



Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

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

Details

Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	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	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	40-DIP (0.600", 15.24mm)
Supplier Device Package	40-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16c74a-04i-p

Email: info@E-XFL.COM

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

Pin Name	DIP Pin#	SSOP Pin#	SOIC Pin#	l/O/P Type	Buffer Type	Description
OSC1/CLKIN	9	9	9	I	ST/CMOS ⁽³⁾	Oscillator crystal input/external clock source input.
OSC2/CLKOUT	10	10	10	0	_	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	1	1	1	I/P	ST	Master clear (reset) input or programming voltage input. This pin is an active low reset to the device.
						PORTA is a bi-directional I/O port.
RA0/AN0	2	2	2	I/O	TTL	RA0 can also be analog input0
RA1/AN1	3	3	3	I/O	TTL	RA1 can also be analog input1
RA2/AN2	4	4	4	I/O	TTL	RA2 can also be analog input2
RA3/AN3/VREF	5	5	5	I/O	TTL	RA3 can also be analog input3 or analog reference voltage
RA4/T0CKI	6	6	6	I/O	ST	RA4 can also be the clock input to the Timer0 module. Output is open drain type.
RA5/SS/AN4	7	7	7	I/O	TTL	RA5 can also be analog input4 or the slave select for the synchronous serial port.
						PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs.
RB0/INT	21	21	21	I/O	TTL/ST ⁽¹⁾	RB0 can also be the external interrupt pin.
RB1	22	22	22	I/O	TTL	
RB2	23	23	23	I/O	TTL	
RB3	24	24	24	I/O	TTL	
RB4	25	25	25	I/O	TTL	Interrupt on change pin.
RB5	26	26	26	I/O	TTL	Interrupt on change pin.
RB6	27	27	27	I/O	TTL/ST ⁽²⁾	Interrupt on change pin. Serial programming clock.
RB7	28	28	28	I/O	TTL/ST ⁽²⁾	Interrupt on change pin. Serial programming data.
						PORTC is a bi-directional I/O port.
RC0/T1OSO/T1CKI	11	11	11	I/O	ST	RC0 can also be the Timer1 oscillator output or Timer1 clock input.
RC1/T1OSI	12	12	12	I/O	ST	RC1 can also be the Timer1 oscillator input.
RC2/CCP1	13	13	13	I/O	ST	RC2 can also be the Capture1 input/Compare1 output/ PWM1 output.
RC3/SCK/SCL	14	14	14	I/O	ST	RC3 can also be the synchronous serial clock input/output for both SPI and I ² C modes.
RC4/SDI/SDA	15	15	15	I/O	ST	RC4 can also be the SPI Data In (SPI mode) or data I/O (I ² C mode).
RC5/SDO	16	16	16	I/O	ST	RC5 can also be the SPI Data Out (SPI mode).
RC6	17	17	17	I/O	ST	
RC7	18	18	18	I/O	ST	
Vss	8, 19	8, 19	8, 19	Р		Ground reference for logic and I/O pins.
Vdd	20	20	20	Р		Positive supply for logic and I/O pins.
Legend: I = input	0 =	output	1	I/O = i	nput/output	P = power
	— =	Not used	1	TTI =	TTI input	ST = Schmitt Trigger input

TABLE 3-1:	PIC16C72	PINOUT	DESCRIPTION
------------	-----------------	--------	-------------

Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt. 2: This buffer is a Schmitt Trigger input when used in serial programming mode.

3: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

FIGURE 4-3: PIC16C76/77 PROGRAM MEMORY MAP AND STACK



4.2 Data Memory Organization

 Applicable Devices

 72
 73
 73
 74
 74
 76
 77

The data memory is partitioned into multiple banks which contain the General Purpose Registers and the Special Function Registers. Bits RP1 and RP0 are the bank select bits.

RP1:RP0 (STATUS<6:5>)

- = 00 \rightarrow Bank0
- = 01 \rightarrow Bank1
- = $10 \rightarrow \text{Bank2}$
- = 11 \rightarrow Bank3

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.

4.2.1 GENERAL PURPOSE REGISTER FILE

The register file can be accessed either directly, or indirectly through the File Select Register FSR (Section 4.5).



FIGURE 5-12: PARALLEL SLAVE PORT WRITE WAVEFORMS





Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 1 Bit 0		Value on all other resets
08h	PORTD	Port dat	a latch	when w	ritten: Port pi	ns when	read			xxxx xxxx	uuuu uuuu
09h	PORTE	—	—	—	—	—	RE2 RE1 RE0		xxx	uuu	
89h	TRISE	IBF	OBF	IBOV	PSPMODE	—	PORTE D	ata Direction	n Bits	0000 -111	0000 -111
0Ch	PIR1	PSPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
9Fh	ADCON1	_	—	—	—	—	PCFG2 PCFG1 PCFG0		000	000	

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

NOTES:

NOTES:

8.0 TIMER1 MODULE Applicable Devices

72 73 73A 74 74A 76 77

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 either of the two CCP modules (Section 10.0). Figure 8-1 shows the Timer1 control register.

For the PIC16C72/73A/74A/76/77, when the Timer1 oscillator is enabled (T1OSCEN is set), the RC1/T1OSI/CCP2 and RC0/T1OSO/T1CKI pins become inputs. That is, the TRISC<1:0> value is ignored.

For the PIC16C73/74, when the Timer1 oscillator is enabled (T1OSCEN is set), RC1/T1OSI/CCP2 pin becomes an input, however the RC0/T1OSO/T1CKI pin will have to be configured as an input by setting the TRISC<0> bit.

FIGURE 8-1: T1CON: TIMER1 CONTROL REGISTER (ADDRESS 10h)



8.5 <u>Resetting Timer1 using a CCP Trigger</u> Output

Applicable Devices

The CCP2 module is not implemented on the PIC16C72 device.

If the CCP1 or CCP2 module is configured in compare mode to generate a "special event trigger" (CCP1M3:CCP1M0 = 1011), this signal will reset Timer1.

Note:	The special event triggers from the CCP1
	and CCP2 modules 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 or CCP2, the write will take precedence.

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

8.6 Resetting of Timer1 Register Pair (TMR1H, TMR1L) Applicable Devices 72|73|73A|74|74A|76|77

TMR1H and TMR1L registers are not reset to 00h on a POR or any other reset except by the CCP1 and CCP2 special event triggers.

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

8.7 <u>Timer1 Prescaler</u> Applicable Devices

72 73 73A 74 74A 76 77

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

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	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF ^(1,2)	ADIF	RCIF ⁽²⁾	TXIF ⁽²⁾	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE ^(1,2)	ADIE	RCIE ⁽²⁾	TXIE ⁽²⁾	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
0Eh	TMR1L	Holding reg	Holding register for the Least Significant Byte of the 16-bit TMR1 register								uuuu uuuu
0Fh	TMR1H	Holding reg	gister fo	xxxx xxxx	uuuu uuuu						
10h	T1CON	_	_	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR10N	00 0000	uu uuuu

TABLE 8-2: REGISTERS ASSOCIATED WITH TIMER1 AS A TIMER/COUNTER

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

Note 1: Bits PSPIE and PSPIF are reserved on the PIC16C73/73A/76, always maintain these bits clear.

2: The PIC16C72 does not have a Parallel Slave Port or a USART, these bits are unimplemented, read as '0'.

The \overline{SS} pin allows a synchronous slave mode. The SPI must be in slave mode (SSPCON<3:0> = 04h) and the TRISA<5> bit must be set the for synchronous slave mode to be enabled. When the \overline{SS} pin is low, transmission and reception are enabled and the SDO pin is driven. When the \overline{SS} pin goes high, the SDO pin is no longer driven, even if in the middle of a transmitted byte, and becomes a floating output. If the \overline{SS} pin is taken low without resetting SPI mode, the transmission will continue from the

point at which it was taken high. External pull-up/ pull-down resistors may be desirable, depending on the application.

To emulate two-wire communication, the SDO pin can be connected to the SDI pin. When the SPI needs to operate as a receiver the SDO pin can be configured as an input. This disables transmissions from the SDO. The SDI can always be left as an input (SDI function) since it cannot create a bus conflict.









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	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF ^(1,2)	ADIF	RCIF ⁽²⁾	TXIF ⁽²⁾	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE ^(1,2)	ADIE	RCIE ⁽²⁾	TXIE ⁽²⁾	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
87h	TRISC	PORTC Da	ta Directio	on Registe	er					1111 1111	1111 1111
13h	SSPBUF	Synchrono	us Serial I	Port Rece	ive Buffer	/Transmit	Register			xxxx xxxx	uuuu uuuu
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000
85h	TRISA	—	_	PORTA Data Direction Register						11 1111	11 1111
94h	SSPSTAT	—	—	D/Ā	Р	S	R/W	UA	BF	00 0000	00 0000

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

Note 1: Bits PSPIE and PSPIF are reserved on the PIC16C73/73A, always maintain these bits clear.

2: The PIC16C72 does not have a Parallel Slave Port or USART, these bits are unimplemented, read as '0'.

11.3.1 SPI MODE FOR PIC16C76/77

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

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

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

Slave Select (SS) RA5/SS/AN4

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

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

The SSP consists of a transmit/receive Shift Register (SSPSR) and a buffer register (SSPBUF). The SSPSR shifts the data in and out of the device, MSb first. The SSPBUF holds the data that was written to the SSPSR until the received data is ready. Once the 8-bits of data have been received, that byte is moved to the SSPBUF register. Then the buffer full detect bit BF (SSPSTAT<0>) and interrupt flag bit SSPIF (PIR1<3>) are set. This double buffering of the received data (SSPBUF) allows the next byte to start reception before reading the data that was just received. Any write to the SSPBUF register during transmission/reception of data will be ignored, and the write collision detect bit WCOL (SSPCON<7>) will be set. User software must clear the WCOL bit so that it can be determined if the following write(s) to the SSPBUF register completed successfully. When the application software is expecting to receive valid data, the SSPBUF should be read before the next byte of data to transfer is written to the SSPBUF. Buffer full bit BF (SSPSTAT<0>) indicates when SSPBUF has been loaded with the received data (transmission is complete). When the SSPBUF is read, bit BF is cleared. This data may be irrelevant if the SPI is only a transmitter. Generally the SSP Interrupt is used to determine when the transmission/reception has completed. The SSPBUF must be read and/or written. If the interrupt method is not going to be used, then software polling can be done to ensure that a write collision does not occur. Example 11-2 shows the loading of the SSPBUF (SSPSR) for data transmission. The shaded instruction is only required if the received data is meaningful.

EXAMPLE 11-2: LOADING THE SSPBUF (SSPSR) REGISTER (PIC16C76/77)

	BCF BSF	STATUS , STATUS ,	RP1 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 11-9), shows that the SSPSR 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 11-9: SSP BLOCK DIAGRAM (SPI MODE)(PIC16C76/77)



Steps to follow when setting up an Asynchronous Transmission:

- 1. Initialize the SPBRG register for the appropriate baud rate. If a high speed baud rate is desired, set bit BRGH. (Section 12.1)
- 2. Enable the asynchronous serial port by clearing bit SYNC and setting bit SPEN.
- 3. If interrupts are desired, then set enable bit TXIE.
- 4. If 9-bit transmission is desired, then set transmit bit TX9.
- 5. Enable the transmission by setting bit TXEN, which will also set bit TXIF.
- 6. If 9-bit transmission is selected, the ninth bit should be loaded in bit TX9D.
- 7. Load data to the TXREG register (starts transmission).

FIGURE 12-8: ASYNCHRONOUS MASTER TRANSMISSION



FIGURE 12-9: ASYNCHRONOUS MASTER TRANSMISSION (BACK TO BACK)



TABLE 12-6: REGISTERS ASSOCIATED WITH ASYNCHRONOUS TRANSMISSION

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR		Bit 0 Value on BoR		Valu all o Res	e on ther sets
0Ch	PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000	0000	0000	0000		
18h	RCSTA	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000	-00x	0000	-00x		
19h	TXREG	USART Trar	nsmit Re	gister						0000	0000	0000	0000		
8Ch	PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000	0000	0000	0000		
98h	TXSTA	CSRC	ТХ9	TXEN	SYNC	_	BRGH	TRMT	TX9D	0000	-010	0000	-010		
99h	SPBRG	Baud Rate (Baud Rate Generator Register								0000	0000	0000		

Legend: x = unknown, - = unimplemented locations read as '0'. Shaded cells are not used for Asynchronous Transmission.

Note 1: Bits PSPIE and PSPIF are reserved on the PIC16C73/73A/76, always maintain these bits clear.

12.4 USART Synchronous Slave Mode

Applicable Devices 72 73 73A 74 74A 76 77

Synchronous slave mode differs from the Master mode in the fact that the shift clock is supplied externally at the RC6/TX/CK pin (instead of being supplied internally in master mode). This allows the device to transfer or receive data while in SLEEP mode. Slave mode is entered by clearing bit CSRC (TXSTA<7>).

12.4.1 USART SYNCHRONOUS SLAVE TRANSMIT

The operation of the synchronous master and slave modes are identical except in the case of the SLEEP mode.

If two words are written to the TXREG and then the SLEEP instruction is executed, the following will occur:

- a) The first word will immediately transfer to the TSR register and transmit.
- b) The second word will remain in TXREG register.
- c) Flag bit TXIF will not be set.
- d) When the first word has been shifted out of TSR, the TXREG register will transfer the second word to the TSR and flag bit TXIF will now be set.
- e) If enable bit TXIE is set, the interrupt will wake the chip from SLEEP and if the global interrupt is enabled, the program will branch to the interrupt vector (0004h).

Steps to follow when setting up a Synchronous Slave Transmission:

- 1. Enable the synchronous slave serial port by setting bits SYNC and SPEN and clearing bit CSRC.
- 2. Clear bits CREN and SREN.
- 3. If interrupts are desired, then set enable bit TXIE.
- 4. If 9-bit transmission is desired, then set bit TX9.
- 5. Enable the transmission by setting enable bit TXEN.
- 6. If 9-bit transmission is selected, the ninth bit should be loaded in bit TX9D.
- 7. Start transmission by loading data to the TXREG register.

12.4.2 USART SYNCHRONOUS SLAVE RECEPTION

The operation of the synchronous master and slave modes is identical except in the case of the SLEEP mode. Also, bit SREN is a don't care in slave mode.

If receive is enabled, by setting bit CREN, prior to the SLEEP instruction, then a word may be received during SLEEP. On completely receiving the word, the RSR register will transfer the data to the RCREG register and if enable bit RCIE bit is set, the interrupt generated will wake the chip from SLEEP. If the global interrupt is enabled, the program will branch to the interrupt vector (0004h).

Steps to follow when setting up a Synchronous Slave Reception:

- 1. Enable the synchronous master serial port by setting bits SYNC and SPEN and clearing bit CSRC.
- 2. If interrupts are desired, then set enable bit RCIE.
- 3. If 9-bit reception is desired, then set bit RX9.
- 4. To enable reception, set enable bit CREN.
- 5. Flag bit RCIF will be set when reception is complete and an interrupt will be generated, if enable bit RCIE was set.
- 6. Read the RCSTA register to get the ninth bit (if enabled) and determine if any error occurred during reception.
- 7. Read the 8-bit received data by reading the RCREG register.
- 8. If any error occurred, clear the error by clearing bit CREN.

14.3 <u>Reset</u> Applicable Devices 72|73|73A|74|74A|76|77

The PIC16CXX differentiates between various kinds of reset:

- Power-on Reset (POR)
- MCLR reset during normal operation
- MCLR reset during SLEEP
- WDT Reset (normal operation)
- Brown-out Reset (BOR) (PIC16C72/73A/74A/76/ 77)

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 Brownout 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 14-5 and Table 14-6. These bits are used in software to determine the nature of the reset. See Table 14-8 for a full description of reset states of all registers.

A simplified block diagram of the on-chip reset circuit is shown in Figure 14-8.

The PIC16C72/73A/74A/76/77 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 14-20: WAKE-UP FROM SLEEP THROUGH INTERRUPT

; a1 a2 a3 a4 ; a1 a2 a osc1/^_/_/_/_//		Q1 Q2 Q3 Q4	; q1 q2 q3 q4 ; ////////////////////////////////////	Q1 Q2 Q3 Q4 ;	Q1 Q2 Q3 Q4; ∕──∕──∕─
CLKOUT(4) ,//	Tost(2)			\'	'
INT pin				1 1 1	
INTF flag (INTCON<1>)	`	1	Interrupt Latency (Note 2)		
GIE bit (INTCON<7>)	Processor in SLEEP			ו ו ו ו	1 1 1
INSTRUCTION FLOW				1	1
PC X PC X PC+1	X PC+2	PC+2	PC + 2	X 0004h	0005h
Instruction $\begin{cases} \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ $	+ 1)	Inst(PC + 2)		Inst(0004h)	Inst(0005h)
Instruction executed I Inst(PC - 1) SLEEP		Inst(PC + 1)	Dummy cycle	Dummy cycle	Inst(0004h)
Note 1. XT US or I D appillator mode on	ourse d				

Note 1: XT, HS or LP oscillator mode assumed.

2: TOST = 1024TOSC (drawing not to scale) This delay will not be there for RC osc mode.

3: GIE = '1' assumed. In this case after wake- up, the processor jumps to the interrupt routine. If GIE = '0', execution will continue in-line.

4: CLKOUT is not available in these osc modes, but shown here for timing reference.

14.9 <u>Program Verification/Code Protection</u>

Applicable Devices

72 73 73A 74 74A 76 77

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

Note:	Microchip does not recommend code pro-						
	tecting windowed devices.						

14.10 ID Locations

Applicable Devices

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.

14.11 In-Circuit Serial Programming Applicable Devices 72/73/73A/74/74A/76/77

PIC16CXX microcontrollers can be serially programmed while in the end application circuit. This is simply done with two lines for clock and data, and three other lines for power, ground, and the programming voltage. This allows customers to manufacture boards with unprogrammed devices, and then program the microcontroller just before shipping the product. This also allows the most recent firmware or a custom firmware to be programmed. The device is placed into a program/verify mode by holding the RB6 and RB7 pins low while raising the $\overline{\text{MCLR}}$ (VPP) pin from VIL to VIHH (see programming specification). RB6 becomes the programming clock and RB7 becomes the programming data. Both RB6 and RB7 are Schmitt Trigger inputs in this mode.

After reset, to place the device into programming/verify mode, the program counter (PC) is at location 00h. A 6bit command is then supplied to the device. Depending on the command, 14-bits of program data are then supplied to or from the device, depending if the command was a load or a read. For complete details of serial programming, please refer to the PIC16C6X/7X Programming Specifications (Literature #DS30228).

FIGURE 14-21: TYPICAL IN-CIRCUIT SERIAL PROGRAMMING CONNECTION



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 <u>C Compiler (MPLAB-C)</u>

The MPLAB-C Code Development System is a complete 'C' compiler and integrated development environment for Microchip's PIC16/17 family of micro-controllers. 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 <u>Fuzzy Logic Development System</u> (fuzzyTECH-MP)

*fuzzy*TECH-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, *fuzzy*TECH-MP, edition for implementing more complex systems.

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

16.14 <u>MP-DriveWay™ – Application Code</u> <u>Generator</u>

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 <u>SEEVAL[®] Evaluation and</u> <u>Programming System</u>

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 tradeoff analysis and reliability calculations. The total kit can significantly reduce time-to-market and result in an optimized system.

16.16 <u>TrueGauge[®] Intelligent Battery</u> <u>Management</u>

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 <u>KEELOQ[®] Evaluation and</u> <u>Programming Tools</u>

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.

Applicable Devices 72 73 73A 74 74A 76 77

	Standard Operating Conditions (unloss otherwise stated)								
		Onerating temperature -40° $< T_{A} < \pm125^{\circ}$ for extended							
		-40° $\leq TA \leq +125^{\circ}$ C for industrial and							
DC CHARA	CTERISTICS	$-40 \circ C \simeq TA \simeq +00 \circ 101$ industrial and $0^{\circ}C \simeq -7A < +70^{\circ}C$ for commercial							
		Operating voltage \sqrt{D} range as described in DC space Section 17.1							
		and Se	ction 17.2		b range a	0 0000			
Param	Characteristic	Sym	Min	Тур	Max	Units	Conditions		
No.				†					
	Output High Voltage								
D090	I/O ports (Note 3)	Voн	Vdd - 0.7	-	-	V	IOH = -3.0 mA, VDD = 4.5V, -40°С to +85°С		
D090A			Vdd - 0.7	-	-	V	IOH = -2.5 mA, VDD = 4.5V, -40°С to +125°С		
D092	OSC2/CLKOUT (RC osc config)		Vdd - 0.7	-	-	V	IOH = -1.3 mA, VDD = 4.5V, -40°С to +85°С		
D092A			Vdd - 0.7	-	-	V	Юн = -1.0 mA, VDD = 4.5V, -40°С to +125°С		
D150*	Open-Drain High Voltage	Vod	-	-	14	V	RA4 pin		
	Capacitive Loading Specs on Out-								
	put Pins								
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)	Cio	-	-	50	pF			
D102	SCL, SDA in I ² C mode	Св	-	-	400	pF			

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 PIC16C7X be driven with external clock in RC mode.

2: The leakage current on the MCLR/VPP 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.

Applicable Devices 72 73 73A 74 74A 76 77





CAPTURE/COMPARE/PWM REQUIREMENTS (CCP1 AND CCP2) **TABLE 19-6**:

Param No.	Sym	Characteristic	Min	Тур†	Max	Units	Conditions		
50* TccL	TccL	CCP1 and CCP2 input low time	No Prescaler		0.5Tcy + 20	-	_	ns	
			With Prescaler	PIC16 C 73A/74A	10	-	_	ns	
				PIC16 LC 73A/74A	20	—	—	ns	
51*	51* TccH CCP1 and CCP2	CCP1 and CCP2	No Prescaler		0.5Tcy + 20	—		ns	
	input high time	With Prescaler	PIC16 C 73A/74A	10	-	—	ns		
			PIC16 LC 73A/74A	20	_	—	ns		
52*	TccP	CCP1 and CCP2 in	nput period	<u>3Tcy + 40</u> N	-	_	ns	N = prescale value (1,4 or 16)	
53*	TccR	CCP1 and CCP2 output rise time		PIC16 C 73A/74A	_	10	25	ns	
			PIC16 LC 73A/74A	—	25	45	ns		
54*	54* TccF CCP1 and CCP2 output fall time		PIC16 C 73A/74A	—	10	25	ns		
				PIC16 LC 73A/74A	_	25	45	ns	

These parameters are characterized but not tested. t

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

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

*

PIC16C7X



FIGURE 21-15: MAXIMUM IDD vs. FREQUENCY (RC MODE @ 100 pF, -40°C TO 85°C)



Data based on matrix samples. See first page of this section for details.

PIC16C7X

22.4 40-Lead Plastic Dual In-line (600 mil) (P)



Package Group: Plastic Dual In-Line (PLA)									
		Millimeters		Inches					
Symbol	Min	Мах	Notes	Min	Max	Notes			
α	0°	10°		0°	10°				
A	_	5.080		_	0.200				
A1	0.381	_		0.015	_				
A2	3.175	4.064		0.125	0.160				
В	0.355	0.559		0.014	0.022				
B1	1.270	1.778	Typical	0.050	0.070	Typical			
С	0.203	0.381	Typical	0.008	0.015	Typical			
D	51.181	52.197		2.015	2.055				
D1	48.260	48.260	Reference	1.900	1.900	Reference			
E	15.240	15.875		0.600	0.625				
E1	13.462	13.970		0.530	0.550				
e1	2.489	2.591	Typical	0.098	0.102	Typical			
eA	15.240	15.240	Reference	0.600	0.600	Reference			
eB	15.240	17.272		0.600	0.680				
L	2.921	3.683		0.115	0.145				
N	40	40		40	40				
S	1.270	_		0.050	_				
S1	0.508	_		0.020	_				

NOTES:

PIC16C7X PRODUCT IDENTIFICATION SYSTEM

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



* JW Devices are UV erasable and can be programmed to any device configuration. JW Devices meet the electrical requirement of each oscillator type (including LC devices).

Sales and Support

Products supported by a preliminary Data Sheet may possibly 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. The Microchip Website at www.microchip.com
- 2. Your local Microchip sales office (see following page)
- 3. The Microchip Corporate Literature Center U.S. FAX: (602) 786-7277
- 4. The Microchip's Bulletin Board, via your local CompuServe number (CompuServe membership NOT required).

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

For latest version information and upgrade kits for Microchip Development Tools, please call 1-800-755-2345 or 1-602-786-7302.