



Welcome to [E-XFL.COM](https://www.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	I <sup>2</sup> C, 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 ~ 5.5V
Data Converters	A/D 11x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SSOP (0.209", 5.30mm Width)
Supplier Device Package	28-SSOP
Purchase URL	<a href="https://www.e-xfl.com/product-detail/microchip-technology/pic16f723a-i-ss">https://www.e-xfl.com/product-detail/microchip-technology/pic16f723a-i-ss</a>

# PIC16(L)F722A/723A

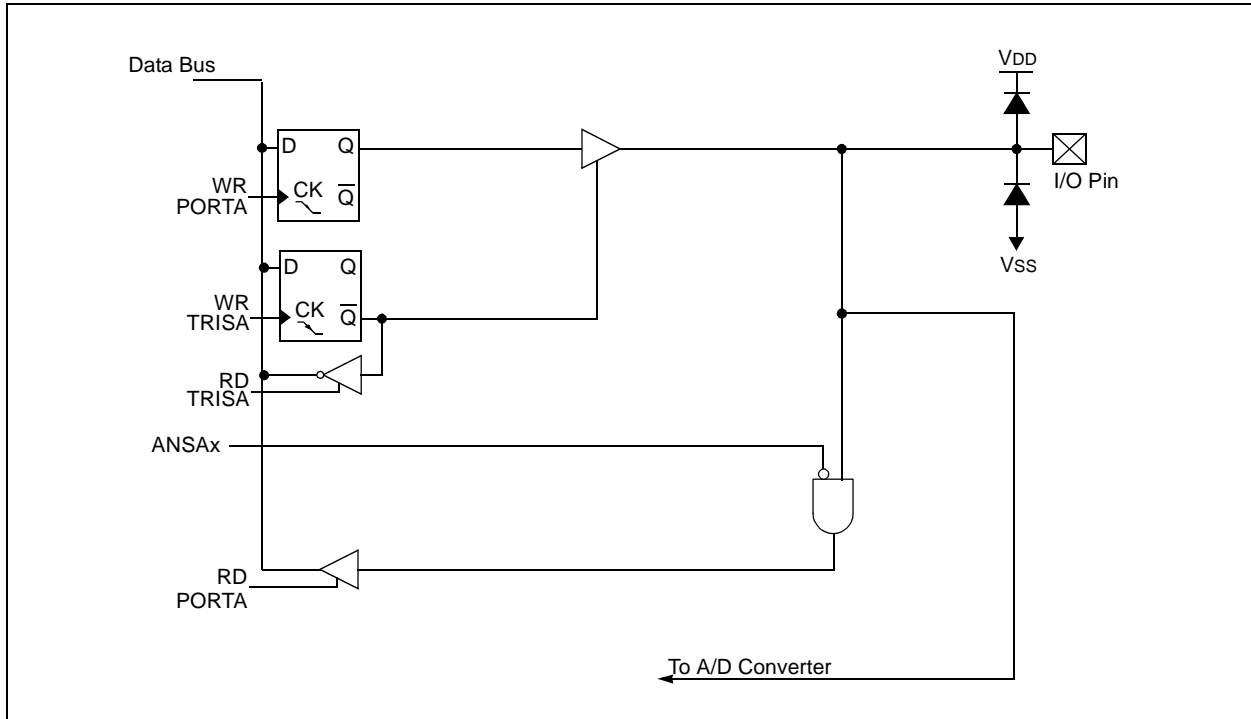
**TABLE 2-1: PIC16(L)F722A/723A SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Page
<b>Bank 1</b>											
80h <sup>(2)</sup>	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								xxxx xxxx	22,30
81h	OPTION_REG	RPBPU	INTEDG	TOCS	T0SE	PSA	PS2	PS1	PS0	1111 1111	19,30
82h <sup>(2)</sup>	PCL	Program Counter (PC) Least Significant Byte								0000 0000	21,30
83h <sup>(2)</sup>	STATUS	IRP	RP1	RP0	T0	PD	Z	DC	C	0001 1xxx	18,30
84h <sup>(2)</sup>	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	22,30
85h	TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	1111 1111	43,30
86h	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	1111 1111	52,30
87h	TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	1111 1111	62,30
89h	TRISE	—	—	—	—	TRISE3 <sup>(5)</sup>	—	—	—	---- 1111	69,30
8Ah <sup>(1, 2)</sup>	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	21,30
8Bh <sup>(2)</sup>	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	36,30
8Ch	PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	37,31
8Dh	PIE2	—	—	—	—	—	—	—	CCP2IE	---- --0	38,31
8Eh	PCON	—	—	—	—	—	—	POR	BOR	---- --q	20,31
8Fh	T1GCON	TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO DONE	T1GVAL	T1GSS1	T1GSS0	0000 0x00	104,31
90h	OSCCON	—	—	IRCF1	IRCF0	ICSL	ICSS	—	—	--10 qq--	73,31
91h	OSCTUNE	—	—	TUN5	TUN4	TUN3	TUN2	TUN1	TUN0	--00 0000	74,31
92h	PR2	Timer2 Period Register								1111 1111	106,31
93h	SSPADD <sup>(4)</sup>	Synchronous Serial Port (I <sup>2</sup> C mode) Address Register								0000 0000	155,31
93h	SSPMSK <sup>(3)</sup>	Synchronous Serial Port (I <sup>2</sup> C mode) Address Mask Register								1111 1111	166,31
94h	SSPSTAT	SMP	CKE	D/A	P	S	R/W	UA	BF	0000 0000	153,31
95h	WPUB	WPUB7	WPUB6	WPUB5	WPUB4	WPUB3	WPUB2	WPUB1	WPUB0	1111 1111	52,31
96h	IOCB	IOCB7	IOCB6	IOCB5	IOCB4	IOCB3	IOCB2	IOCB1	IOCB0	0000 0000	53,31
97h	—	Unimplemented								—	—
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	133,31
99h	SPBRG	BRG7	BRG6	BRG5	BRG4	BRG3	BRG2	BRG1	BRG0	0000 0000	135,31
9Ah	—	Unimplemented								—	—
9Bh	—	Unimplemented								—	—
9Ch	APFCON	—	—	—	—	—	—	SSSEL	CCP2SEL	---- --00	42,31
9Dh	FVRCON	FVRDY	FVREN	—	—	—	—	ADFVR1	ADFVR0	q0-- --00	90,31
9Eh	—	Unimplemented								—	—
9Fh	ADCON1	—	ADCS2	ADCS1	ADCS0	—	—	ADREF1	ADREF0	0000 --00	86,31

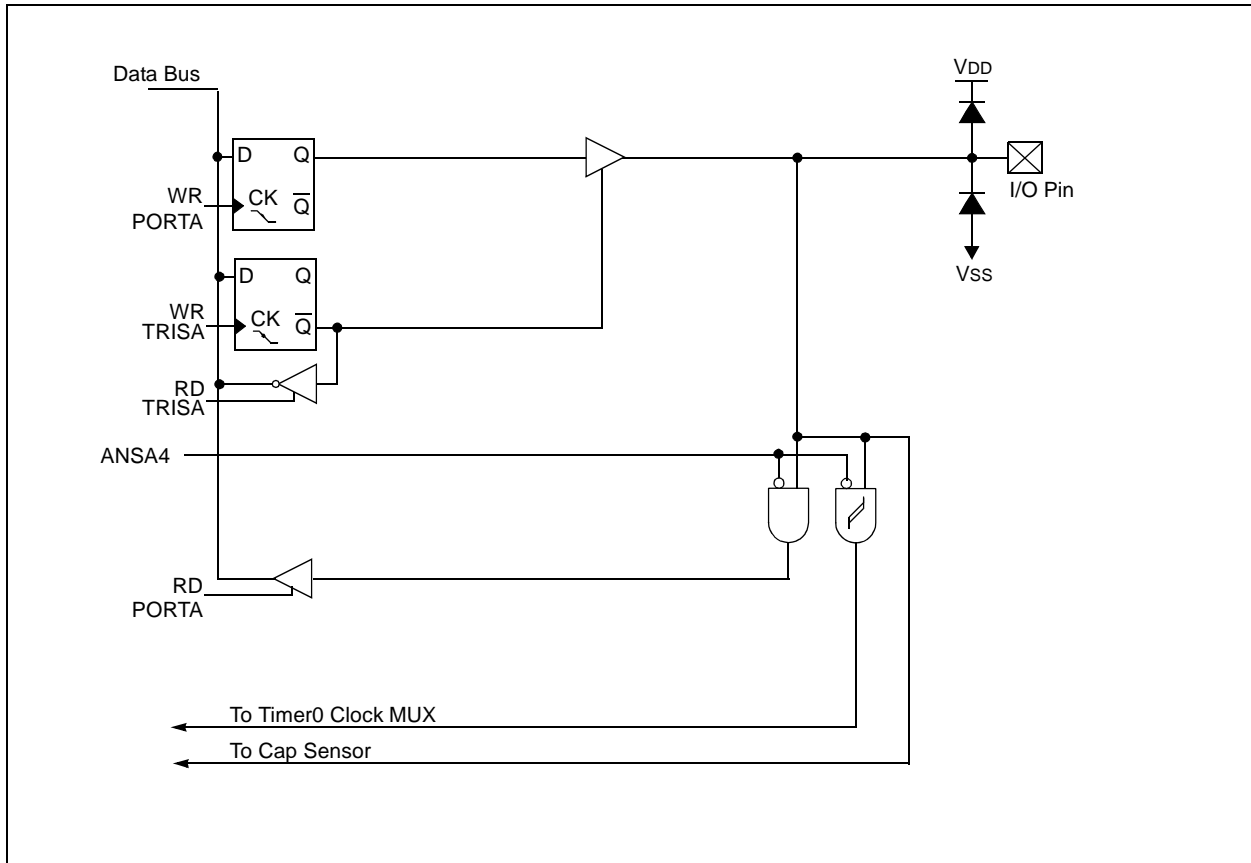
**Legend:** x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, read as '0', r = reserved.  
Shaded locations are unimplemented, read as '0'.

- 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: Accessible only when SSPM<3:0> ≠ 1001.
- 5: This bit is always '1' as RE3 is input-only.

**FIGURE 6-2: RA<3:1> BLOCK DIAGRAM**



**FIGURE 6-3: BLOCK DIAGRAM OF RA4**



# PIC16(L)F722A/723A

**TABLE 6-4: SUMMARY OF REGISTERS ASSOCIATED WITH PORTC**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
APFCON	—	—	—	—	—	—	SSSEL	CCP2SEL	42
CCP1CON	—	—	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	115
CCP2CON	—	—	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	115
PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	62
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	134
SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	152
SSPSTAT	SMP	CKE	D/ $\bar{A}$	P	S	R/ $\bar{W}$	UA	BF	153
T1CON	TMR1CS1	TMR1CS0	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{T1SYNC}$	—	TMR1ON	103
TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	133
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	62

**Legend:** x = unknown, u = unchanged, - = unimplemented locations read as '0'. Shaded cells are not used by PORTC.

## 8.0 DEVICE CONFIGURATION

Device configuration consists of Configuration Word 1 and Configuration Word 2 registers, code protection and device ID.

## 8.1 Configuration Words

There are several Configuration Word bits that allow different oscillator and memory protection options. These are implemented as Configuration Word 1 register at 2007h and Configuration Word 2 register at 2008h. These registers are only accessible during programming.

### REGISTER 8-1: CONFIG1: CONFIGURATION WORD REGISTER 1

R/P-1	R/P-1	U-1 <sup>(4)</sup>	R/P-1	R/P-1	R/P-1
DEBUG	PLLEN	—	BORV	BOREN1	BOREN0
bit 13			bit 8		

U-1 <sup>(4)</sup>	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
—	CP	MCLRE	PWRT	WDTE	FOSC2	FOSC1	FOSC0
bit 7				bit 0			

<b>Legend:</b>	P = Programmable bit
R = Readable bit	W = Writable bit
-n = Value at POR	'1' = Bit is set
	'0' = Bit is cleared
	x = Bit is unknown

bit 13	<b>DEBUG:</b> In-Circuit Debugger Mode bit 1 = In-circuit debugger disabled, RB6/ICSPCLK and RB7/ICSPDAT are general purpose I/O pins 0 = In-circuit debugger enabled, RB6/ICSPCLK and RB7/ICSPDAT are dedicated to the debugger
bit 12	<b>PLLEN:</b> INTOSC PLL Enable bit 0 = INTOSC frequency is 500 kHz 1 = INTOSC frequency is 16 MHz (32x)
bit 11	<b>Unimplemented:</b> Read as '1'
bit 10	<b>BORV:</b> Brown-out Reset Voltage selection bit 0 = Brown-out Reset Voltage (VBOR) set to 2.5 V nominal 1 = Brown-out Reset Voltage (VBOR) set to 1.9 V nominal
bit 9-8	<b>BOREN&lt;1:0&gt;:</b> Brown-out Reset Selection bits <sup>(1)</sup> 0x = BOR disabled (preconditioned state) 10 = BOR enabled during operation and disabled in Sleep 11 = BOR enabled
bit 7	<b>Unimplemented:</b> Read as '1'
bit 6	<b>CP:</b> Code Protection bit <sup>(2)</sup> 1 = Program memory code protection is disabled 0 = Program memory code protection is enabled
bit 5	<b>MCLRE:</b> RE3/MCLR Pin Function Select bit <sup>(3)</sup> 1 = RE3/MCLR pin function is MCLR 0 = RE3/MCLR pin function is digital input, MCLR internally tied to VDD
bit 4	<b>PWRT:</b> Power-up Timer Enable bit 1 = PWRT disabled 0 = PWRT enabled
bit 3	<b>WDTE:</b> Watchdog Timer Enable bit 1 = WDT enabled 0 = WDT disabled

- Note**
- 1: Enabling Brown-out Reset does not automatically enable Power-up Timer.
  - 2: The entire program memory will be erased when the code protection is turned off.
  - 3: When MCLR is asserted in INTOSC or RC mode, the internal clock oscillator is disabled.
  - 4: MPLAB® X IDE masks unimplemented Configuration bits to '0'.

## 9.1 ADC Configuration

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

- Port configuration
- Channel selection
- ADC voltage reference selection
- ADC conversion clock source
- Interrupt control
- Results formatting

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.

### 9.1.1 PORT CONFIGURATION

The ADC can be used to convert both analog and digital signals. When converting analog signals, the I/O pin 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

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 ADC VOLTAGE REFERENCE

The ADREF bits of the ADCON1 register provides control of the positive voltage reference. The positive voltage reference can be either VDD, an external voltage source or the internal Fixed Voltage Reference. The negative voltage reference is always connected to the ground reference. See **Section 10.0 “Fixed Voltage Reference”** for more details.

### 9.1.4 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.

## 12.3 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 prescale counter. The prescale counter is not directly readable or writable; however, the prescaler counter is cleared upon a write to TMR1H or TMR1L.

## 12.4 Timer1 Oscillator

A dedicated low-power 32.768 kHz oscillator circuit is built-in between pins T1OSI (input) and T1OSO (amplifier output). This internal circuit is to be used in conjunction with an external 32.768 kHz crystal.

The oscillator circuit is enabled by setting the T1OSCEN bit of the T1CON register. The oscillator will continue to run during Sleep.

<b>Note:</b> The oscillator requires a start-up and stabilization time before use. Thus, T1OSCEN should be set and a suitable delay observed prior to enabling Timer1.
--

## 12.5 Timer1 Operation in Asynchronous Counter Mode

If control bit  $\overline{T1SYNC}$  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 12.5.1 “Reading and Writing Timer1 in Asynchronous Counter Mode”**).

<b>Note:</b> 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.
--

### 12.5.1 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.

## 14.5 Software Control

The software portion of the capacitive sensing module is required to determine the change in frequency of the capacitive sensing oscillator. This is accomplished by the following:

- Setting a fixed-time base to acquire counts on Timer0 or Timer1
- Establishing the nominal frequency for the capacitive sensing oscillator
- Establishing the reduced frequency for the capacitive sensing oscillator due to an additional capacitive load
- Set the frequency threshold

### 14.5.1 NOMINAL FREQUENCY (NO CAPACITIVE LOAD)

To determine the nominal frequency of the capacitive sensing oscillator:

- Remove any extra capacitive load on the selected CPSx pin
- At the start of the fixed-time base, clear the timer resource
- At the end of the fixed-time base, save the value in the timer resource

The value of the timer resource is the number of oscillations of the capacitive sensing oscillator for the given time base. The frequency of the capacitive sensing oscillator is equal to the number of counts on in the timer divided by the period of the fixed-time base.

### 14.5.2 REDUCED FREQUENCY (ADDITIONAL CAPACITIVE LOAD)

The extra capacitive load will cause the frequency of the capacitive sensing oscillator to decrease. To determine the reduced frequency of the capacitive sensing oscillator:

- Add a typical capacitive load on the selected CPSx pin
- Use the same fixed-time base as the nominal frequency measurement
- At the start of the fixed-time base, clear the timer resource
- At the end of the fixed-time base, save the value in the timer resource

The value of the timer resource is the number of oscillations of the capacitive sensing oscillator with an additional capacitive load. The frequency of the capacitive sensing oscillator is equal to the number of counts on in the timer divided by the period of the fixed-time base. This frequency should be less than the value obtained during the nominal frequency measurement.

### 14.5.3 FREQUENCY THRESHOLD

The frequency threshold should be placed midway between the value of nominal frequency and the reduced frequency of the capacitive sensing oscillator. Refer to Application Note *AN1103, Software Handling for Capacitive Sensing* (DS01103) for more detailed information the software required for capacitive sensing module.

**Note:** For more information on general capacitive sensing refer to Application Notes:

- *AN1101, Introduction to Capacitive Sensing* (DS01101)
- *AN1102, Layout and Physical Design Guidelines for Capacitive Sensing* (DS01102)



# PIC16(L)F722A/723A

**TABLE 16-2: REGISTERS ASSOCIATED WITH ASYNCHRONOUS 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
SPBRG	BRG7	BRG6	BRG5	BRG4	BRG3	BRG2	BRG1	BRG0	0000 0000	0000 0000
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 asynchronous reception.

# PIC16(L)F722A/723A

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

BAUD RATE	SYNC = 0, BRGH = 0											
	Fosc = 20.000 MHz			Fosc = 18.432 MHz			Fosc = 16.0000 MHz			Fosc = 11.0592 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	—	—	—	—	—	—	—	—	—	—	—	—
1200	1221	1.73	255	1200	0.00	239	1201	0.08	207	1200	0.00	143
2400	2404	0.16	129	2400	0.00	119	2403	0.16	103	2400	0.00	71
9600	9470	-1.36	32	9600	0.00	29	9615	0.16	25	9600	0.00	17
10417	10417	0.00	29	10286	-1.26	27	10416	-0.01	23	10165	-2.42	16
19.2k	19.53k	1.73	15	19.20k	0.00	14	19.23k	0.16	12	19.20k	0.00	8
57.6k	—	—	—	57.60k	0.00	7	—	—	—	57.60k	0.00	2
115.2k	—	—	—	—	—	—	—	—	—	—	—	—

BAUD RATE	SYNC = 0, BRGH = 0											
	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	300	0.00	191	300	0.16	51
1200	1202	0.16	103	1202	0.16	51	1200	0.00	47	1202	0.16	12
2400	2404	0.16	51	2404	0.16	25	2400	0.00	23	—	—	—
9600	9615	0.16	12	—	—	—	9600	0.00	5	—	—	—
10417	10417	0.00	11	10417	0.00	5	—	—	—	—	—	—
19.2k	—	—	—	—	—	—	19.20k	0.00	2	—	—	—
57.6k	—	—	—	—	—	—	57.60k	0.00	0	—	—	—
115.2k	—	—	—	—	—	—	—	—	—	—	—	—

BAUD RATE	SYNC = 0, BRGH = 1											
	Fosc = 20.000 MHz			Fosc = 18.432 MHz			Fosc = 16.0000 MHz			Fosc = 11.0592 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	—	—	—	—	—	—	—	—	—	—	—	—
1200	—	—	—	—	—	—	—	—	—	—	—	—
2400	—	—	—	—	—	—	—	—	—	—	—	—
9600	9615	0.16	129	9600	0.00	119	9615	0.16	103	9600	0.00	71
10417	10417	0.00	119	10378	-0.37	110	10417	0.00	95	10473	0.53	65
19.2k	19.23k	0.16	64	19.20k	0.00	59	19.23k	0.16	51	19.20k	0.00	35
57.6k	56.82k	-1.36	21	57.60k	0.00	19	58.8k	2.12	16	57.60k	0.00	11
115.2k	113.64k	-1.36	10	115.2k	0.00	9	—	—	—	115.2k	0.00	5

## 16.3 AUSART Synchronous Mode

Synchronous serial communications are typically used in systems with a single master and one or more slaves. The master device contains the necessary circuitry for baud rate generation and supplies the clock for all devices in the system. Slave devices can take advantage of the master clock by eliminating the internal clock generation circuitry.

There are two signal lines in Synchronous mode: a bidirectional data line and a clock line. Slaves use the external clock supplied by the master to shift the serial data into and out of their respective receive and transmit shift registers. Since the data line is bidirectional, synchronous operation is half-duplex only. Half-duplex refers to the fact that master and slave devices can receive and transmit data but not both simultaneously. The AUSART can operate as either a master or slave device.

Start and Stop bits are not used in synchronous transmissions.

### 16.3.1 SYNCHRONOUS MASTER MODE

The following bits are used to configure the AUSART for Synchronous Master operation:

- SYNC = 1
- CSRC = 1
- SREN = 0 (for transmit); SREN = 1 (for receive)
- CREN = 0 (for transmit); CREN = 1 (for receive)
- SPEN = 1

Setting the SYNC bit of the TXSTA register configures the device for synchronous operation. Setting the CSRC bit of the TXSTA register configures the device as a master. Clearing the SREN and CREN bits of the RCSTA register ensures that the device is in the Transmit mode, otherwise the device will be configured to receive. Setting the SPEN bit of the RCSTA register enables the AUSART.

#### 16.3.1.1 Master Clock

Synchronous data transfers use a separate clock line, which is synchronous with the data. A device configured as a master transmits the clock on the TX/CK line. The TX/CK pin output driver is automatically enabled when the AUSART is configured for synchronous transmit or receive operation. Serial data bits change on the leading edge to ensure they are valid at the trailing edge of each clock. One clock cycle is generated for each data bit. Only as many clock cycles are generated as there are data bits.

#### 16.3.1.2 Synchronous Master Transmission

Data is transferred out of the device on the RX/DT pin. The RX/DT and TX/CK pin output drivers are automatically enabled when the AUSART is configured for synchronous master transmit operation.

A transmission is initiated by writing a character to the TXREG register. If the TSR still contains all or part of a previous character, the new character data is held in the TXREG until the last bit of the previous character has been transmitted. 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. The transmission of the character commences immediately following the transfer of the data to the TSR from the TXREG.

Each data bit changes on the leading edge of the master clock and remains valid until the subsequent leading clock edge.

<b>Note:</b>	The TSR register is not mapped in data memory, so it is not available to the user.
--------------	--

#### 16.3.1.3 Synchronous Master Transmission Setup:

1. Initialize the SPBRG register and the BRGH bit to achieve the desired baud rate (refer to **Section 16.2 “AUSART Baud Rate Generator (BRG)”**).
2. Enable the synchronous master serial port by setting bits SYNC, SPEN and CSRC.
3. Disable Receive mode by clearing bits SREN and CREN.
4. Enable Transmit mode by setting the TXEN bit.
5. If 9-bit transmission is desired, set the TX9 bit.
6. If interrupts are desired, set the TXIE bit of the PIE1 register and the GIE and PEIE bits of the INTCON register.
7. If 9-bit transmission is selected, the ninth bit should be loaded in the TX9D bit.
8. Start transmission by loading data to the TXREG register.

## 16.3.2.3 AUSART Synchronous Slave Reception

The operation of the Synchronous Master and Slave modes is identical (**Section 16.3.1.4 “Synchronous Master Reception”**), with the following exceptions:

- Sleep
- CREN bit is always set, therefore the receiver is never Idle
- SREN bit, which is a “don’t care” in Slave mode

A character may be received while in Sleep mode by setting the CREN bit prior to entering Sleep. Once the word is received, the RSR register will transfer the data to the RCREG register. If the RCIE interrupt enable bit of the PIE1 register is set, the interrupt generated will wake the device from Sleep and execute the next instruction. If the GIE bit is also set, the program will branch to the interrupt vector.

## 16.3.2.4 Synchronous Slave Reception Setup:

1. Set the SYNC and SPEN bits and clear the CSRC bit.
2. If interrupts are desired, set the RCIE bit of the PIE1 register and the GIE and PEIE bits of the INTCON register.
3. If 9-bit reception is desired, set the RX9 bit.
4. Verify address detection is disabled by clearing the ADDEN bit of the RCSTA register.
5. Set the CREN bit to enable reception.
6. The RCIF bit of the PIR1 register will be set when reception is complete. An interrupt will be generated if the RCIE bit of the PIE1 register was set.
7. If 9-bit mode is enabled, retrieve the Most Significant bit from the RX9D bit of the RCSTA register.
8. Retrieve the eight Least Significant bits from the receive FIFO by reading the RCREG register.
9. If an overrun error occurs, clear the error by either clearing the CREN bit of the RCSTA register.

**TABLE 16-9: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE 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 slave reception.

## 18.0 PROGRAM MEMORY READ

The Flash program memory is readable during normal operation over the full VDD range of the device. To read data from program memory, five Special Function Registers (SFRs) are used:

- PMCON1
- PMDATL
- PMDATH
- PMADRL
- PMADRH

The value written to the PMADRH:PMADRL register pair determines which program memory location is read. The read operation will be initiated by setting the RD bit of the PMCON1 register. The program memory Flash controller takes two instructions to complete the read. As a consequence, after the RD bit has been set, the next two instructions will be ignored. To avoid conflict with program execution, it is recommended that the two instructions following the setting of the RD bit are NOP. When the read completes, the result is placed in the PMDATLH:PMDATL register pair. Refer to Example 18-1 for sample code.

**Note:** Code-protect does not effect the CPU from performing a read operation on the program memory. For more information, refer to **Section 8.2 “Code Protection”**

### EXAMPLE 18-1: PROGRAM MEMORY READ

Required Sequence

```
BANKSEL PMADRL ;
MOVWF MS_PROG_ADDR, W;
MOVWF PMADRH ;MS Byte of Program Address to read
MOVWF LS_PROG_ADDR, W;
MOVWF PMADRL ;LS Byte of Program Address to read
BANKSEL PMCON1 ;
BSF PMCON1, RD;Initiate Read
NOP
NOP ;Any instructions here are ignored as program
;memory is read in second cycle after BSF

BANKSEL PMDATL ;
MOVWF PMDATL, W;W = LS Byte of Program Memory Read
MOVWF LOWPMBYTE;
MOVWF PMDATH, W;W = MS Byte of Program Memory Read
MOVWF HIGHPMBYTE;
```

# PIC16(L)F722A/723A

## 20.0 IN-CIRCUIT SERIAL PROGRAMMING™ (ICSP™)

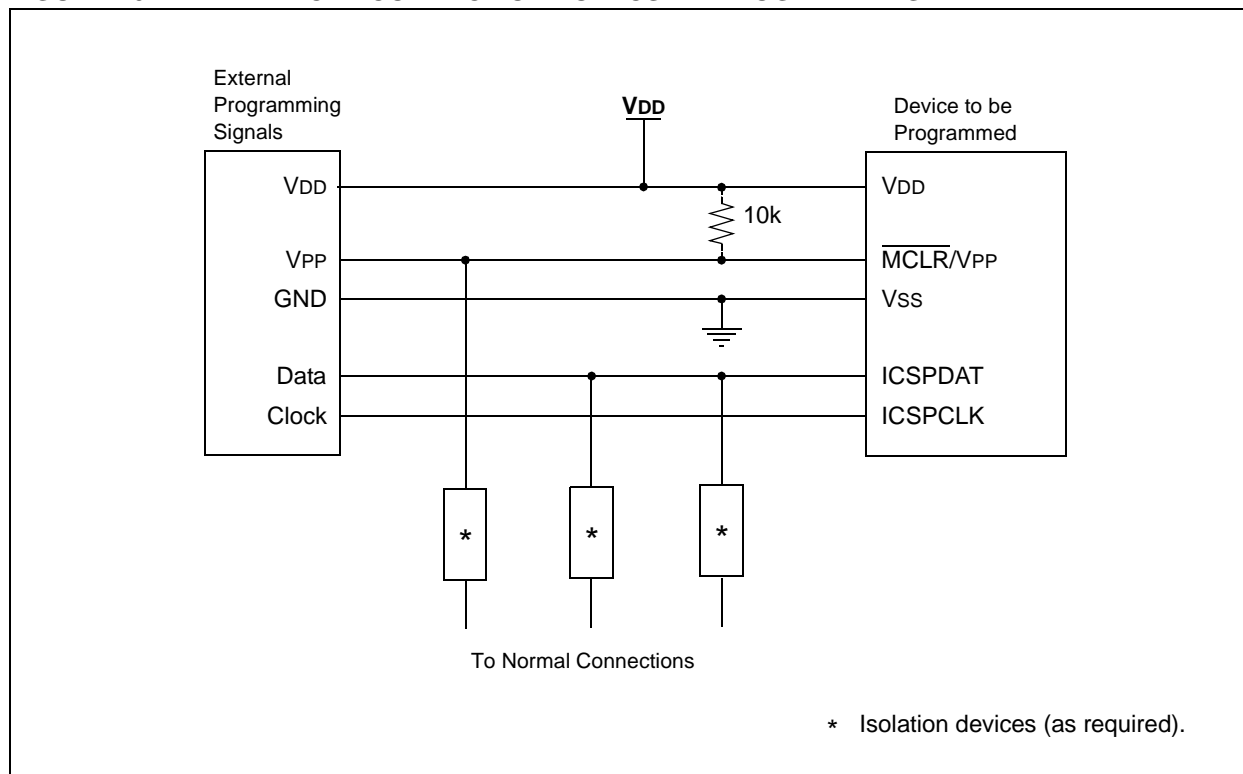
ICSP™ programming allows customers to manufacture circuit boards with unprogrammed devices. Programming can be done after the assembly process allowing the device to be programmed with the most recent firmware or a custom firmware. Five pins are needed for ICSP™ programming:

- ICSPCLK
- ICSPDAT
- MCLR/VPP
- VDD
- VSS

The device is placed into Program/Verify mode by holding the ICSPCLK and ICSPDAT pins low then raising the voltage on  $\overline{\text{MCLR/VPP}}$  from 0v to VPP. In Program/Verify mode the Program Memory, User IDs and the Configuration Words are programmed through serial communications. The ICSPDAT pin is a bidirectional I/O used for transferring the serial data and the ICSPCLK pin is the clock input. For more information on ICSP™ refer to the “PIC16(L)F72X Memory Programming Specification” (DS41332).

**Note:** The ICD 2 produces a VPP voltage greater than the maximum VPP specification of the PIC16(L)F722A/723A. When using this programmer, an external circuit, such as the AC164112 MPLAB ICD 2 VPP voltage limiter, is required to keep the VPP voltage within the device specifications.

FIGURE 20-1: TYPICAL CONNECTION FOR ICSP™ PROGRAMMING



## 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)F722A/723A

## 23.0 ELECTRICAL SPECIFICATIONS

### Absolute Maximum Ratings<sup>(†)</sup>

Ambient temperature under bias .....	-40°C to +125°C
Storage temperature .....	-65°C to +150°C
Voltage on VDD with respect to VSS, PIC16F722A/723A .....	-0.3V to +6.5V
Voltage on VCAP pin with respect to VSS, PIC16F722A/723A .....	-0.3V to +4.0V
Voltage on VDD with respect to VSS, PIC16LF722A/723A .....	-0.3V to +4.0V
Voltage on $\overline{\text{MCLR}}$ with respect to VSS .....	-0.3V to +9.0V
Voltage on all other pins with respect to VSS .....	-0.3V to (VDD + 0.3V)
Total power dissipation <sup>(1)</sup> .....	800 mW
Maximum current out of VSS pin .....	95 mA
Maximum current into VDD pin .....	70 mA
Clamp current, I <sub>K</sub> (V <sub>PIN</sub> < 0 or V <sub>PIN</sub> > VDD).....	± 20 mA
Maximum output current sunk by any I/O pin.....	25 mA
Maximum output current sourced by any I/O pin.....	25 mA
Maximum current sunk by all ports <sup>(2)</sup> , -40°C ≤ T <sub>A</sub> ≤ +85°C for industrial .....	200 mA
Maximum current sunk by all ports <sup>(2)</sup> , -40°C ≤ T <sub>A</sub> ≤ +125°C for extended .....	90 mA
Maximum current sourced by all ports <sup>(2)</sup> , 40°C ≤ T <sub>A</sub> ≤ +85°C for industrial .....	140 mA
Maximum current sourced by all ports <sup>(2)</sup> , -40°C ≤ T <sub>A</sub> ≤ +125°C for extended.....	65 mA

**Note 1:** Power dissipation is calculated as follows:  $P_{DIS} = V_{DD} \times \{I_{DD} - \sum I_{OH}\} + \sum \{(V_{DD} - V_{OH}) \times I_{OH}\} + \sum (V_{OL} \times I_{OL})$ .

† NOTICE: Stresses above those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure above maximum rating conditions for extended periods may affect device reliability.



## 23.5 Thermal Considerations

Standard Operating Conditions (unless otherwise stated)					
Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$					
Param No.	Sym.	Characteristic	Typ.	Units	Conditions
TH01	$\theta_{JA}$	Thermal Resistance Junction to Ambient	60.0	$^{\circ}\text{C/W}$	28-pin SPDIP package
			69.7	$^{\circ}\text{C/W}$	28-pin SOIC package
			71.0	$^{\circ}\text{C/W}$	28-pin SSOP package
			52.5	$^{\circ}\text{C/W}$	28-pin UQFN 4x4mm package
			30.0	$^{\circ}\text{C/W}$	28-pin QFN 6x6mm package
TH02	$\theta_{JC}$	Thermal Resistance Junction to Case	29.0	$^{\circ}\text{C/W}$	28-pin SPDIP package
			18.9	$^{\circ}\text{C/W}$	28-pin SOIC package
			24.0	$^{\circ}\text{C/W}$	28-pin SSOP package
			16.7	$^{\circ}\text{C/W}$	28-pin UQFN 4x4mm package
			5.0	$^{\circ}\text{C/W}$	28-pin QFN 6x6mm package
TH03	$T_{JMAX}$	Maximum Junction Temperature	150	$^{\circ}\text{C}$	
TH04	PD	Power Dissipation	—	W	$PD = P_{INTERNAL} + P_{I/O}$
TH05	$P_{INTERNAL}$	Internal Power Dissipation	—	W	$P_{INTERNAL} = I_{DD} \times V_{DD}^{(1)}$
TH06	$P_{I/O}$	I/O Power Dissipation	—	W	$P_{I/O} = \sum (I_{OL} \times V_{OL}) + \sum (I_{OH} \times (V_{DD} - V_{OH}))$
TH07	$P_{DER}$	Derated Power	—	W	$P_{DER} = P_{DMAX} (T_J - T_A) / \theta_{JA}^{(2)}$

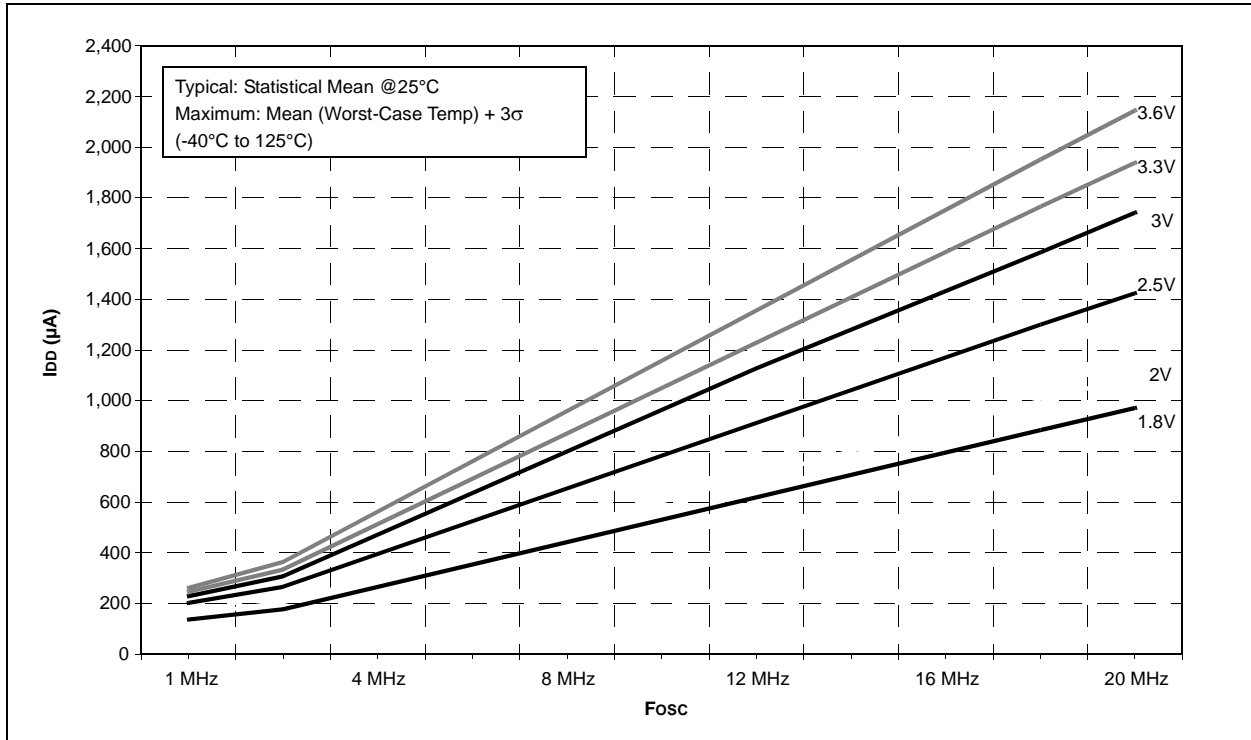
**Note 1:**  $I_{DD}$  is current to run the chip alone without driving any load on the output pins.

**2:**  $T_A$  = Ambient Temperature

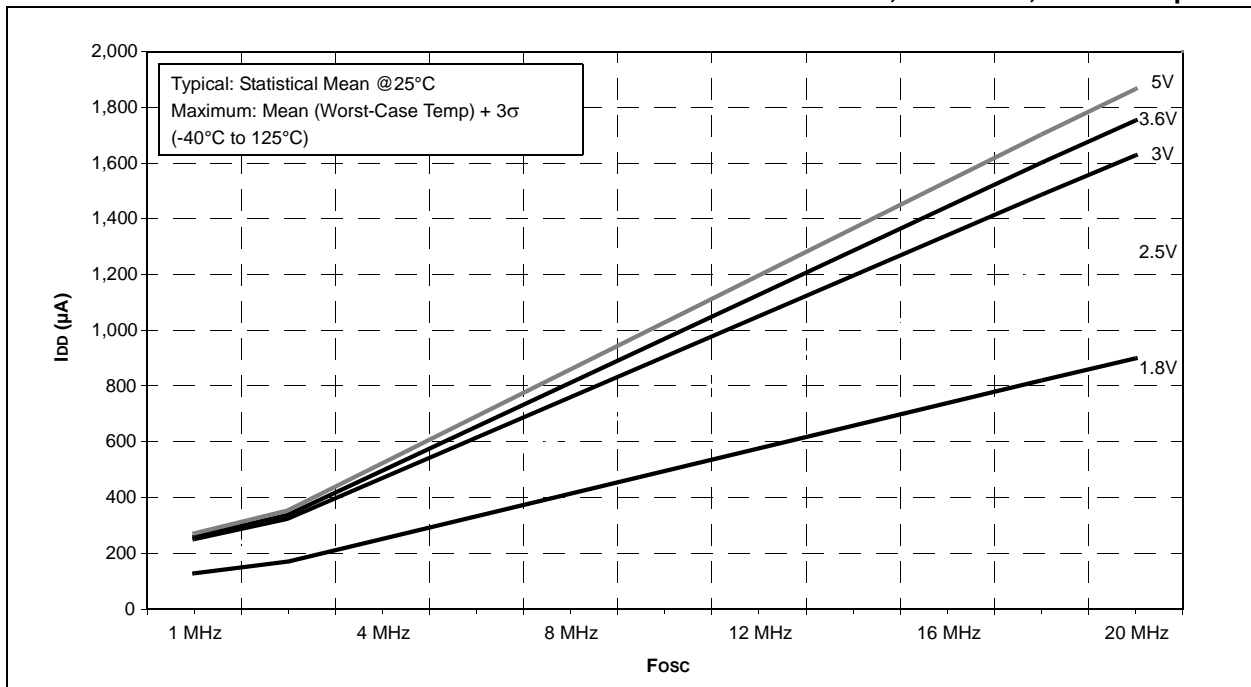
**3:**  $T_J$  = Junction Temperature

# PIC16(L)F722A/723A

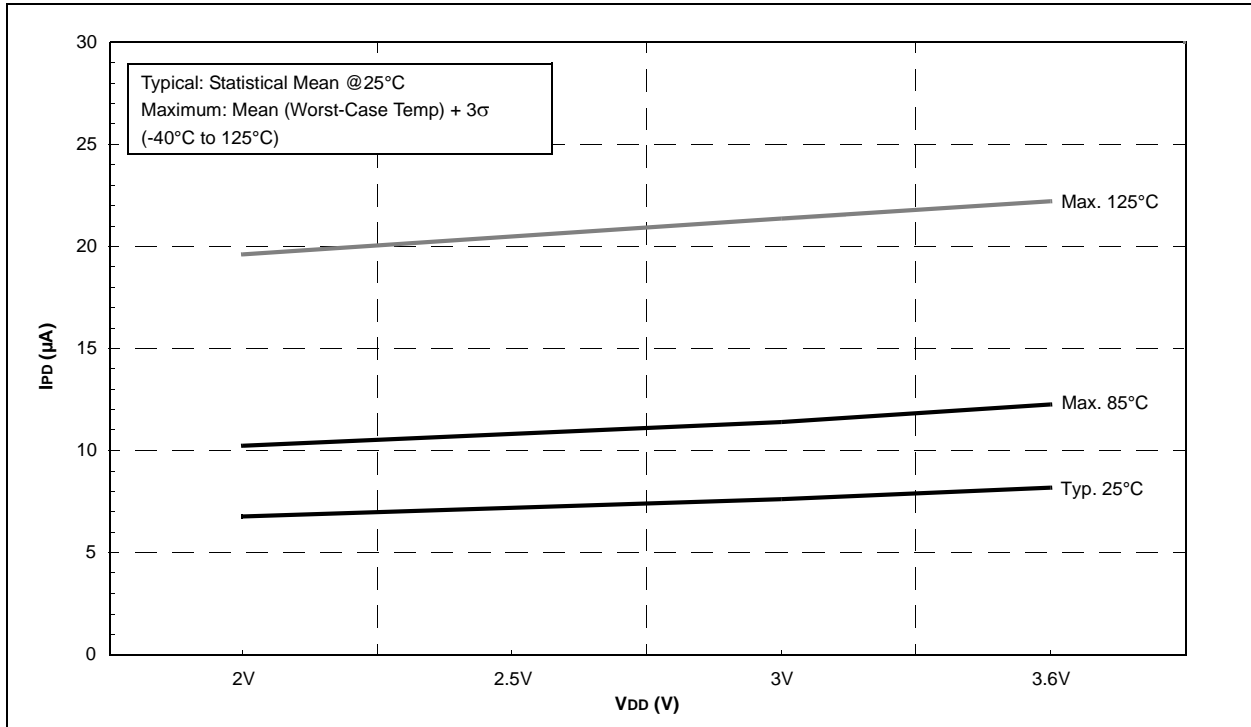
**FIGURE 24-2: PIC16LF722A/723A MAXIMUM  $I_{DD}$  vs.  $F_{OSC}$  OVER  $V_{DD}$ , EC MODE**



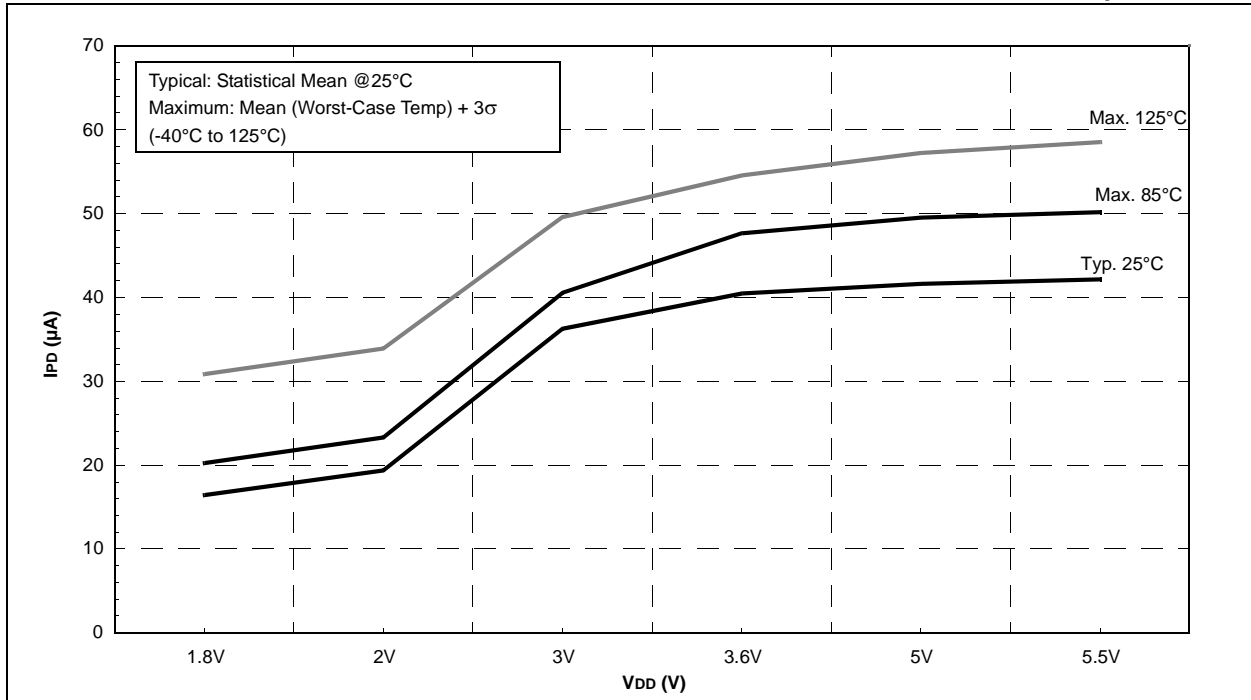
**FIGURE 24-3: PIC16F722A/723A TYPICAL  $I_{DD}$  vs.  $F_{OSC}$  OVER  $V_{DD}$ , EC MODE,  $V_{CAP} = 0.1\mu F$**



**FIGURE 24-34: PIC16LF722A/723A BOR IPD vs. VDD**



**FIGURE 24-35: PIC16F722A/723A CAP SENSE HIGH POWER IPD vs. VDD, VCAP = 0.1µF**



# PIC16(L)F722A/723A

FIGURE 24-36: PIC16LF722A/723A CAP SENSE HIGH POWER I<sub>PD</sub> vs. V<sub>DD</sub>

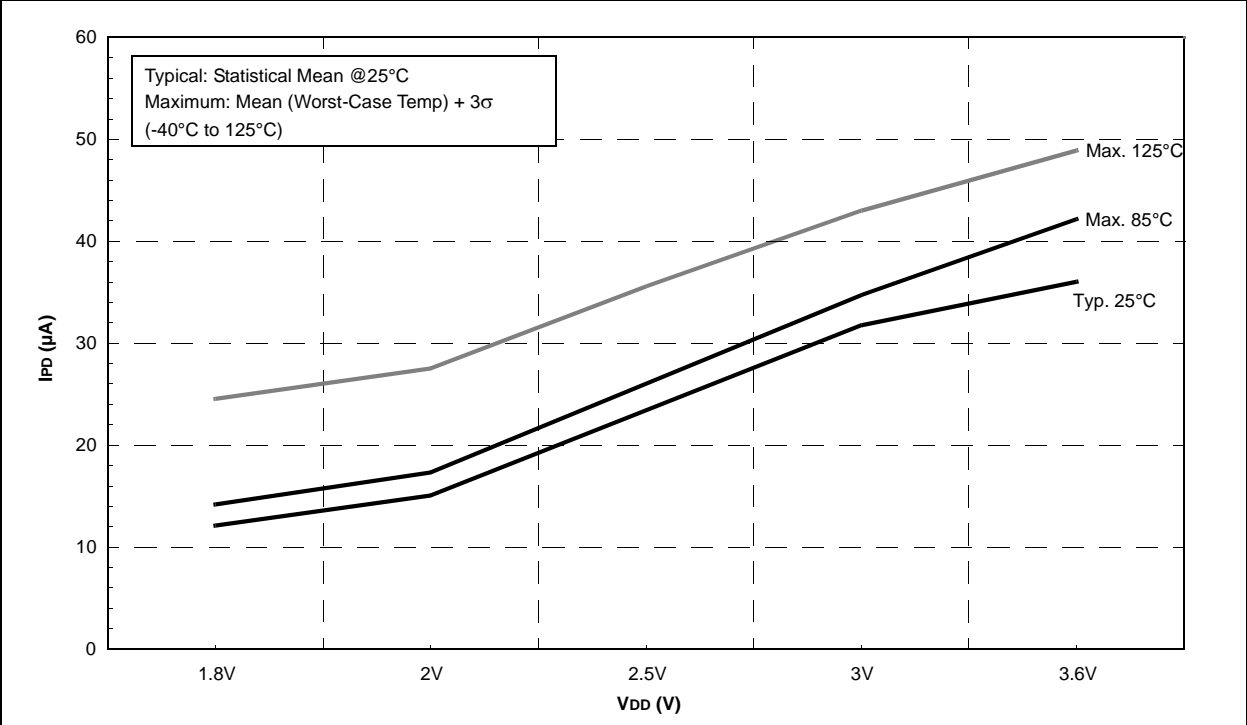
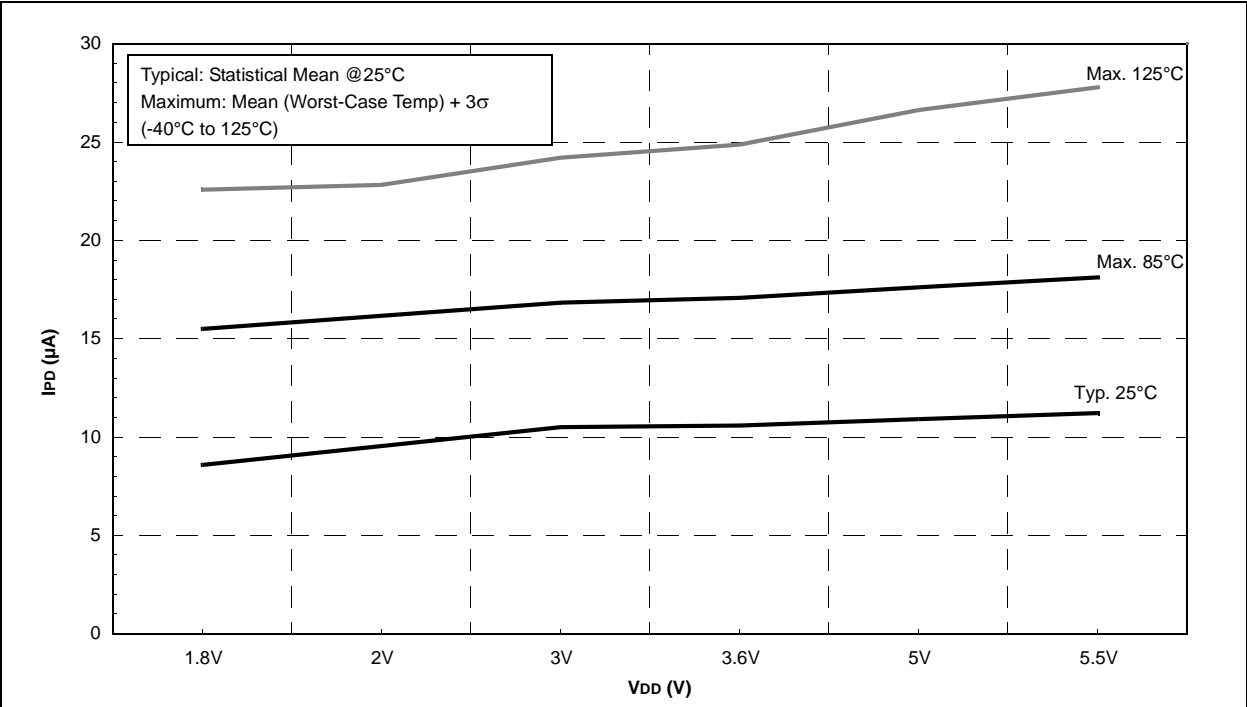


FIGURE 24-37: PIC16F722A/723A CAP SENSE MEDIUM POWER I<sub>PD</sub> vs. V<sub>DD</sub>, V<sub>CAP</sub> = 0.1μF



# PIC16(L)F722A/723A

FIGURE 24-48: PIC16LF722A/723A WDT IPD vs. VDD

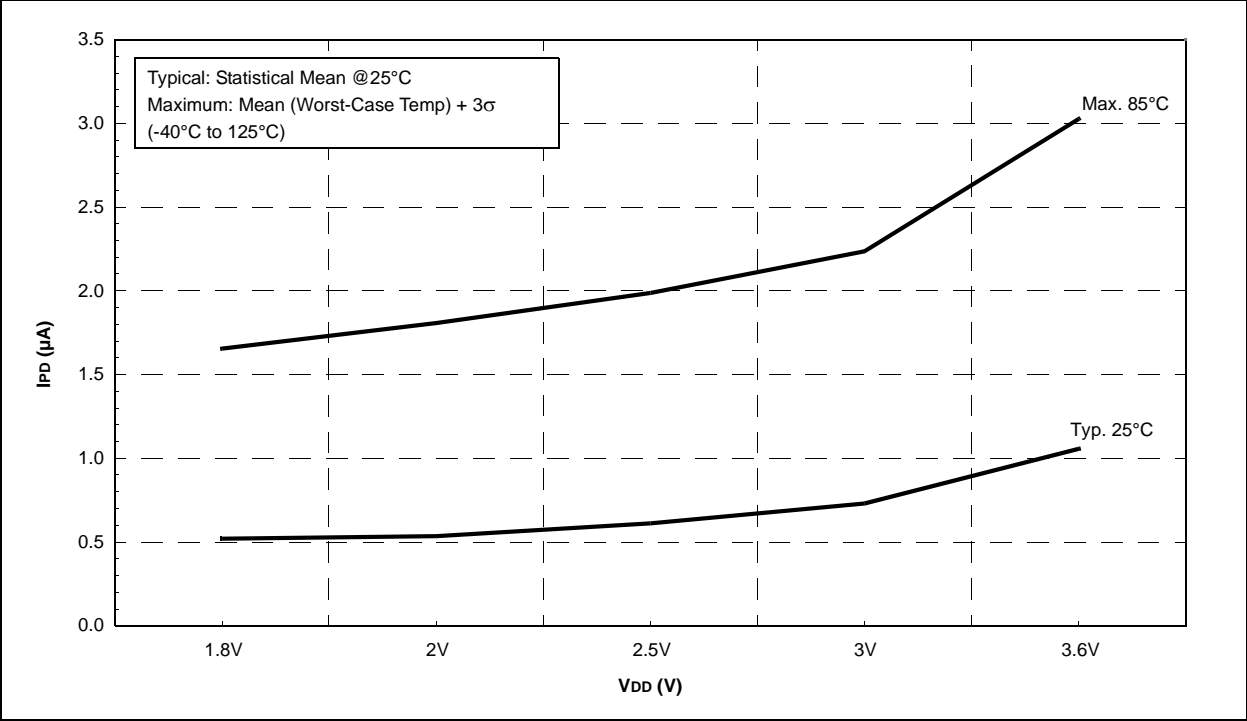


FIGURE 24-49: TTL INPUT THRESHOLD VIN vs. VDD OVER TEMPERATURE

