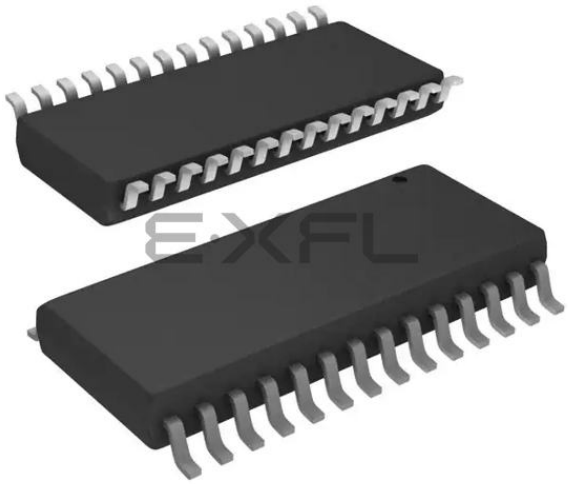


**Welcome to E-XFL.COM**

## 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 | 20MHz |
| Connectivity | I²C, SPI, UART/USART |
| Peripherals | Brown-out Detect/Reset, POR, PWM, WDT |
| Number of I/O | 25 |
| Program Memory Size | 7KB (4K x 14) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 192 x 8 |
| Voltage - Supply (Vcc/Vdd) | 1.8V ~ 3.6V |
| Data Converters | A/D 11x8b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 28-SOIC (0.295", 7.50mm Width) |
| Supplier Device Package | 28-SOIC |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/pic16lf723-i-so |

**TABLE 2-1: PIC16(L)F722/3/4/6/7 SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)**

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on: POR, BOR | Page |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Bank 2** | | | | | | | | | | | |
| 100h(2) | INDF | Addressing this location uses contents of FSR to address data memory (not a physical register) | | | | | | | | xxxx xxxx | 29,37 |
| 101h | TMR0 | Timer0 Module Register | | | | | | | | xxxx xxxx | 105,37 |
| 102h(2) | PCL | Program Counter's (PC) Least Significant Byte | | | | | | | | 0000 0000 | 28,37 |
| 103h(2) | STATUS | IRP | RP1 | RP0 | $\overline{TO}$ | $\overline{PD}$ | Z | DC | C | 0001 1xxx | 25,37 |
| 104h(2) | FSR | Indirect Data Memory Address Pointer | | | | | | | | xxxx xxxx | 29,37 |
| 105h | — | Unimplemented | | | | | | | | — | — |
| 106h | — | Unimplemented | | | | | | | | — | — |
| 107h | — | Unimplemented | | | | | | | | — | — |
| 108h | CPSCON0 | CPSON | — | — | — | CPSRNG1 | CPSRNG0 | CPSOUT | T0XCS | 0--- 0000 | 126,38 |
| 109h | CPSCON1 | — | — | — | — | CPSCH3 | CPSCH2 | CPSCH1 | CPSCH0 | ---- 0000 | 127,38 |
| 10Ah(1, 2) | PCLATH | — | — | — | Write Buffer for the upper 5 bits of the Program Counter | | | | | ---0 0000 | 28,37 |
| 10Bh(2) | INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 0000 000x | 44,37 |
| 10Ch | PMDATL | Program Memory Read Data Register Low Byte | | | | | | | | xxxx xxxx | 181,38 |
| 10Dh | PMADRL | Program Memory Read Address Register Low Byte | | | | | | | | xxxx xxxx | 181,38 |
| 10Eh | PMDATH | — | — | Program Memory Read Data Register High Byte | | | | | | --xx xxxx | 181,38 |
| 10Fh | PMADRH | — | — | — | Program Memory Read Address Register High Byte | | | | | ---x xxxx | 181,38 |
| **Bank 3** | | | | | | | | | | | |
| 180h(2) | INDF | Addressing this location uses contents of FSR to address data memory (not a physical register) | | | | | | | | xxxx xxxx | 29,37 |
| 181h | OPTION_REG | $\overline{RBPU}$ | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 | 1111 1111 | 26,37 |
| 182h(2) | PCL | Program Counter (PC) Least Significant Byte | | | | | | | | 0000 0000 | 28,37 |
| 183h(2) | STATUS | IRP | RP1 | RP0 | $\overline{TO}$ | $\overline{PD}$ | Z | DC | C | 0001 1xxx | 25,37 |
| 184h(2) | FSR | Indirect Data Memory Address Pointer | | | | | | | | xxxx xxxx | 29,37 |
| 185h | ANSELA | — | — | ANSA5 | ANSA4 | ANSA3 | ANSA2 | ANSA1 | ANSA0 | --11 1111 | 52,38 |
| 186h | ANSELB | — | — | ANSB5 | ANSB4 | ANSB3 | ANSB2 | ANSB1 | ANSB0 | --11 1111 | 61,38 |
| 187h | — | Unimplemented | | | | | | | | — | — |
| 188h | ANSELD | ANSD7 | ANSD6 | ANSD5 | ANSD4 | ANSD3 | ANSD2 | ANSD1 | ANSD0 | 1111 1111 | 78,38 |
| 189h(3) | ANSELE | — | — | — | — | — | ANSE2 | ANSE1 | ANSE0 | ---- -111 | 82,38 |
| 18Ah(1, 2) | PCLATH | — | — | — | Write Buffer for the upper 5 bits of the Program Counter | | | | | ---0 0000 | 28,37 |
| 18Bh(2) | INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 0000 000x | 44,37 |
| 18Ch | PMCON1 | Reserved | — | — | — | — | — | — | RD | 1--- ---0 | 182,38 |
| 18Dh | — | Unimplemented | | | | | | | | — | — |
| 18Eh | — | Unimplemented | | | | | | | | — | — |
| 18Fh | — | Unimplemented | | | | | | | | — | — |

**Legend:** x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, read as '0', r = reserved. Shaded locations are unimplemented, read as '0'.

**Note 1:** The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8>, whose contents are transferred to the upper byte of the program counter.

**2:** These registers can be addressed from any bank.

**3:** These registers/bits are not implemented on PIC16F722/723/726/PIC16LF722/723/726 devices, read as '0'.

**4:** Accessible only when SSPM<3:0> = 1001.

**5:** Accessible only when SSPM<3:0> ≠ 1001.

**6:** This bit is always '1' as RE3 is input-only.

## 4.1 Operation

Interrupts are disabled upon any device Reset. They are enabled by setting the following bits:

- GIE bit of the INTCON register
- Interrupt Enable bit(s) for the specific interrupt event(s)
- PEIE bit of the INTCON register (if the Interrupt Enable bit of the interrupt event is contained in the PIE1 and PIE2 registers)

The INTCON, PIR1 and PIR2 registers record individual interrupts via Interrupt Flag bits. Interrupt Flag bits will be set, regardless of the status of the GIE, PEIE and individual Interrupt Enable bits.

The following events happen when an interrupt event occurs while the GIE bit is set:

- Current prefetched instruction is flushed
- GIE bit is cleared
- Current Program Counter (PC) is pushed onto the stack
- PC is loaded with the interrupt vector 0004h

The ISR determines the source of the interrupt by polling the Interrupt Flag bits. The Interrupt Flag bits must be cleared before exiting the ISR to avoid repeated interrupts. Because the GIE bit is cleared, any interrupt that occurs while executing the ISR will be recorded through its Interrupt Flag, but will not cause the processor to redirect to the interrupt vector.

The `RETFIE` instruction exits the ISR by popping the previous address from the stack and setting the GIE bit.

For additional information on a specific interrupt's operation, refer to its peripheral chapter.

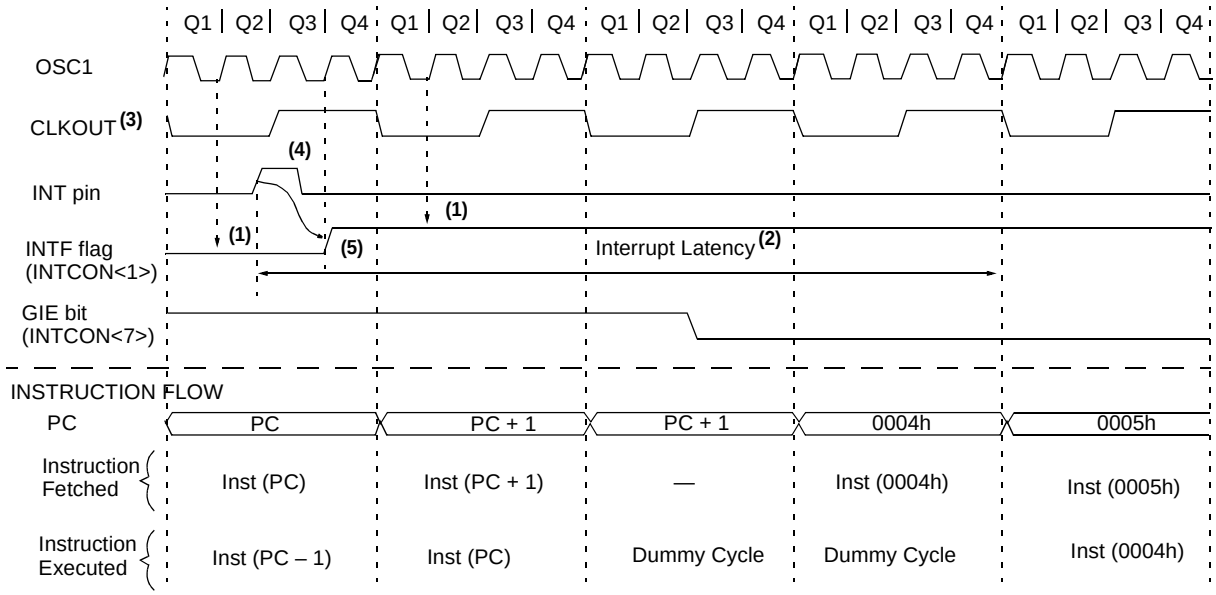
**Note 1:** Individual Interrupt Flag bits are set, regardless of the state of any other enable bits.

**2:** All interrupts will be ignored while the GIE bit is cleared. Any interrupt occurring while the GIE bit is clear will be serviced when the GIE bit is set again.

## 4.2 Interrupt Latency

Interrupt latency is defined as the time from when the interrupt event occurs to the time code execution at the interrupt vector begins. The latency for synchronous interrupts is three instruction cycles. For asynchronous interrupts, the latency is three to four instruction cycles, depending on when the interrupt occurs. See Figure 4-2 for timing details.

**FIGURE 4-2: INT PIN INTERRUPT TIMING**




**Note 1:** INTF flag is sampled here (every Q1).

**2:** Asynchronous interrupt latency = 3-4 TCY. Synchronous latency = 3 TCY, where TCY = instruction cycle time. Latency is the same whether Inst (PC) is a single cycle or a 2-cycle instruction.

**3:** CLKOUT is available only in INTOSC and RC Oscillator modes.

**4:** For minimum width of INT pulse, refer to AC specifications in **Section 23.0 "Electrical Specifications"**.

**5:** INTF is enabled to be set any time during the Q4-Q1 cycles.

## 4.3 Interrupts During Sleep

Some interrupts can be used to wake from Sleep. To wake from Sleep, the peripheral must be able to operate without the system clock. The interrupt source must have the appropriate Interrupt Enable bit(s) set prior to entering Sleep.

On waking from Sleep, if the GIE bit is also set, the processor will branch to the interrupt vector. Otherwise, the processor will continue executing instructions after the SLEEP instruction. The instruction directly after the SLEEP instruction will always be executed before branching to the ISR. Refer to the **Section 19.0 "Power-Down Mode (Sleep)"** for more details.

## 4.4 INT Pin

The external interrupt, INT pin, causes an asynchronous, edge-triggered interrupt. The INTEDG bit of the OPTION register determines on which edge the interrupt will occur. When the INTEDG bit is set, the rising edge will cause the interrupt. When the INTEDG bit is clear, the falling edge will cause the interrupt. The INTF bit of the INTCON register will be set when a valid edge appears on the INT pin. If the GIE and INTE bits are also set, the processor will redirect program execution to the interrupt vector. This interrupt is disabled by clearing the INTE bit of the INTCON register.

## 4.5 Context Saving

When an interrupt occurs, only the return PC value is saved to the stack. If the ISR modifies or uses an instruction that modifies key registers, their values must be saved at the beginning of the ISR and restored when the ISR completes. This prevents instructions following the ISR from using invalid data. Examples of key registers include the W, STATUS, FSR and PCLATH registers.

| | |
|---|---|
| **Note:** | The microcontroller does not normally require saving the PCLATH register. However, if computed GOTO's are used, the PCLATH register must be saved at the beginning of the ISR and restored when the ISR is complete to ensure correct program flow. |

The code shown in Example 4-1 can be used to do the following.

- Save the W register
- Save the STATUS register
- Save the PCLATH register
- Execute the ISR program
- Restore the PCLATH register
- Restore the STATUS register
- Restore the W register

Since most instructions modify the W register, it must be saved immediately upon entering the ISR. The SWAPF instruction is used when saving and restoring the W and STATUS registers because it will not affect any bits in the STATUS register. It is useful to place W_TEMP in shared memory because the ISR cannot predict which bank will be selected when the interrupt occurs.

The processor will branch to the interrupt vector by loading the PC with 0004h. The PCLATH register will remain unchanged. This requires the ISR to ensure that the PCLATH register is set properly before using an instruction that causes PCLATH to be loaded into the PC. See **Section 2.3 "PCL and PCLATH"** for details on PC operation.

**FIGURE 6-4: BLOCK DIAGRAM OF RA5**

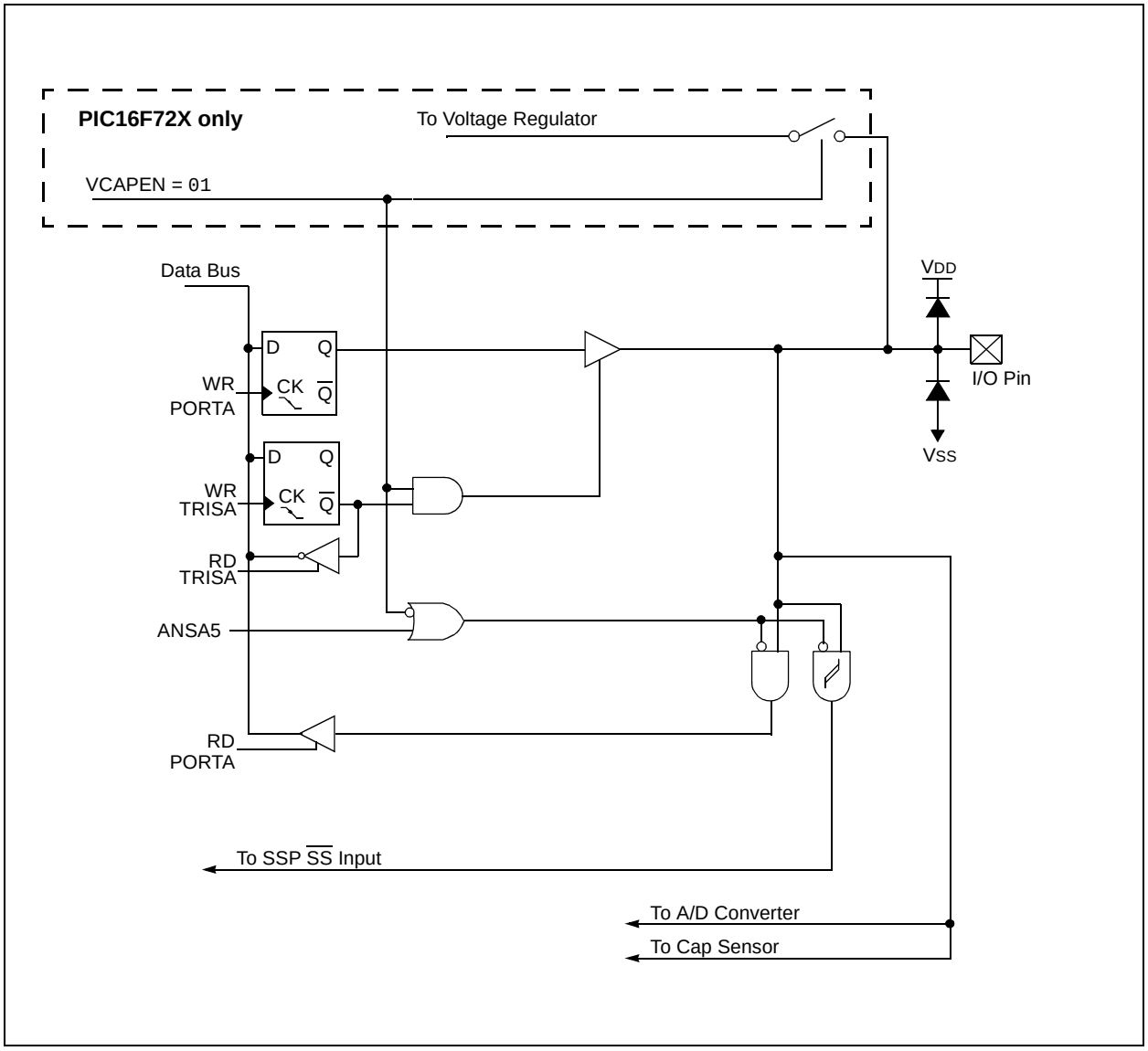
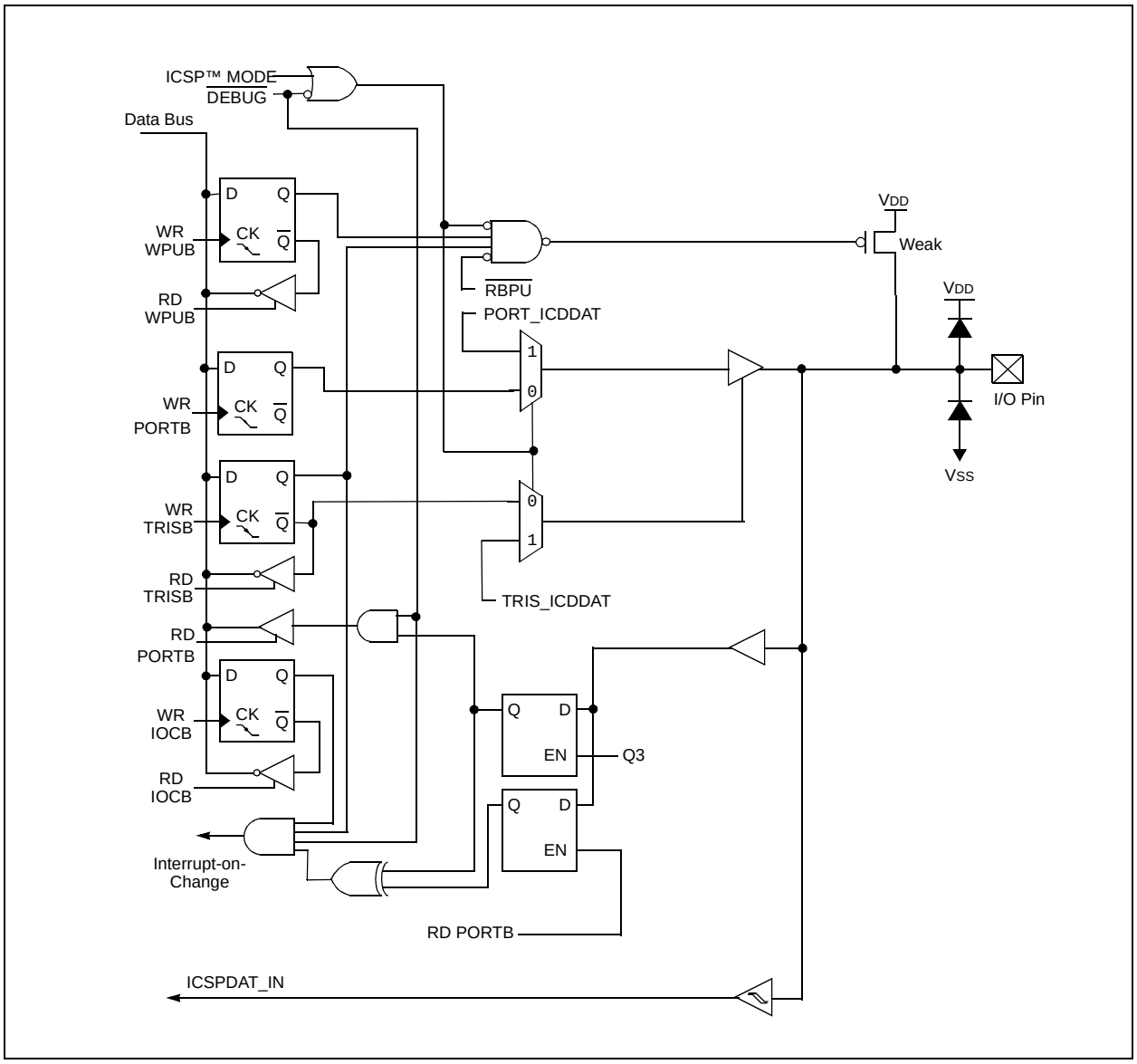


**TABLE 6-1: SUMMARY OF REGISTERS ASSOCIATED WITH PORTA**

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|
| ADCON0 | — | — | CHS3 | CHS2 | CHS1 | CHS0 | GO/$\overline{\text{DONE}}$ | ADON | 0000 0000 | 0000 0000 |
| ADCON1 | — | ADCS2 | ADCS1 | ADCS0 | — | — | ADREF1 | ADREF0 | -000 --00 | -000 --00 |
| ANSELA | — | — | ANSA5 | ANSA4 | ANSA3 | ANSA2 | ANSA1 | ANSA0 | --11 1111 | --11 1111 |
| APFCON | — | — | — | — | — | — | SSSEL | CCP2SEL | ---- --00 | ---- --00 |
| CPSCON0 | CPSON | — | — | — | CPSRNG1 | CPSRNG0 | CPSOUT | T0XCS | 0--- 0000 | 0--- 0000 |
| CPSCON1 | — | — | — | — | CPSCH3 | CPSCH2 | CPSCH1 | CPSCH0 | ---- 0000 | ---- 0000 |
| CONFIG2(1) | — | — | VCAPEN1 | VCAPEN0 | — | — | — | — | — | — |
| OPTION_REG | $\overline{\text{RBPU}}$ | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 | 1111 1111 | 1111 1111 |
| PORTA | RA7 | RA6 | RA5 | RA4 | RA3 | RA2 | RA1 | RA0 | xxxx xxxx | xxxx xxxx |
| SSPCON | WCOL | SSPOV | SSPEN | CKP | SSPM3 | SSPM2 | SSPM1 | SSPM0 | 0000 0000 | 0000 0000 |
| TRISA | TRISA7 | TRISA6 | TRISA5 | TRISA4 | TRISA3 | TRISA2 | TRISA1 | TRISA0 | 1111 1111 | 1111 1111 |

**Legend:** x = unknown, u = unchanged, – = unimplemented locations read as '0'. Shaded cells are not used by PORTA.
**Note 1:** PIC16F72X only.

**FIGURE 6-12: BLOCK DIAGRAM OF RB7**



#### 9.1.5 INTERRUPTS

The ADC module allows for the ability to generate an interrupt upon completion of an Analog-to-Digital conversion. The ADC interrupt flag is the ADIF bit in the PIR1 register. The ADC interrupt enable is the ADIE bit in the PIE1 register. The ADIF bit must be cleared in software.

| | |
|---|---|
| **Note 1:** | The ADIF bit is set at the completion of every conversion, regardless of whether or not the ADC interrupt is enabled. |
| **2:** | The ADC operates during Sleep only when the FRC oscillator is selected. |

This interrupt can be generated while the device is operating or while in Sleep. If the device is in Sleep, the interrupt will wake-up the device. Upon waking from Sleep, the next instruction following the SLEEP instruction is always executed. If the user is attempting to wake-up from Sleep and resume in-line code execution, the GIE and PEIE bits of the INTCON register must be disabled. If the GIE and PEIE bits of the INTCON register are enabled, execution will switch to the Interrupt Service Routine.

Please refer to **Section 9.1.5 "Interrupts"** for more information.

### 9.2 ADC Operation

#### 9.2.1 STARTING A CONVERSION

To enable the ADC module, the ADON bit of the ADCON0 register must be set to a '1'. Setting the GO/$\overline{\text{DONE}}$ bit of the ADCON0 register to a '1' will start the Analog-to-Digital conversion.

| | |
|---|---|
| **Note:** | The GO/$\overline{\text{DONE}}$ bit should not be set in the same instruction that turns on the ADC. Refer to **Section 9.2.6 "A/D Conversion Procedure"**. |

#### 9.2.2 COMPLETION OF A CONVERSION

When the conversion is complete, the ADC module will:

- Clear the GO/$\overline{\text{DONE}}$ bit
- Set the ADIF Interrupt Flag bit
- Update the ADRES register with new conversion result

#### 9.2.3 TERMINATING A CONVERSION

If a conversion must be terminated before completion, the GO/$\overline{\text{DONE}}$ bit can be cleared in software. The ADRES register will be updated with the partially complete Analog-to-Digital conversion sample. Incomplete bits will match the last bit converted.

| | |
|---|---|
| **Note:** | A device Reset forces all registers to their Reset state. Thus, the ADC module is turned off and any pending conversion is terminated. |

#### 9.2.4 ADC OPERATION DURING SLEEP

The ADC module can operate during Sleep. This requires the ADC clock source to be set to the FRC option. When the FRC clock source is selected, the ADC waits one additional instruction before starting the conversion. This allows the SLEEP instruction to be executed, which can reduce system noise during the conversion. If the ADC interrupt is enabled, the device will wake-up from Sleep when the conversion completes. If the ADC interrupt is disabled, the ADC module is turned off after the conversion completes, although the ADON bit remains set.

When the ADC clock source is something other than FRC, a SLEEP instruction causes the present conversion to be aborted and the ADC module is turned off, although the ADON bit remains set.

#### 9.2.5 SPECIAL EVENT TRIGGER

The Special Event Trigger of the CCP module allows periodic ADC measurements without software intervention. When this trigger occurs, the GO/$\overline{\text{DONE}}$ bit is set by hardware and the Timer1 counter resets to zero.

Using the Special Event Trigger does not assure proper ADC timing. It is the user's responsibility to ensure that the ADC timing requirements are met.

Refer to **Section 15.0 "Capture/Compare/PWM (CCP) Module"** for more information.

## 9.3 A/D Acquisition Requirements

For the ADC to meet its specified accuracy, the charge holding capacitor (CHOLD) must be allowed to fully charge to the input channel voltage level. The Analog Input model is shown in Figure 9-3. The source impedance (RS) and the internal sampling switch (RSS) impedance directly affect the time required to charge the capacitor CHOLD. The sampling switch (RSS) impedance varies over the device voltage (VDD), refer to Figure 9-3. **The maximum recommended impedance for analog sources is 10 kΩ**. As the source impedance is decreased, the acquisition time may be decreased. After the analog input channel is selected (or changed), an A/D acquisition must be done before the conversion can be started. To calculate the minimum acquisition time, Equation 9-1 may be used. This equation assumes that 1/2 LSb error is used (256 steps for the ADC). The 1/2 LSb error is the maximum error allowed for the ADC to meet its specified resolution.

**EQUATION 9-1: ACQUISITION TIME EXAMPLE**

*Assumptions:* $Temperature = 50°C$ *and external impedance of* $10k\Omega$ $5.0V$ $V_{DD}$

$$\begin{aligned} T_{ACQ} &= \text{Amplifier Settling Time} + \text{Hold Capacitor Charging Time} + \text{Temperature Coefficient} \\ &= T_{AMP} + T_C + T_{COFF} \\ &= 2\mu s + T_C + [(\text{Temperature} - 25°C)(0.05\mu s/°C)] \end{aligned}$$

*The value for* $T_C$ *can be approximated with the following equations:*

$$V_{APPLIED}\left(1 - \frac{1}{(2^{n+1}) - 1}\right) = V_{CHOLD} \quad ;[1]\ V_{CHOLD} \text{ charged to within 1/2 lsb}$$

$$V_{APPLIED}\left(1 - e^{\frac{-T_C}{RC}}\right) = V_{CHOLD} \quad ;[2]\ V_{CHOLD} \text{ charge response to } V_{APPLIED}$$

$$V_{APPLIED}\left(1 - e^{\frac{-T_C}{RC}}\right) = V_{APPLIED}\left(1 - \frac{1}{(2^{n+1}) - 1}\right) \quad ;\text{combining [1] and [2]}$$

***Note:*** *Where n = number of bits of the ADC.*

*Solving for* $T_C$*:*

$$\begin{aligned} T_C &= -C_{HOLD}(R_{IC} + R_{SS} + R_S)\ ln(1/511) \\ &= -10pF(1k\Omega + 7k\Omega + 10k\Omega)\ ln(0.001957) \\ &= 1.12\mu s \end{aligned}$$

*Therefore:*

$$\begin{aligned} T_{ACQ} &= 2MS + 1.12MS + [(50°C - 25°C)(0.05MS/°C)] \\ &= 4.42MS \end{aligned}$$

**Note 1:** The reference voltage (VREF) has no effect on the equation, since it cancels itself out.

**2:** The charge holding capacitor (CHOLD) is not discharged after each conversion.

**3:** The maximum recommended impedance for analog sources is 10 kΩ. This is required to meet the pin leakage specification.

### 15.3.4 PWM RESOLUTION

The resolution determines the number of available duty cycles for a given period. For example, a 10-bit resolution will result in 1024 discrete duty cycles, whereas an 8-bit resolution will result in 256 discrete duty cycles.

The maximum PWM resolution is ten bits when PR2 is 255. The resolution is a function of the PR2 register value as shown by Equation 15-4.

**EQUATION 15-4: PWM RESOLUTION**

$$Resolution = \frac{log[4(PR2 + 1)]}{log(2)} \ bits$$

**Note:** If the pulse-width value is greater than the period, the assigned PWM pin(s) will remain unchanged.

**TABLE 15-5: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS (FOSC = 20 MHz)**

| PWM Frequency | 1.22 kHz | 4.88 kHz | 19.53 kHz | 78.12 kHz | 156.3 kHz | 208.3 kHz |
|---|---|---|---|---|---|---|
| Timer Prescale (1, 4, 16) | 16 | 4 | 1 | 1 | 1 | 1 |
| PR2 Value | 0xFF | 0xFF | 0xFF | 0x3F | 0x1F | 0x17 |
| Maximum Resolution (bits) | 10 | 10 | 10 | 8 | 7 | 6.6 |

**TABLE 15-6: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS (FOSC = 8 MHz)**

| PWM Frequency | 1.22 kHz | 4.90 kHz | 19.61 kHz | 76.92 kHz | 153.85 kHz | 200.0 kHz |
|---|---|---|---|---|---|---|
| Timer Prescale (1, 4, 16) | 16 | 4 | 1 | 1 | 1 | 1 |
| PR2 Value | 0x65 | 0x65 | 0x65 | 0x19 | 0x0C | 0x09 |
| Maximum Resolution (bits) | 8 | 8 | 8 | 6 | 5 | 5 |

### 15.3.5 OPERATION IN SLEEP MODE

In Sleep mode, the TMR2 register will not increment and the state of the module will not change. If the CCPx pin is driving a value, it will continue to drive that value. When the device wakes up, TMR2 will continue from its previous state.

### 15.3.6 CHANGES IN SYSTEM CLOCK FREQUENCY

The PWM frequency is derived from the system clock frequency (FOSC). Any changes in the system clock frequency will result in changes to the PWM frequency. Refer to **Section 7.0 "Oscillator Module"** for additional details.

### 15.3.7 EFFECTS OF RESET

Any Reset will force all ports to Input mode and the CCP registers to their Reset states.

### 15.3.8 SETUP FOR PWM OPERATION

The following steps should be taken when configuring the CCP module for PWM operation:

1. Disable the PWM pin (CCPx) output driver(s) by setting the associated TRIS bit(s).
2. Load the PR2 register with the PWM period value.
3. Configure the CCP module for the PWM mode by loading the CCPxCON register with the appropriate values.
4. Load the CCPRxL register and the DCxBx bits of the CCPxCON register, with the PWM duty cycle value.
5. Configure and start Timer2:
   - Clear the TMR2IF interrupt flag bit of the PIR1 register. See Note below.
   - Configure the T2CKPS bits of the T2CON register with the Timer2 prescale value.
   - Enable Timer2 by setting the TMR2ON bit of the T2CON register.
6. Enable PWM output pin:
   - Wait until Timer2 overflows, TMR2IF bit of the PIR1 register is set. See Note below.
   - Enable the PWM pin (CCPx) output driver(s) by clearing the associated TRIS bit(s).

**Note:** In order to send a complete duty cycle and period on the first PWM output, the above steps must be included in the setup sequence. If it is not critical to start with a complete PWM signal on the first output, then step 6 may be ignored.

**REGISTER 16-2: RCSTA: RECEIVE STATUS AND CONTROL REGISTER**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R-0 | R-0 | R-x |
|---|---|---|---|---|---|---|---|
| SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| -n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

bit 7 **SPEN:** Serial Port Enable bit(1)
1 = Serial port enabled (configures RX/DT and TX/CK pins as serial port pins)
0 = Serial port disabled (held in Reset)

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
Asynchronous mode:
Don't care
Synchronous mode – Master:
1 = Enables single receive
0 = Disables single receive
This bit is cleared after reception is complete.
Synchronous mode – Slave:
Don't care

bit 4 **CREN:** Continuous Receive Enable bit
Asynchronous mode:
1 = Enables receiver
0 = Disables receiver
Synchronous mode:
1 = Enables continuous receive until enable bit CREN is cleared (CREN overrides SREN)
0 = Disables continuous receive

bit 3 **ADDEN:** Address Detect Enable bit
Asynchronous mode 9-bit (RX9 = 1):
1 = Enables address detection, enable interrupt and load the receive buffer when RSR<8> is set
0 = Disables address detection, all bytes are received and ninth bit can be used as parity bit
Asynchronous mode 8-bit (RX9 = 0):
Don't care
Synchronous mode:
Must be set to '0'

bit 2 **FERR:** Framing Error bit
1 = Framing error (can be updated by reading RCREG register and receive next valid byte)
0 = No framing error

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

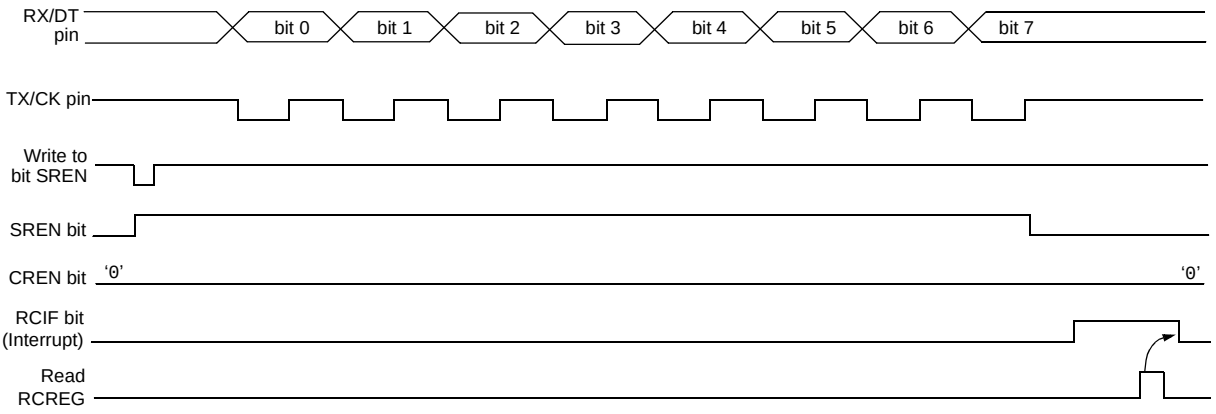
bit 0 **RX9D:** Ninth bit of Received Data
This can be address/data bit or a parity bit and must be calculated by user firmware.

**Note 1:** The AUSART module automatically changes the pin from tri-state to drive as needed. Configure TRISx = 1.

**TABLE 16-5: BAUD RATES FOR ASYNCHRONOUS MODES**

| BAUD RATE | SYNC = 0, BRGH = 1 | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | FOSC = 8.000 MHz | | | FOSC = 4.000 MHz | | | FOSC = 3.6864 MHz | | | FOSC = 1.000 MHz | | |
| | Actual Rate | % Error | SPBRG value (decimal) | Actual Rate | % Error | SPBRG value (decimal) | Actual Rate | % Error | SPBRG value (decimal) | Actual Rate | % Error | SPBRG value (decimal) |
| 300 | — | — | — | — | — | — | — | — | — | 300 | 0.16 | 207 |
| 1200 | — | — | — | 1202 | 0.16 | 207 | 1200 | 0.00 | 191 | 1202 | 0.16 | 51 |
| 2400 | 2404 | 0.16 | 207 | 2404 | 0.16 | 103 | 2400 | 0.00 | 95 | 2404 | 0.16 | 25 |
| 9600 | 9615 | 0.16 | 51 | 9615 | 0.16 | 25 | 9600 | 0.00 | 23 | — | — | — |
| 10417 | 10417 | 0.00 | 47 | 10417 | 0.00 | 23 | 10473 | 0.53 | 21 | 10417 | 0.00 | 5 |
| 19.2k | 19231 | 0.16 | 25 | 19.23k | 0.16 | 12 | 19.2k | 0.00 | 11 | — | — | — |
| 57.6k | 55556 | -3.55 | 8 | — | — | — | 57.60k | 0.00 | 3 | — | — | — |
| 115.2k | — | — | — | — | — | — | 115.2k | 0.00 | 1 | — | — | — |

**FIGURE 16-8: SYNCHRONOUS RECEPTION (MASTER MODE, SREN)**




**Note:** Timing diagram demonstrates Synchronous Master mode with bit SREN = 1 and bit BRGH = 0.

**TABLE 16-7: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER RECEPTION**

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|
| INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 0000 000x | 0000 000x |
| PIE1 | TMR1GIE | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | 0000 0000 | 0000 0000 |
| PIR1 | TMR1GIF | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | 0000 0000 | 0000 0000 |
| RCREG | AUSART Receive Data Register | | | | | | | | 0000 0000 | 0000 0000 |
| RCSTA | SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D | 0000 000X | 0000 000X |
| TRISC | TRISC7 | TRISC6 | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 | 1111 1111 | 1111 1111 |
| TXSTA | CSRC | TX9 | TXEN | SYNC | — | BRGH | TRMT | TX9D | 0000 -010 | 0000 -010 |

**Legend:** x = unknown, - = unimplemented read as '0'. Shaded cells are not used for Synchronous Master Reception.

**TABLE 21-2: PIC16(L)F722/3/4/6/7 INSTRUCTION SET**

| Mnemonic, Operands | | Description | Cycles | 14-Bit Opcode MSb | | | LSb | Status Affected | Notes |
|---|---|---|---|---|---|---|---|---|---|
| **BYTE-ORIENTED FILE REGISTER OPERATIONS** | | | | | | | | | |
| ADDWF | f, d | Add W and f | 1 | 00 | 0111 | dfff | ffff | C, DC, Z | **1, 2** |
| ANDWF | f, d | AND W with f | 1 | 00 | 0101 | dfff | ffff | Z | **1, 2** |
| CLRF | f | Clear f | 1 | 00 | 0001 | lfff | ffff | Z | **2** |
| CLRW | – | Clear W | 1 | 00 | 0001 | 0xxx | xxxx | Z | |
| COMF | f, d | Complement f | 1 | 00 | 1001 | dfff | ffff | Z | **1, 2** |
| DECF | f, d | Decrement f | 1 | 00 | 0011 | dfff | ffff | Z | **1, 2** |
| DECFSZ | f, d | Decrement f, Skip if 0 | 1**(2)** | 00 | 1011 | dfff | ffff | | **1, 2, 3** |
| INCF | f, d | Increment f | 1 | 00 | 1010 | dfff | ffff | Z | **1, 2** |
| INCFSZ | f, d | Increment f, Skip if 0 | 1**(2)** | 00 | 1111 | dfff | ffff | | **1, 2, 3** |
| IORWF | f, d | Inclusive OR W with f | 1 | 00 | 0100 | dfff | ffff | Z | **1, 2** |
| MOVF | f, d | Move f | 1 | 00 | 1000 | dfff | ffff | Z | **1, 2** |
| MOVWF | f | Move W to f | 1 | 00 | 0000 | lfff | ffff | | |
| NOP | – | No Operation | 1 | 00 | 0000 | 0xx0 | 0000 | | |
| RLF | f, d | Rotate Left f through Carry | 1 | 00 | 1101 | dfff | ffff | C | **1, 2** |
| RRF | f, d | Rotate Right f through Carry | 1 | 00 | 1100 | dfff | ffff | C | **1, 2** |
| SUBWF | f, d | Subtract W from f | 1 | 00 | 0010 | dfff | ffff | C, DC, Z | **1, 2** |
| SWAPF | f, d | Swap nibbles in f | 1 | 00 | 1110 | dfff | ffff | | **1, 2** |
| XORWF | f, d | Exclusive OR W with f | 1 | 00 | 0110 | dfff | ffff | Z | **1, 2** |
| **BIT-ORIENTED FILE REGISTER OPERATIONS** | | | | | | | | | |
| BCF | f, b | Bit Clear f | 1 | 01 | 00bb | bfff | ffff | | **1, 2** |
| BSF | f, b | Bit Set f | 1 | 01 | 01bb | bfff | ffff | | **1, 2** |
| BTFSC | f, b | Bit Test f, Skip if Clear | 1 **(2)** | 01 | 10bb | bfff | ffff | | **3** |
| BTFSS | f, b | Bit Test f, Skip if Set | 1 **(2)** | 01 | 11bb | bfff | ffff | | **3** |
| **LITERAL AND CONTROL OPERATIONS** | | | | | | | | | |
| ADDLW | k | Add literal and W | 1 | 11 | 111x | kkkk | kkkk | C, DC, Z | |
| ANDLW | k | AND literal with W | 1 | 11 | 1001 | kkkk | kkkk | Z | |
| CALL | k | Call Subroutine | 2 | 10 | 0kkk | kkkk | kkkk | | |
| CLRWDT | – | Clear Watchdog Timer | 1 | 00 | 0000 | 0110 | 0100 | $\overline{TO}$, $\overline{PD}$ | |
| GOTO | k | Go to address | 2 | 10 | 1kkk | kkkk | kkkk | | |
| IORLW | k | Inclusive OR literal with W | 1 | 11 | 1000 | kkkk | kkkk | Z | |
| MOVLW | k | Move literal to W | 1 | 11 | 00xx | kkkk | kkkk | | |
| RETFIE | – | Return from interrupt | 2 | 00 | 0000 | 0000 | 1001 | | |
| RETLW | k | Return with literal in W | 2 | 11 | 01xx | kkkk | kkkk | | |
| RETURN | – | Return from Subroutine | 2 | 00 | 0000 | 0000 | 1000 | | |
| SLEEP | – | Go into Standby mode | 1 | 00 | 0000 | 0110 | 0011 | $\overline{TO}$, $\overline{PD}$ | |
| SUBLW | k | Subtract W from literal | 1 | 11 | 110x | kkkk | kkkk | C, DC, Z | |
| XORLW | k | Exclusive OR literal with W | 1 | 11 | 1010 | kkkk | kkkk | Z | |

**Note 1:** When an I/O register is modified as a function of itself (e.g., `MOVF PORTA, 1`), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.

**2:** If this instruction is executed on the TMR0 register (and where applicable, d = 1), the prescaler will be cleared if assigned to the Timer0 module.

**3:** If the Program Counter (PC) is modified, or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a `NOP`.

## 22.11 Demonstration/Development Boards, Evaluation Kits, and Starter Kits

A wide variety of demonstration, development and evaluation boards for various PIC MCUs and dsPIC DSCs allows quick application development on fully functional systems. Most boards include prototyping areas for adding custom circuitry and provide application firmware and source code for examination and modification.

The boards support a variety of features, including LEDs, temperature sensors, switches, speakers, RS-232 interfaces, LCD displays, potentiometers and additional EEPROM memory.

The demonstration and development boards can be used in teaching environments, for prototyping custom circuits and for learning about various microcontroller applications.

In addition to the PICDEM™ and dsPICDEM™ demonstration/development board series of circuits, Microchip has a line of evaluation kits and demonstration software for analog filter design, KEELOQ® security ICs, CAN, IrDA®, PowerSmart battery management, SEEVAL® evaluation system, Sigma-Delta ADC, flow rate sensing, plus many more.

Also available are starter kits that contain everything needed to experience the specified device. This usually includes a single application and debug capability, all on one board.

Check the Microchip web page (www.microchip.com) for the complete list of demonstration, development and evaluation kits.

## 22.12 Third-Party Development Tools

Microchip also offers a great collection of tools from third-party vendors. These tools are carefully selected to offer good value and unique functionality.

- Device Programmers and Gang Programmers from companies, such as SoftLog and CCS
- Software Tools from companies, such as Gimpel and Trace Systems
- Protocol Analyzers from companies, such as Saleae and Total Phase
- Demonstration Boards from companies, such as MikroElektronika, Digilent® and Olimex
- Embedded Ethernet Solutions from companies, such as EZ Web Lynx, WIZnet and IPLogika®

## 23.7 AC Characteristics: PIC16F72X-I/E

**FIGURE 23-3: CLOCK TIMING**

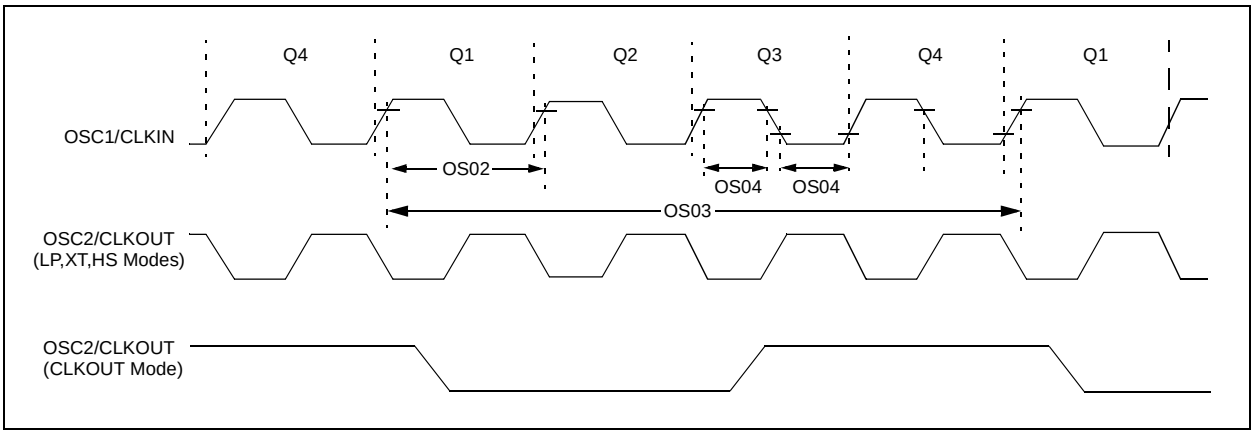
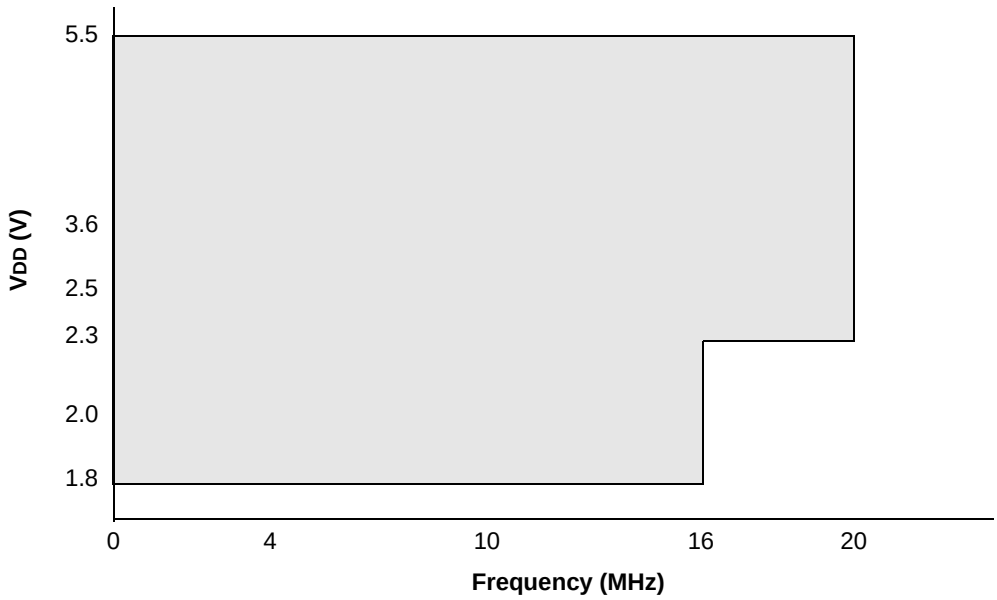
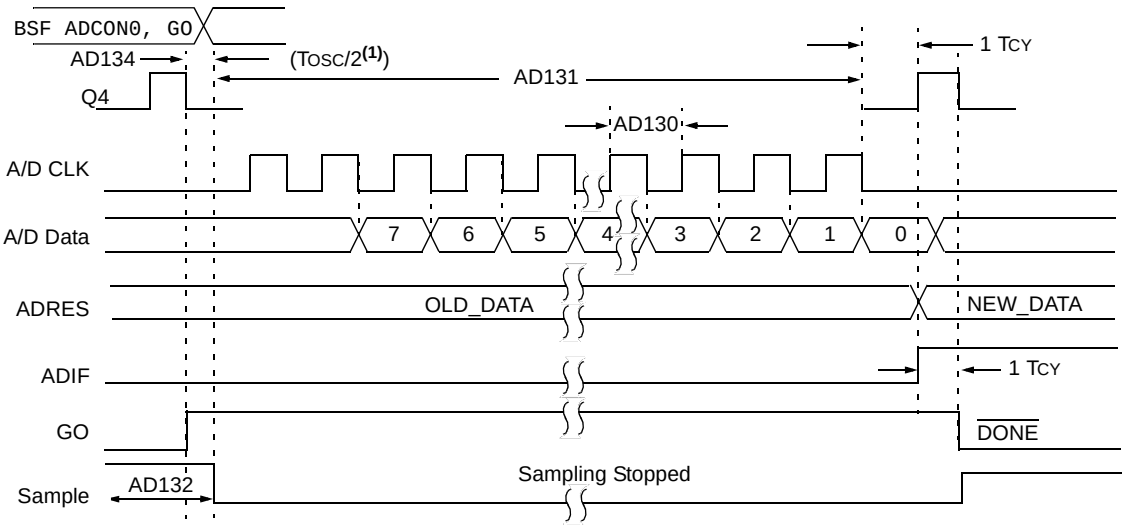



**FIGURE 23-4: PIC16F722/3/4/6/7 VOLTAGE FREQUENCY GRAPH, -40°C ≤ TA ≤ +125°C**



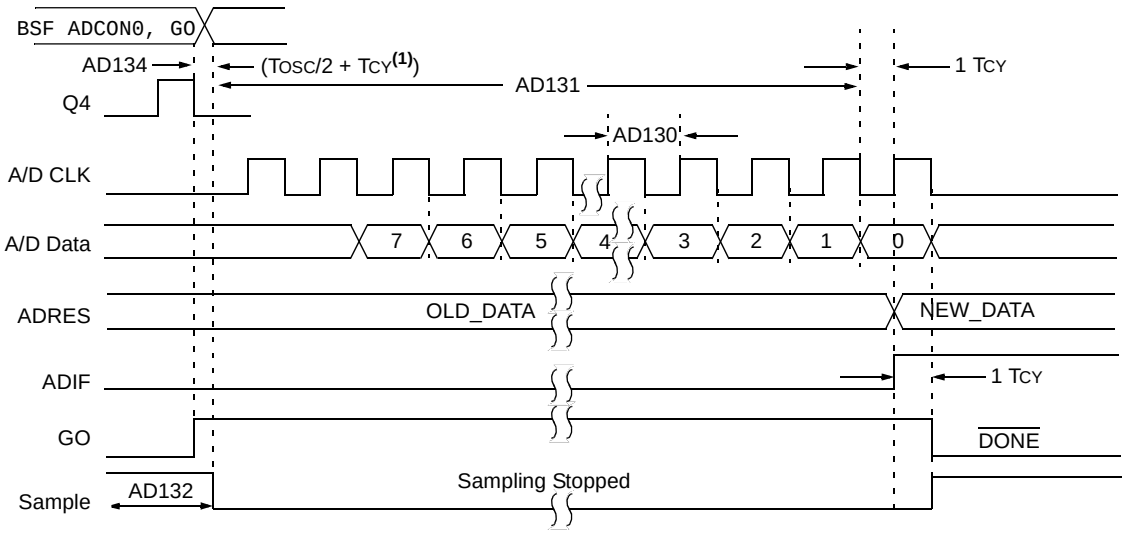

**Note 1:** The shaded region indicates the permissible combinations of voltage and frequency.
**2:** Refer to Table 23-1 for each Oscillator mode's supported frequencies.

**FIGURE 23-12: PIC16F722/3/4/6/7 A/D CONVERSION TIMING (NORMAL MODE)**



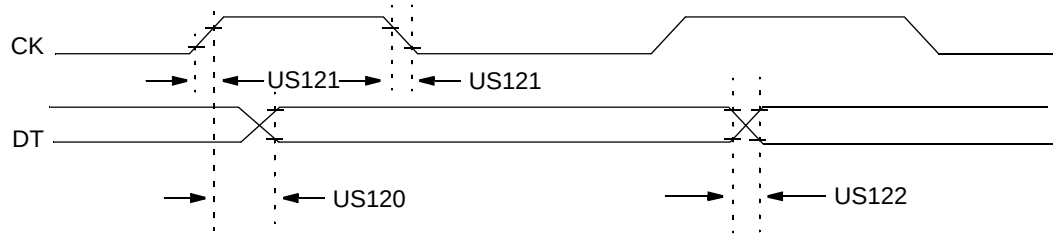

**Note 1:** If the A/D clock source is selected as RC, a time of TCY is added before the A/D clock starts. This allows the SLEEP instruction to be executed.

**FIGURE 23-13: PIC16F722/3/4/6/7 A/D CONVERSION TIMING (SLEEP MODE)**




**Note 1:** If the A/D clock source is selected as RC, a time of TCY is added before the A/D clock starts. This allows the SLEEP instruction to be executed.

**FIGURE 23-14: USART SYNCHRONOUS TRANSMISSION (MASTER/SLAVE) TIMING**

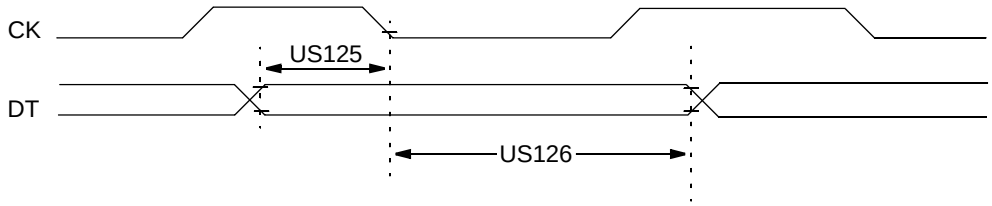



**Note:** Refer to Figure 23-2 for load conditions.

**TABLE 23-9: USART SYNCHRONOUS TRANSMISSION REQUIREMENTS**

| **Standard Operating Conditions (unless otherwise stated)**<br>Operating Temperature -40°C ≤ TA ≤ +125°C | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Param. No.** | **Symbol** | **Characteristic** | | **Min.** | **Max.** | **Units** | **Conditions** |
| US120 | TCKH2DTV | SYNC XMIT (Master and Slave)<br>Clock high to data-out valid | 3.0-5.5V | — | 80 | ns | |
| | | | 1.8-5.5V | — | 100 | ns | |
| US121 | TCKRF | Clock out rise time and fall time (Master mode) | 3.0-5.5V | — | 45 | ns | |
| | | | 1.8-5.5V | — | 50 | ns | |
| US122 | TDTRF | Data-out rise time and fall time | 3.0-5.5V | — | 45 | ns | |
| | | | 1.8-5.5V | — | 50 | ns | |

**FIGURE 23-15: USART SYNCHRONOUS RECEIVE (MASTER/SLAVE) TIMING**




**Note:** Refer to Figure 23-2 for load conditions.

**TABLE 23-10: USART SYNCHRONOUS RECEIVE REQUIREMENTS**

| **Standard Operating Conditions (unless otherwise stated)**<br>Operating Temperature -40°C ≤ TA ≤ +125°C | | | | | | |
|---|---|---|---|---|---|---|
| **Param. No.** | **Symbol** | **Characteristic** | **Min.** | **Max.** | **Units** | **Conditions** |
| US125 | TDTV2CKL | SYNC RCV (Master and Slave)<br>Data-hold before CK ↓ (DT hold time) | 10 | — | ns | |
| US126 | TCKL2DTL | Data-hold after CK ↓ (DT hold time) | 15 | — | ns | |

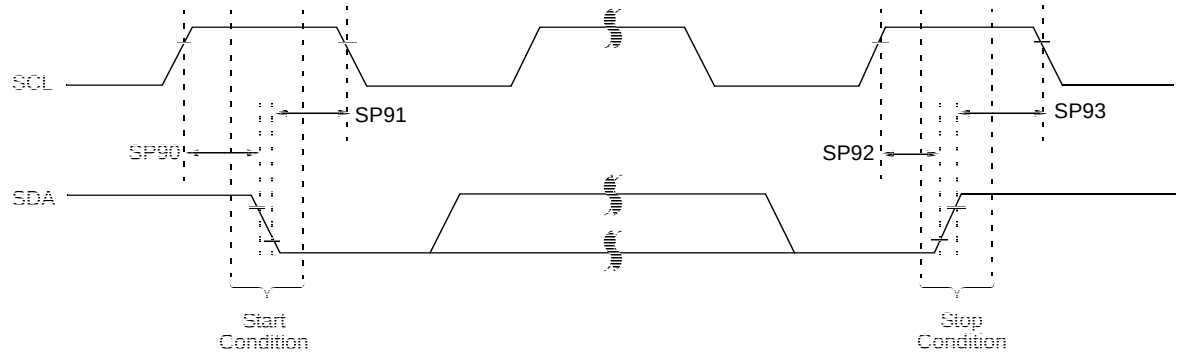
**TABLE 23-11: SPI MODE REQUIREMENTS**

| Param No. | Symbol | Characteristic | | Min. | Typ† | Max. | Units | Conditions |
|---|---|---|---|---|---|---|---|---|
| SP70* | TssL2scH, TssL2scL | $\overline{SS}\downarrow$ to SCK↓ or SCK↑ input | | Tcy | — | — | ns | |
| SP71* | TscH | SCK input high time (Slave mode) | | Tcy + 20 | — | — | ns | |
| SP72* | TscL | SCK input low time (Slave mode) | | Tcy + 20 | — | — | ns | |
| SP73* | TdiV2scH, TdiV2scL | Setup time of SDI data input to SCK edge | | 100 | — | — | ns | |
| SP74* | TscH2diL, TscL2diL | Hold time of SDI data input to SCK edge | | 100 | — | — | ns | |
| SP75* | TdoR | SDO data output rise time | 3.0-5.5V | — | 10 | 25 | ns | |
| | | | 1.8-5.5V | — | 25 | 50 | ns | |
| SP76* | TdoF | SDO data output fall time | | — | 10 | 25 | ns | |
| SP77* | TssH2doZ | $\overline{SS}\uparrow$ to SDO output high-impedance | | 10 | — | 50 | ns | |
| SP78* | TscR | SCK output rise time (Master mode) | 3.0-5.5V | — | 10 | 25 | ns | |
| | | | 1.8-5.5V | — | 25 | 50 | ns | |
| SP79* | TscF | SCK output fall time (Master mode) | | — | 10 | 25 | ns | |
| SP80* | TscH2doV, TscL2doV | SDO data output valid after SCK edge | 3.0-5.5V | — | — | 50 | ns | |
| | | | 1.8-5.5V | — | — | 145 | ns | |
| SP81* | TdoV2scH, TdoV2scL | SDO data output setup to SCK edge | | Tcy | — | — | ns | |
| SP82* | TssL2doV | SDO data output valid after $\overline{SS}\downarrow$ edge | | — | — | 50 | ns | |
| SP83* | TscH2ssH, TscL2ssH | $\overline{SS}\uparrow$ after SCK edge | | 1.5Tcy + 40 | — | — | ns | |

* These parameters are characterized but not tested.

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**FIGURE 23-20: $I^2C$ BUS START/STOP BITS TIMING**




**Note**: Refer to Figure 23-2 for load conditions.

**FIGURE 24-19: PIC16F722/3/4/6/7 MAXIMUM IDD vs. FOSC OVER VDD, INTOSC MODE, VCAP = 0.1 µF**

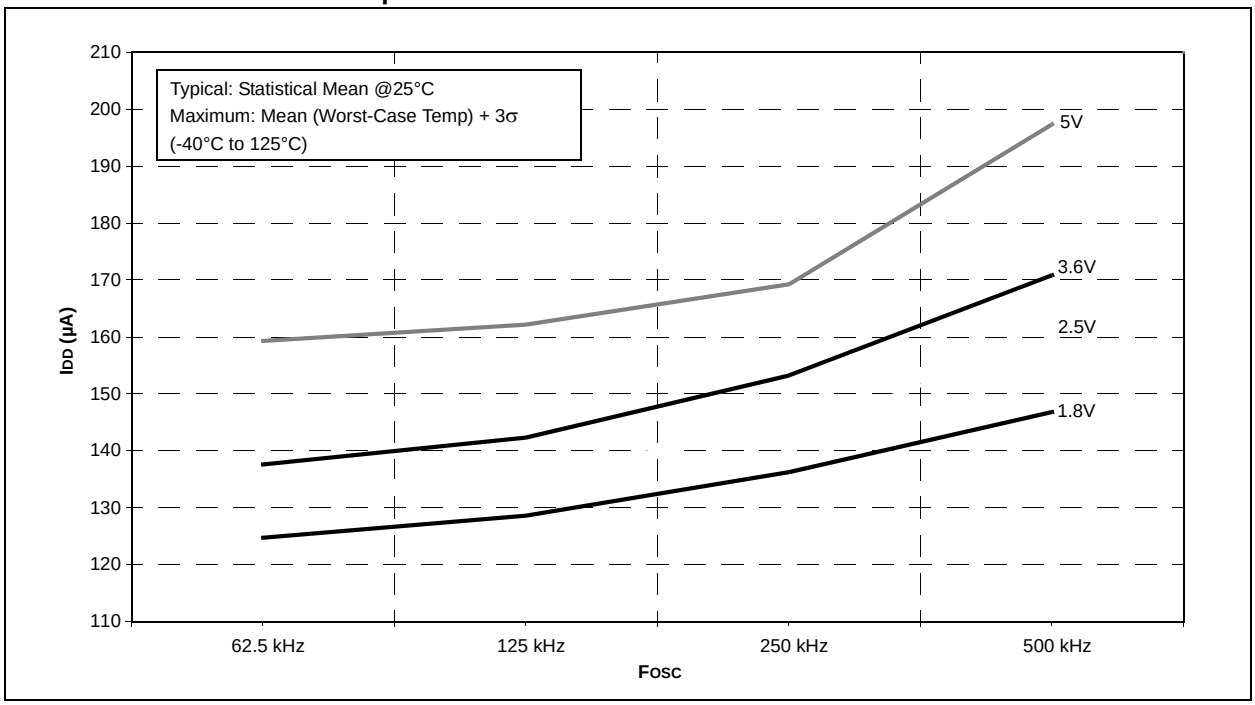
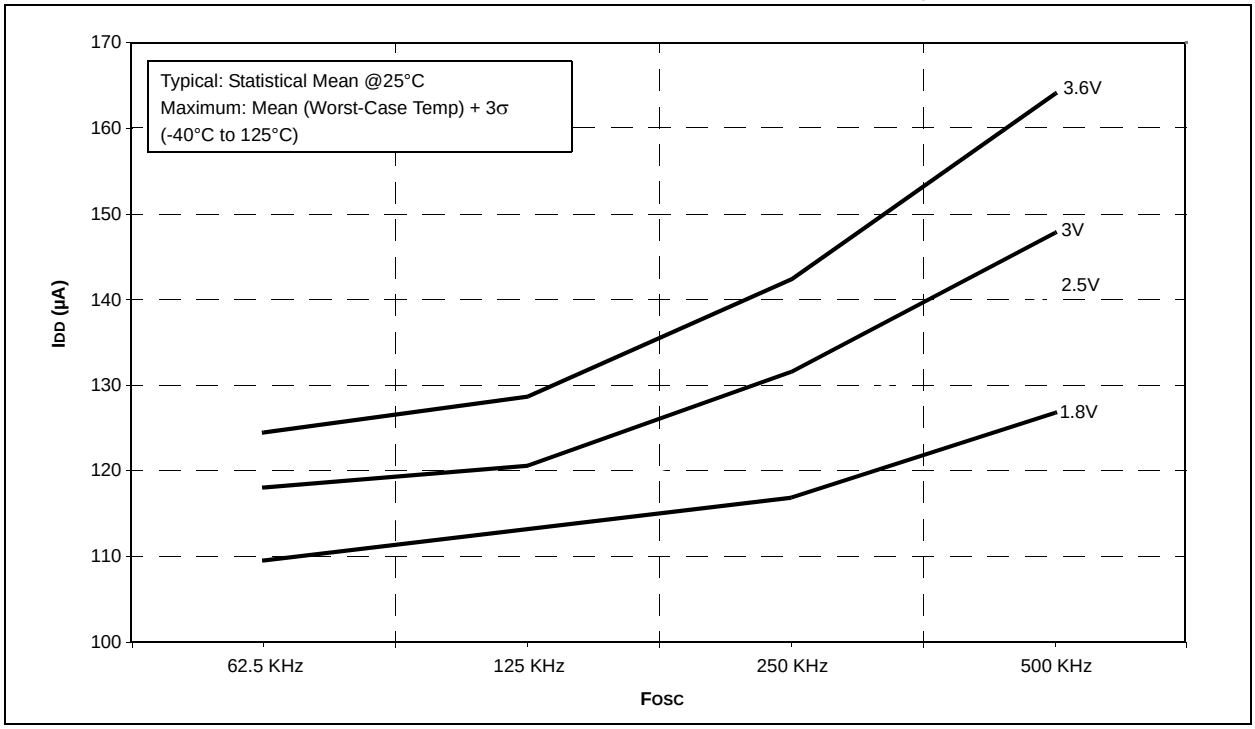



**FIGURE 24-20: PIC16LF722/3/4/6/7 MAXIMUM IDD vs. FOSC OVER VDD, INTOSC MODE**

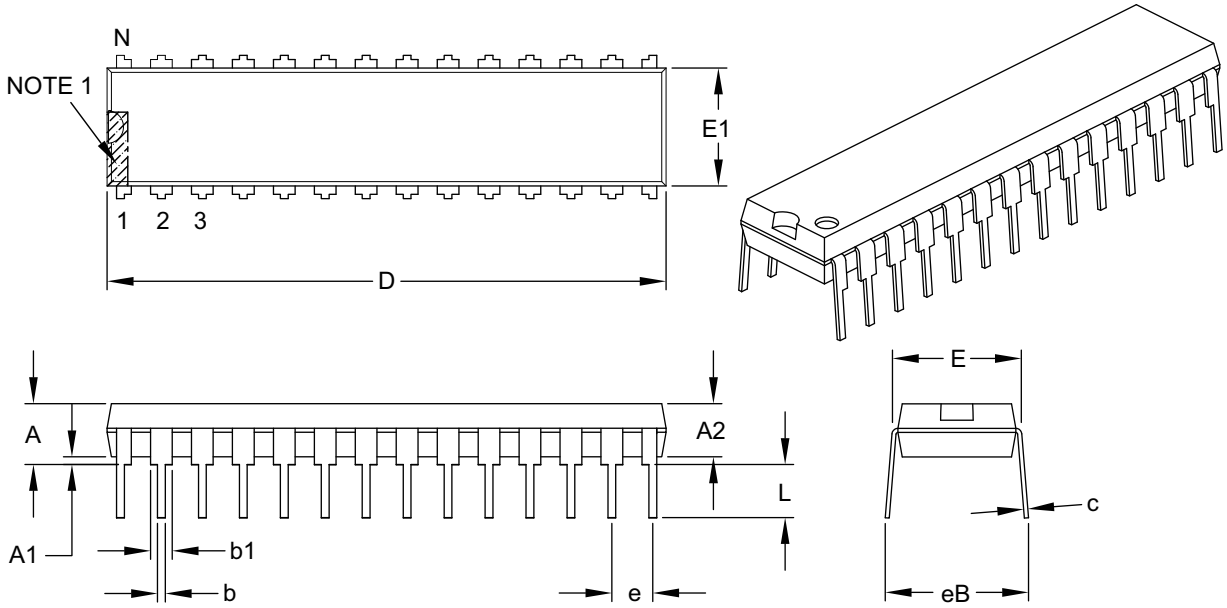


## 25.2 Package Details

The following sections give the technical details of the packages.

### 28-Lead Skinny Plastic Dual In-Line (SP) – 300 mil Body [SPDIP]

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




| Units | | INCHES | | |
|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX |
| Number of Pins | N | | 28 | |
| Pitch | e | | .100 BSC | |
| Top to Seating Plane | A | – | – | .200 |
| Molded Package Thickness | A2 | .120 | .135 | .150 |
| Base to Seating Plane | A1 | .015 | – | – |
| Shoulder to Shoulder Width | E | .290 | .310 | .335 |
| Molded Package Width | E1 | .240 | .285 | .295 |
| Overall Length | D | 1.345 | 1.365 | 1.400 |
| Tip to Seating Plane | L | .110 | .130 | .150 |
| Lead Thickness | c | .008 | .010 | .015 |
| Upper Lead Width | b1 | .040 | .050 | .070 |
| Lower Lead Width | b | .014 | .018 | .022 |
| Overall Row Spacing § | eB | – | – | .430 |

**Notes:**

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic.
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-070B