



Welcome to E-XFL.COM

#### What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "<u>Embedded -</u> <u>Microcontrollers</u>"

#### Details

Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	20MHz
Connectivity	I <sup>2</sup> C, SPI
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	22
Program Memory Size	3.5KB (2K x 14)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	4V ~ 5.5V
Data Converters	A/D 5x8b
Oscillator Type	External
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SSOP (0.209", 5.30mm Width)
Supplier Device Package	28-SSOP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f72t-e-ss

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

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

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

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

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

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

#### Trademarks

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

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

Analog-for-the-Digital Age, Application Maestro, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, ECAN, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, Linear Active Thermistor, Mindi, MiWi, MPASM, MPLIB, MPLINK, PICkit, PICDEM, PICDEM.net, PICLAB, PICtail, PowerCal, PowerInfo, PowerMate, PowerTool, REAL ICE, rfLAB, rfPICDEM, Select Mode, Smart Serial, SmartTel, Total Endurance, UNI/O, WiperLock and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

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

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

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

Printed on recycled paper.

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

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

\_

## 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 7.0).

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.

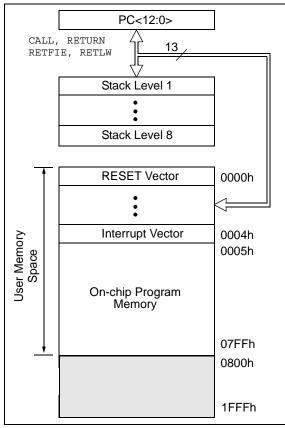
Additional information on device memory may be found in the PIC<sup>™</sup> Mid-Range Reference Manual, (DS33023).

## 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).

#### FIGURE 2-2:

#### -2: PIC16F72 REGISTER FILE MAP

Indirect addr.(*)	00h	Indirect addr.(*)	80h	Indirect addr.(*)	100h	Indirect addr.(*)	180
TMR0	01h	OPTION	81h	TMR0	101h	OPTION	181
PCL	02h	PCL	82h	PCL	102h	PCL	182
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183
FSR	04h	FSR	84h	FSR	104h	FSR	184
PORTA	05h	TRISA	85h		105h		185
PORTB	06h	TRISB	86h	PORTB	106h	TRISB	186
PORTC	07h	TRISC	87h		107h		187
	08h		88h		108h		188
	09h		89h		109h		189
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18A
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18E
PIR1	0Ch	PIE1	8Ch	PMDATL	10Ch	PMCON1	180
	0Dh		8Dh	PMADRL	10Dh		180
TMR1L	0Eh	PCON	8Eh	PMDATH	10Eh		18E
TMR1H	0Fh		8Fh	PMADRH	10Fh		18F
T1CON	10h		90h		110h		190
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		11Fh		19F
ADCON0	1Fh	ADCON1	9Fh				
	20h	General Purpose Register	A0h		120h	accesses A0h -BFh	1A)
General		32 Bytes	BFh				1B
Purpose			C0h	accesses			1C
Register		accesses		20h-7Fh		accesses	
96 Bytes		40h-7Fh				40h -7Fh	
	7Fh		FFh		17Fh		1FF
Bank 0		Bank 1		Bank 2		Bank 3	
Unimplem	ented data	a memory location	s, read as	·'O'.			

#### 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

	·-						· · ·		1	1	
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:
Bank 0											
00h <sup>(1)</sup>	INDF	Addressi	ng this loca	tion uses cor	tents of FSR	to address of	data memor	/ (not a phys	ical register)	0000 0000	19
01h	TMR0	Timer0 N	lodule's Re	gister						xxxx xxxx	27,13
02h <sup>(1)</sup>	PCL	Program	Counter's (	PC) Least S	ignificant By	te				0000 0000	18
03h <sup>(1)</sup>	STATUS	IRP	RP1	RP0	TO	PD	Z	DC	С	0001 1xxx	12
04h <sup>(1)</sup>	FSR	Indirect [	Data Memo	ry Address P	ointer					xxxx xxxx	19
05h	PORTA	_		PORTA Dat	a Latch whe	n written: PC	ORTA pins w	hen read		0x 0000	21
06h	PORTB	PORTB I	Data Latch	when written	: PORTB pir	ns when read	ł			xxxx xxxx	23
07h	PORTC	PORTC I	Data Latch	when written	: PORTC pir	ns when read	b			xxxx xxxx	25
08h	_	Unimpler	mented								_
09h	—	Unimpler	mented							—	—
	PCLATH		—	—	Write Buffer	for the uppe	er 5 bits of th	ne 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	—	Unimpler	mented							—	—
0Eh	TMR1L	Holding F	Register for	the Least Si	ignificant Byt	e of the 16-b	oit TMR1 Re	gister		XXXX XXXX	29
0Fh	TMR1H	Holding I	Register for	the Most Sig	gnificant Byte	e of the 16-b	it TMR1 Reg	gister	_	xxxx xxxx	29
10h	T1CON		—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR10N	00 0000	29
11h	TMR2	Timer2 N	Iodule's Re	gister						0000 0000	33
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	34
13h	SSPBUF	Synchror	nous Serial	Port Receive	e Buffer/Tran	smit Registe	er	-	_	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 3								38,39,41	
16h	CCPR1H	Capture/	Compare/P	WM Registe	r (MSB)	-	-	-	_	xxxx xxxx	38,39,41
17h	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	00 0000	37
18h-1Dh	_	Unimpler	mented							_	_
1Eh	ADRES	A/D Resu	ult 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.

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'.

#### 2.2.2.5 PIR1 Register

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

- n = Value at POR

EGISTER 2-5:	PIR1: PEI	RIPHERAL	INTERRUI	PT FLAG R	EGISTER	1 (ADDRE	SS 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	Unimplem	ented: Read	d as '0'							
bit 6			nterrupt Flag	bit						
		) conversion /D conversio	completed	olete						
bit 5-4	Unimplem	ented: Read	d as '0'							
bit 3	SSPIF: Sy	nchronous S	Serial Port (S	SP) Interrupt	Flag bit					
	<ul> <li>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.</li> <li>0 = No SSP interrupt condition has occurred</li> </ul>									
bit 2	CCP1IF: C	CCP1IF: CCP1 Interrupt Flag bit								
	<u>Capture mode:</u> 1 = A TMR1 register capture occurred (must be cleared in software) 0 = No TMR1 register capture occurred <u>Compare mode:</u> 1 = A TMR1 register compare match occurred (must be cleared in software) 0 = No TMR1 register compare match occurred <u>PWM mode:</u> Unused in this mode									
bit 1	TMR2IF: T	MR2 to PR2	2 Match Inter	rupt Flag bit						
	<ul> <li>1 = TMR2 to PR2 match occurred (must be cleared in software)</li> <li>0 = No TMR2 to PR2 match occurred</li> </ul>									
bit 0	TMR1IF: T	MR1 Overflo	ow Interrupt	Flag bit						
bit 0 <b>TMR1IF:</b> TMR1 Overflow Interrupt Flag bit 1 = TMR1 register overflowed (must be cleared in software) 0 = TMR1 register did not overflow										
	Legend:							]		
	R = Reada	ble bit	W = W	ritable bit	U = Unim	plemented	bit, read as	'0'		
	1									

'1' = Bit is set

x = Bit is unknown

'0' = Bit is cleared

#### 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).

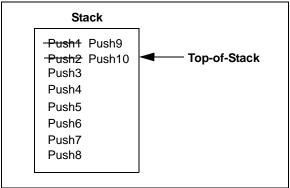
#### 2.3.2 STACK

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 POPs 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

NEXT	movwf clrf incf btfss	FSR INDF FSR FSR,4	<pre>;initialize pointer ;to RAM ;clear INDF register ;inc pointer ;all done? ;NO, clear next</pre>
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.

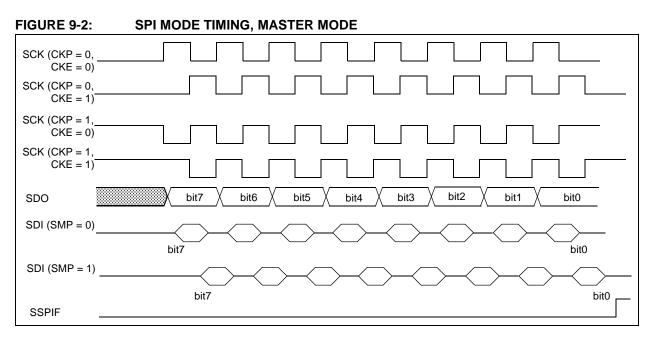
ER 9-2:	SSPCON	SSPCON: SYNC SERIAL PORT CONTROL REGISTER (ADDRESS 14h)									
	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0			
	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0			
	bit 7							bit 0			
bit 7	WCOL: W	/rite Collision	Detect bit								
		d in software		en while it is	still transmi	tting the pr	evious word	d (must be			
bit 6		Receive Over	flow Indicate	or bit							
DILO	In SPI mo		now mulcall								
	1 = A new of ove must mode by wr 0 = No ov	v byte is rece erflow, the da read the SSI , the overflo iting to the S verflow	ata in SSPSF PBUF, even i w bit is not s	e SSPBUF re R is lost. Ove if only transm et since each ster.	erflow can on hitting data, to	ly occur in avoid setti	Slave mode	e. The user . In Master			
		e is received lon't care" in		SPBUF regi ode. SSPOV							
bit 5	SSPEN: S	Synchronous	Serial Port E	Enable bit							
	1 = Enabl	<u>In SPI mode</u> : 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:										
	<ul> <li>1 = Enables the serial port and configures the SDA and SCL pins as serial port pins</li> <li>0 = Disables serial port and configures these pins as I/O port pins</li> </ul>										
	In both modes, when enabled, these pins must be properly configured as input or outp							utput.			
bit 4		ck Polarity S			,	Ū	•	·			
	In SPI mode:										
	<ul> <li>1 = IDLE state for clock is a high level (Microwire<sup>®</sup> default)</li> <li>0 = IDLE state for clock is a low level (Microwire alternate)</li> </ul>										
	In I <sup>2</sup> C mod SCK relea 1 = Enable	se control									
	0 = Holds	clock low (cl	ock stretch -	used to ens	ure data setu	ıp time)					
bit 3-0	SSPM<3:	0>: Synchro	nous Serial F	Port Mode Se	lect bits						
	0000 = SI	0000 = SPI Master mode, clock = Fosc/4									
		PI Master mo									
		PI Master mo			10						
				TMR2 o <u>utp</u> ut CK pin. SS p		abled.					
				CK pin. SS p			an be used	as I/O pin.			
	$0110 = I^2$	C Slave mod	le, 7-bit addr	ess				•			
		C Slave mod									
				ster mode (S		D hit intorru	into onoblo	4			
		1110 = $I^2C$ Slave mode, 7-bit address with START and STOP bit interrupts enabled 1111 = $I^2C$ Slave mode, 10-bit address with START and STOP bit interrupts enabled									
	Legend:										
	R = Read	able bit	W = V	Vritable bit	U = Unim	plemented	bit, read as	'0'			
			vv — v		0 - 01111	plomonicu	, 1000 05	~			

'1' = Bit is set

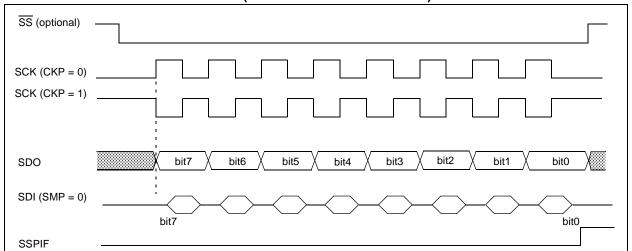
'0' = Bit is cleared

- n = Value at POR

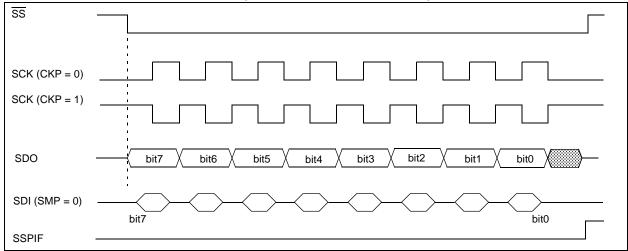
x = Bit is unknown











© 2007 Microchip Technology Inc.



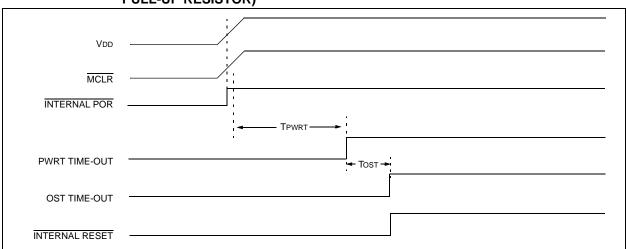


FIGURE 11-7: TIME-OUT SEQUENCE ON POWER-UP (MCLR TIED TO VDD THROUGH RC NETWORK): CASE 1

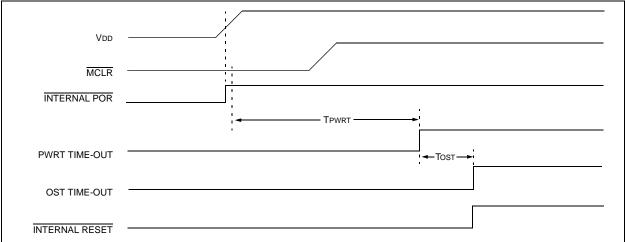
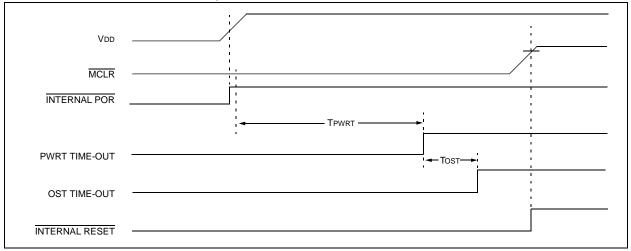


FIGURE 11-8: TIME-OUT SEQUENCE ON POWER-UP (MCLR TIED TO VDD THROUGH RC NETWORK): CASE 2



## 11.13 Watchdog Timer (WDT)

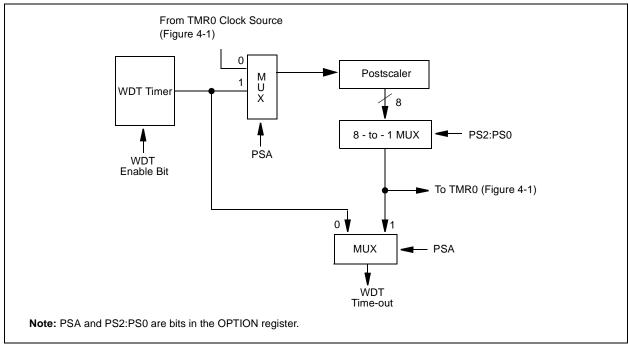
The Watchdog Timer is a free running, on-chip RC oscillator that does not require any external components. This RC oscillator is separate from the RC oscillator of the OSC1/CLKI pin. That means that the WDT will run, even if the clock on the OSC1/CLKI and OSC2/CLKO pins of the device has been stopped, 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 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	PWRTEN <sup>(1)</sup>	WDTEN	FOSC1	FOSC0
81h,181h	OPTION	RBPU	INTEDG	TOCS	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  $\ensuremath{\mathtt{SLEEP}}$  instruction.

If enabled, the Watchdog Timer will be cleared but keeps running, the PD bit (STATUS<3>) is cleared, the 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 VDD or VSS, 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 VDD or VSs for lowest current consumption. The contribution from on-chip pull-ups on PORTB should also be considered.

The MCLR pin must be at a logic high level (VIHMC).

#### 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 MCLR pin.
- 2. Watchdog Timer wake-up (if WDT was enabled).
- 3. Interrupt from INT pin, RB port change or a peripheral interrupt.

External MCLR Reset will cause a device RESET. All other events are considered a continuation of program execution and cause a "wake-up". The TO and PD bits in the STATUS register can be used to determine the cause of the device RESET. The PD bit, which is set on power-up, is cleared when SLEEP is invoked. The 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.
- SSP transmit or receive in Slave mode (SPI/I<sup>2</sup>C).
- 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 TO bit will not be set and 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 TO bit will be set and the 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 PD bit. If the 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.

## 12.1 Instruction Descriptions

ADDLW	Add Literal and W
Syntax:	[label] ADDLW k
Operands:	$0 \le k \le 255$
Operation:	$(W) + k \to (W)$
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] ANDWF f,d
Operands:	$\begin{array}{l} 0 \leq f \leq 127 \\ d \in \ [0,1] \end{array}$
Operation:	(W) .AND. (f) $\rightarrow$ (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] ADDWF f,d
Operands:	$\begin{array}{l} 0 \leq f \leq 127 \\ d \in \ [0,1] \end{array}$
Operation:	(W) + (f) $\rightarrow$ (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 ] BCF f,b
Operands:	$\begin{array}{l} 0 \leq f \leq 127 \\ 0 \leq b \leq 7 \end{array}$
Operation:	$0 \rightarrow (f < b >)$
Status Affected:	None
Description:	Bit 'b' in register 'f' is cleared.

ANDLW	AND Literal with W
Syntax:	[ label ] ANDLW k
Operands:	$0 \le k \le 255$
Operation:	(W) .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:	[ <i>label</i> ] BSF f,b
Operands:	$\begin{array}{l} 0 \leq f \leq 127 \\ 0 \leq b \leq 7 \end{array}$
Operation:	$1 \rightarrow (f < b >)$
Status Affected:	None
Description:	Bit 'b' in register 'f' is set.

## 13.0 DEVELOPMENT SUPPORT

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

- Integrated Development Environment
  - MPLAB® IDE Software
- Assemblers/Compilers/Linkers
  - MPASM<sup>™</sup> Assembler
  - MPLAB C17 and MPLAB C18 C Compilers
  - MPLINK<sup>™</sup> Object Linker/ MPLIB<sup>™</sup> 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<sup>®</sup> II Universal Device Programmer
- PICSTART<sup>®</sup> Plus Entry-Level Development Programmer
- Low Cost Demonstration Boards
  - PICDEM<sup>™</sup>1 Demonstration Board
  - PICDEM 2 Demonstration Board
  - PICDEM 3 Demonstration Board
  - PICDEM 17 Demonstration Board
  - KEELOQ<sup>®</sup> 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<sup>®</sup>-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 PIC 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 PIC 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<sup>®</sup> 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.

## 13.8 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 PIC MCUs and can be used to develop for this and other PIC 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<sup>TM</sup> 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 PIC 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 PIC 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.

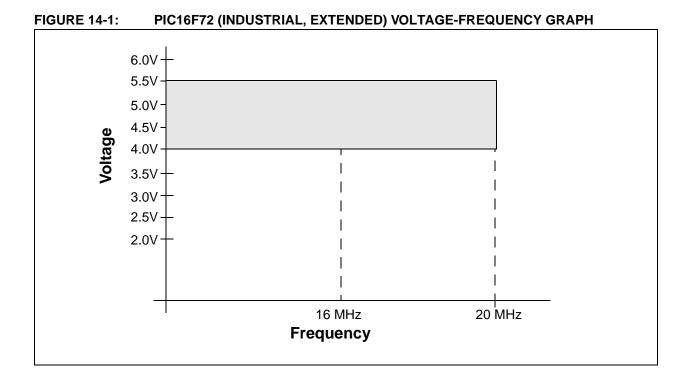
## 13.11 PICDEM 1 Low Cost PIC Demonstration Board

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 incircuit 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.

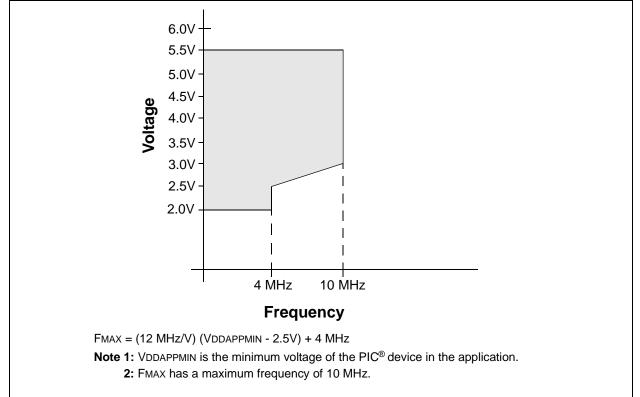
## 13.12 PICDEM 2 Low Cost PIC16CXX Demonstration Board

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  $I^2C^{TM}$  bus and separate headers for connection to an LCD module and a keypad.

# PIC16F72







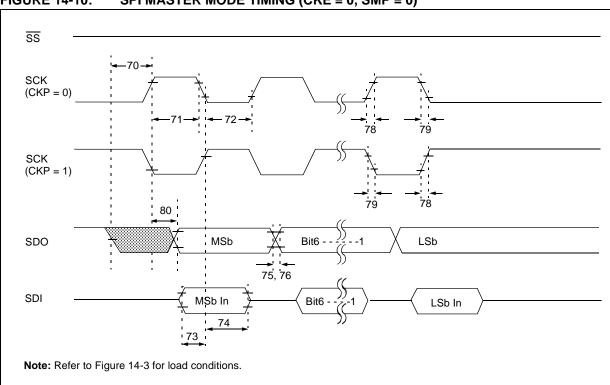
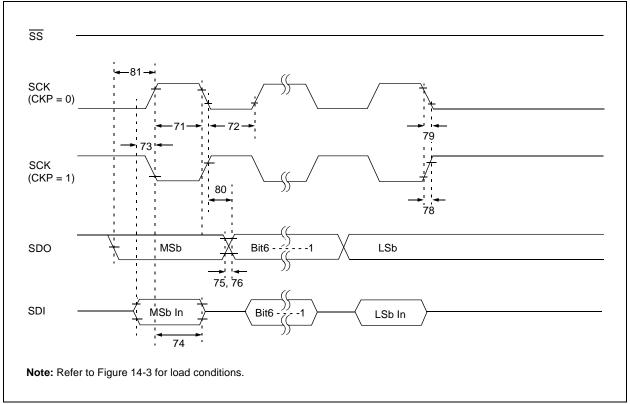


FIGURE 14-10: SPI MASTER MODE TIMING (CKE = 0, SMP = 0)





# **PIC16F72**

Oscillator Configuration	59, 61
Crystal Oscillator/Ceramic Resonators	61
HS	61, 65
LP	61, 65
RC	61, 62, 65
XT	61, 65
Oscillator, WDT	70

## Ρ

-
P44
Package Marking Information117
PCFG0 bit54
PCFG1 bit
PCFG2 bit
PCL Register
PCLATH Register
PCON Register
POR bit
PICDEM 1 Low Cost PIC
Demonstration Board
PICDEM 17 Demonstration Board
PICDEM 2 Low Cost PIC16CXX
Demonstration Board83
PICDEM 3 Low Cost PIC16CXXX
Demonstration Board84
PICSTART Plus Entry Level
Development Programmer83
Pin Functions
MCLR/VPP
OSC1/CLKI
OSC2/CLKO
RA0/AN0
RA1/AN1
RA2/AN26
RA3/AN3/VREF6
RA4/T0CKI
RA5/AN4/SS6
RB0/INT6
RB16
RB26
RB3
RB4
RB5
RB6/PGC
RB7/PGD
RC0/T1OSO/T1CKI
RC1/T10SI6
RC2/CCP16
RC3/SCK/SCL6
RC4/SDI/SDA6
RC5/SDO6
RC66
RC76
VDD6
Vss6
Pinout Descriptions
PIC16F72
POP
POP
PORTA
Associated Registers
Functions22

PORTA Register PORTB		9
Associated Registers		24
Functions		
Pull-up Enable (RBPU bit)		13
RB0/INT Edge Select (INTEDG bit)		13
RB0/INT Pin, External		
RB7:RB4 Interrupt-on-Change Flag (RBIF bit)		14
RB7:RB4 Interrupt-on-Change		69
RB7:RB4 Interrupt-on-Change Enable		
(RBIE bit)		69
RB7:RB4 Interrupt-on-Change Flag		
(RBIF bit)	. 14,	69
PORTB Register		9
Associated Registers		26
Functions		26
PORTC Register		9
Postscaler, WDT		
Assignment (PSA Bit)		13
Rate Select (PS2:PS0 bits)		13
Power-down Mode. See SLEEP		
Power-on Reset (POR) 59, 62, 64	, 65,	66
Brown-out Reset (BOR)		64
Oscillator Start-up Timer (OST)	. 59,	64
POR Status (POR bit)		
Power Control/Status Register (PCON)		64
Power-down (PD bit)		
Power-up Timer (PWRT)		
Time-out (TO bit)		
Time-out Sequence		
PR2 Register		
Prescaler, Timer0		
Assignment (PSA bit)		13
Rate Select (PS2:PS0 bits)		13
PRO MATE II Universal Device Programmer		
Product Identification System	1	33
Program Counter		
RESET Conditions		65
Program Memory		
Paging		19
Program Memory Map and Stack		. 7
Program Verification		
PUSH		

## R

R/W	44
R/W bit	49
RBIF bit	
Read/Write bit Information, R/W	44
Reader Response	132
Reading Program Memory	27
PMADR	27
PMCON1 Register	27
Receive Overflow Indicator bit, SSPOV	45
Register File Map	8

Registers	
ADCON0 (A/D Control 0)	
ADCON1 (A/D Control 1)	54
CCPCON1 (Capture/Compare/PWM Control 1)	37
Initialization Conditions (table)	
INTCON (Interrupt Control)	
OPTION	
PCON (Power Control)	
PIE1 (Peripheral Interrupt Enable 1)	
PIR1 (Peripheral Interrupt Flag 1)	
PMCON1 (Program Memory Control 1)	
SSPCON (Sync Serial Port Control)	
SSPSTAT (Synchronous Serial Port Status)	44
STATUS	12
Summary	
T1CON (Timer1 Control)	
RESET	
Brown-out Reset (BOR). See Brown-out Reset (BOR)	
MCLR RESET. See MCLR	
Power-on Reset (POR). See Power-on Reset (POR)	
RESET Conditions for All Registers	
RESET Conditions for PCON Register	65
RESET Conditions for Program Counter	
RESET Conditions for STATUS Register	65
WDT Reset. See Watchdog Timer (WDT)	
Revision History12	23
RP0, RP1 bit	
	••
S	
S	11
Sales and Support1	33
Slave Mode	
SCL	-
SDA	48
SLEEP	71
SMP	44
Software Simulator (MPLAB SIM)	82
Special Event Trigger	
Special Features of the CPU	
Special Function Registers	
PMADRH	77
PMADRL	
PMCON1	
PMDATH	
PMDATL	27
SPI	
Associated Registers	46
SPI Clock Edge Select bit, CKE	
SPI Data Input Sample Phase Select bit, SMP	
SPI Mode	•••
Serial Clock	<u>4</u> 2
Serial Data In	-
	+0

	Serial Data In	
	Serial Data Out	
	Slave Select	
SSP		
	ACK	
	Addressing	
	BF bit	
	I <sup>2</sup> C Mode Operation	
	R/W bit	
	Reception	
	SCL Clock Input	
	SSPOV bit	
	Transmission	-

SSPADD Register	. 10
SSPEN	. 45
SSPIF	. 16
SSPM3:SSPM0	. 45
SSPOV	
SSPSTAT Register	
Stack	
Overflows	
Underflow	. 19
START bit, S	
STATUS Register	
DC bit	. 12
IRP bit	. 12
PD bit	. 62
TO bit	, 62
STOP bit, P	
Synchronous Serial Port (SSP)	
Overview	
SPI Mode	. 43
Synchronous Serial Port Enable bit, SSPEN	. 45
Synchronous Serial Port Interrupt	. 16
Synchronous Serial Port Mode Select bits,	
SSPM3:SSPM0	. 45

#### Т

T2CKPS0 bit	. 36	
T2CKPS1 bit		
T2CON (Timer2 Control)	. 36	
Tad		
Timer0	. 29	
Clock Source Edge Select (T0SE bit)	. 13	
Clock Source Select (T0CS bit)		
External Clock		
Interrupt	. 29	
Operation	. 29	
Overflow Enable (TMR0IE bit)	. 14	
Overflow Flag (TMR0IF bit)		
Overflow Interrupt		
Prescaler	. 30	
ТОСКІ	. 30	
Timer1		
Associated Registers	. 34	
Asynchronous Counter Mode	. 33	
Capacitor Selection		
Counter Operation	. 32	
Interrupt	. 33	
Operation in Timer Mode	. 32	
Oscillator	. 33	
Prescaler	. 34	
Resetting TMR1H, TMR1L Register Pair	. 34	
Resetting Using a CCP Trigger Output		
Synchronized Counter Mode		
Timer2	. 35	
Interrupt	. 35	
Operation	. 35	
Output	. 35	
Prescaler, Postscaler	. 35	

## **READER RESPONSE**

It is our intention to provide you with the best documentation possible to ensure successful use of your Microchip product. If you wish to provide your comments on organization, clarity, subject matter, and ways in which our documentation can better serve you, please FAX your comments to the Technical Publications Manager at (480) 792-4150.

Please list the following information, and use this outline to provide us with your comments about this document.

To:	Technical Publications Manager	Total Pages Sent		
RE:	Reader Response			
From	n: Name			
	Company			
	Telephone: ()	FAX: ()		
Appli	ication (optional):			
Would you like a reply?YN				
Devi	ce: DSTEMP	Literature Number: DS00000A		
Ques	stions:			
1. What are the best features of this document?				
2. H	2. How does this document meet your hardware and software development needs?			
_				
_				
3. E	3. Do you find the organization of this document easy to follow? If not, why?			
_				
_				
4. V	What additions to the document do y	you think would enhance the structure and subject?		
_				
- -				
5. V	what deletions from the document c	ould be made without affecting the overall usefulness?		
-				
6. I	here any incorrect or misleading information (what and where)?			
0. 1	s there any monteet of misleading i			
_				
- 7. H	low would you improve this docume	ent?		
_				
-				