



**PIC16F72**  
**Data Sheet**

28-Pin, 8-Bit CMOS FLASH  
Microcontroller with A/D Converter

---

**Note the following details of the code protection feature on PICmicro® MCUs.**

- The PICmicro family meets the specifications contained in the Microchip Data Sheet.
- Microchip believes that its family of PICmicro microcontrollers is one of the most secure products 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 PICmicro microcontroller in a manner outside the operating specifications contained in the data sheet. The person doing so may be 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 product.

If you have any further questions about this matter, please contact the local sales office nearest to you.

---

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contained in this publication regarding device applications and the like is intended through suggestion only and may be superseded by updates.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

#### Trademarks

The Microchip name and logo, the Microchip logo, FilterLab, KEELOQ, microID, MPLAB, PIC, PICmicro, PICMASTER, PICSTART, PRO MATE, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

dsPIC, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, microPort, Migratable Memory, MPASM, MPLIB, MPLINK, MPSIM, MXDEV, MXLAB, PICC, PICDEM, PICDEM.net, rPIC, Select Mode and Total Endurance are trademarks of Microchip Technology Incorporated in the U.S.A.

Serialized Quick Turn Programming (SQTP) is a service mark of Microchip Technology Incorporated in the U.S.A.

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

© 2002, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

 Printed on recycled paper.



*Microchip received QS-9000 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona in July 1999 and Mountain View, California in March 2002. The Company's quality system processes and procedures are QS-9000 compliant for its PICmicro® 8-bit MCUs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, non-volatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001 certified.*

## 28-Pin, 8-Bit CMOS FLASH MCU with A/D Converter

### Device Included:

- PIC16F72

### High Performance RISC CPU:

- Only 35 single word instructions to learn
- All single cycle instructions except for program branches, which are two-cycle
- Operating speed: DC - 20 MHz clock input  
DC - 200 ns instruction cycle
- 2K x 14 words of Program Memory,  
128 x 8 bytes of Data Memory (RAM)
- Pinout compatible to PIC16C72/72A and  
PIC16F872
- Interrupt capability
- Eight-level deep hardware stack
- Direct, Indirect and Relative Addressing modes

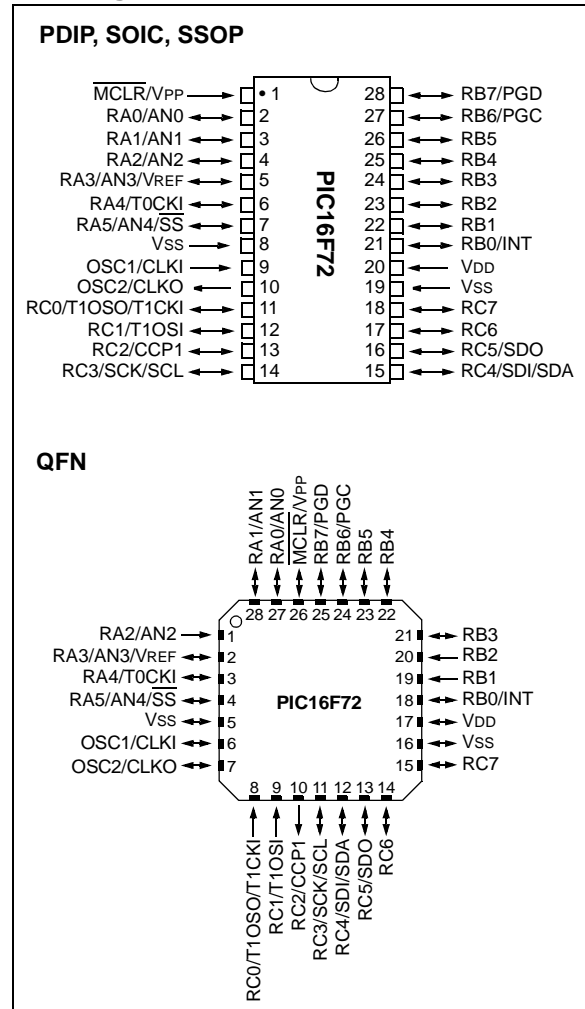
### Peripheral Features:

- High Sink/Source Current: 25 mA
- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler,  
can be incremented during SLEEP via external  
crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period  
register, prescaler and postscaler
- Capture, Compare, PWM (CCP) module
  - Capture is 16-bit, max. resolution is 12.5 ns
  - Compare is 16-bit, max. resolution is 200 ns
  - PWM max. resolution is 10-bit
- 8-bit, 5-channel analog-to-digital converter
- Synchronous Serial Port (SSP) with  
SPI™ (Master/Slave) and I<sup>2</sup>C™ (Slave)
- Brown-out detection circuitry for  
Brown-out Reset (BOR)

### CMOS Technology:

- Low power, high speed CMOS FLASH technology
- Fully static design
- Wide operating voltage range: 2.0V to 5.5V
- Industrial temperature range
- Low power consumption:
  - < 0.6 mA typical @ 3V, 4 MHz
  - 20 µA typical @ 3V, 32 kHz
  - < 1 µA typical standby current

### Pin Diagrams



### Special Microcontroller Features:

- 1,000 erase/write cycle FLASH program memory typical
- Power-on Reset (POR), Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code protection
- Power saving SLEEP mode
- Selectable oscillator options
- In-Circuit Serial Programming™ (ICSP™) via 2 pins
- Processor read access to program memory

# PIC16F72

---

---

Key Reference Manual Features	PIC16F72
Operating Frequency	DC - 20 MHz
RESETS and (Delays)	POR, BOR, (PWRT, OST)
FLASH Program Memory - (14-bit words, 1000 E/W cycles)	2K
Data Memory - RAM (8-bit bytes)	128
Interrupts	8
I/O Ports	PORTA, PORTB, PORTC
Timers	Timer0, Timer1, Timer2
Capture/Compare/PWM Modules	1
Serial Communications	SSP
8-bit A/D Converter	5 channels
Instruction Set (No. of Instructions)	35

## Table of Contents

1.0	Device Overview .....	5
2.0	Memory Organization .....	7
3.0	I/O Ports .....	21
4.0	Reading Program Memory .....	27
5.0	Timer0 Module .....	29
6.0	Timer1 Module .....	31
7.0	Timer2 Module .....	35
8.0	Capture/Compare/PWM (CCP) Module .....	37
9.0	Synchronous Serial Port (SSP) Module .....	43
10.0	Analog-to-Digital Converter (A/D) Module .....	53
11.0	Special Features of the CPU .....	59
12.0	Instruction Set Summary .....	73
13.0	Development Support .....	81
14.0	Electrical Characteristics .....	87
15.0	DC and AC Characteristics Graphs and Tables .....	107
16.0	Package Marking Information .....	117
Appendix A:	Revision History .....	123
Appendix B:	Conversion Considerations .....	123
Index	.....	125
On-Line Support	.....	131
Reader Response	.....	132
Product Identification System	.....	133

## 1.0 DEVICE OVERVIEW

This document contains device specific information for the operation of the PIC16F72 device. Additional information may be found in the PICmicro™ Mid-Range MCU Reference Manual (DS33023), which may be downloaded from the Microchip website. The Reference Manual should be considered a complementary document to this data sheet, and is highly recommended reading for a better understanding of the device architecture and operation of the peripheral modules.

The PIC16F72 belongs to the Mid-Range family of the PICmicro devices. A block diagram of the device is shown in Figure 1-1.

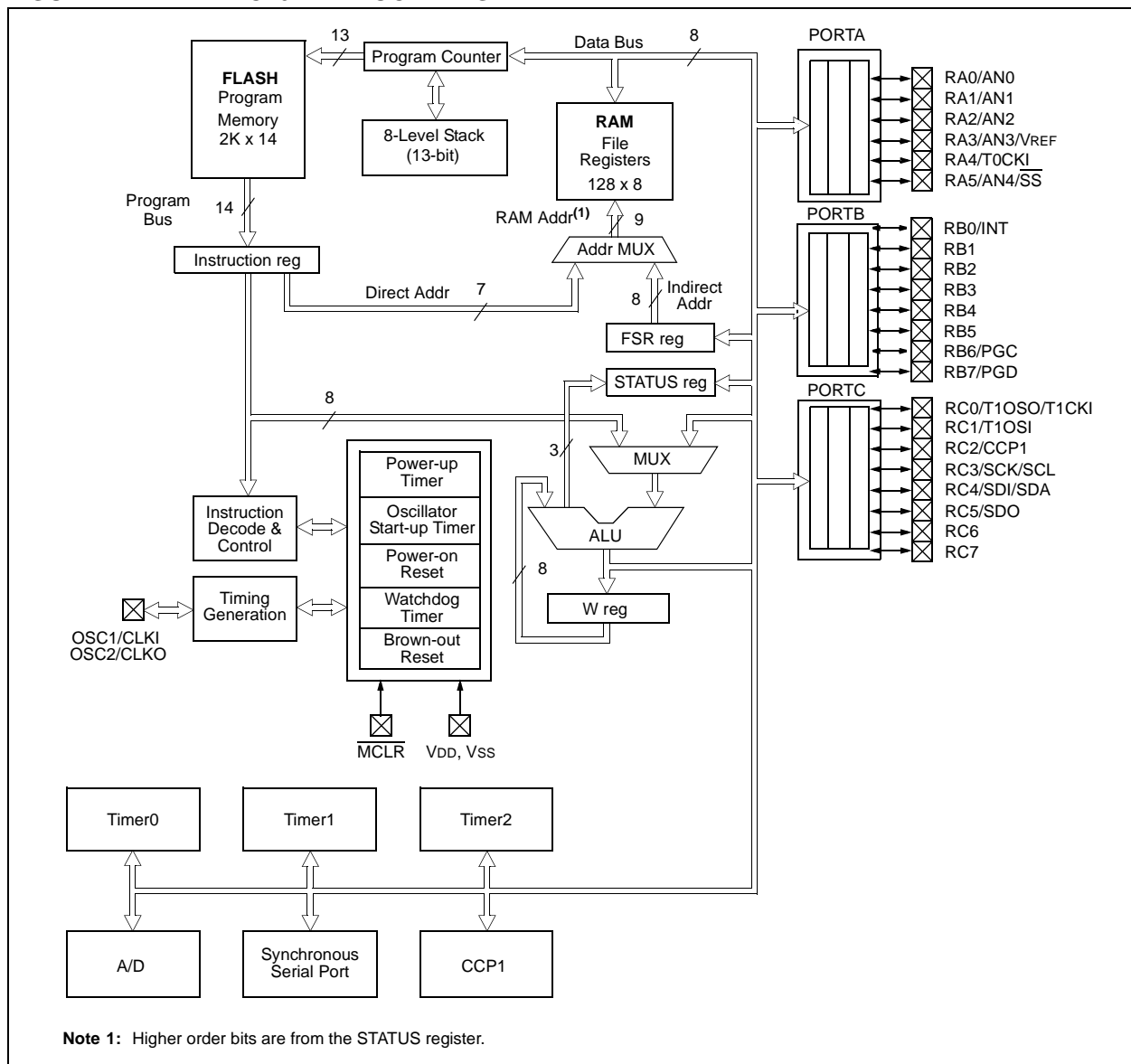
The program memory contains 2K words, which translate to 2048 instructions, since each 14-bit program memory word is the same width as each device instruction. The data memory (RAM) contains 128 bytes.

There are 22 I/O pins that are user configurable on a pin-to-pin basis. Some pins are multiplexed with other device functions. These functions include:

- External interrupt
- Change on PORTB interrupt
- Timer0 clock input
- Timer1 clock/oscillator
- Capture/Compare/PWM
- A/D converter
- SPI/I<sup>2</sup>C

Table 1-1 details the pinout of the device with descriptions and details for each pin.

**FIGURE 1-1: PIC16F72 BLOCK DIAGRAM**



# PIC16F72

**TABLE 1-1: PIC16F72 PINOUT DESCRIPTION**

Pin Name	PDIP, SOIC, SSOP Pin#	MLF Pin#	I/O/P Type	Buffer Type	Description
OSC1/CLKI	9	6	I	ST/CMOS <sup>(3)</sup>	Oscillator crystal input/external clock source input.
OSC2/CLKO	10	7	O	—	Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode. In RC mode, the OSC2 pin outputs CLKO, which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate.
MCLR/VPP	1	26	I/P	ST	Master Clear (Reset) input or programming voltage input. This pin is an active low RESET to the device.
RA0/AN0	2	27	I/O	TTL	PORTA is a bi-directional I/O port. RA0 can also be analog input0.
RA1/AN1	3	28	I/O	TTL	RA1 can also be analog input1.
RA2/AN2	4	1	I/O	TTL	RA2 can also be analog input2.
RA3/AN3/VREF	5	2	I/O	TTL	RA3 can also be analog input3 or analog reference voltage.
RA4/T0CKI	6	3	I/O	ST	RA4 can also be the clock input to the Timer0 module. Output is open drain type.
RA5/AN4/SS	7	4	I/O	TTL	RA5 can also be analog input4 or the slave select for the synchronous serial port.
RB0/INT	21	18	I/O	TTL/ST <sup>(1)</sup>	PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs. RB0 can also be the external interrupt pin.
RB1	22	19	I/O	TTL	
RB2	23	20	I/O	TTL	
RB3	24	21	I/O	TTL	
RB4	25	22	I/O	TTL	Interrupt-on-change pin.
RB5	26	23	I/O	TTL	Interrupt-on-change pin.
RB6/PGC	27	24	I/O	TTL/ST <sup>(2)</sup>	Interrupt-on-change pin. Serial programming clock.
RB7/PGD	28	25	I/O	TTL/ST <sup>(2)</sup>	Interrupt-on-change pin. Serial programming data.
RC0/T1OSO/ T1CKI	11	8	I/O	ST	PORTC is a bi-directional I/O port. RC0 can also be the Timer1 oscillator output or Timer1 clock input.
RC1/T1OSI	12	9	I/O	ST	RC1 can also be the Timer1 oscillator input.
RC2/CCP1	13	10	I/O	ST	RC2 can also be the Capture1 input/Compare1 output/ PWM1 output.
RC3/SCK/SCL	14	11	I/O	ST	RC3 can also be the synchronous serial clock input/output for both SPI and I <sup>2</sup> C modes.
RC4/SDI/SDA	15	12	I/O	ST	RC4 can also be the SPI Data In (SPI mode) or Data I/O (I <sup>2</sup> C mode).
RC5/SDO	16	13	I/O	ST	RC5 can also be the SPI Data Out (SPI mode).
RC6	17	14	I/O	ST	
RC7	18	15	I/O	ST	
VSS	8, 19	5, 16	P	—	Ground reference for logic and I/O pins.
VDD	20	17	P	—	Positive supply 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

- Note 1:** This buffer is a Schmitt Trigger input when configured as the external interrupt.  
**Note 2:** This buffer is a Schmitt Trigger input when used in Serial Programming mode.  
**Note 3:** This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.

## 2.0 MEMORY ORGANIZATION

There are two memory blocks in the PIC16F72 device. These are the program memory and the data memory. Each block has separate buses so that concurrent access can occur. Program memory and data memory are explained in this section. Program memory can be read internally by the user code (see Section 4.0).

The data memory can further be broken down into the general purpose RAM and the Special Function Registers (SFRs). The operation of the SFRs that control the “core” are described here. The SFRs used to control the peripheral modules are described in the section discussing each individual peripheral module.

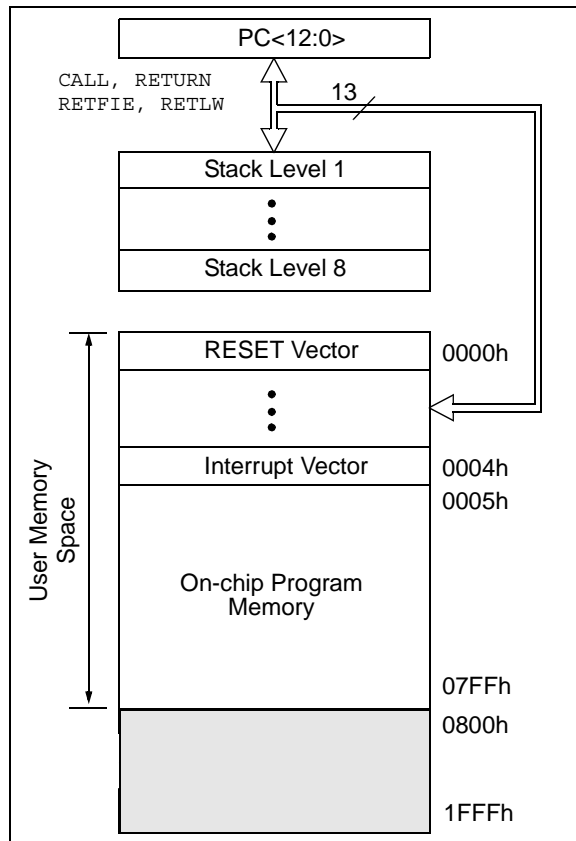
Additional information on device memory may be found in the PICmicro™ Mid-Range Reference Manual, (DS33023). [www.trimmer.ru](http://www.trimmer.ru)

### 2.1 Program Memory Organization

PIC16F72 devices have a 13-bit program counter capable of addressing a 8K x 14 program memory space. The address range for this program memory is 0000h - 07FFh. Accessing a location above the physically implemented address will cause a wraparound.

The RESET Vector is at 0000h and the Interrupt Vector is at 0004h.

**FIGURE 2-1: PROGRAM MEMORY MAP AND STACK**



## 2.2 Data Memory Organization

The Data Memory is partitioned into multiple banks that contain the General Purpose Registers and the Special Function Registers. Bits RP1 (STATUS<6>) and RP0 (STATUS<5>) are the bank select bits.

RP1:RP0	Bank
00	0
01	1
10	2
11	3

Each bank extends up to 7Fh (128 bytes). The lower locations of each bank are reserved for the Special Function Registers. Above the Special Function Registers are General Purpose Registers, implemented as static RAM.

All implemented banks contain SFRs. Some “high use” SFRs from one bank may be mirrored in another bank, for code reduction and quicker access (e.g., the STATUS register is in Banks 0 - 3).

### 2.2.1 GENERAL PURPOSE REGISTER FILE

The register file can be accessed either directly, or indirectly, through the File Select Register FSR (see Section 2.5).



# PIC16F72

**FIGURE 2-2: PIC16F72 REGISTER FILE MAP**

File Address		File Address		File Address		File Address	
Indirect addr.(*)	00h	Indirect addr.(*)	80h	Indirect addr.(*)	100h	Indirect addr.(*)	180h
TMR0	01h	OPTION	81h	TMR0	101h	OPTION	181h
PCL	02h	PCL	82h	PCL	102h	PCL	182h
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183h
FSR	04h	FSR	84h	FSR	104h	FSR	184h
PORTA	05h	TRISA	85h		105h		185h
PORTB	06h	TRISB	86h	PORTB	106h	TRISB	186h
PORTC	07h	TRISC	87h		107h		187h
	08h		88h		108h		188h
	09h		89h		109h		189h
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh
PIR1	0Ch	PIE1	8Ch	PMDATL	10Ch	PMCON1	18Ch
	0Dh		8Dh	PMADRL	10Dh		18Dh
TMR1L	0Eh	PCON	8Eh	PMDATH	10Eh		18Eh
TMR1H	0Fh		8Fh	PMADRH	10Fh		18Fh
T1CON	10h		90h		110h		190h
TMR2	11h		91h				
T2CON	12h	PR2	92h				
SSPBUF	13h	SSPADD	93h				
SSPCON	14h	SSPSTAT	94h				
CCPR1L	15h		95h				
CCPR1H	16h		96h				
CCP1CON	17h		97h				
	18h		98h				
	19h		99h				
	1Ah		9Ah				
	1Bh		9Bh				
	1Ch		9Ch				
	1Dh		9Dh				
ADRES	1Eh		9Eh				
ADCON0	1Fh	ADCON1	9Fh		11Fh		19Fh
General Purpose Register 96 Bytes	20h	General Purpose Register 32 Bytes	A0h	accesses 20h-7Fh	120h	accesses A0h -BFh	1A0h
			BFh C0h				
		accesses 40h-7Fh				accesses 40h -7Fh	1C0h
Bank 0	7Fh	Bank 1	FFh	Bank 2	17Fh	Bank 3	1FFh

Unimplemented data memory locations, read as '0'.  
 \* Not a physical register.

## 2.2.2 SPECIAL FUNCTION REGISTERS

The Special Function Registers are registers used by the CPU and peripheral modules for controlling the desired operation of the device. These registers are implemented as static RAM. A list of these registers is given in Table 2-1.

The Special Function Registers can be classified into two sets: core (CPU) and peripheral. Those registers associated with the core functions are described in detail in this section. Those related to the operation of the peripheral features are described in detail in the peripheral feature section.

**TABLE 2-1: 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	Details on page:
<b>Bank 0</b>											
00h <sup>(1)</sup>	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	19
01h	TMR0	Timer0 Module's Register								xxxx xxxx	29,13
02h <sup>(1)</sup>	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	18
03h <sup>(1)</sup>	STATUS	IRP	RP1	RP0	TO	PD	Z	DC	C	0001 1xxx	12
04h <sup>(1)</sup>	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	19
05h	PORTA	—	—	PORTA Data Latch when written: PORTA pins when read						--0x 0000	21
06h	PORTB	PORTB Data Latch when written: PORTB pins when read								xxxx xxxx	23
07h	PORTC	PORTC Data Latch when written: PORTC pins when read								xxxx xxxx	25
08h	—	Unimplemented								—	—
09h	—	Unimplemented								—	—
0Ah <sup>(1,2)</sup>	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	18
0Bh <sup>(1)</sup>	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF	0000 000x	14
0Ch	PIR1	—	ADIF	—	—	SSPIF	CCP1IF	TMR2IF	TMR1IF	-0-- 0000	16
0Dh	—	Unimplemented								—	—
0Eh	TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	31
0Fh	TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	31
10h	T1CON	—	—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	--00 0000	31
11h	TMR2	Timer2 Module's Register								0000 0000	35
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	36
13h	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								xxxx xxxx	43,48
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	45
15h	CCPR1L	Capture/Compare/PWM Register (LSB)								xxxx xxxx	38,39,41
16h	CCPR1H	Capture/Compare/PWM Register (MSB)								xxxx xxxx	38,39,41
17h	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	37
18h-1Dh	—	Unimplemented								—	—
1Eh	ADRES	A/D Result Register								xxxx xxxx	53
1Fh	ADCON0	ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	—	ADON	0000 00-0	53

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, read as '0', r = reserved.  
Shaded locations are unimplemented, read as '0'.

- Note 1:** These registers can be addressed from any bank.  
**Note 2:** The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8> whose contents are transferred to the upper byte of the program counter.  
**Note 3:** This bit always reads as a '1'.

# PIC16F72

**TABLE 2-1: 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	Details on page:
<b>Bank 1</b>											
80h <sup>(1)</sup>	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	19
81h	OPTION	RBP $\bar{U}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	13
82h <sup>(1)</sup>	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	18
83h <sup>(1)</sup>	STATUS	IRP	RP1	RP0	$\bar{T}O$	$\bar{P}D$	Z	DC	C	0001 1xxx	12
84h <sup>(1)</sup>	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	19
85h	TRISA	—	—	PORTA Data Direction Register						--11 1111	21
86h	TRISB	PORTB Data Direction Register								1111 1111	23
87h	TRISC	PORTC Data Direction Register								1111 1111	25
88h	—	Unimplemented								—	—
89h	—	Unimplemented								—	—
8Ah <sup>(1,2)</sup>	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the PC					---0 0000	18
8Bh <sup>(1)</sup>	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF	0000 000x	14
8Ch	PIE1	—	ADIE	—	—	SSPIE	CCP1IE	TMR2IE	TMR1IE	-0-- 0000	15
8Dh	—	Unimplemented								—	—
8Eh	PCON	—	—	—	—	—	—	$\bar{P}OR$	$\bar{B}OR$	---- --qg	17
8Fh	—	Unimplemented								—	—
90h	—	Unimplemented								—	—
91h	—	Unimplemented								—	—
92h	PR2	Timer2 Period Register								1111 1111	41
93h	SSPADD	Synchronous Serial Port (I <sup>2</sup> C mode) Address Register								0000 0000	43,48
94h	SSPSTAT	SMP	CKE	D $\bar{A}$	P	S	R $\bar{W}$	UA	BF	0000 0000	44
95h	—	Unimplemented								—	—
96h	—	Unimplemented								—	—
97h	—	Unimplemented								—	—
98h	—	Unimplemented								—	—
99h	—	Unimplemented								—	—
9Ah	—	Unimplemented								—	—
9Bh	—	Unimplemented								—	—
9Ch	—	Unimplemented								—	—
9Dh	—	Unimplemented								—	—
9Eh	—	Unimplemented								—	—
9Fh	ADCON1	—	—	—	—	—	PCFG2	PCFG1	PCFG0	---- -000	54

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, read as '0', r = reserved.  
 Shaded locations are unimplemented, read as '0'.

- Note**
- 1: These registers can be addressed from any bank.
  - 2: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8> whose contents are transferred to the upper byte of the program counter.
  - 3: This bit always reads as a '1'.

**TABLE 2-1: 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	Details on page:	
<b>Bank 2</b>												
100h <sup>(1)</sup>	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)									0000 0000	19
101h	TMR0	Timer0 Module's Register									xxxx xxxx	29
102h <sup>(1)</sup>	PCL	Program Counter's (PC) Least Significant Byte									0000 0000	18
103h <sup>(1)</sup>	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	12	
104h <sup>(1)</sup>	FSR	Indirect Data Memory Address Pointer									xxxx xxxx	19
105h	—	Unimplemented									—	—
106h	PORTB	PORTB Data Latch when written: PORTB pins when read									xxxx xxxx	23
107h	—	Unimplemented									—	—
108h	—	Unimplemented									—	—
109h	—	Unimplemented									—	—
10Ah <sup>(1,2)</sup>	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	18	
10Bh <sup>(1)</sup>	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF	0000 000x	14	
10Ch	PMDATL	Data Register Low Byte									xxxx xxxx	27
10Dh	PMADRL	Address Register Low Byte									xxxx xxxx	27
10Eh	PMDATH	—	—	Data Register High Byte					--xx xxxx	27		
10Fh	PMADRH	—	—	—	Address Register High Byte					---x xxxx	27	
<b>Bank 3</b>												
180h <sup>(1)</sup>	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)									0000 0000	19
181h	OPTION	$\overline{RBPU}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	13	
182h <sup>(1)</sup>	PCL	Program Counter's (PC) Least Significant Byte									0000 0000	18
183h <sup>(1)</sup>	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	12	
184h <sup>(1)</sup>	FSR	Indirect Data Memory Address Pointer									xxxx xxxx	19
185h	—	Unimplemented									—	—
186h	TRISB	PORTB Data Direction Register									1111 1111	23
187h	—	Unimplemented									—	—
188h	—	Unimplemented									—	—
189h	—	Unimplemented									—	—
18Ah <sup>(1,2)</sup>	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	18	
18Bh <sup>(1)</sup>	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF	0000 000x	14	
18Ch	PMCON1	— <sup>(3)</sup>	—	—	—	—	—	—	RD	1--- ---0	27	
18Dh	—	Unimplemented									—	—
18Eh	—	Reserved, maintain clear									0000 0000	—
18Fh	—	Reserved, maintain clear									0000 0000	—

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, read as '0', r = reserved.  
Shaded locations are unimplemented, read as '0'.

- Note**
- 1: These registers can be addressed from any bank.
  - 2: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8> whose contents are transferred to the upper byte of the program counter.
  - 3: This bit always reads as a '1'.

# PIC16F72

## 2.2.2.1 STATUS Register

The STATUS register, shown in Register 2-1, contains the arithmetic status of the ALU, the RESET status and the bank select bits for data memory.

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

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

It is recommended, therefore, that only `BCF`, `BSF`, `SWAPF` and `MOVWF` instructions are used to alter the STATUS register, because these instructions do not affect the Z, C or DC bits from the STATUS register. For other instructions, not affecting any status bits, see Section 12.0, Instruction Set Summary.

**Note 1:** The  $\overline{\text{C}}$  and  $\overline{\text{DC}}$  bits operate as a borrow and digit borrow bit, respectively, in subtraction. See the `SUBLW` and `SUBWF` instructions for examples.

### REGISTER 2-1: STATUS REGISTER (ADDRESS 03h, 83h, 103h, 183h)

	R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
	IRP	RP1	RP0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C
bit 7								bit 0

- bit 7 **IRP:** Register Bank Select bit (used for indirect addressing)  
 1 = Bank 2, 3 (100h - 1FFh)  
 0 = Bank 0, 1 (00h - FFh)
- bit 6-5 **RP<1:0>:** Register Bank Select bits (used for direct addressing)  
 11 = Bank 3 (180h - 1FFh)  
 10 = Bank 2 (100h - 17Fh)  
 01 = Bank 1 (80h - FFh)  
 00 = Bank 0 (00h - 7Fh)  
 Each bank is 128 bytes
- bit 4  **$\overline{\text{TO}}$ :** Time-out bit  
 1 = After power-up, `CLRWDT` instruction, or `SLEEP` instruction  
 0 = A WDT time-out occurred
- bit 3 **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 (`ADDWF`, `ADDLW`, `SUBLW` and `SUBWF` instructions)<sup>(1)</sup>  
 1 = A carry-out from the 4th low order bit of the result occurred  
 0 = No carry-out from the 4th low order bit of the result
- bit 0 **C:** Carry/borrow bit (`ADDWF`, `ADDLW`, `SUBLW` and `SUBWF` instructions)<sup>(1,2)</sup>  
 1 = A carry-out from the Most Significant bit of the result occurred  
 0 = No carry-out from the Most Significant bit of the result occurred

**Note 1:** For borrow, the polarity is reversed. A subtraction is executed by adding the two's complement of the second operand.

**2:** For rotate (`RRF`, `RLF`) instructions, this bit is loaded with either the high or low order bit of the source register.

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

## 2.2.2.2 OPTION Register

The OPTION register is a readable and writable register that contains various control bits to configure the TMR0 prescaler/WDT postscaler (single assignable register known also as the prescaler), the External INT Interrupt, TMR0, and the weak pull-ups on PORTB.

**Note:** To achieve a 1:1 prescaler assignment for the TMR0 register, assign the prescaler to the Watchdog Timer.

### REGISTER 2-2: OPTION REGISTER (ADDRESS 81h, 181h)

	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
	$\overline{\text{RBP}}\overline{\text{U}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7								bit 0

- bit 7  **$\overline{\text{RBP}}\overline{\text{U}}$** : PORTB Pull-up Enable bit  
 1 = PORTB pull-ups are disabled  
 0 = PORTB pull-ups are enabled by individual port latch values
- bit 6 **INTEDG**: Interrupt Edge Select bit  
 1 = Interrupt on rising edge of RB0/INT pin  
 0 = Interrupt on falling edge of RB0/INT pin
- bit 5 **T0CS**: TMR0 Clock Source Select bit  
 1 = Transition on RA4/T0CKI pin  
 0 = Internal instruction cycle clock (CLKO)
- bit 4 **T0SE**: TMR0 Source Edge Select bit  
 1 = Increment on high-to-low transition on RA4/T0CKI pin  
 0 = Increment on low-to-high transition on RA4/T0CKI pin
- bit 3 **PSA**: Prescaler Assignment bit  
 1 = Prescaler is assigned to the WDT  
 0 = Prescaler is assigned to the Timer0 module
- bit 2-0 **PS2:PS0**: Prescaler Rate Select bits

Bit Value	TMR0 Rate	WDT Rate
000	1 : 2	1 : 1
001	1 : 4	1 : 2
010	1 : 8	1 : 4
011	1 : 16	1 : 8
100	1 : 32	1 : 16
101	1 : 64	1 : 32
110	1 : 128	1 : 64
111	1 : 256	1 : 128

**Legend:**

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

# PIC16F72

## 2.2.2.3 INTCON Register

The INTCON Register is a readable and writable register that contains various enable and flag bits for the TMR0 register overflow, RB Port change and External RB0/INT pin interrupts.

**Note:** Interrupt flag bits get set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the global enable bit, GIE (INTCON<7>). User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

### REGISTER 2-3: INTCON: INTERRUPT CONTROL REGISTER (ADDRESS 0Bh, 8Bh, 10Bh, 18Bh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF

bit 7

bit 0

- bit 7     **GIE:** Global Interrupt Enable bit  
           1 = Enables all unmasked interrupts  
           0 = Disables all interrupts
- bit 6     **PEIE:** Peripheral Interrupt Enable bit  
           1 = Enables all unmasked peripheral interrupts  
           0 = Disables all peripheral interrupts
- bit 5     **TMR0IE:** TMR0 Overflow Interrupt Enable bit  
           1 = Enables the TMR0 interrupt  
           0 = Disables the TMR0 interrupt
- bit 4     **INTE:** RB0/INT External Interrupt Enable bit  
           1 = Enables the RB0/INT external interrupt  
           0 = Disables the RB0/INT external interrupt
- bit 3     **RBIE:** RB Port Change Interrupt Enable bit  
           1 = Enables the RB port change interrupt  
           0 = Disables the RB port change interrupt
- bit 2     **TMR0IF:** TMR0 Overflow Interrupt Flag bit  
           1 = TMR0 register has overflowed (must be cleared in software)  
           0 = TMR0 register did not overflow
- bit 1     **INTF:** RB0/INT External Interrupt Flag bit  
           1 = The RB0/INT external interrupt occurred (must be cleared in software)  
           0 = The RB0/INT external interrupt did not occur
- bit 0     **RBIF:** RB Port Change Interrupt Flag bit  
           A mismatch condition will continue to set flag bit RBIF. Reading PORTB will end the mismatch condition and allow flag bit RBIF to be cleared.  
           1 = At least one of the RB7:RB4 pins changed state (must be cleared in software)  
           0 = None of the RB7:RB4 pins have changed state

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

## 2.2.2.4 PIE1 Register

This register contains the individual enable bits for the peripheral interrupts.

**Note:** Bit PEIE (INTCON<6>) must be set to enable any peripheral interrupt.

### REGISTER 2-4: PIE1: PERIPHERAL INTERRUPT ENABLE REGISTER 1 (ADDRESS 8Ch)

U-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	ADIE	—	—	SSPIE	CCP1IE	TMR2IE	TMR1IE

bit 7

bit 0

- bit 7      **Unimplemented:** Read as '0'
- bit 6      **ADIE:** A/D Converter Interrupt Enable bit  
1 = Enables the A/D converter interrupt  
0 = Disables the A/D converter interrupt
- bit 5-4    **Unimplemented:** Read as '0'
- bit 3      **SSPIE:** Synchronous Serial Port Interrupt Enable bit  
1 = Enables the SSP interrupt  
0 = Disables the SSP interrupt
- bit 2      **CCP1IE:** CCP1 Interrupt Enable bit  
1 = Enables the CCP1 interrupt  
0 = Disables the CCP1 interrupt
- bit 1      **TMR2IE:** TMR2 to PR2 Match Interrupt Enable bit  
1 = Enables the TMR2 to PR2 match interrupt  
0 = Disables the TMR2 to PR2 match interrupt
- bit 0      **TMR1IE:** TMR1 Overflow Interrupt Enable bit  
1 = Enables the TMR1 overflow interrupt  
0 = Disables the TMR1 overflow interrupt

**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



# PIC16F72

## 2.2.2.5 PIR1 Register

This register contains the individual flag bits for the Peripheral interrupts.

### REGISTER 2-5: PIR1: PERIPHERAL INTERRUPT FLAG REGISTER 1 (ADDRESS 0Ch)

U-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	ADIF	—	—	SSPIF	CCP1IF	TMR2IF	TMR1IF
bit 7							bit 0

- bit 7     **Unimplemented:** Read as '0'
- bit 6     **ADIF:** A/D Converter Interrupt Flag bit  
           1 = An A/D conversion completed  
           0 = The A/D conversion is not complete
- bit 5-4   **Unimplemented:** Read as '0'
- bit 3     **SSPIF:** Synchronous Serial Port (SSP) Interrupt Flag bit  
           1 = The SSP interrupt condition has occurred, and must be cleared in software before returning from the Interrupt Service Routine.  
               The conditions that will set this bit are a transmission/reception has taken place.  
           0 = No SSP interrupt condition has occurred
- bit 2     **CCP1IF:** CCP1 Interrupt Flag bit  
           Capture mode:  
           1 = A TMR1 register capture occurred (must be cleared in software)  
           0 = No TMR1 register capture occurred  
           Compare mode:  
           1 = A TMR1 register compare match occurred (must be cleared in software)  
           0 = No TMR1 register compare match occurred  
           PWM mode:  
           Unused in this mode
- bit 1     **TMR2IF:** TMR2 to PR2 Match Interrupt Flag bit  
           1 = TMR2 to PR2 match occurred (must be cleared in software)  
           0 = No TMR2 to PR2 match occurred
- bit 0     **TMR1IF:** TMR1 Overflow Interrupt Flag bit  
           1 = TMR1 register overflowed (must be cleared in software)  
           0 = TMR1 register did not overflow

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

## 2.2.2.6 PCON Register

**Note:** Interrupt flag bits get set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the global enable bit, GIE (INTCON<7>). User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

**Note:**  $\overline{\text{BOR}}$  is unknown on Power-on Reset. It must then be set by the user and checked on subsequent RESETS to see if  $\overline{\text{BOR}}$  is clear, indicating a brown-out has occurred. The  $\overline{\text{BOR}}$  status bit is a 'don't care' and is not necessarily predictable if the brown-out circuit is disabled (by clearing the BOREN bit in the Configuration word).

The Power Control (PCON) register contains a flag bit to allow differentiation between a Power-on Reset (POR), a Brown-out Reset, an external MCLR Reset and WDT Reset.

### REGISTER 2-6: PCON: POWER CONTROL REGISTER (ADDRESS 8Eh)

	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-x
	—	—	—	—	—	—	$\overline{\text{POR}}$	$\overline{\text{BOR}}$
bit 7							bit 0	

bit 7-2      **Unimplemented:** Read as '0'

bit 1      **POR:** Power-on Reset Status bit

1 = No Power-on Reset occurred

0 = A Power-on Reset occurred (must be set in software after a Power-on Reset occurs)

bit 0      **BOR:** Brown-out Reset Status bit

1 = No Brown-out Reset occurred

0 = A Brown-out Reset occurred (must be set in software after a Brown-out Reset occurs)

**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

# PIC16F72

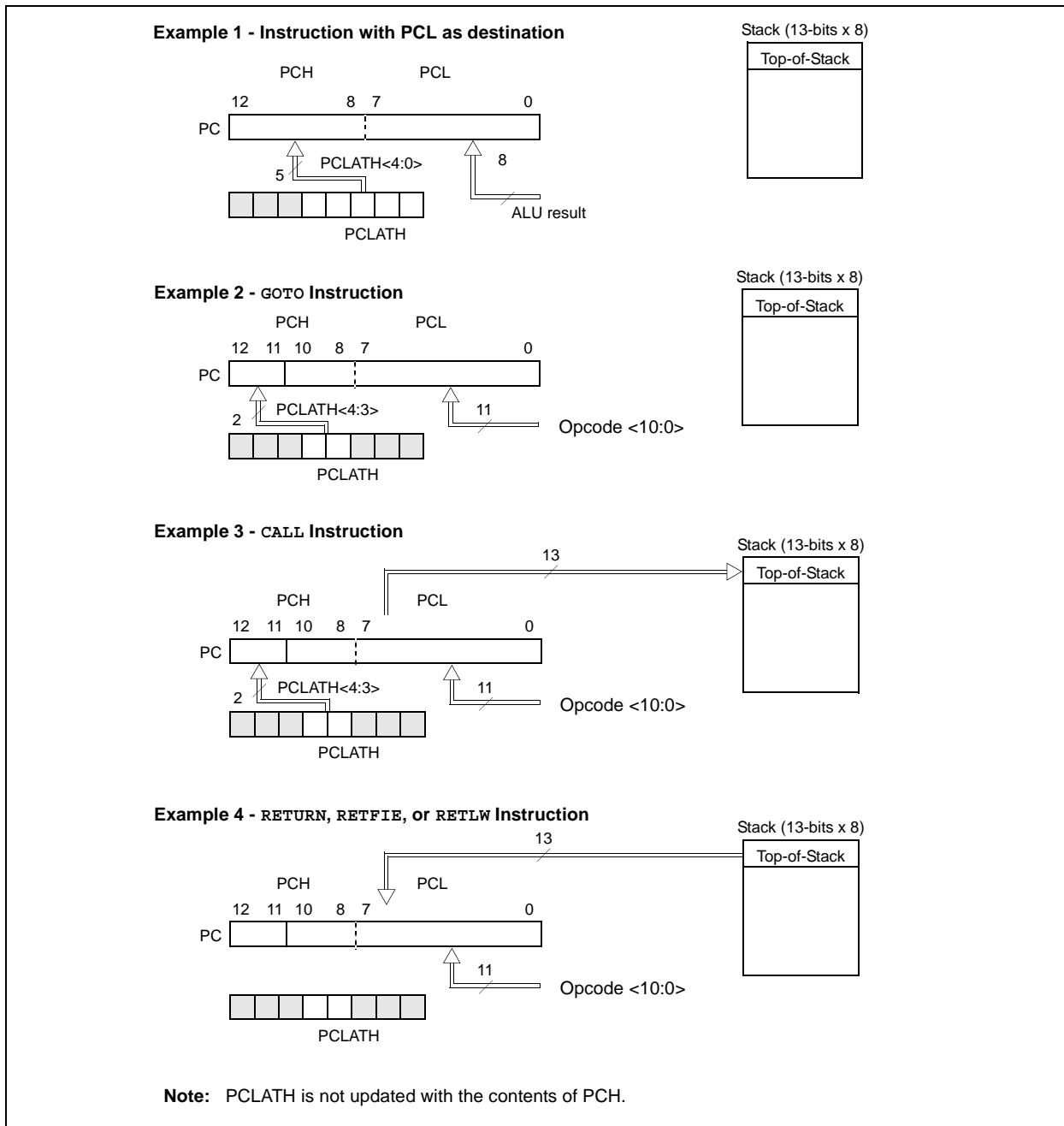
## 2.3 PCL and PCLATH

The program counter (PC) specifies the address of the instruction to fetch for execution. The PC is 13-bits wide. The low byte is called the PCL register. This register is readable and writable. The high byte is called the PCH register. This register contains the PC<12:8> bits and is not directly readable or writable. All updates to the PCH register go through the PCLATH register.

Figure 2-3 shows the four situations for the loading of the PC.

- Example 1 shows how the PC is loaded on a write to PCL (PCLATH<4:0> → PCH).
- Example 2 shows how the PC is loaded during a GOTO instruction (PCLATH<4:3> → PCH).
- Example 3 shows how the PC is loaded during a CALL instruction (PCLATH<4:3> → PCH), with the PC loaded (PUSH'd) onto the Top-of-Stack.
- Example 4 shows how the PC is loaded during one of the return instructions, where the PC is loaded (POP'd) from the Top-of-Stack.

**FIGURE 2-3: LOADING OF PC IN DIFFERENT SITUATIONS**



## 2.3.1 COMPUTED GOTO

A computed `GOTO` is accomplished by adding an offset to the program counter (`ADDWF PCL`). When doing a table read using a computed `GOTO` method, care should be exercised if the table location crosses a PCL memory boundary (each 256-byte block). Refer to the Application Note, "Implementing a Table Read" (AN556).

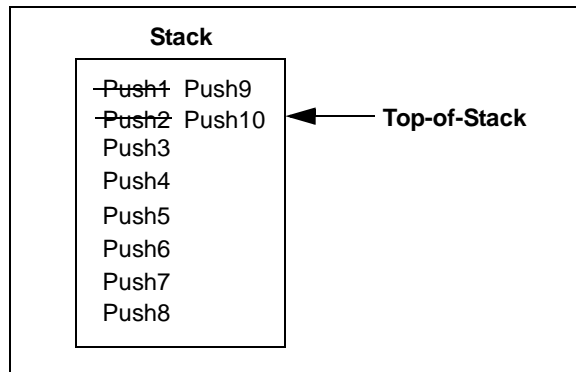
## 2.3.2 STACK

The stack allows a combination of up to eight program calls and interrupts to occur. The stack contains the return address from this branch in program execution.

Mid-range devices have an 8-level deep x 13-bit wide hardware stack. The stack space is not part of either program or data space and the stack pointer is not readable or writable. The PC is `PUSH`'d onto the stack when a `CALL` instruction is executed, or an interrupt causes a branch. The stack is `POP`'d in the event of a `RETURN`, `RETLW` or a `RETFIE` instruction execution. `PCLATH` is not modified when the stack is `PUSH`'d or `POP`'d.

After the stack has been `PUSH`'d eight times, the ninth push overwrites the value that was stored from the first push. The tenth push overwrites the second push (and so on). An example of the overwriting of the stack is shown in Figure 2-4.

**FIGURE 2-4: STACK MODIFICATION**



**Note 1:** There are no status bits to indicate stack overflow or stack underflow conditions.

**2:** There are no instructions/mnemonics called `PUSH` or `POP`. These are actions that occur from the execution of the `CALL`, `RETURN`, `RETLW` and `RETFIE` instructions, or the vectoring to an interrupt address.

## 2.4 Program Memory Paging

The `CALL` and `GOTO` instructions provide 11 bits of address to allow branching within any 2K program memory page. When doing a `CALL` or `GOTO` instruction, the upper two bits of the address are provided by `PCLATH<4:3>`. When doing a `CALL` or `GOTO` instruction, the user must ensure that the page select bits are programmed so that the desired program memory page is addressed. If a return from a `CALL` instruction (or interrupt) is executed, the entire 13-bit PC is pushed onto the stack. Therefore, manipulation of the `PCLATH<4:3>` bits is not required for the return instructions (which `POP`s the address from the stack).

**Note:** The PIC16F72 device ignores the paging bit `PCLATH<4:3>`. The use of `PCLATH<4:3>` as a general purpose read/write bit is not recommended, since this may affect upward compatibility with future products.

## 2.5 Indirect 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.

A simple program to clear RAM locations 20h-2Fh using indirect addressing is shown in Example 2-1.

**EXAMPLE 2-1: INDIRECT ADDRESSING**

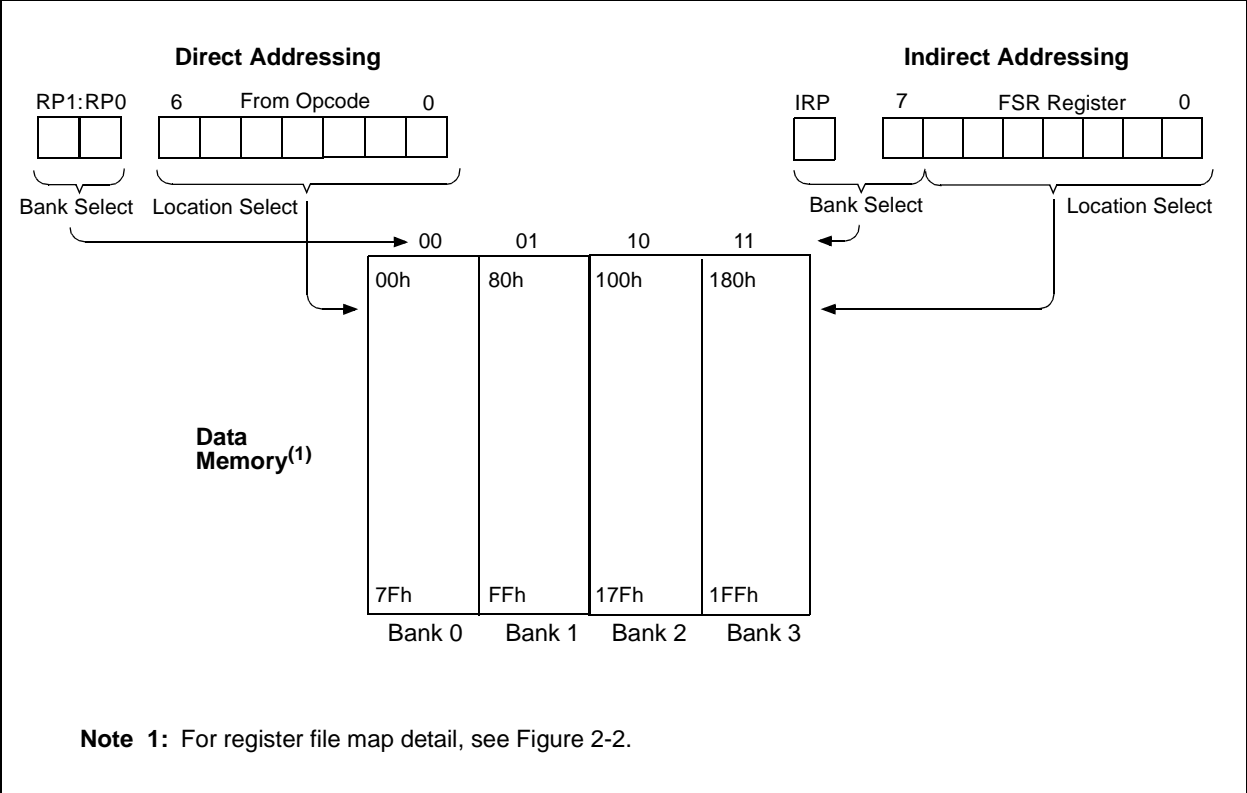
```

        movlw 0x20 ;initialize pointer
        movwf FSR ;to RAM
NEXT    clrf  INDF ;clear INDF register
        incf  FSR ;inc pointer
        btfss FSR,4 ;all done?
        goto  NEXT ;NO, clear next
CONTINUE
        :           ;YES, continue
    
```

An effective 9-bit address is obtained by concatenating the 8-bit `FSR` register and the `IRP` bit (`STATUS<7>`), as shown in Figure 2-5.

# PIC16F72

**FIGURE 2-5: DIRECT/INDIRECT ADDRESSING**



## 3.0 I/O PORTS

Some pins for these I/O ports are multiplexed with an alternate function for the peripheral features on the device. In general, when a peripheral is enabled, that pin may not be used as a general purpose I/O pin.

Additional information on I/O ports may be found in the PICmicro™ Mid-Range MCU Reference Manual, (DS33023).

### 3.1 PORTA and the TRISA Register

PORTA is a 6-bit wide, bi-directional port. The corresponding data direction register is TRISA. Setting a TRISA bit (= 1) will make the corresponding PORTA pin an input (i.e., put the corresponding output driver in a Hi-Impedance mode). Clearing a TRISA bit (= 0) will make the corresponding PORTA pin an output (i.e., put the contents of the output latch on the selected pin).

Reading the PORTA register, reads the status of the pins, whereas writing to it will write to the port latch. All write operations are read-modify-write operations. Therefore, a write to a port implies that the port pins are read, this value is modified and then written to the port data latch.

Pin RA4 is multiplexed with the Timer0 module clock input to become the RA4/T0CKI pin. The RA4/T0CKI pin is a Schmitt Trigger input and an open drain output. All other RA port pins have TTL input levels and full CMOS output drivers.

Other PORTA pins are multiplexed with analog inputs and analog VREF input. The operation of each pin is selected by clearing/setting the control bits in the ADCON1 register (A/D Control Register1).

**Note:** On a Power-on Reset, these pins are configured as analog inputs and read as '0'.

The TRISA register controls the direction of the RA pins, even when they are being used as analog inputs. The user must ensure the bits in the TRISA register are maintained set when using them as analog inputs.

#### EXAMPLE 3-1: INITIALIZING PORTA

```

BANKSEL   PORTA   ; select bank for PORTA
CLRF      PORTA   ; Initialize PORTA by
                ; clearing output
                ; data latches
BANKSEL   ADCON1  ; Select Bank for ADCON1
MOVLW    0x06     ; Configure all pins
MOVWF    ADCON1  ; as digital inputs
MOVLW    0xCF     ; Value used to
                ; initialize data
                ; direction
MOVWF    TRISA   ; Set RA<3:0> as inputs
                ; RA<5:4> as outputs
                ; TRISA<7:6> are always
                ; read as '0'.
    
```

FIGURE 3-1: BLOCK DIAGRAM OF RA3:RA0 AND RA5 PINS

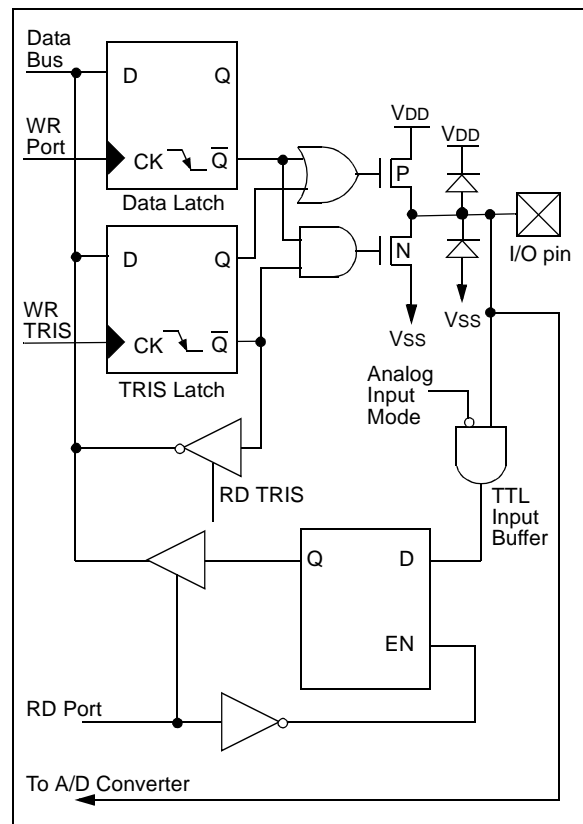
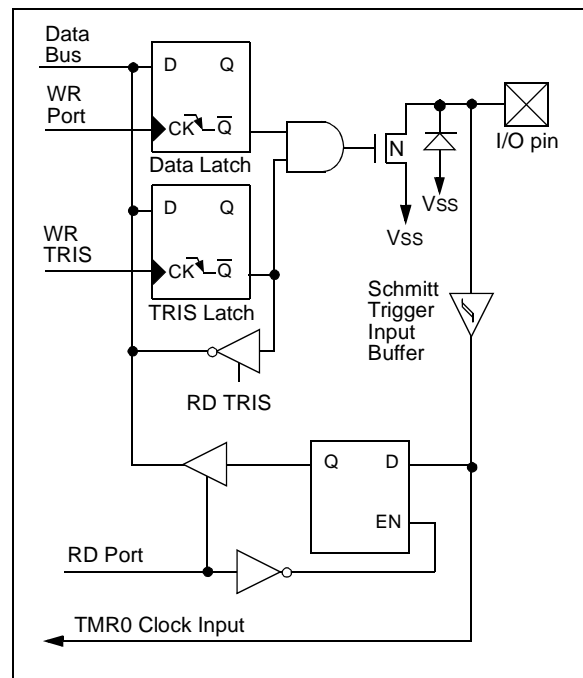


FIGURE 3-2: BLOCK DIAGRAM OF RA4/T0CKI PIN



# PIC16F72

**TABLE 3-1: PORTA FUNCTIONS**

Name	Bit#	Buffer	Function
RA0/AN0	bit 0	TTL	Input/output or analog input.
RA1/AN1	bit 1	TTL	Input/output or analog input.
RA2/AN2	bit 2	TTL	Input/output or analog input.
RA3/AN3/VREF	bit 3	TTL	Input/output or analog input or VREF.
RA4/T0CKI	bit 4	ST	Input/output or external clock input for Timer0. Output is open drain type.
RA5/AN4/SS	bit 5	TTL	Input/output or analog input or slave select input for synchronous serial port.

Legend: TTL = TTL input, ST = Schmitt Trigger input

**TABLE 3-2: SUMMARY OF REGISTERS ASSOCIATED WITH PORTA**

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
05h	PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	--0x 0000	--0u 0000
85h	TRISA	—	—	PORTA Data Direction Register						--11 1111	--11 1111
9Fh	ADCON1	—	—	—	—	—	PCFG2	PCFG1	PCFG0	---- -000	---- -000

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

Shaded cells are not used by PORTA.

**Note:** When using the SSP module in SPI Slave mode and  $\overline{SS}$  enabled, the A/D Port Configuration Control bits (PCFG2:PCFG0) in the A/D Control Register (ADCON1) must be set to one of the following configurations: 100, 101, 11x.

## 3.2 PORTB and the TRISB Register

PORTB is an 8-bit wide, bi-directional port. The corresponding data direction register is TRISB. Setting a TRISB bit (= 1) will make the corresponding PORTB pin an input (i.e., put the corresponding output driver in a Hi-Impedance mode). Clearing a TRISB bit (= 0) will make the corresponding PORTB pin an output (i.e., put the contents of the output latch on the selected pin).

### EXAMPLE 3-2: INITIALIZING PORTB

```

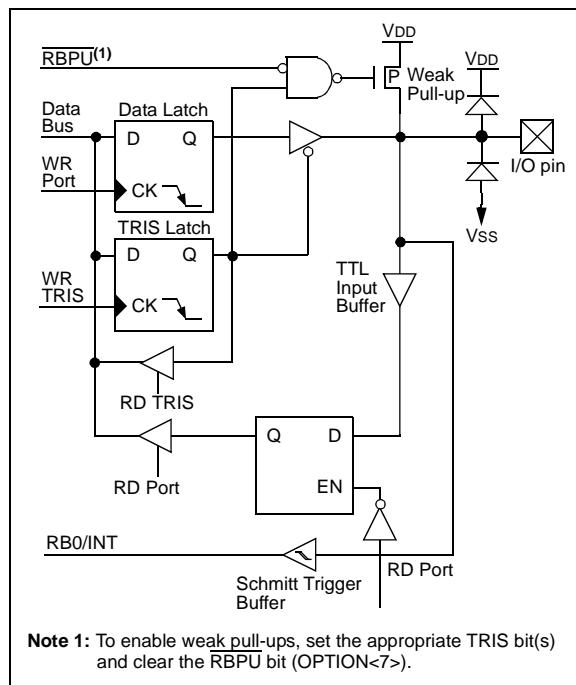
BANKSEL  PORTB    ; Select bank for PORTB
CLRF    PORTB    ; Initialize PORTB by
                ; clearing output
                ; data latches

BANKSEL  TRISB    ; Select Bank for TRISB
MOVLW   0xCF     ; Value used to
                ; initialize data
                ; direction

MOVWF   TRISB    ; Set RB<3:0> as inputs
                ; RB<5:4> as outputs
                ; RB<7:6> as inputs
    
```

Each of the PORTB pins has a weak internal pull-up. A single control bit can turn on all the pull-ups. This is performed by clearing bit  $\overline{\text{RBP}}\text{U}$  (OPTION<7>). The weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are disabled on a Power-on Reset.

**FIGURE 3-3: BLOCK DIAGRAM OF RB3:RB0 PINS**



Four of PORTB's pins, RB7:RB4, have an interrupt-on-change feature. Only pins configured as inputs can cause this interrupt to occur (i.e., any RB7:RB4 pin configured as an output is excluded from the interrupt on change comparison). The input pins (of RB7:RB4)

are compared with the old value latched on the last read of PORTB. The "mismatch" outputs of RB7:RB4 are OR'd together to generate the RB Port Change Interrupt with flag bit RBIF (INTCON<0>).

This interrupt can wake the device from SLEEP. The user, in the Interrupt Service Routine, can clear the interrupt in the following manner:

- Any read or write of PORTB. This will end the mismatch condition.
- Clear flag bit RBIF.

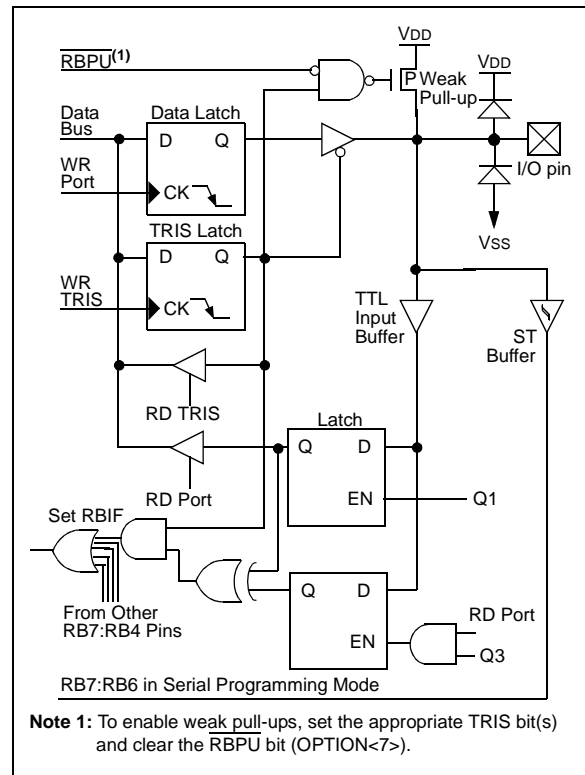
A mismatch condition will continue to set flag bit RBIF. Reading PORTB will end the mismatch condition and allow flag bit RBIF to be cleared.

The interrupt-on-change feature is recommended for wake-up on key depression operation and operations where PORTB is only used for the interrupt-on-change feature. Polling of PORTB is not recommended while using the interrupt-on-change feature.

This interrupt-on-mismatch feature, together with software configurable pull-ups on these four pins, allow easy interface to a keypad and make it possible for wake-up on key depression. Refer to the Embedded Control Handbook, "Implementing Wake-Up on Key Stroke" (AN552).

RB0/INT is an external interrupt input pin and is configured using the INTEDG bit (OPTION<6>).

**FIGURE 3-4: BLOCK DIAGRAM OF RB7:RB4 PINS**





# PIC16F72

**TABLE 3-3: PORTB FUNCTIONS**

Name	Bit#	Buffer	Function
RB0/INT	bit 0	TTL/ST <sup>(1)</sup>	Input/output pin or external interrupt input. Internal software programmable weak pull-up.
RB1	bit 1	TTL	Input/output pin. Internal software programmable weak pull-up.
RB2	bit 2	TTL	Input/output pin. Internal software programmable weak pull-up.
RB3	bit 3	TTL	Input/output pin. Internal software programmable weak pull-up.
RB4	bit 4	TTL	Input/output pin (with interrupt-on-change). Internal software programmable weak pull-up.
RB5	bit 5	TTL	Input/output pin (with interrupt-on-change). Internal software programmable weak pull-up.
RB6	bit 6	TTL/ST <sup>(2)</sup>	Input/output pin (with interrupt-on-change). Internal software programmable weak pull-up. Serial programming clock.
RB7	bit 7	TTL/ST <sup>(2)</sup>	Input/output pin (with interrupt-on-change). Internal software programmable weak pull-up. Serial programming data.

Legend: TTL = TTL input, ST = Schmitt Trigger input

**Note 1:** This buffer is a Schmitt Trigger input when configured as the external interrupt.

**2:** This buffer is a Schmitt Trigger input when used in Serial Programming mode.

**TABLE 3-4: SUMMARY OF REGISTERS ASSOCIATED WITH PORTB**

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
06h, 106h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx	uuuu uuuu
86h, 186h	TRISB	PORTB Data Direction Register								1111 1111	1111 1111
81h, 181h	OPTION	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111

Legend: x = unknown, u = unchanged. Shaded cells are not used by PORTB.

## 3.3 PORTC and the TRISC Register

PORTC is an 8-bit wide, bi-directional port. The corresponding data direction register is TRISC. Setting a TRISC bit (= 1) will make the corresponding PORTC pin an input (i.e., put the corresponding output driver in a Hi-Impedance mode). Clearing a TRISC bit (= 0) will make the corresponding PORTC pin an output (i.e., put the contents of the output latch on the selected pin).

PORTC is multiplexed with several peripheral functions (Table 3-5). PORTC pins have Schmitt Trigger input buffers. [www.trimmer.ru](http://www.trimmer.ru)

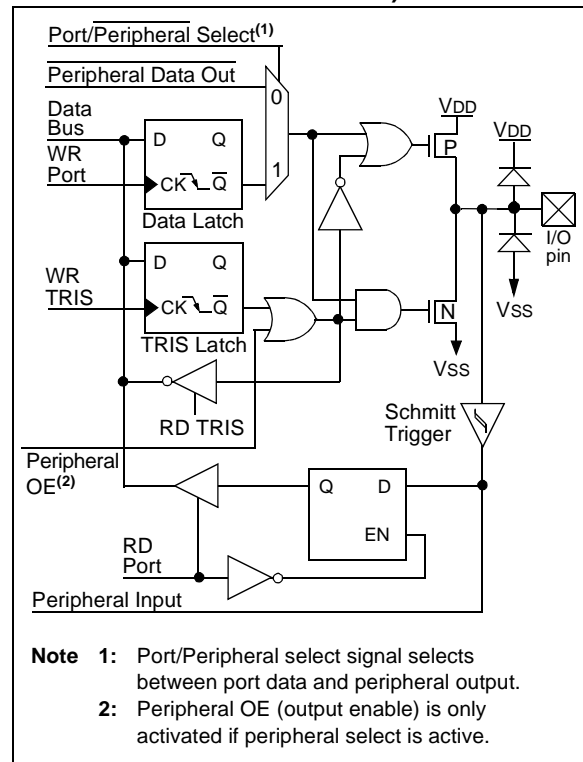
When enabling peripheral functions, care should be taken in defining TRIS bits for each PORTC pin. Some peripherals override the TRIS bit to make a pin an output, while other peripherals override the TRIS bit to make a pin an input. Since the TRIS bit override is in effect while the peripheral is enabled, read-modify-write instructions (BSF, BCF, XORWF) with TRISC as destination should be avoided. The user should refer to the corresponding peripheral section for the correct TRIS bit settings.

### EXAMPLE 3-3: INITIALIZING PORTC

```

BANKSEL  PORTC    ; Select Bank for PORTC
CLRF     PORTC    ; Initialize PORTC by
                ; clearing output
                ; data latches
BANKSEL  TRISC    ; Select Bank for TRISC
MOVLW   0xCF     ; Value used to
                ; initialize data
                ; direction
MOVWF   TRISC    ; Set RC<3:0> as inputs
                ; RC<5:4> as outputs
                ; RC<7:6> as inputs
    
```

**FIGURE 3-5: PORTC BLOCK DIAGRAM (PERIPHERAL OUTPUT OVERRIDE)**



# PIC16F72

**TABLE 3-5: PORTC FUNCTIONS**

Name	Bit#	Buffer Type	Function
RC0/T1OSO/T1CKI	bit 0	ST	Input/output port pin or Timer1 oscillator output/Timer1 clock input.
RC1/T1OSI	bit 1	ST	Input/output port pin or Timer1 oscillator input.
RC2/CCP1	bit 2	ST	Input/output port pin or Capture1 input/Compare1 output/PWM1 output.
RC3/SCK/SCL	bit 3	ST	RC3 can also be the synchronous serial clock for both SPI and I <sup>2</sup> C modes.
RC4/SDI/SDA	bit 4	ST	RC4 can also be the SPI Data In (SPI mode) or data I/O (I <sup>2</sup> C mode).
RC5/SDO	bit 5	ST	Input/output port pin or Synchronous Serial Port data output.
RC6	bit 6	ST	Input/output port pin.
RC7	bit 7	ST	Input/output port pin.

Legend: ST = Schmitt Trigger input

**TABLE 3-6: SUMMARY OF REGISTERS ASSOCIATED WITH PORTC**

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
07h	PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	xxxx xxxx	uuuu uuuu
87h	TRISC	PORTC Data Direction Register								1111 1111	1111 1111

Legend: x = unknown, u = unchanged

## 4.0 READING PROGRAM MEMORY

The FLASH Program Memory is readable during normal operation over the entire VDD range. It is indirectly addressed through Special Function Registers (SFR). Up to 14-bit wide numbers can be stored in memory for use as calibration parameters, serial numbers, packed 7-bit ASCII, etc. Executing a program memory location containing data that forms an invalid instruction results in a NOP.

There are five SFRs used to read the program and memory:

- PMCON1
- PMDATL
- PMDATH
- PMADRL
- PMADRH

The program memory allows word reads. Program memory access allows for checksum calculation and reading calibration tables.

When interfacing to the program memory block, the PMDATH:PMDATL registers form a two-byte word, which holds the 14-bit data for reads. The PMADRH:PMADRL registers form a two-byte word, which holds the 13-bit address of the FLASH location being accessed. This device has up to 2K words of program FLASH, with an address range from 0h to 07FFh. The unused upper bits PMDATH<7:6> and PMADRH<7:5> are not implemented and read as zeros.

## 4.1 PMADR

The address registers can address up to a maximum of 8K words of program FLASH.

When selecting a program address value, the MSByte of the address is written to the PMADRH register and the LSByte is written to the PMADRL register. The upper MSbits of PMADRH must always be clear.

## 4.2 PMCON1 Register

PMCON1 is the control register for memory accesses.

The control bit RD initiates read operations. This bit cannot be cleared, only set, in software. It is cleared in hardware at the completion of the read operation.

### REGISTER 4-1: PMCON1: PROGRAM MEMORY CONTROL REGISTER 1 (ADDRESS 18Ch)

R-1	U-0	U-0	U-0	U-0	U-0	U-0	R/S-0
reserved	—	—	—	—	—	—	RD
bit 7							bit 0

bit 7 **Reserved:** Read as '1'

bit 6-1 **Unimplemented:** Read as '0'

bit 0 **RD:** Read Control bit

1 = Initiates a FLASH read, RD is cleared in hardware. The RD bit can only be set (not cleared) in software.

0 = Does not initiate a FLASH read

**Legend:**

W = Writable bit

U = Unimplemented bit, read as '0'

R = Readable bit

S = Settable bit

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

# PIC16F72

## 4.3 Reading the FLASH Program Memory

To read a program memory location, the user must write two bytes of the address to the PMADRL and PMADRH registers and then set control bit, RD (PMCON1<0>). Once the read control bit is set, the program memory FLASH controller will use the second instruction cycle after to read the data. This causes the second instruction immediately following the “BSF PMCON1, RD” instruction to be ignored. The data is available in the very next cycle in the PMDATL and PMDATH registers; therefore, it can be read as two bytes in the following instructions. PMDATL and PMDATH registers will hold this value until another read, or until it is written to by the user (during a write operation).

## 4.4 Operation During Code Protect

The FLASH program memory control can read anywhere within the program memory, whether or not the program memory is code protected.

This does not compromise the code, because there is no way to rewrite a portion of the program memory, or leave contents of a program memory read in a register while changing modes.

### EXAMPLE 4-1: FLASH PROGRAM READ

```

BANKSEL PMADRH      ; Select Bank for PMADRH
MOVLW  MS_PROG_EE_ADDR ;
MOVWF  PMADRH      ; MS Byte of Program Address to read
MOVLW  LS_PROG_EE_ADDR ;
MOVWF  PMADRL      ; LS Byte of Program Address to read
BANKSEL PMCON1     ; Select Bank for PMCON1
BSF    PMCON1, RD  ; EE Read
;
NOP     ; Any instructions here are ignored as program
NOP     ; memory is read in second cycle after BSF PMCON1,RD
;
; First instruction after BSF PMCON1,RD executes normally
BANKSEL PMDATL     ; Select Bank for PMDATL
MOVF   PMDATL, W   ; W = LS Byte of Program PMDATL
MOVF   PMDATH, W   ; W = MS Byte of Program PMDATH
    
```

TABLE 4-1: REGISTERS ASSOCIATED WITH PROGRAM FLASH

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	
10Dh	PMADRL	Address Register Low Byte								xxxx xxxx	uuuu uuuu	
10Fh	PMADRH	—	—	—	Address Register High Byte						xxxx xxxx	uuuu uuuu
10Ch	PMDATL	Data Register Low Byte								xxxx xxxx	uuuu uuuu	
10Eh	PMDATH	—	—	Data Register High Byte						xxxx xxxx	uuuu uuuu	
18Ch	PMCON1	—(1)	—	—	—	—	—	—	RD	1--- ---0	1--- ---0	

Legend: x = unknown, u = unchanged, r = reserved, - = unimplemented, read as '0'.  
Shaded cells are not used during FLASH access.

**Note 1:** This bit always reads as a '1'.

## 5.0 TIMER0 MODULE

The Timer0 module timer/counter has the following features:

- 8-bit timer/counter
- Readable and writable
- 8-bit software programmable prescaler
- Internal or external clock select
- Interrupt on overflow from FFh to 00h
- Edge select for external clock

Figure 5-1 is a block diagram of the Timer0 module and the prescaler shared with the WDT.

Additional information on the Timer0 module is available in the PICmicro™ Mid-Range MCU Family Reference Manual (DS33023).

### 5.1 Timer0 Operation

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

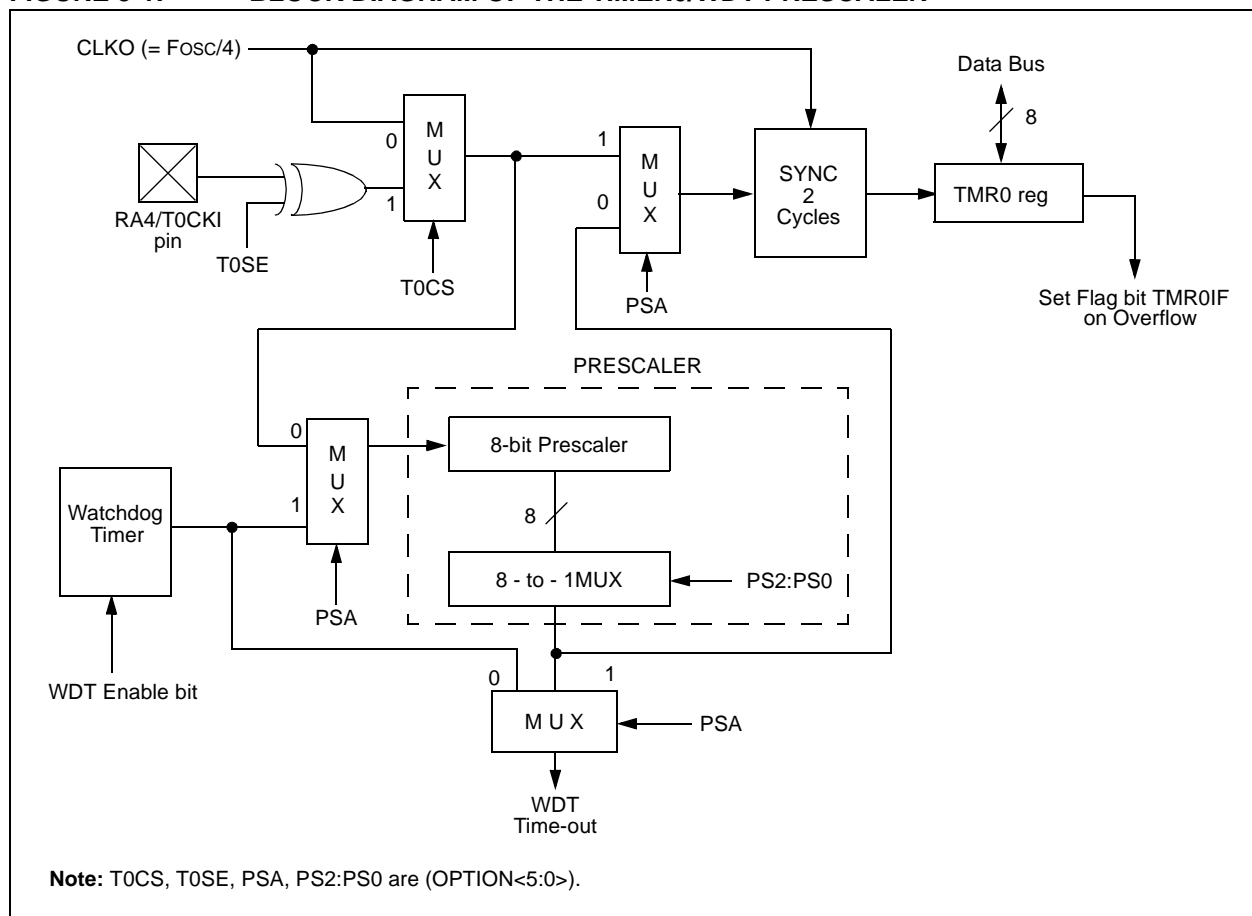
Counter mode is selected by setting bit TOCS (OPTION<5>). In Counter mode, Timer0 will increment, either on every rising or falling edge of pin RA4/T0CKI. The incrementing edge is determined by the Timer0 Source Edge Select bit TOSE (OPTION<4>). Clearing bit TOSE selects the rising edge. Restrictions on the external clock input are discussed in detail in Section 5.3.

The prescaler is mutually exclusively shared between the Timer0 module and the Watchdog Timer. The prescaler is not readable or writable. Section 5.4 details the operation of the prescaler.

### 5.2 Timer0 Interrupt

The TMR0 interrupt is generated when the TMR0 register overflows from FFh to 00h. This overflow sets bit TMR0IF (INTCON<2>). The interrupt can be masked by clearing bit TMR0IE (INTCON<5>). Bit TMR0IF must be cleared in software by the Timer0 module Interrupt Service Routine, before re-enabling this interrupt. The TMR0 interrupt cannot awaken the processor from SLEEP, since the timer is shut-off during SLEEP.

**FIGURE 5-1: BLOCK DIAGRAM OF THE TIMER0/WDT PRESCALER**



# PIC16F72

## 5.3 Using Timer0 with an External Clock

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. Therefore, it is necessary for T0CKI to be high for at least 2 TOSC (and a small RC delay of 20 ns) and low for at least 2 TOSC (and a small RC delay of 20 ns). Refer to the electrical specification of the desired device.

## 5.4 Prescaler

There is only one prescaler available, which is mutually exclusively shared between the Timer0 module and the Watchdog Timer. A prescaler assignment for the

Timer0 module means that there is no prescaler for the Watchdog Timer, and vice-versa. This prescaler is not readable or writable (see Figure 5-1).

The PSA and PS2:PS0 bits (OPTION<3:0>) determine the prescaler assignment and prescale ratio.

When assigned to the Timer0 module, all instructions writing to the TMR0 register (e.g., CLRF 1, MOVWF 1, BSF 1,x...etc.) will clear the prescaler. When assigned to WDT, a CLRWDT instruction will clear the prescaler along with the Watchdog Timer. The prescaler is not readable or writable.

**Note:** Writing to TMR0 when the prescaler is assigned to Timer0, will clear the prescaler count but will not change the prescaler assignment.

**TABLE 5-1: REGISTERS ASSOCIATED WITH TIMER0**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other RESETS
01h,101h	TMR0	Timer0 Module Register								xxxx xxxx	uuuu uuuu
0Bh,8Bh,10Bh,18Bh	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF	0000 000x	0000 000u
81h,181h	OPTION	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111

Legend: x = unknown, u = unchanged, - = unimplemented locations read as '0'. Shaded cells are not used by Timer0.

## 6.0 TIMER1 MODULE

The Timer1 module timer/counter has the following features:

- 16-bit timer/counter  
(Two 8-bit registers; TMR1H and TMR1L)
- Readable and writable (both registers)
- Internal or external clock select
- Interrupt on overflow from FFFFh to 0000h
- RESET from CCP module trigger

Timer1 has a control register, shown in Register 6-1. Timer1 can be enabled/disabled by setting/clearing control bit TMR1ON (T1CON<0>).

Figure 6-2 is a simplified block diagram of the Timer1 module.

Additional information on timer modules is available in the PICmicro™ Mid-Range MCU Reference Manual, (DS33023).

## 6.1 Timer1 Operation

Timer1 can operate in one of these modes:

- As a timer
- As a synchronous counter
- As an asynchronous counter

The Operating mode is determined by the clock select bit, TMR1CS (T1CON<1>).

In Timer mode, Timer1 increments every instruction cycle. In Counter mode, it increments on every rising edge of the external clock input.

When the Timer1 oscillator is enabled (T1OSCEN is set), the RC1/T1OSI and RC0/T1OSO/T1CKI pins become inputs. That is, the TRISC<1:0> value is ignored.

Timer1 also has an internal "RESET input". This RESET can be generated by the CCP module (Section 8.0).

### REGISTER 6-1: T1CON: TIMER1 CONTROL REGISTER (ADDRESS 10h)

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
—	—	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{T1SYNC}$	TMR1CS	TMR1ON	
bit 7								bit 0

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

bit 5-4 **T1CKPS1:T1CKPS0:** Timer1 Input Clock Prescale Select bits

11 = 1:8 Prescale value  
 10 = 1:4 Prescale value  
 01 = 1:2 Prescale value  
 00 = 1:1 Prescale value

bit 3 **T1OSCEN:** Timer1 Oscillator Enable Control bit

1 = Oscillator is enabled  
 0 = Oscillator is shut-off (The oscillator inverter is turned off to eliminate power drain.)

bit 2 **T1SYNC:** Timer1 External Clock Input Synchronization Control bit

TMR1CS = 1:

1 = Do not synchronize external clock input  
 0 = Synchronize external clock input

TMR1CS = 0:

This bit is ignored. Timer1 uses the internal clock when TMR1CS = '0'.

bit 1 **TMR1CS:** Timer1 Clock Source Select bit

1 = External clock from pin RC0/T1OSO/T1CKI (on the rising edge)  
 0 = Internal clock (FOSC/4)

bit 0 **TMR1ON:** Timer1 On bit

1 = Enables Timer1  
 0 = Stops Timer1

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



# PIC16F72

## 6.2 Timer1 Operation in Timer Mode

Timer mode is selected by clearing the TMR1CS (T1CON<1>) bit. In this mode, the input clock to the timer is  $F_{OSC}/4$ . The synchronize control bit  $\overline{T1SYNC}$  (T1CON<2>) has no effect, since the internal clock is always in sync.

## 6.3 Timer1 Counter Operation

Timer1 may operate in Asynchronous or Synchronous mode, depending on the setting of the TMR1CS bit.

When Timer1 is being incremented via an external source, increments occur on a rising edge. After Timer1 is enabled in Counter mode, the module must first have a falling edge before the counter begins to increment.

## 6.4 Timer1 Operation in Synchronized Counter Mode

Counter mode is selected by setting bit TMR1CS. In this mode, the timer increments on every rising edge of clock input on pin RC1/T1OSI when bit T1OSCEN is set, or on pin RC0/T1OSO/T1CKI when bit T1OSCEN is cleared.

If  $\overline{T1SYNC}$  is cleared, then the external clock input is synchronized with internal phase clocks. The synchronization is done after the prescaler stage. The prescaler stage is an asynchronous ripple counter.

In this configuration, during SLEEP mode, Timer1 will not increment even if the external clock is present, since the synchronization circuit is shut-off. The prescaler, however, will continue to increment.

FIGURE 6-1: TIMER1 INCREMENTING EDGE

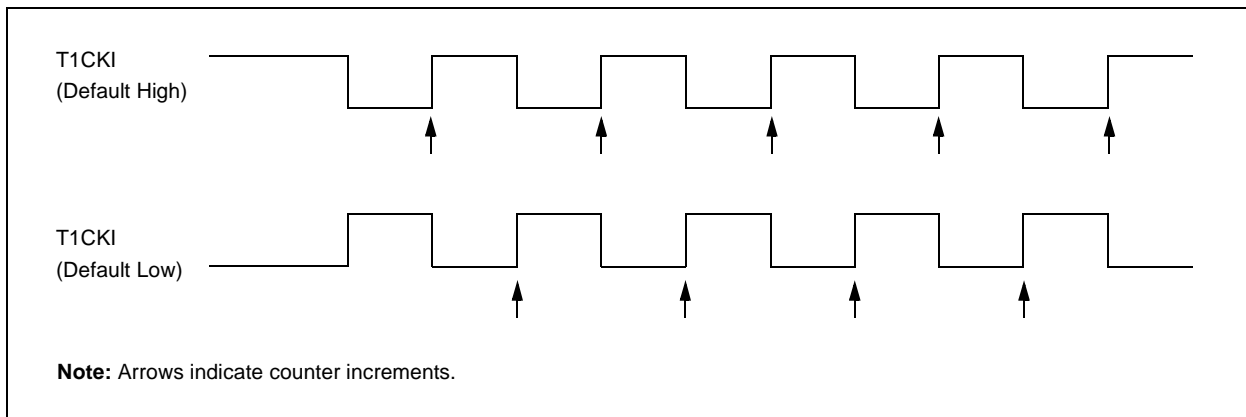
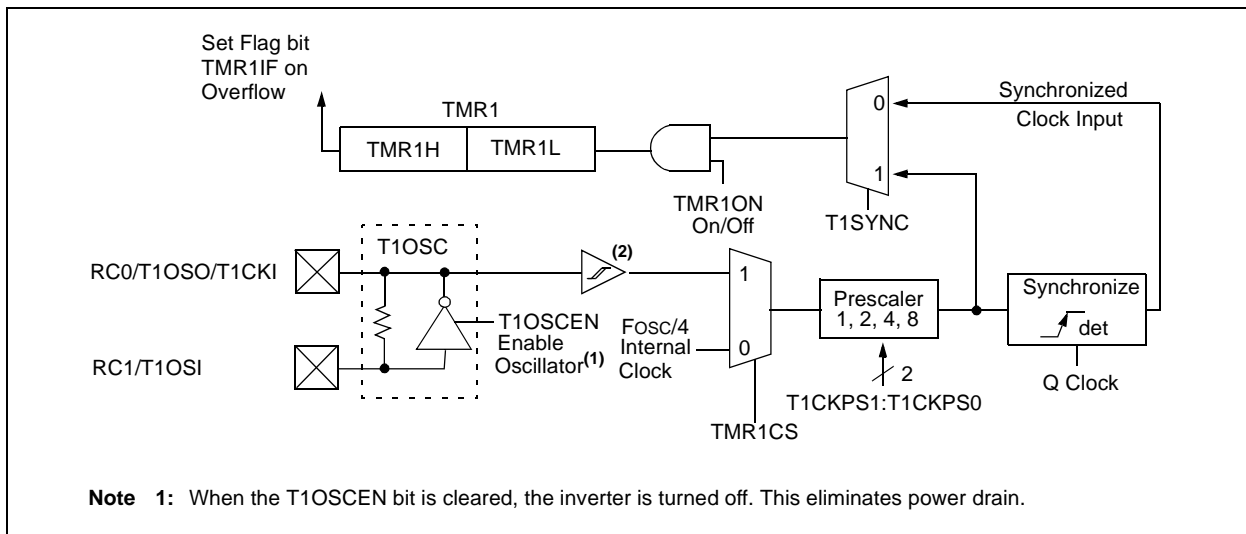


FIGURE 6-2: TIMER1 BLOCK DIAGRAM



## 6.5 Timer1 Operation in Asynchronous Counter Mode

If control bit  $\overline{T1SYNC}$  (T1CON<2>) is set, the external clock input is not synchronized. The timer continues to increment asynchronous to the internal phase clocks. The timer will continue to run during SLEEP and can generate an interrupt on overflow, that will wake-up the processor. However, special precautions in software are needed to read/write the timer (Section 6.5.1).

In Asynchronous Counter mode, Timer1 cannot be used as a time base for capture or compare operations.

### 6.5.1 READING AND WRITING TIMER1 IN ASYNCHRONOUS COUNTER MODE

Reading TMR1H or TMR1L while the timer is running from an external asynchronous clock will ensure a valid read (taken care of in hardware). However, the user should keep in mind that reading the 16-bit timer in two 8-bit values itself, poses certain problems, since the timer may overflow between the reads.

For writes, it is recommended that the user simply stop the timer and write the desired values. A write contention may occur by writing to the timer registers, while the register is incrementing. This may produce an unpredictable value in the timer register. Data in the Timer1 register (TMR1) may become corrupted. Corruption occurs when the timer enable is turned off at the same instant that a ripple carry occurs in the timer module.

Reading the 16-bit value requires some care. Examples 12-2 and 12-3 in the PICmicro™ Mid-Range MCU Family Reference Manual (DS33023) show how to read and write Timer1 when it is running in Asynchronous mode.

## 6.6 Timer1 Oscillator

A crystal oscillator circuit is built between pins T1OSI (input) and T1OSO (amplifier output). It is enabled by setting control bit T1OSCEN (T1CON<3>). The oscillator is a low power oscillator rated up to 200 kHz. It will continue to run during SLEEP. It is primarily intended for a 32 kHz crystal. Table 6-1 shows the capacitor selection for the Timer1 oscillator.

The Timer1 oscillator is identical to the LP oscillator. The user must provide a software time delay to ensure proper oscillator start-up.

**TABLE 6-1: CAPACITOR SELECTION FOR THE TIMER1 OSCILLATOR**

Osc Type	Freq	C1	C2
LP	32 kHz	33 pF	33 pF
	100 kHz	15 pF	15 pF
	200 kHz	15 pF	15 pF
<b>These values are for design guidance only.</b>			
<p><b>Note 1:</b> Higher capacitance increases the stability of oscillator, but also increases the start-up time.</p> <p><b>2:</b> Since each resonator/crystal has its own characteristics, the user should consult the resonator/crystal manufacturer for appropriate values of external components.</p>			

## 6.7 Timer1 Interrupt

The TMR1 register pair (TMR1H:TMR1L) increments from 0000h to FFFFh and rolls over to 0000h. The TMR1 interrupt, if enabled, is generated on overflow, which is latched in interrupt flag bit TMR1IF (PIR1<0>). This interrupt can be enabled/disabled by setting/clearing TMR1 interrupt enable bit TMR1IE (PIE1<0>).

## 6.8 Resetting Timer1 Using a CCP Trigger Output

If the CCP module is configured in Compare mode to generate a "special event trigger" signal (CCP1M3:CCP1M0 = 1011), the signal will reset Timer1 and start an A/D conversion (if the A/D module is enabled).

**Note:** The special event triggers from the CCP1 module will not set interrupt flag bit TMR1IF (PIR1<0>).

Timer1 must be configured for either Timer or Synchronized Counter mode to take advantage of this feature. If Timer1 is running in Asynchronous Counter mode, this RESET operation may not work.

In the event that a write to Timer1 coincides with a special event trigger from CCP1, the write will take precedence.

In this mode of operation, the CCPR1H:CCPR1L registers pair effectively becomes the period register for Timer1.

# PIC16F72

## 6.9 Resetting Timer1 Register Pair (TMR1H, TMR1L)

TMR1H and TMR1L registers are not reset to 00h on a POR, or any other RESET, except by the CCP1 special event triggers.

T1CON register is reset to 00h on a Power-on Reset or a Brown-out Reset, which shuts off the timer and leaves a 1:1 prescale. In all other RESETS, the register is unaffected.

## 6.10 Timer1 Prescaler

The prescaler counter is cleared on writes to the TMR1H or TMR1L registers.

**TABLE 6-2: REGISTERS ASSOCIATED WITH TIMER1 AS A TIMER/COUNTER**

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
0Bh,8Bh,10Bh,18Bh	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	—	ADIF	—	—	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	—	ADIE	—	—	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
0Eh	TMR1L	Holding register for the Least Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	uuuu uuuu
0Fh	TMR1H	Holding register for the Most Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	uuuu uuuu
10h	T1CON	—	—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	--00 0000	--uu uuuu

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used by the Timer1 module.

## 7.0 TIMER2 MODULE

The Timer2 module timer has the following features:

- 8-bit timer (TMR2 register)
- 8-bit period register (PR2)
- Readable and writable (both registers)
- Software programmable prescaler (1:1, 1:4, 1:16)
- Software programmable postscaler (1:1 to 1:16)
- Interrupt on TMR2 match of PR2
- SSP module optional use of TMR2 output to generate clock shift

Timer2 has a control register, shown in Register 7-1. Timer2 can be shut-off by clearing control bit TMR2ON (T2CON<2>) to minimize power consumption.

Figure 7-1 is a simplified block diagram of the Timer2 module.

Additional information on timer modules is available in the PICmicro™ Mid-Range MCU Reference Manual, (DS33023).

### 7.1 Timer2 Operation

Timer2 can be used as the PWM time-base for PWM mode of the CCP module.

The TMR2 register is readable and writable, and is cleared on any device RESET.

The input clock ( $F_{osc}/4$ ) has a prescale option of 1:1, 1:4 or 1:16, selected by control bits T2CKPS1:T2CKPS0 (T2CON<1:0>).

The match output of TMR2 goes through a 4-bit postscaler (which gives a 1:1 to 1:16 scaling inclusive) to generate a TMR2 interrupt (latched in flag bit TMR2IF, (PIR1<1>)).

### 7.2 Timer2 Prescaler and Postscaler

The prescaler and postscaler counters are cleared when any of the following occurs:

- A write to the TMR2 register
- A write to the T2CON register
- Any device RESET (Power-on Reset,  $\overline{MCLR}$ , WDT Reset, or Brown-out Reset)

TMR2 is not cleared when T2CON is written.

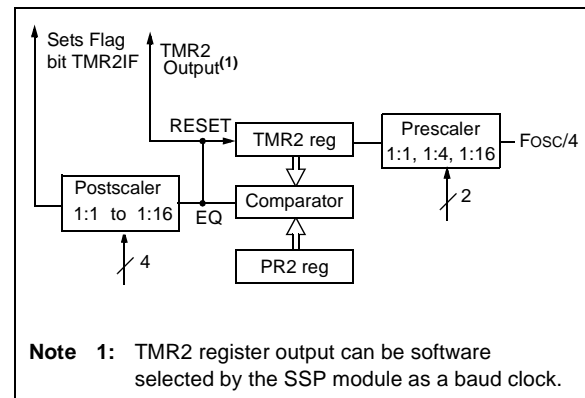
### 7.3 Timer2 Interrupt

The Timer2 module has an 8-bit period register, PR2. Timer2 increments from 00h until it matches PR2 and then resets to 00h on the next increment cycle. PR2 is a readable and writable register. The PR2 register is initialized to FFh upon RESET.

### 7.4 Output of TMR2

The output of TMR2 (before the postscaler) is fed to the Synchronous Serial Port module, which optionally uses it to generate a shift clock.

**FIGURE 7-1: TIMER2 BLOCK DIAGRAM**



# PIC16F72

## REGISTER 7-1: T2CON: TIMER2 CONTROL REGISTER (ADDRESS 12h)

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0
bit 7							bit 0

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

bit 6-3 **TOUTPS3:TOUTPS0:** Timer2 Output Postscale Select bits

0000 = 1:1 Postscale

0001 = 1:2 Postscale

0010 = 1:3 Postscale

•

•

•

1111 = 1:16 Postscale

bit 2 **TMR2ON:** Timer2 On bit

1 = Timer2 is on

0 = Timer2 is off

bit 1-0 **T2CKPS1:T2CKPS0:** Timer2 Clock Prescale Select bits

00 = Prescaler is 1

01 = Prescaler is 4

1x = Prescaler is 16

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

**TABLE 7-1: REGISTERS ASSOCIATED WITH TIMER2 AS A TIMER/COUNTER**

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
0Bh,8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	—	ADIF	—	—	SSPIF	CCP1IF	TMR2IF	TMR1IF	-0-- 0000	0000 0000
8Ch	PIE1	—	ADIE	—	—	SSPIE	CCP1IE	TMR2IE	TMR1IE	-0-- 0000	0000 0000
11h	TMR2	Timer2 Module Register								0000 0000	0000 0000
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000
92h	PR2	Timer2 Period Register								1111 1111	1111 1111

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used by the Timer2 module.

## 8.0 CAPTURE/COMPARE/PWM (CCP) MODULE

Additional information on the CCP module is available in the PICmicro™ Mid-Range MCU Reference Manual, (DS33023).

The CCP (Capture/Compare/PWM) module contains a 16-bit register that can operate as a:

- 16-bit capture register
- 16-bit compare register
- PWM master/slave duty cycle register.

Table 8-1 shows the timer resources of the CCP Module modes.

Capture/Compare/PWM Register1 (CCPR1) is comprised of two 8-bit registers: CCPR1L (low byte) and CCPR1H (high byte). The CCP1CON register controls the operation of CCP1. All are readable and writable.

**TABLE 8-1: CCP MODE - TIMER RESOURCE**

CCP Mode	Timer Resource
Capture	Timer1
Compare	Timer1
PWM	Timer2

### REGISTER 8-1: CCPCON1: CAPTURE/COMPARE/PWM CONTROL REGISTER 1 (ADDRESS 17h)

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
—	—	CCPxX	CCPxY	CCPxM3	CCPxM2	CCPxM1	CCPxM0	
bit 7								bit 0

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

bit 5-4 **CCPxX:CCPxY:** PWM Least Significant bits

Capture mode:

Unused

Compare mode:

Unused

PWM mode:

These bits are the two LSbs of the PWM duty cycle. The eight MSbs are found in CCPRxL.

bit 3-0 **CCPxM3:CCPxM0:** CCPx Mode Select bits

0000 = Capture/Compare/PWM disabled (resets CCPx module)

0100 = Capture mode, every falling edge

0101 = Capture mode, every rising edge

0110 = Capture mode, every 4th rising edge

0111 = Capture mode, every 16th rising edge

1000 = Compare mode, set output on match (CCPxIF bit is set)

1001 = Compare mode, clear output on match (CCPxIF bit is set)

1010 = Compare mode, generate software interrupt on match (CCPxIF bit is set, CCPx pin is unaffected)

1011 = Compare mode, trigger special event (CCPxIF bit is set, CCPx pin is unaffected); CCP1 resets TMR1 and starts an A/D conversion (if A/D module is enabled)

11xx = PWM mode

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

# PIC16F72

## 8.1 Capture Mode

In Capture mode, CCPR1H:CCPR1L captures the 16-bit value of the TMR1 register when an event occurs on pin RC2/CCP1. An event is defined as:

- Every falling edge
- Every rising edge
- Every 4th rising edge
- Every 16th rising edge

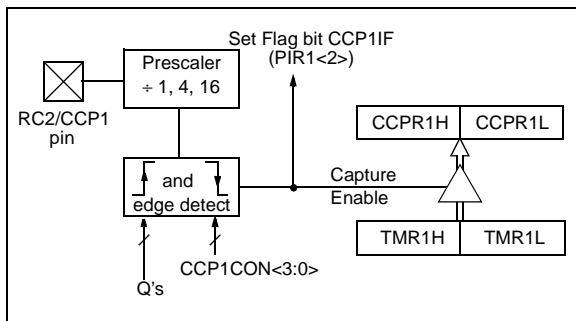
An event is selected by control bits CCP1M3:CCP1M0 (CCP1CON<3:0>). When a capture is made, the interrupt request flag bit CCP1IF (PIR1<2>) is set. It must be cleared in software. If another capture occurs before the value in register CCPR1 is read, the old captured value is overwritten by the new captured value.

### 8.1.1 CCP PIN CONFIGURATION

In Capture mode, the RC2/CCP1 pin should be configured as an input by setting the TRISC<2> bit.

**Note:** If the RC2/CCP1 is configured as an output, a write to the port can cause a capture condition.

**FIGURE 8-1: CAPTURE MODE OPERATION BLOCK DIAGRAM**



### 8.1.2 TIMER1 MODE SELECTION

Timer1 must be running in Timer mode or Synchronized Counter mode for the CCP module to use the capture feature. In Asynchronous Counter mode, the capture operation may not work.

### 8.1.3 SOFTWARE INTERRUPT

When the Capture mode is changed, a false capture interrupt may be generated. The user should keep bit CCP1IE (PIE1<2>) clear to avoid false interrupts and should clear the flag bit CCP1IF, following any such change in Operating mode.

### 8.1.4 CCP PRESCALER

There are four prescaler settings, specified by bits CCP1M3:CCP1M0. Whenever the CCP module is turned off, or the CCP module is not in Capture mode, the prescaler counter is cleared. This means that any RESET will clear the prescaler counter.

Switching from one capture prescaler to another may generate an interrupt. Also, the prescaler counter will not be cleared, therefore, the first capture may be from a non-zero prescaler. Example 8-1 shows the recommended method for switching between capture prescalers. This example also clears the prescaler counter and will not generate the "false" interrupt.

**EXAMPLE 8-1: CHANGING BETWEEN CAPTURE PRESCALERS**

```
CLRF   CCP1CON   ; Turn CCP module off
MOVLW  NEW_CAPT_PS ; Load the W reg with
                ; the new prescaler
                ; mode value and CCP ON
MOVWF  CCP1CON   ; Load CCP1CON with
                ; this value
```

## 8.2 Compare Mode

In Compare mode, the 16-bit CCPR1 register value is constantly compared against the TMR1 register pair value. When a match occurs, the RC2/CCP1 pin is:

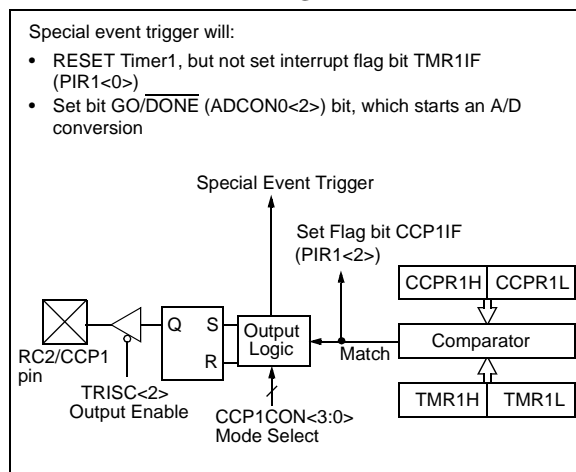
- Driven High
- Driven Low
- Remains Unchanged

The action on the pin is based on the value of control bits CCP1M3:CCP1M0 (CCP1CON<3:0>). At the same time, interrupt flag bit CCP1IF is set.

The output may become inverted when the mode of the module is changed from Compare/Clear on Match (CCPxM<3:0> = '1001') to Compare/Set on Match (CCPxM<3:0> = '1000'). This may occur as a result of any operation that selectively clears bit CCPxM0, such as a BCF instruction.

When this condition occurs, the output becomes inverted when the instruction is executed. It will remain inverted for all following Compare operations, until the module is reset.

**FIGURE 8-2: COMPARE MODE OPERATION BLOCK DIAGRAM**



### 8.2.1 CCP PIN CONFIGURATION

The user must configure the RC2/CCP1 pin as an output by clearing the TRISC<2> bit.

**Note:** Clearing the CCP1CON register will force the RC2/CCP1 compare output latch to the default low level. This is not the data latch.

### 8.2.2 TIMER1 MODE SELECTION

Timer1 must be running in Timer mode or Synchronized Counter mode, if the CCP module is using the compare feature. In Asynchronous Counter mode, the compare operation may not work.

### 8.2.3 SOFTWARE INTERRUPT MODE

When generate software interrupt is chosen, the CCP1 pin is not affected. Only a CCP interrupt is generated (if enabled).

### 8.2.4 SPECIAL EVENT TRIGGER

In this mode, an internal hardware trigger is generated that may be used to initiate an action.

The special event trigger output of CCP1 resets the TMR1 register pair. This allows the CCPR1 register to effectively be a 16-bit programmable period register for Timer1.

The special trigger output of CCP1 resets the TMR1 register pair, and starts an A/D conversion (if the A/D module is enabled).

**Note:** The special event trigger from the CCP1 module will not set interrupt flag bit TMR1IF (PIR1<0>).



# PIC16F72

**TABLE 8-2: REGISTERS ASSOCIATED WITH CAPTURE, COMPARE, AND TIMER1**

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
0Bh,8Bh 10Bh,18Bh	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	—	ADIF	—	—	SSPIF	CCP1IF	TMR2IF	TMR1IF	-0-- 0000	0000 0000
8Ch	PIE1	—	ADIE	—	—	SSPIE	CCP1IE	TMR2IE	TMR1IE	-0-- 0000	0000 0000
87h	TRISC	PORTC Data Direction Register								1111 1111	1111 1111
0Eh	TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	uuuu uuuu
0Fh	TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	uuuu uuuu
10h	T1CON	—	—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	--00 0000	--uu uuuu
15h	CCPR1L	Capture/Compare/PWM Register1 (LSB)								xxxx xxxx	uuuu uuuu
16h	CCPR1H	Capture/Compare/PWM Register1 (MSB)								xxxx xxxx	uuuu uuuu
17h	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	--00 0000

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used by Capture and Timer1.

## 8.3 PWM Mode

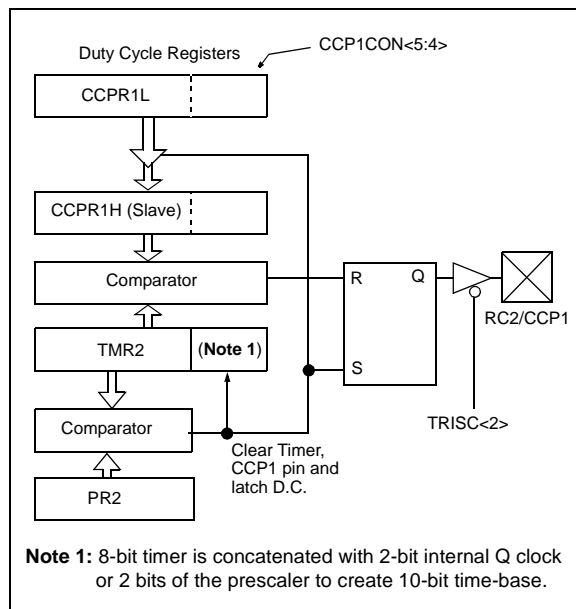
In Pulse Width Modulation (PWM) mode, the CCP1 pin produces up to a 10-bit resolution PWM output. Since the CCP1 pin is multiplexed with the PORTC data latch, the TRISC<2> bit must be cleared to make the CCP1 pin an output.

**Note:** Clearing the CCP1CON register will force the CCP1 PWM output latch to the default low level. This is not the PORTC I/O data latch.

Figure 8-3 shows a simplified block diagram of the CCP module in PWM mode.

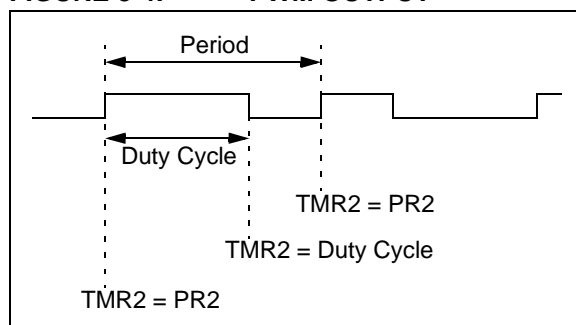
For a step by step procedure on how to set up the CCP module for PWM operation, see Section 8.3.3.

**FIGURE 8-3: SIMPLIFIED PWM BLOCK DIAGRAM**



A PWM output (Figure 8-4) has a time-base (period) and a time that the output stays high (duty cycle). The frequency of the PWM is the inverse of the period (1/period).

**FIGURE 8-4: PWM OUTPUT**



### 8.3.1 PWM PERIOD

The PWM period is specified by writing to the PR2 register. The PWM period can be calculated using the formula in Equation 8-1.

**EQUATION 8-1: PWM PERIOD**

$$\text{PWM period} = [(PR2) + 1] \cdot 4 \cdot T_{osc} \cdot (\text{TMR2 prescale value})$$

PWM frequency is defined as  $1 / [\text{PWM period}]$ .

When TMR2 is equal to PR2, the following three events occur on the next increment cycle:

- TMR2 is cleared
- The CCP1 pin is set (exception: if PWM duty cycle = 0%, the CCP1 pin will not be set)
- The PWM duty cycle is latched from CCPR1L into CCPR1H

**Note:** The Timer2 postscaler (see Section 7.0) is not used in the determination of the PWM frequency. The postscaler could be used to have a servo update rate at a different frequency than the PWM output.

### 8.3.2 PWM DUTY CYCLE

The PWM duty cycle is specified by writing to the CCPR1L register and to the CCP1CON<5:4> bits. Up to 10-bit resolution is available: the CCPR1L contains the eight MSBs and the CCP1CON<5:4> contains the two LSBs. This 10-bit value is represented by CCPR1L:CCP1CON<5:4>. Equation 8-2 is used to calculate the PWM duty cycle in time.

**EQUATION 8-2: PWM DUTY CYCLE**

$$\text{PWM duty cycle} = (\text{CCPR1L:CCP1CON<5:4>}) \cdot T_{osc} \cdot (\text{TMR2 prescale value})$$

CCPR1L and CCP1CON<5:4> can be written to at any time, but the duty cycle value is not latched into CCPR1H until after a match between PR2 and TMR2 occurs (i.e., the period is complete). In PWM mode, CCPR1H is a read only register.

The CCPR1H register and a 2-bit internal latch are used to double buffer the PWM duty cycle. This double buffering is essential for glitchless PWM operation.

When the CCPR1H and 2-bit latch match TMR2, concatenated with an internal 2-bit Q clock or 2 bits of the TMR2 prescaler, the CCP1 pin is cleared.

# PIC16F72

Maximum PWM resolution (bits) for a given PWM frequency is calculated using Equation 8-3.

**EQUATION 8-3: PWM MAX RESOLUTION**

$$\text{PWM Maximum Resolution} = \frac{\log\left(\frac{F_{\text{OSC}}}{F_{\text{PWM}}}\right)}{\log(2)} \text{ bits}$$

**Note:** If the PWM duty cycle value is longer than the PWM period, the CCP1 pin will not be cleared.

For a sample PWM period and duty cycle calculation, see the PICmicro™ Mid-Range MCU Reference Manual (DS33023).

## 8.3.3 SET-UP FOR PWM OPERATION

The following steps should be taken when configuring the CCP module for PWM operation:

1. Set the PWM period by writing to the PR2 register.
2. Set the PWM duty cycle by writing to the CCPR1L register and CCP1CON<5:4> bits.
3. Make the CCP1 pin an output by clearing the TRISC<2> bit.
4. Set the TMR2 prescale value and enable Timer2 by writing to T2CON.
5. Configure the CCP1 module for PWM operation.

**TABLE 8-3: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS AT 20 MHz**

PWM Frequency	1.22 kHz	4.88 kHz	19.53 kHz	78.12 kHz	156.3 kHz	208.3 kHz
Timer Prescaler (1, 4, 16)	16	4	1	1	1	1
PR2 Value	0xFF	0xFF	0xFF	0x3F	0x1F	0x17
Maximum Resolution (bits)	10	10	10	8	7	5.5

**TABLE 8-4: REGISTERS ASSOCIATED WITH PWM AND TIMER2**

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
0Bh,8Bh 10Bh,18Bh	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	—	ADIF	—	—	SSPIF	CCP1IF	TMR2IF	TMR1IF	-0-- 0000	0000 0000
8Ch	PIE1	—	ADIE	—	—	SSPIE	CCP1IE	TMR2IE	TMR1IE	-0-- 0000	0000 0000
87h	TRISC	PORTC Data Direction Register								1111 1111	1111 1111
11h	TMR2	Timer2 Module Register								0000 0000	0000 0000
92h	PR2	Timer2 Module Period Register								1111 1111	1111 1111
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000
15h	CCPR1L	Capture/Compare/PWM Register1 (LSB)								xxxx xxxx	uuuu uuuu
16h	CCPR1H	Capture/Compare/PWM Register1 (MSB)								xxxx xxxx	uuuu uuuu
17h	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	--00 0000

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used by PWM and Timer2.

## 9.0 SYNCHRONOUS SERIAL PORT (SSP) MODULE

### 9.1 SSP Module Overview

The Synchronous Serial Port (SSP) module is a serial interface useful for communicating with other peripheral or microcontroller devices. These peripheral devices may be Serial EEPROMs, shift registers, display drivers, A/D converters, etc. The SSP module can operate in one of two modes:

- Serial Peripheral Interface (SPI)
- Inter-Integrated Circuit (I<sup>2</sup>C)

An overview of I<sup>2</sup>C operations and additional information on the SSP module can be found in the PICmicro™ Mid-Range MCU Family Reference Manual (DS33023).

Refer to Application Note AN578, "Use of the SSP Module in the I<sup>2</sup>C Multi-Master Environment."

### 9.2 SPI Mode

This section contains register definitions and operational characteristics of the SPI module.

SPI mode allows 8 bits of data to be synchronously transmitted and received simultaneously. To accomplish communication, typically three pins are used:

- Serial Data Out (SDO) RC5/SDO
- Serial Data In (SDI) RC4/SDI/SDA
- Serial Clock (SCK) RC3/SCK/SCL

Additionally, a fourth pin may be used when in a Slave mode of operation:

- Slave Select ( $\overline{SS}$ ) RA5/AN4/ $\overline{SS}$

When initializing the SPI, several options need to be specified. This is done by programming the appropriate control bits in the SSPCON register (SSPCON<5:0>) and SSPSTAT<7:6>. These control bits allow the following to be specified:

- Master mode (SCK is the clock output)
- Slave mode (SCK is the clock input)
- Clock Polarity (IDLE state of SCK)
- Clock edge (output data on rising/falling edge of SCK)
- Clock Rate (Master mode only)
- Slave Select mode (Slave mode only)

# PIC16F72

## REGISTER 9-1: SSPSTAT: SYNCHRONOUS SERIAL PORT STATUS REGISTER (ADDRESS 94h)

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
SMP	CKE	D/ $\bar{A}$	P	S	$\bar{R}/\bar{W}$	UA	BF

bit 7

bit 0

- bit 7 **SMP:** SPI Data Input Sample Phase bits  
SPI Master mode:  
 1 = Input data sampled at end of data output time  
 0 = Input data sampled at middle of data output time (Microwire®)  
SPI Slave mode:  
 SMP must be cleared when SPI is used in Slave mode  
I<sup>2</sup>C mode:  
 This bit must be maintained clear
- bit 6 **CKE:** SPI Clock Edge Select bits (Figure 9-2, Figure 9-3, and Figure 9-4)  
SPI mode, CKP = 0:  
 1 = Data transmitted on rising edge of SCK (Microwire alternate)  
 0 = Data transmitted on falling edge of SCK  
SPI mode, CKP = 1:  
 1 = Data transmitted on falling edge of SCK (Microwire default)  
 0 = Data transmitted on rising edge of SCK  
I<sup>2</sup>C mode:  
 This bit must be maintained clear
- bit 5  **$\bar{D}/\bar{A}$ :** Data/Address bit (I<sup>2</sup>C mode only)  
 1 = Indicates that the last byte received or transmitted was data  
 0 = Indicates that the last byte received or transmitted was address
- bit 4 **P:** STOP bit (I<sup>2</sup>C mode only) – This bit is cleared when the SSP module is disabled, or when the START bit is detected last. SSPEN is cleared.  
 1 = Indicates that a STOP bit has been detected last (this bit is '0' on RESET)  
 0 = STOP bit was not detected last
- bit 3 **S:** START bit (I<sup>2</sup>C mode only) – This bit is cleared when the SSP module is disabled, or when the STOP bit is detected last. SSPEN is cleared.  
 1 = Indicates that a START bit has been detected last (this bit is '0' on RESET)  
 0 = START bit was not detected last
- bit 2  **$\bar{R}/\bar{W}$ :** Read/Write Information bit (I<sup>2</sup>C mode only) – This bit holds the R/W bit information following the last address match. This bit is only valid from the address match to the next START bit, STOP bit, or ACK bit.  
 1 = Read  
 0 = Write
- bit 1 **UA:** Update Address bit (10-bit I<sup>2</sup>C mode only)  
 1 = Indicates that the user needs to update the address in the SSPADD register  
 0 = Address does not need to be updated
- bit 0 **BF:** Buffer Full Status bit  
Receive (SPI and I<sup>2</sup>C modes):  
 1 = Receive complete, SSPBUF is full  
 0 = Receive not complete, SSPBUF is empty  
Transmit (I<sup>2</sup>C mode only):  
 1 = Transmit in progress, SSPBUF is full  
 0 = Transmit complete, SSPBUF is empty

**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

## REGISTER 9-2: SSPCON: SYNC SERIAL PORT CONTROL REGISTER (ADDRESS 14h)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0
bit 7					bit 0		

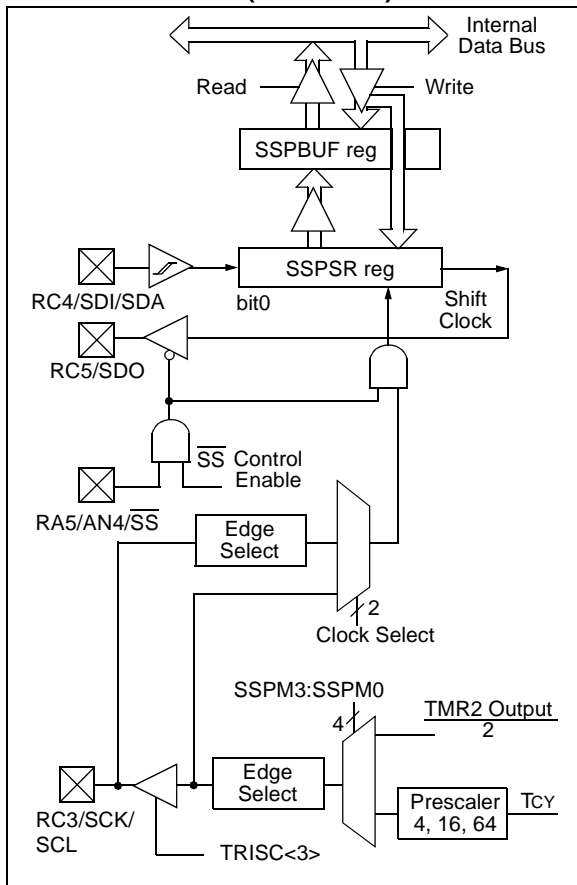
- bit 7 **WCOL:** Write Collision Detect bit  
 1 = The SSPBUF register is written while it is still transmitting the previous word (must be cleared in software)  
 0 = No collision
- bit 6 **SSPOV:** Receive Overflow Indicator bit  
In SPI mode:  
 1 = A new byte is received while the SSPBUF register is still holding the previous data. In case of overflow, the data in SSPSR is lost. Overflow can only occur in Slave mode. The user must read the SSPBUF, even if only transmitting data, to avoid setting overflow. In Master mode, the overflow bit is not set since each new reception (and transmission) is initiated by writing to the SSPBUF register.  
 0 = No overflow  
In I<sup>2</sup>C mode:  
 1 = A byte is received while the SSPBUF register is still holding the previous byte. SSPOV is a "don't care" in Transmit mode. SSPOV must be cleared in software in either mode.  
 0 = No overflow
- bit 5 **SSPEN:** Synchronous Serial Port Enable bit  
In SPI mode:  
 1 = Enables serial port and configures SCK, SDO, and SDI as serial port pins  
 0 = Disables serial port and configures these pins as I/O port pins  
In I<sup>2</sup>C mode:  
 1 = Enables the serial port and configures the SDA and SCL pins as serial port pins  
 0 = Disables serial port and configures these pins as I/O port pins  
 In both modes, when enabled, these pins must be properly configured as input or output.
- bit 4 **CKP:** Clock Polarity Select bit  
In SPI mode:  
 1 = IDLE state for clock is a high level (Microwire<sup>®</sup> default)  
 0 = IDLE state for clock is a low level (Microwire alternate)  
In I<sup>2</sup>C mode:  
**SCK release control**  
 1 = Enable clock  
 0 = Holds clock low (clock stretch - used to ensure data setup time)
- bit 3-0 **SSPM<3:0>:** Synchronous Serial Port Mode Select bits  
 0000 = SPI Master mode, clock = Fosc/4  
 0001 = SPI Master mode, clock = Fosc/16  
 0010 = SPI Master mode, clock = Fosc/64  
 0011 = SPI Master mode, clock = TMR2 output/2  
 0100 = SPI Slave mode, clock = SCK pin.  $\overline{SS}$  pin control enabled.  
 0101 = SPI Slave mode, clock = SCK pin.  $\overline{SS}$  pin control disabled.  $\overline{SS}$  can be used as I/O pin.  
 0110 = I<sup>2</sup>C Slave mode, 7-bit address  
 0111 = I<sup>2</sup>C Slave mode, 10-bit address  
 1011 = I<sup>2</sup>C firmware controlled Master mode (Slave IDLE)  
 1110 = I<sup>2</sup>C Slave mode, 7-bit address with START and STOP bit interrupts enabled  
 1111 = I<sup>2</sup>C Slave mode, 10-bit address with START and STOP bit interrupts enabled

**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

# PIC16F72

**FIGURE 9-1: SSP BLOCK DIAGRAM (SPI MODE)**



To enable the serial port, SSP enable bit, SSPEN (SSPCON<5>) must be set. To reset or reconfigure SPI mode, clear bit SSPEN, re-initialize the SSPCON register, and then set bit SSPEN. This configures the SDI, SDO, SCK, and SS pins as serial port pins. For the pins to behave as the serial port function, they must have their data direction bits (in the TRISC register) appropriately programmed. That is:

- SDI must have TRISC<4> set
- SDO must have TRISC<5> cleared
- SCK (Master mode) must have TRISC<3> cleared
- SCK (Slave mode) must have TRISC<3> set
- $\overline{SS}$  must have TRISA<5> set and ADCON must be configured such that RA5 is a digital I/O

**Note 1:** When the SPI is in Slave mode with  $\overline{SS}$  pin control enabled (SSPCON<3:0> = 0100), the SPI module will reset if the  $\overline{SS}$  pin is set to VDD.

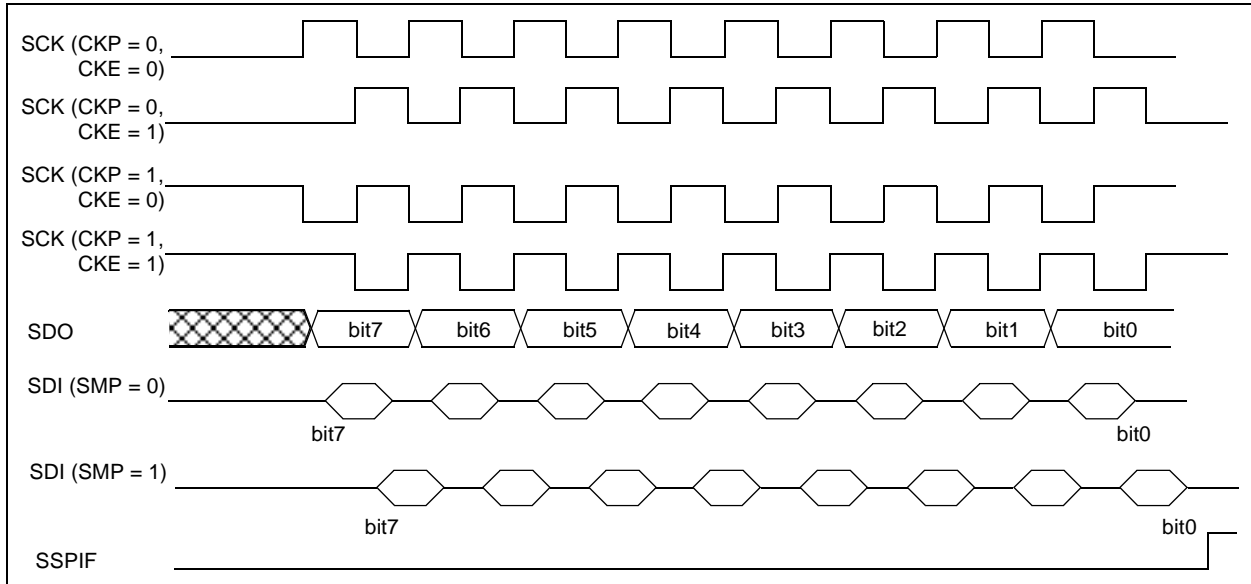
**Note 2:** If the SPI is used in Slave mode with CKE = '1', then the  $\overline{SS}$  pin control must be enabled.

**TABLE 9-1: REGISTERS ASSOCIATED WITH SPI OPERATION**

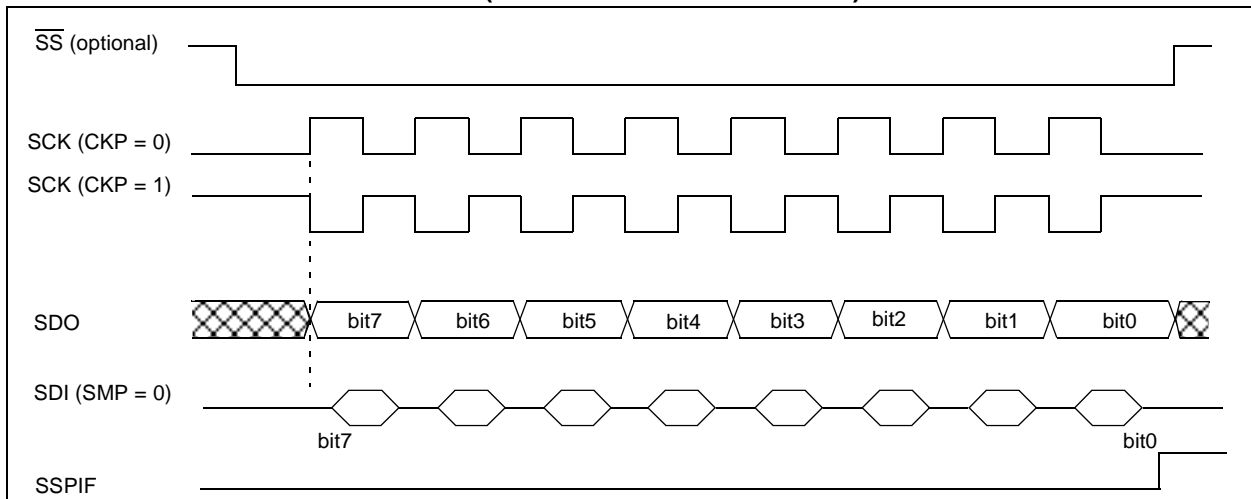
Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other RESETS
0Bh,8Bh 10Bh,18Bh	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	—	ADIF	—	—	SSPIF	CCP1IF	TMR2IF	TMR1IF	-0-- 0000	0000 0000
8Ch	PIE1	—	ADIE	—	—	SSPIE	CCP1IE	TMR2IE	TMR1IE	-0-- 0000	0000 0000
87h	TRISC	PORTC Data Direction Register								1111 1111	1111 1111
13h	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								xxxxx xxxxx	uuuu uuuu
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000
85h	TRISA	—	—	PORTA Data Direction Register						--11 1111	--11 1111
94h	SSPSTAT	—	—	D/ $\overline{A}$	P	S	R/ $\overline{W}$	UA	BF	--00 0000	--00 0000

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used by the SSP in SPI mode.

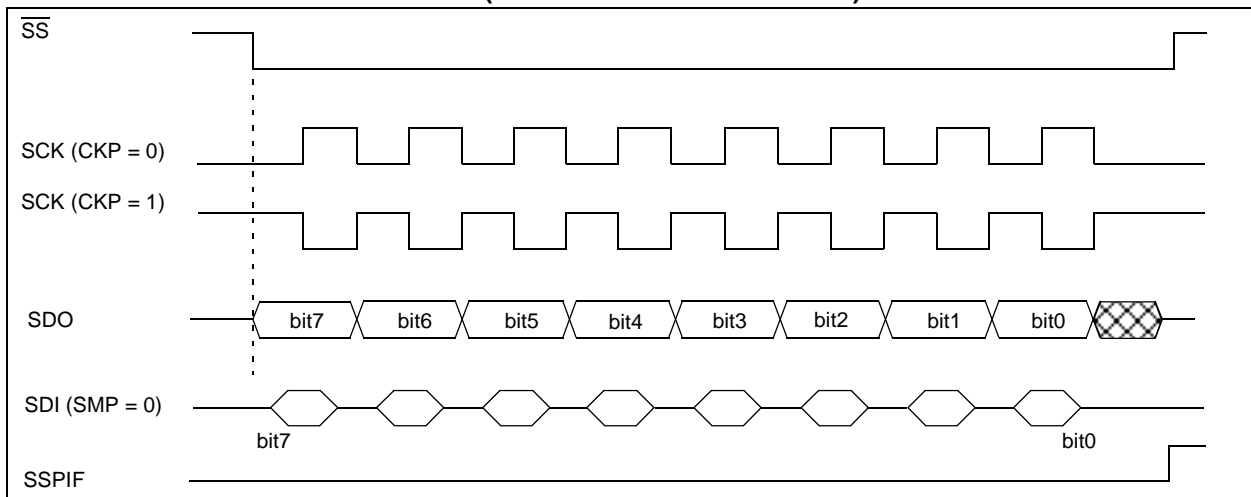
**FIGURE 9-2: SPI MODE TIMING, MASTER MODE**



**FIGURE 9-3: SPI MODE TIMING (SLAVE MODE WITH CKE = 0)**



**FIGURE 9-4: SPI MODE TIMING (SLAVE MODE WITH CKE = 1)**





# PIC16F72

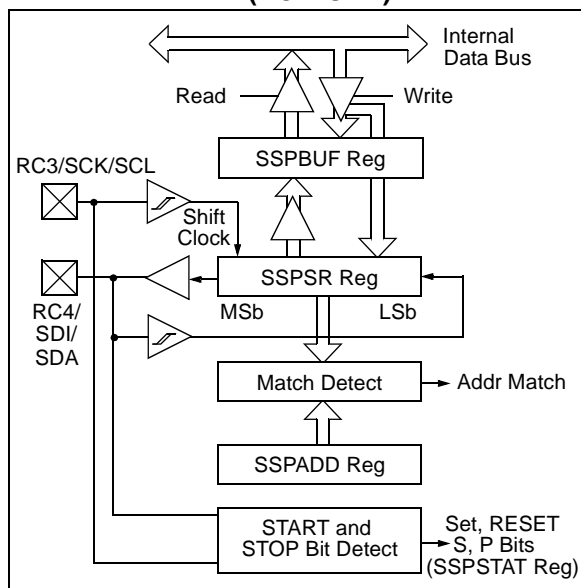
## 9.3 SSP I<sup>2</sup>C Mode Operation

The SSP module in I<sup>2</sup>C mode fully implements all slave functions, except general call support and provides interrupts on START and STOP bits in hardware to facilitate firmware implementations of the master functions. The SSP module implements the Standard mode specifications, as well as 7-bit and 10-bit addressing.

Two pins are used for data transfer. These are the RC3/SCK/SCL pin, which is the clock (SCL), and the RC4/SDI/SDA pin, which is the data (SDA). The user must configure these pins as inputs or outputs through the TRISC<4:3> bits.

The SSP module functions are enabled by setting SSP Enable bit SSPEN (SSPCON<5>).

**FIGURE 9-5: SSP BLOCK DIAGRAM (I<sup>2</sup>C MODE)**



The SSP module has five registers for I<sup>2</sup>C operation:

- SSP Control Register (SSPCON)
- SSP Status Register (SSPSTAT)
- Serial Receive/Transmit Buffer (SSPBUF)
- SSP Shift Register (SSPSR) - Not directly accessible
- SSP Address Register (SSPADD)

The SSPCON register allows control of the I<sup>2</sup>C operation. Four mode selection bits (SSPCON<3:0>) allow one of the following I<sup>2</sup>C modes to be selected:

- I<sup>2</sup>C Slave mode (7-bit address)
- I<sup>2</sup>C Slave mode (10-bit address)
- I<sup>2</sup>C Slave mode (7-bit address), with START and STOP bit interrupts enabled
- I<sup>2</sup>C Slave mode (10-bit address), with START and STOP bit interrupts enabled
- I<sup>2</sup>C Firmware controlled Master operation, Slave is IDLE

Selection of any I<sup>2</sup>C mode, with the SSPEN bit set, forces the SCL and SDA pins to be open drain, provided these pins are programmed to inputs by setting the appropriate TRISC bits.

Additional information on SSP I<sup>2</sup>C operation may be found in the PICmicro™ Mid-Range MCU Reference Manual (DS33023).

### 9.3.1 SLAVE MODE

In Slave mode, the SCL and SDA pins must be configured as inputs (TRISC<4:3> set). The SSP module will override the input state with the output data when required (slave-transmitter).

When an address is matched or the data transfer after an address match is received, the hardware automatically will generate the Acknowledge ( $\overline{ACK}$ ) pulse, and then load the SSPBUF register with the received value currently in the SSPSR register.

Either or both of the following conditions will cause the SSP module not to give this  $\overline{ACK}$  pulse.

- a) The buffer full bit BF (SSPSTAT<0>) was set before the transfer was received.
- b) The overflow bit SSPOV (SSPCON<6>) was set before the transfer was received.

In this case, the SSPSR register value is not loaded into the SSPBUF, but bit SSPIF (PIR1<3>) is set. Table 9-2 shows what happens when a data transfer byte is received, given the status of bits BF and SSPOV. The shaded cells show the condition where user software did not properly clear the overflow condition. Flag bit BF is cleared by reading the SSPBUF register while bit SSPOV is cleared through software.

The SCL clock input must have a minimum high and low for proper operation. The high and low times of the I<sup>2</sup>C specification, as well as the requirement of the SSP module are shown in timing parameter #100 and parameter #101.

#### 9.3.1.1 Addressing

Once the SSP module has been enabled, it waits for a START condition to occur. Following the START condition, the eight bits are shifted into the SSPSR register. All incoming bits are sampled with the rising edge of the clock (SCL) line. The value of register SSPSR<7:1> is compared to the value of the SSPADD register. The address is compared on the falling edge of the eighth clock (SCL) pulse. If the addresses match, and the BF and SSPOV bits are clear, the following events occur:

- a) The SSPSR register value is loaded into the SSPBUF register.
- b) The buffer full bit, BF is set.
- c) An  $\overline{ACK}$  pulse is generated.
- d) SSP interrupt flag bit, SSPIF (PIR1<3>) is set (interrupt is generated, if enabled) - on the falling edge of the ninth SCL pulse.

In 10-bit Address mode, two address bytes need to be received by the slave device. The five Most Significant bits (MSBs) of the first address byte specify if this is a 10-bit address. Bit  $\overline{R/W}$  (SSPSTAT<2>) must specify a write so the slave device will receive the second address byte. For a 10-bit address the first byte would equal '1111 0 A9 A8 0', where A9 and A8 are the two MSBs of the address.

The sequence of events for 10-bit address is as follows, with steps 7- 9 for slave-transmitter:

1. Receive first (high) byte of address (bits SSPIF, BF, and bit UA (SSPSTAT<1>) are set).
2. Update the SSPADD register with second (low) byte of address (clears bit UA and releases the SCL line).
3. Read the SSPBUF register (clears bit BF) and clear flag bit SSPIF.
4. Receive second (low) byte of address (bits SSPIF, BF, and UA are set).
5. Update the SSPADD register with the first (high) byte of Address, if match releases SCL line, this will clear bit UA.
6. Read the SSPBUF register (clears bit BF) and clear flag bit SSPIF.
7. Receive Repeated START condition.
8. Receive first (high) byte of address (bits SSPIF and BF are set).
9. Read the SSPBUF register (clears bit BF) and clear flag bit SSPIF.

### 9.3.1.2 Reception

When the  $\overline{R/W}$  bit of the address byte is clear and an address match occurs, the  $\overline{R/W}$  bit of the SSPSTAT register is cleared. The received address is loaded into the SSPBUF register.

When the address byte overflow condition exists, then a no Acknowledge ( $\overline{ACK}$ ) pulse is given. An overflow condition is indicated if either bit BF (SSPSTAT<0>) is set, or bit SSPOV (SSPCON<6>) is set.

An SSP interrupt is generated for each data transfer byte. Flag bit SSPIF (PIR1<3>) must be cleared in software. The SSPSTAT register is used to determine the status of the byte.

### 9.3.1.3 Transmission

When the  $\overline{R/W}$  bit of the incoming address byte is set and an address match occurs, the  $\overline{R/W}$  bit of the SSPSTAT register is set. The received address is loaded into the SSPBUF register. The  $\overline{ACK}$  pulse will be sent on the ninth bit, and pin RC3/SCK/SCL is held low. The transmit data must be loaded into the SSPBUF register, which also loads the SSPSR register. Then pin RC3/SCK/SCL should be enabled by setting bit CKP (SSPCON<4>). The master device must monitor the SCL pin prior to asserting another clock pulse. The slave devices may be holding off the master device by stretching the clock. The eight data bits are shifted out on the falling edge of the SCL input. This ensures that the SDA signal is valid during the SCL high time (Figure 9-7).

An SSP interrupt is generated for each data transfer byte. Flag bit SSPIF must be cleared in software and the SSPSTAT register is used to determine the status of the byte. Flag bit SSPIF is set on the falling edge of the ninth clock pulse.

As a slave-transmitter, the  $\overline{ACK}$  pulse from the master-receiver is latched on the rising edge of the ninth SCL input pulse. If the SDA line was high (not  $\overline{ACK}$ ), then the data transfer is complete. When the  $\overline{ACK}$  is latched by the slave device, the slave logic is reset (resets SSPSTAT register) and the slave device then monitors for another occurrence of the START bit. If the SDA line was low ( $\overline{ACK}$ ), the transmit data must be loaded into the SSPBUF register, which also loads the SSPSR register. Then, pin RC3/SCK/SCL should be enabled by setting bit CKP.

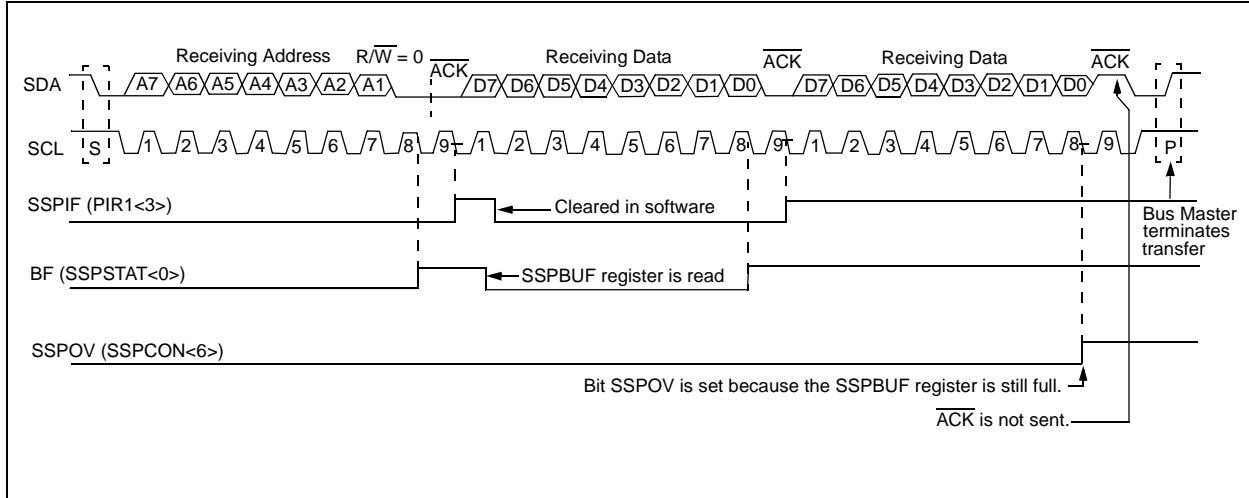
**TABLE 9-2: DATA TRANSFER RECEIVED BYTE ACTIONS**

Status Bits as Data Transfer is Received		SSPSR → SSPBUF	Generate $\overline{ACK}$ Pulse	Set bit SSPIF (SSP Interrupt occurs if enabled)
BF	SSPOV			
0	0	Yes	Yes	Yes
1	0	No	No	Yes
1	1	No	No	Yes
0	1	No	No	Yes

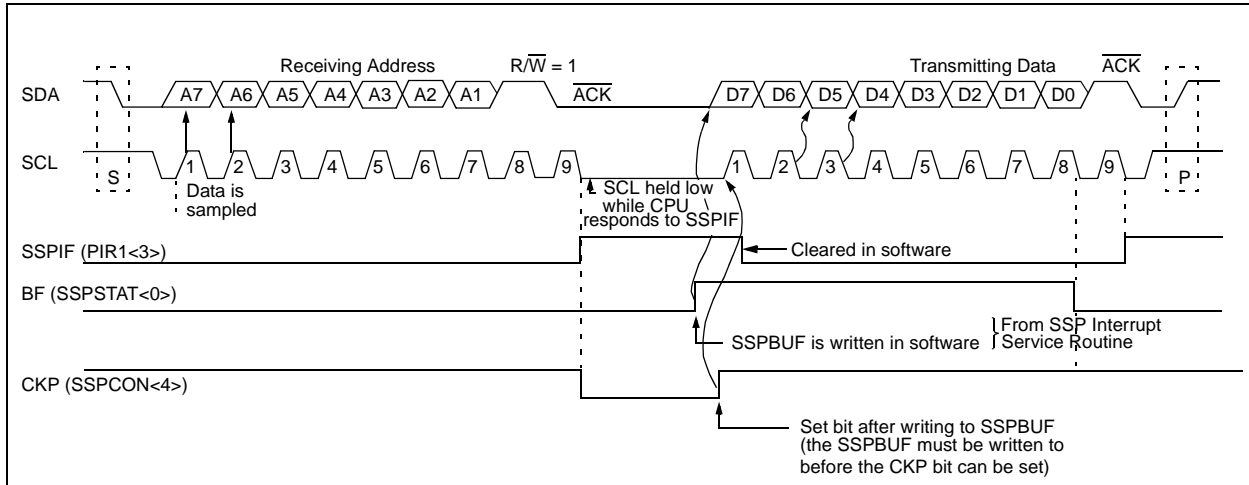
**Note 1:** Shaded cells show the conditions where the user software did not properly clear the overflow condition.

# PIC16F72

**FIGURE 9-6: I<sup>2</sup>C WAVEFORMS FOR RECEPTION (7-BIT ADDRESS)**



**FIGURE 9-7: I<sup>2</sup>C WAVEFORMS FOR TRANSMISSION (7-BIT ADDRESS)**



## 9.3.2 MASTER MODE OPERATION

Master mode operation is supported in firmware using interrupt generation on the detection of the START and STOP conditions. The STOP (P) and START (S) bits are cleared from a RESET or when the SSP module is disabled. The STOP (P) and START (S) bits will toggle, based on the START and STOP conditions. Control of the I<sup>2</sup>C bus may be taken when the P bit is set, or the bus is IDLE and both the S and P bits are clear.

In Master mode operation, the SCL and SDA lines are manipulated in firmware by clearing the corresponding TRISC<4:3> bit(s). The output level is always low, irrespective of the value(s) in PORTC<4:3>. So, when transmitting data, a '1' data bit must have the TRISC<4> bit set (input) and a '0' data bit must have the TRISC<4> bit cleared (output). The same scenario is true for the SCL line with the TRISC<3> bit.

The following events will cause the SSP Interrupt Flag bit, SSPIF, to be set (SSP Interrupt if enabled):

- START condition
- STOP condition
- Data transfer byte transmitted/received

Master mode operation can be done with either the Slave mode IDLE (SSPM3:SSPM0 = 1011), or with the Slave mode active. When both Master mode operation and Slave modes are used, the software needs to differentiate the source(s) of the interrupt.

For more information on Master mode operation, see AN554 - Software Implementation of I<sup>2</sup>C Bus Master.

## 9.3.3 MULTI-MASTER MODE OPERATION

In Multi-Master mode operation, the interrupt generation on the detection of the START and STOP conditions allows the determination of when the bus is free. The STOP (P) and START (S) bits are cleared from a RESET or when the SSP module is disabled. The STOP (P) and START (S) bits will toggle, based on the START and STOP conditions. Control of the I<sup>2</sup>C bus may be taken when bit P (SSPSTAT<4>) is set, or the bus is IDLE and both the S and P bits clear. When the bus is busy, enabling the SSP interrupt will generate the interrupt when the STOP condition occurs.

In Multi-Master mode operation, the SDA line must be monitored to see if the signal level is the expected output level. This check only needs to be done when a high level is output. If a high level is expected and a low level is present, the device needs to release the SDA and SCL lines (set TRISC<4:3>). There are two stages where this arbitration can be lost:

- Address Transfer
- Data Transfer

When the slave logic is enabled, the Slave device continues to receive. If arbitration was lost during the address transfer stage, communication to the device may be in progress. If addressed, an ACK pulse will be generated. If arbitration was lost during the data transfer stage, the device will need to retransfer the data at a later time.

For more information on Multi-Master mode operation, see AN578 - Use of the SSP Module in the I<sup>2</sup>C Multi-Master Environment.

**TABLE 9-3: REGISTERS ASSOCIATED WITH I<sup>2</sup>C OPERATION**

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
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	—	ADIF	—	—	SSPIF	CCP1IF	TMR2IF	TMR1IF	-0-- 0000	0000 0000
8Ch	PIE1	—	ADIE	—	—	SSPIE	CCP1IE	TMR2IE	TMR1IE	-0-- 0000	0000 0000
13h	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								xxxx xxxx	uuuu uuuu
93h	SSPADD	Synchronous Serial Port (I <sup>2</sup> C mode) Address Register								0000 0000	0000 0000
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000
94h	SSPSTAT	SMP <sup>(1)</sup>	CKE <sup>(1)</sup>	D/ $\bar{A}$	P	S	R/ $\bar{W}$	UA	BF	0000 0000	0000 0000
87h	TRISC	PORTC Data Direction Register								1111 1111	1111 1111

Legend: x = unknown, u = unchanged, - = unimplemented locations read as '0'.  
Shaded cells are not used by SSP module in SPI mode.

**Note 1:** Maintain these bits clear in I<sup>2</sup>C mode.

# PIC16F72

---

NOTES:

## 10.0 ANALOG-TO-DIGITAL CONVERTER (A/D) MODULE

The analog-to-digital (A/D) converter module has five inputs for the PIC16F72.

The A/D allows conversion of an analog input signal to a corresponding 8-bit digital number. The output of the sample and hold is the input into the converter, which generates the result via successive approximation. The analog reference voltage is software selectable to either the device's positive supply voltage (VDD) or the voltage level on the RA3/AN3/VREF pin.

The A/D converter has a unique feature of being able to operate while the device is in SLEEP mode. To operate in SLEEP, the A/D conversion clock must be derived from the A/D's internal RC oscillator.

The A/D module has three registers:

- A/D Result Register      ADRES
- A/D Control Register 0    ADCON0
- A/D Control Register 1    ADCON1

A device RESET forces all registers to their RESET state. This forces the A/D module to be turned off and any conversion is aborted.

The ADCON0 register, shown in Register 10-1, controls the operation of the A/D module. The ADCON1 register, shown in Register 10-2, configures the functions of the port pins. The port pins can be configured as analog inputs (RA3 can also be a voltage reference) or a digital I/O.

For more information on use of the A/D Converter, see *AN546 - Use of A/D Converter*, or refer to the PICmicro™ Mid-Range MCU Family Reference Manual (DS33023).

### REGISTER 10-1: ADCON0: A/D CONTROL REGISTER 0 (ADDRESS 1Fh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	—	ADON
							bit 0

- bit 7-6      **ADCS<1:0>**: A/D Conversion Clock Select bits  
 00 = FOSC/2  
 01 = FOSC/8  
 10 = FOSC/32  
 11 = FRC (clock derived from the internal A/D module RC oscillator)
- bit 5-3      **CHS<2:0>**: Analog Channel Select bits  
 000 = Channel 0, (RA0/AN0)  
 001 = Channel 1, (RA1/AN1)  
 010 = Channel 2, (RA2/AN2)  
 011 = Channel 3, (RA3/AN3)  
 100 = Channel 4, (RA5/AN4)
- bit 2      **GO/DONE**: A/D Conversion Status bit  
If ADON = 1:  
 1 = A/D conversion in progress (setting this bit starts the A/D conversion)  
 0 = A/D conversion not in progress (this bit is automatically cleared by hardware when the A/D conversion is complete)
- bit 1      **Unimplemented**: Read as '0'
- bit 0      **ADON**: A/D On bit  
 1 = A/D converter module is operating  
 0 = A/D converter module is shut-off and consumes no operating current

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

# PIC16F72

## REGISTER 10-2: ADCON1: A/D CONTROL REGISTER 1 (ADDRESS 9Fh)

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	PCFG2	PCFG1	PCFG0
bit 7					bit 0		

bit 7-3 **Unimplemented:** Read as '0'

bit 2-0 **PCFG<2:0>:** A/D Port Configuration Control bits

PCFG2:PCFG0	RA0	RA1	RA2	RA5	RA3	VREF
000	A	A	A	A	A	VDD
001	A	A	A	A	VREF	RA3
010	A	A	A	A	A	VDD
011	A	A	A	A	VREF	RA3
100	A	A	D	D	A	VDD
101	A	A	D	D	VREF	RA3
11x	D	D	D	D	D	VDD

A = Analog input

D = Digital I/O

### 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

The ADRES register contains the result of the A/D conversion. When the A/D conversion is complete, the result is loaded into the ADRES register, the  $\overline{\text{GO/DONE}}$  bit (ADCON0<2>) is cleared, and A/D interrupt flag bit ADIF is set. The block diagram of the A/D module is shown in Figure 10-1.

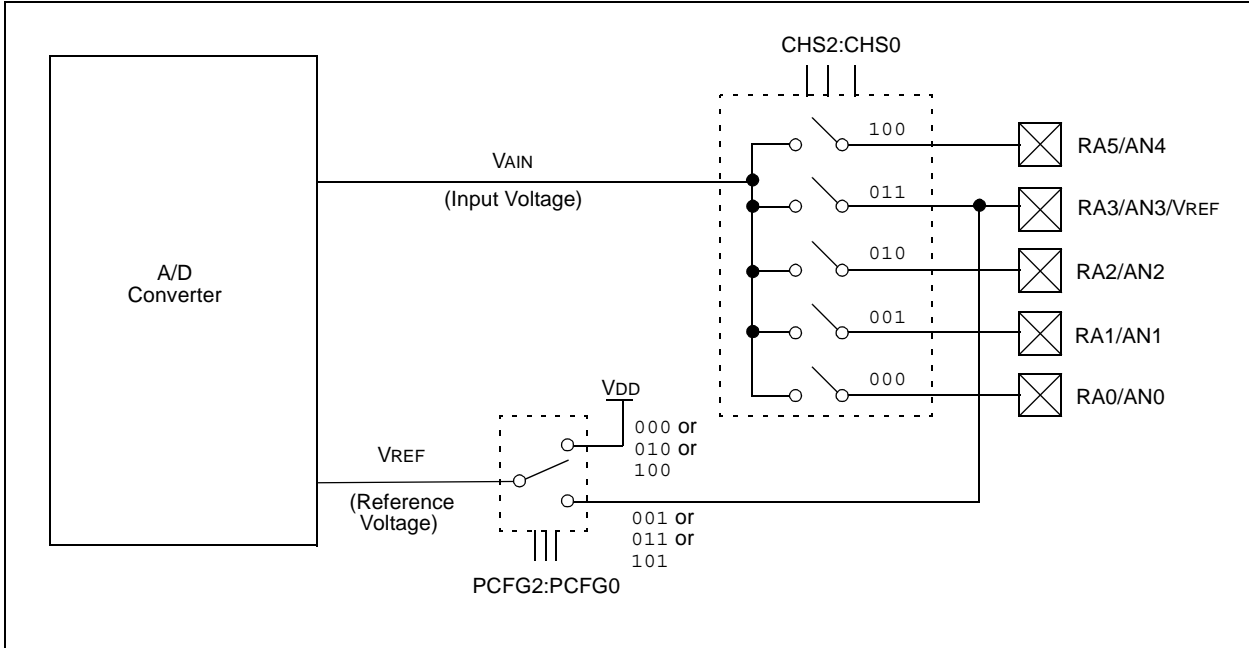
The value in the ADRES register is not modified for a Power-on Reset. The ADRES register will contain unknown data after a Power-on Reset.

After the A/D module has been configured as desired, the selected channel must be acquired before the conversion is started. The analog input channels must have their corresponding TRIS bits selected as an input. To determine acquisition time, see Section 10.1. After this acquisition time has elapsed, the A/D conversion can be started.

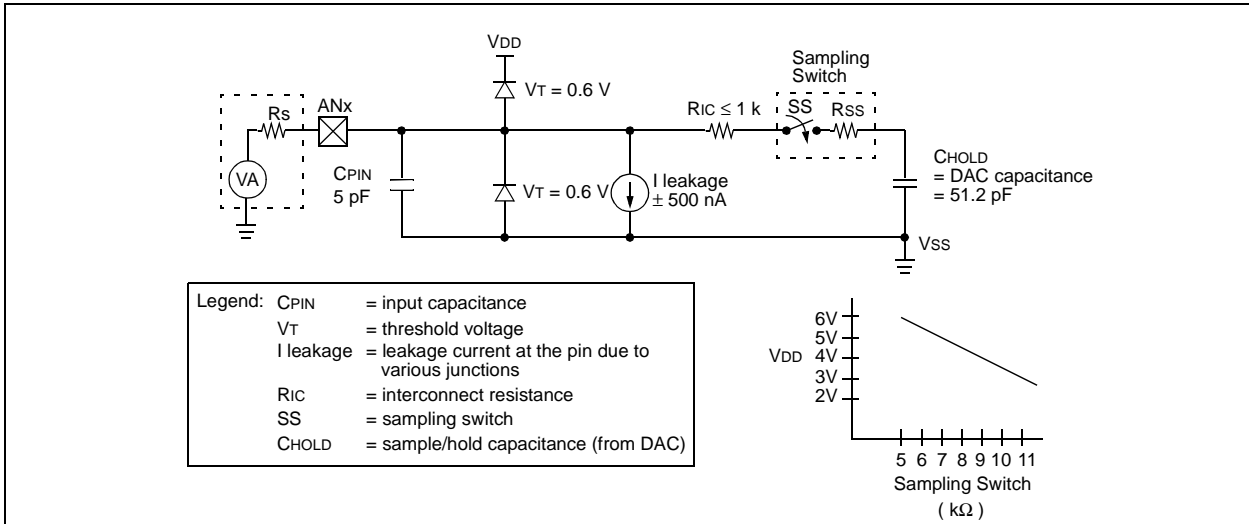
The following steps should be followed for doing an A/D conversion:

- Configure the A/D module:
  - Configure analog pins/voltage reference and digital I/O (ADCON1)
  - Select A/D input channel (ADCON0)
  - Select A/D conversion clock (ADCON0)
  - Turn on A/D module (ADCON0)
- Configure A/D interrupt (if desired):
  - Clear ADIF bit
  - Set ADIE bit
  - Set GIE bit
- Wait the required acquisition time.
- Start conversion:
  - Set  $\overline{\text{GO/DONE}}$  bit (ADCON0)
- Wait for A/D conversion to complete, by either:
  - Polling for the  $\overline{\text{GO/DONE}}$  bit to be cleared
 OR
  - Waiting for the A/D interrupt
- Read A/D Result register (ADRES), clear bit ADIF if required.
- For next conversion, go to step 1 or step 2 as required. The A/D conversion time per bit is defined as TAD. A minimum wait of 2 TAD is required before the next acquisition starts.

**FIGURE 10-1: A/D BLOCK DIAGRAM**



**FIGURE 10-2: ANALOG INPUT MODEL**





# PIC16F72

## 10.1 A/D Acquisition Requirements

For the A/D converter 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 10-2. The source impedance (RS) and the internal sampling switch (RSS) impedance directly affect the time required to charge the capacitor CHOLD. The sampling switch (RSS) impedance varies over the device voltage (VDD). The source impedance affects the offset voltage at the analog input (due to pin leakage current).

**The maximum recommended impedance for analog sources is 10 kΩ.** After the analog input channel is selected (changed), this acquisition must be done before the conversion can be started.

To calculate the minimum acquisition time, TACQ, see the PICmicro™ Mid-Range MCU Reference Manual, (DS33023). In general, however, given a max of 10 kΩ and at a temperature of 100°C, TACQ will be no more than 16 μs. [www.trimmer.ru](http://www.trimmer.ru)

## 10.2 Selecting the A/D Conversion Clock

The A/D conversion time per bit is defined as TAD. The A/D conversion requires 9.0 TAD per 8-bit conversion. The source of the A/D conversion clock is software selectable. The four possible options for TAD are:

- 2 TOSC
- 8 TOSC
- 32 TOSC
- Internal RC oscillator (2 - 6 μs)

For correct A/D conversions, the A/D conversion clock (TAD) must be selected to ensure a minimum TAD time as small as possible, but no less than 1.6 μs and not greater than 6.4 μs.

Table 10-1 shows the resultant TAD times derived from the device operating frequencies and the A/D clock source selected.

## 10.3 Configuring Analog Port Pins

The ADCON1, and TRISA registers control the operation of the A/D port pins. The port pins that are desired as analog inputs must have their corresponding TRIS bits set (input). If the TRIS bit is cleared (output), the digital output level (VOH or VOL) will be converted.

The A/D operation is independent of the state of the CHS<2:0> bits and the TRIS bits.

**Note 1:** When reading the port register, all pins configured as analog input channels will read as cleared (a low level). Pins configured as digital inputs, will convert an analog input. Analog levels on a digitally configured input will not affect the conversion accuracy.

**2:** Analog levels on any pin that is defined as a digital input (including the AN4:AN0 pins), may cause the input buffer to consume current out of the device specification.

## 10.4 A/D Conversions

**Note:** The GO/DONE bit should **NOT** be set in the same instruction that turns on the A/D.

Clearing the GO/DONE bit during a conversion will abort the current conversion. The ADRES register will NOT be updated with the partially completed A/D conversion sample. That is, the ADRES register will continue to contain the value of the last completed conversion (or the last value written to the ADRES register). After the A/D conversion is aborted, a 2 TAD wait is required before the next acquisition is started. After this 2 TAD wait, an acquisition is automatically started on the selected channel. The GO/DONE bit can then be set to start the conversion.

**TABLE 10-1: TAD vs. MAXIMUM DEVICE OPERATING FREQUENCIES (STANDARD DEVICES (C))**

AD Clock Source (TAD)		Maximum Device Frequency
Operation	ADCS<1:0>	Max.
2 TOSC	00	1.25 MHz
8 TOSC	01	5 MHz
32 TOSC	10	20 MHz
RC <sup>(1, 2)</sup>	11	<b>(Note 1)</b>

**Note 1:** The RC source has a typical TAD time of 4 μs, but can vary between 2-6 μs.

**2:** When the device frequencies are greater than 1 MHz, the RC A/D conversion clock source is only recommended for SLEEP operation.

## 10.5 A/D Operation During SLEEP

The A/D module can operate during SLEEP mode. This requires that the A/D clock source be set to RC (ADCS1:ADCS0 = 11). When the RC clock source is selected, the A/D module waits one instruction cycle before starting the conversion. This allows the SLEEP instruction to be executed, which eliminates all digital switching noise from the conversion. When the conversion is completed, the GO/DONE bit will be cleared, and the result loaded into the ADRES register. If the A/D interrupt is enabled, the device will wake-up from SLEEP. If the A/D interrupt is not enabled, the A/D module will then be turned off, although the ADON bit will remain set.

When the A/D clock source is another clock option (not RC), a SLEEP instruction will cause the present conversion to be aborted and the A/D module to be turned off, though the ADON bit will remain set.

Turning off the A/D places the A/D module in its lowest current consumption state.

**Note:** For the A/D module to operate in SLEEP, the A/D clock source must be set to RC (ADCS1:ADCS0 = 11). To perform an A/D conversion in SLEEP, ensure the SLEEP instruction immediately follows the instruction that sets the GO/DONE bit.

## 10.6 Effects of a RESET

A device RESET forces all registers to their RESET state. The A/D module is disabled and any conversion in progress is aborted. All A/D input pins are configured as analog inputs.

The ADRES register will contain unknown data after a Power-on Reset.

## 10.7 Use of the CCP Trigger

An A/D conversion can be started by the “special event trigger” of the CCP1 module. This requires that the CCP1M3:CCP1M0 bits (CCP1CON<3:0>) be programmed as 1011 and that the A/D module is enabled (ADON bit is set). When the trigger occurs, the GO/DONE bit will be set, starting the A/D conversion, and the Timer1 counter will be reset to zero. Timer1 is reset to automatically repeat the A/D acquisition period with minimal software overhead (moving the ADRES to the desired location). The appropriate analog input channel must be selected and the minimum acquisition done before the “special event trigger” sets the GO/DONE bit (starts a conversion).

If the A/D module is not enabled (ADON is cleared), then the “special event trigger” will be ignored by the A/D module, but will still reset the Timer1 counter.

**TABLE 10-2: REGISTERS/BITS ASSOCIATED WITH A/D**

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
0Bh,8Bh 10Bh,18Bh	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	—	ADIF	—	—	SSPIF	CCP1IF	TMR2IF	TMR1IF	-0-- 0000	-0-- 0000
8Ch	PIE1	—	ADIE	—	—	SSPIE	CCP1IE	TMR2IE	TMR1IE	-0-- 0000	-0-- 0000
1Eh	ADRES	A/D Result Register								xxxx xxxx	uuuu uuuu
1Fh	ADCON0	ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	—	ADON	0000 00-0	0000 00-0
9Fh	ADCON1	—	—	—	—	—	PCFG2	PCFG1	PCFG0	---- -000	---- -000
05h	PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	--0x 0000	--0u 0000
85h	TRISA	—	—	PORTA Data Direction Register						--11 1111	--11 1111

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used for A/D conversion.

## 11.0 SPECIAL FEATURES OF THE CPU

These devices have a host of features intended to maximize system reliability, minimize cost through elimination of external components, provide power saving Operating modes and offer code protection:

- Oscillator Selection
- RESET
  - Power-on Reset (POR)
  - Power-up Timer (PWRT)
  - Oscillator Start-up Timer (OST)
  - Brown-out Reset (BOR)
- Interrupts
- Watchdog Timer (WDT)
- SLEEP
- Code Protection
- ID Locations
- In-Circuit Serial Programming

These devices have a Watchdog Timer, which can be enabled or disabled using a configuration bit. It runs off its own RC oscillator for added reliability.

There are two timers that offer necessary delays on power-up. One is the Oscillator Start-up Timer (OST), intended to keep the chip in RESET until the crystal oscillator is stable. The other is the Power-up Timer (PWRT), which provides a fixed delay of 72 ms (nominal) on power-up only. It is designed to keep the part in RESET while the power supply stabilizes, and is enabled or disabled using a configuration bit. With these two timers on-chip, most applications need no external RESET circuitry.

SLEEP mode is designed to offer a very low current Power-down mode. The user can wake-up from SLEEP through external RESET, Watchdog Timer Wake-up, or through an interrupt.

Several oscillator options are also made available to allow the part to fit the application. The RC oscillator option saves system cost while the LP crystal option saves power. Configuration bits are used to select the desired oscillator mode.

Additional information on special features is available in the PICmicro™ Mid-Range Reference Manual (DS33023).

### 11.1 Configuration Bits

The configuration bits can be programmed (read as '0'), or left unprogrammed (read as '1'), to select various device configurations. These bits are mapped in program memory location 2007h.

The user will note that address 2007h is beyond the user program memory space, which can be accessed only during programming.

# PIC16F72

## REGISTER 11-1: CONFIGURATION WORD (ADDRESS 2007h)<sup>(1)</sup>

U-1	U-1	U-1	U-1	U-1	U-1	U-1	u-1	U-1	u-1	u-1	u-1	u-1	u-1	
—	—	—	—	—	—	—	BOREN	—	CP	$\overline{\text{PWRTEN}}$	WDTEN	F0SC1	F0SC0	
bit13														bit0

bit 13-7 **Unimplemented:** Read as '1'

bit 6 **BOREN:** Brown-out Reset Enable bit<sup>(2)</sup>

1 = BOR enabled  
0 = BOR disabled

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

bit 4 **CP:** FLASH Program Memory Code Protection bit

1 = Code protection off  
0 = All memory locations code protected

bit 3  **$\overline{\text{PWRTEN}}$ :** Power-up Timer Enable bit

1 = PWRT disabled  
0 = PWRT enabled

bit 2 **WDTEN:** Watchdog Timer Enable bit

1 = WDT enabled  
0 = WDT disabled

bit 1-0 **FOSC1:FOSC0:** Oscillator Selection bits

11 = RC oscillator  
10 = HS oscillator  
01 = XT oscillator  
00 = LP oscillator

**Note 1:** The erased (unprogrammed) value of the configuration word is 3FFFh.

**2:** Enabling Brown-out Reset automatically enables Power-up Timer (PWRT), regardless of the value of bit  $\overline{\text{PWRTEN}}$ . Ensure the Power-up Timer is enabled any time Brown-out Reset is enabled.

**Legend:**

R = Readable bit

P = Programmable bit

U = Unimplemented bit, read as '1'

- n = Value when device is unprogrammed

u = Unchanged from programmed state

## 11.2 Oscillator Configurations

### 11.2.1 OSCILLATOR TYPES

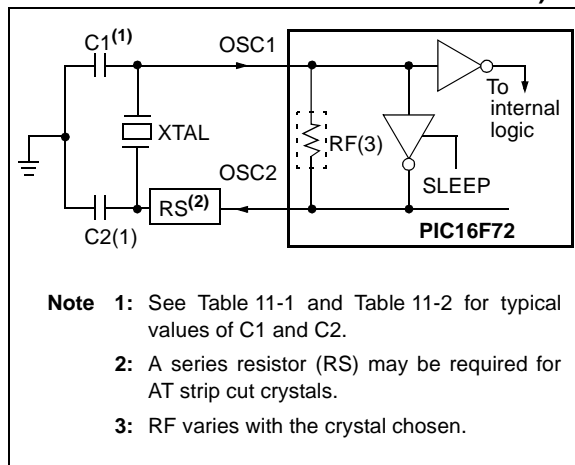
The PIC16F72 can be operated in four different Oscillator modes. The user can program two configuration bits (FOSC1 and FOSC0) to select one of these four modes:

- LP Low Power Crystal
- XT Crystal/Resonator
- HS High Speed Crystal/Resonator
- RC Resistor/Capacitor

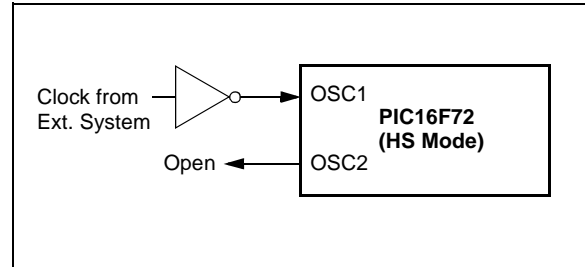
### 11.2.2 CRYSTAL OSCILLATOR/CERAMIC RESONATORS

In XT, LP or HS modes, a crystal or ceramic resonator is connected to the OSC1/CLKI and OSC2/CLKO pins to establish oscillation (Figure 11-1). The PIC16F72 oscillator design requires 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 mode, the device can accept an external clock source to drive the OSC1/CLKI pin (Figure 11-2). See Figure 14-1 or Figure 14-2 (depending on the part number and VDD range) for valid external clock frequencies.

**FIGURE 11-1: CRYSTAL/CERAMIC RESONATOR OPERATION (HS, XT OR LP OSC CONFIGURATION)**



**FIGURE 11-2: EXTERNAL CLOCK INPUT OPERATION (HS OSC CONFIGURATION)**



**TABLE 11-1: CERAMIC RESONATORS (FOR DESIGN GUIDANCE ONLY)**

Typical Capacitor Values Used:			
Mode	Freq	OSC1	OSC2
XT	455 kHz	56 pF	56 pF
	2.0 MHz	47 pF	47 pF
	4.0 MHz	33 pF	33 pF
HS	8.0 MHz	27 pF	27 pF
	16.0 MHz	22 pF	22 pF

**Capacitor values are for design guidance only.**

These capacitors were tested with the resonators listed below for basic start-up and operation. These values were not optimized.

Different capacitor values may be required to produce acceptable oscillator operation. The user should test the performance of the oscillator over the expected VDD and temperature range for the application.

See the notes at the bottom of page 62 for additional information.

# PIC16F72

**TABLE 11-2: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR (FOR DESIGN GUIDANCE ONLY)**

Osc Type	Crystal Freq	Typical Capacitor Values Tested:	
		C1	C2
LP	32 kHz	33 pF	33 pF
	200 kHz	15 pF	15 pF
XT	200 kHz	56 pF	56 pF
	1 MHz	15 pF	15 pF
	4 MHz	15 pF	15 pF
HS	4 MHz	15 pF	15 pF
	8 MHz	15 pF	15 pF
	20 MHz	15 pF	15 pF

**Capacitor values are for design guidance only.**

These capacitors were tested with the crystals listed below for basic start-up and operation. These values were not optimized.

Different capacitor values may be required to produce acceptable oscillator operation. The user should test the performance of the oscillator over the expected VDD and temperature range for the application.

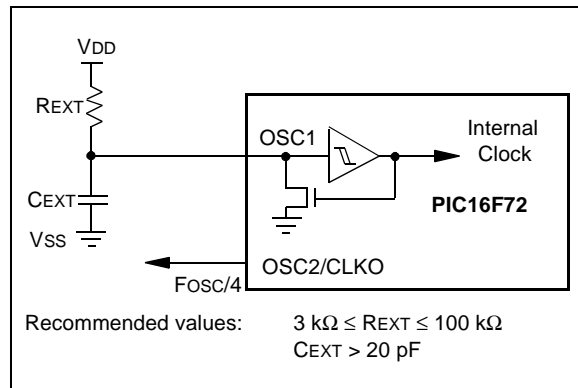
See the notes following this table for additional information.

- Note 1:** Higher capacitance increases the stability of oscillator, but also increases the start-up time.
- Since each resonator/crystal has its own characteristics, the user should consult the resonator/crystal manufacturer for appropriate values of external components.
  - Rs may be required in HS mode, as well as XT mode, to avoid overdriving crystals with low drive level specification.
  - Always verify oscillator performance over the VDD and temperature range that is expected for the application.

## 11.2.3 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 (REXT) and capacitor (CEXT) 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 CEXT values. The user also needs to take into account variation due to tolerance of external R and C components used. Figure 11-3 shows how the R/C combination is connected to the PIC16F72.

**FIGURE 11-3: RC OSCILLATOR MODE**



## 11.3 RESET

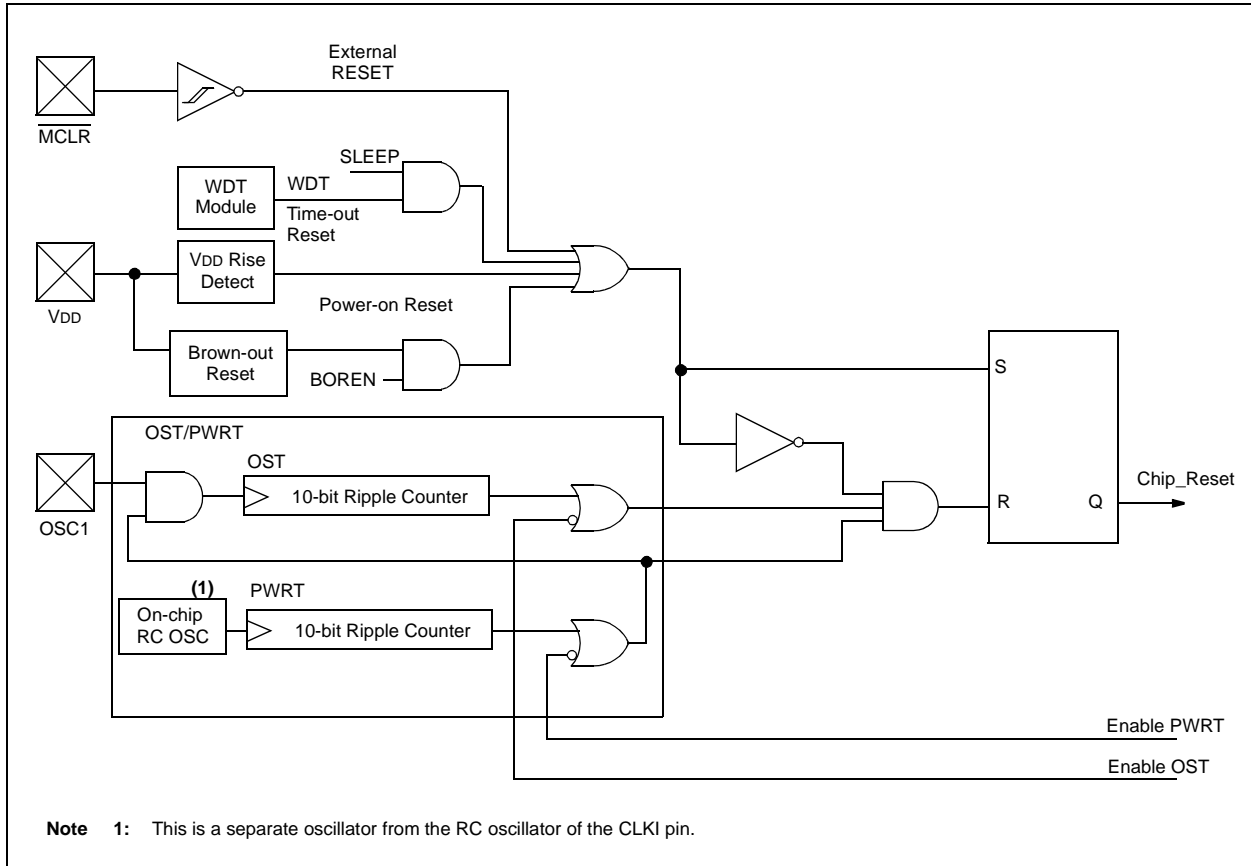
The PIC16F72 differentiates between various kinds of RESET:

- Power-on Reset (POR)
- $\overline{\text{MCLR}}$  Reset during normal operation
- $\overline{\text{MCLR}}$  Reset during SLEEP
- WDT Reset (during normal operation)
- WDT Wake-up (during SLEEP)
- Brown-out Reset (BOR)

Some registers are not affected in any RESET condition. Their status is unknown on POR and unchanged in any other RESET. Most other registers are reset to a “RESET state” on Power-on Reset (POR), on the  $\overline{\text{MCLR}}$  and WDT Reset, on  $\overline{\text{MCLR}}$  Reset during SLEEP, and Brown-out Reset (BOR). They are not affected by a WDT Wake-up, which is viewed as the resumption of normal operation. The  $\overline{\text{TO}}$  and  $\overline{\text{PD}}$  bits are set or cleared differently in different RESET situations, as indicated in Table 11-4. These bits are used in software to determine the nature of the RESET. See Table 11-6 for a full description of RESET states of all registers.

A simplified block diagram of the on-chip RESET circuit is shown in Figure 11-4.

**FIGURE 11-4: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT**



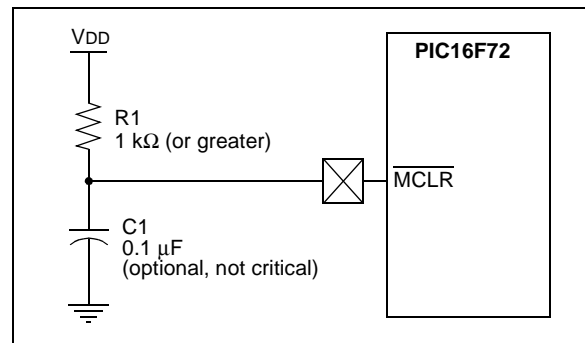
## 11.4 MCLR

PIC16F72 device has a noise filter in the  $\overline{\text{MCLR}}$  Reset path. The filter will detect and ignore small pulses.

It should be noted that a WDT Reset does not drive  $\overline{\text{MCLR}}$  pin low.

The behavior of the ESD protection on the  $\overline{\text{MCLR}}$  pin has been altered from previous devices of this family. Voltages applied to the pin that exceed its specification can result in both  $\overline{\text{MCLR}}$  and excessive current beyond the device specification during the ESD event. For this reason, Microchip recommends that the  $\overline{\text{MCLR}}$  pin no longer be tied directly to VDD. The use of an RC network, as shown in Figure 11-5, is suggested.

**FIGURE 11-5: RECOMMENDED MCLR CIRCUIT**



# PIC16F72

---

## 11.5 Power-on Reset (POR)

A Power-on Reset pulse is generated on-chip when VDD rise is detected (in the range of 1.2V - 1.7V). To take advantage of the POR, tie the  $\overline{\text{MCLR}}$  pin to VDD, as described in Section 11.4. A maximum rise time for VDD is specified. See Section 14.0, Electrical Characteristics for details.

When the device starts normal operation (exits 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 conditions are met. For more information, see Application Note, AN607- Power-up Trouble Shooting (DS00607).

## 11.6 Power-up Timer (PWRT)

The Power-up Timer provides a fixed 72 ms nominal time-out on power-up only from the POR. The Power-up Timer operates on an internal RC oscillator. The chip is kept in RESET as long as the PWRT is active. The PWRT's time delay allows VDD to rise to an acceptable level. A configuration bit is provided to enable/disable the PWRT.

The power-up time delay will vary from chip to chip due to VDD, temperature and process variation. See DC parameters for details (TPWRT, parameter #33).

## 11.7 Oscillator Start-up Timer (OST)

The Oscillator Start-up Timer (OST) provides 1024 oscillator cycles (from OSC1 input) delay after the PWRT delay is over (if enabled). This helps to ensure that the crystal oscillator or resonator has started and stabilized.

The OST time-out is invoked only for XT, LP and HS modes and only on Power-on Reset or wake-up from SLEEP.

## 11.8 Brown-out Reset (BOR)

The configuration bit, BOREN, can enable or disable the Brown-out Reset circuit. If VDD falls below VBOR (parameter D005, about 4V) for longer than TBOR (parameter #35, about 100  $\mu\text{s}$ ), the brown-out situation will reset the device. If VDD falls below VBOR for less than TBOR, a RESET may not occur.

Once the brown-out occurs, the device will remain in Brown-out Reset until VDD rises above VBOR. The Power-up Timer then keeps the device in RESET for TPWRT (parameter #33, about 72 ms). If VDD should fall below VBOR during TPWRT, the Brown-out Reset process will restart when VDD rises above VBOR, with the Power-up Timer Reset. The Power-up Timer is always enabled when the Brown-out Reset circuit is enabled, regardless of the state of the PWRT configuration bit.

## 11.9 Time-out Sequence

On power-up, the time-out sequence is as follows: the PWRT delay starts (if enabled) when a POR occurs. Then, OST starts counting 1024 oscillator cycles when PWRT ends (LP, XT, HS). When the OST ends, the device comes out of RESET.

If  $\overline{\text{MCLR}}$  is kept low long enough, all delays will expire. Bringing  $\overline{\text{MCLR}}$  high will begin execution immediately. This is useful for testing purposes or to synchronize more than one PIC16F72 device operating in parallel.

Table 11-5 shows the RESET conditions for the STATUS, PCON and PC registers, while Table 11-6 shows the RESET conditions for all the registers.

## 11.10 Power Control/Status Register (PCON)

The Power Control/Status Register, PCON, has two bits to indicate the type of RESET that last occurred.

Bit0 is Brown-out Reset Status bit,  $\overline{\text{BOR}}$ . Bit  $\overline{\text{BOR}}$  is unknown on a Power-on Reset. It must then be set by the user and checked on subsequent RESETS to see if bit  $\overline{\text{BOR}}$  cleared, indicating a Brown-out Reset occurred. When the Brown-out Reset is disabled, the state of the  $\overline{\text{BOR}}$  bit is unpredictable.

Bit1 is  $\overline{\text{POR}}$  (Power-on Reset Status bit). It is cleared on a Power-on Reset and unaffected otherwise. The user must set this bit following a Power-on Reset.



**TABLE 11-3: TIME-OUT IN VARIOUS SITUATIONS**

Oscillator Configuration	Power-up		Brown-out	Wake-up from SLEEP
	$\overline{\text{PWRTE}} = 0$	$\overline{\text{PWRTE}} = 1$		
XT, HS, LP	72 ms + 1024 TOSC	1024 TOSC	72 ms + 1024 TOSC	1024 TOSC
RC	72 ms	—	72 ms	—

**TABLE 11-4: STATUS BITS AND THEIR SIGNIFICANCE**

$\overline{\text{POR}}$ (PCON<1>)	$\overline{\text{BOR}}$ (PCON<0>)	$\overline{\text{TO}}$ (STATUS<4>)	$\overline{\text{PD}}$ (STATUS<3>)	Significance
0	x	1	1	Power-on Reset
0	x	0	x	Illegal, $\overline{\text{TO}}$ is set on $\overline{\text{POR}}$
0	x	x	0	Illegal, $\overline{\text{PD}}$ is set on $\overline{\text{POR}}$
u	0	1	1	Brown-out Reset
u	u	0	1	WDT Reset
u	u	0	0	WDT Wake-up
u	u	u	u	$\overline{\text{MCLR}}$ Reset during normal operation
u	u	1	0	$\overline{\text{MCLR}}$ Reset during SLEEP or interrupt wake-up from SLEEP

**TABLE 11-5: RESET CONDITION FOR SPECIAL REGISTERS**

Condition	Program Counter	STATUS Register	PCON Register
Power-on Reset	000h	0001 1xxx	---- --0x
$\overline{\text{MCLR}}$ Reset during normal operation	000h	000u uuuu	---- --uu
$\overline{\text{MCLR}}$ Reset during SLEEP	000h	0001 0uuu	---- --uu
WDT Reset	000h	0000 1uuu	---- --uu
WDT Wake-up	PC + 1	uuu0 0uuu	---- --uu
Brown-out Reset	000h	0001 1uuu	---- --u0
Interrupt Wake-up from SLEEP	PC + 1 <sup>(1)</sup>	uuu1 0uuu	---- --uu

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

**Note 1:** When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).

# PIC16F72

**TABLE 11-6: INITIALIZATION CONDITIONS FOR ALL REGISTERS**

Register	Power-on Reset, Brown-out Reset	MCLR Reset, WDT Reset	Wake-up via WDT or Interrupt
W	xxxx xxxx	uuuu uuuu	uuuu uuuu
INDF	N/A	N/A	N/A
TMR0	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCL	0000h	0000h	PC + 1 <sup>(2)</sup>
STATUS	0001 1xxx	000q quuu <sup>(3)</sup>	uuuq quuu <sup>(3)</sup>
FSR	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTA	--0x 0000	--0u 0000	--uu uuuu
PORTB	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTC	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCLATH	---0 0000	---0 0000	---u uuuu
INTCON	0000 000x	0000 000u	uuuu uuuu <sup>(1)</sup>
PIR1	-0-- 0000	-0-- 0000	-u-- uuuu <sup>(1)</sup>
TMR1L	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR1H	xxxx xxxx	uuuu uuuu	uuuu uuuu
T1CON	--00 0000	--uu uuuu	--uu uuuu
TMR2	0000 0000	0000 0000	uuuu uuuu
T2CON	-000 0000	-000 0000	-uuu uuuu
SSPBUF	xxxx xxxx	uuuu uuuu	uuuu uuuu
SSPCON	0000 0000	0000 0000	uuuu uuuu
CCPR1L	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR1H	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP1CON	--00 0000	--00 0000	--uu uuuu
ADRES	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADCON0	0000 00-0	0000 00-0	uuuu uu-u
OPTION	1111 1111	1111 1111	uuuu uuuu
TRISA	--11 1111	--11 1111	--uu uuuu
TRISB	1111 1111	1111 1111	uuuu uuuu
TRISC	1111 1111	1111 1111	uuuu uuuu
PIE1	-0-- 0000	-0-- 0000	-u-- uuuu
PCON	---- -qqr	---- --uu	---- --uu
PR2	1111 1111	1111 1111	1111 1111
SSPADD	0000 0000	0000 0000	uuuu uuuu
SSPSTAT	--00 0000	--00 0000	--uu uuuu
ADCON1	---- -000	---- -000	---- -uuu
PMDATL	0--- 0000	0--- 0000	u--- uuuu
PMADRL	xxxx xxxx	uuuu uuuu	uuuu uuuu
PMDATH	xxxx xxxx	uuuu uuuu	uuuu uuuu
PMADRH	xxxx xxxx	uuuu uuuu	uuuu uuuu
PMCON1	1--- ---0	1--- ---0	1--- ---u

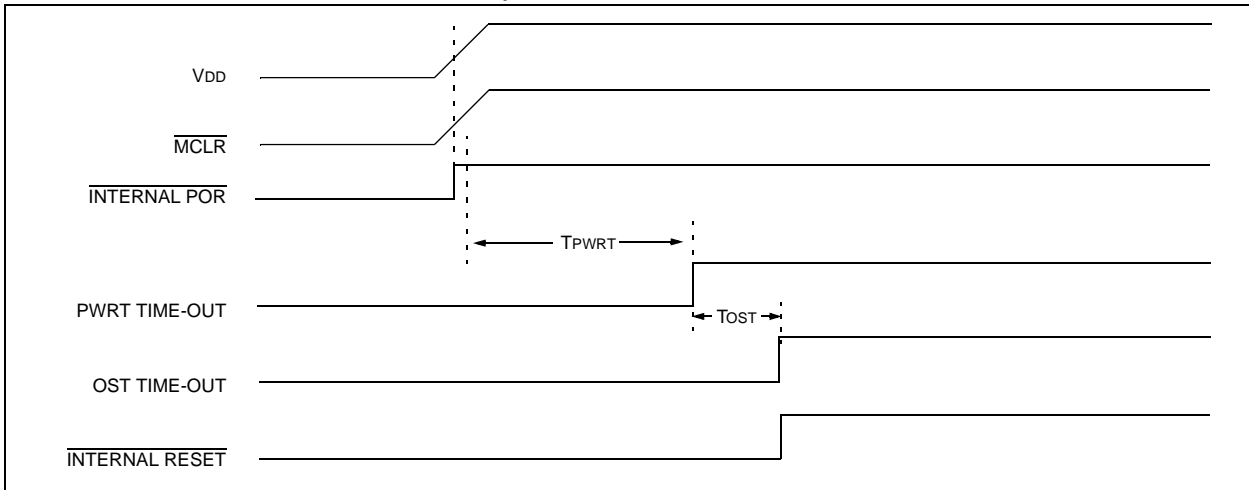
Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition, r = reserved, maintain clear.

**Note 1:** One or more bits in INTCON, PIR1 will be affected (to cause wake-up).

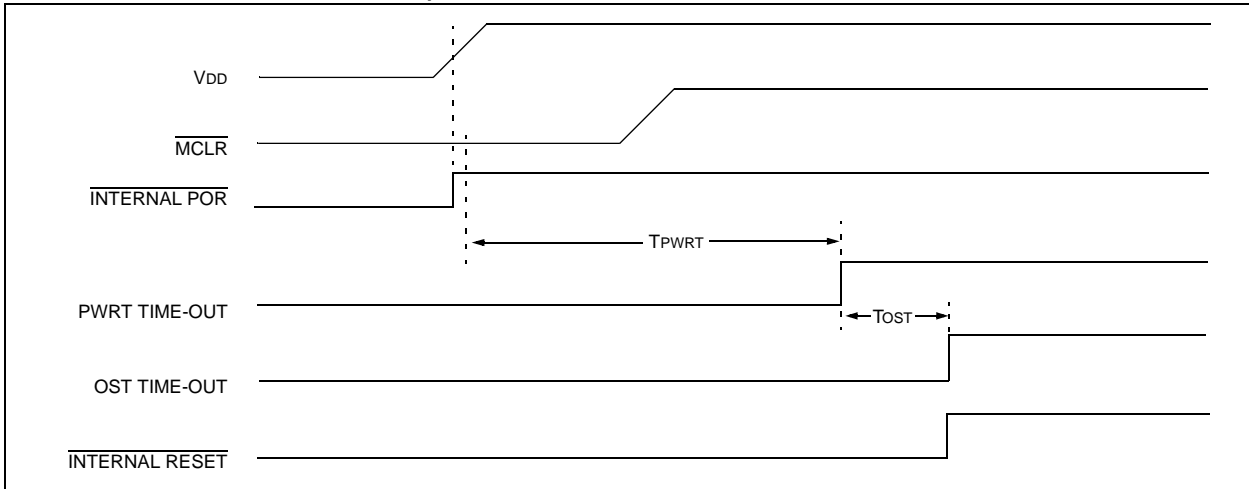
**Note 2:** When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).

**Note 3:** See Table 11-5 for RESET value for specific condition.

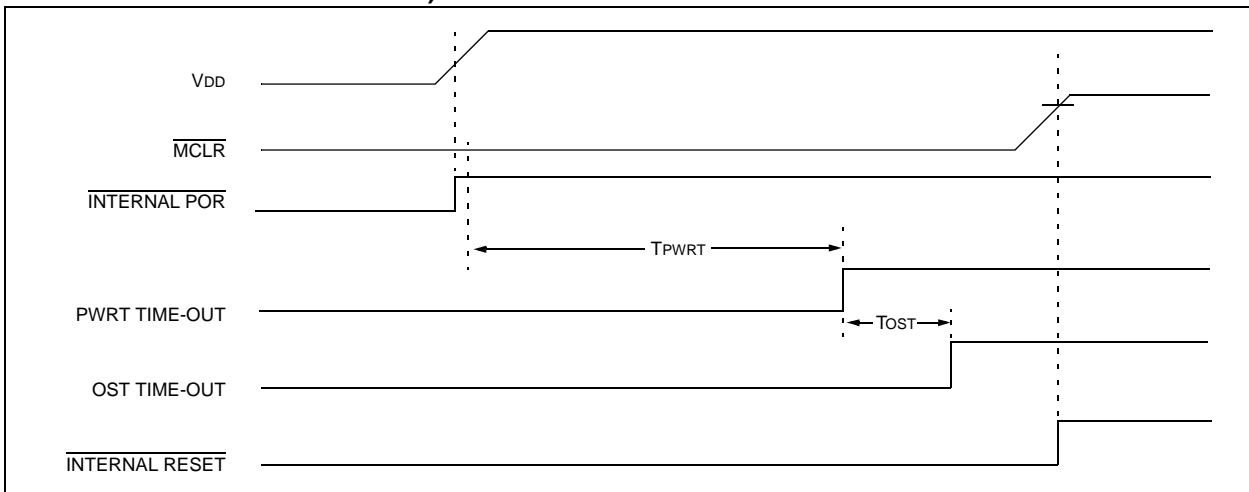
**FIGURE 11-6: TIME-OUT SEQUENCE ON POWER-UP ( $\overline{\text{MCLR}}$  TIED TO  $V_{DD}$  THROUGH PULL-UP RESISTOR)**



**FIGURE 11-7: TIME-OUT SEQUENCE ON POWER-UP ( $\overline{\text{MCLR}}$  TIED TO  $V_{DD}$  THROUGH RC NETWORK): CASE 1**

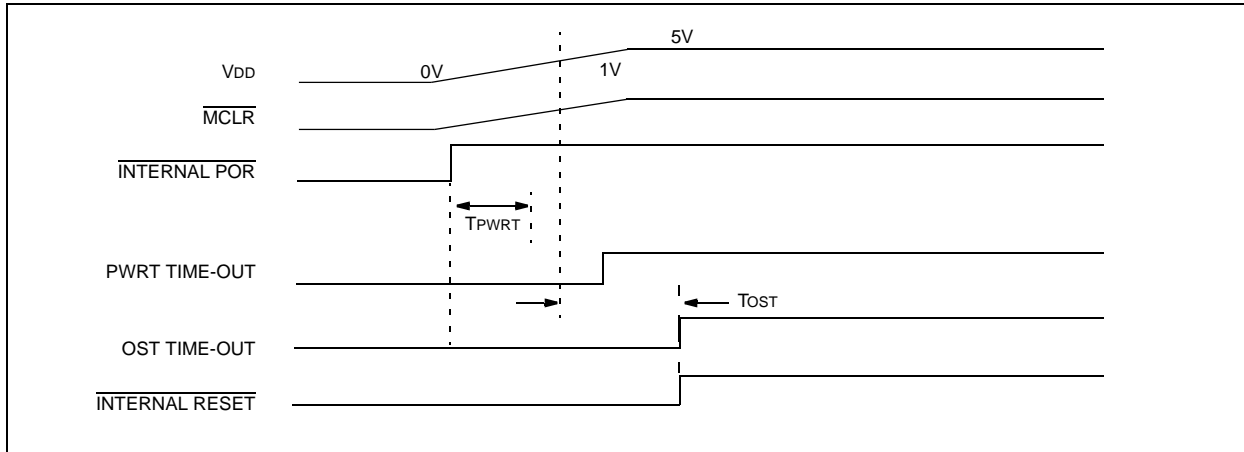


**FIGURE 11-8: TIME-OUT SEQUENCE ON POWER-UP ( $\overline{\text{MCLR}}$  TIED TO  $V_{DD}$  THROUGH RC NETWORK): CASE 2**



# PIC16F72

**FIGURE 11-9: SLOW RISE TIME ( $\overline{\text{MCLR}}$  TIED TO  $V_{DD}$  THROUGH RC NETWORK)**



## 11.11 Interrupts

The PIC16F72 has up to eight sources of interrupt. The interrupt control register (INTCON) records individual interrupt requests in flag bits. It also has individual and global interrupt enable bits.

**Note:** Individual interrupt flag bits are set, regardless of the status of their corresponding mask bit, or the GIE bit.

A global interrupt enable bit, GIE (INTCON<7>) enables (if set) all unmasked interrupts, or disables (if cleared) all interrupts. When bit GIE is enabled, and an interrupt's flag bit and mask bit are set, the interrupt will vector immediately. Individual interrupts can be disabled through their corresponding enable bits in various registers. Individual interrupt bits are set, regardless of the status of the GIE bit. The GIE bit is cleared on RESET.

The "return from interrupt" instruction, RETFIE, exits the interrupt routine, as well as sets the GIE bit, which re-enables interrupts.

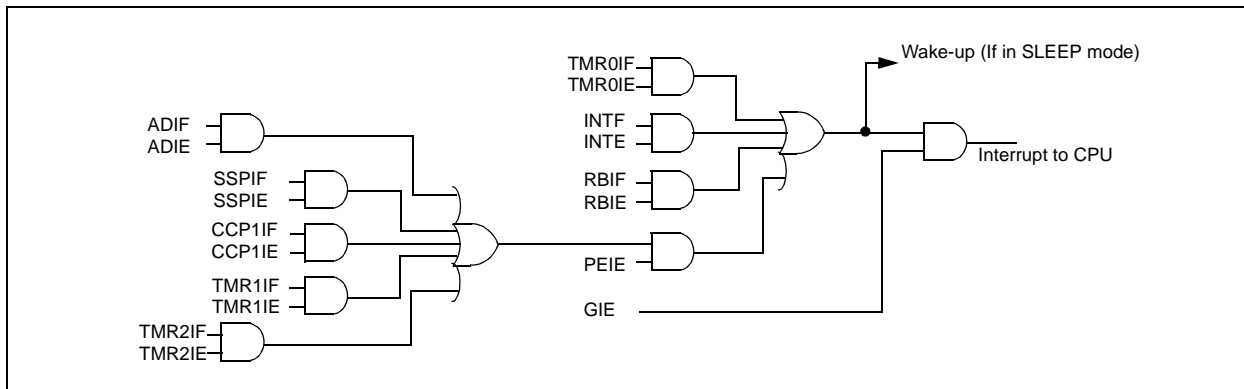
The RB0/INT pin interrupt, the RB port change interrupt and the TMR0 overflow interrupt flags are contained in the INTCON register.

The peripheral interrupt flags are contained in the Special Function Register, PIR1. The corresponding interrupt enable bits are contained in Special Function Register, PIE1, and the peripheral interrupt enable bit is contained in Special Function Register INTCON.

When an interrupt is serviced, the GIE bit is cleared to disable any further interrupt, the return address is pushed onto the stack, and the PC is loaded with 0004h. Once in the Interrupt Service Routine, the source(s) of the interrupt can be determined by polling the interrupt flag bits. The interrupt flag bit(s) must be cleared in software before re-enabling interrupts to avoid recursive interrupts.

For external interrupt events, such as the INT pin or PORTB change interrupt, the interrupt latency will be three or four instruction cycles. The exact latency depends when the interrupt event occurs, relative to the current Q cycle. The latency is the same for one or two cycle instructions. Individual interrupt flag bits are set, regardless of the status of their corresponding mask bit, PEIE bit, or the GIE bit.

**FIGURE 11-10: INTERRUPT LOGIC**



## 11.11.1 INT INTERRUPT

External interrupt on the RB0/INT pin is edge triggered, either rising, if bit INTEDG (OPTION<6>) is set, or falling, if the INTEDG bit is clear. When a valid edge appears on the RB0/INT pin, flag bit INTF (INTCON<1>) is set. This interrupt can be disabled by clearing enable bit INTE (INTCON<4>). Flag bit INTF must be cleared in software in the Interrupt Service Routine before re-enabling this interrupt. The INT interrupt can wake-up the processor from SLEEP, if bit INTE was set prior to going into SLEEP. The status of global interrupt enable bit GIE decides whether or not the processor branches to the interrupt vector following wake-up. See Section 11.14 for details on SLEEP mode.

## 11.11.2 TMR0 INTERRUPT

An overflow (FFh → 00h) in the TMR0 register will set flag bit TMR0IF (INTCON<2>). The interrupt can be enabled/disabled by setting/clearing enable bit TMR0IE (INTCON<5>) (see Section 5.0).

## 11.11.3 PORTB INTCON CHANGE

An input change on PORTB<7:4> sets flag bit RBIF (INTCON<0>). The interrupt can be enabled/disabled by setting/clearing enable bit RBIE (INTCON<4>) (see Section 3.2).

## 11.12 Context Saving During Interrupts

During an interrupt, only the return PC value is saved on the stack. Typically, users may wish to save key registers during an interrupt (i.e., W, STATUS registers). This will have to be implemented in software, as shown in Example 11-1.

For the PIC16F72 device, the register W\_TEMP must be defined in both banks 0 and 1 and must be defined at the same offset from the bank base address (i.e., if W\_TEMP is defined at 20h in bank 0, it must also be defined at A0h in bank 1). The register STATUS\_TEMP is only defined in bank 0.

### EXAMPLE 11-1: SAVING STATUS, W AND PCLATH REGISTERS IN RAM

```

MOVWF  W_TEMP           ;Copy W to TEMP register
SWAPF  STATUS,W        ;Swap status to be saved into W
CLRF   STATUS           ;bank 0, regardless of current bank, Clears IRP,RP1,RP0
MOVWF  STATUS_TEMP     ;Save status to bank zero STATUS_TEMP register
:
:(ISR)                  ;Insert user code here
:
SWAPF  STATUS_TEMP,W   ;Swap STATUS_TEMP register into W
                        ;(sets bank to original state)
MOVWF  STATUS          ;Move W into STATUS register
SWAPF  W_TEMP,F        ;Swap W_TEMP
SWAPF  W_TEMP,W        ;Swap W_TEMP into W
    
```

# PIC16F72

## 11.13 Watchdog Timer (WDT)

The Watchdog Timer is a free running, on-chip RC oscillator that does not require any external components. This RC oscillator is separate from the RC oscillator of the OSC1/CLKI pin. That means that the WDT will run, even if the clock on the OSC1/CLKI and OSC2/CLKO pins of the device has been stopped, for example, by execution of a SLEEP instruction.

During normal operation, a WDT time-out generates a device RESET (Watchdog Timer Reset). If the device is in SLEEP mode, a WDT time-out causes the device to wake-up and continue with normal operation (Watchdog Timer Wake-up). The  $\overline{TO}$  bit in the STATUS register will be cleared upon a Watchdog Timer time-out.

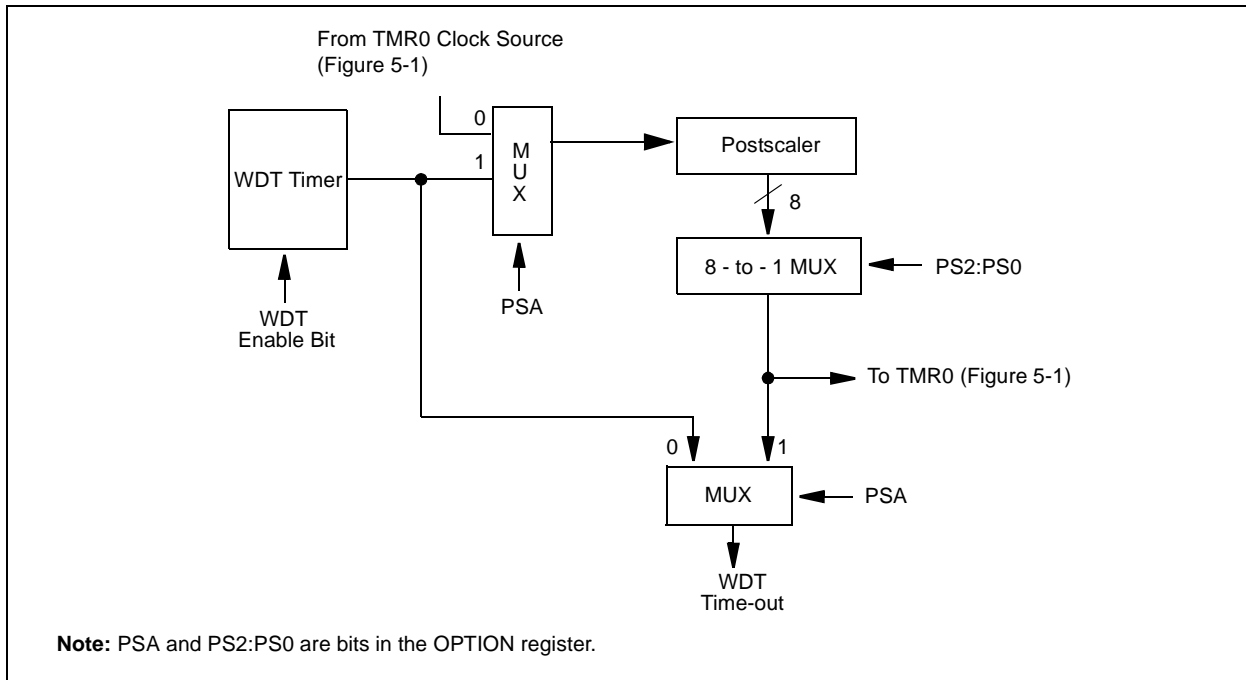
The WDT can be permanently disabled by clearing configuration bit WDTEN (see Section 11.1).

WDT time-out period values may be found in the Electrical Specifications section under parameter #31. Values for the WDT prescaler (actually a postscaler, but shared with the Timer0 prescaler) may be assigned using the OPTION register.

**Note 1:** The CLRWDT and SLEEP instructions clear the WDT and the postscaler, if assigned to the WDT, and prevent it from timing out and generating a device RESET condition.

**2:** When a CLRWDT instruction is executed and the prescaler is assigned to the WDT, the prescaler count will be cleared, but the prescaler assignment is not changed.

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



**TABLE 11-7: SUMMARY OF WATCHDOG TIMER REGISTERS**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
2007h	Config. bits	(1)	BOREN <sup>(1)</sup>	—	CP	$\overline{PWRRTEN}$ <sup>(1)</sup>	WDTEN	FOSC1	FOSC0
81h,181h	OPTION	$\overline{RBPU}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0

Legend: Shaded cells are not used by the Watchdog Timer.

**Note 1:** See Register 11-1 for operation of these bits.

## 11.14 Power-down Mode (SLEEP)

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

If enabled, the Watchdog Timer will be cleared but keeps running, the  $\overline{PD}$  bit (`STATUS<3>`) is cleared, the  $\overline{TO}$  (`STATUS<4>`) bit is set, and the oscillator driver is turned off. The I/O ports maintain the status they had before the `SLEEP` instruction was executed (driving high, low, or hi-impedance).

For lowest current consumption in this mode, place all I/O pins at either  $V_{DD}$  or  $V_{SS}$ , ensure no external circuitry is drawing current from the I/O pin, power-down the A/D and disable external clocks. Pull all I/O pins that are hi-impedance inputs, high or low externally, to avoid switching currents caused by floating inputs. The `TOCKI` input should also be at  $V_{DD}$  or  $V_{SS}$  for lowest current consumption. The contribution from on-chip pull-ups on `PORTB` should also be considered.

The  $\overline{MCLR}$  pin must be at a logic high level ( $V_{IHMC}$ ).

### 11.14.1 WAKE-UP FROM SLEEP

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

1. External `RESET` input on  $\overline{MCLR}$  pin.
2. Watchdog Timer wake-up (if `WDT` was enabled).
3. Interrupt from `INT` pin, `RB` port change or a peripheral interrupt.

External  $\overline{MCLR}$  Reset will cause a device `RESET`. All other events are considered a continuation of program execution and cause a "wake-up". The  $\overline{TO}$  and  $\overline{PD}$  bits in the `STATUS` register can be used to determine the cause of the device `RESET`. The  $\overline{PD}$  bit, which is set on power-up, is cleared when `SLEEP` is invoked. The  $\overline{TO}$  bit is cleared if a `WDT` time-out occurred and caused wake-up.

The following peripheral interrupts can wake the device from `SLEEP`:

1. `TMR1` interrupt. `Timer1` must be operating as an asynchronous counter.
2. `CCP` Capture mode interrupt.
3. Special event trigger (`Timer1` in Asynchronous mode using an external clock).
4. `SSP` (`START/STOP`) bit detect interrupt.
5. `SSP` transmit or receive in Slave mode (`SPI/I2C`).
6. A/D conversion (when A/D clock source is `RC`).

Other peripherals cannot generate interrupts since during `SLEEP`, no on-chip clocks are present.

When the `SLEEP` instruction is being executed, the next instruction (`PC + 1`) is pre-fetched. For the device to wake-up through an interrupt event, the corresponding interrupt enable bit must be set (enabled). Wake-up occurs regardless of the state of the `GIE` bit. If the `GIE` bit is clear (disabled), the device continues execution at the instruction after the `SLEEP` instruction. If the `GIE` bit is set (enabled), the device executes the instruction after the `SLEEP` instruction and then branches to the interrupt address (`0004h`). In cases where the execution of the instruction following `SLEEP` is not desirable, the user should have a `NOP` after the `SLEEP` instruction.

### 11.14.2 WAKE-UP USING INTERRUPTS

When global interrupts are disabled (`GIE` cleared) and any interrupt source has both its interrupt enable bit and interrupt flag bit set, one of the following will occur:

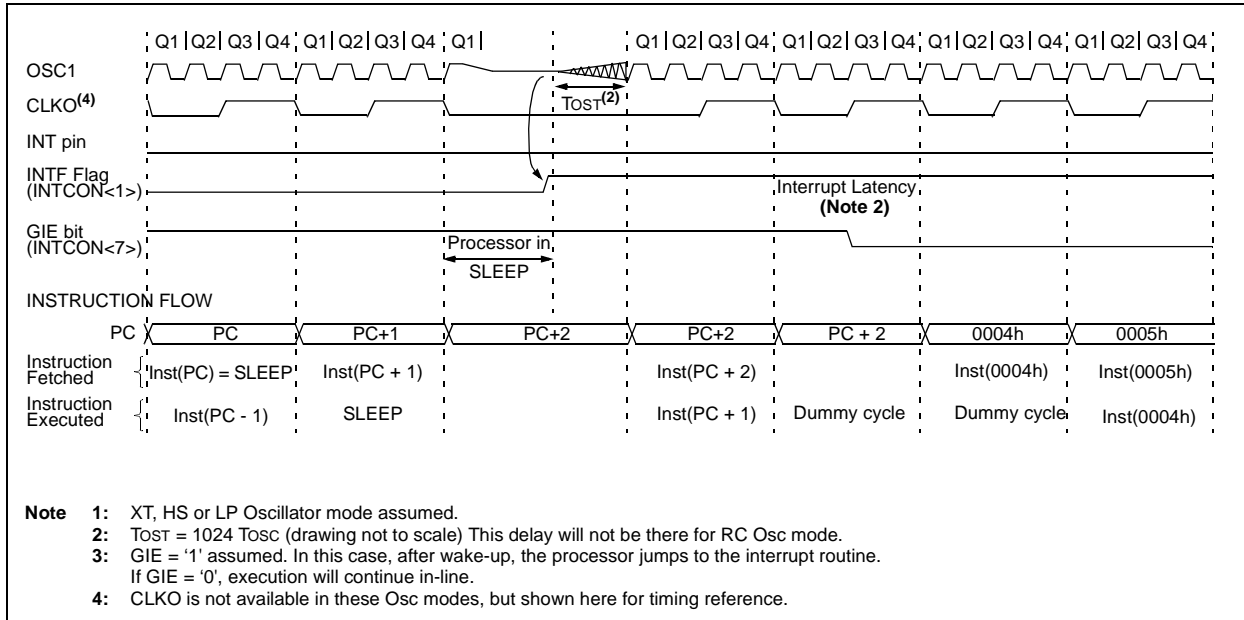
- If the interrupt occurs **before** the execution of a `SLEEP` instruction, the `SLEEP` instruction will complete as a `NOP`. Therefore, the `WDT` and `WDT` postscaler will not be cleared, the  $\overline{TO}$  bit will not be set and  $\overline{PD}$  bits will not be cleared.
- If the interrupt occurs **during or after** the execution of a `SLEEP` instruction, the device will immediately wake-up from `SLEEP`. The `SLEEP` instruction will be completely executed before the wake-up. Therefore, the `WDT` and `WDT` postscaler will be cleared, the  $\overline{TO}$  bit will be set and the  $\overline{PD}$  bit will be cleared.

Even if the flag bits were checked before executing a `SLEEP` instruction, it may be possible for flag bits to become set before the `SLEEP` instruction completes. To determine whether a `SLEEP` instruction executed, test the  $\overline{PD}$  bit. If the  $\overline{PD}$  bit is set, the `SLEEP` instruction was executed as a `NOP`.

To ensure that the `WDT` is cleared, a `CLRWDT` instruction should be executed before a `SLEEP` instruction.

# PIC16F72

**FIGURE 11-12: WAKE-UP FROM SLEEP THROUGH INTERRUPT**



## 11.15 Program Verification/ Code Protection

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

## 11.16 ID Locations

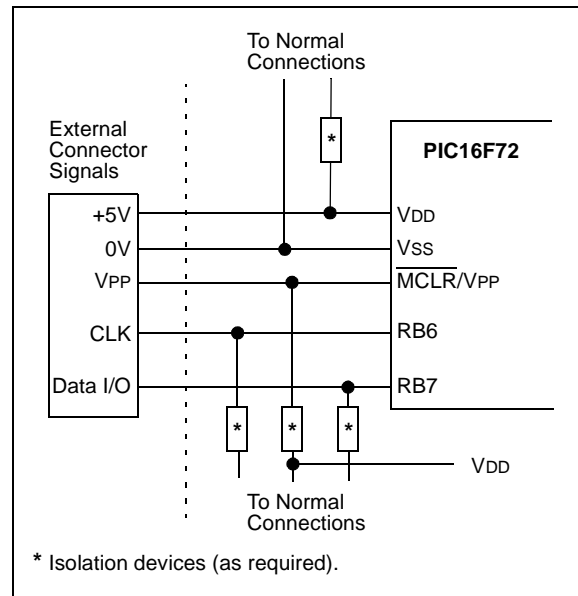
Four memory locations (2000h - 2003h) are designated as ID locations, where the user can store checksum or other code identification numbers. These locations are not accessible during normal execution, but are readable and writable during program/verify. It is recommended that only the four Least Significant bits of the ID location are used.

## 11.17 In-Circuit Serial Programming

PIC16F72 microcontrollers can be serially programmed while in the end application circuit. This is simply done with two lines for clock and data and three other lines for power, ground, and the programming voltage (see Figure 11-13 for an example). 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.

For general information of serial programming, please refer to the In-Circuit Serial Programming™ (ICSP™) Guide (DS30277). For specific details on programming commands and operations for the PIC16F72 devices, please refer to the latest version of the PIC16F72 FLASH Program Memory Programming Specification (DS39588).

**FIGURE 11-13: TYPICAL IN-CIRCUIT SERIAL PROGRAMMING CONNECTION**





## 12.0 INSTRUCTION SET SUMMARY

Each PIC16F72 instruction is a 14-bit word divided into an OPCODE that specifies the instruction type and one or more operands that further specify the operation of the instruction. The PIC16F72 instruction set summary in Table 12-2 lists **byte-oriented**, **bit-oriented**, and **literal and control** operations. Table 12-1 shows the opcode field descriptions.

For **byte-oriented** instructions, 'f' represents a file register designator and 'd' represents a destination designator. The file register designator 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 zero, the result is placed in the W register. If 'd' is one, 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 eight or eleven-bit constant or literal value.

**TABLE 12-1: OPCODE FIELD DESCRIPTIONS**

Field	Description
f	Register file address (0x00 to 0x7F)
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.
PC	Program Counter
TO	Time-out bit
PD	Power-down bit

The instruction set is highly orthogonal and is grouped into three basic categories:

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

All instructions are executed within one single instruction cycle, unless a conditional test is true or the program counter is changed as a result of an instruction. In this case, the execution takes two instruction cycles, with the second cycle executed as a **NOE**. 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.

Table 12-2 lists the instructions recognized by the MPASM™ assembler.

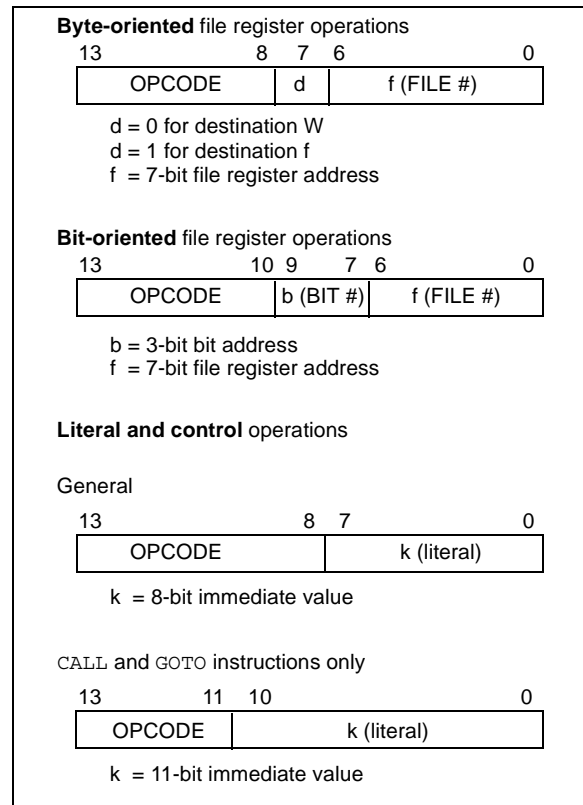
Figure 12-1 shows the general formats that the instructions can have.

All examples use the following format to represent a hexadecimal number:

0xhh

where h signifies a hexadecimal digit.

**FIGURE 12-1: GENERAL FORMAT FOR INSTRUCTIONS**



A description of each instruction is available in the PICmicro™ Mid-Range MCU Family Reference Manual (DS33023).

# PIC16F72

**TABLE 12-2: PIC16F72 INSTRUCTION SET**

Mnemonic, Operands	Description	Cycles	14-Bit Opcode		Status Affected	Notes	
			MSb	LSb			
<b>BYTE-ORIENTED FILE REGISTER OPERATIONS</b>							
ADDWF	f, d	Add W and f	1	00 0111	dfff ffff	C,DC,Z	1,2
ANDWF	f, d	AND W with f	1	00 0101	dfff ffff	Z	1,2
CLRF	f	Clear f	1	00 0001	1fff ffff	Z	2
CLRWF	-	Clear W	1	00 0001	0xxx xxxx	Z	
COMF	f, d	Complement f	1	00 1001	dfff ffff	Z	1,2
DECF	f, d	Decrement f	1	00 0011	dfff ffff	Z	1,2
DECFSZ	f, d	Decrement f, Skip if 0	1(2)	00 1011	dfff ffff		1,2,3
INCF	f, d	Increment f	1	00 1010	dfff ffff	Z	1,2
INCFSZ	f, d	Increment f, Skip if 0	1(2)	00 1111	dfff ffff		1,2,3
IORWF	f, d	Inclusive OR W with f	1	00 0100	dfff ffff	Z	1,2
MOVF	f, d	Move f	1	00 1000	dfff ffff	Z	1,2
MOVWF	f	Move W to f	1	00 0000	1fff ffff		
NOP	-	No Operation	1	00 0000	0xx0 0000		
RLF	f, d	Rotate Left f through Carry	1	00 1101	dfff ffff	C	1,2
RRF	f, d	Rotate Right f through Carry	1	00 1100	dfff ffff	C	1,2
SUBWF	f, d	Subtract W from f	1	00 0010	dfff ffff	C,DC,Z	1,2
SWAPF	f, d	Swap nibbles in f	1	00 1110	dfff ffff		1,2
XORWF	f, d	Exclusive OR W with f	1	00 0110	dfff ffff	Z	1,2
<b>BIT-ORIENTED FILE REGISTER OPERATIONS</b>							
BCF	f, b	Bit Clear f	1	01 00bb	bfff ffff		1,2
BSF	f, b	Bit Set f	1	01 01bb	bfff ffff		1,2
BTFSC	f, b	Bit Test f, Skip if Clear	1 (2)	01 10bb	bfff ffff		3
BTFSS	f, b	Bit Test f, Skip if Set	1 (2)	01 11bb	bfff ffff		3
<b>LITERAL AND CONTROL OPERATIONS</b>							
ADDLW	k	Add literal and W	1	11 111x	kkkk kkkk	C,DC,Z	
ANDLW	k	AND literal with W	1	11 1001	kkkk kkkk	Z	
CALL	k	Call subroutine	2	10 0kkk	kkkk kkkk		
CLRWDTC	-	Clear Watchdog Timer	1	00 0000	0110 0100	$\overline{TO,PD}$	
GOTO	k	Go to address	2	10 1kkk	kkkk kkkk		
IORLW	k	Inclusive OR literal with W	1	11 1000	kkkk kkkk	Z	
MOVLW	k	Move literal to W	1	11 00xx	kkkk kkkk		
RETFIE	-	Return from interrupt	2	00 0000	0000 1001		
RETLW	k	Return with literal in W	2	11 01xx	kkkk kkkk		
RETURN	-	Return from Subroutine	2	00 0000	0000 1000		
SLEEP	-	Go into Standby mode	1	00 0000	0110 0011	$\overline{TO,PD}$	
SUBLW	k	Subtract W from literal	1	11 110x	kkkk kkkk	C,DC,Z	
XORLW	k	Exclusive OR literal with W	1	11 1010	kkkk kkkk	Z	

**Note 1:** 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'.

**2:** If this instruction is executed on the TMR0 register (and where applicable, `d = 1`), the prescaler will be cleared if assigned to the Timer0 module.

**3:** If Program Counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a `NOP`.

**Note:** Additional information on the mid-range instruction set is available in the PICmicro™ Mid-Range MCU Family Reference Manual (DS33023).

## 12.1 Instruction Descriptions

### **ADDLW**      **Add Literal and W**

**Syntax:**             $[label] \text{ ADDLW } k$

**Operands:**         $0 \leq k \leq 255$

**Operation:**         $(W) + k \rightarrow (W)$

**Status Affected:** C, DC, Z

**Description:**      The contents of the W register are added to the eight-bit literal 'k' and the result is placed in the W register.

### **ANDWF**        **AND W with f**

**Syntax:**             $[label] \text{ ANDWF } f,d$

**Operands:**         $0 \leq f \leq 127$   
 $d \in [0,1]$

**Operation:**         $(W) \text{ .AND. } (f) \rightarrow (\text{destination})$

**Status Affected:** Z

**Description:**      AND the W register with register 'f'. If 'd' = '0', the result is stored in the W register. If 'd' = '1', the result is stored back in register 'f'.

### **ADDWF**        **Add W and f**

**Syntax:**             $[label] \text{ ADDWF } f,d$

**Operands:**         $0 \leq f \leq 127$   
 $d \in [0,1]$

**Operation:**         $(W) + (f) \rightarrow (\text{destination})$

**Status Affected:** C, DC, Z

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

### **BCF**            **Bit Clear f**

**Syntax:**             $[label] \text{ BCF } f,b$

**Operands:**         $0 \leq f \leq 127$   
 $0 \leq b \leq 7$

**Operation:**         $0 \rightarrow (f<b>)$

**Status Affected:** None

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

### **ANDLW**        **AND Literal with W**

**Syntax:**             $[label] \text{ ANDLW } k$

**Operands:**         $0 \leq k \leq 255$

**Operation:**         $(W) \text{ .AND. } (k) \rightarrow (W)$

**Status Affected:** Z

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

### **BSF**            **Bit Set f**

**Syntax:**             $[label] \text{ BSF } f,b$

**Operands:**         $0 \leq f \leq 127$   
 $0 \leq b \leq 7$

**Operation:**         $1 \rightarrow (f<b>)$

**Status Affected:** None

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

# PIC16F72

---

<b>BTFSS</b>	<b>Bit Test f, Skip if Set</b>
Syntax:	[ <i>label</i> ] BTFSS f,b
Operands:	$0 \leq f \leq 127$ $0 \leq b < 7$
Operation:	skip if (f<b>) = 1
Status Affected:	None
Description:	If bit 'b' in register 'f' = '0', the next instruction is executed. If bit 'b' = '1', then the next instruction is discarded and a NOP is executed instead, making this a 2 TCY instruction.

<b>CLRF</b>	<b>Clear f</b>
Syntax:	[ <i>label</i> ] CLRF f
Operands:	$0 \leq f \leq 127$
Operation:	00h → (f) 1 → Z
Status Affected:	Z
Description:	The contents of register 'f' are cleared and the Z bit is set.

<b>BTFSC</b>	<b>Bit Test, Skip if Clear</b>
Syntax:	[ <i>label</i> ] BTFSC f,b
Operands:	$0 \leq f \leq 127$ $0 \leq b \leq 7$
Operation:	skip if (f<b>) = 0
Status Affected:	None
Description:	If bit 'b' in register 'f' = '1', the next instruction is executed. If bit 'b' in register 'f' = '0', the next instruction is discarded, and a NOP is executed instead, making this a 2 TCY instruction.

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

<b>CALL</b>	<b>Call Subroutine</b>
Syntax:	[ <i>label</i> ] CALL k
Operands:	$0 \leq k \leq 2047$
Operation:	(PC) + 1 → TOS, k → PC<10:0>, (PCLATH<4:3>) → PC<12:11>
Status Affected:	None
Description:	Call Subroutine. First, return address (PC+1) is pushed onto the stack. The eleven-bit immediate address is loaded into PC bits <10:0>. The upper bits of the PC are loaded from PCLATH. CALL is a two-cycle instruction.

<b>CLRWDT</b>	<b>Clear Watchdog Timer</b>
Syntax:	[ <i>label</i> ] CLRWDT
Operands:	None
Operation:	00h → WDT 0 → WDT prescaler, 1 → $\overline{TO}$ 1 → $\overline{PD}$
Status Affected:	$\overline{TO}$ , $\overline{PD}$
Description:	CLRWDT instruction resets the Watchdog Timer. It also resets the prescaler of the WDT. Status bits $\overline{TO}$ and $\overline{PD}$ are set.

<b>COMF</b>	<b>Complement f</b>
Syntax:	[ <i>label</i> ] COMF f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	$(f) \rightarrow (\text{destination})$
Status Affected:	Z
Description:	The contents of register 'f' are complemented. If 'd' = '0', the result is stored in W. If 'd' = '1', the result is stored back in register 'f'.

<b>GOTO</b>	<b>Unconditional Branch</b>
Syntax:	[ <i>label</i> ] GOTO k
Operands:	$0 \leq k \leq 2047$
Operation:	$k \rightarrow PC<10:0>$ $PCLATH<4:3> \rightarrow PC<12:11>$
Status Affected:	None
Description:	GOTO is an unconditional branch. The eleven-bit immediate value is loaded into PC bits <10:0>. The upper bits of PC are loaded from PCLATH<4:3>. GOTO is a two-cycle instruction.

<b>DECF</b>	<b>Decrement f</b>
Syntax:	[ <i>label</i> ] DECF f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	$(f) - 1 \rightarrow (\text{destination})$
Status Affected:	Z
Description:	Decrement register 'f'. If 'd' = '0', the result is stored in the W register. If 'd' = '1', the result is stored back in register 'f'.

<b>INCF</b>	<b>Increment f</b>
Syntax:	[ <i>label</i> ] INCF f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	$(f) + 1 \rightarrow (\text{destination})$
Status Affected:	Z
Description:	The contents of register 'f' are incremented. If 'd' = '0', the result is placed in the W register. If 'd' = '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:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	$(f) - 1 \rightarrow (\text{destination});$ skip if result = 0
Status Affected:	None
Description:	The contents of register 'f' are decremented. If 'd' = '0', the result is placed in the W register. If 'd' = '1', the result is placed back in register 'f'. If the result is '1', the next instruction is executed. If the result is '0', then a NOP is executed instead, making it a 2 TCY instruction.

<b>INCFSZ</b>	<b>Increment f, Skip if 0</b>
Syntax:	[ <i>label</i> ] INCFSZ f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	$(f) + 1 \rightarrow (\text{destination}),$ skip if result = 0
Status Affected:	None
Description:	The contents of register 'f' are incremented. If 'd' = '0', the result is placed in the W register. If 'd' = '1', the result is placed back in register 'f'. If the result is '1', the next instruction is executed. If the result is '0', a NOP is executed instead, making it a 2 TCY instruction.

# PIC16F72

---

---

**IORLW**            **Inclusive OR Literal with W**

---

Syntax:            [ *label* ] IORLW k

Operands:         $0 \leq k \leq 255$

Operation:        (W) .OR. k  $\rightarrow$  (W)

Status Affected: Z

Description:      The contents of the W register are OR'd with the eight-bit literal 'k'. The result is placed in the W 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 eight-bit literal 'k' is loaded into W register. The don't cares will assemble as '0's.

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

---

Syntax:            [ *label* ] IORWF f,d

Operands:         $0 \leq f \leq 127$   
                      d  $\in$  [0,1]

Operation:        (W) .OR. (f)  $\rightarrow$  (destination)

Status Affected: Z

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

**MOVWF**           **Move W to f**

---

Syntax:            [ *label* ] MOVWF f

Operands:         $0 \leq f \leq 127$

Operation:        (W)  $\rightarrow$  (f)

Status Affected: None

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

**MOVF**            **Move f**

---

Syntax:            [ *label* ] MOVF f,d

Operands:         $0 \leq f \leq 127$   
                      d  $\in$  [0,1]

Operation:        (f)  $\rightarrow$  (destination)

Status Affected: Z

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

**NOP**              **No Operation**

---

Syntax:            [ *label* ] NOP

Operands:        None

Operation:        No operation

Status Affected: None

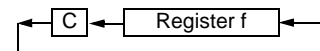
Description:      No operation.

## **RETFIE**      **Return from Interrupt**

Syntax:      [ *label* ] RETFIE  
 Operands:    None  
 Operation:    TOS → PC,  
                   1 → GIE  
 Status Affected: None

## **RLF**            **Rotate Left f through Carry**

Syntax:      [ *label* ] RLF f,d  
 Operands:     $0 \leq f \leq 127$   
                    $d \in [0,1]$   
 Operation:    See description below  
 Status Affected: C  
 Description:    The contents of register 'f' are rotated one bit to the left through the Carry Flag. If 'd' = '0', the result is placed in the W register. If 'd' = '1', the result is stored back in register 'f'.

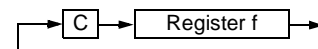


## **RETLW**        **Return with Literal in W**

Syntax:      [ *label* ] RETLW k  
 Operands:     $0 \leq k \leq 255$   
 Operation:     $k \rightarrow (W)$ ;  
                   TOS → PC  
 Status Affected: None  
 Description:    The W register is loaded with the eight-bit literal 'k'. The program counter is loaded from the top of the stack (the return address). This is a two-cycle instruction.

## **RRF**            **Rotate Right f through Carry**

Syntax:      [ *label* ] RRF f,d  
 Operands:     $0 \leq f \leq 127$   
                    $d \in [0,1]$   
 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' = '0', the result is placed in the W register. If 'd' = '1', the result is placed back in register 'f'.



## **RETURN**        **Return from Subroutine**

Syntax:      [ *label* ] RETURN  
 Operands:    None  
 Operation:    TOS → PC  
 Status Affected: None  
 Description:    Return from subroutine. The stack is POPed and the top of the stack (TOS) is loaded into the program counter. This is a two-cycle instruction.

## **SLEEP**

Syntax:      [ *label* ] SLEEP  
 Operands:    None  
 Operation:    00h → WDT,  
                   0 → WDT prescaler,  
                   1 →  $\overline{TO}$ ,  
                   0 →  $\overline{PD}$   
 Status Affected:  $\overline{TO}$ ,  $\overline{PD}$   
 Description:    The power-down status bit,  $\overline{PD}$  is cleared. Time-out status bit,  $\overline{TO}$  is set. Watchdog Timer and its prescaler are cleared. The processor is put into SLEEP mode with the oscillator stopped.

# PIC16F72

---

---

**SUBLW**      **Subtract W from Literal**

---

Syntax:      [ *label* ]    SUBLW   k

Operands:     $0 \leq k \leq 255$

Operation:     $k - (W) \rightarrow (W)$

Status Affected:    C, DC, Z

Description:    The W register is subtracted (2's complement method) from the eight-bit literal 'k'. The result is placed in the W register.

**XORLW**      **Exclusive OR Literal with W**

---

Syntax:      [ *label* ]    XORLW   k

Operands:     $0 \leq k \leq 255$

Operation:     $(W) .XOR. k \rightarrow (W)$

Status Affected:    Z

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

**SUBWF**      **Subtract W from f**

---

Syntax:      [ *label* ]    SUBWF   f,d

Operands:     $0 \leq f \leq 127$   
                 $d \in [0,1]$

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

Status Affected:    C, DC, Z

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

**XORWF**      **Exclusive OR W with f**

---

Syntax:      [ *label* ]    XORWF   f,d

Operands:     $0 \leq f \leq 127$   
                 $d \in [0,1]$

Operation:     $(W) .XOR. (f) \rightarrow (\text{destination})$

Status Affected:    Z

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

**SWAPF**      **Swap Nibbles in f**

---

Syntax:      [ *label* ]    SWAPF   f,d

Operands:     $0 \leq f \leq 127$   
                 $d \in [0,1]$

Operation:     $(f<3:0>) \rightarrow (\text{destination}<7:4>),$   
                 $(f<7:4>) \rightarrow (\text{destination}<3:0>)$

Status Affected:    None

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



## 13.0 DEVELOPMENT SUPPORT

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

- Integrated Development Environment
  - MPLAB® IDE Software
- Assemblers/Compilers/Linkers
  - MPASM™ Assembler
  - MPLAB C17 and MPLAB C18 C Compilers
  - MPLINK™ Object Linker/  
MPLIB™ Object Librarian
- Simulators
  - MPLAB SIM Software Simulator
- Emulators
  - MPLAB ICE 2000 In-Circuit Emulator
  - ICEPIC™ In-Circuit Emulator
- In-Circuit Debugger
  - MPLAB ICD
- Device Programmers
  - PRO MATE® II Universal Device Programmer
  - PICSTART® Plus Entry-Level Development Programmer
- Low Cost Demonstration Boards
  - PICDEM™ 1 Demonstration Board
  - PICDEM 2 Demonstration Board
  - PICDEM 3 Demonstration Board
  - PICDEM 17 Demonstration Board
  - KEELOQ® Demonstration Board

### 13.1 MPLAB Integrated Development Environment Software

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

- An interface to debugging tools
  - simulator
  - programmer (sold separately)
  - emulator (sold separately)
  - in-circuit debugger (sold separately)
- A full-featured editor
- A project manager
- Customizable toolbar and key mapping
- A status bar
- On-line help

The MPLAB IDE allows you to:

- Edit your source files (either assembly or 'C')
- One touch assemble (or compile) and download to PICmicro emulator and simulator tools (automatically updates all project information)
- Debug using:
  - source files
  - absolute listing file
  - machine code

The ability to use MPLAB IDE with multiple debugging tools allows users to easily switch from the cost-effective simulator to a full-featured emulator with minimal retraining.

### 13.2 MPASM Assembler

The MPASM assembler is a full-featured universal macro assembler for all PICmicro MCU's.

The MPASM assembler has a command line interface and a Windows shell. It can be used as a stand-alone application on a Windows 3.x or greater system, or it can be used through MPLAB IDE. 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, an absolute LST file that contains source lines and generated machine code, and a COD file for debugging.

The MPASM assembler features include:

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

### 13.3 MPLAB C17 and MPLAB C18 C Compilers

The MPLAB C17 and MPLAB C18 Code Development Systems are complete ANSI 'C' compilers for Microchip's PIC17CXXX and PIC18CXXX family of microcontrollers, respectively. These compilers provide powerful integration capabilities and ease of use not found with other compilers.

For easier source level debugging, the compilers provide symbol information that is compatible with the MPLAB IDE memory display.

## 13.4 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK object linker combines relocatable objects created by the MPASM assembler and the MPLAB C17 and MPLAB C18 C compilers. It can also link relocatable objects from pre-compiled libraries, using directives from a linker script.

The MPLIB object librarian is a librarian for pre-compiled code to be used with the MPLINK object linker. When a routine from a library is called from another 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 MPLIB object librarian manages the creation and modification of library files.

The MPLINK object linker features include:

- Integration with MPASM assembler and MPLAB C17 and MPLAB C18 C compilers.
- Allows all memory areas to be defined as sections to provide link-time flexibility.

The MPLIB object librarian features include:

- Easier linking because single libraries can be included instead of many smaller files.
- Helps keep code maintainable by grouping related modules together.
- Allows libraries to be created and modules to be added, listed, replaced, deleted or extracted.

## 13.5 MPLAB SIM Software Simulator

The MPLAB SIM software simulator allows code development in a PC-hosted environment by simulating the PICmicro series microcontrollers on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a file, or user-defined key press, to any of the pins. The execution can be performed in single step, execute until break, or trace mode.

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

## 13.6 MPLAB ICE High Performance Universal In-Circuit Emulator with MPLAB IDE

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

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

The MPLAB ICE in-circuit emulator system has been designed as a real-time emulation system, with advanced features that are generally found on more expensive development tools. The PC platform and Microsoft® Windows environment were chosen to best make these features available to you, the end user.

## 13.7 ICEPIC In-Circuit Emulator

The ICEPIC low cost, in-circuit emulator is a solution for the Microchip Technology PIC16C5X, PIC16C6X, PIC16C7X and PIC16CXXX families of 8-bit One-Time-Programmable (OTP) microcontrollers. The modular system can support different subsets of PIC16C5X or PIC16CXXX products through the use of interchangeable personality modules, or daughter boards. The emulator is capable of emulating without target application circuitry being present.

## 13.8 MPLAB ICD In-Circuit Debugger

Microchip's In-Circuit Debugger, MPLAB ICD, is a powerful, low cost, run-time development tool. This tool is based on the FLASH PICmicro MCUs and can be used to develop for this and other PICmicro microcontrollers. The MPLAB ICD utilizes the in-circuit debugging capability built into the FLASH devices. This feature, along with Microchip's In-Circuit Serial Programming™ protocol, offers cost-effective in-circuit FLASH debugging from the graphical user interface of the MPLAB Integrated Development Environment. This enables a designer to develop and debug source code by watching variables, single-stepping and setting break points. Running at full speed enables testing hardware in real-time.

## 13.9 PRO MATE II Universal Device Programmer

The PRO MATE II universal device programmer is a full-featured programmer, capable of operating in stand-alone mode, as well as PC-hosted mode. The PRO MATE II device programmer is CE compliant.

The PRO MATE II device programmer has programmable VDD and VPP supplies, which allow it to verify programmed memory at VDD min and VDD max for maximum reliability. It has an LCD display for instructions and error messages, keys to enter commands and a modular detachable socket assembly to support various package types. In stand-alone mode, the PRO MATE II device programmer can read, verify, or program PICmicro devices. It can also set code protection in this mode.

## 13.10 PICSTART Plus Entry Level Development Programmer

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

The PICSTART Plus development programmer supports all PICmicro devices with up to 40 pins. Larger pin count devices, such as the PIC16C92X and PIC17C76X, may be supported with an adapter socket. The PICSTART Plus development programmer is CE compliant.

## 13.11 PICDEM 1 Low Cost PICmicro Demonstration Board

The PICDEM 1 demonstration board is a simple board which demonstrates the capabilities of several of Microchip's microcontrollers. The microcontrollers supported are: PIC16C5X (PIC16C54 to PIC16C58A), PIC16C61, PIC16C62X, PIC16C71, PIC16C8X, PIC17C42, PIC17C43 and PIC17C44. All necessary hardware and software is included to run basic demo programs. The user can program the sample microcontrollers provided with the PICDEM 1 demonstration board on a PRO MATE II device programmer, or a PICSTART Plus development programmer, and easily test firmware. The user can also connect the PICDEM 1 demonstration board to the MPLAB ICE in-circuit emulator and download the firmware to the emulator for testing. A prototype area is available for the user to build some additional hardware and connect it to the microcontroller socket(s). Some of the features include an RS-232 interface, a potentiometer for simulated analog input, push button switches and eight LEDs connected to PORTB.

## 13.12 PICDEM 2 Low Cost PIC16CXX Demonstration Board

The PICDEM 2 demonstration board is a simple demonstration board that supports the PIC16C62, PIC16C64, PIC16C65, PIC16C73 and PIC16C74 microcontrollers. All the necessary hardware and software is included to run the basic demonstration programs. The user can program the sample microcontrollers provided with the PICDEM 2 demonstration board on a PRO MATE II device programmer, or a PICSTART Plus development programmer, and easily test firmware. The MPLAB ICE in-circuit emulator may also be used with the PICDEM 2 demonstration board to test firmware. A prototype area has been provided to the user for adding additional hardware and connecting it to the microcontroller socket(s). Some of the features include a RS-232 interface, push button switches, a potentiometer for simulated analog input, a serial EEPROM to demonstrate usage of the I<sup>2</sup>C™ bus and separate headers for connection to an LCD module and a keypad.

# PIC16F72

---

## 13.13 PICDEM 3 Low Cost PIC16CXXX Demonstration Board

The PICDEM 3 demonstration board is a simple demonstration board that supports the PIC16C923 and PIC16C924 in the PLCC package. It will also support future 44-pin PLCC microcontrollers with an LCD Module. All the necessary hardware and software is included to run the basic demonstration programs. The user can program the sample microcontrollers provided with the PICDEM 3 demonstration board on a PRO MATE II device programmer, or a PICSTART Plus development programmer with an adapter socket, and easily test firmware. The MPLAB ICE in-circuit emulator may also be used with the PICDEM 3 demonstration board to test firmware. A prototype area has been provided to the user for adding hardware and connecting it to the microcontroller socket(s). Some of the features include a RS-232 interface, push button switches, a potentiometer for simulated analog input, a thermistor and separate headers for connection to an external LCD module and a keypad. Also provided on the PICDEM 3 demonstration board is a LCD panel, with 4 commons and 12 segments, that is capable of displaying time, temperature and day of the week. The PICDEM 3 demonstration board provides an additional RS-232 interface and Windows software for showing the demultiplexed LCD signals on a PC. A simple serial interface allows the user to construct a hardware demultiplexer for the LCD signals.

## 13.14 PICDEM 17 Demonstration Board

The PICDEM 17 demonstration board is an evaluation board that demonstrates the capabilities of several Microchip microcontrollers, including PIC17C752, PIC17C756A, PIC17C762 and PIC17C766. All necessary hardware is included to run basic demo programs, which are supplied on a 3.5-inch disk. A programmed sample is included and the user may erase it and program it with the other sample programs using the PRO MATE II device programmer, or the PICSTART Plus development programmer, and easily debug and test the sample code. In addition, the PICDEM 17 demonstration board supports downloading of programs to and executing out of external FLASH memory on board. The PICDEM 17 demonstration board is also usable with the MPLAB ICE in-circuit emulator, or the PICMASTER emulator and all of the sample programs can be run and modified using either emulator. Additionally, a generous prototype area is available for user hardware.

## 13.15 KEELOQ Evaluation and Programming Tools

KEELOQ evaluation and programming tools support Microchip's HCS Secure Data Products. The HCS evaluation kit includes a LCD display to show changing codes, a decoder to decode transmissions and a programming interface to program test transmitters.

**TABLE 13-1: DEVELOPMENT TOOLS FROM MICROCHIP**

Tools	PIC12CXXXX	PIC14000	PIC16C5X	PIC16C6X	PIC16CXX	PIC16F62X	PIC16C7X	PIC16C7XX	PIC16C8X/	PIC16F8XX	PIC16C9XX	PIC17C4X	PIC17C7XX	PIC18CXX2	PIC18FXX	24CXX/ 25CXX/ 93CXX	HCSXX	MCRFXXX	MCP2510
<b>Software Tools</b>																			
MPLAB® Integrated Development Environment	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
MPLAB® C17 C Compiler																			
MPLAB® C18 C Compiler																			
MPASM™ Assembler/ MPLINK™ Object Linker	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
MPLAB® ICE In-Circuit Emulator	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ICEPIC™ In-Circuit Emulator	✓		✓	✓	✓		✓	✓	✓	✓	✓								
<b>Debugger</b>																			
MPLAB® ICD In-Circuit Debugger				✓			✓			✓					✓				
<b>Programmers</b>																			
PICSTART® Plus Entry Level Development Programmer	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
PRO MATE® II Universal Device Programmer	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
<b>Demo Boards and Eval Kits</b>																			
PICDEM™ 1 Demonstration Board			✓				†		✓										
PICDEM™ 2 Demonstration Board				†			†							✓					
PICDEM™ 3 Demonstration Board										✓									
PICDEM™ 14A Demonstration Board		✓																	
PICDEM™ 17 Demonstration Board												✓							
KEELOQ® Evaluation Kit																	✓		
KEELOQ® Transponder Kit																	✓		
microID™ Programmer's Kit																		✓	
125 kHz microID™ Developer's Kit																		✓	
125 kHz Anticollision microID™ Developer's Kit																		✓	
13.56 MHz Anticollision microID™ Developer's Kit																		✓	
MCP2510 CAN Developer's Kit																		✓	✓

\* Contact the Microchip Technology Inc. web site at [www.microchip.com](http://www.microchip.com) for information on how to use the MPLAB® ICD In-Circuit Debugger (DV164001) with PIC16C62, 63, 64, 65, 72, 73, 74, 76, 77.

\*\* Contact Microchip Technology Inc. for availability date.

† Development tool is available on select devices.

## 14.0 ELECTRICAL CHARACTERISTICS

### Absolute Maximum Ratings †

Ambient temperature under bias .....	-55 to +125°C
Storage temperature .....	-65°C to +150°C
Voltage on any pin with respect to VSS (except VDD, $\overline{\text{MCLR}}$ , and RA4) .....	-0.3V to (VDD + 0.3V)
Voltage on VDD with respect to VSS .....	-0.3 to +6.5V
Voltage on $\overline{\text{MCLR}}$ with respect to VSS ( <b>Note 2</b> ) .....	0 to +13.5V
Voltage on RA4 with respect to Vss .....	0 to +12V
Total power dissipation ( <b>Note 1</b> ) .....	1.0W
Maximum current out of VSS pin .....	300 mA
Maximum current into VDD pin .....	250 mA
Input clamp current, I <sub>IK</sub> (V <sub>I</sub> < 0 or V <sub>I</sub> > VDD) .....	± 20 mA
Output clamp current, I <sub>OK</sub> (V <sub>O</sub> < 0 or V <sub>O</sub> > VDD) .....	± 20 mA
Maximum output current sunk by any I/O pin .....	25 mA
Maximum output current sourced by any I/O pin .....	25 mA
Maximum current sunk by PORTA, PORTB .....	200 mA
Maximum current sourced by PORTA, PORTB .....	200 mA
Maximum current sunk by PORTC .....	200 mA
Maximum current sourced by PORTC .....	200 mA

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

**2:** Voltage spikes at the  $\overline{\text{MCLR}}$  pin may cause unpredictable results. A series resistor of greater than 1 kΩ should be used to pull  $\overline{\text{MCLR}}$  to VDD, rather than tying the pin directly to VDD.

† 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.

# PIC16F72

FIGURE 14-1: PIC16F72 (INDUSTRIAL, EXTENDED) VOLTAGE-FREQUENCY GRAPH

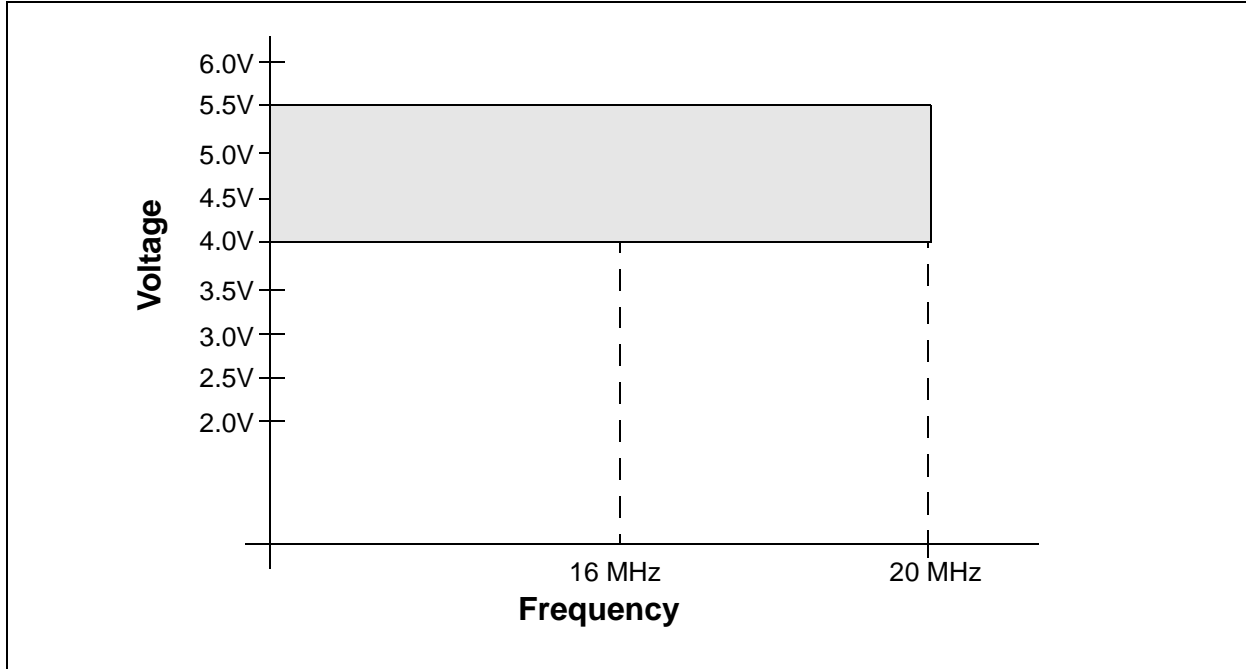
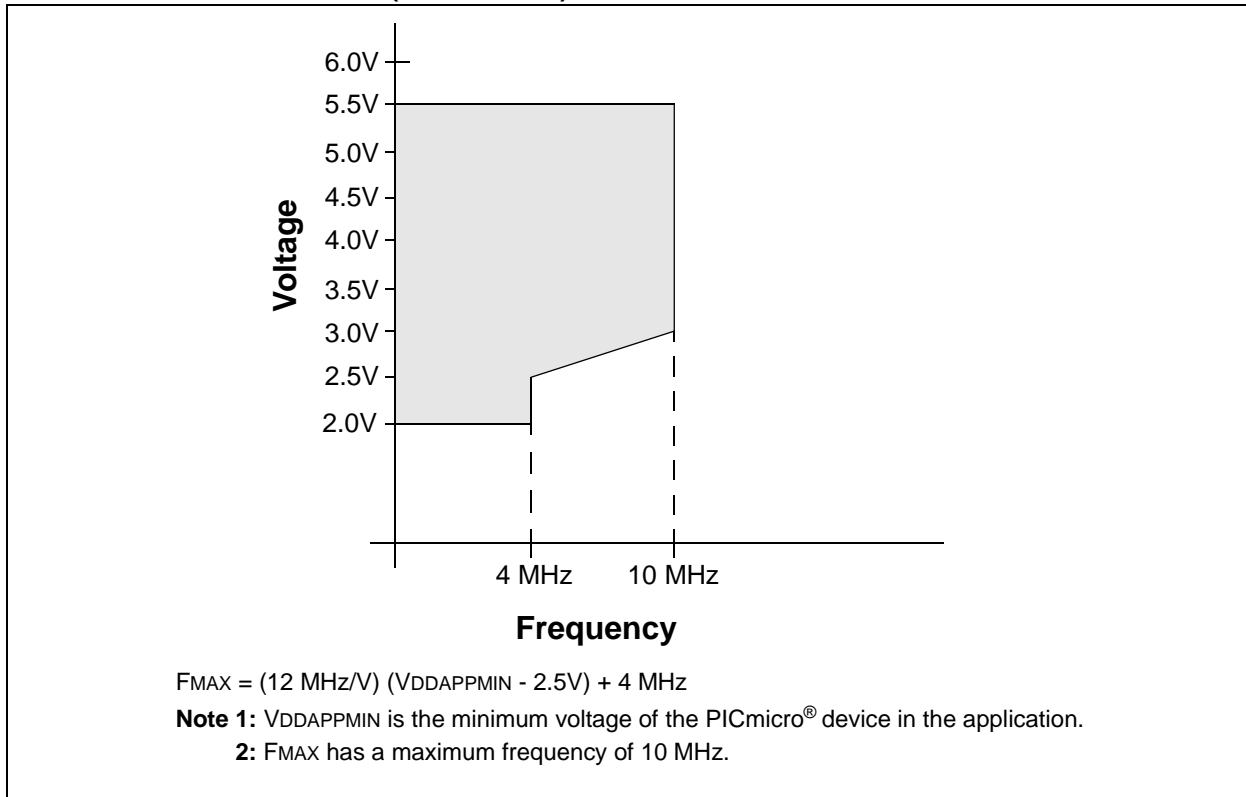


FIGURE 14-2: PIC16LF72 (INDUSTRIAL) VOLTAGE-FREQUENCY GRAPH



## 14.1 DC Characteristics: PIC16F72 (Industrial, Extended) PIC16LF72 (Industrial)

PIC16LF72 (Industrial)		Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial					
PIC16F72 (Industrial, Extended)		Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for extended					
Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
D001	VDD	<b>Supply Voltage</b>					
		PIC16LF72	2.0	—	5.5	V	A/D not used, $-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$
			2.5	—	5.5	V	A/D in use, $-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$
			2.2	—	5.5	V	A/D in use, $0^{\circ}\text{C}$ to $+85^{\circ}\text{C}$
D001 D001A		PIC16F72	4.0	—	5.5	V	All configurations
		VBOR*	—	—	5.5	V	BOR enabled ( <b>Note 7</b> )
D002*	VDR	<b>RAM Data Retention Voltage (Note 1)</b>	—	1.5	—	V	
D003	VPOR	<b>VDD Start Voltage</b> to ensure internal Power-on Reset signal	—	VSS	—	V	See section on Power-on Reset for details
D004*	SVDD	<b>VDD Rise Rate</b> to ensure internal Power-on Reset signal	0.05	—	—	V/ms	See section on Power-on Reset for details
D005	VBOR	<b>Brown-out Reset Voltage</b>	3.65	4.0	4.35	V	BOREN bit in configuration word enabled

\* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V,  $25^{\circ}\text{C}$  unless otherwise stated. These parameters are for design guidance only and are not tested.

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

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

MCLR = VDD; WDT enabled/disabled as specified.

**3:** The power-down current in SLEEP mode does not depend on the oscillator type. Power-down current is measured with the part in SLEEP mode, with all I/O pins in hi-impedance state and tied to VDD and VSS.

**4:** For RC osc configuration, current through REXT is not included. The current through the resistor can be estimated by the formula  $I_r = V_{DD}/2R_{EXT}$  (mA) with REXT in kΩ.

**5:** Timer1 oscillator (when enabled) adds approximately 20 μA to the specification. This value is from characterization and is for design guidance only. This is not tested.

**6:** The Δ current is the additional current consumed when this peripheral is enabled. This current should be added to the base IDD or IPD measurement.

**7:** When BOR is enabled, the device will operate correctly until the VBOR voltage trip point is reached.



# PIC16F72

## 14.1 DC Characteristics: PIC16F72 (Industrial, Extended) PIC16LF72 (Industrial) (Continued)

PIC16F72 (Industrial)		Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial					
PIC16F72 (Industrial, Extended)		Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for extended					
Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
	IDD	<b>Supply Current (Notes 2, 5)</b>					
D010		PIC16LF72	—	0.4	2.0	mA	XT, RC osc configuration FOSC = 4 MHz, VDD = 3.0V ( <b>Note 4</b> )
D010A			—	25	48	μA	LP osc configuration FOSC = 32 kHz, VDD = 3.0V, WDT disabled
D010		PIC16F72	-	0.9	4	mA	XT, RC osc configuration FOSC = 4 MHz, VDD = 5.5V ( <b>Note 4</b> )
D013			—	5.2	15	mA	HS osc configuration FOSC = 20 MHz, VDD = 5.5V
D015*	ΔIBOR	<b>Brown-out Reset Current (Note 6)</b>	—	25	200	μA	BOR enabled, VDD = 5.0V
	IPD	<b>Power-down Current (Notes 3, 5)</b>					
D020		PIC16LF72	—	2.0	30	μA	VDD = 3.0V, WDT enabled, $-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$
D021			—	0.1	5	μA	VDD = 3.0V, WDT disabled, $-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$
D020		PIC16F72	—	5.0	42	μA	VDD = 4.0V, WDT enabled, $-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$
D021			—	0.1	19	μA	VDD = 4.0V, WDT disabled, $-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$
D023*	ΔIBOR	<b>Brown-out Reset Current (Note 6)</b>	—	25	200	μA	BOR enabled, VDD = 5.0V

\* These parameters are characterized but not tested.

† 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:** This is the limit to which VDD can be lowered without losing RAM data.
- Note 2:** The supply current is mainly a function of the operating voltage and frequency. Other factors, such as I/O pin loading and switching rate, oscillator type, internal code execution pattern and temperature, also have an impact on the current consumption. 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 VDD  
 MCLR = VDD; WDT enabled/disabled as specified.
- Note 3:** The power-down current in SLEEP mode does not depend on the oscillator type. Power-down current is measured with the part in SLEEP mode, with all I/O pins in hi-impedance state and tied to VDD and VSS.
- Note 4:** For RC osc configuration, current through REXT is not included. The current through the resistor can be estimated by the formula  $I_r = V_{DD}/2R_{EXT}$  (mA) with REXT in kΩ.
- Note 5:** Timer1 oscillator (when enabled) adds approximately 20 μA to the specification. This value is from characterization and is for design guidance only. This is not tested.
- Note 6:** The Δ current is the additional current consumed when this peripheral is enabled. This current should be added to the base IDD or IPD measurement.
- Note 7:** When BOR is enabled, the device will operate correctly until the VBOR voltage trip point is reached.

## 14.2 DC Characteristics: PIC16F72 (Industrial, Extended) PIC16LF72 (Industrial)

DC CHARACTERISTICS		Standard Operating Conditions (unless otherwise stated)					
		Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for extended					
		Operating voltage $V_{DD}$ range as described in DC Specification, Section 14.1.					
Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
	$V_{IL}$	<b>Input Low Voltage</b>					
D030		I/O ports					
D030A		with TTL buffer	$V_{SS}$	—	$0.15 V_{DD}$	V	For entire $V_{DD}$ range
D031		with Schmitt Trigger buffer	$V_{SS}$	—	$0.8V$	V	$4.5V \leq V_{DD} \leq 5.5V$
D032		$\overline{\text{MCLR}}$ , OSC1 (in RC mode)	$V_{SS}$	—	$0.2 V_{DD}$	V	
D033		OSC1 (in XT and LP mode)	$V_{SS}$	—	$0.3V$	V	<b>(Note 1)</b>
		OSC1 (in HS mode)	$V_{SS}$	—	$0.3 V_{DD}$	V	<b>(Note 1)</b>
	$V_{IH}$	<b>Input High Voltage</b>					
D040		I/O ports					
D040A		with TTL buffer	2.0	—	$V_{DD}$	V	$4.5V \leq V_{DD} \leq 5.5V$
D041		with Schmitt Trigger buffer	$0.25 V_{DD} + 0.8V$	—	$V_{DD}$	V	For entire $V_{DD}$ range
D042		$\overline{\text{MCLR}}$	$0.8 V_{DD}$	—	$V_{DD}$	V	For entire $V_{DD}$ range
D042A		OSC1 (in XT and LP mode)	1.6V	—	$V_{DD}$	V	<b>(Note 1)</b>
		OSC1 (in HS mode)	$0.7 V_{DD}$	—	$V_{DD}$	V	<b>(Note 1)</b>
D043		OSC1 (in RC mode)	$0.9 V_{DD}$	—	$V_{DD}$	V	
D070	IPURB	<b>PORTB Weak Pull-up Current</b>	50	250	400	$\mu\text{A}$	$V_{DD} = 5V, V_{PIN} = V_{SS}$
	$I_{IL}$	<b>Input Leakage Current (Notes 2, 3)</b>					
D060		I/O ports	—	—	$\pm 1$	$\mu\text{A}$	$V_{SS} \leq V_{PIN} \leq V_{DD}$ , Pin at hi-impedance
D061		$\overline{\text{MCLR}}$ , RA4/T0CKI	—	—	$\pm 5$	$\mu\text{A}$	$V_{SS} \leq V_{PIN} \leq V_{DD}$
D063		OSC1	—	—	$\pm 5$	$\mu\text{A}$	$V_{SS} \leq V_{PIN} \leq V_{DD}$ , XT, HS and LP osc configuration

\* These parameters are characterized but not tested.

† 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 RC oscillator configuration, the OSC1/CLKI pin is a Schmitt Trigger input. It is not recommended that the PIC16F72 be driven with external clock in RC mode.
- 2:** The leakage current on the  $\overline{\text{MCLR}}$  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 current sourced by the pin.

# PIC16F72

## 14.2 DC Characteristics: PIC16F72 (Industrial, Extended) PIC16LF72 (Industrial) (Continued)

DC CHARACTERISTICS		Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for extended Operating voltage $V_{DD}$ range as described in DC Specification, Section 14.1.					
Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
		<b>Output Low Voltage</b>					
D080	VOL	I/O ports	—	—	0.6	V	$I_{OL} = 8.5\text{ mA}$ , $V_{DD} = 4.5\text{V}$ , $-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$
D083		OSC2/CLKO (RC osc config)	—	—	0.6	V	$I_{OL} = 1.6\text{ mA}$ , $V_{DD} = 4.5\text{V}$ , $-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$
		<b>Output High Voltage</b>					
D090	VOH	I/O ports ( <b>Note 3</b> )	$V_{DD} - 0.7$	—	—	V	$I_{OH} = -3.0\text{ mA}$ , $V_{DD} = 4.5\text{V}$ , $-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$
D092		OSC2/CLKO (RC osc config)	$V_{DD} - 0.7$	—	—	V	$I_{OH} = -1.3\text{ mA}$ , $V_{DD} = 4.5\text{V}$ , $-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$
D150*	VOD	<b>Open Drain High Voltage</b>	—	—	12	V	RA4 pin
		<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 (in RC mode)	—	—	50	pF	
D102	CB	SCL, SDA in I <sup>2</sup> C mode	—	—	400	pF	
		<b>Program FLASH Memory</b>					
D130	EP	Endurance	100	1000	—	E/W	25°C at 5V
D131	VPR	VDD for read	2.0	—	5.5	V	

\* These parameters are characterized but not tested.

† 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 RC oscillator configuration, the OSC1/CLKI pin is a Schmitt Trigger input. It is not recommended that the PIC16F72 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 current sourced by the pin.

## 14.3 Timing Parameter Symbology

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

- |             |           |  |
|-------------|-----------|--|
| 1. TppS2ppS | 3. TCC:ST | (I <sup>2</sup> C specifications only) |
| 2. TppS     | 4. Ts     | (I <sup>2</sup> C specifications only) |

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

Lowercase letters (pp) and their meanings:

<b>pp</b>		
cc	CCP1	osc
ck	CLKO	rd
cs	$\overline{CS}$	rw
di	SDI	sc
do	SDO	ss
dt	Data in	t0
io	I/O port	t1
mc	$\overline{MCLR}$	wr
		OSC1
		$\overline{RD}$
		$\overline{RD}$ or $\overline{WR}$
		SCK
		$\overline{SS}$
		T0CKI
		T1CKI
		$\overline{WR}$

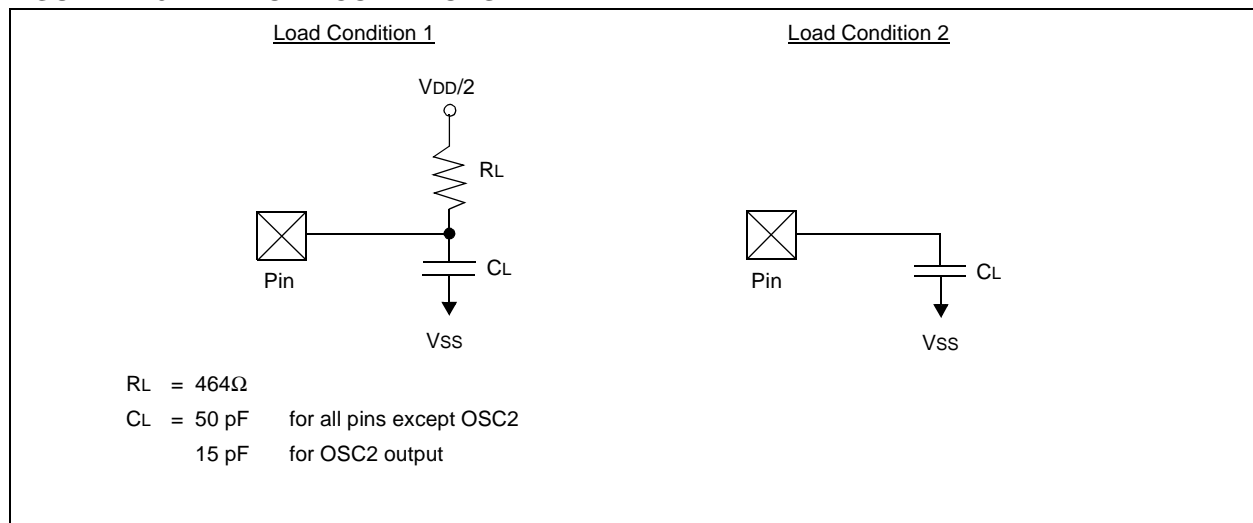
Uppercase letters and their meanings:

<b>S</b>		
F	Fall	P
H	High	R
I	Invalid (Hi-impedance)	V
L	Low	Z
		Hi-impedance
<b>I<sup>2</sup>C only</b>		
AA	output access	High
BUF	Bus free	Low
		Period
		Rise
		Valid
		Hi-impedance

TCC:ST (I<sup>2</sup>C specifications only)

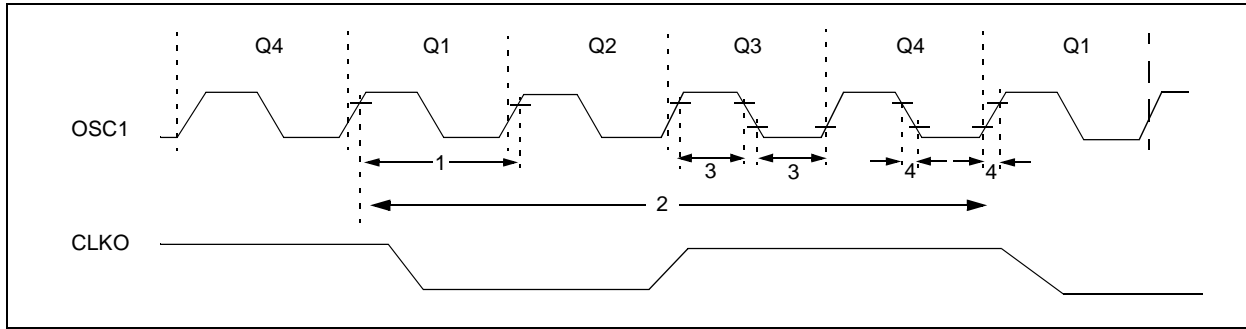
<b>CC</b>		
HD	Hold	SU
		Setup
<b>ST</b>		
DAT	DATA input hold	STO
STA	START condition	STOP condition

**FIGURE 14-3: LOAD CONDITIONS**



# PIC16F72

**FIGURE 14-4: EXTERNAL CLOCK TIMING**



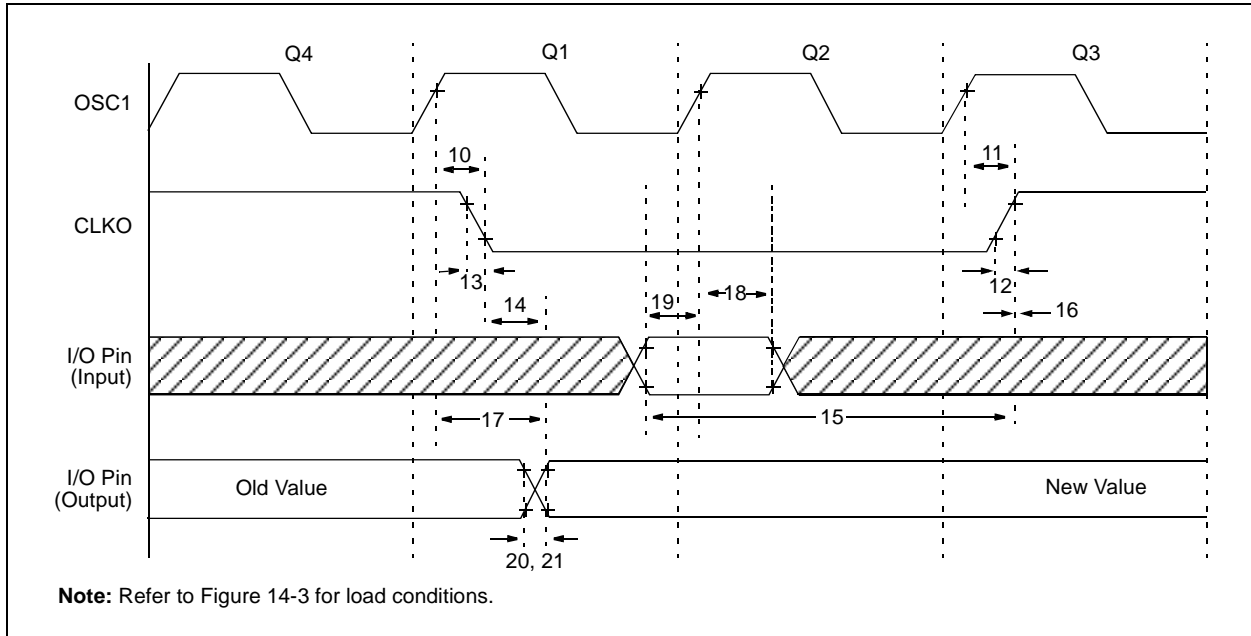
**TABLE 14-1: EXTERNAL CLOCK TIMING REQUIREMENTS**

Parameter No.	Symbol	Characteristic	Min	Typ†	Max	Units	Conditions
	FOSC	<b>External CLKI Frequency (Note 1)</b>	DC	—	1	MHz	XT Osc mode
			DC	—	20	MHz	HS Osc mode
			DC	—	32	kHz	LP Osc mode
		<b>Oscillator Frequency (Note 1)</b>	DC	—	4	MHz	RC osc mode
			0.1	—	4	MHz	XT Osc mode
1	TOSC	<b>External CLKI Period (Note 1)</b>	1000	—	—	ns	XT Osc mode
			50	—	—	ns	HS Osc mode
			5	—	—	ms	LP Osc mode
		<b>Oscillator Period (Note 1)</b>	250	—	—	ns	RC Osc mode
			250	—	10,000	ns	XT Osc mode
2	TCY	<b>Instruction Cycle Time (Note 1)</b>	50	—	250	ns	HS Osc mode
			5	—	—	ms	LP Osc mode
			200	TCY	DC	ns	TCY = 4/FOSC
3	TosL, TosH	<b>External Clock in (OSC1) High or Low Time</b>	500	—	—	ns	XT oscillator
			2.5	—	—	ms	LP oscillator
			15	—	—	ns	HS oscillator
4	TosR, TosF	<b>External Clock in (OSC1) Rise or Fall Time</b>	—	—	25	ns	XT oscillator
			—	—	50	ns	LP oscillator
			—	—	15	ns	HS oscillator

† 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:** Instruction cycle period (TCY) equals four times the input oscillator time-base period. 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. All devices are tested to operate at "min" values with an external clock applied to the OSC1/CLKI pin. When an external clock input is used, the "max" cycle time limit is "DC" (no clock) for all devices.

**FIGURE 14-5: CLKO AND I/O TIMING**



**TABLE 14-2: CLKO AND I/O TIMING REQUIREMENTS**

Param No.	Symbol	Characteristic	Min	Typ†	Max	Units	Conditions	
10*	TosH2ckL	OSC1↑ to CLKO↓	—	75	200	ns	(Note 1)	
11*	TosH2ckH	OSC1↑ to CLKO↑	—	75	200	ns	(Note 1)	
12*	TckR	CLKO rise time	—	35	100	ns	(Note 1)	
13*	TckF	CLKO fall time	—	35	100	ns	(Note 1)	
14*	TckL2ioV	CLKO↓ to Port out valid	—	—	0.5 T <sub>CY</sub> + 20	ns	(Note 1)	
15*	TioV2ckH	Port in valid before CLKO↑	T <sub>OSC</sub> + 200	—	—	ns	(Note 1)	
16*	TckH2ioI	Port in hold after CLKO↑	0	—	—	ns	(Note 1)	
17*	TosH2ioV	OSC1↑ (Q1 cycle) to Port out valid	—	100	255	ns		
18*	TosH2ioI	OSC1↑ (Q2 cycle) to Port input invalid (I/O in hold time)	Standard (F)	100	—	—	ns	
			Extended (LF)	200	—	—	ns	
19*	TioV2osH	Port input valid to OSC1↑ (I/O in setup time)	0	—	—	ns		
20*	TioR	Port output rise time	Standard (F)	—	10	40	ns	
			Extended (LF)	—	—	145	ns	
21*	TioF	Port output fall time	Standard (F)	—	10	40	ns	
			Extended (LF)	—	—	145	ns	
22††*	TiNP	INT pin high or low time	T <sub>CY</sub>	—	—	ns		
23††*	TRBP	RB7:RB4 change INT high or low time	T <sub>CY</sub>	—	—	ns		

\* These parameters are characterized but not tested.

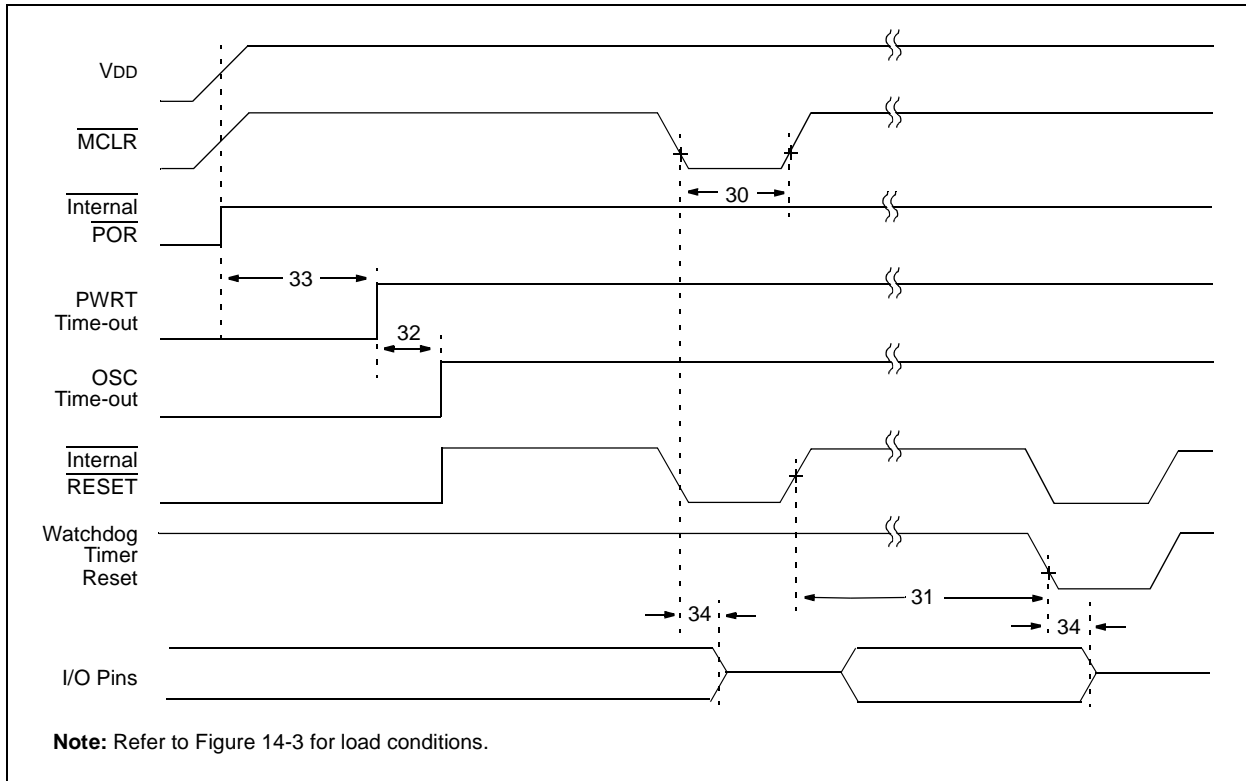
† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

†† These parameters are asynchronous events, not related to any internal clock edges.

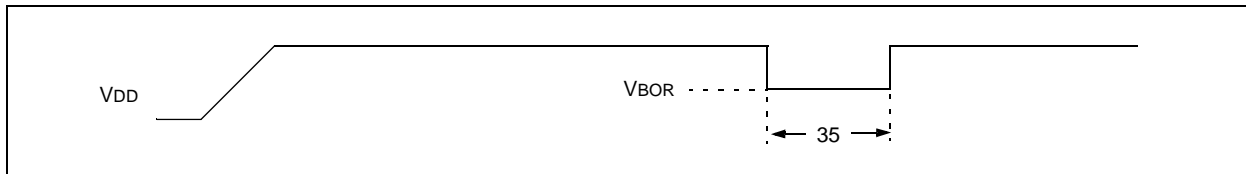
**Note 1:** Measurements are taken in RC mode, where CLKO output is 4 x T<sub>OSC</sub>.

# PIC16F72

**FIGURE 14-6: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING**



**FIGURE 14-7: BROWN-OUT RESET TIMING**



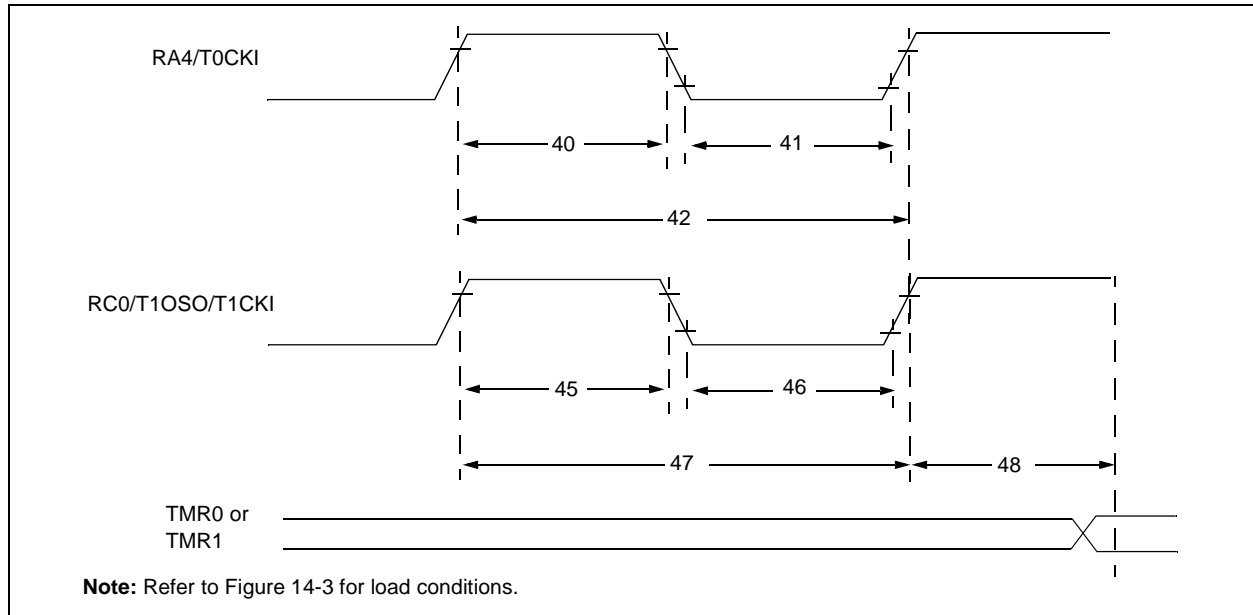
**TABLE 14-3: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER, POWER-UP TIMER, AND BROWN-OUT RESET REQUIREMENTS**

Parameter No.	Symbol	Characteristic	Min	Typ†	Max	Units	Conditions
30	TmCL	MCLR Pulse Width (low)	2	—	—	μs	VDD = 5V, -40°C to +85°C
31*	TWDT	Watchdog Timer Time-out Period (No Prescaler)	7	18	33	ms	VDD = 5V, -40°C to +85°C
32	TOST	Oscillation Start-up Timer Period	—	1024 TOSC	—	—	TOSC = OSC1 period
33*	TPWRT	Power-up Timer Period	28	72	132	ms	VDD = 5V, -40°C to +85°C
34	TIOZ	I/O Hi-impedance from MCLR Low or Watchdog Timer Reset	—	—	2.1	μs	
35	TBOR	Brown-out Reset Pulse Width	100	—	—	μs	VDD ≤ VBOR (D005)

\* These parameters are characterized but not tested.

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

**FIGURE 14-8: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS**



**TABLE 14-4: TIMER0 AND TIMER1 EXTERNAL CLOCK REQUIREMENTS**

Param No.	Symbol	Characteristic		Min	Typ†	Max	Units	Conditions	
40*	Tt0H	T0CKI High Pulse Width	No Prescaler	$0.5 T_{CY} + 20$	—	—	ns	Must also meet parameter 42	
			With Prescaler	10	—	—	ns		
41*	Tt0L	T0CKI Low Pulse Width	No Prescaler	$0.5 T_{CY} + 20$	—	—	ns	Must also meet parameter 42	
			With Prescaler	10	—	—	ns		
42*	Tt0P	T0CKI Period	No Prescaler	$T_{CY} + 40$	—	—	ns	N = prescale value (2, 4, ..., 256)	
			With Prescaler	Greater of: $20$ or $\frac{T_{CY} + 40}{N}$	—	—	ns		
45*	Tt1H	T1CKI High Time	Synchronous, Prescaler = 1	$0.5 T_{CY} + 20$	—	—	ns	Must also meet parameter 47	
			Synchronous, Prescaler = 2,4,8	Standard(F)	15	—	—		ns
				Extended(LF)	25	—	—		ns
			Asynchronous	Standard(F)	30	—	—		ns
Extended(LF)	50	—		—	ns				
46*	Tt1L	T1CKI Low Time	Synchronous, Prescaler = 1	$0.5 T_{CY} + 20$	—	—	ns	Must also meet parameter 47	
			Synchronous, Prescaler = 2,4,8	Standard(F)	15	—	—		ns
				Extended(LF)	25	—	—		ns
			Asynchronous	Standard(F)	30	—	—		ns
Extended(LF)	50	—		—	ns				
47*	Tt1P	T1CKI Input Period	Synchronous	Standard(F)	Greater of: $30$ or $\frac{T_{CY} + 40}{N}$	—	—	ns	N = prescale value (1, 2, 4, 8)
				Extended(LF)	Greater of: $50$ or $\frac{T_{CY} + 40}{N}$	—	—	ns	N = prescale value (1, 2, 4, 8)
			Asynchronous	Standard(F)	60	—	—	ns	
				Extended(LF)	100	—	—	ns	
	Ft1	Timer1 Oscillator Input Frequency Range (oscillator enabled by setting bit T1OSCEN)		DC	—	200	kHz		
48	TCKEZtmr1	Delay from External Clock Edge to Timer Increment		$2 T_{OSC}$	—	$7 T_{OSC}$	—		

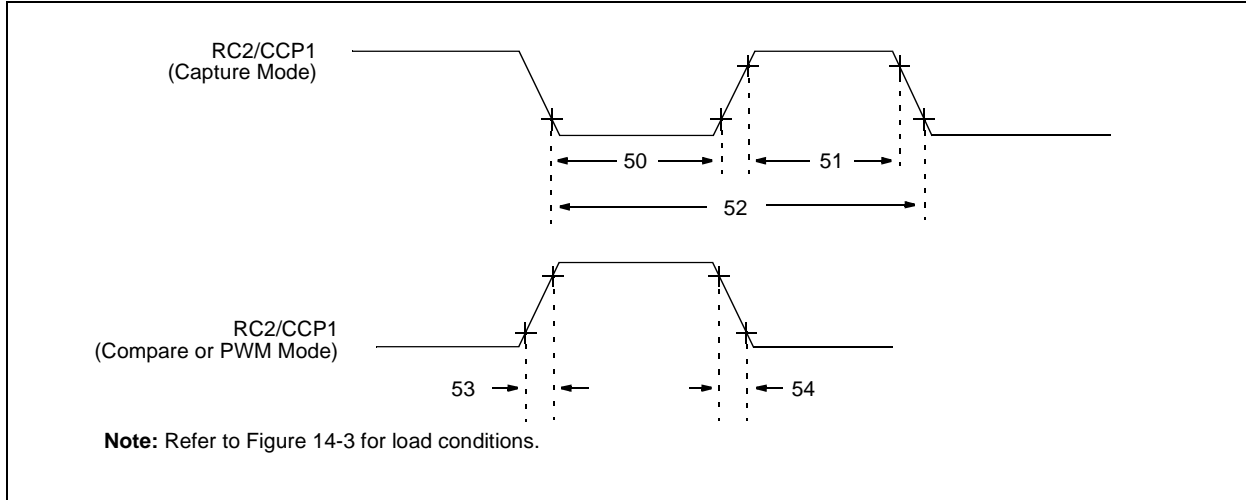
\* These parameters are characterized but not tested.

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



# PIC16F72

**FIGURE 14-9: CAPTURE/COMPARE/PWM TIMINGS (CCP1)**



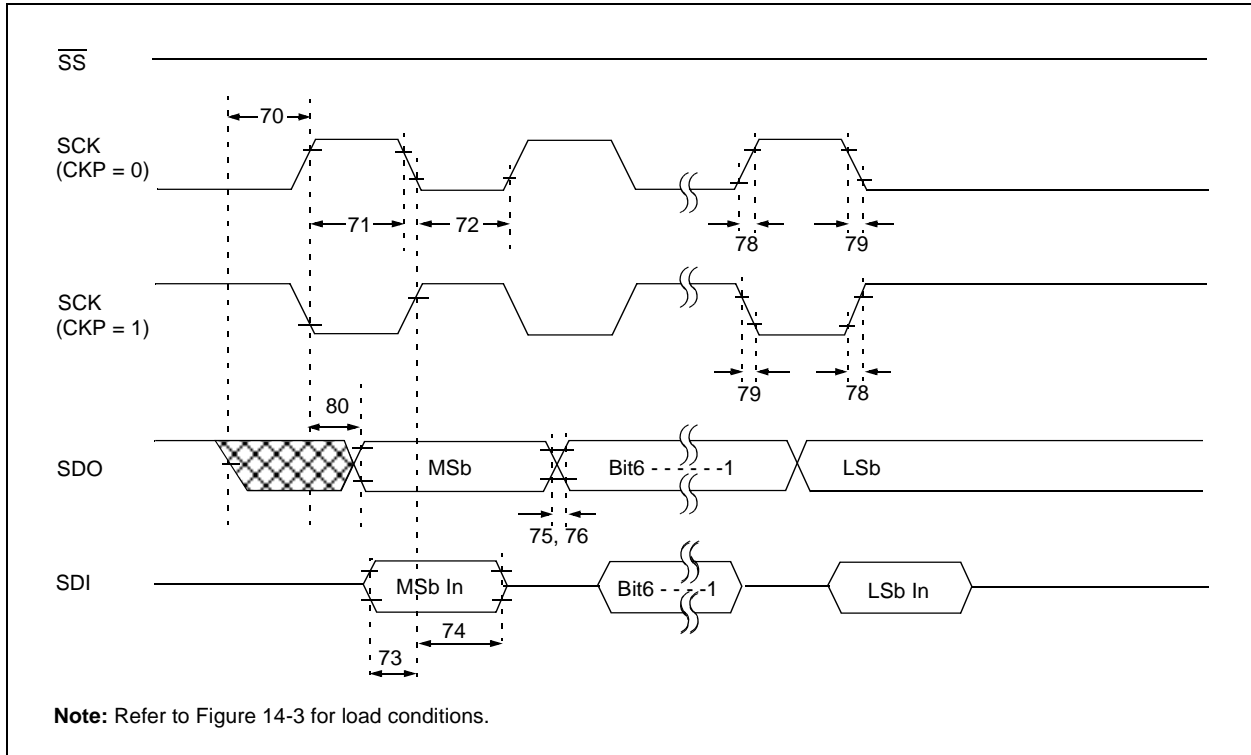
**TABLE 14-5: CAPTURE/COMPARE/PWM REQUIREMENTS (CCP1)**

Param No.	Symbol	Characteristic		Min	Typ†	Max	Units	Conditions	
50*	TccL	CCP1 input low time	No Prescaler	$0.5 T_{CY} + 20$	—	—	ns		
			With Prescaler	Standard(F)	10	—	—		ns
				Extended(LF)	20	—	—		ns
51*	TccH	CCP1 input high time	No Prescaler	$0.5 T_{CY} + 20$	—	—	ns		
			With Prescaler	Standard(F)	10	—	—		ns
				Extended(LF)	20	—	—		ns
52*	TccP	CCP1 input period		$\frac{3 T_{CY} + 40}{N}$	—	—	ns	N = prescale value (1,4 or 16)	
53*	TccR	CCP1 output rise time	Standard(F)	—	10	25	ns		
			Extended(LF)	—	25	50	ns		
54*	TccF	CCP1 output fall time	Standard(F)	—	10	25	ns		
			Extended(LF)	—	25	45	ns		

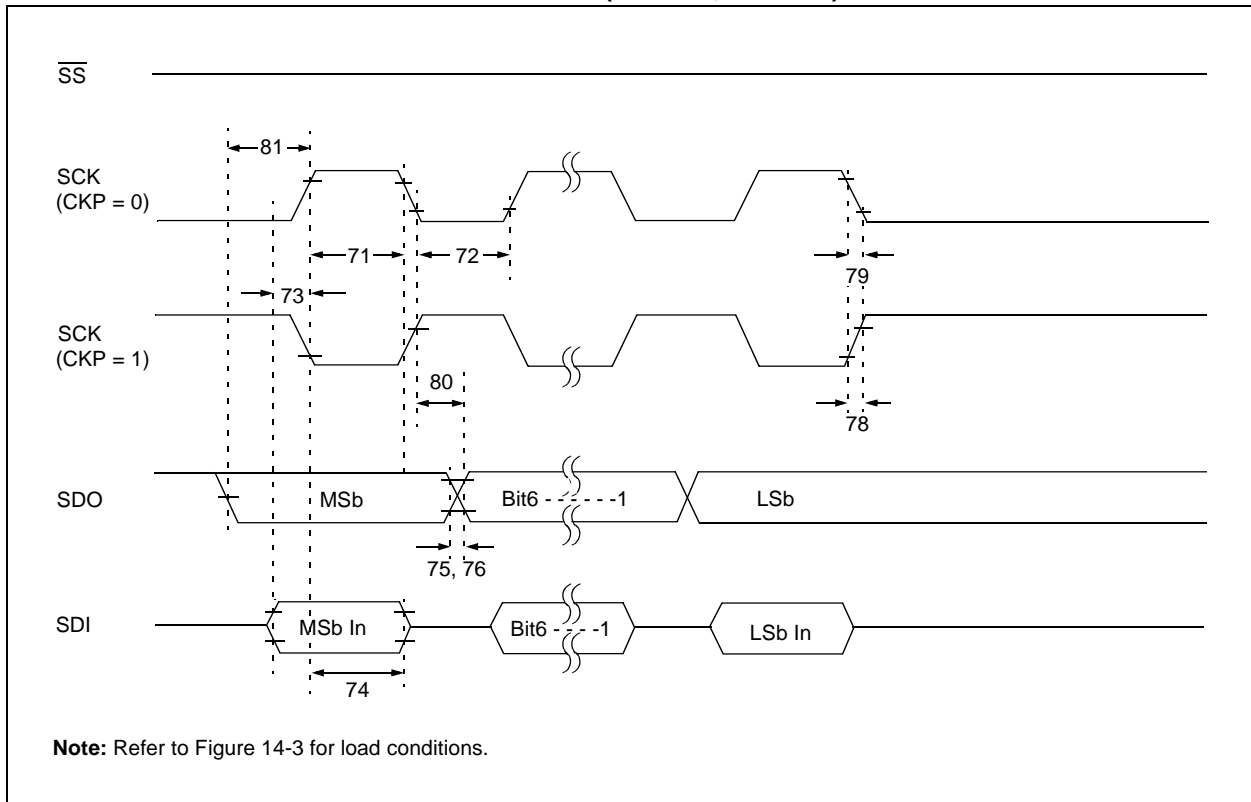
\* These parameters are characterized but not tested.

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

**FIGURE 14-10: SPI MASTER MODE TIMING (CKE = 0, SMP = 0)**

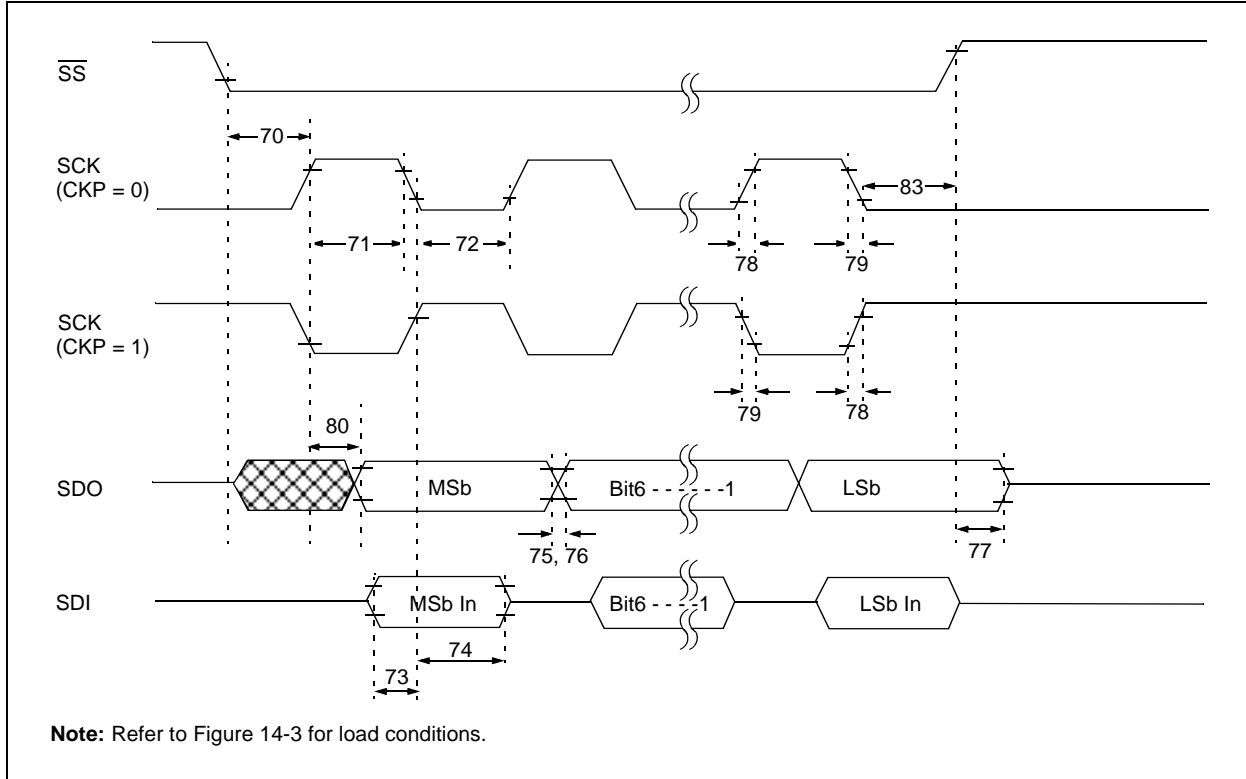


**FIGURE 14-11: SPI MASTER MODE TIMING (CKE = 1, SMP = 1)**

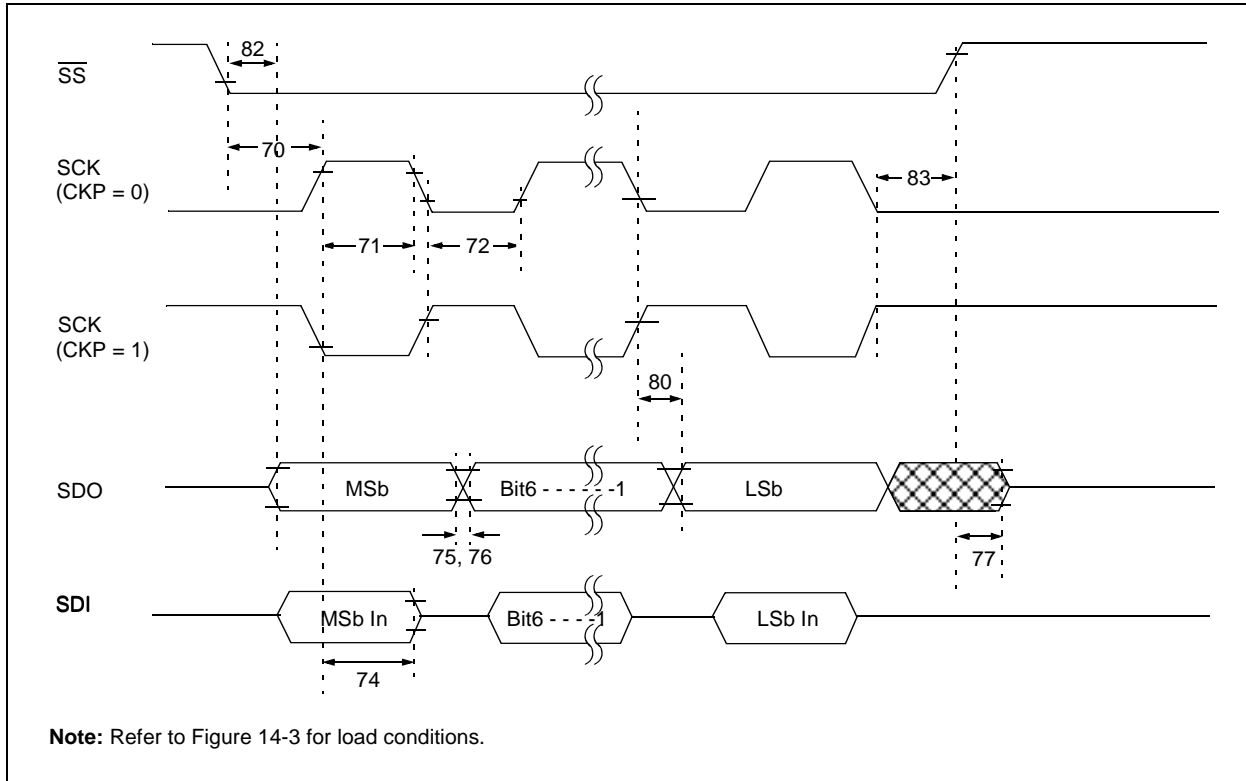


# PIC16F72

**FIGURE 14-12: SPI SLAVE MODE TIMING (CKE = 0)**



**FIGURE 14-13: SPI SLAVE MODE TIMING (CKE = 1)**



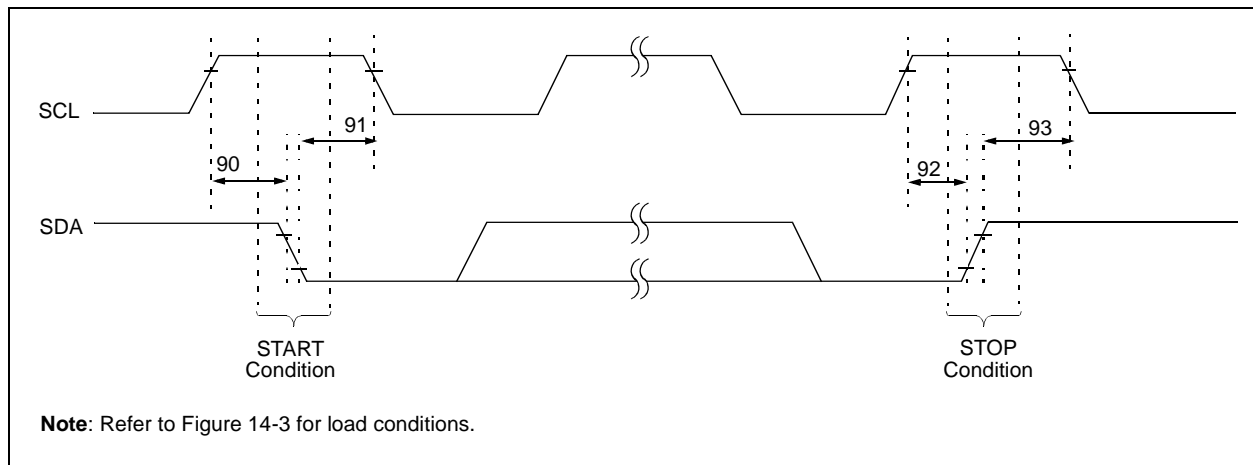
**TABLE 14-6: SPI MODE REQUIREMENTS**

Param No.	Symbol	Characteristic	Min	Typ†	Max	Units	Conditions
70*	TssL2scH, TssL2scL	$\overline{SS}\downarrow$ to SCK $\downarrow$ or SCK $\uparrow$ input	T <sub>CY</sub>	—	—	ns	
71*	TscH	SCK input high time (Slave mode)	T <sub>CY</sub> + 20	—	—	ns	
72*	TscL	SCK input low time (Slave mode)	T <sub>CY</sub> + 20	—	—	ns	
73*	TdiV2scH, TdiV2scL	Setup time of SDI data input to SCK edge	100	—	—	ns	
74*	Tsch2diL, TscL2diL	Hold time of SDI data input to SCK edge	100	—	—	ns	
75*	TdoR	SDO data output rise time	Standard(F) Extended(LF)	10 25	25 50	ns ns	
76*	TdoF	SDO data output fall time	—	10	25	ns	
77*	TssH2doZ	$\overline{SS}\uparrow$ to SDO output hi-impedance	10	—	50	ns	
78*	TscR	SCK output rise time (Master mode)	Standard(F) Extended(LF)	10 25	25 50	ns ns	
79*	TscF	SCK output fall time (Master mode)	—	10	25	ns	
80*	Tsch2doV, TscL2doV	SDO data output valid after SCK edge	Standard(F) Extended(LF)	— —	50 145	ns ns	
81*	TdoV2scH, TdoV2scL	SDO data output setup to SCK edge	T <sub>CY</sub>	—	—	ns	
82*	TssL2doV	SDO data output valid after $\overline{SS}\downarrow$ edge	—	—	50	ns	
83*	Tsch2ssH, TscL2ssH	$\overline{SS}\uparrow$ after SCK edge	1.5 T <sub>CY</sub> + 40	—	—	ns	

\* These parameters are characterized but not tested.

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

**FIGURE 14-14: I<sup>2</sup>C BUS START/STOP BITS TIMING**



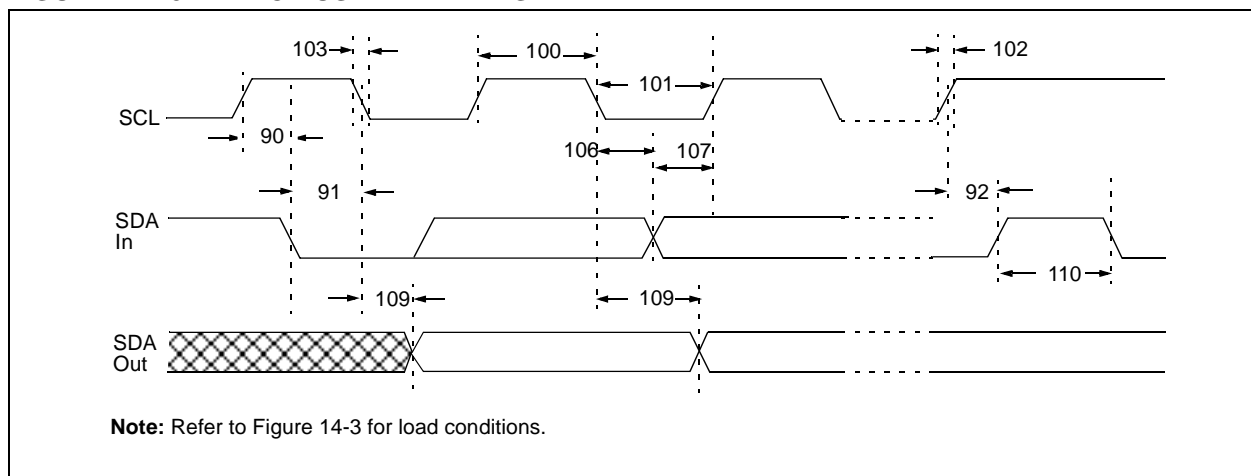
# PIC16F72

**TABLE 14-7: I<sup>2</sup>C BUS START/STOP BITS REQUIREMENTS**

Param No.	Symbol	Characteristic	Min	Typ	Max	Units	Conditions	
90*	TSU:STA	START condition	100 kHz mode	4700	—	—	ns	Only relevant for Repeated START condition
		Setup time	400 kHz mode	600	—	—		
91*	THD:STA	START condition	100 kHz mode	4000	—	—	ns	After this period, the first clock pulse is generated
		Hold time	400 kHz mode	600	—	—		
92*	TSU:STO	STOP condition	100 kHz mode	4700	—	—	ns	
		Setup time	400 kHz mode	600	—	—		
93	THD:STO	STOP condition	100 kHz mode	4000	—	—	ns	
		Hold time	400 kHz mode	600	—	—		

\* These parameters are characterized but not tested.

**FIGURE 14-15: I<sup>2</sup>C BUS DATA TIMING**



**TABLE 14-8: I<sup>2</sup>C BUS DATA REQUIREMENTS**

Param No.	Symbol	Characteristic	Min	Max	Units	Conditions	
100*	THIGH	Clock High Time	100 kHz mode	4.0	—	μs	Device must operate at a minimum of 1.5 MHz
			400 kHz mode	0.6	—	μs	Device must operate at a minimum of 10 MHz
			SSP Module	1.5 T <sub>CY</sub>	—		
101*	TLOW	Clock Low Time	100 kHz mode	4.7	—	μs	Device must operate at a minimum of 1.5 MHz
			400 kHz mode	1.3	—	μs	Device must operate at a minimum of 10 MHz
			SSP Module	1.5 T <sub>CY</sub>	—		
102*	TR	SDA and SCL Rise Time	100 kHz mode	—	1000	ns	
			400 kHz mode	20 + 0.1 C <sub>B</sub>	300	ns	C <sub>B</sub> is specified to be from 10 - 400 pF
103*	TF	SDA and SCL Fall Time	100 kHz mode	—	300	ns	
			400 kHz mode	20 + 0.1 C <sub>B</sub>	300	ns	C <sub>B</sub> is specified to be from 10 - 400 pF
90*	TSU:STA	START Condition Setup Time	100 kHz mode	4.7	—	μs	Only relevant for Repeated START condition
			400 kHz mode	0.6	—	μs	
91*	THD:STA	START Condition Hold Time	100 kHz mode	4.0	—	μs	After this period, the first clock pulse is generated
			400 kHz mode	0.6	—	μs	
106*	THD:DAT	Data Input Hold Time	100 kHz mode	0	—	ns	
			400 kHz mode	0	0.9	μs	
107*	TSU:DAT	Data Input Setup Time	100 kHz mode	250	—	ns	<b>(Note 2)</b>
			400 kHz mode	100	—	ns	
92*	TSU:STO	STOP Condition Setup Time	100 kHz mode	4.7	—	μs	
			400 kHz mode	0.6	—	μs	
109*	TAA	Output Valid from Clock	100 kHz mode	—	3500	ns	<b>(Note 1)</b>
			400 kHz mode	—	—	ns	
110*	TBUF	Bus Free Time	100 kHz mode	4.7	—	μs	Time the bus must be free before a new transmission can start
			400 kHz mode	1.3	—	μs	
	C <sub>B</sub>	Bus Capacitive Loading	—	400	pF		

\* These parameters are characterized but not tested.

**Note 1:** As a transmitter, the device must provide this internal minimum delay time to bridge the undefined region (min. 300 ns) of the falling edge of SCL to avoid unintended generation of START or STOP conditions.

**2:** A Fast mode (400 kHz) I<sup>2</sup>C bus device can be used in a Standard mode (100 kHz) I<sup>2</sup>C bus system, but the requirement TSU:DAT ≥ 250 ns must then be met. This will automatically be the case if the device does not stretch the LOW period of the SCL signal. If such a device does stretch the LOW period of the SCL signal, it must output the next data bit to the SDA line TR max.+TSU:DAT = 1000 + 250 = 1250 ns (according to the Standard mode I<sup>2</sup>C bus specification), before the SCL line is released.

# PIC16F72

**TABLE 14-9: A/D CONVERTER CHARACTERISTICS: PIC16F72 (INDUSTRIAL)  
PIC16LF72 (INDUSTRIAL)**

Param No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
A01	NR	Resolution	PIC16F72	—	—	8 bits	bit	$V_{REF} = V_{DD} = 5.12V$ , $V_{SS} \leq V_{AIN} \leq V_{REF}$
			PIC16LF72	—	—	8 bits	bit	$V_{REF} = V_{DD} = 2.2V$
A02	EABS	Total Absolute Error		—	—	$< \pm 1$	LSb	$V_{REF} = V_{DD} = 5.12V$ , $V_{SS} \leq V_{AIN} \leq V_{REF}$
A03	EIL	Integral Linearity Error		—	—	$< \pm 1$	LSb	$V_{REF} = V_{DD} = 5.12V$ , $V_{SS} \leq V_{AIN} \leq V_{REF}$
A04	EDL	Differential Linearity Error		—	—	$< \pm 1$	LSb	$V_{REF} = V_{DD} = 5.12V$ , $V_{SS} \leq V_{AIN} \leq V_{REF}$
A05	EFS	Full Scale Error		—	—	$< \pm 1$	LSb	$V_{REF} = V_{DD} = 5.12V$ , $V_{SS} \leq V_{AIN} \leq V_{REF}$
A06	E0FF	Offset Error		—	—	$< \pm 1$	LSb	$V_{REF} = V_{DD} = 5.12V$ , $V_{SS} \leq V_{AIN} \leq V_{REF}$
A10	—	Monotonicity ( <b>Note 3</b> )		—	guaranteed	—	—	$V_{SS} \leq V_{AIN} \leq V_{REF}$
A20	VREF	Reference Voltage		2.5	—	$V_{DD}+0.3$	V	$-40^{\circ}C$ to $+85^{\circ}C$
				2.2	—	$V_{DD}+0.3$	V	$0^{\circ}C$ to $+85^{\circ}C$
A25	VAIN	Analog Input Voltage		$V_{SS} - 0.3$	—	$V_{REF} + 0.3$	V	
A30	ZAIN	Recommended Impedance of Analog Voltage Source		—	—	10.0	k $\Omega$	
A40	IAD	A/D Conversion Current ( $V_{DD}$ )	PIC16F72	—	180	—	$\mu A$	Average current consumption when A/D is on ( <b>Note 1</b> ).
			PIC16LF72	—	90	—	$\mu A$	
A50	IREF	VREF input current ( <b>Note 2</b> )		N/A	—	$\pm 5$	$\mu A$	During VAIN acquisition. During A/D Conversion cycle.
				—	—	500	$\mu A$	

\* These parameters are characterized but not tested.

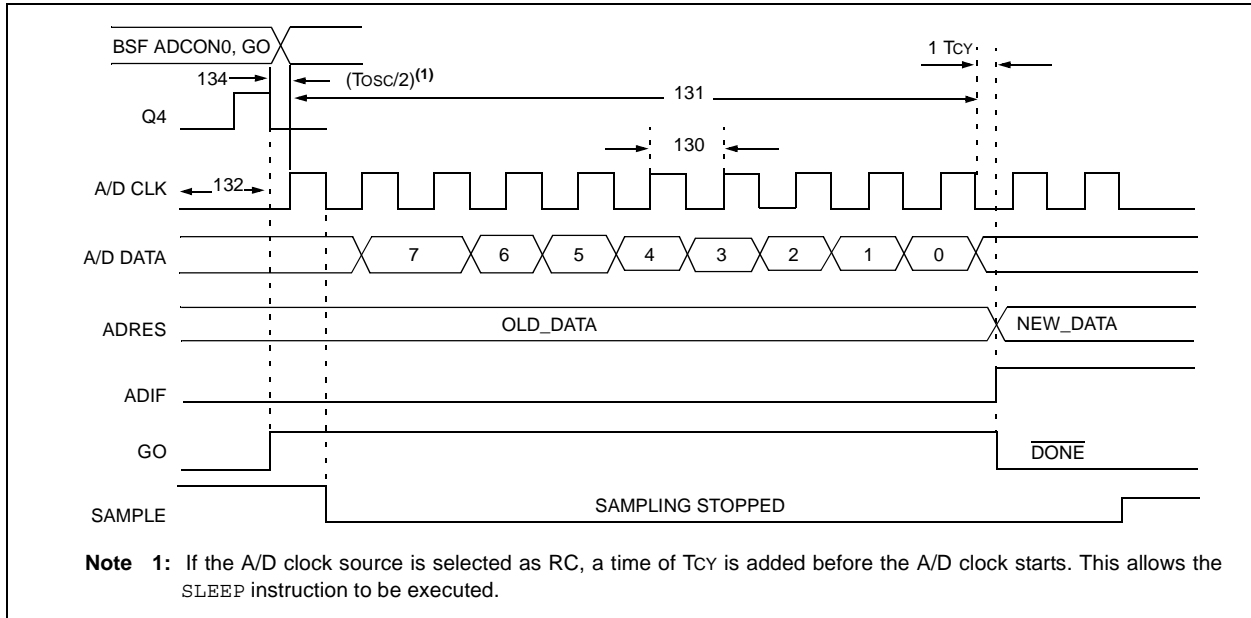
† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** When A/D is off, it will not consume any current other than minor leakage current. The power-down current spec includes any such leakage from the A/D module.

**2:** VREF current is from the RA3 pin or the VDD pin, whichever is selected as a reference input.

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

**FIGURE 14-16: A/D CONVERSION TIMING**



**TABLE 14-10: A/D CONVERSION REQUIREMENTS**

Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions	
130	TAD	A/D Clock Period	PIC16F72	1.6	—	—	$\mu s$	TOSC based, $V_{REF} \geq 3.0V$
			PIC16LF72	2.0	—	—	$\mu s$	TOSC based, $2.0V \leq V_{REF} \leq 5.5V$
			PIC16F72	2.0	4.0	6.0	$\mu s$	A/D RC mode
			PIC16LF72	3.0	6.0	9.0	$\mu s$	A/D RC mode
131	TCNV	Conversion Time (not including S/H time) <b>(Note 1)</b>	9	—	9	TAD		
132	TACQ	Acquisition Time	5*	—	—	$\mu s$	The minimum time is the amplifier settling time. This may be used if the "new" input voltage has not changed by more than 1 LSb (i.e., 20.0 mV @ 5.12V) from the last sampled voltage (as stated on CHOLD).	
134	TGo	Q4 to A/D Clock Start	—	TOSC/2	—	—	If the A/D clock source is selected as RC, a time of $T_{cy}$ is added before the A/D clock starts. This allows the SLEEP instruction to be executed.	

\* These parameters are characterized but not tested.

† 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:** ADRES register may be read on the following  $T_{cy}$  cycle.

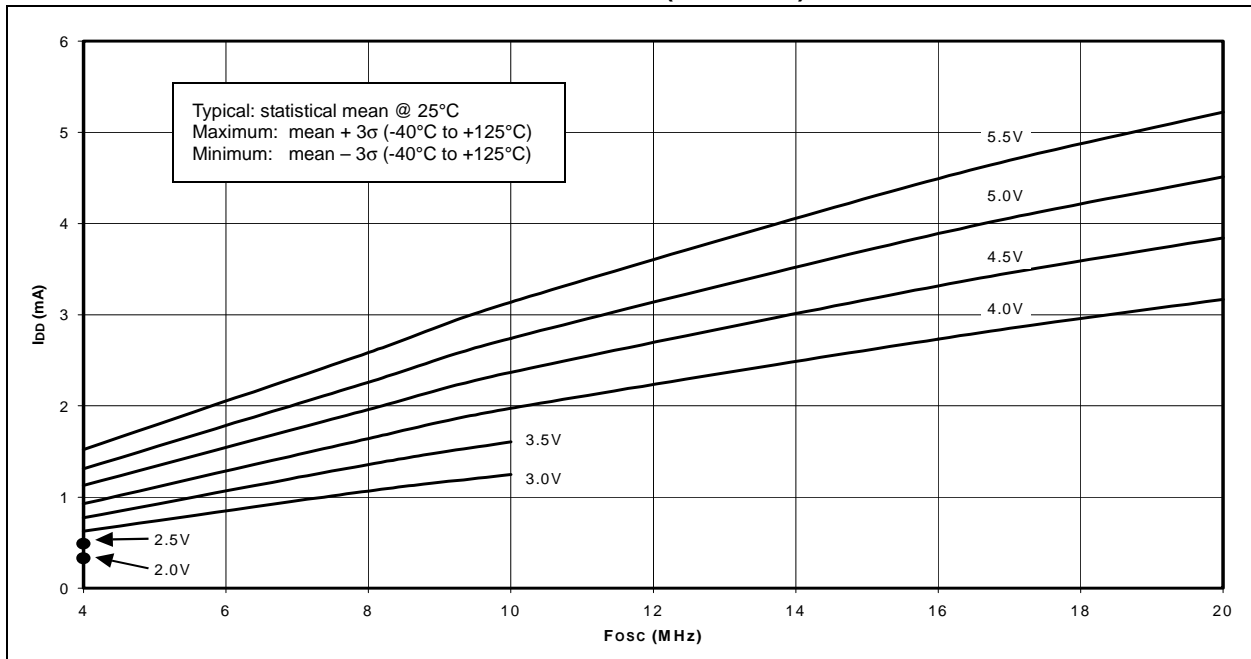


## 15.0 DC AND AC CHARACTERISTICS GRAPHS AND TABLES

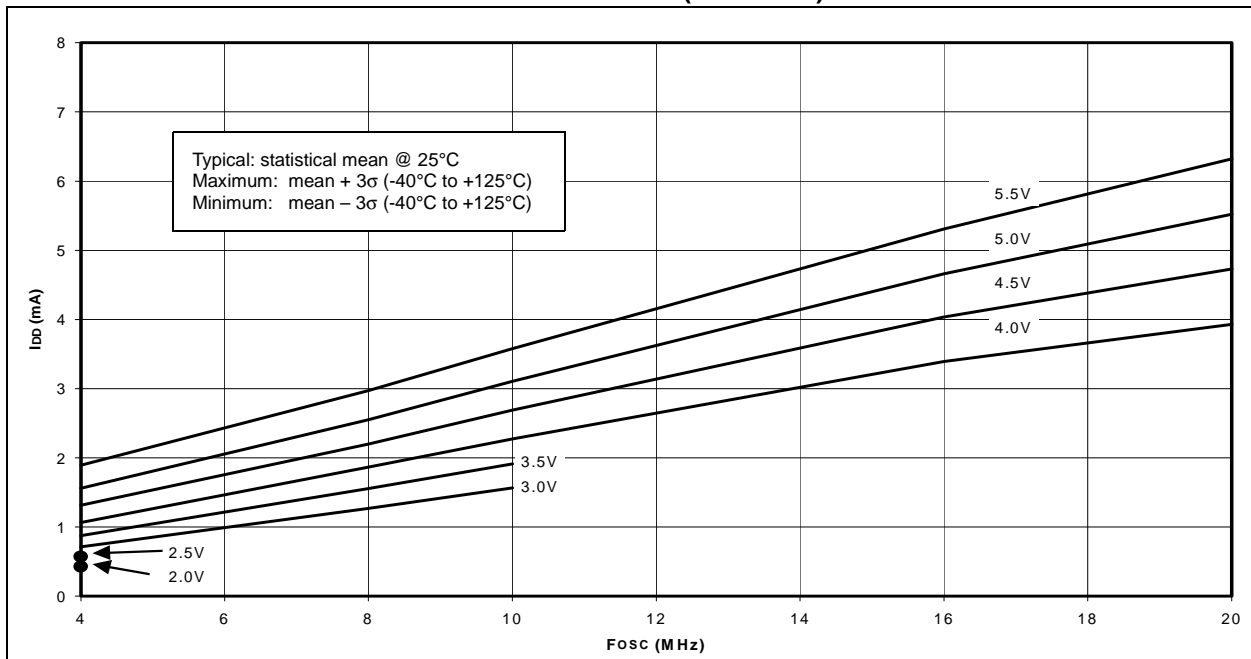
**Note:** The graphs and tables provided following this note are a statistical summary based on a limited number of samples and are provided for informational purposes only. The performance characteristics listed herein are not tested or guaranteed. In some graphs or tables, the data presented may be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

“Typical” represents the mean of the distribution at 25°C. “Maximum” or “minimum” represents (mean + 3σ) or (mean - 3σ) respectively, where σ is a standard deviation, over the whole temperature range.

**FIGURE 15-1: TYPICAL I<sub>DD</sub> vs. F<sub>osc</sub> OVER V<sub>DD</sub> (HS MODE)**



**FIGURE 15-2: MAXIMUM I<sub>DD</sub> vs. F<sub>osc</sub> OVER V<sub>DD</sub> (HS MODE)**



# PIC16F72

FIGURE 15-3: TYPICAL  $I_{DD}$  vs.  $F_{osc}$  OVER  $V_{DD}$  (XT MODE)

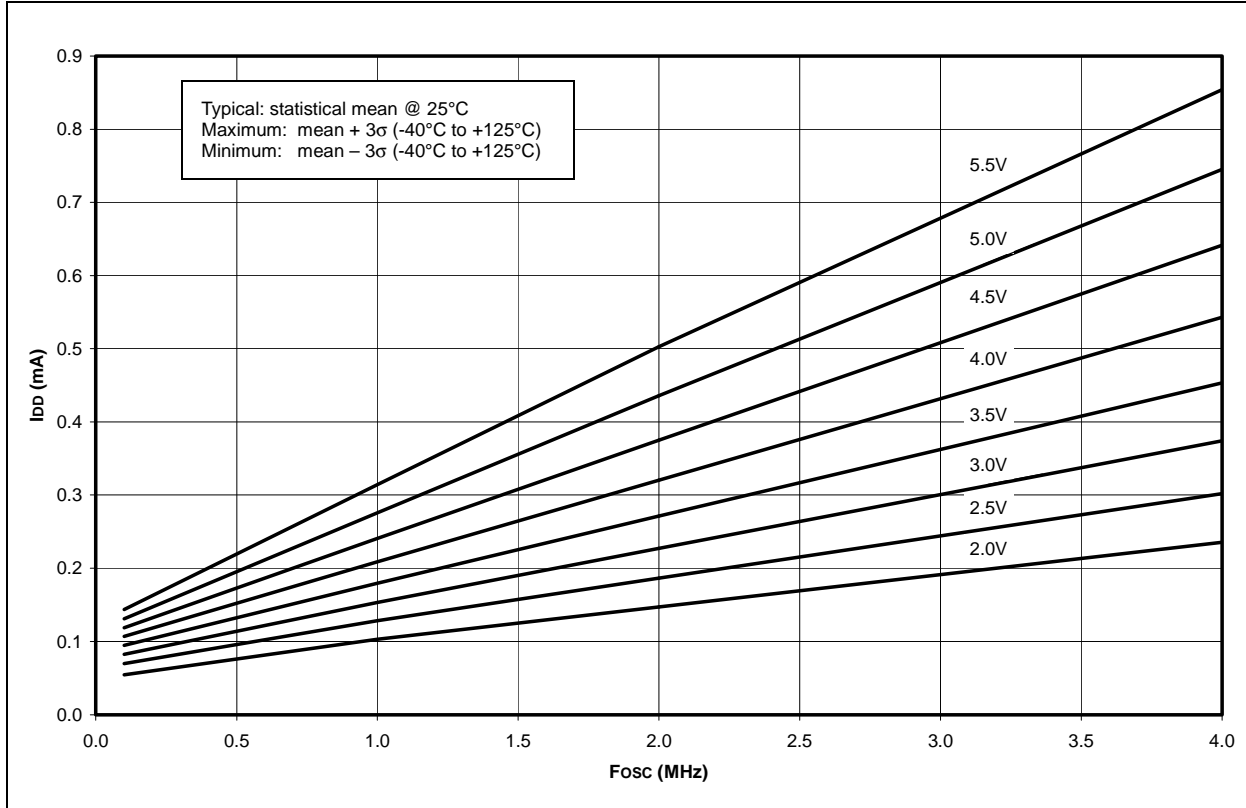
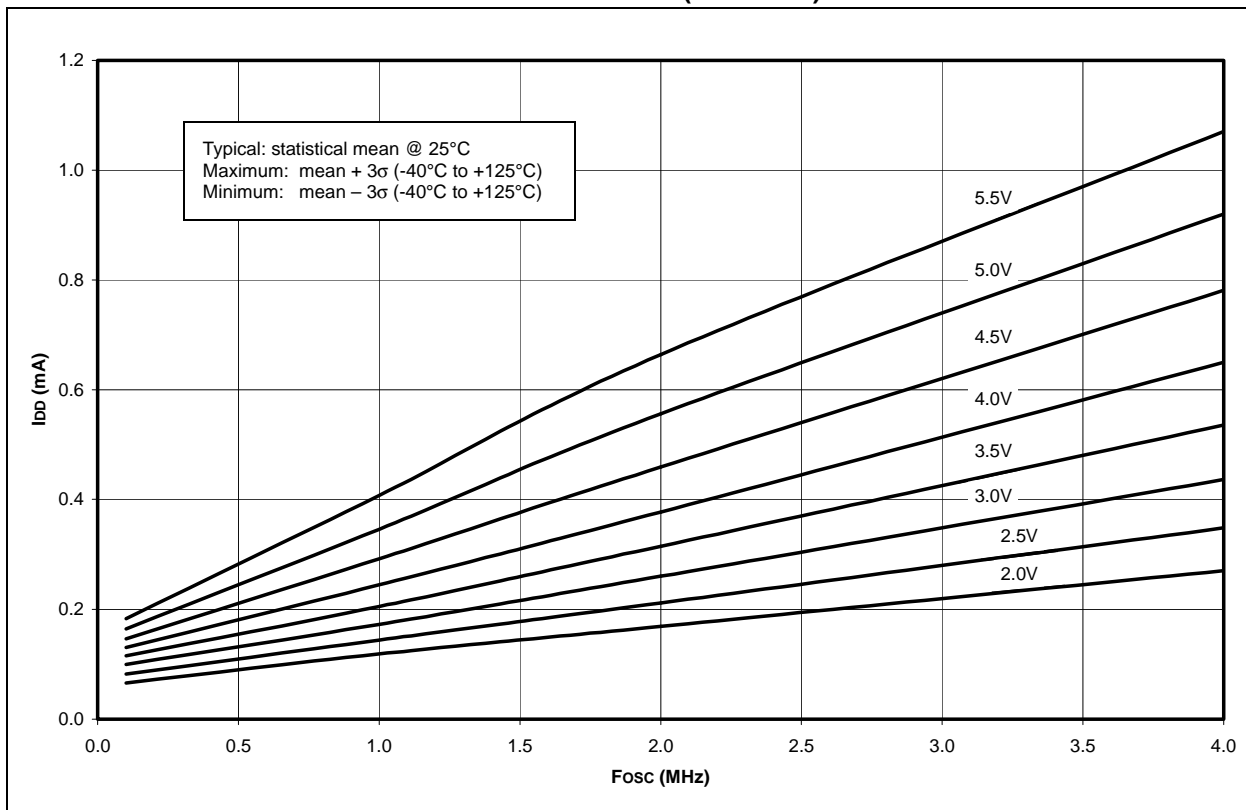
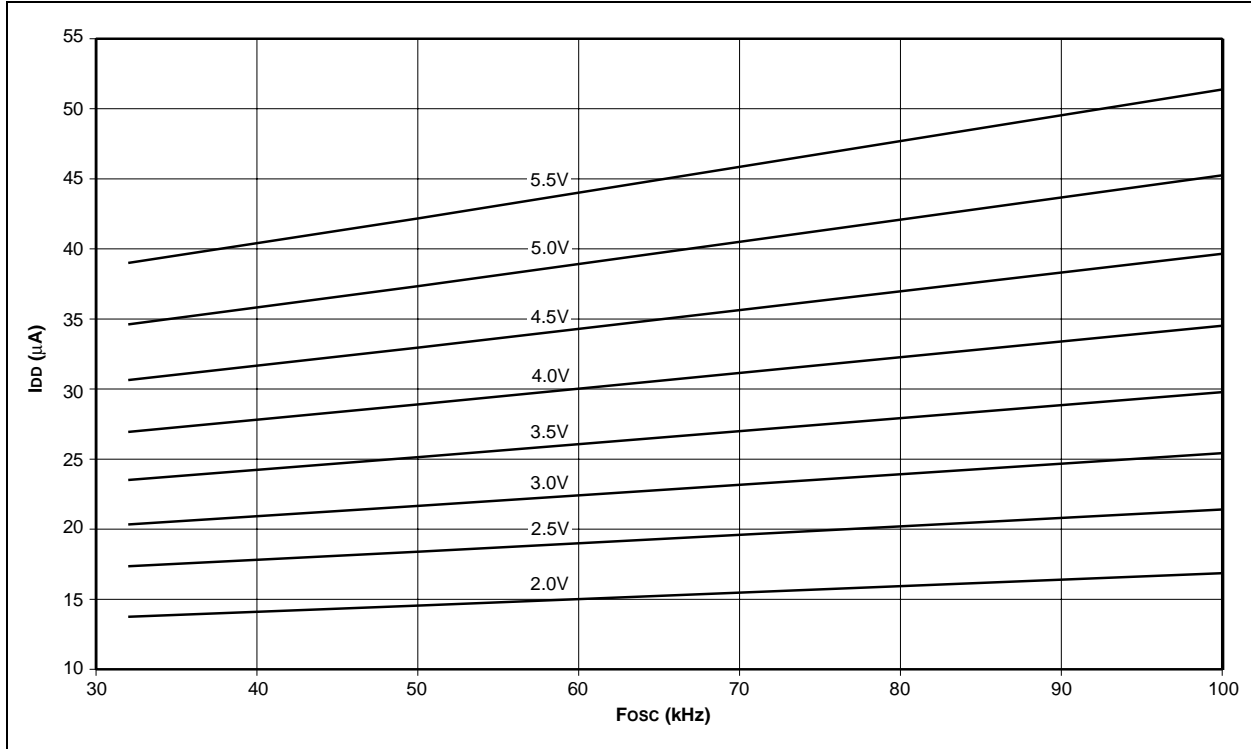


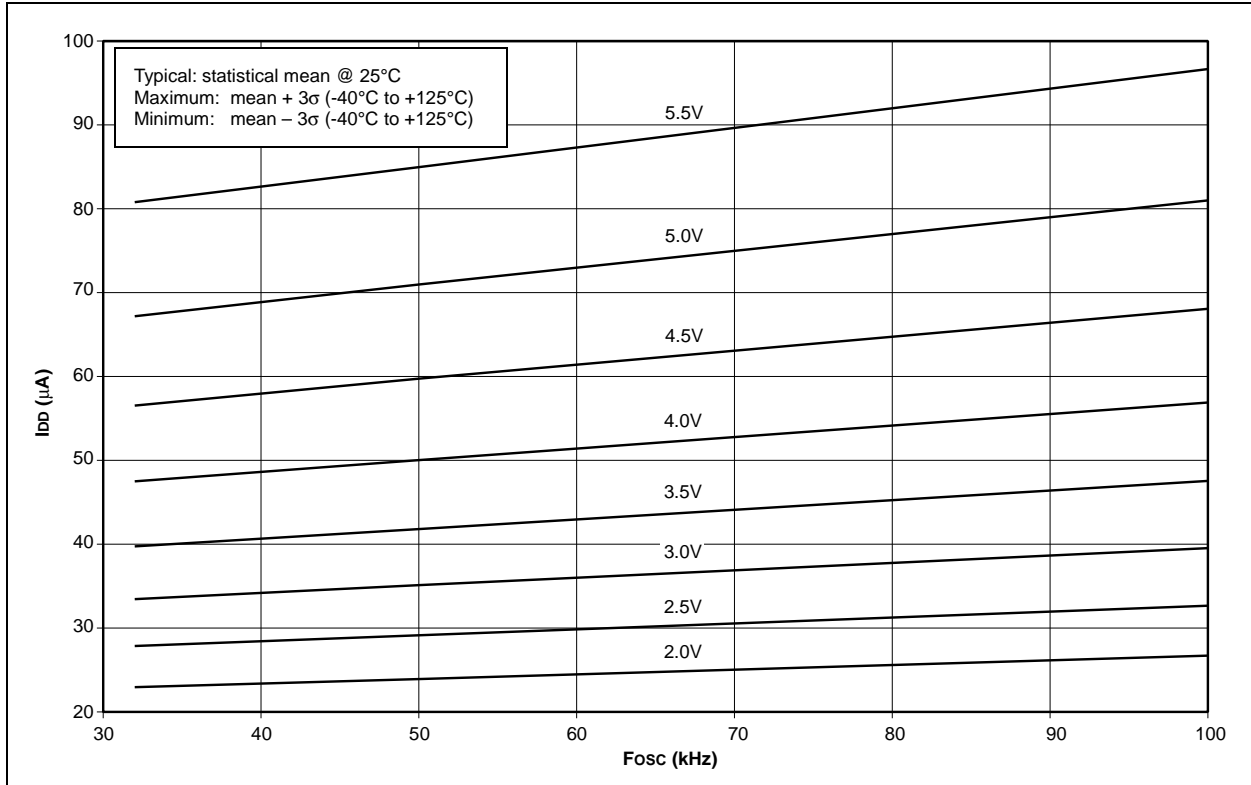
FIGURE 15-4: MAXIMUM  $I_{DD}$  vs.  $F_{osc}$  OVER  $V_{DD}$  (XT MODE)



**FIGURE 15-5: TYPICAL  $I_{DD}$  vs.  $F_{OSC}$  OVER  $V_{DD}$  (LP MODE)**

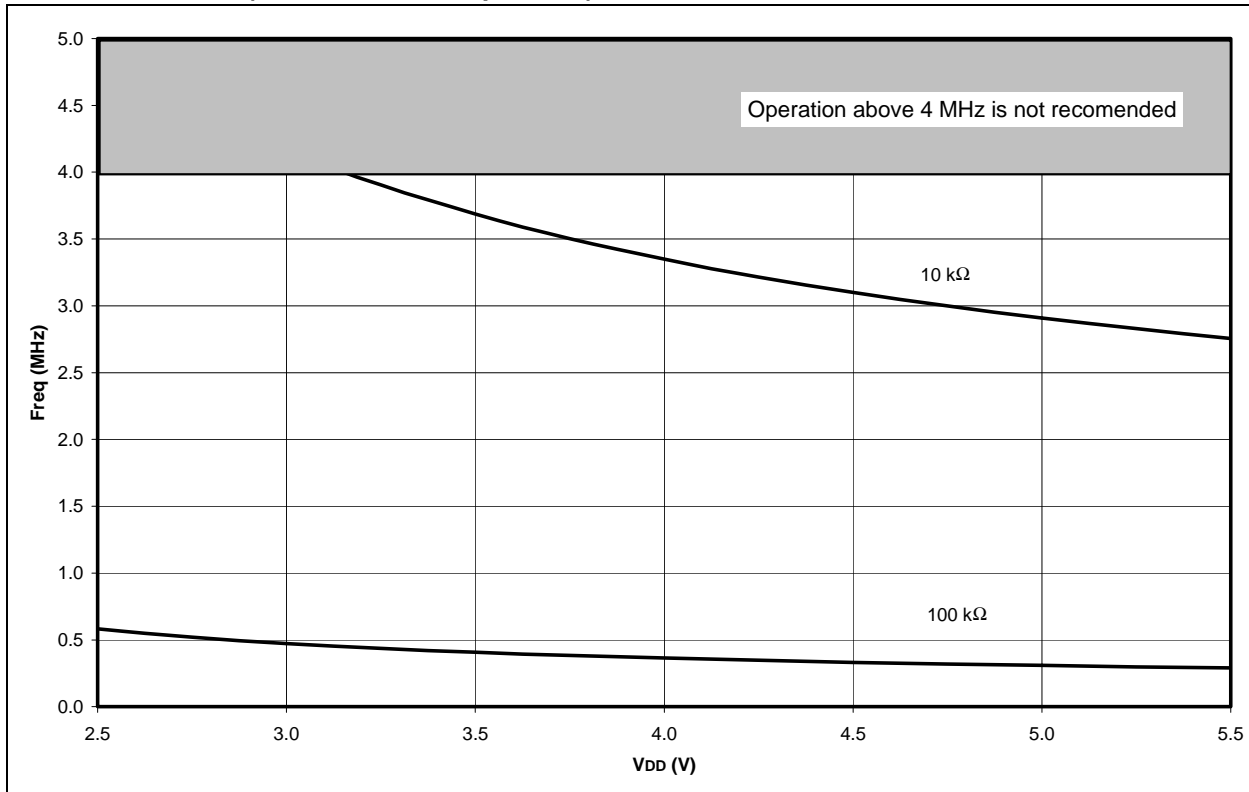


**FIGURE 15-6: MAXIMUM  $I_{DD}$  vs.  $F_{OSC}$  OVER  $V_{DD}$  (LP MODE)**

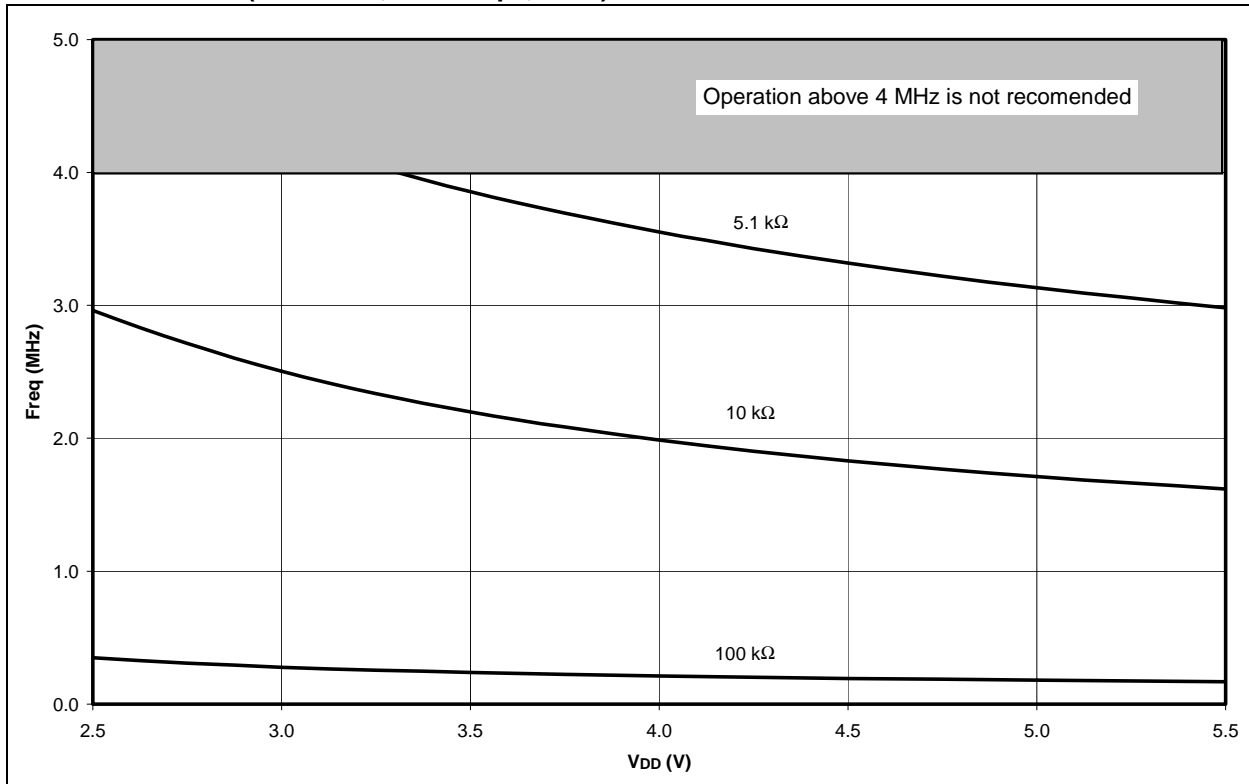


# PIC16F72

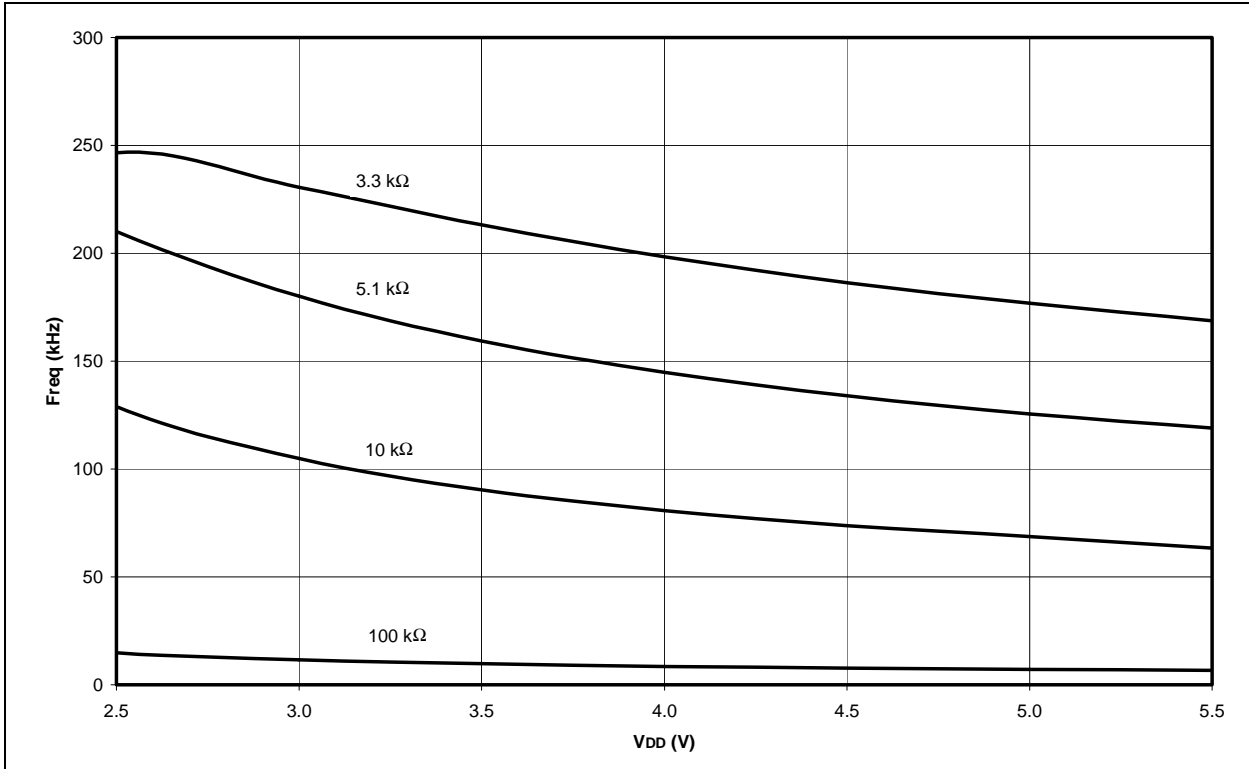
**FIGURE 15-7: AVERAGE  $F_{osc}$  vs.  $V_{DD}$  FOR VARIOUS VALUES OF R (RC MODE, C = 20 pF, 25°C)**



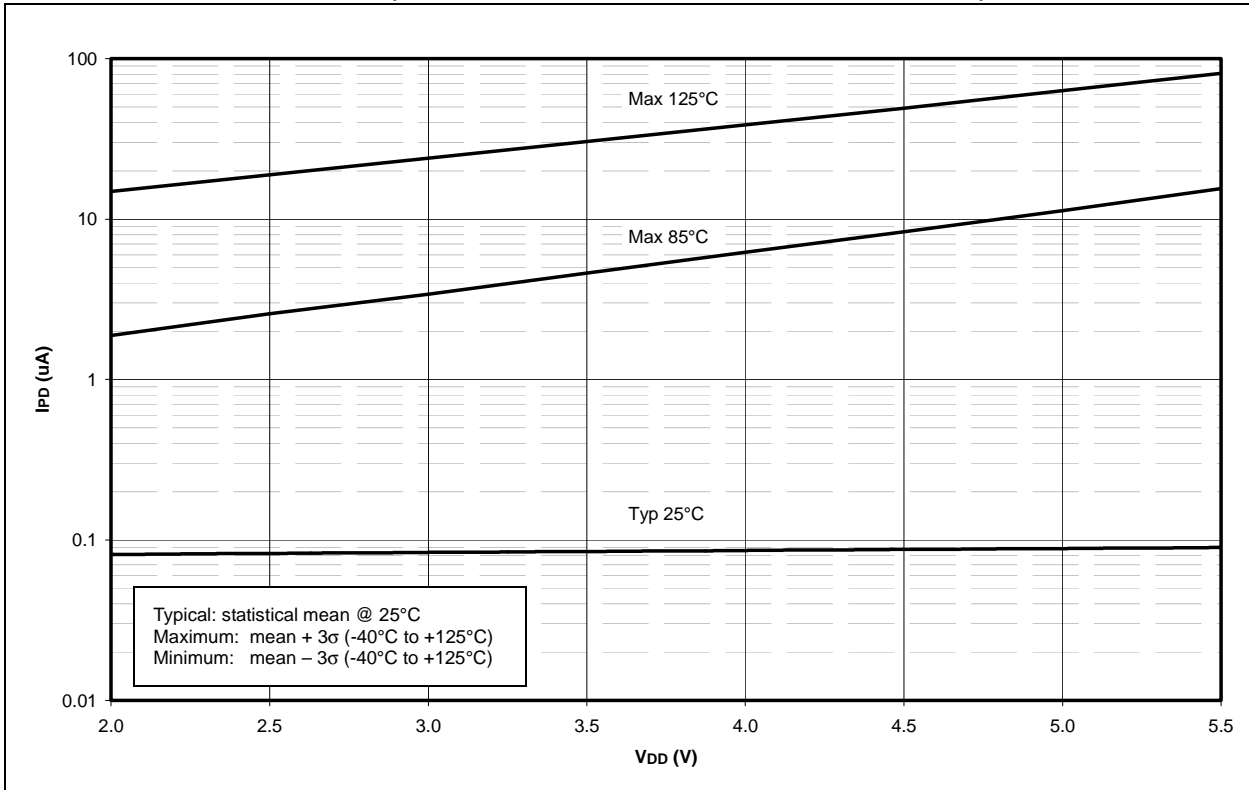
**FIGURE 15-8: AVERAGE  $F_{osc}$  vs.  $V_{DD}$  FOR VARIOUS VALUES OF R (RC MODE, C = 100 pF, 25°C)**



**FIGURE 15-9: AVERAGE Fosc vs. VDD FOR VARIOUS VALUES OF R (RC MODE, C = 300 pF, 25°C)**

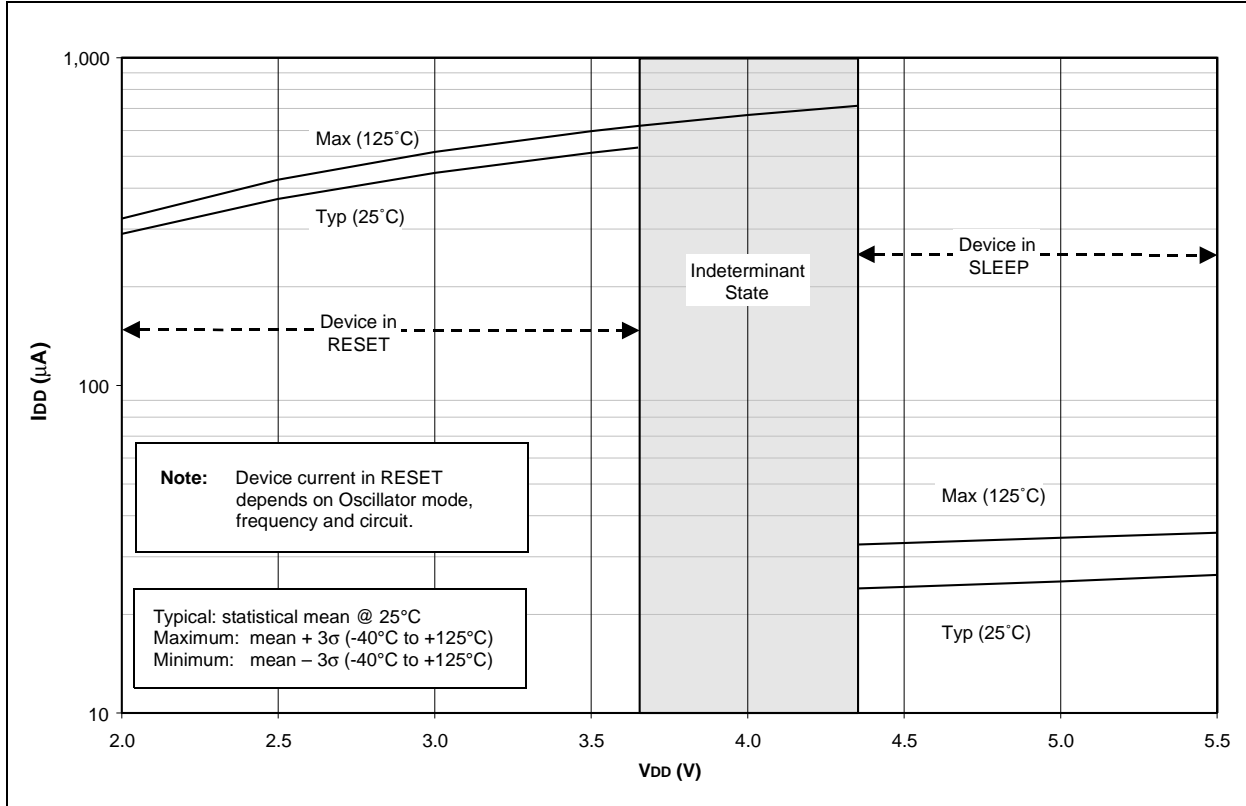


**FIGURE 15-10: IPD vs. VDD (SLEEP MODE, ALL PERIPHERALS DISABLED)**

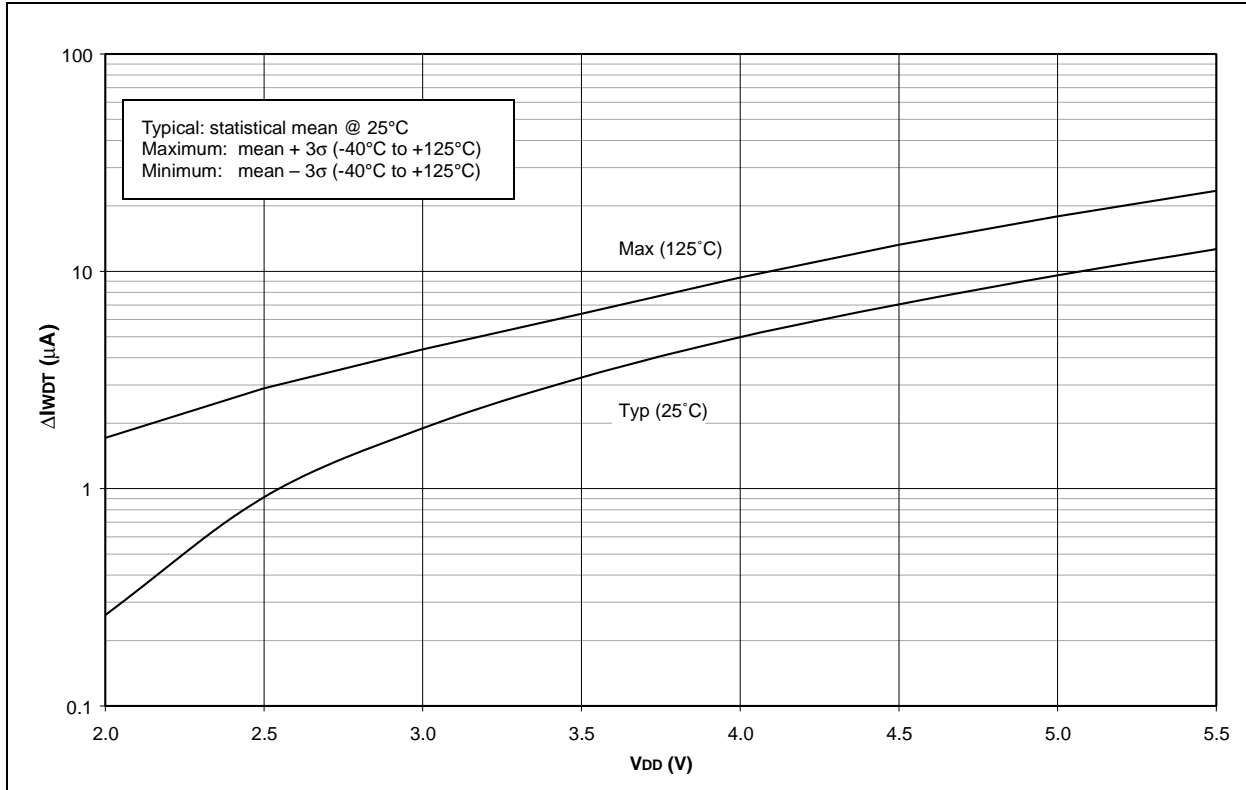


# PIC16F72

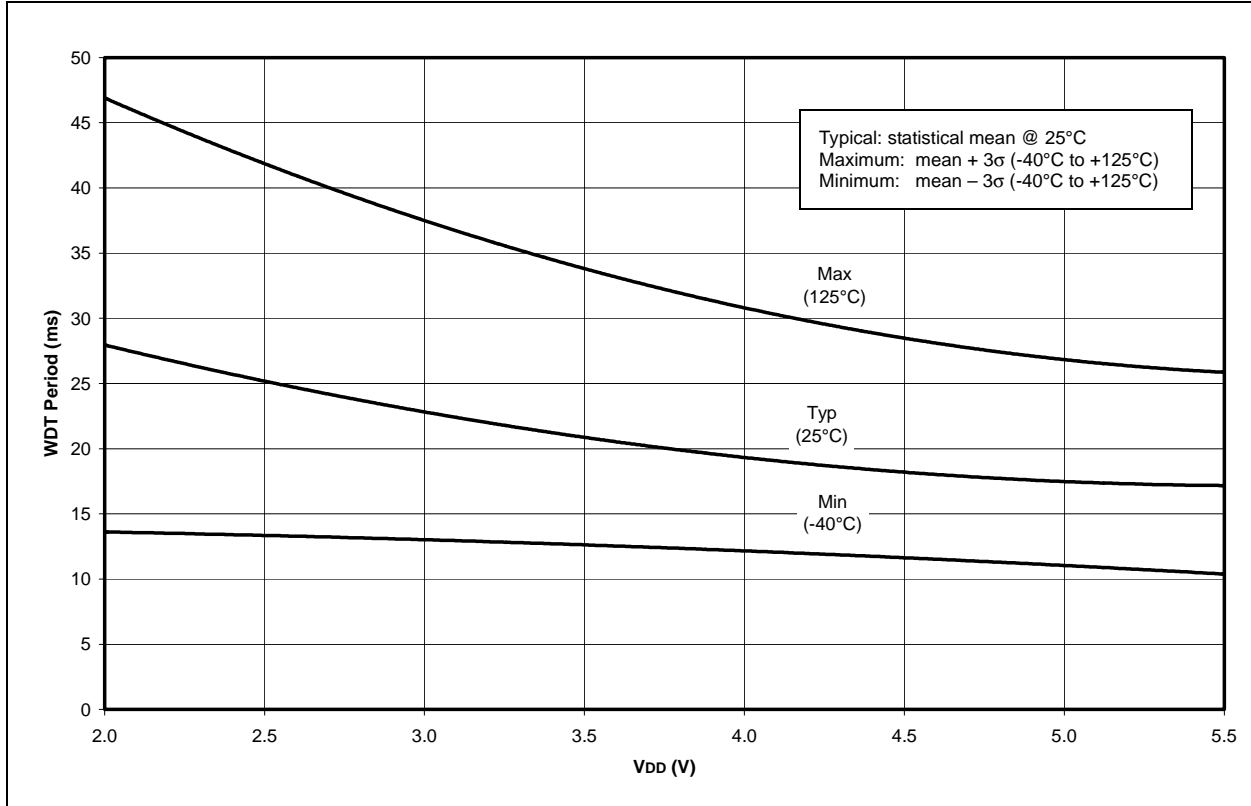
**FIGURE 15-11:  $\Delta I_{BOR}$  vs.  $V_{DD}$  OVER TEMPERATURE**



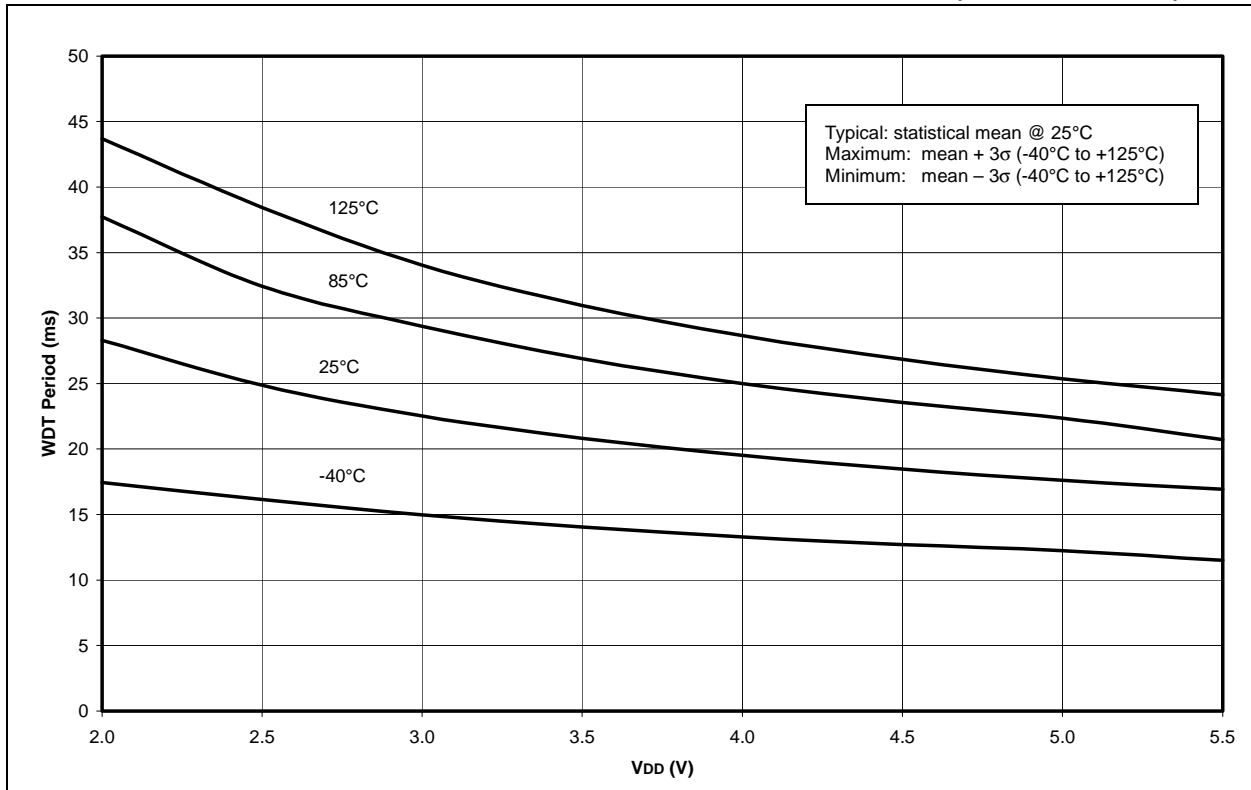
**FIGURE 15-12: TYPICAL AND MAXIMUM  $\Delta I_{WDT}$  vs.  $V_{DD}$  OVER TEMPERATURE**



**FIGURE 15-13: TYPICAL, MINIMUM AND MAXIMUM WDT PERIOD vs. V<sub>DD</sub> (-40°C TO +125°C)**



**FIGURE 15-14: AVERAGE WDT PERIOD vs. V<sub>DD</sub> OVER TEMPERATURE (-40°C TO +125°C)**



# PIC16F72

FIGURE 15-15: TYPICAL, MINIMUM AND MAXIMUM  $V_{OH}$  vs.  $I_{OH}$  ( $V_{DD} = 5V$ ,  $-40^{\circ}C$  TO  $+125^{\circ}C$ )

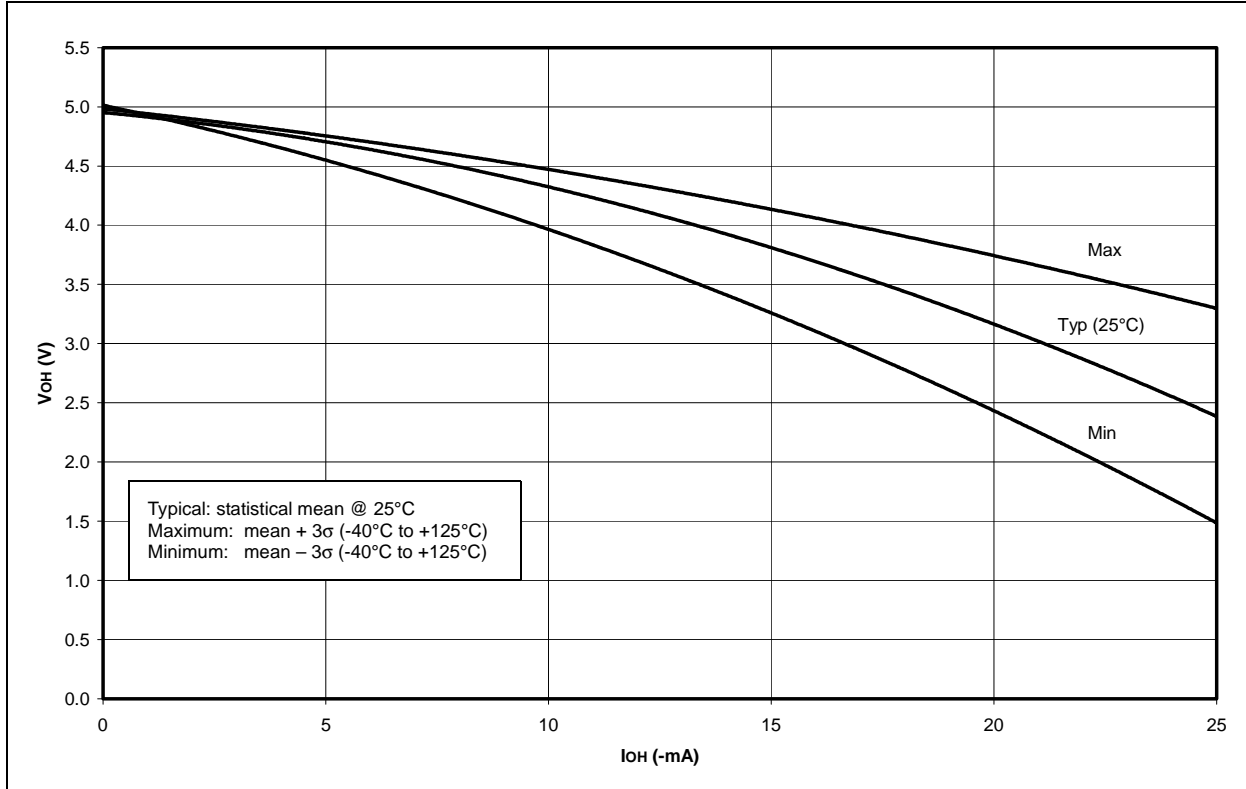
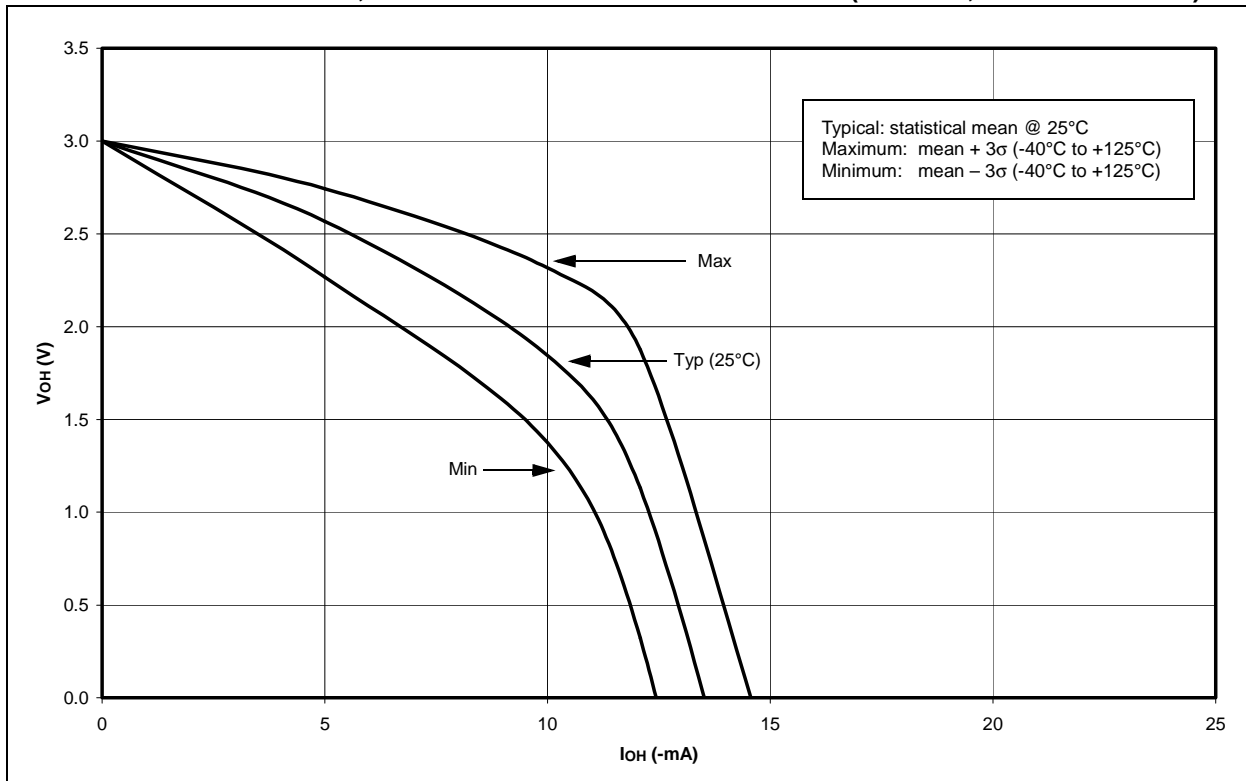
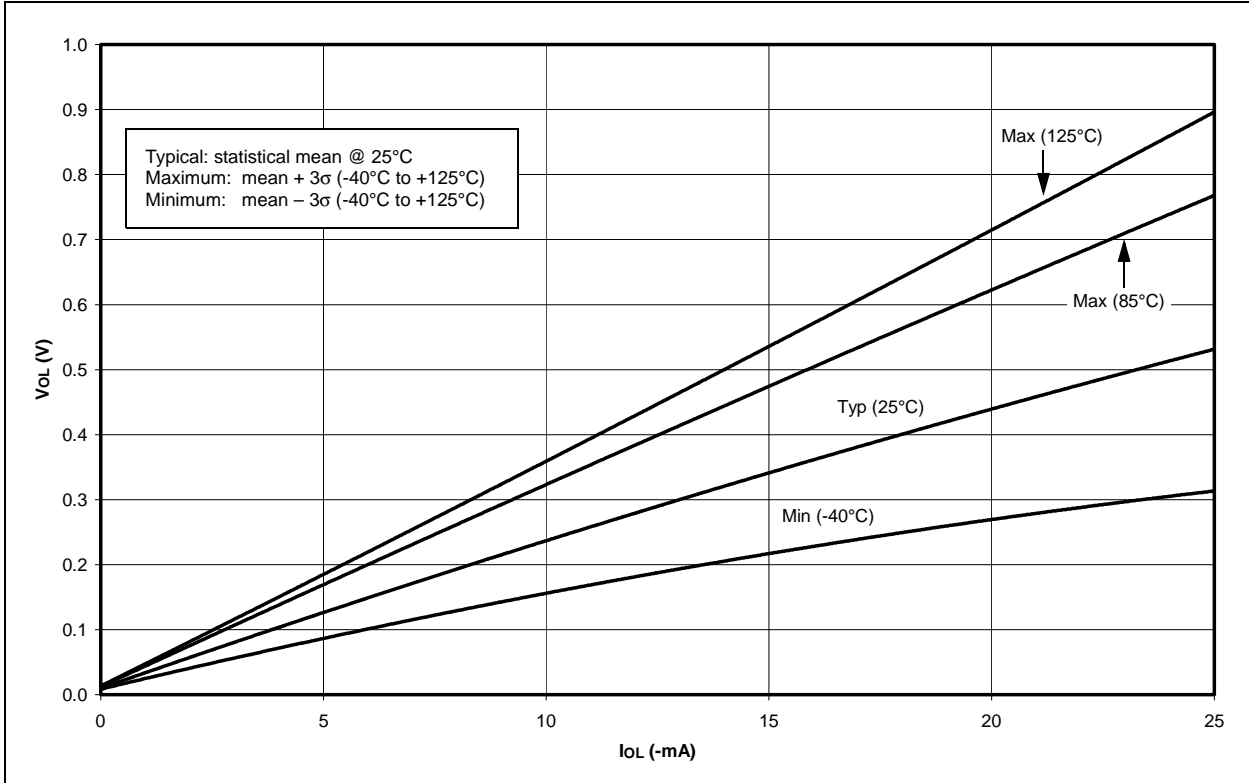


FIGURE 15-16: TYPICAL, MINIMUM AND MAXIMUM  $V_{OH}$  vs.  $I_{OH}$  ( $V_{DD} = 3V$ ,  $-40^{\circ}C$  TO  $+125^{\circ}C$ )

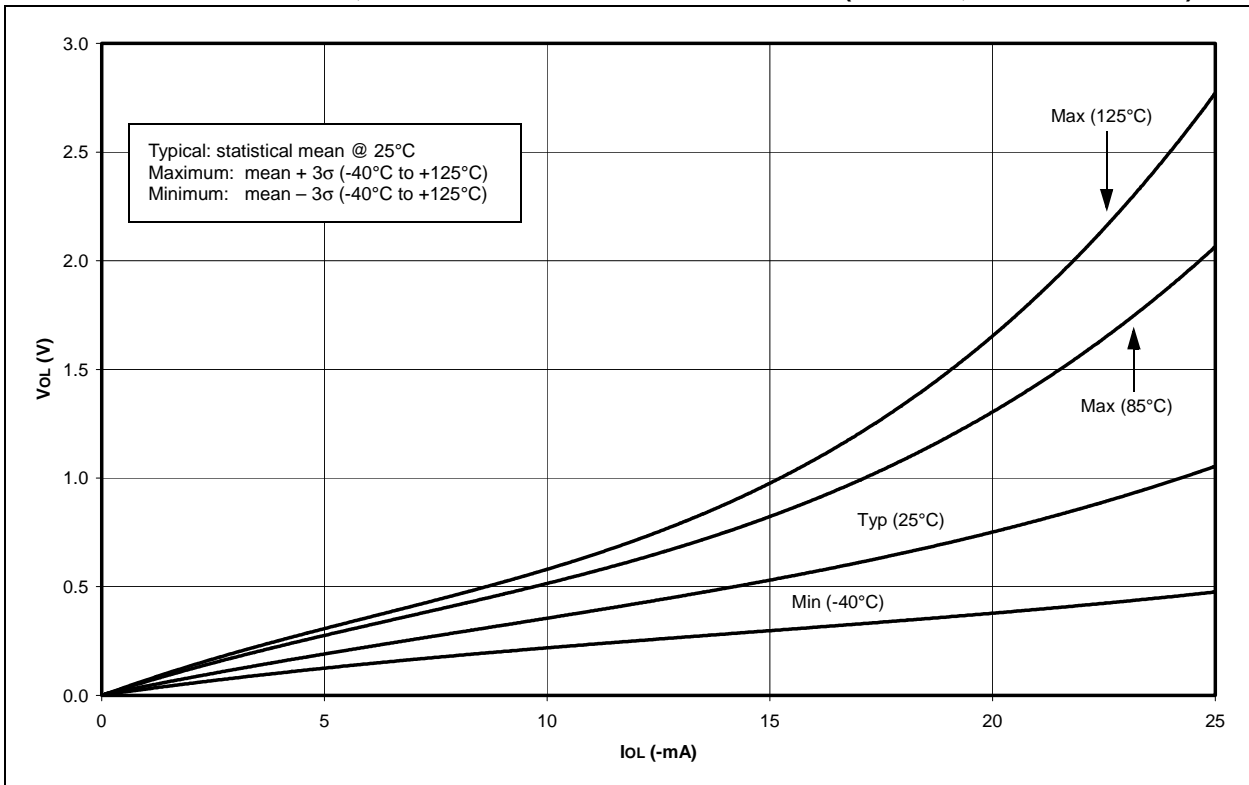




**FIGURE 15-17: TYPICAL, MINIMUM AND MAXIMUM  $V_{OL}$  vs.  $I_{OL}$  ( $V_{DD} = 5V$ ,  $-40^{\circ}C$  TO  $+125^{\circ}C$ )**

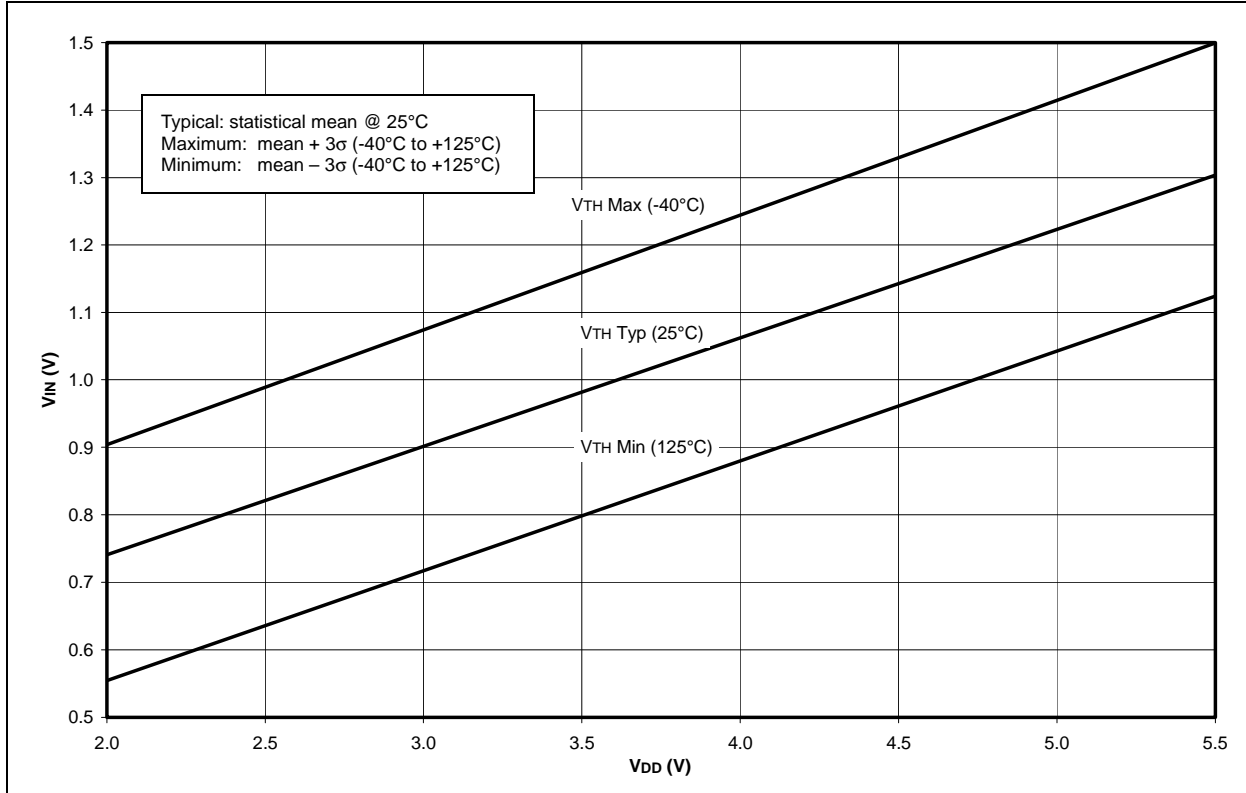


**FIGURE 15-18: TYPICAL, MINIMUM AND MAXIMUM  $V_{OL}$  vs.  $I_{OL}$  ( $V_{DD} = 3V$ ,  $-40^{\circ}C$  TO  $+125^{\circ}C$ )**

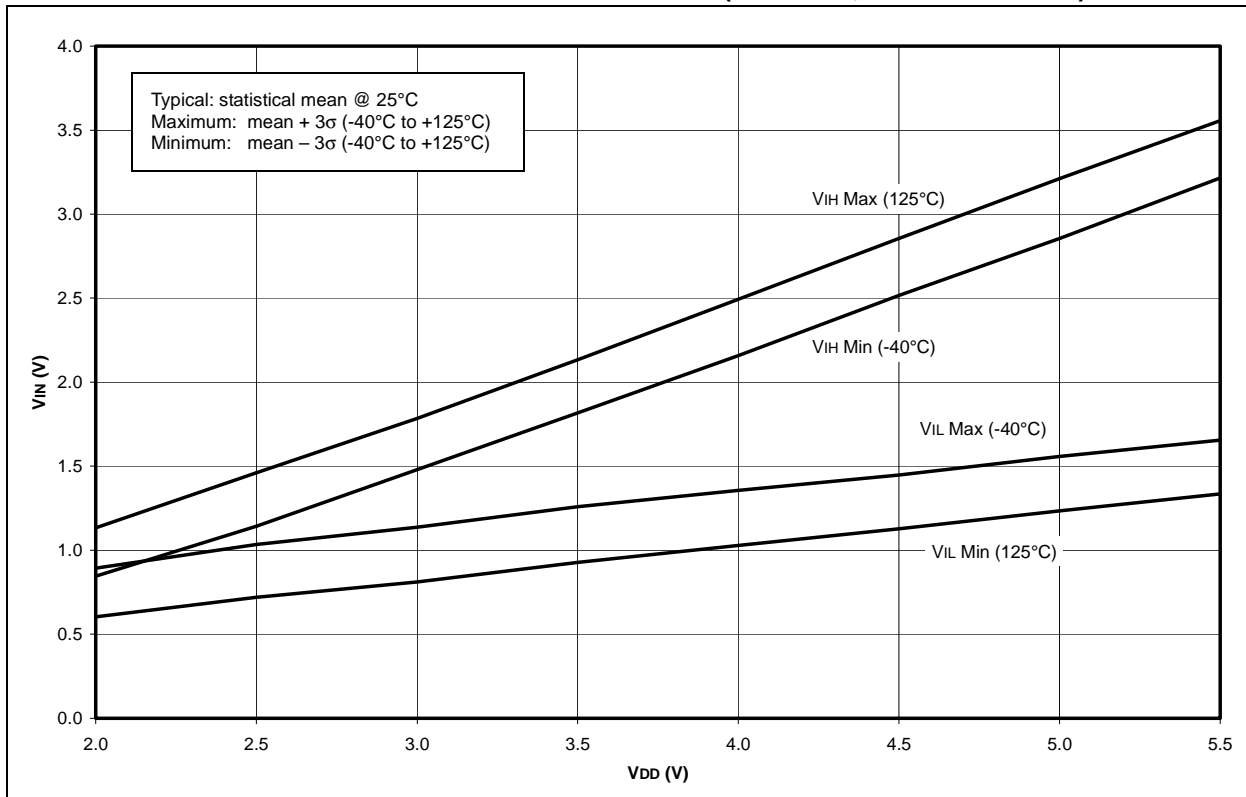


# PIC16F72

**FIGURE 15-19: MINIMUM AND MAXIMUM  $V_{IN}$  vs.  $V_{DD}$ , (TTL INPUT,  $-40^{\circ}\text{C}$  TO  $+125^{\circ}\text{C}$ )**

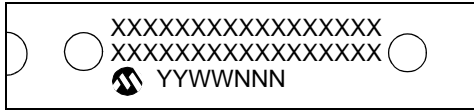


**FIGURE 15-20: MINIMUM AND MAXIMUM  $V_{IN}$  vs.  $V_{DD}$  (ST INPUT,  $-40^{\circ}\text{C}$  TO  $+125^{\circ}\text{C}$ )**

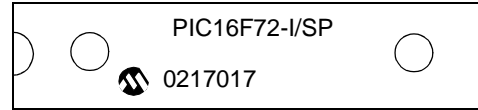


## 16.0 PACKAGE MARKING INFORMATION

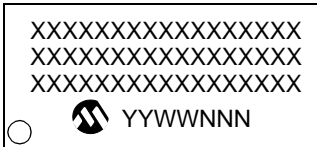
28-Lead PDIP (Skinny DIP)



Example



28-Lead SOIC

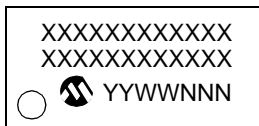


Example



[www.trimmer.ru](http://www.trimmer.ru)

28-Lead SSOP



Example



28-Lead QFN



Example



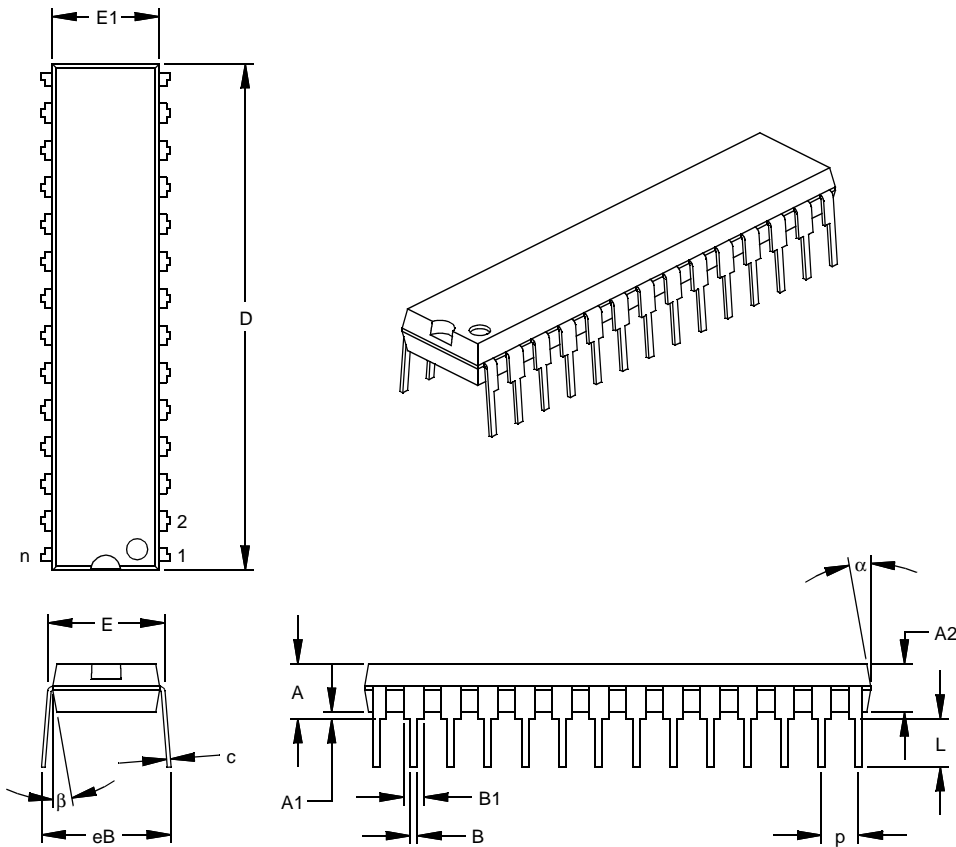
**Legend:** XX...X Customer specific information\*  
 Y Year code (last digit of calendar year)  
 YY Year code (last 2 digits of calendar year)  
 WW Week code (week of January 1 is week '01')  
 NNN Alphanumeric traceability code

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

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

# PIC16F72

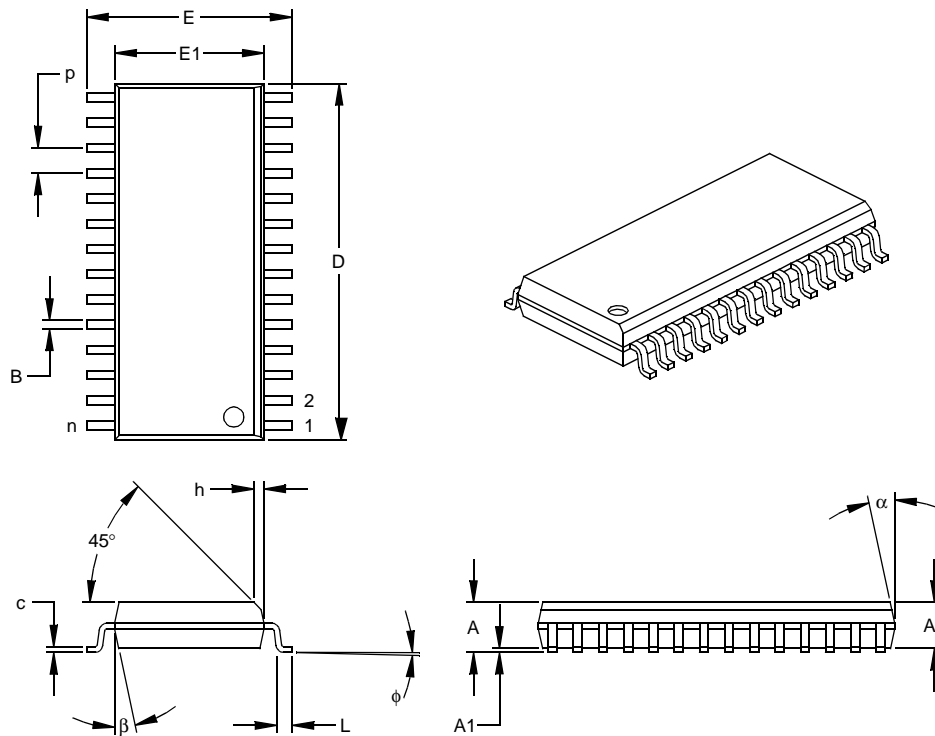
## 28-Lead Skinny Plastic Dual In-line (SP) – 300 mil (PDIP)



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

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

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



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

\* Controlling Parameter  
 § Significant Characteristic

Notes:

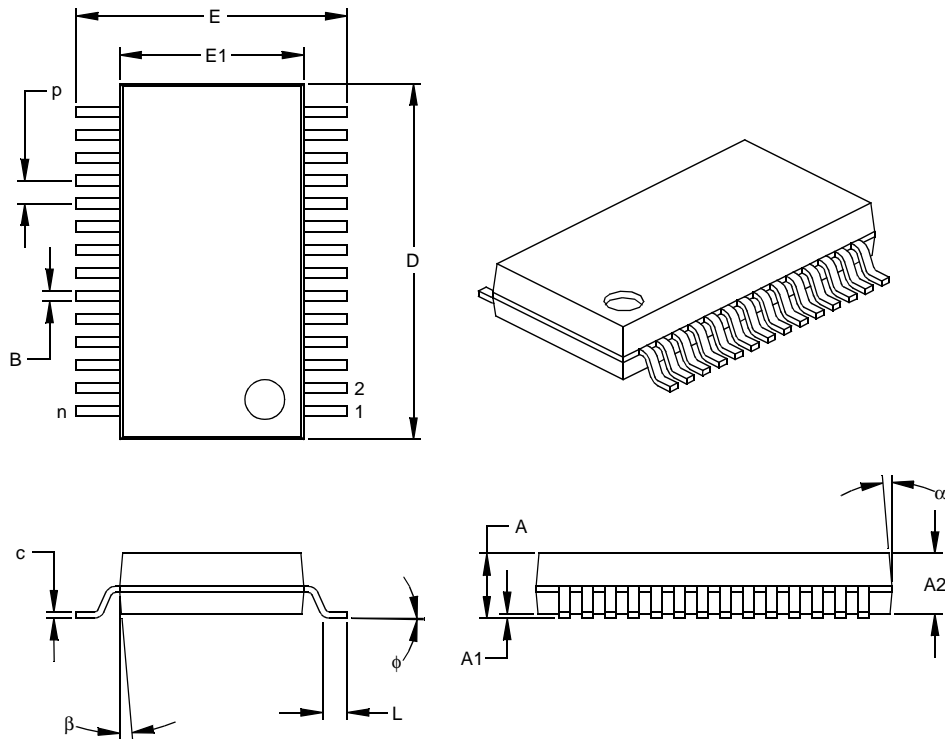
Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" (0.254mm) per side.

JEDEC Equivalent: MS-013

Drawing No. C04-052

# PIC16F72

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

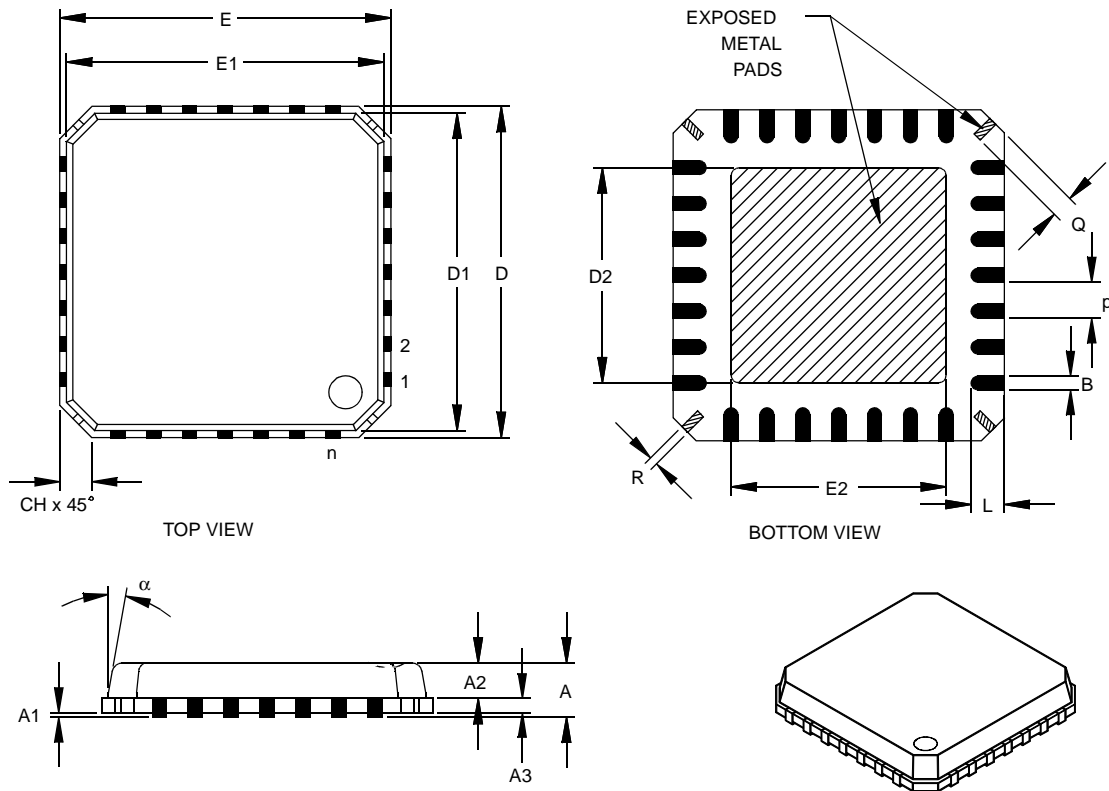


Dimension Limits	Units	INCHES			MILLIMETERS*		
	n	MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		28			28	
Pitch	p		.026			0.65	
Overall Height	A	.068	.073	.078	1.73	1.85	1.98
Molded Package Thickness	A2	.064	.068	.072	1.63	1.73	1.83
Standoff §	A1	.002	.006	.010	0.05	0.15	0.25
Overall Width	E	.299	.309	.319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

\* Controlling Parameter  
 § Significant Characteristic

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

## 28-Lead Plastic Quad Flat No Leads Package (ML) 6x6 mm Body (QFN)



Units		INCHES			MILLIMETERS*		
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		28			28	
Pitch	p	.026 BSC			0.65 BSC		
Overall Height	A		.033	.039		0.85	1.00
Molded Package Thickness	A2		.026	.031		0.65	0.80
Standoff	A1	.000	.0004	.002	0.00	0.01	0.05
Base Thickness	A3	.008 REF.			0.20 REF.		
Overall Width	E	.236 BSC			6.00 BSC		
Molded Package Width	E1	.226 BSC			5.75 BSC		
Exposed Pad Width	E2	.140	.146	.152	3.55	3.70	3.85
Overall Length	D	.236 BSC			6.00 BSC		
Molded Package Length	D1	.226 BSC			5.75 BSC		
Exposed Pad Length	D2	.140	.146	.152	3.55	3.70	3.85
Lead Width	B	.009	.011	.014	0.23	0.28	0.35
Lead Length	L	.020	.024	.030	0.50	0.60	0.75
Tie Bar Width	R	.005	.007	.010	0.13	0.17	0.23
Tie Bar Length	Q	.012	.016	.026	0.30	0.40	0.65
Chamfer	CH	.009	.017	.024	0.24	0.42	0.60
Mold Draft Angle Top	$\alpha$			12°			12°

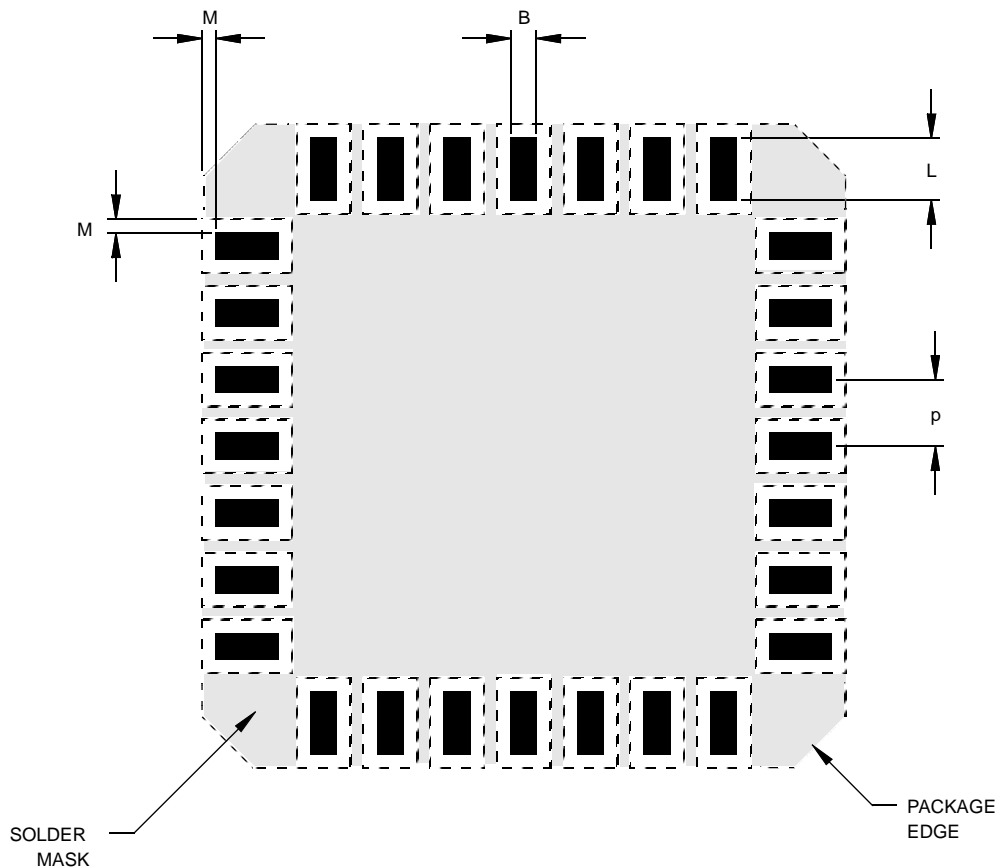
\* Controlling Parameter

**Notes:** Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" (0.254mm) per side.  
JEDEC equivalent: pending

Drawing No. C04-114

# PIC16F72

## 28-Lead Plastic Quad Flat No Leads Package (ML) 6x6 mm Body (QFN) (Continued)



Dimension Limits	Units	INCHES			MILLIMETERS*		
		MIN	NOM	MAX	MIN	NOM	MAX
Pitch	P	.026 BSC			0.65 BSC		
Pad Width	B	.009	.011	.014	0.23	0.28	0.35
Pad Length	L	.020	.024	.030	0.50	0.60	0.75
Pad to Solder Mask	M	.005		.006	0.13		0.15

\*Controlling Parameter

Drawing No. C04-2114



## APPENDIX A: REVISION HISTORY

Version	Date	Revision Description
A	April 2002	This is a new data sheet. However, this device is similar to the PIC16C72 device found in the PIC16C7X Data Sheet (DS30390), the PIC16C72A Data Sheet (DS35008) or the PIC16F872 device (DS30221).
B	May 2002	Final data sheet. Includes device characterization data. Minor typographic revisions throughout.

## APPENDIX B: CONVERSION CONSIDERATIONS

Considerations for converting from previous versions of devices to the ones listed in this data sheet are listed in Table B-1.

**TABLE B-1: CONVERSION CONSIDERATIONS**

Characteristic	PIC16C72/72A	PIC16F872	PIC16F72
Pins	28	28	28
Timers	3	3	3
Interrupts	8	10	8
Communication	Basic SSP/SSP (SPI, I <sup>2</sup> C Slave)	MSSP (SPI, I <sup>2</sup> C Master/Slave)	SSP (SPI, I <sup>2</sup> C Slave)
Frequency	20 MHz	20 MHz	20 MHz
A/D	8-bit, 5 Channels	10-bit, 5 Channels	8-bit, 5 Channels
CCP	1	1	1
Program Memory	2K EPROM	2K FLASH (1,000 E/W cycles)	2K FLASH (1000 E/W cycles)
RAM	128 bytes	128 bytes	128 bytes
EEPROM Data	None	64 bytes	None
Other	—	In-Circuit Debugger, Low Voltage Programming	—

## INDEX

### A

A/D	
Acquisition Requirements	56
ADCON0 Register	53
ADCON1 Register	53
ADIF bit	54
ADRES Register	53
Analog-to-Digital Converter	53
Associated Registers	57
Configuring Analog Port Pins	56
Configuring the Interrupt	54
Configuring the Module	54
Conversion Clock	56
Conversions	56
Converter Characteristics	104
Effects of a RESET	57
Internal Sampling Switch (R <sub>ss</sub> ) Impedance	56
Operation During SLEEP	57
Source Impedance	56
Use of the the CCP Trigger	57
Absolute Maximum Ratings	87
ACK	49
ADCON0	
GO/DONE bit	54
ADRES Register	9, 54
Application Notes	
AN546 (Using the Analog-to-Digital Converter)	53
AN552 (Implementing Wake-up on Key Strokes Using PIC16F7X)	23
AN556 (Implementing a Table Read)	19
AN578 (Use of the SSP Module in the I <sup>2</sup> C Multi-Master Environment)	43
AN607 (Power-up Trouble Shooting)	64
Assembler	
MPASM Assembler	81
<b>B</b>	
BF	44
Block Diagrams	
A/D	55
Analog Input Model	55
Capture Mode Operation	38
Compare Mode Operation	39
In-Circuit Serial Programming Connections	72
Interrupt Logic	68
On-Chip Reset Circuit	63
PIC16F72	5
PORTC	25
PWM	41
RA3:RA0 and RA5 Port Pins	21
RA4/T0CKI Pin	21
RB3:RB0 Port Pins	23
RB7:RB4 Port Pins	23
Recommended MCLR Circuit	63
SSP in I <sup>2</sup> C Mode	48
SSP in SPI Mode	46
Timer0/WDT Prescaler	29
Timer1	32
Timer2	35
Watchdog Timer (WDT)	70
BOR. See Brown-out Reset	
Brown-out Reset (BOR)	59, 62, 65, 66
Buffer Full Status bit, BF	44

### C

Capture/Compare/PWM	37
Associated Registers with PWM and Timer2	42
Associated Registers, Capture, Compare and Timer1	40
Capture	
CCP1IF	38
CCPR1	38
CCPR1H:CCPR1L	38
Capture Mode	38
CCP Mode Timer Resources	37
CCP Pin Configuration	38, 39
CCP Prescaler	38
CCPR1L Register	37
Compare Mode	39
PWM Mode	41
PWM, Example Frequencies/Resolutions	42
Software Interrupt	38
Software Interrupt Mode	39
Special Event Trigger and A/D Conversions	39
Special Event Trigger Output of CCP1	39
Timer1 Mode Selection	38, 39
CCPR1H Register	37
CCPxM0 bit	37
CCPxM1 bit	37
CCPxM2 bit	37
CCPxM3 bit	37
CCPxX bit	37
CCPxY bit	37
CKE	44
CKP	45
Clock Polarity Select bit, CKP	45
Code Examples	
Changing Between Capture Prescalers	38
FLASH Program Read	28
Indirect Addressing	19
Initializing PORTA	21
Initializing PORTB	23
Initializing PORTC	25
Saving STATUS, W and PCLATH	
Registers in RAM	69
Code Protection	59, 72
Configuration Bits	59
Configuration Word	60
Conversion Considerations	123

### D

D/A	44
Data Memory	
General Purpose Register File	7
Special Function Registers	9
Data/Address bit, D/A	44
DC and AC Characteristics	
Graphs and Tables	107
DC Characteristics	89
Development Support	81
Device Overview	5
Direct Addressing	20

### E

Electrical Characteristics	87
Errata	3

# PIC16F72

## F

FLASH Program Memory	
Associated Registers .....	28
Operation During Code Protect.....	28
Reading.....	28
FSR Register.....	9, 10

## I

I/O Ports .....	21
PORTA .....	21
PORTB.....	23
PORTC.....	25

## I<sup>2</sup>C

Associated Registers .....	51
Master Mode .....	51
Mode Selection .....	48
Multi-Master Mode .....	51
SCL and SDA pins .....	48
Slave Mode .....	48

ICEPIC In-Circuit Emulator .....	82
----------------------------------	----

ID Locations .....	72
--------------------	----

In-Circuit Serial Programming (ICSP) .....	72
--	----

INDF Register .....	10
---------------------	----

Indirect Addressing .....	20
---------------------------	----

FSR Register .....	19
--------------------	----

INDF Register .....	19
---------------------	----

Instruction Format .....	73
--------------------------	----

Instruction Set .....	73
-----------------------	----

ADDLW .....	75
-------------	----

ADDWF .....	75
-------------	----

ANDLW .....	75
-------------	----

ANDWF .....	75
-------------	----

BCF .....	75
-----------	----

BSF .....	75
-----------	----

BTFSC .....	76
-------------	----

BTFSS .....	76
-------------	----

CALL .....	76
------------	----

CLRF .....	76
------------	----

CLRW .....	76
------------	----

CLRWDT.....	76
-------------	----

COMF .....	77
------------	----

DECF .....	77
------------	----

DECFSZ.....	77
-------------	----

GOTO.....	77
-----------	----

INCF.....	77
-----------	----

INCFSZ.....	77
-------------	----

IORLW.....	78
------------	----

IORWF .....	78
-------------	----

MOVF.....	78
-----------	----

MOVLW.....	78
------------	----

MOVWF .....	78
-------------	----

NOP .....	78
-----------	----

RETFIE .....	79
--------------	----

RETLW.....	79
------------	----

RETURN .....	79
--------------	----

RLF .....	79
-----------	----

RRF.....	79
----------	----

SLEEP .....	79
-------------	----

SUBLW.....	80
------------	----

SUBWF .....	80
-------------	----

Summary Table .....	74
---------------------	----

SWAPF .....	80
-------------	----

XORLW .....	80
-------------	----

XORWF .....	80
-------------	----

INT Interrupt (RB0/INT). See Interrupt Sources

INTCON Register	
GIE bit.....	14
INTE bit.....	14
INTF bit.....	14
RBIF bit.....	14
TMR0IE bit.....	14

Internal Sampling Switch (Rss) Impedance.....	56
---	----

Interrupt Sources .....	59, 68
-------------------------	--------

RB0/INT Pin, External.....	69
----------------------------	----

TMR0 Overflow.....	69
--------------------	----

## Interrupts

RB7:RB4 Port Change.....	23
--------------------------	----

Synchronous Serial Port Interrupt.....	16
--	----

Interrupts, Context Saving During.....	69
--	----

## Interrupts, Enable Bits

Global Interrupt Enable (GIE bit) .....	14, 68
---	--------

Interrupt-on-Change (RB7:RB4)	
-------------------------------	--

Enable (RBIE bit) .....	69
-------------------------	----

RB0/INT Enable (INTE bit) .....	14
---------------------------------	----

TMR0 Overflow Enable (TMR0IE bit) .....	14
---	----

## Interrupts, Flag bits

Interrupt-on-Change (RB7:RB4) Flag	
------------------------------------	--

(RBIF bit) .....	14
------------------	----

Interrupt-on-Change (RB7:RB4) Flag	
------------------------------------	--

(RBIF bit) .....	14, 69
------------------	--------

RB0/INT Flag (INTF bit).....	14
------------------------------	----

TMR0 Overflow Flag (TMR0IF bit).....	69
--------------------------------------	----

## K

KEELOQ Evaluation and Programming Tools .....	84
---	----

## L

Loading of PC .....	18
---------------------	----

## M

### Master Clear ( $\overline{\text{MCLR}}$ )

MCLR Reset, Normal Operation.....	62, 65, 66
-----------------------------------	------------

MCLR Reset, SLEEP.....	62, 65, 66
------------------------	------------

Operation and ESD Protection .....	63
------------------------------------	----

### Memory

Data Memory .....	7
-------------------	---

Program Memory .....	7
----------------------	---

MPLAB C17 and MPLAB C18 C Compilers .....	81
---	----

MPLAB ICD In-Circuit Debugger .....	83
-------------------------------------	----

### MPLAB ICE High Performance Universal

In-Circuit Emulator with MPLAB IDE .....	82
--	----

### MPLAB Integrated Development

Environment Software .....	81
----------------------------	----

MPLINK Object Linker/MPLIB Object Librarian .....	82
---	----

## O

On-Line Support .....	131
-----------------------	-----

OPCODE Field Descriptions.....	73
--------------------------------	----

### OPTION\_REG Register

INTEDG bit .....	13
------------------	----

PS2:PS0 bits.....	13
-------------------	----

PSA bit.....	13
--------------	----

RBPV bit.....	13
---------------	----

T0CS bit.....	13
---------------	----

T0SE bit.....	13
---------------	----

Oscillator Configuration.....	59, 61	PORTA Register .....	9
Crystal Oscillator/Ceramic Resonators .....	61	PORTB	
HS .....	61, 65	Associated Registers .....	24
LP .....	61, 65	Functions .....	24
RC .....	61, 62, 65	Pull-up Enable (RBP $\bar{U}$ bit) .....	13
XT .....	61, 65	RB0/INT Edge Select (INTEDG bit) .....	13
Oscillator, WDT .....	70	RB0/INT Pin, External .....	69
<b>P</b>		RB7:RB4 Interrupt-on-Change Flag (RBIF bit) .....	14
P .....	44	RB7:RB4 Interrupt-on-Change .....	69
Package Marking Information .....	117	RB7:RB4 Interrupt-on-Change Enable	
PCFG0 bit .....	54	(RBIE bit) .....	69
PCFG1 bit .....	54	RB7:RB4 Interrupt-on-Change Flag	
PCFG2 bit .....	54	(RBIF bit) .....	14, 69
PCL Register.....	9, 10, 18	PORTB Register .....	9
PCLATH Register .....	9, 10, 18	PORTC	
PCON Register .....	64	Associated Registers .....	26
POR bit .....	17	Functions .....	26
PICDEM 1 Low Cost PICmicro		PORTC Register.....	9
Demonstration Board.....	83	Postscaler, WDT	
PICDEM 17 Demonstration Board .....	84	Assignment (PSA Bit) .....	13
PICDEM 2 Low Cost PIC16CXX		Rate Select (PS2:PS0 bits) .....	13
Demonstration Board.....	83	Power-down Mode. See SLEEP	
PICDEM 3 Low Cost PIC16CXXX		Power-on Reset (POR).....	59, 62, 64, 65, 66
Demonstration Board.....	84	Brown-out Reset (BOR).....	64
PICSTART Plus Entry Level		Oscillator Start-up Timer (OST) .....	59, 64
Development Programmer .....	83	POR Status ( $\overline{POR}$ bit) .....	17
Pin Functions		Power Control/Status Register (PCON).....	64
MCLR/VPP.....	6	Power-down ( $\overline{PD}$ bit) .....	62
OSC1/CLKI .....	6	Power-up Timer (PWRT) .....	59, 64
OSC2/CLKO .....	6	Time-out (TO bit) .....	12, 62
RA0/AN0 .....	6	Time-out Sequence .....	64
RA1/AN1 .....	6	PR2 Register .....	35
RA2/AN2 .....	6	Prescaler, Timer0	
RA3/AN3/VREF .....	6	Assignment (PSA bit) .....	13
RA4/T0CKI.....	6	Rate Select (PS2:PS0 bits) .....	13
RA5/AN4/ $\overline{SS}$ .....	6	PRO MATE II Universal Device Programmer .....	83
RB0/INT .....	6	Product Identification System .....	133
RB1 .....	6	Program Counter	
RB2 .....	6	RESET Conditions.....	65
RB3 .....	6	Program Memory	
RB4 .....	6	Paging .....	19
RB5 .....	6	Program Memory Map and Stack .....	7
RB6/PGC .....	6	Program Verification .....	72
RB7/PGD .....	6	PUSH.....	19
RC0/T1OSO/T1CKI .....	6	<b>R</b>	
RC1/T1OSI .....	6	R/ $\overline{W}$ .....	44
RC2/CCP1 .....	6	R/ $\overline{W}$ bit .....	49
RC3/SCK/SCL .....	6	RBIF bit.....	23
RC4/SDI/SDA .....	6	Read/Write bit Information, R/ $\overline{W}$ .....	44
RC5/SDO .....	6	Reader Response.....	132
RC6 .....	6	Reading Program Memory.....	27
RC7 .....	6	PMADR.....	27
VDD .....	6	PMCON1 Register.....	27
Vss.....	6	Receive Overflow Indicator bit, SSPOV.....	45
Pinout Descriptions		Register File Map.....	8
PIC16F72.....	6		
POP .....	19		
POR. See Power-on Reset			
PORTA			
Associated Registers .....	22		
Functions .....	22		

# PIC16F72

Registers .....	36	SSPADD Register .....	10
ADCON0 (A/D Control 0) .....	53	SSPEN .....	45
ADCON1 (A/D Control 1) .....	54	SSPIF .....	16
CCPCON1 (Capture/Compare/PWM Control 1) .....	37	SSPM3:SSPM0 .....	45
Initialization Conditions (table) .....	66	SSPOV .....	45
INTCON (Interrupt Control) .....	14	SSPSTAT Register .....	10
OPTION .....	13	Stack .....	19
PCON (Power Control) .....	17	Overflows .....	19
PIE1 (Peripheral Interrupt Enable 1) .....	15	Underflow .....	19
PIR1 (Peripheral Interrupt Flag 1) .....	16	START bit, S .....	44
PMCON1 (Program Memory Control 1) .....	27	STATUS Register .....	
SSPCON (Sync Serial Port Control) .....	45	DC bit .....	12
SSPSTAT (Synchronous Serial Port Status) .....	44	IRP bit .....	12
STATUS .....	12	PD bit .....	62
Summary .....	9	TO bit .....	12, 62
T1CON (Timer1 Control) .....	31	STOP bit, P .....	44
RESET .....	59, 62	Synchronous Serial Port (SSP) .....	43
Brown-out Reset (BOR). See Brown-out Reset (BOR)		Overview .....	43
MCLR RESET. See MCLR		SPI Mode .....	43
Power-on Reset (POR). See Power-on Reset (POR)		Synchronous Serial Port Enable bit, SSPEN .....	45
RESET Conditions for All Registers .....	66	Synchronous Serial Port Interrupt .....	16
RESET Conditions for PCON Register .....	65	Synchronous Serial Port Mode Select bits,	
RESET Conditions for Program Counter .....	65	SSPM3:SSPM0 .....	45
RESET Conditions for STATUS Register .....	65	<b>T</b>	
WDT Reset. See Watchdog Timer (WDT)		T2CKPS0 bit .....	36
Revision History .....	123	T2CKPS1 bit .....	36
RP0, RP1 bit .....	7	T2CON (Timer2 Control) .....	36
<b>S</b>		TAD .....	56
S .....	44	Timer0 .....	29
Sales and Support .....	133	Clock Source Edge Select (T0SE bit) .....	13
Slave Mode .....		Clock Source Select (T0CS bit) .....	13
SCL .....	48	External Clock .....	30
SDA .....	48	Interrupt .....	29
SLEEP .....	59, 62, 71	Operation .....	29
SMP .....	44	Overflow Enable (TMR0IE bit) .....	14
Software Simulator (MPLAB SIM) .....	82	Overflow Flag (TMR0IF bit) .....	69
Special Event Trigger .....	57	Overflow Interrupt .....	69
Special Features of the CPU .....	59	Prescaler .....	30
Special Function Registers .....		T0CKI .....	30
PMADRH .....	27	Timer1 .....	
PMADRL .....	27	Associated Registers .....	34
PMCON1 .....	27	Asynchronous Counter Mode .....	33
PMDATH .....	27	Capacitor Selection .....	33
PMDATL .....	27	Counter Operation .....	32
SPI .....		Interrupt .....	33
Associated Registers .....	46	Operation in Timer Mode .....	32
SPI Clock Edge Select bit, CKE .....	44	Oscillator .....	33
SPI Data Input Sample Phase Select bit, SMP .....	44	Prescaler .....	34
SPI Mode .....		Resetting TMR1H, TMR1L Register Pair .....	34
Serial Clock .....	43	Resetting Using a CCP Trigger Output .....	33
Serial Data In .....	43	Synchronized Counter Mode .....	32
Serial Data Out .....	43	Timer2 .....	35
Slave Select .....	43	Interrupt .....	35
SSP .....		Operation .....	35
ACK .....	48	Output .....	35
Addressing .....	48	Prescaler, Postscaler .....	35
BF bit .....	48		
I <sup>2</sup> C Mode Operation .....	48		
R/W bit .....	49		
Reception .....	49		
SCL Clock Input .....	48		
SSPOV bit .....	48		
Transmission .....	49		

## Timing Diagrams

A/D Conversion .....	105
Brown-out Reset .....	96
Capture/Compare/PWM (CCP1) .....	98
CLKO and I/O .....	95
External Clock .....	94
I <sup>2</sup> C Bus Data .....	102
I <sup>2</sup> C Bus START/STOP bits .....	101
I <sup>2</sup> C Reception (7-bit Address) .....	50
I <sup>2</sup> C Transmission (7-bit Address) .....	50
RESET, Watchdog Timer, Oscillator Start-up Timer and Power-up Timer .....	96
Slow Rise Time (MCLR Tied to VDD Through RC Network) .....	68
SPI Master Mode .....	47
SPI Master Mode (CKE = 0, SMP = 0) .....	99
SPI Master Mode (CKE = 1, SMP = 1) .....	99
SPI Slave Mode (CKE = 0) .....	47, 100
SPI Slave Mode (CKE = 1) .....	47, 100
Time-out Sequence on Power-up (MCLR Tied to VDD Through Pull-up Resistor) .....	67
Time-out Sequence on Power-up (MCLR Tied to VDD Through RC Network): Case 1 .....	67
Time-out Sequence on Power-up (MCLR Tied to VDD Through RC Network): Case 2 .....	67
Timer0 and Timer1 External Clock .....	97
Wake-up from SLEEP through Interrupt .....	72
Timing Parameter Symbology .....	93
TMR1H Register .....	9
TMR1L Register .....	9
TMR2 Register .....	9
TMR2ON bit .....	36
TOUTPS0 bit .....	36
TOUTPS1 bit .....	36
TOUTPS2 bit .....	36
TOUTPS3 bit .....	36
TRISA Register .....	10, 21
TRISB Register .....	10, 23
TRISC Register .....	10, 25

## U

UA .....	44
Update Address bit, UA .....	44

## W

Wake-up from SLEEP .....	59, 71
Interrupts .....	65, 66
MCLR Reset .....	66
WDT Reset .....	66
Watchdog Timer (WDT) .....	59, 70
Associated Registers .....	70
Enable (WDTEN bit) .....	70
Postscaler. See Postscaler, WDT	
Programming Considerations .....	70
RC Oscillator .....	70
Time-out Period .....	70
WDT Reset, Normal Operation .....	62, 65, 66
WDT Reset, SLEEP .....	62, 65, 66
WCOL .....	45
Write Collision Detect bit, WCOL .....	45
WWW, On-Line Support .....	3

## 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>	X	/XX	XXX
Device	Temperature Range	Package	Pattern
Device	PIC16F72: Standard VDD range PIC16F72T: (Tape and Reel) PIC16LF72: Extended VDD range		
Temperature Range	- = 0°C to +70°C I = -40°C to +85°C		
Package	SO = SOIC SS = SSOP ML = QFN P = PDIP		
Pattern	QTP, SQTP, ROM Code (factory specified) or Special Requirements. Blank for OTP and Windowed devices.		

**Examples:**

- a) PIC16F72-04I/SO = Industrial Temp., SOIC package, normal VDD limits
- b) PIC16LF72-20I/SS = Industrial Temp., SSOP package, extended VDD limits
- c) PIC16F72-20I/ML = Industrial Temp., QFN package, normal VDD limits

\* JW Devices are UV erasable and can be programmed to any device configuration. JW Devices meet the electrical requirement of each oscillator type.