

## 28-Pin, 8-Bit Flash Microcontroller

### Processor Features

- Interrupt Capability
- PIC16F570 Operating Speed:
  - DC – 20 MHz Crystal oscillator
  - DC – 200 ns Instruction cycle
- High-Endurance Program and Flash Data Memory Cells:
  - 2048 x 12 user execution memory
  - 64 x 8 self-writable data memory
  - 100,000 write program memory endurance
  - 1,000,000 write Flash data memory endurance
  - Program and Flash data retention: >40 years
- General Purpose Registers (SRAM):
  - 132 x 8 memory
- Only 36 Single-Word Instructions to Learn:
  - Modified baseline CPU
  - Added RETURN and RETFIE instructions
  - Added MOVLB instruction
- All Instructions are Single-Cycle except for Program Branches which are Two-Cycle
- Four-Level Deep Hardware Stack
- Direct, Indirect and Relative Addressing modes for Data and Instructions

### Peripheral Features

- Device Features:
  - 24 I/Os
  - Individual direction control
  - High-current source/sink
- 8-Bit Real-Time Clock/Counter (TMR0) with 8-Bit Programmable Prescaler
- In-Circuit Serial Programming™ (ICSP™) via Two External Pin Connections
- Analog Comparator (CMP):
  - Two analog comparators
  - Absolute and programmable references
- Analog-to-Digital Converter (ADC):
  - 8-bit resolution
  - Eight external input channels
  - 0.6V reference input
- Operational Amplifiers (op amps):
  - Two operational amplifiers
  - Fully-accessible visibility

### eXtreme Low-Power (XLP) Features

- Sleep Mode:
  - 50 nA @ 2.0V, typical
- Watchdog Timer:
  - 500 nA @ 2.0V, typical

### Microcontroller Features

- Brown-out Reset (BOR)
- Power-on Reset (POR)
- Device Reset Timer (DRT)
- Watchdog Timer (WDT) with its own On-Chip RC Oscillator for Reliable Operation
- Programmable Code Protection (CP)
- Power-Saving Sleep mode with Wake-up on Change Feature
- Selectable Oscillator Options:
  - INTOSC: Precision 4 or 8 MHz internal oscillator
  - EXTRC: Low-cost external RC oscillator
  - LP: Power-saving, low-frequency crystal
  - XT: Standard crystal/resonator
  - HS: High-speed crystal/resonator
  - EC: High-speed external clock
- Variety of Packaging Options:
  - 28-Lead SPDIP, SOIC, SSOP, QFN, UQFN

### CMOS Technology

- Low-Power, High-Speed CMOS Flash Technology
- Fully-Static Design
- Wide Operating Voltage and Temperature Range:
  - Industrial: 2.0V to 5.5V
  - Extended: 2.0V to 5.5V
- Operating Current:
  - 175 uA @ 2V, 4 MHz, typical
  - 13 uA @ 2V, 32 kHz, typical
- Standby Current:
  - 100 nA @ 2V, typical

# PIC16F570

**TABLE 1: PIC16F527 AND PIC16F570 FAMILY TYPES**

Device	Data Sheet Index	I/O Pins <sup>(1)</sup>	Flash	Data EE (B)	SRAM (B)	8-Bit ADC Channels	Op Amp	Comparator	8-Bit Timers	BOR	Stack Levels	Interrupts	8 MHz Int. Osc.	Interrupt-on-Change Pins	Weak Pull-up Pins	XLP
PIC16F527	(1)	18	1 KW	64	68	8	2	2	1	Y	4	Y	Y	4	4	Y
PIC16F570	(2)	25	2 KW	64	132	8	2	2	1	Y	4	Y	Y	8	8	Y

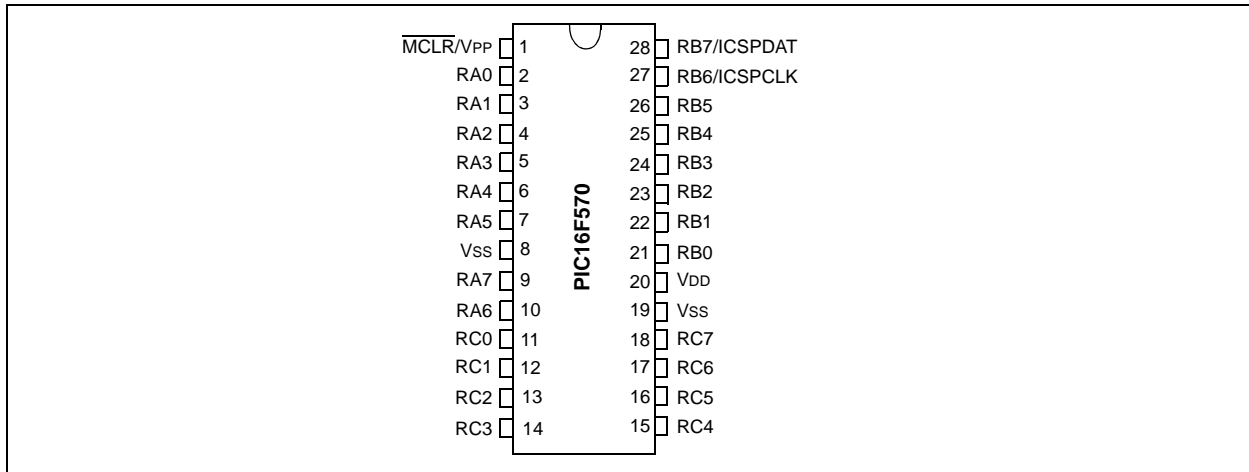
**Note 1:** One pin is input-only.

**Data Sheet Index:** (Unshaded devices are described in this document.)

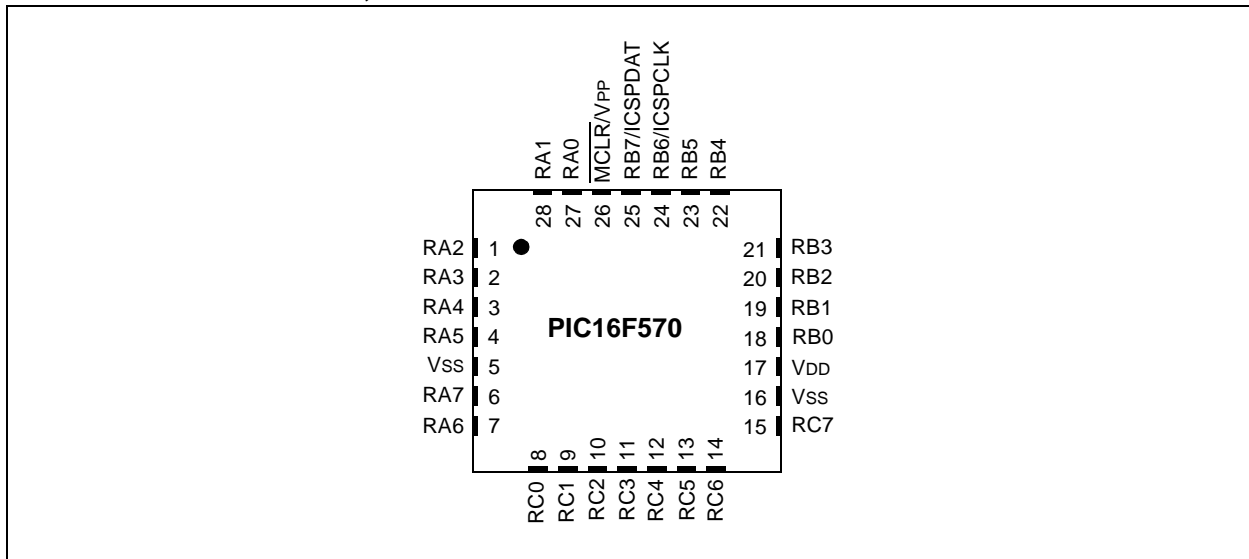
- 1: DS40001652 [PIC16F527 Data Sheet, 20-Pin, 8-bit Flash Microcontroller.](#)
- 2: DS40001684 [PIC16F570 Data Sheet, 28-Pin, 8-bit Flash Microcontroller.](#)

## Pin Diagrams

**FIGURE 1: 28-PIN SPDIP, SSOP, SOIC**



**FIGURE 2: 28-PIN QFN, UQFN**



**TABLE 2: 28-PIN ALLOCATION TABLE**

I/O	28-Pin SPDIP, SSOP, SOIC	28-Pin QFN, UQFN	ADC	Reference	Comparator	Op Amp	Timers	Interrupt-on-Change	Pull-up	Basic
$\overline{\text{MCLR}}$	1	26	—	—	—	—	—	N	Y	$\overline{\text{MCLR}}$ V <sub>PP</sub>
RA0	2	27	AN0	—	—	—	—	N	N	—
RA1	3	28	AN1	—	C1IN+	—	—	N	N	—
RA2	4	1	AN2	CVREF1	—	—	—	N	N	—
RA3	5	2	AN3	—	C2IN+	—	—	N	N	—
RA4	6	3	AN4	—	—	—	T0CKI	N	N	—
RA5	7	4	AN5	—	—	—	—	N	N	—
RA6	10	7	—	—	—	—	—	N	N	OSC2 CLKOUT
RA7	9	6	—	—	—	—	—	N	N	OSC1 CLKIN
RB0	21	18	—	—	—	—	—	Y	Y	—
RB1	22	19	—	—	—	—	—	Y	Y	—
RB2	23	20	—	—	—	—	—	Y	Y	—
RB3	24	21	—	—	C1OUT	—	—	Y	Y	—
RB4	25	22	—	—	C2OUT	—	—	Y	Y	—
RB5	26	23	—	—	—	—	—	Y	Y	—
RB6	27	24	—	—	—	—	—	Y	Y	ICSPCLK
RB7	28	25	—	CVREF2	C1IN-	—	—	Y	Y	ICSPDAT
RC0	11	8	—	—	—	—	—	N	N	—
RC1	12	9	AN6	—	—	OP1	—	N	N	—
RC2	13	10	—	—	—	OP1-	—	N	N	—
RC3	14	11	—	—	—	OP1+	—	N	N	—
RC4	15	12	—	—	—	OP2+	—	N	N	—
RC5	16	13	—	—	—	OP2-	—	N	N	—
RC6	17	14	AN7	—	—	OP2	—	N	N	—
RC7	18	15	—	—	C2IN-	—	—	N	N	—
V <sub>DD</sub>	20	17	—	—	—	—	—	—	—	V <sub>DD</sub>
V <sub>SS</sub>	8, 19	5, 16	—	—	—	—	—	—	—	V <sub>SS</sub>

# PIC16F570

## Table of Contents

1.0	General Description	5
2.0	PIC16F570 Device Varieties	6
3.0	Architectural Overview	7
4.0	Memory Organization	12
5.0	Self-Writable Flash Data Memory Control	23
6.0	I/O Port	27
7.0	Timer0 Module and TMR0 Register	32
8.0	Special Features of the CPU	37
9.0	Analog-to-Digital (A/D) Converter	54
10.0	Comparator(s)	59
11.0	Comparator Voltage Reference Module	64
12.0	Operational Amplifier (OPA) Module	66
13.0	Instruction Set Summary	68
14.0	Development Support	76
15.0	Electrical Characteristics	80
16.0	DC and AC Characteristics Graphs and Charts	97
17.0	Packaging Information	111
	Appendix	125
	The Microchip Website	126
	Customer Change Notification Service	126
	Customer Support	126
	Product Identification System	127

## 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). 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 Website 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., DS30000000A is version A of document DS30000000).

### 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 Website; <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 website at [www.microchip.com](http://www.microchip.com) to receive the most current information on all of our products.

## 1.0 GENERAL DESCRIPTION

The PIC16F570 device from Microchip Technology is a low-cost, high-performance, 8-bit, fully-static, Flash-based CMOS microcontroller. It employs a RISC architecture with only 36 single-word/single-cycle instructions. All instructions are single cycle except for program branches, which take two cycles. The PIC16F570 device 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 a typical 2:1 code compression over other 8-bit microcontrollers in its class. The easy-to-use and easy to remember instruction set reduces development time significantly.

The PIC16F570 product is equipped with special features that reduce system cost and power requirements. The Power-on Reset (POR), Brown-out Reset (BOR) and Device Reset Timer (DRT) eliminate the need for external Reset circuitry. There are several oscillator configurations to choose from, including INTRC Internal Oscillator mode and the power-saving LP (Low-Power) Oscillator mode. Power-Saving Sleep mode, Watchdog Timer and code protection features improve system cost, power and reliability.

The PIC16F570 device is available in the cost-effective Flash programmable version, which is suitable for production in any volume. The customer can take full advantage of Microchip's price leadership in Flash programmable microcontrollers, while benefiting from the Flash programmable flexibility.

The PIC16F570 product is supported by a full-featured macro assembler, a software simulator, an in-circuit emulator, a 'C' compiler, a low-cost development programmer and a full-featured programmer. All the tools are supported on IBM® PC and compatible machines.

## 1.1 Applications

The PIC16F570 device fits in applications ranging from personal care appliances and security systems to low-power remote transmitters/receivers. The Flash technology makes customizing application programs (transmitter codes, appliance settings, receiver frequencies, etc.) extremely fast and convenient. The small footprint packages, for through hole or surface mounting, make these microcontrollers perfect for applications with space limitations. Low cost, low power, high performance, ease of use and I/O flexibility make the PIC16F570 device very versatile, even in areas where no microcontroller use has been considered before (e.g., timer functions, logic and PLDs in larger systems and co-processor applications).

**TABLE 1-1: FEATURES AND MEMORY OF PIC16F570**

Clock	Maximum Frequency of Operation (MHz)	20
Memory	Flash Program Memory	2048
	SRAM Data Memory (bytes)	132
	Flash Data Memory (bytes)	64
Peripherals	Timer Module(s)	TMR0
	Wake-up from Sleep on Pin Change	Yes
Features	I/O Pins	24
	Input Pins	1
	Internal Pull-ups	Yes
	In-Circuit Serial Programming™	Yes
	Number of Instructions	36
	Packages	28-pin SPDIP, SOIC, SSOP, QFN, UQFN
	Interrupts	Yes

# PIC16F570

---

## 2.0 PIC16F570 DEVICE VARIETIES

A variety of packaging options are available. Depending on application and production requirements, the proper device option can be selected using the information in this section. When placing orders, please use the PIC16F570 Product Identification System at the back of this data sheet to specify the correct part number.

### 2.1 Quick Turn Programming (QTP) Devices

Microchip offers a QTP programming service for factory production orders. This service is made available for users who choose not to program medium-to-high quantity units and whose code patterns have stabilized. The devices are identical to the Flash devices but with all Flash locations and fuse options already programmed by the factory. Certain code and prototype verification procedures do apply before production shipments are available. Please contact your local Microchip Technology sales office for more details.

### 2.2 Serialized Quick Turn Programming<sup>SM</sup> (SQTP<sup>SM</sup>) Devices

Microchip offers a unique programming service, where a few user-defined locations in each device are programmed with different serial numbers. The serial numbers may be random, pseudo-random or sequential.

Serial programming allows each device to have a unique number, which can serve as an entry code, password or ID number.

## 3.0 ARCHITECTURAL OVERVIEW

The high performance of the PIC16F570 device can be attributed to a number of architectural features commonly found in RISC microprocessors. To begin with, the PIC16F570 device uses a Harvard architecture in which program and data are accessed on separate buses. This improves bandwidth over traditional von Neumann architectures where program and data are fetched on the same bus. Separating program and data memory further allows instructions to be sized differently than the 8-bit wide data word. Instruction opcodes are 12 bits wide, making it possible to have all single-word instructions. A 12-bit wide program memory access bus fetches a 12-bit instruction in a single cycle. A two-stage pipeline overlaps fetch and execution of instructions. Consequently, all instructions execute in a single cycle (200 ns @ 20 MHz, 1  $\mu$ s @ 4 MHz) except for program branches.

[Table 3-1](#) below lists memory supported by the PIC16F570 device.

**TABLE 3-1: PIC16F570 MEMORY**

Device	Program Memory	Data Memory	
	Flash (words)	SRAM (bytes)	Flash (bytes)
PIC16F570	2048	132	64

The PIC16F570 device can directly or indirectly address its register files and data memory. All Special Function Registers (SFR), including the PC, are mapped in the data memory. The PIC16F570 device 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 PIC16F570 device simple, yet efficient. In addition, the learning curve is reduced significantly.

The PIC16F570 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-bit wide and capable of addition, subtraction, shift and logical operations. Unless otherwise mentioned, arithmetic operations are two's complement in nature. In two-operand instructions, one operand is typically the W (working) register. The other operand is either a file register or an immediate constant. In single operand instructions, the operand is either the W register or a file register.

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

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

A simplified block diagram is shown in [Figure 3-2](#), with the corresponding device pins described in [Table 3-2](#).



**TABLE 3-2: PIC16F570 PINOUT DESCRIPTION**

Name	Function	Input Type	Output Type	Description
VPP/MCLR	VPP	HV	—	Test mode high-voltage pin
	MCLR	ST	—	Master Clear (Reset). This pin is an active-low Reset to the device. Voltage on MCLR/VPP must not exceed VDD during normal device operation or the device will enter Programming mode. Weak pull-up is always on.
RA0/AN0	RA0	TTL	CMOS	Bidirectional I/O pin
	AN0	AN	—	ADC channel input
RA1/AN1/C1IN+	RA1	TTL	CMOS	Bidirectional I/O pin
	AN1	AN	—	ADC channel input
	C1IN+	AN	—	Comparator 1 non-inverting input
RA2/AN2/CVREF1	RA2	TTL	CMOS	Bidirectional I/O port
	AN2	AN	—	ADC channel input
	CVREF1	—	AN	Programmable Voltage Reference output 1
RA3/AN3/C2IN+	RA3	TTL	CMOS	Bidirectional I/O pin
	AN3	AN	—	ADC channel input
	C2IN+	AN	—	Comparator 2 non-inverting input
RA4/AN4/TOCKI	RA4	TTL	CMOS	Bidirectional I/O pin
	AN4	AN	—	ADC channel input
	TOCKI	ST	—	Timer0 Schmitt Trigger input pin
RA5/AN5	RA5	TTL	CMOS	Bidirectional I/O port
	AN5	AN	—	ADC channel input
RA7/CLKIN/OSC1	RA7	TTL	CMOS	Bidirectional I/O port
	CLKIN	ST	—	EXTRC Schmitt Trigger input
	OSC1	XTAL	—	XTAL oscillator input pin
RA6/CLKOUT/OSC2	RA6	TTL	CMOS	Bidirectional I/O port
	CLKOUT	—	CMOS	EXTRC/INTRC CLKOUT pin ( $F_{osc}/4$ )
	OSC2	—	XTAL	Oscillator crystal output. Connections to crystal or resonator in Crystal Oscillator mode (XT, HS and LP modes only, PORTB in other modes).
RC0	RC0	ST	CMOS	Bidirectional I/O port
RC1/AN6/OP1	RC1	ST	CMOS	Bidirectional I/O port
	AN6	AN	—	ADC channel input
	OP1	—	AN	Op amp 1 output
RC2/OP1-	RC2	ST	CMOS	Bidirectional I/O port
	OP1-	AN	—	Op amp 1 inverting input
RC3/OP1+	RC3	ST	CMOS	Bidirectional I/O port
	OP1+	AN	—	Op amp 1 non-inverting input
RC4/OP2+	RC4	ST	CMOS	Bidirectional I/O port
	OP2+	AN	—	Op amp 2 non-inverting input
RC5/OP2-	RC5	ST	CMOS	Bidirectional I/O port
	OP2-	AN	—	Op amp 2 inverting input
RC6/AN7/OP2	RC6	ST	CMOS	Bidirectional I/O port
	AN7	AN	—	ADC channel input
	OP2	—	AN	Op amp 2 output

**Legend:** I = Input, O = Output, I/O = Input/Output, P = Power, — = Not Used, TTL = TTL input, ST = Schmitt Trigger input, HV = High Voltage, AN = Analog Voltage

# PIC16F570

**TABLE 3-2: PIC16F570 PINOUT DESCRIPTION (CONTINUED)**

Name	Function	Input Type	Output Type	Description
RC7/C2IN-	RC7	ST	CMOS	Bidirectional I/O port
	C2IN-	AN	—	Comparator 2 inverting input
RB0	RB0	TTL	CMOS	Bidirectional I/O pin. It can be software programmed for internal weak pull-up and wake-up from Sleep on pin change.
RB1	RB1	TTL	CMOS	Bidirectional I/O pin. It can be software programmed for internal weak pull-up and wake-up from Sleep on pin change.
RB2	RB2	TTL	CMOS	Bidirectional I/O pin. It can be software programmed for internal weak pull-up and wake-up from Sleep on pin change.
RB3/C1OUT	RB3	TTL	CMOS	Bidirectional I/O pin. It can be software programmed for internal weak pull-up and wake-up from Sleep on pin change.
	C1OUT	—	CMOS	Comparator 1 output
RB4/C2OUT	RB4	TTL	CMOS	Bidirectional I/O pin. It can be software programmed for internal weak pull-up and wake-up from Sleep on pin change.
	C2OUT	—	CMOS	Comparator 2 output
RB5	RB5	TTL	CMOS	Bidirectional I/O pin. It can be software programmed for internal weak pull-up and wake-up from Sleep on pin change.
RB6/ICSPCLK	RB6	TTL	CMOS	Bidirectional I/O pin. It can be software programmed for internal weak pull-up and wake-up from Sleep on pin change.
	ICSPCLK	ST	—	ICSP™ mode Schmitt Trigger
RB7/ICSPDAT/C1IN-/CVREF2	RB7	TTL	CMOS	Bidirectional I/O pin. It can be software programmed for internal weak pull-up and wake-up from Sleep on pin change.
	ICSPDAT	ST	CMOS	ICSP™ mode Schmitt Trigger
	C1IN-	AN	—	Comparator 1 inverting input
	CVREF2	—	AN	Programmable Voltage Reference output 2
VDD	VDD	—	P	Positive supply for logic and I/O pins
VSS	VSS	—	P	Ground reference for logic and I/O pins

**Legend:** I = Input, O = Output, I/O = Input/Output, P = Power, — = Not Used, TTL = TTL input, ST = Schmitt Trigger input, HV = High Voltage, AN = Analog Voltage

## 3.1 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 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 is shown in Figure 3-2 and Example 3-1.

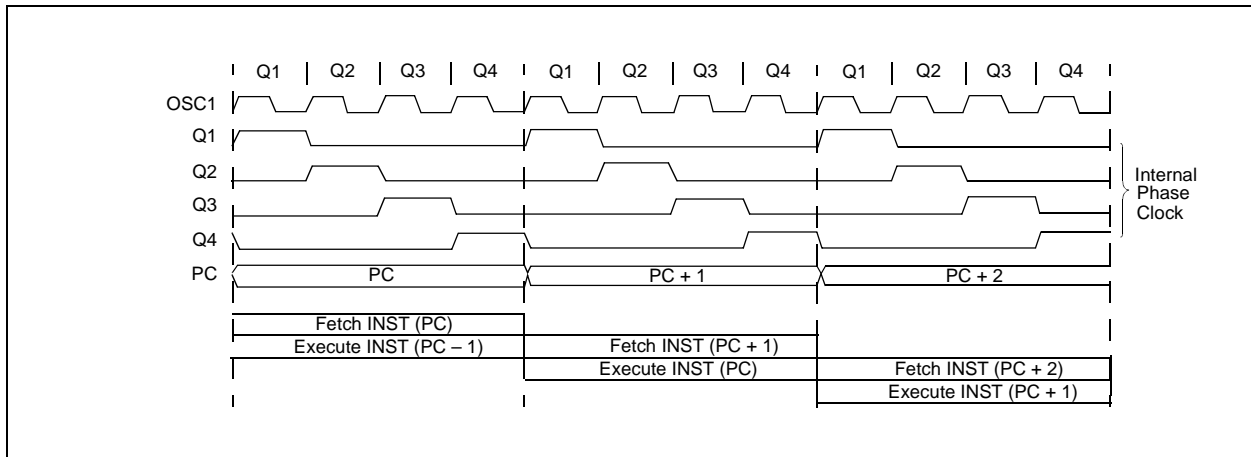
## 3.2 Instruction Flow/Pipelining

An instruction cycle consists of four Q cycles (Q1, Q2, Q3 and Q4). The instruction fetch and execute are pipelined such that fetch takes one instruction cycle, while decode and execute take another instruction cycle. However, due to the pipelining, each instruction effectively executes in one cycle. If an instruction causes the PC to change (e.g., GOTO or an interrupt), then two cycles are required to complete the instruction (Example 3-1).

A fetch cycle begins with the PC incrementing in Q1.

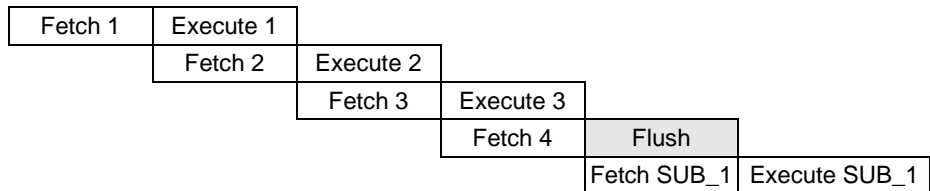
In the execution cycle, the fetched instruction is latched into the Instruction Register (IR) in cycle Q1. This instruction is then decoded and executed during the Q2, Q3 and Q4 cycles. Data memory is read during Q2 (operand read) and written during Q4 (destination write).

**FIGURE 3-2: CLOCK/INSTRUCTION CYCLE**



**EXAMPLE 3-1: INSTRUCTION PIPELINE FLOW**

1. MOVLW 03H
2. MOVWF PORTB
3. CALL SUB\_1
4. BSF PORTB, BIT1



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.

# PIC16F570

## 4.0 MEMORY ORGANIZATION

The PIC16F570 memories are organized into program memory and data memory (SRAM). The self-writable portion of the program memory called self-writable Flash data memory is located at addresses 800h-83Fh. All program mode commands that work on the normal Flash memory, work on the Flash data memory. This includes bulk erase, row/column/cycling toggles, Load and Read data commands (Refer to [Section 5.0 “Self-Writable Flash Data Memory Control”](#) for more details). For devices with more than 512 bytes of program memory, a paging scheme is used. Program memory pages are accessed using one STATUS register bit. For the PIC16F570, with data memory register files of more than 32 registers, a banking scheme is used. Data memory banks are accessed directly using the Bank Select Register (BSR).

### 4.1 Program Memory Organization for PIC16F570

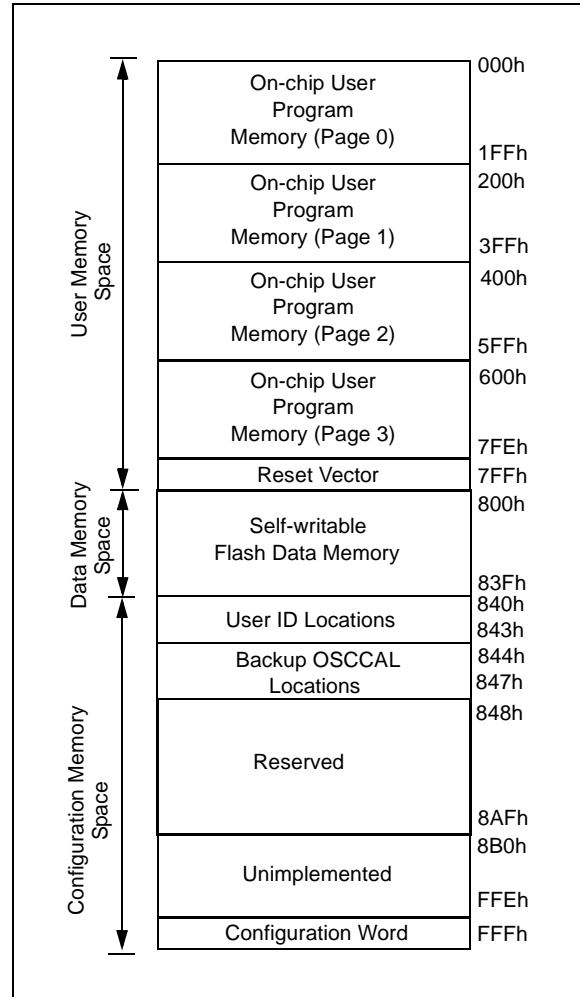
The PIC16F570 device has an 11-bit Program Counter (PC) capable of addressing a 2K x 12 program memory space. Program memory is partitioned into user memory, data memory and configuration memory spaces.

The user memory space is the on-chip user program memory. As shown in [Figure 4-1](#), it extends from 0x000 to 0x7FF and partitions into pages, including an Interrupt vector at address 0x004 and a Reset vector at address 0x7FF.

The Configuration memory space extends from 0x840 to 0xFFFF. Locations from 0x848 through 0x8AF are reserved. The user ID locations extend from 0x840 through 0x843. The Backup OSCCAL locations extend from 0x844 through 0x847. The Configuration Word is physically located at 0xFFFF.

Refer to “*PIC16F570 Memory Programming Specification*” (DS41670) for more details.

FIGURE 4-1: MEMORY MAP



## 4.2 Data Memory (SRAM and SFRs)

Data memory is composed of registers or bytes of SRAM. 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 are registers used by the CPU and peripheral functions for controlling desired operations of the PIC16F570. See [Section 4.3 “STATUS Register”](#) for details.

### 4.2.1 GENERAL PURPOSE REGISTER FILE

The General Purpose Register file is accessed directly or indirectly. See [Section 4.8 “Direct and Indirect Addressing”](#).

### 4.2.2 SPECIAL FUNCTION REGISTERS

The Special Function Registers (SFRs) are registers used by the CPU and peripheral functions to control the operation of the device ([Section 4.3 “STATUS Register”](#)).

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.

**FIGURE 4-2: PIC16F570 REGISTER FILE MAP**

BSR<2:0>	000	001	010	011	100	101	110	111
File Address	20h	40h	60h	80h	A0h	C0h	E0h	
00h	INDF <sup>(1)</sup>	INDF <sup>(1)</sup>	INDF <sup>(1)</sup>	INDF <sup>(1)</sup>	INDF <sup>(1)</sup>	INDF <sup>(1)</sup>	INDF <sup>(1)</sup>	INDF <sup>(1)</sup>
01h	TMR0	EECON	TMR0	IW	TMR0	EECON	TMR0	IW
02h	PCL	PCL	PCL	PCL	PCL	PCL	PCL	PCL
03h	STATUS	STATUS	STATUS	STATUS	STATUS	STATUS	STATUS	STATUS
04h	FSR	FSR	FSR	FSR	FSR	FSR	FSR	FSR
05h	OSCCAL	EEDATA	OSCCAL	INTCON1	OSCCAL	EEDATA	OSCCAL	INTCON1
06h	PORTA	EEADR	PORTA	ISTATUS	PORTA	EEADR	PORTA	ISTATUS
07h	PORTB	CM1CON0	PORTB	IFSR	PORTB	CM1CON0	PORTB	IFSR
08h	PORTC	CM2CON0	PORTC	IBSR	PORTC	CM2CON0	PORTC	IBSR
09h	ADCON0	VRCON	ADCON0	OPACON	ADCON0	VRCON	ADCON0	OPACON
0Ah	ADRES	ANSEL	ADRES	ANSEL	ADRES	ANSEL	ADRES	ANSEL
0Bh	INTCON0	INTCON0	INTCON0	INTCON0	INTCON0	INTCON0	INTCON0	INTCON0
0Ch	General Purpose Registers Addresses map back to addresses in Bank 0.							
0Fh	2Fh	4Fh	6Fh	8Fh	AFh	CFh	EFh	
10h	30h	50h	70h	90h	B0h	D0h	F0h	
	General Purpose Registers	General Purpose Registers	General Purpose Registers	General Purpose Registers	General Purpose Registers	General Purpose Registers	General Purpose Registers	General Purpose Registers
1Fh	3Fh	5Fh	7Fh	9Fh	BFh	DFh	FFh	
	Bank 0	Bank 1	Bank 2	Bank 3	Bank 4	Bank 5	Bank 6	Bank 7

**Note 1:** Not a physical register. See [Section 4.8 “Direct and Indirect Addressing”](#).

**TABLE 4-1: SPECIAL FUNCTION REGISTER SUMMARY**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR/BOR	Value on all other Resets
<b>Bank 0/4</b>											
N/A	W <sup>(2)</sup>	Working Register (W)								xxxx xxxx	xxxx xxxx
N/A	TRIS	I/O Control Registers (TRISA, TRISB, TRISC)								1111 1111	1111 1111
N/A	OPTION	Contains control bits to configure Timer0 and Timer0/WDT prescaler								1111 1111	1111 1111
N/A	BSR <sup>(2)</sup>	—	—	—	—	—	BSR2	BSR1	BSR0	---- -000	---- -uuu
00h	INDF	Uses contents of FSR to address data memory (not a physical register)								xxxx xxxx	uuuu uuuu
01h	TMR0	Timer0 module Register								xxxx xxxx	uuuu uuuu
02h	PCL <sup>(1)</sup>	Low-order eight bits of PC								1111 1111	1111 1111
03h	STATUS <sup>(2)</sup>	PA2	PA1	PA0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	000q qqqq
04h	FSR <sup>(2)</sup>	Indirect data memory Address Pointer								xxxx xxxx	uuuu uuuu
05h	OSCCAL	CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0	—	1111 111-	uuuu uuu-
06h	PORTA	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0	xxxx xxxx	uuuu uuuu
07h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx	uuuu uuuu
08h	PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	xxxx xxxx	uuuu uuuu
09h	ADCON0	ADCS1	ADCS0	CHS3	CHS2	CHS1	CHS0	GO/DONE	ADON	1111 1100	1111 1100
0Ah	ADRES	ADC Conversion Result								xxxx xxxx	uuuu uuuu
0Bh	INTCON0	ADIF	CWIF	T0IF	RBIF	—	—	—	GIE	0000 ---0	0000 ---0
<b>Bank 1/5</b>											
N/A	W <sup>(2)</sup>	Working Register (W)								xxxx xxxx	xxxx xxxx
N/A	TRIS	I/O Control Registers (TRISA, TRISB, TRISC)								1111 1111	1111 1111
N/A	OPTION	Contains control bits to configure Timer0 and Timer0/WDT prescaler								1111 1111	1111 1111
N/A	BSR <sup>(2)</sup>	—	—	—	—	—	BSR2	BSR1	BSR0	---- -000	---- -uuu
20h	INDF	Uses contents of FSR to address data memory (not a physical register)								xxxx xxxx	uuuu uuuu
21h	EECON	—	—	—	FREE	WRERR	WREN	WR	RD	---0 0000	---0 0000
22h	PCL <sup>(1)</sup>	Low-order eight bits of PC								1111 1111	1111 1111
23h	STATUS <sup>(2)</sup>	PA2	PA1	PA0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	000q qqqq
24h	FSR <sup>(2)</sup>	Indirect data memory Address Pointer								xxxx xxxx	uuuu uuuu
25h	EEDATA	Self Read/Write Data								xxxx xxxx	uuuu uuuu
26h	EEADR	—	—	Self Read/Write Address						--xx xxxx	--uu uuuu
27h	CM1CON0	C1OUT	$\overline{C1OUTEN}$	C1POL	$\overline{C1T0CS}$	C1ON	C1NREF	C1PREF	$\overline{C1WU}$	1111 1111	quuu uuuu
28h	CM2CON0	C2OUT	$\overline{C2OUTEN}$	C2POL	C2PREF2	C2ON	C2NREF	C2PREF1	$\overline{C2WU}$	1111 1111	quuu uuuu
29h	VRCON	VREN	VROE1	VROE2	VRR	VR3	VR2	VR1	VR0	0001 1111	uuuu uuuu
2Ah	ANSEL	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0	1111 1111	1111 1111
2Bh	INTCON0	ADIF	CWIF	T0IF	RBIF	—	—	—	GIE	0000 ---0	0000 ---0

**Legend:** x = unknown, u = unchanged, — = unimplemented, read as '0' (if applicable), q = value depends on condition.

Shaded cells = unimplemented or unused

- Note 1:** The upper byte of the Program Counter is not directly accessible. See [Section 4.6 “Program Counter”](#) for an explanation of how to access these bits.
- 2:** Registers are implemented as two physical registers. When executing from within an ISR, a secondary register is used at the same logical location. Both registers are persistent. See [Section 8.11 “Interrupts”](#).
- 3:** These registers show the contents of the registers in the other context: ISR or main line code. See [Section 8.11 “Interrupts”](#).

# PIC16F570

**TABLE 4-1: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR/BOR	Value on all other Resets
<b>Bank 2/6</b>											
N/A	W <sup>(2)</sup>	Working Register (W)								xxxx xxxx	xxxx xxxx
N/A	TRIS	I/O Control Registers (TRISA, TRISB, TRISC)								1111 1111	1111 1111
N/A	OPTION	Contains control bits to configure Timer0 and Timer0/WDT prescaler								1111 1111	1111 1111
N/A	BSR <sup>(2)</sup>	—	—	—	—	—	BSR2	BSR1	BSR0	---- -000	---- -uuu
40h	INDF	Uses contents of FSR to address data memory (not a physical register)								xxxx xxxx	uuuu uuuu
41h	TMR0	Timer0 module Register								xxxx xxxx	uuuu uuuu
42h	PCL <sup>(1)</sup>	Low-order eight bits of PC								1111 1111	1111 1111
43h	STATUS <sup>(2)</sup>	PA2	PA1	PA0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	000q qqqq
44h	FSR <sup>(2)</sup>	Indirect data memory Address Pointer								xxxx xxxx	uuuu uuuu
45h	OSCCAL	CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0	—	1111 111-	uuuu uu-
46h	PORTA	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0	xxxx xxxx	uuuu uuuu
47h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx	uuuu uuuu
48h	PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	xxxx xxxx	uuuu uuuu
49h	ADCON0	ADCS1	ADCS0	CHS3	CHS2	CHS1	CHS0	GO/DONE	ADON	1111 1100	1111 1100
4Ah	ADRES	ADC Conversion Result								xxxx xxxx	uuuu uuuu
4Bh	INTCON0	ADIF	CWIF	TOIF	RBIF	—	—	—	GIE	0000 ---0	0000 ---0
<b>Bank 3/7</b>											
N/A	W <sup>(2)</sup>	Working Register (W)								xxxx xxxx	xxxx xxxx
N/A	TRIS	I/O Control Registers (TRISA, TRISB, TRISC)								1111 1111	1111 1111
N/A	OPTION	Contains control bits to configure Timer0 and Timer0/WDT prescaler								1111 1111	1111 1111
N/A	BSR <sup>(2)</sup>	—	—	—	—	—	BSR2	BSR1	BSR0	---- -000	---- -0uu
60h	INDF	Uses contents of FSR to address data memory (not a physical register)								xxxx xxxx	uuuu uuuu
61h	IW <sup>(3)</sup>	Interrupt Working Register. (Addressed also as W register when within ISR)								xxxx xxxx	xxxx xxxx
62h	PCL <sup>(1)</sup>	Low-order eight bits of PC								1111 1111	1111 1111
63h	STATUS <sup>(2)</sup>	PA2	PA1	PA0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	000q qqqq
64h	FSR <sup>(2)</sup>	Indirect data memory Address Pointer								xxxx xxxx	uuuu uuuu
65h	INTCON1	ADIE	CWIE	TOIE	RBIE	—	—	—	WUR	0000 ---0	0000 ---0
66h	ISTATUS <sup>(3)</sup>	PA2	PA1	PA0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	xxxx xxxx	000q qqqq
67h	IFSR <sup>(3)</sup>	Indirect data memory Address Pointer								xxxx xxxx	uuuu uuuu
68h	IBSR <sup>(3)</sup>	—	—	—	—	—	BSR2	BSR1	BSR0	---- -xxx	---- -uuu
69h	OPACON	—	—	—	—	—	—	OPA2ON	OPA1ON	---- --00	---- --00
6Ah	ANSEL	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0	1111 1111	1111 1111
6Bh	INTCON0	ADIF	CWIF	TOIF	RBIF	—	—	—	GIE	0000 ---0	0000 ---0

**Legend:** x = unknown, u = unchanged, — = unimplemented, read as '0' (if applicable), q = value depends on condition.  
Shaded cells = unimplemented or unused

- Note 1:** The upper byte of the Program Counter is not directly accessible. See [Section 4.6 “Program Counter”](#) for an explanation of how to access these bits.
- 2:** Registers are implemented as two physical registers. When executing from within an ISR, a secondary register is used at the same logical location. Both registers are persistent. See [Section 8.11 “Interrupts”](#).
- 3:** These registers show the contents of the registers in the other context: ISR or main line code. See [Section 8.11 “Interrupts”](#).

## 4.3 STATUS Register

This register contains the arithmetic status of the ALU, the Reset status and the page preselect bit.

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

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

Therefore, it is recommended that only `BCF`, `BSF` and `MOVWF` instructions be used to alter the STATUS register. These instructions do not affect the Z, DC or C bits from the STATUS register. For other instructions which do affect Status bits, see [Section 13.0 “Instruction Set Summary”](#).

**REGISTER 4-1: STATUS: STATUS REGISTER**

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

**Legend:**

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

- bit 7-5      **PA<2:0>**: Program Page Preselect bits  
x00 = Page 0 (000h-1FFh)  
x01 = Page 1 (200h-3FFh)  
x10 = Page 2 (400h-5FFh)  
x11 = Page 3 (600h-7FFh)
- 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 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/borrow bit (for `ADDWF`, `SUBWF` and `RRF`, `RLF` instructions)  
**ADDWF**: **SUBWF**: **RRF** or **RLF**:  
1 = A carry occurred    1 = A borrow did not occur; Load bit with LSb or MSb, respectively  
0 = A carry did not occur    0 = A borrow occurred

# PIC16F570

## 4.4 OPTION Register

The OPTION register is an 8-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 <7:0> bits.

**Note:** If TRIS bit is set to '0', the wake-up on change and pull-up functions are disabled for that pin (i.e., note that TRIS overrides Option control of  $\overline{\text{RBPU}}$  and  $\overline{\text{RBWU}}$ ).

**REGISTER 4-2: OPTION: OPTION REGISTER**

W-1	W-1	W-1	W-1	W-1	W-1	W-1	W-1
$\overline{\text{RBWU}}^{(2)}$	$\overline{\text{RBPU}}$	T0CS <sup>(1)</sup>	T0SE	PSA	PS2	PS1	PS0
bit 7							bit 0

**Legend:**

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

bit 7  **$\overline{\text{RBWU}}$** : Enable PORTB Interrupt Flag on Pin Change bit<sup>(2)</sup>  
 1 = Disabled  
 0 = Enabled

bit 6  **$\overline{\text{RBPU}}$** : Enable PORTB Weak Pull-Ups bit  
 1 = Disabled  
 0 = Enabled

bit 5 **T0CS**: Timer0 Clock Source Select bit<sup>(1)</sup>  
 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

- Note 1:** If the T0CS bit is set to '1', it will override the TRIS function on the T0CKI pin.  
**Note 2:** The  $\overline{\text{RBWU}}$  bit of the OPTION register must be cleared to enable the RBIF function in the INTCON0 register.

## 4.5 OSCCAL Register

The Oscillator Calibration (OSCCAL) register is used to calibrate the 8 MHz internal oscillator macro. It contains seven bits of calibration that uses a two's complement scheme for controlling the oscillator speed. See [Register 4-3](#) for details.

**REGISTER 4-3: OSCCAL: OSCILLATOR CALIBRATION REGISTER**

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

**Legend:**

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

bit 7-1      **CAL<6:0>**: Oscillator Calibration bits  
               0111111 = Maximum frequency  
               •  
               •  
               •  
               0000001  
               0000000 = Center frequency  
               1111111  
               •  
               •  
               •  
               1000000 = Minimum frequency

bit 0      **Unimplemented:** Read as '0'

# PIC16F570

## 4.6 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>. Bits 5 and 6 of the STATUS register provide page information to bits 9 and 10 of the PC (Figure 4-3).

For a `CALL` instruction, or any instruction where the PCL is the destination, bits <7:0> of the PC again are provided by the instruction word. However, PC<8> does not come from the instruction word, but is always cleared (Figure 4-3).

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

**Note:** Because bit 8 of the PC 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).

### 4.6.1 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 oscillator calibration instruction). After executing `MOVLW XX`, the PC will roll over to location 00h and begin executing user code.

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

Therefore, upon a Reset, a `GOTO` instruction will automatically cause the program to jump to page 0 until the value of the page bits is altered.

## 4.7 Stack

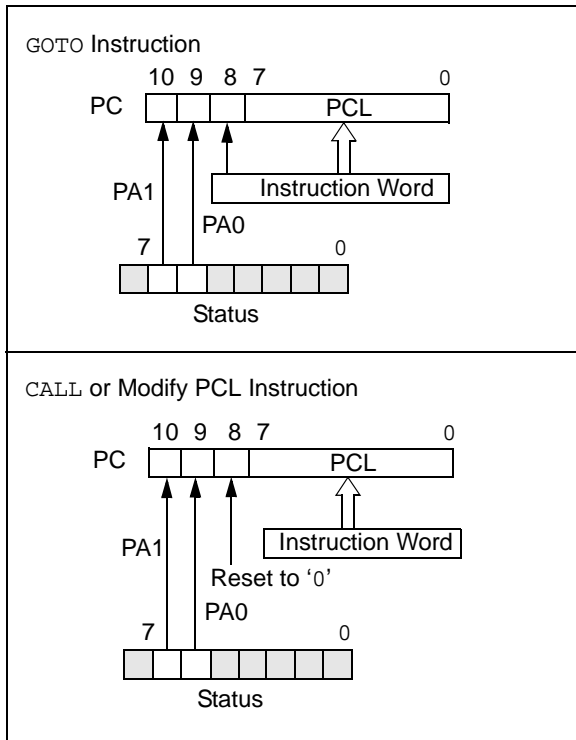
The PIC16F570 device has a 4-deep, 12-bit wide hardware PUSH/POP stack.

A `CALL` instruction or an interrupt will PUSH the current PC value, incremented by one, into Stack Level 1. If there was a previous value in the Stack 1 location, it will be pushed into the Stack 2 location. This process will be continued throughout the remaining stack locations populated with values. If more than four sequential `CALL`s are executed, only the most recent four return addresses are stored.

A `RETLW`, `RETURN` or `RETFIE` instruction will POP the contents of Stack Level 1 into the PC. If there was a previous value in the Stack 2 location, it will be copied into the Stack Level 1 location. This process will be continued throughout the remaining stack locations populated with values. If more than four sequential `RETLW`s are executed, the stack will be filled with the address previously stored in Stack Level 4. Note that 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.

**Note 1:** There are no Status bits to indicate Stack Overflows or Stack Underflow conditions.  
**2:** There are no instruction mnemonics called `PUSH` or `POP`. These are actions that occur from the execution of the `CALL`, `RETURN`, `RETFIE` and `RETLW` instructions.

**FIGURE 4-3: LOADING OF PC BRANCH INSTRUCTIONS**



## 4.8 Direct and Indirect Addressing

### 4.8.1 DIRECT DATA ADDRESSING: BSR REGISTER

Traditional data memory addressing is performed in the Direct Addressing mode. In Direct Addressing, the Bank Select Register bits BSR<2:0>, in the new BSR register, are used to select the data memory bank. The address location within that bank comes directly from the opcode being executed.

BSR<2:0> 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, etc.).

A new instruction supports the addition of the BSR register, called the `MOVLB` instruction. See [Section 13.0 “Instruction Set Summary”](#) for more information.

### 4.8.2 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.

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).

The FSR is an 8-bit wide register. It is used in conjunction with the INDF register to indirectly address the data memory area.

The FSR<7:0> bits are used to select data memory addresses 00h to 1Fh.

A simple program to clear RAM locations 10h-1Fh using indirect addressing is shown in [Example 4-1](#).

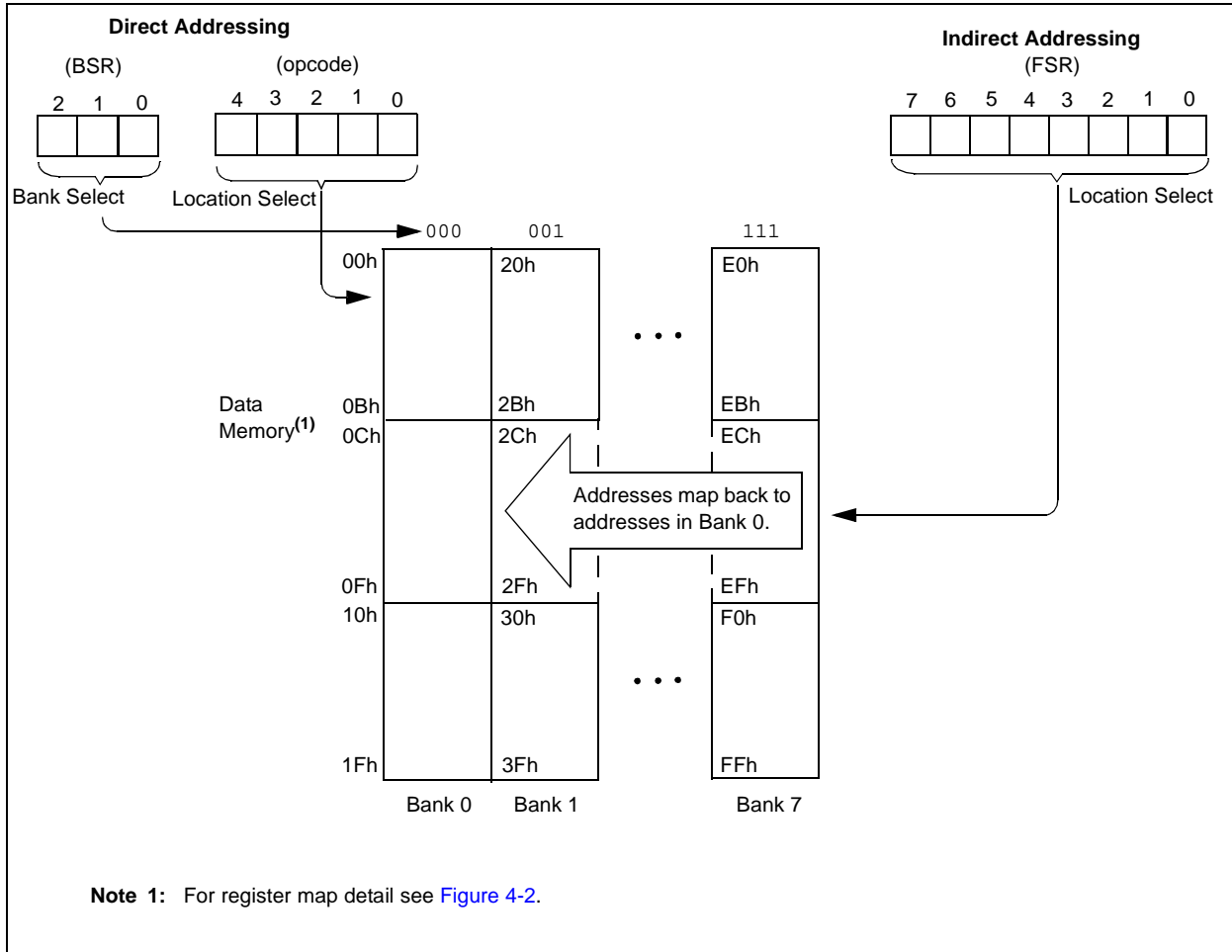
#### EXAMPLE 4-1: HOW TO CLEAR RAM USING INDIRECT ADDRESSING

```

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

# PIC16F570

**FIGURE 4-4: DIRECT/INDIRECT ADDRESSING**



## 5.0 SELF-WRITABLE FLASH DATA MEMORY CONTROL

Flash data memory consists of 64 bytes of self-writable memory and supports a self-write capability that can write four bytes of memory at one time. Data to be written to the self-writable data memory is first written into four write latches before writing the data to Flash memory.

Although each Flash data memory location is 12 bits wide, access is limited to the lower eight bits. The upper four bits will automatically default to '1' in any self-write procedure. The lower eight bits are fully readable and writable during normal operation and throughout the full VDD range.

The self-writable Flash data memory is not directly mapped in the register file space. Instead, it is indirectly addressed through the Special Function Registers, EECON, EEDATA and EEADR.

### 5.1 Reading Flash Data Memory

To read a Flash data memory location the user must:

- Write the EEADR register
- Set the RD bit of the EECON register

The value written to the EEADR register determines which Flash data memory location is read. Setting the RD bit of the EECON register initiates the read. Data from the Flash data memory read is available in the EEDATA register immediately. The EEDATA register will hold this value until another read is initiated or it is modified by a write operation. Program execution is suspended while the read cycle is in progress. Execution will continue with the instruction following the one that sets the WR bit. See [Example 5-1](#) for sample code.

#### EXAMPLE 5-1: READING FROM FLASH DATA MEMORY

```

MOVLB    0x01          ; Switch to Bank 1
MOVF     DATA_EE_ADDR,W;
MOVWF    EEADR         ; Data Memory
                    ; Address to read
BSF      EECON, RD     ; EE Read
MOVWF    EEDATA, W    ; W = EEDATA
    
```

**Note:** Only a BSF command will work to enable the Flash data memory read documented in [Example 5-1](#). No other sequence of commands will work, no exceptions.

**Note 1:** To prevent accidental corruption of the Flash data memory, an unlock sequence is required to initiate a write or erase cycle. This sequence requires that the bit set instructions used to configure the EECON register happen exactly as shown in [Example 5-2](#) and [Example 5-3](#), depending on the operation requested.

**2:** In order to prevent any disruptions of self-writes or row erases performed on the self-writable Flash data memory, interrupts should be disabled prior to executing those routines.

#### 5.1.1 ERASING FLASH DATA MEMORY

A row must be manually erased before writing new data. The following sequence must be performed for a single row erase.

1. Load EEADR with an address in the row to be erased.
2. Set the FREE bit to enable the erase.
3. Set the WREN bit to enable write access to the array.
4. Disable interrupts.
5. Set the WR bit to initiate the erase cycle.

If the WREN bit is not set in the instruction cycle after the FREE bit is set, the FREE bit will be cleared in hardware.

If the WR bit is not set in the instruction cycle after the WREN bit is set, the WREN bit will be cleared in hardware.

Sample code that follows this procedure is included in [Example 5-2](#).

Program execution is suspended while the erase cycle is in progress. Execution will continue with the instruction following the one that sets the WR bit.

#### EXAMPLE 5-2: ERASING A FLASH DATA MEMORY ROW

```

MOVLB    0x01          ; Switch to Bank 1
MOVLW    EE_ADR_ERASE ; LOAD ADDRESS OF ROW TO
                    ; ERASE
MOVWF    EEADR         ;
BSF      EECON,FREE   ; SELECT ERASE
BSF      EECON,WREN   ; ENABLE WRITES
BSF      EECON,WR     ; INITIATE ERASE
    
```

# PIC16F570

**Note 1:** The FREE bit may be set by any command normally used by the core. However, the WREN and WR bits can only be set using a series of BSF commands, as documented in [Example 5-1](#). No other sequence of commands will work, no exceptions.

**2:** Bits <5:3> of the EEADR register indicate which row is to be erased.

## 5.1.2 WRITING TO FLASH DATA MEMORY

Once a cell is erased, new data can be written. Program execution is suspended during the write cycle.

The self-write operation writes four bytes of data at one time. The data must first be loaded into four write latches. Once the write latches are loaded, the data will be written to Flash data memory.

The self-write sequence is shown below.

The following self-write sequence must be performed for four bytes to be written.

1. Load EEADR with the address.
2. Load EEDATA with the data to be written.
3. Set the WREN bit to enable write access to the array.
4. Disable interrupts.
5. Set the WR bit to load the data into the write latch.
6. Steps 1 through 5 are repeated three more times to load the remaining write latches.

On the fourth and final loop, the EEADR register will contain an address in the format of b'00xxxx11.

When the WR bit is set for the final time, the processor will recognize that this is the last write latch to be loaded, and will automatically load the write latch and then, immediately perform the Flash data memory write of all four bytes.

The specific sequence of setting the WREN bit and setting the WR bit must be executed to properly initiate each load of the write latches and the write to Flash data memory.

If the WR bit is not set in the instruction cycle after the WREN bit is set, the WREN bit will be cleared in hardware.

Sample code that follows this procedure is included in [Example 5-3](#).

## EXAMPLE 5-3: WRITING TO FLASH DATA MEMORY

```
MOVLB 0x01           ;SWITCH TO BANK 1
MOVLW 0x04           ;LOAD 4 DATA BYTES
MOVWF LoopCount      ;WRITE LOOP
WRITE_LOOP          ;VARIABLE STORED
  MOVLW EE_ADR_WRITE ;LOAD ADDRESS TO
                    ;WRITE
  MOVWF EEADR        ;INTO EEADR
                    ;REGISTER
  MOVLW EE_DATA_TO_WRITE ;LOAD DATA TO
  MOVWF EEDATA       ;INTO EEDATA
                    ;REGISTER
  BSF  EECON,WREN    ;ENABLE WRITES
  BSF  EECON,WR      ;LOAD WRITE LATCH
  BTFSC LoopCount    ;TEST IF 4th BYTE
  GOTO WRITE_LOOP    ;
                    ;WRITE IS DONE
                    ;
```

**Note 1:** Only a series of BSF commands will work to enable the memory write sequence documented in [Example 5-3](#). No other sequence of commands will work, no exceptions.

**2:** For reads, erases and writes to the Flash data memory, there is no need to insert a NOP into the user code as is done on mid-range devices. The instruction immediately following the "BSF EECON,WR/RD" will be fetched and executed properly.

## 5.2 Write/Verify

Depending on the application, good programming practice may dictate that data written to the Flash data memory be verified. [Example 5-4](#) is an example of a write/verify.

## EXAMPLE 5-4: WRITE/VERIFY OF FLASH DATA MEMORY

```
MOVWF EEDATA, W      ;EEDATA has not changed
                    ;from previous write
BSF  EECON, RD       ;Read the value written
XORWF EEDATA, W      ;
BTFSS STATUS, Z      ;Is data the same
GOTO  WRITE_ERR      ;No, handle error
                    ;Yes, continue
```

## 5.3 Register Definitions — Memory Control

### REGISTER 5-1: EEDATA: FLASH DATA REGISTER

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
EEDATA7	EEDATA6	EEDATA5	EEDATA4	EEDATA3	EEDATA2	EEDATA1	EEDATA0
bit 7							bit 0

**Legend:**

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

bit 7-0                      **EEDATA<7:0>**: Eight bits of data to be read from/written to data Flash

### REGISTER 5-2: EEADR: FLASH ADDRESS REGISTER

U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	EEADR5	EEADR4	EEADR3	EEADR2	EEADR1	EEADR0
bit 7							bit 0

**Legend:**

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

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

bit 5-0                      **EEADR<5:0>**: Six bits of data to be read from/written to data Flash

# PIC16F570

## REGISTER 5-3: EECON: FLASH CONTROL REGISTER

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	FREE	WRERR	WREN	WR	RD
bit 7							bit 0

### Legend:

S = Bit can only be set

R = Readable bit

-n = Value at POR

W = Writable bit

'1' = Bit is set

U = Unimplemented bit, read as '0'

'0' = Bit is cleared

x = Bit is unknown

bit 7-5 **Unimplemented:** Read as '0'.

bit 4 **FREE:** Flash Data Memory Row Erase Enable bit

1 = Program memory row being pointed to by EEADR will be erased on the next write cycle. No write will be performed. This bit is cleared at the completion of the erase operation.

0 = Perform write only

bit 3 **WRERR:** Write Error Flag bit

1 = A write operation terminated prematurely (by device Reset)

0 = Write operation completed successfully

bit 2 **WREN:** Write Enable bit

1 = Allows write cycle to Flash data memory

0 = Inhibits write cycle to Flash data memory

bit 1 **WR:** Write Control bit

1 = Initiate a erase or write cycle

0 = Write/Erase cycle is complete

bit 0 **RD:** Read Control bit

1 = Initiate a read of Flash data memory

0 = Do not read Flash data memory

## 5.4 Code Protection

Code protection does not prevent the CPU from performing read or write operations on the Flash data memory. Refer to the code protection chapter for more information.

## 6.0 I/O PORT

As with any other register, the I/O register(s) can be written and read under program control. However, read instructions (e.g., `MOVF 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 are all set.

### 6.1 PORTA

PORTA is an 8-bit I/O register. The Configuration Word can set several I/Os to alternate functions. When acting as alternate functions, the pins will read as '0' during a port read.

### 6.2 PORTB

PORTB is an 8-bit I/O register. The PORTB pins can be configured with weak pull-ups and also for wake-up on change.

### 6.3 PORTC

PORTC is an 8-bit I/O register.

### 6.4 TRIS Register

The Output Driver Control register is loaded with the contents of the W register by executing the `TRIS` instruction. A '1' from a TRIS register bit puts the corresponding output driver in a High-Impedance mode. A '0' puts the contents of the output data latch on the selected pins, enabling the output buffer. The only exception is the T0CKI pin, which may be controlled by the OPTION register (see [Register 4-2](#)).

TRIS registers are "write-only". Active bits in these registers are set (output drivers disabled) upon Reset.



## 6.6 Register Definitions — PORT Control

**REGISTER 6-1: PORTA: PORTA REGISTER**

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0
bit 7							bit 0

**Legend:**

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

bit 7-0                      **RA<7:0>**: PORTA I/O Pin bits  
 1 = Port pin is >VIH min.  
 0 = Port pin is <VIL max.

**TABLE 6-1: PORTA PINS ORDER OF PRECEDENCE**

Priority	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0
1	OSC1	OSC2	AN5	AN4	C2IN+	CVREF1	C1IN+	AN0
2	CLKIN	CLKOUT	TRISA5	T0CKI	AN3	AN2	AN1	TRISA0
3	TRISA7	TRISA6	—	TRISA4	TRISA3	TRISA2	TRISA1	—

**REGISTER 6-2: PORTB: PORTB REGISTER**

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
bit 7							bit 0

**Legend:**

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

bit 7-0                      **RB<7:0>**: PORTB I/O Pin bits  
 1 = Port pin is >VIH min.  
 0 = Port pin is <VIL max.

**TABLE 6-2: PORTB PINS ORDER OF PRECEDENCE**

Priority	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
1	ICSPDAT	ICSPCLK	TRISB5	C2OUT	C1OUT	TRISB2	TRISB1	TRISB0
2	CVREF2	TRISB6	—	TRISB4	TRISB3	—	—	—
3	C1IN-	—	—	—	—	—	—	—
4	TRISB7	—	—	—	—	—	—	—

# PIC16F570

**REGISTER 6-3: PORTC: PORTC REGISTER**

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0
bit 7							bit 0

**Legend:**

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

bit 7-0                      **RC<7:0>**: PORTC I/O Pin bits  
 1 = Port pin is >VIH min.  
 0 = Port pin is <VIL max.

**TABLE 6-3: PORTC PINS ORDER OF PRECEDENCE**

Priority	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0
1	C2IN-	OP2	OP2-	OP2+	OP1+	OP1-	OP1	TRISC0
2	TRISC7	AN7	TRISC5	TRISC4	TRISC3	TRISC2	AN6	—
3	—	TRISC6	—	—	—	—	TRISC1	—

**REGISTER 6-4: ANSEL REGISTER**

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0
bit 7							bit 0

**Legend:**

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

bit 7-0                      **ANS<7:0>**: ADC Analog Input Pin Select bits<sup>(1,2)</sup>  
 0 = Analog function on selected ANx pin is disabled  
 1 = ANx configured as an analog input

**Note 1:** When the ANSx bits are set, the channels selected will automatically be forced into Analog mode, regardless of the pin function previously defined, and the digital output drivers and input buffers will be also disabled. Exceptions exist when there is more than one analog function active on the ANx pin. It is the user's responsibility to ensure that the ADC loading on the other analog functions does not affect their application.

**2:** The ANS<7:0> bits are active regardless of the condition of ADON.

**TABLE 6-4: REGISTERS ASSOCIATED WITH THE I/O PORTS**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-On Reset	Value on MCLR and WDT Reset
N/A	TRIS	I/O Control Registers (TRISA, TRISB, TRISC)								1111 1111	1111 1111
06h	PORTA	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0	xxxx xxxx	uuuu uuuu
07h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx	uuuu uuuu
08h	PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	xxxx xxxx	uuuu uuuu

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

## 6.7 I/O Programming Considerations

### 6.7.1 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 rewrite 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 (e.g., PIC16F570)

```

;Initial PORTB Settings
;PORTB<5:3> Inputs
;PORTB<2:0> Outputs
;
; PORTB latch PORTB pins
; -----
BCF    PORTB, 5    ;--01 -ppp--11 pppp
BCF    PORTB, 4    ;--10 -ppp--11 pppp
MOVLW  007h       ;
TRIS   PORTB      ;--10 -ppp--11 pppp
;

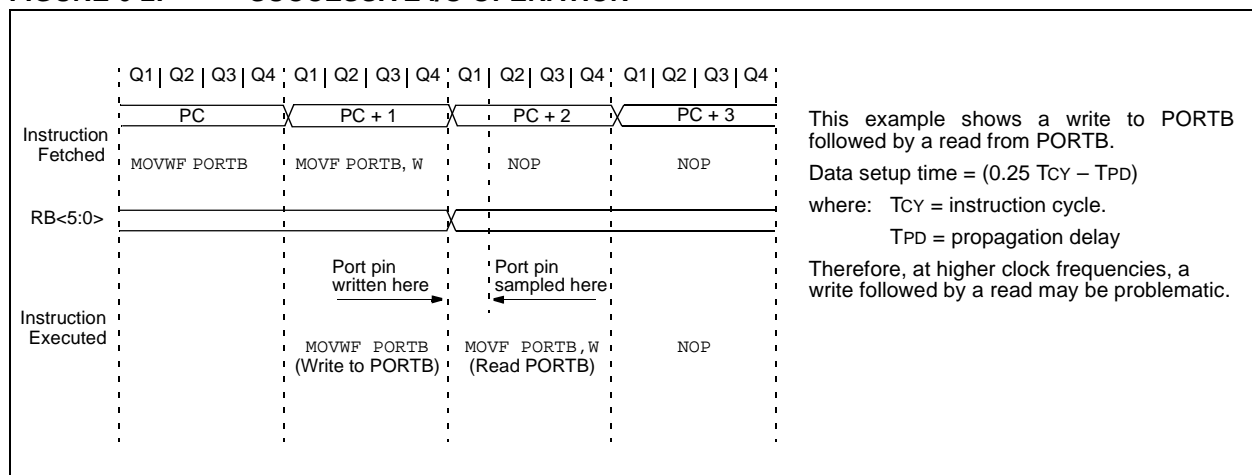
```

**Note 1:** The user may have expected the pin values to be ‘--00 pppp’. The 2nd BCF caused RB5 to be latched as the pin value (High).

### 6.7.2 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 causes that file to be read into the CPU. Otherwise, the previous state of that pin may be read into the CPU rather than the new state. When in doubt, it is better to separate these instructions with a NOP or another instruction not accessing this I/O port.

**FIGURE 6-2: SUCCESSIVE I/O OPERATION**



# PIC16F570

## 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 of the OPTION register. 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.

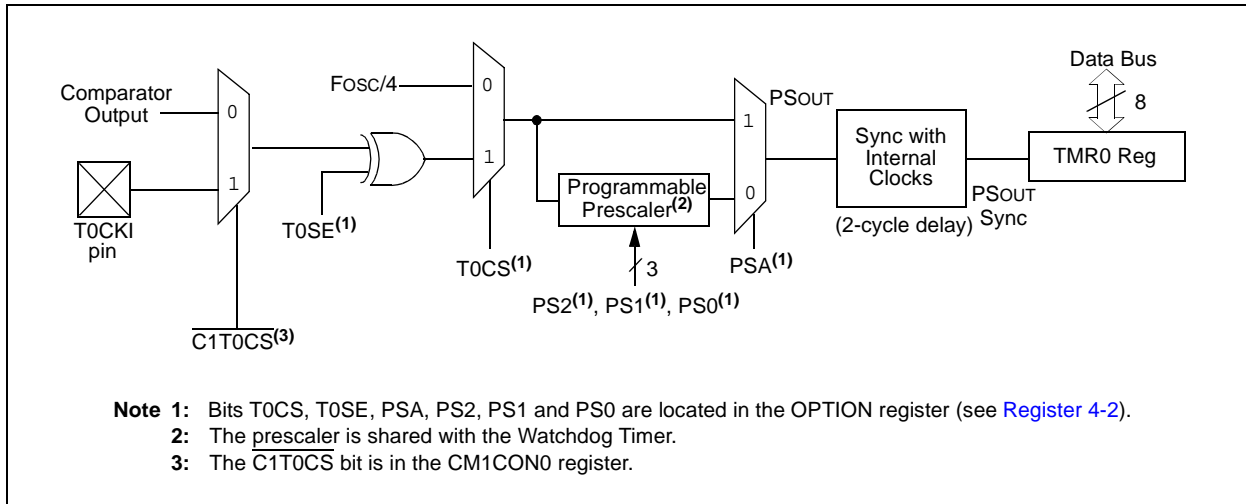
There are two types of Counter mode. The first Counter mode uses the T0CKI pin to increment Timer0. It is selected by setting the T0CS bit of the OPTION register, setting the C1T0CS bit of the CM1CON0 register and setting the C1OUTEN bit of the CM1CON0 register. In this mode, Timer0 will increment either on every rising or falling edge of pin T0CKI. The T0SE bit of the OPTION register determines the source edge. 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”.

The second Counter mode uses the output of the comparator to increment Timer0. It can be entered in by setting the T0CS bit of the OPTION register, and clearing the C1T0CS bit of the CM1CON0 register (C1OUTEN [CM1CON0<6>] does not affect this mode of operation). This enables an internal connection between the comparator and the Timer0.

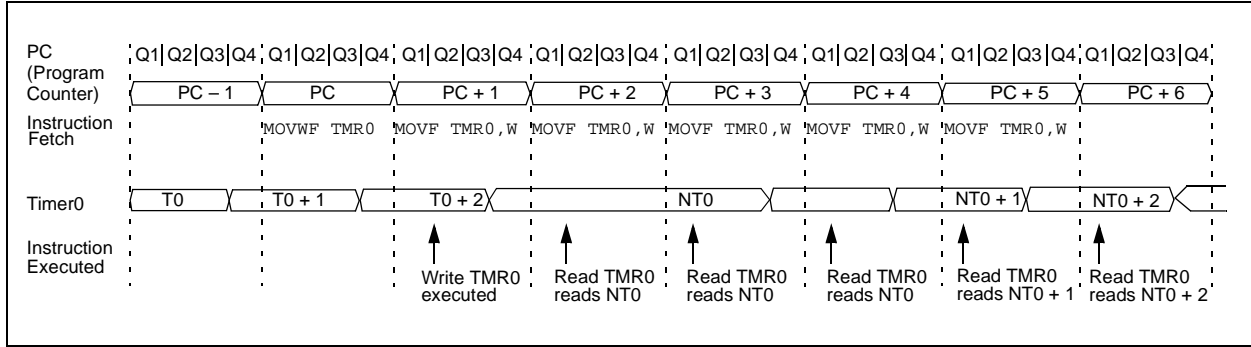
The prescaler may be used by either the Timer0 module or the Watchdog Timer, but not both. The prescaler assignment is controlled in software by the control bit, PSA of the OPTION register. Clearing the PSA bit will assign the prescaler to Timer0. The prescaler is not readable or writable. When the prescaler 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.

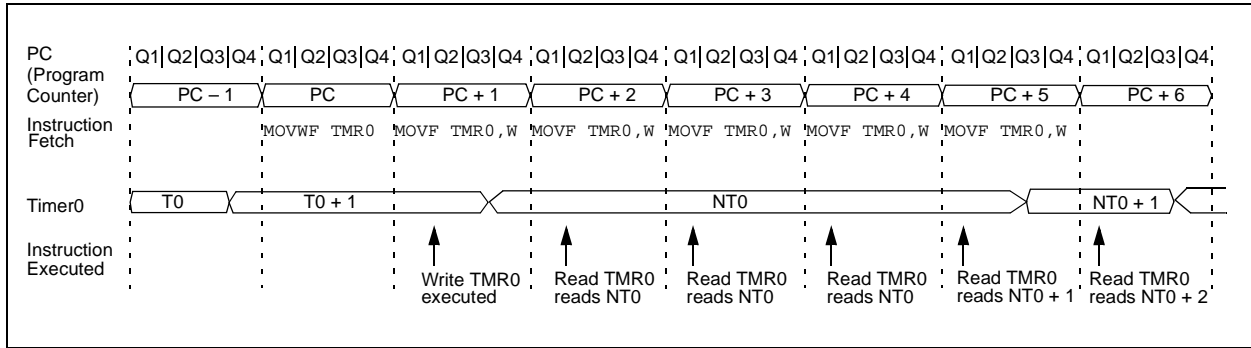
FIGURE 7-1: TIMER0 BLOCK DIAGRAM



**FIGURE 7-2: TIMER0 TIMING: INTERNAL CLOCK/NO PRESCALE**



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



**TABLE 7-1: REGISTERS ASSOCIATED WITH TIMER0**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
TMR0	Timer0 module Register								—
CM1CON0	C1OUT	$\overline{C1OUTEN}$	C1POL	$\overline{C1T0CS}$	C1ON	C1NREF	C1PREF	$\overline{C1WU}$	62
CM2CON0	C2OUT	$\overline{C2OUTEN}$	C2POL	C2PREF2	C2ON	C2NREF	C2PREF1	$\overline{C2WU}$	63
OPTION	$\overline{RBWU}$	$\overline{RBPU}$	T0CS	T0SE	PSA	PS2	PS1	PS0	18
TRIS <sup>(1)</sup>	I/O Control Registers (TRISA, TRISB, TRISC)								—

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

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

# PIC16F570

## 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

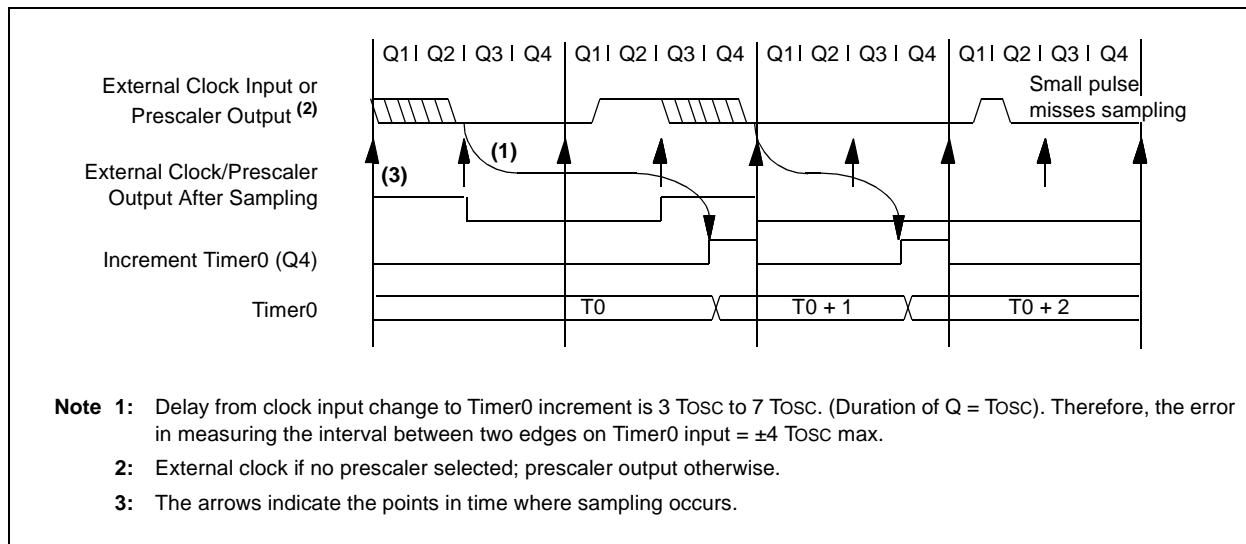
When no prescaler is used, the external clock input is the same as the prescaler output. The synchronization of T0CKI with the internal phase clocks is accomplished by sampling the prescaler output on the Q2 and Q4 cycles of the internal phase clocks (Figure 7-4). Therefore, it is necessary for T0CKI to be high for at least 2 Tosc (and a small RC delay of 2 Tt0H) and low for at least 2 Tosc (and a small RC delay of 2 Tt0H). 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 4 Tt0H) divided by the prescaler value. The only requirement on T0CKI high and low time is that they do not violate the minimum pulse width requirement of Tt0H. Refer to parameters 40, 41 and 42 in the electrical specification of the desired device.

### 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



## 7.2 Prescaler

An 8-bit counter is available as a prescaler for the Timer0 module or as a postscale for the Watchdog Timer (WDT), respectively (see [Section 8.7 “Watchdog Timer \(WDT\)”](#)). For simplicity, this counter is being referred to as “prescaler” throughout this data sheet.

**Note:** The prescaler may be used by either the Timer0 module or the WDT, but not both. Thus, a prescaler assignment for the Timer0 module means that there is no prescaler for the WDT and vice versa.

The PSA and PS<2:0> bits of the OPTION register determine prescaler assignment and prescale ratio.

When assigned to the Timer0 module, all instructions writing to the TMR0 register (e.g., CLRWF TMR0, MOVWF TMR0, 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
CLRWF   TMR0    ;Clear TMR0 & Prescaler
MOVLW  b'00xx1111'
CLRWDT          ;PS<2:0> are 000 or 001
MOVLW  b'00xx1xxx' ;Set Postscale to
OPTION          ;desired WDT rate
```

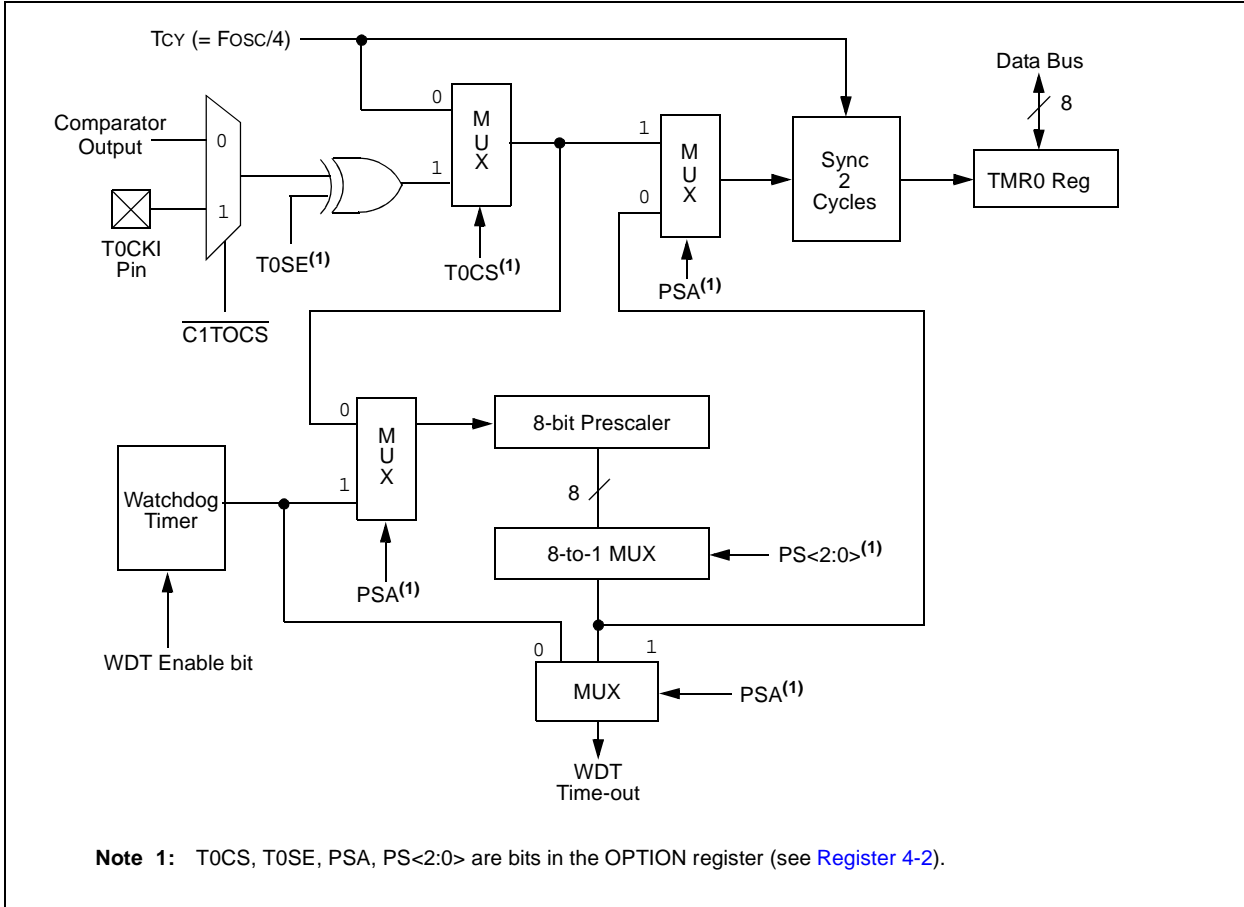
To change the prescaler from the WDT to the Timer0 module, use the sequence shown in [Example 7-2](#). This sequence must be used even if the WDT is disabled. A CLRWDT instruction should be executed before switching the prescaler.

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

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

# PIC16F570

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



## 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 PIC16F570 microcontrollers have a host of such features intended to maximize system reliability, minimize cost through elimination of external components, provide power-saving operating modes and offer code protection. These features are:

- Oscillator Selection
- Reset:
  - Power-on Reset (POR)
  - Brown-out Reset (BOR)
  - Device Reset Timer (DRT)
  - Wake-up from Sleep on Pin Change
- Interrupts
- Automatic Context Saving
- Watchdog Timer (WDT)
- Sleep
- Code Protection
- ID Locations
- In-Circuit Serial Programming™
- Clock Out

The device has a Watchdog Timer, which can be shut off only through Configuration bit WDTE. The Watchdog Timer runs off of its own RC oscillator for added reliability.

There is also a Device Reset Timer (DRT), intended to keep the chip in Reset until the crystal oscillator is stable. The DRT can be enabled with the DRTEN Configuration bit. For the HS, XT or LP oscillator options, the 18 ms (nominal) delay is always provided by the Device Reset Timer and the DRTEN bit is ignored. When using the EC clock, INTRC or EXTRC oscillator options, there is a standard delay of 10 us on power-up, which can be extended to 18 ms with the use of the DRT timer. With the DRT timer on-chip, most applications require no additional external Reset circuitry.

The Sleep mode is designed to offer a very low current Power-Down mode. The user can wake-up from Sleep through a change on input pin or through a Watchdog Timer time-out. Several oscillator options are also made available to allow the part to fit the application, including an internal 4/8 MHz oscillator. The EXTRC 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

The PIC16F570 Configuration Words consist of 12 bits, although some bits may be unimplemented and read as '1'. Configuration bits can be programmed to select various device configurations (see [Register 8-1](#)).

# PIC16F570

## 8.2 Register Definitions — Configuration Word

### REGISTER 8-1: CONFIG: CONFIGURATION WORD REGISTER<sup>(1)</sup>

U-1	U-1	R/P-1	R/P-1	R/P-1	R/P-1	U-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
—	—	DRTEN	BOREN	$\overline{\text{CPSW}}$	IOSCF5	—	$\overline{\text{CP}}$	WDTE	FOSC2	FOSC1	FOSC0
bit 11											bit 0

#### Legend:

R = Readable bit  
'0' = Bit is cleared

P = Programmable bit  
'1' = Bit is set

U = Unimplemented bit, read as '1'  
-n = Value when blank or after Bulk Erase

bit 11-10 **Unimplemented:** Read as '1'

bit 9 **DRTEN:** Device Reset Timer Enable bit  
1 = DRT Enabled (18 ms)  
0 = DRT Disabled

bit 8 **BOREN:** Brown-out Reset Enable bit  
1 = BOR Enabled  
0 = BOR Disabled

bit 7 **CPSW:** Code Protection bit – Self-Writable Memory  
1 = Code protection off  
0 = Code protection on

bit 6 **IOSCF5:** Internal Oscillator Frequency Select bit  
1 = 8 MHz INTOSC speed  
0 = 4 MHz INTOSC speed

bit 5 **Unimplemented:** Read as '1'

bit 4 **CP:** Code Protection bit – User Program Memory  
1 = Code protection off  
0 = Code protection on

bit 3 **WDTE:** Watchdog Timer Enable bit  
1 = WDT enabled  
0 = WDT disabled

bit 2-0 **FOSC<2:0>:** Oscillator Selection bits  
000 = LP oscillator and automatic 18 ms DRT (DRTEN fuse ignored)  
001 = XT oscillator and automatic 18 ms DRT (DRTEN fuse ignored)  
010 = HS oscillator and automatic 18 ms DRT (DRTEN fuse ignored)  
011 = EC oscillator with I/O function on OSC2/CLKOUT and 10 us start-up time<sup>(2,3)</sup>  
100 = INTRC with I/O function on OSC2/CLKOUT and 10 us start-up time<sup>(2,3)</sup>  
101 = INTRC with CLKOUT function on OSC2/CLKOUT and 10 us start-up time<sup>(2,3)</sup>  
110 = EXTRC with I/O function on OSC2/CLKOUT and 10 us start-up time<sup>(2,3)</sup>  
111 = EXTRC with CLKOUT function on OSC2/CLKOUT and 10 us start-up time<sup>(2,3)</sup>

**Note 1:** Refer to the “PIC16F570 Memory Programming Specification” (DS41670), to determine how to access the Configuration Word.

**2:** DRT length and start-up time are functions of the Clock mode selection. It is the responsibility of the application designer to ensure the use of either will result in acceptable operation. Refer to [Section 15.0 “Electrical Characteristics”](#) for VDD rise time and stability requirements for this mode of operation.

**3:** The optional DRDEN fuse can be used to extend the start-up time to 18 ms.

## 8.3 Oscillator Configurations

### 8.3.1 OSCILLATOR TYPES

The PIC16F570 device can be operated in up to six different oscillator modes. The user can program up to three Configuration bits (FOSC<2:0>). To select one of these modes:

- LP: Low-Power Crystal
- XT: Crystal/Resonator
- HS: High-Speed Crystal/Resonator
- INTRC: Internal 4/8 MHz Oscillator
- EXTRC: External Resistor/Capacitor
- EC: External High-Speed Clock Input

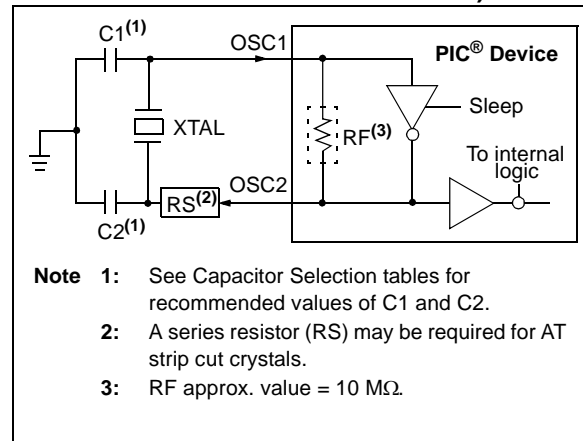
### 8.3.2 CRYSTAL OSCILLATOR/CERAMIC RESONATORS

In HS, XT or LP modes, a crystal or ceramic resonator is connected to the OSC1/CLKIN and OSC2/CLKOUT pins to establish oscillation (Figure 8-1). The PIC16F570 oscillator designs require the use of a parallel cut crystal. Use of a series cut crystal may give a frequency out of the crystal manufacturers specifications. When in HS, XT or LP modes, the device can have an external clock source drive the OSC1/CLKIN pin (Figure 8-2). In this mode, the output drive levels on the OSC2 pin are very weak. If the part is used in this fashion, then this pin should be left open and unloaded. Also when using this mode, the external clock should observe the frequency limits for the Clock mode chosen (HS, XT or LP).

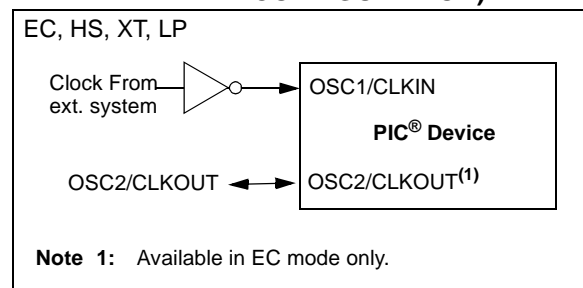
**Note 1:** This device has been designed to perform to the parameters of its data sheet. It has been tested to an electrical specification designed to determine its conformance with these parameters. Due to process differences in the manufacture of this device, this device may have different performance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

**FIGURE 8-1: CRYSTAL OPERATION (OR CERAMIC RESONATOR) (HS, XT OR LP OSC CONFIGURATION)**



**FIGURE 8-2: EXTERNAL CLOCK INPUT OPERATION (HS, XT, LP OR EC OSC CONFIGURATION)**



**TABLE 8-1: CAPACITOR SELECTION FOR CERAMIC RESONATORS**

Osc. Type	Resonator Freq.	Cap. Range C1	Cap. Range C2
XT	4.0 MHz	30 pF	30 pF
HS	16 MHz	10-47 pF	10-47 pF

- Note 1:** 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.

# PIC16F570

**TABLE 8-2: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR<sup>(2)</sup>**

Osc. Type	Resonator Freq.	Cap. Range C1	Cap. Range C2
LP	32 kHz <sup>(1)</sup>	15 pF	15 pF
XT	200 kHz	47-68 pF	47-68 pF
	1 MHz	15 pF	15 pF
	4 MHz	15 pF	15 pF
HS	20 MHz	15-47 pF	15-47 pF

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

**2:** These values are for design guidance only. Rs may be required 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.

### 8.3.3 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 8-3 shows implementation of a parallel resonant oscillator circuit. The circuit is designed to use the fundamental frequency of the crystal. The 74AS04 inverter performs the 180-degree 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 8-3: EXTERNAL PARALLEL RESONANT CRYSTAL OSCILLATOR CIRCUIT**

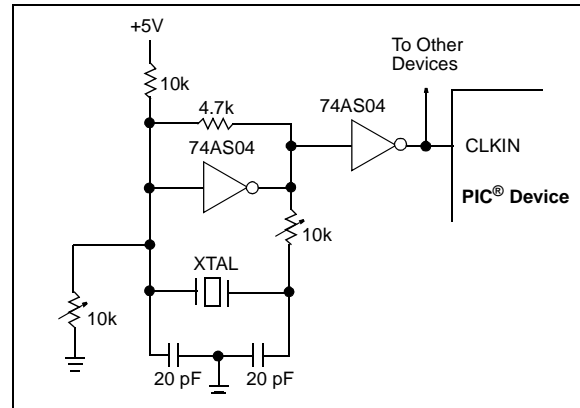
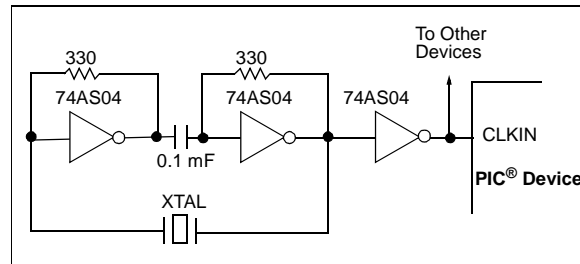


Figure 8-4 shows a series resonant oscillator circuit. This circuit is also designed to use the fundamental frequency of the crystal. The inverter performs a 180-degree phase shift in a series resonant oscillator circuit. The 330 Ω resistors provide the negative feedback to bias the inverters in their linear region.

**FIGURE 8-4: EXTERNAL SERIES RESONANT CRYSTAL OSCILLATOR CIRCUIT**



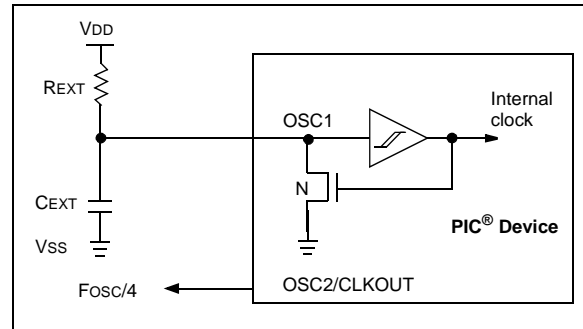
### 8.3.4 EXTERNAL RC OSCILLATOR

For timing insensitive applications, the RC device option offers additional cost savings. The RC oscillator frequency is a function of the supply voltage, the resistor ( $R_{EXT}$ ) and capacitor ( $C_{EXT}$ ) values, and the operating temperature. In addition to this, the oscillator frequency will vary from unit-to-unit due to normal process parameter variation. Furthermore, the difference in lead frame capacitance between package types will also affect the oscillation frequency, especially for low  $C_{EXT}$  values. The user also needs to take into account variation due to tolerance of external R and C components used.

Figure 8-5 shows how the R/C combination is connected to the PIC16F570 device. For  $R_{EXT}$  values below  $3.0\text{ k}\Omega$ , the oscillator operation may become unstable or stop completely. For very high  $R_{EXT}$  values (e.g.,  $1\text{ M}\Omega$ ), the oscillator becomes sensitive to noise, humidity and leakage. Thus, we recommend keeping  $R_{EXT}$  between  $5.0\text{ k}\Omega$  and  $100\text{ k}\Omega$ .

Although the oscillator will operate with no external capacitor ( $C_{EXT} = 0\text{ pF}$ ), we recommend using values above  $20\text{ pF}$  for noise and stability reasons. With no external capacitance or with values below  $20\text{ pF}$ , the oscillation frequency can vary dramatically due to changes in external capacitances, such as PCB trace capacitance or package lead frame capacitance.

**FIGURE 8-5: EXTERNAL RC OSCILLATOR MODE**



### 8.3.5 INTERNAL 4/8 MHz RC OSCILLATOR

The internal RC oscillator provides a fixed 4/8 MHz (nominal) system clock at  $V_{DD} = 5\text{ V}$  and  $25^\circ\text{C}$ , (see [Section 15.0 “Electrical Characteristics”](#) for information on variation overvoltage and temperature).

In addition, a calibration instruction is programmed into the last address of memory, which contains the calibration value for the internal RC oscillator. This location is always non-code protected, regardless of the code-protect settings. This value is programmed as a `MOVLW XX` instruction where `XX` is the calibration value, and is placed at the Reset vector. This will load the W register with the calibration value upon Reset and the PC will then roll over to the users program at address `0x000`. The user then has the option of writing the value to the `OSCCAL` register or ignoring it.

`OSCCAL`, when written to with the calibration value, will “trim” the internal oscillator to remove process variation from the oscillator frequency.

**Note:** Erasing the device will also erase the pre-programmed internal calibration value for the internal oscillator. The calibration value must be read prior to erasing the part so it can be reprogrammed correctly later.

For the PIC16F570 device, only bits `<7:1>` of `OSCCAL` are used for calibration. See [Register 4-3](#) for more information.

**Note:** The bit 0 of the `OSCCAL` register is unimplemented and should be written as ‘0’ when modifying `OSCCAL` for compatibility with future devices.

# PIC16F570

## 8.4 Reset

The device differentiates between various kinds of Reset:

- Power-on Reset (POR)
- Brown-out Reset (BOR)
- $\overline{\text{MCLR}}$  Reset during normal operation
- $\overline{\text{MCLR}}$  Reset during Sleep
- WDT Time-out Reset during normal operation
- WDT Time-out Reset during Sleep
- Wake-up from Sleep on pin change

Some registers are not reset in any way, they are unknown on POR/BOR and unchanged in any other Reset. Most other registers are reset to “Reset state” on Power-on Reset (POR)/Brown-out Reset (BOR),  $\overline{\text{MCLR}}$ , WDT or Wake-up on pin change Reset during normal operation. They are not affected by a WDT Reset during Sleep or  $\overline{\text{MCLR}}$  Reset during Sleep, since these Resets are viewed as resumption of normal operation. The exceptions to this are the  $\overline{\text{TO}}$  and  $\overline{\text{PD}}$  bits. They are set or cleared differently in different Reset situations. These bits are used in software to determine the nature of Reset. See [Table 4-1](#) for a full description of Reset states of all registers.

**TABLE 8-3: RESET CONDITION FOR SPECIAL REGISTERS**

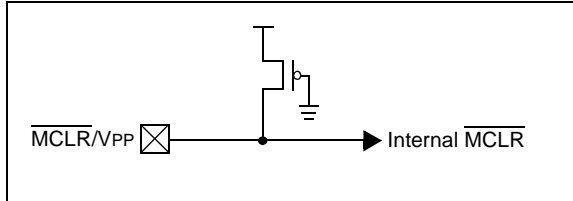
	STATUS Addr: 03h
Power-on Reset (POR) or Brown-out Reset (BOR)	0001 1xxx
$\overline{\text{MCLR}}$ Reset during normal operation	000u uuuu
$\overline{\text{MCLR}}$ Reset during Sleep	0001 0uuu
WDT Reset during Sleep	0000 0uuu
WDT Reset normal operation	0000 uuuu
Wake-up from Sleep on pin change	1001 0uuu
Wake-up from Sleep on comparator change	0101 0uuu

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

## 8.4.1 $\overline{\text{MCLR}}$ ENABLE

This Master Clear ( $\overline{\text{MCLR}}$ ) feature and its associated pull-up are always enabled on this device. The pin is assigned to be an input-only pin function.

**FIGURE 8-6:  $\overline{\text{MCLR}}$  INPUT PIN**



## 8.5 Power-on Reset (POR)

The PIC16F570 device incorporates an on-chip Power-on Reset (POR) circuitry, which provides an internal chip Reset for most power-up situations.

The on-chip POR circuit holds the chip in Reset until  $V_{DD}$  has reached a high enough level for proper operation. To take advantage of the internal POR, an internal weak pull-up resistor is implemented using a transistor (refer to [Table 15-11](#) for the pull-up resistor ranges). This will eliminate external RC components usually needed to create a Power-on Reset. A maximum rise time for  $V_{DD}$  is specified. See [Section 15.0 “Electrical Characteristics”](#) for details.

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

A simplified block diagram of the on-chip Power-on Reset circuit is shown in [Figure 8-7](#).

The Power-on Reset circuit and the Device Reset Timer (see [Section 8.6 “Device Reset Timer \(DRT\)”](#)) circuit are closely related. On Power-up, the Reset latch is set and the DRT is reset. The DRT timer begins counting once it detects  $\overline{\text{MCLR}}$  to be high. After the time-out period, it will reset the Reset latch and thus end the on-chip Reset signal.

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

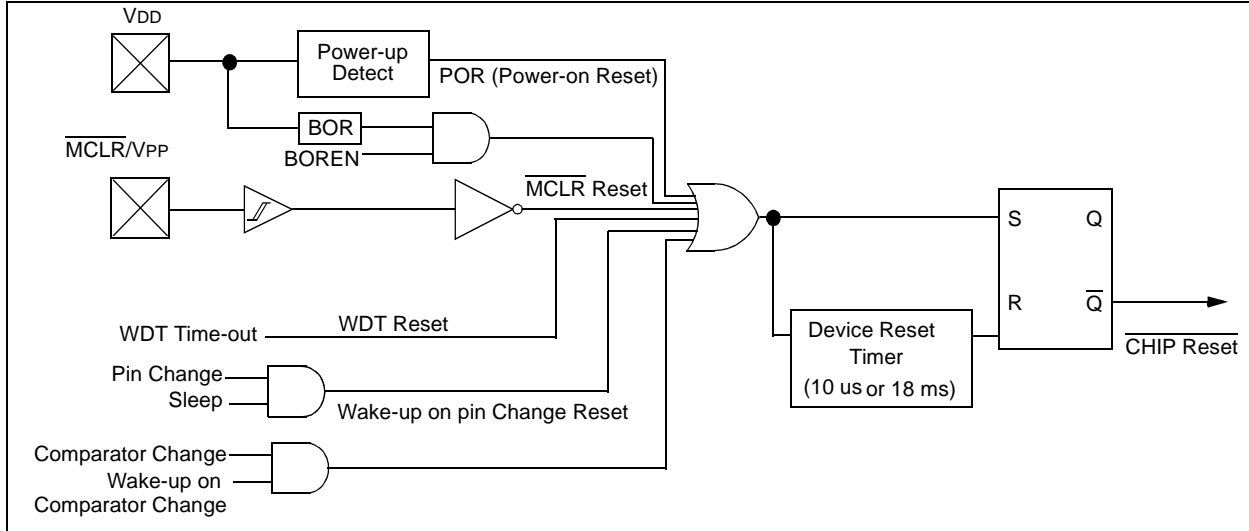
In [Figure 8-9](#), the on-chip Power-on Reset feature is being used ( $\overline{\text{MCLR}}$  is tied to  $V_{DD}$  or pulled high by its internal pull-up). The  $V_{DD}$  is stable before the start-up timer times out and there is no problem in getting a proper Reset. However, [Figure 8-10](#) depicts a problem situation where  $V_{DD}$  rises too slowly. The time between when the DRT senses that  $\overline{\text{MCLR}}$  is high and when  $\overline{\text{MCLR}}$  and  $V_{DD}$  actually reach their full value, is too long. In this situation, when the start-up timer times out,  $V_{DD}$  has not reached the  $V_{DD}(\text{min})$  value and the chip may not function correctly. For such situations, we recommend that external RC circuits be used to achieve longer POR delay times ([Figure 8-9](#)).

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

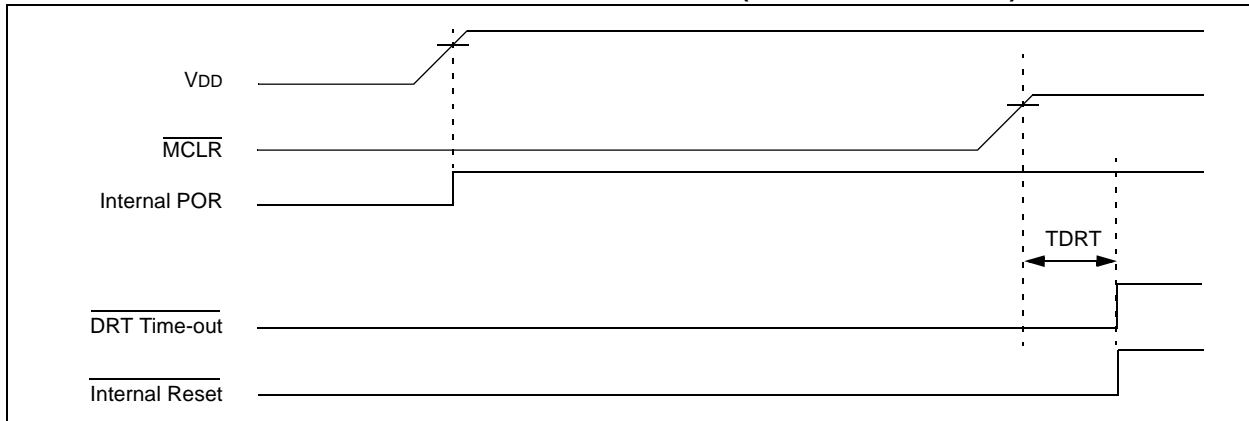
For additional information, refer to Application Notes AN522, “*Power-Up Considerations*” (DS00522) and AN607, “*Power-up Trouble Shooting*” (DS00607).

# PIC16F570

**FIGURE 8-7: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT**



**FIGURE 8-8: TIME-OUT SEQUENCE ON POWER-UP ( $\overline{\text{MCLR}}$  PULLED LOW)**



**FIGURE 8-9: TIME-OUT SEQUENCE ON POWER-UP ( $\overline{\text{MCLR}}$  TIED TO  $V_{DD}$ ): FAST  $V_{DD}$  RISE TIME**

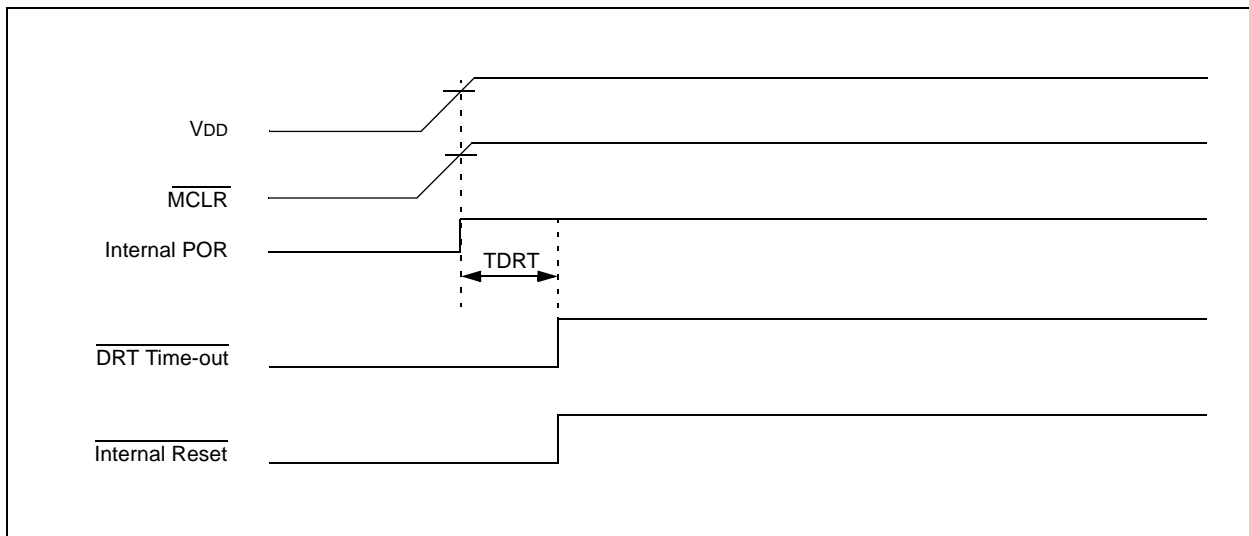
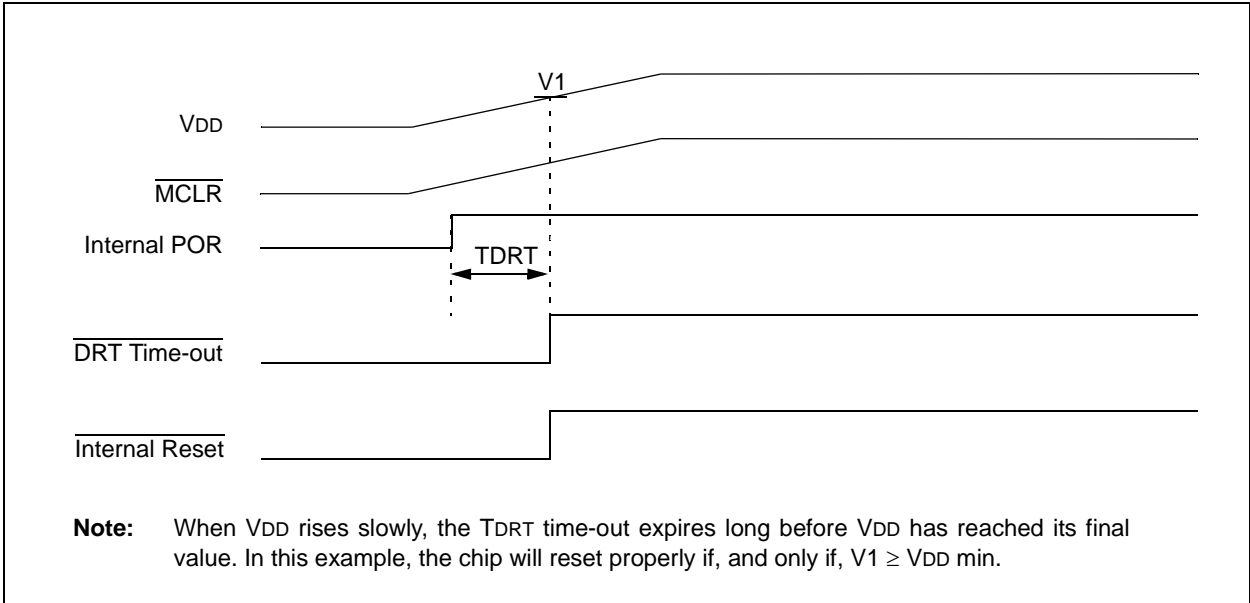


FIGURE 8-10: TIME-OUT SEQUENCE ON POWER-UP ( $\overline{\text{MCLR}}$  TIED TO  $V_{\text{DD}}$ ): SLOW  $V_{\text{DD}}$  RISE TIME



# PIC16F570

## 8.6 Device Reset Timer (DRT)

On the PIC16F570 device, the DRT runs any time the device is powered up. DRT runs from Reset and varies based on oscillator selection and Reset type (see [Table 8-4](#)).

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 after  $\overline{MCLR}$  has reached a logic high ( $V_{IH\ MCLR}$ ) level. Using an external RC network connected to the  $\overline{MCLR}$  input is not required in most cases. This allows savings in cost-sensitive and/or space restricted applications, as well as allowing the use of that pin as a general purpose input.

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

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

Reset sources are POR,  $\overline{MCLR}$ , WDT time-out and wake-up on pin or comparator change. See [Section 8.10.2 “Wake-up from Sleep”](#), [Notes 1, 2 and 3](#).

## 8.7 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 external RC oscillator of the OSC1/CLKIN pin and the internal 4/8 MHz oscillator. This means that the WDT will run even if the main processor clock has been stopped, for example, by execution of a SLEEP instruction. During normal operation or Sleep, a WDT Reset or wake-up Reset, generates a device Reset.

The  $\overline{TO}$  bit of the STATUS register will be cleared upon a Watchdog Timer Reset.

The WDT can be permanently disabled by programming the configuration WDTE as a '0' (see [Section 8.1 “Configuration Bits”](#)). Refer to the PIC16F570 Programming Specifications to determine how to access the Configuration Word.

TABLE 8-4: TYPICAL DRT PERIODS

Oscillator Configuration	POR Reset	Subsequent Resets
HS, XT, LP	18 ms	18 ms
EC	10 $\mu$ s	10 $\mu$ s
INTOSC, EXTRC	10 $\mu$ s	10 $\mu$ s

### 8.7.1 WDT PERIOD

The WDT has a nominal time-out period of 18 ms, (with no prescaler). If a longer time-out period is desired, a prescaler with a division ratio of up to 1:128 can be assigned to the WDT (under software control) by writing to the OPTION register. Thus, a time-out period of a nominal 2.3 seconds can be realized. These periods vary with temperature,  $V_{DD}$  and part-to-part process variations (see DC specs).

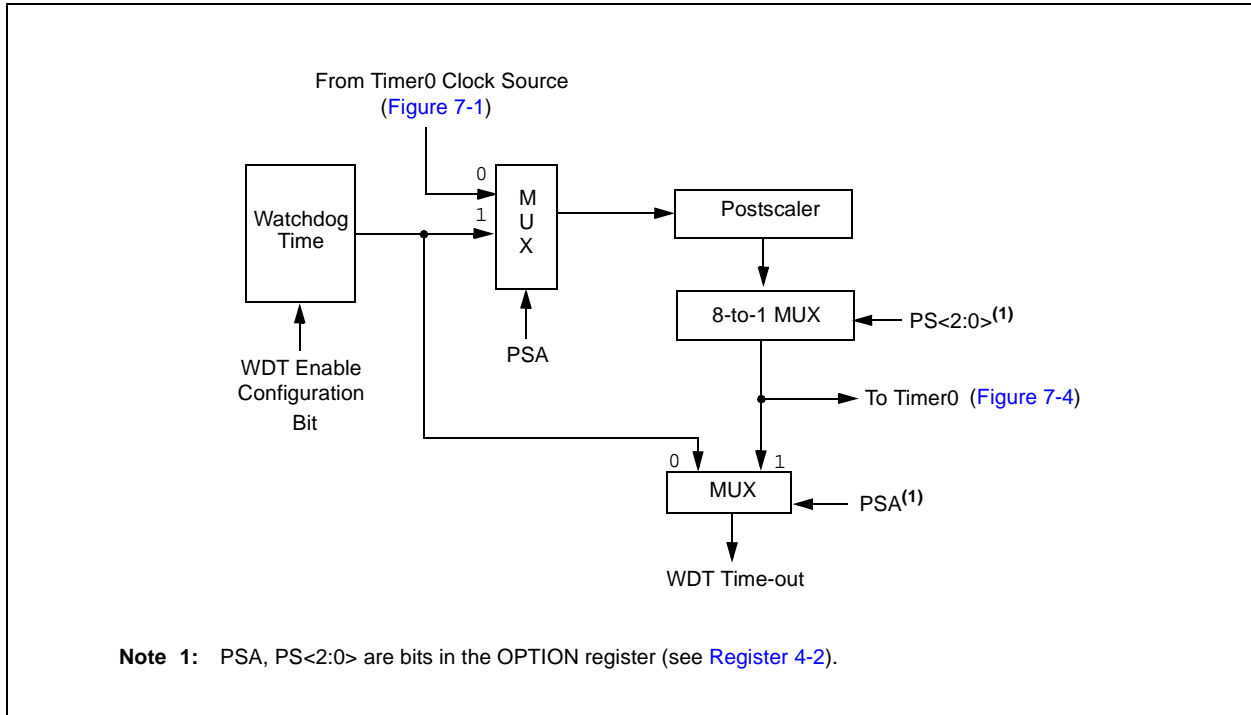
Under worst-case conditions ( $V_{DD} = \text{Min.}$ , Temperature = Max., max. WDT prescaler), it may take several seconds before a WDT time-out occurs.

### 8.7.2 WDT PROGRAMMING CONSIDERATIONS

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

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

**FIGURE 8-11: WATCHDOG TIMER BLOCK DIAGRAM**



**TABLE 8-5: REGISTERS ASSOCIATED WITH THE WATCHDOG TIMER**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
OPTION	RBWU	RBPU	T0CS	T0SE	PSA	PS2	PS1	PS0	18

**Legend:** Shaded boxes = Not used by Watchdog Timer.

## 8.8 Time-out Sequence ( $\overline{TO}$ ) and Power-down ( $\overline{PD}$ ) Reset Status

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

**TABLE 8-6:  $\overline{TO}/\overline{PD}$  STATUS AFTER RESET**

$\overline{TO}$	$\overline{PD}$	Reset Caused By
0	0	WDT wake-up from Sleep
0	u	WDT time-out (not from Sleep)
1	0	$\overline{MCLR}$ wake-up from Sleep
1	1	Power-up or Brown-out Reset
u	u	$\overline{MCLR}$ not during Sleep

**Legend:** u = unchanged

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

# PIC16F570

## 8.9 Brown-out Reset (BOR)

A brown-out is a condition where device power (VDD) dips below its minimum value, but not to zero, and then recovers. The device should be reset in the event of a brown-out. The Brown-out Reset feature is enabled by the BOREN Configuration bit.

If VDD falls below VBOR for greater than parameter (TBOR) (see Figure 8-12), the brown-out situation will reset the device. This will occur regardless of VDD slew rate. A Reset is not insured to occur if VDD falls below VBOR for less than parameter (TBOR).

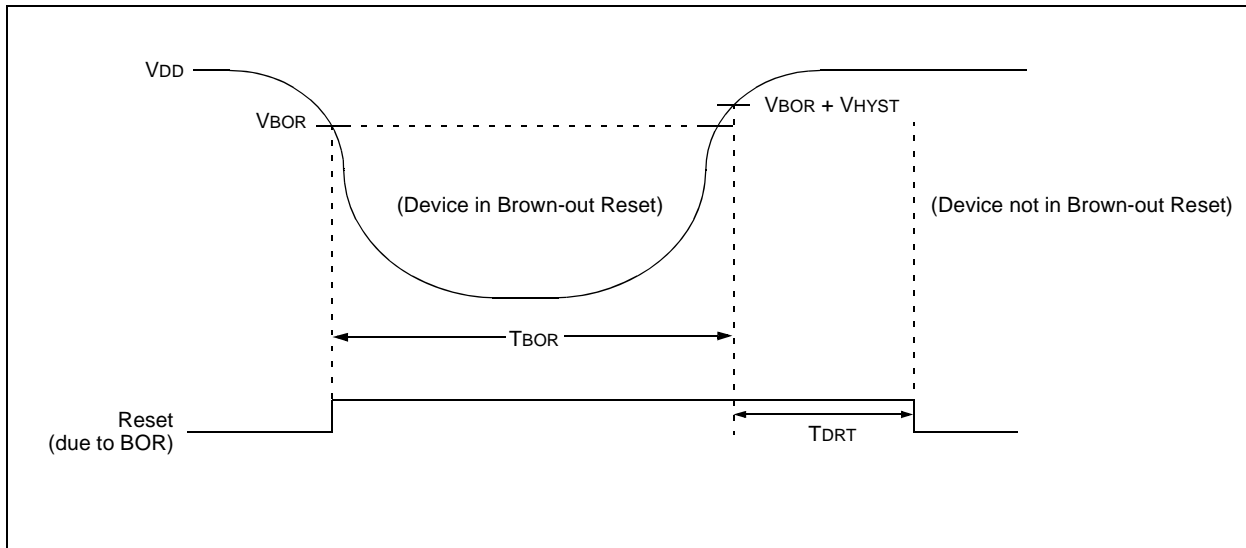
Please see Section 15.0 “Electrical Characteristics” for the VBOR specification and other parameters shown in Figure 8-12.

On any Reset (Power-on, Brown-out Reset, Watchdog Timer, etc.), the chip will remain in Reset until VDD rises above VBOR (see Figure 8-12). If enabled, the Device Reset Timer will now be invoked, and will keep the chip in Reset an additional 18 ms.

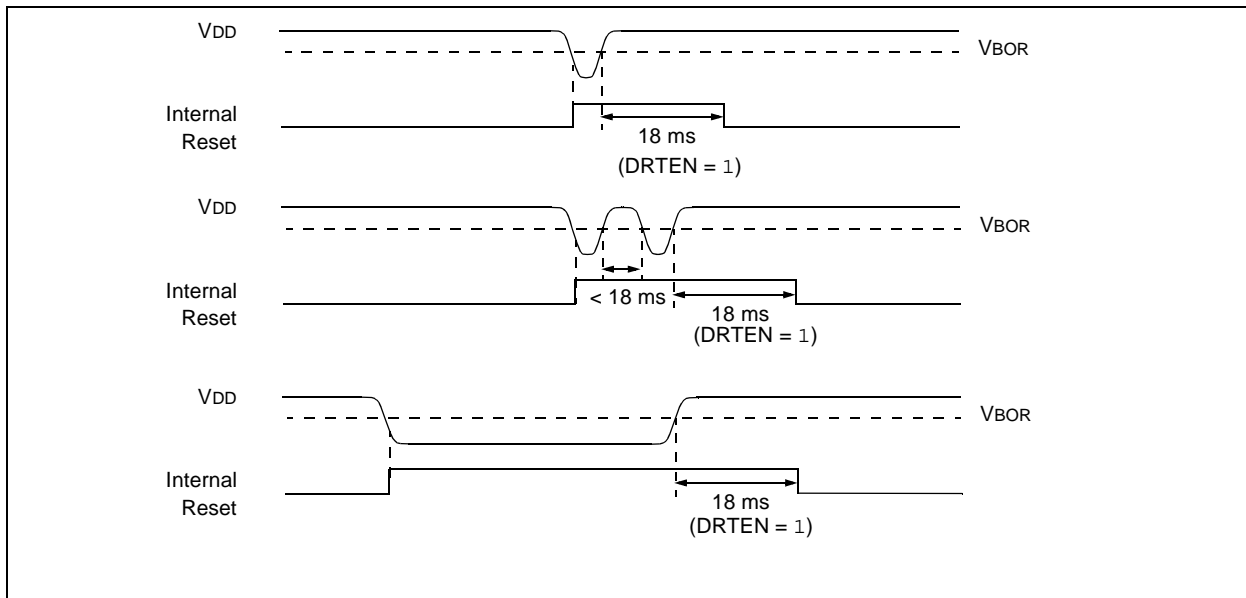
**Note:** The Device Reset Timer is enabled by the DRTEN bit in the Configuration Word register.

If VDD drops below VBOR while the Device Reset Timer is running, the chip will go back into a Brown-out Reset and the Device Reset Timer will be re-initialized. Once VDD rises above VBOR, the Device Reset Timer will execute a 18 ms Reset.

**FIGURE 8-12: BROWN-OUT RESET TIMING AND CHARACTERISTICS**



**FIGURE 8-13: BROWN-OUT SITUATIONS**



## 8.10 Power-down Mode (Sleep)

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

### 8.10.1 SLEEP

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

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

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

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

### 8.10.2 WAKE-UP FROM SLEEP

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

1. An external Reset input on  $\overline{MCLR}/V_{PP}$  pin.
2. A Watchdog Timer Time-out Reset (if WDT was enabled).
3. From an interrupt source, see [Section 8.11 "Interrupts"](#) for more information.

On waking from Sleep, the processor will continue to execute the instruction immediately following the `SLEEP` instruction. If the  $\overline{WUR}$  bit is also set, upon waking from Sleep, the device will reset. If the  $\overline{GIE}$  bit is also set, upon waking from Sleep, the processor will branch to the interrupt vector. Please see [Section 8.11 "Interrupts"](#) for more information.

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

**CAUTION:** Right before entering Sleep, read the input pins. When in Sleep, wake-up occurs when the values at the pins change from the state they were in at the last reading. If a wake-up on change occurs and the pins are not read before re-entering Sleep, a wake-up will occur immediately even if no pins change while in Sleep mode.

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

**CAUTION:** Right before entering Sleep, read the comparator Configuration register(s) `CM1CON0` and `CM2CON0`. When in Sleep, wake-up occurs when the comparator output bit `C1OUT` and `C2OUT` change from the state they were in at the last reading. If a wake-up on comparator change occurs and the pins are not read before re-entering Sleep, a wake-up will occur immediately, even if no pins change while in Sleep mode.

## 8.11 Interrupts

The interrupt feature allows certain events to preempt normal program flow. Firmware is used to determine the source of the interrupt and act accordingly. Some interrupts can be configured to wake the MCU from Sleep mode.

These following interrupt sources are available on the PIC16F570 device:

- Timer0 Overflow
- ADC Completion
- Comparator Output Change
- Interrupt-on-change pin

Refer to the corresponding chapters for details.

### 8.11.1 OPERATION

Interrupts are disabled upon any device Reset. They are enabled by setting the following bits:

- $\overline{GIE}$  bit of the `INTCON0` register
- Interrupt Enable bit(s) for the specific interrupt event(s)

The enable bits for specific interrupts can be found in the `INTCON1` register. An interrupt is recorded for a specific interrupt via flag bits found in the `INTCON0` register.

The ADC Conversion flag and the Timer0 Overflow flags will be set regardless of the status of the  $\overline{GIE}$  and individual interrupt enable bits.

The Comparator and Interrupt-on-change flags must be enabled for use. One or both of the comparator outputs can be enabled to affect the interrupt flag by clearing the `C1WU` bit in the `CM1CON0` register and the `C2WU` bit in the `CM2CON0` register. The Interrupt-on-change flag is enabled by clearing the `RBWU` bit in the `OPTION` register.

# PIC16F570

The following events happen when an interrupt event occurs while the GIE bit is set:

- Current prefetched instruction is flushed
- GIE bit is cleared
- Current Program Counter (PC) is pushed onto the stack
- Several registers are automatically switched to a secondary set of registers to store critical data. (See [Section 8.12 “Automatic Context Switching”](#))
- PC is loaded with the interrupt vector 0004h

The firmware within the Interrupt Service Routine (ISR) should determine the source of the interrupt by polling the interrupt flag bits. The interrupt flag bits must be cleared before exiting the ISR to avoid repeated interrupts. Because the GIE bit is cleared, any interrupt that occurs while executing the ISR will be recorded through its interrupt flag, but will not cause the processor to redirect to the interrupt vector.

## 8.12 Automatic Context Switching

While the device is executing from the ISR, a secondary set of W, STATUS, FSR and BSR registers are used by the CPU. These registers are still addressed at the same location, but hold persistent, independent values for use inside the ISR. This allows the contents of the primary set of registers to be unaffected by interrupts in the main line execution. The contents of the secondary set of context registers are visible in the SFR map as the IW, ISTATUS, IFSR and IBSR registers. When executing code from within the ISR, these registers will read back the main line context, and vice versa.

The `RETFIE` instruction exits the ISR by popping the previous address from the stack, switching back to the original set of critical registers and setting the GIE bit.

For additional information on a specific interrupt's operation, refer to its peripheral chapter.

- Note 1:** Individual interrupt flag bits may be set, regardless of the state of any other enable bits.
- 2:** All interrupts will be ignored while the GIE bit is cleared. Any interrupt occurring while the GIE bit is clear will be serviced when the GIE bit is set again.
  - 3:** All interrupts should be disabled prior to executing writes or row erases in the self-writable flash data memory.
  - 4:** The user must manage the contents of the context registers if they are using interrupts that will vector to the Interrupt Service Routine (ISR). The context registers (IW, ISTATUS, IFSR and IBSR) power-up in unknown states following POR and BOR events.

## 8.13 Interrupts during Sleep

Any of the interrupt sources can be used to wake from Sleep. To wake from Sleep, the peripheral must be operating without the system clock. The interrupt source must have the appropriate Interrupt Enable bit(s) set prior to entering Sleep.

On waking from Sleep, if the GIE bit is also set, the processor will branch to the interrupt vector. Otherwise, the processor will continue executing instructions after the `SLEEP` instruction. The instruction directly after the `SLEEP` instruction will always be executed before branching to the ISR. Refer to the [Section 8.10 “Power-down Mode \(Sleep\)”](#) for more details.

**TABLE 8-7: INTERRUPT PRIORITIES**

	In Sleep	GIE	WUR
Vector or Wake-up and Vector	X	1	0
Wake-up Reset	1	X	1
Wake-up Inline	1	0	0
Watchdog Wake-up Inline	1	X	0
Watchdog Wake-up Reset	1	X	1

## 8.14 Register Definitions — Interrupt Control

### REGISTER 8-2: INTCON0 REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	R/W-0
ADIF	CWIF	TOIF	RBIF	—	—	—	GIE
bit 7							bit 0

#### Legend:

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

- bit 7      **ADIF:** A/D Converter Interrupt Flag bit  
           1 = A/D conversion complete (must be cleared by software)  
           0 = A/D conversion has not completed or has not been started
- bit 6      **CWIF:** Comparator 1 or 2 Interrupt Flag bit  
           1 = Comparator interrupt-on-change has occurred<sup>(1)</sup>  
           0 = No change in Comparator 1 or 2 output
- bit 5      **TOIF:** Timer0 Overflow Interrupt Flag bit  
           1 = TMR0 register has overflowed (must be cleared by software)  
           0 = TMR0 register did not overflow
- bit 4      **RBIF:** PORTB Interrupt-on-change Flag bit  
           1 = Wake-up or interrupt has occurred (cleared in software)<sup>(2)</sup>  
           0 = Wake-up or interrupt has not occurred
- bit 3-1    **Unimplemented:** Read as '0'
- bit 0      **GIE:** Global Interrupt Enable bit  
           1 = Interrupt sets PC to address 0x004 (Vector to ISR)  
           0 = Interrupt causes wake-up and inline code execution

**Note 1:** This bit only functions when the  $\overline{C1WU}$  or  $\overline{C2WU}$  bits are cleared (see [Register 10-1](#) and [Register 10-2](#)).  
**2:** The RBWU bit of the OPTION register must be cleared to enable this function (see [Register 4-2](#)).

# PIC16F570

## REGISTER 8-3: INTCON1 REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	R/W-0
ADIE	CWIE	TOIE	RBIE	—	—	—	WUR
bit 7							bit 0

### Legend:

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

- bit 7      **ADIE:** A/D Converter (ADC) Interrupt Enable bit  
 1 = Enables the ADC interrupt  
 0 = Disables the ADC interrupt
- bit 6      **CWIE:** Comparator 1 and 2 Interrupt Enable bit  
 1 = Enables the Comparator 1 and 2 Interrupt  
 0 = Disables the Comparator 1 and 2 Interrupt
- bit 5      **TOIE:** Timer0 Overflow Interrupt Enable bit  
 1 = Enables the Timer0 interrupt  
 0 = Disables the Timer0 interrupt
- bit 4      **RBIE:** PORTB on Pin Change Interrupt Enable bit  
 1 = Interrupt-on-change pin enabled  
 0 = Interrupt-on-change pin disabled
- bit 3-1    **Unimplemented:** Read as '0'
- bit 0      **WUR:** Wake-up Reset Enable bit  
 1 = Interrupt source causes device Reset on wake-up  
 0 = Interrupt source wakes up device from Sleep (Vector to ISR or inline execution)

## 8.15 Program Verification/Code Protection

If the code protection bit has not been programmed, the on-chip program memory can be read out for verification purposes.

The first 64 locations and the last location (OSCCAL) can be read, regardless of the code protection bit setting.

## 8.16 ID Locations

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

Use only the lower four bits of the ID locations and always program the upper eight bits as '0's.

## 8.17 In-Circuit Serial Programming™

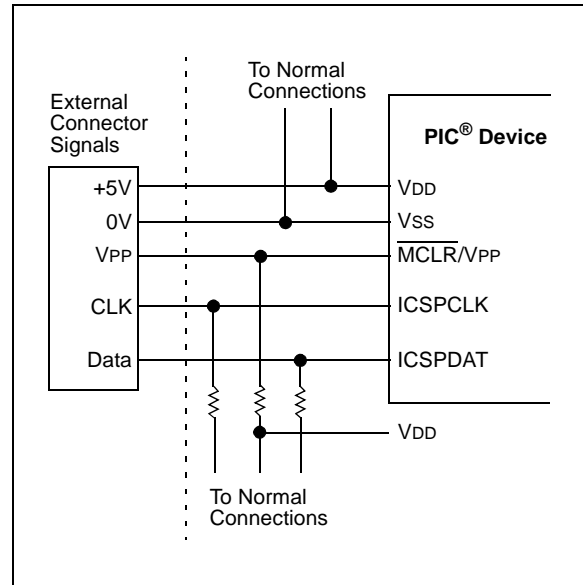
The PIC16F570 microcontroller can be serially programmed while in the end application circuit. This is simply done with two lines for clock and data, and three other lines for power, ground and the programming voltage. This allows customers to manufacture boards with unprogrammed devices and then program the microcontroller just before shipping the product. This also allows the most recent firmware, or a custom firmware, to be programmed.

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

After Reset, 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 PIC16F570 Programming Specifications.

A typical In-Circuit Serial Programming connection is shown in [Figure 8-14](#).

**FIGURE 8-14: TYPICAL IN-CIRCUIT SERIAL PROGRAMMING CONNECTION**



# PIC16F570

## 9.0 ANALOG-TO-DIGITAL (A/D) CONVERTER

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

### 9.1 Clock Divisors

The ADC has four clock source settings  $ADCS\langle 1:0 \rangle$ . There are three divisor values 16, 8 and 4. The fourth setting is INTOSC with a divisor of four. These settings will allow a proper conversion when using an external oscillator at speeds from 20 MHz to 350 kHz. Using an external oscillator at a frequency below 350 kHz requires the ADC oscillator setting to be INTOSC/4 ( $ADCS\langle 1:0 \rangle = 11$ ) for valid ADC results.

The ADC requires 13 TAD periods to complete a conversion. The divisor values do not affect the number of TAD periods required to perform a conversion. The divisor values determine the length of the TAD period.

When the  $ADCS\langle 1:0 \rangle$  bits are changed while an ADC conversion is in process, the new ADC clock source will not be selected until the next conversion is started. This clock source selection will be lost when the device enters Sleep.

**Note:** The ADC clock is derived from the instruction clock. The ADCS divisors are then applied to create the ADC clock

#### 9.1.1 VOLTAGE REFERENCE

There is no external voltage reference for the ADC. The ADC reference voltage will always be  $V_{DD}$ .

#### 9.1.2 ANALOG MODE SELECTION

The  $ANS\langle 7:0 \rangle$  bits are used to configure pins for analog input. Upon any Reset,  $ANS\langle 7:0 \rangle$  defaults to FF. This configures the affected pins as analog inputs. Pins configured as analog inputs are not available for digital output. Users should not change the ANS bits while a conversion is in process. ANS bits are active regardless of the condition of ADON.

#### 9.1.3 ADC CHANNEL SELECTION

The CHS bits are used to select the analog channel to be sampled by the ADC. The  $CHS\langle 3:0 \rangle$  bits can be changed at any time without adversely effecting a conversion. To acquire an external analog signal, the  $CHS\langle 3:0 \rangle$  selection must match one of the pin(s) selected by the  $ANS\langle 7:0 \rangle$  bits. When the ADC is on ( $ADON = 1$ ) and a channel is selected that is also being used by the comparator, then both the comparator and the ADC will see the analog voltage on the pin.

**Note:** It is the user's responsibility to ensure that the use of the ADC and op amp simultaneously on the same pin does not adversely affect the signal being monitored or adversely effect device operation.

When the  $CHS\langle 3:0 \rangle$  bits are changed during an ADC conversion, the new channel will not be selected until the current conversion is completed. This allows the current conversion to complete with valid results. All channel selection information will be lost when the device enters Sleep.

#### 9.1.4 THE $\overline{GO/DONE}$ BIT

The  $\overline{GO/DONE}$  bit is used to determine the status of a conversion, to start a conversion and to manually halt a conversion in process. Setting the  $\overline{GO/DONE}$  bit starts a conversion. When the conversion is complete, the ADC module clears the  $\overline{GO/DONE}$  bit and sets the ADIF bit in the INTCON0 register.

A conversion can be terminated by manually clearing the  $\overline{GO/DONE}$  bit while a conversion is in process. Manual termination of a conversion may result in a partially converted result in ADRES.

The  $\overline{GO/DONE}$  bit is cleared when the device enters Sleep, stopping the current conversion. The ADC does not have a dedicated oscillator, it runs off of the instruction clock. Therefore, no conversion can occur in Sleep.

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

#### 9.1.5 A/D ACQUISITION REQUIREMENTS

For the ADC to meet its specified accuracy, the charge holding capacitor (CHOLD) must be allowed to fully charge to the input channel voltage level. The Analog Input model is shown in [Figure 9-1](#). The source impedance ( $R_s$ ) and the internal sampling switch ( $R_{ss}$ ) impedance directly affect the time required to charge the capacitor CHOLD. The sampling switch ( $R_{ss}$ ) impedance varies over the device voltage ( $V_{DD}$ ), see [Figure 9-1](#). **The maximum recommended impedance for analog sources is 10 k $\Omega$ .** As the source impedance is decreased, the acquisition time may be decreased. After the analog input channel is selected (or changed), an A/D acquisition must be done before the conversion can be started. To calculate the minimum acquisition time, [Equation 9-1](#) may be used. This equation assumes that 1/2 LSb error is used (256 steps for the ADC). The 1/2 LSb error is the maximum error allowed for the ADC to meet its specified resolution.

## EQUATION 9-1: ACQUISITION TIME EXAMPLE

Assumptions:

Temperature = 50°C and external impedance of 10 kΩ 5.0V VDD

$T_{acq}$  = Amplifier Settling Time + Hold Capacitor Charging Time + Temperature Coefficient

=  $T_{AMP} + T_C + T_{COFF}$

=  $2 \mu s + T_C + [(Temperature - 25^\circ C)(0.05 \mu s/^\circ C)]$

Solving for  $T_C$ :

$T_C$  =  $CHOLD (RIC + R_{SS} + R_S) \ln(1/512)$

=  $-25pF (1 k\Omega + 7 k\Omega + 10 k\Omega) \ln(0.00196)$

=  $2.81 \mu s$

Therefore:

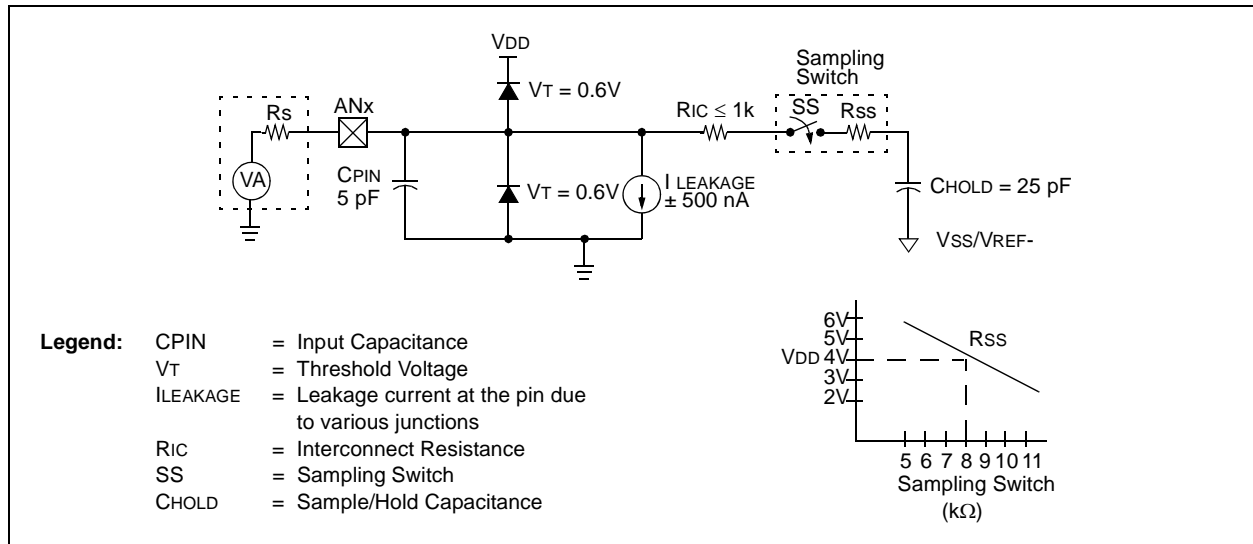
$T_{acq}$  =  $2 \mu s + 2.81 \mu s + [(50^\circ C - 25^\circ C)(0.05 \mu s/^\circ C)]$

=  $6.06 \mu s$

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

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

## FIGURE 9-1: ANALOG INPUT MODULE



# PIC16F570

## 9.1.6 ANALOG CONVERSION RESULT REGISTER

The ADRES register contains the results of the last conversion. These results are present during the sampling period of the next analog conversion process. After the sampling period is over, ADRES is cleared (= 0). A 'leading one' is then right shifted into the ADRES to serve as an internal conversion complete bit. As each bit weight, starting with the MSB, is converted, the leading one is shifted right and the converted bit is stuffed into ADRES. After a total of nine

right shifts of the 'leading one' have taken place, the conversion is complete; the 'leading one' has been shifted out and the  $\overline{\text{GO/DONE}}$  bit is cleared.

If the  $\overline{\text{GO/DONE}}$  bit is cleared in software during a conversion, the conversion stops and the ADIF bit will not be set to a '1'. The data in ADRES is the partial conversion result. This data is valid for the bit weights that have been converted. The position of the 'leading one' determines the number of bits that have been converted. The bits that were not converted before the  $\overline{\text{GO/DONE}}$  was cleared are unrecoverable.

### REGISTER 9-1: ADCON0: A/D CONTROL REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-0	R/W-0
ADCS1	ADCS0	CHS3	CHS2	CHS1	CHS0	$\overline{\text{GO/DONE}}$	ADON
bit 7						bit 0	

#### Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

bit 7-6 **ADCS<1:0>**: ADC Conversion Clock Select bits

00 = Fosc/16

01 = Fosc/8

10 = Fosc/4

11 = INTOSC/4

bit 5-2 **CHS<3:0>**: ADC Channel Select bits<sup>(1)</sup>

0000 = Channel 0 (AN0)

0001 = Channel 1 (AN1)

0010 = Channel 2 (AN2)

0011 = Channel 3 (AN3)

0100 = Channel 4 (AN4)

0101 = Channel 5 (AN5)

0110 = Channel 6 (AN6)

0111 = Channel 7 (AN7)

1xxx = Reserved

1111 = 0.6V Fixed Input Reference (VFIR)

bit 1  **$\overline{\text{GO/DONE}}$** : ADC Conversion Status bit<sup>(2)</sup>

1 = ADC conversion in progress. Setting this bit starts an ADC conversion cycle. This bit is automatically cleared by hardware when the ADC is done converting.

0 = ADC conversion completed/not in progress. Manually clearing this bit while a conversion is in process terminates the current conversion.

bit 0 **ADON**: ADC Enable bit

1 = ADC module is operating

0 = ADC module is shut-off and consumes no power

**Note 1:** CHS<3:0> bits default to 1 after any Reset.

**Note 2:** If the ADON bit is clear, the  $\overline{\text{GO/DONE}}$  bit cannot be set.

## REGISTER 9-2: ADRES: A/D CONVERSION RESULTS REGISTER

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

### Legend:

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

bit 7-0                      **ADRES<7:0>**: ADC Result Register bits

### EXAMPLE 9-1: PERFORMING AN ANALOG-TO-DIGITAL CONVERSION

```

;Sample code operates out of BANK0
    MOVLW 0xF1      ;configure A/D
    MOVWF ADCON0
    BSF ADCON0, 1  ;start conversion
loop0  BTFSC ADCON0, 1;wait for 'DONE'
       GOTO loop0
       MOVF ADRES, W ;read result
       MOVWF result0 ;save result

       BSF ADCON0, 2 ;setup for read of
                       ;channel 1
loop1  BSF ADCON0, 1 ;start conversion
       BTFSC ADCON0, 1;wait for 'DONE'
       GOTO loop1
       MOVF ADRES, W ;read result
       MOVWF result1 ;save result

       BSF ADCON0, 3 ;setup for read of
       BCF ADCON0, 2 ;channel 2
       BSF ADCON0, 1 ;start conversion
loop2  BTFSC ADCON0, 1;wait for 'DONE'
       GOTO loop2
       MOVF ADRES, W ;read result
       MOVWF result2 ;save result
  
```

### EXAMPLE 9-2: CHANNEL SELECTION CHANGE DURING CONVERSION

```

    MOVLW 0xF1      ;configure A/D
    MOVWF ADCON0
    BSF ADCON0, 1  ;start conversion
    BSF ADCON0, 2  ;setup for read of
                       ;channel 1
loop0  BTFSC ADCON0, 1;wait for 'DONE'
       GOTO loop0
       MOVF ADRES, W ;read result
       MOVWF result0 ;save result

       BSF ADCON0, 1 ;start conversion
       BSF ADCON0, 3 ;setup for read of
       BCF ADCON0, 2 ;channel 2
loop1  BTFSC ADCON0, 1;wait for 'DONE'
       GOTO loop1
       MOVF ADRES, W ;read result
       MOVWF result1 ;save result

       BSF ADCON0, 1 ;start conversion
loop2  BTFSC ADCON0, 1;wait for 'DONE'
       GOTO loop2
       MOVF ADRES, W ;read result
       MOVWF result2 ;save result
       CLRF ADCON0 ;optional: returns
                       ;pins to Digital mode and turns off
                       ;the ADC module
  
```

# PIC16F570

## 9.1.7 SLEEP

This ADC does not have a dedicated ADC clock, and therefore, no conversion in Sleep is possible. If a conversion is underway and a Sleep command is executed, the GO/DONE and ADON bit will be cleared. This will stop any conversion in process and power-down the ADC module to conserve power. Due to the nature of the conversion process, the ADRES may contain a partial conversion. At least one bit must have been converted prior to Sleep to have partial conversion data in ADRES. The ADCS and CHS bits are reset to their default condition; ANS<7:0> = 1s and CHS<3:0> = 1s.

- For accurate conversions, TAD must meet the following:
  - 500 ns < TAD < 50 μs
  - TAD = 1/(FOSC/divisor)

Shaded areas indicate TAD out of range for accurate conversions. If analog input is desired at these frequencies, use INTOSC/8 for the ADC clock source.

**TABLE 9-1: TAD FOR ADCS SETTINGS WITH VARIOUS OSCILLATORS**

Source	ADCS <1:0>	Divisor	20 MHz	16 MHz	8 MHz	4 MHz	1 MHz	500 kHz	350 kHz	200 kHz	100 kHz	32 kHz
INTOSC	11	4	—	—	.5 μs	1 μs	—	—	—	—	—	—
FOSC	10	4	.2 μs	.25 μs	.5 μs	1 μs	4 μs	8 μs	11 μs	20 μs	40 μs	125 μs
FOSC	01	8	.4 μs	.5 μs	1 μs	2 μs	8 μs	16 μs	23 μs	40 μs	80 μs	250 μs
FOSC	00	16	.8 μs	1 μs	2 μs	4 μs	16 μs	32 μs	46 μs	80 μs	160 μs	500 μs

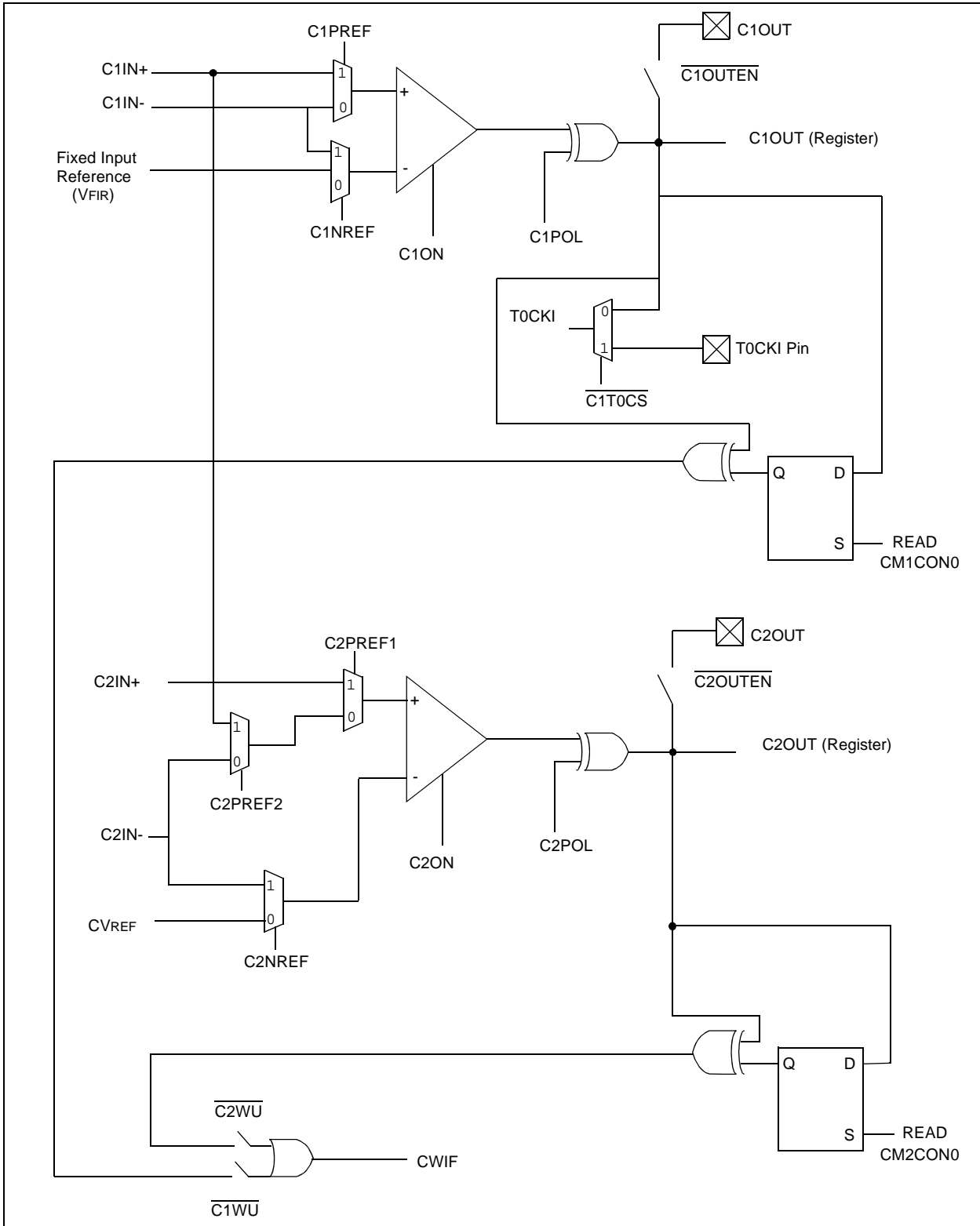
**TABLE 9-2: EFFECTS OF SLEEP ON ADCON0**

	ANS<7:0>	ADCS1	ADCS0	CHS<3:0>	GO/DONE	ADON
Entering Sleep	Unchanged	1	1	1	0	0
Wake or Reset	1	1	1	1	0	0

10.0 COMPARATOR(S)

This device contains two comparators and a comparator voltage reference.

FIGURE 10-1: COMPARATORS BLOCK DIAGRAM

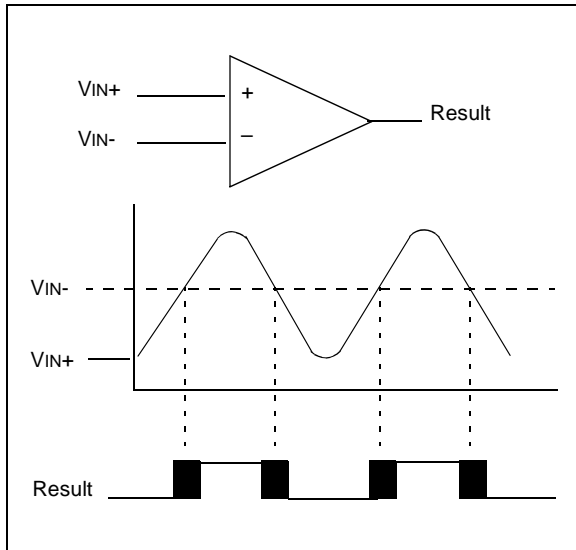


# PIC16F570

## 10.1 Comparator Operation

A single comparator is shown in [Figure 10-2](#) along with the relationship between the analog input levels and the digital output. When the analog input at  $V_{IN+}$  is less than the analog input  $V_{IN-}$ , the output of the comparator is a digital low level. The shaded area of the output of the comparator in [Figure 10-2](#) represent the uncertainty due to input offsets and response time. See [Table 15-2](#) for Common Mode Voltage.

**FIGURE 10-2: SINGLE COMPARATOR**



## 10.2 Comparator Reference

An internal reference signal may be used depending on the comparator operating mode. The analog signal that is present at  $V_{IN-}$  is compared to the signal at  $V_{IN+}$ , and the digital output of the comparator is adjusted accordingly ([Figure 10-2](#)). Please see [Section 11.0 “Comparator Voltage Reference Module”](#) for internal reference specifications.

## 10.3 Comparator Response Time

Response time is the minimum time after selecting a new reference voltage or input source before the comparator output is to have a valid level. If the comparator inputs are changed, a delay must be used to allow the comparator to settle to its new state. Please see [Table 15-2](#) for comparator response time specifications.

## 10.4 Comparator Output

The comparator output is read through the  $CxOUT$  bit in the  $CM1CON0$  or  $CM2CON0$  register. This bit is read-only. The comparator output may also be used externally, see [Section 10.1 “Comparator Operation”](#).

**Note:** Analog levels on any pin that is defined as a digital input may cause the input buffer to consume more current than is specified.

## 10.5 Comparator Wake-up Flag

The Comparator Wake-up Flag bit,  $CWIF$ , in the  $INTCON0$  register, is set whenever all of the following conditions are met:

- $\overline{C1WU} = 0$  ( $CM1CON0<0>$ ) or  $\overline{C2WU} = 0$  ( $CM2CON0<0>$ )
- $CM1CON0$  or  $CM2CON0$  has been read to latch the last known state of the  $C1OUT$  and  $C2OUT$  bit (`MOVF CM1CON0, W`)
- The output of a comparator has changed state

The wake-up flag may be cleared in software or by another device Reset.

## 10.6 Comparator Operation During Sleep

When the comparator is enabled it is active. To minimize power consumption while in Sleep mode, turn off the comparator before entering Sleep.

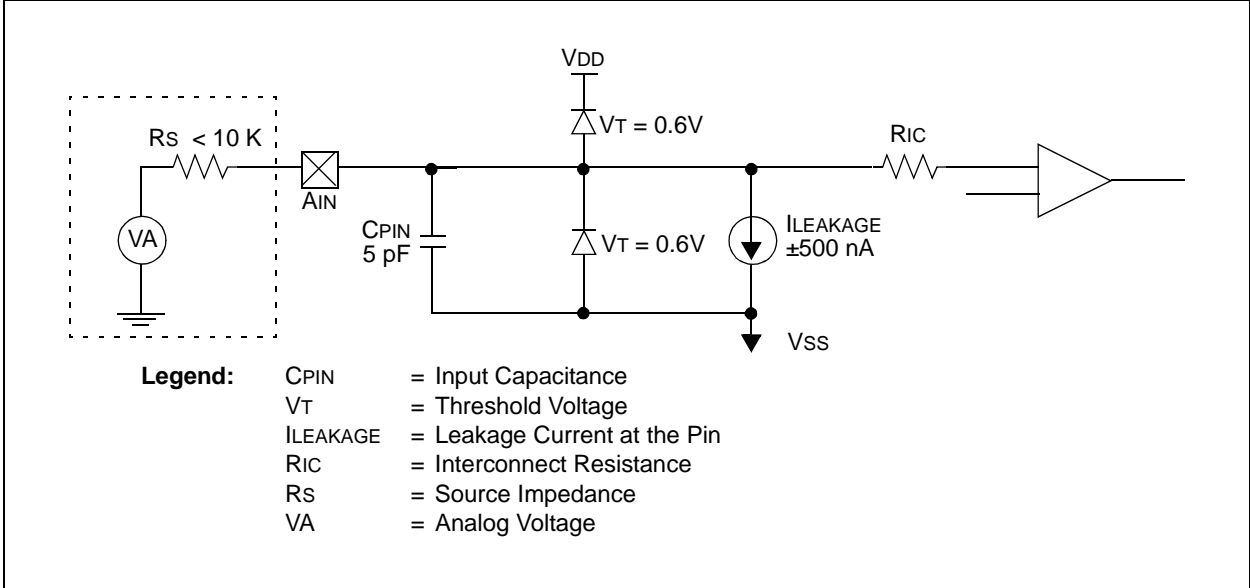
## 10.7 Effects of Reset

A Power-on Reset (POR) forces the  $CMxCON0$  register to its Reset state. This forces the Comparator input pins to analog Reset mode. Device current is minimized when analog inputs are present at Reset time.

## 10.8 Analog Input Connection Considerations

A simplified circuit for an analog input is shown in [Figure 10-3](#). Since the analog pins are connected to a digital output, they have reverse biased diodes to  $V_{DD}$  and  $V_{SS}$ . The analog input, therefore, must be between  $V_{SS}$  and  $V_{DD}$ . If the input voltage deviates from this range by more than 0.6V in either direction, one of the diodes is forward biased and a latch-up may occur. A maximum source impedance of 10 k $\Omega$  is recommended for the analog sources. Any external component connected to an analog input pin, such as a capacitor or a Zener diode, should have very little leakage current.

FIGURE 10-3: ANALOG INPUT MODE



# PIC16F570

## 10.9 Register Definitions — Comparator Control

### REGISTER 10-1: CM1CON0: COMPARATOR C1 CONTROL REGISTER

R-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
C1OUT	$\overline{\text{C1OUTEN}}$	C1POL	$\overline{\text{C1T0CS}}$	C1ON	C1NREF	C1PREF	$\overline{\text{C1WU}}$
bit 7							bit 0

#### Legend:

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

- bit 7      **C1OUT**: Comparator Output bit  
 1 =  $V_{IN+} > V_{IN-}$   
 0 =  $V_{IN+} < V_{IN-}$
- bit 6       **$\overline{\text{C1OUTEN}}$** : Comparator Output Enable bit<sup>(1)</sup>  
 1 = Output of comparator is NOT placed on the C1OUT pin  
 0 = Output of comparator is placed in the C1OUT pin
- bit 5      **C1POL**: Comparator Output Polarity bit  
 1 = Output of comparator is not inverted  
 0 = Output of comparator is inverted
- bit 4       **$\overline{\text{C1T0CS}}$** : Comparator TMR0 Clock Source bit  
 1 = TMR0 clock source selected by T0CS control bit  
 0 = Comparator output used as TMR0 clock source
- bit 3      **C1ON**: Comparator Enable bit  
 1 = Comparator is on  
 0 = Comparator is off
- bit 2      **C1NREF**: Comparator Negative Reference Select bit<sup>(2)</sup>  
 1 = C1IN- pin  
 0 = 0.6V Fixed Input Reference (VFIR)
- bit 1      **C1PREF**: Comparator Positive Reference Select bit<sup>(2)</sup>  
 1 = C1IN+ pin  
 0 = C1IN- pin
- bit 0       **$\overline{\text{C1WU}}$** : Comparator Wake-up On Change Enable bit<sup>(3)</sup>  
 1 = Wake-up On Comparator Change is disabled  
 0 = Wake-up On Comparator Change is enabled

**Note 1:** Overrides TRIS control of the port.

**2:** When this bit selects an I/O pin and the comparator is turned on, this feature will override the TRIS and ANSEL settings to make the respective pin an analog input. The value in the ANSEL register, however, is not over-written. When the comparator is turned off, the respective pin will revert back to the original TRIS and ANSEL settings.

**3:** The  $\overline{\text{C1WU}}$  bit must be cleared to enable the CWIF function. See the INTCON0 register ([Register 8-2](#)) for more information.

**REGISTER 10-2: CM2CON0: COMPARATOR C2 CONTROL REGISTER**

R-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
C2OUT	$\overline{\text{C2OUTEN}}$	C2POL	C2PREF2	C2ON	C2NREF	C2PREF1	$\overline{\text{C2WU}}$
bit 7							bit 0

**Legend:**

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

- bit 7            **C2OUT:** Comparator Output bit  
                   1 =  $V_{IN+} > V_{IN-}$   
                   0 =  $V_{IN+} < V_{IN-}$
- bit 6            **C2OUTEN:** Comparator Output Enable bit<sup>(1)</sup>  
                   1 = Output of comparator is NOT placed on the C2OUT pin  
                   0 = Output of comparator is placed in the C2OUT pin
- bit 5            **C2POL:** Comparator Output Polarity bit  
                   1 = Output of comparator not inverted  
                   0 = Output of comparator inverted
- bit 4            **C2PREF2:** Comparator Positive Reference Select bit  
                   1 = C1IN+ pin  
                   0 = C2IN- pin
- bit 3            **C2ON:** Comparator Enable bit  
                   1 = Comparator is on  
                   0 = Comparator is off
- bit 2            **C2NREF:** Comparator Negative Reference Select bit<sup>(2)</sup>  
                   1 = C2IN- pin  
                   0 = CVREF
- bit 1            **C2PREF1:** Comparator Positive Reference Select bit<sup>(2)</sup>  
                   1 = C2IN+ pin  
                   0 = C2PREF2 controls analog input selection
- bit 0            **C2WU:** Comparator Wake-up on Change Enable bit<sup>(3)</sup>  
                   1 = Wake-up on Comparator change is disabled  
                   0 = Wake-up on Comparator change is enabled.

**Note 1:** Overrides TRIS control of the port.

**2:** When this bit selects an I/O pin and the comparator is turned on, this feature will override the TRIS and ANSEL settings to make the respective pin an analog input. The value in the ANSEL register, however, is not over-written. When the comparator is turned off, the respective pin will revert back to the original TRIS and ANSEL settings.

**3:** The  $\overline{\text{C2WU}}$  bit must be cleared to enable the CWIF function. See the INTCON0 register ([Register 8-2](#)) for more information.

**TABLE 10-1: REGISTERS ASSOCIATED WITH COMPARATOR MODULE**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
STATUS	PA2	PA1	PA0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	17
CM1CON0	C1OUT	$\overline{\text{C1OUTEN}}$	C1POL	$\overline{\text{C1T0CS}}$	C1ON	C1NREF	C1PREF	$\overline{\text{C1WU}}$	62
CM2CON0	C2OUT	$\overline{\text{C2OUTEN}}$	C2POL	C2PREF2	C2ON	C2NREF	C2PREF1	$\overline{\text{C2WU}}$	63
TRIS	I/O Control Register (TRISA, TRISB, TRISC)								—

**Legend:** x = Unknown, u = Unchanged, – = Unimplemented, read as '0',  $\alpha$  = Depends on condition.

# PIC16F570

## 11.0 COMPARATOR VOLTAGE REFERENCE MODULE

The Comparator Voltage Reference module also allows the selection of an internally generated voltage reference for one of the C2 comparator inputs. The VRCON register (Register 11-1) controls the voltage reference module shown in Figure 11-1.

### 11.1 Configuring the Voltage Reference

The voltage reference can output 32 voltage levels; 16 in a high range and 16 in a low range.

Equation 11-1 determines the output voltages:

#### EQUATION 11-1:

$$\begin{aligned}
 &VRR = 1 \text{ (low range):} \\
 &CVREF = (VR<3:0>/24) \times VDD \\
 &VRR = 0 \text{ (high range):} \\
 &CVREF = (VDD/4) + (VR<3:0> \times VDD/32)
 \end{aligned}$$

## 11.2 Voltage Reference Accuracy

The full range of VSS to VDD cannot be realized due to construction of the module. The transistors on the top and bottom of the resistor ladder network (Figure 11-1) keep CVREF from approaching VSS or VDD. The exception is when the module is disabled by clearing the VREN bit of the VRCON register. When disabled, the reference voltage is VSS when VR<3:0> is '0000' and the VRR bit of the VRCON register is set. This allows the comparator to detect a zero-crossing and not consume the CVREF module current.

The voltage reference is VDD derived and, therefore, the CVREF output changes with fluctuations in VDD. The tested absolute accuracy of the comparator voltage reference can be found in Section 15.0 "Electrical Characteristics".

### REGISTER 11-1: VRCON: VOLTAGE REFERENCE CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
VREN	VROE1	VROE2	VRR	VR3	VR2	VR1	VR0
bit 7							bit 0

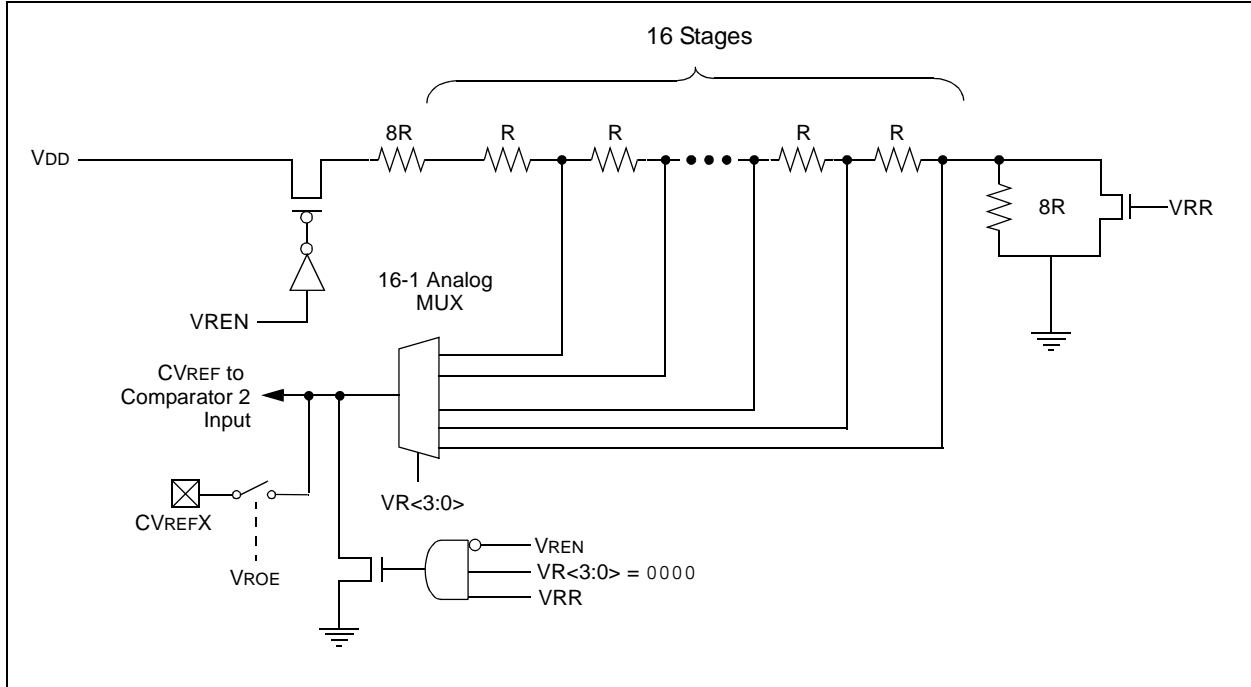
#### Legend:

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

- bit 7            **VREN:** CVREF Enable bit  
                   1 = CVREF is powered on  
                   0 = CVREF is powered down, no current is drawn
- bit 6            **VROE1:** CVREF1 Output Enable bit<sup>(1)</sup>  
                   1 = CVREF1 output is enabled  
                   0 = CVREF1 output is disabled
- bit 5            **VROE2:** CVREF2 Output Enable bit<sup>(1)</sup>  
                   1 = CVREF2 output is enabled  
                   0 = CVREF2 output is disabled
- bit 4            **VRR:** CVREF Range Selection bit  
                   1 = Low range  
                   0 = High range
- bit 3-0        **VR<3:0>** CVREF Value Selection bits  
                   When VRR = 1: CVREF = (VR<3:0>/24)\*VDD  
                   When VRR = 0: CVREF = VDD/4 + (VR<3:0>/32)\*VDD

**Note 1:** When this bit is set, the TRIS for the CVREFX pin is overridden and the analog voltage is placed on the CVREFX pin.

**FIGURE 11-1: COMPARATOR VOLTAGE REFERENCE BLOCK DIAGRAM**



**TABLE 11-1: REGISTERS ASSOCIATED WITH COMPARATOR VOLTAGE REFERENCE**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
VRCON	VREN	VROE1	VROE2	VRR	VR3	VR2	VR1	VR0	64
CM2CON0	C2OUT	$\overline{\text{C2OUTEN}}$	C2POL	C2PREF2	C2ON	C2NREF	C2PREF1	$\overline{\text{C2WU}}$	63

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

# PIC16F570

## 12.0 OPERATIONAL AMPLIFIER (OPA) MODULE

The OPA module has the following features:

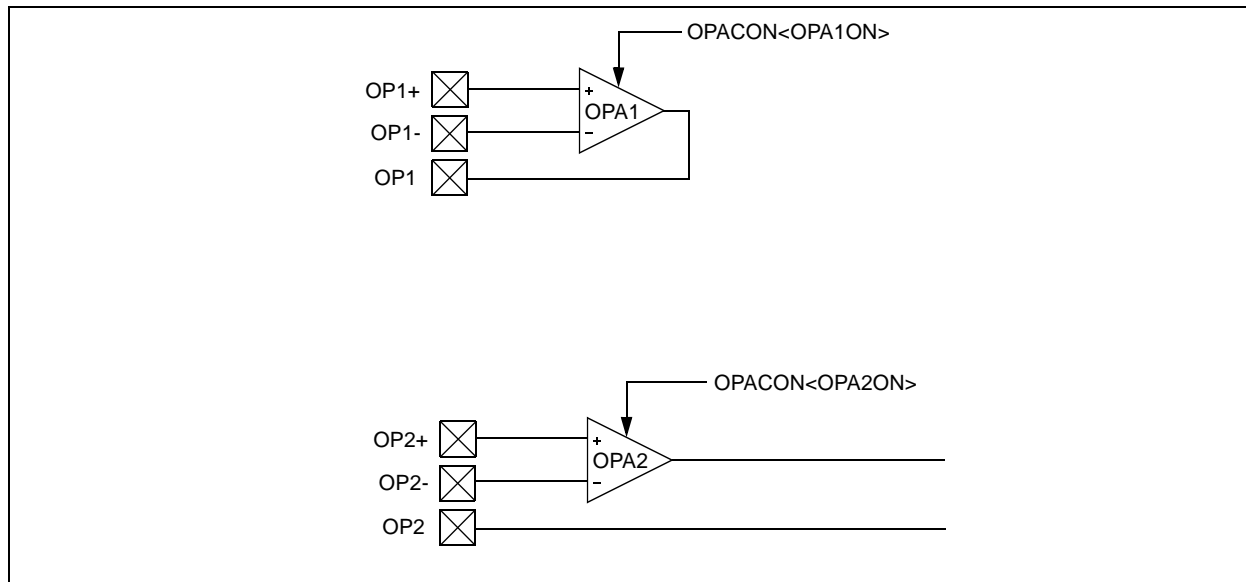
- Two independent Operational Amplifiers
- External connections to all ports
- 3 MHz Gain Bandwidth Product (GBWP)

### 12.1 OPACON Register

The OPA module is enabled by setting the OPAxON bit of the OPACON register.

**Note:** When OPA1 or OPA2 is enabled, the OP1 pin or OP2 pin, respectively, is driven by the op amp output, not by the port driver. Refer to [Table 15-13](#) for the electrical specifications for the op amp output drive capability.

FIGURE 12-1: OPA MODULE BLOCK DIAGRAM



**REGISTER 12-1: OPACON: OP AMP CONTROL REGISTER**

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	OPA2ON	OPA1ON
bit 7						bit 0	

**Legend:**

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

- bit 7-2                      **Unimplemented:** Read as '0'
- bit 1                      **OPA2ON:** Op Amp Enable bit  
                                  1 = Op amp 2 is enabled  
                                  0 = Op amp 2 is disabled
- bit 0                      **OPA1ON:** Op Amp Enable bit  
                                  1 = Op amp 1 is enabled  
                                  0 = Op amp 1 is disabled

## 12.2 Effects of a Reset

A device Reset forces all registers to their Reset state. This disables both op amps.

## 12.3 OPA Module Performance

Common AC and DC performance specifications for the OPA module:

- Common Mode Voltage Range
- Leakage Current
- Input Offset Voltage
- Open Loop Gain
- Gain Bandwidth Product (GBWP)

**Common mode voltage range** is the specified voltage range for the OP+ and OP- inputs, for which the OPA module will perform to within its specifications. The OPA module is designed to operate with input voltages between 0 and VDD-1.4V. Behavior for common mode voltages greater than VDD-1.4V, or below 0V, are beyond the normal operating range.

**Leakage current** is a measure of the small source or sink currents on the OP+ and OP- inputs. To minimize the effect of leakage currents, the effective impedances connected to the OP+ and OP- inputs should be kept as small as possible and equal.

**Input offset voltage** is a measure of the voltage difference between the OP+ and OP- inputs in a closed loop circuit with the OPA in its linear region. The offset voltage will appear as a DC offset in the output equal to the input offset voltage, multiplied by the gain of the circuit. The input offset voltage is also affected by the common mode voltage.

**Open loop gain** is the ratio of the output voltage to the differential input voltage, (OP+) - (OP-). The gain is greatest at DC and falls off with frequency.

**Gain Bandwidth Product** or GBWP is the frequency at which the open loop gain falls off to 0 dB.

## 12.4 Effects of Sleep

When enabled, the op amps continue to operate and consume current while the processor is in Sleep mode.

**TABLE 12-1: REGISTERS ASSOCIATED WITH THE OPA MODULE**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
ANSEL	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0	30
OPACON	—	—	—	—	—	—	OPA2ON	OPA1ON	67
TRIS	I/O Control Registers (TRISA, TRISB, TRISC)								—

**Legend:** x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used for the OPA module.

# PIC16F570

## 13.0 INSTRUCTION SET SUMMARY

The PIC16 instruction set is highly orthogonal and is comprised of three basic categories.

- **Byte-oriented** operations
- **Bit-oriented** operations
- **Literal and control** operations

Each PIC16 instruction is a 12-bit word divided into an **opcode**, which specifies the instruction type, and one or more **operands** which further specify the operation of the instruction. The formats for each of the categories is presented in [Figure 13-1](#), while the various opcode fields are summarized in [Table 13-1](#).

For **byte-oriented** instructions, 'f' represents a file register designator and 'd' represents a destination designator. The file register designator specifies which file register is to be used by the instruction.

The destination designator specifies where the result of the operation is to be placed. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed in the file register specified in the instruction.

For **bit-oriented** instructions, 'b' represents a bit field designator which selects the number of the bit affected by the operation, while 'f' represents the number of the file in which the bit is located.

For **literal and control** operations, 'k' represents an 8 or 9-bit constant or literal value.

**TABLE 13-1: OPCODE FIELD DESCRIPTIONS**

Field	Description
f	Register file address (0x00 to 0x1F)
W	Working register (accumulator)
b	Bit address within an 8-bit file register
k	Literal field, constant data or label
x	Don't care location (= 0 or 1) The assembler will generate code with x = 0. It is the recommended form of use for compatibility with all Microchip software tools.
d	Destination select; d = 0 (store result in W) d = 1 (store result in file register 'f') Default is d = 1
label	Label name
TOS	Top-of-Stack
PC	Program Counter
WDT	Watchdog Timer counter
$\overline{TO}$	Time-out bit
$\overline{PD}$	Power-down bit
dest	Destination, either the W register or the specified register file location
[ ]	Options
( )	Contents
$\mathbb{E}$	Assigned to
< >	Register bit field
$\mathbb{C}$	In the set of
italics	User defined term (font is courier)

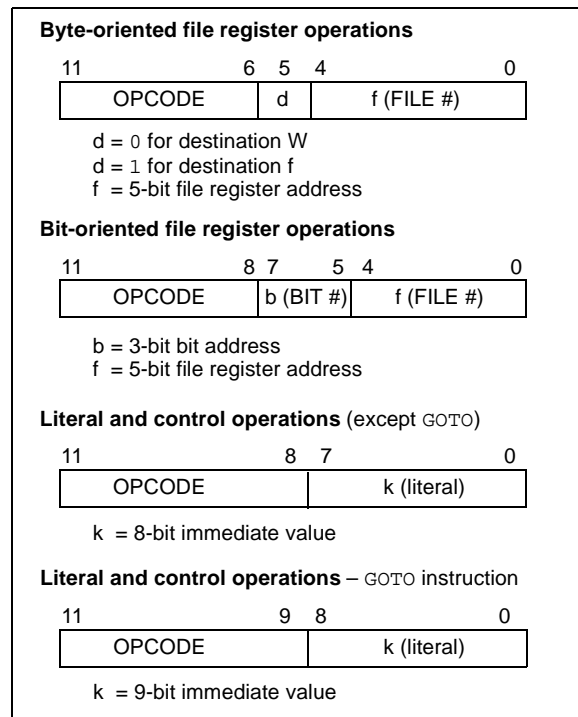
All instructions are executed within a single instruction cycle, unless a conditional test is true or the program counter is changed as a result of an instruction. In this case, the execution takes two instruction cycles. One instruction cycle consists of four oscillator periods. Thus, for an oscillator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

[Figure 13-1](#) 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 13-1: GENERAL FORMAT FOR INSTRUCTIONS**



**TABLE 13-2: 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
<b>BIT-ORIENTED FILE REGISTER OPERATIONS</b>						
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	
<b>LITERAL AND CONTROL OPERATIONS</b>						
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	$\overline{TO}$ , $\overline{PD}$	
GOTO k	Unconditional branch	2	101k	kkkk kkkk	None	
IORLW k	Inclusive OR literal with W	1	1101	kkkk kkkk	Z	
MOVLB k	Move Literal to BSR Register	1	0000	0001 0kkk	None	
MOVLW k	Move literal to W	1	1100	kkkk kkkk	None	
OPTION —	Load OPTION register	1	0000	0000 0010	None	
RETFIE —	Return from Interrupt	2	0000	0001 1111	None	
RETLW k	Return, place literal in W	2	1000	kkkk kkkk	None	3
RETURN —	Return, maintain W	2	0000	0001 1110	None	
SLEEP —	Go into Standby mode	1	0000	0000 0011	$\overline{TO}$ , $\overline{PD}$	
TRIS f	Load TRIS register	1	0000	0000 0fff	None	
XORLW k	Exclusive OR literal to W	1	1111	kkkk kkkk	Z	

**Note 1:** The 9th bit of the program counter will be forced to a '0' by any instruction that writes to the PC except for GOTO. See [Section 4.6 "Program Counter"](#).

- 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'.
- The instruction `TRIS f`, where  $f = 6$ , causes the contents of the W register to be written to the tri-state latches of PORTA. A '1' forces the pin to a high-impedance state and disables the output buffers.
- If this instruction is executed on the TMR0 register (and, where applicable,  $d = 1$ ), the prescaler will be cleared (if assigned to TMR0).

# PIC16F570

---

## **ADDWF**      **Add W and f**

---

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

## **ANDLW**      **AND literal with W**

---

Syntax:      [ *label* ] ANDLW   k  
Operands:     $0 \leq k \leq 255$   
Operation:    (W).AND. (k) → (W)  
Status Affected: Z  
Description:    The contents of the W register are AND'ed with the 8-bit literal 'k'. The result is placed in the W register.

## **ANDWF**      **AND W with f**

---

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

## **BCF**          **Bit Clear f**

---

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

## **BSF**          **Bit Set f**

---

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

## **BTFSC**      **Bit Test f, Skip if Clear**

---

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

<b>BTFSS</b>	<b>Bit Test f, Skip if Set</b>
Syntax:	[ <i>label</i> ] BTFSS f,b
Operands:	$0 \leq f \leq 31$ $0 \leq b < 7$
Operation:	skip if (f<b>) = 1
Status Affected:	None
Description:	If bit 'b' in register 'f' is '1', then the next instruction is skipped. If bit 'b' is '1', then the next instruction fetched during the current instruction execution, is discarded and a NOP is executed instead, making this a 2-cycle instruction.

<b>CLRW</b>	<b>Clear W</b>
Syntax:	[ <i>label</i> ] CLRW
Operands:	None
Operation:	00h → (W); 1 → Z
Status Affected:	Z
Description:	The W register is cleared. Zero bit (Z) is set.

<b>CALL</b>	<b>Subroutine Call</b>
Syntax:	[ <i>label</i> ] CALL k
Operands:	$0 \leq k \leq 255$
Operation:	(PC) + 1 → Top-of-Stack; k → PC<7:0>; (STATUS<6:5>) → PC<10:9>; 0 → PC<8>
Status Affected:	None
Description:	Subroutine call. First, return address (PC + 1) is PUSHed onto the stack. The 8-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 2-cycle instruction.

<b>CLRWDT</b>	<b>Clear Watchdog Timer</b>
Syntax:	[ <i>label</i> ] CLRWDT
Operands:	None
Operation:	00h → WDT; 0 → WDT prescaler (if assigned); 1 → $\overline{TO}$ ; 1 → PD
Status Affected:	$\overline{TO}$ , PD
Description:	The CLRWDT instruction resets the WDT. It also resets the prescaler, if the prescaler is assigned to the WDT and not Timer0. Status bits $\overline{TO}$ and PD are set.

<b>CLRF</b>	<b>Clear f</b>
Syntax:	[ <i>label</i> ] CLRF f
Operands:	$0 \leq f \leq 31$
Operation:	00h → (f); 1 → Z
Status Affected:	Z
Description:	The contents of register 'f' are cleared and the Z bit is set.

<b>COMF</b>	<b>Complement f</b>
Syntax:	[ <i>label</i> ] COMF f,d
Operands:	$0 \leq f \leq 31$ d ∈ [0,1]
Operation:	(f) → (dest)
Status Affected:	Z
Description:	The contents of register 'f' are complemented. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'.

# PIC16F570

---

<b>DECf</b>	<b>Decrement f</b>
Syntax:	[ <i>label</i> ] DECf f,d
Operands:	$0 \leq f \leq 31$ $d \in [0,1]$
Operation:	$(f) - 1 \rightarrow (\text{dest})$
Status Affected:	Z
Description:	Decrement register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'.

<b>INCF</b>	<b>Increment f</b>
Syntax:	[ <i>label</i> ] INCF f,d
Operands:	$0 \leq f \leq 31$ $d \in [0,1]$
Operation:	$(f) + 1 \rightarrow (\text{dest})$
Status Affected:	Z
Description:	The contents of register 'f' are incremented. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'.

<b>DECFSZ</b>	<b>Decrement f, Skip if 0</b>
Syntax:	[ <i>label</i> ] 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
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 2-cycle instruction.

<b>INCFSZ</b>	<b>Increment f, Skip if 0</b>
Syntax:	[ <i>label</i> ] INCFSZ f,d
Operands:	$0 \leq f \leq 31$ $d \in [0,1]$
Operation:	$(f) + 1 \rightarrow (\text{dest})$ , skip if result = 0
Status Affected:	None
Description:	The contents of register 'f' are incremented. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'. If the result is '0', then the next instruction, which is already fetched, is discarded and a NOP is executed instead making it a 2-cycle instruction.

<b>GOTO</b>	<b>Unconditional Branch</b>
Syntax:	[ <i>label</i> ] GOTO k
Operands:	$0 \leq k \leq 511$
Operation:	$k \rightarrow \text{PC}\langle 8:0 \rangle$ ; $\text{STATUS}\langle 6:5 \rangle \rightarrow \text{PC}\langle 10:9 \rangle$
Status Affected:	None
Description:	GOTO is an unconditional branch. The 9-bit immediate value is loaded into PC bits $\langle 8:0 \rangle$ . The upper bits of PC are loaded from STATUS $\langle 6:5 \rangle$ . GOTO is a 2-cycle instruction.

<b>IORLW</b>	<b>Inclusive OR literal with W</b>
Syntax:	[ <i>label</i> ] IORLW k
Operands:	$0 \leq k \leq 255$
Operation:	$(W) .\text{OR.} (k) \rightarrow (W)$
Status Affected:	Z
Description:	The contents of the W register are OR'ed with the 8-bit literal 'k'. The result is placed in the W register.

---

**IORWF**      **Inclusive OR W with f**

Syntax:      [ *label* ] IORWF f,d

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

Operation:    (W).OR. (f) → (dest)

Status Affected: Z

Description:    Inclusive OR the W register with register 'f'. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'.

---

**MOVWF**      **Move W to f**

Syntax:      [ *label* ] MOVWF f

Operands:     $0 \leq f \leq 31$

Operation:    (W) → (f)

Status Affected: None

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

---

**MOVF**        **Move f**

Syntax:      [ *label* ] MOVF f,d

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

Operation:    (f) → (dest)

Status Affected: Z

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

---

**NOP**         **No Operation**

Syntax:      [ *label* ] NOP

Operands:    None

Operation:    No operation

Status Affected: None

Description:    No operation.

---

**MOVLB**      **Move Literal to BSR**

Syntax:      [ *label* ] MOVLB k

Operands:     $0 \leq k \leq 7$

Operation:     $k \rightarrow$  (BSR)

Status Affected: None

Description:    The 3-bit literal 'k' is loaded into the BSR register.

---

**OPTION**     **Load OPTION Register**

Syntax:      [ *label* ] OPTION

Operands:    None

Operation:    (W) → OPTION

Status Affected: None

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

---

**MOVLW**      **Move Literal to W**

Syntax:      [ *label* ] MOVLW k

Operands:     $0 \leq k \leq 255$

Operation:     $k \rightarrow$  (W)

Status Affected: None

Description:    The 8-bit literal 'k' is loaded into the W register. The "don't cares" will be assembled as '0's.

---

**RETFIE**     **Return From Interrupt**

Syntax:      [ *label* ] RETFIE

Operands:    None

Operation:    TOS → PC  
                   1 → GIE

Status Affected: None

Description:    The program counter is loaded from the top of the stack (the return address).  
                   GIE bit of INTCON0 is set.  
                   This is a 2-cycle instruction.

# PIC16F570

## RETLW Return with Literal in W

Syntax: [ *label* ] RETLW *k*

Operands:  $0 \leq k \leq 255$

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

Status Affected: None

Description: The W register is loaded with the 8-bit literal 'k'. The program counter is loaded from the top of the stack (the return address). This is a 2-cycle instruction.

## RETURN Return

Syntax: [ *label* ] RETURN

Operands: None

Operation: TOS  $\rightarrow$  PC

Status Affected: None

Description: The program counter is loaded from the top of the stack (the return address). This is a 2-cycle instruction.

## RLF Rotate Left f through Carry

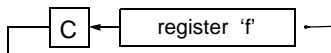
Syntax: [ *label* ] RLF *f,d*

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

Operation: See description below

Status Affected: C

Description: The contents of register 'f' are rotated 1 bit to the left through the Carry flag. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is stored back in register 'f'.



## RRF Rotate Right f through Carry

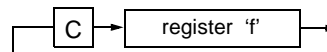
Syntax: [ *label* ] RRF *f,d*

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

Operation: See description below

Status Affected: C

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



## SLEEP Enter SLEEP Mode

Syntax: [ *label* ] SLEEP

Operands: None

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

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

Description: Time-out Status bit ( $\overline{TO}$ ) is set. The Power-down Status bit ( $\overline{PD}$ ) is cleared. RBWUF is unaffected. The WDT and its prescaler are cleared. The processor is put into Sleep mode with the oscillator stopped. See [Section 8.10 "Power-down Mode \(Sleep\)"](#) on Sleep for more details.

## SUBWF Subtract W from f

Syntax: [ *label* ] SUBWF *f,d*

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

Operation:  $(f) - (W) \rightarrow (\text{dest})$

Status Affected: C, DC, Z

Description: Subtract (2's complement method) the W register from register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'.

**SWAPF**            **Swap Nibbles in f**

---

Syntax:            [ *label* ] SWAPF f,d

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

Operation:        (f<3:0>) → (dest<7:4>);  
                    (f<7:4>) → (dest<3:0>)

Status Affected: None

Description:      The upper and lower nibbles of register 'f' are exchanged. If 'd' is '0', the result is placed in W register. If 'd' is '1', the result is placed in register 'f'.

**TRIS**             **Load TRIS Register**

---

Syntax:            [ *label* ] TRIS f

Operands:        f = 6

Operation:        (W) → TRIS register f

Status Affected: None

Description:      TRIS register 'f' (f = 6, 7 or 8) is loaded with the contents of the W register

**XORLW**           **Exclusive OR literal with W**

---

Syntax:            [ *label* ] XORLW k

Operands:         $0 \leq k \leq 255$

Operation:        (W) .XOR. k → (W)

Status Affected: Z

Description:      The contents of the W register are XOR'ed with the 8-bit literal 'k'. The result is placed in the W register.

## 14.0 DEVELOPMENT SUPPORT

The PIC<sup>®</sup> microcontrollers (MCU) and dsPIC<sup>®</sup> digital signal controllers (DSC) are supported with a full range of software and hardware development tools:

- Integrated Development Environment
  - MPLAB<sup>®</sup> X IDE Software
- Compilers/Assemblers/Linkers
  - MPLAB XC Compiler
  - MPASM<sup>™</sup> Assembler
  - MPLINK<sup>™</sup> Object Linker/  
MPLIB<sup>™</sup> Object Librarian
  - MPLAB Assembler/Linker/Librarian for  
Various Device Families
- Simulators
  - MPLAB X SIM Software Simulator
- Emulators
  - MPLAB REAL ICE<sup>™</sup> In-Circuit Emulator
- In-Circuit Debuggers/Programmers
  - MPLAB ICD 3
  - PICKit<sup>™</sup> 3
- Device Programmers
  - MPLAB PM3 Device Programmer
- Low-Cost Demonstration/Development Boards,  
Evaluation Kits and Starter Kits
- Third-party development tools

## 14.1 MPLAB X Integrated Development Environment Software

The MPLAB X IDE is a single, unified graphical user interface for Microchip and third-party software, and hardware development tool that runs on Windows<sup>®</sup>, Linux and Mac OS<sup>®</sup> X. Based on the NetBeans IDE, MPLAB X IDE is an entirely new IDE with a host of free software components and plug-ins for high-performance application development and debugging. Moving between tools and upgrading from software simulators to hardware debugging and programming tools is simple with the seamless user interface.

With complete project management, visual call graphs, a configurable watch window and a feature-rich editor that includes code completion and context menus, MPLAB X IDE is flexible and friendly enough for new users. With the ability to support multiple tools on multiple projects with simultaneous debugging, MPLAB X IDE is also suitable for the needs of experienced users.

Feature-Rich Editor:

- Color syntax highlighting
- Smart code completion makes suggestions and provides hints as you type
- Automatic code formatting based on user-defined rules
- Live parsing

User-Friendly, Customizable Interface:

- Fully customizable interface: toolbars, toolbar buttons, windows, window placement, etc.
- Call graph window

Project-Based Workspaces:

- Multiple projects
- Multiple tools
- Multiple configurations
- Simultaneous debugging sessions

File History and Bug Tracking:

- Local file history feature
- Built-in support for Bugzilla issue tracker

## 14.2 MPLAB XC Compilers

The MPLAB XC Compilers are complete ANSI C compilers for all of Microchip's 8, 16, and 32-bit MCU and DSC devices. These compilers provide powerful integration capabilities, superior code optimization and ease of use. MPLAB XC Compilers run on Windows, Linux or MAC OS X.

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

The free MPLAB XC Compiler editions support all devices and commands, with no time or memory restrictions, and offer sufficient code optimization for most applications.

MPLAB XC Compilers include an assembler, linker and utilities. 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. MPLAB XC Compiler uses the assembler to produce its object file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command-line interface
- Rich directive set
- Flexible macro language
- MPLAB X IDE compatibility

## 14.3 MPASM Assembler

The MPASM Assembler is a full-featured, universal macro assembler for PIC10/12/16/18 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 X IDE projects
- User-defined macros to streamline assembly code
- Conditional assembly for multipurpose source files
- Directives that allow complete control over the assembly process

## 14.4 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler. 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/librarian 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

## 14.5 MPLAB Assembler, Linker and Librarian for Various Device Families

MPLAB Assembler produces relocatable machine code from symbolic assembly language for PIC24, PIC32 and dsPIC DSC devices. MPLAB XC Compiler uses the assembler to produce its object file. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command-line interface
- Rich directive set
- Flexible macro language
- MPLAB X IDE compatibility

## 14.6 MPLAB X SIM Software Simulator

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

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

## 14.7 MPLAB REAL ICE In-Circuit Emulator System

The MPLAB REAL ICE In-Circuit Emulator System is Microchip's next generation high-speed emulator for Microchip Flash DSC and MCU devices. It debugs and programs all 8, 16 and 32-bit MCU, and DSC devices with the easy-to-use, powerful graphical user interface of the MPLAB X IDE.

The emulator is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with either a connector compatible with in-circuit debugger systems (RJ-11) or with the new high-speed, noise tolerant, Low-Voltage Differential Signal (LVDS) interconnection (CAT5).

The emulator is field upgradable through future firmware downloads in MPLAB X IDE. MPLAB REAL ICE offers significant advantages over competitive emulators including full-speed emulation, run-time variable watches, trace analysis, complex breakpoints, logic probes, a ruggedized probe interface and long (up to three meters) interconnection cables.

## 14.8 MPLAB ICD 3 In-Circuit Debugger System

The MPLAB ICD 3 In-Circuit Debugger System is Microchip's most cost-effective, high-speed hardware debugger/programmer for Microchip Flash DSC and MCU devices. It debugs and programs PIC Flash microcontrollers and dsPIC DSCs with the powerful, yet easy-to-use graphical user interface of the MPLAB IDE.

The MPLAB ICD 3 In-Circuit Debugger probe is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with a connector compatible with the MPLAB ICD 2 or MPLAB REAL ICE systems (RJ-11). MPLAB ICD 3 supports all MPLAB ICD 2 headers.

## 14.9 PICkit 3 In-Circuit Debugger/Programmer

The MPLAB PICkit 3 allows debugging and programming of PIC and dsPIC Flash microcontrollers at a most affordable price point using the powerful graphical user interface of the MPLAB IDE. The MPLAB PICkit 3 is connected to the design engineer's PC using a full-speed USB interface and can be connected to the target via a Microchip debug (RJ-11) connector (compatible with MPLAB ICD 3 and MPLAB REAL ICE). The connector uses two device I/O pins and the Reset line to implement in-circuit debugging and In-Circuit Serial Programming™ (ICSP™).

## 14.10 MPLAB PM3 Device Programmer

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

## 14.11 Demonstration/Development Boards, Evaluation Kits, and Starter Kits

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

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

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

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

Also available are starter kits that contain everything needed to experience the specified device. This usually includes a single application and debug capability, all on one board.

Check the Microchip web page ([www.microchip.com](http://www.microchip.com)) for the complete list of demonstration, development and evaluation kits.

## 14.12 Third-Party Development Tools

Microchip also offers a great collection of tools from third-party vendors. These tools are carefully selected to offer good value and unique functionality.

- Device Programmers and Gang Programmers from companies, such as SoftLog and CCS
- Software Tools from companies, such as Gimpel and Trace Systems
- Protocol Analyzers from companies, such as Saleae and Total Phase
- Demonstration Boards from companies, such as MikroElektronika, Digilent® and Olimex
- Embedded Ethernet Solutions from companies, such as EZ Web Lynx, WIZnet and IPLogika®

# PIC16F570

## 15.0 ELECTRICAL CHARACTERISTICS

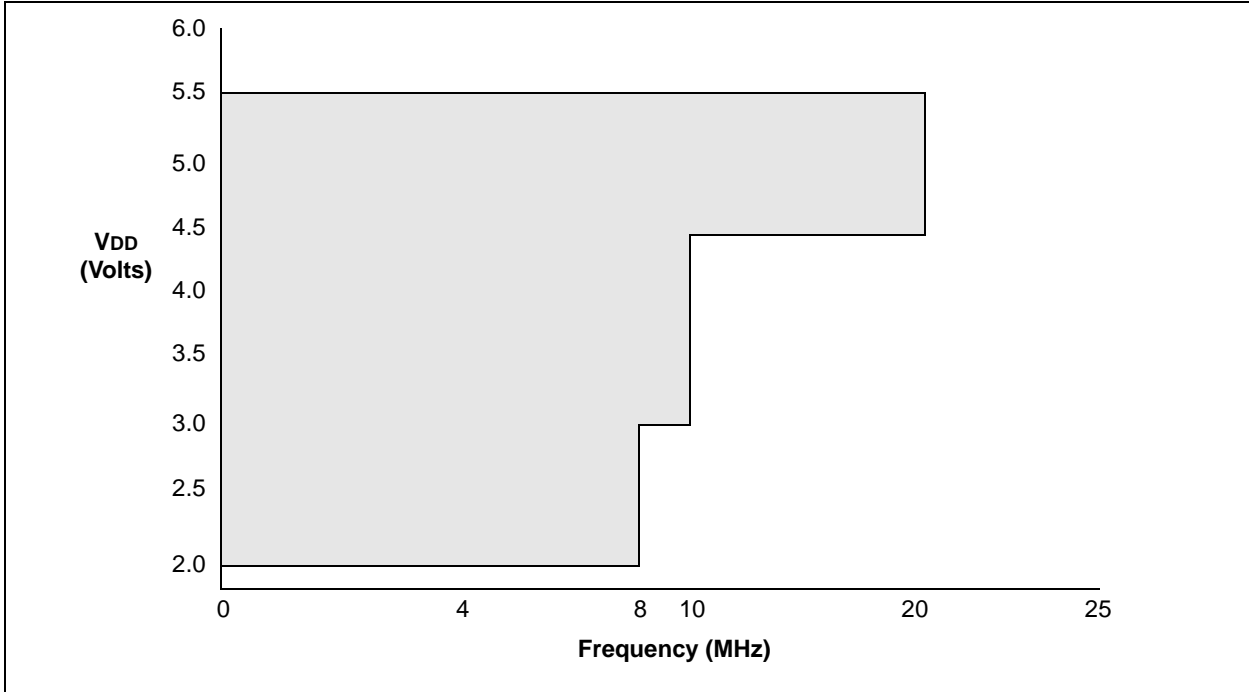
### Absolute Maximum Ratings<sup>(†)</sup>

Ambient temperature under bias .....	-40°C to +125°C
Storage temperature .....	-65°C to +150°C
Voltage on V <sub>DD</sub> with respect to V <sub>SS</sub> .....	0 to +6.5V
Voltage on $\overline{\text{MCLR}}$ with respect to V <sub>SS</sub> .....	0 to +13.5V
Voltage on all other pins with respect to V <sub>SS</sub> .....	-0.3V to (V <sub>DD</sub> + 0.3V)
Total power dissipation <sup>(1)</sup> .....	700 mW
Max. current out of V <sub>SS</sub> pin .....	200 mA
Max. current into V <sub>DD</sub> pin .....	150 mA
Input clamp current, I <sub>IK</sub> (V <sub>I</sub> < 0 or V <sub>I</sub> > V <sub>DD</sub> ).....	±20 mA
Output clamp current, I <sub>OK</sub> (V <sub>O</sub> < 0 or V <sub>O</sub> > V <sub>DD</sub> ) .....	±20 mA
Max. output current sunk by any I/O pin.....	25 mA
Max. output current sourced by any I/O pin .....	25 mA
Max. output current sourced by I/O port .....	75 mA
Max. output current sunk by I/O port .....	75 mA

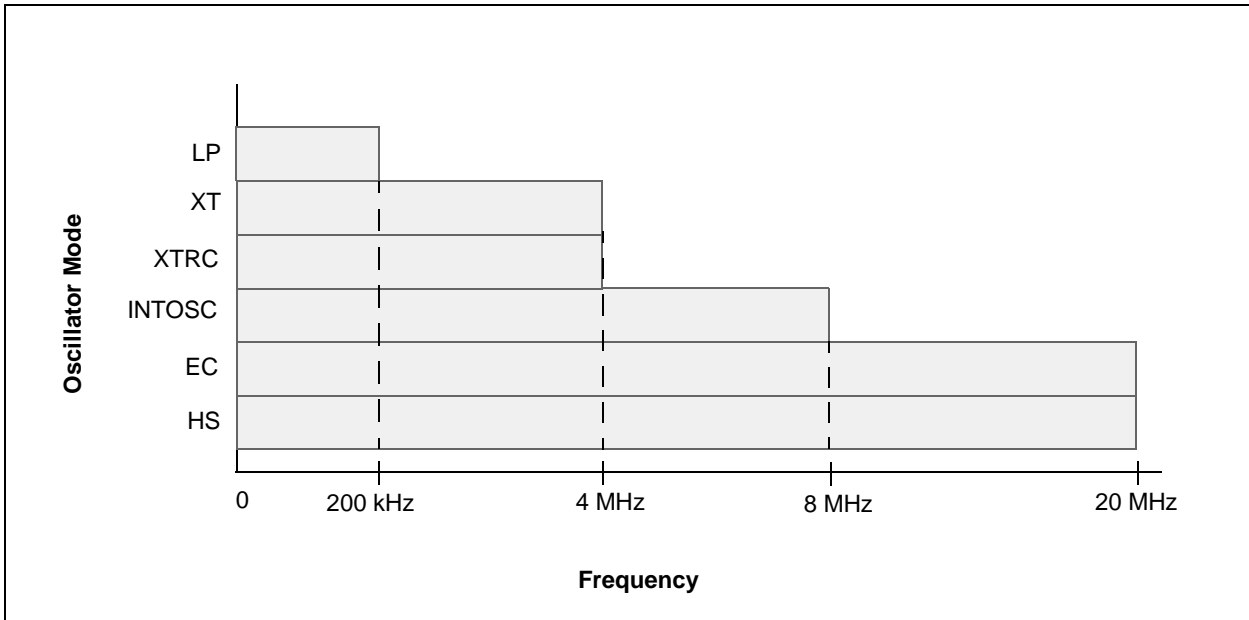
**Note 1:** Power dissipation is calculated as follows:  $P_{DIS} = V_{DD} \times \{I_{DD} - \sum I_{OH}\} + \sum \{(V_{DD} - V_{OH}) \times I_{OH}\} + \sum (V_{OL} \times I_{OL})$

<sup>†</sup>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 15-1: PIC16F570 VOLTAGE-FREQUENCY GRAPH,  $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$**



**FIGURE 15-2: MAXIMUM OSCILLATOR FREQUENCY TABLE**



# PIC16F570

## 15.1 DC Characteristics: PIC16F570 (Industrial)

DC Characteristics			Standard Operating Conditions (unless otherwise specified) Operating Temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (industrial)				
Param. No.	Sym.	Characteristic	Min.	Typ. <sup>(1)</sup>	Max.	Units	Conditions
D001	VDD	Supply Voltage	2.0	—	5.5	V	See Figure 15-1
D002	VDR	RAM Data Retention Voltage <sup>(2)</sup>	—	1.5*	—	V	Device in Sleep mode
D003	VPOR	VDD Start Voltage to ensure Power-on Reset	—	VSS	—	V	See Section 8.5 “Power-on Reset (POR)” for details
D004	SVDD	VDD Rise Rate to ensure Power-on Reset	0.05*	—	—	V/ms	See Section 8.5 “Power-on Reset (POR)” for details
D005	IDDP	Supply Current During Prog./ Erase	—	1.0*	—	mA	VDD = 5.0V
D010	IDD	Supply Current <sup>(3,4,6)</sup>	—	195	300	$\mu\text{A}$	FOSC = 4 MHz, VDD = 2.0V
			—	505	750	$\mu\text{A}$	FOSC = 4 MHz, VDD = 5.0V
			—	310	500	$\mu\text{A}$	FOSC = 8 MHz, VDD = 2.0V
			—	880	1300	$\mu\text{A}$	FOSC = 8 MHz, VDD = 5.0V
			—	1900	2300	$\mu\text{A}$	FOSC = 20 MHz, VDD = 5.0V
D020	IPD	Power-down Current <sup>(5)</sup>	—	0.05	1.2	$\mu\text{A}$	VDD = 2.0V
			—	0.35	3.0	$\mu\text{A}$	VDD = 5.0V
D021	IBOR	BOR Current <sup>(5)</sup>	—	3.5	7.0	$\mu\text{A}$	VDD = 3.0V
			—	4.0	9.0	$\mu\text{A}$	VDD = 5.0V
D022	IWDT	WDT Current <sup>(5)</sup>	—	0.5	3.0	$\mu\text{A}$	VDD = 2.0V
			—	8.0	16.0	$\mu\text{A}$	VDD = 5.0V
D023	ICMP	Comparator Current <sup>(5)</sup>	—	15	26	$\mu\text{A}$	VDD = 2.0V (per comparator)
			—	60	78	$\mu\text{A}$	VDD = 5.0V (per comparator)
D024	ICVREF	CVREF Current <sup>(5)</sup>	—	30	70	$\mu\text{A}$	VDD = 2.0V (high range)
			—	75	125	$\mu\text{A}$	VDD = 5.0V (high range)
D025	IVFIR	Internal 0.6V Fixed Voltage Reference Current <sup>(5)</sup>	—	100	125	$\mu\text{A}$	VDD = 2.0V (reference and 1 comparator enabled)
			—	175	205	$\mu\text{A}$	VDD = 5.0V (reference and 1 comparator enabled)
D026	IAD2	A/D Current	—	0.5	2.0	$\mu\text{A}$	2.0V, No conversion in progress
			—	0.8	4.0	$\mu\text{A}$	5.0V, No conversion in progress
D027	IOPA	Op Amp Current <sup>(5)</sup>	—	310	425	$\mu\text{A}$	VDD = 2.0V
			—	330	475	$\mu\text{A}$	VDD = 5.0V

\* These parameters are characterized but not tested.

**Note 1:** Data in the Typical (“Typ.”) column is based on characterization results at 25°C. This data is for design guidance only and is not tested.

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

**3:** The supply current is mainly a function of the operating voltage and frequency. Other factors such as bus loading, oscillator type, bus rate, internal code execution pattern and temperature also have an impact on the current consumption.

**4:** 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, MCLR = VDD; WDT enabled/disabled as specified.

**5:** For standby current measurements, the conditions are the same as IDD, except that the device is in Sleep mode. If a module current is listed, the current is for that specific module enabled and the device in Sleep.

**6:** Does not include current through REXT. The current through the resistor can be estimated by the formula:  $I = V_{DD}/2R_{EXT}$  (mA) with REXT in k $\Omega$ .

## 15.2 DC Characteristics: PIC16F570 (Extended)

DC Characteristics			Standard Operating Conditions (unless otherwise specified) Operating Temperature $40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ (extended)				
Param. No.	Sym.	Characteristic	Min.	Typ. <sup>(1)</sup>	Max.	Units	Conditions
D001	VDD	<b>Supply Voltage</b>	2.0	—	5.5	V	See <a href="#">Figure 15-1</a>
D002	VDR	<b>RAM Data Retention Voltage<sup>(2)</sup></b>	—	1.5*	—	V	Device in Sleep mode
D003	VPOR	<b>VDD Start Voltage</b> to ensure Power-on Reset	—	VSS	—	V	See <a href="#">Section 8.5 “Power-on Reset (POR)”</a> for details.
D004	SVDD	<b>VDD Rise Rate</b> to ensure Power-on Reset	0.05*	—	—	V/ms	See <a href="#">Section 8.5 “Power-on Reset (POR)”</a> for details.
D005	IDDP	Supply Current During Prog./Erase	—	1.0*	—	mA	VDD = 5.0V
D010	IDD	<b>Supply Current<sup>(3,4,6)</sup></b>	—	195	300	$\mu\text{A}$	FOSC = 4 MHz, VDD = 2.0V
			—	505	750	$\mu\text{A}$	FOSC = 4 MHz, VDD = 5.0V
			—	310	500	$\mu\text{A}$	FOSC = 8 MHz, VDD = 2.0V
			—	880	1300	$\mu\text{A}$	FOSC = 8 MHz, VDD = 5.0V
			—	1900	2300	$\mu\text{A}$	FOSC = 20 MHz, VDD = 5.0V
D020	IPD	<b>Power-down Current<sup>(5)</sup></b>	—	0.1	9.0	$\mu\text{A}$	VDD = 2.0V
			—	0.35	15.0	$\mu\text{A}$	VDD = 5.0V
D021	IBOR	<b>BOR Current<sup>(5)</sup></b>	—	3.5	10	$\mu\text{A}$	VDD = 3.0V
			—	4.0	12	$\mu\text{A}$	VDD = 5.0V
D022	IWDT	<b>WDT Current<sup>(5)</sup></b>	—	1.0	18	$\mu\text{A}$	VDD = 2.0V
			—	8.0	22	$\mu\text{A}$	VDD = 5.0V
D023	ICMP	<b>Comparator Current<sup>(5)</sup></b>	—	15	28	$\mu\text{A}$	VDD = 2.0V (per comparator)
			—	60	92	$\mu\text{A}$	VDD = 5.0V (per comparator)
D024	ICVREF	<b>CVREF Current<sup>(5)</sup></b>	—	30	75	$\mu\text{A}$	VDD = 2.0V (high range)
			—	75	135	$\mu\text{A}$	VDD = 5.0V (high range)
D025	IVFIR	<b>Internal 0.6V Fixed Voltage Reference Current<sup>(5)</sup></b>	—	100	135	$\mu\text{A}$	VDD = 2.0V (reference and 1 comparator enabled)
			—	175	220	$\mu\text{A}$	VDD = 5.0V (reference and 1 comparator enabled)
D026	IAD2	<b>A/D Current</b>	—	0.5	10.0	$\mu\text{A}$	2.0V, No conversion in progress
			—	0.8	16.0	$\mu\text{A}$	5.0V, No conversion in progress
D027	IOPA	<b>Op Amp Current<sup>(5)</sup></b>	—	310	450	$\mu\text{A}$	VDD = 2.0V
			—	330	505	$\mu\text{A}$	VDD = 5.0V

\* These parameters are characterized but not tested.

**Note 1:** Data in the Typical (“Typ.”) column is based on characterization results at 25°C. This data is for design guidance only and is not tested.

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

**3:** The supply current is mainly a function of the operating voltage and frequency. Other factors such as bus loading, oscillator type, bus rate, internal code execution pattern and temperature also have an impact on the current consumption.

**4:** 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, MCLR = VDD; WDT enabled/disabled as specified.

**5:** For standby current measurements, the conditions are the same as IDD, except that the device is in Sleep mode. If a module current is listed, the current is for that specific module enabled and the device in Sleep.

**6:** Does not include current through REXT. The current through the resistor can be estimated by the formula:  $I = VDD/2REXT$  (mA) with REXT in k $\Omega$ .

# PIC16F570

**TABLE 15-1: DC CHARACTERISTICS: PIC16F570 (INDUSTRIAL, EXTENDED)**

DC CHARACTERISTICS		Standard Operating Conditions (unless otherwise specified) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (industrial) $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ (extended) Operating voltage $V_{DD}$ range is described in <a href="#">Section 15.1 "DC Characteristics: PIC16F570 (Industrial)"</a> .					
Param. No.	Sym.	Characteristic	Min.	Typ.†	Max.	Units	Conditions
	VIL	<b>Input Low Voltage</b>					
D030		I/O ports with TTL buffer	Vss	—	0.8V	V	For all $4.5 \leq V_{DD} \leq 5.5\text{V}$
D030A			Vss	—	0.15VDD	V	Otherwise
D031		with Schmitt Trigger buffer	Vss	—	0.15VDD	V	
D032		MCLR, T0CKI	Vss	—	0.15VDD	V	
D033		OSC1 (EXTRC mode), (EC mode)	Vss	—	0.15VDD	V	<b>(Note 1)</b>
D033		OSC1 (HS mode)	Vss	—	0.3VDD	V	
D033		OSC1 (XT and LP modes)	Vss	—	0.3	V	
	VIH	<b>Input High Voltage</b>					
D040		I/O ports with TTL buffer	2.0	—	VDD	V	$4.5 \leq V_{DD} \leq 5.5\text{V}$
D040A			0.25VDD + 0.8VDD	—	VDD	V	Otherwise
D041		with Schmitt Trigger buffer	0.85VDD	—	VDD	V	For entire VDD range
D042		MCLR, T0CKI	0.85VDD	—	VDD	V	
D042A		OSC1 (EXTRC mode), (EC mode)	0.85VDD	—	VDD	V	<b>(Note 1)</b>
D042A		OSC1 (HS mode)	0.7VDD	—	VDD	V	
D043		OSC1 (XT and LP modes)	1.6	—	VDD	V	
D070	IPUR	<b>PORTB and MCLR weak pull-up current<sup>(4)</sup></b>	50	250	400	$\mu\text{A}$	$V_{DD} = 5\text{V}$ , $V_{PIN} = V_{SS}$
	IIL	<b>Input Leakage Current<sup>(2,3)</sup></b>					
D060		I/O ports	—	—	$\pm 1$	$\mu\text{A}$	$V_{SS} \leq V_{PIN} \leq V_{DD}$ , Pin at high-impedance
D061		MCLR	—	$\pm 0.7$	$\pm 5$	$\mu\text{A}$	$V_{SS} \leq V_{PIN} \leq V_{DD}$
D063		OSC1	—	—	$\pm 5$	$\mu\text{A}$	$V_{SS} \leq V_{PIN} \leq V_{DD}$ , XT, HS and LP osc configuration
	VOL	<b>Output Low Voltage</b>					
D080		I/O ports/CLKOUT	—	—	0.6	V	IOL = 8.5 mA, $V_{DD} = 4.5\text{V}$ , $-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$
D080A			—	—	0.6	V	IOL = 7.0 mA, $V_{DD} = 4.5\text{V}$ , $-40^{\circ}\text{C}$ to $+125^{\circ}\text{C}$
D083		OSC2	—	—	0.6	V	IOL = 1.6 mA, $V_{DD} = 4.5\text{V}$ , $-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$
D083A			—	—	0.6	V	IOL = 1.2 mA, $V_{DD} = 4.5\text{V}$ , $-40^{\circ}\text{C}$ to $+125^{\circ}\text{C}$

† Data in "Typ." column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

- Note 1:** In EXTRC oscillator configuration, the OSC1/CLKIN pin is a Schmitt Trigger input. It is not recommended that the PIC16F570 be driven with external clock in RC mode.
- 2:** The leakage current on the MCLR pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.
- 3:** Negative current is defined as coming out of the pin.
- 4:** This spec applies to all weak pull-up devices, including the weak pull-up found on MCLR.

**TABLE 15-1: DC CHARACTERISTICS: PIC16F570 (INDUSTRIAL, EXTENDED) (CONTINUED)**

DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise specified)				
			Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (industrial) $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ (extended)				
			Operating voltage VDD range is described in <a href="#">Section 15.1</a> “DC Characteristics: PIC16F570 (Industrial)”.				
Param. No.	Sym.	Characteristic	Min.	Typ.†	Max.	Units	Conditions
<b>Output High Voltage</b>							
D090	VOH	I/O ports/CLKOUT	$V_{DD} - 0.7$	—	—	V	$I_{OH} = -3.0\text{ mA}$ , $V_{DD} = 4.5\text{V}$ , $-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$
D090A			$V_{DD} - 0.7$	—	—	V	$I_{OH} = -2.5\text{ mA}$ , $V_{DD} = 4.5\text{V}$ , $-40^{\circ}\text{C}$ to $+125^{\circ}\text{C}$
D092		OSC2	$V_{DD} - 0.7$	—	—	V	$I_{OH} = -1.3\text{ mA}$ , $V_{DD} = 4.5\text{V}$ , $-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$
D092A			$V_{DD} - 0.7$	—	—	V	$I_{OH} = -1.0\text{ mA}$ , $V_{DD} = 4.5\text{V}$ , $-40^{\circ}\text{C}$ to $+125^{\circ}\text{C}$
<b>Capacitive Loading Specs on Output Pins</b>							
D100	COSC2	OSC2 pin	—	—	15	pF	In XT, HS and LP modes when external clock is used to drive OSC1.
D101	CIO	All I/O pins and OSC2	—	—	50	pF	
<b>Flash Data Memory</b>							
D120	ED	Byte endurance	100K	1M	—	E/W	$-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$
D120A	ED	Byte endurance	10K	100K	—	E/W	$+85^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$
D121	VDRW	VDD for read/write	V <sub>MIN</sub>	—	5.5	V	

† Data in “Typ.” column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

- Note 1:** In EXTRC oscillator configuration, the OSC1/CLKIN pin is a Schmitt Trigger input. It is not recommended that the PIC16F570 be driven with external clock in RC mode.
- Note 2:** The leakage current on the MCLR pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.
- Note 3:** Negative current is defined as coming out of the pin.
- Note 4:** This spec applies to all weak pull-up devices, including the weak pull-up found on  $\overline{\text{MCLR}}$ .

# PIC16F570

**TABLE 15-2: COMPARATOR SPECIFICATIONS**

Comparator Specifications	Standard Operating Conditions (unless otherwise stated)					
	Operating temperature -40°C to 125°C					
Characteristics	Sym.	Min.	Typ.	Max.	Units	Comments
Input offset voltage	VOS	—	± 5.0	±10.0	mV	
Input common mode voltage*	VCM	0	—	VDD – 1.5	V	
CMRR*	CMRR	55	—	—	db	
Response Time <sup>(1)*</sup>	TRT	—	150	—	ns	
Comparator Mode Change to Output Valid*	TMC2COV	—	—	10	µs	

\* These parameters are characterized but not tested.

**Note 1:** Response time measured with one comparator input at  $(V_{DD} - 1.5)/2$  while the other input transitions from VSS to VDD – 1.5V.

**TABLE 15-3: COMPARATOR VOLTAGE REFERENCE (CVREF) SPECIFICATIONS**

Sym.	Characteristics	Min.	Typ.	Max.	Units	Comments
CVRES	Resolution	—	VDD/24*	—	LSb	Low Range (VRR = 1)
		—	VDD/32	—	LSb	High Range (VRR = 0)
	Absolute Accuracy <sup>(2)</sup>	—	—	±1/2*	LSb	Low Range (VRR = 1)
		—	—	±1/2*	LSb	High Range (VRR = 0)
Unit Resistor Value (R)	—	2K*	—	Ω		
Settling Time <sup>(1)</sup>	—	—	10*	µs		

\* These parameters are characterized but not tested.

**Note 1:** Settling time measured while VRR = 1 and VR<3:0> transitions from 0000 to 1111.

**Note 2:** Do not use reference externally when VDD < 2.7V. Under this condition, reference should only be used with comparator Voltage Common mode observed.

**TABLE 15-4: FIXED INPUT REFERENCE SPECIFICATION**

Input Reference Specifications	Standard Operating Conditions (unless otherwise stated)					
	Operating temperature -40°C to 125°C					
Characteristics	Sym.	Min.	Typ.	Max.	Units	Comments
Absolute Accuracy	VFIR	0.5	0.60	0.7	V	

**TABLE 15-5: A/D CONVERTER CHARACTERISTICS**

			Standard Operating Conditions (unless otherwise stated)				
			Operating temperature: 25°C				
Param. No.	Sym.	Characteristic	Min.	Typ.†	Max.	Units	Conditions
A01	NR	Resolution	—	—	8	bit	
A03	EINL	Integral Error	—	—	±1.5	LSb	VDD = 5.0V
A04	EDNL	Differential Error	—	—	-1 < EDNL ≤ 1.7	LSb	No missing codes VDD = 5.0V
A05	EFS	Full Scale Range	2.0*	—	5.5*	V	
A06	E0FF	Offset Error	—	—	±1.5	LSb	VDD = 5.0V
A07	EGN	Gain Error	-0.7	—	±2.2	LSb	VDD = 5.0V
A10	—	Monotonicity	—	guaranteed <sup>(1)</sup>	—	—	VSS ≤ VAIN ≤ VDD
A25*	VAIN	Analog Input Voltage	VSS	—	VDD	V	
A30*	ZAIN	Recommended Impedance of Analog Voltage Source	—	—	10	KΩ	

\* These parameters are characterized but not tested.

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

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

# PIC16F570

## 15.3 Timing Parameter Symbolology and Load Conditions

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

1. TppS2ppS
2. TppS

<b>T</b>	F Frequency	T Time
----------	-------------	--------

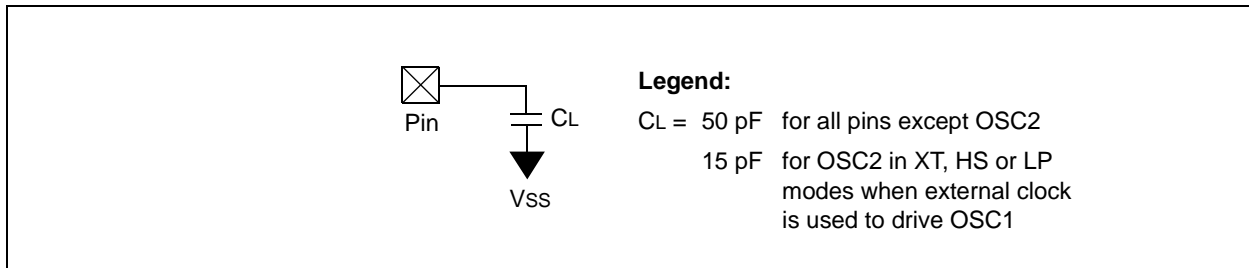
Lowercase subscripts (pp) and their meanings:

<b>pp</b>	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

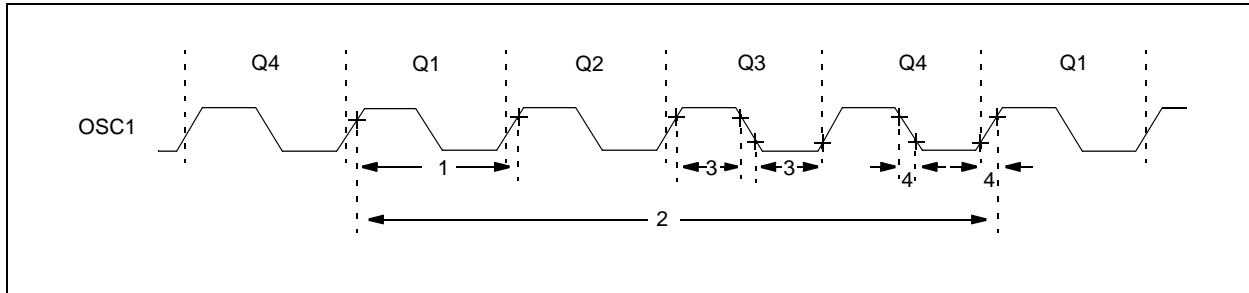
Uppercase letters and their meanings:

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

**FIGURE 15-3: LOAD CONDITIONS**



**FIGURE 15-4: EXTERNAL CLOCK TIMING**



**TABLE 15-6: EXTERNAL CLOCK TIMING REQUIREMENTS**

AC Characteristics			Standard Operating Conditions (unless otherwise specified) Operating Temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (industrial), $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ (extended) Operating voltage $V_{DD}$ range is described in <a href="#">Section 15.1 “DC Characteristics: PIC16F570 (Industrial)”</a> .				
Param. No.	Sym.	Characteristic	Min.	Typ. <sup>(1)</sup>	Max.	Units	Conditions
1A	FOSC	External CLKIN Frequency <sup>(2)</sup>	DC	—	4	MHz	XT Oscillator
			DC	—	20	MHz	HS/EC Oscillator
			DC	—	200	kHz	LP Oscillator
	Oscillator Frequency <sup>(2)</sup>	DC	—	4	MHz	EXTRC Oscillator	
		0.1	—	4	MHz	XT Oscillator	
		4	—	20	MHz	HS/EC Oscillator	
DC		—	200	kHz	LP Oscillator		
1	Tosc	External CLKIN Period <sup>(2)</sup>	250	—	—	ns	XT Oscillator
			50	—	—	ns	HS/EC Oscillator
			5	—	—	μs	LP Oscillator
	Oscillator Period <sup>(2)</sup>	250	—	—	ns	EXTRC Oscillator	
		250	—	10,000	ns	XT Oscillator	
		50	—	250	ns	HS/EC Oscillator	
5		—	—	μs	LP Oscillator		
2	Tcy	Instruction Cycle Time	200	4/FOSC	DC	ns	
3	TosL, TosH	Clock in (OSC1) Low or High Time	50*	—	—	ns	XT Oscillator
			2*	—	—	μs	LP Oscillator
			10*	—	—	ns	HS/EC Oscillator
4	TosR, TosF	Clock in (OSC1) Rise or Fall Time	—	—	25*	ns	XT Oscillator
			—	—	50*	ns	LP Oscillator
			—	—	15*	ns	HS/EC Oscillator

\* These parameters are characterized but not tested.

**Note 1:** Data in the Typical (“Typ.”) column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**2:** 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.

# PIC16F570

**TABLE 15-7: CALIBRATED INTERNAL RC FREQUENCIES**

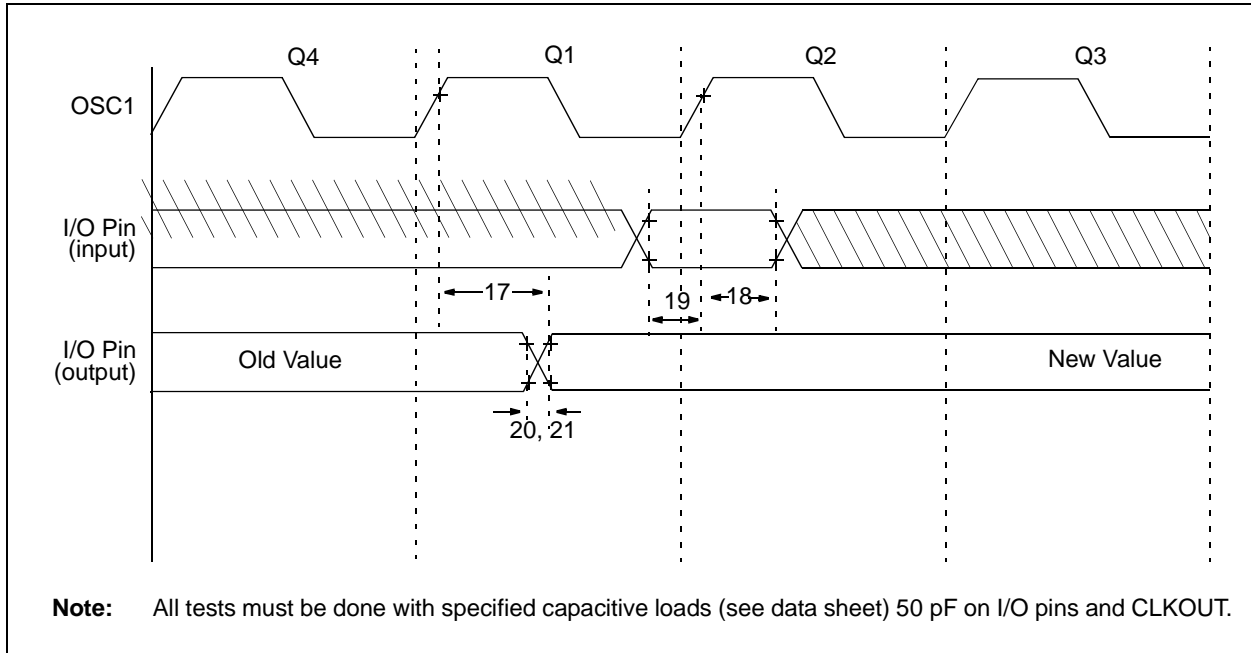
AC Characteristics		Standard Operating Conditions (unless otherwise specified) Operating Temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (industrial), $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ (extended) Operating voltage $V_{DD}$ range is described in <a href="#">Section 15.1 “DC Characteristics: PIC16F570 (Industrial)”</a> .						
Param. No.	Sym.	Characteristic	Freq. Tolerance	Min.	Typ.†	Max.	Units	Conditions
F10	FOSC	Internal Calibrated INTOSC Frequency <sup>(1)</sup>	$\pm 1\%$	7.92	8.00	8.08	MHz	3.5V, 25C
			$\pm 2\%$	7.84	8.00	8.16	MHz	$2.5\text{V} \leq V_{DD} \leq 5.5\text{V}$ $0^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$
			$\pm 5\%$	7.60	8.00	8.40	MHz	$2.0\text{V} \leq V_{DD} \leq 5.5\text{V}$ $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (Ind.) $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ (Ext.)

\* These parameters are characterized but not tested.

† Data in the Typical (“Typ.”) column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** To ensure these oscillator frequency tolerances,  $V_{DD}$  and  $V_{SS}$  must be capacitively decoupled as close to the device as possible. 0.1 uF and 0.01 uF values in parallel are recommended.

**FIGURE 15-5: I/O TIMING**



**TABLE 15-8: TIMING REQUIREMENTS**

AC Characteristics		Standard Operating Conditions (unless otherwise specified)				
		Operating Temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (industrial) $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ (extended)				
		Operating voltage $V_{DD}$ range is described in <a href="#">Section 15.1 "DC Characteristics: PIC16F570 (Industrial)"</a> .				
Param. No.	Sym.	Characteristic	Min.	Typ. <sup>(1)</sup>	Max.	Units
17	TosH2ioV	OSC1 $\uparrow$ (Q1 cycle) to Port Out Valid <sup>(2,3)</sup>	—	—	100*	ns
18	TosH2ioI	OSC1 $\uparrow$ (Q2 cycle) to Port Input Invalid (I/O in hold time) <sup>(2)</sup>	50*	—	—	ns
19	TioV2osH	Port Input Valid to OSC1 $\uparrow$ (I/O in setup time)	20*	—	—	ns
20	TioR	Port Output Rise Time <sup>(3)</sup>	—	10	50**	ns
21	TioF	Port Output Fall Time <sup>(3)</sup>	—	10	50**	ns

\* These parameters are characterized but not tested.

\*\* These parameters are design targets and are not tested.

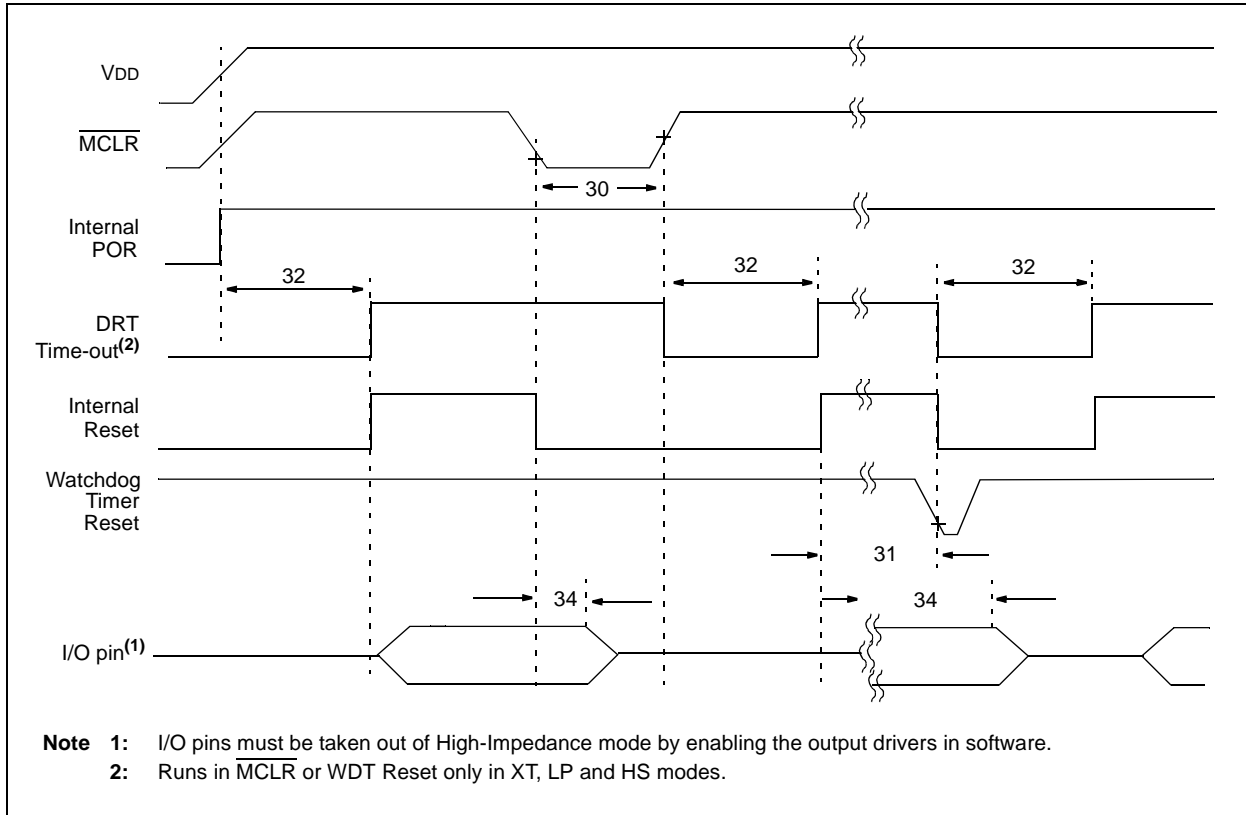
**Note 1:** Data in the Typical ("Typ.") column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**2:** Measurements are taken in EXTRC mode.

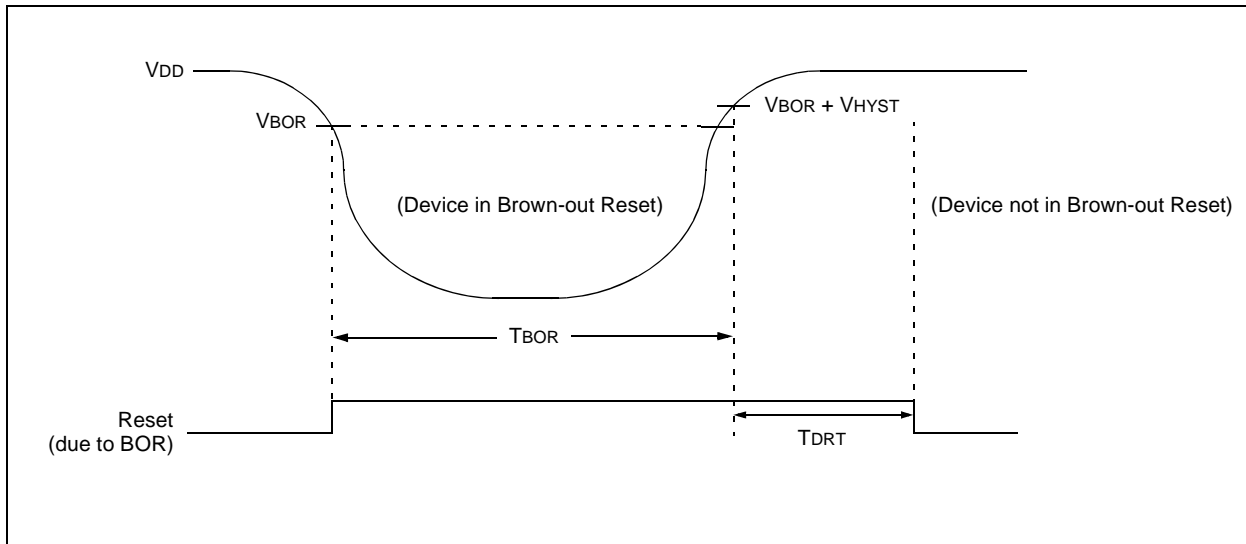
**3:** See [Figure 15-3](#) for loading conditions.

# PIC16F570

**FIGURE 15-6: RESET, WATCHDOG TIMER AND DEVICE RESET TIMER TIMING**



**FIGURE 15-7: BROWN-OUT RESET TIMING AND CHARACTERISTICS**



**TABLE 15-9: BOR, POR, WATCHDOG TIMER AND DEVICE RESET TIMER**

AC Characteristics		Standard Operating Conditions (unless otherwise specified) Operating Temperature -40°C ≤ TA ≤ +85°C (industrial) -40°C ≤ TA ≤ +125°C (extended) Operating voltage VDD range is described in <a href="#">Section 15.1 “DC Characteristics: PIC16F570 (Industrial)”</a> .					
Param. No.	Sym.	Characteristic	Min.	Typ. <sup>(1)</sup>	Max.	Units	Conditions
30	TMCL	MCLR Pulse Width (low)	2000*	—	—	ns	VDD = 5.0V
31	TWDT	Watchdog Timer Time-out Period (no prescaler)	9*	18*	30*	ms	VDD = 5.0V (Industrial)
			9*	18	40*	ms	VDD = 5.0V (Extended)
32	TDRT	Device Reset Timer Period	9*	18*	30*	ms	VDD = 5.0V (Industrial)
			9*	18	40*	ms	VDD = 5.0V (Extended)
34	TIOZ	I/O High-impedance from MCLR low	—	—	2000*	ns	
35	VBOR	Brown-out Reset Voltage	1.95	—	2.25	V	(Note 2)
36*	VHYST	Brown-out Reset Hysteresis	—	50	—	mV	
37*	TBOR	Brown-out Reset Minimum Detection Period	100	—	—	μs	VDD ≤ VBOR

\* These parameters are characterized but not tested.

**Note 1:** Data in the Typical (“Typ.”) column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**2:** To ensure these voltage tolerances, VDD and VSS must be capacitively decoupled as close to the device as possible. 0.1 μF and 0.01 μF values in parallel are recommended.

**TABLE 15-10: DRT (DEVICE RESET TIMER PERIOD)**

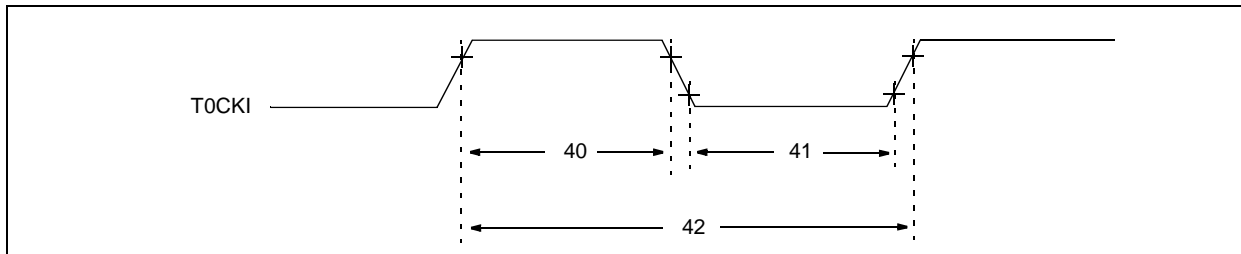
Oscillator Configuration	POR Reset	Subsequent Resets
IntRC, ExtRC, and EC	10 μs (typical) + 18 ms (DRTEN = 1)	10 μs (typical) + 18 ms (DRTEN = 1)
XT, HS and LP	18 ms (typical)	18 ms (typical)

**TABLE 15-11: PULL-UP RESISTOR RANGES**

VDD (Volts)	Temperature (°C)	Min.	Typ.	Max.	Units
<b>RB0-RB7</b>					
2.0	-40	73K	105K	186K	Ω
	25	73K	113K	187K	Ω
	85	82K	123K	190K	Ω
	125	86K	132k	190K	Ω
5.5	-40	15K	21K	33K	Ω
	25	15K	22K	34K	Ω
	85	19K	26k	35K	Ω
	125	23K	29K	35K	Ω
<b>MCLR</b>					
2.0	-40	63K	81K	96K	Ω
	25	77K	93K	116K	Ω
	85	82K	96k	116K	Ω
	125	86K	100K	119K	Ω
5.5	-40	16K	20k	22K	Ω
	25	16K	21K	23K	Ω
	85	24K	25k	28K	Ω
	125	26K	27K	29K	Ω

# PIC16F570

**FIGURE 15-8: TIMER0 CLOCK TIMINGS**



**TABLE 15-12: TIMER0 CLOCK REQUIREMENTS**

AC Characteristics		Standard Operating Conditions (unless otherwise specified)					
		Operating Temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (industrial) $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ (extended)					
		Operating voltage $V_{DD}$ range is described in <a href="#">Section 15.1 “DC Characteristics: PIC16F570 (Industrial)”</a> .					
Param. No.	Sym.	Characteristic	Min.	Typ. <sup>(1)</sup>	Max.	Units	Conditions
40	Tt0H	T0CKI High Pulse Width	No Prescaler	$0.5 T_{CY} + 20^*$	—	—	ns
			With Prescaler	10*	—	—	ns
41	Tt0L	T0CKI Low Pulse Width	No Prescaler	$0.5 T_{CY} + 20^*$	—	—	ns
			With Prescaler	10*	—	—	ns
42	Tt0P	T0CKI Period	$20$ or $T_{CY} + 40^* N$	—	—	ns	Whichever is greater. $N = \text{Prescale Value}$ (1, 2, 4, ..., 256)

\* These parameters are characterized but not tested.

**Note 1:** Data in the Typical (“Typ.”) column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

## 15.4 Operational Amplifiers

**TABLE 15-13: OPERATIONAL AMPLIFIER (OPA) MODULE DC SPECIFICATIONS**

OPA DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise stated) V <sub>DD</sub> = 5.0V Operating temperature: 25°C				
Param. No.	Sym.	Characteristics	Min.	Typ.	Max.	Units	Comments
OPA01	V <sub>OS</sub>	<b>Input Offset Voltage</b>	—	±5	±13	mV	
OPA02*	I <sub>B</sub>	<b>Input current and impedance</b> Input bias current	—	±2*	—	nA	
OPA03*	I <sub>OS</sub>		Input offset bias current	—	±1*	—	pA
OPA04*	V <sub>CM</sub>	<b>Common Mode</b> Common mode input range	V <sub>SS</sub>	—	V <sub>DD</sub> – 1.4	V	
OPA05*	CMR		Common mode rejection	60	70	—	dB
OPA06A*	AOL	<b>Open Loop Gain</b> DC Open loop gain	—	70	—	dB	Standard load
OPA07*	V <sub>OUT</sub>	<b>Output</b> Output voltage swing	V <sub>SS</sub> + 50	—	V <sub>DD</sub> – 50	mV	To V <sub>DD</sub> /2 (10 kΩ connected to V <sub>DD</sub> , 10 kΩ + 20 pF to V <sub>SS</sub> )
OPA08*	I <sub>SC</sub>	Output short circuit current	—	25	28	mA	
OPA10*	PSR	<b>Power Supply</b> Power supply rejection	—	70	—	dB	

\* These parameters are characterized but not tested.

**TABLE 15-14: AC CHARACTERISTICS: OPERATIONAL AMPLIFIER (OPA)**

AC CHARACTERISTICS		Standard Operating Conditions (unless otherwise stated) Operating Temperature: 25°C V <sub>DD</sub> = 5.0V					
Param. No.	Parameters	Symbol	Min.	Typ. <sup>(1)</sup>	Max.	Units	Conditions
OPA12*	Gain Bandwidth Product	GBWP	—	3	—	MHz	V <sub>DD</sub> = 5V
OPA13*	Turn on Time	T <sub>ON</sub>	—	—	10	μs	V <sub>DD</sub> = 5V
OPA14*	Phase Margin	∅M	—	55	—	degrees	V <sub>DD</sub> = 5V
OPA15*	Slew Rate	SR	2	—	—	V/μs	V <sub>DD</sub> = 5V

\* These parameters are characterized but not tested.

**Note 1:** Data in "Typ." column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

# PIC16F570

TABLE 15-15: FLASH DATA MEMORY WRITE/ERASE TIME

AC CHARACTERISTICS			Standard Operating Conditions (unless otherwise specified)				
			Operating Temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (industrial) $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ (extended)				
			Operating Voltage VDD range is described in <a href="#">Section 15.1 “DC Characteristics: PIC16F570 (Industrial)”</a> .				
Param. No.	Sym.	Characteristic	Min.	Typ. <sup>(1)</sup>	Max.	Units	Conditions
43	TDW	Flash Data Memory Write Cycle Time	2	3.5	5	ms	
44	TDE	Flash Data Memory Erase Cycle Time	2	3.5	5	ms	

**Note 1:** Data in the Typical (“Typ.”) column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

## 16.0 DC AND AC CHARACTERISTICS GRAPHS AND CHARTS

The graphs and tables provided in this section are for **design guidance** and are **not tested**.

In some graphs or tables, the data presented are **outside specified operating range** (i.e., outside specified  $V_{DD}$  range). This is for **information only** and devices are ensured to operate properly only within the specified range.

<p><b>Note:</b> The graphs and tables provided following this note are a statistical summary based on a limited number of samples and are provided for informational purposes only. The performance characteristics listed herein are not tested or guaranteed. In some graphs or tables, the data presented may be outside the specified operating range (e.g., outside specified power supply range) and therefore, outside the warranted range.</p>
--

**“Typical”** represents the mean of the distribution at 25°C. **“MAXIMUM”**, **“Max.”**, **“MINIMUM”** or **“Min.”** represents  $(\text{mean} + 3\sigma)$  or  $(\text{mean} - 3\sigma)$  respectively, where  $\sigma$  is a standard deviation, over each temperature range.

# PIC16F570

FIGURE 16-1: I<sub>DD</sub>, LP OSCILLATOR, F<sub>osc</sub> = 32 kHz

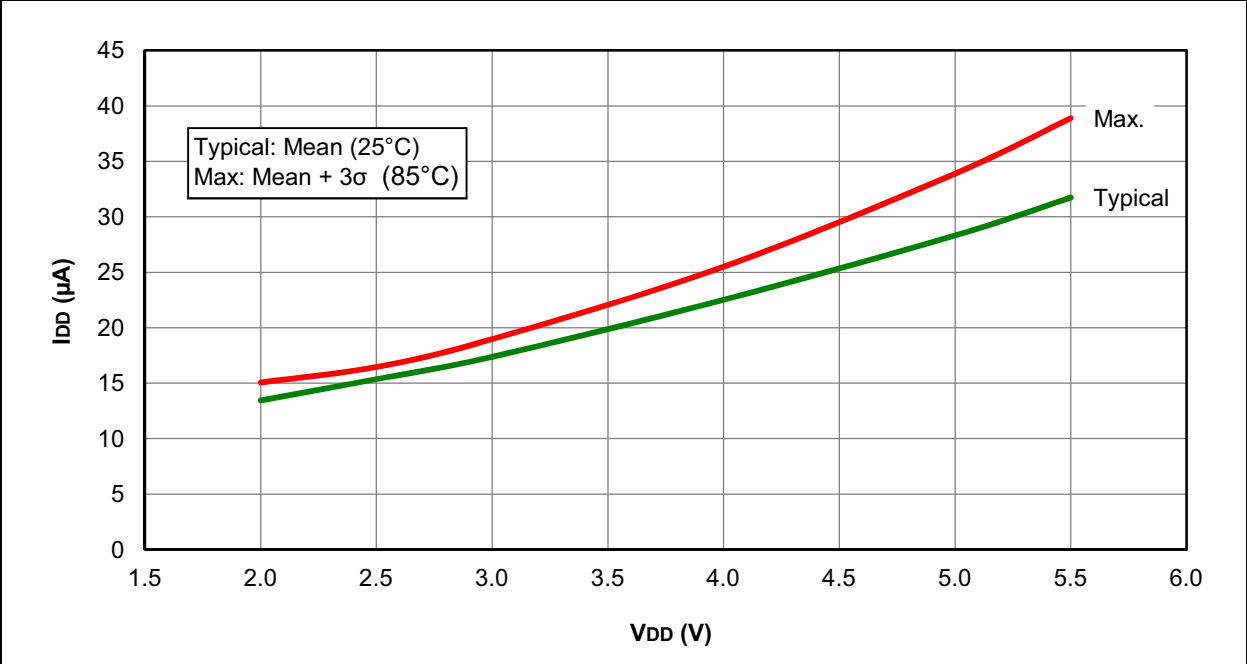


FIGURE 16-2: I<sub>DD</sub> TYPICAL, XT AND EXTRC OSCILLATOR

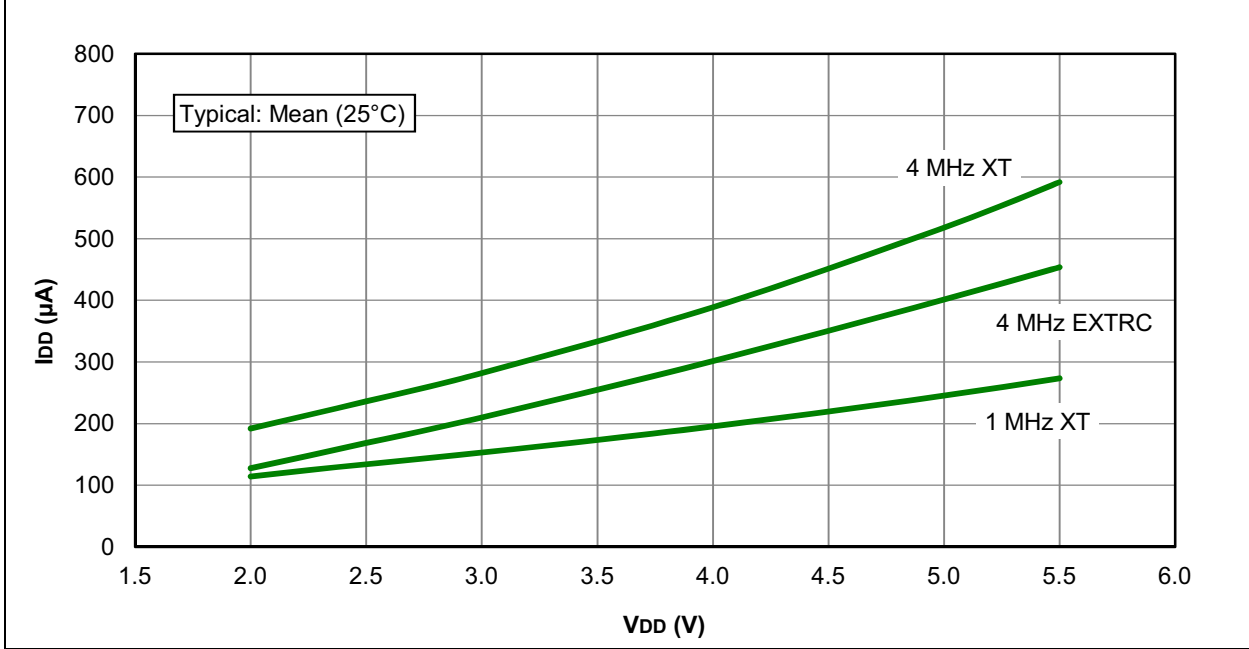
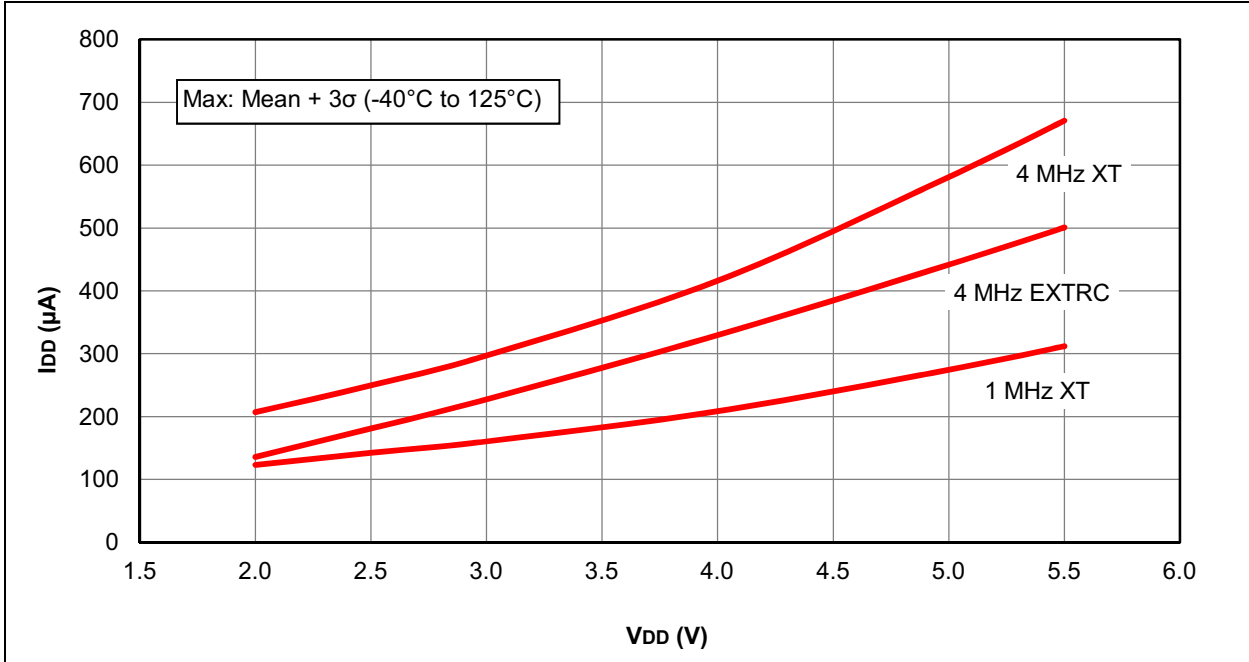


FIGURE 16-3: I<sub>DD</sub> MAXIMUM, XT AND EXTRC OSCILLATOR



# PIC16F570

FIGURE 16-4: I<sub>DD</sub> TYPICAL, EXTERNAL CLOCK MODE

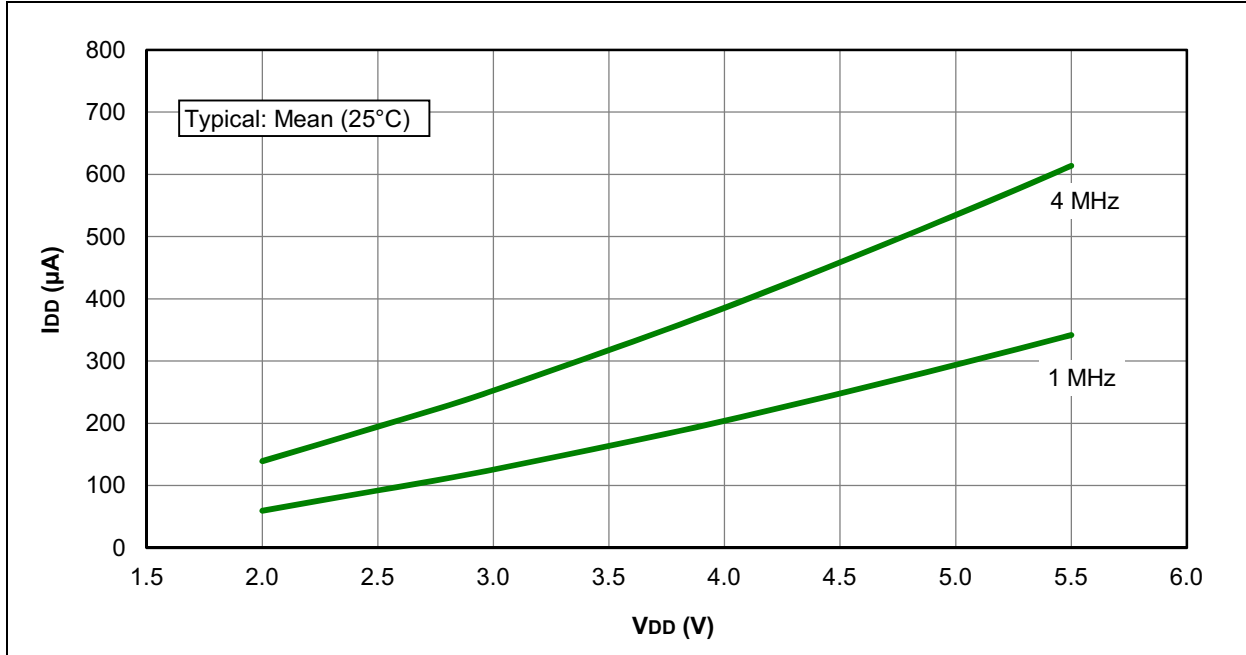


FIGURE 16-5: I<sub>DD</sub> MAXIMUM, EXTERNAL CLOCK MODE

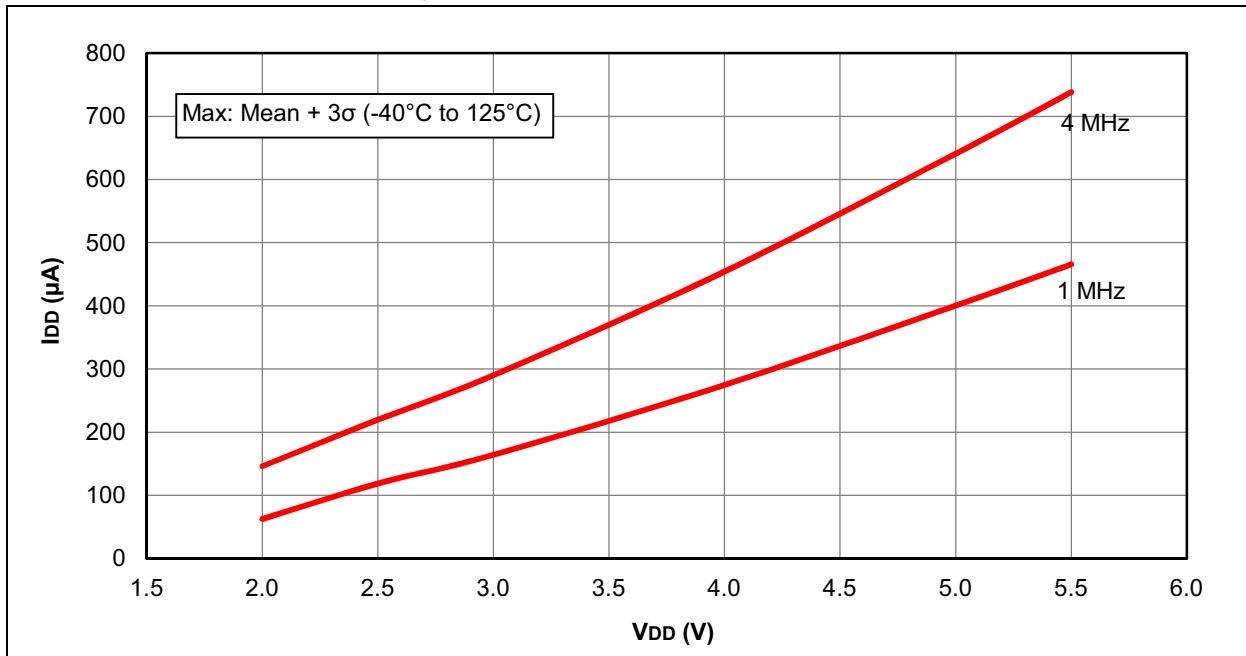


FIGURE 16-6: I<sub>DD</sub> TYPICAL, INTOSC

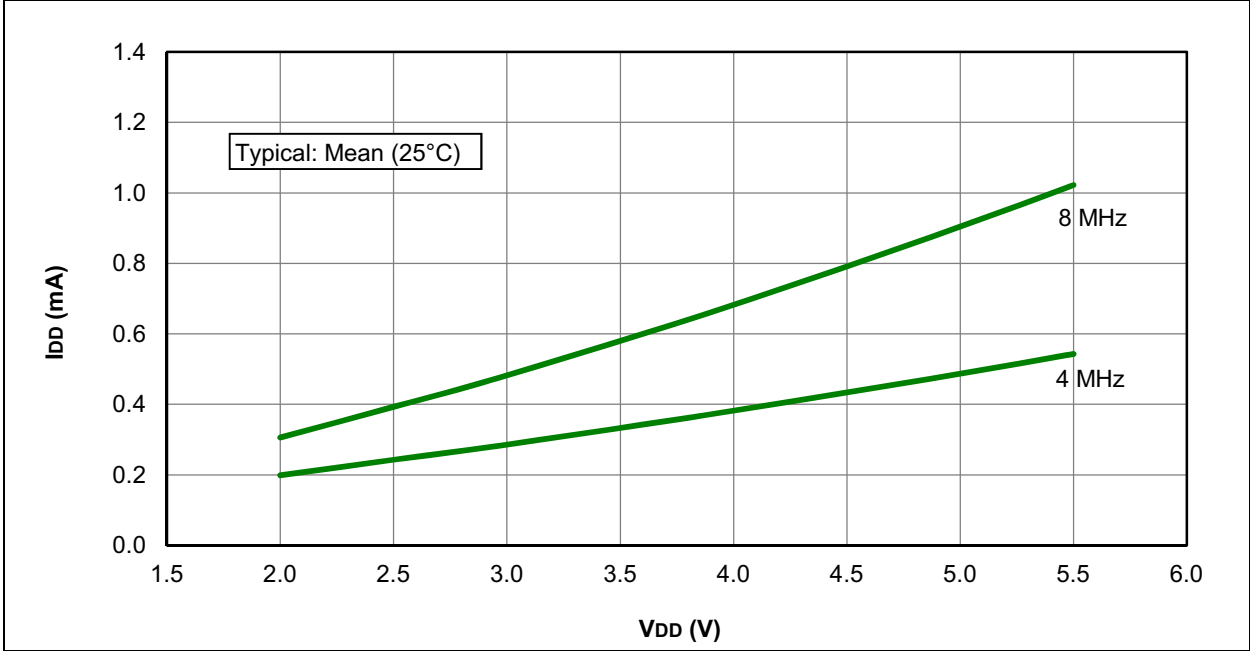
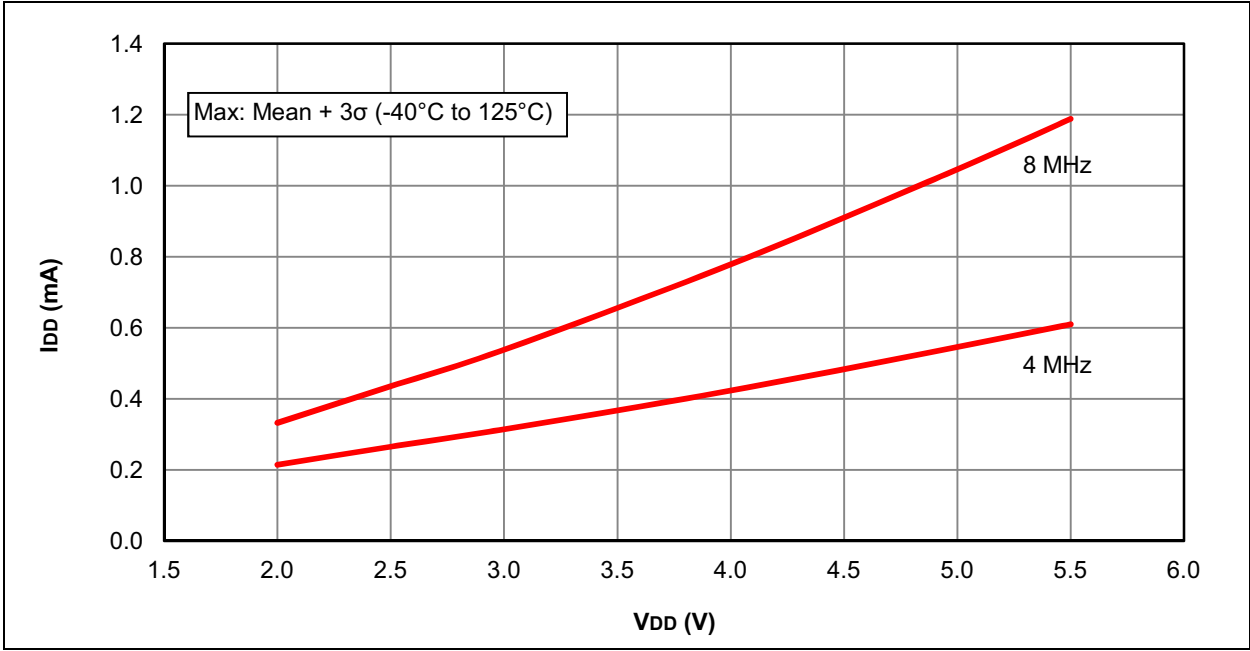


FIGURE 16-7: I<sub>DD</sub> MAXIMUM, INTOSC



# PIC16F570

FIGURE 16-8: I<sub>DD</sub> TYPICAL, HS OSCILLATOR

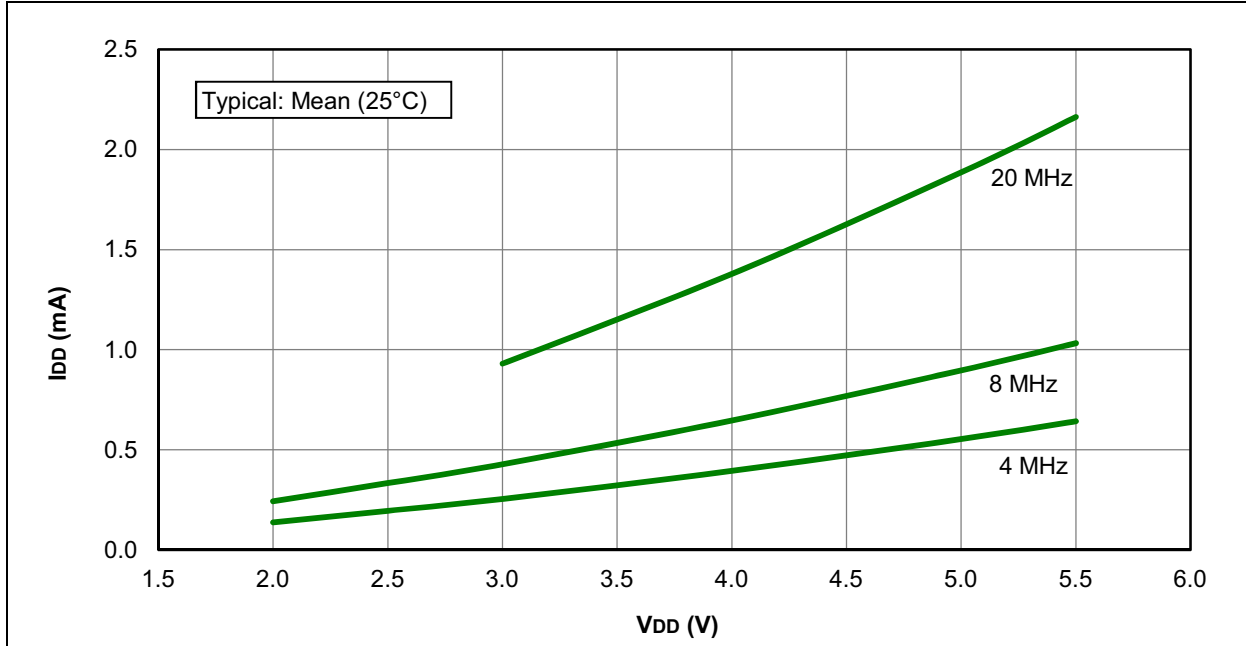


FIGURE 16-9: I<sub>DD</sub> MAXIMUM, HS OSCILLATOR

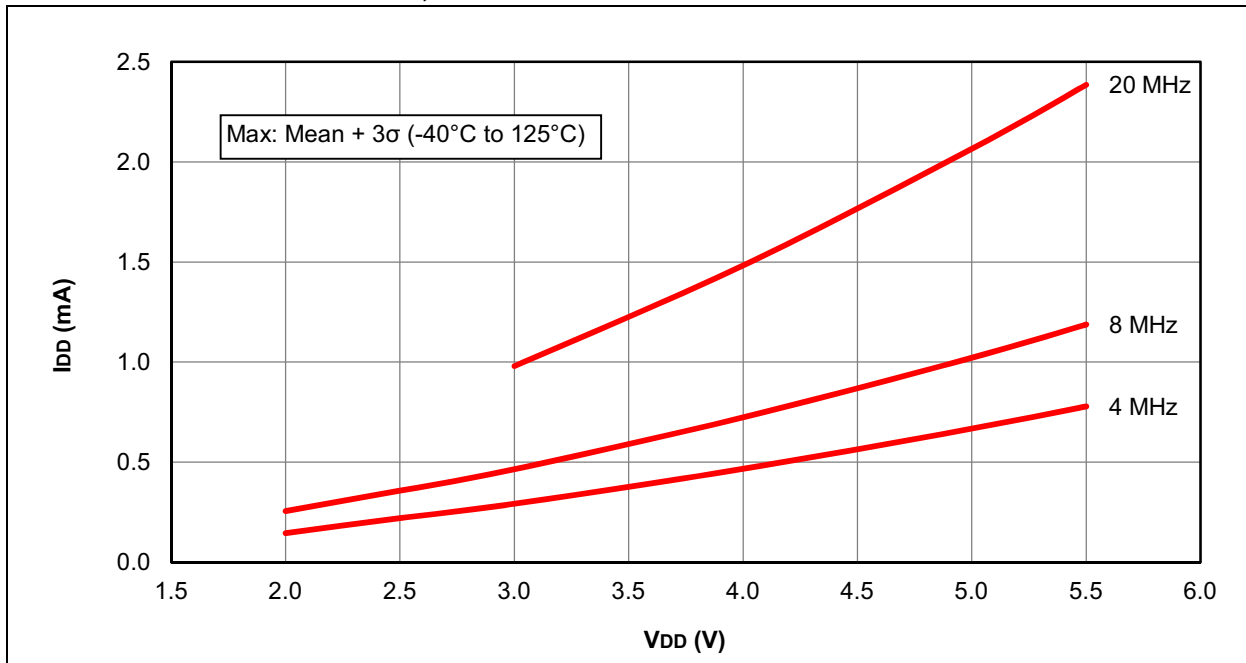


FIGURE 16-10: I<sub>PD</sub> BASE, LOW-POWER SLEEP MODE

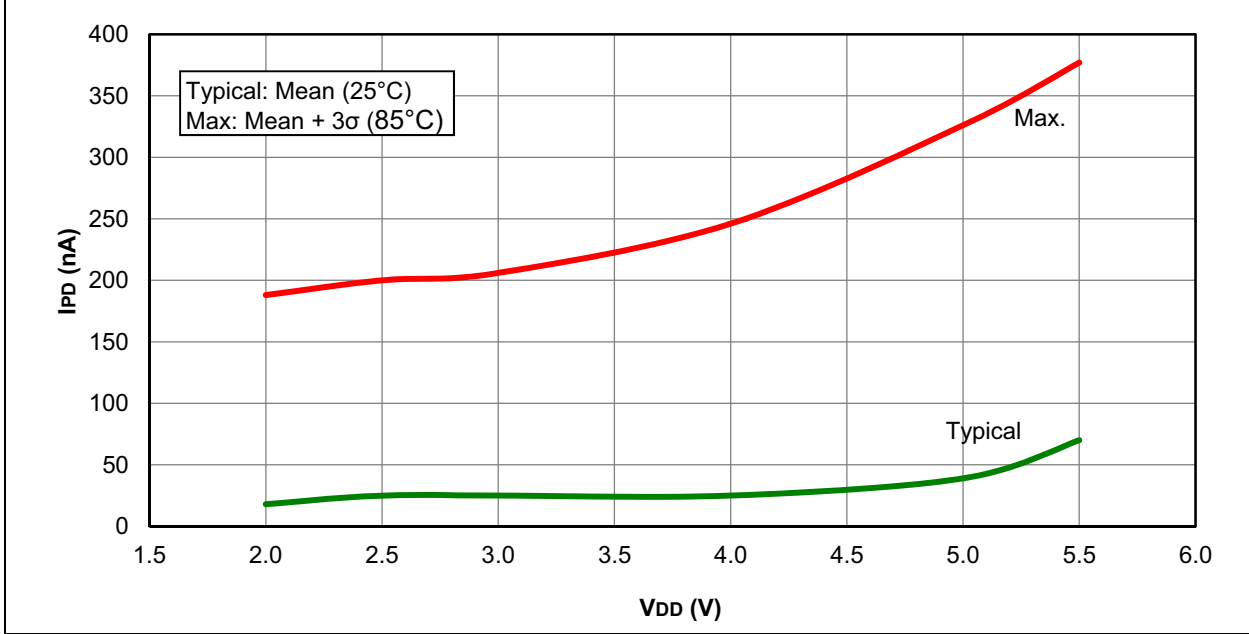
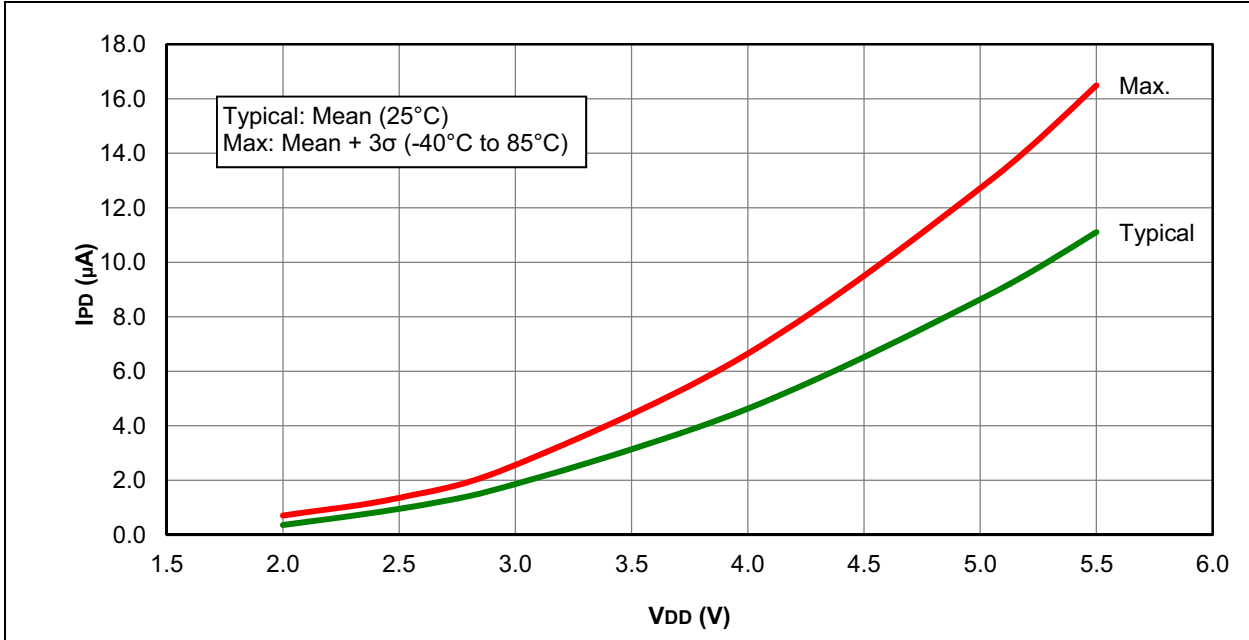


FIGURE 16-11: I<sub>PD</sub>, WATCHDOG TIMER (WDT)



# PIC16F570

FIGURE 16-12:  $I_{PD}$ , FIXED VOLTAGE REFERENCE (FVR)

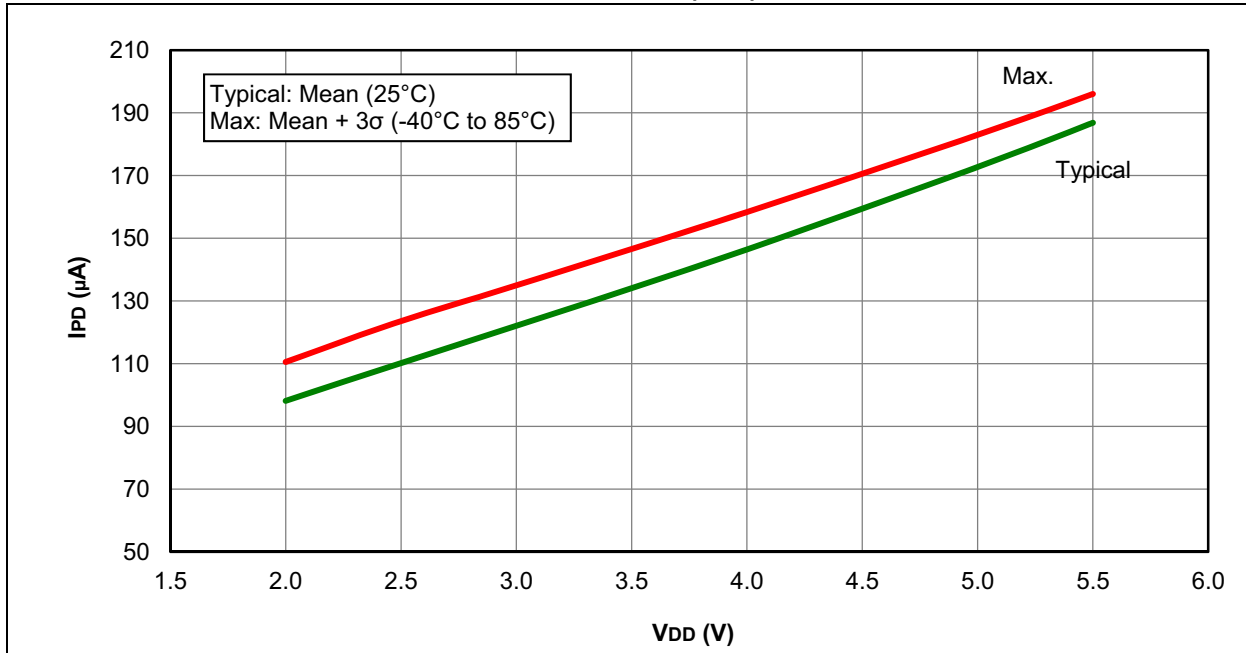
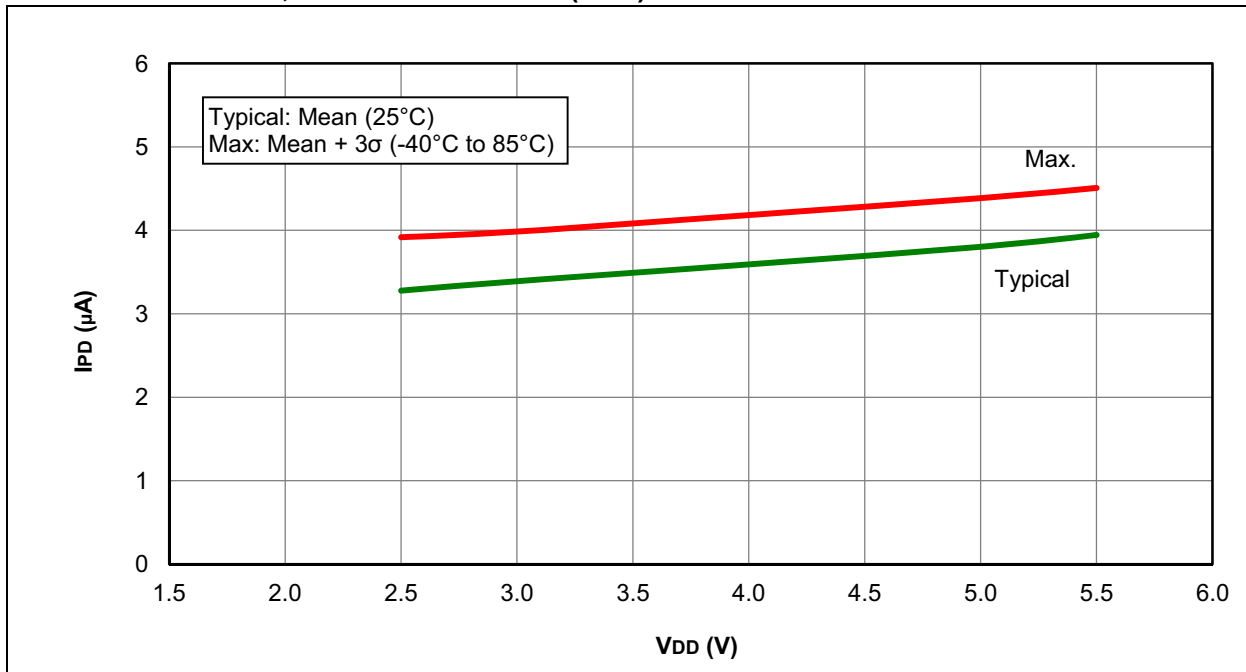
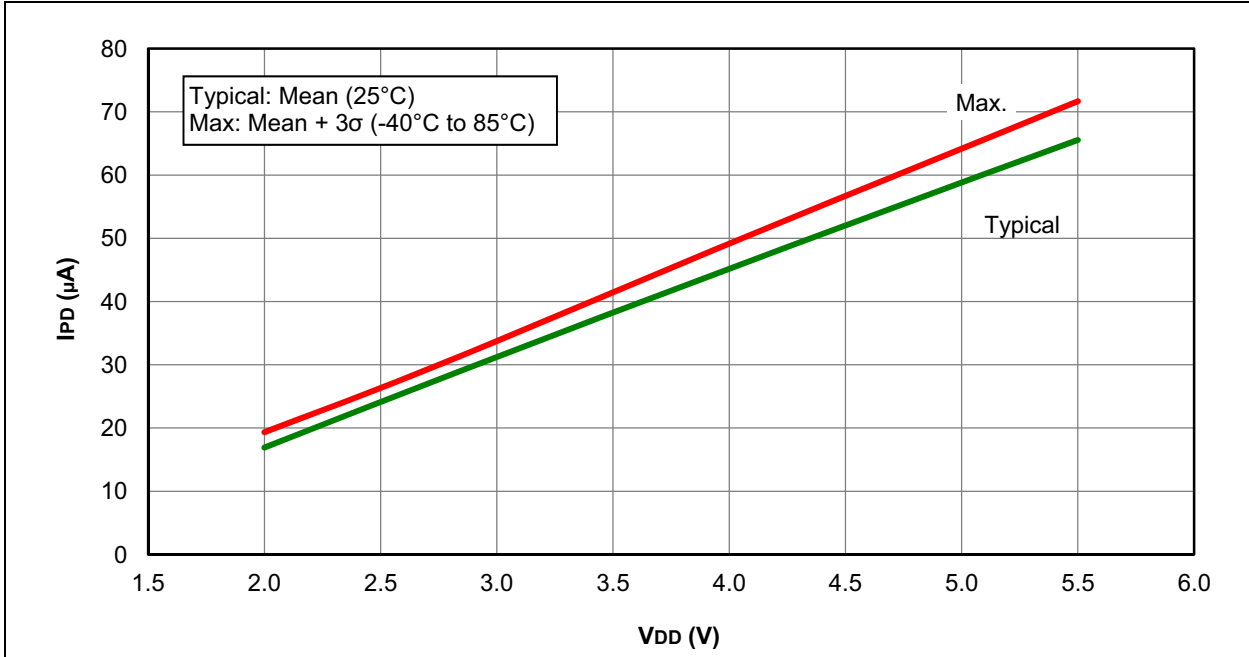


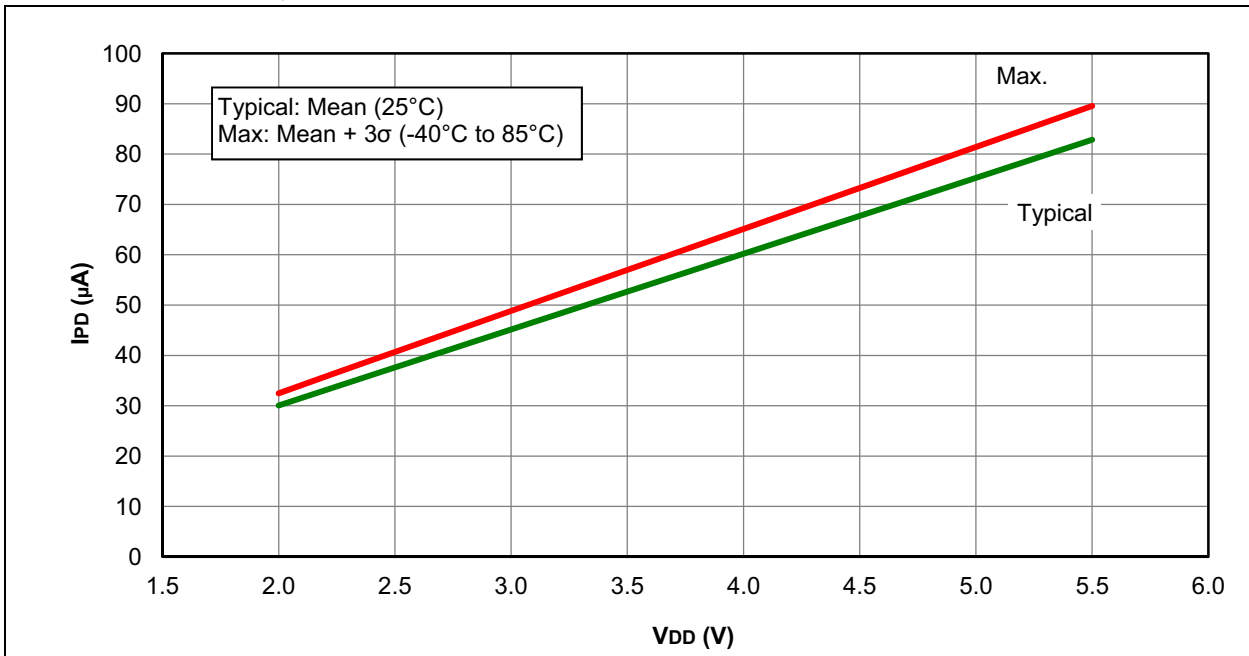
FIGURE 16-13:  $I_{PD}$ , BROWN-OUT RESET (BOR)



**FIGURE 16-14: I<sub>PD</sub>, SINGLE COMPARATOR**

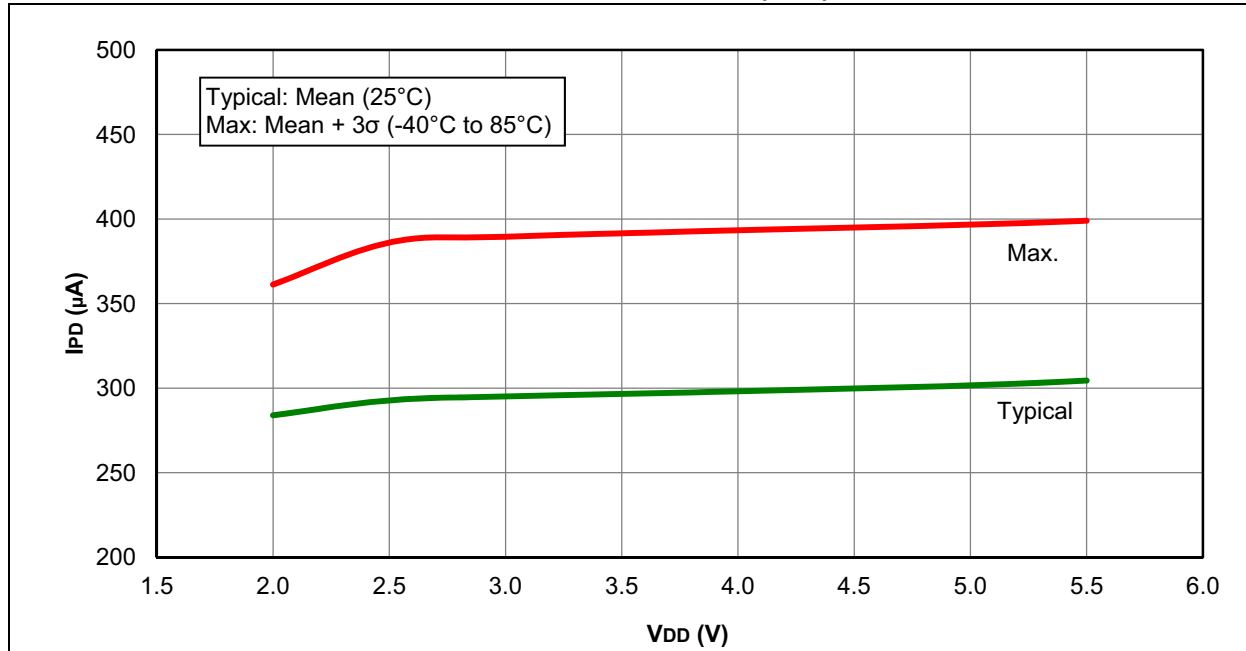


**FIGURE 16-15: I<sub>PD</sub>, CVREF**

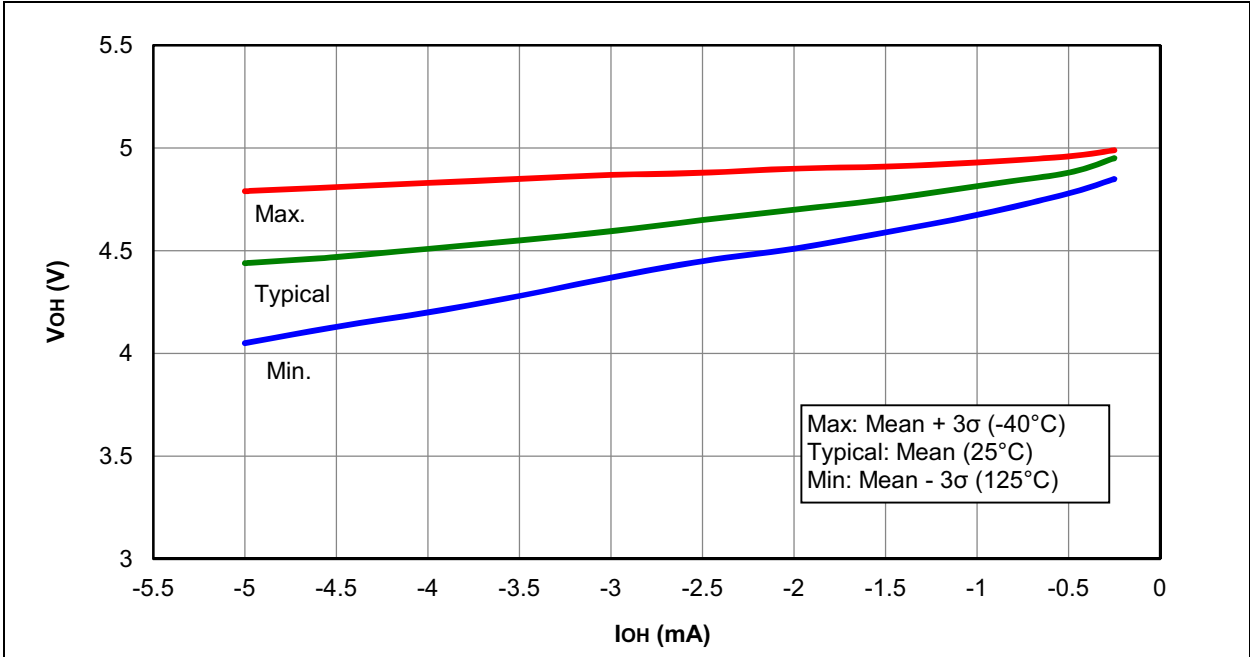


# PIC16F570

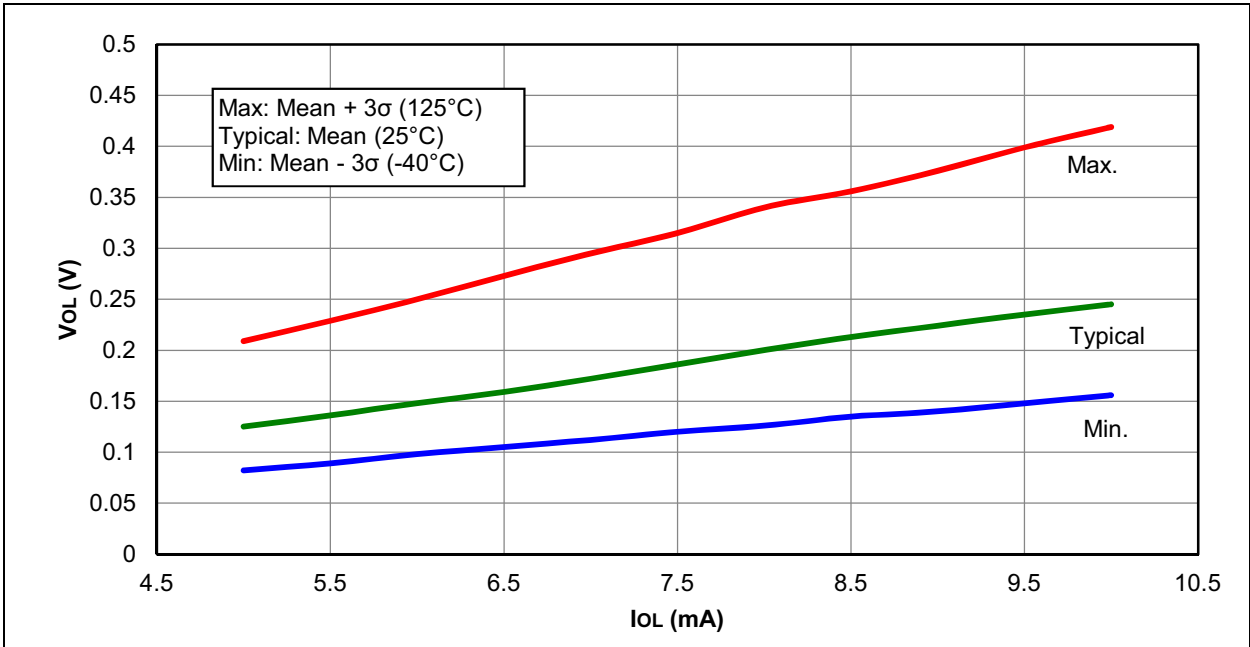
FIGURE 16-16:  $I_{PD}$ , SINGLE OPERATIONAL AMPLIFIER (OPA) – UNITY GAIN MODE



**FIGURE 16-17:  $V_{OH}$  vs.  $I_{OH}$  OVER TEMPERATURE,  $V_{DD} = 5.0V$**



**FIGURE 16-18:  $V_{OL}$  vs.  $I_{OL}$  OVER TEMPERATURE,  $V_{DD} = 5.0 V$**



# PIC16F570

FIGURE 16-19:  $V_{OH}$  vs.  $I_{OH}$  OVER TEMPERATURE,  $V_{DD} = 3.0\text{ V}$

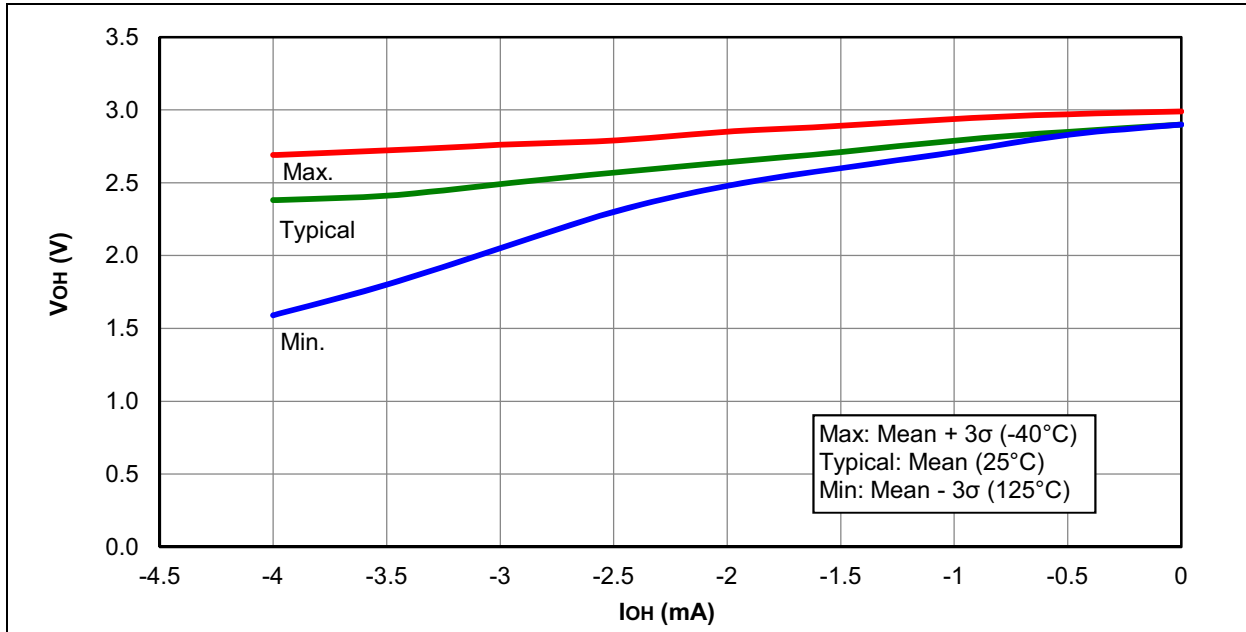
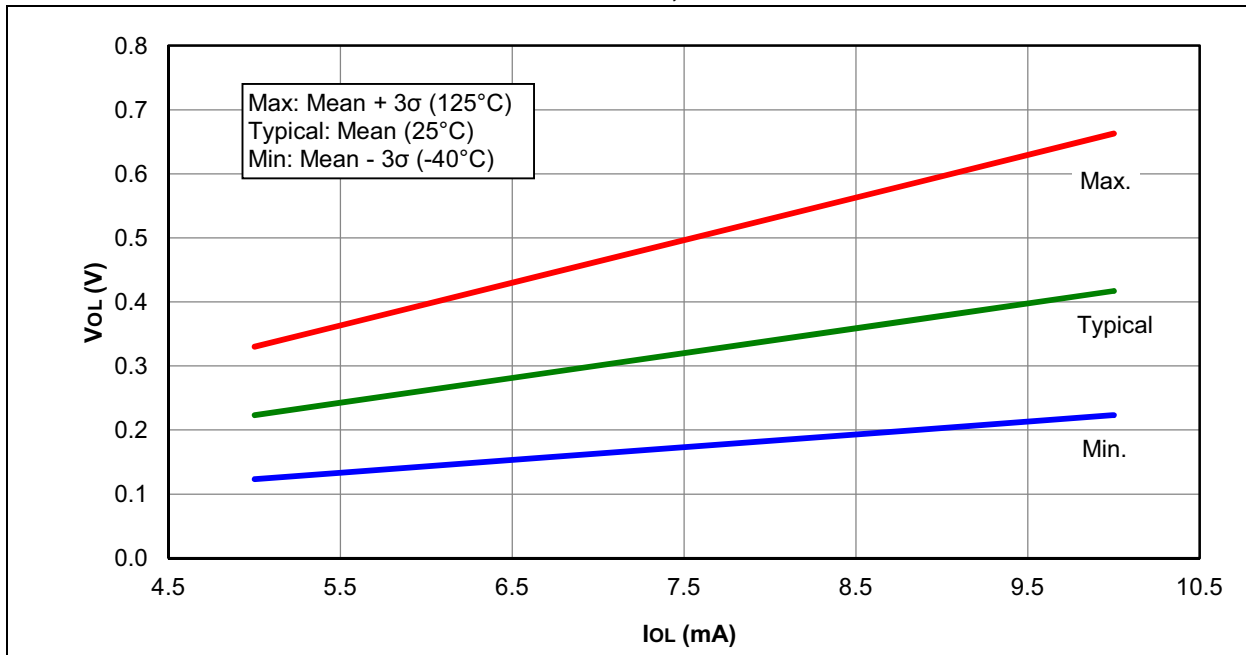
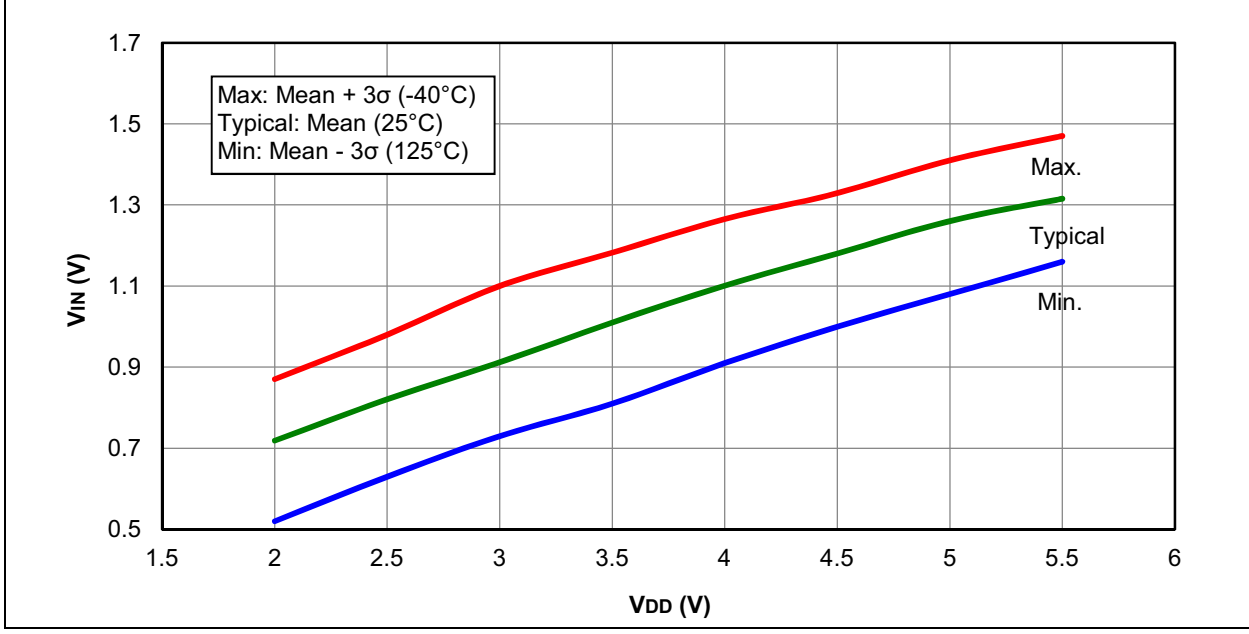


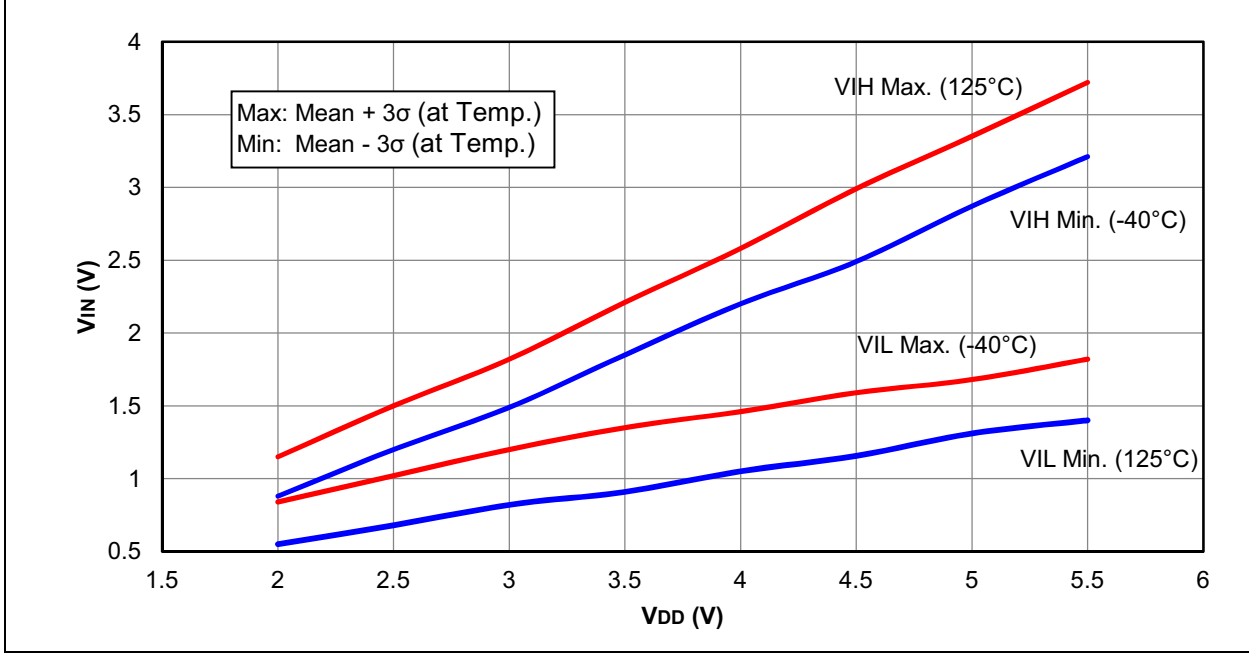
FIGURE 16-20:  $V_{OL}$  vs.  $I_{OL}$  OVER TEMPERATURE,  $V_{DD} = 3.0\text{ V}$



**FIGURE 16-21: TTL INPUT THRESHOLD  $V_{IN}$  vs.  $V_{DD}$**



**FIGURE 16-22: SCHMITT TRIGGER INPUT THRESHOLD  $V_{IN}$  vs.  $V_{DD}$**



# PIC16F570

FIGURE 16-23: BROWN-OUT RESET VOLTAGE

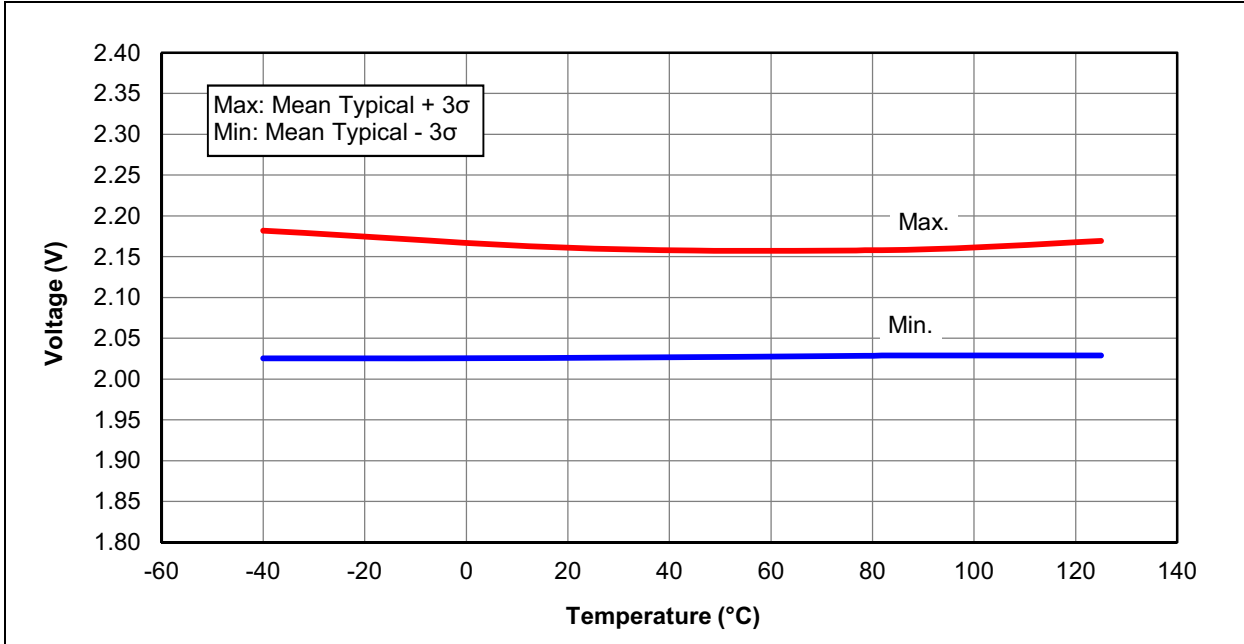
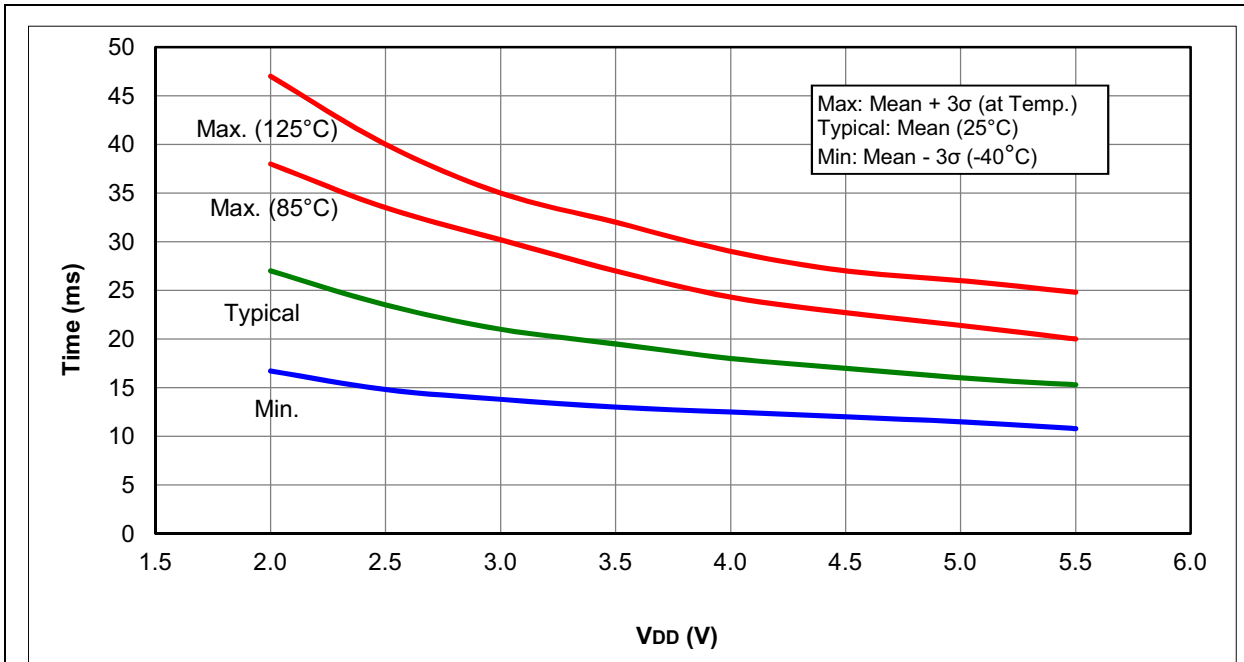


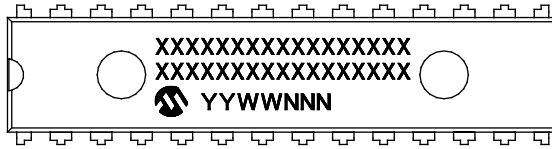
FIGURE 16-24: WDT TIME-OUT PERIOD



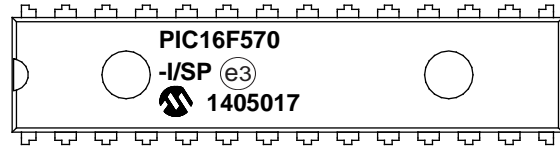
## 17.0 PACKAGING INFORMATION

### 17.1 Package Marking Information

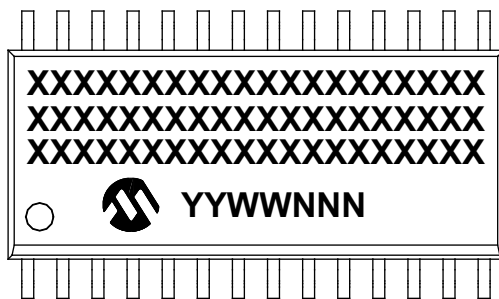
28-Lead SPDIP (.300")



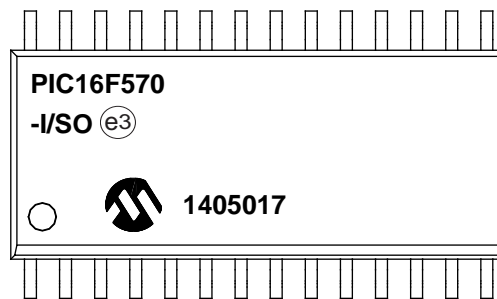
Example



28-Lead SOIC (7.50 mm)



Example



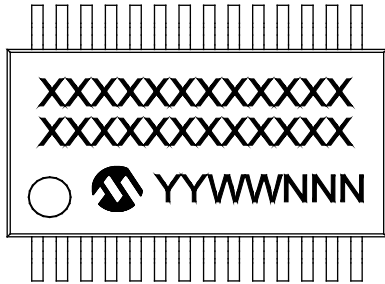
<b>Legend:</b>	XX...X	Customer-specific information
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week '01')
	NNN	Alphanumeric traceability code
	(e3)	Pb-free JEDEC® designator for Matte Tin (Sn)
	*	This package is Pb-free.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

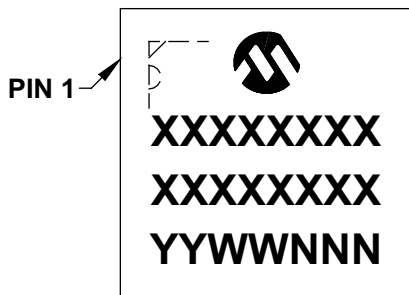
# PIC16F570

## Package Marking Information (Continued)

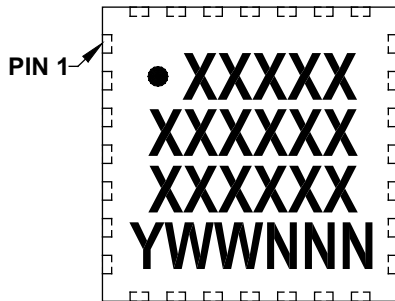
28-Lead SSOP (5.30 mm)



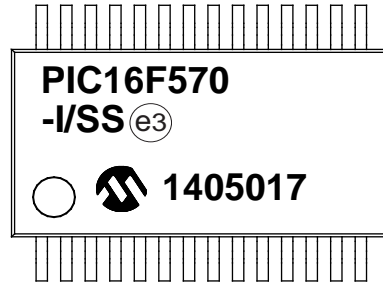
28-Lead QFN (6x6 mm)



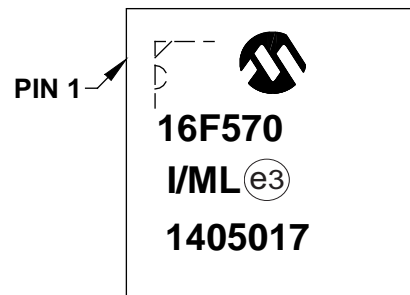
28-Lead UQFN (4x4x0.5 mm)



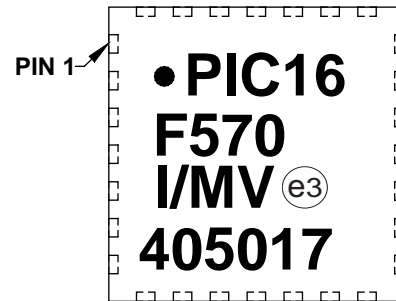
Example



Example



Example

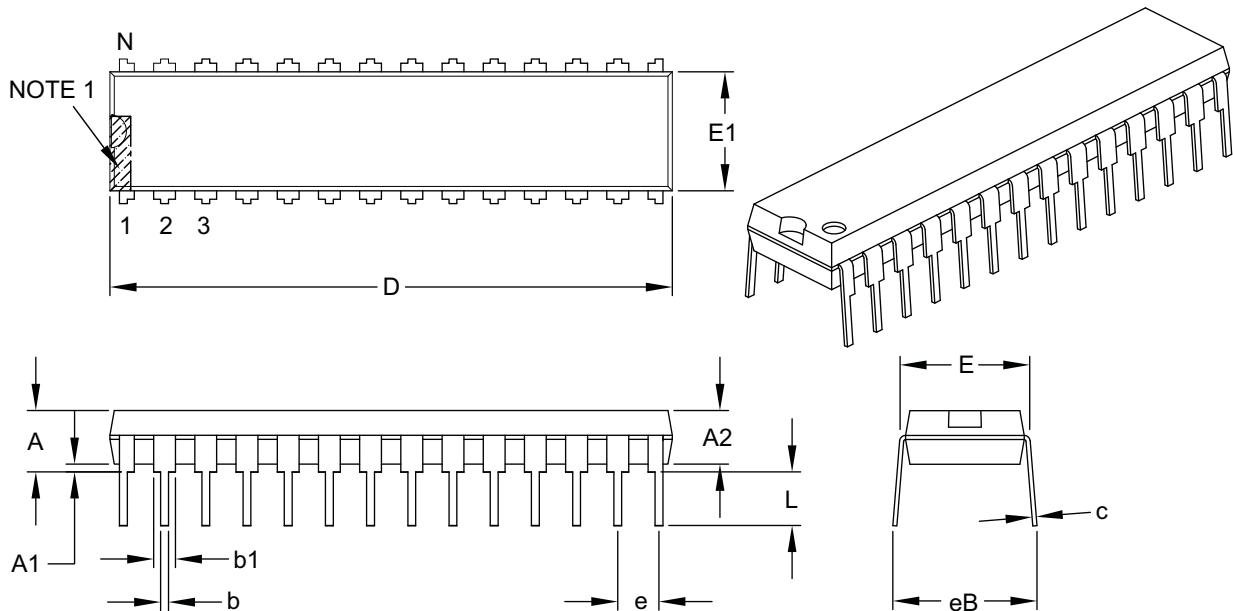


<b>Legend:</b>	XX...X	Customer-specific information
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week '01')
	NNN	Alphanumeric traceability code
	(e3)	Pb-free JEDEC® designator for Matte Tin (Sn)
	*	This package is Pb-free.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

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

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



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

**Notes:**

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

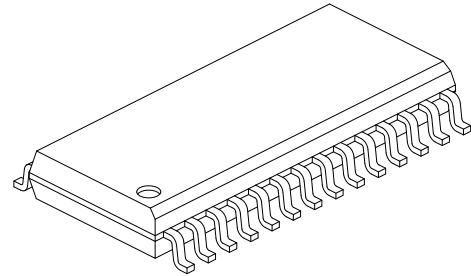
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-070B



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

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



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

**Notes:**

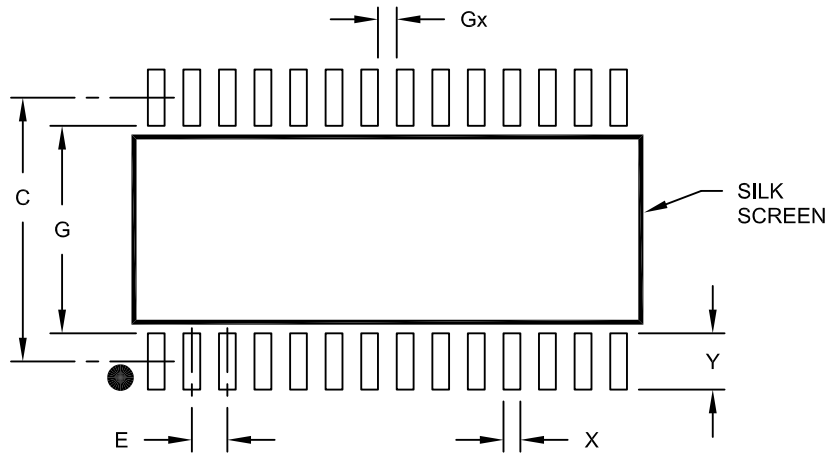
- Pin 1 visual index feature may vary, but must be located within the hatched area.
- § Significant Characteristic
- Dimension D does not include mold flash, protrusions or gate burrs, which shall not exceed 0.15 mm per end. Dimension E1 does not include interlead flash or protrusion, which shall not exceed 0.25 mm per side.
- Dimensioning and tolerancing per ASME Y14.5M  
 BSC: Basic Dimension. Theoretically exact value shown without tolerances.  
 REF: Reference Dimension, usually without tolerance, for information purposes only.
- Datums A & B to be determined at Datum H.

Microchip Technology Drawing C04-052C Sheet 2 of 2

# PIC16F570

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

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



## RECOMMENDED LAND PATTERN

Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Contact Pitch	E	1.27 BSC		
Contact Pad Spacing	C		9.40	
Contact Pad Width (X28)	X			0.60
Contact Pad Length (X28)	Y			2.00
Distance Between Pads	Gx	0.67		
Distance Between Pads	G	7.40		

**Notes:**

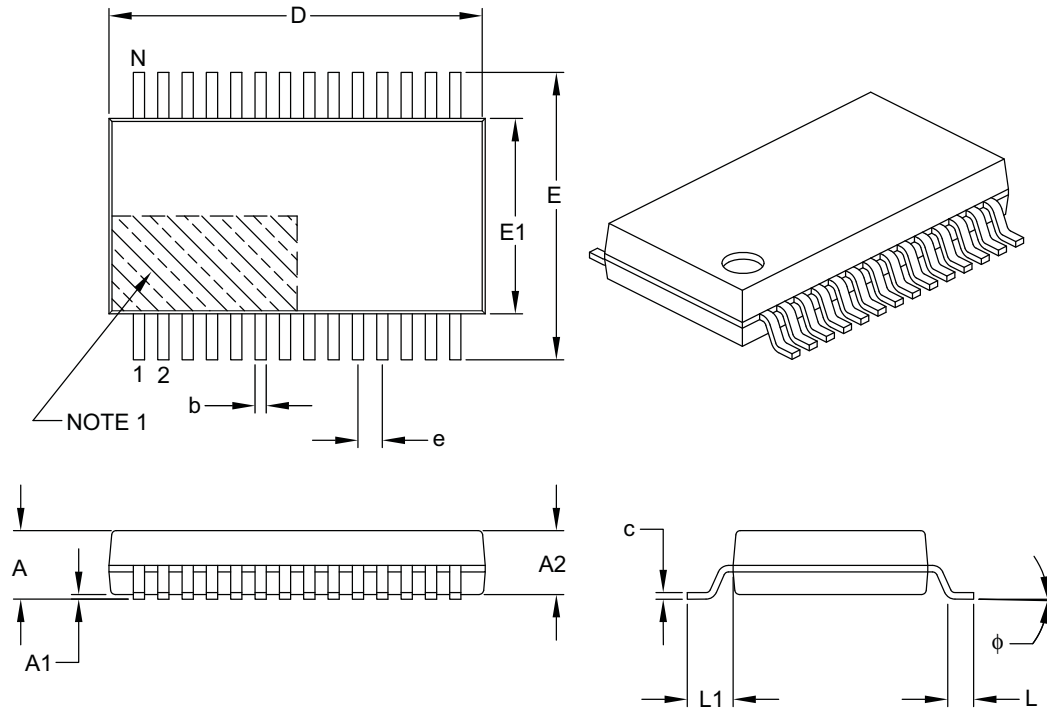
1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-2052A

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

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



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

**Notes:**

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

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

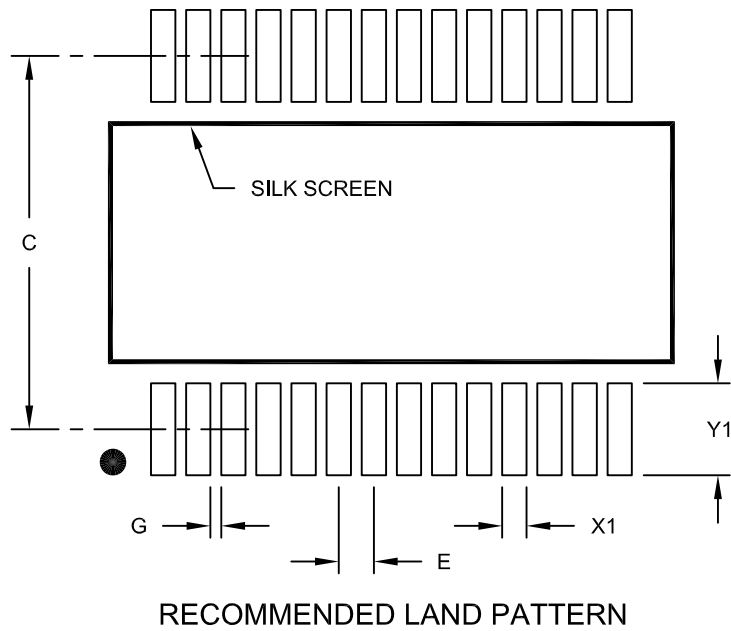
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-073B

# PIC16F570

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

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



Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Contact Pitch	E	0.65 BSC		
Contact Pad Spacing	C		7.20	
Contact Pad Width (X28)	X1			0.45
Contact Pad Length (X28)	Y1			1.75
Distance Between Pads	G	0.20		

Notes:

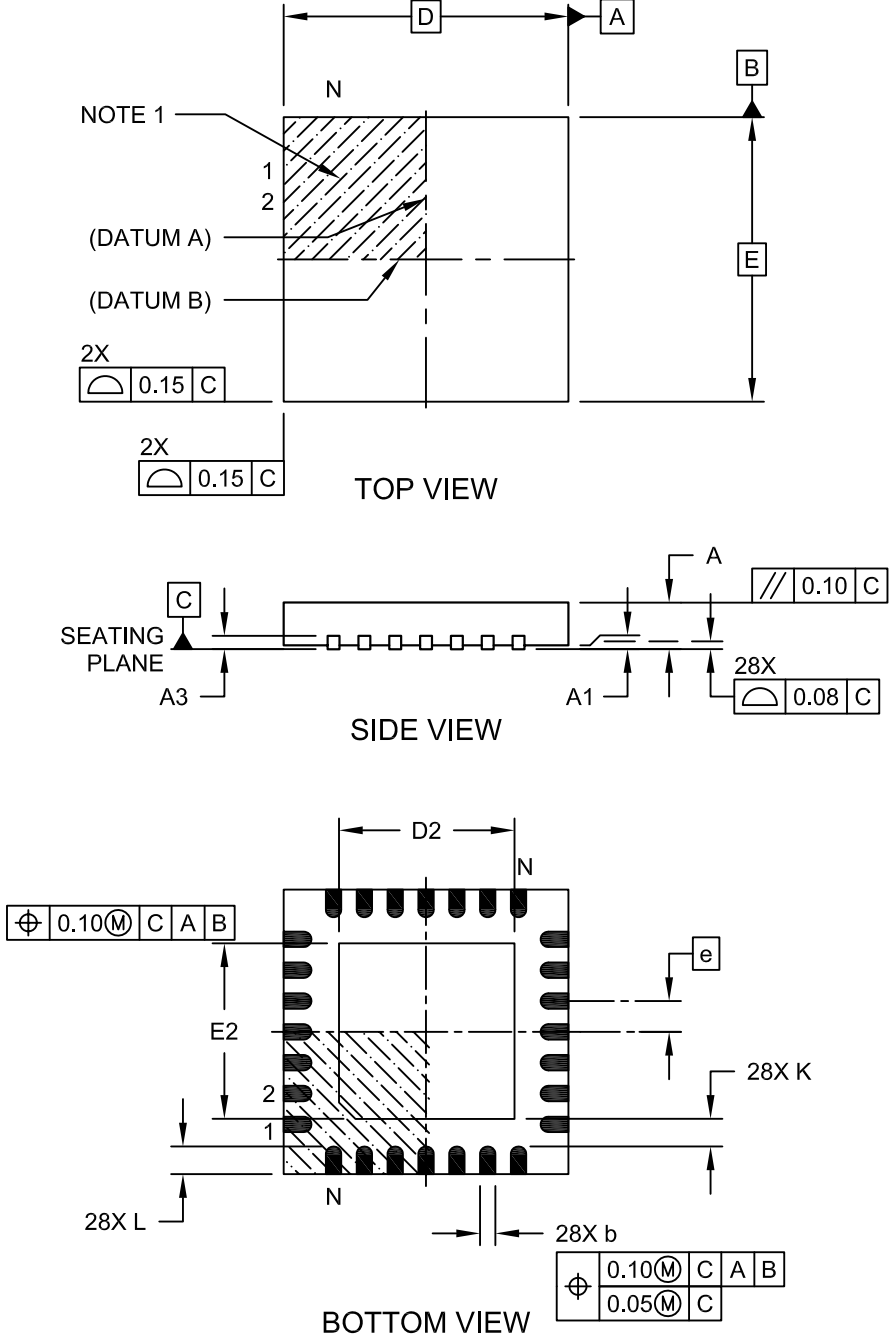
1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-2073A

## 28-Lead Plastic Quad Flat, No Lead Package (ML) - 6x6 mm Body [QFN] With 0.55 mm Terminal Length

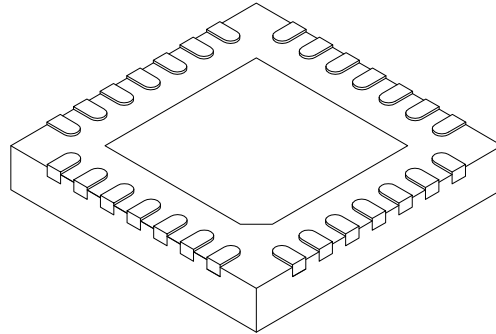
**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



# PIC16F570

## 28-Lead Plastic Quad Flat, No Lead Package (ML) - 6x6 mm Body [QFN] With 0.55 mm Terminal Length

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



Dimension	Units	MILLIMETERS		
		Limits	MIN	NOM
Number of Pins	N	28		
Pitch	e	0.65 BSC		
Overall Height	A	0.80	0.90	1.00
Standoff	A1	0.00	0.02	0.05
Terminal Thickness	A3	0.20 REF		
Overall Width	E	6.00 BSC		
Exposed Pad Width	E2	3.65	3.70	4.20
Overall Length	D	6.00 BSC		
Exposed Pad Length	D2	3.65	3.70	4.20
Terminal Width	b	0.23	0.30	0.35
Terminal Length	L	0.50	0.55	0.70
Terminal-to-Exposed Pad	K	0.20	-	-

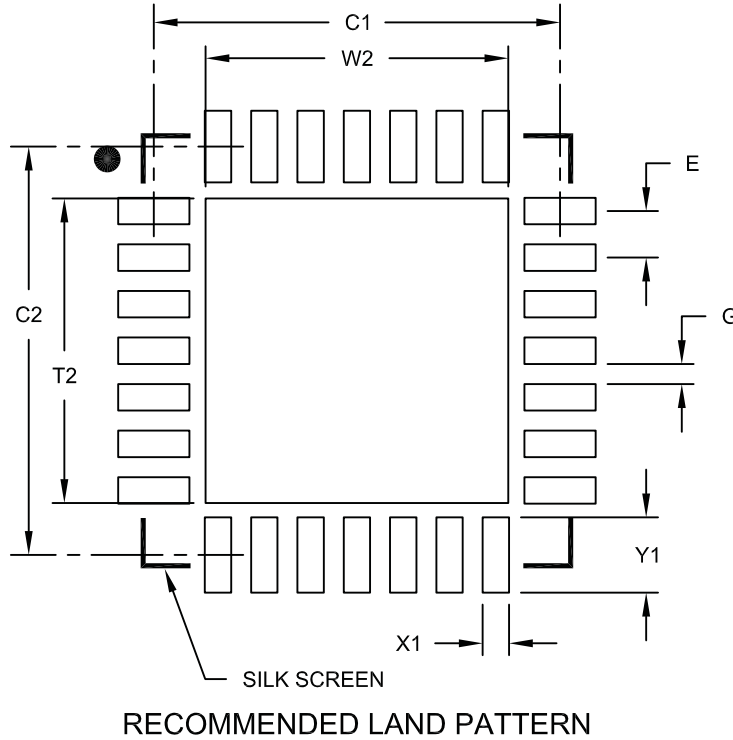
**Notes:**

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package is saw singulated
3. Dimensioning and tolerancing per ASME Y14.5M.  
BSC: Basic Dimension. Theoretically exact value shown without tolerances.  
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-105C Sheet 2 of 2

## 28-Lead Plastic Quad Flat, No Lead Package (ML) – 6x6 mm Body [QFN] with 0.55 mm Contact Length

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



Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Contact Pitch	E	0.65 BSC		
Optional Center Pad Width	W2			4.25
Optional Center Pad Length	T2			4.25
Contact Pad Spacing	C1		5.70	
Contact Pad Spacing	C2		5.70	
Contact Pad Width (X28)	X1			0.37
Contact Pad Length (X28)	Y1			1.00
Distance Between Pads	G	0.20		

**Notes:**

1. Dimensioning and tolerancing per ASME Y14.5M

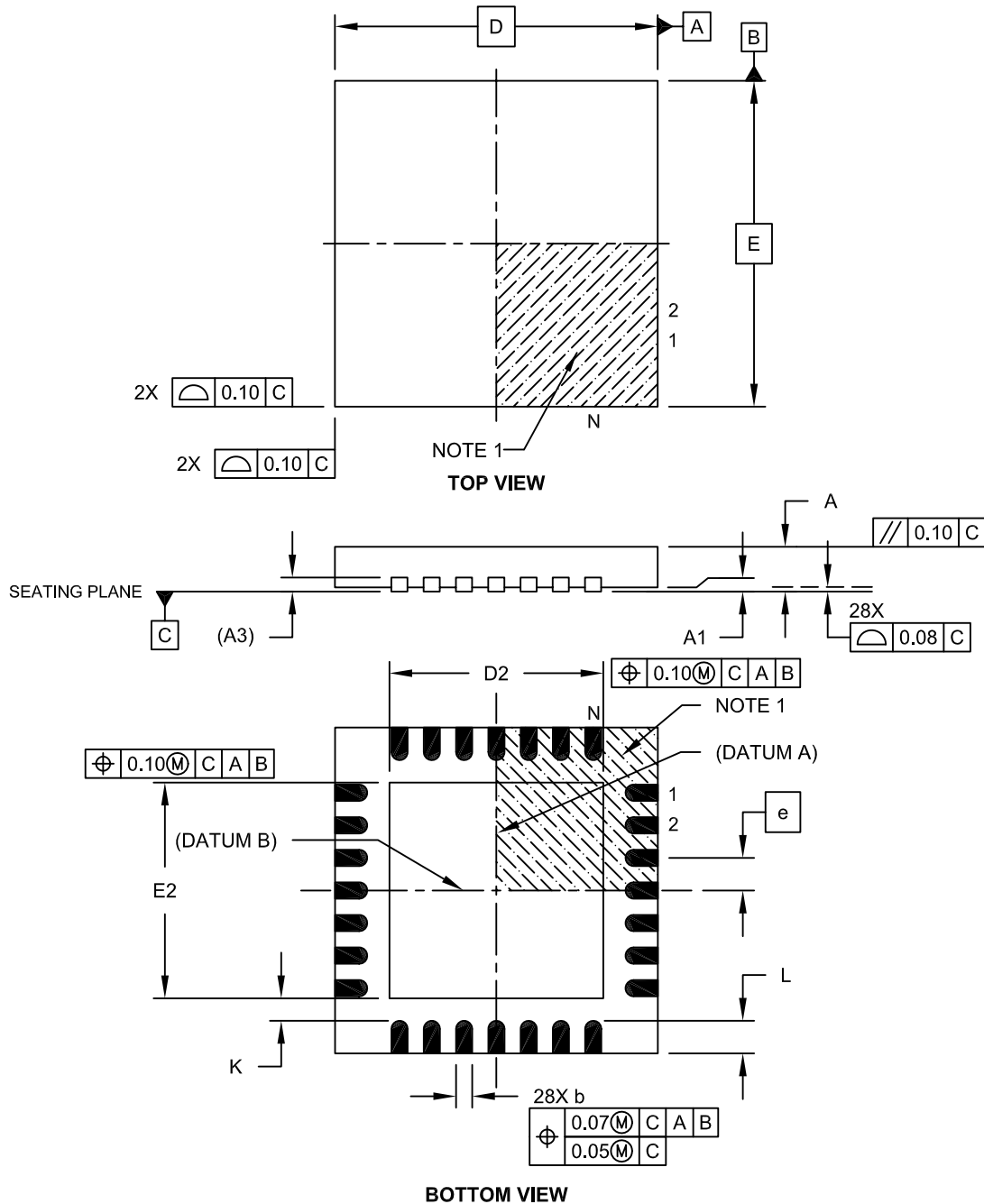
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2105A

# PIC16F570

## 28-Lead Plastic Ultra Thin Quad Flat, No Lead Package (MV) – 4x4x0.5 mm Body [UQFN]

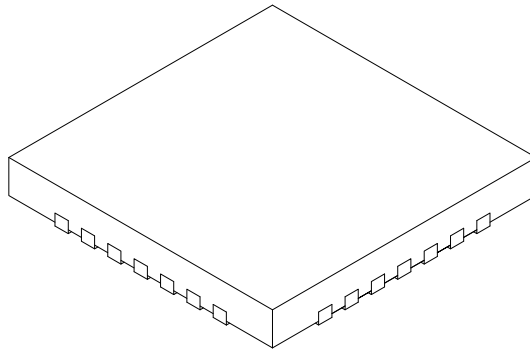
**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



Microchip Technology Drawing C04-152A Sheet 1 of 2

## 28-Lead Plastic Ultra Thin Quad Flat, No Lead Package (MV) – 4x4x0.5 mm Body [UQFN]

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



		Units	MILLIMETERS		
Dimension Limits			MIN	NOM	MAX
Number of Pins	N		28		
Pitch	e		0.40 BSC		
Overall Height	A	0.45	0.50	0.55	
Standoff	A1	0.00	0.02	0.05	
Contact Thickness	A3	0.127 REF			
Overall Width	E	4.00 BSC			
Exposed Pad Width	E2	2.55	2.65	2.75	
Overall Length	D	4.00 BSC			
Exposed Pad Length	D2	2.55	2.65	2.75	
Contact Width	b	0.15	0.20	0.25	
Contact Length	L	0.30	0.40	0.50	
Contact-to-Exposed Pad	K	0.20	-	-	

**Notes:**

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- Package is saw singulated.
- Dimensioning and tolerancing per ASME Y14.5M.

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

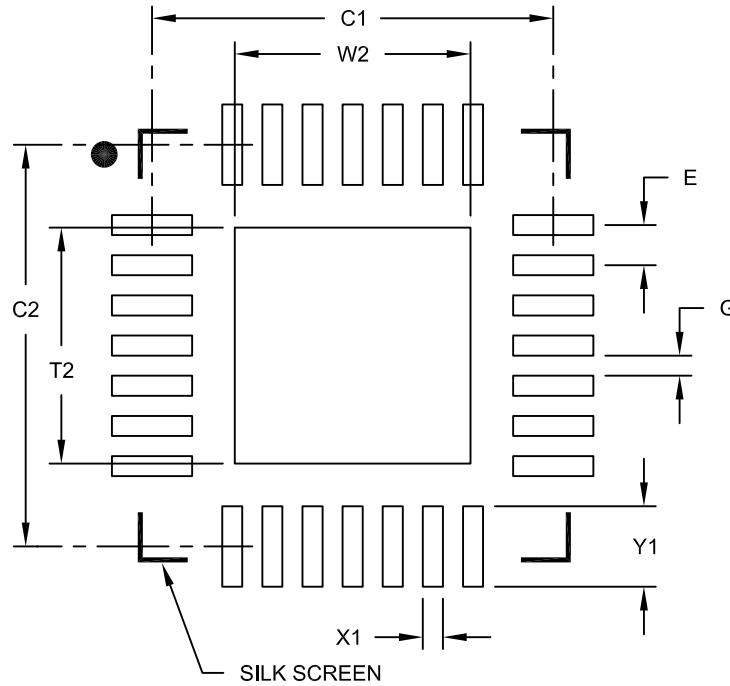
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-152A Sheet 2 of 2

# PIC16F570

28-Lead Ultra Thin Plastic Quad Flat, No Lead Package (MV) - 4x4 mm Body [UQFN]  
With 0.40 mm Contact Length

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



RECOMMENDED LAND PATTERN

Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Contact Pitch	E	0.40 BSC		
Optional Center Pad Width	W2			2.35
Optional Center Pad Length	T2			2.35
Contact Pad Spacing	C1		4.00	
Contact Pad Spacing	C2		4.00	
Contact Pad Width (X28)	X1			0.20
Contact Pad Length (X28)	Y1			0.80
Distance Between Pads	G	0.20		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-2152A

## **APPENDIX A: DATA SHEET REVISION HISTORY**

### **Revision A (02/2013)**

Initial release of this document.

### **Revision B (05/2013)**

Updated the Family Types table (Table 1); Updated Figure 3-1; Updated the Memory Organization section; Updated Examples 5-1, 5-2 and 5-3; Updated Figure 6-1; Updated Note 1 in Register 6-5; Updated Register 8-1; Updated Figure 8-6; Added new Section 9.1.5, A/D Acquisition Requirements; Updated Register 9-1; Updated Figure 10-1 and Register 10-1; Updated section 12, OPA module; Updated the Electrical Specifications section; Other minor corrections.

### **Revision C (07/2013)**

Updated the Electrical Characteristics section.

### **Revision D (10/2013)**

Added graphs to the DC and AC Characteristics Graphs and Charts chapter; Updated the Electrical Characteristics chapter.

### **Revision E (02/2014)**

Updated for Final status; Updated DC Characteristics 15.1 and 15.2.

### **Revision F (01/2016)**

Added “eXtreme Low-Power (XLP) Features” section. Updated Table 1. Other minor corrections.

# PIC16F570

---

## THE MICROCHIP WEBSITE

Microchip provides online support via our website at [www.microchip.com](http://www.microchip.com). This website is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the website contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

## CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip website at [www.microchip.com](http://www.microchip.com). Under "Support", click on "Customer Change Notification" and follow the registration instructions.

## CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or Field Application Engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

**Technical support is available through the website at: <http://microchip.com/support>**

## PRODUCT IDENTIFICATION SYSTEM

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

<u>PART NO.</u>	<u>[X]<sup>(1)</sup></u>	-	<u>X</u>	<u>/XX</u>	<u>XXX</u>
Device	Tape and Reel Option		Temperature Range	Package	Pattern
<b>Device:</b>	PIC16F570				
<b>Tape and Reel Option:</b>	Blank	=	Standard packaging (tube or tray)		
	T	=	Tape and Reel <sup>(1)</sup>		
<b>Temperature Range:</b>	I	=	-40°C to +85°C (Industrial)		
	E	=	-40°C to +125°C (Extended)		
<b>Package:</b>	ML	=	Micro Lead Frame (QFN) 6x6		
	MV	=	Micro Lead Frame (UQFN) 4x4		
	SP	=	Skinny Plastic DIP (SPDIP)		
	S0	=	Small Outline (7.50 mm) (SOIC)		
	SS	=	Shrink Small Outline (5.30 mm) (SSOP)		
<b>Pattern:</b>	QTP, SQTP, Code or Special Requirements (blank otherwise)				

**Examples:**

- a) PIC16F570T - I/ML 301  
Tape and Reel, Industrial temperature, QFN 6x6 package, QTP pattern #301
- b) PIC16F570 - E/SP  
Extended temperature SPDIP package
- c) PIC16F570 - E/SO  
Extended temperature, SOIC package

**Note 1:** Tape and Reel identifier only appears in the catalog part number description. This identifier is used for ordering purposes and is not printed on the device package. Check with your Microchip Sales Office for package availability with the Tape and Reel option.

---

---

**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 unless otherwise stated.

#### **Trademarks**

The Microchip name and logo, the Microchip logo, dsPIC, FlashFlex, flexPWR, JukeBlox, KEELOQ, KEELOQ logo, Klear, LANCheck, MediaLB, MOST, MOST logo, MPLAB, OptoLyzer, PIC, PICSTART, PIC<sup>32</sup> logo, RightTouch, SpyNIC, SST, SST Logo, SuperFlash and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

The Embedded Control Solutions Company and mTouch are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, BodyCom, chipKIT, chipKIT logo, CodeGuard, dsPICDEM, dsPICDEM.net, ECAN, In-Circuit Serial Programming, ICSP, Inter-Chip Connectivity, KlearNet, KlearNet logo, MiWi, motorBench, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, RightTouch logo, REAL ICE, SQI, Serial Quad I/O, Total Endurance, TSHARC, USBCheck, VariSense, ViewSpan, WiperLock, Wireless DNA, 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.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

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

© 2013-2016, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

ISBN: 978-1-5224-0193-3

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

*Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.*



# MICROCHIP

## Worldwide Sales and Service

### AMERICAS

#### Corporate Office

2355 West Chandler Blvd.  
Chandler, AZ 85224-6199

Tel: 480-792-7200

Fax: 480-792-7277

Technical Support:

[http://www.microchip.com/  
support](http://www.microchip.com/support)

Web Address:

[www.microchip.com](http://www.microchip.com)

#### Atlanta

Duluth, GA

Tel: 678-957-9614

Fax: 678-957-1455

#### Austin, TX

Tel: 512-257-3370

#### Boston

Westborough, MA

Tel: 774-760-0087

Fax: 774-760-0088

#### Chicago

Itasca, IL

Tel: 630-285-0071

Fax: 630-285-0075

#### Cleveland

Independence, OH

Tel: 216-447-0464

Fax: 216-447-0643

#### Dallas

Addison, TX

Tel: 972-818-7423

Fax: 972-818-2924

#### Detroit

Novi, MI

Tel: 248-848-4000

#### Houston, TX

Tel: 281-894-5983

#### Indianapolis

Noblesville, IN

Tel: 317-773-8323

Fax: 317-773-5453

#### Los Angeles

Mission Viejo, CA

Tel: 949-462-9523

Fax: 949-462-9608

#### New York, NY

Tel: 631-435-6000

#### San Jose, CA

Tel: 408-735-9110

#### Canada - Toronto

Tel: 905-673-0699

Fax: 905-673-6509

### ASIA/PACIFIC

#### Asia Pacific Office

Suites 3707-14, 37th Floor  
Tower 6, The Gateway  
Harbour City, Kowloon

#### Hong Kong

Tel: 852-2943-5100

Fax: 852-2401-3431

#### Australia - Sydney

Tel: 61-2-9868-6733

Fax: 61-2-9868-6755

#### China - Beijing

Tel: 86-10-8569-7000

Fax: 86-10-8528-2104

#### China - Chengdu

Tel: 86-28-8665-5511

Fax: 86-28-8665-7889

#### China - Chongqing

Tel: 86-23-8980-9588

Fax: 86-23-8980-9500

#### China - Dongguan

Tel: 86-769-8702-9880

#### China - Hangzhou

Tel: 86-571-8792-8115

Fax: 86-571-8792-8116

#### China - Hong Kong SAR

Tel: 852-2943-5100

Fax: 852-2401-3431

#### China - Nanjing

Tel: 86-25-8473-2460

Fax: 86-25-8473-2470

#### China - Qingdao

Tel: 86-532-8502-7355

Fax: 86-532-8502-7205

#### China - Shanghai

Tel: 86-21-5407-5533

Fax: 86-21-5407-5066

#### China - Shenyang

Tel: 86-24-2334-2829

Fax: 86-24-2334-2393

#### China - Shenzhen

Tel: 86-755-8864-2200

Fax: 86-755-8203-1760

#### China - Wuhan

Tel: 86-27-5980-5300

Fax: 86-27-5980-5118

#### China - Xian

Tel: 86-29-8833-7252

Fax: 86-29-8833-7256

### ASIA/PACIFIC

#### China - Xiamen

Tel: 86-592-2388138

Fax: 86-592-2388130

#### China - Zhuhai

Tel: 86-756-3210040

Fax: 86-756-3210049

#### India - Bangalore

Tel: 91-80-3090-4444

Fax: 91-80-3090-4123

#### India - New Delhi

Tel: 91-11-4160-8631

Fax: 91-11-4160-8632

#### India - Pune

Tel: 91-20-3019-1500

#### Japan - Osaka

Tel: 81-6-6152-7160

Fax: 81-6-6152-9310

#### Japan - Tokyo

Tel: 81-3-6880-3770

Fax: 81-3-6880-3771

#### Korea - Daegu

Tel: 82-53-744-4301

Fax: 82-53-744-4302

#### Korea - Seoul

Tel: 82-2-554-7200

Fax: 82-2-558-5932 or

82-2-558-5934

#### Malaysia - Kuala Lumpur

Tel: 60-3-6201-9857

Fax: 60-3-6201-9859

#### Malaysia - Penang

Tel: 60-4-227-8870

Fax: 60-4-227-4068

#### Philippines - Manila

Tel: 63-2-634-9065

Fax: 63-2-634-9069

#### Singapore

Tel: 65-6334-8870

Fax: 65-6334-8850

#### Taiwan - Hsin Chu

Tel: 886-3-5778-366

Fax: 886-3-5770-955

#### Taiwan - Kaohsiung

Tel: 886-7-213-7828

#### Taiwan - Taipei

Tel: 886-2-2508-8600

Fax: 886-2-2508-0102

#### Thailand - Bangkok

Tel: 66-2-694-1351

Fax: 66-2-694-1350

### EUROPE

#### Austria - Wels

Tel: 43-7242-2244-39

Fax: 43-7242-2244-393

#### Denmark - Copenhagen

Tel: 45-4450-2828

Fax: 45-4485-2829

#### France - Paris

Tel: 33-1-69-53-63-20

Fax: 33-1-69-30-90-79

#### Germany - Dusseldorf

Tel: 49-2129-3766400

#### Germany - Karlsruhe

Tel: 49-721-625370

#### Germany - Munich

Tel: 49-89-627-144-0

Fax: 49-89-627-144-44

#### Italy - Milan

Tel: 39-0331-742611

Fax: 39-0331-466781

#### Italy - Venice

Tel: 39-049-7625286

#### Netherlands - Drunen

Tel: 31-416-690399

Fax: 31-416-690340

#### Poland - Warsaw

Tel: 48-22-3325737

#### Spain - Madrid

Tel: 34-91-708-08-90

Fax: 34-91-708-08-91

#### Sweden - Stockholm

Tel: 46-8-5090-4654

#### UK - Wokingham

Tel: 44-118-921-5800

Fax: 44-118-921-5820

07/14/15

# Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

## Microchip:

[PIC16F570-E/MV](#) [PIC16F570-E/SO](#) [PIC16F570-E/SP](#) [PIC16F570-E/SS](#) [PIC16F570-I/MV](#) [PIC16F570-I/SO](#)  
[PIC16F570-I/SP](#) [PIC16F570-I/SS](#) [PIC16F570T-I/MV](#) [PIC16F570T-I/SO](#) [PIC16F570T-I/SS](#)

## Данный компонент на территории Российской Федерации

### Вы можете приобрести в компании MosChip.

Для оперативного оформления запроса Вам необходимо перейти по данной ссылке:

<http://moschip.ru/get-element>

Вы можете разместить у нас заказ для любого Вашего проекта, будь то серийное производство или разработка единичного прибора.

В нашем ассортименте представлены ведущие мировые производители активных и пассивных электронных компонентов.

Нашей специализацией является поставка электронной компонентной базы двойного назначения, продукции таких производителей как XILINX, Intel (ex.ALTERA), Vicor, Microchip, Texas Instruments, Analog Devices, Mini-Circuits, Amphenol, Glenair.

Сотрудничество с глобальными дистрибьюторами электронных компонентов, предоставляет возможность заказывать и получать с международных складов практически любой перечень компонентов в оптимальные для Вас сроки.

На всех этапах разработки и производства наши партнеры могут получить квалифицированную поддержку опытных инженеров.

Система менеджмента качества компании отвечает требованиям в соответствии с ГОСТ Р ИСО 9001, ГОСТ РВ 0015-002 и ЭС РД 009

### Офис по работе с юридическими лицами:

105318, г.Москва, ул.Щербаковская д.3, офис 1107, 1118, ДЦ «Щербаковский»

Телефон: +7 495 668-12-70 (многоканальный)

Факс: +7 495 668-12-70 (доб.304)

E-mail: [info@moschip.ru](mailto:info@moschip.ru)

Skype отдела продаж:

moschip.ru

moschip.ru\_4

moschip.ru\_6

moschip.ru\_9