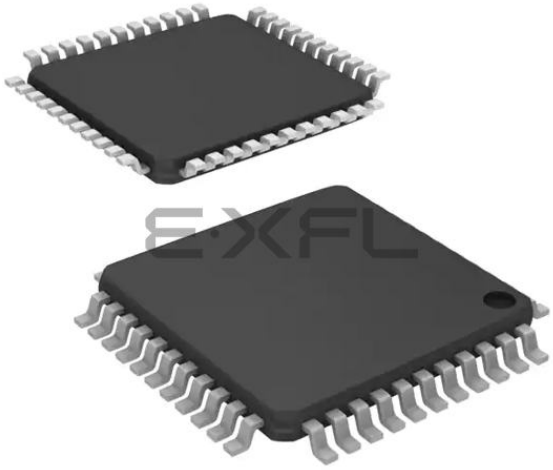


Welcome to E-XFL.COM

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	24MHz
Connectivity	SCI, UART/USART, USB
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	33
Program Memory Size	14KB (8K x 14)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	4.35V ~ 5.25V
Data Converters	A/D 8x8b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-TQFP
Supplier Device Package	44-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16c765-i-pt

3.1 Clocking Scheme/Instruction Cycle

The clock input feeds either an on-chip PLL, or directly drives (FINT). The clock output from either the PLL or direct drive (FINT) is internally divided by four to generate four non-overlapping quadrature clocks namely, Q1, Q2, Q3 and Q4. Internally, the program counter (PC) is incremented every Q1, the instruction is fetched from the program memory and latched into the instruction register in Q4. The instruction is decoded and executed during the following Q1 through Q4. The clocks and instruction execution flow is shown in Figure 3-2.

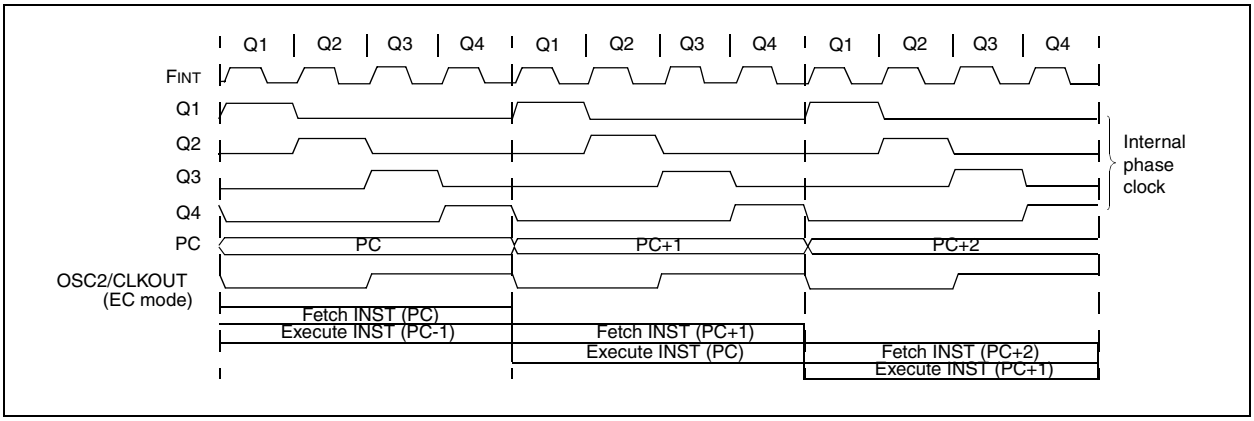
3.2 Instruction Flow/Pipelining

An “Instruction Cycle” consists of four Q cycles (Q1, Q2, Q3 and Q4). The instruction fetch and execute are pipelined such that fetch takes one instruction cycle, while decode and execute takes another instruction cycle. However, due to the pipelining, each instruction effectively executes in one cycle. If an instruction causes the program counter to change (e.g., `GOTO`), then two cycles are required to complete the instruction (Example 3-1).

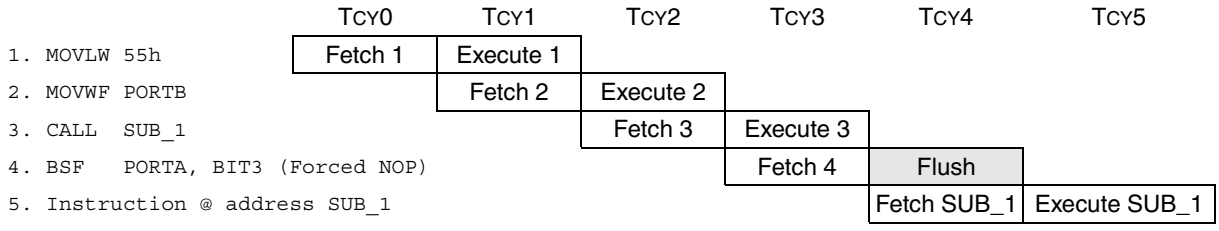
A fetch cycle begins with the program counter (PC) incrementing in Q1.

In the execution cycle, the fetched instruction is latched into the “Instruction Register" (IR) in cycle Q1. This instruction is then decoded and executed during the Q2, Q3 and Q4 cycles. Data memory is read during Q2 (operand read) and written during Q4 (destination write).

FIGURE 3-2: CLOCK/INSTRUCTION CYCLE



EXAMPLE 3-1: INSTRUCTION PIPELINE FLOW



Note: All instructions are single cycle, except for any program branches. These take two cycles, since the fetch instruction is “flushed” from the pipeline, while the new instruction is being fetched and then executed.

4.2.2.4 PIE1 REGISTER

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

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

REGISTER 4-4: PERIPHERAL INTERRUPT ENABLE1 REGISTER (PIE1: 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
PSPIE(1)	ADIE	RCIE	TXIE	USBIE	CCP1IE	TMR2IE	TMR1IE
bit7							bit0

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

bit 7: **PSPIE(1):** Parallel Slave Port Read/Write Interrupt Enable bit
1 = Enables the PSP read/write interrupt
0 = Disables the PSP read/write interrupt

bit 6: **ADIE:** A/D Converter Interrupt Enable bit
1 = Enables the A/D interrupt
0 = Disables the A/D interrupt

bit 5: **RCIE**: USART Receive Interrupt Enable bit
1 = Enables the USART receive interrupt
0 = Disables the USART receive interrupt

bit 4: **TXIE**: USART Transmit Interrupt Enable bit
1 = Enables the USART transmit interrupt
0 = Disables the USART transmit interrupt

bit 3: **USBIE**: Universal Serial Bus Interrupt Enable bit
1 = Enables the USB interrupt
0 = Disables the USB 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

Note 1: Parallel slave ports not implemented on the PIC16C745; always maintain this bit clear.

4.2.2.8 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 Watchdog Reset (WDT) and an external $\overline{MCLR}$ Reset.

Note:	$\overline{BOR}$ is unknown on POR. It must be set by the user and checked on subsequent RESETS to see if $\overline{BOR}$ is clear, indicating a brown-out has occurred.

REGISTER 4-8: POWER CONTROL REGISTER REGISTER (PCON: 8Eh)

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-q
—	—	—	—	—	—	$\overline{POR}$	$\overline{BOR}$
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: **$\overline{POR}$**: 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: **$\overline{BOR}$**: 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)

4.5 Indirect Addressing, INDF and FSR Registers

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

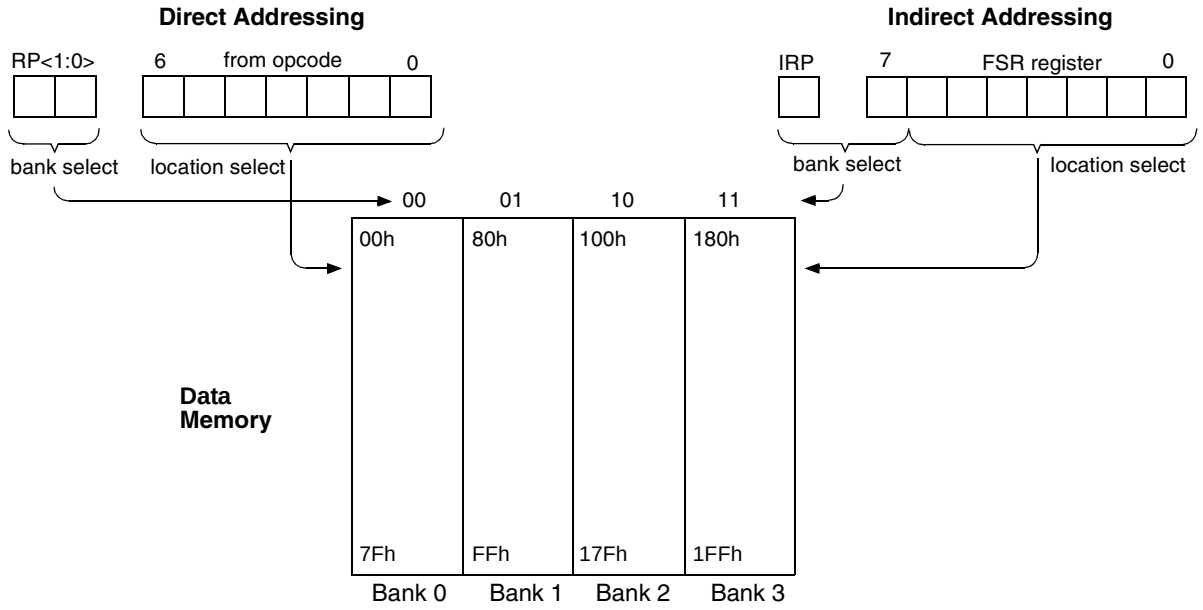
Indirect addressing is possible by using the INDF register. Any instruction using the INDF register actually accesses the register pointed to by the File Select Register, FSR. Reading the INDF register itself indirectly (FSR = '0') will read 00h. Writing to the INDF register indirectly results in a no-operation (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 4-4.

A simple program to clear RAM locations 20h-2Fh using indirect addressing is shown in Example 4-2.

EXAMPLE 4-2: INDIRECT ADDRESSING

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

FIGURE 4-4: DIRECT/INDIRECT ADDRESSING



Note: For register file map detail see Figure 4-2.

7.3 Timer1 Operation in Asynchronous Counter Mode

If control bit $\overline{\text{T1SYNC}}$ (T1CON<2>) is set, the external clock input is not synchronized. The timer continues to increment asynchronous to the internal phase clocks. The timer will continue to run during SLEEP and can generate an interrupt-on-overflow, which will wake-up the processor. However, special precautions in software are needed to read/write the timer (Section 7.3.1).

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

7.3.1 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 PIC Mid-Range MCU Family Reference Manual (DS33023) show how to read and write Timer1 when it is running in asynchronous mode.

7.4 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 7-1 shows the capacitor selection for the Timer1 oscillator.

TABLE 7-1: 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.			
Crystals Tested:			
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

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.

7.5 Resetting Timer1 using a CCP Trigger Output

If the CCP1 or CCP2 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 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 register pair effectively becomes the period register for Timer1.

7.6 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 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.

7.7 Timer1 Prescaler

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

REGISTER 9-1: CAPTURE/COMPARE/PWM CONTROL REGISTER (CCP1CON: 17H, CCP2CON: 1Dh)

U	U	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	DCnB1	DCnB0	CCPnM3	CCPnM2	CCPnM1	CCPnM0
bit7							bit0

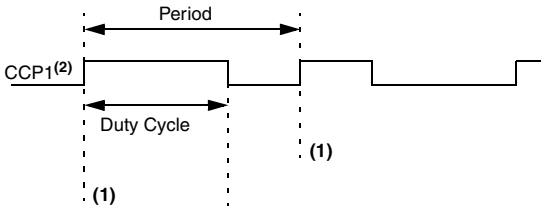
R = Readable bit
W = Writable bit
U = Unimplemented bit, read as '0'
- n = Value at POR reset

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

bit 5-4: **DCnB<1:0>:** 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 CCPRnL.

bit 3-0: **CCPnM<3:0>:** CCPx Mode Select bits
0000 = Capture/Compare/PWM off (resets CCPn 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 (CCPnIF bit is set)
1001 = Compare mode, clear output on match (CCPnIF bit is set)
1010 = Compare mode, generate software interrupt on match (CCPnIF bit is set, CCPn pin is unaffected)
1011 = Compare mode, trigger special event (CCPnIF bit is set; CCPn resets TMR1or TMR3)
11xx = PWM mode

FIGURE 9-4: PWM OUTPUT



Note 1: At this time, the TMR2 register is equal to the PR2 register.
2: Output signal is shown as asserted high.

9.3.1 PWM PERIOD

The PWM period is specified by writing to the PR2 register. The PWM period can be calculated using the following formula:

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

PWM frequency is defined as 1 / [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.

9.3.2 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:

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

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:

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

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

9.3.3 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.

10.5.1.4 Error Interrupt Enable Register (UEIE)

The USB Error Interrupt Enable Register (UEIE) contains enable bits for each of the error interrupt sources within the USB. Setting any of these bits will enable the respective error interrupt source in the UEIR register.

REGISTER 10-4: USB ERROR INTERRUPT ENABLE REGISTER (UEIE: 193h)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
BTS_ERR	OWN_ERR	WRT_ERR	BTO_ERR	DFN8	CRC16	CRC5	PID_ERR
bit7							bit0

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

bit 7: **BTS_ERR:** Set this bit to enable BTS_ERR interrupts
1 = BTS_ERR interrupt enabled
0 = BTS_ERR interrupt disabled

bit 6: **OWN_ERR:** Set this bit to enable OWN_ERR interrupts
1 = OWN_ERR interrupt enabled
0 = OWN_ERR interrupt disabled

bit 5: **WRT_ERR:** Set this bit to enable WRT_ERR interrupts
1 = WRT_ERR interrupt enabled
0 = WRT_ERR interrupt disabled

bit 4: **BTO_ERR:** Set this bit to enable BTO_ERR interrupts
1 = BTO_ERR interrupt enabled
0 = BTO_ERR interrupt disabled

bit 3: **DFN8:** Set this bit to enable DFN8 interrupts
1 = DFN8 interrupt enabled
0 = DFN8 interrupt disabled

bit 2: **CRC16:** Set this bit to enable CRC16 interrupts
1 = CRC16 interrupt enabled
0 = CRC16 interrupt disabled

bit 1: **CRC5:** Set this bit to enable CRC5 interrupts
1 = CRC5 interrupt enabled
0 = CRC5 interrupt disabled

bit 0: **PID_ERR:** Set this bit to enable PID_ERR interrupts
1 = PID_ERR interrupt enabled
0 = PID_ERR interrupt disabled

10.5.1.7 USB Address Register (UADDR)

The Address Register (UADDR) contains the unique USB address that the USB will decode. The register is reset to 00h after the RESET input has gone active or the USB has decoded a USB Reset signaling. That will initialize the address register to decode address 00h as required by the USB specification. The USB address must be written by the MCU during the USB SETUP phase.

REGISTER 10-7: USB ADDRESS REGISTER (UADDR: 196h)

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	ADDR6	ADDR5	ADDR4	ADDR3	ADDR2	ADDR1	ADDR0
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 6-0: **ADDR<6:0>:** This 7-bit value defines the USB address that the USB will decode

10.5.1.8 USB Software Status Register (USWSTAT)

This register is used by the USB firmware libraries for USB status.

Warning: Writing to this register may cause the SIE to drop off the Bus.

REGISTER 10-8: RESERVED SOFTWARE LIBRARY REGISTER (USWSTAT: 197H):.

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
7	6	5	4	3	2	1	0
Reserved for CH9 Firmware						Enumeration Status	

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

bit 7-2(1):**Reserved:** Read as "X'

bit 1-0: **W Enumeration Status <1:0>:** Status of USB peripheral during enumeration
00 = Powered
01 = Default
10 = Addressed
11 = Configured

Note 1: Application should not modify these bits.

10.5.1.9 Endpoint Registers

Each endpoint is controlled by an Endpoint Control Register. The PIC16C745/765 supports Buffer Descriptors (BD) for the following endpoints:

- EP0 Out
- EP0 In
- EP1 Out
- EP1 In
- EP2 Out
- EP2 In

The user will be required to disable unused Endpoints and directions using the Endpoint Control Registers.

10.5.1.10 USB Endpoint Control Register (EPCn)

The Endpoint Control Register contains the endpoint control bits for each of the 6 endpoints available on USB for a decoded address. These four bits define the control necessary for any one endpoint. Endpoint 0 (ENDP0) is associated with control pipe 0 which is required by USB for all functions (IN, OUT, and SETUP). Therefore, after a USB_RST interrupt has been received, the microprocessor should set UEP0 to contain 06h.

Note: These registers are initialized in response to a RESET from the host. The user must modify function USBReset in USB_CH9.ASM to configure the endpoints as needed for the application.

REGISTER 10-9: USB ENDPOINT CONTROL REGISTER (UEPn: 198H-19Ah)

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	EP_CTL_DIS	EP_OUT_EN	EP_IN_EN	EP_STALL
bit7							bit0

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

bit 7-4: **Unimplemented:** Read as '0'

bit 3-1: **EP_CTL_DIS, EP_OUT_EN, EP_IN_EN:** These three bits define if an endpoint is enabled and the direction of the endpoint. The endpoint enable/direction control is defined as follows:

EP_CTL_DIS	EP_OUT_EN	EP_IN_EN	Endpoint Enable/Direction Control
X	0	0	Disable Endpoint
X	0	1	Enable Endpoint for IN tokens only
X	1	0	Enable Endpoint for OUT tokens only
1	1	1	Enable Endpoint for IN and OUT tokens
0	1	1	Enable Endpoint for IN, OUT, and SETUP tokens

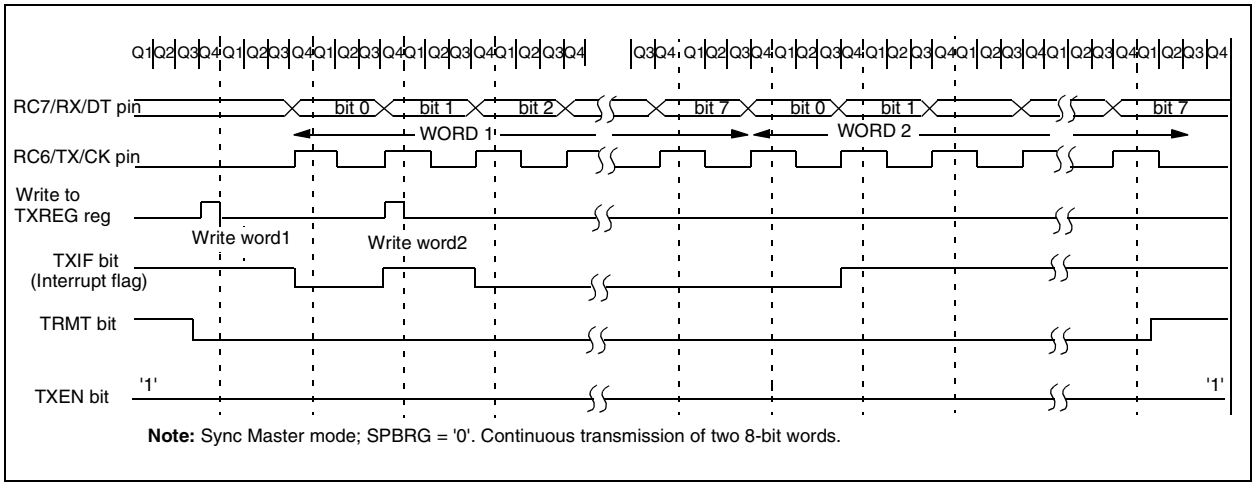
bit 0: **EP_STALL:** When this bit is set it indicates that the endpoint is stalled. This bit has priority over all other control bits in the Endpoint Enable register, but is only valid if EP_IN_EN=1 or EP_OUT_EN=1. Any access to this endpoint will cause the USB to return a STALL handshake. The EP_STALL bit can be set or cleared by the SIE. Refer to the USB 1.1 Specification, Sections 4.4.4 and 8.5.2 for more details on the STALL protocol.

TABLE 11-8: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER TRANSMISSION

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
0Ch	PIR1	PSPIF(1)	ADIF	RCIF	TXIF	USBIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000 -00x	0000 -00x
19h	TXREG	USART Transmit Register								0000 0000	0000 0000
8Ch	PIE1	PSPIE(1)	ADIE	RCIE	TXIE	USBIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

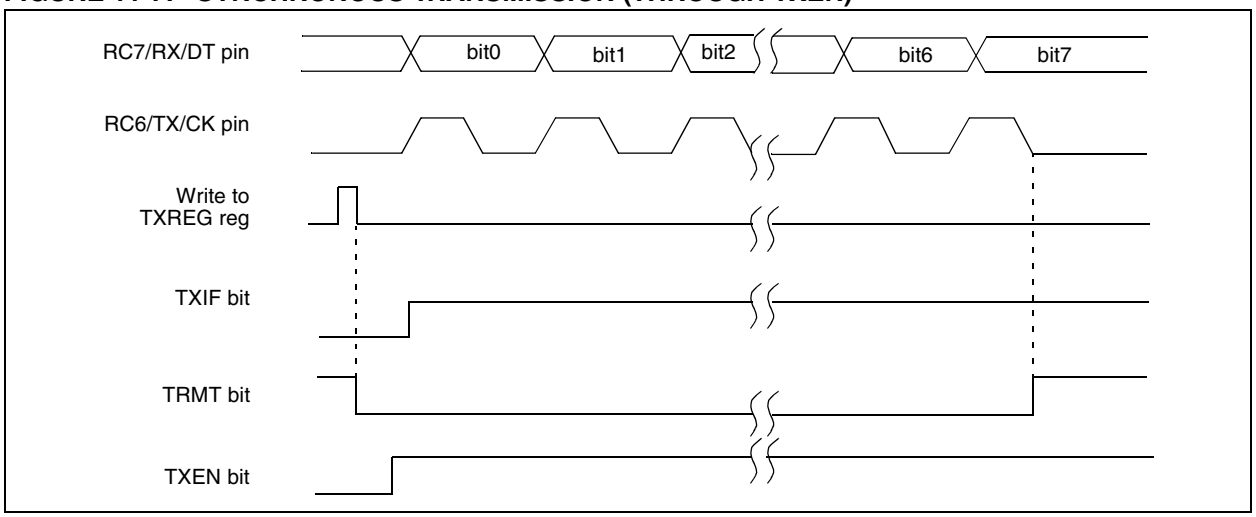
Legend: x = unknown, - = unimplemented, read as '0'. Shaded cells are not used for synchronous master transmission.
Note 1: Bits PSPIE and PSPIF are reserved on the PIC16C745; always maintain these bits clear.

FIGURE 11-6: SYNCHRONOUS TRANSMISSION



Note: Sync Master mode; SPBRG = '0'. Continuous transmission of two 8-bit words.

FIGURE 11-7: SYNCHRONOUS TRANSMISSION (THROUGH TXEN)



12.7 Use of the CCP Trigger

An A/D conversion can be started by the "special event trigger" of the CCP2 module. This requires that the CCP2M<3:0> 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/$\overline{\text{DONE}}$ 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/$\overline{\text{DONE}}$ bit (starts a conversion).

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

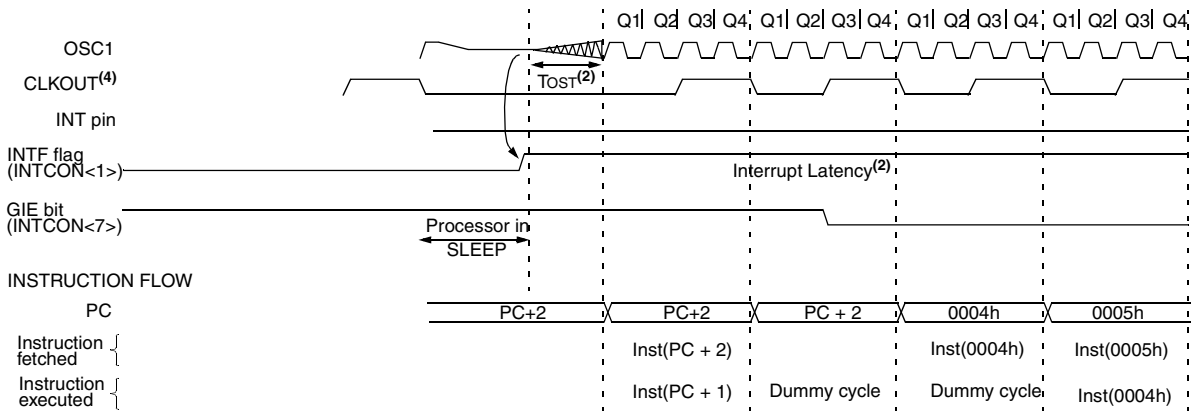
TABLE 12-2: SUMMARY OF A/D REGISTERS

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	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF(1)	ADIF	RCIF	TXIF	USBIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE(1)	ADIE	RCIE	TXIE	USBIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
1Eh	ADRES	A/D Result Register								xxxx xxxx	uuuu uuuu
1Fh	ADCON0	ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/$\overline{\text{DONE}}$	—	ADON	0000 00-0	0000 00-0
9Fh	ADCON1	—	—	—	—	—	PCFG2	PCFG1	PCFG0	---- -000	---- -000
05h	PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	--0x 0000	--0u 0000
85h	TRISA	—	—	PORTA Data Direction Register						--11 1111	--11 1111
09h	PORTE	—	—	—	—	—	RE2(1)	RE1(1)	RE0(1)	---- -xxx	---- -uuu
89h	TRISE	IBF(1)	OBF(1)	IBOV(1)	PSP-MODE(1)	—	PORTE(1) Data Direction Bits			0000 -111	0000 -111

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

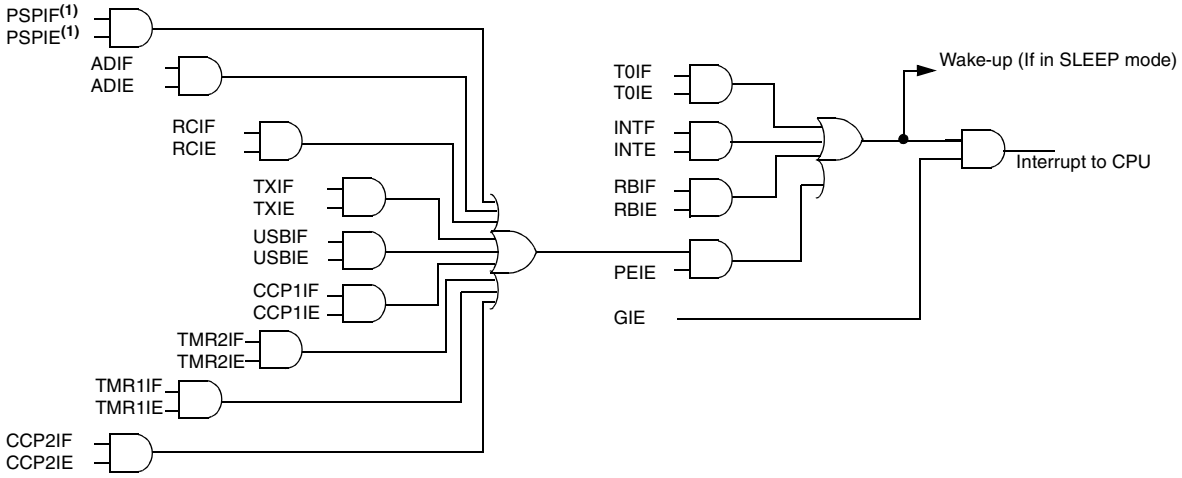
Note 1: These bits are reserved on the PIC6C745; always maintain these bits clear.

FIGURE 13-5: WAKE-UP FROM SLEEP THROUGH INTERRUPT



Note 1: HS oscillator mode assumed.
2: TOST = 1024TOSC (drawing not to scale). This delay is not present in EC osc mode.
3: GIE = '1' assumed. 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.

FIGURE 13-6: INTERRUPT LOGIC



The following table shows the interrupts for each device.

Device	T0IF	INTF	RBIF	PSPIF	ADIF	RCIF	TXIF	USBIF	CCP1IF	TMR2IF	TMR1IF	CCP2IF
PIC16C745	Yes	Yes	Yes	—	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
PIC16C765	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Note 1: PIC16C765 only.

13.9 Power-Down Mode (SLEEP)

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

If enabled, the WDT 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 VDD or VSS, ensure no external circuitry is drawing current from the I/O pin, power-down the A/D, and disable external clocks. Pull all I/O pins that are hi-impedance inputs, high or low externally, to avoid switching currents caused by floating inputs. The T0CKI input should also be at VDD or VSS 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 (VIHMC).

13.9.1 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. USB interrupt.
3. CCP capture mode interrupt.
4. Parallel slave port read or write (PIC16C765 only).
5. A/D conversion (when A/D clock source is dedicated internal oscillator).
6. USART TX or RX (Synchronous Slave mode).

Other peripherals cannot generate interrupts, since during SLEEP, no on-chip Q 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.

13.9.2 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 $\overline{PD}$ bit 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.

16.0 ELECTRICAL CHARACTERISTICS

Absolute Maximum Ratings (†)

Ambient temperature under bias ..... -55°C to +125°C
Storage temperature ..... -65°C to +150°C
Voltage on any pin with respect to V_{SS} (except V_{DD}, $\overline{MCLR}$ and RA4) ..... -0.3V to (V_{DD} + 0.3V)
Voltage on V_{DD} with respect to V_{SS} ..... -0.3V to +7.5V
Voltage on $\overline{MCLR}$ with respect to V_{SS} ..... -0.3V to +13.25V
Voltage on RA4 with respect to Vss ..... -0.3V to +10.5V
Total power dissipation **(Note 1)** 1.0W
Maximum current out of V_{SS} pin ..... 300 mA
Maximum current into V_{DD} pin ..... 250 mA
Input clamp current, I_{IK} (V_I < 0 or V_I > V_{DD}) ..... ±20 mA
Output clamp current, I_{OK} (V_O < 0 or V_O > V_{DD}) ..... ±20 mA
Maximum output current sunk by any I/O pin ..... 25 mA
Maximum output current sourced by any I/O pin ..... 25 mA
Maximum current sunk by PORTA, PORTB, and PORTE **(Note 2)** (combined) ..... 200 mA
Maximum current sourced by PORTA, PORTB, and PORTE **(Note 2)** (combined) ..... 200 mA
Maximum current sunk by PORTC and PORTD **(Note 2)** (combined) ..... 200 mA
Maximum current sourced by PORTC and PORTD **(Note 2)** (combined) ..... 200 mA

Note 1: Power dissipation is calculated as follows: Pdis = V_{DD} x {I_{DD} - $\sum I_{OH}$} + $\sum$ {(V_{DD}-V_{OH}) x I_{OH}} + $\sum$(V_{OL} x I_{OL})

2: PORTD and PORTE not available on the PIC16C745.

† 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.

16.3 AC (Timing) Characteristics

16.3.1 TIMING PARAMETER SYMBOLOGY

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

1. TppS2ppS
2. TppS

T			
F	Frequency	T	Time

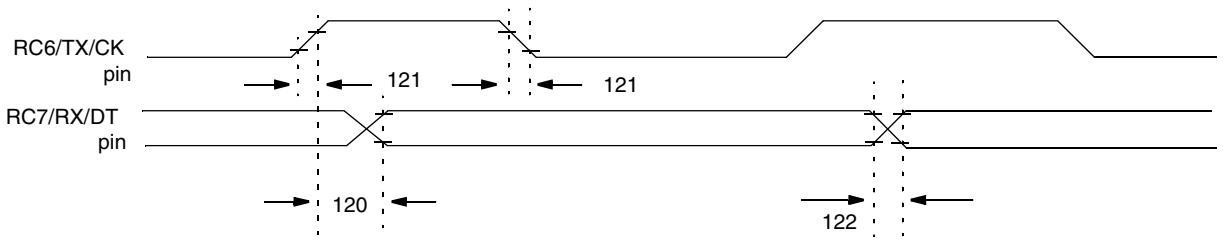
Lowercase letters (pp) and their meanings:

pp			
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:

S			
F	Fall	P	Period
H	High	R	Rise
I	Invalid (Hi-impedance)	V	Valid
L	Low	Z	Hi-impedance

FIGURE 16-11: USART SYNCHRONOUS TRANSMISSION (MASTER/SLAVE) TIMING



Note: Refer to Figure 16-2 for load conditions.

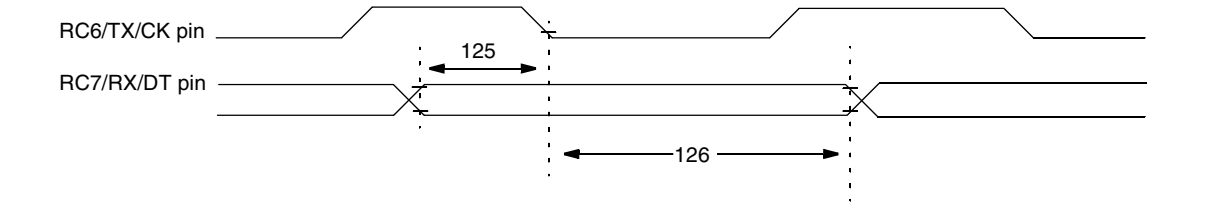
TABLE 16-8: USART SYNCHRONOUS TRANSMISSION REQUIREMENTS

Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
120*	TCKH2DTV	SYNC XMIT (MASTER & SLAVE) Clock high to data out valid	—	—	80	ns	
121*	TCKRF	Clock out rise time and fall time (Master mode)	—	—	45	ns	
122*	TDTRF	Data out rise time and fall time	—	—	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 16-12: USART SYNCHRONOUS RECEIVE (MASTER/SLAVE) TIMING



Note: Refer to Figure 16-2 for load conditions.

TABLE 16-9: USART SYNCHRONOUS RECEIVE REQUIREMENTS

Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
125*	TDTV2CKL	SYNC RCV (MASTER & SLAVE) Data setup before CK ↓ (DT setup time)	15	—	—	ns	
126*	TCKL2DTL	Data hold after CK ↓ (DT hold time)	15	—	—	ns	

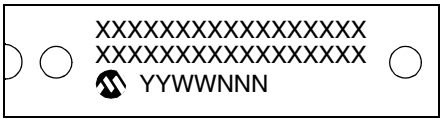
*These parameters are characterized but not tested.

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

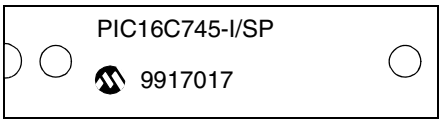
18.0 PACKAGING INFORMATION

18.1 Package Marking Information

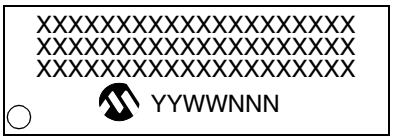
28-Lead PDIP (Skinny DIP)



Example

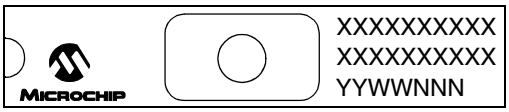


28-Lead SOIC

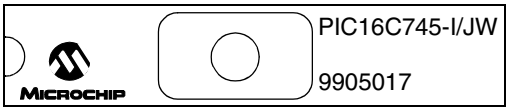


Example

28-Lead Side Braze Windowed (JW)



Example

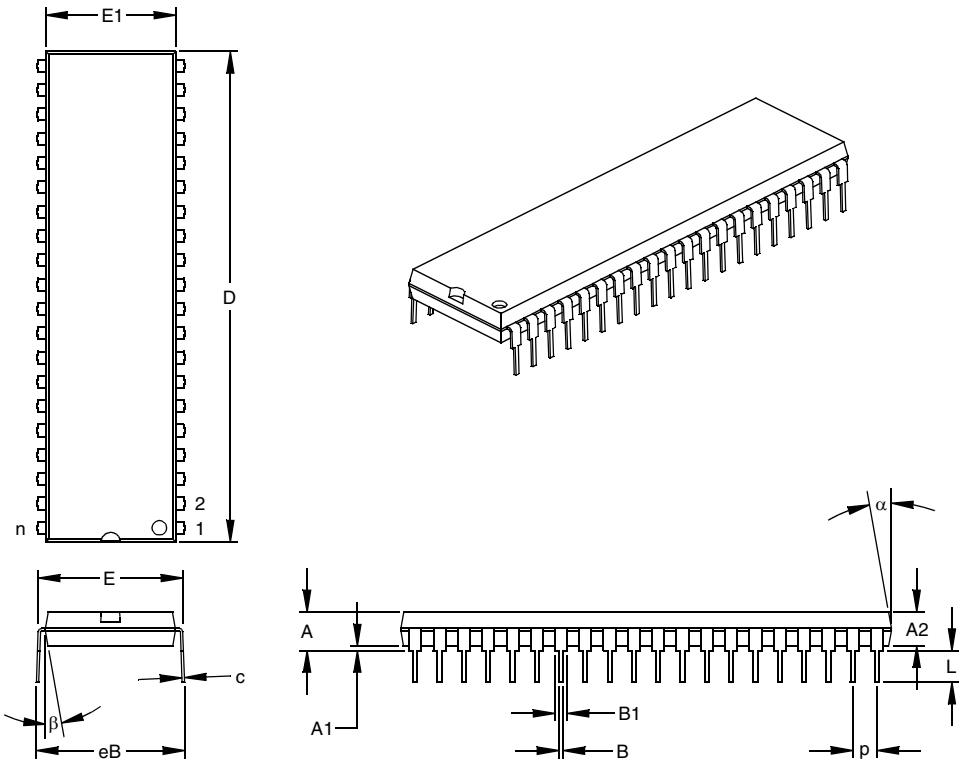


Legend:	XX...X	Customer-specific information
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week ‘01’)
	NNN	Alphanumeric traceability code
	(e3)	Pb-free JEDEC designator for Matte Tin (Sn)
	*	This package is Pb-free. The Pb-free JEDEC designator ((e3)) can be found on the outer packaging for this package.

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.

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

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging

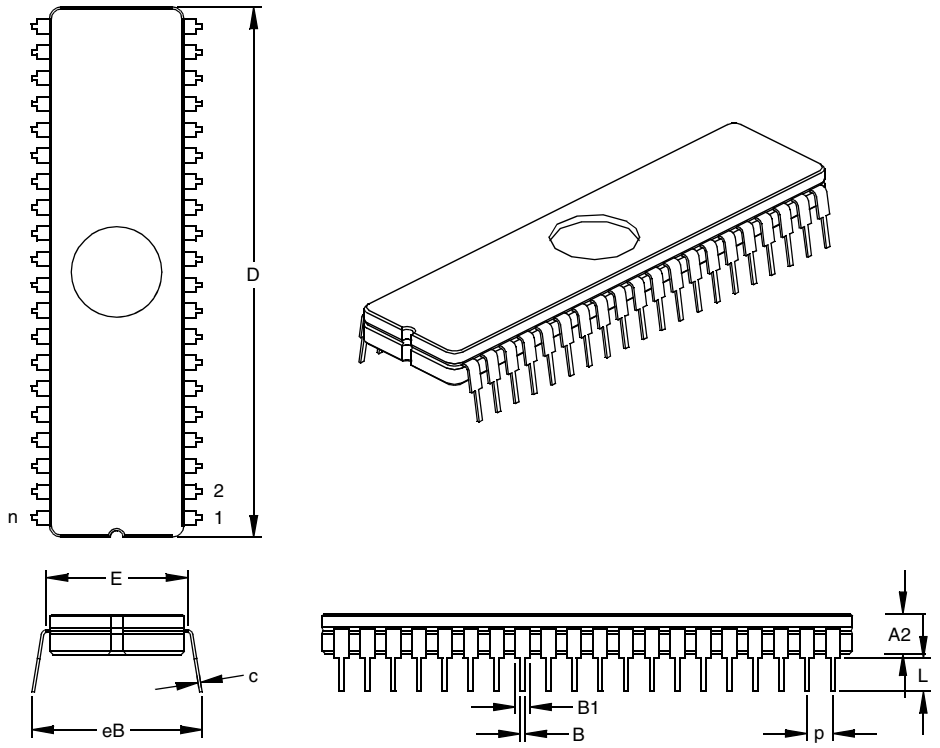


Units		INCHES*			MILLIMETERS		
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		40			40	
Pitch	p		.100			2.54	
Top to Seating Plane	A	.160	.175	.190	4.06	4.45	4.83
Molded Package Thickness	A2	.140	.150	.160	3.56	3.81	4.06
Base to Seating Plane	A1	.015			0.38		
Shoulder to Shoulder Width	E	.595	.600	.625	15.11	15.24	15.88
Molded Package Width	E1	.530	.545	.560	13.46	13.84	14.22
Overall Length	D	2.045	2.058	2.065	51.94	52.26	52.45
Tip to Seating Plane	L	.120	.130	.135	3.05	3.30	3.43
Lead Thickness	c	.008	.012	.015	0.20	0.29	0.38
Upper Lead Width	B1	.030	.050	.070	0.76	1.27	1.78
Lower Lead Width	B	.014	.018	.022	0.36	0.46	0.56
Overall Row Spacing §	eB	.620	.650	.680	15.75	16.51	17.27
Mold Draft Angle Top	α	5	10	15	5	10	15
Mold Draft Angle Bottom	β	5	10	15	5	10	15

* Controlling Parameter
§ Significant Characteristic
Notes:
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: MO-011
Drawing No. C04-016

40-Lead Ceramic Dual In-line with Window (JW) – 600 mil (CERDIP)

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Units		INCHES*			MILLIMETERS		
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		40			40	
Pitch	p		.100			2.54	
Top to Seating Plane	A	.185	.205	.225	4.70	5.21	5.72
Ceramic Package Height	A2	.155	.160	.165	3.94	4.06	4.19
Standoff	A1	.030	.045	.060	0.76	1.14	1.52
Shoulder to Shoulder Width	E	.595	.600	.625	15.11	15.24	15.88
Ceramic Pkg. Width	E1	.514	.520	.526	13.06	13.21	13.36
Overall Length	D	2.040	2.050	2.060	51.82	52.07	52.32
Tip to Seating Plane	L	.135	.140	.145	3.43	3.56	3.68
Lead Thickness	c	.008	.011	.014	0.20	0.28	0.36
Upper Lead Width	B	.050	.053	.055	1.27	1.33	1.40
Lower Lead Width	B1	.016	.020	.023	0.41	0.51	0.58
Overall Row Spacing §	eB	.610	.660	.710	15.49	16.76	18.03
Window Diameter	W	.340	.350	.360	8.64	8.89	9.14

* Controlling Parameter
§ Significant Characteristic
JEDEC Equivalent: MO-103
Drawing No. C04-014