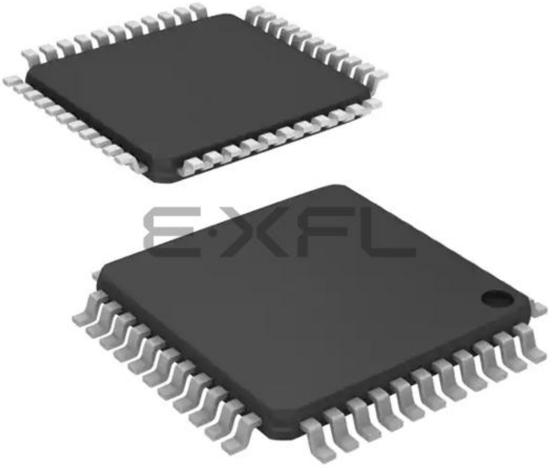
E·XFL

# Microchip Technology - PIC17C42AT-33E/PT Datasheet



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 | Obsolete |
| Core Processor | PIC |
| Core Size | 8-Bit |
| Speed | 33MHz |
| Connectivity | UART/USART |
| Peripherals | POR, PWM, WDT |
| Number of I/O | 33 |
| Program Memory Size | 4KB (2K x 16) |
| Program Memory Type | OTP |
| EEPROM Size | - |
| RAM Size | 232 x 8 |
| Voltage - Supply (Vcc/Vdd) | 4.5V ~ 6V |
| Data Converters | - |
| Oscillator Type | External |
| Operating Temperature | -40°C ~ 125°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/pic17c42at-33e-pt |

NOTES:

© 1996 Microchip Technology Inc.

## 1.0 OVERVIEW

This data sheet covers the PIC17C4X group of the PIC17CXX family of microcontrollers. The following devices are discussed in this data sheet:

- PIC17C42
- PIC17CR42
- PIC17C42A
- PIC17C43
- PIC17CR43
- PIC17C44

The PIC17CR42, PIC17C42A, PIC17C43, PIC17CR43, and PIC17C44 devices include architectural enhancements over the PIC17C42. These enhancements will be discussed throughout this data sheet.

The PIC17C4X devices are 40/44-Pin, EPROM/ROM-based members of the versatile PIC17CXX family of low-cost, high-performance, CMOS, fully-static, 8-bit microcontrollers.

All PIC16/17 microcontrollers employ an advanced RISC architecture. The PIC17CXX has enhanced core features, 16-level deep stack, and multiple internal and external interrupt sources. The separate instruction and data buses of the Harvard architecture allow a 16-bit wide instruction word with a separate 8-bit wide data. The two stage instruction pipeline allows all instructions to execute in a single cycle, except for program branches (which require two cycles). A total of 55 instructions (reduced instruction set) are available in the PIC17C42 and 58 instructions in all the other devices. Additionally, a large register set gives some of the architectural innovations used to achieve a very high performance. For mathematical intensive applications all devices, except the PIC17C42, have a single cycle 8 x 8 Hardware Multiplier.

PIC17CXX microcontrollers typically achieve a 2:1 code compression and a 4:1 speed improvement over other 8-bit microcontrollers in their class.

PIC17C4X devices have up to 454 bytes of RAM and 33 I/O pins. In addition, the PIC17C4X adds several peripheral features useful in many high performance applications including:

- Four timer/counters
- Two capture inputs
- Two PWM outputs
- A Universal Synchronous Asynchronous Receiver Transmitter (USART)

These special features reduce external components, thus reducing cost, enhancing system reliability and reducing power consumption. There are four oscillator options, of which the single pin RC oscillator provides a low-cost solution, the LF oscillator is for low frequency crystals and minimizes power consumption, XT is a standard crystal, and the EC is for external clock input. The SLEEP (power-down) mode offers additional power saving. The user can wake-up the chip from SLEEP through several external and internal interrupts and device resets.

There are four configuration options for the device operational modes:

- Microprocessor
- Microcontroller
- Extended microcontroller
- Protected microcontroller

The microprocessor and extended microcontroller modes allow up to 64K-words of external program memory.

A highly reliable Watchdog Timer with its own on-chip RC oscillator provides protection against software malfunction.

Table 1-1 lists the features of the PIC17C4X devices.

A UV-erasable CERDIP-packaged version is ideal for code development while the cost-effective One-Time Programmable (OTP) version is suitable for production in any volume.

The PIC17C4X fits perfectly in applications ranging from precise motor control and industrial process control to automotive, instrumentation, and telecom applications. Other applications that require extremely fast execution of complex software programs or the flexibility of programming the software code as one of the last steps of the manufacturing process would also be well suited. The EPROM technology makes customization of application programs (with unique security codes, combinations, model numbers, parameter storage, etc.) fast and convenient. Small footprint package options make the PIC17C4X ideal for applications with space limitations that require high performance. High speed execution, powerful peripheral features, flexible I/O, and low power consumption all at low cost make the PIC17C4X ideal for a wide range of embedded control applications.

### 1.1 Family and Upward Compatibility

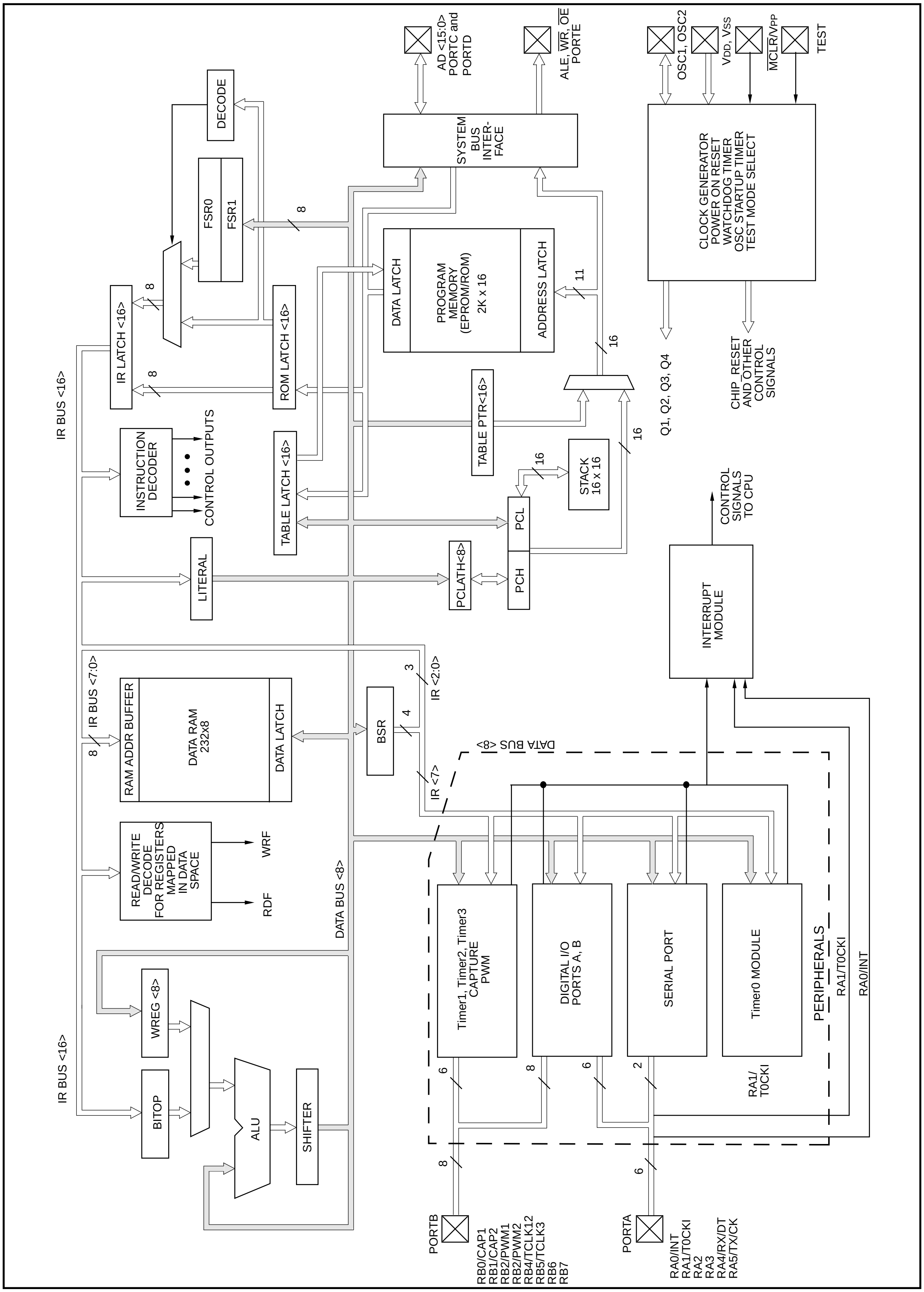
Those users familiar with the PIC16C5X and PIC16CXX families of microcontrollers will see the architectural enhancements that have been implemented. These enhancements allow the device to be more efficient in software and hardware requirements. Please refer to Appendix A for a detailed list of enhancements and modifications. Code written for PIC16C5X or PIC16CXX can be easily ported to PIC17CXX family of devices (Appendix B).

### 1.2 Development Support

The PIC17CXX family is supported by a full-featured macro assembler, a software simulator, an in-circuit emulator, a universal programmer, a "C" compiler, and fuzzy logic support tools.

This document was created with FrameMaker 4.0.4

**FIGURE 3-1: PIC17C42 BLOCK DIAGRAM**



© 1996 Microchip Technology Inc.

## 5.1 Interrupt Status Register (INTSTA)

The Interrupt Status/Control register (INTSTA) records the individual interrupt requests in flag bits, and contains the individual interrupt enable bits (not for the peripherals).

The PEIF bit is a read only, bit wise OR of all the peripheral flag bits in the PIR register (Figure 5-4).

| Note: | T0IF, INTF, T0CKIF, or PEIF will be set by the specified condition, even if the corresponding interrupt enable bit is clear (interrupt disabled) or the GLINTD bit is set (all interrupts disabled). |
|---|---|

Care should be taken when clearing any of the INTSTA register enable bits when interrupts are enabled (GLINTD is clear). If any of the INTSTA flag bits (T0IF, INTF, T0CKIF, or PEIF) are set in the same instruction cycle as the corresponding interrupt enable bit is cleared, the device will vector to the reset address (0x00).

When disabling any of the INTSTA enable bits, the GLINTD bit should be set (disabled).

### FIGURE 5-2: INTSTA REGISTER (ADDRESS: 07h, UNBANKED)

| R - 0 | R/W - 0 | R/W - 0 | R/W - 0 | R/W - 0 | R/W - 0 | R/W - 0 | R/W - 0 |
|---|---|---|---|---|---|---|---|
| PEIF | T0CKIF | T0IF | INTF | PEIE | T0CKIE | T0IE | INTE |
| bit7 | | | | | | | bit0 |

R = Readable bit
W = Writable bit
- n = Value at POR reset

bit 7: **PEIF**: Peripheral Interrupt Flag bit
This bit is the OR of all peripheral interrupt flag bits AND'ed with their corresponding enable bits.
1 = A peripheral interrupt is pending
0 = No peripheral interrupt is pending

bit 6: **T0CKIF**: External Interrupt on T0CKI Pin Flag bit
This bit is cleared by hardware, when the interrupt logic forces program execution to vector (18h).
1 = The software specified edge occurred on the RA1/T0CKI pin
0 = The software specified edge did not occur on the RA1/T0CKI pin

bit 5: **T0IF**: TMR0 Overflow Interrupt Flag bit
This bit is cleared by hardware, when the interrupt logic forces program execution to vector (10h).
1 = TMR0 overflowed
0 = TMR0 did not overflow

bit 4: **INTF**: External Interrupt on INT Pin Flag bit
This bit is cleared by hardware, when the interrupt logic forces program execution to vector (08h).
1 = The software specified edge occurred on the RA0/INT pin
0 = The software specified edge did not occur on the RA0/INT pin

bit 3: **PEIE**: Peripheral Interrupt Enable bit
This bit enables all peripheral interrupts that have their corresponding enable bits set.
1 = Enable peripheral interrupts
0 = Disable peripheral interrupts

bit 2: **T0CKIE**: External Interrupt on T0CKI Pin Enable bit
1 = Enable software specified edge interrupt on the RA1/T0CKI pin
0 = Disable interrupt on the RA1/T0CKI pin

bit 1: **T0IE**: TMR0 Overflow Interrupt Enable bit
1 = Enable TMR0 overflow interrupt
0 = Disable TMR0 overflow interrupt

bit 0: **INTE**: External Interrupt on RA0/INT Pin Enable bit
1 = Enable software specified edge interrupt on the RA0/INT pin
0 = Disable software specified edge interrupt on the RA0/INT pin

© 1996 Microchip Technology Inc.

## 5.2 Peripheral Interrupt Enable Register (PIE)

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

### FIGURE 5-3: PIE REGISTER (ADDRESS: 17h, BANK 1)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| R/W - 0 | R/W - 0 | R/W - 0 | R/W - 0 | R/W - 0 | R/W - 0 | R/W - 0 | R/W - 0 |
| RBIE | TMR3IE | TMR2IE | TMR1IE | CA2IE | CA1IE | TXIE | RCIE |
| bit7 | | | | | | | bit0 |

R = Readable bit
W = Writable bit
-n = Value at POR reset

bit 7: **RBIE**: PORTB Interrupt on Change Enable bit
1 = Enable PORTB interrupt on change
0 = Disable PORTB interrupt on change

bit 6: **TMR3IE**: Timer3 Interrupt Enable bit
1 = Enable Timer3 interrupt
0 = Disable Timer3 interrupt

bit 5: **TMR2IE**: Timer2 Interrupt Enable bit
1 = Enable Timer2 interrupt
0 = Disable Timer2 interrupt

bit 4: **TMR1IE**: Timer1 Interrupt Enable bit
1 = Enable Timer1 interrupt
0 = Disable Timer1 interrupt

bit 3: **CA2IE**: Capture2 Interrupt Enable bit
1 = Enable Capture interrupt on RB1/CAP2 pin
0 = Disable Capture interrupt on RB1/CAP2 pin

bit 2: **CA1IE**: Capture1 Interrupt Enable bit
1 = Enable Capture interrupt on RB2/CAP1 pin
0 = Disable Capture interrupt on RB2/CAP1 pin

bit 1: **TXIE**: USART Transmit Interrupt Enable bit
1 = Enable Transmit buffer empty interrupt
0 = Disable Transmit buffer empty interrupt

bit 0: **RCIE**: USART Receive Interrupt Enable bit
1 = Enable Receive buffer full interrupt
0 = Disable Receive buffer full interrupt

© 1996 Microchip Technology Inc.

## 5.4 Interrupt Operation

Global Interrupt Disable bit, GLINTD (CPUSTA<4>), enables all unmasked interrupts (if clear) or disables all interrupts (if set). Individual interrupts can be disabled through their corresponding enable bits in the INTSTA register. Peripheral interrupts need either the global peripheral enable PEIE bit disabled, or the specific peripheral enable bit disabled. Disabling the peripherals via the global peripheral enable bit, disables all peripheral interrupts. GLINTD is set on reset (interrupts disabled).

The `RETFIE` instruction allows returning from interrupt and re-enable interrupts at the same time.

When an interrupt is responded to, the GLINTD bit is automatically set to disable any further interrupt, the return address is pushed onto the stack and the PC is loaded with interrupt vector. There are four interrupt vectors to reduce interrupt latency.

The peripheral interrupt vector has multiple interrupt sources. Once in the peripheral interrupt service routine, the source(s) of the interrupt can be determined by polling the interrupt flag bits. The peripheral interrupt flag bit(s) must be cleared in software before re-enabling interrupts to avoid continuous interrupts.

The PIC17C4X devices have four interrupt vectors. These vectors and their hardware priority are shown in Table 5-1. If two enabled interrupts occur "at the same time", the interrupt of the highest priority will be serviced first. This means that the vector address of that interrupt will be loaded into the program counter (PC).

**TABLE 5-1: INTERRUPT VECTORS/ PRIORITIES**

| Address | Vector | Priority |
|---|---|---|
| 0008h | External Interrupt on RA0/ INT pin (INTF) | 1 (Highest) |
| 0010h | TMR0 overflow interrupt (T0IF) | 2 |
| 0018h | External Interrupt on T0CKI (T0CKIF) | 3 |
| 0020h | Peripherals (PEIF) | 4 (Lowest) |

**Note 1:** Individual interrupt flag bits are set regardless of the status of their corresponding mask bit or the GLINTD bit.

**Note 2:** When disabling any of the INTSTA enable bits, the GLINTD bit should be set (disabled).

**Note 3:** For the PIC17C42 only:
If an interrupt occurs while the Global Interrupt Disable (GLINTD) bit is being set, the GLINTD 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 interrupt occurs simultaneously with an instruction that sets the GLINTD bit.
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 GLINTD bit to be cleared (enables interrupts), and the program returns to the instruction after the one which was meant to disable interrupts.

The method to ensure that interrupts are globally disabled is:

1. Ensure that the GLINTD bit was set by the instruction, as shown in the following code:

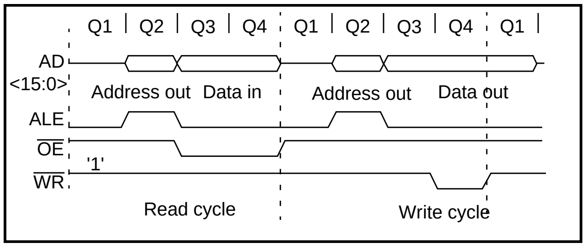
```
LOOP   BSF    CPUSTA, GLINTD ; Disable Global
                             ; Interrupt
       BTFSS  CPUSTA, GLINTD ; Global Interrupt
                             ; Disabled?
       GOTO   LOOP           ; NO, try again
                             ; YES, continue
                             ; with program
                             ; low
```

© 1996 Microchip Technology Inc.

### 6.1.2 EXTERNAL MEMORY INTERFACE

When either microprocessor or extended microcontroller mode is selected, PORTC, PORTD and PORTE are configured as the system bus. PORTC and PORTD are the multiplexed address/data bus and PORTE is for the control signals. External components are needed to demultiplex the address and data. This can be done as shown in Figure 6-4. The waveforms of address and data are shown in Figure 6-3. For complete timings, please refer to the electrical specification section.

FIGURE 6-3: EXTERNAL PROGRAM MEMORY ACCESS WAVEFORMS



The system bus requires that there is no bus conflict (minimal leakage), so the output value (address) will be capacitively held at the desired value.

As the speed of the processor increases, external EPROM memory with faster access time must be used. Table 6-2 lists external memory speed requirements for a given PIC17C4X device frequency.

In extended microcontroller mode, when the device is executing out of internal memory, the control signals will continue to be active. That is, they indicate the action that is occurring in the internal memory. The external memory access is ignored.

This following selection is for use with Microchip EPROMs. For interfacing to other manufacturers memory, please refer to the electrical specifications of the desired PIC17C4X device, as well as the desired memory device to ensure compatibility.

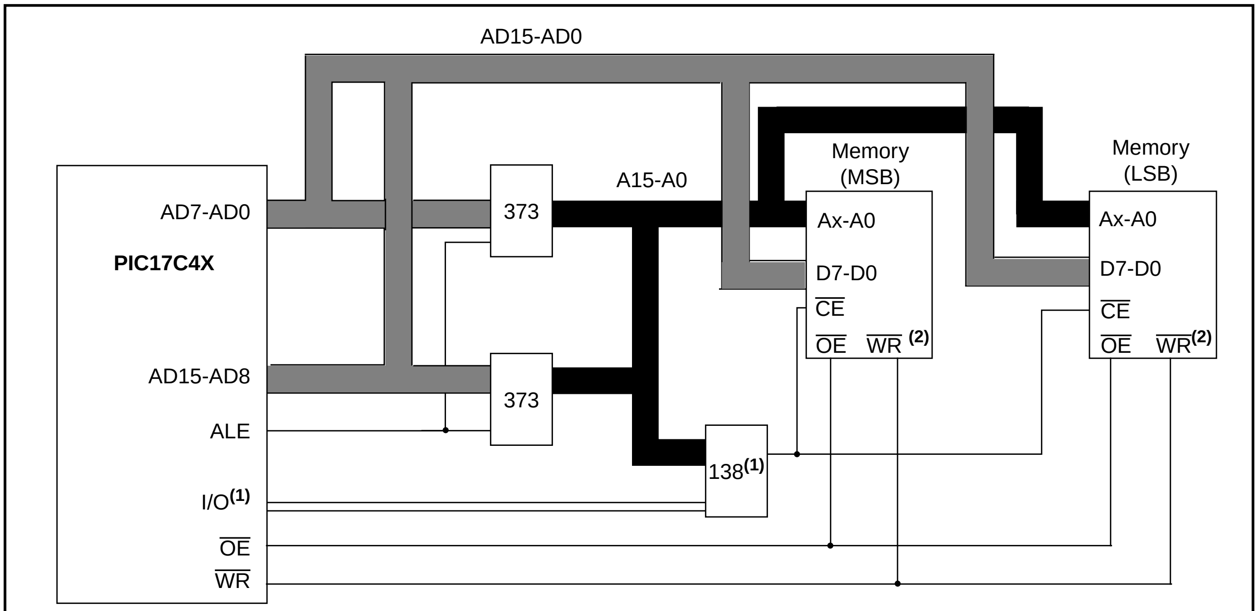
TABLE 6-2: EPROM MEMORY ACCESS TIME ORDERING SUFFIX

| PIC17C4X Oscillator Frequency | Instruction Cycle Time (TCY) | EPROM Suffix | |
|---|---|---|---|
| | | PIC17C42 | PIC17C43<br>PIC17C44 |
| 8 MHz | 500 ns | -25 | -25 |
| 16 MHz | 250 ns | -12 | -15 |
| 20 MHz | 200 ns | -90 | -10 |
| 25 MHz | 160 ns | N.A. | -70 |
| 33 MHz | 121 ns | N.A. | (1) |

Note 1: The access times for this requires the use of fast SRAMS.

**Note:** The external memory interface is not supported for the LC devices.

FIGURE 6-4: TYPICAL EXTERNAL PROGRAM MEMORY CONNECTION DIAGRAM



Note 1: Use of I/O pins is only required for paged memory.
2: This signal is unused for ROM and EPROM devices.

Example 8-4 shows the sequence to do an 16 x 16 signed multiply. Equation 8-2 shows the algorithm that used. The 32-bit result is stored in four registers RES3:RES0. To account for the sign bits of the arguments, each argument pairs most significant bit (MSb) is tested and the appropriate subtractions are done.

### EQUATION 8-2: 16 x 16 SIGNED MULTIPLICATION ALGORITHM

RES3:RES0

= ARG1H:ARG1L * ARG2H:ARG2L

= (ARG1H * ARG2H * $2^{16}$) +

(ARG1H * ARG2L * $2^{8}$) +

(ARG1L * ARG2H * $2^{8}$) +

(ARG1L * ARG2L) +

(-1 * ARG2H<7> * ARG1H:ARG1L * $2^{16}$) +

(-1 * ARG1H<7> * ARG2H:ARG2L * $2^{16}$)

### EXAMPLE 8-4: 16 x 16 SIGNED MULTIPLY ROUTINE

```
   MOVFP    ARG1L, WREG
   MULWF    ARG2L       ; ARG1L * ARG2L ->
                        ;   PRODH:PRODL
   MOVPF    PRODH, RES1 ;
   MOVPF    PRODL, RES0 ;
;
   MOVFP    ARG1H, WREG
   MULWF    ARG2H       ; ARG1H * ARG2H ->
                        ;   PRODH:PRODL
   MOVPF    PRODH, RES3 ;
   MOVPF    PRODL, RES2 ;
;
   MOVFP    ARG1L, WREG
   MULWF    ARG2H       ; ARG1L * ARG2H ->
                        ;   PRODH:PRODL
   MOVFP    PRODL, WREG ;
   ADDWF    RES1, F     ; Add cross
   MOVFP    PRODH, WREG ;    products
   ADDWFC   RES2, F     ;
   CLRF     WREG, F     ;
   ADDWFC   RES3, F     ;
;
   MOVFP    ARG1H, WREG ;
   MULWF    ARG2L       ; ARG1H * ARG2L ->
                        ;   PRODH:PRODL

   MOVFP    PRODL, WREG ;
   ADDWF    RES1, F     ; Add cross
   MOVFP    PRODH, WREG ;    products
   ADDWFC   RES2, F     ;
   CLRF     WREG, F     ;
   ADDWFC   RES3, F     ;
;
   BTFSS    ARG2H, 7    ; ARG2H:ARG2L neg?
   GOTO     SIGN_ARG1   ; no, check ARG1
   MOVFP    ARG1L, WREG ;
   SUBWF    RES2        ;
   MOVFP    ARG1H, WREG ;
   SUBWFB   RES3
;
SIGN_ARG1
   BTFSS    ARG1H, 7    ; ARG1H:ARG1L neg?
   GOTO     CONT_CODE   ; no, done
   MOVFP    ARG2L, WREG ;
   SUBWF    RES2        ;
   MOVFP    ARG2H, WREG ;
   SUBWFB   RES3
;
CONT_CODE
     :
```

### TABLE 9-5: PORTC FUNCTIONS

| Name | Bit | Buffer Type | Function |
|---|---|---|---|
| RC0/AD0 | bit0 | TTL | Input/Output or system bus address/data pin. |
| RC1/AD1 | bit1 | TTL | Input/Output or system bus address/data pin. |
| RC2/AD2 | bit2 | TTL | Input/Output or system bus address/data pin. |
| RC3/AD3 | bit3 | TTL | Input/Output or system bus address/data pin. |
| RC4/AD4 | bit4 | TTL | Input/Output or system bus address/data pin. |
| RC5/AD5 | bit5 | TTL | Input/Output or system bus address/data pin. |
| RC6/AD6 | bit6 | TTL | Input/Output or system bus address/data pin. |
| RC7/AD7 | bit7 | TTL | Input/Output or system bus address/data pin. |

Legend: TTL = TTL input.

### TABLE 9-6: REGISTERS/BITS ASSOCIATED WITH PORTC

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on Power-on Reset | Value on all other resets (Note1) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 11h, Bank 1 | PORTC | RC7/<br>AD7 | RC6/<br>AD6 | RC5/<br>AD5 | RC4/<br>AD4 | RC3/<br>AD3 | RC2/<br>AD2 | RC1/<br>AD1 | RC0/<br>AD0 | `xxxx xxxx` | `uuuu uuuu` |
| 10h, Bank 1 | DDRC | Data direction register for PORTC | | | | | | | | `1111 1111` | `1111 1111` |

Legend: x = unknown, u = unchanged.

Note 1: Other (non power-up) resets include: external reset through $\overline{\text{MCLR}}$ and the Watchdog Timer Reset.

### FIGURE 12-2: TCON2 REGISTER (ADDRESS: 17h, BANK 3)

| R - 0 | R - 0 | R/W - 0 | R/W - 0 | R/W - 0 | R/W - 0 | R/W - 0 | R/W - 0 |
|---|---|---|---|---|---|---|---|
| CA2OVF | CA1OVF | PWM2ON | PWM1ON | CA1/$\overline{\text{PR3}}$ | TMR3ON | TMR2ON | TMR1ON |
| bit7 | | | | | | | bit0 |

R = Readable bit
W = Writable bit
-n = Value at POR reset

bit 7: **CA2OVF**: Capture2 Overflow Status bit
This bit indicates that the capture value had not been read from the capture register pair (CA2H:CA2L) before the next capture event occurred. The capture register retains the oldest unread capture value (last capture before overflow). Subsequent capture events will not update the capture register with the Timer3 value until the capture register has been read (both bytes).
1 = Overflow occurred on Capture2 register
0 = No overflow occurred on Capture2 register

bit 6: **CA1OVF**: Capture1 Overflow Status bit
This bit indicates that the capture value had not been read from the capture register pair (PR3H/CA2H:PR3L/CA2L) before the next capture event occurred. The capture register retains the oldest unread capture value (last capture before overflow). Subsequent capture events will not update the capture register with the TMR3 value until the capture register has been read (both bytes).
1 = Overflow occurred on Capture1 register
0 = No overflow occurred on Capture1 register

bit 5: **PWM2ON**: PWM2 On bit
1 = PWM2 is enabled (The RB3/PWM2 pin ignores the state of the DDRB<3> bit)
0 = PWM2 is disabled (The RB3/PWM2 pin uses the state of the DDRB<3> bit for data direction)

bit 4: **PWM1ON**: PWM1 On bit
1 = PWM1 is enabled (The RB2/PWM1 pin ignores the state of the DDRB<2> bit)
0 = PWM1 is disabled (The RB2/PWM1 pin uses the state of the DDRB<2> bit for data direction)

bit 3: **CA1/$\overline{\text{PR3}}$**: CA1/$\overline{\text{PR3}}$ Register Mode Select bit
1 = Enables Capture1 (PR3H/CA1H:PR3L/CA1L is the Capture1 register. Timer3 runs without a period register)
0 = Enables the Period register (PR3H/CA1H:PR3L/CA1L is the Period register for Timer3)

bit 2: **TMR3ON**: Timer3 On bit
1 = Starts Timer3
0 = Stops Timer3

bit 1: **TMR2ON**: Timer2 On bit
This bit controls the incrementing of the Timer2 register. When Timer2:Timer1 form the 16-bit timer (T16 is set), TMR2ON must be set. This allows the MSB of the timer to increment.
1 = Starts Timer2 (Must be enabled if the T16 bit (TCON1<3>) is set)
0 = Stops Timer2

bit 0: **TMR1ON**: Timer1 On bit
When T16 is set (in 16-bit Timer Mode)
1 = Starts 16-bit Timer2:Timer1
0 = Stops 16-bit Timer2:Timer1

When T16 is clear (in 8-bit Timer Mode)
1 = Starts 8-bit Timer1
0 = Stops 8-bit Timer1

© 1996 Microchip Technology Inc.

#### 12.1.3.3.1 MAX RESOLUTION/FREQUENCY FOR EXTERNAL CLOCK INPUT

The use of an external clock for the PWM time-base (Timer1 or Timer2) limits the PWM output to a maximum resolution of 8-bits. The PWxDCL<7:6> bits must be kept cleared. Use of any other value will distort the PWM output. All resolutions are supported when internal clock mode is selected. The maximum attainable frequency is also lower. This is a result of the timing requirements of an external clock input for a timer (see the Electrical Specification section). The maximum PWM frequency, when the timers clock source is the RB4/TCLK12 pin, is shown in Table 12-3 (standard resolution mode).

## 12.2 Timer3

Timer3 is a 16-bit timer consisting of the TMR3H and TMR3L registers. TMR3H is the high byte of the timer and TMR3L is the low byte. This timer has an associated 16-bit period register (PR3H/CA1H:PR3L/CA1L). This period register can be software configured to be a second 16-bit capture register.

When the TMR3CS bit (TCON1<2>) is clear, the timer increments every instruction cycle (Fosc/4). When TMR3CS is set, the timer increments on every falling edge of the RB5/TCLK3 pin. In either mode, the TMR3ON bit must be set for the timer to increment. When TMR3ON is clear, the timer will not increment or set the TMR3IF bit.

Timer3 has two modes of operation, depending on the CA1/$\overline{\text{PR3}}$ bit (TCON2<3>). These modes are:

- One capture and one period register mode
- Dual capture register mode

The PIC17C4X has up to two 16-bit capture registers that capture the 16-bit value of TMR3 when events are detected on capture pins. There are two capture pins (RB0/CAP1 and RB1/CAP2), one for each capture register. The capture pins are multiplexed with PORTB pins. An event can be:

- a rising edge
- a falling edge
- every 4th rising edge
- every 16th rising edge

Each 16-bit capture register has an interrupt flag associated with it. The flag is set when a capture is made. The capture module is truly part of the Timer3 block. Figure 12-7 and Figure 12-8 show the block diagrams for the two modes of operation.

### TABLE 12-4: REGISTERS/BITS ASSOCIATED WITH PWM

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on Power-on Reset | Value on all other resets (Note1) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 16h, Bank 3 | TCON1 | CA2ED1 | CA2ED0 | CA1ED1 | CA1ED0 | T16 | TMR3CS | TMR2CS | TMR1CS | 0000 0000 | 0000 0000 |
| 17h, Bank 3 | TCON2 | CA2OVF | CA1OVF | PWM2ON | PWM1ON | CA1/$\overline{\text{PR3}}$ | TMR3ON | TMR2ON | TMR1ON | 0000 0000 | 0000 0000 |
| 10h, Bank 2 | TMR1 | Timer1 register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 11h, Bank 2 | TMR2 | Timer2 register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 16h, Bank 1 | PIR | RBIF | TMR3IF | TMR2IF | TMR1IF | CA2IF | CA1IF | TXIF | RCIF | 0000 0010 | 0000 0010 |
| 17h, Bank 1 | PIE | RBIE | TMR3IE | TMR2IE | TMR1IE | CA2IE | CA1IE | TXIE | RCIE | 0000 0000 | 0000 0000 |
| 07h, Unbanked | INTSTA | PEIF | T0CKIF | T0IF | INTF | PEIE | T0CKIE | T0IE | INTE | 0000 0000 | 0000 0000 |
| 06h, Unbanked | CPUSTA | — | — | STKAV | GLINTD | $\overline{\text{TO}}$ | $\overline{\text{PD}}$ | — | — | --11 11-- | --11 qq-- |
| 10h, Bank 3 | PW1DCL | DC1 | DC0 | — | — | — | — | — | — | xx-- ---- | uu-- ---- |
| 11h, Bank 3 | PW2DCL | DC1 | DC0 | TM2PW2 | — | — | — | — | — | xx0- ---- | uu0- ---- |
| 12h, Bank 3 | PW1DCH | DC9 | DC8 | DC7 | DC6 | DC5 | DC4 | DC3 | DC2 | xxxx xxxx | uuuu uuuu |
| 13h, Bank 3 | PW2DCH | DC9 | DC8 | DC7 | DC6 | DC5 | DC4 | DC3 | DC2 | xxxx xxxx | uuuu uuuu |

Legend: x = unknown, u = unchanged, - = unimplemented read as '0', q = value depends on conditions, shaded cells are not used by PWM.

**FIGURE 13-2: RCSTA REGISTER (ADDRESS: 13h, BANK 0)**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| R/W - 0 | R/W - 0 | R/W - 0 | R/W - 0 | U - 0 | R - 0 | R - 0 | R - x |
| SPEN | RX9 | SREN | CREN | — | FERR | OERR | RX9D |

bit7 bit 0

R = Readable bit
W = Writable bit
-n = Value at POR reset
(x = unknown)

bit 7: **SPEN**: Serial Port Enable bit
1 = Configures RA5/RX/DT and RA4/TX/CK pins as serial port pins
0 = Serial port disabled

bit 6: **RX9**: 9-bit Receive Enable bit
1 = Selects 9-bit reception
0 = Selects 8-bit reception

bit 5: **SREN**: Single Receive Enable bit
This bit enables the reception of a single byte. After receiving the byte, this bit is automatically cleared.
Synchronous mode:
1 = Enable reception
0 = Disable reception
Note: This bit is ignored in synchronous slave reception.
Asynchronous mode:
Don’t care

bit 4: **CREN**: Continuous Receive Enable bit
This bit enables the continuous reception of serial data.
Asynchronous mode:
1 = Enable reception
0 = Disables reception
Synchronous mode:
1 = Enables continuous reception until CREN is cleared (CREN overrides SREN)
0 = Disables continuous reception

bit 3: **Unimplemented**: Read as '0'

bit 2: **FERR**: Framing Error bit
1 = Framing error (Updated by reading RCREG)
0 = No framing error

bit 1: **OERR**: Overrun Error bit
1 = Overrun (Cleared by clearing CREN)
0 = No overrun error

bit 0: **RX9D**: 9th bit of receive data (can be the software calculated parity bit)

© 1996 Microchip Technology Inc.

### TABLE 13-9: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE TRANSMISSION

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on Power-on Reset | Value on all other resets (Note1) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 16h, Bank 1 | PIR | RBIF | TMR3IF | TMR2IF | TMR1IF | CA2IF | CA1IF | TXIF | RCIF | 0000 0010 | 0000 0010 |
| 13h, Bank 0 | RCSTA | SPEN | RX9 | SREN | CREN | — | FERR | OERR | RX9D | 0000 -00x | 0000 -00u |
| 16h, Bank 0 | TXREG | TX7 | TX6 | TX5 | TX4 | TX3 | TX2 | TX1 | TX0 | xxxx xxxx | uuuu uuuu |
| 17h, Bank 1 | PIE | RBIE | TMR3IE | TMR2IE | TMR1IE | CA2IE | CA1IE | TXIE | RCIE | 0000 0000 | 0000 0000 |
| 15h, Bank 0 | TXSTA | CSRC | TX9 | TXEN | SYNC | — | — | TRMT | TX9D | 0000 --1x | 0000 --1u |
| 17h, Bank 0 | SPBRG | Baud rate generator register | | | | | | | | xxxx xxxx | uuuu uuuu |

Legend: x = unknown, u = unchanged, - = unimplemented read as a '0', shaded cells are not used for synchronous slave transmission.

Note 1: Other (non power-up) resets include: external reset through $\overline{MCLR}$ and Watchdog Timer Reset.

### TABLE 13-10: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE RECEPTION

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on Power-on Reset | Value on all other resets (Note1) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 16h, Bank1 | PIR | RBIF | TMR3IF | TMR2IF | TMR1IF | CA2IF | CA1IF | TXIF | RCIF | 0000 0010 | 0000 0010 |
| 13h, Bank0 | RCSTA | SPEN | RX9 | SREN | CREN | — | FERR | OERR | RX9D | 0000 -00x | 0000 -00u |
| 14h, Bank0 | RCREG | RX7 | RX6 | RX5 | RX4 | RX3 | RX2 | RX1 | RX0 | xxxx xxxx | uuuu uuuu |
| 17h, Bank1 | PIE | RBIE | TMR3IE | TMR2IE | TMR1IE | CA2IE | CA1IE | TXIE | RCIE | 0000 0000 | 0000 0000 |
| 15h, Bank 0 | TXSTA | CSRC | TX9 | TXEN | SYNC | — | — | TRMT | TX9D | 0000 --1x | 0000 --1u |
| 17h, Bank0 | SPBRG | Baud rate generator register | | | | | | | | xxxx xxxx | uuuu uuuu |

Legend: x = unknown, u = unchanged, - = unimplemented read as a '0', shaded cells are not used for synchronous slave reception.

Note 1: Other (non power-up) resets include: external reset through $\overline{MCLR}$ and Watchdog Timer Reset.

© 1996 Microchip Technology Inc.

### MULLW Multiply Literal with WREG

Syntax: [ *label* ] MULLW k

Operands: 0 ≤ k ≤ 255

Operation: (k x WREG) → PRODH:PRODL

Status Affected: None

Encoding:

| | | | |
|---|---|---|---|
| 1011 | 1100 | kkkk | kkkk |

Description: An unsigned multiplication is carried out between the contents of WREG and the 8-bit literal 'k'. The 16-bit result is placed in PRODH:PRODL register pair. PRODH contains the high byte.

WREG is unchanged.

None of the status flags are affected.

Note that neither overflow nor carry is possible in this operation. A zero result is possible but not detected.

Words: 1

Cycles: 1

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|
| Decode | Read literal 'k' | Execute | Write registers PRODH: PRODL |

Example: `MULLW 0xC4`

Before Instruction

| | | |
|---|---|---|
| WREG | = | 0xE2 |
| PRODH | = | ? |
| PRODL | = | ? |

After Instruction

| | | |
|---|---|---|
| WREG | = | 0xC4 |
| PRODH | = | 0xAD |
| PRODL | = | 0x08 |

**Note:** This instruction is not available in the PIC17C42 device.

### MULWF Multiply WREG with f

Syntax: [ *label* ] MULWF f

Operands: 0 ≤ f ≤ 255

Operation: (WREG x f) → PRODH:PRODL

Status Affected: None

Encoding:

| | | | |
|---|---|---|---|
| 0011 | 0100 | ffff | ffff |

Description: An unsigned multiplication is carried out between the contents of WREG and the register file location 'f'. The 16-bit result is stored in the PRODH:PRODL register pair. PRODH contains the high byte.

Both WREG and 'f' are unchanged.

None of the status flags are affected.

Note that neither overflow nor carry is possible in this operation. A zero result is possible but not detected.

Words: 1

Cycles: 1

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|
| Decode | Read register 'f' | Execute | Write registers PRODH: PRODL |

Example: `MULWF REG`

Before Instruction

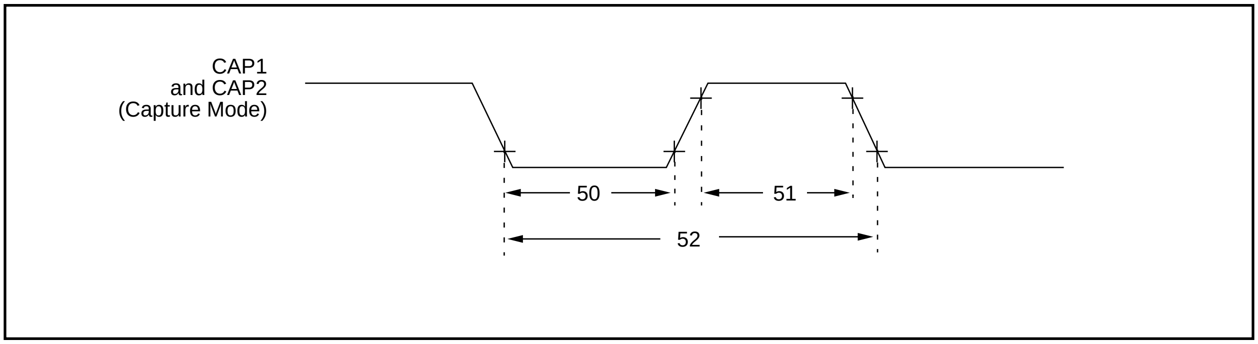
| | | |
|---|---|---|
| WREG | = | 0xC4 |
| REG | = | 0xB5 |
| PRODH | = | ? |
| PRODL | = | ? |

After Instruction

| | | |
|---|---|---|
| WREG | = | 0xC4 |
| REG | = | 0xB5 |
| PRODH | = | 0x8A |
| PRODL | = | 0x94 |

**Note:** This instruction is not available in the PIC17C42 device.

**FIGURE 17-7: CAPTURE TIMINGS**



### TABLE 17-7: CAPTURE REQUIREMENTS

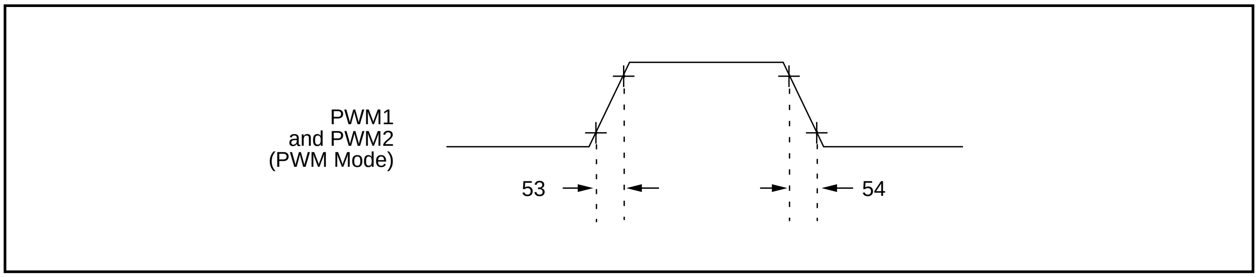
| Parameter No. | Sym | Characteristic | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| 50 | TccL | Capture1 and Capture2 input low time | 10 * | — | — | ns | |
| 51 | TccH | Capture1 and Capture2 input high time | 10 * | — | — | ns | |
| 52 | TccP | Capture1 and Capture2 input period | $\frac{2\,T_{CY}}{N}$ § | — | — | ns | N = prescale value (4 or 16) |

* 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.

**FIGURE 17-8: PWM TIMINGS**



### TABLE 17-8: PWM REQUIREMENTS

| Parameter No. | Sym | Characteristic | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| 53 | TccR | PWM1 and PWM2 output rise time | — | 10 * | 35 *§ | ns | |
| 54 | TccF | PWM1 and PWM2 output fall time | — | 10 * | 35 *§ | ns | |

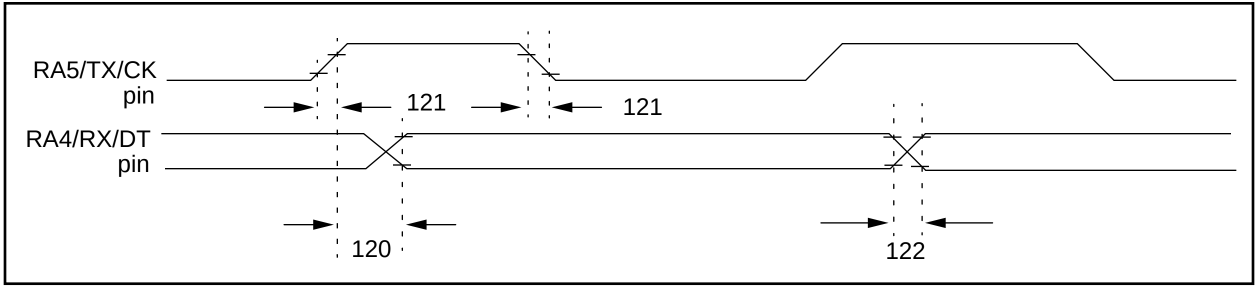
* These parameters are characterized but not tested.

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

§ This specification ensured by design.

| Applicable Devices | 42 | R42 | 42A | 43 | R43 | 44 |
|---|---|---|---|---|---|---|

### FIGURE 17-9: USART MODULE: SYNCHRONOUS TRANSMISSION (MASTER/SLAVE) TIMING

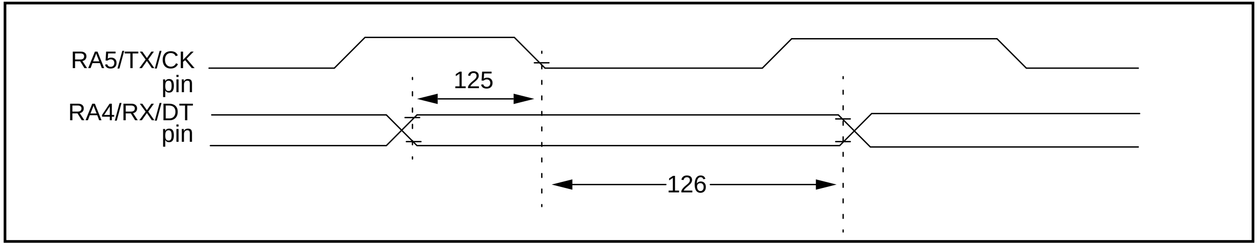


### TABLE 17-9: SERIAL PORT SYNCHRONOUS TRANSMISSION REQUIREMENTS

| Parameter No. | Sym | Characteristic | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| 120 | TckH2dtV | SYNC XMIT (MASTER & SLAVE)<br>Clock high to data out valid | — | — | 65 | ns | |
| 121 | TckRF | Clock out rise time and fall time (Master Mode) | — | 10 | 35 | ns | |
| 122 | TdtRF | Data out rise time and fall time | — | 10 | 35 | ns | |

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

### FIGURE 17-10: USART MODULE: SYNCHRONOUS RECEIVE (MASTER/SLAVE) TIMING



### TABLE 17-10: SERIAL PORT SYNCHRONOUS RECEIVE REQUIREMENTS

| Parameter No. | Sym | Characteristic | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| 125 | TdtV2ckL | SYNC RCV (MASTER & SLAVE)<br>Data hold before CK↓ (DT hold time) | 15 | — | — | ns | |
| 126 | TckL2dtl | Data hold after CK↓ (DT hold time) | 15 | — | — | ns | |

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

© 1996 Microchip Technology Inc.

| Applicable Devices | 42 | R42 | 42A | 43 | R43 | 44 |
|---|---|---|---|---|---|---|

FIGURE 18-15: IOH vs. VOH, VDD = 5V

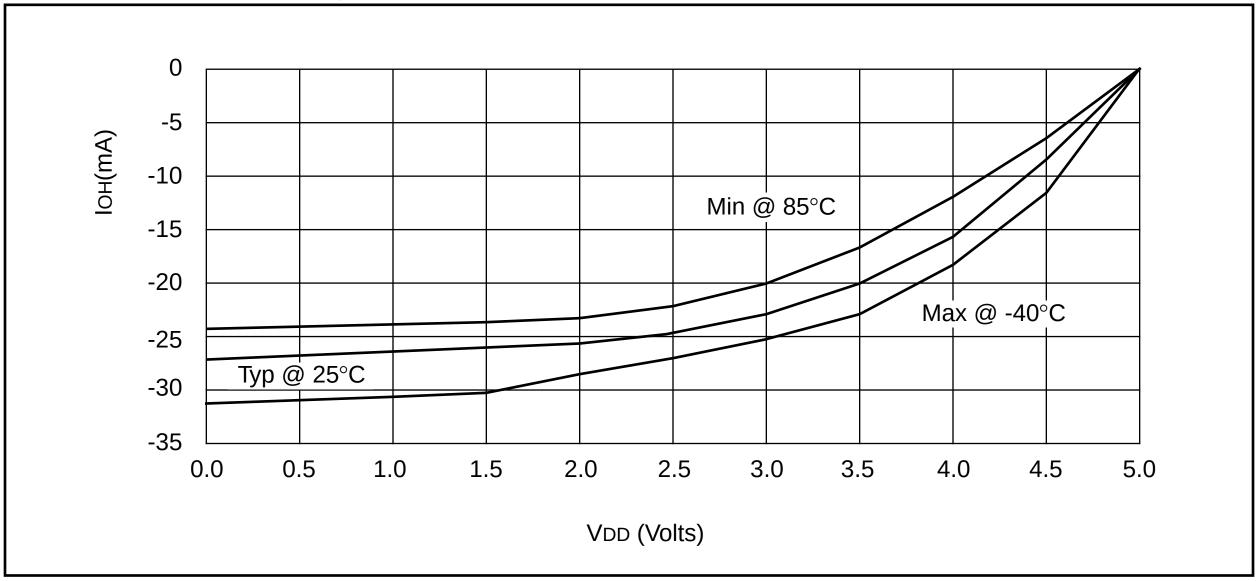
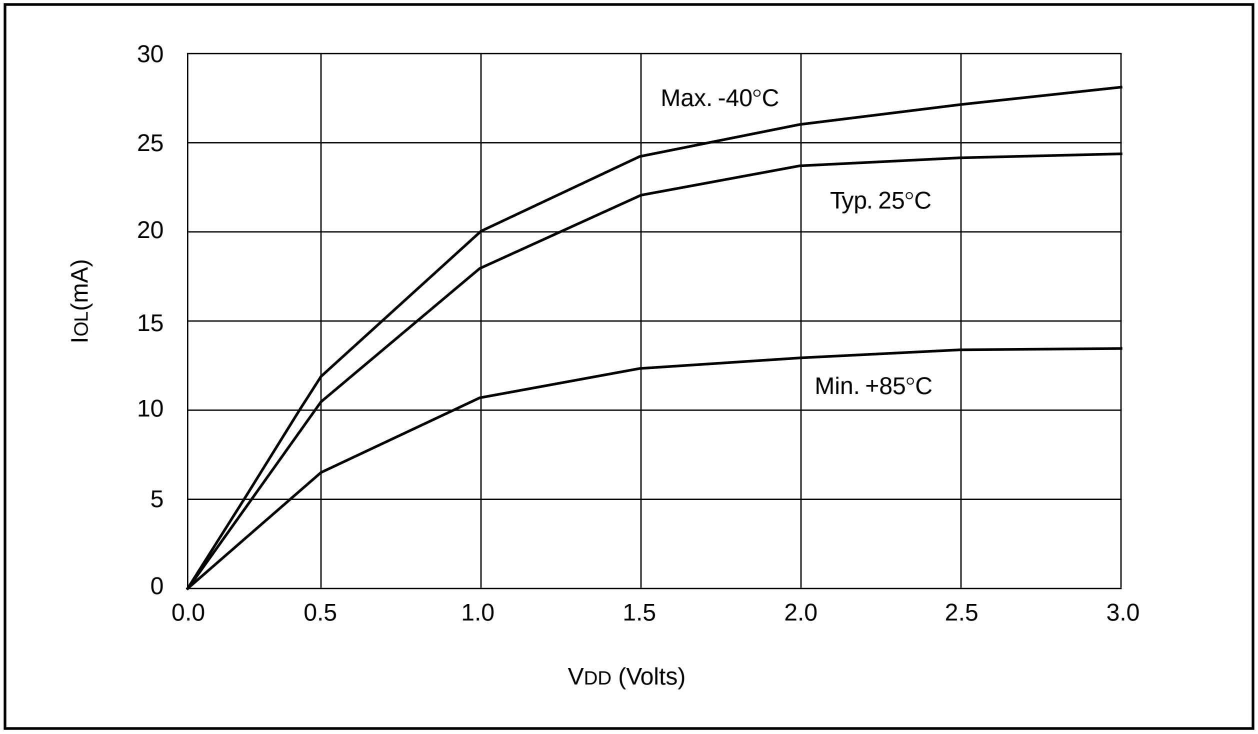


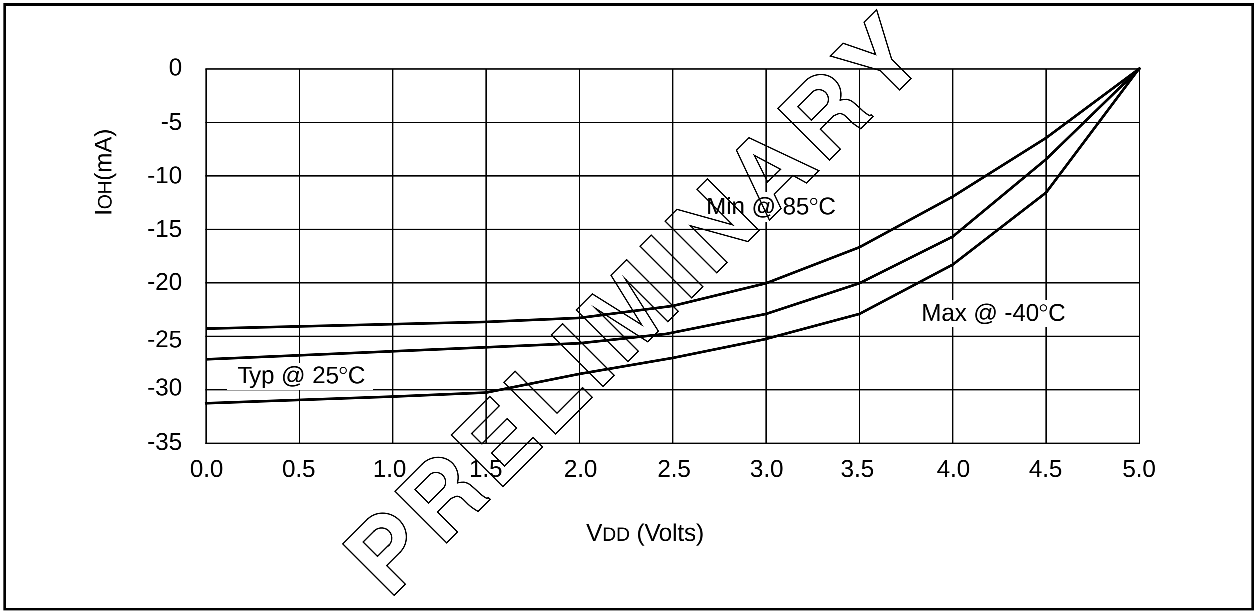
FIGURE 18-16: IOL vs. VOL, VDD = 3V



© 1996 Microchip Technology Inc.

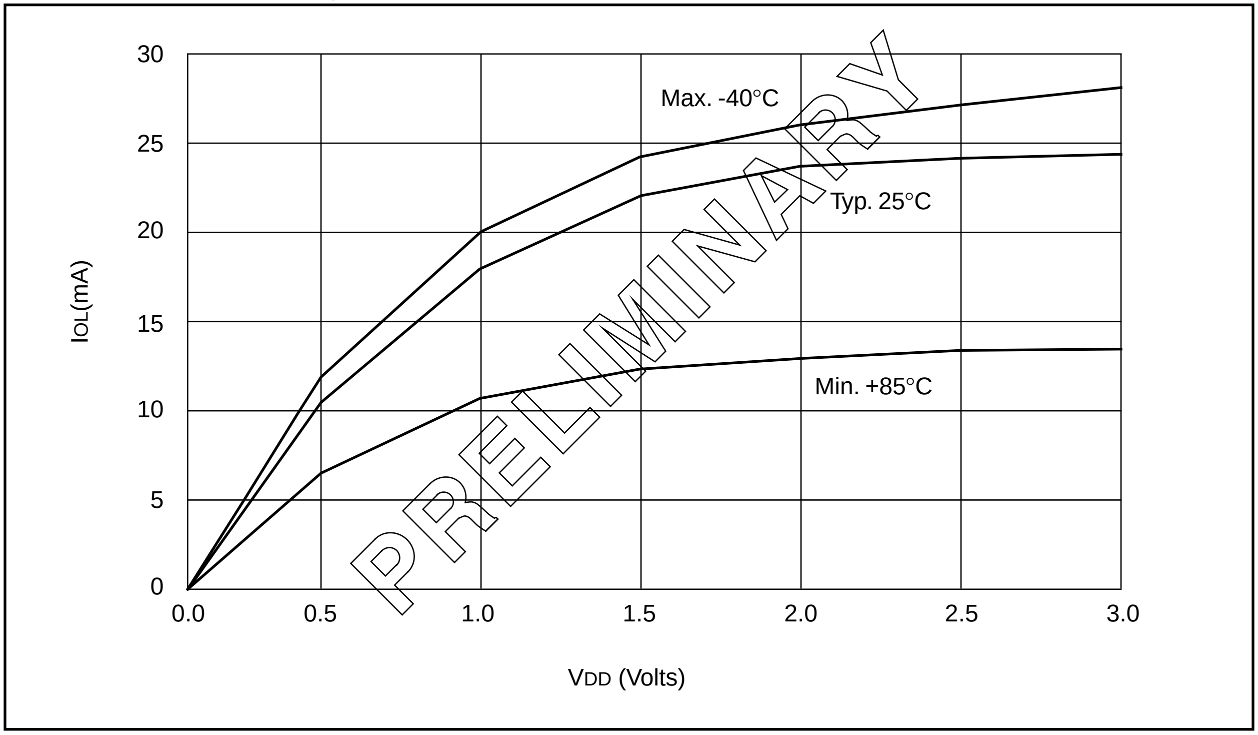
| Applicable Devices | 42 | R42 | 42A | 43 | R43 | 44 |
|---|---|---|---|---|---|---|

**FIGURE 20-15: IOH vs. VOH, VDD = 5V**



PRELIMINARY

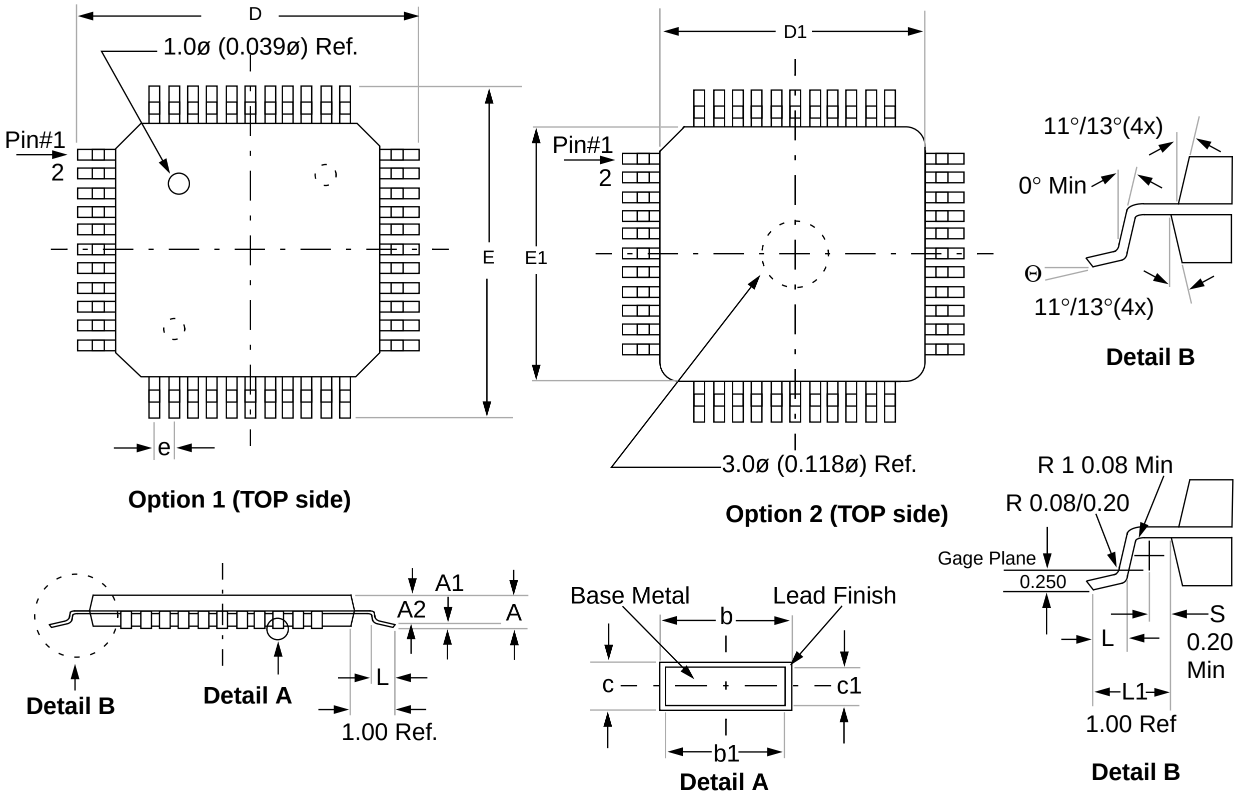
**FIGURE 20-16: IOL vs. VOL, VDD = 3V**



PRELIMINARY

© 1996 Microchip Technology Inc.

## 21.5 44-Lead Plastic Surface Mount (TQFP 10x10 mm Body 1.0/0.10 mm Lead Form)



| Package Group: Plastic TQFP | | | | | | |
|---|---|---|---|---|---|---|
| | Millimeters | | | Inches | | |
| Symbol | Min | Max | Notes | Min | Max | Notes |
| A | 1.00 | 1.20 | | 0.039 | 0.047 | |
| A1 | 0.05 | 0.15 | | 0.002 | 0.006 | |
| A2 | 0.95 | 1.05 | | 0.037 | 0.041 | |
| D | 11.75 | 12.25 | | 0.463 | 0.482 | |
| D1 | 9.90 | 10.10 | | 0.390 | 0.398 | |
| E | 11.75 | 12.25 | | 0.463 | 0.482 | |
| E1 | 9.90 | 10.10 | | 0.390 | 0.398 | |
| L | 0.45 | 0.75 | | 0.018 | 0.030 | |
| e | 0.80 BSC | | | 0.031 BSC | | |
| b | 0.30 | 0.45 | | 0.012 | 0.018 | |
| b1 | 0.30 | 0.40 | | 0.012 | 0.016 | |
| c | 0.09 | 0.20 | | 0.004 | 0.008 | |
| c1 | 0.09 | 0.16 | | 0.004 | 0.006 | |
| N | 44 | 44 | | 44 | 44 | |
| Θ | 0° | 7° | | 0° | 7° | |

Note 1: Dimensions D1 and E1 do not include mold protrusion. Allowable mold protrusion is 0.25m/m (0.010”) per side. D1 and E1 dimensions including mold mismatch.

2: Dimension “b” does not include Dambar protrusion, allowable Dambar protrusion shall be 0.08m/m (0.003”)max.

3: This outline conforms to JEDEC MS-026.

© 1996 Microchip Technology Inc.

NOTES:

© 1996 Microchip Technology Inc.