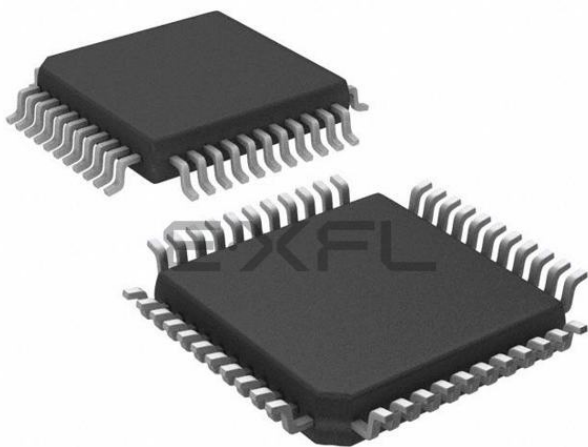




Welcome to [E-XFL.COM](#)

What is "[Embedded - Microcontrollers](#)"?

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

Applications of "[Embedded - Microcontrollers](#)"

Details

Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	4MHz
Connectivity	I ² C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	33
Program Memory Size	7KB (4K x 14)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	192 x 8
Voltage - Supply (Vcc/Vdd)	4V ~ 6V
Data Converters	A/D 8x8b
Oscillator Type	External
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	44-QFP
Supplier Device Package	44-MQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16c74a-04-pq

TABLE 4-2: PIC16C73/73A/74/74A 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)
Bank 0											
00h ⁽⁴⁾	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 ⁽⁴⁾	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	0000 0000
03h ⁽⁴⁾	STATUS	IRP ⁽⁷⁾	RP1 ⁽⁷⁾	RP0	T $\overline{O}$	P $\overline{D}$	Z	DC	C	0001 1xxx	000q quuu
04h ⁽⁴⁾	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 ⁽⁵⁾	PORTD	PORTD Data Latch when written: PORTD pins when read								xxxx xxxx	uuuu uuuu
09h ⁽⁵⁾	PORTE	—	—	—	—	—	RE2	RE1	RE0	---- -xxx	---- -uuu
0Ah ^(1,4)	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	---0 0000
0Bh ⁽⁴⁾	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF ⁽³⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
0Dh	PIR2	—	—	—	—	—	—	—	CCP2IF	---- --0	---- --0
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	T1SYNCR	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	RCSTA	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000 -00x	0000 -00x
19h	TXREG	USART Transmit Data Register								0000 0000	0000 0000
1Ah	RCREG	USART Receive Data Register								0000 0000	0000 0000
1Bh	CCPR2L	Capture/Compare/PWM Register2 (LSB)								xxxx xxxx	uuuu uuuu
1Ch	CCPR2H	Capture/Compare/PWM Register2 (MSB)								xxxx xxxx	uuuu uuuu
1Dh	CCP2CON	—	—	CCP2X	CCP2Y	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000	--00 0000
1Eh	ADRES	A/D Result Register								xxxx xxxx	uuuu uuuu
1Fh	ADCON0	ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	—	ADON	0000 00-0	0000 00-0

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented read as '0'.
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: Bits PSPIE and PSPIF are reserved on the PIC16C73/73A, always maintain these bits clear.
- 4: These registers can be addressed from either bank.
- 5: PORTD and PORTE are not physically implemented on the PIC16C73/73A, read as '0'.
- 6: Brown-out Reset is not implemented on the PIC16C73 or the PIC16C74, read as '0'.
- 7: The IRP and RP1 bits are reserved on the PIC16C73/73A/74/74A, always maintain these bits clear.

TABLE 4-3: PIC16C76/77 SPECIAL FUNCTION REGISTER SUMMARY (Cont.'d)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other resets (2)
Bank 2											
100h ⁽⁴⁾	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 ⁽⁴⁾	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	0000 0000
103h ⁽⁴⁾	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	000q quuu
104h ⁽⁴⁾	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 ^(1,4)	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					-- -0 0000	-- -0 0000
10Bh ⁽⁴⁾	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
10Ch-10Fh	—	Unimplemented								—	—
Bank 3											
180h ⁽⁴⁾	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	0000 0000
181h	OPTION	$\overline{RBP}\overline{U}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
182h ⁽⁴⁾	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	0000 0000
183h ⁽⁴⁾	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	000q quuu
184h ⁽⁴⁾	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 ^(1,4)	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					-- -0 0000	-- -0 0000
18Bh ⁽⁴⁾	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
18Ch-18Fh	—	Unimplemented								—	—

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented read as '0'.
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: Bits PSPIE and PSPIF are reserved on the PIC16C76, always maintain these bits clear.
- 4: These registers can be addressed from any bank.
- 5: PORTD and PORTE are not physically implemented on the PIC16C76, read as '0'.

PIC16C7X

4.2.2.3 INTCON REGISTER

Applicable Devices

72	73	73A	74	74A	76	77
----	----	-----	----	-----	----	----

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>).

FIGURE 4-9: 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-0	R/W-x
GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF
bit7							bit0
bit 7: GIE:⁽¹⁾ 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 RB7:RB4 pins changed state (must be cleared in software) 0 = None of the RB7:RB4 pins have changed state Note 1: For the PIC16C73 and PIC16C74, if an interrupt occurs while the GIE bit is being cleared, the GIE bit may be unintentionally re-enabled by the RETFIE instruction in the user's Interrupt Service Routine. Refer to Section 14.5 for a detailed description.							
R = Readable bit W = Writable bit U = Unimplemented bit, read as '0' - n = Value at POR reset							
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.							

Applicable Devices						
72	73	73A	74	74A	76	77

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

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

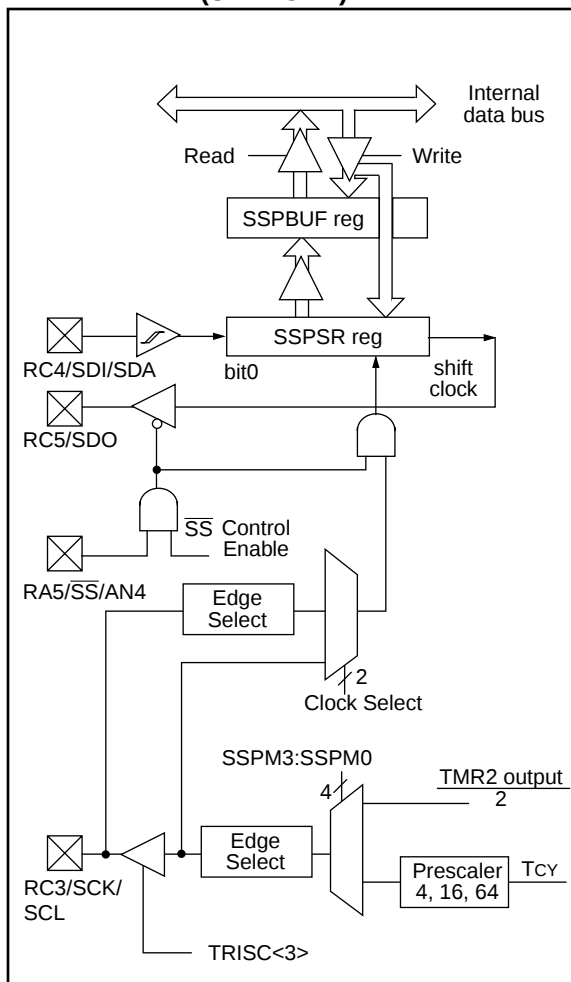
- Master Mode (SCK is the clock output)
- Slave Mode (SCK is the clock input)
- Clock Polarity (Output/Input data on the Rising/Falling edge of SCK)
- Clock Rate (Master mode only)
- Slave Select Mode (Slave mode only)

EXAMPLE 11-1: LOADING THE SSPBUF (SSPSR) REGISTER

	BSF	STATUS, RP0	;Specify Bank 1
LOOP	BTFSS	SSPSTAT, BF	;Has data been
			;received
			;(transmit
			;complete)?
	GOTO	LOOP	;No
	BCF	STATUS, RP0	;Specify Bank 0
	MOVF	SSPBUF, W	;W reg = contents
			;of SSPBUF
	MOVWF	RXDATA	;Save in user RAM
	MOVF	TXDATA, W	;W reg = contents
			; of TXDATA
	MOVWF	SSPBUF	;New data to xmit

The block diagram of the SSP module, when in SPI mode (Figure 11-3), shows that the SSPSR register is not directly readable or writable, and can only be accessed from addressing the SSPBUF register. Additionally, the SSP status register (SSPSTAT) indicates the various status conditions.

FIGURE 11-3: SSP BLOCK DIAGRAM (SPI MODE)



13.2 Selecting the A/D Conversion Clock

Applicable Devices						
72	73	73A	74	74A	76	77

The A/D conversion time per bit is defined as T_{AD} . The A/D conversion requires $9.5T_{AD}$ per 8-bit conversion. The source of the A/D conversion clock is software selectable. The four possible options for T_{AD} are:

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

For correct A/D conversions, the A/D conversion clock (T_{AD}) must be selected to ensure a minimum T_{AD} time of 1.6 μ s.

Table 13-1 shows the resultant T_{AD} times derived from the device operating frequencies and the A/D clock source selected.

13.3 Configuring Analog Port Pins

Applicable Devices						
72	73	73A	74	74A	76	77

The ADCON1, TRISA, and TRISE 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 (V_{OH} or V_{OL}) will be converted.

The A/D operation is independent of the state of the CHS2:CHS0 bits and the TRIS bits.

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

Note 2: Analog levels on any pin that is defined as a digital input (including the AN7:AN0 pins), may cause the input buffer to consume current that is out of the devices specification.

TABLE 13-1: T_{AD} vs. DEVICE OPERATING FREQUENCIES

AD Clock Source (T_{AD})		Device Frequency			
Operation	ADCS1:ADCS0	20 MHz	5 MHz	1.25 MHz	333.33 kHz
2Tosc	00	100 ns ⁽²⁾	400 ns ⁽²⁾	1.6 μ s	6 μ s
8Tosc	01	400 ns ⁽²⁾	1.6 μ s	6.4 μ s	24 μ s ⁽³⁾
32Tosc	10	1.6 μ s	6.4 μ s	25.6 μ s ⁽³⁾	96 μ s ⁽³⁾
RC ⁽⁵⁾	11	2 - 6 μ s ^(1,4)	2 - 6 μ s ^(1,4)	2 - 6 μ s ^(1,4)	2 - 6 μ s ⁽¹⁾

Legend: Shaded cells are outside of recommended range.

Note 1: The RC source has a typical T_{AD} time of 4 μ s.

2: These values violate the minimum required T_{AD} time.

3: For faster conversion times, the selection of another clock source is recommended.

4: When device frequency is greater than 1 MHz, the RC A/D conversion clock source is recommended for sleep operation only.

5: For extended voltage devices (LC), please refer to Electrical Specifications section.

13.8 Use of the CCP Trigger

Applicable Devices					
72	73	73A	74	74A	76 77

Note: In the PIC16C72, the "special event trigger" is implemented in the CCP1 module.

An A/D conversion can be started by the "special event trigger" of the CCP2 module (CCP1 on the PIC16C72 only). This requires that the CCP2M3:CCP2M0 bits (CCP2CON<3:0>) be programmed as 1011 and that the A/D module is enabled (ADON bit is set). When the trigger occurs, the GO/DONE bit will be set, starting the A/D conversion, and the Timer1 counter will be reset to zero. Timer1 is reset to automatically repeat the A/D acquisition period with minimal software overhead (moving the ADRES to the desired location). The appropriate analog input channel must be selected and the minimum acquisition done before the "special event trigger" sets the GO/DONE bit (starts a conversion).

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

13.9 Connection Considerations

Applicable Devices					
72	73	73A	74	74A	76 77

If the input voltage exceeds the rail values (V_{SS} or V_{DD}) by greater than 0.2V, then the accuracy of the conversion is out of specification.

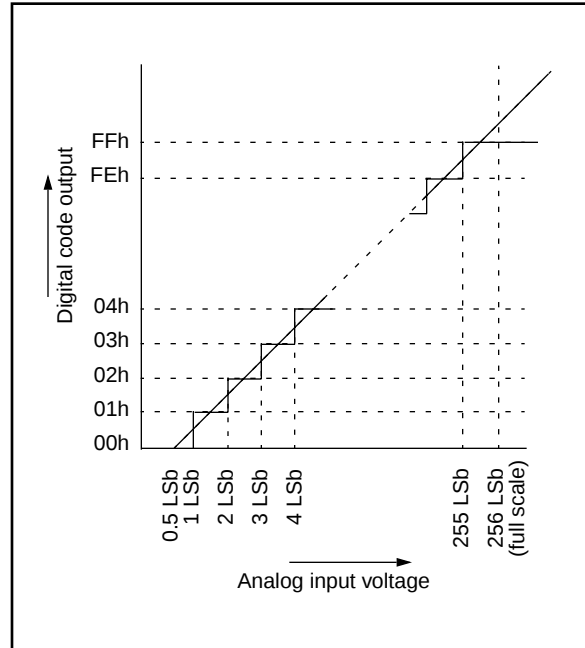
An external RC filter is sometimes added for anti-aliasing of the input signal. The R component should be selected to ensure that the total source impedance is kept under the 10 k Ω recommended specification. Any external components connected (via hi-impedance) to an analog input pin (capacitor, zener diode, etc.) should have very little leakage current at the pin.

13.10 Transfer Function

Applicable Devices					
72	73	73A	74	74A	76 77

The ideal transfer function of the A/D converter is as follows: the first transition occurs when the analog input voltage (V_{AIN}) is Analog $V_{REF}/256$ (Figure 13-5).

FIGURE 13-5: A/D TRANSFER FUNCTION



13.11 References

A very good reference for understanding A/D converters is the "Analog-Digital Conversion Handbook" third edition, published by Prentice Hall (ISBN 0-13-03-2848-0).

14.2.3 EXTERNAL CRYSTAL OSCILLATOR CIRCUIT

Either a prepackaged oscillator can be used or a simple oscillator circuit with TTL gates can be built. Prepackaged oscillators provide a wide operating range and better stability. A well-designed crystal oscillator will provide good performance with TTL gates. Two types of crystal oscillator circuits can be used; one with series resonance, or one with parallel resonance.

Figure 14-5 shows implementation of a parallel resonant oscillator circuit. The circuit is designed to use the fundamental frequency of the crystal. The 74AS04 inverter performs the 180-degree phase shift that a parallel oscillator requires. The 4.7 k Ω resistor provides the negative feedback for stability. The 10 k Ω potentiometer biases the 74AS04 in the linear region. This could be used for external oscillator designs.

FIGURE 14-5: EXTERNAL PARALLEL RESONANT CRYSTAL OSCILLATOR CIRCUIT

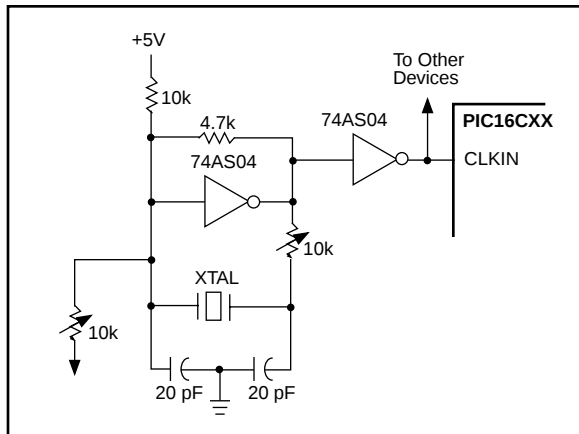
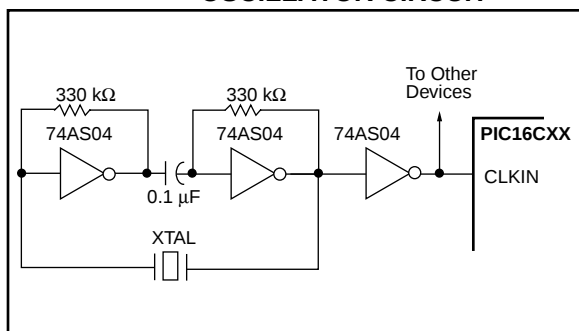


Figure 14-6 shows a series resonant oscillator circuit. This circuit is also designed to use the fundamental frequency of the crystal. The inverter performs a 180-degree phase shift in a series resonant oscillator circuit. The 330 k Ω resistors provide the negative feedback to bias the inverters in their linear region.

FIGURE 14-6: EXTERNAL SERIES RESONANT CRYSTAL OSCILLATOR CIRCUIT



14.2.4 RC OSCILLATOR

For timing insensitive applications the "RC" device option offers additional cost savings. The RC oscillator frequency is a function of the supply voltage, the resistor (R_{ext}) and capacitor (C_{ext}) values, and the operating temperature. In addition to this, the oscillator frequency will vary from unit to unit due to normal process parameter variation. Furthermore, the difference in lead frame capacitance between package types will also affect the oscillation frequency, especially for low C_{ext} values. The user also needs to take into account variation due to tolerance of external R and C components used. Figure 14-7 shows how the R/C combination is connected to the PIC16CXX. For R_{ext} values below 2.2 k Ω , the oscillator operation may become unstable, or stop completely. For very high R_{ext} values (e.g. 1 M Ω), the oscillator becomes sensitive to noise, humidity and leakage. Thus, we recommend to keep R_{ext} between 3 k Ω and 100 k Ω .

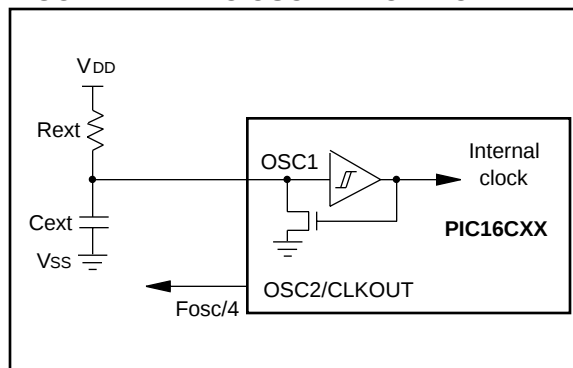
Although the oscillator will operate with no external capacitor ($C_{ext} = 0$ pF), we recommend using values above 20 pF for noise and stability reasons. With no or small external capacitance, the oscillation frequency can vary dramatically due to changes in external capacitances, such as PCB trace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

See characterization data for desired device for RC frequency variation from part to part due to normal process variation. The variation is larger for larger R (since leakage current variation will affect RC frequency more for large R) and for smaller C (since variation of input capacitance will affect RC frequency more).

See characterization data for desired device for variation of oscillator frequency due to V_{DD} for given R_{ext}/C_{ext} values as well as frequency variation due to operating temperature for given R, C, and V_{DD} values.

The oscillator frequency, divided by 4, is available on the OSC2/CLKOUT pin, and can be used for test purposes or to synchronize other logic (see Figure 3-4 for waveform).

FIGURE 14-7: RC OSCILLATOR MODE



14.4.5 TIME-OUT SEQUENCE

On power-up the time-out sequence is as follows: First PWRT time-out is invoked after the POR time delay has expired. Then OST is activated. The total time-out will vary based on oscillator configuration and the status of the PWRT. For example, in RC mode with the PWRT disabled, there will be no time-out at all. Figure 14-10, Figure 14-11, and Figure 14-12 depict time-out sequences on power-up.

Since the time-outs occur from the POR pulse, if $\overline{\text{MCLR}}$ is kept low long enough, the time-outs will expire. Then bringing MCLR high will begin execution immediately (Figure 14-11). This is useful for testing purposes or to synchronize more than one PIC16CXX device operating in parallel.

Table 14-7 shows the reset conditions for some special function registers, while Table 14-8 shows the reset conditions for all the registers.

14.4.6 POWER CONTROL/STATUS REGISTER (PCON)

Applicable Devices					
72	73	73A	74	74A	76/77

The Power Control/Status Register, PCON has up to two bits, depending upon the device. Bit0 is not implemented on the PIC16C73 or PIC16C74.

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 14-3: TIME-OUT IN VARIOUS SITUATIONS, PIC16C73/74

Oscillator Configuration	Power-up		Wake-up from SLEEP
	PWRTE = 1	PWRTE = 0	
XT, HS, LP	72 ms + 1024Tosc	1024Tosc	1024 Tosc
RC	72 ms	—	—

TABLE 14-4: TIME-OUT IN VARIOUS SITUATIONS, PIC16C72/73A/74A/76/77

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 14-5: STATUS BITS AND THEIR SIGNIFICANCE, PIC16C73/74

$\overline{\text{POR}}$	$\overline{\text{TO}}$	$\overline{\text{PD}}$	
0	1	1	Power-on Reset
0	0	x	Illegal, $\overline{\text{TO}}$ is set on $\overline{\text{POR}}$
0	x	0	Illegal, $\overline{\text{PD}}$ is set on $\overline{\text{POR}}$
1	0	1	WDT Reset
1	0	0	WDT Wake-up
1	u	u	MCLR Reset during normal operation
1	1	0	MCLR Reset during SLEEP or interrupt wake-up from SLEEP

Legend: u = unchanged, x = unknown

14.5 Interrupts

Applicable Devices							
72	73	73A	74	74A	76	77	

The PIC16C7X family has up to 12 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 (Figure 14-17). 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.

Note: For the PIC16C73/74, if an interrupt occurs while the Global Interrupt Enable (GIE) bit is being cleared, the GIE bit may unintentionally be re-enabled by the user's Interrupt Service Routine (the RETFIE instruction). The events that would cause this to occur are:

1. An instruction clears the GIE bit while an interrupt is acknowledged.
2. The program branches to the Interrupt vector and executes the Interrupt Service Routine.
3. The Interrupt Service Routine completes with the execution of the RETFIE instruction. This causes the GIE bit to be set (enables interrupts), and the program returns to the instruction after the one which was meant to disable interrupts.

Perform the following to ensure that interrupts are globally disabled:

```

LOOP BCF    INTCON, GIE    ; Disable global
                               ; interrupt bit
      BTFSC INTCON, GIE    ; Global interrupt
                               ; disabled?
      GOTO  LOOP           ; NO, try again
      :                   ; Yes, continue
                               ; with program
                               ; flow

```

15.1 Instruction Descriptions

ADDLW Add Literal and W

Syntax:	[<i>label</i>] ADDLW <i>k</i>			
Operands:	$0 \leq k \leq 255$			
Operation:	$(W) + k \rightarrow (W)$			
Status Affected:	C, DC, Z			
Encoding:	11	111x	kkkk	kkkk
Description:	The contents of the W register are added to the eight bit literal 'k' and the result is placed in the W register.			
Words:	1			
Cycles:	1			
Q Cycle Activity:	Q1	Q2	Q3	Q4
	Decode	Read literal 'k'	Process data	Write to W

Example: ADDLW 0x15
 Before Instruction
 W = 0x10
 After Instruction
 W = 0x25

ADDWF Add W and f

Syntax:	[<i>label</i>] ADDWF <i>f</i> , <i>d</i>			
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$			
Operation:	$(W) + (f) \rightarrow (\text{destination})$			
Status Affected:	C, DC, Z			
Encoding:	00	0111	dfff	ffff
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'.			
Words:	1			
Cycles:	1			
Q Cycle Activity:	Q1	Q2	Q3	Q4
	Decode	Read register 'f'	Process data	Write to destination

Example ADDWF FSR, 0
 Before Instruction
 W = 0x17
 FSR = 0xC2
 After Instruction
 W = 0xD9
 FSR = 0xC2

ANDLW AND Literal with W

Syntax:	[<i>label</i>] ANDLW <i>k</i>			
Operands:	$0 \leq k \leq 255$			
Operation:	$(W) .\text{AND.} (k) \rightarrow (W)$			
Status Affected:	Z			
Encoding:	11	1001	kkkk	kkkk
Description:	The contents of W register are AND'ed with the eight bit literal 'k'. The result is placed in the W register.			
Words:	1			
Cycles:	1			
Q Cycle Activity:	Q1	Q2	Q3	Q4
	Decode	Read literal "k"	Process data	Write to W

Example ANDLW 0x5F
 Before Instruction
 W = 0xA3
 After Instruction
 W = 0x03

ANDWF AND W with f

Syntax:	[<i>label</i>] ANDWF <i>f</i> , <i>d</i>			
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$			
Operation:	$(W) .\text{AND.} (f) \rightarrow (\text{destination})$			
Status Affected:	Z			
Encoding:	00	0101	dfff	ffff
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'.			
Words:	1			
Cycles:	1			
Q Cycle Activity:	Q1	Q2	Q3	Q4
	Decode	Read register 'f'	Process data	Write to destination

Example ANDWF FSR, 1
 Before Instruction
 W = 0x17
 FSR = 0xC2
 After Instruction
 W = 0x17
 FSR = 0x02

BTFSS		Bit Test f, Skip if Set			
Syntax:	[label] BTFSS f,b				
Operands:	0 ≤ f ≤ 127 0 ≤ b < 7				
Operation:	skip if (f) = 1				
Status Affected:	None				
Encoding:	01	11bb	bfff	ffff	
Description:	If bit 'b' in register 'f' is '0' then 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.				
Words:	1				
Cycles:	1(2)				
Q Cycle Activity:	Q1	Q2	Q3	Q4	
	Decode	Read register 'f'	Process data	No-Operation	
If Skip:	(2nd Cycle)				
	Q1	Q2	Q3	Q4	
	No-Operation	No-Operation	No-Operation	No-Operation	
Example	HERE FALSE TRUE	BTFSC GOTO • • •	FLAG,1 PROCESS_CODE		
	Before Instruction PC = address HERE				
	After Instruction if FLAG<1> = 0, PC = address FALSE if FLAG<1> = 1, PC = address TRUE				

CALL				
Call Subroutine				
Syntax:	[label] CALL k			
Operands:	0 ≤ k ≤ 2047			
Operation:	(PC)+ 1→ TOS, k → PC<10:0>, (PCLATH<4:3>) → PC<12:11>			
Status Affected:	None			
Encoding:	10	0kkk	kkkk	kkkk
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.			
Words:	1			
Cycles:	2			
Q Cycle Activity:	Q1	Q2	Q3	Q4
1st Cycle	Decode	Read literal 'k', Push PC to Stack	Process data	Write to PC
2nd Cycle	No-Operation	No-Operation	No-Operation	No-Operation
Example	HERE	CALL	THERE	
	Before Instruction			
	PC = Address HERE			
	After Instruction			
	PC = Address THERE			
	TOS = Address HERE+1			

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

Encoding:

00	0010	dfff	ffff
----	------	------	------

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'.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process data	Write to destination

Example 1: SUBWF REG1, 1

Before Instruction

REG1 = 3
W = 2
C = ?
Z = ?

After Instruction

REG1 = 1
W = 2
C = 1; result is positive
Z = 0

Example 2: Before Instruction

REG1 = 2
W = 2
C = ?
Z = ?

After Instruction

REG1 = 0
W = 2
C = 1; result is zero
Z = 1

Example 3: Before Instruction

REG1 = 1
W = 2
C = ?
Z = ?

After Instruction

REG1 = 0xFF
W = 2
C = 0; result is negative
Z = 0

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

Encoding:

00	1110	dfff	ffff
----	------	------	------

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'.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process data	Write to destination

Example SWAPF REG, 0

Before Instruction

REG1 = 0xA5

After Instruction

REG1 = 0xA5
W = 0x5A

TRIS Load TRIS Register

Syntax: [label] TRIS f

Operands: $5 \leq f \leq 7$

Operation: $(W) \rightarrow \text{TRIS register } f$;

Status Affected: None

Encoding:

00	0000	0110	0fff
----	------	------	------

Description: The instruction is supported for code compatibility with the PIC16C5X products. Since TRIS registers are readable and writable, the user can directly address them.

Words: 1

Cycles: 1

Example

To maintain upward compatibility with future PIC16CXX products, do not use this instruction.

MPASM has the following features to assist in developing software for specific use applications.

- Provides translation of Assembler source code to object code for all Microchip microcontrollers.
- Macro assembly capability.
- Produces all the files (Object, Listing, Symbol, and special) required for symbolic debug with Microchip's emulator systems.
- Supports Hex (default), Decimal and Octal source and listing formats.

MPASM provides a rich directive language to support programming of the PIC16/17. Directives are helpful in making the development of your assemble source code shorter and more maintainable.

16.11 Software Simulator (MPLAB-SIM)

The MPLAB-SIM Software Simulator allows code development in a PC host environment. It allows the user to simulate the PIC16/17 series microcontrollers on an instruction level. On any given instruction, the user may examine or modify any of the data areas or provide external stimulus to any of the pins. The input/output radix can be set by the user and the execution can be performed in; single step, execute until break, or in a trace mode.

MPLAB-SIM fully supports symbolic debugging using MPLAB-C and MPASM. The Software Simulator offers the low cost flexibility to develop and debug code outside of the laboratory environment making it an excellent multi-project software development tool.

16.12 C Compiler (MPLAB-C)

The MPLAB-C Code Development System is a complete 'C' compiler and integrated development environment for Microchip's PIC16/17 family of microcontrollers. The compiler provides powerful integration capabilities and ease of use not found with other compilers.

For easier source level debugging, the compiler provides symbol information that is compatible with the MPLAB IDE memory display (PICMASTER emulator software versions 1.13 and later).

16.13 Fuzzy Logic Development System (*fuzzyTECH-MP*)

fuzzyTECH-MP fuzzy logic development tool is available in two versions - a low cost introductory version, MP Explorer, for designers to gain a comprehensive working knowledge of fuzzy logic system design; and a full-featured version, *fuzzyTECH-MP*, edition for implementing more complex systems.

Both versions include Microchip's *fuzzyLAB*™ demonstration board for hands-on experience with fuzzy logic systems implementation.

16.14 MP-DriveWay™ – Application Code Generator

MP-DriveWay is an easy-to-use Windows-based Application Code Generator. With MP-DriveWay you can visually configure all the peripherals in a PIC16/17 device and, with a click of the mouse, generate all the initialization and many functional code modules in C language. The output is fully compatible with Microchip's MPLAB-C C compiler. The code produced is highly modular and allows easy integration of your own code. MP-DriveWay is intelligent enough to maintain your code through subsequent code generation.

16.15 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.

16.16 TrueGauge® Intelligent Battery Management

The TrueGauge development tool supports system development with the MTA11200B TrueGauge Intelligent Battery Management IC. System design verification can be accomplished before hardware prototypes are built. User interface is graphically-oriented and measured data can be saved in a file for exporting to Microsoft Excel.

16.17 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.

PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

FIGURE 17-11: A/D CONVERSION TIMING

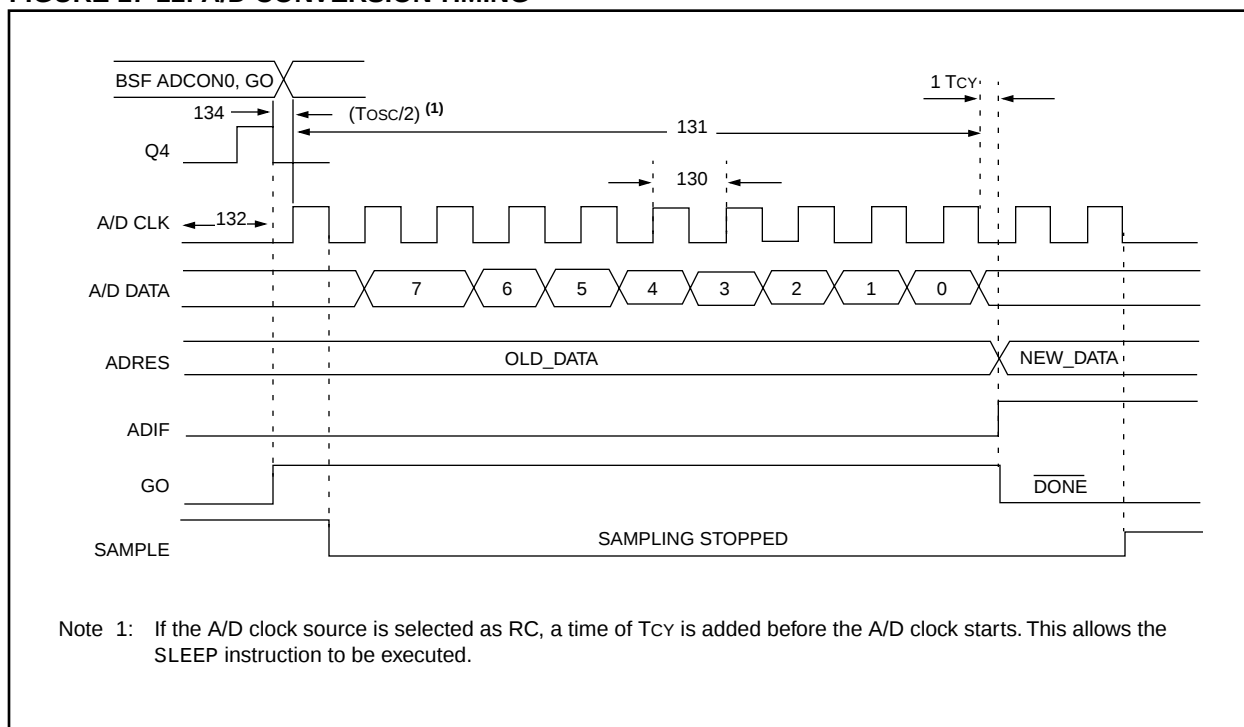


TABLE 17-11: A/D CONVERSION REQUIREMENTS

Param No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
130	TAD	A/D clock period	PIC16C72	1.6	—	—	μs	TOSC based, VREF ≥ 3.0V
			PIC16LC72	2.0	—	—	μs	TOSC based, VREF full range
			PIC16C72	2.0	4.0	6.0	μs	A/D RC Mode
			PIC16LC72	3.0	6.0	9.0	μs	A/D RC Mode
131	T _{CV}	Conversion time (not including S/H time) (Note 1)		—	9.5	—	TAD	
132	TACQ	Acquisition time		Note 2	20	—	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).
				5*	—	—	μs	
134	T _{GO}	Q4 to A/D clock start		—	Tosc/2 §	—	—	If the A/D clock source is selected as RC, a time of T _{CY} is added before the A/D clock starts. This allows the SLEEP instruction to be executed.
135	T _{SWC}	Switching from convert → sample time		1.5 §	—	—	TAD	

* 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 13.1 for min conditions.

18.2 DC Characteristics: PIC16LC73/74-04 (Commercial, Industrial)

DC CHARACTERISTICS		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial and 0°C ≤ TA ≤ +70°C for commercial					
Param No.	Characteristic	Sym	Min	Typ†	Max	Units	Conditions
D001	Supply Voltage	VDD	3.0	-	6.0	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
D010	Supply Current (Note 2,5)	IDD	-	2.0	3.8	mA	XT, RC osc configuration FOSC = 4 MHz, VDD = 3.0V (Note 4)
D010A			-	22.5	48	μA	LP osc configuration FOSC = 32 kHz, VDD = 3.0V, WDT disabled
D020	Power-down Current (Note 3,5)	IPD	-	7.5	30	μA	VDD = 3.0V, WDT enabled, -40°C to +85°C
D021			-	0.9	13.5	μA	VDD = 3.0V, WDT disabled, 0°C to +70°C
D021A			-	0.9	18	μA	VDD = 3.0V, WDT disabled, -40°C to +85°C

* 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 μA to the specification. This value is from characterization and is for design guidance only. This is not tested.

PIC16C7X

Applicable Devices 72 73 73A 74 74A 76 77

TABLE 20-8: SPI MODE REQUIREMENTS

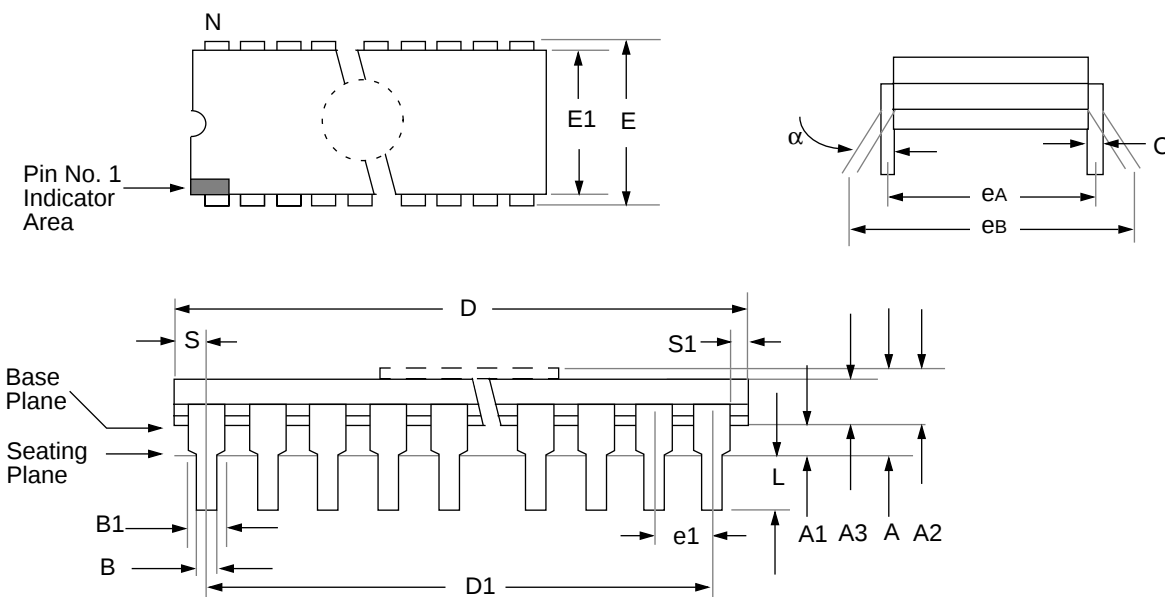
Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
70*	TssL2scH, TssL2scL	$\overline{SS}\downarrow$ to SCK $\downarrow$ or SCK $\uparrow$ input	T _{CY}	—	—	ns	
71*	TscH	SCK input high time (slave mode)	T _{CY} + 20	—	—	ns	
72*	TscL	SCK input low time (slave mode)	T _{CY} + 20	—	—	ns	
73*	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	
76*	TdoF	SDO data output fall time	—	10	25	ns	
77*	TssH2doZ	$\overline{SS}\uparrow$ to SDO output hi-impedance	10	—	50	ns	
78*	TscR	SCK output rise time (master mode)	—	10	25	ns	
79*	TscF	SCK output fall time (master mode)	—	10	25	ns	
80*	Tsch2doV, TscL2doV	SDO data output valid after SCK edge	—	—	50	ns	
81*	TdoV2scH, TdoV2scL	SDO data output setup to SCK edge	T _{CY}	—	—	ns	
82*	TssL2doV	SDO data output valid after $\overline{SS}\downarrow$ edge	—	—	50	ns	
83*	Tsch2ssH, TscL2ssH	$\overline{SS}\uparrow$ after SCK edge	1.5T _{CY} + 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.

PIC16C7X

22.2 40-Lead Ceramic Cerdip Dual In-line with Window (600 mil) (JW)



Package Group: Ceramic Cerdip Dual In-Line (CDP)						
Symbol	Millimeters			Inches		
	Min	Max	Notes	Min	Max	Notes
α	0°	10°		0°	10°	
A	4.318	5.715		0.170	0.225	
A1	0.381	1.778		0.015	0.070	
A2	3.810	4.699		0.150	0.185	
A3	3.810	4.445		0.150	0.175	
B	0.355	0.585		0.014	0.023	
B1	1.270	1.651	Typical	0.050	0.065	Typical
C	0.203	0.381	Typical	0.008	0.015	Typical
D	51.435	52.705		2.025	2.075	
D1	48.260	48.260	Reference	1.900	1.900	Reference
E	15.240	15.875		0.600	0.625	
E1	12.954	15.240		0.510	0.600	
e1	2.540	2.540	Reference	0.100	0.100	Reference
eA	14.986	16.002	Typical	0.590	0.630	Typical
eB	15.240	18.034		0.600	0.710	
L	3.175	3.810		0.125	0.150	
N	40	40		40	40	
S	1.016	2.286		0.040	0.090	
S1	0.381	1.778		0.015	0.070	

Instruction Set

ADDLW	149
ADDWF	149
ANDLW	149
ANDWF	149
BCF	150
BSF	150
BTFSC	150
BTFSS	151
CALL	151
CLRF	152
CLRWF	152
CLRWDW	152
COMF	153
DECF	153
DECFSZ	153
GOTO	154
INCF	154
INCFSSZ	155
IORLW	155
IORWF	156
MOVF	156
MOVLW	156
MOVWF	156
NOP	157
OPTION	157
RETFIE	157
RETLW	158
RETURN	158
RLF	159
RRF	159
SLEEP	160
SUBLW	160
SUBWF	161
SWAPF	161
TRIS	161
XORLW	162
XORWF	162
Section	147
Summary Table	148
INT Interrupt	143
INTCON	29
INTCON Register	32
INTEDG bit	31, 143
Internal Sampling Switch (Rss) Impedance	120
Interrupts	129
PortB Change	143
RB7:RB4 Port Change	45
Section	141
TMR0	143
IRP bit	30

L

Loading of PC	40
---------------	----

M

MCLR	133, 136
Memory	
Data Memory	20
Program Memory	19
Program Memory Maps	
PIC16C72	19
PIC16C73	19
PIC16C73A	19
PIC16C74	19
PIC16C74A	19
PIC16C76	20
PIC16C77	20
Register File Maps	
PIC16C72	21
PIC16C73	21
PIC16C73A	21
PIC16C74	21
PIC16C74A	21
PIC16C76	21
PIC16C77	21
MPASM Assembler	163
MPLAB-C	165
MPSIM Software Simulator	163, 165

O

OERR bit	100
OPCODE	147
OPTION	29
OPTION Register	31
Orthogonal	9
OSC selection	129
Oscillator	
HS	131, 135
LP	131, 135
RC	131
XT	131, 135
Oscillator Configurations	131
Output of TMR2	69

P

P	78, 83
Packaging	
28-Lead Ceramic w/Window	251
28-Lead PDIP	253
28-Lead SOIC	255
28-Lead SSOP	256
40-Lead CERDIP w/Window	252
40-Lead PDIP	254
44-Lead MQFP	258
44-Lead PLCC	257
44-Lead TQFP	259
Paging, Program Memory	40
Parallel Slave Port	50, 54
PCFG0 bit	118
PCFG1 bit	118
PCFG2 bit	118
PCL Register	23, 24, 25, 26, 27, 28, 29, 40
PCLATH	136
PCLATH Register	23, 24, 25, 26, 27, 28, 29, 40
PCON Register	29, 39, 135
PD bit	30, 133, 135
PICDEM-1 Low-Cost PIC16/17 Demo Board	163, 164
PICDEM-2 Low-Cost PIC16CXX Demo Board	163, 164
PICDEM-3 Low-Cost PIC16C9XXX Demo Board	164
PICMASTER In-Circuit Emulator	163

PIC16C7X

LIST OF FIGURES

Figure 3-1:	PIC16C72 Block Diagram	10	Figure 8-1:	T1CON: Timer1 Control Register (Address 10h)	65
Figure 3-2:	PIC16C73/73A/76 Block Diagram	11	Figure 8-2:	Timer1 Block Diagram	66
Figure 3-3:	PIC16C74/74A/77 Block Diagram	12	Figure 9-1:	Timer2 Block Diagram	69
Figure 3-4:	Clock/Instruction Cycle	17	Figure 9-2:	T2CON: Timer2 Control Register (Address 12h)	70
Figure 4-1:	PIC16C72 Program Memory Map and Stack	19	Figure 10-1:	CCP1CON Register (Address 17h)/ CCP2CON Register (Address 1Dh)	72
Figure 4-2:	PIC16C73/73A/74/74A Program Memory Map and Stack	19	Figure 10-2:	Capture Mode Operation Block Diagram	72
Figure 4-3:	PIC16C76/77 Program Memory Map and Stack	20	Figure 10-3:	Compare Mode Operation Block Diagram	73
Figure 4-4:	PIC16C72 Register File Map	21	Figure 10-4:	Simplified PWM Block Diagram	74
Figure 4-5:	PIC16C73/73A/74/74A Register File Map	21	Figure 10-5:	PWM Output	74
Figure 4-6:	PIC16C76/77 Register File Map	22	Figure 11-1:	SSPSTAT: Sync Serial Port Status Register (Address 94h)	78
Figure 4-7:	Status Register (Address 03h, 83h, 103h, 183h)	30	Figure 11-2:	SSPCON: Sync Serial Port Control Register (Address 14h)	79
Figure 4-8:	OPTION Register (Address 81h, 181h)	31	Figure 11-3:	SSP Block Diagram (SPI Mode)	80
Figure 4-9:	INTCON Register (Address 0Bh, 8Bh, 10Bh, 18Bh)	32	Figure 11-4:	SPI Master/Slave Connection	81
Figure 4-10:	PIE1 Register PIC16C72 (Address 8Ch)	33	Figure 11-5:	SPI Mode Timing, Master Mode or Slave Mode w/o SS Control	82
Figure 4-11:	PIE1 Register PIC16C73/73A/ 74/74A/76/77 (Address 8Ch)	34	Figure 11-6:	SPI Mode Timing, Slave Mode with SS Control	82
Figure 4-12:	PIR1 Register PIC16C72 (Address 0Ch)	35	Figure 11-7:	SSPSTAT: Sync Serial Port Status Register (Address 94h)(PIC16C76/77)	83
Figure 4-13:	PIR1 Register PIC16C73/73A/ 74/74A/76/77 (Address 0Ch)	36	Figure 11-8:	SSPCON: Sync Serial Port Control Register (Address 14h)(PIC16C76/77)	84
Figure 4-14:	PIE2 Register (Address 8Dh)	37	Figure 11-9:	SSP Block Diagram (SPI Mode) (PIC16C76/77)	85
Figure 4-15:	PIR2 Register (Address 0Dh)	38	Figure 11-10:	SPI Master/Slave Connection PIC16C76/77)	86
Figure 4-16:	PCON Register (Address 8Eh)	39	Figure 11-11:	SPI Mode Timing, Master Mode (PIC16C76/77)	87
Figure 4-17:	Loading of PC In Different Situations	40	Figure 11-12:	SPI Mode Timing (Slave Mode With CKE = 0) (PIC16C76/77)	87
Figure 4-18:	Direct/Indirect Addressing	41	Figure 11-13:	SPI Mode Timing (Slave Mode With CKE = 1) (PIC16C76/77)	88
Figure 5-1:	Block Diagram of RA3:RA0 and RA5 Pins	43	Figure 11-14:	Start and Stop Conditions	89
Figure 5-2:	Block Diagram of RA4/T0CKI Pin	43	Figure 11-15:	7-bit Address Format	90
Figure 5-3:	Block Diagram of RB3:RB0 Pins	45	Figure 11-16:	I ² C 10-bit Address Format	90
Figure 5-4:	Block Diagram of RB7:RB4 Pins (PIC16C73/74)	46	Figure 11-17:	Slave-receiver Acknowledge	90
Figure 5-5:	Block Diagram of RB7:RB4 Pins (PIC16C72/73A/ 74A/76/77)	46	Figure 11-18:	Data Transfer Wait State	90
Figure 5-6:	PORTC Block Diagram (Peripheral Output Override)	48	Figure 11-19:	Master-transmitter Sequence	91
Figure 5-7:	PORTD Block Diagram (in I/O Port Mode)	50	Figure 11-20:	Master-receiver Sequence	91
Figure 5-8:	PORTE Block Diagram (in I/O Port Mode)	51	Figure 11-21:	Combined Format	91
Figure 5-9:	TRISE Register (Address 89h)	51	Figure 11-22:	Multi-master Arbitration (Two Masters)	92
Figure 5-10:	Successive I/O Operation	53	Figure 11-23:	Clock Synchronization	92
Figure 5-11:	PORTD and PORTE Block Diagram (Parallel Slave Port)	54	Figure 11-24:	SSP Block Diagram (I ² C Mode)	93
Figure 5-12:	Parallel Slave Port Write Waveforms	55	Figure 11-25:	I ² C Waveforms for Reception (7-bit Address)	95
Figure 5-13:	Parallel Slave Port Read Waveforms	55	Figure 11-26:	I ² C Waveforms for Transmission (7-bit Address)	96
Figure 7-1:	Timer0 Block Diagram	59	Figure 11-27:	Operation of the I ² C Module in IDLE_MODE, RCV_MODE or XMIT_MODE	98
Figure 7-2:	Timer0 Timing: Internal Clock/No Prescale	59	Figure 12-1:	TXSTA: Transmit Status and Control Register (Address 98h)	99
Figure 7-3:	Timer0 Timing: Internal Clock/Prescale 1:2	60	Figure 12-2:	RCSTA: Receive Status and Control Register (Address 18h)	100
Figure 7-4:	Timer0 Interrupt Timing	60	Figure 12-3:	RX Pin Sampling Scheme. BRGH = 0 (PIC16C73/73A/74/74A)	104
Figure 7-5:	Timer0 Timing with External Clock	61	Figure 12-4:	RX Pin Sampling Scheme, BRGH = 1 (PIC16C73/73A/74/74A)	104
Figure 7-6:	Block Diagram of the Timer0/WDT Prescaler	62			

Figure 12-5:	RX Pin Sampling Scheme, BRGH = 1 (PIC16C73/73A/74/74A)	104	Figure 17-4:	Reset, Watchdog Timer, Oscillator Start-up Timer and Power-up Timer Timing	175
Figure 12-6:	RX Pin Sampling Scheme, BRGH = 0 OR BRGH = 1 (PIC16C76/77)	105	Figure 17-5:	Brown-out Reset Timing	175
Figure 12-7:	USART Transmit Block Diagram.....	106	Figure 17-6:	Timer0 and Timer1 External Clock Timings	176
Figure 12-8:	Asynchronous Master Transmission	107	Figure 17-7:	Capture/Compare/PWM Timings (CCP1)	177
Figure 12-9:	Asynchronous Master Transmission (Back to Back).....	107	Figure 17-8:	SPI Mode Timing	178
Figure 12-10:	USART Receive Block Diagram.....	108	Figure 17-9:	I ² C Bus Start/Stop Bits Timing.....	179
Figure 12-11:	Asynchronous Reception	108	Figure 17-10:	I ² C Bus Data Timing	180
Figure 12-12:	Synchronous Transmission.....	111	Figure 17-11:	A/D Conversion Timing.....	182
Figure 12-13:	Synchronous Transmission (Through TXEN).....	111	Figure 18-1:	Load Conditions.....	188
Figure 12-14:	Synchronous Reception (Master Mode, SREN).....	113	Figure 18-2:	External Clock Timing.....	189
Figure 13-1:	ADCON0 Register (Address 1Fh)	117	Figure 18-3:	CLKOUT and I/O Timing.....	190
Figure 13-2:	ADCON1 Register (Address 9Fh)	118	Figure 18-4:	Reset, Watchdog Timer, Oscillator Start-up Timer and Power-up Timer Timing.....	191
Figure 13-3:	A/D Block Diagram.....	119	Figure 18-5:	Timer0 and Timer1 External Clock Timings	192
Figure 13-4:	Analog Input Model	120	Figure 18-6:	Capture/Compare/PWM Timings (CCP1 and CCP2)	193
Figure 13-5:	A/D Transfer Function.....	125	Figure 18-7:	Parallel Slave Port Timing (PIC16C74).....	194
Figure 13-6:	Flowchart of A/D Operation.....	126	Figure 18-8:	SPI Mode Timing	195
Figure 14-1:	Configuration Word for PIC16C73/74.....	129	Figure 18-9:	I ² C Bus Start/Stop Bits Timing.....	196
Figure 14-2:	Configuration Word for PIC16C72/73A/74A/76/77.....	130	Figure 18-10:	I ² C Bus Data Timing	197
Figure 14-3:	Crystal/Ceramic Resonator Operation (HS, XT or LP OSC Configuration).....	131	Figure 18-11:	USART Synchronous Transmission (Master/Slave) Timing.....	198
Figure 14-4:	External Clock Input Operation (HS, XT or LP OSC Configuration)	131	Figure 18-12:	USART Synchronous Receive (Master/Slave) Timing.....	198
Figure 14-5:	External Parallel Resonant Crystal Oscillator Circuit	132	Figure 18-13:	A/D Conversion Timing.....	200
Figure 14-6:	External Series Resonant Crystal Oscillator Circuit	132	Figure 19-1:	Load Conditions.....	206
Figure 14-7:	RC Oscillator Mode	132	Figure 19-2:	External Clock Timing	207
Figure 14-8:	Simplified Block Diagram of On-chip Reset Circuit.....	133	Figure 19-3:	CLKOUT and I/O Timing.....	208
Figure 14-9:	Brown-out Situations	134	Figure 19-4:	Reset, Watchdog Timer, Oscillator Start-up Timer and Power-up Timer Timing	209
Figure 14-10:	Time-out Sequence on Power-up (MCLR not Tied to VDD): Case 1.....	139	Figure 19-5:	Brown-out Reset Timing	209
Figure 14-11:	Time-out Sequence on Power-up (MCLR Not Tied To VDD): Case 2.....	139	Figure 19-6:	Timer0 and Timer1 External Clock Timings	210
Figure 14-12:	Time-out Sequence on Power-up (MCLR Tied to VDD)	139	Figure 19-7:	Capture/Compare/PWM Timings (CCP1 and CCP2)	211
Figure 14-13:	External Power-on Reset Circuit (for Slow VDD Power-up).....	140	Figure 19-8:	Parallel Slave Port Timing (PIC16C74A)	212
Figure 14-14:	External Brown-out Protection Circuit 1	140	Figure 19-9:	SPI Mode Timing	213
Figure 14-15:	External Brown-out Protection Circuit 2	140	Figure 19-10:	I ² C Bus Start/Stop Bits Timing.....	214
Figure 14-16:	Interrupt Logic	142	Figure 19-11:	I ² C Bus Data Timing	215
Figure 14-17:	INT Pin Interrupt Timing	142	Figure 19-12:	USART Synchronous Transmission (Master/Slave) Timing.....	216
Figure 14-18:	Watchdog Timer Block Diagram	144	Figure 19-13:	USART Synchronous Receive (Master/Slave) Timing.....	216
Figure 14-19:	Summary of Watchdog Timer Registers.....	144	Figure 19-14:	A/D Conversion Timing.....	218
Figure 14-20:	Wake-up from Sleep Through Interrupt.....	146	Figure 20-1:	Load Conditions.....	225
Figure 14-21:	Typical In-Circuit Serial Programming Connection	146	Figure 20-2:	External Clock Timing	226
Figure 15-1:	General Format for Instructions	147	Figure 20-3:	CLKOUT and I/O Timing.....	227
Figure 17-1:	Load Conditions	172	Figure 20-4:	Reset, Watchdog Timer, Oscillator Start-up Timer and Power-up Timer Timing	228
Figure 17-2:	External Clock Timing	173	Figure 20-5:	Brown-out Reset Timing	228
Figure 17-3:	CLKOUT and I/O Timing	174	Figure 20-6:	Timer0 and Timer1 External Clock Timings	229
			Figure 20-7:	Capture/Compare/PWM Timings (CCP1 and CCP2)	230
			Figure 20-8:	Parallel Slave Port Timing (PIC16C77).....	231