

## 28-Pin, 8-Bit CMOS FLASH Microcontroller

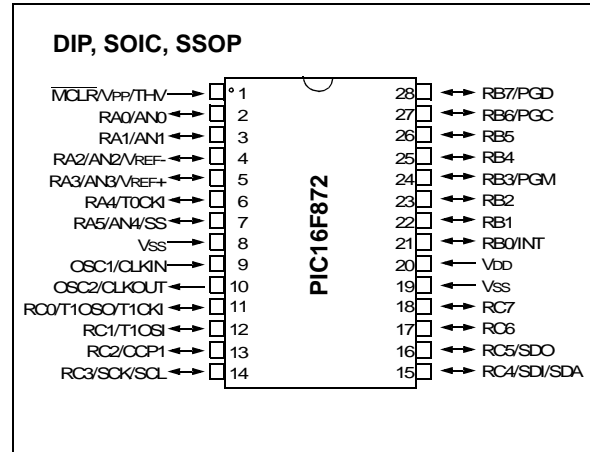
### Devices Included in this Data Sheet:

- PIC16F872

### Microcontroller Core Features:

- 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 FLASH Program Memory  
128 x 8 bytes of Data Memory (RAM)  
64 x 8 bytes of EEPROM Data Memory
- Pinout compatible to the PIC16C72A
- Interrupt capability (up to 10 sources)
- Eight level deep hardware stack
- Direct, indirect and relative addressing modes
- 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
- Low-power, high-speed CMOS FLASH/EEPROM technology
- Fully static design
- In-Circuit Serial Programming™ (ICSP) via two pins
- Single 5V In-Circuit Serial Programming capability
- In-Circuit Debugging via two pins
- Processor read/write access to program memory
- Wide operating voltage range: 2.0V to 5.5V
- High Sink/Source Current: 25 mA
- Commercial and Industrial temperature ranges
- Low-power consumption:
  - < 2 mA typical @ 5V, 4 MHz
  - 20 µA typical @ 3V, 32 kHz
  - < 1 µA typical standby current

### Pin Diagram



### Peripheral Features:

- 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
- One Capture, Compare, PWM 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
- 10-bit multi-channel Analog-to-Digital converter
- Synchronous Serial Port (SSP) with SPI™ (Master Mode) and I<sup>2</sup>C™ (Master/Slave)
- Brown-out detection circuitry for Brown-out Reset (BOR)

# PIC16F872

---

---

<b>Key Features PICmicro™ Mid-Range Reference Manual (DS33023)</b>	<b>PIC16F872</b>
Operating Frequency	DC - 20 MHz
Resets (and Delays)	POR, BOR (PWRT, OST)
FLASH Program Memory (14-bit words)	2K
Data Memory (bytes)	128
EEPROM Data Memory	64
Interrupts	10
I/O Ports	Ports A,B,C
Timers	3
Capture/Compare/PWM module	1
Serial Communications	MSSP
10-bit Analog-to-Digital Module	5 input channels
Instruction Set	35 Instructions

## Table of Contents

1.0 Device Overview .....	5
2.0 Memory Organization.....	7
3.0 I/O Ports.....	23
4.0 Data EEPROM and FLASH Program Memory .....	29
5.0 Timer0 Module .....	37
6.0 Timer1 Module .....	41
7.0 Timer2 Module .....	45
8.0 Capture/Compare/PWM (CCP) Module(s).....	47
9.0 Master Synchronous Serial Port (MSSP) Module.....	53
10.0 Analog-to-Digital Converter (A/D) Module .....	85
11.0 Special Features of the CPU .....	95
12.0 Instruction Set Summary.....	111
13.0 Development Support .....	119
14.0 Electrical Characteristics.....	125
15.0 DC and AC Characteristics Graphs and Tables .....	143
16.0 Packaging Information .....	145
Appendix A: Revision History.....	149
Appendix B: Conversion Considerations.....	149
Index .....	151
On-Line Support.....	157
Product Identification System .....	159

### *To Our Valued Customers*

#### **Most Current Data Sheet**

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

<http://www.microchip.com>

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

#### **New Customer Notification System**

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

#### **Errata**

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

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

- Microchip's Worldwide Web site; <http://www.microchip.com>
- Your local Microchip sales office (see last page)
- The Microchip Corporate Literature Center; U.S. FAX: (480) 786-7277

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

#### **Corrections to this Data Sheet**

We constantly strive to improve the quality of all our products and documentation. We have spent a great deal of time to ensure that this document is correct. However, we realize that we may have missed a few things. If you find any information that is missing or appears in error, please:

- Fill out and mail in the reader response form in the back of this data sheet.
- E-mail us at [webmaster@microchip.com](mailto:webmaster@microchip.com).

We appreciate your assistance in making this a better document.

# PIC16F872

---

NOTES:

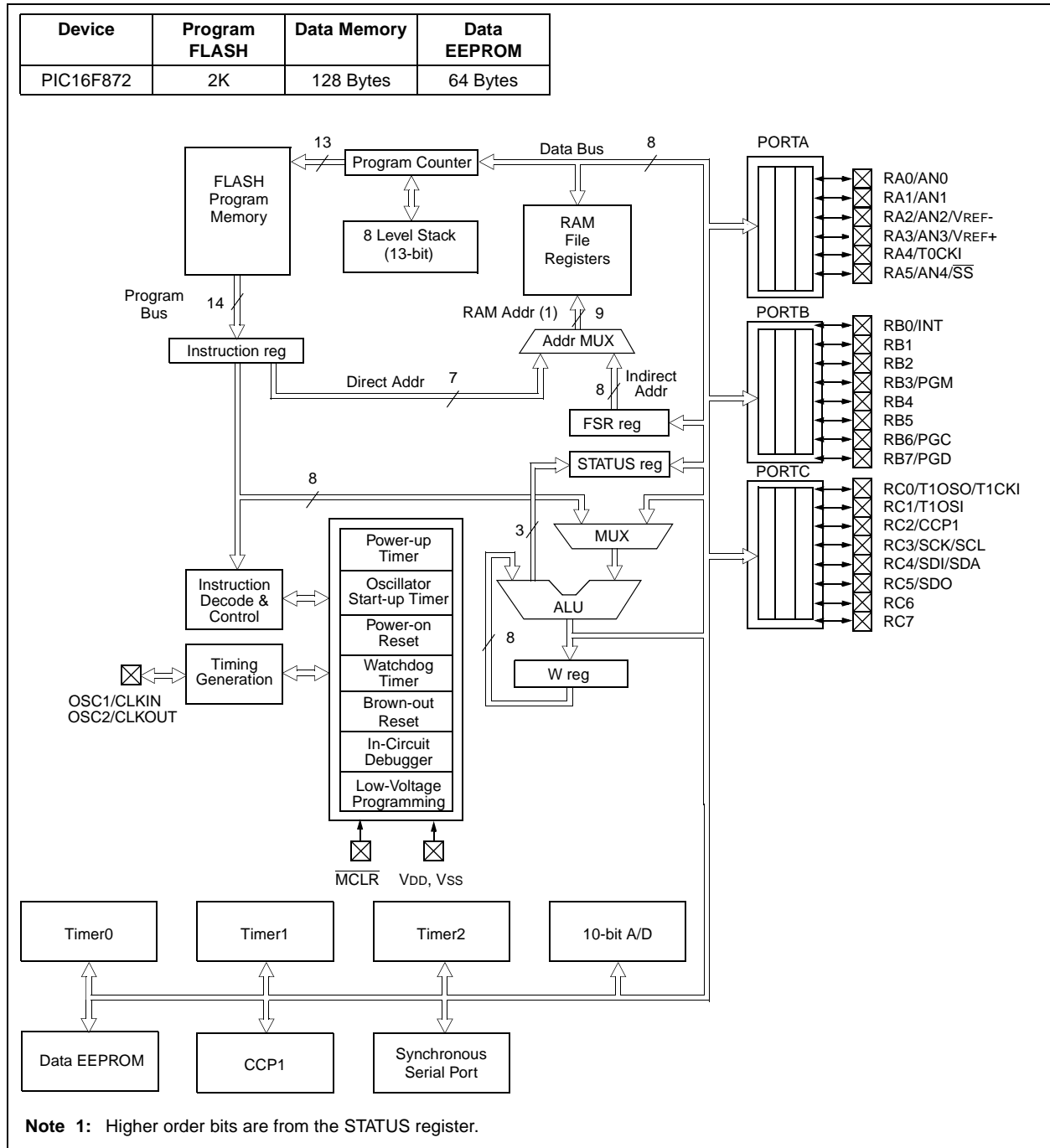
## 1.0 DEVICE OVERVIEW

This document contains device-specific information. Additional information may be found in the PICmicro™ Mid-Range Reference Manual, (DS33023), which may be obtained from your local Microchip Sales Representative or downloaded from the Microchip website. The Reference Manual should be considered a comple-

mentary document to this data sheet, and is highly recommended reading for a better understanding of the device architecture and operation of the peripheral modules.

This data sheet covers the PIC16F872 device. The PIC16F872 is a 28-pin device and its block diagram is shown in Figure 1-1.

**FIGURE 1-1: PIC16F872 BLOCK DIAGRAM**



# PIC16F872

**TABLE 1-1: PIC16F872 PINOUT DESCRIPTION**

Pin Name	DIP Pin#	SOIC Pin#	I/O/P Type	Buffer Type	Description
OSC1/CLKIN	9	9	I	ST/CMOS <sup>(3)</sup>	Oscillator crystal input/external clock source input.
OSC2/CLKOUT	10	10	O	—	Oscillator crystal output. Connects to crystal or resonator in crystal oscillator mode. In RC mode, the OSC2 pin outputs CLKOUT, which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate.
MCLR/VPP/THV	1	1	I/P	ST	Master clear (reset) input or programming voltage input or high voltage test mode control. This pin is an active low reset to the device.
RA0/AN0	2	2	I/O	TTL	<p>PORTA is a bi-directional I/O port.</p> <p>RA0 can also be analog input0.</p> <p>RA1 can also be analog input1.</p> <p>RA2 can also be analog input2 or negative analog reference voltage.</p> <p>RA3 can also be analog input3 or positive analog reference voltage.</p> <p>RA4 can also be the clock input to the Timer0 module. Output is open drain type.</p> <p>RA5 can also be analog input4 or the slave select for the synchronous serial port.</p>
RA1/AN1	3	3	I/O	TTL	
RA2/AN2/VREF-	4	4	I/O	TTL	
RA3/AN3/VREF+	5	5	I/O	TTL	
RA4/T0CKI	6	6	I/O	ST	
RA5/SS/AN4	7	7	I/O	TTL	
RB0/INT	21	21	I/O	TTL/ST <sup>(1)</sup>	<p>PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs.</p> <p>RB0 can also be the external interrupt pin.</p> <p>RB3 can also be the low voltage programming input.</p> <p>Interrupt on change pin.</p> <p>Interrupt on change pin.</p> <p>Interrupt on change pin or In-Circuit Debugger pin. Serial programming clock.</p> <p>Interrupt on change pin or In-Circuit Debugger pin. Serial programming data.</p>
RB1	22	22	I/O	TTL	
RB2	23	23	I/O	TTL	
RB3/PGM	24	24	I/O	TTL/ST <sup>(1)</sup>	
RB4	25	25	I/O	TTL	
RB5	26	26	I/O	TTL	
RB6/PGC	27	27	I/O	TTL/ST <sup>(2)</sup>	
RB7/PGD	28	28	I/O	TTL/ST <sup>(2)</sup>	
RC0/T1OSO/T1CKI	11	11	I/O	ST	<p>PORTC is a bi-directional I/O port.</p> <p>RC0 can also be the Timer1 oscillator output or Timer1 clock input.</p> <p>RC1 can also be the Timer1 oscillator input.</p> <p>RC2 can also be the Capture1 input/Compare1 output/PWM1 output.</p> <p>RC3 can also be the synchronous serial clock input/output for both SPI and I<sup>2</sup>C modes.</p> <p>RC4 can also be the SPI Data In (SPI mode) or data I/O (I<sup>2</sup>C mode).</p> <p>RC5 can also be the SPI Data Out (SPI mode).</p>
RC1/T1OSI	12	12	I/O	ST	
RC2/CCP1	13	13	I/O	ST	
RC3/SCK/SCL	14	14	I/O	ST	
RC4/SDI/SDA	15	15	I/O	ST	
RC5/SDO	16	16	I/O	ST	
RC6	17	17	I/O	ST	
RC7	18	18	I/O	ST	
VSS	8, 19	8, 19	P	—	Ground reference for logic and I/O pins.
VDD	20	20	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 or LVP.  
**2:** This buffer is a Schmitt Trigger input when used in serial programming mode.  
**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 three memory blocks in each of these PICmicro® MCUs. The Program Memory and Data Memory have separate buses, so that concurrent access can occur, and is detailed in this section. The EEPROM data memory block is detailed in Section 4.0.

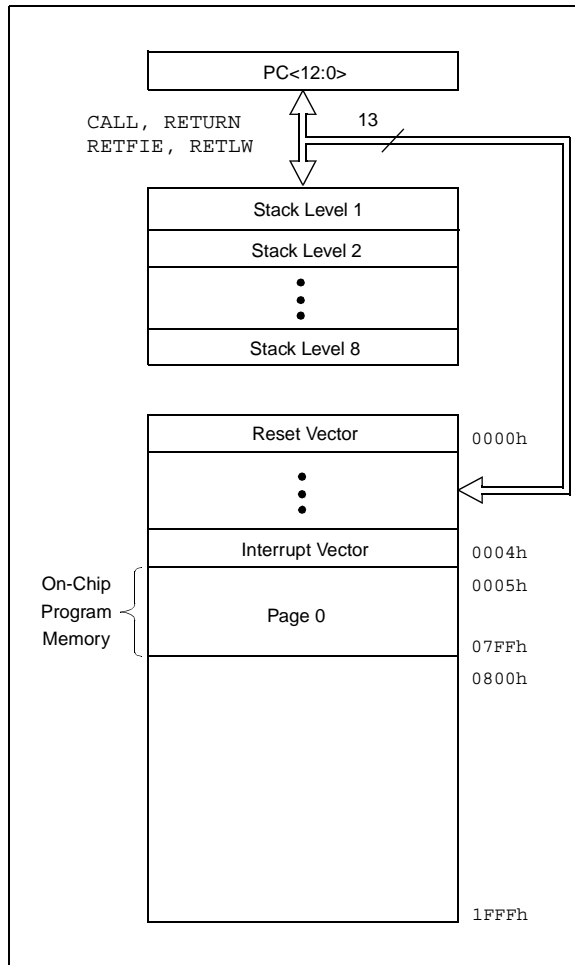
Additional information on device memory may be found in the PICmicro™ Mid-Range Reference Manual, (DS33023).

### 2.1 Program Memory Organization

The PIC16F872 devices have a 13-bit program counter capable of addressing an 8K x 14 program memory space. The PIC16F872 device has 2K x 14 words of FLASH program memory. Accessing a location above the physically implemented address will cause a wrap-around.

The reset vector is at 0000h and the interrupt vector is at 0004h.

**FIGURE 2-1: PIC16F872 PROGRAM MEMORY MAP AND STACK**



## 2.2 Data Memory Organization

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

RP<1:0>	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 Special Function Registers. Some "high use" Special Function Registers from one bank may be mirrored in another bank for code reduction and quicker access.

**Note:** EEPROM Data Memory description can be found in Section 4.0 of this Data Sheet

### 2.2.1 GENERAL PURPOSE REGISTER FILE

The register file can be accessed either directly, or indirectly through the File Select Register FSR.

# PIC16F872

**FIGURE 2-2: PIC16F872 REGISTER FILE MAP**

						File Address	
Indirect addr. (*)	00h	Indirect addr. (*)	80h	Indirect addr. (*)	100h	Indirect addr. (*)	180h
TMR0	01h	OPTION_REG	81h	TMR0	101h	OPTION_REG	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	EEDATA	10Ch	EECON1	18Ch
PIR2	0Dh	PIE2	8Dh	EEADR	10Dh	EECON2	18Dh
TMR1L	0Eh	PCON	8Eh	EEDATH	10Eh	Reserved <sup>(1)</sup>	18Eh
TMR1H	0Fh		8Fh	EEADRH	10Fh	Reserved <sup>(1)</sup>	18Fh
T1CON	10h		90h		110h		190h
TMR2	11h	SSPCON2	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				
ADRESH	1Eh	ADRESL	9Eh				
ADCON0	1Fh	ADCON1	9Fh				
	20h	General Purpose Register	A0h				
		32 Bytes	BFh	accesses 20h-7Fh	120h	accesses A0h - BFh	1A0h
			C0h				1BFh
			EFh				1C0h
			F0h		16Fh		1EFh
		accesses 70h-7Fh	FFh	accesses 70h-7Fh	170h	accesses 70h-7Fh	1F0h
					17Fh		1FFh
Bank 0		Bank 1		Bank 2		Bank 3	
General Purpose Register 96 Bytes		General Purpose Register 32 Bytes		accesses 20h-7Fh		accesses A0h - BFh	

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

**Note 1:** These registers are reserved; maintain these registers clear.



## 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	Value on all other resets (2)		
<b>Bank 0</b>													
00h <sup>(3)</sup>	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	0000 0000		
01h	TMR0	Timer0 module's register								xxxx xxxx	uuuu uuuu		
02h <sup>(3)</sup>	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	0000 0000		
03h <sup>(3)</sup>	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	000q quuu		
04h <sup>(3)</sup>	FSR	Indirect data memory address pointer								xxxx xxxx	uuuu uuuu		
05h	PORTA	—	—	PORTA Data Latch when written: PORTA pins when read								--0x 0000	--0u 0000
06h	PORTB	PORTB Data Latch when written: PORTB pins when read								xxxx xxxx	uuuu uuuu		
07h	PORTC	PORTC Data Latch when written: PORTC pins when read								xxxx xxxx	uuuu uuuu		
08h	—	Unimplemented								—	—		
09h	—	Unimplemented								—	—		
0Ah <sup>(1,3)</sup>	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	---0 0000		
0Bh <sup>(3)</sup>	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u		
0Ch	PIR1	(4)	ADIF	(4)	(4)	SSPIF	CCP1IF	TMR2IF	TMR1IF	r0rr 0000	r0rr 0000		
0Dh	PIR2	—	(4)	—	EEIF	BCLIF	—	—	(4)	-r-0 0--r	-r-0 0--r		
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	T1OSCN	T1SYNC	TMR1CS	TMR1ON	--00 0000	--uu uuuu		
11h	TMR2	Timer2 module's register								0000 0000	0000 0000		
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000		
13h	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								xxxx xxxx	uuuu uuuu		
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 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		
18h	—	Unimplemented								—	—		
19h	—	Unimplemented								—	—		
1Ah	—	Unimplemented								—	—		
1Bh	—	Unimplemented								—	—		
1Ch	—	Unimplemented								—	—		
1Dh	—	Unimplemented								—	—		
1Eh	ADRESH	A/D Result Register High Byte								xxxx xxxx	uuuu uuuu		
1Fh	ADCON0	ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/ DONE	—	ADON	0000 00-0	0000 00-0		

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:** 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.
- 2:** Other (non power-up) resets include external reset through  $\overline{MCLR}$  and Watchdog Timer Reset.
- 3:** These registers can be addressed from any bank.
- 4:** These bits are reserved; always maintain these bits clear.

# PIC16F872

**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	Value on all other resets (2)		
<b>Bank 1</b>													
80h <sup>(3)</sup>	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	0000 0000		
81h	OPTION_REG	RBP $\bar{U}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111		
82h <sup>(3)</sup>	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	0000 0000		
83h <sup>(3)</sup>	STATUS	IRP	RP1	RP0	$\bar{T}O$	$\bar{P}D$	Z	DC	C	0001 1xxx	000q quuu		
84h <sup>(3)</sup>	FSR	Indirect data memory address pointer								xxxx xxxx	uuuu uuuu		
85h	TRISA	—	—	PORTA Data Direction Register								--11 1111	--11 1111
86h	TRISB	PORTB Data Direction Register								1111 1111	1111 1111		
87h	TRISC	PORTC Data Direction Register								1111 1111	1111 1111		
88h	—	Unimplemented								—	—		
89h	—	Unimplemented								—	—		
8Ah <sup>(1,3)</sup>	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	---0 0000		
8Bh <sup>(3)</sup>	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u		
8Ch	PIE1	(4)	ADIE	(4)	(4)	SSPIE	CCP1IE	TMR2IE	TMR1IE	r0rr 0000	r0rr 0000		
8Dh	PIE2	—	(4)	—	EEIE	BCLIE	—	—	(4)	-r-0 0--r	-r-0 0--r		
8Eh	PCON	—	—	—	—	—	—	$\bar{P}OR$	BOR	---- --qg	---- --uu		
8Fh	—	Unimplemented								—	—		
90h	—	Unimplemented								—	—		
91h	SSPCON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	0000 0000	0000 0000		
92h	PR2	Timer2 Period Register								1111 1111	1111 1111		
93h	SSPADD	Synchronous Serial Port (I <sup>2</sup> C mode) Address Register								0000 0000	0000 0000		
94h	SSPSTAT	SMP	CKE	D/ $\bar{A}$	P	S	R/ $\bar{W}$	UA	BF	0000 0000	0000 0000		
95h	—	Unimplemented								—	—		
96h	—	Unimplemented								—	—		
97h	—	Unimplemented								—	—		
98h	—	Unimplemented								—	—		
99h	—	Unimplemented								—	—		
9Ah	—	Unimplemented								—	—		
9Bh	—	Unimplemented								—	—		
9Ch	—	Unimplemented								—	—		
9Dh	—	Unimplemented								—	—		
9Eh	ADRESL	A/D Result Register Low Byte								xxxx xxxx	uuuu uuuu		
9Fh	ADCON1	ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0	0--- 0000	0--- 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:** 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.
- 2:** Other (non power-up) resets include external reset through MCLR and Watchdog Timer Reset.
- 3:** These registers can be addressed from any bank.
- 4:** These bits are reserved; always maintain these bits clear.

**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	Value on all other resets (2)
<b>Bank 2</b>											
100h <sup>(3)</sup>	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	0000 0000
101h	TMR0	Timer0 module's register								xxxx xxxx	uuuu uuuu
102h <sup>(3)</sup>	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	0000 0000
103h <sup>(3)</sup>	STATUS	IRP	RP1	RP0	TO	PD	Z	DC	C	0001 1xxx	000q quuu
104h <sup>(3)</sup>	FSR	Indirect data memory address pointer								xxxx xxxx	uuuu uuuu
105h	—	Unimplemented								—	—
106h	PORTB	PORTB Data Latch when written: PORTB pins when read								xxxx xxxx	uuuu uuuu
107h	—	Unimplemented								—	—
108h	—	Unimplemented								—	—
109h	—	Unimplemented								—	—
10Ah <sup>(1,3)</sup>	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	---0 0000
10Bh <sup>(3)</sup>	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
10Ch	EEDATA	EEPROM data register								xxxx xxxx	uuuu uuuu
10Dh	EEADR	EEPROM address register								xxxx xxxx	uuuu uuuu
10Eh	EEDATH	—	—	EEPROM data register high byte					xxxx xxxx	uuuu uuuu	
10Fh	EEADRH	—	—	—	EEPROM address register high byte					xxxx xxxx	uuuu uuuu
<b>Bank 3</b>											
180h <sup>(3)</sup>	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	0000 0000
181h	OPTION_REG	RBP $\bar{U}$	INTEDG	TOCS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
182h <sup>(3)</sup>	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	0000 0000
183h <sup>(3)</sup>	STATUS	IRP	RP1	RP0	TO	PD	Z	DC	C	0001 1xxx	000q quuu
184h <sup>(3)</sup>	FSR	Indirect data memory address pointer								xxxx xxxx	uuuu uuuu
185h	—	Unimplemented								—	—
186h	TRISB	PORTB Data Direction Register								1111 1111	1111 1111
187h	—	Unimplemented								—	—
188h	—	Unimplemented								—	—
189h	—	Unimplemented								—	—
18Ah <sup>(1,3)</sup>	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	---0 0000
18Bh <sup>(3)</sup>	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
18Ch	EECON1	EEPGD	—	—	—	WRERR	WREN	WR	RD	x--- x000	x--- u000
18Dh	EECON2	EEPROM control register2 (not a physical register)								---- ----	---- ----
18Eh	—	Reserved maintain clear								0000 0000	0000 0000
18Fh	—	Reserved maintain clear								0000 0000	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:** 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.
- 2:** Other (non power-up) resets include external reset through  $\overline{MCLR}$  and Watchdog Timer Reset.
- 3:** These registers can be addressed from any bank.
- 4:** These bits are reserved; always maintain these bits clear.

# PIC16F872

## 2.2.2.1 STATUS REGISTER

The STATUS register 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  $\overline{TO}$  and  $\overline{PD}$  bits are not writable, therefore, the result of an instruction with the STATUS register as destination may be different than intended.

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

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 the "Instruction Set Summary."

**Note:** The C and 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	$\overline{TO}$	$\overline{PD}$	Z	DC	C
bit7								bit0
bit 7:	<b>IRP:</b> Register Bank Select bit (used for indirect addressing) 1 = Bank 2, 3 (100h - 1FFh) 0 = Bank 0, 1 (00h - FFh)							
bit 6-5:	<b>RP&lt;1:0&gt;:</b> 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:	<b><math>\overline{TO}</math>:</b> Time-out bit 1 = After power-up, <code>CLRWDT</code> instruction, or <code>SLEEP</code> instruction 0 = A WDT time-out occurred							
bit 3:	<b><math>\overline{PD}</math>:</b> Power-down bit 1 = After power-up or by the <code>CLRWDT</code> instruction 0 = By execution of the <code>SLEEP</code> instruction							
bit 2:	<b>Z:</b> 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:	<b>DC:</b> Digit carry/borrow bit ( <code>ADDWF</code> , <code>ADDLW</code> , <code>SUBLW</code> , <code>SUBWF</code> instructions) (for borrow the polarity is reversed) 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:	<b>C:</b> Carry/borrow bit ( <code>ADDWF</code> , <code>ADDLW</code> , <code>SUBLW</code> , <code>SUBWF</code> instructions) 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 <b>Note:</b> For borrow the polarity is reversed. A subtraction is executed by adding the two's complement of the second operand. For rotate ( <code>RRF</code> , <code>RLF</code> ) instructions, this bit is loaded with either the high or low order bit of the source register.							

R = Readable bit  
 W = Writable bit  
 U = Unimplemented bit, read as '0'  
 - n = Value at POR reset

## 2.2.2.2 OPTION\_REG REGISTER

The OPTION\_REG Register is a readable and writable register, which 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\_REG 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
RBP $\bar{U}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit7							bit0

R = Readable bit  
 W = Writable bit  
 U = Unimplemented bit, read as '0'  
 - n= Value at POR reset

bit 7: **RBP $\bar{U}$** : 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 (CLKOUT)

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: **PS<2:0>**: 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

**Note:** When using Low Voltage ICSP Programming (LVP) and the pull-ups on PORTB are enabled, bit 3 in the TRISB register must be cleared to disable the pull-up on RB3 and ensure the proper operation of the device.

# PIC16F872

## 2.2.2.3 INTCON REGISTER

The INTCON Register is a readable and writable register, which 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 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-x
bit7	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF
							RBIF bit0

R = Readable bit  
 W = Writable bit  
 U = Unimplemented bit, read as '0'  
 - n= Value at POR reset

bit 7: **GIE:** Global Interrupt Enable bit  
 1 = Enables all un-masked interrupts  
 0 = Disables all interrupts

bit 6: **PEIE:** Peripheral Interrupt Enable bit  
 1 = Enables all un-masked peripheral interrupts  
 0 = Disables all peripheral interrupts

bit 5: **TOIE:** 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: **TOIF:** 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  
 1 = At least one of the RB<7:4> pins changed state (must be cleared in software)  
 0 = None of the RB<7:4> pins have changed state

## 2.2.2.4 PIE1 REGISTER

The PIE1 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 REGISTER (ADDRESS 8Ch)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	ADIE	—	—	SSPIE	CCP1IE	TMR2IE	TMR1IE
bit7							bit0

R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as '0'  
- n= Value at POR reset

bit 7: **Reserved:** Always maintain this bit clear

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: **Reserved:** Always maintain this bit clear

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

# PIC16F872

## 2.2.2.5 PIR1 REGISTER

The PIR1 register contains the individual flag bits for the peripheral 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 bits are clear prior to enabling an interrupt.

### REGISTER 2-5: PIR1 REGISTER (ADDRESS 0Ch)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	ADIF	—	—	SSPIF	CCP1IF	TMR2IF	TMR1IF

bit7

bit0

R = Readable bit  
 W = Writable bit  
 U = Unimplemented bit, read as '0'  
 - n= Value at POR reset

- bit 7: **Reserved:** Always maintain this bit clear
- 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: **Reserved:** Always maintain this bit clear
- bit 3: **SSPIF:** Synchronous Serial Port (SSP) Interrupt Flag  
 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:  
SPI  
 A transmission/reception has taken place.  
I<sup>2</sup>C Slave  
 A transmission/reception has taken place.  
I<sup>2</sup>C Master  
 A transmission/reception has taken place.  
 The initiated start condition was completed by the SSP module.  
 The initiated stop condition was completed by the SSP module.  
 The initiated restart condition was completed by the SSP module.  
 The initiated acknowledge condition was completed by the SSP module.  
 A start condition occurred while the SSP module was idle (Multimaster system).  
 A stop condition occurred while the SSP module was idle (Multimaster system).  
 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



## 2.2.2.6 PIE2 REGISTER

The PIE2 register contains the individual enable bits for the SSP bus collision interrupt and the EEPROM write operation interrupt.

### REGISTER 2-6: PIE2 REGISTER (ADDRESS 8Dh)

U-0	R/W-0	U-0	R/W-0	R/W-0	U-0	U-0	R/W-0
—	—	—	EEIE	BCLIE	—	—	—
bit7							bit0

R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as '0'  
- n = Value at POR reset

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

bit 6: **Reserved:** Always maintain this bit clear

bit 5: **Unimplemented:** Read as '0'

bit 4: **EEIE:** EEPROM Write Operation Interrupt Enable  
1 = Enable EE Write Interrupt  
0 = Disable EE Write Interrupt

bit 3: **BCLIE:** Bus Collision Interrupt Enable  
1 = Enable Bus Collision Interrupt  
0 = Disable Bus Collision Interrupt

bit 2-1: **Unimplemented:** Read as '0'

bit 0: **Reserved:** Always maintain this bit clear

# PIC16F872

## 2.2.2.7 PIR2 REGISTER

The PIR2 register contains the flag bits for the SSP bus collision interrupt and the EEPROM write operation interrupt.

**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 2-7: PIR2 REGISTER (ADDRESS 0Dh)

U-0	R/W-0	U-0	R/W-0	R/W-0	U-0	U-0	R/W-0	
—	—	—	EEIF	BCLIF	—	—	—	
bit7								bit0

R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as '0'  
- n = Value at POR reset

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

bit 6: **Reserved:** Always maintain this bit clear

bit 5: **Unimplemented:** Read as '0'

bit 4: **EEIF:** EEPROM Write Operation Interrupt Flag bit  
1 = The write operation completed (must be cleared in software)  
0 = The write operation is not complete or has not been started

bit 3: **BCLIF:** Bus Collision Interrupt Flag  
1 = A bus collision has occurred in the SSP, when configured for I<sup>2</sup>C master mode  
0 = No bus collision has occurred

bit 2-1: **Unimplemented:** Read as '0'

bit 0: **Reserved:** Always maintain this bit clear

## 2.2.2.8 PCON REGISTER

The Power Control (PCON) Register contains flag bits to allow differentiation between a Power-on Reset (POR), a Brown-out Reset (BOR), a Watch-dog Reset (WDT) and an external MCLR Reset.

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

### REGISTER 2-8: PCON REGISTER (ADDRESS 8Eh)

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-1
—	—	—	—	—	—	$\overline{\text{POR}}$	$\overline{\text{BOR}}$
bit7						bit0	

R = Readable bit  
 W = Writable bit  
 U = Unimplemented bit, read as '0'  
 - n = Value at POR reset

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

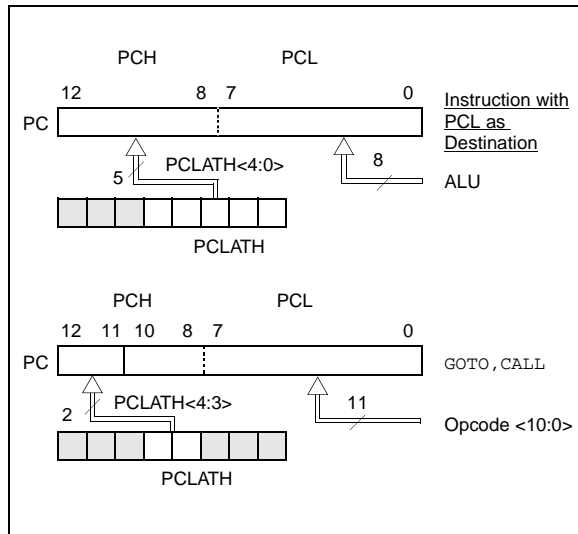
bit 1:  **$\overline{\text{POR}}$ :** Power-on Reset Status bit  
 1 = No Power-on Reset occurred  
 0 = A Power-on Reset occurred (must be set in software after a Power-on Reset occurs)

bit 0:  **$\overline{\text{BOR}}$ :** 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)

## 2.3 PCL and PCLATH

The program counter (PC) is 13-bits wide. The low byte comes from the PCL register, which is a readable and writable register. The upper bits (PC<12:8>) are not readable, but are indirectly writable through the PCLATH register. On any reset, the upper bits of the PC will be cleared. Figure 2-3 shows the two situations for the loading of the PC. The upper example in the figure shows how the PC is loaded on a write to PCL (PCLATH<4:0> → PCH). The lower example in the figure shows how the PC is loaded during a CALL or GOTO instruction (PCLATH<4:3> → PCH).

**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 PIC16CXX family has 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 PUSHed onto the stack when a CALL instruction is executed or an interrupt causes a branch. The stack is POPed in the event of a RETURN, RETLW or a RETFIE instruction execution. PCLATH is not affected by a PUSH or POP operation.

The stack operates as a circular buffer. This means that after the stack has been PUSHed 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).

**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 PIC16CXXX architecture is capable of addressing a continuous 8K word block of program memory. The CALL and GOTO instructions provide 11 bits of the address, which allows branches within any 2K program memory page. Therefore, the 8K words of program memory are broken into four pages. Since the PIC16FC872 has only 2K words of program memory or one page, additional code is not required to ensure that the correct page is selected before a CALL or GOTO instruction is executed. The PCLATH<4:3> bits should always be maintained as zeros. If a return from a CALL instruction (or interrupt) is executed, the entire 13-bit PC is popped off the stack. Manipulation of the PCLATH is not required for the return instructions.

## 2.5 Indirect Addressing, INDF and FSR Registers

The INDF register is not a physical register. Addressing the INDF register will cause indirect addressing.

Indirect addressing is possible by using the INDF register. Any instruction using the INDF register actually accesses the register pointed to by the File Select Register, FSR. Reading the INDF register itself indirectly (FSR = '0') will read 00h. Writing to the INDF register indirectly results in a no-operation (although status bits may be affected). An effective 9-bit address is obtained by concatenating the 8-bit FSR register and the IRP bit (STATUS<7>), as shown in Figure 2-4.

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

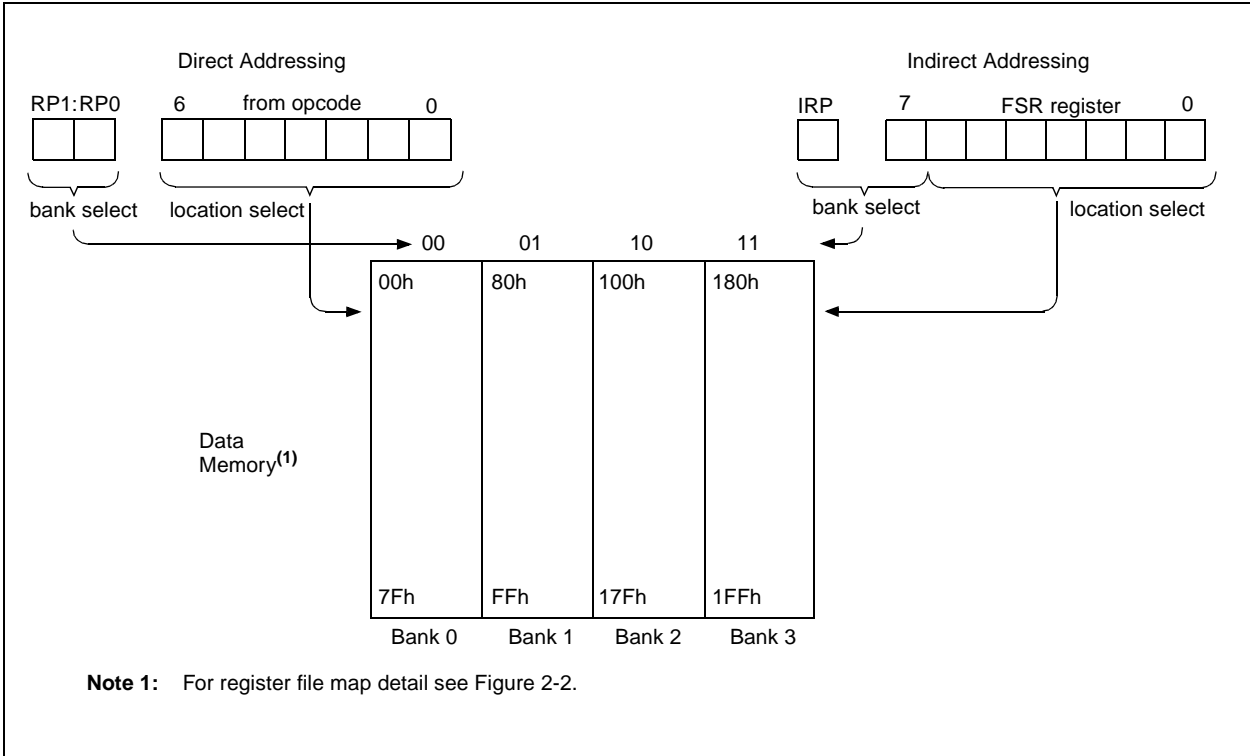
### EXAMPLE 2-1: INDIRECT ADDRESSING

```

movlw 0x20 ;initialize pointer
movwf FSR ;to RAM
NEXT   clrf INDF ;clear INDF register
       incf FSR,F ;inc pointer
       btfs FSR,4 ;all done?
       goto NEXT ;no clear next

CONTINUE
       : ;yes continue
    
```

FIGURE 2-4: DIRECT/INDIRECT ADDRESSING



# PIC16F872

---

NOTES:

## 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 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, the 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 PORTA 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).

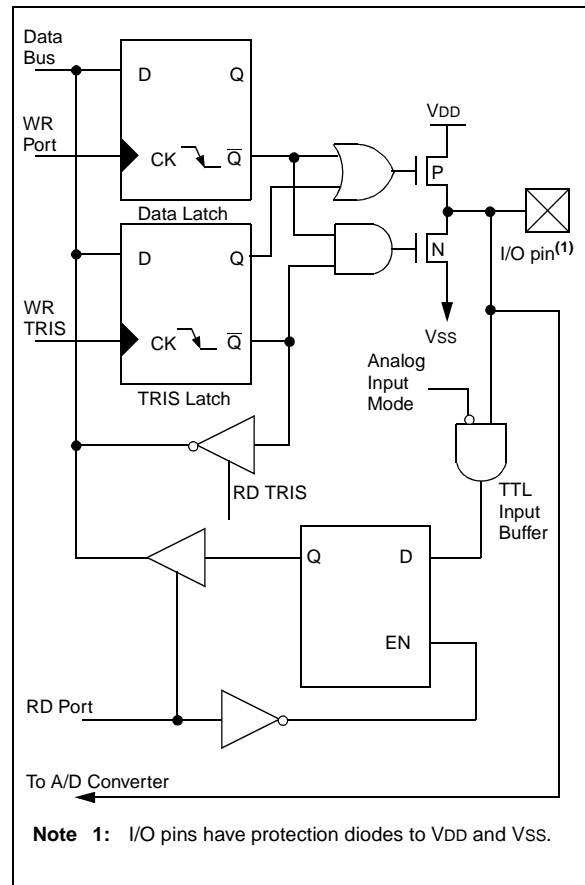
**Note:** On a Power-on Reset, these pins are configured as analog inputs and read as '0'.

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.

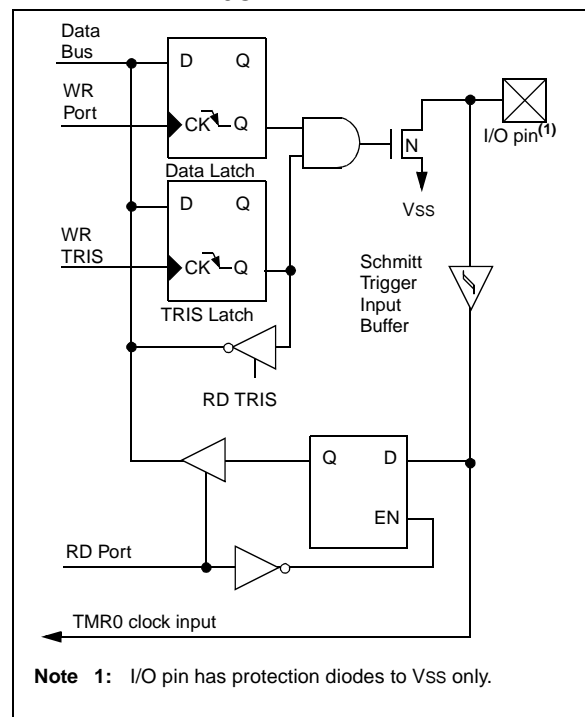
#### EXAMPLE 3-1: INITIALIZING PORTA

```
BCF STATUS, RP0 ;
BCF STATUS, RP1 ; Bank0
CLRF PORTA ; Initialize PORTA by
; clearing output
; data latches
BSF STATUS, RP0 ; Select Bank 1
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 RA<3:0> AND RA5 PINS**



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



# PIC16F872

**TABLE 3-1: PORTA FUNCTIONS**

Name	Bit#	Buffer	Function
RA0/AN0	bit0	TTL	Input/output or analog input
RA1/AN1	bit1	TTL	Input/output or analog input
RA2/AN2	bit2	TTL	Input/output or analog input
RA3/AN3/VREF	bit3	TTL	Input/output or analog input or VREF
RA4/T0CKI	bit4	ST	Input/output or external clock input for Timer0 Output is open drain type
RA5/ $\overline{SS}$ /AN4	bit5	TTL	Input/output or slave select input for synchronous serial port or analog input

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	ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0	--0- 0000	--0- 0000

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 converter must be set to one of the following modes where PCFG<3:0> = 0100, 0101, 011x, 1101, 1110, 1111.



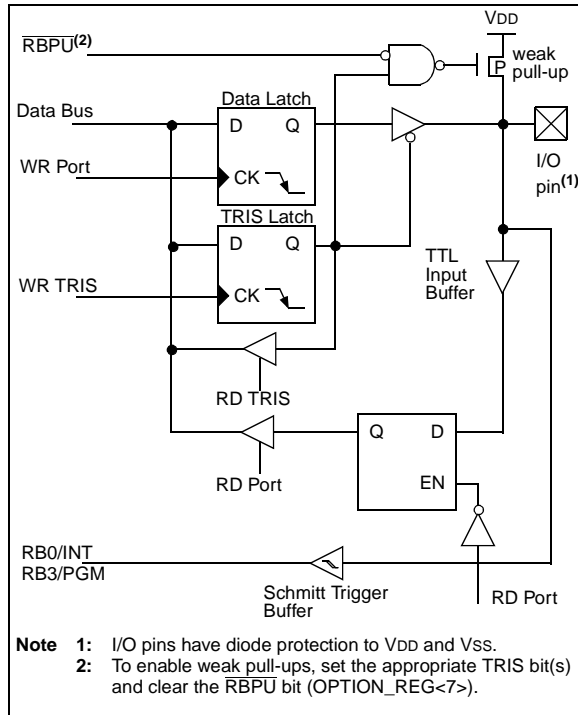
## 3.2 PORTB and the TRISB Register

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).

Three pins of PORTB are multiplexed with the Low Voltage Programming function; RB3/PGM, RB6/PGC and RB7/PGD. The alternate functions of these pins are described in the Special Features Section.

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\_REG<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 RB<3:0> PINS**



Four of PORTB's pins, RB<7:4>, have an interrupt on change feature. Only pins configured as inputs can cause this interrupt to occur (i.e., any RB<7:4> pin configured as an output is excluded from the interrupt on change comparison). The input pins (of RB<7:4>) are compared with the old value latched on the last read of PORTB. The "mismatch" outputs of RB<7:4> are OR'ed 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.

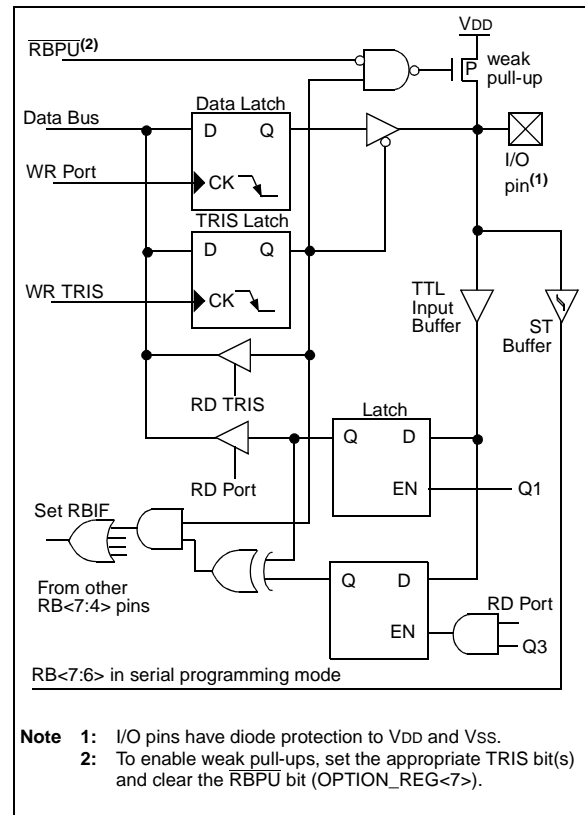
The interrupt on change feature is recommended for wake-up on key depression operation and operations where PORTB is only used for the interrupt on change feature. Polling of PORTB is not recommended while using the interrupt on change feature.

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\_REG<6>).

RB0/INT is discussed in detail in Section 11.10.1.

**FIGURE 3-4: BLOCK DIAGRAM OF RB<7:4> PINS**



**Note:** When using Low Voltage ICSP Programming (LVP) and the pull-ups on PORTB are enabled, bit 3 in the TRISB register must be cleared to disable the pull-up on RB3 and ensure the proper operation of the device.

# PIC16F872

**TABLE 3-3: PORTB FUNCTIONS**

Name	Bit#	Buffer	Function
RB0/INT	bit0	TTL/ST <sup>(1)</sup>	Input/output pin or external interrupt input. Internal software programmable weak pull-up.
RB1	bit1	TTL	Input/output pin. Internal software programmable weak pull-up.
RB2	bit2	TTL	Input/output pin. Internal software programmable weak pull-up.
RB3/PGM	bit3	TTL/ST <sup>(1)</sup>	Input/output pin or programming pin in LVP mode. Internal software programmable weak pull-up.
RB4	bit4	TTL	Input/output pin (with interrupt on change). Internal software programmable weak pull-up.
RB5	bit5	TTL	Input/output pin (with interrupt on change). Internal software programmable weak pull-up.
RB6/PGC	bit6	TTL/ST <sup>(2)</sup>	Input/output pin (with interrupt on change) or In-Circuit Debugger pin. Internal software programmable weak pull-up. Serial programming clock.
RB7/PGD	bit7	TTL/ST <sup>(2)</sup>	Input/output pin (with interrupt on change) or In-Circuit Debugger pin. 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 or LVP mode.

**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_REG	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

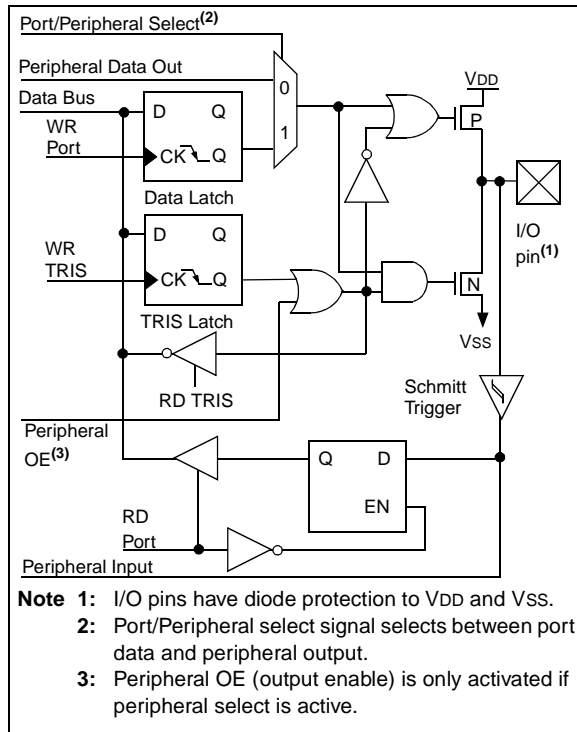
PORTC is an 8-bit wide, bi-directional port. The corresponding data direction register is TRISC. Setting a TRISC bit (=1) will make the corresponding PORTC pin an input (i.e., put the corresponding output driver in a hi-impedance mode). Clearing a TRISC bit (=0) will make the corresponding PORTC pin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

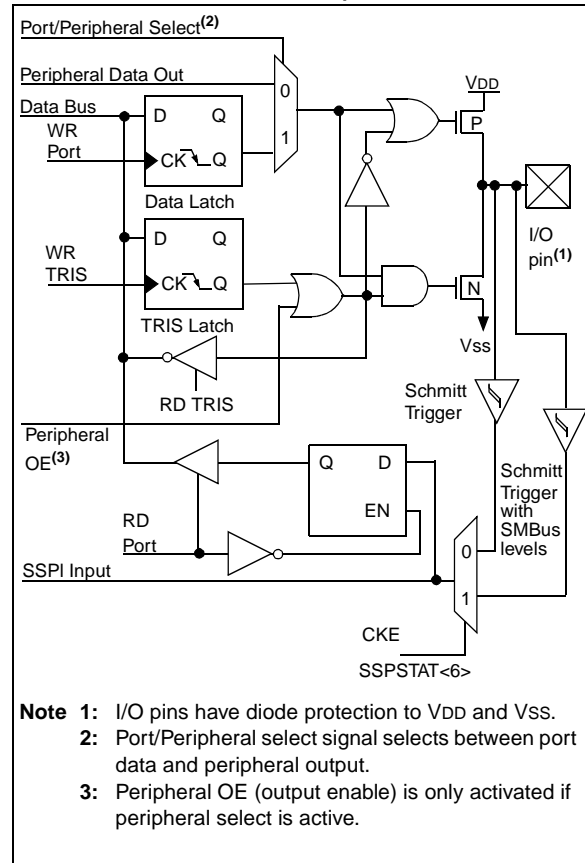
When the I<sup>2</sup>C module is enabled, the PORTC (3:4) pins can be configured with normal I<sup>2</sup>C levels or with SMBUS levels by using the CKE bit (SSPSTAT<6>).

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.

**FIGURE 3-5: PORTC BLOCK DIAGRAM (PERIPHERAL OUTPUT OVERRIDE) RC<0:2> RC<5:7>**



**FIGURE 3-6: PORTC BLOCK DIAGRAM (PERIPHERAL OUTPUT OVERRIDE) RC<3:4>**



# PIC16F872

**TABLE 3-5: PORTC FUNCTIONS**

Name	Bit#	Buffer Type	Function
RC0/T1OSO/T1CKI	bit0	ST	Input/output port pin or Timer1 oscillator output/Timer1 clock input.
RC1/T1OSI	bit1	ST	Input/output port pin or Timer1 oscillator input.
RC2/CCP1	bit2	ST	Input/output port pin or Capture1 input/Compare1 output/PWM1 output.
RC3/SCK/SCL	bit3	ST	RC3 can also be the synchronous serial clock for both SPI and I <sup>2</sup> C modes.
RC4/SDI/SDA	bit4	ST	RC4 can also be the SPI Data In (SPI mode) or data I/O (I <sup>2</sup> C mode).
RC5/SDO	bit5	ST	Input/output port pin or Synchronous Serial Port data output.
RC6	bit6	ST	Input/output port pin.
RC7	bit7	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 DATA EEPROM AND FLASH PROGRAM MEMORY

The Data EEPROM and FLASH Program Memory are readable and writable during normal operation over the entire VDD range. A bulk erase operation may not be issued from user code (which includes removing code protection). The data memory is not directly mapped in the register file space. Instead, it is indirectly addressed through the Special Function Registers (SFR).

There are six SFRs used to read and write the program and data EEPROM memory. These registers are:

- EECON1
- EECON2
- EEDATA
- EEDATH
- EEADR
- EEADRH

The EEPROM data memory allows byte read and write. When interfacing to the data memory block, EEDATA holds the 8-bit data for read/write and EEADR holds the address of the EEPROM location being accessed. The registers EEDATH and EEADRH are not used for data EEPROM access. The PIC16F872 device has 64 bytes of data EEPROM with an address range from 0h to 3Fh.

The EEPROM data memory is rated for high erase/write cycles. The write time is controlled by an on-chip timer. The write time will vary with voltage and temperature, as well as from chip-to-chip. Please refer to the specifications for exact limits.

The program memory allows word reads and writes. Program memory access allows for checksum calculation and calibration table storage. A byte or word write automatically erases the location and writes the new data (erase before write). Writing to program memory will cease operation until the write is complete. The program memory cannot be accessed during the write, therefore code cannot execute. During the write operation, the oscillator continues to clock the peripherals, and therefore, they continue to operate. Interrupt events will be detected and essentially “queued” until the write is completed. When the write completes, the next instruction in the pipeline is executed and the branch to the interrupt vector address will occur.

When interfacing to the program memory block, the EEDATH:EEDATA registers form a two byte word, which holds the 14-bit data for read/write. The EEADRH:EEADR registers form a two byte word, which holds the 13-bit address of the FLASH location being accessed. The PIC16F872 device has 2K words of program FLASH with an address range from 0h to 7FFh. The unused upper bits in both the EEDATH and EEDATA registers all read as “0’s”.

The value written to program memory does not need to be a valid instruction. Therefore, up to 14-bit 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.

### 4.1 EEADR

The address registers can address up to a maximum of 256 bytes of data EEPROM or up to a maximum of 8K words of program FLASH. However, the PIC16F872 has 64 bytes of data EEPROM and 2K words of program FLASH.

When selecting a program address value, the MSByte of the address is written to the EEADRH register and the LSByte is written to the EEADR register. When selecting a data address value, only the LSByte of the address is written to the EEADR register.

On the PIC16F872 device, the upper two bits of the EEADR must always be cleared to prevent inadvertent access to the wrong location in data EEPROM. This also applies to the program memory. The upper five MSbits of EEADRH must always be clear during program FLASH access.

### 4.2 EECON1 and EECON2 Registers

EECON1 is the control register for memory accesses.

EECON2 is not a physical register. Reading EECON2 will read all '0's. The EECON2 register is used exclusively in the memory write sequence.

Control bit EEPGD determines if the access will be a program or a data memory access. When clear, any subsequent operations will operate on the data memory. When set, any subsequent operations will operate on the program memory.

Control bits RD and WR initiate read and write operations, respectively. These bits cannot be cleared, only set, in software. They are cleared in hardware at the completion of the read or write operation. The inability to clear the WR bit in software prevents the accidental or premature termination of a write operation.

The WREN bit, when set, will allow a write operation. On power-up, the WREN bit is clear. The WRERR bit is set when a write operation is interrupted by a MCLR reset or a WDT time-out reset during normal operation. In these situations, following reset, the user can check the WRERR bit and rewrite the location. The value of the data and address registers and the EEPGD bit remains unchanged.

Interrupt flag bit EEIF, in the PIR2 register, is set when write is complete. It must be cleared in software.

# PIC16F872

## REGISTER 4-1: EECON1 REGISTER (ADDRESS 18Ch)

R/W-x	U-0	U-0	U-0	R/W-x	R/W-0	R/S-0	R/S-0
EEPGD	—	—	—	WRERR	WREN	WR	RD

bit7 bit0

R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as '0'  
- n = Value at POR reset

bit 7: **EEPGD**: Program / Data EEPROM Select bit  
1 = Accesses Program memory  
0 = Accesses data memory  
(This bit cannot be changed while a read or write operation is in progress)

bit 6-4: **Unimplemented**: Read as '0'

bit 3: **WRERR**: EEPROM Error Flag bit  
1 = A write operation is prematurely terminated  
(any MCLR reset or any WDT reset during normal operation)  
0 = The write operation completed

bit 2: **WREN**: EEPROM Write Enable bit  
1 = Allows write cycles  
0 = Inhibits write to the EEPROM

bit 1: **WR**: Write Control bit  
1 = initiates a write cycle. (The bit is cleared by hardware once write is complete.) The WR bit can only be set (not cleared) in software.  
0 = Write cycle to the EEPROM is complete

bit 0: **RD**: Read Control bit  
1 = Initiates an EEPROM read RD is cleared in hardware. The RD bit can only be set (not cleared) in software.  
0 = Does not initiate an EEPROM read

## 4.3 Reading the Data EEPROM Memory

To read a data memory location, the user must write the address to the EEADR register, clear the EEPGD control bit (EECON1<7>) and then set control bit RD (EECON1<0>). The data is available in the very next instruction cycle of the EEDATA register, therefore it can be read by the next instruction. EEDATA will hold this value until another read operation or until it is written to by the user (during a write operation).

## 4.4 Writing to the Data EEPROM Memory

To write an EEPROM data location, the address must first be written to the EEADR register and the data written to the EEDATA register. Then the sequence in Example 4-2 must be followed to initiate the write cycle.

### EXAMPLE 4-1: DATA EEPROM READ

```
BSF STATUS, RP1 ;
BCF STATUS, RP0 ;Bank 2
MOVLW DATA_EE_ADDR ;
MOVWF EEADR ;Data Memory Address to read
BSF STATUS, RP0 ;Bank 3
BCF EECON1, EEPGD ;Point to DATA memory
BSF EECON1, RD ;EEPROM Read
BCF STATUS, RP0 ;Bank 2
MOVF EEDATA, W ;W = EEDATA
```

### EXAMPLE 4-2: DATA EEPROM WRITE

```
BSF STATUS, RP1 ;
BCF STATUS, RP0 ; Bank 2
MOVLW DATA_EE_ADDR ;
MOVWF EEADR ; Data Memory Address to write
MOVLW DATA_EE_DATA ;
MOVWF EEDATA ; Data Memory Value to write
BSF STATUS, RP0 ; Bank 3
BCF EECON1, EEPGD ; Point to DATA memory
BSF EECON1, WREN ; Enable writes
```

```
BCF INTCON, GIE ; Disable Interrupts
```

Required Sequence	MOVLW 55h ;
	MOVWF EECON2 ; Write 55h
	MOVLW AAh ;
	MOVWF EECON2 ; Write AAh
	BSF EECON1, WR ; Set WR bit to begin write
	BSF INTCON, GIE ; Enable Interrupts
	SLEEP ; Wait for interrupt to signal write complete
	BCF EECON1, WREN ; Disable writes

The write will not initiate if the above sequence is not exactly followed (write 55h to EECON2, write AAh to EECON2, then set WR bit) for each byte. It is strongly recommended that interrupts be disabled during this code segment.

Additionally, the WREN bit in EECON1 must be set to enable writes. This mechanism prevents accidental writes to data EEPROM due to unexpected code execution (i.e., runaway programs). The WREN bit should be kept clear at all times, except when updating the EEPROM. The WREN bit is not cleared by hardware

After a write sequence has been initiated, clearing the WREN bit will not affect the current write cycle. The WR bit will be inhibited from being set unless the WREN bit

is set. The WREN bit must be set on a previous instruction. Both WR and WREN cannot be set with the same instruction.

At the completion of the write cycle, the WR bit is cleared in hardware and the EEPROM Write Complete Interrupt Flag bit (EEIF) is set. EEIF must be cleared by software.

## 4.5 Reading the FLASH Program Memory

A program memory location may be read by writing two bytes of the address to the EEADR and EEADRH registers, setting the EEPGD control bit (EECON1<7>) and then setting control bit RD (EECON1<0>). Once the read control bit is set, the microcontroller will use the next two instruction cycles to read the data. The

data is available in the EEDATA and EEDATH registers after the second NOP instruction. Therefore, it can be read as two bytes in the following instructions. The EEDATA and EEDATH registers will hold this value until another read operation or until it is written to by the user (during a write operation).

### EXAMPLE 4-3: FLASH PROGRAM READ

```
BSF    STATUS, RP1    ;
BCF    STATUS, RP0    ; Bank 2
MOVLW  ADDRH         ;
MOVWF  EEADRH        ; MSByte of Program Address to read
MOVLW  ADDRL         ;
MOVWF  EEADR         ; LSByte of Program Address to read
BSF    STATUS, RP0    ; Bank 3
BSF    EECON1, EEPGD ; Point to PROGRAM memory
Required BSF    EECON1, RD    ; EEPROM Read
Sequence
NOP                                ; memory is read in the next two cycles after BSF EECON1,RD
NOP                                ;

BCF    STATUS, RP0    ; Bank 2

MOVF   EEDATA, W      ; W = LSByte of Program EEDATA
MOVF   EEDATH, W     ; W = MSByte of Program EEDATA
```



## 4.6 Writing to the FLASH Program Memory

When the PIC16F872 is fully code protected or not code protected, a word of the FLASH program memory may be written provided the WRT configuration bit is set. If the PIC16F872 is partially code protected, then a word of FLASH program memory may be written if the word is in a non-code protected segment of memory and the WRT configuration bit is set. To write a FLASH program location, the first two bytes of the address must be written to the EEADR and EEADRH registers and two bytes of the data to the EEDATA and EEDATH registers, set the EEPGD control bit (EECON1<7>),

and then set control bit WR (EECON1<1>). The sequence in Example 4-4 must be followed to initiate a write to program memory.

The microcontroller will then halt internal operations during the next two instruction cycles for the TPEW (parameter D133) in which the write takes place. This is not SLEEP mode, as the clocks and peripherals will continue to run. Therefore, the two instructions following the “BSF EECON, WR” should be NOP instructions. After the write cycle, the microcontroller will resume operation with the 3rd instruction after the EECON1 write instruction.

### EXAMPLE 4-4: FLASH PROGRAM WRITE

```

BSF    STATUS, RP1    ;
BCF    STATUS, RP0    ; Bank 2
MOVLW  ADDRH          ;
MOVWF  EEADRH         ; MSByte of Program Address to read
MOVLW  ADDRRL         ;
MOVWF  EEADR          ; LSByte of Program Address to read
MOVLW  DATAH         ;
MOVWF  EEDATH         ; MS Program Memory Value to write
MOVLW  DATAL          ;
MOVWF  EEDATA         ; LS Program Memory Value to write
BSF    STATUS, RP0    ; Bank 3
BSF    EECON1, EEPGD  ; Point to PROGRAM memory
BSF    EECON1, WREN   ; Enable writes

BCF    INTCON, GIE    ; Disable Interrupts
MOVLW  55h            ;
Required MOVWF  EECON2        ; Write 55h
Sequence MOVLW  AAh            ;
MOVWF  EECON2        ; Write AAh
BSF    EECON1, WR     ; Set WR bit to begin write

NOP                                ; Instructions here are ignored by the microcontroller
NOP                                ; Microcontroller will halt operation and wait for
                                ; a write complete. After the write
                                ; the microcontroller continues with 3rd instruction

BSF    INTCON, GIE    ; Enable Interrupts
BCF    EECON1, WREN   ; Disable writes

```

## 4.7 Write Verify

Depending on the application, good programming practice may dictate that the value written to the memory should be verified against the original value. This should be used in applications where excessive writes can stress bits near the specification limit.

Generally a write failure will be a bit which was written as a '1', but reads back as a '0' (due to leakage off the bit).

## 4.8 Protection Against Spurious Write

### 4.8.1 EEPROM DATA MEMORY

There are conditions when the device may not want to write to the data EEPROM memory. To protect against spurious EEPROM writes, various mechanisms have been built-in. On power-up, the WREN bit is cleared. Also, the Power-up Timer (72 ms duration) prevents EEPROM write.

The write initiate sequence and the WREN bit together help prevent an accidental write during brown-out, power glitch, or software malfunction.

### 4.8.2 PROGRAM FLASH MEMORY

To protect against spurious writes to FLASH program memory, the WRT bit in the configuration word may be programmed to '0' to prevent writes. The write initiate sequence must also be followed. WRT and the configuration word cannot be programmed by user code, only through the use of an external programmer.

## 4.9 Operation during Code Protect

Each reprogrammable memory block has its own code protect mechanism. External Read and Write operations are disabled if either of these mechanisms are enabled.

### 4.9.1 DATA EEPROM MEMORY

The microcontroller itself can both read and write to the internal Data EEPROM, regardless of the state of the code protect configuration bit.

### 4.9.2 PROGRAM FLASH MEMORY

The microcontroller can read and execute instructions out of the internal FLASH program memory, regardless of the state of the code protect configuration bits. However, the WRT configuration bit and the code protect bits have different effects on writing to program memory. Table 4-1 shows the various configurations and status of reads and writes. To erase the WRT or code protection bits in the configuration word requires that the device be fully erased.

**Note:** The PIC16F872 devices can perform self writes to any location in program memory when not code protected or fully code protected.

**TABLE 4-1: READ/WRITE STATE OF INTERNAL FLASH PROGRAM MEMORY**

Configuration Bits			Memory Location	Internal Read	Internal Write	ICSP Read	ICSP Write
CP1	CP0	WRT					
0	0	1	All program memory	Yes	Yes	No	No
0	0	0	All program memory	Yes	No	No	No
0	1	0	Unprotected areas	Yes	No	Yes	No
0	1	0	Protected areas	Yes	No	No	No
0	1	1	Unprotected areas	Yes	Yes	Yes	No
0	1	1	Protected areas	Yes	No	No	No
1	0	0	Unprotected areas	Yes	No	Yes	No
1	0	0	Protected areas	Yes	No	No	No
1	0	1	Unprotected areas	Yes	Yes	Yes	No
1	0	1	Protected areas	Yes	No	No	No
1	1	0	All program memory	Yes	No	Yes	Yes
1	1	1	All program memory	Yes	Yes	Yes	Yes

**TABLE 4-2: REGISTERS ASSOCIATED WITH DATA EEPROM/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
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
10Dh	EEADR	EEPROM address register								xxxx xxxx	uuuu uuuu
10Fh	EEADRH	—	—	—	—	—	EEPROM address high			xxxx xxxx	uuuu uuuu
10Ch	EEDATA	EEPROM data register								xxxx xxxx	uuuu uuuu
10Eh	EEDATH	—	—	EEPROM data register high						xxxx xxxx	uuuu uuuu
18Ch	EECON1	EEPGD	—	—	—	WRERR	WREN	WR	RD	x--- x000	x--- u000
18Dh	EECON2	EEPROM control register2 (not a physical register)									
8Dh	PIE2	—	(1)	—	EEIE	BCLIE	—	—	(1)	-r-0 0--r	-r-0 0--r
0Dh	PIR2	—	(1)	—	EEIF	BCLIF	—	—	(1)	-r-0 0--r	-r-0 0--r

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

**Note 1:** These bits are reserved; always maintain these bits clear.

# PIC16F872

---

NOTES:

## 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).

Timer mode is selected by clearing bit T0CS (OPTION\_REG<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.

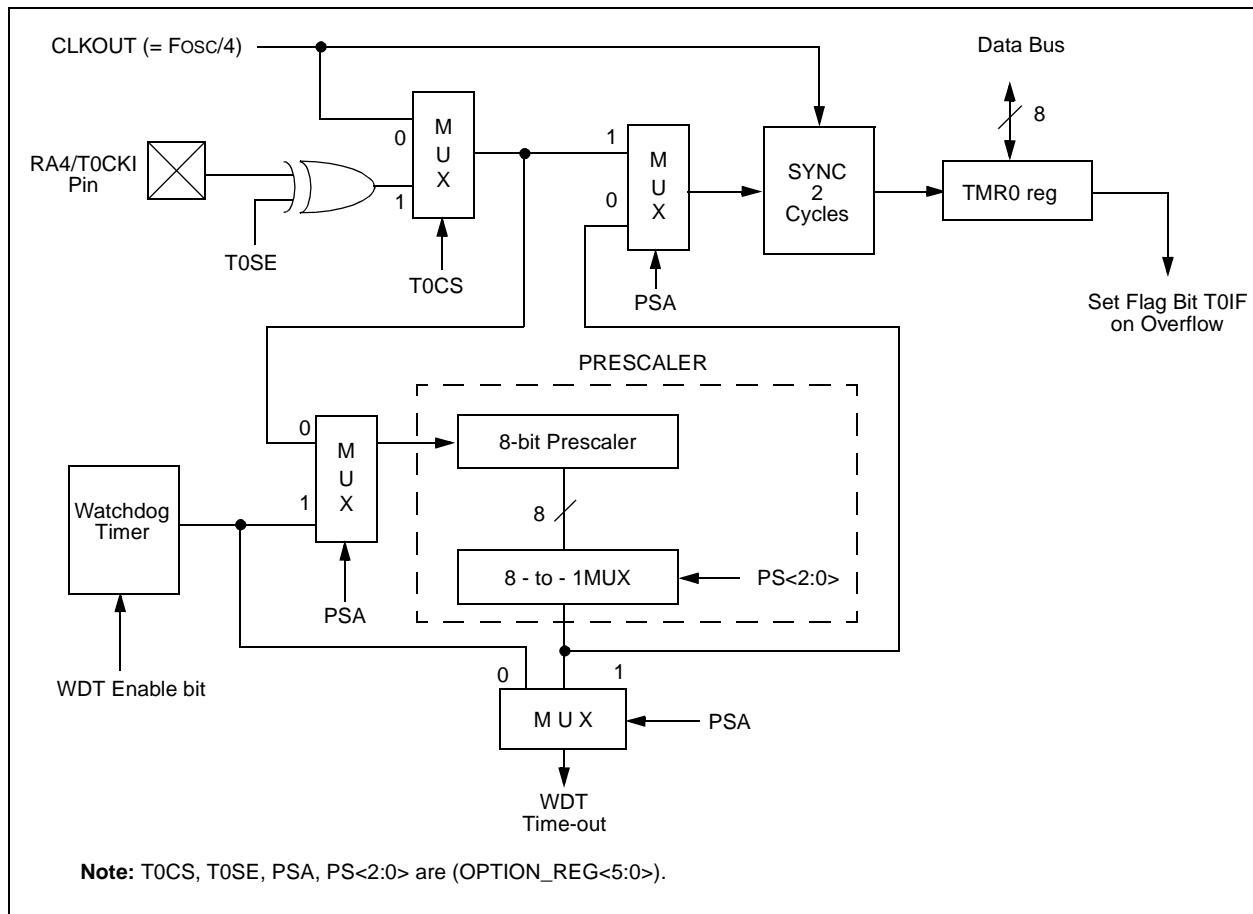
Counter mode is selected by setting bit T0CS (OPTION\_REG<5>). In counter mode, Timer0 will increment either on every rising or falling edge of pin RA4/T0CKI. The incrementing edge is determined by the Timer0 Source Edge Select bit T0SE (OPTION\_REG<4>). Clearing bit T0SE selects the rising edge. Restrictions on the external clock input are discussed in detail in Section 5.2.

The prescaler is mutually exclusively shared between the Timer0 module and the watchdog timer. The prescaler is not readable or writable. Section 5.3 details the operation of the prescaler.

### 5.1 Timer0 Interrupt

The TMR0 interrupt is generated when the TMR0 register overflows from FFh to 00h. This overflow sets bit T0IF (INTCON<2>). The interrupt can be masked by clearing bit T0IE (INTCON<5>). Bit T0IF 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**



# PIC16F872

## 5.2 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 2Tosc (and a small RC delay of 20 ns) and low for at least 2Tosc (and a small RC delay of 20 ns). Refer to the electrical specification of the desired device.

## 5.3 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 PS<2:0> bits (OPTION\_REG<3:0>) determine the prescaler assignment and prescale ratio.

When assigned to the Timer0 module, all instructions writing to the TMR0 register (i.e., 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.

**REGISTER 5-1: OPTION\_REG REGISTER**

	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7								bit 0
bit 7:	<b>RBPU</b>							
bit 6:	<b>INTEDG</b>							
bit 5:	<b>T0CS:</b> TMR0 Clock Source Select bit							
	1 = Transition on T0CKI pin							
	0 = Internal instruction cycle clock (CLKOUT)							
bit 4:	<b>T0SE:</b> TMR0 Source Edge Select bit							
	1 = Increment on high-to-low transition on T0CKI pin							
	0 = Increment on low-to-high transition on T0CKI pin							
bit 3:	<b>PSA:</b> Prescaler Assignment bit							
	1 = Prescaler is assigned to the WDT							
	0 = Prescaler is assigned to the Timer0 module							
bit 2-0:	<b>PS&lt;2:0&gt;:</b> 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					

R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as '0'  
-n = Value at POR reset

**Note:** To avoid an unintended device RESET, the instruction sequence shown in the PICmicro™ Mid-Range MCU Family Reference Manual (DS33023) must be executed when changing the prescaler assignment from Timer0 to the WDT. This sequence must be followed even if the WDT is disabled.

**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's register								xxxx xxxx	uuuu uuuu
0Bh,8Bh, 10Bh,18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
81h,181h	OPTION_REG	RBP <u>U</u>	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.

# PIC16F872

---

NOTES:



## 6.0 TIMER1 MODULE

The Timer1 module is a 16-bit timer/counter consisting of two 8-bit registers (TMR1H and TMR1L), which are readable and writable. 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>).

Timer1 can operate in one of two modes:

- As a timer
- As a 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.

Timer1 can be enabled/disabled by setting/clearing control bit TMR1ON (T1CON<0>).

Timer1 also has an internal "reset input". This reset can be generated by the CCP module (Section 8.0). Register 6-1 shows the Timer1 control register.

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.

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

### 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	T1SYNC	TMR1CS	TMR1ON	
bit7								bit0

R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as '0'  
- n = Value at POR reset

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

bit 5-4: **T1CKPS<1:0>**: 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

# PIC16F872

## 6.1 Timer1 Operation in Timer Mode

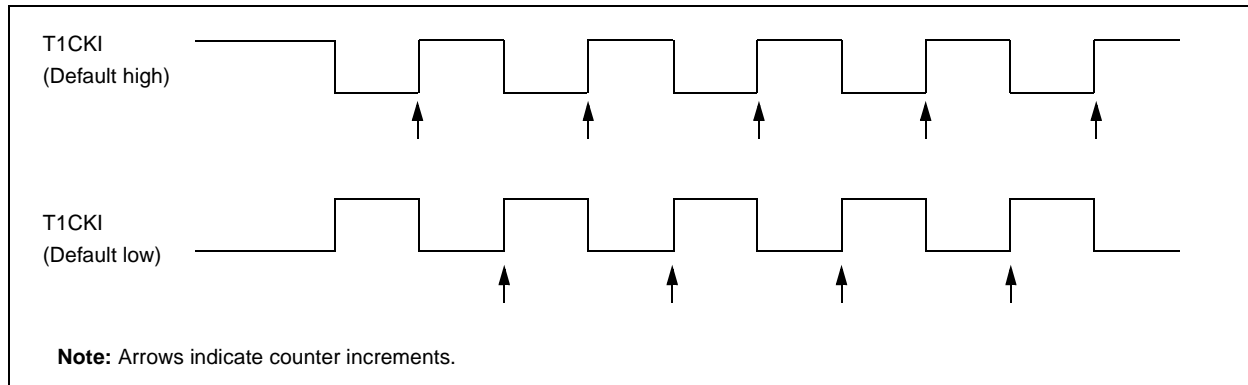
Timer mode is selected by clearing the TMR1CS (T1CON<1>) bit. In this mode, the input clock to the timer is FOSC/4. The synchronize control bit  $\overline{T1SYNC}$  (T1CON<2>) has no effect since the internal clock is always in sync.

## 6.2 Timer1 Counter Operation

Timer1 may operate in asynchronous or usynchronous 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.

**FIGURE 6-1: TIMER1 INCREMENTING EDGE**



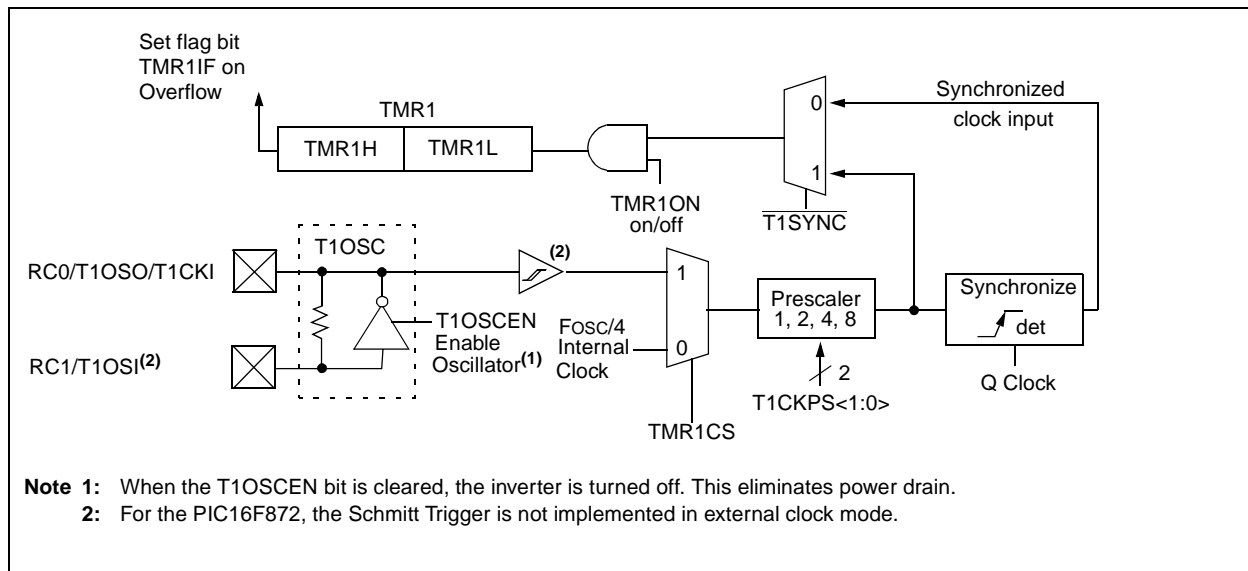
## 6.3 Timer1 Operation in Synchronized Counter Mode

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 6-2: TIMER1 BLOCK DIAGRAM**



## 6.4 Timer1 Operation in Asynchronous Counter Mode

If control bit  $\overline{T1SYN\bar{C}}$  (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, which will wake-up the processor. However, special precautions in software are needed to read/write the timer (Section 6.4.1).

In asynchronous counter mode, Timer1 can not be used as a time-base for capture or compare operations.

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

Reading TMR1H or TMR1L while the timer is running from an external asynchronous clock will guarantee 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.

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.

## 6.5 Timer1 Oscillator

A crystal oscillator circuit is built-in 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 use with 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
These values are for design guidance only.			
<b>Crystals Tested:</b>			
32.768 kHz	Epson C-001R32.768K-A	± 20 PPM	
100 kHz	Epson C-2 100.00 KC-P	± 20 PPM	
200 kHz	STD XTL 200.000 kHz	± 20 PPM	
<b>Note 1:</b> Higher capacitance increases the stability of oscillator, but also increases the start-up time. <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.			

## 6.6 Resetting Timer1 using CCP1 Trigger Output

If the CCP1 module is configured in compare mode to generate a "special event trigger" (CCP1M<3:0> = 1011), this signal will reset Timer1.

**Note:** The special event trigger 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 register pair effectively becomes the period register for Timer1.

## 6.7 Resetting of 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 trigger.

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.

## 6.8 Timer1 Prescaler

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

# PIC16F872

**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	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	(1)	ADIF	(1)	(1)	SSPIF	CCP1IF	TMR2IF	TMR1IF	r0xr 0000	r0xr 0000
8Ch	PIE1	(1)	ADIE	(1)	(1)	SSPIE	CCP1IE	TMR2IE	TMR1IE	r0xr 0000	r0xr 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	$\overline{T1SYNC}$	TMR1CS	TMR1ON	--00 0000	--uu uuuu

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

**Note 1:** These bits are reserved; always maintain these bits clear.

## 7.0 TIMER2 MODULE

Timer2 is an 8-bit timer with a prescaler and a postscaler. It can be used as the PWM time-base for the 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 T2CKPS<1:0> (T2CON<1:0>).

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.

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>)).

Timer2 can be shut off by clearing control bit TMR2ON (T2CON<2>) to minimize power consumption.

Register 7-1 shows the Timer2 control register.

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

## 7.1 Timer2 Prescaler and Postscaler

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

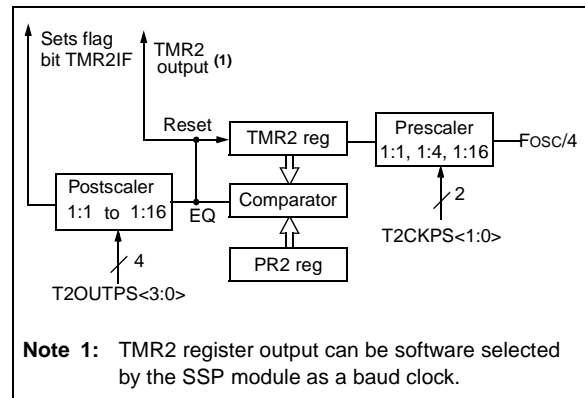
- a write to the TMR2 register
- a write to the T2CON register
- any device reset (POR, MCLR reset, WDT reset or BOR)

TMR2 is not cleared when T2CON is written.

## 7.2 Output of TMR2

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

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



**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
bit7							bit0

R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as '0'  
- n = Value at POR reset

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

bit 6-3: **TOUTPS<3:0>**: 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: **T2CKPS<1:0>**: Timer2 Clock Prescale Select bits  
00 = Prescaler is 1  
01 = Prescaler is 4  
1x = Prescaler is 16

# PIC16F872

**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	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	(1)	ADIF	(1)	(1)	SSPIF	CCP1IF	TMR2IF	TMR1IF	r0rr 0000	r0rr 0000
8Ch	PIE1	(1)	ADIE	(1)	(1)	SSPIE	CCP1IE	TMR2IE	TMR1IE	r0rr 0000	r0rr 0000
11h	TMR2	Timer2 module's 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, r = reserved, - = unimplemented read as '0'. Shaded cells are not used by the Timer2 module.

**Note 1:** These bits are reserved; always maintain these bits clear.

## 8.0 CAPTURE/COMPARE/PWM MODULE

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

- 16-bit Capture register
- 16-bit Compare register
- PWM master/slave Duty Cycle register

Table 8-1 shows the resources used by the CCP module. In the following sections, the operation of a CCP module is described.

### CCP1 Module:

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. The special event trigger is

generated by a compare match and will reset Timer1 and start an A/D conversion (if the A/D module is enabled).

Additional information on CCP modules is available in the PICmicro™ Mid-Range MCU Family Reference Manual (DS33023) and in Application Note 594, "Using the CCP Modules" (DS00594).

**TABLE 8-1: CCP MODE - TIMER RESOURCES REQUIRED**

CCP Mode	Timer Resource
Capture	Timer1
Compare	Timer1
PWM	Timer2

### REGISTER 8-1: CCP1CON REGISTER (ADDRESS: 17h)

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	
bit7								bit0

R = Readable bit  
 W = Writable bit  
 U = Unimplemented bit, read as '0'  
 - n = Value at POR reset

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

bit 5-4: **CCP1<X:Y>:** 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 CCPR1L.

bit 3-0: **CCP1M<3:0>:** CCPx Mode Select bits  
 0000 = Capture/Compare/PWM off (resets CCP 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 (CCP1IF bit is set)  
 1001 = Compare mode, clear output on match (CCP1IF bit is set)  
 1010 = Compare mode, generate software interrupt on match (CCP1IF bit is set, CCP pin is unaffected)  
 1011 = Compare mode, trigger special event (CCP1IF bit is set, CCP1 pin is unaffected); CCP1 resets TMR1 and starts an A/D conversion (if A/D module is enabled)  
 11xx = PWM mode

# PIC16F872

## 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

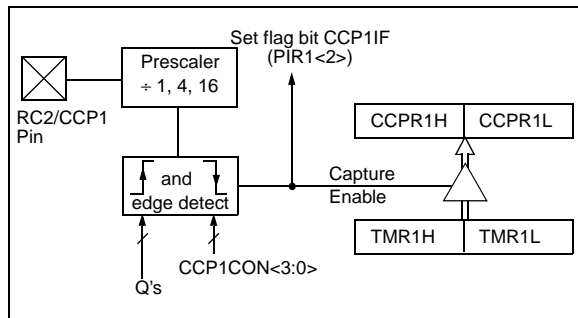
An event is selected by control bits CCP1M<3:0> (CCP1CON<3:0>). When a capture is made, the interrupt request flag bit CCP1IF (PIR1<2>) is set. The interrupt flag must be cleared in software. If another capture occurs before the value in register CCPR1 is read, the old captured value will be lost.

### 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 pin 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 CCP1M<3:0>. Whenever the CCP module is turned off, or the CCP module is not in capture mode, the prescaler counter is cleared. 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
MOVWF  CCP1CON      ; move value and CCP ON

MOVWF  CCP1CON      ;Load CCP1CON with this
                        ; value
```



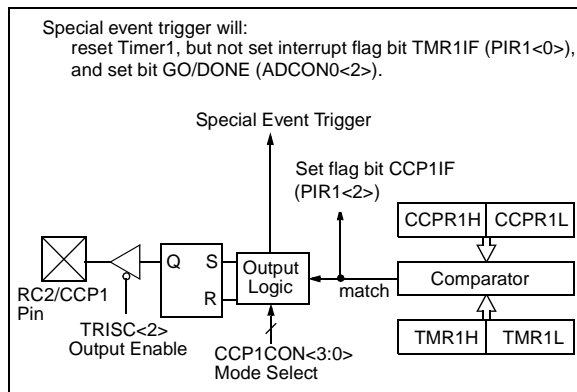
## 8.2 Compare Mode

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 CCP1M<3:0> (CCP1CON<3:0>). At the same time, interrupt flag bit CCP1IF is set.

**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 mode is chosen, the CCP1 pin is not affected. The CCP1IF bit is set causing a CCP interrupt (if enabled).

### 8.2.4 SPECIAL EVENT TRIGGER

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

The special event trigger output of CCP1 resets the TMR1 register pair and starts an A/D conversion (if the A/D module is enabled). This allows the CCPR1 register to effectively be a 16-bit programmable period register for Timer1.

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

## 8.3 PWM Mode (PWM)

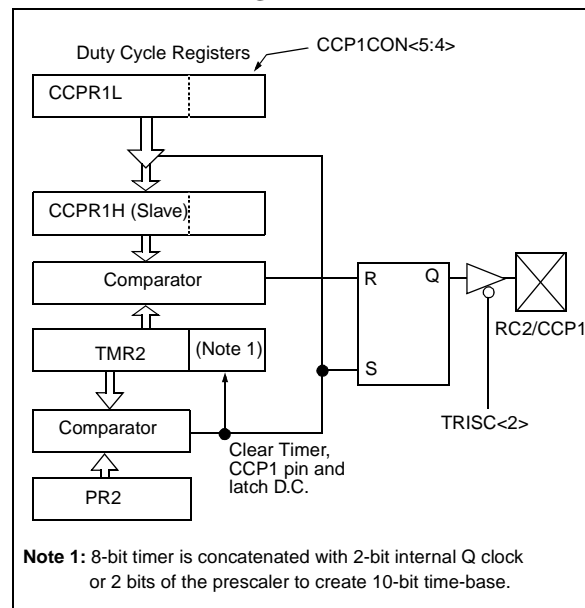
In pulse width modulation mode, the CCP1 pin produces up to a 10-bit resolution PWM output. Since the CCP1 pin is multiplexed with the PORTC data latch, the TRISC<2> bit must be cleared to make the CCP1 pin an output.

**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.

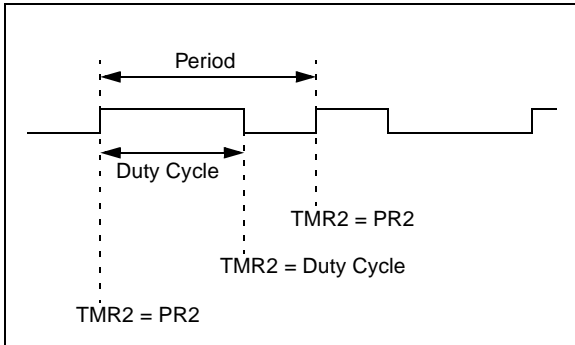
For a step-by-step procedure on how to set up the CCP module for PWM operation, see Section 8.3.3.

**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 following formula:

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

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 8.1) 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>. The following equation is used to calculate the PWM duty cycle in time:

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

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.

Maximum PWM resolution (bits) for a given PWM frequency:

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

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

### 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 8-2: 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	TOIE	INTE	RBIE	TOIF	INTF	RBFIF	0000 000x	0000 000u
0Ch	PIR1	(1)	ADIF	(1)	(1)	SSPIF	CCP1IF	TMR2IF	TMR1IF	r0rr 0000	r0rr 0000
8Ch	PIE1	(1)	ADIE	(1)	(1)	SSPIE	CCP1IE	TMR2IE	TMR1IE	r0rr 0000	r0rr 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, r = reserved, - = unimplemented read as '0'. Shaded cells are not used by Capture and Timer1.

**Note 1:** These bits are reserved; always maintain these bits clear.

**TABLE 8-3: 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	TOIE	INTE	RBIE	TOIF	INTF	RBFIF	0000 000x	0000 000u
0Ch	PIR1	(1)	ADIF	(1)	(1)	SSPIF	CCP1IF	TMR2IF	TMR1IF	r0rr 0000	r0rr 0000
8Ch	PIE1	(1)	ADIE	(1)	(1)	SSPIE	CCP1IE	TMR2IE	TMR1IE	r0rr 0000	r0rr 0000
87h	TRISC	PORTC Data Direction Register								1111 1111	1111 1111
11h	TMR2	Timer2 module's register								0000 0000	0000 0000
92h	PR2	Timer2 module's 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, r = reserved, - = unimplemented read as '0'. Shaded cells are not used by PWM and Timer2.

**Note 1:** These bits are reserved; always maintain these bits clear.

# PIC16F872

---

NOTES:

## 9.0 MASTER SYNCHRONOUS SERIAL PORT (MSSP) MODULE

The Master Synchronous Serial Port (MSSP) 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 MSSP module can operate in one of two modes:

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

Figure 9-1 shows a block diagram for the SPI mode, while Figure 9-5 and Figure 9-9 show the block diagrams for the two different I<sup>2</sup>C modes of operation.

# PIC16F872

## REGISTER 9-1: SSPSTAT: SYNC 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	R/ $\bar{W}$	UA	BF
bit7							bit0

R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as '0'  
-n = Value at POR reset

- bit 7: **SMP**: Sample bit  
SPI Master Mode  
1 = Input data sampled at end of data output time  
0 = Input data sampled at middle of data output time  
SPI Slave Mode  
SMP must be cleared when SPI is used in slave mode  
In I<sup>2</sup>C master or slave mode:  
1 = Slew rate control disabled for standard speed mode (100 kHz and 1 MHz)  
0 = Slew rate control enabled for high speed mode (400 kHz)
- bit 6: **CKE**: SPI Clock Edge Select (Figure 9-4, Figure 9-5 and Figure 9-6)  
SPI Mode:  
CKP = 0  
1 = Transmit happens on transition from active clock state to idle clock state  
0 = Transmit happens on transition from idle clock state to active clock state  
CKP = 1  
1 = Data transmitted on falling edge of SCK  
0 = Data transmitted on rising edge of SCK  
In I<sup>2</sup>C Master or Slave Mode:  
1 = Input levels conform to SMBUS spec  
0 = Input levels conform to I<sup>2</sup>C specs
- bit 5: **D/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 MSSP module is disabled, 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 MSSP module is disabled, 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: **R/W**: Read/Write bit information (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 not  $\bar{A}CK$  bit.  
In I<sup>2</sup>C slave mode:  
1 = Read  
0 = Write  
In I<sup>2</sup>C master mode:  
1 = Transmit is in progress  
0 = Transmit is not in progress.  
Or'ing this bit with SEN, RSEN, PEN, RCEN or ACKEN will indicate if the MSSP is in IDLE mode.
- bit 1: **UA**: Update Address (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 = Data Transmit in progress (does not include the  $\bar{A}CK$  and stop bits), SSPBUF is full  
0 = Data Transmit complete (does not include the  $\bar{A}CK$  and stop bits), SSPBUF is empty

## 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
bit7							bit0

R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as '0'  
- n= Value at POR reset

bit 7: **WCOL**: Write Collision Detect bit  
Master Mode:  
1 = A write to SSPBUF was attempted while the I<sup>2</sup>C conditions were not valid  
0 = No collision  
Slave Mode:  
1 = SSPBUF register is written while 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 SSPBUF holds previous data. Data in SSPSR is lost on overflow. In slave mode, the user must read the SSPBUF, even if only transmitting data to avoid overflows. In master mode, the overflow bit is not set since each operation is initiated by writing to the SSPBUF register. (Must be cleared in software.)  
0 = No overflow  
In I<sup>2</sup>C mode  
1 = A byte is received while the SSPBUF is holding the previous byte. SSPOV is a "don't care" in transmit mode. (Must be cleared in software.)  
0 = No overflow

bit 5: **SSPEN**: Synchronous Serial Port Enable bit  
In SPI mode, when enabled, these pins must be properly configured as input or output.  
1 = Enables serial port and configures SCK, SDO, SDI, and  $\overline{SS}$  as the source of the serial port pins  
0 = Disables serial port and configures these pins as I/O port pins  
In I<sup>2</sup>C mode, when enabled, these pins must be properly configured as input or output.  
1 = Enables the serial port and configures the SDA and SCL pins as the source of the serial port pins  
0 = Disables serial port and configures these pins as I/O port pins

bit 4: **CKP**: Clock Polarity Select bit  
In SPI mode  
1 = Idle state for clock is a high level  
0 = Idle state for clock is a low level  
In I<sup>2</sup>C slave mode, SCK release control  
1 = Enable clock  
0 = Holds clock low (clock stretch) (Used to ensure data setup time)  
In I<sup>2</sup>C master mode  
Unused in this mode

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  
1000 = I<sup>2</sup>C master mode, clock = Fosc / (4 \* (SSPADD+1))  
1011 = I<sup>2</sup>C firmware controlled master mode (slave idle)  
1110 = I<sup>2</sup>C firmware controlled master mode, 7-bit address with start and stop bit interrupts enabled.  
1111 = I<sup>2</sup>C firmware controlled master mode, 10-bit address with start and stop bit interrupts enabled.  
1001, 1010, 1100, 1101 = reserved

# PIC16F872

## REGISTER 9-3: SSPCON2: SYNC SERIAL PORT CONTROL REGISTER2 (ADDRESS 91h)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	
bit7								bit0

R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as '0'  
- n= Value at POR reset

- bit 7: **GCEN**: General Call Enable bit (In I<sup>2</sup>C slave mode only)  
1 = Enable interrupt when a general call address (0000h) is received in the SSPSR.  
0 = General call address disabled.
- bit 6: **ACKSTAT**: Acknowledge Status bit (In I<sup>2</sup>C master mode only)  
In master transmit mode:  
1 = Acknowledge was not received from slave.  
0 = Acknowledge was received from slave.
- bit 5: **ACKDT**: Acknowledge Data bit (In I<sup>2</sup>C master mode only)  
In master receive mode:  
Value that will be transmitted when the user initiates an Acknowledge sequence at the end of a receive.  
1 = Not Acknowledge.  
0 = Acknowledge.
- bit 4: **ACKEN**: Acknowledge Sequence Enable bit (In I<sup>2</sup>C master mode only).  
In master receive mode:  
1 = Initiate Acknowledge sequence on SDA and SCL pins, and transmit ACKDT data bit. Automatically cleared by hardware.  
0 = Acknowledge sequence idle.
- bit 3: **RCEN**: Receive Enable bit (In I<sup>2</sup>C master mode only).  
1 = Enables Receive mode for I<sup>2</sup>C.  
0 = Receive idle.
- bit 2: **PEN**: Stop Condition Enable bit (In I<sup>2</sup>C master mode only).  
SCK release control  
1 = Initiate Stop condition on SDA and SCL pins. Automatically cleared by hardware.  
0 = Stop condition idle.
- bit 1: **RSEN**: Repeated Start Condition Enabled bit (In I<sup>2</sup>C master mode only)  
1 = Initiate Repeated Start condition on SDA and SCL pins. Automatically cleared by hardware.  
0 = Repeated Start condition idle.
- bit 0: **SEN**: Start Condition Enabled bit (In I<sup>2</sup>C master mode only)  
1 = Initiate Start condition on SDA and SCL pins. Automatically cleared by hardware.  
0 = Start condition idle.

**Note:** For bits ACKEN, RCEN, PEN, RSEN, SEN: If the I<sup>2</sup>C module is not in the idle mode, this bit may not be set (no spooling), and the SSPBUF may not be written (or writes to the SSPBUF are disabled).



## 9.1 SPI Mode

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

- Serial Data Out (SDO)
- Serial Data In (SDI)
- Serial Clock (SCK)

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

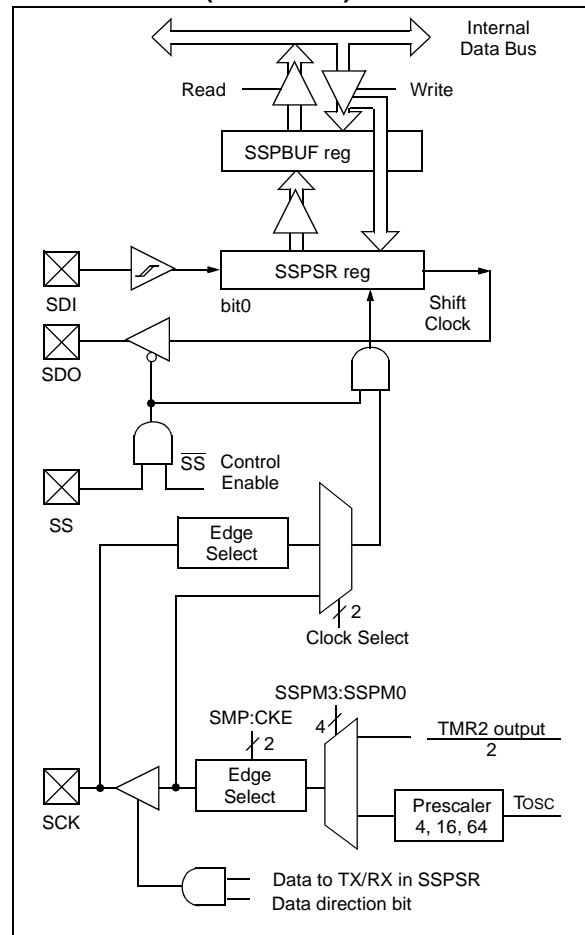
- Slave Select ( $\overline{SS}$ )

When initializing the SPI, several options need to be specified. This is done by programming the appropriate control bits (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)
- Data input sample phase (middle or end of data output time)
- Clock edge (output data on rising/falling edge of SCK)
- Clock Rate (Master mode only)
- Slave Select Mode (Slave mode only)

Figure 9-4 shows the block diagram of the MSSP module when in SPI mode.

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



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

- SDI is automatically controlled by the SPI module
- 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

Any serial port function that is not desired may be overridden by programming the corresponding data direction (TRIS) register to the opposite value.

## 9.1.1 MASTER MODE

The master can initiate the data transfer at any time because it controls the SCK. The master determines when the slave (Processor 2, Figure 9-5) is to broadcast data by the software protocol.

In Master mode, the data is transmitted/received as soon as the SSPBUF register is written to. If the SPI module is only going to receive, the SDO output could be disabled (programmed as an input). The SSPSR register will continue to shift in the signal present on the SDI pin at the programmed clock rate. As each byte is received, it will be loaded into the SSPBUF register as if a normal received byte (interrupts and status bits appropriately set). This could be useful in receiver applications as a "line activity monitor".

The clock polarity is selected by appropriately programming bit CKP (SSPCON<4>). This then would give waveforms for SPI communication as shown in

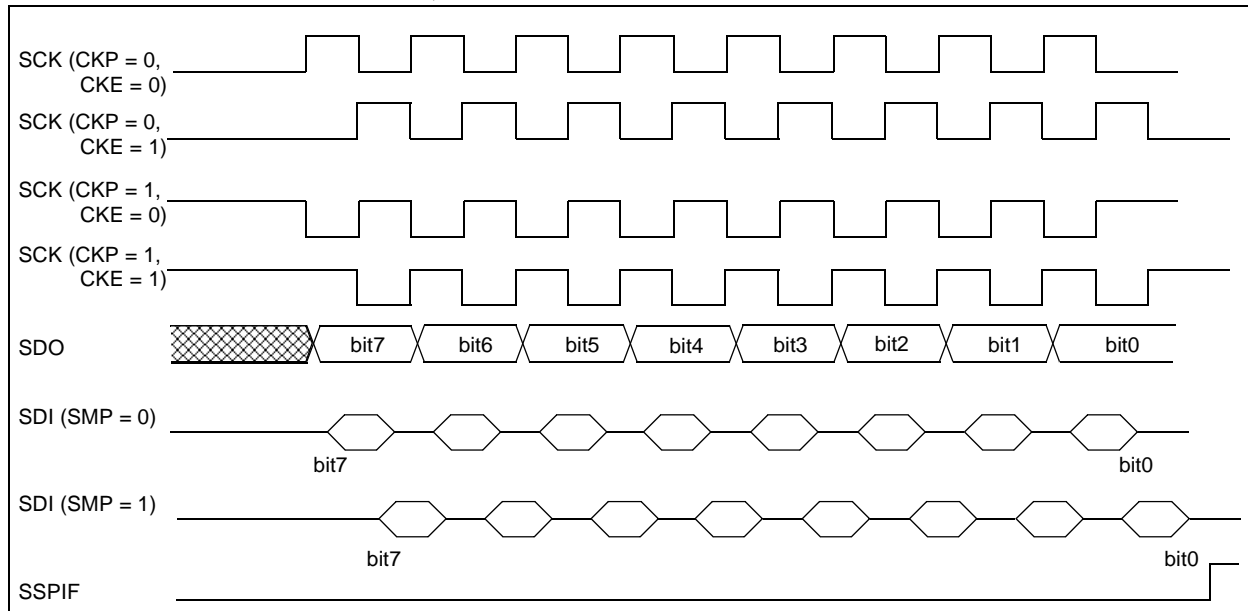
Figure 9-6, Figure 9-8 and Figure 9-9 where the MSb is transmitted first. In Master mode, the SPI clock rate (bit rate) is user programmable to be one of the following:

- $F_{OSC}/4$  (or  $T_{CY}$ )
- $F_{OSC}/16$  (or  $4 \cdot T_{CY}$ )
- $F_{OSC}/64$  (or  $16 \cdot T_{CY}$ )
- Timer2 output/2

This allows a maximum bit clock frequency (at 20 MHz) of 5.0 MHz.

Figure 9-6 shows the waveforms for Master mode. When  $CKE = 1$ , the SDO data is valid before there is a clock edge on SCK. The change of the input sample is shown based on the state of the SMP bit. The time when the SSPBUF is loaded with the received data is shown.

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



## 9.1.2 SLAVE MODE

In Slave mode, the data is transmitted and received as the external clock pulses appear on SCK. When the last bit is latched, the interrupt flag bit SSPIF (PIR1<3>) is set.

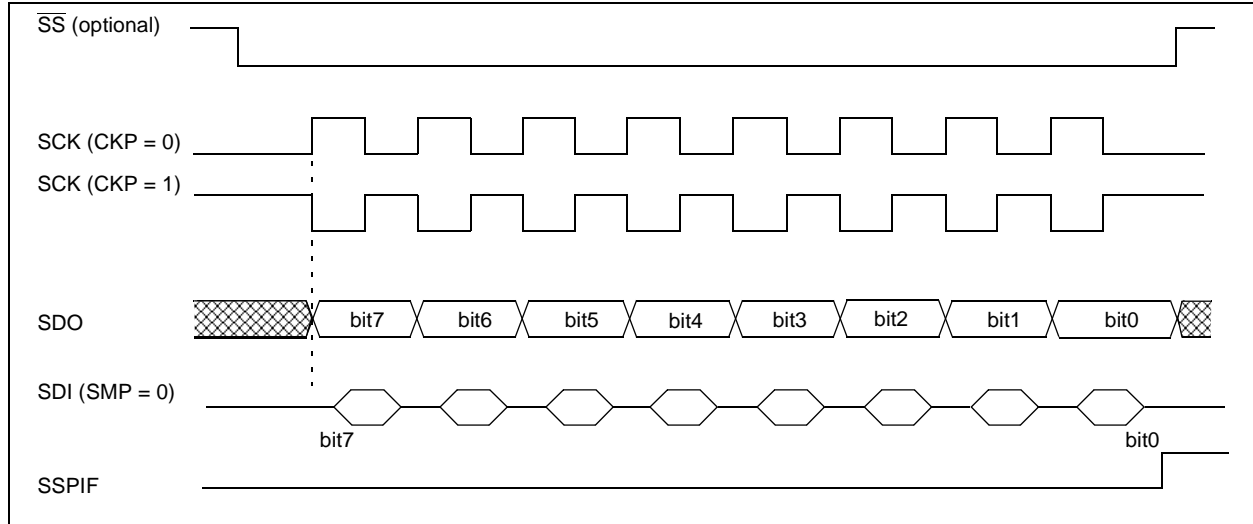
While in Slave mode, the external clock is supplied by the external clock source on the SCK pin. This external clock must meet the minimum high and low times as specified in the electrical specifications.

While in SLEEP mode, the slave can transmit/receive data. When a byte is received, the device will wake-up from sleep.

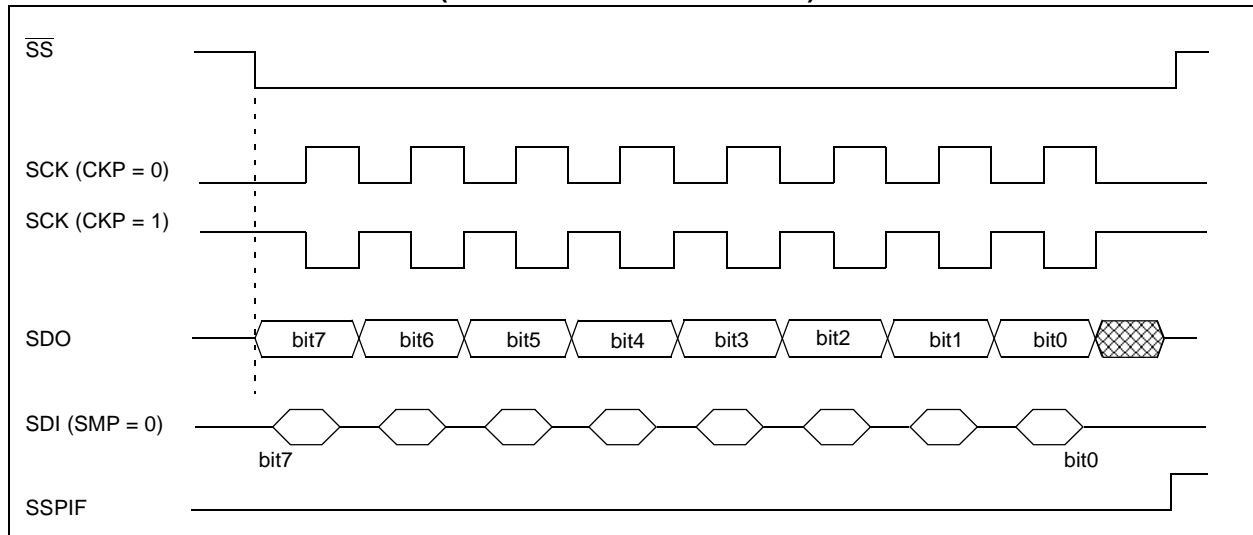
**Note:** When the SPI module is in Slave mode with  $\overline{SS}$  pin control enabled, (SSP-CON<3:0> = 0100) the SPI module will reset if the  $\overline{SS}$  pin is set to VDD.

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

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



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



# PIC16F872

**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	POR, BOR	MCLR, WDT
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
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
94h	SSPSTAT	SMP	CKE	D/ $\bar{A}$	P	S	R/ $\bar{W}$	UA	BF	0000 0000	0000 0000

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

**Note 1:** These bits are reserved on the 28-pin devices; always maintain these bits clear.

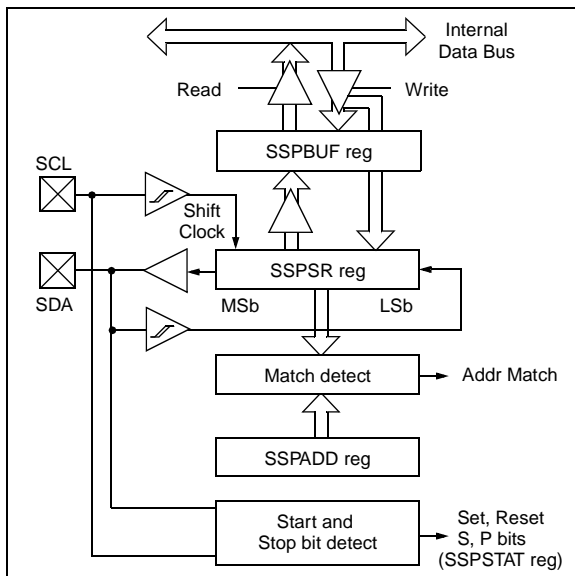
## 9.2 MSSP I<sup>2</sup>C Operation

The MSSP module in I<sup>2</sup>C mode fully implements all master and slave functions (including general call support) and provides interrupts-on-start and stop bits in hardware to determine a free bus (multi-master function). The MSSP module implements the standard mode specifications, as well as 7-bit and 10-bit addressing.

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

A "glitch" filter is on the SCL and SDA pins when the pin is an input. This filter operates in both the 100 kHz and 400 kHz modes. In the 100 kHz mode, when these pins are an output, there is a slow rate control of the pin that is independent of device frequency.

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



Two pins are used for data transfer. These are the SCL pin, which is the clock, and the SDA pin, which is the data. The SDA and SCL pins are automatically configured when the I<sup>2</sup>C mode is enabled. The SSP module functions are enabled by setting SSP Enable bit SSPEN (SSPCON<5>).

The MSSP module has six registers for I<sup>2</sup>C operation. They are the:

- SSP Control Register (SSPCON)
- SSP Control Register2 (SSPCON2)
- 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 Master mode, clock = OSC/4 (SSPADD +1)

Before selecting any I<sup>2</sup>C mode, the SCL and SDA pins must be programmed to inputs by setting the appropriate TRIS bits. Selecting an I<sup>2</sup>C mode, by setting the SSPEN bit, enables the SCL and SDA pins to be used as the clock and data lines in I<sup>2</sup>C mode.

The CKE bit (SSPSTAT<6:7>) sets the levels of the SDA and SCL pins in either Master or Slave mode. When CKE = 1, the levels will conform to the SMBUS specification. When CKE = 0, the levels will conform to the I<sup>2</sup>C specification.

The SSPSTAT register gives the status of the data transfer. This information includes detection of a START (S) or STOP (P) bit, specifies if the received byte was data or address, if the next byte is the completion of 10-bit address, and if this will be a read or write data transfer.

SSPBUF is the register to which the transfer data is written to or read from. The SSPSR register shifts the data in or out of the device. In receive operations, the SSPBUF and SSPSR create a doubled buffered receiver. This allows reception of the next byte to begin before reading the last byte of received data. When the complete byte is received, it is transferred to the SSPBUF register and flag bit SSPIF is set. If another complete byte is received before the SSPBUF register is read, a receiver overflow has occurred and bit SSPOV (SSPCON<6>) is set and the byte in the SSPSR is lost.

The SSPADD register holds the slave address. In 10-bit mode, the user needs to write the high byte of the address (1111 0 A9 A8 0). Following the high byte address match, the low byte of the address needs to be loaded (A7:A0).

## 9.2.1 SLAVE MODE

In Slave mode, the SCL and SDA pins must be configured as inputs. The MSSP 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.

There are certain conditions that will cause the MSSP module not to give this  $\overline{ACK}$  pulse. These are if either (or both):

- 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.

If the BF bit is set, the SSPSR register value is not loaded into the SSPBUF, but bit SSPIF and SSPOV are 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 time for proper operation. The high and low times of the I<sup>2</sup>C specification, as well as the requirement of the MSSP module, is shown in timing parameter #100 and parameter #101 of the electrical specifications.

## 9.2.1.1 ADDRESSING

Once the MSSP module has been enabled, it waits for a START condition to occur. Following the START condition, the 8-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 on the falling edge of the 8th SCL pulse.
- b) The buffer full bit, BF, is set on the falling edge of the 8th SCL pulse.
- 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 9th SCL pulse.

In 10-bit address mode, two address bytes need to be received by the slave. The five Most Significant bits (MSBs) of the first address byte specify if this is a 10-bit address. Bit 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 a 10-bit address is as follows, with steps 7- 9 for slave-transmitter:

1. Receive first (high) byte of Address (bits SSPIF, BF and UA (SSPSTAT<1>) are set).
2. Update the SSPADD register with the 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. This will clear bit UA and release the SCL line.
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.

**Note:** Following the Repeated Start condition (step 7) in 10-bit mode, the user only needs to match the first 7-bit address. The user does not update the SSPADD for the second half of the address.

## 9.2.1.2 SLAVE 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 no acknowledge ( $\overline{ACK}$ ) pulse is given. An overflow condition is defined as 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 received byte.

**Note:** The SSPBUF will be loaded if the SSPOV bit is set and the BF flag is cleared. If a read of the SSPBUF was performed, but the user did not clear the state of the SSPOV bit before the next receive occurred, the  $\overline{ACK}$  is not sent and the SSPBUF is updated.

**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	Yes	No	Yes

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

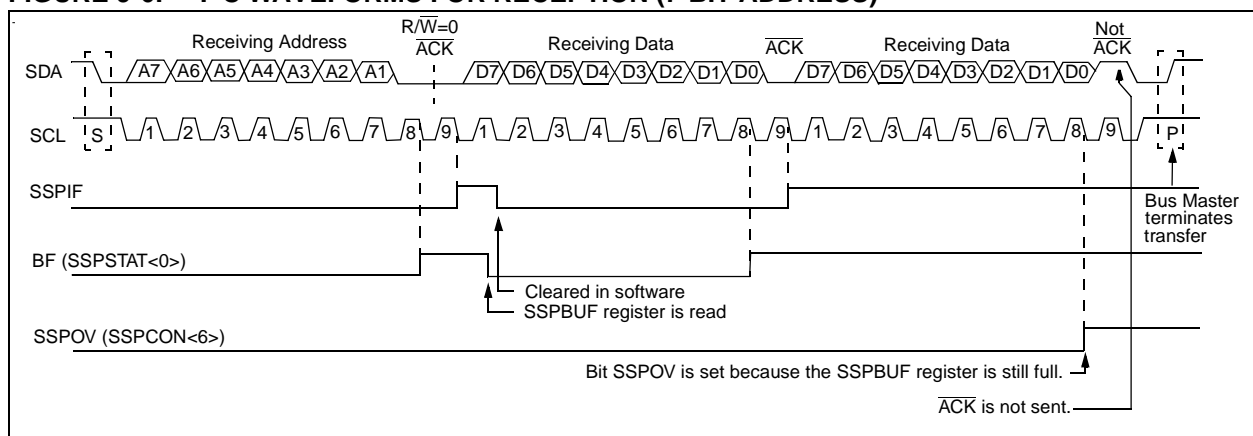
## 9.2.1.3 SLAVE TRANSMISSION

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 the SCL pin is held low. The transmit data must be loaded into the SSPBUF register, which also loads the SSPSR register. Then the SCL pin should be enabled by setting bit CKP (SSPCON<4>). The master must monitor the SCL pin prior to asserting another clock pulse. The slave devices may be holding off the master 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).

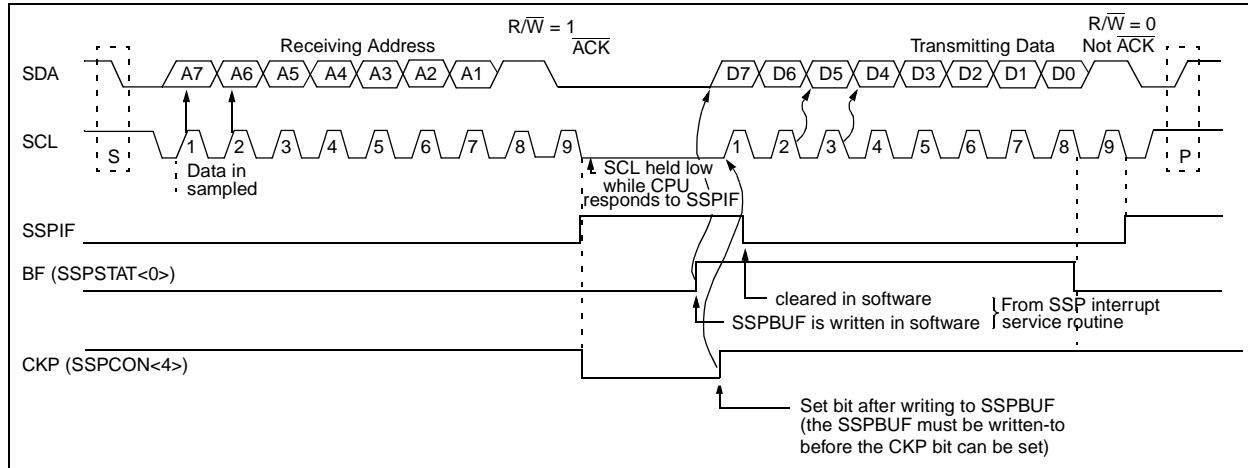
An SSP interrupt is generated for each data transfer byte. The SSPIF flag bit must be cleared in software and the SSPSTAT register is used to determine the status of the byte transfer. The SSPIF flag bit is set on the falling edge of the ninth clock pulse.

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 is high (not  $\overline{ACK}$ ), then the data transfer is complete. When the not  $\overline{ACK}$  is latched by the slave, the slave logic is reset and the slave 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 the SCL pin should be enabled by setting the CKP bit.

**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)**



## 9.2.2 GENERAL CALL ADDRESS SUPPORT

The addressing procedure for the I<sup>2</sup>C bus is such that the first byte after the START condition usually determines which device will be the slave addressed by the master. The exception is the general call address, which can address all devices. When this address is used, all devices should, in theory, respond with an acknowledge.

The general call address is one of eight addresses reserved for specific purposes by the I<sup>2</sup>C protocol. It consists of all 0's with R/W = 0

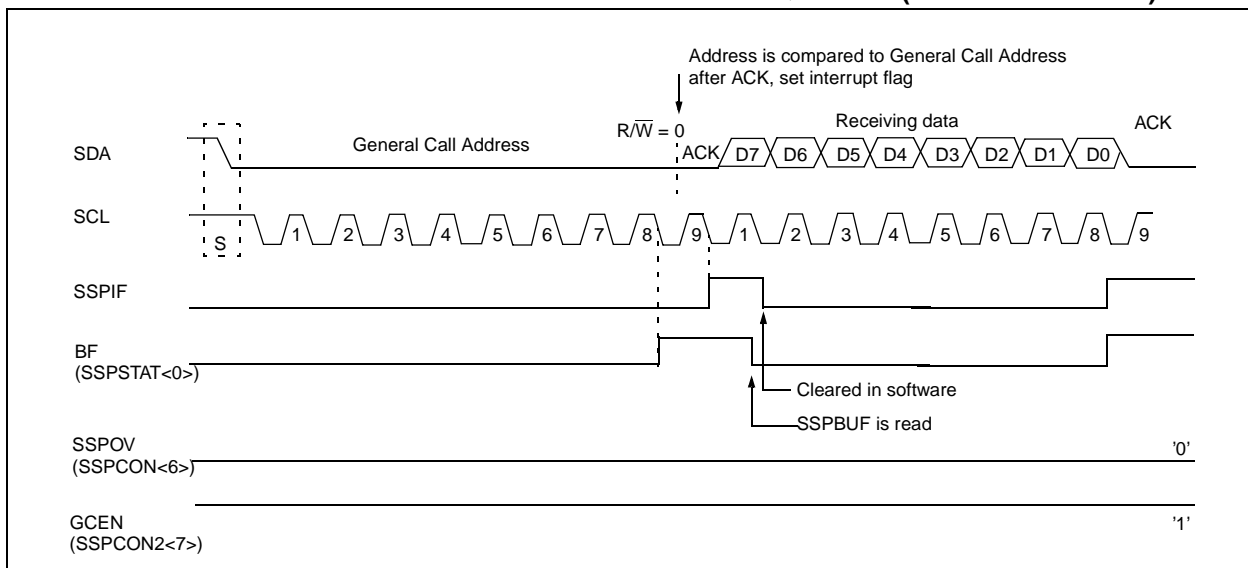
The general call address is recognized when the General Call Enable bit (GCEN) is enabled (SSPCON2<7> is set). Following a start-bit detect, 8 bits are shifted into SSPSR and the address is compared against SSPADD. It is also compared to the general call address and fixed in hardware.

If the general call address matches, the SSPSR is transferred to the SSPBUF, the BF flag is set (eighth bit), and on the falling edge of the ninth bit (ACK bit), the SSPIF flag is set.

When the interrupt is serviced, the source for the interrupt can be checked by reading the contents of the SSPBUF to determine if the address was device specific or a general call address.

In 10-bit mode, the SSPADD is required to be updated for the second half of the address to match, and the UA bit is set (SSPSTAT<1>). If the general call address is sampled when GCEN is set while the slave is configured in 10-bit address mode, then the second half of the address is not necessary, the UA bit will not be set, and the slave will begin receiving data after the acknowledge (Figure 9-8).

**FIGURE 9-8: SLAVE MODE GENERAL CALL ADDRESS SEQUENCE (7 OR 10-BIT MODE)**





## 9.2.3 SLEEP OPERATION

While in SLEEP mode, the I<sup>2</sup>C module can receive addresses or data. When an address match or complete byte transfer occurs, wake the processor from sleep (if the SSP interrupt is enabled).

## 9.2.4 EFFECTS OF A RESET

A RESET disables the SSP module and terminates the current transfer.

**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	POR, BOR	MCLR, WDT
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	(1)	ADIF	(1)	(1)	SSPIF	CCP1IF	TMR2IF	TMR1IF	r0rr 0000	r0rr 0000
8Ch	PIE1	(1)	ADIE	(1)	(1)	SSPIE	CCP1IE	TMR2IE	TMR1IE	r0rr 0000	r0rr 0000
0Dh	PIR2	—	(1)	—	EEIF	BCLIF	—	—	(1)	-r-0 0--r	-r-0 0--r
8Dh	PIE2	—	(1)	—	EEIE	BCLIE	—	—	(1)	-r-0 0--r	-r-0 0--r
13h	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								xxxx xxxx	uuuu uuuu
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000
91h	SSPCON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	0000 0000	0000 0000
94h	SSPSTAT	SMP	CKE	D/A	P	S	R/W	UA	BF	0000 0000	0000 0000

Legend: x = unknown, u = unchanged, r = reserved, - = unimplemented read as '0'. Shaded cells are not used by the SSP in I<sup>2</sup>C mode.

**Note 1:** These bits are reserved; always maintain these bits clear.

# PIC16F872

## 9.2.5 MASTER MODE

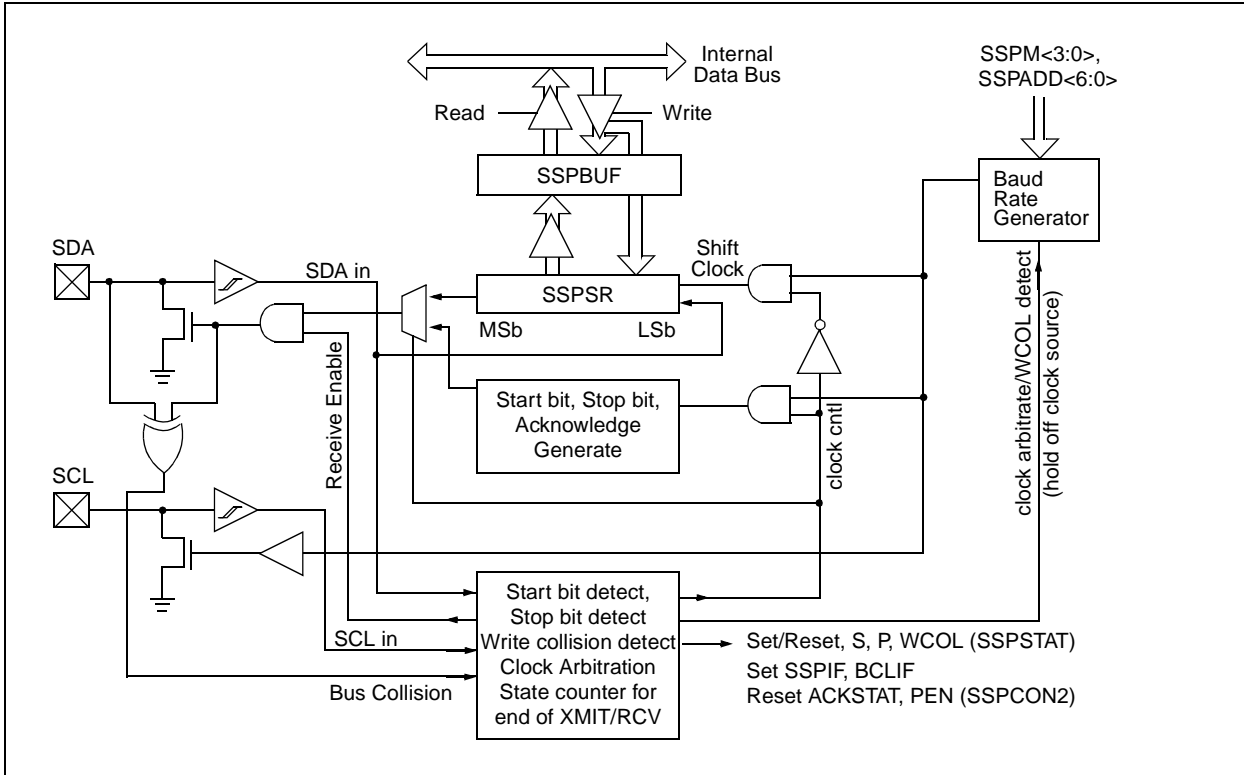
Master mode of operation is supported by interrupt generation on the detection of the START and STOP conditions. The STOP (P) and START (S) bits are cleared from a RESET or when the MSSP module is disabled. Control of the I<sup>2</sup>C bus may be TAKEN when the P bit is set, or the bus is idle with both the S and P bits clear.

In Master mode, the SCL and SDA lines are manipulated by the MSSP hardware.

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
- Acknowledge transmit
- Repeated Start

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



## 9.2.6 MULTI-MASTER MODE

In Multi-Master mode, 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 MSSP module is disabled. Control of the I<sup>2</sup>C bus may be TAKEN when bit P (SSPSTAT<4>) is set, or the bus is idle with 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 operation, the SDA line must be monitored for arbitration to see if the signal level is the expected output level. This check is performed in hardware, with the result placed in the BCLIF bit.

The states where arbitration can be lost are:

- Address Transfer
- Data Transfer
- A Start Condition
- A Repeated Start Condition
- An Acknowledge Condition

## 9.2.7 I<sup>2</sup>C MASTER MODE SUPPORT

Master mode is enabled by setting and clearing the appropriate SSPM bits in SSPCON and by setting the SSPEN bit. Once Master mode is enabled, the user has six options.

- Assert a start condition on SDA and SCL.
- Assert a Repeated Start condition on SDA and SCL.
- Write to the SSPBUF register initiating transmission of data/address.
- Generate a stop condition on SDA and SCL.
- Configure the I<sup>2</sup>C port to receive data.
- Generate an Acknowledge condition at the end of a received byte of data.

**Note:** The MSSP Module, when configured in I<sup>2</sup>C Master mode, does not allow queuing of events. For instance, the user is not allowed to initiate a start condition and immediately write the SSPBUF register to initiate transmission before the START condition is complete. In this case, the SSPBUF will not be written to and the WCOL bit will be set, indicating that a write to the SSPBUF did not occur.

## 9.2.7.4 I<sup>2</sup>C MASTER MODE OPERATION

The master device generates all of the serial clock pulses and the START and STOP conditions. A transfer is ended with a STOP condition or with a Repeated Start condition. Since the Repeated Start condition is also the beginning of the next serial transfer, the I<sup>2</sup>C bus will not be released.

In Master Transmitter mode, serial data is output through SDA, while SCL outputs the serial clock. The first byte transmitted contains the slave address of the receiving device (7 bits) and the Read/Write (R/W) bit. In this case, the R/W bit will be logic '0'. Serial data is transmitted 8 bits at a time. After each byte is transmitted, an acknowledge bit is received. START and STOP conditions are output to indicate the beginning and the end of a serial transfer.

In Master Receive mode, the first byte transmitted contains the slave address of the transmitting device (7 bits) and the R/W bit. In this case, the R/W bit will be logic '1'. Thus, the first byte transmitted is a 7-bit slave address followed by a '1' to indicate receive bit. Serial data is received via SDA, while SCL outputs the serial clock. Serial data is received 8 bits at a time. After each byte is received, an acknowledge bit is transmitted. START and STOP conditions indicate the beginning and end of transmission.

The baud rate generator used for SPI mode operation is now used to set the SCL clock frequency for either 100 kHz, 400 kHz or 1 MHz I<sup>2</sup>C operation. The baud rate generator reload value is contained in the lower 7 bits of the SSPADD register. The baud rate generator will automatically begin counting on a write to the SSPBUF. Once the given operation is complete (i.e., transmission of the last data bit is followed by ACK), the internal clock will automatically stop counting and the SCL pin will remain in its last state

A typical transmit sequence would go as follows:

- a) The user generates a Start Condition by setting the START enable bit (SEN) in SSPCON2.
- b) SSPIF is set. The module will wait the required start time before any other operation takes place.
- c) The user loads the SSPBUF with address to transmit.
- d) Address is shifted out the SDA pin until all 8 bits are transmitted.
- e) The MSSP Module shifts in the ACK bit from the slave device and writes its value into the SSPCON2 register (SSPCON2<6>).
- f) The module generates an interrupt at the end of the ninth clock cycle by setting SSPIF.
- g) The user loads the SSPBUF with eight bits of data.
- h) DATA is shifted out the SDA pin until all 8 bits are transmitted.

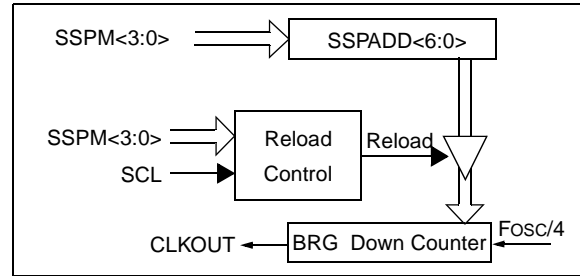
- i) The MSSP module shifts in the ACK bit from the slave device, and writes its value into the SSPCON2 register ( SSPCON2<6>).
- j) The MSSP module generates an interrupt at the end of the ninth clock cycle by setting the SSPIF bit.
- k) The user generates a STOP condition by setting the STOP enable bit PEN in SSPCON2.
- l) Interrupt is generated once the STOP condition is complete.

## 9.2.8 BAUD RATE GENERATOR

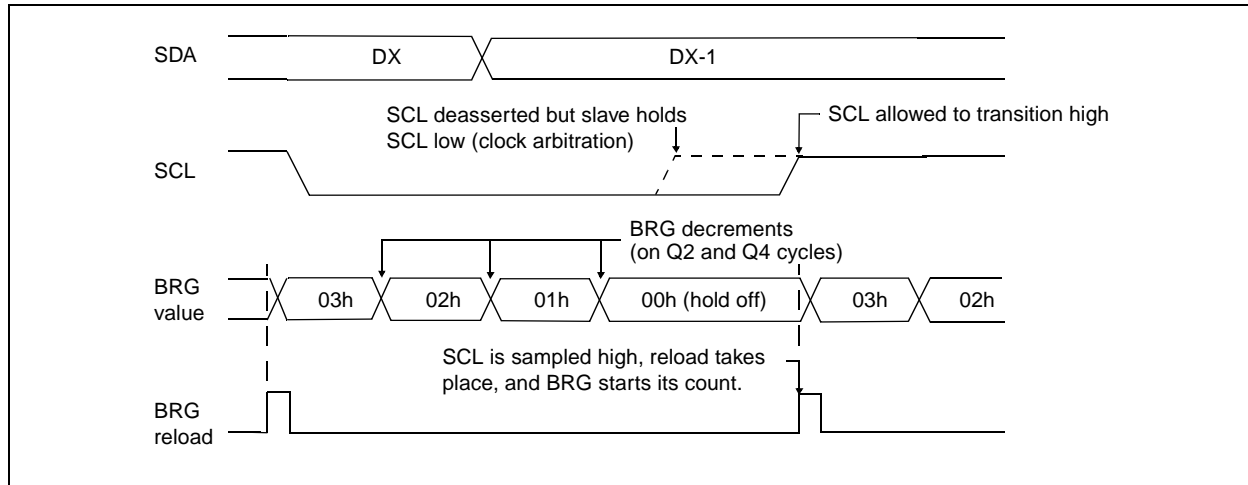
In I<sup>2</sup>C Master mode, the reload value for the BRG is located in the lower 7 bits of the SSPADD register (Figure 9-10). When the BRG is loaded with this value, the BRG counts down to 0 and stops until another reload has TAKEN place. The BRG count is decremented twice per instruction cycle (TCY), on the Q2 and Q4 clock.

In I<sup>2</sup>C Master mode, the BRG is reloaded automatically. If Clock Arbitration is taking place for instance, the BRG will be reloaded when the SCL pin is sampled high (Figure 9-11).

**FIGURE 9-10: BAUD RATE GENERATOR BLOCK DIAGRAM**



**FIGURE 9-11: BAUD RATE GENERATOR TIMING WITH CLOCK ARBITRATION**



## 9.2.9 I<sup>2</sup>C MASTER MODE START CONDITION TIMING

To initiate a START condition, the user sets the start condition enable bit, SEN (SSPCON2<0>). If the SDA and SCL pins are sampled high, the baud rate generator is re-loaded with the contents of SSPADD<6:0> and starts its count. If SCL and SDA are both sampled high when the baud rate generator times out (TBRG), the SDA pin is driven low. The action of the SDA being driven low while SCL is high is the START condition, and causes the S bit (SSPSTAT<3>) to be set. Following this, the baud rate generator is reloaded with the contents of SSPADD<6:0> and resumes its count. When the baud rate generator times out (TBRG), the SEN bit (SSPCON2<0>) will be automatically cleared by hardware. The baud rate generator is suspended leaving the SDA line held low, and the START condition is complete.

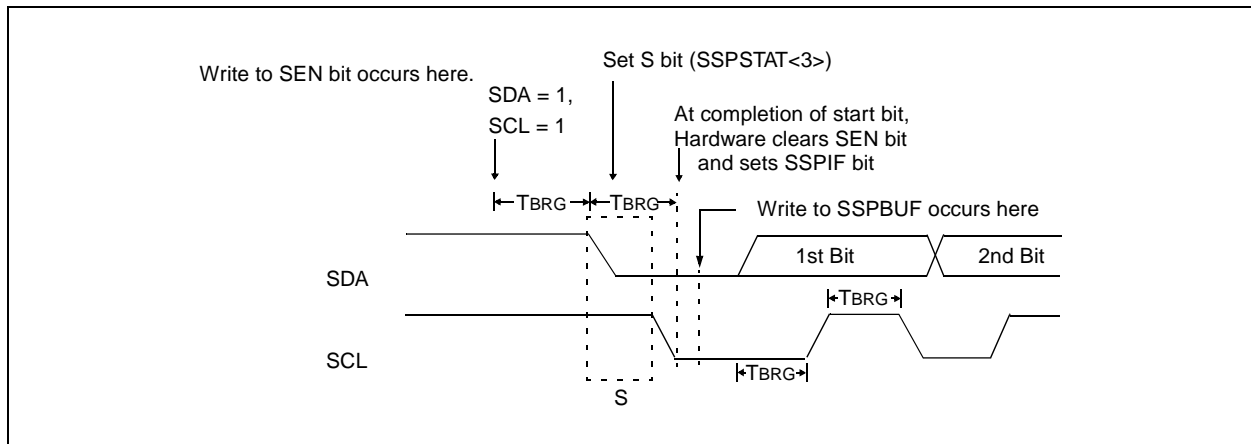
**Note:** If at the beginning of START condition the SDA and SCL pins are already sampled low, or if during the START condition the SCL line is sampled low before the SDA line is driven low, a bus collision occurs, the Bus Collision Interrupt Flag (BCLIF) is set, the START condition is aborted, and the I<sup>2</sup>C module is reset into its IDLE state.

### 9.2.9.5 WCOL STATUS FLAG

If the user writes the SSPBUF when an START sequence is in progress, then WCOL is set and the contents of the buffer are unchanged (the write doesn't occur).

**Note:** Because queueing of events is not allowed, writing to the lower 5 bits of SSPCON2 is disabled until the START condition is complete.

**FIGURE 9-12: FIRST START BIT TIMING**



## 9.2.10 I<sup>2</sup>C MASTER MODE REPEATED START CONDITION TIMING

A Repeated Start condition occurs when the RSEN bit (SSPCON2<1>) is programmed high and the I<sup>2</sup>C module is in the idle state. When the RSEN bit is set, the SCL pin is asserted low. When the SCL pin is sampled low, the baud rate generator is loaded with the contents of SSPADD<6:0> and begins counting. The SDA pin is released (brought high) for one baud rate generator count (TBRG). When the baud rate generator times out if SDA is sampled high, the SCL pin will be deasserted (brought high). When SCL is sampled high the baud rate generator is reloaded with the contents of SSPADD<6:0> and begins counting. SDA and SCL must be sampled high for one TBRG. This action is then followed by assertion of the SDA pin (SDA is low) for one TBRG, while SCL is high. Following this, the RSEN bit in the SSPCON2 register will be automatically cleared and the baud rate generator will not be reloaded, leaving the SDA pin held low. As soon as a start condition is detected on the SDA and SCL pins, the S bit (SSPSTAT<3>) will be set. The SSPIF bit will not be set until the baud rate generator has timed-out.

**Note 1:** If RSEN is programmed while any other event is in progress, it will not take effect.

**Note 2:** A bus collision during the Repeated Start condition occurs if:

- SDA is sampled low when SCL goes from low to high.
- SCL goes low before SDA is asserted low. This may indicate that another master is attempting to transmit a data "1".

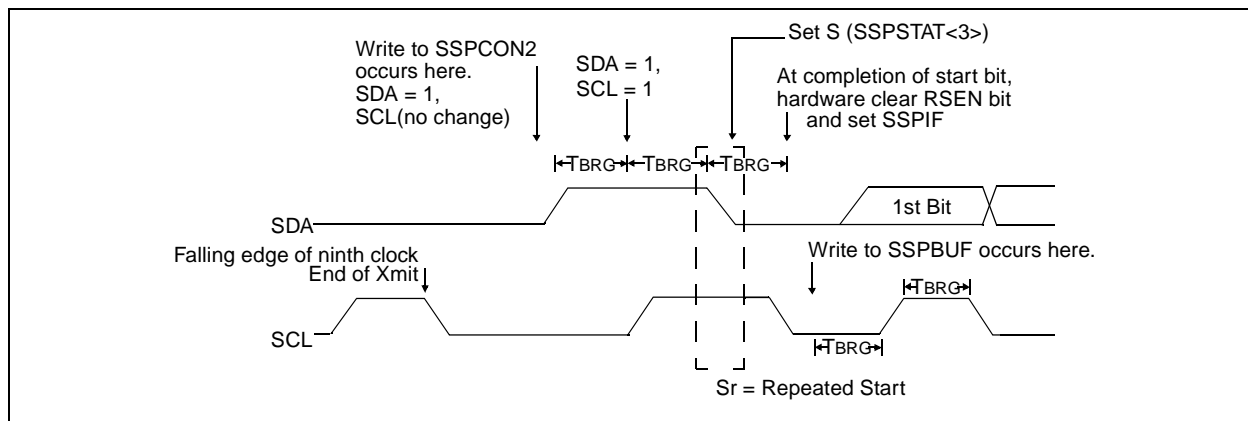
Immediately following the SSPIF bit getting set, the user may write the SSPBUF with the 7-bit address in 7-bit mode, or the default first address in 10-bit mode. After the first eight bits are transmitted and an ACK is received, the user may then transmit an additional eight bits of address (10-bit mode) or eight bits of data (7-bit mode).

### 9.2.10.6 WCOL STATUS FLAG

If the user writes the SSPBUF when a Repeated Start sequence is in progress, then WCOL is set and the contents of the buffer are unchanged (the write doesn't occur).

**Note:** Because queuing of events is not allowed, writing of the lower 5 bits of SSPCON2 is disabled until the Repeated Start condition is complete.

**FIGURE 9-13: REPEAT START CONDITION WAVEFORM**



### 9.2.11 I<sup>2</sup>C MASTER MODE TRANSMISSION

Transmission of a data byte, a 7-bit address or either half of a 10-bit address, is accomplished by simply writing a value to SSPBUF register. This action will set the buffer full flag (BF) and allow the baud rate generator to begin counting and start the next transmission. Each bit of address/data will be shifted out onto the SDA pin after the falling edge of SCL is asserted (see data hold time spec). SCL is held low for one baud rate generator rollover count (TBRG). Data should be valid before SCL is released high (see data setup time spec). When the SCL pin is released high, it is held that way for TBRG. The data on the SDA pin must remain stable for that duration and some hold time after the next falling edge of SCL. After the eighth bit is shifted out (the falling edge of the eighth clock), the BF flag is cleared and the master releases SDA allowing the slave device being addressed to respond with an  $\overline{\text{ACK}}$  bit during the ninth bit time, if an address match occurs or if data was received properly. The status of  $\overline{\text{ACK}}$  is read into the ACKDT on the falling edge of the ninth clock. If the master receives an acknowledge, the acknowledge status bit (ACKSTAT) is cleared. If not, the bit is set. After the ninth clock, the SSPIF is set and the master clock (baud rate generator) is suspended until the next data byte is loaded into the SSPBUF, leaving SCL low and SDA unchanged (Figure 9-14).

After the write to the SSPBUF, each bit of address will be shifted out on the falling edge of SCL until all seven address bits and the  $\overline{\text{R}/\text{W}}$  bit are completed. On the falling edge of the eighth clock, the master will de-assert the SDA pin allowing the slave to respond with an acknowledge. On the falling edge of the ninth clock, the master will sample the SDA pin to see if the address was recognized by a slave. The status of the ACK bit is loaded into the ACKSTAT status bit (SSPCON2<6>). Following the falling edge of the ninth clock transmission of the address, the SSPIF is set, the BF flag is cleared, and the baud rate generator is turned off until another write to the SSPBUF takes place, holding SCL low and allowing SDA to float.

#### 9.2.11.7 BF STATUS FLAG

In Transmit mode, the BF bit (SSPSTAT<0>) is set when the CPU writes to SSPBUF and is cleared when all 8 bits are shifted out.

#### 9.2.11.8 WCOL STATUS FLAG

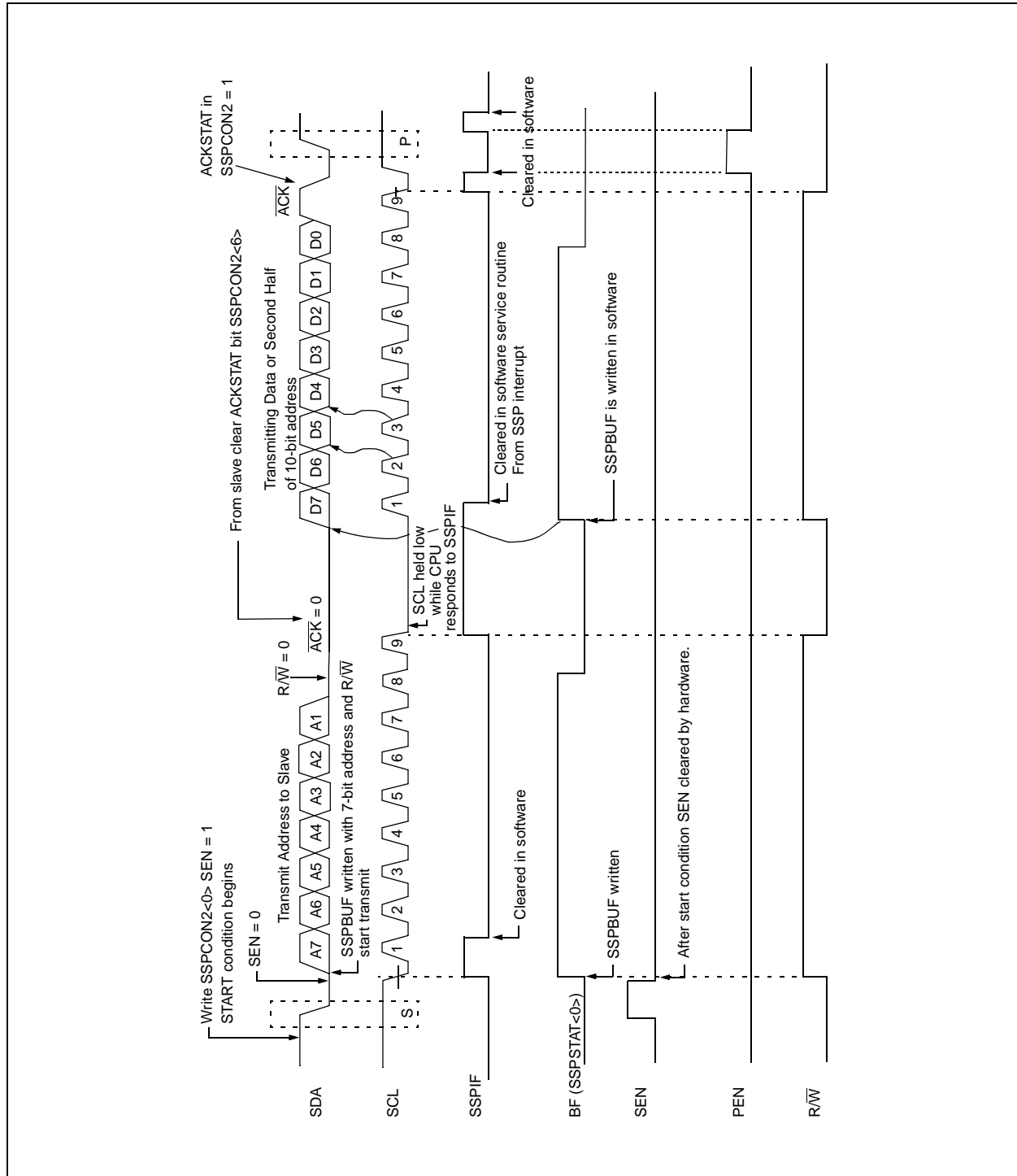
If the user writes the SSPBUF when a transmit is already in progress (i.e., SSPSR is still shifting out a data byte), then WCOL is set and the contents of the buffer are unchanged (the write doesn't occur).

WCOL must be cleared in software.

#### 9.2.11.9 ACKSTAT STATUS FLAG

In Transmit mode, the ACKSTAT bit (SSPCON2<6>) is cleared when the slave has sent an acknowledge ( $\overline{\text{ACK}} = 0$ ) and is set when the slave does not acknowledge ( $\overline{\text{ACK}} = 1$ ). A slave sends an acknowledge when it has recognized its address (including a general call) or when the slave has properly received its data.

FIGURE 9-14: I<sup>2</sup>C MASTER MODE TIMING (TRANSMISSION, 7 OR 10-BIT ADDRESS)





---

### 9.2.12 I<sup>2</sup>C MASTER MODE RECEPTION

Master mode reception is enabled by programming the receive enable bit, RCEN (SSPCON2<3>).

<p><b>Note:</b> The SSP module must be in an IDLE STATE before the RCEN bit is set or the RCEN bit will be disregarded.</p>
---

The baud rate generator begins counting, and on each rollover, the state of the SCL pin changes (high to low/low to high), and data is shifted into the SSPSR. After the falling edge of the eighth clock, the receive enable flag is automatically cleared, the contents of the SSPSR are loaded into the SSPBUF, the BF flag is set, the SSPIF is set, and the baud rate generator is suspended from counting, holding SCL low. The SSP is now in IDLE state, awaiting the next command. When the buffer is read by the CPU, the BF flag is automatically cleared. The user can then send an acknowledge bit at the end of reception, by setting the acknowledge sequence enable bit, ACKEN (SSPCON2<4>).

#### 9.2.12.10 BF STATUS FLAG

In receive operation, BF is set when an address or data byte is loaded into SSPBUF from SSPSR. It is cleared when SSPBUF is read.

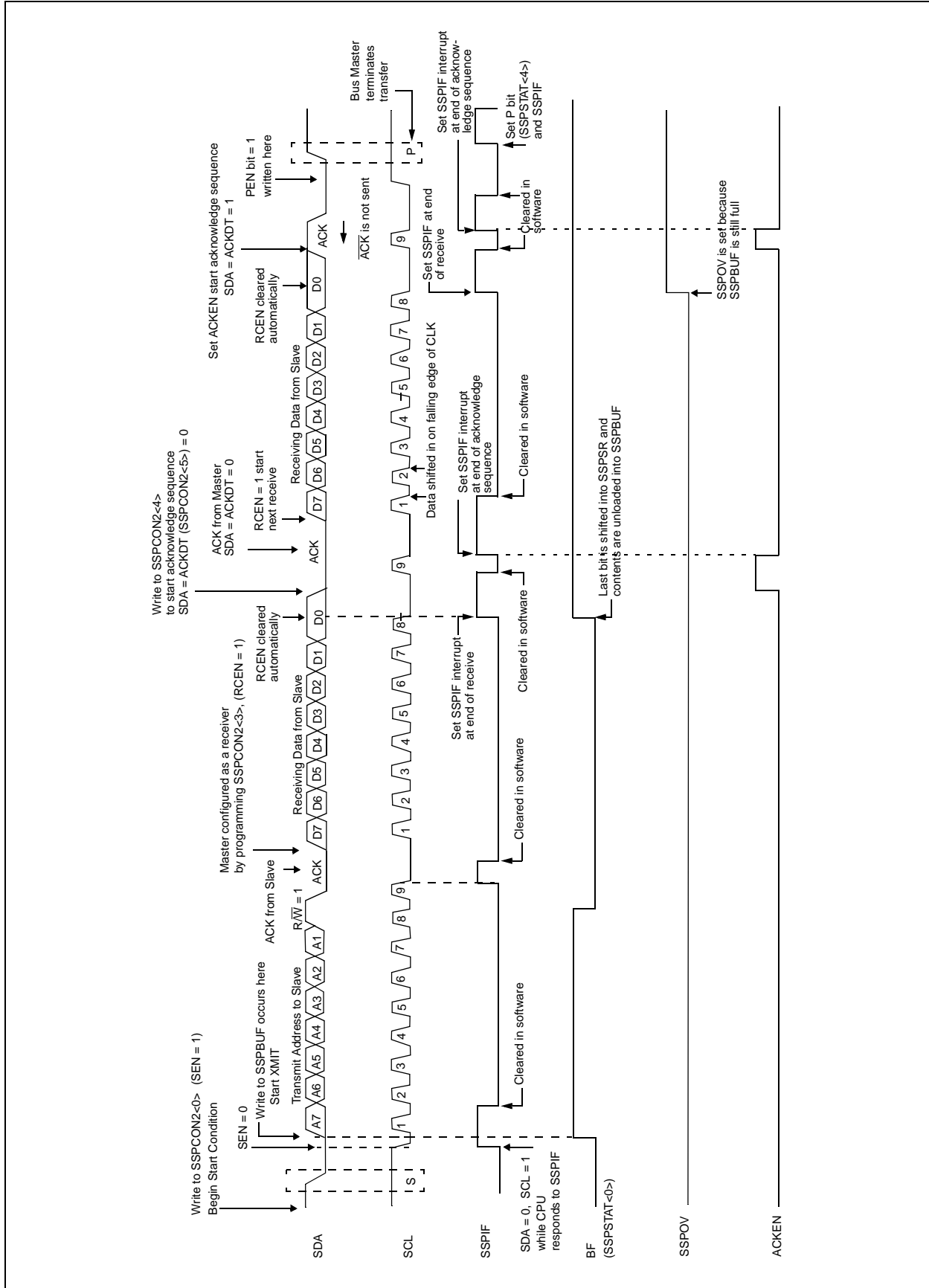
#### 9.2.12.11 SSPOV STATUS FLAG

In receive operation, SSPOV is set when 8 bits are received into the SSPSR and the BF flag is already set from a previous reception.

#### 9.2.12.12 WCOL STATUS FLAG

If the user writes the SSPBUF when a receive is already in progress (i.e., SSPSR is still shifting in a data byte), then WCOL is set and the contents of the buffer are unchanged (the write doesn't occur).

**FIGURE 9-15: I<sup>2</sup>C MASTER MODE TIMING (RECEPTION 7-BIT ADDRESS)**



## 9.2.13 ACKNOWLEDGE SEQUENCE TIMING

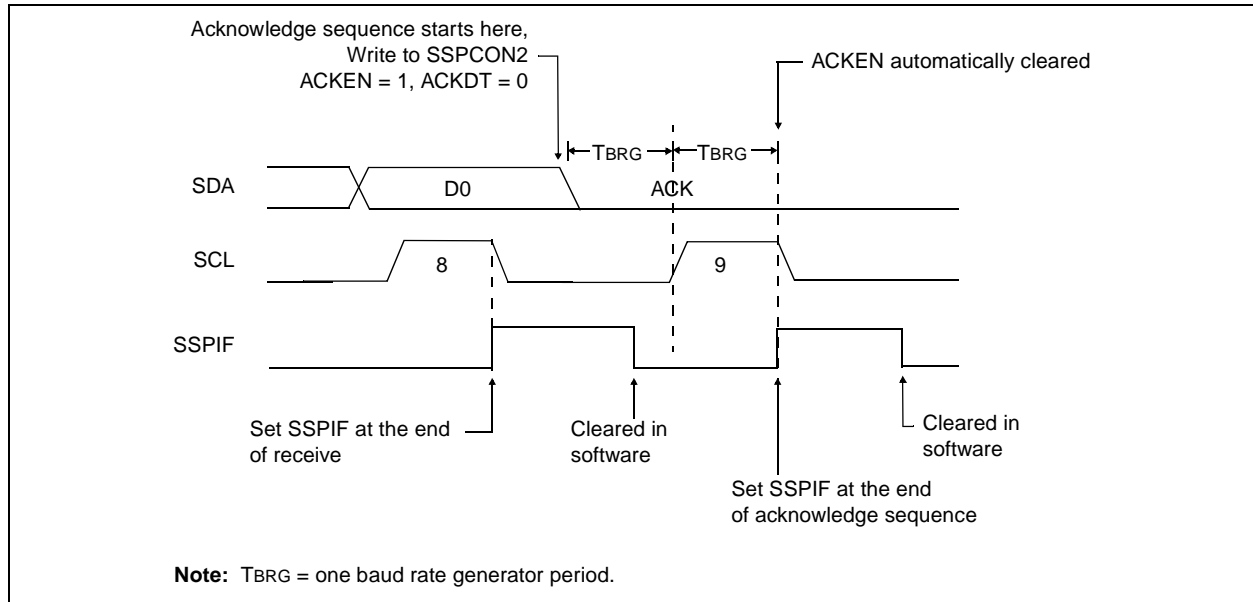
An acknowledge sequence is enabled by setting the acknowledge sequence enable bit, ACKEN (SSPCON2<4>). When this bit is set, the SCL pin is pulled low and the contents of the acknowledge data bit is presented on the SDA pin. If the user wishes to generate an acknowledge, the ACKDT bit should be cleared. If not, the user should set the ACKDT bit before starting an acknowledge sequence. The baud rate generator then counts for one rollover period (TBRG), and the SCL pin is deasserted (pulled high). When the SCL pin is sampled high (clock arbitration),

the baud rate generator counts for TBRG. The SCL pin is then pulled low. Following this, the ACKEN bit is automatically cleared, the baud rate generator is turned off, and the SSP module then goes into IDLE mode. (Figure 9-16)

### 9.2.13.13 WCOL STATUS FLAG

If the user writes the SSPBUF when an acknowledge sequence is in progress, the WCOL is set and the contents of the buffer are unchanged (the write doesn't occur).

**FIGURE 9-16: ACKNOWLEDGE SEQUENCE WAVEFORM**



## 9.2.14 STOP CONDITION TIMING

A stop bit is asserted on the SDA pin at the end of a receive/transmit by setting the Stop Sequence Enable bit PEN (SSPCON2<2>). At the end of a receive/transmit, the SCL line is held low after the falling edge of the ninth clock. When the PEN bit is set, the master will assert the SDA line low. When the SDA line is sampled low, the baud rate generator is reloaded and counts down to 0. When the baud rate generator times out, the SCL pin will be brought high, and one TBRG (baud rate generator rollover count) later, the SDA pin will be deasserted. When the SDA pin is sampled high

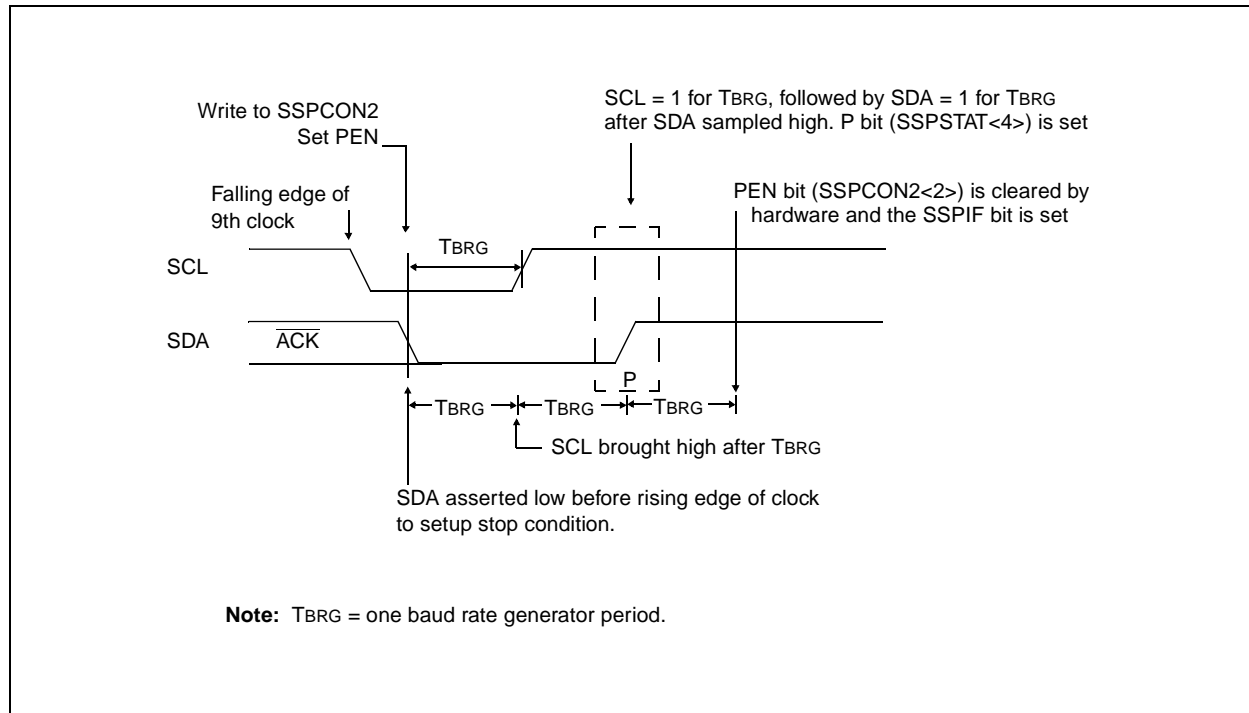
while SCL is high, the P bit (SSPSTAT<4>) is set. A TBRG later, the PEN bit is cleared and the SSPIF bit is set (Figure 9-17).

Whenever the firmware decides to take control of the bus, it will first determine if the bus is busy by checking the S and P bits in the SSPSTAT register. If the bus is busy, then the CPU can be interrupted (notified) when a Stop bit is detected (i.e., bus is free).

### 9.2.14.14 WCOL STATUS FLAG

If the user writes the SSPBUF when a STOP sequence is in progress, then WCOL is set and the contents of the buffer are unchanged (the write doesn't occur).

**FIGURE 9-17: STOP CONDITION RECEIVE OR TRANSMIT MODE**



## 9.2.15 CLOCK ARBITRATION

Clock arbitration occurs when the master, during any receive, transmit, or repeated start/stop condition, deasserts the SCL pin (SCL allowed to float high). When the SCL pin is allowed to float high, the baud rate generator (BRG) is suspended from counting until the SCL pin is actually sampled high. When the SCL pin is sampled high, the baud rate generator is reloaded with the contents of SSPADD<6:0> and begins counting. This ensures that the SCL high time will always be at least one BRG rollover count in the event that the clock is held low by an external device (Figure 9-18).

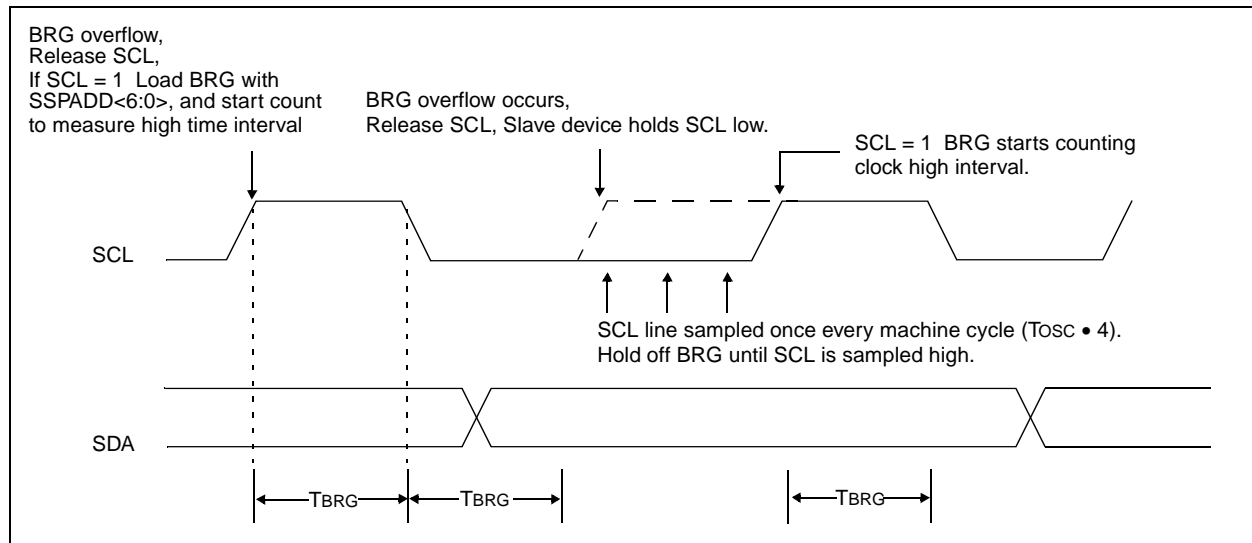
## 9.2.16 SLEEP OPERATION

While in SLEEP mode, the I<sup>2</sup>C module can receive addresses or data, and when an address match or complete byte transfer occurs, wake the processor from sleep (if the SSP interrupt is enabled).

## 9.2.17 EFFECTS OF A RESET

A RESET disables the SSP module and terminates the current transfer.

**FIGURE 9-18: CLOCK ARBITRATION TIMING IN MASTER TRANSMIT MODE**



## 9.2.18 MULTI-MASTER COMMUNICATION, BUS COLLISION, AND BUS ARBITRATION

Multi-Master mode support is achieved by bus arbitration. When the master outputs address/data bits onto the SDA pin, arbitration takes place when the master outputs a '1' on SDA by letting SDA float high and another master asserts a '0'. When the SCL pin floats high, data should be stable. If the expected data on SDA is a '1' and the data sampled on the SDA pin = '0', a bus collision has taken place. The master will set the Bus Collision Interrupt Flag, BCLIF, and reset the I<sup>2</sup>C port to its IDLE state. (Figure 9-19).

If a transmit was in progress when the bus collision occurred, the transmission is halted, the BF flag is cleared, the SDA and SCL lines are deasserted, and the SSPBUF can be written to. When the user services the bus collision interrupt service routine, and if the I<sup>2</sup>C bus is free, the user can resume communication by asserting a START condition.

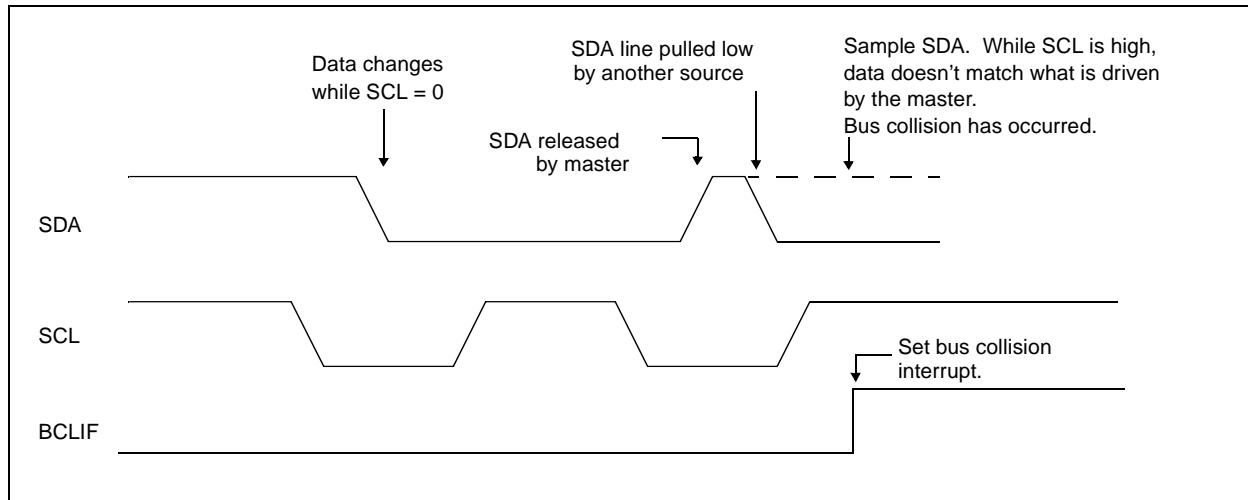
If a START, Repeated Start, STOP or Acknowledge condition was in progress when the bus collision occurred, the condition is aborted, the SDA and SCL lines are deasserted, and the respective control bits in the SSPCON2 register are cleared. When the user services the bus collision interrupt service routine, and if the I<sup>2</sup>C bus is free, the user can resume communication by asserting a START condition.

The master will continue to monitor the SDA and SCL pins, and if a STOP condition occurs, the SSPIF bit will be set.

A write to the SSPBUF will start the transmission of data at the first data bit, regardless of where the transmitter left off when the bus collision occurred.

In Multi-Master mode, the interrupt generation on the detection of start and stop conditions allows the determination of when the bus is free. Control of the I<sup>2</sup>C bus can be taken when the P bit is set in the SSPSTAT register, or the bus is idle and the S and P bits are cleared.

**FIGURE 9-19: BUS COLLISION TIMING FOR TRANSMIT AND ACKNOWLEDGE**



## 9.2.18.15 BUS COLLISION DURING A START CONDITION

During a START condition, a bus collision occurs if:

- SDA or SCL are sampled low at the beginning of the START condition (Figure 9-20).
- SCL is sampled low before SDA is asserted low. (Figure 9-21).

During a START condition both the SDA and the SCL pins are monitored.

If:

the SDA pin is already low  
or the SCL pin is already low,

then:

the START condition is aborted,  
and the BCLIF flag is set,  
and the SSP module is reset to its IDLE state  
 (Figure 9-20).

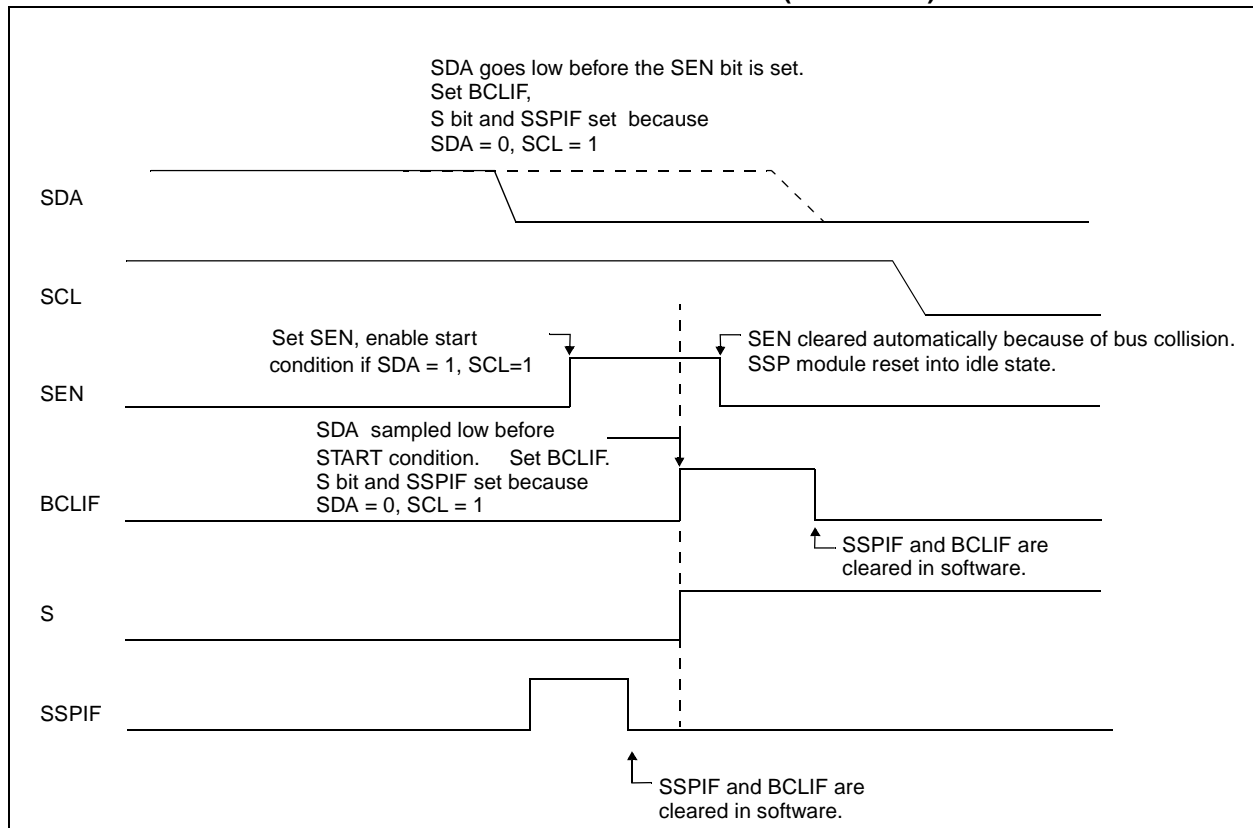
The START condition begins with the SDA and SCL pins deasserted. When the SDA pin is sampled high, the baud rate generator is loaded from SSPADD<6:0> and counts down to 0. If the SCL pin is sampled low

while SDA is high, a bus collision occurs, because it is assumed that another master is attempting to drive a data '1' during the START condition.

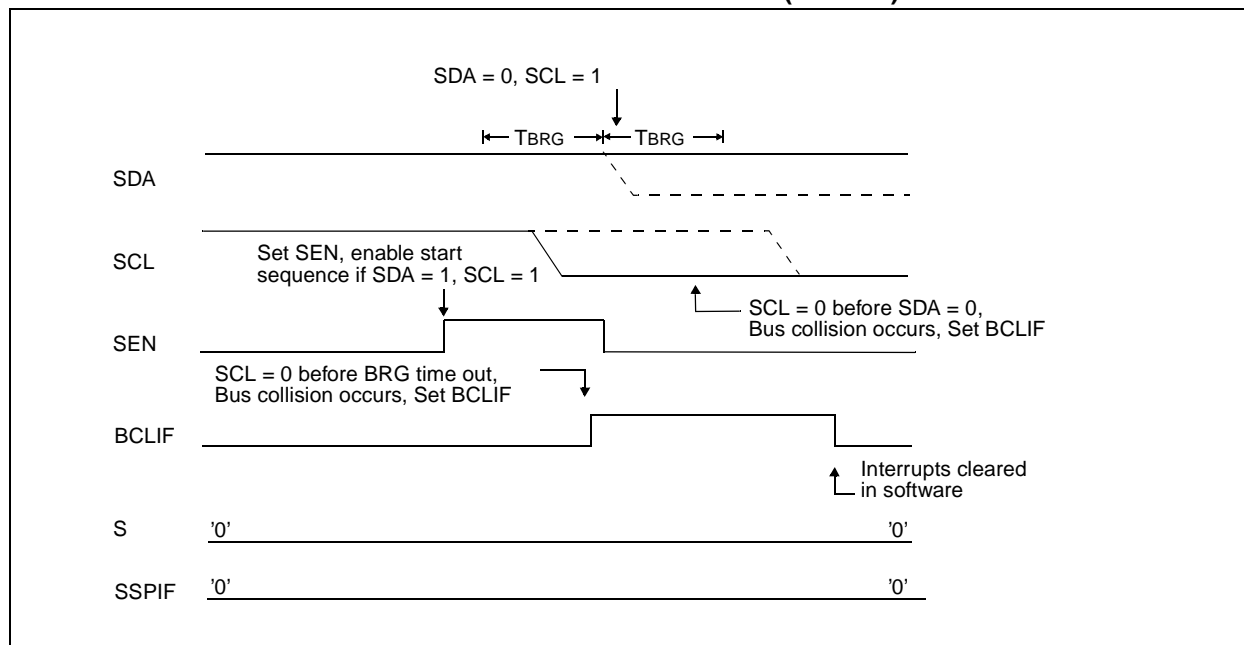
If the SDA pin is sampled low during this count, the BRG is reset and the SDA line is asserted early (Figure 9-22). If, however, a '1' is sampled on the SDA pin, the SDA pin is asserted low at the end of the BRG count. The baud rate generator is then reloaded and counts down to 0. During this time, if the SCL pins are sampled as '0', a bus collision does not occur. At the end of the BRG count, the SCL pin is asserted low.

**Note:** The reason that bus collision is not a factor during a START condition is that no two bus masters can assert a START condition at the exact same time. Therefore, one master will always assert SDA before the other. This condition does not cause a bus collision, because the two masters must be allowed to arbitrate the first address following the START condition. If the address is the same, arbitration must be allowed to continue into the data portion, REPEATED START or STOP conditions.

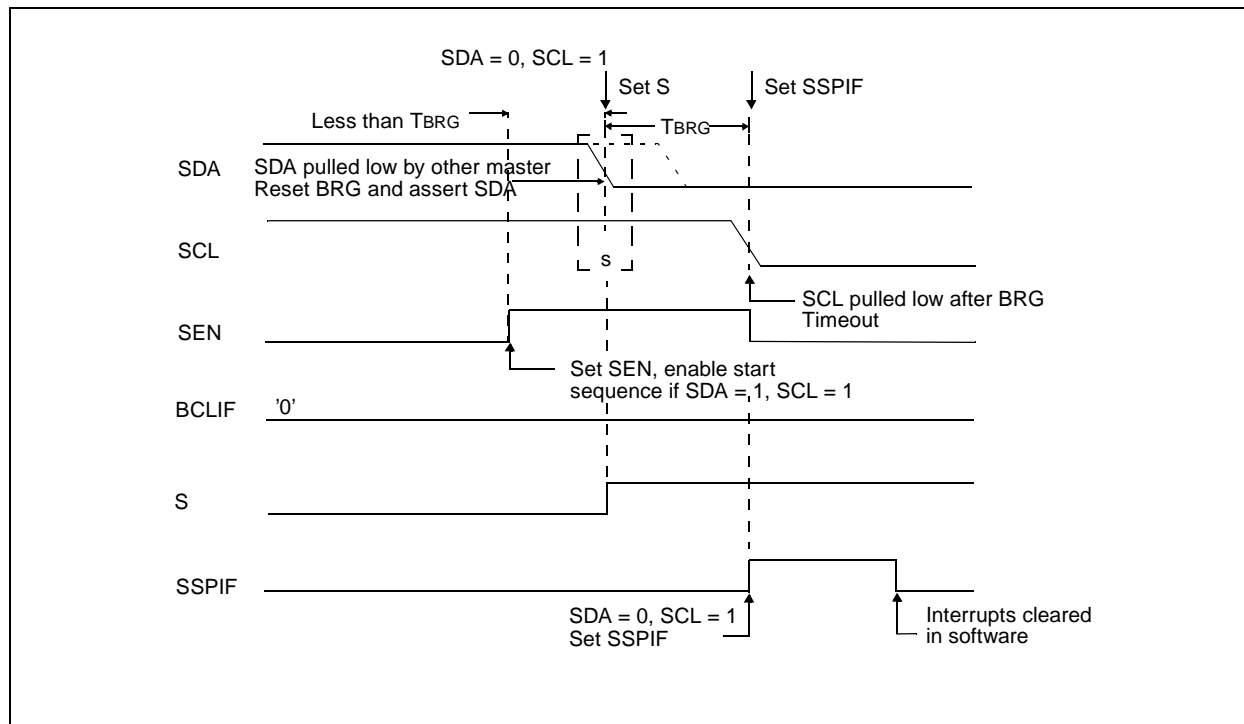
**FIGURE 9-20: BUS COLLISION DURING START CONDITION (SDA ONLY)**



**FIGURE 9-21: BUS COLLISION DURING START CONDITION (SCL = 0)**



**FIGURE 9-22: BRG RESET DUE TO SDA COLLISION DURING START CONDITION**





## 9.2.18.16 BUS COLLISION DURING A REPEATED START CONDITION

During a Repeated Start condition, a bus collision occurs if:

- A low level is sampled on SDA when SCL goes from low level to high level.
- SCL goes low before SDA is asserted low, indicating that another master is attempting to transmit a data '1'.

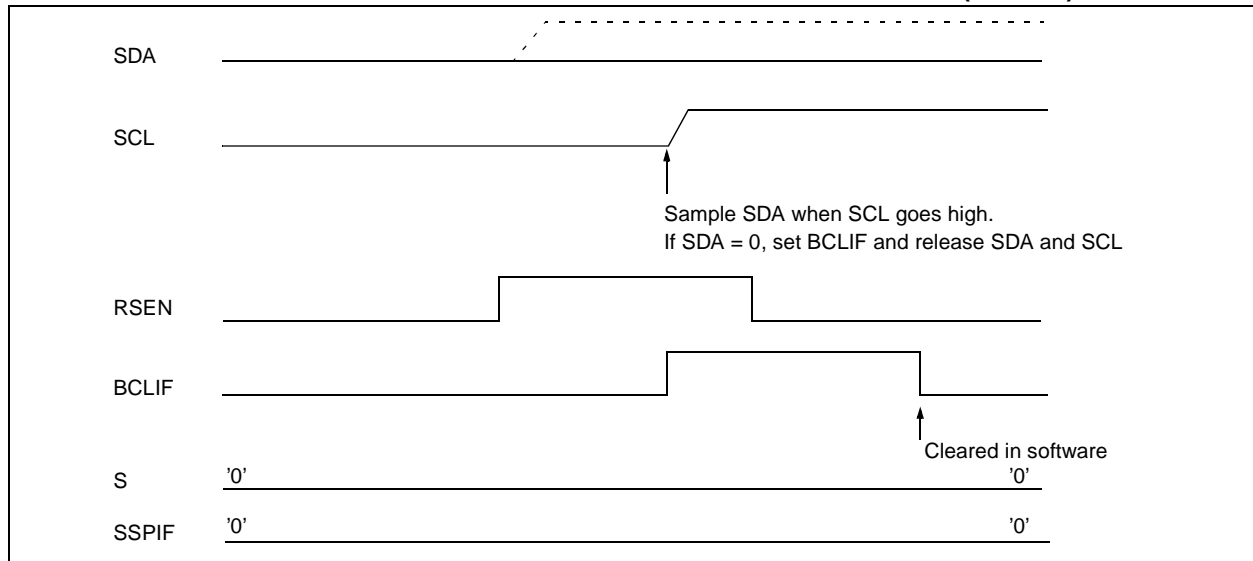
When the user deasserts SDA and the pin is allowed to float high, the BRG is loaded with SSPADD<6:0> and counts down to 0. The SCL pin is then deasserted, and when sampled high, the SDA pin is sampled. If SDA is low, a bus collision has occurred (i.e., another master is attempting to transmit a data '0'). If, however, SDA is

sampled high, the BRG is reloaded and begins counting. If SDA goes from high to low before the BRG times out, no bus collision occurs, because no two masters can assert SDA at exactly the same time.

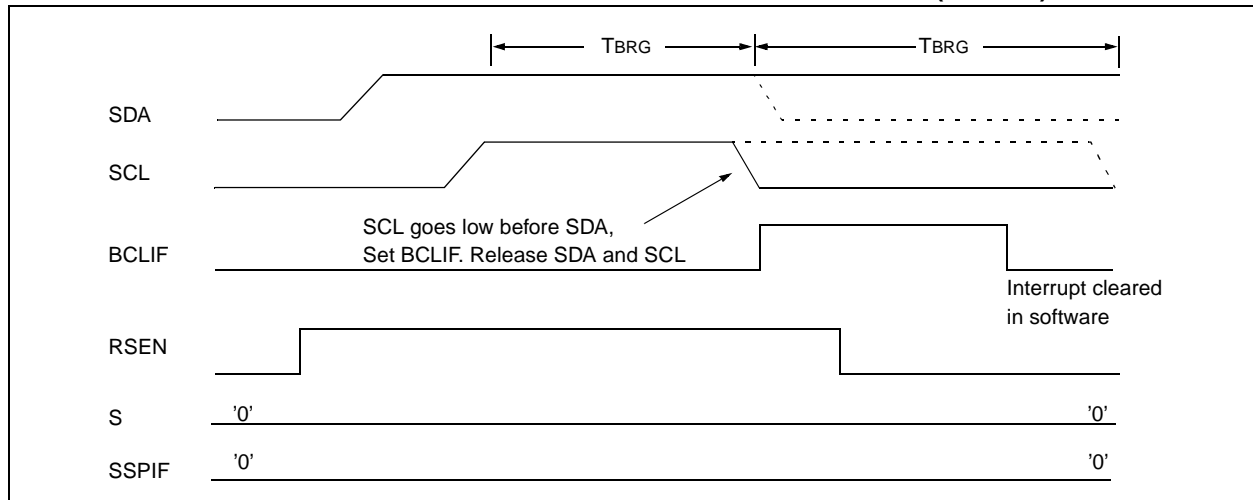
If, however, SCL goes from high to low before the BRG times out and SDA has not already been asserted, a bus collision occurs. In this case, another master is attempting to transmit a data '1' during the Repeated Start condition.

If, at the end of the BRG time out, both SCL and SDA are still high, the SDA pin is driven low, the BRG is reloaded and begins counting. At the end of the count, regardless of the status of the SCL pin, the SCL pin is driven low and the Repeated Start condition is complete (Figure 9-23).

**FIGURE 9-23: BUS COLLISION DURING A REPEATED START CONDITION (CASE 1)**



**FIGURE 9-24: BUS COLLISION DURING REPEATED START CONDITION (CASE 2)**



# PIC16F872

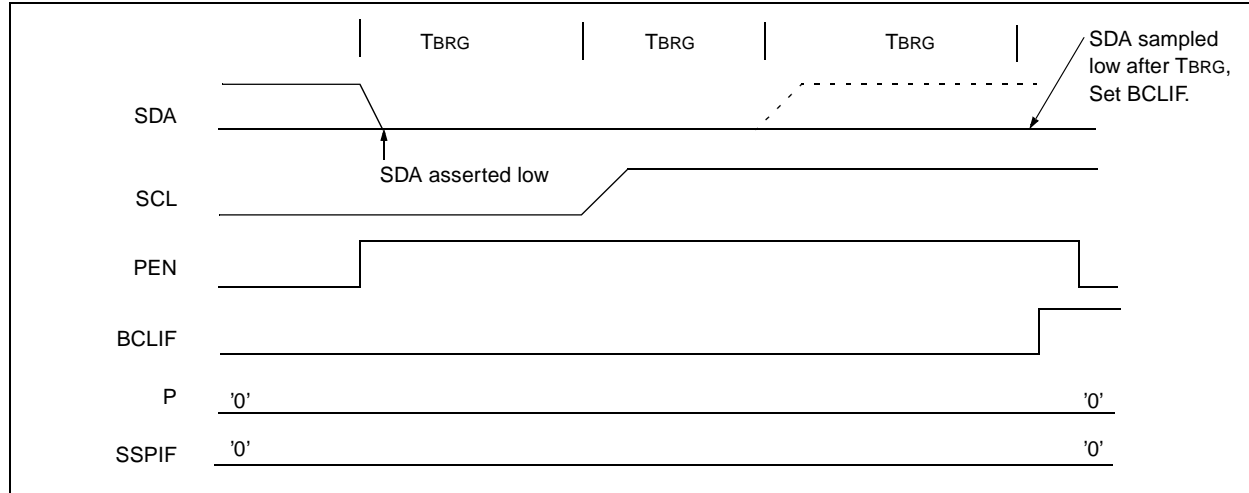
## 9.2.18.17 BUS COLLISION DURING A STOP CONDITION

Bus collision occurs during a STOP condition if:

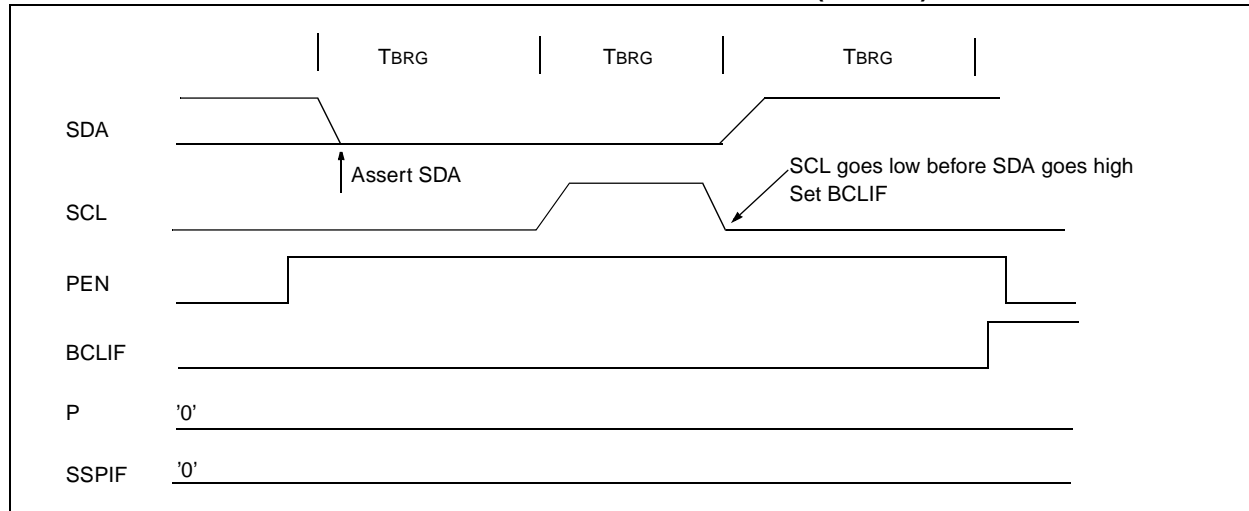
- After the SDA pin has been deasserted and allowed to float high, SDA is sampled low after the BRG has timed out.
- After the SCL pin is deasserted, SCL is sampled low before SDA goes high.

The STOP condition begins with SDA asserted low. When SDA is sampled low, the SCL pin is allowed to float. When the pin is sampled high (clock arbitration), the baud rate generator is loaded with SSPADD<6:0> and counts down to 0. After the BRG times out, SDA is sampled. If SDA is sampled low, a bus collision has occurred. This is due to another master attempting to drive a data '0'. If the SCL pin is sampled low before SDA is allowed to float high, a bus collision occurs. This is a case of another master attempting to drive a data '0' (Figure 9-25).

**FIGURE 9-25: BUS COLLISION DURING A STOP CONDITION (CASE 1)**



**FIGURE 9-26: BUS COLLISION DURING A STOP CONDITION (CASE 2)**



## 9.3 Connection Considerations for I<sup>2</sup>C Bus

For standard-mode I<sup>2</sup>C bus devices, the values of resistors  $R_p$  and  $R_s$  in Figure 9-27 depend on the following parameters:

- Supply voltage
- Bus capacitance
- Number of connected devices (input current + leakage current).

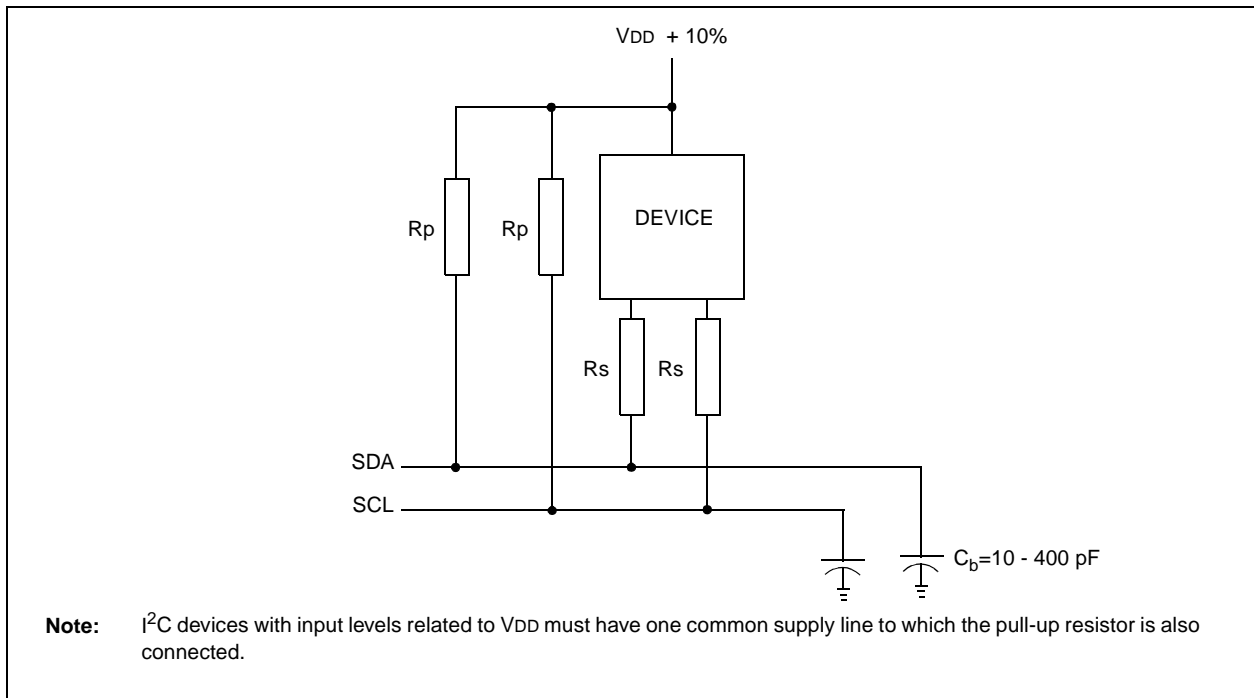
The supply voltage limits the minimum value of resistor  $R_p$  due to the specified minimum sink current of 3 mA at  $V_{OL\ max} = 0.4V$  for the specified output stages. For

example, with a supply voltage of  $V_{DD} = 5V \pm 10\%$  and  $V_{OL\ max} = 0.4V$  at 3 mA,  $R_{p\ min} = (5.5 - 0.4) / 0.003 = 1.7\ k\Omega$ .  $V_{DD}$  as a function of  $R_p$  is shown in Figure 9-27. The desired noise margin of  $0.1V_{DD}$  for the low level limits the maximum value of  $R_s$ . Series resistors are optional and used to improve ESD susceptibility.

The bus capacitance is the total capacitance of wire, connections, and pins. This capacitance limits the maximum value of  $R_p$  due to the specified rise time (Figure 9-27).

The SMP bit is the slew rate control enabled bit. This bit is in the SSPSTAT register, and controls the slew rate of the I/O pins when in I<sup>2</sup>C mode (master or slave).

**FIGURE 9-27: SAMPLE DEVICE CONFIGURATION FOR I<sup>2</sup>C BUS**



# PIC16F872

---

NOTES:

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

The Analog-to-Digital (A/D) Converter module has five inputs.

The analog input charges a sample and hold capacitor. The output of the sample and hold capacitor is the input into the converter. The converter then generates a digital result of this analog level via successive approximation. The A/D conversion of the analog input signal results in a corresponding 10-bit digital number. The A/D module has high and low voltage reference input that is software selectable to some combination of VDD, VSS, RA2 or RA3.

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 clock must be derived from the A/D's internal RC oscillator.

The A/D module has four registers. These registers are:

- A/D Result High Register (ADRESH)
- A/D Result Low Register (ADRESL)
- A/D Control Register0 (ADCON0)
- A/D Control Register1 (ADCON1)

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 the voltage reference) or as digital I/O.

Additional information on using the A/D module can be found in the PICmicro™ Mid-Range MCU Family Reference Manual (DS33023).

### REGISTER 10-1: ADCON0 REGISTER (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
							bit0
<p>bit7</p> <p>bit 7-6: <b>ADCS&lt;1:0&gt;</b>: A/D Conversion Clock Select bits            00 = Fosc/2            01 = Fosc/8            10 = Fosc/32            11 = FRC (clock derived from an RC oscillation)</p> <p>bit 5-3: <b>CHS&lt;2:0&gt;</b>: 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)</p> <p>bit 2: <b>GO/DONE</b>: 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)</p> <p>bit 1: <b>Unimplemented</b>: Read as '0'</p> <p>bit 0: <b>ADON</b>: A/D On bit            1 = A/D converter module is operating            0 = A/D converter module is shutoff and consumes no operating current</p>							

R = Readable bit  
 W = Writable bit  
 U = Unimplemented bit, read as '0'  
 - n = Value at POR reset

# PIC16F872

## REGISTER 10-2: ADCON1 REGISTER (ADDRESS 9Fh)

U-0	U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0

bit7

bit0

R = Readable bit  
W = Writable bit  
U = Unimplemented bit,  
read as '0'  
- n = Value at POR reset

bit 7: **ADFM:** A/D Result format select  
1 = Right Justified. 6 most significant bits of ADRESH are read as '0'.  
0 = Left Justified. 6 least significant bits of ADRESL are read as '0'.

bit 6-4: **Unimplemented:** Read as '0'

bit 3-0: **PCFG<3:0>:** A/D Port Configuration Control bits

PCFG<3:0>	AN4 RA5	AN3 RA3	AN2 RA2	AN1 RA1	AN0 RA0	VREF+	VREF-	CHAN / Refs <sup>(1)</sup>
0000	A	A	A	A	A	VDD	VSS	5/0
0001	A	VREF+	A	A	A	RA3	VSS	4/1
0010	A	A	A	A	A	VDD	VSS	5/0
0011	A	VREF+	A	A	A	RA3	VSS	4/1
0100	D	A	D	A	A	VDD	VSS	3/0
0101	D	VREF+	D	A	A	RA3	VSS	2/1
011x	D	D	D	D	D	VDD	VSS	0/0
1000	A	VREF+	VREF-	A	A	RA3	RA2	3/2
1001	A	A	A	A	A	VDD	VSS	5/0
1010	A	VREF+	A	A	A	RA3	VSS	4/1
1011	A	VREF+	VREF-	A	A	RA3	RA2	3/2
1100	A	VREF+	VREF-	A	A	RA3	RA2	3/2
1101	D	VREF+	VREF-	A	A	RA3	RA2	2/2
1110	D	D	D	D	A	VDD	VSS	1/0
1111	D	VREF+	VREF-	D	A	RA3	RA2	1/2

A = Analog input

D = Digital I/O

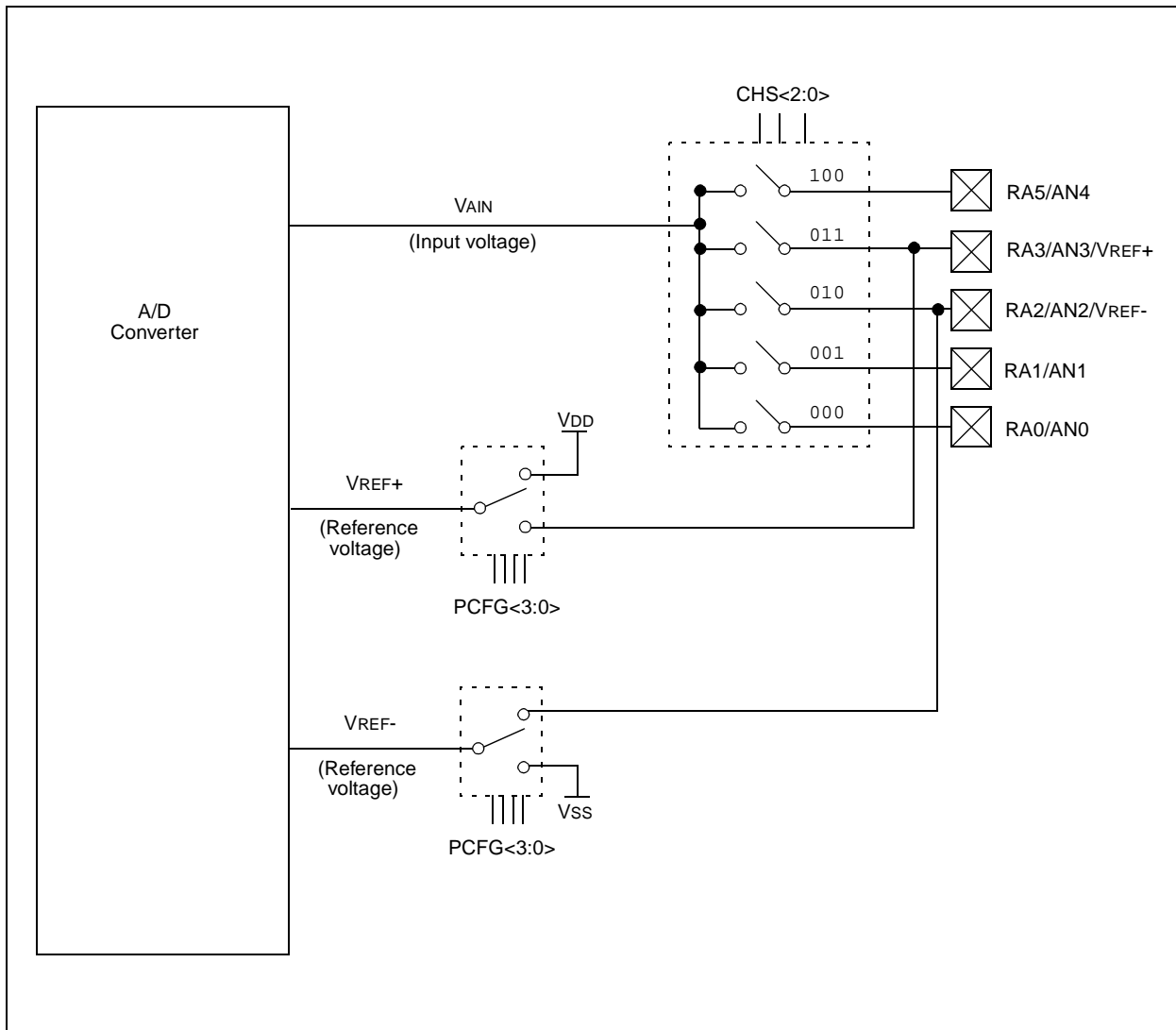
**Note 1:** This column indicates the number of analog channels available as A/D inputs and the number of analog channels used as voltage reference inputs.

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

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 inputs. To determine sample 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:

1. 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)
2. Configure A/D interrupt (if desired):
  - Clear ADIF bit
  - Set ADIE bit
  - Set GIE bit
3. Wait the required acquisition time.
4. Start conversion:
  - Set  $\overline{\text{GO/DONE}}$  bit (ADCON0)
5. Wait for A/D conversion to complete, by either:
  - Polling for the  $\overline{\text{GO/DONE}}$  bit to be clearedOR
  - Waiting for the A/D interrupt
6. Read A/D Result register pair (ADRESH:ADRESL), clear bit ADIF if required.
7. For next conversion, go to step 1 or step 2 as required. The A/D conversion time per bit is required. The A/D conversion time per bit is defined as TAD. A minimum wait of 2TAD is required before next acquisition starts.

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



## 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 ( $R_s$ ) and the internal sampling switch ( $R_{SS}$ ) impedance directly affect the time required to charge the capacitor CHOLD. The sampling switch ( $R_{SS}$ ) impedance varies over the device voltage ( $V_{DD}$ ), Figure 10-2. **The maximum recommended impedance for analog sources is 10 k $\Omega$ .** As the impedance is decreased, the acquisition time may be decreased. 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, Equation 10-1 may be used. This equation assumes that 1/2 LSb error is used (1024 steps for the A/D). The 1/2 LSb error is the maximum error allowed for the A/D to meet its specified resolution.

To calculate the minimum acquisition time,  $T_{ACQ}$ , see the PICmicro™ Mid-Range Reference Manual (DS33023).



## EQUATION 10-1: ACQUISITION TIME

$$\begin{aligned}
 T_{ACQ} &= \text{Amplifier Settling Time} + \\
 &\quad \text{Hold Capacitor Charging Time} + \\
 &\quad \text{Temperature Coefficient} \\
 &= T_{AMP} + T_C + T_{COFF} \\
 &= 2\mu\text{S} + T_C + [(\text{Temperature} - 25^\circ\text{C})(0.05\mu\text{S}/^\circ\text{C})] \\
 T_C &= \text{CHOLD} (\text{RIC} + \text{RSS} + \text{RS}) \ln(1/2047) \\
 &= -120\text{pF} (1\text{k}\Omega + 7\text{k}\Omega + 10\text{k}\Omega) \ln(0.0004885) \\
 &= 16.47\mu\text{S} \\
 T_{ACQ} &= 2\mu\text{S} + 16.47\mu\text{S} + [(50^\circ\text{C} - 25^\circ\text{C})(0.05\mu\text{S}/^\circ\text{C})] \\
 &= 19.72\mu\text{S}
 \end{aligned}$$

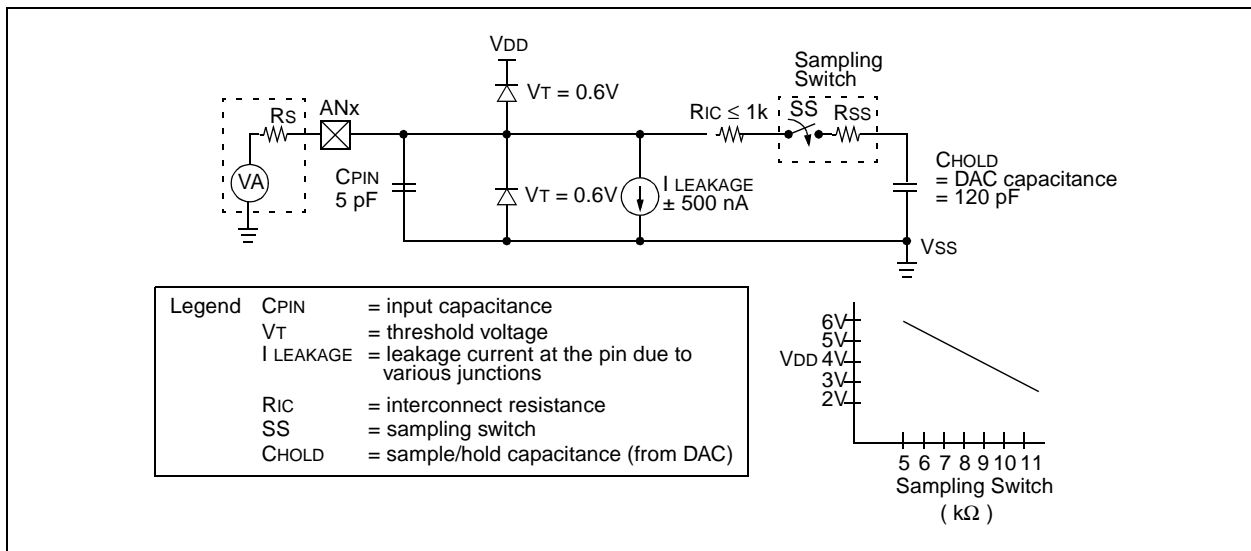
**Note 1:** The reference voltage ( $V_{REF}$ ) has no effect on the equation, since it cancels itself out.

**2:** The charge holding capacitor ( $\text{CHOLD}$ ) is not discharged after each conversion.

**3:** The maximum recommended impedance for analog sources is  $10\text{ k}\Omega$ . This is required to meet the pin leakage specification.

**4:** After a conversion has completed, a  $2.0T_{AD}$  delay must complete before acquisition can begin again. During this time, the holding capacitor is not connected to the selected A/D input channel.

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



## 10.2 Selecting the A/D Conversion Clock

The A/D conversion time per bit is defined as TAD. The A/D conversion requires a minimum 12TAD per 10-bit conversion. The source of the A/D conversion clock is software selected. The four possible options for TAD are:

- 2Tosc
- 8Tosc
- 32Tosc
- Internal RC oscillator

For correct A/D conversions, the A/D conversion clock (TAD) must be selected to ensure a minimum TAD time of 1.6  $\mu$ s.

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

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

AD Clock Source (TAD)		Maximum Device Frequency
Operation	ADCS<1:0>	Max.
2Tosc	00	1.25 MHz
8Tosc	01	5 MHz
32Tosc	10	20 MHz
RC <sup>(1, 2, 3)</sup>	11	Note 1

- Note 1:** The RC source has a typical TAD time of 4  $\mu$ s but can vary between 2-6  $\mu$ s.  
**2:** When the device frequencies are greater than 1 MHz, the RC A/D conversion clock source is only recommended for sleep operation.  
**3:** For extended voltage devices (LC), please refer to the Electrical Specifications section.

## 10.3 Configuring Analog Port Pins

The ADCON1, and TRIS 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, any pin configured as an analog input channel 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 AN<4:0> pins), may cause the input buffer to consume current that is out of the device specifications.

## 10.4 A/D Conversions

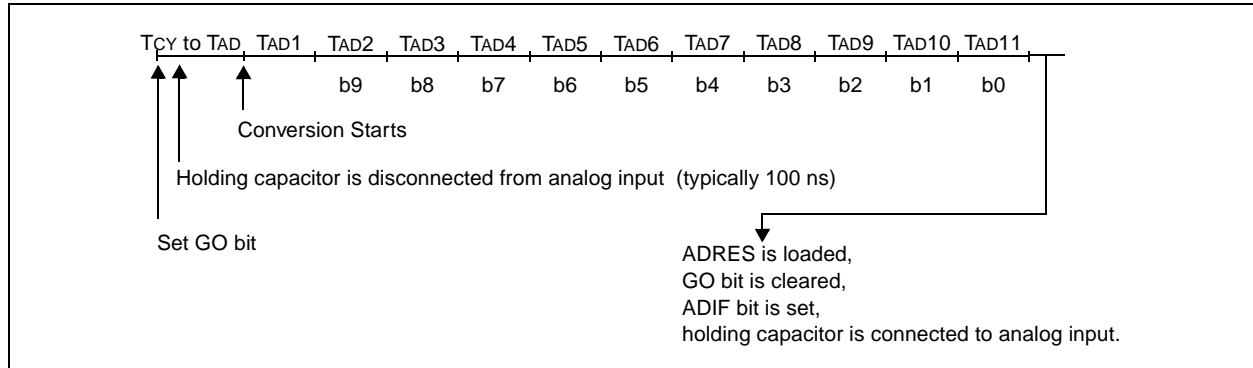
Clearing the  $\overline{\text{GO/DONE}}$  bit during a conversion will abort the current conversion. The A/D result register pair will NOT be updated with the partially completed A/D conversion sample. That is, the ADRESH:ADRESL registers will continue to contain the value of the last completed conversion (or the last value written to the ADRESH:ADRESL registers). After the A/D conversion is aborted, a 2TAD wait is

required before the next acquisition is started. After this 2TAD wait, acquisition on the selected channel is automatically started.

In Figure 10-3, after the GO bit is set, the first time segment has a minimum of TCY and a maximum of TAD.

**Note:** The  $\overline{\text{GO/DONE}}$  bit should **NOT** be set in the same instruction that turns on the A/D.

**FIGURE 10-3: A/D CONVERSION TAD CYCLES**



# PIC16F872

## 10.4.1 A/D RESULT REGISTERS

The ADRESH:ADRESL register pair is the location where the 10-bit A/D result is loaded at the completion of the A/D conversion. This register pair is 16-bits wide. The A/D module gives the flexibility to left or right justify the 10-bit result in the 16-bit result register. The A/D Format Select bit (ADFM) controls this justification. Figure 10-4 shows the operation of the A/D result justification. The extra bits are loaded with '0's'. When an A/D result will not overwrite these locations (A/D disable), these registers may be used as two general purpose 8-bit registers.

## 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 (ADCS<1:0> = 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.

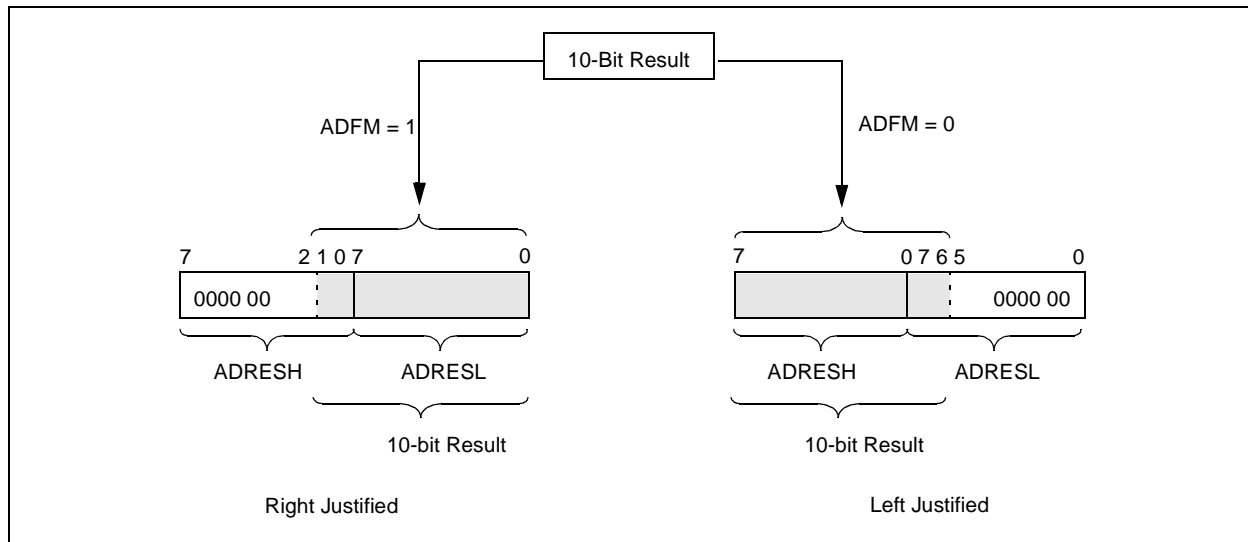
**Note:** For the A/D module to operate in SLEEP, the A/D clock source must be set to RC (ADCS<1:0> = 11). To allow the conversion to occur during SLEEP, ensure the SLEEP instruction immediately follows the instruction that sets the GO/DONE bit.

## 10.6 Effects of a Reset

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

The value that is in the ADRESH:ADRESL registers is not modified for a Power-on Reset. The ADRESH:ADRESL registers will contain unknown data after a Power-on Reset.

FIGURE 10-4: A/D RESULT JUSTIFICATION



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

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR, BOR	MCLR, WDT
0Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	(1)	ADIF	(1)	(1)	SSPIF	CCP1IF	TMR2IF	TMR1IF	r0rr 0000	r0rr 0000
8Ch	PIE1	(1)	ADIE	(1)	(1)	SSPIE	CCP1IE	TMR2IE	TMR1IE	r0rr 0000	r0rr 0000
1Eh	ADRESH	A/D Result Register High Byte								xxxx xxxx	uuuu uuuu
9Eh	ADRESL	A/D Result Register Low Byte								xxxx xxxx	uuuu uuuu
1Fh	ADCON0	ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	—	ADON	0000 00-0	0000 00-0
9Fh	ADCON1	ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0	--0- 0000	--0- 0000
85h	TRISA	—	—	PORTA Data Direction Register						--11 1111	--11 1111
05h	PORTA	—	—	PORTA Data Latch when written: PORTA pins when read						--0x 0000	--0u 0000

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

**Note 1:** These bits are reserved; always maintain these bits clear.

# PIC16F872

---

NOTES:

## 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. These are:

- OSC 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
- Low Voltage In-Circuit Serial Programming
- In-Circuit Debugger

These devices have a Watchdog Timer, which can be shut off only through configuration bits. 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. 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. A set of configuration bits are used to select various options.

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. In fact, it belongs to the special test/configuration memory space (2000h through 3FFFh), which can be accessed only during programming.

# PIC16F872

## REGISTER 11-1: CONFIGURATION WORD

CP1	CP0	DEBUG	—	WRT	CPD	LVP	BODEN	CP1	CP0	PWRTE	WDTE	F0SC1	F0SC0
-----	-----	-------	---	-----	-----	-----	-------	-----	-----	-------	------	-------	-------

bit13

bit0

Register: CONFIG  
Address 2007h

bit 13-12:

bit 5-4: **CP<1:0>**: Flash Program Memory Code Protection bits <sup>(2)</sup>

- 11 = Code protection off
- 10 = 0000h to 06FFh code protected
- 01 = 0000h to 03FFh code protected
- 00 = 0000h to 07FFh code protected

bit 11: **DEBUG**: In-Circuit Debugger Mode

- 1 = In-Circuit Debugger disabled, RB6 and RB7 are general purpose I/O pins.
- 0 = In-Circuit Debugger enabled, RB6 and RB7 are dedicated to the debugger.

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

bit 9: **WRT**: Flash Program Memory Write Enable

- 1 = Unprotected program memory may be written to by EECON control
- 0 = Unprotected program memory may not be written to by EECON control

bit 8: **CPD**: Data EE Memory Code Protection

- 1 = Code protection off
- 0 = Data EEPROM memory code protected

bit 7: **LVP**: Low Voltage In-Circuit Serial Programming Enable bit

- 1 = RB3/PGM pin has PGM function, low voltage programming enabled
- 0 = RB3 is digital I/O, HV on  $\overline{MCLR}$  must be used for programming

bit 6: **BODEN**: Brown-out Reset Enable bit <sup>(1)</sup>

- 1 = BOR enabled
- 0 = BOR disabled

bit 3: **PWRTE**: Power-up Timer Enable bit <sup>(1)</sup>

- 1 = PWRT disabled
- 0 = PWRT enabled

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

**2:** All of the CP<1:0> pairs have to be given the same value to enable the code protection scheme listed.



## 11.2 Oscillator Configurations

### 11.2.1 OSCILLATOR TYPES

The PIC16F872 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/CLKIN and OSC2/CLKOUT pins to establish oscillation (Figure 11-1). The PIC16F872 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 XT, LP or HS modes, the device can have an external clock source to drive the OSC1/CLKIN pin (Figure 11-2).

TABLE 11-1: CERAMIC RESONATORS

Ranges Tested:			
Mode	Freq	OSC1	OSC2
XT	455 kHz	68 - 100 pF	68 - 100 pF
	2.0 MHz	15 - 68 pF	15 - 68 pF
	4.0 MHz	15 - 68 pF	15 - 68 pF
HS	8.0 MHz	10 - 68 pF	10 - 68 pF
	16.0 MHz	10 - 22 pF	10 - 22 pF
These values are for design guidance only. See notes at bottom of page.			
Resonators Used:			
455 kHz	Panasonic EFO-A455K04B	± 0.3%	
2.0 MHz	Murata Erie CSA2.00MG	± 0.5%	
4.0 MHz	Murata Erie CSA4.00MG	± 0.5%	
8.0 MHz	Murata Erie CSA8.00MT	± 0.5%	
16.0 MHz	Murata Erie CSA16.00MX	± 0.5%	
All resonators used did not have built-in capacitors.			

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

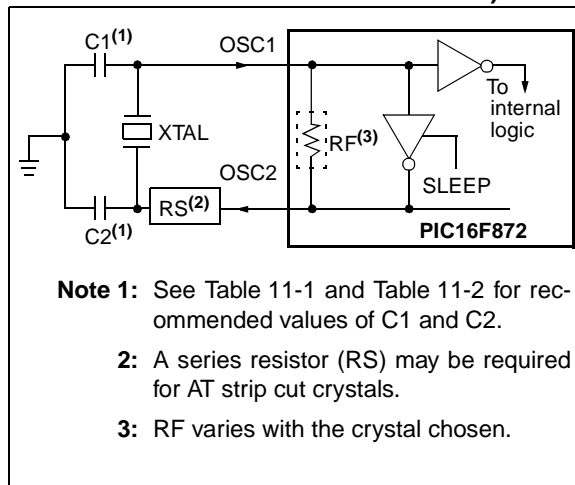
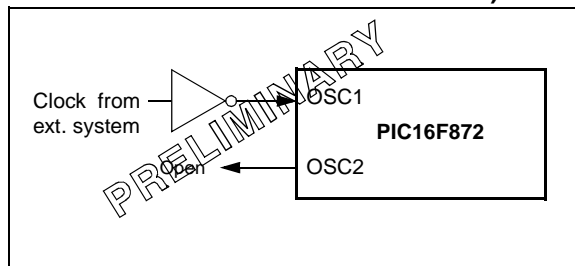


FIGURE 11-2: EXTERNAL CLOCK INPUT OPERATION (HS, XT OR LP OSC CONFIGURATION)



# PIC16F872

**TABLE 11-2: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR**

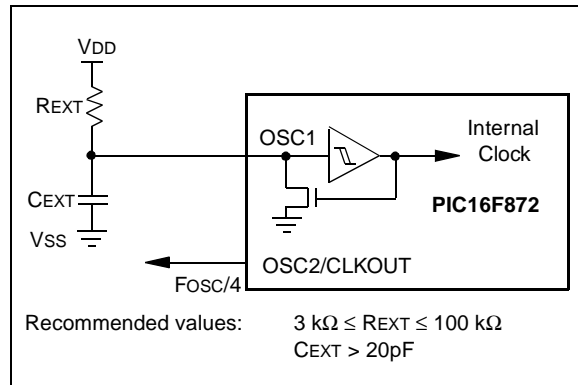
Osc Type	Crystal Freq	Cap. Range C1	Cap. Range C2
LP	32 kHz	33 pF	33 pF
	200 kHz	15 pF	15 pF
XT	200 kHz	47-68 pF	47-68 pF
	1 MHz	15 pF	15 pF
	4 MHz	15 pF	15 pF
HS	4 MHz	15 pF	15 pF
	8 MHz	15-33 pF	15-33 pF
	20 MHz	15-33 pF	15-33 pF
These values are for design guidance only. See notes at bottom of page.			
<b>Crystals Used</b>			
32 kHz	Epson C-001R32.768K-A	± 20 PPM	
200 kHz	STD XTL 200.000KHz	± 20 PPM	
1 MHz	ECS ECS-10-13-1	± 50 PPM	
4 MHz	ECS ECS-40-20-1	± 50 PPM	
8 MHz	EPSON CA-301 8.000M-C	± 30 PPM	
20 MHz	EPSON CA-301 20.000M-C	± 30 PPM	

- Note 1:** Higher capacitance increases the stability of oscillator but also increases the start-up time.
- 2:** Since each resonator/crystal has its own characteristics, the user should consult the resonator/crystal manufacturer for appropriate values of external components.
- 3:** Rs may be required in HS mode, as well as XT mode, to avoid overdriving crystals with low drive level specification.
- 4:** When migrating from other PICmicro devices, oscillator performance should be verified.

## 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 (R<sub>EXT</sub>) and capacitor (C<sub>EXT</sub>) values, and the operating temperature. In addition to this, the oscillator frequency will vary from unit to unit due to normal process parameter variation. Furthermore, the difference in lead frame capacitance between package types will also affect the oscillation frequency, especially for low C<sub>EXT</sub> 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 PIC16F872.

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



## 11.3 Reset

The PIC16F872 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

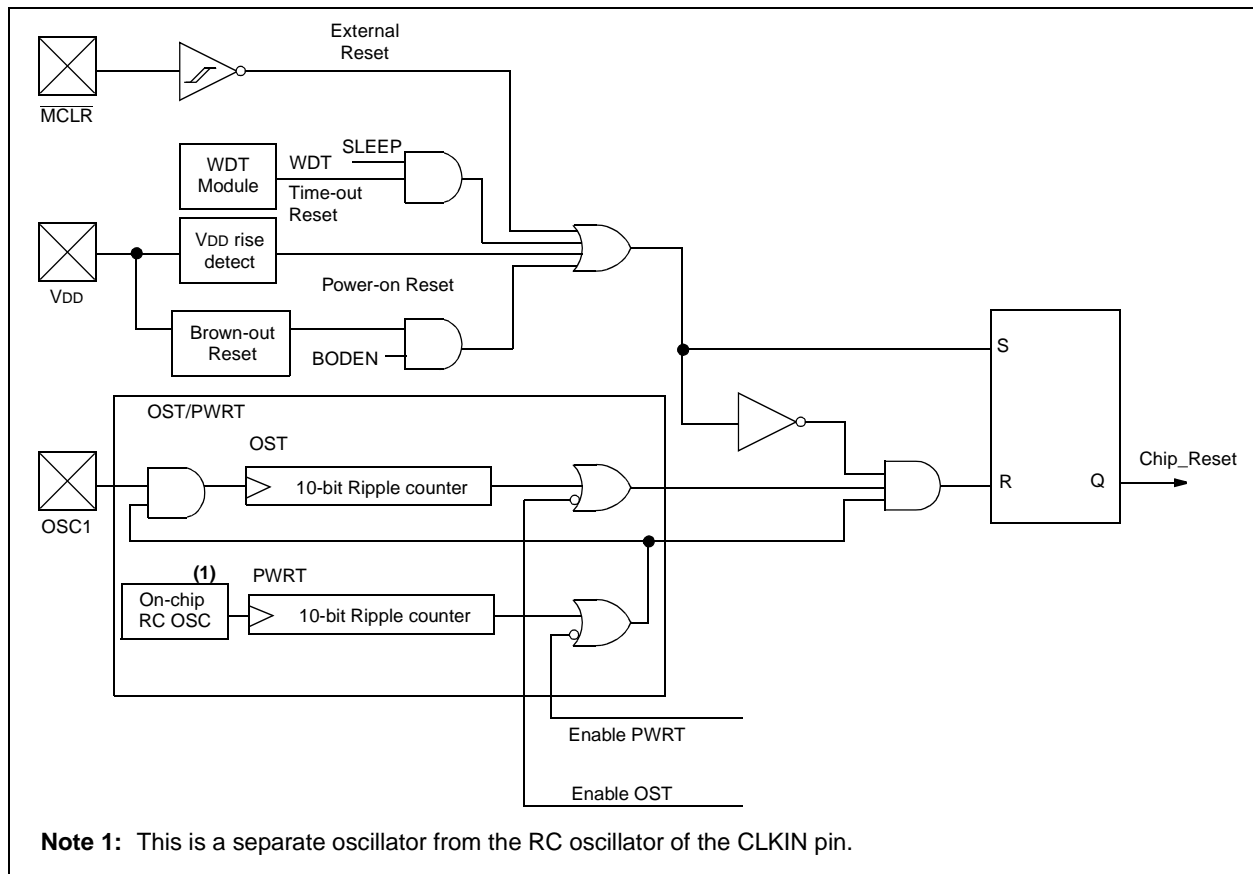
Brown-out Reset (BOR). They are not affected by a WDT Wake-up, which is viewed as the resumption of normal operation. The  $\overline{\text{TO}}$  and  $\overline{\text{PD}}$  bits are set or cleared differently in different reset situations as indicated in Table 11-4. These bits are used in software 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.

These devices have a  $\overline{\text{MCLR}}$  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.

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



## 11.4 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 directly (or through a resistor) to VDD. This will eliminate external RC components usually needed to create a Power-on Reset. A maximum rise time for VDD is specified. See Electrical Specifications 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. Brown-out Reset may be used to meet the start-up conditions. For additional information, refer to Application Note, AN007, "Power-up Trouble Shooting", (DS00007).

## 11.5 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).

## 11.6 Oscillator Start-up Timer (OST)

The Oscillator Start-up Timer (OST) provides 1024 oscillator cycle (from OSC1 input) delay after the PWRT delay is over. This ensures 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.

## 11.7 Brown-Out Reset (BOR)

The configuration bit, BODEN, 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 $\mu$ 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 72mS). 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.

## 11.8 Time-out Sequence

On power-up, the Time-out Sequence is as follows: The PWRT delay starts (if enabled) when a POR reset 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, the time-outs will expire. Bringing  $\overline{\text{MCLR}}$  high will begin execution immediately. This is useful for testing purposes or to synchronize more than one PIC16CXX 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.

## 11.9 Power Control/Status Register (PCON)

The Power Control/Status Register, PCON, has up to two bits depending upon the device.

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 BOR occurred. The  $\overline{\text{BOR}}$  bit is a "don't care" bit and is not necessarily predictable if the Brown-out Reset circuitry is disabled (by clearing bit BODEN in the Configuration Word).

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
	PWRTE = 0	PWRTE = 1		
XT, HS, LP	72 ms + 1024Tosc	1024Tosc	72 ms + 1024Tosc	1024Tosc
RC	72 ms	—	72 ms	—

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

POR	BOR	TO	PD	
0	x	1	1	Power-on Reset
0	x	0	x	Illegal, $\overline{TO}$ is set on $\overline{POR}$
0	x	x	0	Illegal, $\overline{PD}$ is set on $\overline{POR}$
1	0	1	1	Brown-out Reset
1	1	0	1	WDT Reset
1	1	0	0	WDT Wake-up
1	1	u	u	$\overline{MCLR}$ Reset during normal operation
1	1	1	0	$\overline{MCLR}$ 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
$\overline{MCLR}$ Reset during normal operation	000h	000u uuuu	---- --uu
$\overline{MCLR}$ 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).

# PIC16F872

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

Register	Power-on Reset, Brown-out Reset	MCLR Resets 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	r0rr 0000	r0rr 0000	rurr uuuu <sup>(1)</sup>
PIR2	-r-0 0--r	-r-0 0--r	-r-u u--r <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
ADRESH	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADCON0	0000 00-0	0000 00-0	uuuu uu-u
OPTION_REG	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	r0rr 0000	r0rr 0000	rurr uuuu
PIE2	-r-0 0--r	-r-0 0--r	-r-u u--r
PCON	---- -qg	---- --uu	---- --uu
PR2	1111 1111	1111 1111	1111 1111
SSPADD	0000 0000	0000 0000	uuuu uuuu
SSPSTAT	--00 0000	--00 0000	--uu uuuu
ADRESL	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADCON1	0--- 0000	0--- 0000	u--- uuuu
EEDATA	0--- 0000	0--- 0000	u--- uuuu
EEADR	xxxx xxxx	uuuu uuuu	uuuu uuuu
EEDATH	xxxx xxxx	uuuu uuuu	uuuu uuuu
EEADRH	xxxx xxxx	uuuu uuuu	uuuu uuuu
EECON1	x--- x000	u--- u000	u--- uuuu
EECON2	---- ----	---- ----	---- ----

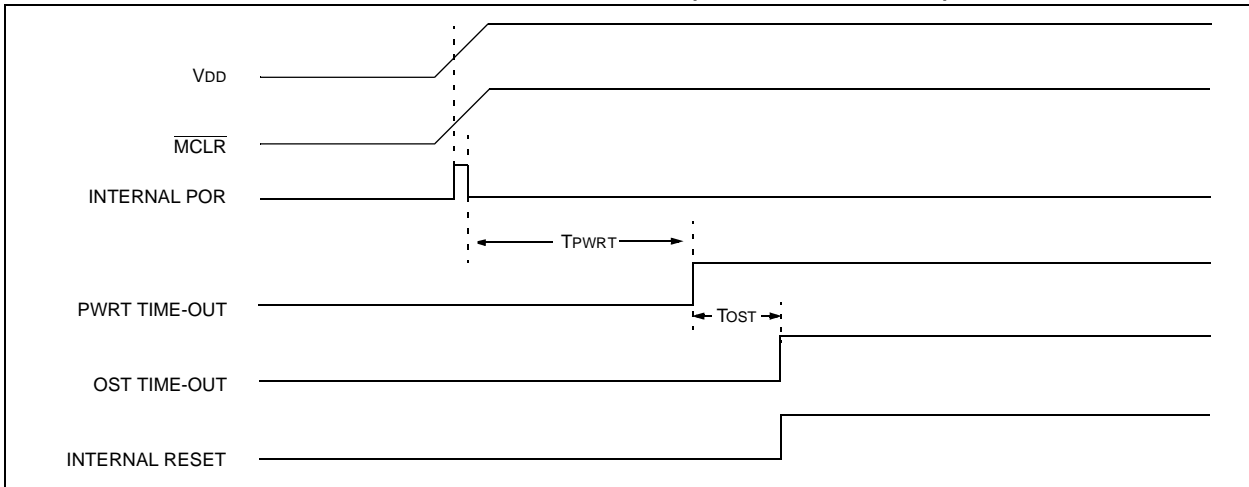
Legend: u = unchanged, x = unknown, r = reserved, - = unimplemented bit, read as '0', q = value depends on condition, r = reserved maintain clear.

**Note 1:** One or more bits in INTCON, PIR1 and/or PIR2 will be affected (to cause wake-up).

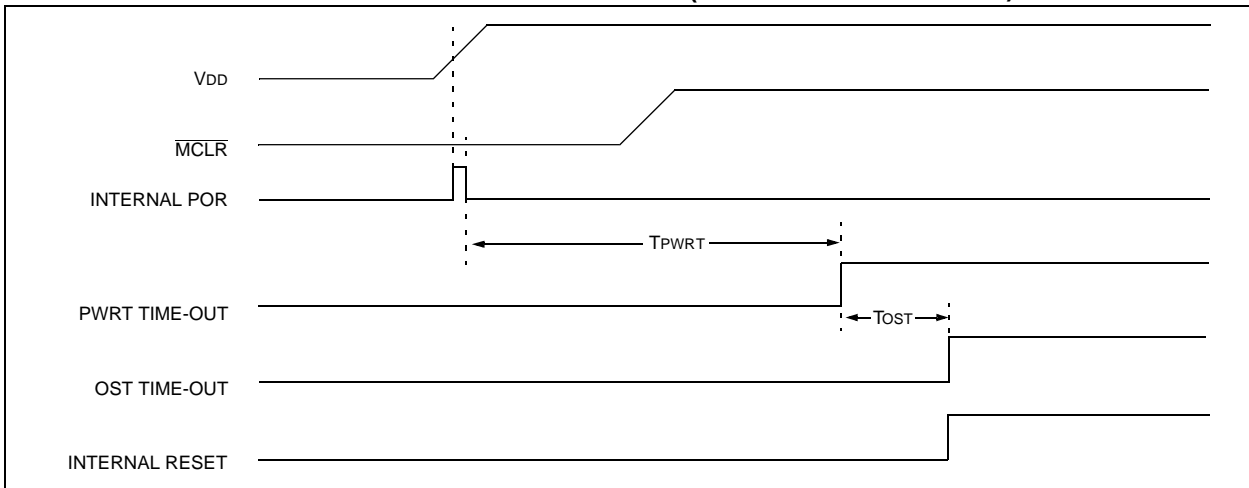
**2:** When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).

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

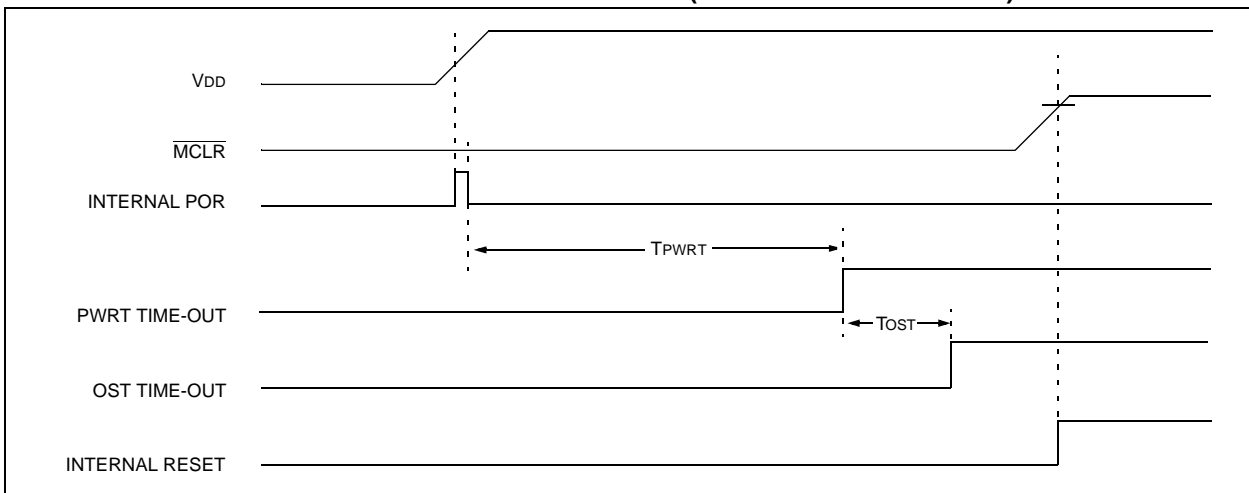
**FIGURE 11-5: TIME-OUT SEQUENCE ON POWER-UP ( $\overline{\text{MCLR}}$  TIED TO  $V_{\text{DD}}$ )**



**FIGURE 11-6: TIME-OUT SEQUENCE ON POWER-UP ( $\overline{\text{MCLR}}$  NOT TIED TO  $V_{\text{DD}}$ ): CASE 1**

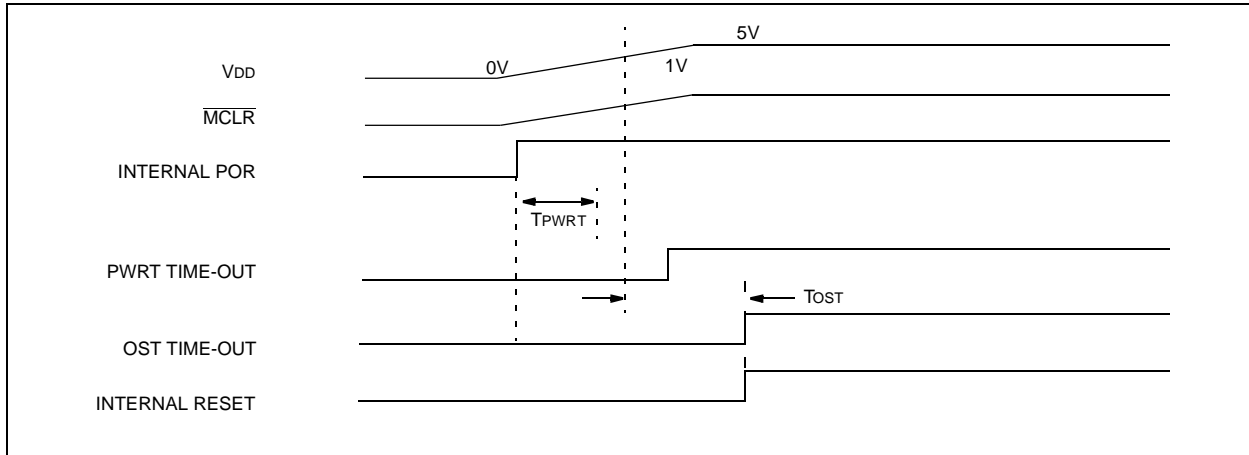


**FIGURE 11-7: TIME-OUT SEQUENCE ON POWER-UP ( $\overline{\text{MCLR}}$  NOT TIED TO  $V_{\text{DD}}$ ): CASE 2**



# PIC16F872

FIGURE 11-8: SLOW RISE TIME ( $\overline{\text{MCLR}}$  TIED TO  $V_{\text{DD}}$ )





## 11.10 Interrupts

The PIC16F872 has 10 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 un-masked 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.

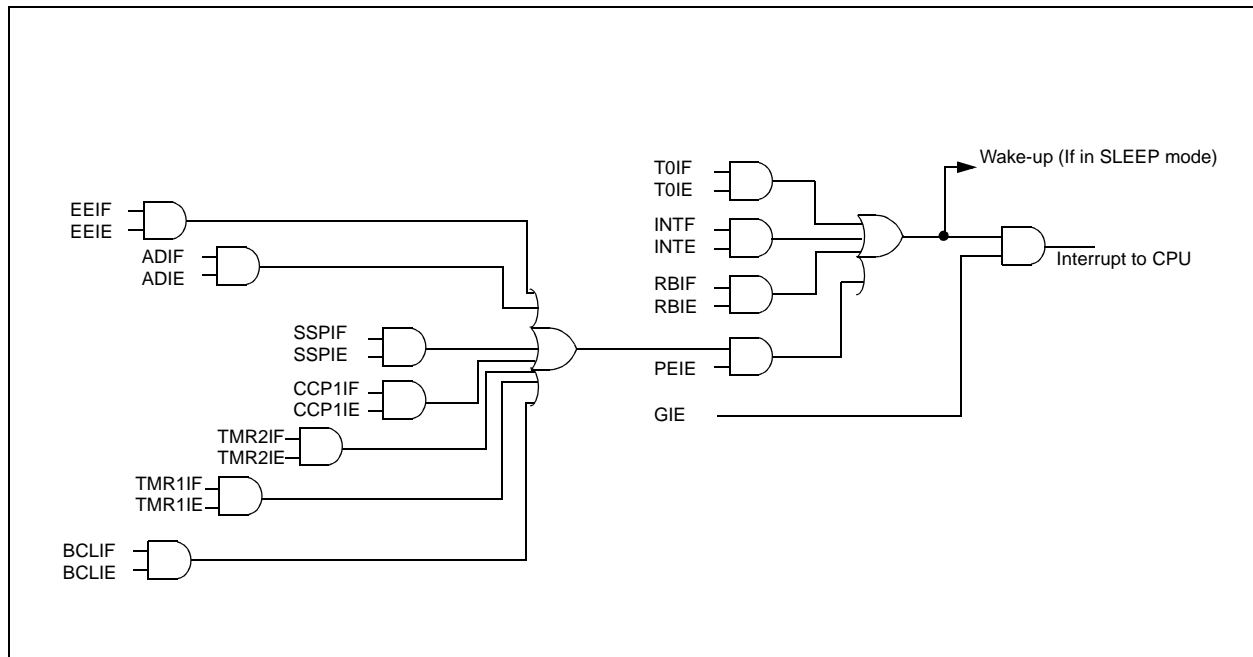
The RB0/INT pin interrupt, the RB port change interrupt and the TMR0 overflow interrupt flags are contained in the INTCON register.

The peripheral interrupt flags are contained in the Special Function Registers, PIR1 and PIR2. The corresponding interrupt enable bits are contained in Special Function Registers, PIE1 and PIE2, and the peripheral interrupt enable bit is contained in Special Function Register INTCON.

When an interrupt is responded to, 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. 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 or the GIE bit

**FIGURE 11-9: INTERRUPT LOGIC**



## 11.10.1 INT INTERRUPT

External interrupt on the RB0/INT pin is edge triggered, either rising, if bit INTEDG (OPTION\_REG<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 11.13 for details on SLEEP mode.

## 11.10.2 TMR0 INTERRUPT

An overflow (FFh → 00h) in the TMR0 register will set flag bit T0IF (INTCON<2>). The interrupt can be enabled/disabled by setting/clearing enable bit T0IE (INTCON<5>) (Section 5.0).

## 11.10.3 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>) (Section 3.2).

## 11.11 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 register and STATUS register). This will have to be implemented in software.

Since the upper 16 bytes of each bank are common in the PIC16F872 devices, temporary holding registers W\_TEMP, STATUS\_TEMP and PCLATH\_TEMP should be placed in here. These 16 locations don't require banking and therefore, make it easier for context save and restore. Example 11-1 can be used to save and restore context for interrupts.

### 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
MOVF     PCLATH, W        ;Only required if using pages 1, 2 and/or 3
MOVWF    PCLATH_TEMP      ;Save PCLATH into W
CLRF     PCLATH           ;Page zero, regardless of current page
:
:(ISR)
:
MOVF     PCLATH_TEMP, W   ;Restore PCLATH
MOVWF    PCLATH           ;Move W into PCLATH
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
```

## 11.12 Watchdog Timer (WDT)

The Watchdog Timer is as a free running on-chip RC oscillator which does not require any external components. This RC oscillator is separate from the RC oscillator of the OSC1/CLKIN pin. That means that the WDT will run, even if the clock on the OSC1/CLKIN and OSC2/CLKOUT pins 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.

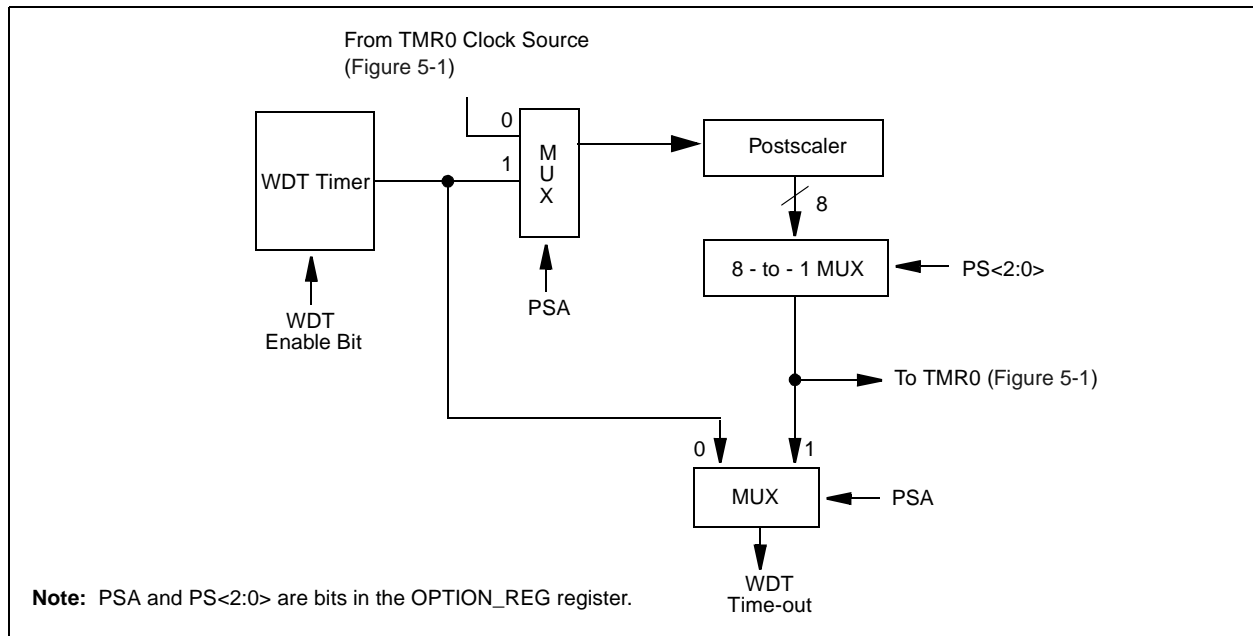
The WDT can be permanently disabled by clearing configuration bit WDTE (Section 11.1).

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\_REG register.

**Note:** 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.

**Note:** 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 11-10: WATCHDOG TIMER BLOCK DIAGRAM**



**FIGURE 11-11: 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)	BODEN <sup>(1)</sup>	CP1	CP0	PWRTE <sup>(1)</sup>	WDTE	FOSC1	FOSC0
81h,181h	OPTION_REG	$\overline{RBPU}$	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.

## 11.13 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 `T0CKI` input should also be at  $V_{DD}$  or  $V_{SS}$  for lowest current consumption. The contribution from on-chip pull-ups on `PORTB` should be considered.

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

### 11.13.1 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 some Peripheral Interrupts.

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 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`).
7. `EEPROM` write operation completion.

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 is 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.

### 11.13.2 WAKE-UP USING INTERRUPTS

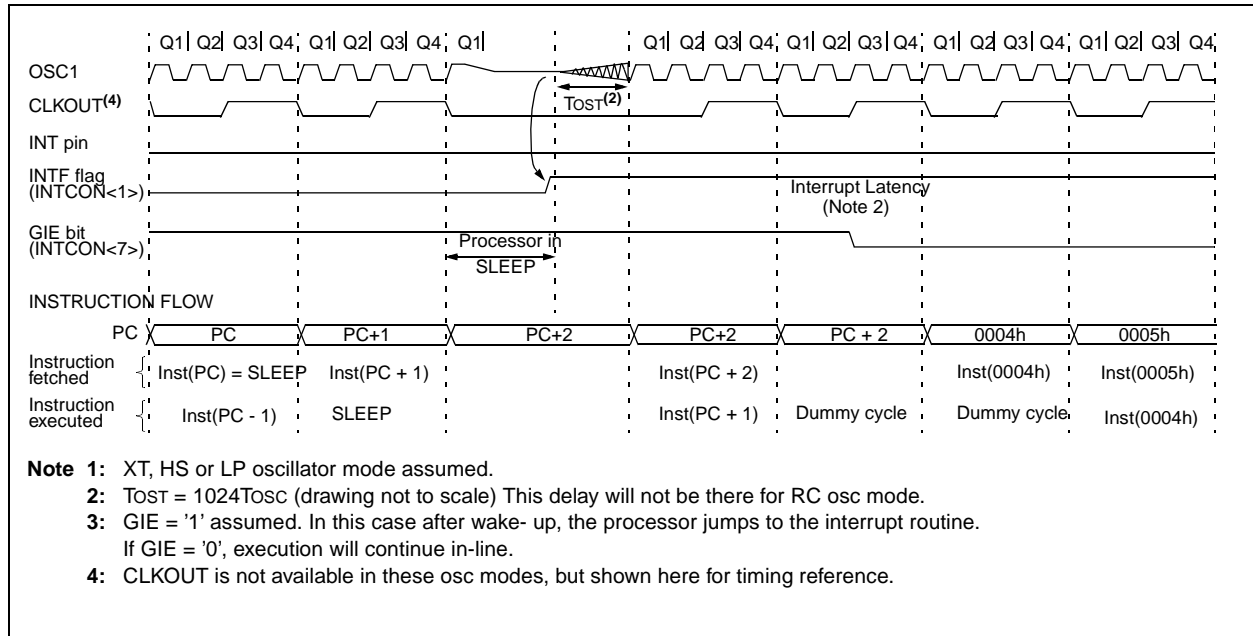
When global interrupts are disabled (`GIE` cleared) and any interrupt source has both its interrupt enable bit and interrupt flag bit set, one of the following will occur:

- 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.

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



## 11.14 In-Circuit Debugger

When the DEBUG bit in the configuration word is programmed to a '0', the In-Circuit Debugger functionality is enabled. This function allows simple debugging functions when used with MPLAB<sup>®</sup>. When the microcontroller has this feature enabled, some of the resources are not available for general use. Table 11-7 shows which features are consumed by the background debugger.

**TABLE 11-7: DEBUGGER RESOURCES**

I/O pins	RB6, RB7
Stack	1 level
Program Memory	Address 0000h must be NOP
	Last 100h words
Data Memory	0x070(0x0F0, 0x170, 0x1F0) 0x1EB - 0x1EF

To use the In-Circuit Debugger function of the microcontroller, the design must implement In-Circuit Serial Programming connections to MCLR/VPP, VDD, GND, RB7 and RB6. This will interface to the In-Circuit Debugger module available from Microchip or one of the third party development tool companies.

## 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 4 least significant bits of the ID location are used.

## 11.17 In-Circuit Serial Programming

The PIC16F872 microcontroller can be serially programmed while in the end application circuit. This is simply done with two lines for clock and data and three other lines for power, ground, and the programming voltage. This allows customers to manufacture boards with unprogrammed devices, and then program the microcontroller just before shipping the product. This also allows the most recent firmware or a custom firmware to be programmed.

When using ICSP, the part must be supplied 4.5V to 5.5V if a bulk erase will be executed. This includes reprogramming of the code protect both from an on-state to off-state. For all other cases of ICSP, the part may be programmed at the normal operating voltages. This means calibration values, unique user IDs or user code can be reprogrammed or added.

For complete details of serial programming, please refer to the In-Circuit Serial Programming (ICSP™) Guide, (DS30277).

## 11.18 Low Voltage ICSP Programming

The LVP bit of the configuration word enables low voltage ICSP programming. This mode allows the microcontroller to be programmed via ICSP using a VDD source in the operating voltage range. This only means that VPP does not have to be brought to VIH, but can instead be left at the normal operating voltage. In this mode, the RB3/PGM pin is dedicated to the programming function and ceases to be a general purpose I/O pin. During programming, VDD is applied to the MCLR pin. To enter programming mode, VDD must be applied to the RB3/PGM provided the LVP bit is set. The LVP bit defaults to on ('1') from the factory.

- |   |
|---|
| <p><b>Note 1:</b> The high voltage programming mode is always available, regardless of the state of the LVP bit, by applying VIH to the MCLR pin.</p> <p><b>2:</b> While in low voltage ICSP mode, the RB3 pin can no longer be used as a general purpose I/O pin.</p> <p><b>3:</b> When using low voltage ICSP programming (LVP) and the pull-ups on PORTB are enabled, bit 3 in the TRISB register must be cleared to disable the pull-up on RB3 and ensure the proper operation of the device.</p> |
|---|

If low-voltage programming mode is not used, the LVP bit can be programmed to a '0' and RB3/PGM becomes a digital I/O pin. However, the LVP bit may only be programmed when programming is entered with VIH on MCLR. The LVP bit can only be changed when using high voltage on MCLR.

It should be noted, that once the LVP bit is programmed to 0, only the high voltage programming mode is available and only high voltage programming mode can be used to program the device.

When using low voltage ICSP, the part must be supplied 4.5V to 5.5V if a bulk erase will be executed. This includes reprogramming of the code protect bits from an on-state to off-state. For all other cases of low voltage ICSP, the part may be programmed at the normal operating voltage. This means calibration values, unique user IDs or user code can be reprogrammed or added.

## 12.0 INSTRUCTION SET SUMMARY

Each PIC16CXX instruction is a 14-bit word divided into an OPCODE which specifies the instruction type and one or more operands which further specify the operation of the instruction. The PIC16CXX 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 NOP. 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.

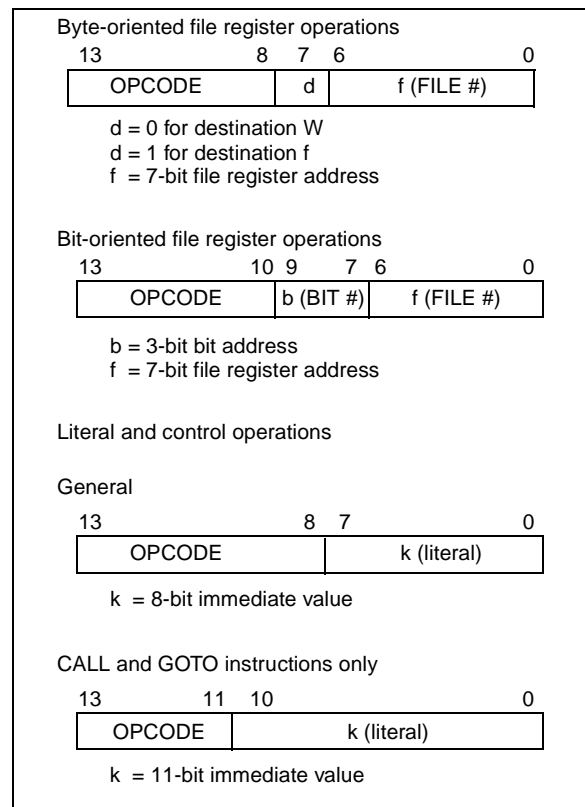
**Note:** To maintain upward compatibility with future PIC16CXX products, do not use the OPTION and TRIS instructions.

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 Reference Manual, (DS33023).

# PIC16F872

**TABLE 12-2: PIC16CXXX 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 lfff 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 lfff 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		
CLRWDI	-	Clear Watchdog Timer	1	00	0000 0110 0100	$\overline{TO}, \overline{PD}$	
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	$\overline{TO}, \overline{PD}$	
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]* 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]* ANDWF    *f,d*

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

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

**Status Affected:**    Z

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

### **ADDWF**      **Add W and f**

**Syntax:**      *[label]* 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' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.

### **BCF**          **Bit Clear f**

**Syntax:**      *[label]* BCF      *f,b*

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

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

**Status Affected:**    None

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

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

**Syntax:**      *[label]* ANDLW    *k*

**Operands:**     $0 \leq k \leq 255$

**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:**      *[label]* 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.

# PIC16F872

<b>BTFS</b>	<b>Bit Test f, Skip if Set</b>
Syntax:	[ <i>label</i> ] BTFS 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' is '0', the next instruction is executed. If bit 'b' is '1', then the next instruction is discarded and a NOP is executed instead making this a 2TCY instruction.

<b>BTFS</b>	<b>Bit Test, Skip if Clear</b>
Syntax:	[ <i>label</i> ] BTFS 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' is '1', the next instruction is executed. If bit 'b', in register 'f', is '0', the next instruction is discarded, and a NOP is executed instead, making this a 2TCY instruction.

<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>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>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>CLRWD</b>	<b>Clear Watchdog Timer</b>
Syntax:	[ <i>label</i> ] CLRWD
Operands:	None
Operation:	00h → WDT 0 → WDT prescaler, 1 → $\overline{TO}$ 1 → $\overline{PD}$
Status Affected:	$\overline{TO}$ , $\overline{PD}$
Description:	CLRWD 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 ≤ f ≤ 127 d ∈ [0,1]
Operation:	( $\bar{f}$ ) → (destination)
Status Affected:	Z
Description:	The contents of register 'f' are complemented. If 'd' is 0, the result is stored in W. If 'd' is 1, the result is stored back in register 'f'.

<b>GOTO</b>	<b>Unconditional Branch</b>
Syntax:	[ <i>label</i> ] GOTO k
Operands:	0 ≤ k ≤ 2047
Operation:	k → PC<10:0> PCLATH<4:3> → 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 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(f) - 1 → (destination)
Status Affected:	Z
Description:	Decrement register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.

<b>INCF</b>	<b>Increment f</b>
Syntax:	[ <i>label</i> ] INCF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(f) + 1 → (destination)
Status Affected:	Z
Description:	The contents of register 'f' are incremented. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'.

<b>DECFSZ</b>	<b>Decrement f, Skip if 0</b>
Syntax:	[ <i>label</i> ] DECFSZ f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(f) - 1 → (destination); skip if result = 0
Status Affected:	None
Description:	The contents of register 'f' are decremented. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'. If the result is 1, the next instruction is executed. If the result is 0, then a NOP is executed instead making it a 2TCY instruction.

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

# PIC16F872

---

---

**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'ed with the eight bit literal 'k'. The result is placed in the W register.

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

---

Syntax:            [ *label* ] IORWF f,d  
Operands:         $0 \leq f \leq 127$   
                       $d \in [0,1]$   
Operation:        (W) .OR. (f)  $\rightarrow$  (destination)  
Status Affected: Z  
Description:      Inclusive OR the W register with register 'f'. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in 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, 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.

**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.

**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'.

**NOP**               **No Operation**

---

Syntax:            [ *label* ] NOP  
Operands:        None  
Operation:        No operation  
Status Affected: None  
Description:      No operation.

---

## RETFIE      Return from Interrupt

---

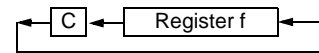
Syntax:      [ *label* ] RETFIE  
 Operands:    None  
 Operation:    TOS → PC,  
                   1 → GIE  
 Status Affected: None

---

## 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' is 0, the result is placed in the W register. If 'd' is 1, the result is stored back in register 'f'.




---

## RETLW        Return with Literal in W

---

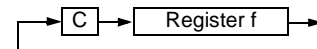
Syntax:      [ *label* ] RETLW k  
 Operands:     $0 \leq k \leq 255$   
 Operation:     $k \rightarrow (W)$ ;  
                   TOS → PC  
 Status Affected: None  
 Description:    The W register is loaded with the eight bit literal 'k'. The program counter is loaded from the top of the stack (the 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' is 0, the result is placed in the W register. If 'd' is 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.

# PIC16F872

---

## **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' is 0, the result is stored in the W register. If 'd' is 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' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.

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

---

Syntax:      *[label]*    SWAPF *f,d*

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

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' is 0, the result is placed in W register. If 'd' is 1, the result is placed in register 'f'.

## 13.0 DEVELOPMENT SUPPORT

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

- Integrated Development Environment
  - MPLAB<sup>®</sup> IDE Software
- Assemblers/Compilers/Linkers
  - MPASM Assembler
  - MPLAB-C17 and MPLAB-C18 C Compilers
  - MPLINK/MPLIB Linker/Librarian
- Simulators
  - MPLAB-SIM Software Simulator
- Emulators
  - MPLAB-ICE Real-Time In-Circuit Emulator
  - PICMASTER<sup>®</sup>/PICMASTER-CE In-Circuit Emulator
  - ICEPIC<sup>™</sup>
- In-Circuit Debugger
  - MPLAB-ICD for PIC16F877
- Device Programmers
  - PRO MATE<sup>®</sup> II Universal Programmer
  - PICSTART<sup>®</sup> Plus Entry-Level Prototype Programmer
- Low-Cost Demonstration Boards
  - SIMICE
  - PICDEM-1
  - PICDEM-2
  - PICDEM-3
  - PICDEM-17
  - SEEVAL<sup>®</sup>
  - KEELOQ<sup>®</sup>

### 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. MPLAB is a Windows<sup>®</sup>-based application which contains:

- Multiple functionality
  - editor
  - simulator
  - programmer (sold separately)
  - emulator (sold separately)
- A full featured editor
- A project manager
- Customizable tool bar and key mapping
- A status bar
- On-line help

MPLAB allows you to:

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

The ability to use MPLAB with Microchip's simulator, MPLAB-SIM, allows a consistent platform and the ability to easily switch from the cost-effective simulator to the full featured emulator with minimal retraining.

### 13.2 MPASM Assembler

MPASM is a full featured universal macro assembler for all PICmicro MCU's. It can produce absolute code directly in the form of HEX files for device programmers, or it can generate relocatable objects for MPLINK.

MPASM has a command line interface and a Windows shell and can be used as a standalone application on a Windows 3.x or greater system. MPASM generates relocatable object files, Intel standard HEX files, MAP files to detail memory usage and symbol reference, an absolute LST file which contains source lines and generated machine code, and a COD file for MPLAB debugging.

MPASM features include:

- MPASM and MPLINK are integrated into MPLAB projects.
- MPASM allows user defined macros to be created for streamlined assembly.
- MPASM allows conditional assembly for multi purpose source files.
- MPASM directives 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 and integrated development environments 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/MPLIB Linker/Librarian

MPLINK is a relocatable linker for MPASM and MPLAB-C17 and MPLAB-C18. It can link relocatable objects from assembly or C source files along with pre-compiled libraries using directives from a linker script.

MPLIB is a librarian for pre-compiled code to be used with MPLINK. When a routine from a library is called from another source file, only the modules that contains that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications. MPLIB manages the creation and modification of library files.

MPLINK features include:

- MPLINK works with MPASM and MPLAB-C17 and MPLAB-C18.
- MPLINK allows all memory areas to be defined as sections to provide link-time flexibility.

MPLIB features include:

- MPLIB makes linking easier because single libraries can be included instead of many smaller files.
- MPLIB helps keep code maintainable by grouping related modules together.
- MPLIB commands allow 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 host 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.

MPLAB-SIM fully supports symbolic debugging using MPLAB-C17 and MPLAB-C18 and MPASM. 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 MPLAB-ICE is provided by the MPLAB Integrated Development Environment (IDE), which allows editing, "make" and download, and source debugging from a single environment.

Interchangeable processor modules allow the system to be easily reconfigured for emulation of different processors. The universal architecture of the MPLAB-ICE allows expansion to support new PICmicro microcontrollers.

The MPLAB-ICE 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 3.x/95/98 environment were chosen to best make these features available to you, the end user.

MPLAB-ICE 2000 is a full-featured emulator system with enhanced trace, trigger, and data monitoring features. Both systems use the same processor modules and will operate across the full operating speed range of the PICmicro MCU.

## 13.7 PICMASTER/PICMASTER CE

The PICMASTER system from Microchip Technology is a full-featured, professional quality emulator system. This flexible in-circuit emulator provides a high-quality, universal platform for emulating Microchip 8-bit PICmicro microcontrollers (MCUs). PICMASTER systems are sold worldwide, with a CE compliant model available for European Union (EU) countries.

## 13.8 ICEPIC

ICEPIC is a low-cost in-circuit emulation 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.

## 13.9 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 PIC16F877 and can be used to develop for this and other PICmicro microcontrollers from the PIC16CXXX family. MPLAB-ICD utilizes the In-Circuit Debugging capability built into the PIC16F87X. This feature, along with Microchip's In-Circuit Serial Programming protocol, offers cost-effective in-circuit flash programming and 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. The MPLAB-ICD is also a programmer for the flash PIC16F87X family.



## **13.10 PRO MATE II Universal Programmer**

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

The PRO MATE II has programmable VDD and VPP supplies which allows 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 can read, verify or program PICmicro devices. It can also set code-protect bits in this mode.

## **13.11 PICSTART Plus Entry Level Development System**

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

PICSTART Plus 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. PICSTART Plus is CE compliant.

## **13.12 SIMICE Entry-Level Hardware Simulator**

SIMICE is an entry-level hardware development system designed to operate in a PC-based environment with Microchip's simulator MPLAB-SIM. Both SIMICE and MPLAB-SIM run under Microchip Technology's MPLAB Integrated Development Environment (IDE) software. Specifically, SIMICE provides hardware simulation for Microchip's PIC12C5XX, PIC12CE5XX, and PIC16C5X families of PICmicro 8-bit microcontrollers. SIMICE works in conjunction with MPLAB-SIM to provide non-real-time I/O port emulation. SIMICE enables a developer to run simulator code for driving the target system. In addition, the target system can provide input to the simulator code. This capability allows for simple and interactive debugging without having to manually generate MPLAB-SIM stimulus files. SIMICE is a valuable debugging tool for entry-level system development.

## **13.13 PICDEM-1 Low-Cost PICmicro Demonstration Board**

The PICDEM-1 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 users can program the sample microcontrollers provided with

the PICDEM-1 board, on a PRO MATE II or PICSTART-Plus programmer, and easily test firmware. The user can also connect the PICDEM-1 board to the MPLAB-ICE emulator and download the firmware to the emulator for testing. Additional 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.

## **13.14 PICDEM-2 Low-Cost PIC16CXX Demonstration Board**

The PICDEM-2 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 board, on a PRO MATE II programmer or PICSTART-Plus, and easily test firmware. The MPLAB-ICE emulator may also be used with the PICDEM-2 board to test firmware. Additional 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.

## **13.15 PICDEM-3 Low-Cost PIC16CXXX Demonstration Board**

The PICDEM-3 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 a 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 board, on a PRO MATE II programmer or PICSTART Plus with an adapter socket, and easily test firmware. The MPLAB-ICE emulator may also be used with the PICDEM-3 board to test firmware. Additional prototype area has been provided to the user for adding hardware and connecting it to the microcontroller socket(s). Some of the features include an 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 board is an LCD panel, with 4 commons and 12 segments, that is capable of displaying time, temperature and day of the week. The PICDEM-3 provides an additional RS-232 interface and Windows 3.1 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.

## **PICDEM-17**

The PICDEM-17 is an evaluation board that demonstrates the capabilities of several Microchip microcontrollers, including PIC17C752, PIC17C756, 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 or PICSTART Plus device programmers and easily debug and test the sample code. In addition, PICDEM-17 supports down-loading of programs to and executing out of external FLASH memory on board. The PICDEM-17 is also usable with the MPLAB-ICE or 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.

## **SEEVAL Evaluation and Programming System**

The SEEVAL SEEPROM Designer's Kit supports all Microchip 2-wire and 3-wire Serial EEPROMs. The kit includes everything necessary to read, write, erase or program special features of any Microchip SEEPROM product including Smart Serials™ and secure serials. The Total Endurance™ Disk is included to aid in trade-off analysis and reliability calculations. The total kit can significantly reduce time-to-market and result in an optimized system.

## **KEELOQ Evaluation and Programming Tools**

KEELOQ evaluation and programming tools support Microchips HCS Secure Data Products. The HCS evaluation kit includes an 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**

Development Tool	PIC12CXX	PIC14000	PIC16C5X	PIC16C6X	PIC16CXX	PIC16F62X	PIC16C7X	PIC16C7XX	PIC16C8X	PIC16F8XX	PIC16C9XX	PIC17C4X	PIC17C7XX	PIC18CXX2	24CXX/ 25CXX/ 93CXX	HCSXX	MGRFXX	MCP2510
MPLAB® Integrated Development Environment	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓				
MPLAB® C17 Compiler																		
MPLAB® C18 Compiler																✓		
MPASM/MPLINK	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
MPLAB®-ICE	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
PICMASTER/PICMASTER-CE	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
ICEPIC™ Low-Cost In-Circuit Emulator	✓		✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓				
MPLAB®-ICD In-Circuit Debugger				✓			✓			✓								
PICSTART® Plus Low-Cost Universal Dev. Kit	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
PRO MATE® II Universal Programmer	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		
SIMICE	✓		✓															
PICDEM-1			✓		✓		✓		✓			✓						
PICDEM-2							✓							✓				
PICDEM-3											✓							
PICDEM-14A																		
PICDEM-17		✓										✓						
KEELOO® Evaluation Kit																✓		
KEELOO 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.

# PIC16F872

---

NOTES:

## 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 +7.5V
Voltage on $\overline{\text{MCLR}}$ with respect to VSS (Note 2).....	0 to +13.25V
Voltage on RA4 with respect to Vss .....	0 to +8.5V
Total power dissipation (Note 1).....	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 and PORTB (combined) .....	200 mA
Maximum current sourced by PORTA and PORTB (combined) .....	200 mA
Maximum current sunk by PORTC.....	200 mA
Maximum current sourced by PORTC .....	200 mA

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

**2:** Voltage spikes below VSS at the  $\overline{\text{MCLR}}$  pin, inducing currents greater than 80 mA, may cause latch-up. Thus, a series resistor of 50-100Ω should be used when applying a “low” level to the  $\overline{\text{MCLR}}$  pin, rather than pulling this pin directly to VSS..

† 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.

# PIC16F872

FIGURE 14-1: PIC16FXXX VOLTAGE-FREQUENCY GRAPH

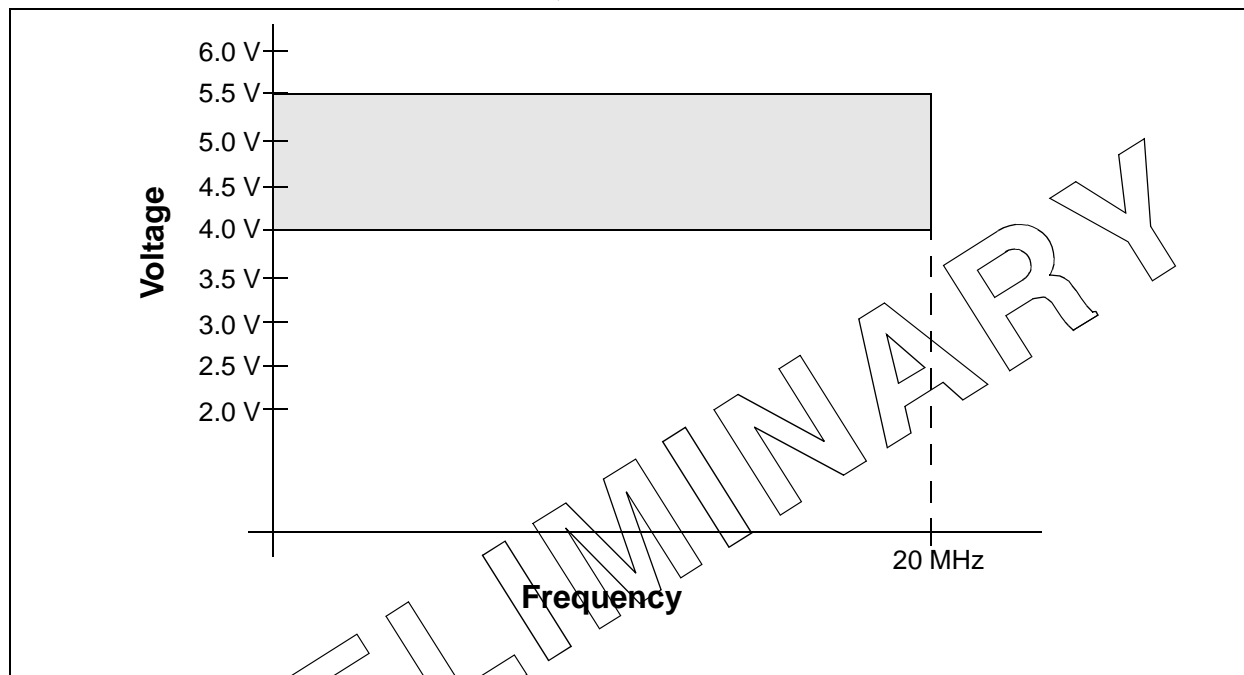
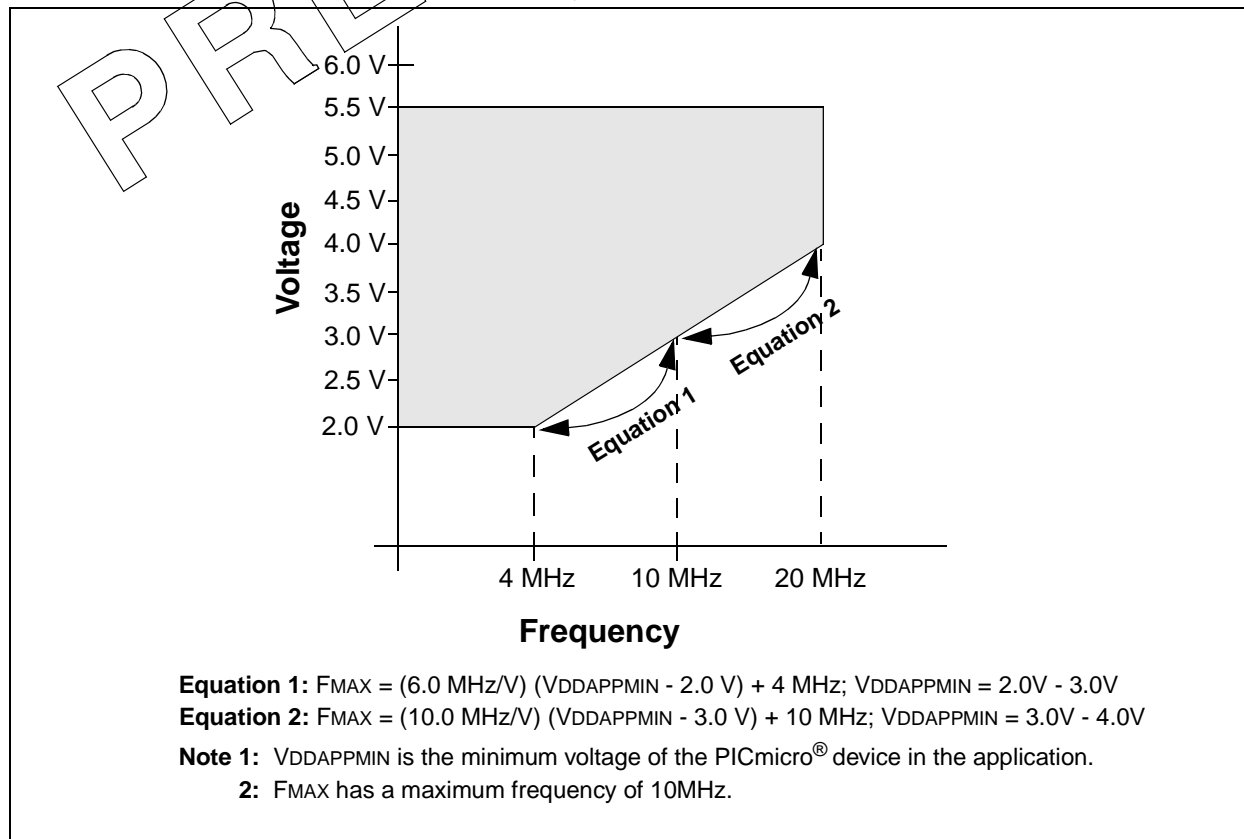


FIGURE 14-2: PIC16LFXXX VOLTAGE-FREQUENCY GRAPH



## 14.1 DC Characteristics: PIC16F872 (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					
Param No.	Characteristic	Sym	Min	Typ†	Max	Units	Conditions
D001 D001A	Supply Voltage	VDD	4.0 4.5 VBOR*	- - -	5.5 5.5 5.5	V V V	XT, RC and LP osc configuration HS osc configuration BOR enabled, Fmax = 14MHz (Note 7)
D002*	RAM Data Retention Voltage (Note 1)	VDR	-	1.5	-	V	
D003	VDD start voltage to ensure internal Power-on Reset signal	VPOR	-	VSS	-	V	See section on Power-on Reset for details
D004*	VDD rise rate to ensure internal Power-on Reset signal	SVDD	0.05	-	-	V/ms	See section on Power-on Reset for details
D005	Brown-out Reset Voltage	VBOR	3.7	4.0	4.35	V	BODEN bit in configuration word enabled
D010	Supply Current (Note 2,5)	IDD	-	1.6	4	mA	XT, RC osc configuration FOSC = 4 MHz, VDD = 5.5V (Note 4)
D013			-	7	15	mA	HS osc configuration FOSC = 20 MHz, VDD = 5.5V
D015*	Brown-out Reset Current (Note 6)	$\Delta$ IBOR	-	85	200	$\mu\text{A}$	BOR enabled VDD = 5.0V
D020 D021 D021A	Power-down Current (Note 3,5)	IPD	-	10.5 1.5 1.5	42 16 19	$\mu\text{A}$ $\mu\text{A}$ $\mu\text{A}$	VDD = 4.0V, WDT enabled, $-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$ VDD = 4.0V, WDT disabled, $-0^{\circ}\text{C}$ to $+70^{\circ}\text{C}$ VDD = 4.0V, WDT disabled, $-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$
D023*	Brown-out Reset Current (Note 6)	$\Delta$ IBOR	-	85	200	$\mu\text{A}$	BOR enabled VDD = 5.0V

Legend: \* 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.

**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 tristated, 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 kOhm.

**5:** Timer1 oscillator (when enabled) adds approximately 20  $\mu\text{A}$  to the specification. This value is from characterization and is for design guidance only. This is not tested.

**6:** The  $\Delta$  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.

# PIC16F872

## 14.2 DC Characteristics: PIC16LF872 (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					
Param No.	Characteristic	Sym	Min	Typ†	Max	Units	Conditions
D001	Supply Voltage	VDD	2.0	-	5.5	V	LP, XT, RC osc configuration (DC - 4 MHz)
D002*	RAM Data Retention Voltage (Note 1)	VDR	-	1.5	-	V	
D003	VDD start voltage to ensure internal Power-on Reset signal	VPOR	-	VSS	-	V	See section on Power-on Reset for details
D004*	VDD rise rate to ensure internal Power-on Reset signal	SVDD	0.05	-	-	V/ms	See section on Power-on Reset for details
D005	Brown-out Reset Voltage	VBOR	3.7	4.0	4.35	V	BODEN bit in configuration word enabled
D010	Supply Current (Note 2,5)	IDD	-	0.6	2.0	mA	XT, RC osc configuration FOSC = 4 MHz, VDD = 3.0V (Note 4)
D010A				20	35	$\mu\text{A}$	LP osc configuration FOSC = 32 kHz, VDD = 3.0V, WDT disabled
D015*	Brown-out Reset Current (Note 6)	$\Delta\text{IBOR}$	-	85	200	$\mu\text{A}$	BOR enabled VDD = 5.0V
D020	Power-down Current (Note 3,5)	IPD	-	7.5	30	$\mu\text{A}$	VDD = 3.0V, WDT enabled, $-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$
D021			-	0.9	5	$\mu\text{A}$	VDD = 3.0V, WDT disabled, $0^{\circ}\text{C}$ to $+70^{\circ}\text{C}$
D021A			-	0.9	5	$\mu\text{A}$	VDD = 3.0V, WDT disabled, $-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$
D023*	Brown-out Reset Current (Note 6)	$\Delta\text{IBOR}$	-	85	200	$\mu\text{A}$	BOR enabled VDD = 5.0V

Legend: \* 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 tristated, 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 kOhm.

**5:** Timer1 oscillator (when enabled) adds approximately 20  $\mu\text{A}$  to the specification. This value is from characterization and is for design guidance only. This is not tested.

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



## 14.3 DC Characteristics: PIC16F872 and PIC16LF872 (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 Operating voltage VDD range as described in DC spec Section 14.1 and Section 14.2.					
Param No.	Characteristic	Sym	Min	Typ†	Max	Units	Conditions
D030 D030A D031 D032 D033 D034 D034A	<b>Input Low Voltage</b> I/O ports with TTL buffer with Schmitt Trigger buffer MCLR, OSC1 (in RC mode) OSC1 (in XT, HS and LP) Ports RC3 and RC4 with Schmitt Trigger buffer with SMBus	V <sub>IL</sub>	V <sub>SS</sub> V <sub>SS</sub> V <sub>SS</sub> V <sub>SS</sub> V <sub>SS</sub> V <sub>SS</sub> -0.5	- - - - - - -	0.15V <sub>DD</sub> 0.8V 0.2V <sub>DD</sub> 0.2V <sub>DD</sub> 0.3V <sub>DD</sub> 0.3V <sub>DD</sub> 0.6	V V V V V V V	For entire VDD range $4.5\text{V} \leq V_{DD} \leq 5.5\text{V}$ Note 1 For entire VDD range for VDD = 4.5 to 5.5V
D040 D040A D041 D042 D042A D043 D044 D044A	<b>Input High Voltage</b> I/O ports with TTL buffer with Schmitt Trigger buffer MCLR OSC1 (XT, HS and LP) OSC1 (in RC mode) Ports RC3 and RC4 with Schmitt Trigger buffer with SMBus	V <sub>IH</sub>	2.0 0.25V <sub>DD</sub> +0.8V 0.8V <sub>DD</sub> 0.8V <sub>DD</sub> 0.7V <sub>DD</sub> 0.9V <sub>DD</sub> 0.7V <sub>DD</sub> 1.4	- - - - - - - - -	V <sub>DD</sub> V <sub>DD</sub> V <sub>DD</sub> V <sub>DD</sub> V <sub>DD</sub> V <sub>DD</sub> V <sub>DD</sub> V <sub>DD</sub> 5.5	V V V V V V V V V	$4.5\text{V} \leq V_{DD} \leq 5.5\text{V}$ For entire VDD range For entire VDD range Note 1 For entire VDD range for VDD = 4.5 to 5.5V
D070	PORTB weak pull-up current	IPURB	50	250	400	μA	VDD = 5V, VPIN = VSS
D060 D061 D063	<b>Input Leakage Current</b> (Notes 2, 3) I/O ports MCLR, RA4/T0CKI OSC1	I <sub>IL</sub>	- - -	- - -	±1 ±5 ±5	μA μA μA	VSS ≤ VPIN ≤ VDD, Pin at hi-impedance VSS ≤ VPIN ≤ VDD VSS ≤ VPIN ≤ VDD, XT, HS and LP osc configuration
D080 D083 D090 D092	<b>Output Low Voltage</b> I/O ports OSC2/CLKOUT (RC osc config) <b>Output High Voltage</b> I/O ports (Note 3) OSC2/CLKOUT (RC osc config)	V <sub>OL</sub> V <sub>OH</sub>	- - VDD - 0.7 VDD - 0.7	- - - -	0.6 0.6 - -	V V V V	I <sub>OL</sub> = 8.5 mA, VDD = 4.5V, -40°C to +85°C I <sub>OL</sub> = 1.6 mA, VDD = 4.5V, -40°C to +85°C I <sub>OH</sub> = -3.0 mA, VDD = 4.5V, -40°C to +85°C I <sub>OH</sub> = -1.3 mA, VDD = 4.5V, -40°C to +85°C

Legend: \* 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/CLKIN pin is a Schmitt Trigger input. It is not recommended that the PIC16F87X be driven with external clock in RC mode.

- 2:** The leakage current on the MCLR pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.
- 3:** Negative current is defined as current sourced by the pin.

# PIC16F872

DC CHARACTERISTICS		Standard Operating Conditions (unless otherwise stated)					
		Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial					
		Operating voltage $V_{DD}$ range as described in DC spec Section 14.1 and Section 14.2.					
Param No.	Characteristic	Sym	Min	Typ†	Max	Units	Conditions
D150*	Open-Drain High Voltage	VOD	-	-	8.5	V	RA4 pin
<b>Capacitive Loading Specs on Output Pins</b>							
D100	OSC2 pin	Cosc2	-	-	15	pF	In XT, HS and LP modes when external clock is used to drive OSC1.
D101	All I/O pins and OSC2 (in RC mode) SCL, SDA in I <sup>2</sup> C mode	CIO	-	-	50	pF	
D102		CB	-	-	400	pF	
<b>Data EEPROM Memory</b>							
D120	Endurance	Ed	100K	-	-	E/W	25°C at 5V Using EECON to read/write Vmin = min operating voltage
D121	VDD for read/write	VDRW	Vmin	-	5.5	V	
D122	Erase/write cycle time	TDEW	-	4	8	ms	
<b>Program FLASH Memory</b>							
D130	Endurance	EP	1000	-	-	E/W	25°C at 5V Vmin = min operating voltage using EECON to read/write, Vmin = min operating voltage
D131	VDD for read	VPR	Vmin	-	5.5	V	
D132a	VDD for erase/write		Vmin	-	5.5	V	
D133	Erase/Write cycle time	TPEW	-	4	8	ms	

Legend: \* 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/CLKIN pin is a Schmitt Trigger input. It is not recommended that the PIC16F87X 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.

## 14.4 Timing Parameter Symbology

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

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

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

Lowercase letters (pp) and their meanings:

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

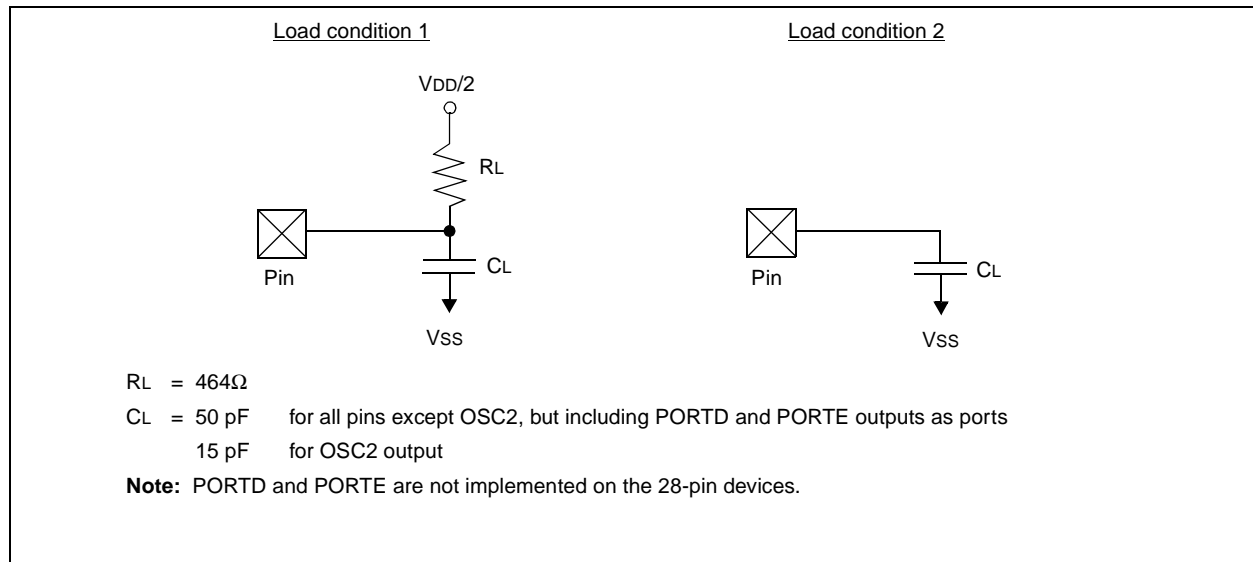
Uppercase letters and their meanings:

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

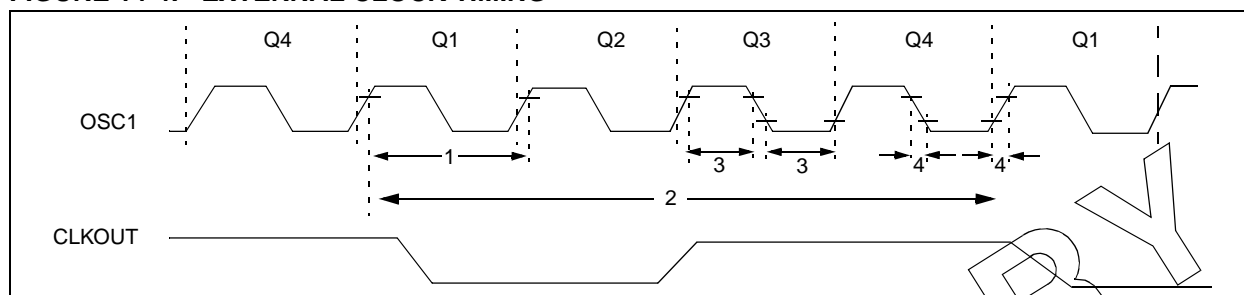
Tcc:ST (I<sup>2</sup>C specifications only)

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

**FIGURE 14-3: LOAD CONDITIONS**



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



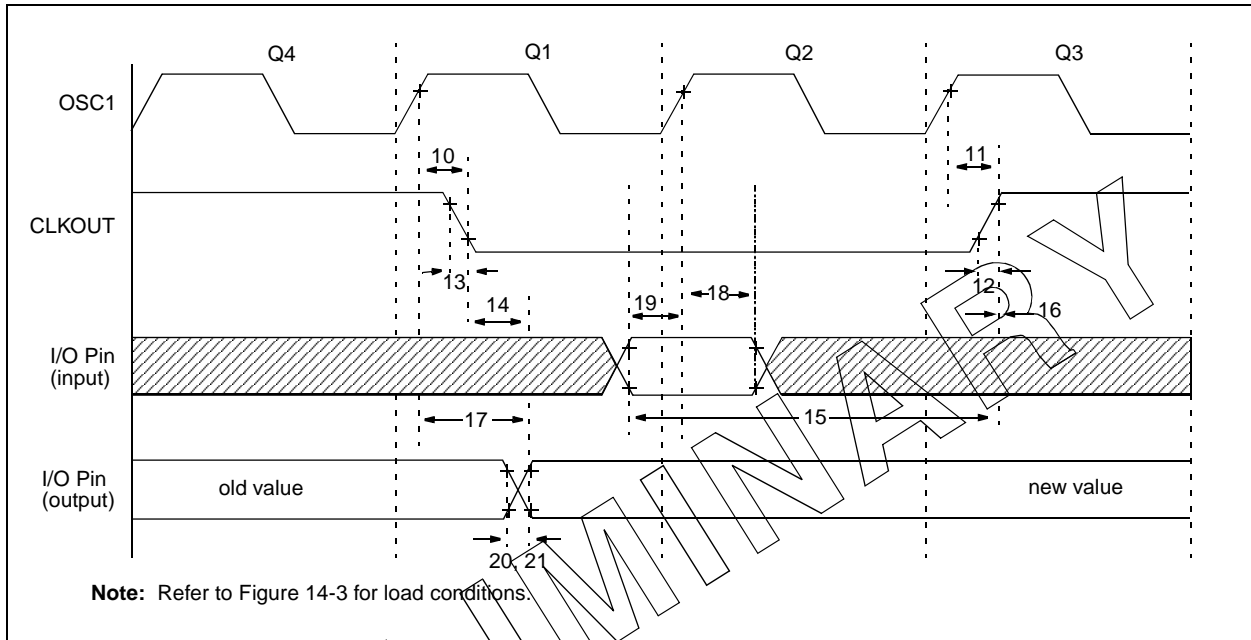
**TABLE 14-1: EXTERNAL CLOCK TIMING REQUIREMENTS**

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
	FOSC	<b>External CLKIN Frequency (Note 1)</b>	DC	—	4	MHz	XT and RC osc mode
			DC	—	4	MHz	HS osc mode (-04)
			DC	—	20	MHz	HS osc mode (-20)
			DC	—	200	kHz	LP osc mode
		<b>Oscillator Frequency (Note 1)</b>	DC	—	4	MHz	RC osc mode
			0.1	—	4	MHz	XT osc mode
			4	—	20	MHz	HS osc mode
			5	—	200	kHz	LP osc mode
1	TOSC	<b>External CLKIN Period (Note 1)</b>	250	—	—	ns	XT and RC osc mode
			250	—	—	ns	HS osc mode (-04)
			50	—	—	ns	HS osc mode (-20)
			5	—	—	μs	LP osc mode
		<b>Oscillator Period (Note 1)</b>	250	—	—	ns	RC osc mode
			250	—	10,000	ns	XT osc mode
			250	—	250	ns	HS osc mode (-04)
			50	—	250	ns	HS osc mode (-20)
			5	—	—	μs	LP osc mode
2	TCY	<b>Instruction Cycle Time (Note 1)</b>	200	TCY	DC	ns	TCY = 4/FOSC
3	TosL, TosH	<b>External Clock in (OSC1) High or Low Time</b>	100	—	—	ns	XT oscillator
			2.5	—	—	μs	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

Legend: † 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/CLKIN pin. When an external clock input is used, the "Max." cycle time limit is "DC" (no clock) for all devices.

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



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

Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
10*	TosH2ckL	OSC1↑ to CLKOUT↓	—	75	200	ns	Note 1
11*	TosH2ckH	OSC1↑ to CLKOUT↑	—	75	200	ns	Note 1
12*	TckR	CLKOUT↑ rise time	—	35	100	ns	Note 1
13*	TckF	CLKOUT fall time	—	35	100	ns	Note 1
14*	TckL2ioV	CLKOUT ↓ to Port out valid	—	—	0.5TCY + 20	ns	Note 1
15*	TioV2ckH	Port in valid before CLKOUT ↑	ToSC + 200	—	—	ns	Note 1
16*	TckH2ioL	Port in hold after CLKOUT ↑	0	—	—	ns	Note 1
17*	TosH2ioV	OSC1↑ (Q1 cycle) to Port out valid	—	100	255	ns	
18*	TosH2ioL	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	TCY	—	—	ns	
23††*	Trbp	RB7:RB4 change INT high or low time	TCY	—	—	ns	

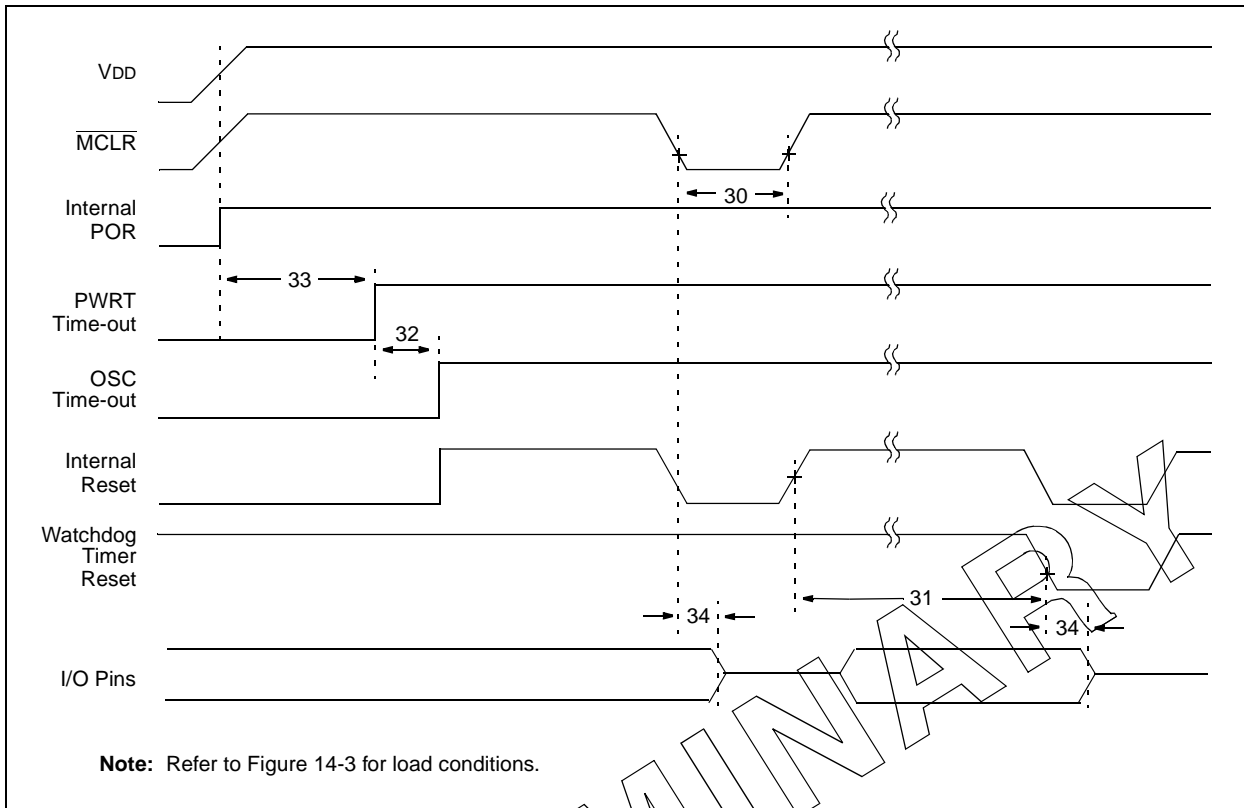
Legend: \* These parameters are characterized but not tested.

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

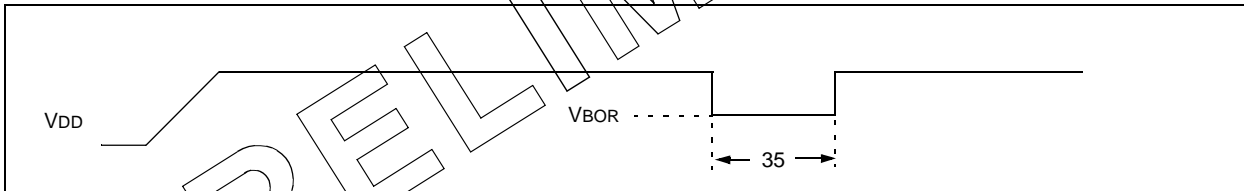
†† These parameters are asynchronous events not related to any internal clock edges.

**Note 1:** Measurements are taken in RC Mode where CLKOUT output is 4 x Tosc.

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



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



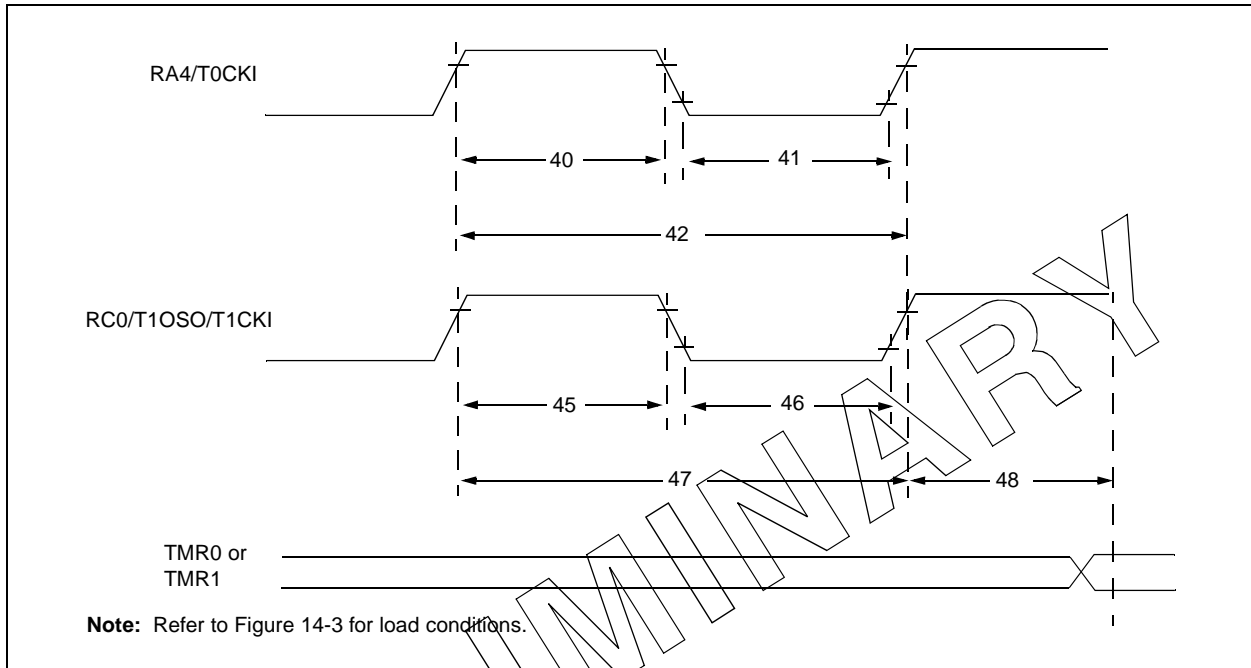
**TABLE 14-3: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER, POWER-UP TIMER, AND BROWN-OUT RESET REQUIREMENTS**

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
30	T <sub>mcl</sub>	MCLR Pulse Width (low)	2	—	—	μs	V <sub>DD</sub> = 5V, -40°C to +85°C
31*	T <sub>wdt</sub>	Watchdog Timer Time-out Period (No Prescaler)	7	18	33	ms	V <sub>DD</sub> = 5V, -40°C to +85°C
32	T <sub>ost</sub>	Oscillation Start-up Timer Period	—	1024 T <sub>osc</sub>	—	—	T <sub>osc</sub> = OSC1 period
33*	T <sub>pwrt</sub>	Power up Timer Period	28	72	132	ms	V <sub>DD</sub> = 5V, -40°C to +85°C
34	T <sub>ioz</sub>	I/O Hi-impedance from MCLR Low or Watchdog Timer Reset	—	—	2.1	μs	
35	T <sub>bor</sub>	Brown-out Reset pulse width	100	—	—	μs	V <sub>DD</sub> ≤ V <sub>BOR</sub> (D005)

Legend: \* 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**



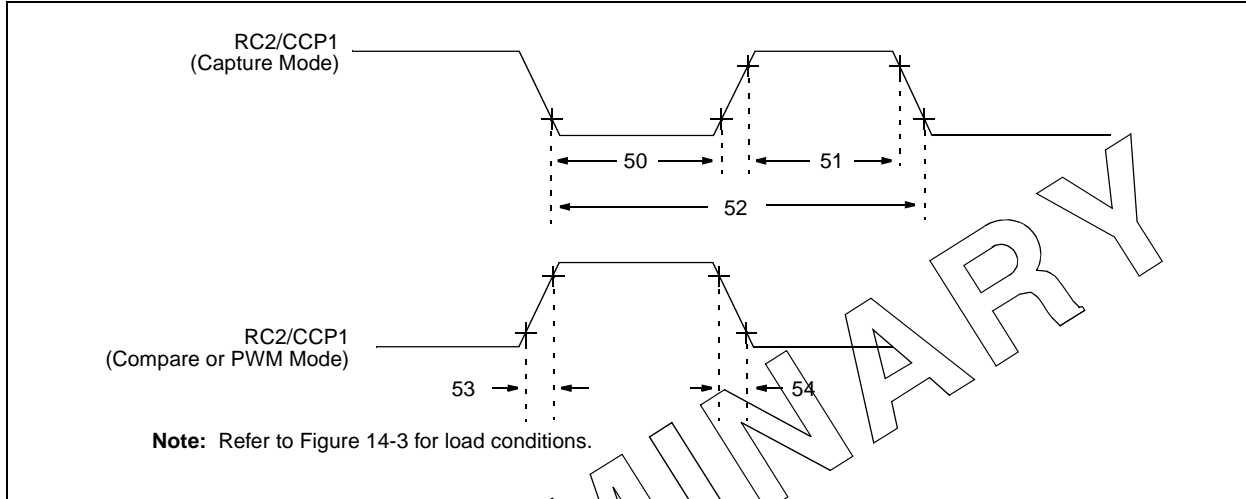
**TABLE 14-4: TIMER0 AND TIMER1 EXTERNAL CLOCK REQUIREMENTS**

Param No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions	
40*	Tt0H	T0CKI High Pulse Width	No Prescaler	$0.5TCY + 20$	—	—	ns	Must also meet parameter 42	
			With Prescaler	10	—	—	ns		
41*	Tt0L	T0CKI Low Pulse Width	No Prescaler	$0.5TCY + 20$	—	—	ns	Must also meet parameter 42	
			With Prescaler	10	—	—	ns		
42*	Tt0P	T0CKI Period	No Prescaler	$TCY + 40$	—	—	ns	N = prescale value (2, 4, ..., 256)	
			With Prescaler	Greater of: $20$ or $\frac{TCY + 40}{N}$	—	—	ns		
45*	Tt1H	T1CKI High Time	Synchronous, Prescaler = 1	$0.5TCY + 20$	—	—	ns	Must also meet parameter 47	
			Synchronous, Prescaler = 2,4,8	Standard(F)	15	—	—		ns
			Asynchronous	Standard(F)	30	—	—		ns
				Extended(LF)	50	—	—		ns
46*	Tt1L	T1CKI Low Time	Synchronous, Prescaler = 1	$0.5TCY + 20$	—	—	ns	Must also meet parameter 47	
			Synchronous, Prescaler = 2,4,8	Standard(F)	15	—	—		ns
			Asynchronous	Standard(F)	30	—	—		ns
				Extended(LF)	50	—	—		ns
47*	Tt1P	T1CKI input period	Synchronous	Standard(F)	Greater of: $30$ or $\frac{TCY + 40}{N}$	—	—	ns	N = prescale value (1, 2, 4, 8)
				Extended(LF)	Greater of: $50$ or $\frac{TCY + 40}{N}$				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		$2Tosc$	—	$7Tosc$	—		

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



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

Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions	
50*	TccL	CCP1 input low time	No Prescaler	$0.5T_{CY} + 20$	—	—	ns	
			With Prescaler	Standard(F)	10	—	—	ns
				Extended(LF)	20	—	—	ns
51*	TccH	CCP1 input high time	No Prescaler	$0.5T_{CY} + 20$	—	—	ns	
			With Prescaler	Standard(F)	10	—	—	ns
				Extended(LF)	20	—	—	ns
52*	TccP	CCP1 input period	$\frac{3T_{CY} + 40}{N}$	—	—	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	

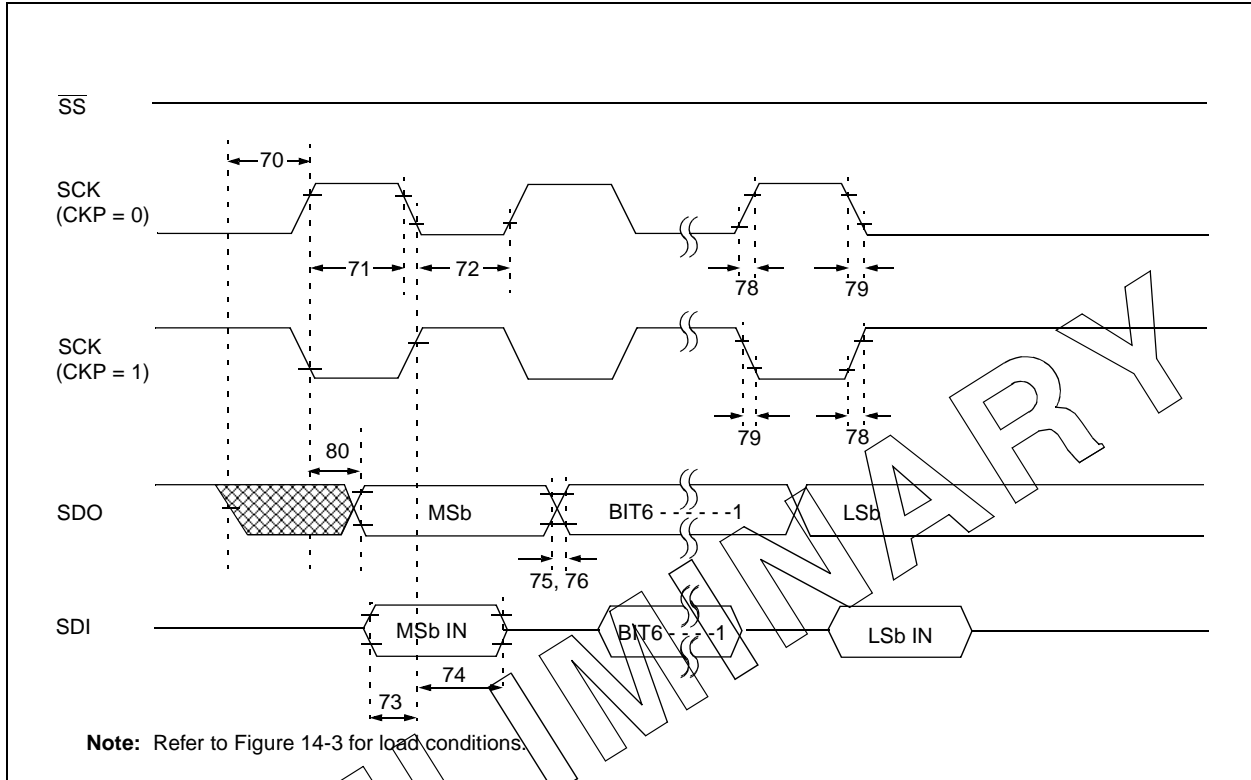
\* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated.

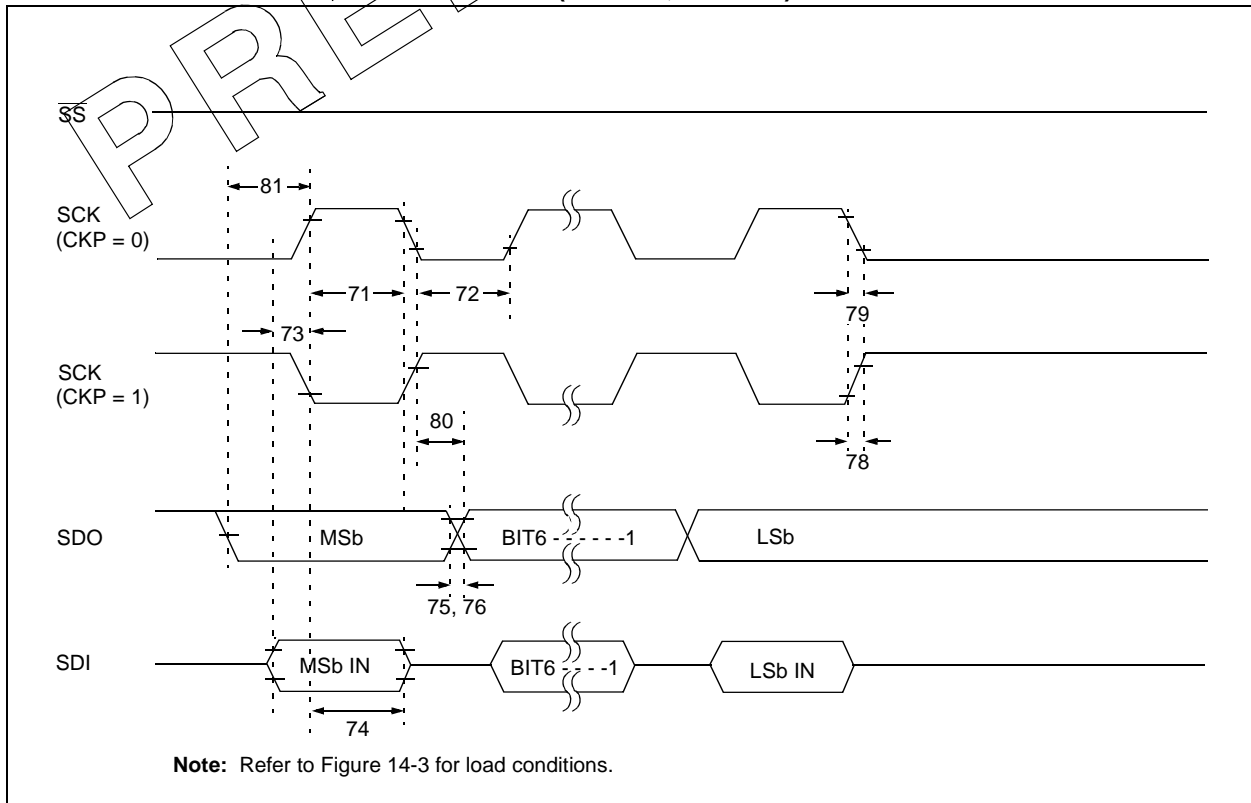
These parameters are for design guidance only and are not tested.



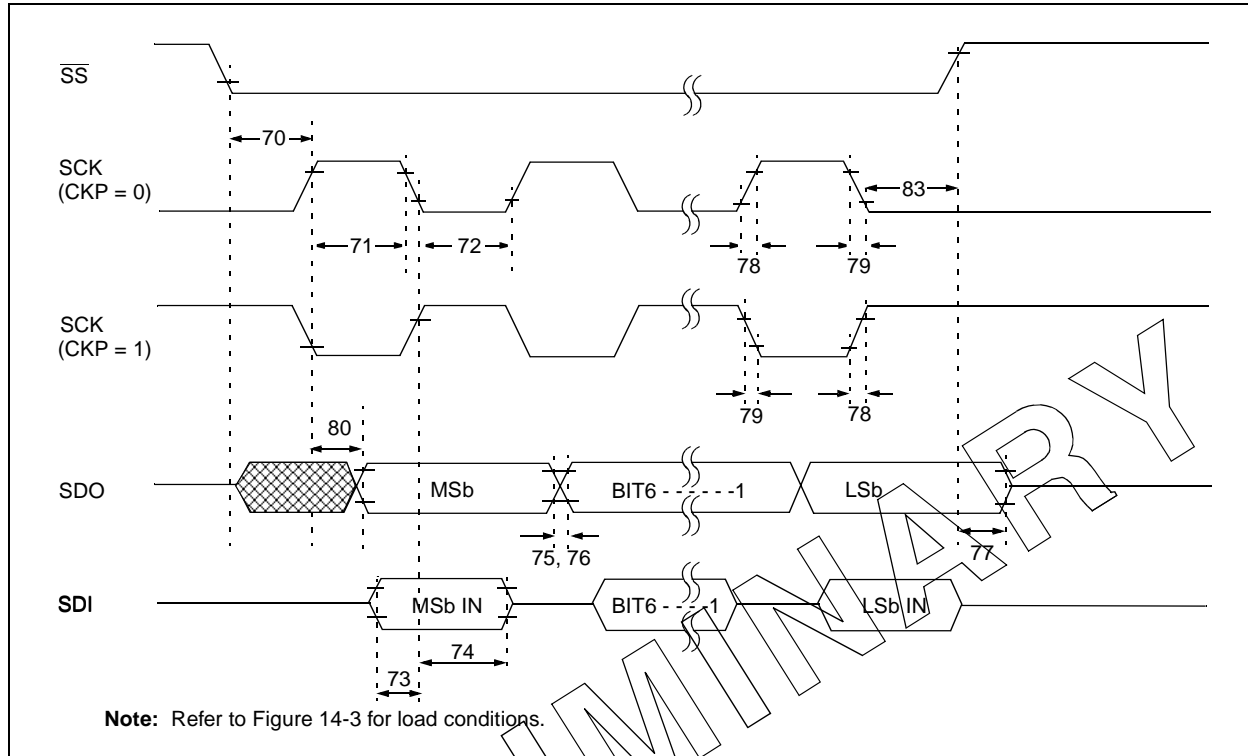
**FIGURE 14-10: SPI MASTER MODE TIMING (CKE = 0, SMP = 0)**



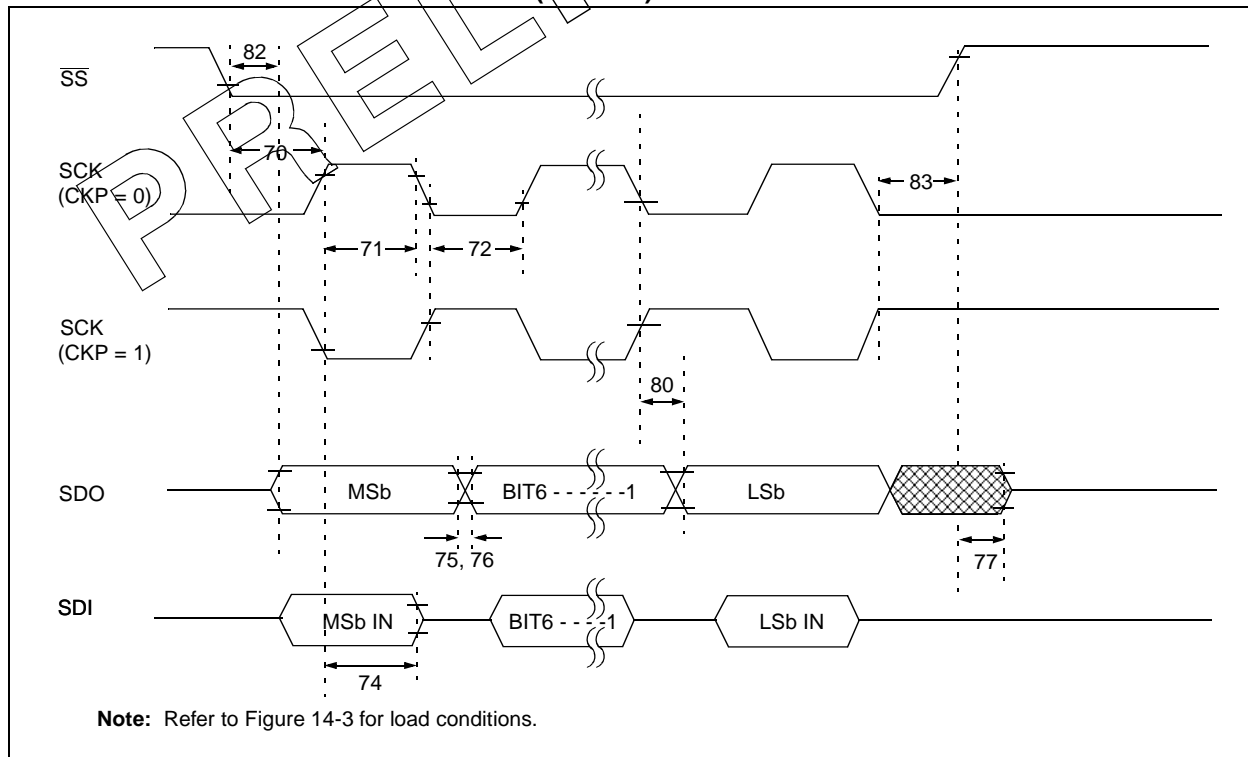
**FIGURE 14-11: SPI MASTER MODE TIMING (CKE = 1, SMP = 1)**



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



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



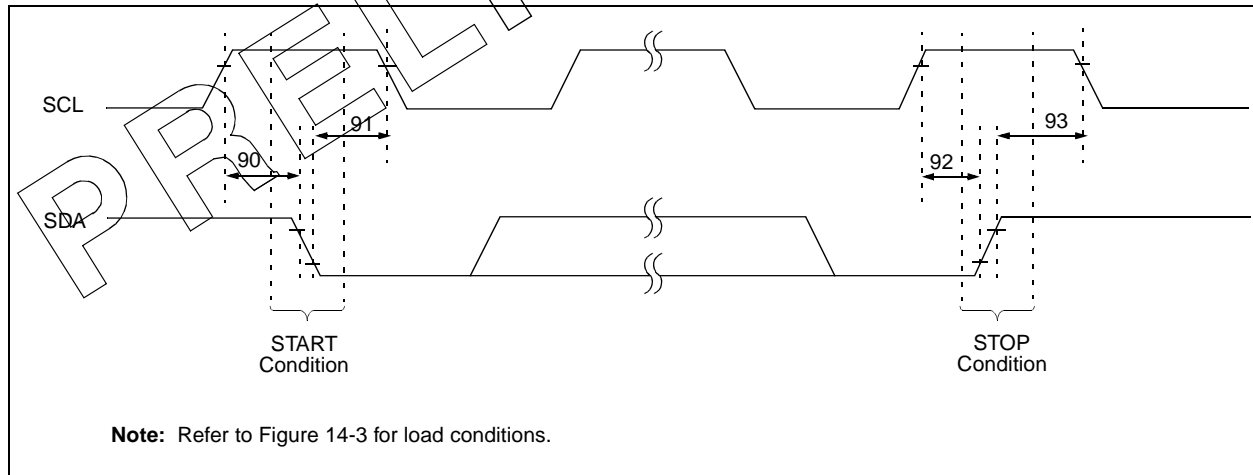
**TABLE 14-6: SPI MODE REQUIREMENTS**

Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
70*	Tssl2sch, Tssl2scL	SS↓ to SCK↓ or SCK↑ input	Tcy	—	—	ns	
71*	Tsch	SCK input high time (slave mode)	Tcy + 20	—	—	ns	
72*	TscL	SCK input low time (slave mode)	Tcy + 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	—	10	25	ns	Standard(F) Extended(LF)
76*	TdoF	SDO data output fall time	—	10	25	ns	
77*	TssH2doZ	SS↑ to SDO output hi-impedance	10	—	50	ns	
78*	Tscr	SCK output rise time (master mode)	—	10	25	ns	Standard(F) Extended(LF)
79*	TscF	SCK output fall time (master mode)	—	10	25	ns	
80*	Tsch2doV, TscL2doV	SDO data output valid after SCK edge	—	—	50	ns	Standard(F) Extended(LF)
81*	TdoV2sch, TdoV2scL	SDO data output setup to SCK edge	Tcy	—	—	ns	
82*	Tssl2doV	SDO data output valid after SS↓ edge	—	—	50	ns	
83*	Tsch2ssH, TscL2ssH	SS ↑ after SCK edge	1.5Tcy + 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**

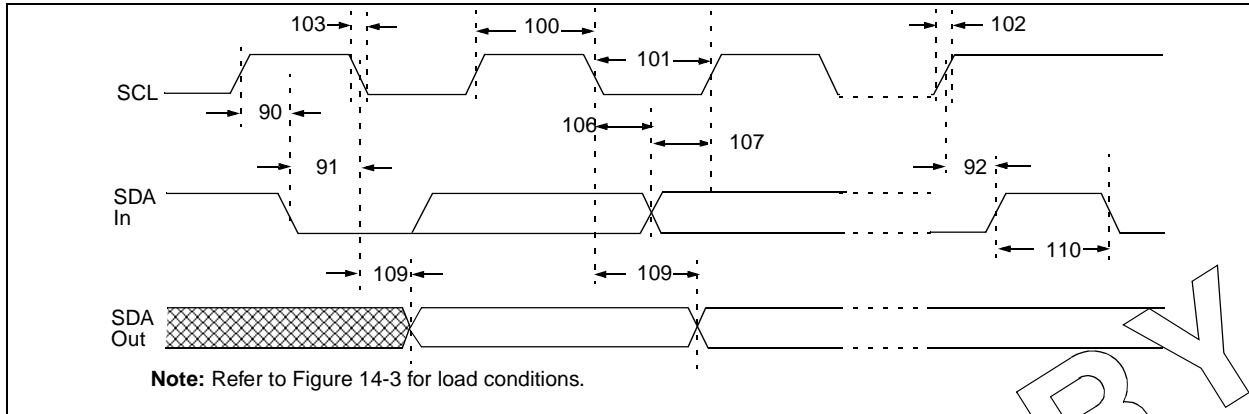


**TABLE 14-7: I<sup>2</sup>C BUS START/STOP BITS REQUIREMENTS**

Parameter No.	Sym	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	—	—		

# PIC16F872

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



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

Param No.	Sym	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	1.5T <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	1.5T <sub>cy</sub>	—	—	—
102	Tr	SDA and SCL rise time	100 kHz mode	—	1000	ns	—
			400 kHz mode	20 + 0.1Cb	300	ns	Cb is specified to be from 10 to 400 pF
103	Tf	SDA and SCL fall time	100 kHz mode	—	300	ns	—
			400 kHz mode	20 + 0.1Cb	300	ns	Cb is specified to be from 10 to 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	Note 2
			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	Note 1
			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	
	Cb	Bus capacitive loading	—	400	pF	—	

**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.

**Note 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.

**TABLE 14-9: PIC16F872 AND PIC16LF872 (INDUSTRIAL)**

Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions	
A01	NR	Resolution	—	—	10-bits	bit	$V_{REF} = V_{DD} = 5.12V$ , $V_{SS} \leq V_{AIN} \leq V_{REF}$	
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	$V_{REF} = V_{DD} = 5.12V$ , $V_{SS} \leq V_{AIN} \leq V_{REF}$	
A06	EOFF	Offset error	—	—	$< \pm 1$	LSb	$V_{REF} = V_{DD} = 5.12V$ , $V_{SS} \leq V_{AIN} \leq V_{REF}$	
A07	EGN	Gain error	—	—	$< \pm 1$	LSb	$V_{REF} = V_{DD} = 5.12V$ , $V_{SS} \leq V_{AIN} \leq V_{REF}$	
A10	—	Monotonicity <sup>(3)</sup>	—	guaranteed	—	—	$V_{SS} \leq V_{AIN} \leq V_{REF}$	
A20	VREF	Reference voltage ( $V_{REF+} - V_{REF-}$ )	2.0V	—	$V_{DD} + 0.3$	V	Absolute minimum electrical spec. To ensure 10-bit accuracy.	
A21	VREF+	Reference voltage High	$V_{DD} - 2.5V$	—	$V_{DD} + 0.3V$	V		
A22	VREF-	Reference voltage low	$V_{SS} - 0.3V$	—	$V_{REF+} - 2.0V$	V		
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 ( $V_{DD}$ )	Standard	—	220	—	$\mu A$	Average current consumption when A/D is on. (Note 1)
			Extended	—	90	—	$\mu A$	
A50	IREF	VREF input current (Note 2)	—	10	—	1000	$\mu A$	During VAIN acquisition. Based on differential of V <sub>HOLD</sub> to VAIN to charge CHOLD, see Section 10.1.
			—	—	—	10	$\mu A$	During A/D Conversion cycle

\* These parameters are characterized but not tested.

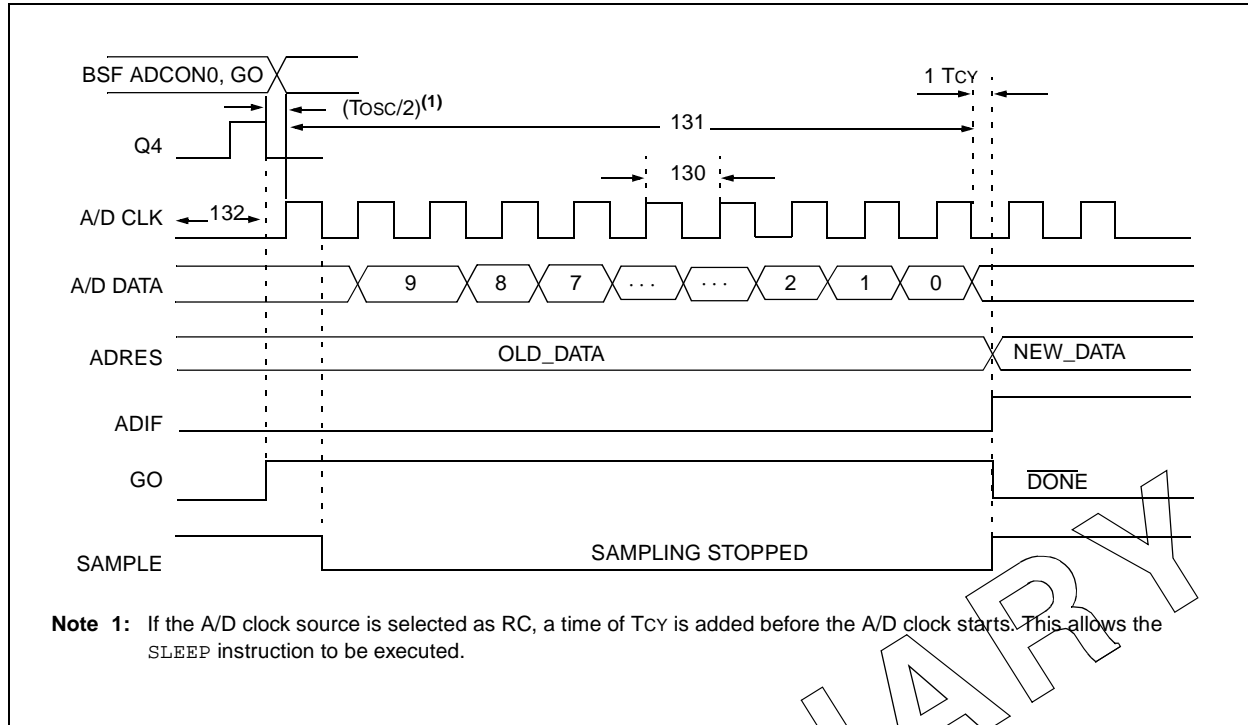
† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**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 RA3 pin or VDD pin, whichever is selected as 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	Standard(F)	1.6	—	—	μs	TOSC based, VREF ≥ 3.0V
			Extended(LF)	3.0	—	—	μs	TOSC based, VREF ≥ 2.0V
		Standard(F)	2.0	4.0	6.0	μs	A/D RC Mode	
			Extended(LF)	3.0	6.0	9.0	μs	A/D RC Mode
131	TCNV	Conversion time (not including S/H time) (Note 1)	—	—	12	TAD		
132	TACQ	Acquisition time	Note 2	40	—	—	μs	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).
			10*	—	—	—	μs	
134	TGO	Q4 to A/D clock start	—	$T_{osc}/2$ §	—	—	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.

§ This specification ensured by design.

**Note 1:** ADRES register may be read on the following  $T_{CY}$  cycle.

**2:** See Section 10.1 for min conditions.

## 15.0 DC AND AC CHARACTERISTICS GRAPHS AND TABLES

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

In some graphs or tables, the data presented are **outside specified operating range** (i.e., outside specified V<sub>DD</sub> range). This is for **information only** and devices are ensured to operate properly only within the specified range.

The data presented in this section is a **statistical summary** of data collected on units from different lots over a period of time and matrix samples. 'Typical' represents the mean of the distribution at 25°C. 'Max' or 'min' represents (mean + 3 $\sigma$ ) or (mean - 3 $\sigma$ ) respectively, where  $\sigma$  is standard deviation, over the whole temperature range.

**Graphs and Tables not available at this time.**

# PIC16F872

---

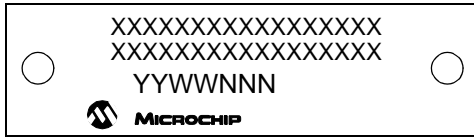
NOTES:



## 16.0 PACKAGING INFORMATION

### 16.1 Package Marking Information

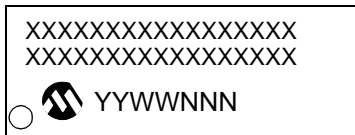
28-Lead PDIP (Skinny DIP)



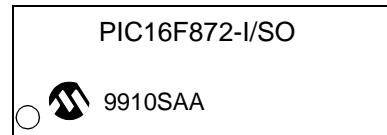
Example



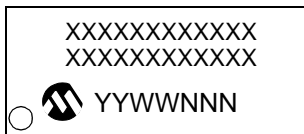
28-Lead SOIC



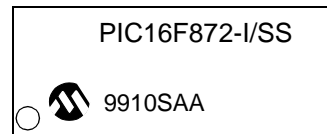
Example



28-Lead SSOP



Example



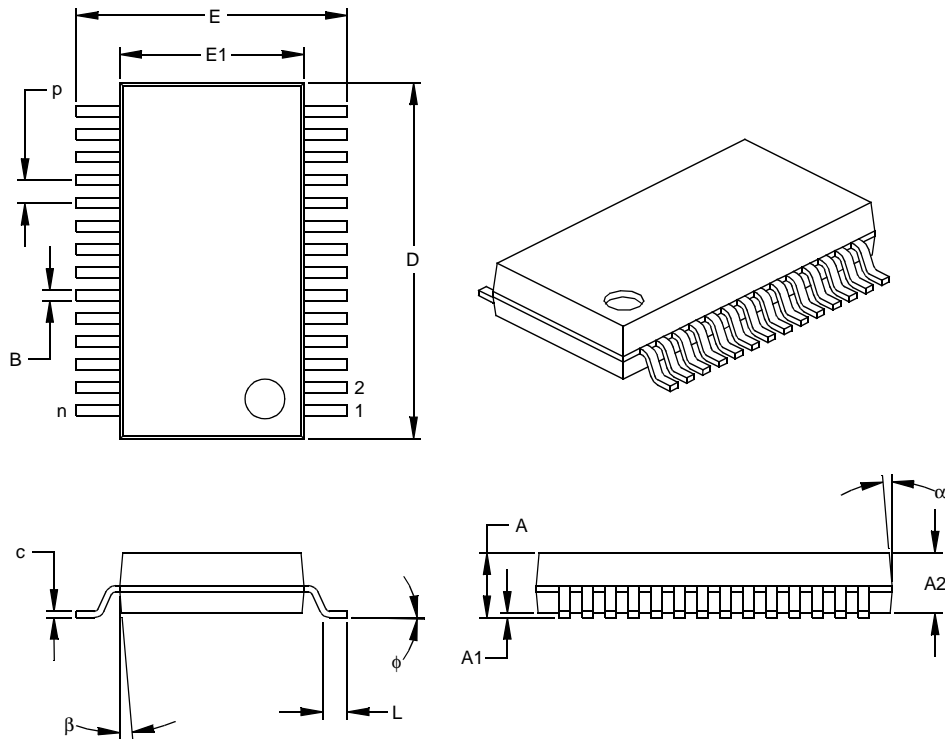
<b>Legend:</b>	MM...M	Microchip part number information
	XX...X	Customer specific information*
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week '01')
	NNN	Alphanumeric traceability code

**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 OTP marking consists of Microchip part number, year code, week code, facility code, mask rev#, and assembly code. For OTP 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.

# PIC16F872

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



Dimension Limits	Units	INCHES			MILLIMETERS*		
		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

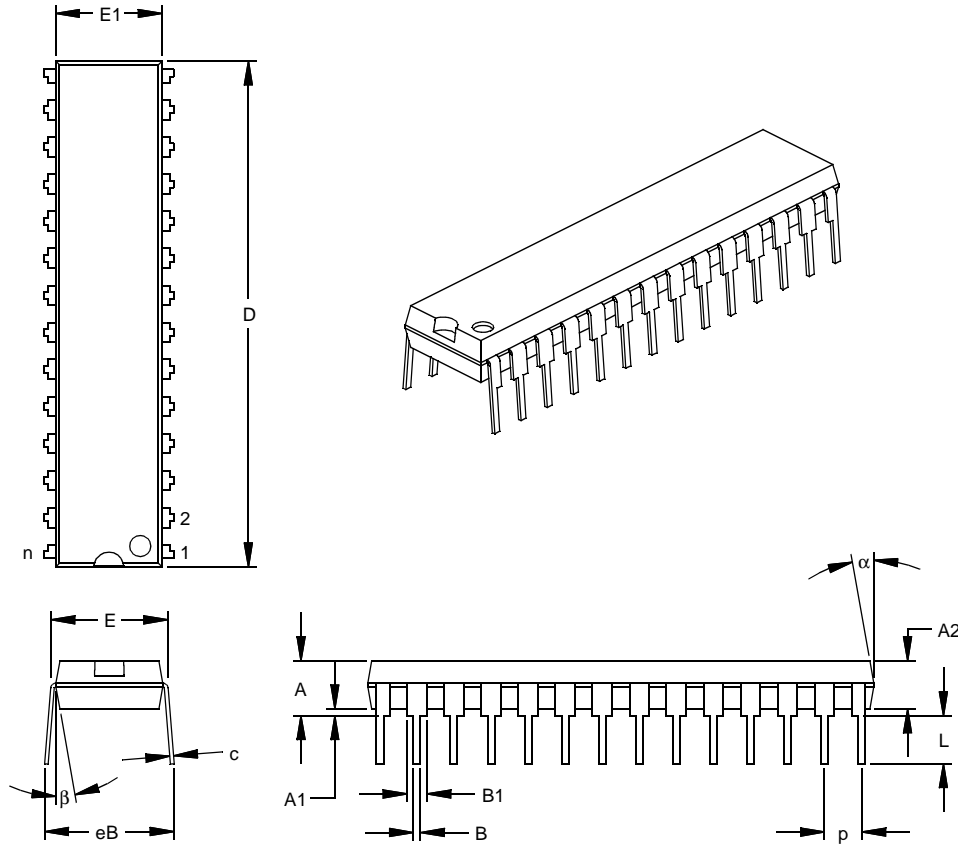
Notes:

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



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

\*Controlling Parameter

Notes:

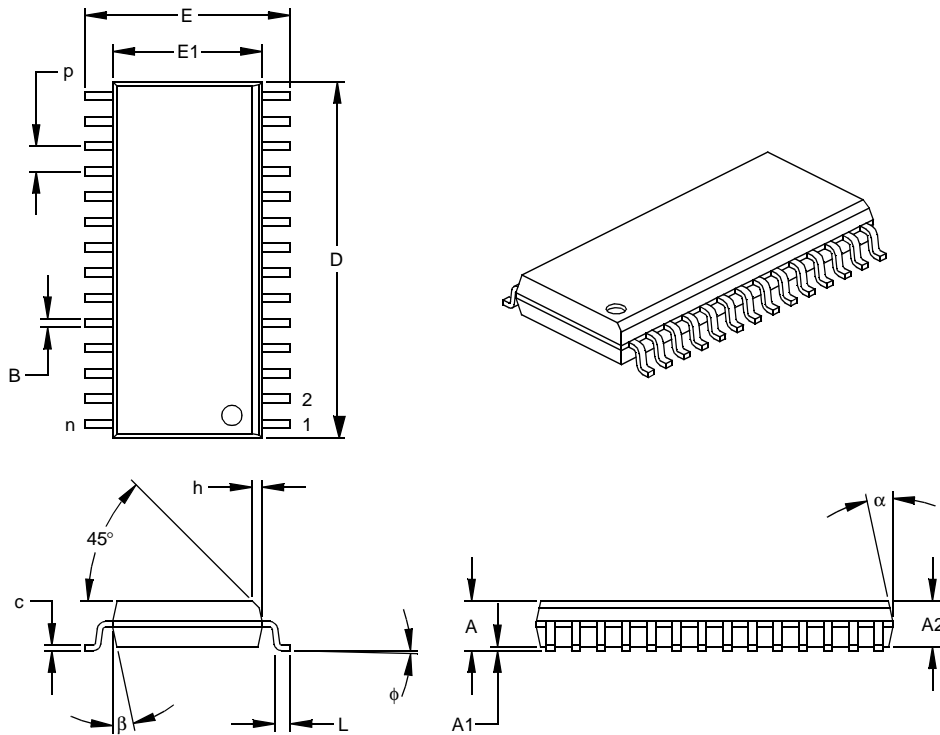
Dimension D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" (0.254mm) per side.

JEDEC Equivalent: MO-095

Drawing No. C04-070

# PIC16F872

## 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	$\phi$	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	$\alpha$	0	12	15	0	12	15
Mold Draft Angle Bottom	$\beta$	0	12	15	0	12	15

\*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: MS-013

Drawing No. C04-052

## APPENDIX A: REVISION HISTORY

Version	Date	Revision Description
A	1999	This is a new data sheet. However, these devices are similar to the PIC16C72A devices found in the PIC16C62B/72A Data Sheet (DS35008A).

## 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	PIC16C72A	PIC16F872
Pins	28	28
Timers	3	3
Interrupts	8	10
Communication	SSP (SPI, I <sup>2</sup> C Slave)	SSP (SPI, I <sup>2</sup> C Master/Slave)
Frequency	20 MHz	20 MHz
A/D	8-bit	10-bit
CCP	1	1
Program Memory	2K EPROM	2K FLASH
RAM	128 bytes	128 bytes
EEPROM data	None	64 bytes
Other	—	In-Circuit Debugger, Low Voltage Programming

# PIC16F872

---

NOTES:

## INDEX

### A

A/D	85
ADCON0 Register	85
ADCON1 Register	86
ADIF bit	87
Analog Input Model Block Diagram	89
Analog Port Pins	6
Block Diagram	88
Configuring Analog Port Pins	90
Configuring the Interrupt	87
Configuring the Module	87
Conversion Clock	90
Conversions	91
Delays	89
Effects of a Reset	92
GO/DONE bit	87
Internal Sampling Switch (Rss) Impedence	88
Operation During Sleep	92
Sampling Requirements	88
Source Impedence	88
Time Delays	89
Absolute Maximum Ratings	125
ACK	62
Acknowledge Data bit	56
Acknowledge Pulse	62
Acknowledge Sequence Enable bit	56
Acknowledge Status bit	56
ADRES Register	9, 85
Application Note AN578, "Use of the SSP Module in the I2C Multi-Master Environment."	61
Application Notes	
AN552 (Implementing Wake-up on Key Strokes Using PIC16CXXX)	25
AN556 (Table Reading Using PIC16CXX)	20
Architecture	
PIC16F872 Block Diagram	5
Assembler	
MPASM Assembler	119

### B

Banking, Data Memory	7, 12
Baud Rate Generator	68
BCLIF	18
BF	54, 62, 71, 73
Block Diagrams	
A/D	88
Analog Input Model	89
Baud Rate Generator	68
Capture	48
Compare	49
I <sup>2</sup> C Master Mode	66
I <sup>2</sup> C Module	61
PWM	49
SSP (I <sup>2</sup> C Mode)	61
SSP (SPI Mode)	57
Timer0/WDT Prescaler	37
Timer2	45
BRG	68
Brown-out Reset (BOR)	95, 99, 101, 102
BOR Status (BOR Bit)	19
Buffer Full bit, BF	62
Buffer Full Status bit, BF	54
Bus Arbitration	78
Bus Collision During a RESTART Condition	81

Bus Collision During a Start Condition	79
Bus Collision During a Stop Condition	82
Bus Collision Interrupt Flag bit, BCLIF	18
Bus Collision Section	78

### C

Capture/Compare/PWM	
Capture	
Block Diagram	48
CCP1CON Register	47
CCP1IF	48
Mode	48
Prescaler	48
CCP Timer Resources	47
Compare	
Block Diagram	49
Mode	49
Software Interrupt Mode	49
Special Event Trigger	49
Special Trigger Output of CCP1	49
Section	47
Special Event Trigger and A/D Conversions	49
Capture/Compare/PWM (CCP)	
CCP1	
RC2/CCP1 Pin	6
PWM Block Diagram	49
PWM Mode	49
CCP1CON	11
CCP1M0 bit	47
CCP1M1 bit	47
CCP1M2 bit	47
CCP1M3 bit	47
CCP1X bit	47
CCP1Y bit	47
CCPR1H Register	9, 11, 47
CCPR1L Register	11, 47
CKE	54
CKP	55
Clock Polarity Select bit, CKP	55
Code Examples	
Indirect Addressing	20
Code Protection	95, 109
Computed GOTO	20
Configuration Bits	95
Conversion Considerations	149

### D

D/A	54
Data Memory	7
Bank Select (RP1:RP0 Bits)	7, 12
General Purpose Registers	7
Register File Map	8
Special Function Registers	9
Data/Address bit, D/A	54
DC Characteristics	127
Development Support	119
Device Overview	5
Direct Addressing	21

### E

Electrical Characteristics	125
Errata	3

### F

Firmware Instructions	111
FSR Register	9, 10, 11, 20

# PIC16F872

## G

General Call Address Sequence .....	64
General Call Address Support .....	64
General Call Enable bit .....	56

## I

I/O Ports .....	23
I <sup>2</sup> C .....	61
I <sup>2</sup> C Master Mode Reception .....	73
I <sup>2</sup> C Master Mode Restart Condition .....	70
I <sup>2</sup> C Mode Selection .....	61
I <sup>2</sup> C Module	
Acknowledge Sequence timing .....	75
Addressing .....	62
Baud Rate Generator .....	68
Block Diagram .....	66
BRG Block Diagram .....	68
BRG Reset due to SDA Collision .....	80
BRG Timing .....	68
Bus Arbitration .....	78
Bus Collision .....	78
Acknowledge .....	78
Restart Condition .....	81
Restart Condition Timing (Case1) .....	81
Restart Condition Timing (Case2) .....	81
Start Condition .....	79
Start Condition Timing .....	79, 80
Stop Condition .....	82
Stop Condition Timing (Case1) .....	82
Stop Condition Timing (Case2) .....	82
Transmit Timing .....	78
Bus Collision timing .....	78
Clock Arbitration .....	77
Clock Arbitration Timing (Master Transmit) .....	77
Conditions to not give ACK Pulse .....	62
General Call Address Support .....	64
Master Mode .....	66
Master Mode 7-bit Reception timing .....	74
Master Mode Operation .....	67
Master Mode Start Condition .....	69
Master Mode Transmission .....	71
Master Mode Transmit Sequence .....	67
Multi-Master Communication .....	78
Multi-master Mode .....	67
Operation .....	61
Repeat Start Condition timing .....	70
Slave Mode .....	62
Slave Reception .....	63
Slave Transmission .....	63
SSPBUF .....	62
Stop Condition Receive or Transmit timing .....	76
Stop Condition timing .....	76
Waveforms for 7-bit Reception .....	63
Waveforms for 7-bit Transmission .....	64
I <sup>2</sup> C Module Address Register, SSPADD .....	62
I <sup>2</sup> C Slave Mode .....	62
ID Locations .....	95, 109
In-Circuit Serial Programming (ICSP) .....	95, 110
INDF .....	11
INDF Register .....	9, 10, 20
Indirect Addressing .....	20, 21
FSR Register .....	7
Instruction Format .....	111

Instruction Set .....	111
ADDLW .....	113
ADDWF .....	113
ANDLW .....	113
ANDWF .....	113
BCF .....	113
BSF .....	113
BTFSC .....	114
BTFSS .....	114
CALL .....	114
CLRF .....	114
CLRWF .....	114
CLRWDW .....	114
COMF .....	115
DECF .....	115
DECFSZ .....	115
GOTO .....	115
INCF .....	115
INCFSZ .....	115
IORLW .....	116
IORWF .....	116
MOVF .....	116
MOVLW .....	116
MOVWF .....	116
NOP .....	116
RETFIE .....	117
RETLW .....	117
RETURN .....	117
RLF .....	117
RRF .....	117
SLEEP .....	117
SUBLW .....	118
SUBWF .....	118
SWAPF .....	118
XORLW .....	118
XORWF .....	118
Summary Table .....	112
INTCON .....	11
INTCON Register .....	14
GIE Bit .....	14
INTE Bit .....	14
INTF Bit .....	14
PEIE Bit .....	14
RBIE Bit .....	14
RBIF Bit .....	14, 25
TOIE Bit .....	14
TOIF Bit .....	14
Inter-Integrated Circuit (I <sup>2</sup> C) .....	53
Internal Sampling Switch (R <sub>ss</sub> ) Impedance .....	88
Interrupt Sources .....	95, 105
Block Diagram .....	105
Interrupt on Change (RB7:RB4) .....	25
RB0/INT Pin, External .....	6, 106
TMR0 Overflow .....	106
Interrupts	
Bus Collision Interrupt .....	18
Synchronous Serial Port Interrupt .....	16
Interrupts, Context Saving During .....	106
Interrupts, Enable Bits	
Global Interrupt Enable (GIE Bit) .....	14, 105
Interrupt on Change (RB7:RB4) Enable (RBIE Bit) .....	14, 106
Peripheral Interrupt Enable (PEIE Bit) .....	14
RB0/INT Enable (INTE Bit) .....	14
TMR0 Overflow Enable (TOIE Bit) .....	14



Interrupts, Flag Bits	
Interrupt on Change (RB7:RB4) Flag	
(RBIF Bit)	14, 25, 106
RB0/INT Flag (INTF Bit)	14
TMR0 Overflow Flag (TOIF Bit)	14, 106
<b>K</b>	
KeeLoq® Evaluation and Programming Tools	122
<b>L</b>	
Loading of PC	20
<b>M</b>	
Master Clear ( $\overline{\text{MCLR}}$ )	6
$\overline{\text{MCLR}}$ Reset, Normal Operation	99, 101, 102
$\overline{\text{MCLR}}$ Reset, SLEEP	99, 101, 102
Memory Organization	
Data Memory	7
Program Memory	7
MPLAB Integrated Development Environment Software	119
Multi-Master Communication	78
Multi-Master Mode	67
<b>O</b>	
OPCODE Field Descriptions	111
OPTION	11
OPTION_REG Register	13
INTEDG Bit	13
PS2:PS0 Bits	13
PSA Bit	13
$\overline{\text{RBPU}}$ Bit	13
T0CS Bit	13
T0SE Bit	13
OSC1/CLKIN Pin	6
OSC2/CLKOUT Pin	6
Oscillator Configuration	95, 97
HS	97, 101
LP	97, 101
RC	97, 98, 101
XT	97, 101
Oscillator, WDT	107
Output of TMR2	45
<b>P</b>	
P (Stop bit)	54
Packaging	145
Paging, Program Memory	7, 20
PCL Register	9, 10, 11, 20
PCLATH Register	9, 10, 11, 20
PCON Register	11, 19, 100
BOR Bit	19
POR Bit	19
PICDEM-1 Low-Cost PICmicro Demo Board	121
PICDEM-2 Low-Cost PIC16CXX Demo Board	121
PICDEM-3 Low-Cost PIC16CXXX Demo Board	121
PICSTART® Plus Entry Level Development System	121
PIE1 Register	11, 15
PIE2 Register	11, 17
Pinout Descriptions	
PIC16F872	6
PIR1 Register	16
PIR2 Register	18
POP	20
PORTA	6, 11
Analog Port Pins	6
Initialization	23
PORTA Register	23
RA3, RA0 and RA5 Port Pins	23
RA4/T0CKI Pin	6, 23
RA5/ $\overline{\text{SS}}$ /AN4 Pin	6
TRISA Register	23
PORTA Register	9
PORTB	6, 11
PORTB Register	25
Pull-up Enable ( $\overline{\text{RBPU}}$ Bit)	13
RB0/INT Edge Select (INTEDG Bit)	13
RB0/INT Pin, External	6, 106
RB3:RB0 Port Pins	25
RB7:RB4 Interrupt on Change	106
RB7:RB4 Interrupt on Change Enable	
(RBIE Bit)	14, 106
RB7:RB4 Interrupt on Change Flag	
(RBIF Bit)	14, 25, 106
RB7:RB4 Port Pins	25
TRISB Register	25
PORTB Register	9
PORTC	6, 11
Block Diagram	27
PORTC Register	27
RC0/T1OSO/T1CKI Pin	6
RC1/T1OSI Pin	6
RC2/CCP1 Pin	6
RC3/SCK/SCL Pin	6
RC4/SDI/SDA Pin	6
RC5/SDO Pin	6
RC6 Pin	6
RC7 Pin	6
TRISC Register	27
PORTC Register	9
Postscaler, WDT	
Assignment (PSA Bit)	13
Rate Select (PS2:PS0 Bits)	13
Power-on Reset (POR)	95, 99, 100, 101, 102
Oscillator Start-up Timer (OST)	95, 100
POR Status ( $\overline{\text{POR}}$ Bit)	19
Power Control (PCON) Register	100
Power-down (PD Bit)	12, 99
Power-up Timer (PWRT)	95, 100
Time-out ( $\overline{\text{TO}}$ Bit)	12, 99
Time-out Sequence on Power-up	103, 104
PR2	11
PR2 Register	10, 45
Prescaler, Timer0	
Assignment (PSA Bit)	13
Rate Select (PS2:PS0 Bits)	13
PRO MATE® II Universal Programmer	121
Product Identification System	158
Program Counter	
Reset Conditions	101
Program Memory	7
Interrupt Vector	7
Paging	7, 20
Program Memory Map	7
Reset Vector	7
Program Verification	109
Programming Pin (VPP)	6
Programming, Device Instructions	111
PUSH	20

# PIC16F872

## R

R/ $\overline{W}$ .....	54
R/W bit .....	62
R/W bit .....	63
Read/Write bit, R/ $\overline{W}$ .....	54
Receive Enable bit .....	56
Receive Overflow Indicator bit, SSPOV .....	55
Register File .....	7
Register File Map .....	8
Registers	
FSR Summary .....	11
INDF Summary .....	11
INTCON Summary .....	11
OPTION Summary .....	11
PCL Summary .....	11
PCLATH Summary .....	11
PORTB Summary .....	11
SSPSTAT .....	54
STATUS Summary .....	11
Summary .....	9
TMR0 Summary .....	11
TRISB Summary .....	11
Reset .....	95, 99
Block Diagram .....	99
Reset Conditions for All Registers .....	102
Reset Conditions for PCON Register .....	101
Reset Conditions for Program Counter .....	101
Reset Conditions for STATUS Register .....	101
Restart Condition Enabled bit .....	56
Revision History .....	149

## S

S (Start bit) .....	54
SCK .....	57
SCL .....	62
SDA .....	62
SDI .....	57
SDO .....	57
SEEVAL® Evaluation and Programming System .....	122
Serial Clock, SCK .....	57
Serial Clock, SCL .....	62
Serial Data Address, SDA .....	62
Serial Data In, SDI .....	57
Serial Data Out, SDO .....	57
Slave Select, $\overline{SS}$ .....	57
SLEEP .....	95, 99, 108
SMP .....	54
Software Simulator (MPLAB-SIM) .....	120
Special Features of the CPU .....	95
Special Function Registers .....	9
Speed, Operating .....	1
SPI	
Master Mode .....	58
Master Mode Timing .....	58
Serial Clock .....	57
Serial Data In .....	57
Serial Data Out .....	57
Serial Peripheral Interface (SPI) .....	53
Slave Mode Timing .....	59
Slave Mode Timing Diagram .....	59
Slave Select .....	57
SPI clock .....	58
SPI Mode .....	57
SPI Clock Edge Select, CKE .....	54
SPI Data Input Sample Phase Select, SMP .....	54

## SPI Module

Slave Mode .....	59
$\overline{SS}$ .....	57
SSP .....	53
Block Diagram (SPI Mode) .....	57
RA5/ $\overline{SS}$ /AN4 Pin .....	6
RC3/SCK/SCL Pin .....	6
RC4/SDI/SDA Pin .....	6
RC5/SDO Pin .....	6
SPI Mode .....	57
SSPADD .....	62
SSPBUF .....	58, 62
SSPCON1 .....	55
SSPCON2 .....	56
SSPSR .....	58, 62
SSPSTAT .....	54, 62
SSP I <sup>2</sup> C	
SSP I <sup>2</sup> C Operation .....	61
SSP Module	
SPI Master Mode .....	58
SPI Slave Mode .....	59
SSPCON1 Register .....	61
SSP Overflow Detect bit, SSPOV .....	62
SSPADD Register .....	10, 11
SSPBUF .....	11, 62
SSPBUF Register .....	9
SSPCON Register .....	9
SSPCON1 .....	55, 61
SSPCON2 .....	56
SSPEN .....	55
SSPIF .....	16, 63
SSPM3:SSPM0 .....	55
SSPOV .....	55, 62, 73
SSPSTAT .....	11, 54, 62
SSPSTAT Register .....	10
Stack .....	20
Overflows .....	20
Underflow .....	20
Start bit (S) .....	54
Start Condition Enabled bit .....	56
STATUS Register .....	11, 12
C Bit .....	12
DC Bit .....	12
IRP Bit .....	12
$\overline{PD}$ Bit .....	12, 99
RP1:RP0 Bits .....	12
$\overline{TO}$ Bit .....	12, 99
Z Bit .....	12
Stop bit (P) .....	54
Stop Condition Enable bit .....	56
Synchronous Serial Port .....	53
Synchronous Serial Port Enable bit, SSPEN .....	55
Synchronous Serial Port Interrupt .....	16
Synchronous Serial Port Mode Select bits, SSPM3:SSPM0 .....	55
T	
T1CKPS0 bit .....	41
T1CKPS1 bit .....	41
T1CON .....	11
T1CON Register .....	11, 41
T1OSCEN bit .....	41
$\overline{T1SYNC}$ bit .....	41
T2CKPS0 bit .....	45
T2CKPS1 bit .....	45
T2CON Register .....	11, 45

TAD .....	90	Timer0 .....	135
Timer0		Timer1 .....	135
Clock Source Edge Select (T0SE Bit) .....	13	Wake-up from SLEEP via Interrupt .....	109
Clock Source Select (T0CS Bit) .....	13	Watchdog Timer .....	134
Overflow Enable (T0IE Bit) .....	14	TMR0 .....	11
Overflow Flag (T0IF Bit) .....	14, 106	TMR0 Register .....	9
Overflow Interrupt .....	106	TMR1CS bit .....	41
RA4/T0CKI Pin, External Clock .....	6	TMR1H .....	11
Timer1 .....	41	TMR1H Register .....	9
RC0/T1OSO/T1CKI Pin .....	6	TMR1L .....	11
RC1/T1OSI Pin .....	6	TMR1L Register .....	9
Timers		TMR1ON bit .....	41
Timer0		TMR2 .....	11
External Clock .....	38	TMR2 Register .....	9
Interrupt .....	37	TMR2ON bit .....	45
Prescaler .....	38	TOUTPS0 bit .....	45
Prescaler Block Diagram .....	37	TOUTPS1 bit .....	45
Section .....	37	TOUTPS2 bit .....	45
T0CKI .....	38	TOUTPS3 bit .....	45
Timer1		TRISA .....	11
Asynchronous Counter Mode .....	43	TRISA Register .....	10
Capacitor Selection .....	43	TRISB .....	11
Operation in Timer Mode .....	42	TRISB Register .....	10
Oscillator .....	43	TRISC .....	11
Prescaler .....	43	TRISC Register .....	10
Resetting of Timer1 Registers .....	43		
Resetting Timer1 using a CCP Trigger Output ..	43	<b>U</b>	
Synchronized Counter Mode .....	42	UA .....	54
T1CON .....	41	Update Address, UA .....	54
TMR1H .....	43	<b>W</b>	
TMR1L .....	43	Wake-up from SLEEP .....	95, 108
Timer2		Interrupts .....	101, 102
Block Diagram .....	45	MCLR Reset .....	102
Postscaler .....	45	Timing Diagram .....	109
Prescaler .....	45	WDT Reset .....	102
T2CON .....	45	Watchdog Timer (WDT) .....	95, 107
Timing Diagrams		Block Diagram .....	107
A/D Conversion .....	142	Enable (WDTE Bit) .....	107
Acknowledge Sequence Timing .....	75	Programming Considerations .....	107
Baud Rate Generator with Clock Arbitration .....	68	RC Oscillator .....	107
BRG Reset Due to SDA Collision .....	80	Time-out Period .....	107
Brown-out Reset .....	134	WDT Reset, Normal Operation .....	99, 101, 102
Bus Collision		WDT Reset, SLEEP .....	99, 101, 102
Start Condition Timing .....	79	Waveform for General Call Address Sequence .....	64
Bus Collision During a Restart Condition (Case 1) ....	81	WCOL .....	55, 69, 71, 73, 75, 76
Bus Collision During a Restart Condition (Case2) ....	81	WCOL Status Flag .....	69
Bus Collision During a Start Condition (SCL = 0) .....	80	Write Collision Detect bit, WCOL .....	55
Bus Collision During a Stop Condition .....	82	WWW, On-Line Support .....	3
Bus Collision for Transmit and Acknowledge .....	78		
Capture/Compare/PWM .....	136		
CLKOUT and I/O .....	133		
I <sup>2</sup> C Bus Data .....	140		
I <sup>2</sup> C Bus Start/Stop bits .....	139		
I <sup>2</sup> C Master Mode First Start bit timing .....	69		
I <sup>2</sup> C Master Mode Reception timing .....	74		
I <sup>2</sup> C Master Mode Transmission timing .....	72		
Master Mode Transmit Clock Arbitration .....	77		
Power-up Timer .....	134		
Repeat Start Condition .....	70		
Reset .....	134		
SPI Master Mode .....	58		
SPI Slave Mode (CKE = 1) .....	59		
SPI Slave Mode Timing (CKE = 0) .....	59		
Start-up Timer .....	134		
Stop Condition Receive or Transmit .....	76		
Time-out Sequence on Power-up .....	103, 104		

# PIC16F872

---

NOTES:

## ON-LINE SUPPORT

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

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

### Connecting to the Microchip Internet Web Site

The Microchip web site is available by using your favorite Internet browser to attach to:

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

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

**<ftp://ftp.microchip.com>**

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

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

## Systems Information and Upgrade Hot Line

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

1-800-755-2345 for U.S. and most of Canada, and  
1-480-786-7302 for the rest of the world.

981103

**Trademarks:** The Microchip name, logo, PIC, PICmicro, PICSTART, PICMASTER, PRO MATE and MPLAB are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries. *FlexROM* and *fuzzyLAB* are trademarks and SQTP is a service mark of Microchip in the U.S.A.

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

# PIC16F872

---

## 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) 786-7578.

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

To: Technical Publications Manager Total Pages Sent  
RE: Reader Response  
From: Name \_\_\_\_\_  
Company \_\_\_\_\_  
Address \_\_\_\_\_  
City / State / ZIP / Country \_\_\_\_\_  
Telephone: (\_\_\_\_\_) \_\_\_\_\_ - \_\_\_\_\_ FAX: (\_\_\_\_\_) \_\_\_\_\_ - \_\_\_\_\_

Application (optional):

Would you like a reply? \_\_\_Y \_\_\_N

Device: **PIC16F872** Literature Number: **DS30221A**

Questions:

1. What are the best features of this document?

---

---

2. How does this document meet your hardware and software development needs?

---

---

3. Do you find the organization of this data sheet easy to follow? If not, why?

---

---

4. What additions to the data sheet do you think would enhance the structure and subject?

---

---

5. What deletions from the data sheet could be made without affecting the overall usefulness?

---

---

6. Is there any incorrect or misleading information (what and where)?

---

---

7. How would you improve this document?

---

---

8. How would you improve our software, systems, and silicon products?

---

---

## PIC16F872 PRODUCT IDENTIFICATION SYSTEM

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

PART NO.	-X	/XX	XXX
Device	Temperature Range	Package	Pattern
<p>Device</p> <p>PIC16F872, PIC16F872T; V<sub>DD</sub> range 4.0V to 5.5V            PIC16LF872, PIC16LF872T; V<sub>DD</sub> range 2.0V to 5.5V</p> <p>F = CMOS FLASH            LF = Low Power CMOS FLASH            T = in tape and reel - SOIC, SSOP, packages only.</p>	<p>Temperature Range</p> <p>blank = 0°C to 70°C (Commercial)            I = -40°C to +85°C (Industrial)</p>	<p>Package</p> <p>SO = SOIC            SP = Skinny plastic dip            SS = SSOP</p>	<p>Pattern</p> <p>QTP, SQTP, Code or Special Requirements            (blank otherwise)</p>
<p><b>Examples:</b></p> <p>a) PIC16F872-I/SP 301 = Industrial temp., PDIP package, 20MHz, normal V<sub>DD</sub> limits, QTP pattern #301.</p> <p>b) PIC16F872-I/SO = Industrial temp., SOIC package, 20 MHz, normal V<sub>DD</sub> limits.</p> <p>c) PIC16F872/P = Industrial temp., PDIP package, 10MHz, normal V<sub>DD</sub> limits.</p> <p>d) PIC16LF872-I/SS = Industrial temp., SSOP package, DC - 20MHz, extended V<sub>DD</sub> limits.</p>			

## Sales and Support

### Data Sheets

Products supported by a preliminary Data Sheet may have an errata sheet describing minor operational differences and recommended workarounds. To determine if an errata sheet exists for a particular device, please contact one of the following:

1. Your local Microchip sales office
2. The Microchip Corporate Literature Center U.S. FAX: (480) 786-7277
3. The Microchip Worldwide Site ([www.microchip.com](http://www.microchip.com))

Please specify which device, revision of silicon and Data Sheet (include Literature #) you are using.

### New Customer Notification System

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



## WORLDWIDE SALES AND SERVICE

### AMERICAS

#### Corporate Office

Microchip Technology Inc.  
2355 West Chandler Blvd.  
Chandler, AZ 85224-6199  
Tel: 480-786-7200 Fax: 480-786-7277  
Technical Support: 480-786-7627  
Web Address: <http://www.microchip.com>

#### Atlanta

Microchip Technology Inc.  
500 Sugar Mill Road, Suite 200B  
Atlanta, GA 30350  
Tel: 770-640-0034 Fax: 770-640-0307

#### Boston

Microchip Technology Inc.  
5 Mount Royal Avenue  
Marlborough, MA 01752  
Tel: 508-480-9990 Fax: 508-480-8575

#### Chicago

Microchip Technology Inc.  
333 Pierce Road, Suite 180  
Itasca, IL 60143  
Tel: 630-285-0071 Fax: 630-285-0075

#### Dallas

Microchip Technology Inc.  
4570 Westgrove Drive, Suite 160  
Addison, TX 75248  
Tel: 972-818-7423 Fax: 972-818-2924

#### Dayton

Microchip Technology Inc.  
Two Prestige Place, Suite 150  
Miamisburg, OH 45342  
Tel: 937-291-1654 Fax: 937-291-9175

#### Detroit

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

#### Los Angeles

Microchip Technology Inc.  
18201 Von Karman, Suite 1090  
Irvine, CA 92612  
Tel: 949-263-1888 Fax: 949-263-1338

#### New York

Microchip Technology Inc.  
150 Motor Parkway, Suite 202  
Hauppauge, NY 11788  
Tel: 631-273-5305 Fax: 631-273-5335

#### San Jose

Microchip Technology Inc.  
1201 North First Street, Suite 590  
San Jose, CA 95131  
Tel: 408-436-7950 Fax: 408-436-7955

### AMERICAS (continued)

#### Toronto

Microchip Technology Inc.  
5925 Airport Road, Suite 200  
Mississauga, Ontario L4V 1W1, Canada  
Tel: 905-405-6279 Fax: 905-405-6253

### ASIA/PACIFIC

#### Hong Kong

Microchip Asia Pacific  
Unit 2101, Tower 2  
Metroplaza  
223 Hing Fong Road  
Kwai Fong, N.T., Hong Kong  
Tel: 852-2-401-1200 Fax: 852-2-401-3431

#### Beijing

Microchip Technology, Beijing  
Unit 915, 6 Chaoyangmen Bei Dajie  
Dong Erhuan Road, Dongcheng District  
New China Hong Kong Manhattan Building  
Beijing 100027 PRC  
Tel: 86-10-85282100 Fax: 86-10-85282104

#### India

Microchip Technology Inc.  
India Liaison Office  
No. 6, Legacy, Convent Road  
Bangalore 560 025, India  
Tel: 91-80-229-0061 Fax: 91-80-229-0062

#### Japan

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

#### Korea

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

#### Shanghai

Microchip Technology  
RM 406 Shanghai Golden Bridge Bldg.  
2077 Yan'an Road West, Hong Qiao District  
Shanghai, PRC 200335  
Tel: 86-21-6275-5700 Fax: 86 21-6275-5060

### ASIA/PACIFIC (continued)

#### Singapore

Microchip Technology Singapore Pte Ltd.  
200 Middle Road  
#07-02 Prime Centre  
Singapore 188980  
Tel: 65-334-8870 Fax: 65-334-8850

#### Taiwan, R.O.C

Microchip Technology Taiwan  
10F-1C 207  
Tung Hua North Road  
Taipei, Taiwan, ROC  
Tel: 886-2-2717-7175 Fax: 886-2-2545-0139

### EUROPE

#### United Kingdom

Arizona Microchip Technology Ltd.  
505 Eskdale Road  
Wokingham  
Berkshire, England RG41 5TU  
Tel: 44 118 921 5858 Fax: 44-118 921-5835

#### Denmark

Microchip Technology Denmark ApS  
Regus Business Centre  
Lautrup hof 1-3  
Ballerup DK-2750 Denmark  
Tel: 45 4420 9895 Fax: 45 4420 9910

#### France

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

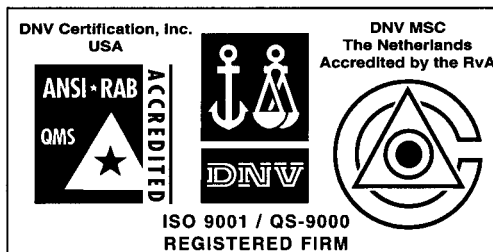
#### Germany

Arizona Microchip Technology GmbH  
Gustav-Heinemann-Ring 125  
D-81739 München, Germany  
Tel: 49-89-627-144 0 Fax: 49-89-627-144-44

#### Italy

Arizona Microchip Technology SRL  
Centro Direzionale Colleoni  
Palazzo Taurus 1 V. Le Colleoni 1  
20041 Agrate Brianza  
Milan, Italy  
Tel: 39-039-65791-1 Fax: 39-039-6899883

11/15/99



Microchip received QS-9000 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona in July 1999. The Company's quality system processes and procedures are QS-9000 compliant for its PICmicro® 8-bit MCUs, KEELOC® code hopping devices, Serial EEPROMs and microperipheral products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001 certified.

All rights reserved. © 1999 Microchip Technology Incorporated. Printed in the USA. 12/99 Printed on recycled paper.

Information contained in this publication regarding device applications and the like is intended for suggestion only and may be superseded by updates. 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. The Microchip logo and name are registered trademarks of Microchip Technology Inc. in the U.S.A. and other countries. All rights reserved. All other trademarks mentioned herein are the property of their respective companies.