



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	16MHz
Connectivity	I <sup>2</sup> C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	17
Program Memory Size	7KB (4K x 14)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 12x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	20-SSOP (0.209", 5.30mm Width)
Supplier Device Package	20-SSOP
Purchase URL	<a href="https://www.e-xfl.com/product-detail/microchip-technology/pic16lf721t-i-ss">https://www.e-xfl.com/product-detail/microchip-technology/pic16lf721t-i-ss</a>

# PIC16(L)F720/721

## PIC16(L)F72X Family Types

Device	Data Sheet Index	Program Memory Flash (words)	Data SRAM (bytes)	High-Endurance Flash Memory (bytes)	I/O's <sup>(2)</sup>	8-bit ADC (ch)	CapSense (ch)	Timers (8/16-bit)	AUSART	SSP (I <sup>2</sup> C/SPI)	CCP	Debug <sup>(1)</sup>	XLP
PIC16(L)F707	(1)	8192	363	0	36	14	32	4/2	1	1	2	I	Y
PIC16(L)F720	(2)	2048	128	128	18	12	—	2/1	1	1	1	I	Y
PIC16(L)F721	(2)	4096	256	128	18	12	—	2/1	1	1	1	I	Y
PIC16(L)F722	(4)	2048	128	0	25	11	8	2/1	1	1	2	I	Y
PIC16(L)F722A	(3)	2048	128	0	25	11	8	2/1	1	1	2	I	Y
PIC16(L)F723	(4)	4096	192	0	25	11	8	2/1	1	1	2	I	Y
PIC16(L)F723A	(3)	4096	192	0	25	11	8	2/1	1	1	2	I	Y
PIC16(L)F724	(4)	4096	192	0	36	14	16	2/1	1	1	2	I	Y
PIC16(L)F726	(4)	8192	368	0	25	11	8	2/1	1	1	2	I	Y
PIC16(L)F727	(4)	8192	368	0	36	14	16	2/1	1	1	2	I	Y

**Note 1:** I - Debugging, Integrated on Chip; H - Debugging, Requires Debug Header.

**2:** One pin is input-only.

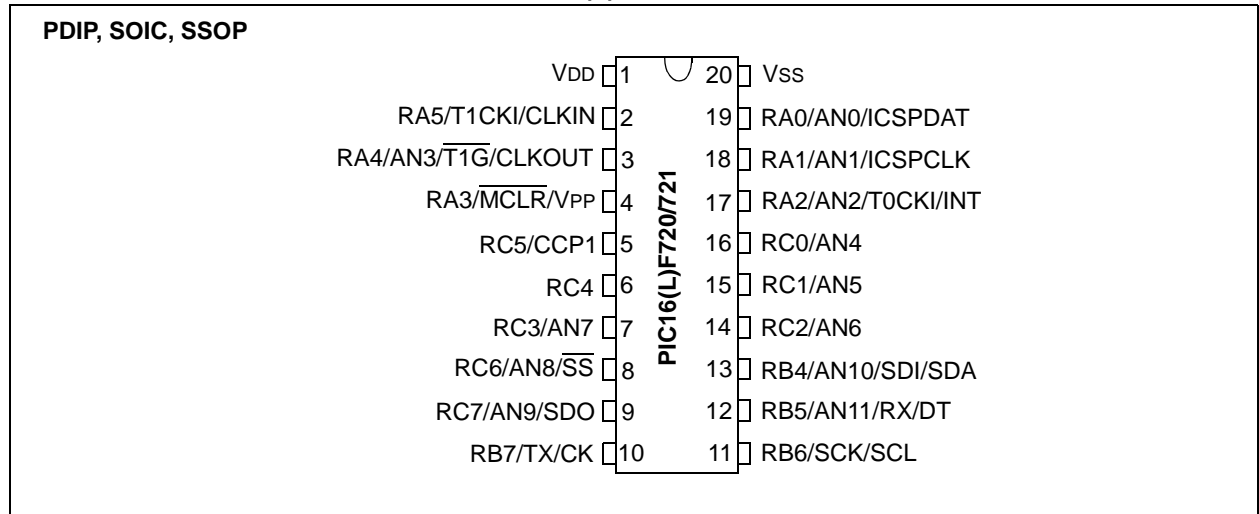
**Data Sheet Index:** (Unshaded devices are described in this document.)

- 1:** DS41418 PIC16(L)F707 Data Sheet, 40/44-Pin Flash, 8-bit Microcontrollers
- 2:** DS41430 PIC16(L)F720/721 Data Sheet, 20-Pin Flash, 8-bit Microcontrollers
- 3:** DS41417 PIC16(L)F722A/723A Data Sheet, 28-Pin Flash, 8-bit Microcontrollers
- 4:** DS41341 PIC16(L)F72X Data Sheet, 28/40/44-Pin Flash, 8-bit Microcontrollers

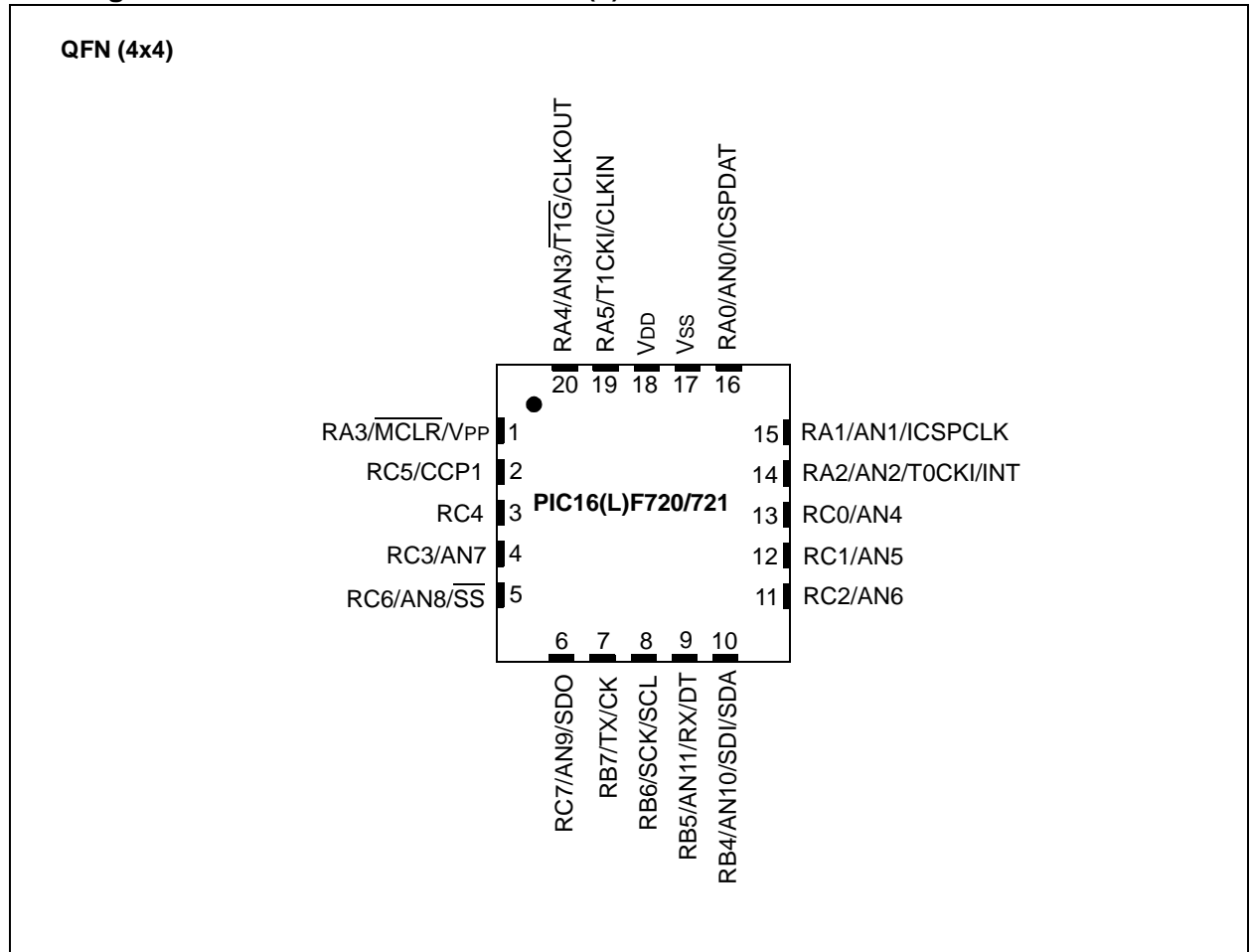
**Note:** For other small form-factor package availability and marking information, please visit <http://www.microchip.com/packaging> or contact your local sales office.

## PIN DIAGRAMS

FIGURE 1: 20-PIN DIAGRAM FOR PIC16(L)F720/721



Pin Diagrams – 20-PIN DIAGRAM FOR PIC16(L)F720/721



# PIC16(L)F720/721

**TABLE 2-2: 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	Value on all other Resets
Bank 2											
100h <sup>(2)</sup>	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx
101h	TMR0	Timer0 module Register								xxxx xxxx	uuuu uuuu
102h <sup>(2)</sup>	PCL	Program Counter (PC) Least Significant Byte								0000 0000	0000 0000
103h <sup>(2)</sup>	STATUS	IRP	RP1	RP0	TO	PD	Z	DC	C	0001 1xxx	000q quuu
104h <sup>(2)</sup>	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	uuuu uuuu
105h	—	Unimplemented								—	—
106h	—	Unimplemented								—	—
107h	—	Unimplemented								—	—
108h	—	Unimplemented								—	—
109h	—	Unimplemented								—	—
10Ah <sup>(1)(2)</sup>	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	---0 0000
10Bh <sup>(2)</sup>	INTCON	GIE	PEIE	TMR0IE	INTE	RABIE	TMR0IF	INTF	RABIF	0000 000x	0000 000x
10Ch	PMDATL	Program Memory Read Data Register Low Byte								xxxx xxxx	xxxx xxxx
10Dh	PMADRL	Program Memory Read Address Register Low Byte								0000 0000	0000 0000
10Eh	PMDATH	—	—	Program Memory Read Data Register High Byte					--xx xxxx	--xx xxxx	
10Fh	PMADRH	—	—	—	Program Memory Read Address Register High Byte					---0 0000	---0 0000
110h	—	Unimplemented								—	—
111h	—	Unimplemented								—	—
112h	—	Unimplemented								—	—
113h	—	Unimplemented								—	—
114h	—	Unimplemented								—	—
115h	WPUB	WPUB7	WPUB6	WPUB5	WPUB4	—	—	—	—	1111 ----	1111 ----
116h	IOCB	IOCB7	IOCB6	IOCB5	IOCB4	—	—	—	—	0000 ----	0000 ----
117h	—	Unimplemented								—	—
118h	—	Unimplemented								—	—
119h	—	Unimplemented								—	—
11Ah	—	Unimplemented								—	—
11Bh	—	Unimplemented								—	—
11Ch	—	Unimplemented								—	—
11Dh	—	Unimplemented								—	—
11Eh	—	Unimplemented								—	—
11Fh	—	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:** Accessible only when SSPM<3:0> = 1001.
- 4:** This bit is unimplemented and reads as '1'.
- 5:** See Register 6-2.

# PIC16(L)F720/721

## 4.0 INTERRUPTS

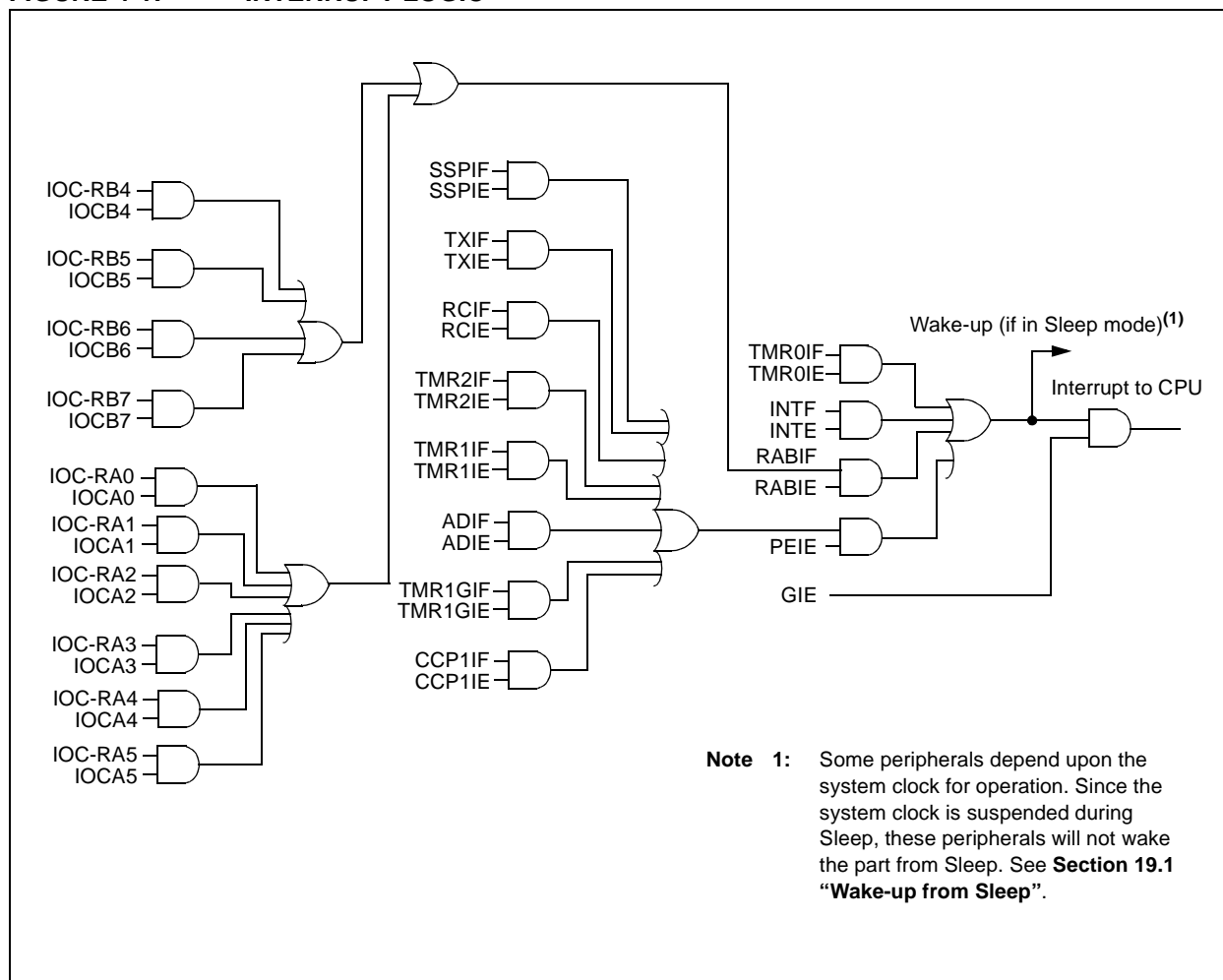
The PIC16(L)F720/721 device family features an interruptible core, allowing certain events to preempt normal program flow. An Interrupt Service Routine (ISR) is used to determine the source of the interrupt and act accordingly. Some interrupts can be configured to wake the MCU from Sleep mode.

The PIC16(L)F720/721 device family has 11 interrupt sources, differentiated by corresponding interrupt enable and flag bits:

- Timer0 Overflow Interrupt
- External Edge Detect on INT Pin Interrupt
- Interrupt-on-change, PORTA and PORTB pins
- Timer1 Gate Interrupt
- A/D Conversion Complete Interrupt
- AUSART Receive Interrupt
- AUSART Transmit Interrupt
- SSP Event Interrupt
- CCP1 Event Interrupt
- Timer2 Match with PR2 Interrupt
- Timer1 Overflow Interrupt

A block diagram of the interrupt logic is shown in Figure 4-1.

**FIGURE 4-1: INTERRUPT LOGIC**



## REGISTER 6-1: PORTA: PORTA REGISTER

U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	RA5	RA4	RA3 <sup>(1)</sup>	RA2	RA1	RA0
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 7-6                      **Unimplemented:** Read as '0'

bit 5-0                      **RA<5:0>:** PORTA I/O Pin bit

1 = Port pin is > V<sub>IH</sub>

0 = Port pin is < V<sub>IL</sub>

**Note 1:** RA<3> is input only.

## REGISTER 6-2: TRISA: PORTA TRI-STATE REGISTER

U-0	U-0	R/W-1	R/W-1	U-1	R/W-1	R/W-1	R/W-1
—	—	TRISA5	TRISA4	— <sup>(1)</sup>	TRISA2	TRISA1	TRISA0
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 7-6                      **Unimplemented:** Read as '0'

bit 5-4                      **TRISA<5:4>:** PORTA Tri-State Control bit

1 = PORTA pin configured as an input (tri-stated)

0 = PORTA pin configured as an output

bit 3                      **Unimplemented:** Read as '1'

bit 2-0                      **TRISA<2:0>:** PORTA Tri-State Control bit

1 = PORTA pin configured as an input (tri-stated)

0 = PORTA pin configured as an output

**Note 1:** TRISA<3> is unimplemented and read as 1.

## REGISTER 6-3: WPUA: WEAK PULL-UP PORTA REGISTER

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	WPUA5	WPUA4	WPUA3 <sup>(2)</sup>	WPUA2	WPUA1	WPUA0
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 7-6                      **Unimplemented:** Read as '0'

bit 5-0                      **WPUA<5:0>:** Weak Pull-up PORTA Control bits

1 = Weak pull-up enabled<sup>(1)</sup>

0 = Weak pull-up disabled

**Note 1:** Enabling weak pull-ups also requires that the RABPU bit of the OPTION\_REG register be cleared.

**2:** If MCLREN = 1, WPUA3 is always enabled.

# PIC16(L)F720/721

## 9.1 ADC Configuration

When configuring and using the ADC, the following functions must be considered:

- Port Configuration
- Channel selection
- ADC conversion clock source
- Interrupt control

### 9.1.1 PORT CONFIGURATION

When converting analog signals, the I/O pin selected as the input channel should be configured for analog by setting the associated TRIS and ANSEL bits. Refer to **Section 6.0 “I/O Ports”** for more information.

**Note:** Analog voltages on any pin that is defined as a digital input may cause the input buffer to conduct excess current.

### 9.1.2 CHANNEL SELECTION

There are 14 channel selections available:

- AN<11:0> pins
- Temperature Indicator
- FVR (Fixed Voltage Reference) Output

Refer to **Section 11.0 “Temperature Indicator Module”** and **Section 10.0 “Fixed Voltage Reference”** for more information on these channel selections.

The CHS bits of the ADCON0 register determine which channel is connected to the sample and hold circuit.

When changing channels, a delay is required before starting the next conversion. Refer to **Section 9.2 “ADC Operation”** for more information.

### 9.1.3 CONVERSION CLOCK

The source of the conversion clock is software-selectable via the ADCS bits of the ADCON1 register. There are seven possible clock options:

- Fosc/2
- Fosc/4
- Fosc/8
- Fosc/16
- Fosc/32
- Fosc/64
- FRC (dedicated internal oscillator)

The time to complete one bit conversion is defined as TAD. One full 8-bit conversion requires 10 TAD periods as shown in Figure 9-2.

For correct conversion, the appropriate TAD specification must be met. Refer to the A/D conversion requirements in **Section 23.0 “Electrical Specifications”** for more information. Table 9-1 gives examples of appropriate ADC clock selections.

**Note:** Unless using the FRC, any changes in the system clock frequency will change the ADC clock frequency, which may adversely affect the ADC result.

**TABLE 9-1: ADC CLOCK PERIOD (TAD) Vs. DEVICE OPERATING FREQUENCIES**

ADC Clock Period (TAD)		Device Frequency (Fosc)			
ADC Clock Source	ADCS<2:0>	16 MHz	8 MHz	4 MHz	1 MHz
Fosc/2	000	125 ns <sup>(2)</sup>	250 ns <sup>(2)</sup>	500 ns <sup>(2)</sup>	2.0 µs
Fosc/4	100	250 ns <sup>(2)</sup>	500 ns <sup>(2)</sup>	1.0 µs	4.0 µs
Fosc/8	001	0.5 µs <sup>(2)</sup>	1.0 µs	2.0 µs	8 µs <sup>(5)</sup>
Fosc/16	101	1.0 µs	2.0 µs	4.0 µs	16.0 µs <sup>(5)</sup>
Fosc/32	010	2.0 µs	4.0 µs	8 µs <sup>(5)</sup>	32.0 µs <sup>(3)</sup>
Fosc/64	110	4.0 µs	8 µs <sup>(5)</sup>	16.0 µs <sup>(5)</sup>	64.0 µs <sup>(3)</sup>
FRC	x11	1.0-6.0 µs <sup>(1,4)</sup>	1.0-6.0 µs <sup>(1,4)</sup>	1.0-6.0 µs <sup>(1,4)</sup>	1.0-6.0 µs <sup>(1,4)</sup>

**Legend:** Shaded cells are outside of the recommended range.

**Note 1:** The FRC source has a typical TAD time of 1.6 µs for VDD.

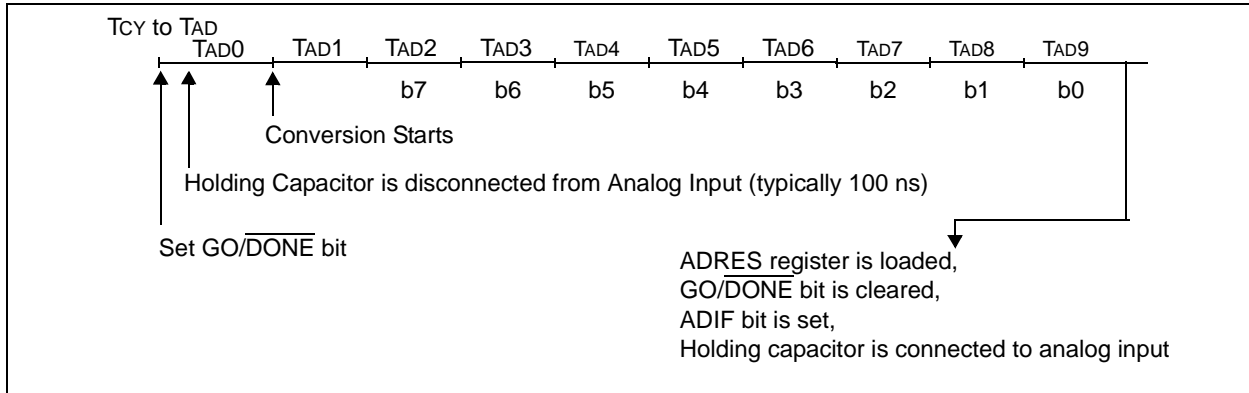
**2:** These values violate the minimum required TAD time.

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

**4:** When the device frequency is greater than 1 MHz, the FRC clock source is only recommended if the conversion will be performed during Sleep.

**5:** Recommended values for VDD ≤ 2.0V and temperature -40°C to 85°C. The 16.0 µs setting should be avoided for temperature > 85°C.

**FIGURE 9-2: ANALOG-TO-DIGITAL CONVERSION TAD CYCLES**



## 9.1.4 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 9.1.4 “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/DONE bit of the ADCON0 register to a ‘1’ will start the Analog-to-Digital conversion.

**Note:** The GO/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/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/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.



## 13.3 Timer1 Prescaler

Timer1 has four prescaler options allowing 1, 2, 4 or 8 divisions of the clock input. The T1CKPS bits of the T1CON register control the prescaler counter. The prescale counter is not directly readable or writable; however, the prescaler counter is cleared upon a write to TMR1H or TMR1L.

## 13.4 Timer1 Operation in Asynchronous Counter Mode

If the control bit  $\overline{\text{T1SYNC}}$  of the T1CON register is set, the external clock input is not synchronized. The timer increments asynchronously to the internal phase clocks. If external clock source is selected then 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 (see **Section 13.4.1 “Reading and Writing Timer1 in Asynchronous Counter Mode”**).

**Note:** When switching from synchronous to asynchronous operation, it is possible to skip an increment. When switching from asynchronous to synchronous operation, it is possible to produce an additional increment.

### 13.4.1 READING AND WRITING TIMER1 IN ASYNCHRONOUS COUNTER MODE

Reading TMR1H or TMR1L while the timer is running from an external asynchronous clock will ensure 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 TMR1H:TMR1L register pair.

## 13.5 Timer1 Gate

Timer1 can be configured to count freely or the count can be enabled and disabled using Timer1 gate circuitry. This is also referred to as Timer1 gate count enable.

Timer1 gate can also be driven by multiple selectable sources.

### 13.5.1 TIMER1 GATE COUNT ENABLE

The Timer1 gate is enabled by setting the TMR1GE bit of the T1GCON register. The polarity of the Timer1 gate is configured using the T1GPOL bit of the T1GCON register.

When Timer1 Gate ( $\overline{\text{T1G}}$ ) input is active, Timer1 will increment on the rising edge of the Timer1 clock source. When Timer1 gate input is inactive, no incrementing will occur and Timer1 will hold the current count. See Figure 13-3 for timing details.

**TABLE 13-3: TIMER1 GATE ENABLE SELECTIONS**

T1CLK	T1GPOL	$\overline{\text{T1G}}$	Timer1 Operation
↑	0	0	Counts
↑	0	1	Holds Count
↑	1	0	Holds Count
↑	1	1	Counts

### 13.5.2 TIMER1 GATE SOURCE SELECTION

The Timer1 gate source can be selected from one of four different sources. Source selection is controlled by the T1GSS bits of the T1GCON register. The polarity for each available source is also selectable. Polarity selection is controlled by the T1GPOL bit of the T1GCON register.

**TABLE 13-4: TIMER1 GATE SOURCES**

T1GSS	Timer1 Gate Source
00	Timer1 Gate Pin
01	Overflow of Timer0 (TMR0 increments from FFh to 00h)
10	Timer2 match PR2 (TMR2 increments to match PR2)
11	Count Enabled by WDT Overflow (Watchdog Time-out interval expired)

# PIC16(L)F720/721

## 15.0 CAPTURE/COMPARE/PWM (CCP) MODULE

The Capture/Compare/PWM module is a peripheral which allows the user to time and control different events. In Capture mode, the peripheral allows the timing of the duration of an event. The Compare mode allows the user to trigger an external event when a predetermined amount of time has expired. The PWM mode can generate a Pulse-Width Modulated signal of varying frequency and duty cycle.

The timer resources used by the module are shown in Table 15-1.

Additional information on CCP modules is available in the Application Note AN594, "Using the CCP Modules" (DS00594).

TABLE 15-1: CCP MODE – TIMER RESOURCES REQUIRED

CCP Mode	Timer Resource
Capture	Timer1
Compare	Timer1
PWM	Timer2

### REGISTER 15-1: CCP1CON: CCP1 CONTROL REGISTER

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	DC1	B1	CCP1M3	CCP1M2	CCP1M1	CCP1M0
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 7-6 **Unimplemented:** Read as '0'

bit 5-4 **DC1:B1:** PWM Duty Cycle 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 CCPR1L.

bit 3-0 **CCP1M<3:0>:** CCP mode Select bits

0000 = Capture/Compare/PWM off (resets CCP module)

0001 = Unused (reserved)

0010 = Compare mode, toggle output on match (CCP1IF bit of the PIRx register is set)

0011 = Unused (reserved)

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 (CCP1IF bit of the PIR1 register is set)

1001 = Compare mode, clear output on match (CCP1IF bit of the PIR1 register is set)

1010 = Compare mode, generate software interrupt on match (CCP1IF bit is set of the PIRx register, CCP1 pin is unaffected)

1011 = Compare mode, trigger special event (CCP1IF bit of the PIR1 register is set, TMR1 is reset and A/D conversion is started if the ADC module is enabled. CCP1 pin is unaffected.)

11xx = PWM mode.

# PIC16(L)F720/721

## 15.3.2 PWM PERIOD

The PWM period is specified by the PR2 register of Timer2. The PWM period can be calculated using the formula of Equation 15-1.

### EQUATION 15-1: PWM PERIOD

$$PWM\ Period = [(PR2) + 1] \cdot 4 \cdot T_{OSC} \cdot (TMR2\ Prescale\ Value)$$

**Note:**  $T_{OSC} = 1/F_{OSC}$

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 the PWM duty cycle = 0%, the pin will not be set.)
- The PWM duty cycle is latched from CCPR1L into CCPR1H.

**Note:** The Timer2 postscaler (refer to **Section 14.1 “Timer2 Operation”**) is not used in the determination of the PWM frequency.

## 15.3.3 PWM DUTY CYCLE

The PWM duty cycle is specified by writing a 10-bit value to multiple registers: CCPR1L register and DC1 and B1 bits of the CCP1CON register. The CCPR1L contains the eight MSbs and the DC1 and B1 bits of the CCP1CON register contain the two LSbs. CCPR1L and DC1 and B1 bits of the CCP1CON register can be written to at any time. The duty cycle value is not latched into CCPR1H until after the period completes (i.e., a match between PR2 and TMR2 registers occurs). While using the PWM, the CCPR1H register is read-only.

Equation 15-2 is used to calculate the PWM pulse width.

Equation 15-3 is used to calculate the PWM duty cycle ratio.

### EQUATION 15-2: PULSE WIDTH

$$Pulse\ Width = (CCPR1L:CCP1CON<5:4>) \cdot T_{OSC} \cdot (TMR2\ Prescale\ Value)$$

**Note:**  $T_{OSC} = 1/F_{OSC}$

### EQUATION 15-3: DUTY CYCLE RATIO

$$Duty\ Cycle\ Ratio = \frac{(CCPR1L:CCP1CON<5:4>)}{4(PR2 + 1)}$$

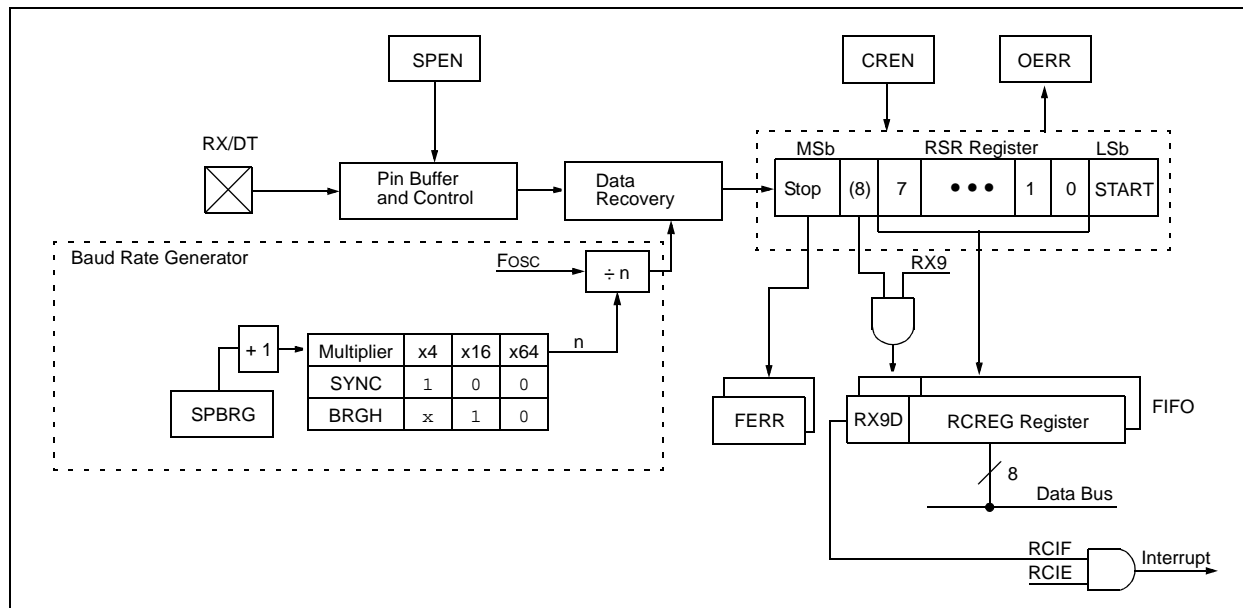
The CCPR1H register and a 2-bit internal latch are used to double buffer the PWM duty cycle. This double buffering is essential for glitchless PWM operation.

The 8-bit timer TMR2 register is concatenated with either the 2-bit internal system clock ( $F_{OSC}$ ), or two bits of the prescaler, to create the 10-bit time base. The system clock is used if the Timer2 prescaler is set to 1:1.

When the 10-bit time base matches the CCPR1H and 2-bit latch, then the CCP1 pin is cleared (refer to Figure 15-3).

# PIC16(L)F720/721

**FIGURE 16-2: AUSART RECEIVE BLOCK DIAGRAM**



The operation of the AUSART module is controlled through two registers:

- Transmit Status and Control (TXSTA)
- Receive Status and Control (RCSTA)

These registers are detailed in Register 16-1 and Register 16-2, respectively.

## 16.1 AUSART Asynchronous Mode

The AUSART transmits and receives data using the standard non-return-to-zero (NRZ) format. NRZ is implemented with two levels: a V<sub>OH</sub> Mark state which represents a '1' data bit, and a V<sub>OL</sub> Space state which represents a '0' data bit. NRZ refers to the fact that consecutively transmitted data bits of the same value stay at the output level of that bit without returning to a neutral level between each bit transmission. An NRZ transmission port idles in the Mark state. Each character transmission consists of one Start bit followed by eight or nine data bits and is always terminated by one or more Stop bits. The Start bit is always a space and the Stop bits are always marks. The most common data format is eight bits. Each transmitted bit persists for a period of 1/(baud rate). An on-chip dedicated 8-bit Baud Rate Generator is used to derive standard baud rate frequencies from the system oscillator. Refer to Table 16-5 for examples of baud rate Configurations.

The AUSART transmits and receives the LSb first. The AUSART's transmitter and receiver are functionally independent, but share the same data format and baud rate. Parity is not supported by the hardware, but can be implemented in software and stored as the ninth data bit.

### 16.1.1 AUSART ASYNCHRONOUS TRANSMITTER

The AUSART transmitter block diagram is shown in Figure 16-1. The heart of the transmitter is the serial Transmit Shift Register (TSR), which is not directly accessible by software. The TSR obtains its data from the transmit buffer, which is the TXREG register.

#### 16.1.1.1 Enabling the Transmitter

The AUSART transmitter is enabled for asynchronous operations by configuring the following three control bits:

- TXEN = 1
- SYNC = 0
- SPEN = 1

All other AUSART control bits are assumed to be in their default state.

Setting the TXEN bit of the TXSTA register enables the transmitter circuitry of the AUSART. Clearing the SYNC bit of the TXSTA register configures the AUSART for asynchronous operation. Setting the SPEN bit of the RCSTA register enables the AUSART and automatically configures the TX/CK I/O pin as an output.

**Note 1:** When the SPEN bit is set the RX/DT I/O pin is automatically configured as an input, regardless of the state of the corresponding TRIS bit and whether or not the AUSART receiver is enabled. The RX/DT pin data can be read via a normal PORT read but PORT latch data output is precluded.

- 2:** The TXIF transmitter interrupt flag is set when the TXEN enable bit is set.

#### 16.1.1.2 Transmitting Data

A transmission is initiated by writing a character to the TXREG register. If this is the first character, or the previous character has been completely flushed from the TSR, the data in the TXREG is immediately transferred to the TSR register. If the TSR still contains all or part of a previous character, the new character data is held in the TXREG until the Stop bit of the previous character has been transmitted. The pending character in the TXREG is then transferred to the TSR in one T<sub>cy</sub> immediately following the Stop bit transmission. The transmission of the Start bit, data bits and Stop bit sequence commences immediately following the transfer of the data to the TSR from the TXREG.

#### 16.1.1.3 Transmit Interrupt Flag

The TXIF interrupt flag bit of the PIR1 register is set whenever the AUSART transmitter is enabled and no character is being held for transmission in TXREG. In other words, the TXIF bit is only clear when TSR is busy with a character and a new character has been queued for transmission in TXREG. The TXIF flag bit is not cleared immediately upon writing TXREG. TXIF becomes valid in the second instruction cycle following the write execution. Polling TXIF immediately following the TXREG write will return invalid results. The TXIF bit is read-only, it cannot be set or cleared by software.

The TXIF interrupt can be enabled by setting the TXIE interrupt enable bit of the PIE1 register. However, the TXIF flag bit will be set whenever TXREG is empty, regardless of the state of the TXIE enable bit.

To use interrupts when transmitting data, set the TXIE bit only when there is more data to send. Clear the TXIE interrupt enable bit upon writing the last character of the transmission to TXREG.

## 17.1.2 SLAVE MODE

For any SPI device acting as a slave, the data is transmitted and received as external clock pulses appear on SCK pin. This external clock must meet the minimum high and low times as specified in the electrical specifications.

### 17.1.2.1 Slave Mode Operation

The SSP consists of a transmit/receive shift register (SSPSR) and a buffer register (SSPBUF). The SSPSR shifts the data in and out of the device, MSb first. The SSPBUF holds the data that was written to the SSPSR until the received data is ready.

The slave has no control as to when data will be clocked in or out of the device. All data that is to be transmitted, to a master or another slave, must be loaded into the SSPBUF register before the first clock pulse is received.

Once eight bits of data have been received:

- Received byte is moved to the SSPBUF register
- BF bit of the SSPSTAT register is set
- SSPIF bit of the PIR1 register is set

Any write to the SSPBUF register during transmission/reception of data will be ignored and the Write Collision Detect bit, WCOL of the SSPCON register, will be set. User software must clear the WCOL bit so that it can be determined if the following write(s) to the SSPBUF register completed successfully.

The user's firmware must read SSPBUF, clearing the BF flag, or the SSPOV bit of the SSPCON register will be set with the reception of the next byte and communication will be disabled.

A SPI module transmits and receives at the same time, occasionally causing dummy data to be transmitted/received. It is up to the user to determine which data is to be used and what can be discarded.

### 17.1.2.2 Enabling Slave I/O

To enable the serial port, the SSPEN bit of the SSPCON register must be set. If a Slave mode of operation is selected in the SSPM bits of the SSPCON register, the SDI, SDO and SCK pins will be assigned as serial port pins.

For these pins to function as serial port pins, they must have their corresponding data direction bits set or cleared in the associated TRIS register as follows:

- SDI configured as input
- SDO configured as output
- SCK configured as input

Optionally, a fourth pin, Slave Select ( $\overline{SS}$ ) may be used in Slave mode. Slave Select may be configured to operate on the RC6/ $\overline{SS}$  pin via the SSSEL bit in the APFCON register.

Upon selection of a Slave Select pin, the appropriate bits must be set in the ANSELA and TRISA registers. Slave Select must be set as an input by setting the corresponding bit in TRISA, and digital I/O must be enabled on the  $\overline{SS}$  pin by clearing the corresponding bit of the ANSELA register.

### 17.1.2.3 Slave Mode Setup

When initializing the SSP module to SPI Slave mode, compatibility must be ensured with the master device. This is done by programming the appropriate control bits of the SSPCON and SSPSTAT registers. These control bits allow the following to be specified:

- SCK as clock input
- Idle state of SCK (CKP bit)
- Data input sample phase (SMP bit)
- Output data on rising/falling edge of SCK (CKE bit)

Figure 17-4 and Figure 17-5 show example waveforms of Slave mode operation.

## 22.2 MPLAB XC Compilers

The MPLAB XC Compilers are complete ANSI C compilers for all of Microchip's 8, 16, and 32-bit MCU and DSC devices. These compilers provide powerful integration capabilities, superior code optimization and ease of use. MPLAB XC Compilers run on Windows, Linux or MAC OS X.

For easy source level debugging, the compilers provide debug information that is optimized to the MPLAB X IDE.

The free MPLAB XC Compiler editions support all devices and commands, with no time or memory restrictions, and offer sufficient code optimization for most applications.

MPLAB XC Compilers include an assembler, linker and utilities. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. MPLAB XC Compiler uses the assembler to produce its object file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command-line interface
- Rich directive set
- Flexible macro language
- MPLAB X IDE compatibility

## 22.3 MPASM Assembler

The MPASM Assembler is a full-featured, universal macro assembler for PIC10/12/16/18 MCUs.

The MPASM Assembler generates relocatable object files for the MPLINK Object Linker, Intel® standard HEX files, MAP files to detail memory usage and symbol reference, absolute LST files that contain source lines and generated machine code, and COFF files for debugging.

The MPASM Assembler features include:

- Integration into MPLAB X IDE projects
- User-defined macros to streamline assembly code
- Conditional assembly for multipurpose source files
- Directives that allow complete control over the assembly process

## 22.4 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler. It can link relocatable objects from precompiled libraries, using directives from a linker script.

The MPLIB Object Librarian manages the creation and modification of library files of precompiled code. When a routine from a library is called from a source file, only the modules that contain that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications.

The object linker/librarian features include:

- Efficient linking of single libraries instead of many smaller files
- Enhanced code maintainability by grouping related modules together
- Flexible creation of libraries with easy module listing, replacement, deletion and extraction

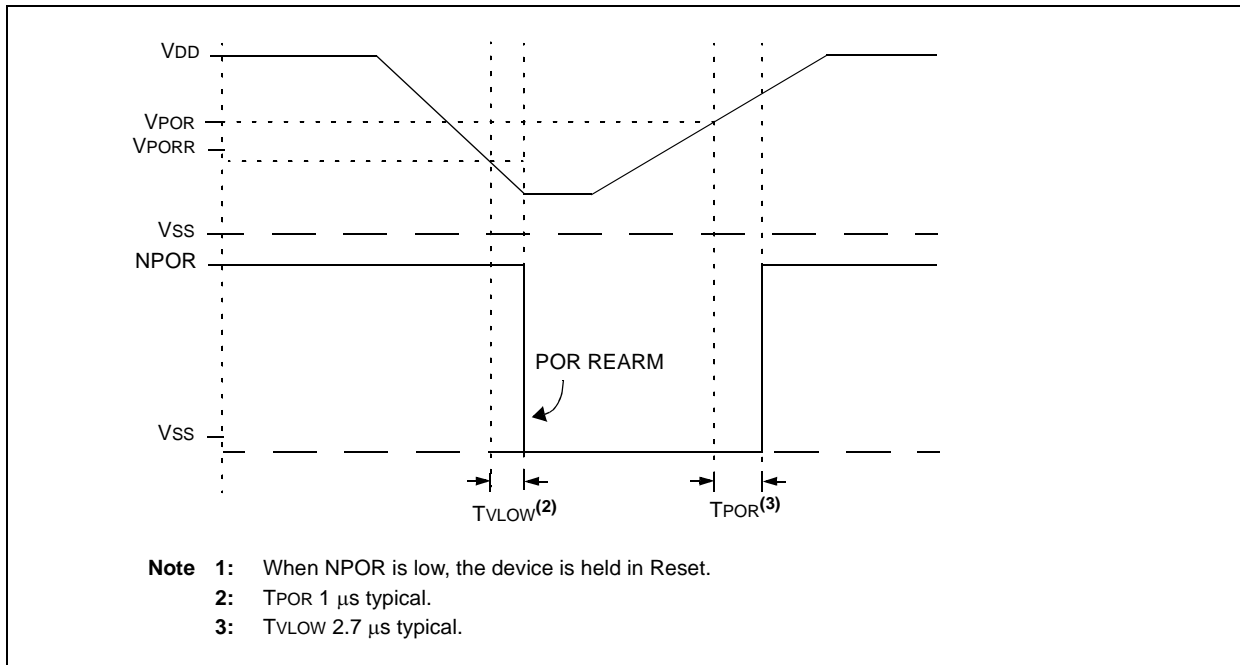
## 22.5 MPLAB Assembler, Linker and Librarian for Various Device Families

MPLAB Assembler produces relocatable machine code from symbolic assembly language for PIC24, PIC32 and dsPIC DSC devices. MPLAB XC Compiler uses the assembler to produce its object file. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command-line interface
- Rich directive set
- Flexible macro language
- MPLAB X IDE compatibility

# PIC16(L)F720/721

FIGURE 23-1: POR AND POR REARM WITH SLOW RISING  $V_{DD}$





**TABLE 23-6: CAPTURE/COMPARE/PWM REQUIREMENTS (CCP)**

Standard Operating Conditions (unless otherwise stated)								
Operating Temperature $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$								
Param. No.	Sym.	Characteristic		Min.	Typ†	Max.	Units	Conditions
CC01*	TccL	CCP Input Low Time	No Prescaler	$0.5T_{CY} + 20$	—	—	ns	
			With Prescaler	20	—	—	ns	
CC02*	TccH	CCP Input High Time	No Prescaler	$0.5T_{CY} + 20$	—	—	ns	
			With Prescaler	20	—	—	ns	
CC03*	TccP	CCP Input Period		$\frac{3T_{CY} + 40}{N}$	—	—	ns	N = prescale value (1, 4 or 16)

\* These parameters are characterized but not tested.

† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**TABLE 23-7: PIC16F720/721 A/D CONVERTER (ADC) CHARACTERISTICS**

Operating Conditions (unless otherwise stated)							
VDD = 3.0V, TA = 25°C							
Param. No.	Sym.	Characteristic	Min.	Typ†	Max.	Units	Conditions
AD01	NR	Resolution	—	—	8	bit	
AD02	EIL	Integral Error	—	—	$\pm 1.7$	LSb	VDD = 3.0V
AD03	EDL	Differential Error	—	—	$\pm 1$	LSb	No missing codes VDD = 3.0V
AD07	EGN	Gain Error	—	—	$\pm 1.5$	LSb	VDD = 3.0V
AD07	VAIN	Full-Scale Range	VSS	—	VDD	V	
AD08*	ZAIN	Recommended Impedance of Analog Voltage Source	—	—	10	kΩ	

\* These parameters are characterized but not tested.

† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

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

Param. No.	Symbol	Characteristic	Min.	Typ†	Max.	Units	Conditions
SP70*	TssL2scH, TssL2scL	$\overline{SS}\downarrow$ to SCK $\downarrow$ or SCK $\uparrow$ 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*	TssH2boZ	$\overline{SS}\uparrow$ 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.

# PIC16(L)F720/721

FIGURE 24-20: PIC16F720/721 BOR IPD vs. VDD

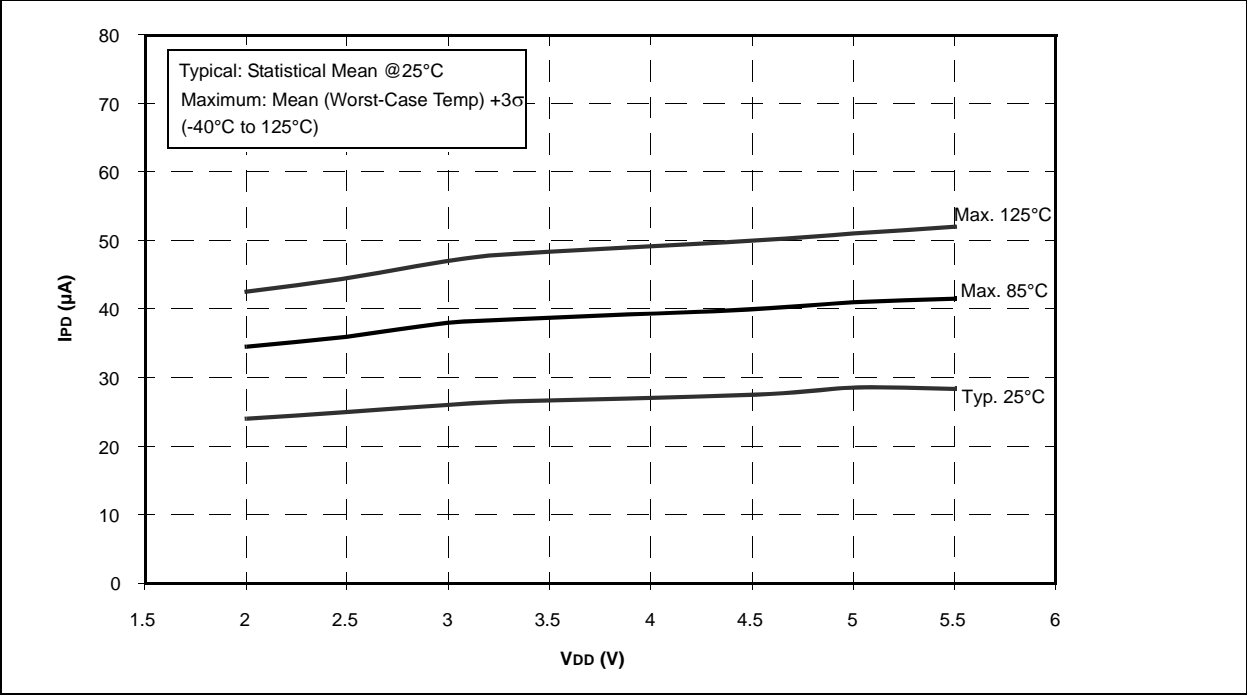
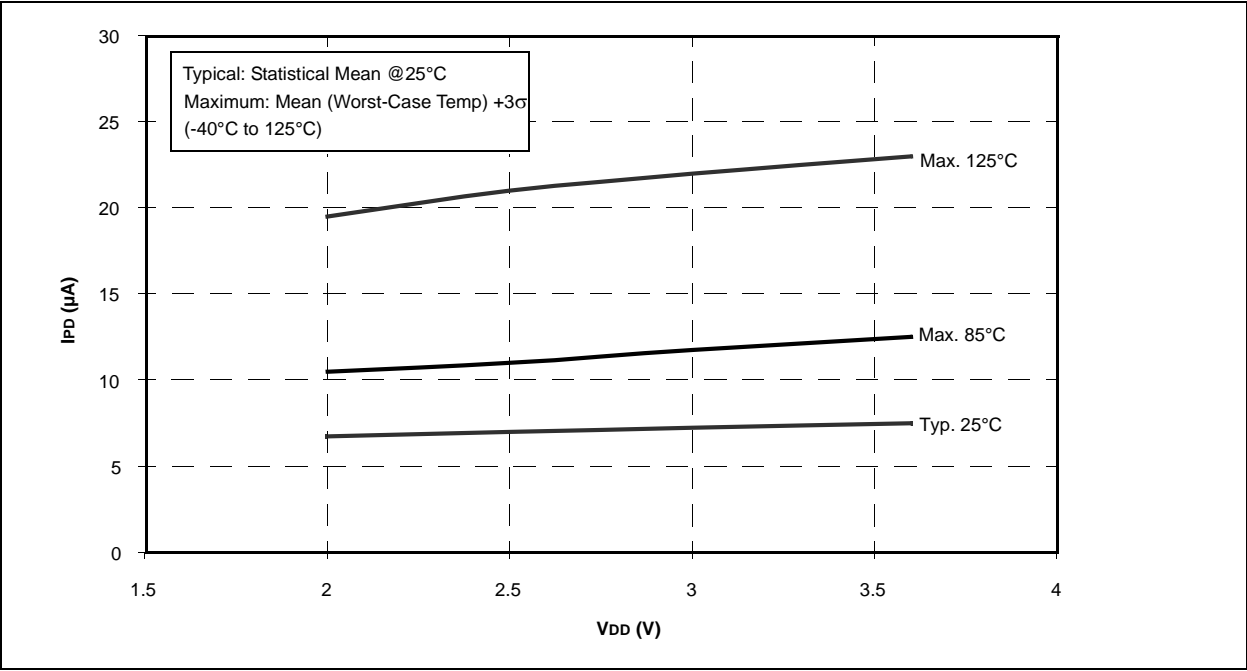


FIGURE 24-21: PIC16LF720/721 BOR IPD vs. VDD



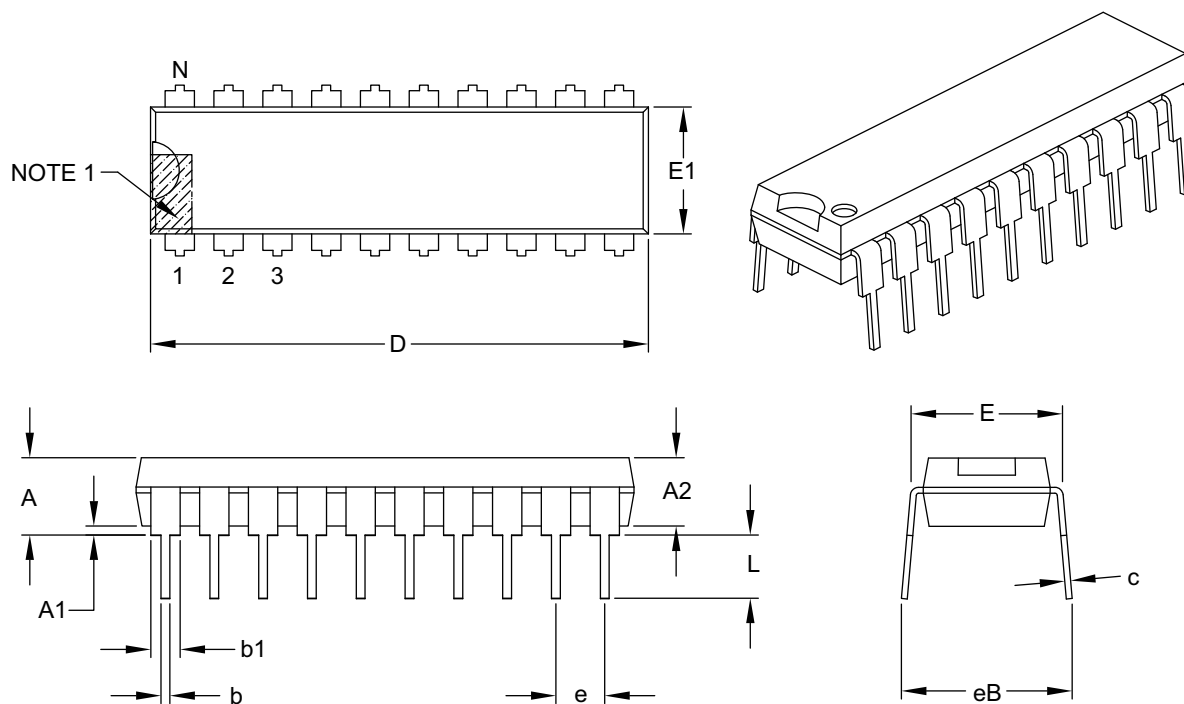
# PIC16(L)F720/721

## 25.2 Package Details

The following sections give the technical details of the packages.

### 20-Lead Plastic Dual In-Line (P) – 300 mil Body [PDIP]

**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	20		
Pitch	e	.100 BSC		
Top to Seating Plane	A	–	–	.210
Molded Package Thickness	A2	.115	.130	.195
Base to Seating Plane	A1	.015	–	–
Shoulder to Shoulder Width	E	.300	.310	.325
Molded Package Width	E1	.240	.250	.280
Overall Length	D	.980	1.030	1.060
Tip to Seating Plane	L	.115	.130	.150
Lead Thickness	c	.008	.010	.015
Upper Lead Width	b1	.045	.060	.070
Lower Lead Width	b	.014	.018	.022
Overall Row Spacing §	eB	–	–	.430

#### Notes:

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

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

Microchip Technology Drawing C04-019B

---

**Note the following details of the code protection feature on Microchip devices:**

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

---

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

#### **Trademarks**

The Microchip name and logo, the Microchip logo, dsPIC, FlashFlex, flexPWR, JukeBlox, KEELOQ, KEELOQ logo, Klear, LANCheck, MediaLB, MOST, MOST logo, MPLAB, OptoLyzer, PIC, PICSTART, PIC<sup>32</sup> logo, RightTouch, SpyNIC, SST, SST Logo, SuperFlash and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

The Embedded Control Solutions Company and mTouch are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, BodyCom, chipKIT, chipKIT logo, CodeGuard, dsPICDEM, dsPICDEM.net, ECAN, In-Circuit Serial Programming, ICSP, Inter-Chip Connectivity, KlearNet, KlearNet logo, MiWi, motorBench, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, RightTouch logo, REAL ICE, SQI, Serial Quad I/O, Total Endurance, TSHARC, USBCheck, VariSense, ViewSpan, WiperLock, Wireless DNA, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

© 2010-2015, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

ISBN: 978-1-5224-0041-7

**QUALITY MANAGEMENT SYSTEM**  
**CERTIFIED BY DNV**  
**== ISO/TS 16949 ==**

*Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.*