PIC16(L)F722A/723A
Data Sheet

28-Pin Flash Microcontrollers with nanoWatt XLP Technology

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.

Trademarks

ISBN: 978-1-61341-971-7

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
ISO/TS 16949:2009

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.

PIC16(L)F722A/723A

28-Pin Flash Microcontrollers with nanoWatt XLP Technology

Devices Included In This Data Sheet:

PIC16F722A/723A Devices:

- PIC16F722A
- PIC16F723A

PIC16LF722A/723A Devices:

- PIC16LF722A
- PIC16LF723A

High-Performance RISC CPU:

- Only 35 Instructions to Learn:
  - All single-cycle instructions except branches
- Operating Speed:
  - DC – 20 MHz oscillator/clock input
  - DC – 200 ns instruction cycle
- Up to 4K x 14 Words of Flash Program Memory
- Up to 192 Bytes of Data Memory (RAM)
- Interrupt Capability
- 8-Level Deep Hardware Stack
- Direct, Indirect and Relative Addressing modes
- Processor Read Access to Program Memory
- Pinout Compatible to other 28-pin PIC16CXXX and PIC16FXXX Microcontrollers

Special Microcontroller Features:

- Precision Internal Oscillator:
  - 16 MHz or 500 kHz operation
  - Factory calibrated to ±1%, typical
  - Software tunable
  - Software selectable ÷1, ÷2, ÷4 or ÷8 divider
- 1.8V-5.5V Operation – PIC16F722A/723A
- 1.8V-3.6V Operation – PIC16LF722A/723A
- Power-on Reset (POR), Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Brown-out Reset (BOR):
  - Selectable between two trip points
  - Disable in Sleep option
- Programmable Code Protection
- In-Circuit Serial Programming™ (ICSP™) via Two Pins
- Multiplexed Master Clear with Pull-up/Input Pin
- Industrial and Extended Temperature Range
- High-Endurance Flash Cell:
  - 1,000 write Flash endurance (typical)
  - Flash retention: > 40 years
- Power-Saving Sleep mode

Extreme Low-Power Management PIC16LF722A/723A with nanoWatt XLP:

- Sleep Mode: 20 nA
- Watchdog Timer: 500 nA
- Timer1 Oscillator: 600 nA @ 32 kHz

Analog Features:

- A/D Converter:
  - 8-bit resolution, 11 channels
  - Conversion available during Sleep
  - Selectable 1.024/2.048/4.096V voltage reference
- On-chip 3.2V Regulator (PIC16F722A/723A devices only)

Peripheral Highlights:

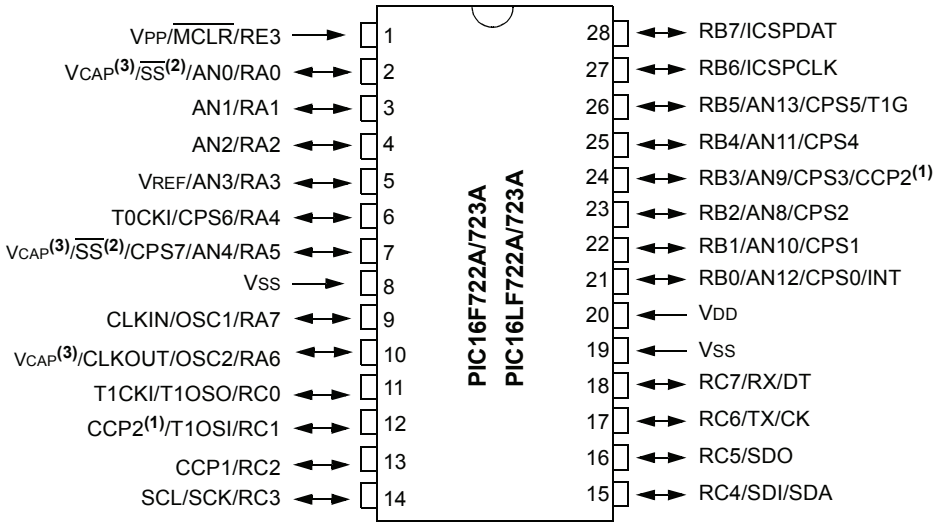
- 25 I/O Pins (1 Input-only Pin):
  - High-current source/sink for direct LED drive
  - Interrupt-on-pin change
  - Individually programmable weak pull-ups
- Timer0: 8-Bit Timer/Counter with 8-Bit Prescaler
- Enhanced Timer1:
  - Dedicated low-power 32 kHz oscillator
  - 16-bit timer/counter with prescaler
  - External Gate Input mode with toggle and single shot modes
  - Interrupt-on-gate completion
- Timer2: 8-Bit Timer/Counter with 8-Bit Period Register, Prescaler and Postscaler
- Two Capture, Compare, PWM (CCP) modules:
  - 16-bit Capture, max. resolution 12.5 ns
  - 16-bit Compare, max. resolution 200 ns
  - 10-bit PWM, max. frequency 20 kHz
- Addressable Universal Synchronous Asynchronous Receiver Transmitter (AUSART)
- Synchronous Serial Port (SSP):
  - SPI (Master/Slave)
  - I^2C™ (Slave) with Address Mask
- mTouch™ Sensing Oscillator module:
  - Up to 8 input channels

Device	Program Memory Flash (words)	SRAM (bytes)	I/Os(1)	Interrupts	8-bit A/D (ch)	AUSART	CCP	Timers 8/16-bit
PIC16F722A/ PIC16LF722A	2048	128	25	12	11	Yes	2	2/1
PIC16F723A/ PIC16LF723A	4096	192	25	12	11	Yes	2	2/1

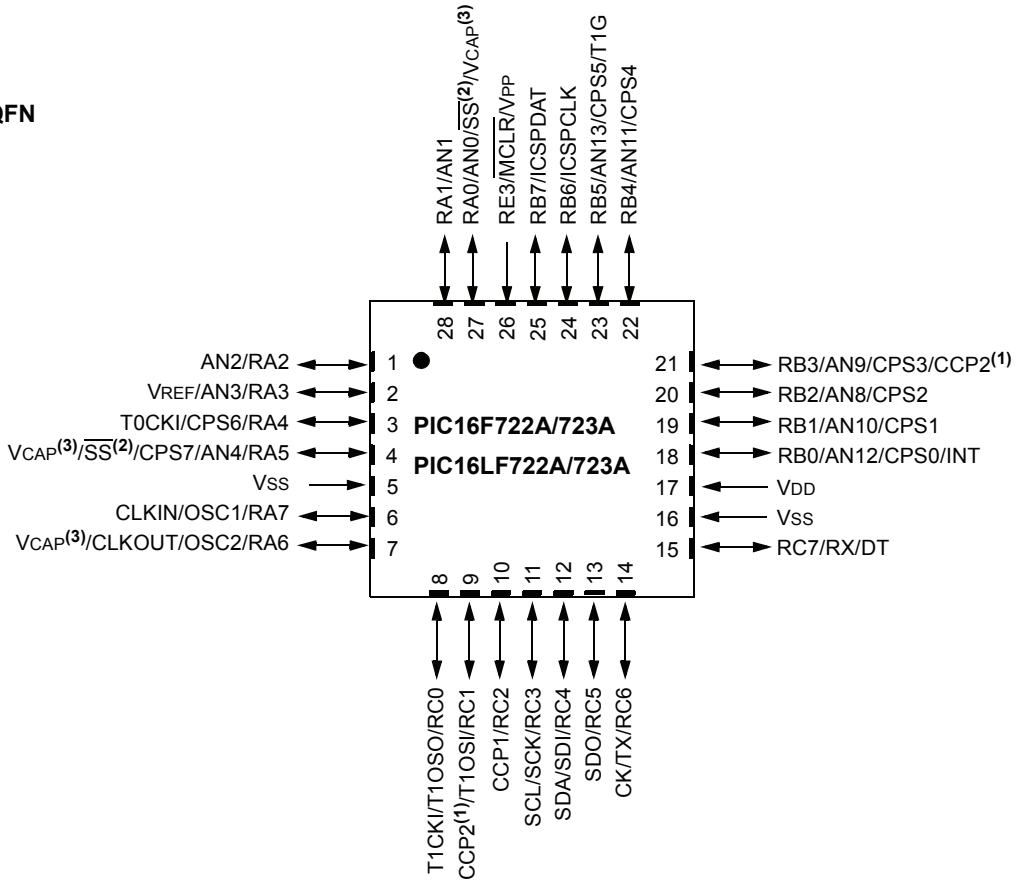
Note 1: One pin is input-only.

Pin Diagrams – 28-PIN SPDIP/SOIC/SSOP/QFN/UQFN (PIC16(L)F722A/723A)

SPDIP, SOIC, SSOP



QFN, UQFN



Note 1: CCP2 pin location may be selected as RB3 or RC1.
2: SS pin location may be selected as RA5 or RA0.
3: PIC16F722A/723A devices only.

TABLE 1: 28-PIN SPDIP/SOIC/SSOP/QFN/UQFN SUMMARY (PIC16(L)F722A/723A)

I/O	28-Pin SPDIP, SOIC, SSOP	28-Pin QFN, UQFN	A/D	Cap Sensor	Timers	CCP	AUSART	SSP	Interrupt	Pull-Up	Basic
RA0	2	27	AN0	—	—	—	—	$\overline{SS}$(3)	—	—	VCAP(4)
RA1	3	28	AN1	—	—	—	—	—	—	—	—
RA2	4	1	AN2	—	—	—	—	—	—	—	—
RA3	5	2	AN3/VREF	—	—	—	—	—	—	—	—
RA4	6	3	—	CPS6	T0CKI	—	—	—	—	—	—
RA5	7	4	AN4	CPS7	—	—	—	$\overline{SS}$(3)	—	—	VCAP(4)
RA6	10	7	—	—	—	—	—	—	—	—	OSC2/CLKOUT/VCAP(4)
RA7	9	6	—	—	—	—	—	—	—	—	OSC1/CLKIN
RB0	21	18	AN12	CPS0	—	—	—	—	IOC/INT	Y	—
RB1	22	19	AN10	CPS1	—	—	—	—	IOC	Y	—
RB2	23	20	AN8	CPS2	—	—	—	—	IOC	Y	—
RB3	24	21	AN9	CPS3	—	CCP2(2)	—	—	IOC	Y	—
RB4	25	22	AN11	CPS4	—	—	—	—	IOC	Y	—
RB5	26	23	AN13	CPS5	T1G	—	—	—	IOC	Y	—
RB6	27	24	—	—	—	—	—	—	IOC	Y	ICSPCLK/ICDCLK
RB7	28	25	—	—	—	—	—	—	IOC	Y	ICSPDAT/ICDDAT
RC0	11	8	—	—	T1OSO/T1CKI	—	—	—	—	—	—
RC1	12	9	—	—	T1OSI	CCP2(2)	—	—	—	—	—
RC2	13	10	—	—	—	CCP1	—	—	—	—	—
RC3	14	11	—	—	—	—	—	SCK/SCL	—	—	—
RC4	15	12	—	—		—	—	SDI/SDA	—	—	—
RC5	16	13	—	—	—	—	—	SDO	—	—	—
RC6	17	14	—	—	—	—	TX/CK	—	—	—	—
RC7	18	15	—	—	—	—	RX/DT	—	—	—	—
RE3	1	26	—	—	—	—	—	—	—	Y(1)	$\overline{MCLR}$/VPP
—	20	17	—	—	—	—	—	—	—	—	VDD
—	8,19	5,16	—	—	—	—	—	—	—	—	VSS

Note 1: Pull-up enabled only with external $\overline{MCLR}$ configuration.
2: RC1 is the default pin location for CCP2. RB3 may be selected by changing the CCP2SEL bit in the APFCON register.
3: RA5 is the default pin location for $\overline{SS}$. RA0 may be selected by changing the SSSEL bit in the APFCON register.
4: PIC16F722A/723A devices only.

Note: The PIC16F722A/723A devices have an internal low dropout voltage regulator. An external capacitor must be connected to one of the available VCAP pins to stabilize the regulator. For more information, see **Section 5.0 "Low Dropout (LDO) Voltage Regulator"**. The PIC16LF722A/723A devices do not have the voltage regulator and therefore no external capacitor is required.

Table of Contents

TO OUR VALUED CUSTOMERS

It is our intention to provide our valued customers with the best documentation possible to ensure successful use of your Microchip products. To this end, we will continue to improve our publications to better suit your needs. Our publications will be refined and enhanced as new volumes and updates are introduced.

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via E-mail at **docerrors@microchip.com** or fax the **Reader Response Form** in the back of this data sheet to (480) 792-4150. We welcome your feedback.

Most Current Data Sheet

To obtain the most up-to-date version of this data sheet, please register at our Worldwide Web site at:

http://www.microchip.com

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number, (e.g., DS30000A is version A of document DS30000).

Errata

An errata sheet, describing minor operational differences from the data sheet and recommended workarounds, may exist for current devices. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please check with one of the following:

- Microchip's Worldwide Web site; **http://www.microchip.com**
- Your local Microchip sales office (see last page)

When contacting a sales office, please specify which device, revision of silicon and data sheet (include literature number) you are using.

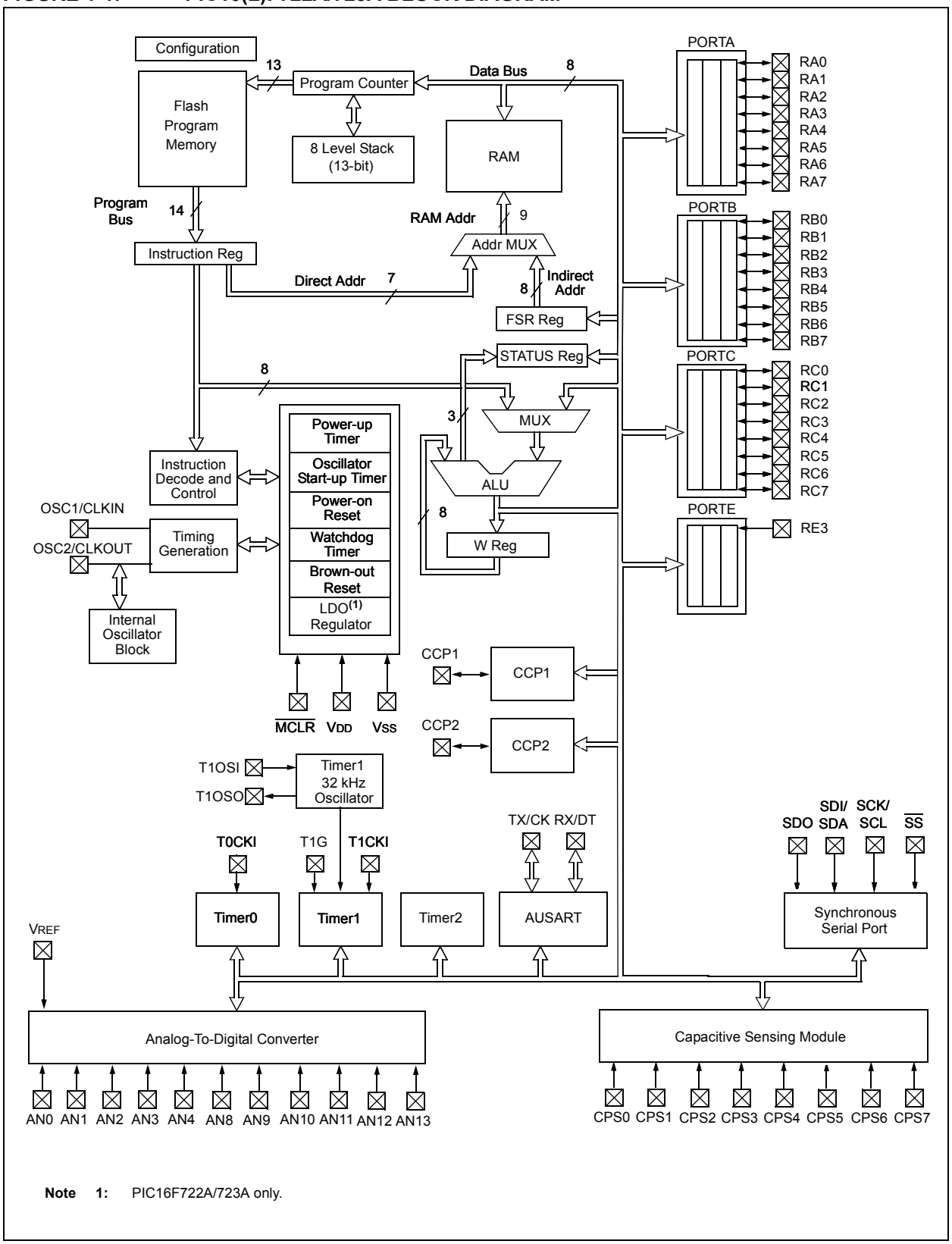
Customer Notification System

Register on our web site at **www.microchip.com** to receive the most current information on all of our products.

1.0 DEVICE OVERVIEW

The PIC16(L)F722A/723A devices are covered by this data sheet. They are available in 28-pin packages. Figure 1-1 shows a block diagram of the PIC16(L)F722A/723A devices. Table 1-1 shows the pinout descriptions.

FIGURE 1-1: PIC16(L)F722A/723A BLOCK DIAGRAM



Note 1: PIC16F722A/723A only.

TABLE 1-1: PIC16F722A/723A PINOUT DESCRIPTION

Name	Function	Input Type	Output Type	Description
RA0/AN0/$\overline{SS}$/V_{CAP}	RA0	TTL	CMOS	General purpose I/O.
	AN0	AN	—	A/D Channel 0 input.
	$\overline{SS}$	ST	—	Slave Select input.
	V_{CAP}	Power	Power	Filter capacitor for Voltage Regulator (PIC16F722A/723A only).
RA1/AN1	RA1	TTL	CMOS	General purpose I/O.
	AN1	AN	—	A/D Channel 1 input.
RA2/AN2	RA2	TTL	CMOS	General purpose I/O.
	AN2	AN	—	A/D Channel 2 input.
RA3/AN3/V_{REF}	RA3	TTL	CMOS	General purpose I/O.
	AN3	AN	—	A/D Channel 3 input.
	V_{REF}	AN	—	A/D Voltage Reference input.
RA4/CPS6/T0CKI	RA4	TTL	CMOS	General purpose I/O.
	CPS6	AN	—	Capacitive sensing input 6.
	T0CKI	ST	—	Timer0 clock input.
RA5/AN4/CPS7/$\overline{SS}$/V_{CAP}	RA5	TTL	CMOS	General purpose I/O.
	AN4	AN	—	A/D Channel 4 input.
	CPS7	AN	—	Capacitive sensing input 7.
	$\overline{SS}$	ST	—	Slave Select input.
	V_{CAP}	Power	Power	Filter capacitor for Voltage Regulator (PIC16F722A/723A only).
RA6/OSC2/CLKOUT/V_{CAP}	RA6	TTL	CMOS	General purpose I/O.
	OSC2	—	XTAL	Crystal/Resonator (LP, XT, HS modes).
	CLKOUT	—	CMOS	F_{OSC}/4 output.
	V_{CAP}	Power	Power	Filter capacitor for Voltage Regulator (PIC16F722A/723A only).
RA7/OSC1/CLKIN	RA7	TTL	CMOS	General purpose I/O.
	OSC1	XTAL	—	Crystal/Resonator (LP, XT, HS modes).
	CLKIN	CMOS	—	External clock input (EC mode).
	CLKIN	ST	—	RC oscillator connection (RC mode).
RB0/AN12/CPS0/INT	RB0	TTL	CMOS	General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up.
	AN12	AN	—	A/D Channel 12 input.
	CPS0	AN	—	Capacitive sensing input 0.
	INT	ST	—	External interrupt.
RB1/AN10/CPS1	RB1	TTL	CMOS	General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up.
	AN10	AN	—	A/D Channel 10 input.
	CPS1	AN	—	Capacitive sensing input 1.
RB2/AN8/CPS2	RB2	TTL	CMOS	General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up.
	AN8	AN	—	A/D Channel 8 input.
	CPS2	AN	—	Capacitive sensing input 2.
RB3/AN9/CPS3/CCP2	RB3	TTL	CMOS	General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up.
	AN9	AN	—	A/D Channel 9 input.
	CPS3	AN	—	Capacitive sensing input 3.
	CCP2	ST	CMOS	Capture/Compare/PWM2.

Legend: AN = Analog input or output CMOS = CMOS compatible input or output OD = Open Drain
TTL = TTL compatible input ST = Schmitt Trigger input with CMOS levels I^2C™ = Schmitt Trigger input with I^2C
HV = High Voltage XTAL = Crystal levels

TABLE 1-1: PIC16F722A/723A PINOUT DESCRIPTION (CONTINUED)

Name	Function	Input Type	Output Type	Description
RB4/AN11/CPS4	RB4	TTL	CMOS	General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up.
	AN11	AN	—	A/D Channel 11 input.
	CPS4	AN	—	Capacitive sensing input 4.
RB5/AN13/CPS5/T1G	RB5	TTL	CMOS	General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up.
	AN13	AN	—	A/D Channel 13 input.
	CPS5	AN	—	Capacitive sensing input 5.
	T1G	ST	—	Timer1 gate input.
RB6/ICSPCLK/ICDCLK	RB6	TTL	CMOS	General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up.
	ICSPCLK	ST	—	Serial Programming Clock.
	ICDCLK	ST	—	In-Circuit Debug Clock.
RB7/ICSPDAT/ICDDAT	RB7	TTL	CMOS	General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up.
	ICSPDAT	ST	CMOS	ICSP™ Data I/O.
	ICDDAT	ST	—	In-Circuit Data I/O.
RC0/T1OSO/T1CKI	RC0	ST	CMOS	General purpose I/O.
	T1OSO	XTAL	XTAL	Timer1 oscillator connection.
	T1CKI	ST	—	Timer1 clock input.
RC1/T1OSI/CCP2	RC1	ST	CMOS	General purpose I/O.
	T1OSI	XTAL	XTAL	Timer1 oscillator connection.
	CCP2	ST	CMOS	Capture/Compare/PWM2.
RC2/CCP1	RC2	ST	CMOS	General purpose I/O.
	CCP1	ST	CMOS	Capture/Compare/PWM1.
RC3/SCK/SCL	RC3	ST	CMOS	General purpose I/O.
	SCK	ST	CMOS	SPI clock.
	SCL	I^2C™	OD	I^2C™ clock.
RC4/SDI/SDA	RC4	ST	CMOS	General purpose I/O.
	SDI	ST	—	SPI data input.
	SDA	I^2C™	OD	I^2C™ data input/output.
RC5/SDO	RC5	ST	CMOS	General purpose I/O.
	SDO	—	CMOS	SPI data output.
RC6/TX/CK	RC6	ST	CMOS	General purpose I/O.
	TX	—	CMOS	USART asynchronous transmit.
	CK	ST	CMOS	USART synchronous clock.
RC7/RX/DT	RC7	ST	CMOS	General purpose I/O.
	RX	ST	—	USART asynchronous input.
	DT	ST	CMOS	USART synchronous data.
RE3/$\overline{MCLR}$/V_{PP}	RE3	TTL	—	General purpose input.
	$\overline{MCLR}$	ST	—	Master Clear with internal pull-up.
	V_{PP}	HV	—	Programming voltage.
V_{DD}	V_{DD}	Power	—	Positive supply.
V_{SS}	V_{SS}	Power	—	Ground reference.

Legend: AN = Analog input or output CMOS = CMOS compatible input or output OD = Open Drain
TTL = TTL compatible input ST = Schmitt Trigger input with CMOS levels I^2C™ = Schmitt Trigger input with I^2C
HV = High Voltage XTAL = Crystal levels

Note: The PIC16F722A/723A devices have an internal low dropout voltage regulator. An external capacitor must be connected to one of the available VCAP pins to stabilize the regulator. For more information, see **Section 5.0 "Low Dropout (LDO) Voltage Regulator"**. The PIC16LF722A/723A devices do not have the voltage regulator and therefore no external capacitor is required.

NOTES:

2.0 MEMORY ORGANIZATION

2.1 Program Memory Organization

The PIC16(L)F722A/723A has a 13-bit program counter capable of addressing a 2K x 14 program memory space for the PIC16(L)F722A (0000h-07FFh) and a 4K x 14 program memory space for the PIC16(L)F723A (0000h-0FFFh). Accessing a location above the memory boundaries for the PIC16(L)F722A will cause a wrap-around within the first 2K x 14 program memory space. Accessing a location above the memory boundaries for the PIC16(L)F723A will cause a wrap-around within the first 4K x 14 program memory space. The Reset vector is at 0000h and the interrupt vector is at 0004h.

FIGURE 2-1: PROGRAM MEMORY MAP AND STACK FOR THE PIC16(L)F722A

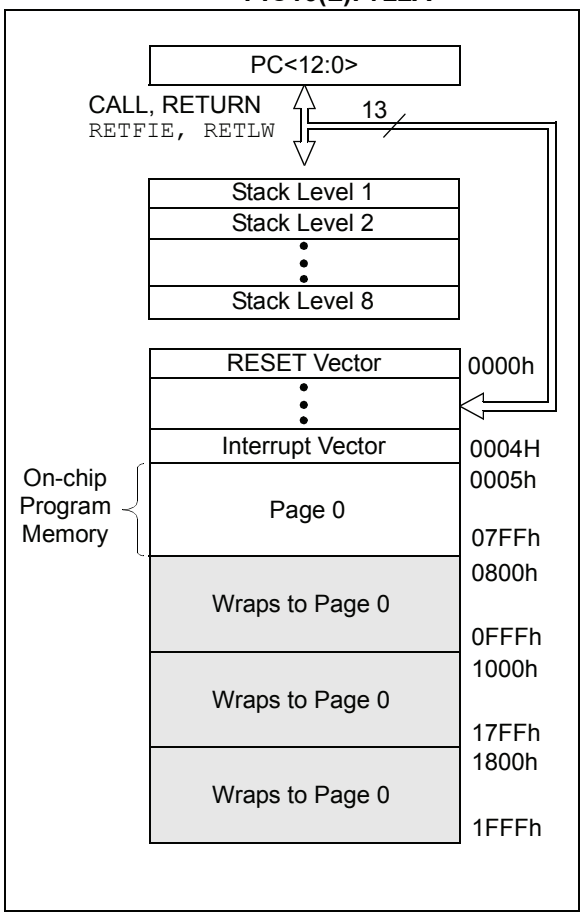
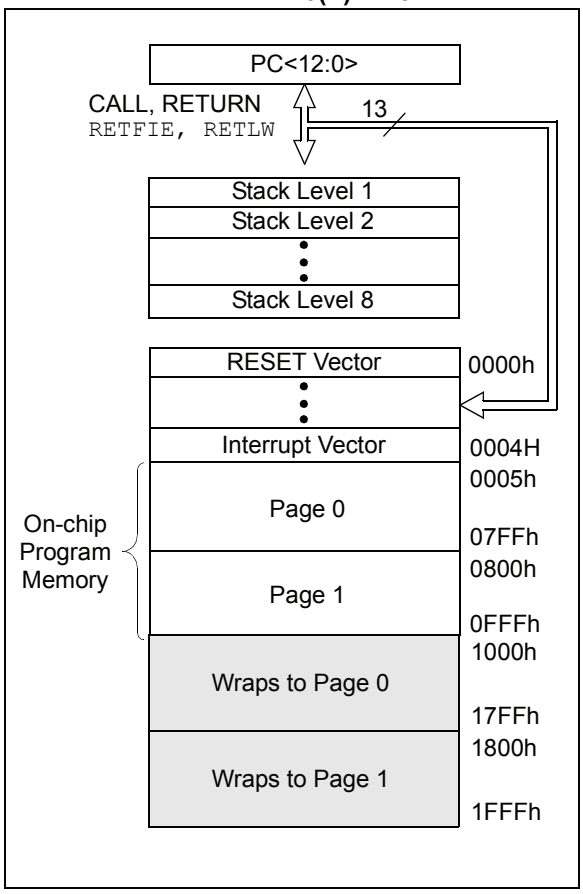


FIGURE 2-2: PROGRAM MEMORY MAP AND STACK FOR THE PIC16(L)F723A



2.2 Data Memory Organization

The data memory is partitioned into multiple banks which contain the General Purpose Registers (GPRs) and the Special Function Registers (SFRs). Bits RP0 and RP1 are bank select bits.

RP1	RP0		
0	0	→	Bank 0 is selected
0	1	→	Bank 1 is selected
1	0	→	Bank 2 is selected
1	1	→	Bank 3 is selected

Each bank extends up to 7Fh (128 bytes). The lower locations of each bank are reserved for the Special Function Registers. Above the Special Function Registers are the General Purpose Registers, implemented as static RAM. All implemented banks contain Special Function Registers. Some frequently used Special Function Registers from one bank are mirrored in another bank for code reduction and quicker access.

2.2.1 GENERAL PURPOSE REGISTER FILE

The register file is organized as 128 x 8 bits in the PIC16(L)F722A and 192 x 8 bits in the PIC16(L)F723A. Each register is accessed either directly or indirectly through the File Select Register (FSR), (Refer to **Section 2.5 "Indirect Addressing, INDF and FSR Registers"**).

2.2.2 SPECIAL FUNCTION REGISTERS

The Special Function Registers are registers used by the CPU and peripheral functions for controlling the desired operation of the device (refer to Table 2-1). These registers are static RAM.

The Special Function Registers can be classified into two sets: core and peripheral. The Special Function Registers associated with the "core" are described in this section. Those related to the operation of the peripheral features are described in the section of that peripheral feature.

FIGURE 2-3: PIC16(L)F722A SPECIAL FUNCTION REGISTERS

	Bank 0		Bank 1		Bank 2		Bank 3	File Address
	Indirect addr.(*)	00h	Indirect addr.(*)	80h	Indirect addr.(*)	100h	Indirect addr.(*)	180h
	TMR0	01h	OPTION	81h	TMR0	101h	OPTION	181h
	PCL	02h	PCL	82h	PCL	102h	PCL	182h
	STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183h
	FSR	04h	FSR	84h	FSR	104h	FSR	184h
	PORTA	05h	TRISA	85h		105h	ANSELA	185h
	PORTB	06h	TRISB	86h		106h	ANSELB	186h
	PORTC	07h	TRISC	87h		107h		187h
		08h		88h	CPSCON0	108h		188h
	PORTE	09h	TRISE	89h	CPSCON1	109h		189h
	PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah
	INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh
	PIR1	0Ch	PIE1	8Ch	PMDATL	10Ch	PMCON1	18Ch
	PIR2	0Dh	PIE2	8Dh	PMADRL	10Dh	Reserved	18Dh
	TMR1L	0Eh	PCON	8Eh	PMDATH	10Eh	Reserved	18Eh
	TMR1H	0Fh	T1GCON	8Fh	PMADRH	10Fh	Reserved	18Fh
	T1CON	10h	OSCCON	90h		110h		190h
	TMR2	11h	OSCTUNE	91h		111h		191h
	T2CON	12h	PR2	92h		112h		192h
	SSPBUF	13h	SSPADD/SSPMSK	93h		113h		193h
	SSPCON	14h	SSPSTAT	94h		114h		194h
	CCPR1L	15h	WPUB	95h		115h		195h
	CCPR1H	16h	IOCB	96h		116h		196h
	CCP1CON	17h		97h		117h		197h
	RCSTA	18h	TXSTA	98h		118h		198h
	TXREG	19h	SPBRG	99h		119h		199h
	RCREG	1Ah		9Ah		11Ah		19Ah
	CCPR2L	1Bh		9Bh		11Bh		19Bh
	CCPR2H	1Ch	APFCON	9Ch		11Ch		19Ch
	CCP2CON	1Dh	FVRCON	9Dh		11Dh		19Dh
	ADRES	1Eh		9Eh		11Eh		19Eh
	ADCON0	1Fh	ADCON1	9Fh		11Fh		19Fh
	General Purpose Register 96 Bytes	20h	General Purpose Register 32 Bytes	A0h		120h		1A0h
				BFh				
				C0h				
				EFh		16Fh		1EFh
			Accesses 70h-7Fh	F0h	Accesses 70h-7Fh	170h	Accesses 70h-7Fh	1F0h
		7Fh		FFh		17Fh		1FFh

Legend: = Unimplemented data memory locations, read as '0'.
* = Not a physical register.

FIGURE 2-4: PIC16(L)F723A SPECIAL FUNCTION REGISTERS

	File Address		File Address		File Address		File Address
Indirect addr.(*)	00h	Indirect addr.(*)	80h	Indirect addr.(*)	100h	Indirect addr.(*)	180h
TMR0	01h	OPTION	81h	TMR0	101h	OPTION	181h
PCL	02h	PCL	82h	PCL	102h	PCL	182h
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183h
FSR	04h	FSR	84h	FSR	104h	FSR	184h
PORTA	05h	TRISA	85h		105h	ANSELA	185h
PORTB	06h	TRISB	86h		106h	ANSELB	186h
PORTC	07h	TRISC	87h		107h		187h
	08h		88h	CPSCON0	108h		188h
PORTE	09h	TRISE	89h	CPSCON1	109h		189h
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh
PIR1	0Ch	PIE1	8Ch	PMDATL	10Ch	PMCON1	18Ch
PIR2	0Dh	PIE2	8Dh	PMADRL	10Dh	Reserved	18Dh
TMR1L	0Eh	PCON	8Eh	PMDATH	10Eh	Reserved	18Eh
TMR1H	0Fh	T1GCON	8Fh	PMADRH	10Fh	Reserved	18Fh
T1CON	10h	OSCCON	90h		110h		190h
TMR2	11h	OSCTUNE	91h		111h		191h
T2CON	12h	PR2	92h		112h		192h
SSPBUF	13h	SSPADD/SSPMSK	93h		113h		193h
SSPCON	14h	SSPSTAT	94h		114h		194h
CCPR1L	15h	WPUB	95h		115h		195h
CCPR1H	16h	IOCB	96h		116h		196h
CCP1CON	17h		97h		117h		197h
RCSTA	18h	TXSTA	98h		118h		198h
TXREG	19h	SPBRG	99h		119h		199h
RCREG	1Ah		9Ah		11Ah		19Ah
CCPR2L	1Bh		9Bh		11Bh		19Bh
CCPR2H	1Ch	APFCON	9Ch		11Ch		19Ch
CCP2CON	1Dh	FVRCON	9Dh		11Dh		19Dh
ADRES	1Eh		9Eh		11Eh		19Eh
ADCON0	1Fh	ADCON1	9Fh		11Fh		19Fh
General Purpose Register 96 Bytes	20h – 7Fh	General Purpose Register 80 Bytes	A0h – EFh	General Purpose Register 16 Bytes	120h – 12Fh		1A0h – 1EFh
					130h – 16Fh		
		Accesses 70h-7Fh	F0h – FFh	Accesses 70h-7Fh	170h – 17Fh	Accesses 70h-7Fh	1F0h – 1FFh
Bank 0		Bank 1		Bank 2		Bank 3	

Legend: = Unimplemented data memory locations, read as '0'.
* = Not a physical register.

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

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Page
Bank 0											
00h(2)	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								xxxx xxxx	26,34
01h	TMR0	Timer0 Module Register								xxxx xxxx	99,34
02h(2)	PCL	Program Counter (PC) Least Significant Byte								0000 0000	25,34
03h(2)	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	22,34
04h(2)	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	26,34
05h	PORTA	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0	xxxx xxxx	48,34
06h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx	57,34
07h	PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	xxxx xxxx	67,34
09h	PORTE	—	—	—	—	RE3	—	—	—	---- xxxx	74,34
0Ah(1, 2)	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	25,34
0Bh(2)	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	40,34
0Ch	PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	43,34
0Dh	PIR2	—	—	—	—	—	—	—	CCP2IF	---- ---0	44,34
0Eh	TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	108,34
0Fh	TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	108,34
10h	T1CON	TMR1CS1	TMR1CS0	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{T1SYNC}$	—	TMR1ON	0000 00-0	112,34
11h	TMR2	Timer2 Module Register								0000 0000	115,34
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	116,34
13h	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								xxxx xxxx	157,34
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	174,34
15h	CCPR1L	Capture/Compare/PWM Register (LSB)								xxxx xxxx	125,34
16h	CCPR1H	Capture/Compare/PWM Register (MSB)								xxxx xxxx	125,34
17h	CCP1CON	—	—	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	124,34
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	143,34
19h	TXREG	USART Transmit Data Register								0000 0000	142,34
1Ah	RCREG	USART Receive Data Register								0000 0000	140,34
1Bh	CCPR2L	Capture/Compare/PWM Register 2 (LSB)								xxxx xxxx	125,34
1Ch	CCPR2H	Capture/Compare/PWM Register 2 (MSB)								xxxx xxxx	125,34
1Dh	CCP2CON	—	—	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000	124,34
1Eh	ADRES	A/D Result Register								xxxx xxxx	93,34
1Fh	ADCON0	—	—	CHS3	CHS2	CHS1	CHS0	GO/$\overline{DONE}$	ADON	--00 0000	92,34

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.

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
Bank 1											
80h(2)	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								xxxx xxxx	26,34
81h	OPTION_REG	$\overline{RBPU}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	23,35
82h(2)	PCL	Program Counter (PC) Least Significant Byte								0000 0000	25,34
83h(2)	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	22,34
84h(2)	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	26,34
85h	TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	1111 1111	48,35
86h	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	1111 1111	57,35
87h	TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	1111 1111	67,35
89h	TRISE	—	—	—	—	TRISE3(5)	—	—	—	---- 1111	74,35
8Ah(1, 2)	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	25,34
8Bh(2)	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	40,34
8Ch	PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	41,35
8Dh	PIE2	—	—	—	—	—	—	—	CCP2IE	---- ---0	42,35
8Eh	PCON	—	—	—	—	—	—	$\overline{POR}$	$\overline{BOR}$	---- --qq	24,35
8Fh	T1GCON	TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO $\overline{DONE}$	T1GVAL	T1GSS1	T1GSS0	0000 0x00	113,35
90h	OSCCON	—	—	IRCF1	IRCF0	ICSL	ICSS	—	—	--10 qq--	79,35
91h	OSCTUNE	—	—	TUN5	TUN4	TUN3	TUN2	TUN1	TUN0	--00 0000	80,35
92h	PR2	Timer2 Period Register								1111 1111	115,35
93h	SSPADD(4)	Synchronous Serial Port (I^2C™ mode) Address Register								0000 0000	165,35
93h	SSPMSK(3)	Synchronous Serial Port (I^2C™ mode) Address Mask Register								1111 1111	176,35
94h	SSPSTAT	SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF	0000 0000	163,35
95h	WPUB	WPUB7	WPUB6	WPUB5	WPUB4	WPUB3	WPUB2	WPUB1	WPUB0	1111 1111	57,35
96h	IOCB	IOCB7	IOCB6	IOCB5	IOCB4	IOCB3	IOCB2	IOCB1	IOCB0	0000 0000	58,35
97h	—	Unimplemented								—	—
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	142,35
99h	SPBRG	BRG7	BRG6	BRG5	BRG4	BRG3	BRG2	BRG1	BRG0	0000 0000	144,35
9Ah	—	Unimplemented								—	—
9Bh	—	Unimplemented								—	—
9Ch	APFCON	—	—	—	—	—	—	SSSEL	CCP2SEL	---- --00	47,35
9Dh	FVRCON	FVRRDY	FVREN	—	—	—	—	ADFVR1	ADFVR0	q0-- --00	97,35
9Eh	—	Unimplemented								—	—
9Fh	ADCON1	—	ADCS2	ADCS1	ADCS0	—	—	ADREF1	ADREF0	0000 --00	93,35

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.

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
Bank 2											
100h(2)	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								xxxx xxxx	26,34
101h	TMR0	Timer0 Module Register								xxxx xxxx	99,34
102h(2)	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	25,34
103h(2)	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	22,34
104h(2)	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	26,34
105h	—	Unimplemented								—	—
106h	—	Unimplemented								—	—
107h	—	Unimplemented								—	—
108h	CPSCON0	CPSON	—	—	—	CPSRNG1	CPSRNG0	CPSOUT	T0XCS	0--- 0000	121,35
109h	CPSCON1	—	—	—	—	CPSCH3	CPSCH2	CPSCH1	CPSCH0	---- 0000	122,35
10Ah(1, 2)	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	25,34
10Bh(2)	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	40,34
10Ch	PMDATL	Program Memory Read Data Register Low Byte								xxxx xxxx	177,35
10Dh	PMADRL	Program Memory Read Address Register Low Byte								xxxx xxxx	177,35
10Eh	PMDATH	—	—	Program Memory Read Data Register High Byte						--xx xxxx	177,35
10Fh	PMADRH	—	—	—	Program Memory Read Address Register High Byte					---x xxxx	177,35
Bank 3											
180h(2)	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								xxxx xxxx	26,34
181h	OPTION_REG	$\overline{RBPU}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	23,35
182h(2)	PCL	Program Counter (PC) Least Significant Byte								0000 0000	25,34
183h(2)	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	22,34
184h(2)	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	26,34
185h	ANSELA	—	—	ANSA5	ANSA4	ANSA3	ANSA2	ANSA1	ANSA0	--11 1111	49,35
186h	ANSELB	—	—	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0	--11 1111	58,35
187h	—	Unimplemented								—	—
18Ah(1, 2)	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	25,34
18Bh(2)	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	40,34
18Ch	PMCON1	Reserved	—	—	—	—	—	—	RD	1--- ---0	178,35
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: 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.

2.2.2.1 STATUS Register

The STATUS register, shown in Register 2-1, contains:

- the arithmetic status of the ALU
- the Reset status
- the bank select bits for data memory (SRAM)

The STATUS register can be the destination for any instruction, like any other register. If the STATUS register is the destination for an instruction that affects the Z, DC or C bits, then the write to these three bits is disabled. These bits are set or cleared according to the device logic. Furthermore, the $\overline{TO}$ and $\overline{PD}$ bits are not writable. Therefore, the result of an instruction with the STATUS register as destination may be different than intended.

For example, `CLRF STATUS` will clear the upper three bits and set the Z bit. This leaves the STATUS register as '`000u u1uu`' (where `u` = unchanged).

It is recommended, therefore, that only `BCF`, `BSF`, `SWAPF` and `MOVWF` instructions are used to alter the STATUS register, because these instructions do not affect any Status bits. For other instructions not affecting any Status bits (Refer to **Section 21.0 "Instruction Set Summary"**).

Note 1: The C and DC bits operate as $\overline{\text{Borrow}}$ and $\overline{\text{Digit Borrow}}$ out bits, respectively, in subtraction.

REGISTER 2-1: STATUS: STATUS REGISTER

R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC(1)	C(1)
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 **IRP:** Register Bank Select bit (used for indirect addressing)
`1` = Bank 2, 3 (100h-1FFh)
`0` = Bank 0, 1 (00h-FFh)

bit 6-5 **RP<1:0>:** Register Bank Select bits (used for direct addressing)
`00` = Bank 0 (00h-7Fh)
`01` = Bank 1 (80h-FFh)
`10` = Bank 2 (100h-17Fh)
`11` = Bank 3 (180h-1FFh)

bit 4 **$\overline{TO}$:** Time-out bit
`1` = After power-up, `CLRWDT` instruction or `SLEEP` instruction
`0` = A WDT time-out occurred

bit 3 **$\overline{PD}$:** Power-down bit
`1` = After power-up or by the `CLRWDT` instruction
`0` = By execution of the `SLEEP` instruction

bit 2 **Z:** Zero bit
`1` = The result of an arithmetic or logic operation is zero
`0` = The result of an arithmetic or logic operation is not zero

bit 1 **DC:** Digit Carry/$\overline{\text{Digit Borrow}}$ bit (`ADDWF`, `ADDLW`, `SUBLW`, `SUBWF` instructions)(1)
`1` = A carry-out from the 4th low-order bit of the result occurred
`0` = No carry-out from the 4th low-order bit of the result

bit 0 **C:** Carry/$\overline{\text{Borrow}}$ bit(1) (`ADDWF`, `ADDLW`, `SUBLW`, `SUBWF` instructions)(1)
`1` = A carry-out from the Most Significant bit of the result occurred
`0` = No carry-out from the Most Significant bit of the result occurred

Note 1: For $\overline{\text{Borrow}}$, the polarity is reversed. A subtraction is executed by adding the two's complement of the second operand. For rotate (`RRF`, `RLF`) instructions, this bit is loaded with either the high-order or low-order bit of the source register.

2.2.2.2 OPTION register

The OPTION register, shown in Register 2-2, is a readable and writable register, which contains various control bits to configure:

- Timer0/WDT prescaler
- External RB0/INT interrupt
- Timer0
- Weak pull-ups on PORTB

Note: To achieve a 1:1 prescaler assignment for Timer0, assign the prescaler to the WDT by setting PSA bit of the OPTION register to '1'. Refer to **Section 12.3 "Timer1 Prescaler"**.

REGISTER 2-2: OPTION_REG: OPTION REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
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 **$\overline{\text{RBPU}}$:** PORTB Pull-up Enable bit
1 = PORTB pull-ups are disabled
0 = PORTB pull-ups are enabled by individual bits in the WPUB register

bit 6 **INTEDG:** Interrupt Edge Select bit
1 = Interrupt on rising edge of RB0/INT pin
0 = Interrupt on falling edge of RB0/INT pin

bit 5 **T0CS:** Timer0 Clock Source Select bit
1 = Transition on RA4/T0CKI pin
0 = Internal instruction cycle clock (FOSC/4)

bit 4 **T0SE:** Timer0 Source Edge Select bit
1 = Increment on high-to-low transition on RA4/T0CKI pin
0 = Increment on low-to-high transition on RA4/T0CKI pin

bit 3 **PSA:** Prescaler Assignment bit
1 = Prescaler is assigned to the WDT
0 = Prescaler is assigned to the Timer0 module

bit 2-0 **PS<2:0>:** Prescaler Rate Select bits

Bit Value	Timer0 Rate	WDT Rate
000	1 : 2	1 : 1
001	1 : 4	1 : 2
010	1 : 8	1 : 4
011	1 : 16	1 : 8
100	1 : 32	1 : 16
101	1 : 64	1 : 32
110	1 : 128	1 : 64
111	1 : 256	1 : 128

2.2.2.3 PCON Register

The Power Control (PCON) register contains flag bits (refer to Table 3-2) to differentiate between a:

- Power-on Reset ($\overline{POR}$)
- Brown-out Reset ($\overline{BOR}$)
- Watchdog Timer Reset (WDT)
- External $\overline{MCLR}$ Reset

The PCON register also controls the software enable of the BOR.

The PCON register bits are shown in Register 2-3.

REGISTER 2-3: PCON: POWER CONTROL REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	R/W-q	R/W-q
—	—	—	—	—	—	$\overline{POR}$	$\overline{BOR}$
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
q = Value depends on condition			

bit 7-2 **Unimplemented:** Read as '0'

bit 1 **$\overline{POR}$:** Power-on Reset Status bit
1 = No Power-on Reset occurred
0 = A Power-on Reset occurred (must be set in software after a Power-on Reset occurs)

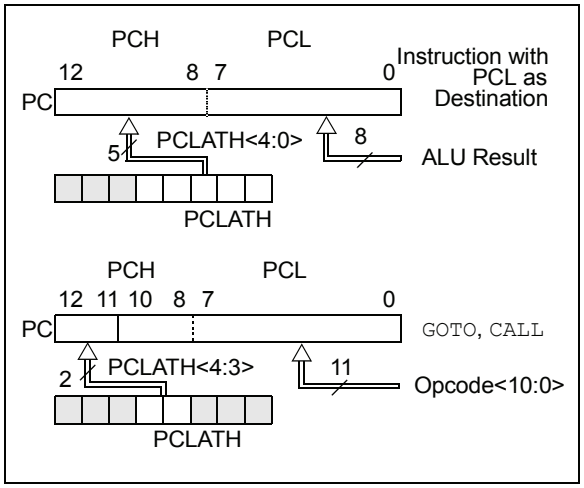
bit 0 **$\overline{BOR}$:** Brown-out Reset Status bit
1 = No Brown-out Reset occurred
0 = A Brown-out Reset occurred (must be set in software after a Power-on Reset or Brown-out Reset occurs)

Note 1: Set BOREN<1:0> = 01 in the Configuration Word register for this bit to control the $\overline{BOR}$.

2.3 PCL and PCLATH

The Program Counter (PC) is 13 bits wide. The low byte comes from the PCL register, which is a readable and writable register. The high byte (PC<12:8>) is not directly readable or writable and comes from PCLATH. On any Reset, the PC is cleared. Figure 2-5 shows the two situations for the loading of the PC. The upper example in Figure 2-5 shows how the PC is loaded on a write to PCL (PCLATH<4:0> → PCH). The lower example in Figure 2-5 shows how the PC is loaded during a CALL or GOTO instruction (PCLATH<4:3> → PCH).

FIGURE 2-5: LOADING OF PC IN DIFFERENT SITUATIONS



2.3.1 COMPUTED GOTO

A computed GOTO is accomplished by adding an offset to the program counter (ADDWF PCL). When performing a table read using a computed GOTO method, care should be exercised if the table location crosses a PCL memory boundary (each 256-byte block). Refer to Application Note AN556, *"Implementing a Table Read"* (DS00556).

2.3.2 STACK

All devices have an 8-level x 13-bit wide hardware stack (refer to Figures 2-1 and 2-2). The stack space is not part of either program or data space and the Stack Pointer is not readable or writable. The PC is PUSHed onto the stack when a CALL instruction is executed or an interrupt causes a branch. The stack is POPed in the event of a RETURN, RETLW or a RETFIE instruction execution. PCLATH is not affected by a PUSH or POP operation.

The stack operates as a circular buffer. This means that after the stack has been PUSHed eight times, the ninth PUSH overwrites the value that was stored from the first PUSH. The tenth PUSH overwrites the second PUSH (and so on).

Note 1: There are no Status bits to indicate stack overflow or stack underflow conditions.

2: There are no instructions/mnemonics called PUSH or POP. These are actions that occur from the execution of the CALL, RETURN, RETLW and RETFIE instructions or the vectoring to an interrupt address.

2.4 Program Memory Paging

All devices are capable of addressing a continuous 8K word block of program memory. The CALL and GOTO instructions provide only 11 bits of address to allow branching within any 2K program memory page. When doing a CALL or GOTO instruction, the upper 2 bits of the address are provided by PCLATH<4:3>. When doing a CALL or GOTO instruction, the user must ensure that the page select bits are programmed so that the desired program memory page is addressed. If a return from a CALL instruction (or interrupt) is executed, the entire 13-bit PC is POPed off the stack. Therefore, manipulation of the PCLATH<4:3> bits is not required for the RETURN instructions (which POPs the address from the stack).

Note: The contents of the PCLATH register are unchanged after a RETURN or RETFIE instruction is executed. The user must rewrite the contents of the PCLATH register for any subsequent subroutine calls or GOTO instructions.

Example 2-1 shows the calling of a subroutine in page 1 of the program memory. This example assumes that PCLATH is saved and restored by the Interrupt Service Routine (if interrupts are used).

EXAMPLE 2-1: CALL OF A SUBROUTINE IN PAGE 1 FROM PAGE 0

```
        ORG 500h
        PAGESEL SUB_P1  ;Select page 1
                        ;(800h-FFFh)
        CALL    SUB1_P1 ;Call subroutine in
        :               ;page 1 (800h-FFFh)
        :
        ORG     900h    ;page 1 (800h-FFFh)
SUB1_P1
        :               ;called subroutine
                        ;page 1 (800h-FFFh)
        :
        RETURN          ;return to
                        ;Call subroutine
                        ;in page 0
                        ;(000h-7FFh)
```

2.5 Indirect Addressing, INDF and FSR Registers

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

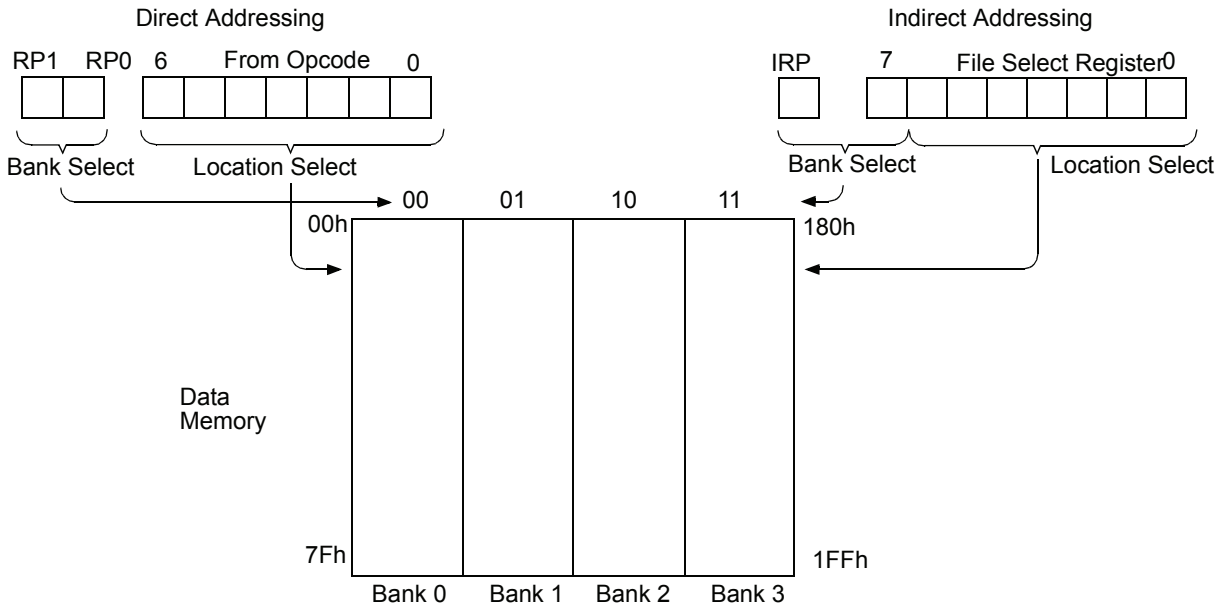
Indirect addressing is possible by using the INDF register. Any instruction using the INDF register actually accesses data pointed to by the File Select Register (FSR). Reading INDF itself indirectly will produce 00h. Writing to the INDF register indirectly results in a no operation (although Status bits may be affected). An effective 9-bit address is obtained by concatenating the 8-bit FSR register and the IRP bit of the STATUS register, as shown in Figure 2-6.

A simple program to clear RAM location 020h-02Fh using indirect addressing is shown in Example 2-2.

EXAMPLE 2-2: INDIRECT ADDRESSING

```
    MOVLW020h  ;initialize pointer
    MOVWFFSR   ;to RAM
    BANKISEL020h
NEXTCLRFINDF   ;clear INDF register
    INCFFSR    ;inc pointer
    BTFSSFSR,4 ;all done?
    GOTONEXT   ;no clear next
CONTINUE       ;yes continue
```

FIGURE 2-6: DIRECT/INDIRECT ADDRESSING



Note: For memory map detail, refer to Figures 2-3 and 2-4.

3.0 RESETS

The PIC16(L)F722A/723A differentiates between various kinds of Reset:

a) Power-on Reset (POR)
b) WDT Reset during normal operation
c) WDT Reset during Sleep
d) $\overline{\text{MCLR}}$ Reset during normal operation
e) $\overline{\text{MCLR}}$ Reset during Sleep
f) Brown-out Reset (BOR)

Some registers are not affected in any Reset condition; their status is unknown on POR and unchanged in any other Reset. Most other registers are reset to a "Reset state" on:

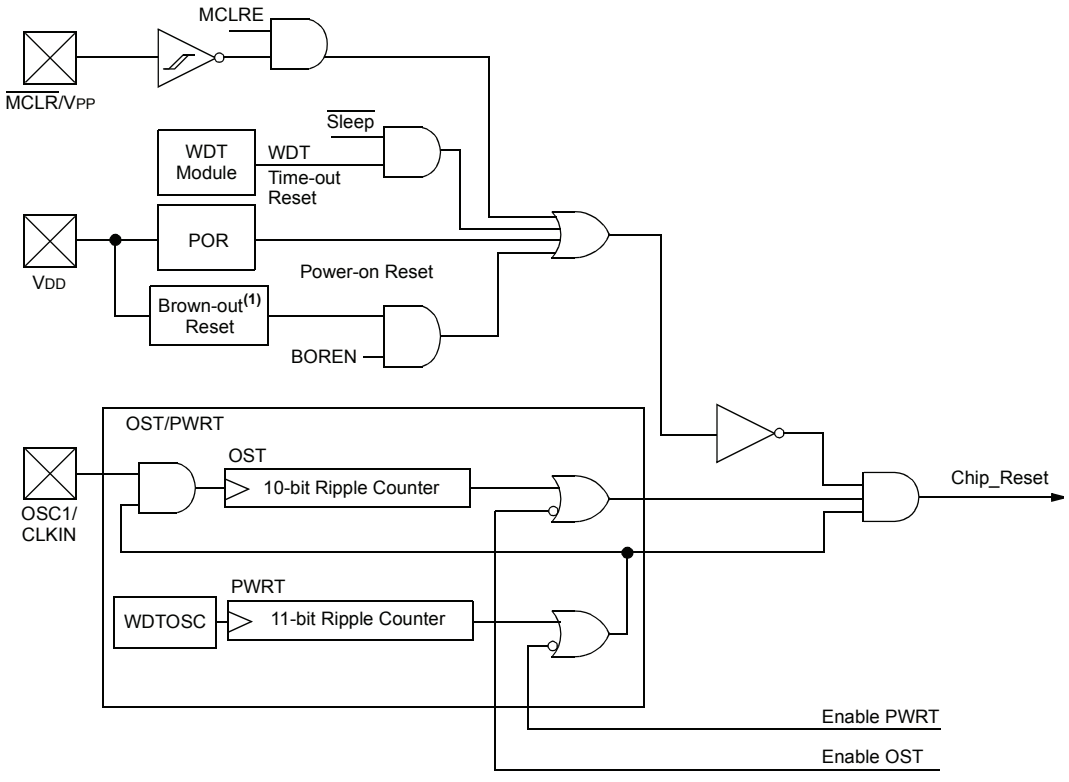
- Power-on Reset (POR)
- $\overline{\text{MCLR}}$ Reset
- $\overline{\text{MCLR}}$ Reset during Sleep
- WDT Reset
- Brown-out Reset (BOR)

Most registers are not affected by a WDT wake-up since this is viewed as the resumption of normal operation. $\overline{\text{TO}}$ and $\overline{\text{PD}}$ bits are set or cleared differently in different Reset situations, as indicated in Table 3-3. These bits are used in software to determine the nature of the Reset.

A simplified block diagram of the On-Chip Reset Circuit is shown in Figure 3-1.

The $\overline{\text{MCLR}}$ Reset path has a noise filter to detect and ignore small pulses. See **Section 23.0 "Electrical Specifications"** for pulse width specifications.

FIGURE 3-1: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT



Note 1: Refer to the Configuration Word Register 1 (Register 8-1).

TABLE 3-1: STATUS BITS AND THEIR SIGNIFICANCE

$\overline{POR}$	$\overline{BOR}$	$\overline{TO}$	$\overline{PD}$	Condition
0	x	1	1	Power-on Reset or LDO Reset
0	x	0	x	Illegal, $\overline{TO}$ is set on $\overline{POR}$
0	x	x	0	Illegal, $\overline{PD}$ is set on $\overline{POR}$
1	0	1	1	Brown-out Reset
1	1	0	1	WDT Reset
1	1	0	0	WDT Wake-up
1	1	u	u	$\overline{MCLR}$ Reset during normal operation
1	1	1	0	$\overline{MCLR}$ Reset during Sleep or interrupt wake-up from Sleep

TABLE 3-2: RESET CONDITION FOR SPECIAL REGISTERS(2)

Condition	Program Counter	STATUS Register	PCON Register
Power-on Reset	0000h	0001 1xxx	---- --0x
$\overline{MCLR}$ Reset during normal operation	0000h	000u uuuu	---- --uu
$\overline{MCLR}$ Reset during Sleep	0000h	0001 0uuu	---- --uu
WDT Reset	0000h	0000 1uuu	---- --uu
WDT Wake-up	PC + 1	uuu0 0uuu	---- --uu
Brown-out Reset	0000h	0001 1uuu	---- --u0
Interrupt Wake-up from Sleep	PC + 1(1)	uuu1 0uuu	---- --uu

Legend: u = unchanged, x = unknown, - = unimplemented bit, reads as '0'.

Note 1: When the wake-up is due to an interrupt and Global Enable bit (GIE) is set, the return address is pushed on the stack and PC is loaded with the interrupt vector (0004h) after execution of PC + 1.

2: If a Status bit is not implemented, that bit will be read as '0'.

3.1 MCLR

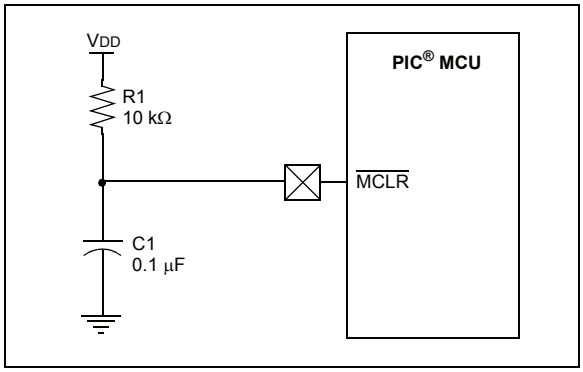
The PIC16(L)F722A/723A has a noise filter in the MCLR Reset path. The filter will detect and ignore small pulses.

It should be noted that a Reset does not drive the MCLR pin low.

Voltages applied to the pin that exceed its specification can result in both MCLR Resets and excessive current beyond the device specification during the ESD event. For this reason, Microchip recommends that the MCLR pin no longer be tied directly to VDD. The use of an RC network, as shown in Figure 3-2, is suggested.

An internal MCLR option is enabled by clearing the MCLRE bit in the Configuration Word register. When MCLRE = 0, the Reset signal to the chip is generated internally. When the MCLRE = 1, the RE3/MCLR pin becomes an external Reset input. In this mode, the RE3/MCLR pin has a weak pull-up to VDD. In-Circuit Serial Programming is not affected by selecting the internal MCLR option.

FIGURE 3-2: RECOMMENDED MCLR CIRCUIT



3.2 Power-on Reset (POR)

The on-chip POR circuit holds the chip in Reset until VDD has reached a high enough level for proper operation. A maximum rise time for VDD is required. See **Section 23.0 "Electrical Specifications"** for details. If the BOR is enabled, the maximum rise time specification does not apply. The BOR circuitry will keep the device in Reset until VDD reaches VBOR (see **Section 3.5 "Brown-Out Reset (BOR)"**).

When the device starts normal operation (exits the Reset condition), device operating parameters (i.e., voltage, frequency, temperature, etc.) must be met to ensure operation. If these conditions are not met, the device must be held in Reset until the operating conditions are met.

For additional information, refer to Application Note AN607, *"Power-up Trouble Shooting"* (DS00607).

3.3 Power-up Timer (PWRT)

The Power-up Timer provides a fixed 64 ms (nominal) time-out on power-up only, from POR or Brown-out Reset. The Power-up Timer operates from the WDT oscillator. For more information, see **Section 7.3 "Internal Clock Modes"**. The chip is kept in Reset as long as PWRT is active. The PWRT delay allows the VDD to rise to an acceptable level. A Configuration bit, PWRTE, can disable (if set) or enable (if cleared or programmed) the Power-up Timer. The Power-up Timer should be enabled when Brown-out Reset is enabled, although it is not required.

The Power-up Timer delay will vary from chip-to-chip and vary due to:

- VDD variation
- Temperature variation
- Process variation

See DC parameters for details (**Section 23.0 "Electrical Specifications"**).

Note:	The Power-up Timer is enabled by the PWRTE bit in the Configuration Word.

3.4 Watchdog Timer (WDT)

The WDT has the following features:

- Shares an 8-bit prescaler with Timer0
- Time-out period is from 17 ms to 2.2 seconds, nominal
- Enabled by a Configuration bit

WDT is cleared under certain conditions described in Table 3-1.

3.4.1 WDT OSCILLATOR

The WDT derives its time base from 31 kHz internal oscillator.

Note:	When the Oscillator Start-up Timer (OST) is invoked, the WDT is held in Reset, because the WDT Ripple Counter is used by the OST to perform the oscillator delay count. When the OST count has expired, the WDT will begin counting (if enabled).

3.4.2 WDT CONTROL

The WDTE bit is located in the Configuration Word Register 1. When set, the WDT runs continuously.

The PSA and PS<2:0> bits of the OPTION register control the WDT period. See **Section 11.0 "Timer0 Module"** for more information.

FIGURE 3-1: WATCHDOG TIMER BLOCK DIAGRAM

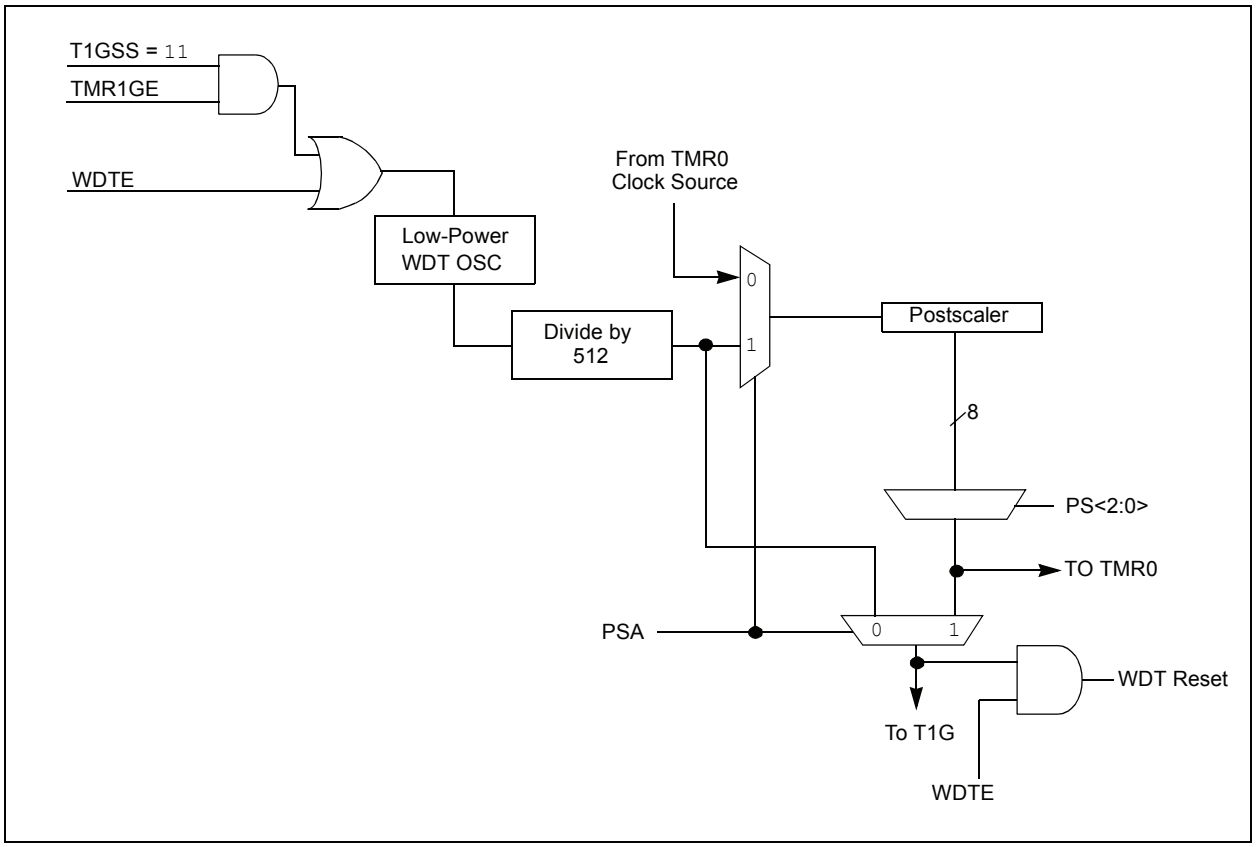


TABLE 3-1: WDT STATUS

Conditions	WDT
WDTE = 0	Cleared
CLRWDT Command	
Exit Sleep + System Clock = T1OSC, EXTRC, INTOSC, EXTCLK	
Exit Sleep + System Clock = XT, HS, LP	Cleared until the end of OST

3.5 Brown-Out Reset (BOR)

Brown-out Reset is enabled by programming the BOREN<1:0> bits in the Configuration register. The brown-out trip point is selectable from two trip points via the BORV bit in the Configuration register.

Between the POR and BOR, complete voltage range coverage for execution protection can be implemented.

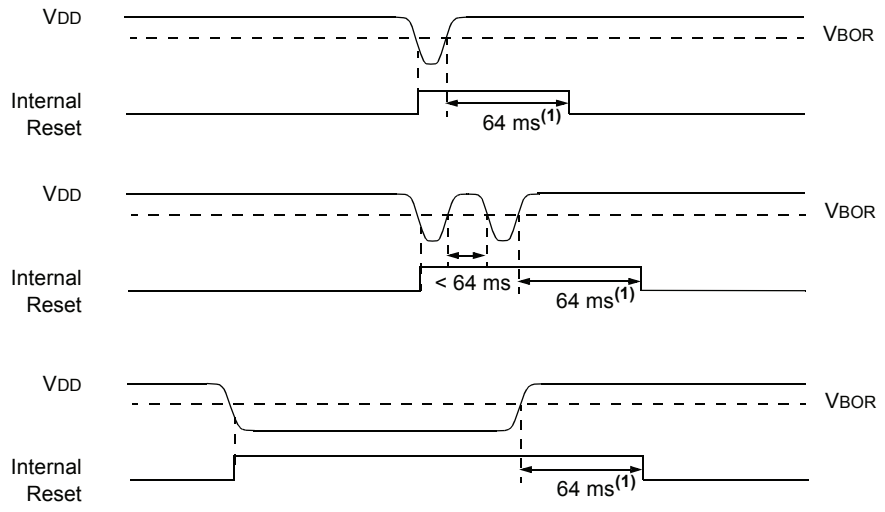
Two bits are used to enable the BOR. When BOREN = 11, the BOR is always enabled. When BOREN = 10, the BOR is enabled, but disabled during Sleep. When BOREN = 0X, the BOR is disabled.

If VDD falls below VBOR for greater than parameter (TBOR) (see **Section 23.0 "Electrical Specifications"**), the brown-out situation will reset the device. This will occur regardless of VDD slew rate. A Reset is not ensured to occur if VDD falls below VBOR for more than parameter (TBOR).

If VDD drops below VBOR while the Power-up Timer is running, the chip will go back into a Brown-out Reset and the Power-up Timer will be re-initialized. Once VDD rises above VBOR, the Power-up Timer will execute a 64 ms Reset.

Note:	When erasing Flash program memory, the BOR is forced to enabled at the minimum BOR setting to ensure that any code protection circuitry is operating properly.

FIGURE 3-3: BROWN-OUT SITUATIONS



Note 1: 64 ms delay only if $\overline{\text{PWRTE}}$ bit is programmed to '0'.

3.6 Time-out Sequence

On power-up, the time-out sequence is as follows: first, PWRT time-out is invoked after POR has expired, then OST is activated after the PWRT time-out has expired. The total time-out will vary based on oscillator configuration and $\overline{\text{PWRTE}}$ bit status. For example, in EC mode with $\overline{\text{PWRTE}}$ bit = 1 (PWRT disabled), there will be no time-out at all. Figure 3-4, Figure 3-5 and Figure 3-6 depict time-out sequences.

Since the time-outs occur from the POR pulse, if $\overline{\text{MCLR}}$ is kept low long enough, the time-outs will expire. Then, bringing $\overline{\text{MCLR}}$ high will begin execution immediately (see Figure 3-5). This is useful for testing purposes or to synchronize more than one PIC16(L)F722A/723A device operating in parallel.

Table 3-3 shows the Reset conditions for some special registers.

3.7 Power Control (PCON) Register

The Power Control (PCON) register has two Status bits to indicate what type of Reset that last occurred.

Bit 0 is $\overline{\text{BOR}}$ (Brown-out Reset). $\overline{\text{BOR}}$ is unknown on Power-on Reset. It must then be set by the user and checked on subsequent Resets to see if $\overline{\text{BOR}}$ = 0, indicating that a brown-out has occurred. The $\overline{\text{BOR}}$ Status bit is a “don’t care” and is not necessarily predictable if the brown-out circuit is disabled (BOREN<1:0> = 00 in the Configuration Word register).

Bit 1 is $\overline{\text{POR}}$ (Power-on Reset). It is a ‘0’ on Power-on Reset and unaffected otherwise. The user must write a ‘1’ to this bit following a Power-on Reset. On a subsequent Reset, if $\overline{\text{POR}}$ is ‘0’, it will indicate that a Power-on Reset has occurred (i.e., V_{DD} may have gone too low).

For more information, see **Section 3.5 “Brown-Out Reset (BOR)”**.

TABLE 3-2: TIME-OUT IN VARIOUS SITUATIONS

Oscillator Configuration	Power-up		Brown-out Reset		Wake-up from Sleep
	$\overline{\text{PWRTE}}$ = 0	$\overline{\text{PWRTE}}$ = 1	$\overline{\text{PWRTE}}$ = 0	$\overline{\text{PWRTE}}$ = 1	
XT, HS, LP(1)	TPWRT + 1024 • TOSC	1024 • TOSC	TPWRT + 1024 • TOSC	1024 • TOSC	1024 • TOSC
RC, EC, INTOSC	TPWRT	—	TPWRT	—	—

Note 1: LP mode with T1OSC disabled.

TABLE 3-3: RESET BITS AND THEIR SIGNIFICANCE

$\overline{\text{POR}}$	$\overline{\text{BOR}}$	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Condition
0	u	1	1	Power-on Reset
1	0	1	1	Brown-out Reset
u	u	0	u	WDT Reset
u	u	0	0	WDT Wake-up
u	u	u	u	$\overline{\text{MCLR}}$ Reset during normal operation
u	u	1	0	$\overline{\text{MCLR}}$ Reset during Sleep

Legend: u = unchanged, x = unknown

FIGURE 3-4: TIME-OUT SEQUENCE ON POWER-UP (DELAYED $\overline{\text{MCLR}}$): CASE 1

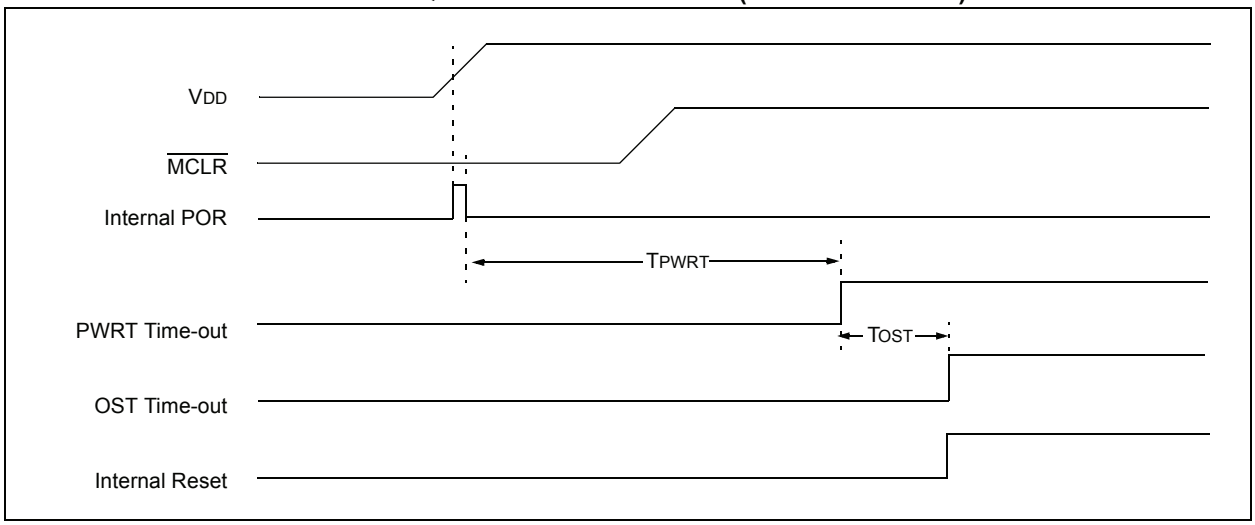


FIGURE 3-5: TIME-OUT SEQUENCE ON POWER-UP (DELAYED $\overline{\text{MCLR}}$): CASE 2

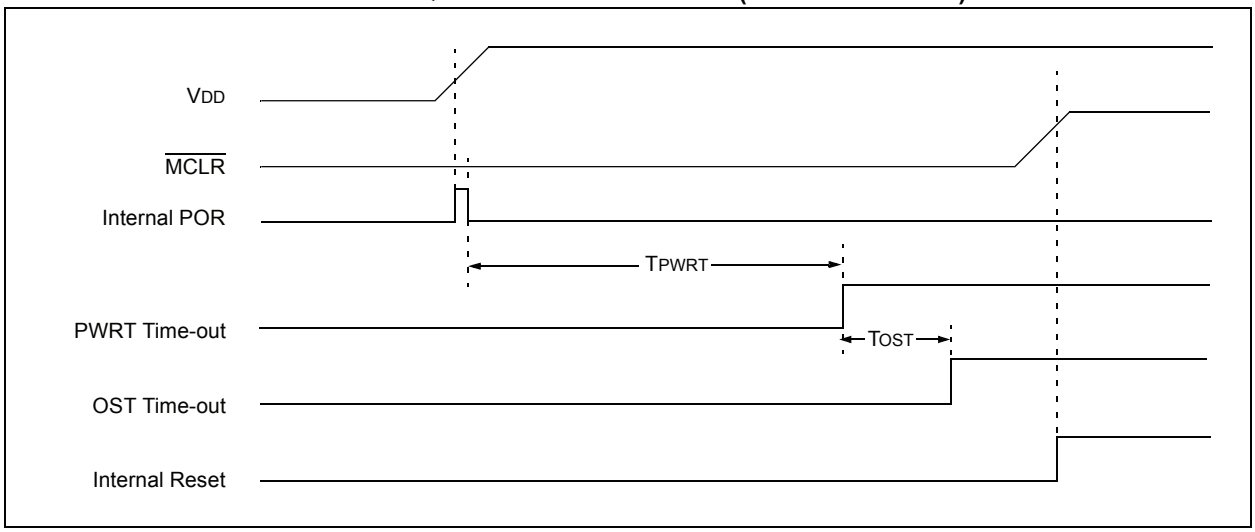


FIGURE 3-6: TIME-OUT SEQUENCE ON POWER-UP ($\overline{\text{MCLR}}$ WITH VDD): CASE 3

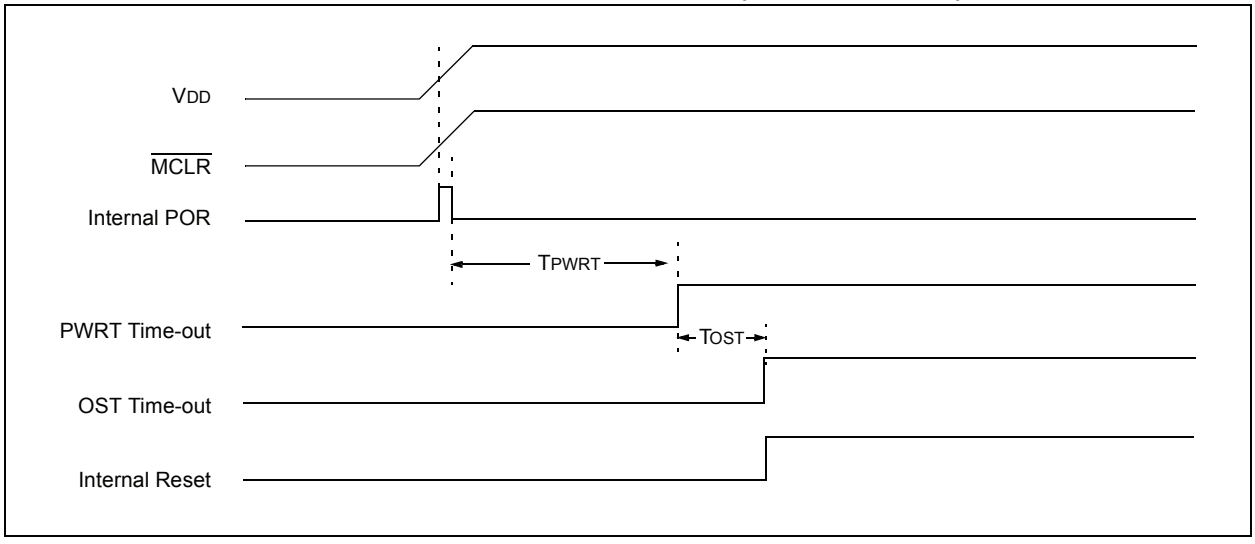


TABLE 3-4: INITIALIZATION CONDITION FOR REGISTERS

Register	Address	Power-on Reset/ Brown-out Reset(1)	$\overline{\text{MCLR}}$ Reset/ WDT Reset	Wake-up from Sleep through Interrupt/Time-out
W	—	xxxx xxxx	uuuu uuuu	uuuu uuuu
INDF	00h/80h/ 100h/180h	xxxx xxxx	xxxx xxxx	uuuu uuuu
TMR0	01h/101h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCL	02h/82h/ 102h/182h	0000 0000	0000 0000	PC + 1(3)
STATUS	03h/83h/ 103h/183h	0001 1xxx	000q quuu(4)	uuuq quuu(4)
FSR	04h/84h/ 104h/184h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTA	05h	xxxx xxxx	xxxx xxxx	uuuu uuuu
PORTB	06h	xxxx xxxx	xxxx xxxx	uuuu uuuu
PORTC	07h	xxxx xxxx	xxxx xxxx	uuuu uuuu
PORTE	09h	---- x---	---- x---	---- u---
PCLATH	0Ah/8Ah/ 10Ah/18Ah	---0 0000	---0 0000	---u uuuu
INTCON	0Bh/8Bh/ 10Bh/18Bh	0000 000x	0000 000x	uuuu uuuu(2)
PIR1	0Ch	0000 0000	0000 0000	uuuu uuuu(2)
PIR2	0Dh	---- ---0	---- ---0	---- ---u
TMR1L	0Eh	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR1H	0Fh	xxxx xxxx	uuuu uuuu	uuuu uuuu
T1CON	10h	0000 00-0	uuuu uu-u	uuuu uu-u
TMR2	11h	0000 0000	0000 0000	uuuu uuuu
T2CON	12h	-000 0000	-000 0000	-uuu uuuu
SSPBUF	13h	xxxx xxxx	xxxx xxxx	uuuu uuuu
SSPCON	14h	0000 0000	0000 0000	uuuu uuuu
CCPR1L	15h	xxxx xxxx	xxxx xxxx	uuuu uuuu
CCPR1H	16h	xxxx xxxx	xxxx xxxx	uuuu uuuu
CCP1CON	17h	--00 0000	--00 0000	--uu uuuu
RCSTA	18h	0000 000x	0000 000x	uuuu uuuu
TXREG	19h	0000 0000	0000 0000	uuuu uuuu
RCREG	1Ah	0000 0000	0000 0000	uuuu uuuu
CCPR2L	1Bh	xxxx xxxx	xxxx xxxx	uuuu uuuu
CCPR2H	1Ch	xxxx xxxx	xxxx xxxx	uuuu uuuu
CCP2CON	1Dh	--00 0000	--00 0000	--uu uuuu
ADRES	1Eh	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADCON0	1Fh	--00 0000	--00 0000	--uu uuuu

Legend: u = unchanged, x = unknown, - = unimplemented bit, reads as '0', q = value depends on condition.

Note 1: If VDD goes too low, Power-on Reset will be activated and registers will be affected differently.

2: One or more bits in INTCON and/or PIR1 and PIR2 will be affected (to cause wake-up).

3: When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).

4: See Table 3-5 for Reset value for specific condition.

5: If Reset was due to brown-out, then bit 0 = 0. All other Resets will cause bit 0 = u.

TABLE 3-4: INITIALIZATION CONDITION FOR REGISTERS (CONTINUED)

Register	Address	Power-on Reset/ Brown-out Reset(1)	$\overline{\text{MCLR}}$ Reset/ WDT Reset	Wake-up from Sleep through Interrupt/Time-out
OPTION_REG	81h/181h	1111 1111	1111 1111	uuuu uuuu
TRISA	85h	1111 1111	1111 1111	uuuu uuuu
TRISB	86h	1111 1111	1111 1111	uuuu uuuu
TRISC	87h	1111 1111	1111 1111	uuuu uuuu
TRISE	89h	---- 1---	---- 1---	---- u---
PIE1	8Ch	0000 0000	0000 0000	uuuu uuuu
PIE2	8Dh	---- ---0	---- ---0	---- ---u
PCON	8Eh	---- --qq	---- --uu(1,5)	---- --uu
T1GCON	8Fh	0000 0x00	uuuu uxuu	uuuu uxuu
OSCCON	90h	--10 qq--	--10 qq--	--uu qq--
OSCTUNE	91h	--00 0000	--uu uuuu	--uu uuuu
PR2	92h	1111 1111	1111 1111	uuuu uuuu
SSPADD	93h	0000 0000	0000 0000	uuuu uuuu
SSPMSK	93h	1111 1111	1111 1111	uuuu uuuu
SSPSTAT	94h	0000 0000	0000 0000	uuuu uuuu
WPUB	95h	1111 1111	1111 1111	uuuu uuuu
IOCB	96h	0000 0000	0000 0000	uuuu uuuu
TXSTA	98h	0000 -010	0000 -010	uuuu -uuu
SPBRG	99h	0000 0000	0000 0000	uuuu uuuu
APFCON	9Ch	---- --00	---- --00	---- --uu
FVRCON	9Dh	q000 --00	q000 --00	uuuu --uu
ADCON1	9Fh	-000 --00	-000 --00	-uuu --uu
CPSCON0	108h	0--- 0000	0--- 0000	u--- uuuu
CPSCON1	109h	---- 0000	---- 0000	---- uuuu
PMDATL	10Ch	xxxx xxxx	xxxx xxxx	uuuu uuuu
PMADRL	10Dh	xxxx xxxx	xxxx xxxx	uuuu uuuu
PMDATH	10Eh	--xx xxxx	--xx xxxx	--uu uuuu
PMADRH	10Fh	---x xxxx	---x xxxx	---u uuuu
ANSELA	185h	--11 1111	--11 1111	--uu uuuu
ANSELB	186h	--11 1111	--11 1111	--uu uuuu
PMCON1	18Ch	1--- ---0	1--- ---0	u--- ---u

Legend: u = unchanged, x = unknown, - = unimplemented bit, reads as '0', q = value depends on condition.

Note 1: If VDD goes too low, Power-on Reset will be activated and registers will be affected differently.

2: One or more bits in INTCON and/or PIR1 and PIR2 will be affected (to cause wake-up).

3: When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).

4: See Table 3-5 for Reset value for specific condition.

5: If Reset was due to brown-out, then bit 0 = 0. All other Resets will cause bit 0 = u.

TABLE 3-5: INITIALIZATION CONDITION FOR SPECIAL REGISTERS

Condition	Program Counter	STATUS Register	PCON Register
Power-on Reset	0000h	0001 1xxx	---- --0x
$\overline{MCLR}$ Reset during normal operation	0000h	000u uuuu	---- --uu
$\overline{MCLR}$ Reset during Sleep	0000h	0001 0uuu	---- --uu
WDT Reset	0000h	0000 uuuu	---- --uu
WDT Wake-up	PC + 1	uuu0 0uuu	---- --uu
Brown-out Reset	0000h	0001 1xxx	---- --10
Interrupt Wake-up from Sleep	PC + 1(1)	uuu1 0uuu	---- --uu

Legend: u = unchanged, x = unknown, - = unimplemented bit, reads as '0'.

Note 1: When the wake-up is due to an interrupt and Global Interrupt Enable bit, GIE, is set, the PC is loaded with the interrupt vector (0004h) after execution of PC + 1.

TABLE 3-6: SUMMARY OF REGISTERS ASSOCIATED WITH RESETS

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	22
PCON	—	—	—	—	—	—	$\overline{POR}$	$\overline{BOR}$	24

Legend: u = unchanged, x = unknown, - = unimplemented bit, reads as '0', q = value depends on condition. Shaded cells are not used by Resets.

Note 1: Other (non Power-up) Resets include $\overline{MCLR}$ Reset and Watchdog Timer Reset during normal operation.

4.0 INTERRUPTS

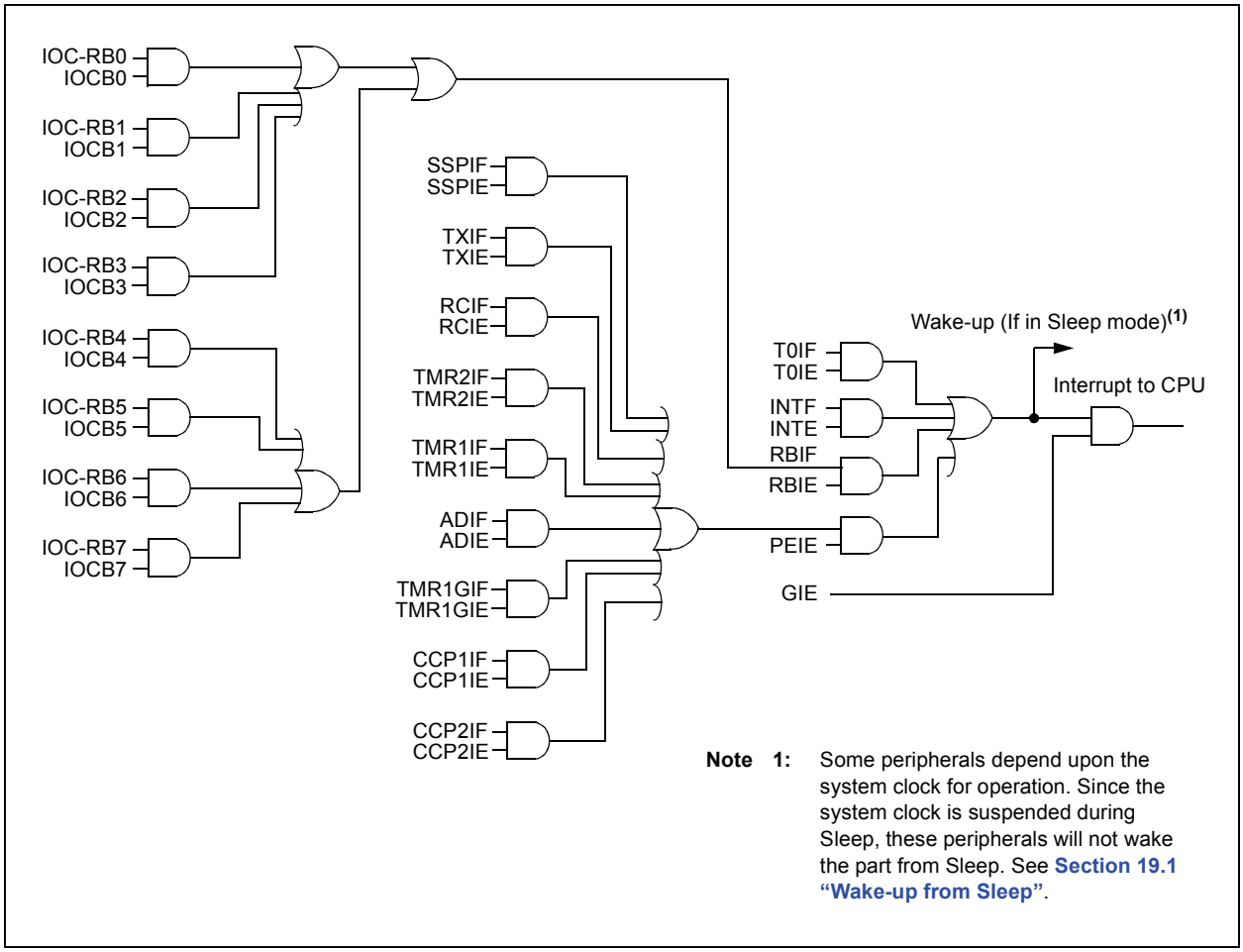
The PIC16(L)F722A/723A device family features an interruptible core, allowing certain events to preempt normal program flow. An Interrupt Service Routine (ISR) is used to determine the source of the interrupt and act accordingly. Some interrupts can be configured to wake the MCU from Sleep mode.

The PIC16(L)F722A/723A device family has 12 interrupt sources, differentiated by corresponding interrupt enable and flag bits:

- Timer0 Overflow Interrupt
- External Edge Detect on INT Pin Interrupt
- PORTB Change Interrupt
- 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
- CCP2 Event Interrupt

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

FIGURE 4-1: INTERRUPT LOGIC



Note 1: Some peripherals depend upon the system clock for operation. Since the system clock is suspended during Sleep, these peripherals will not wake the part from Sleep. See **Section 19.1 "Wake-up from Sleep"**.

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.

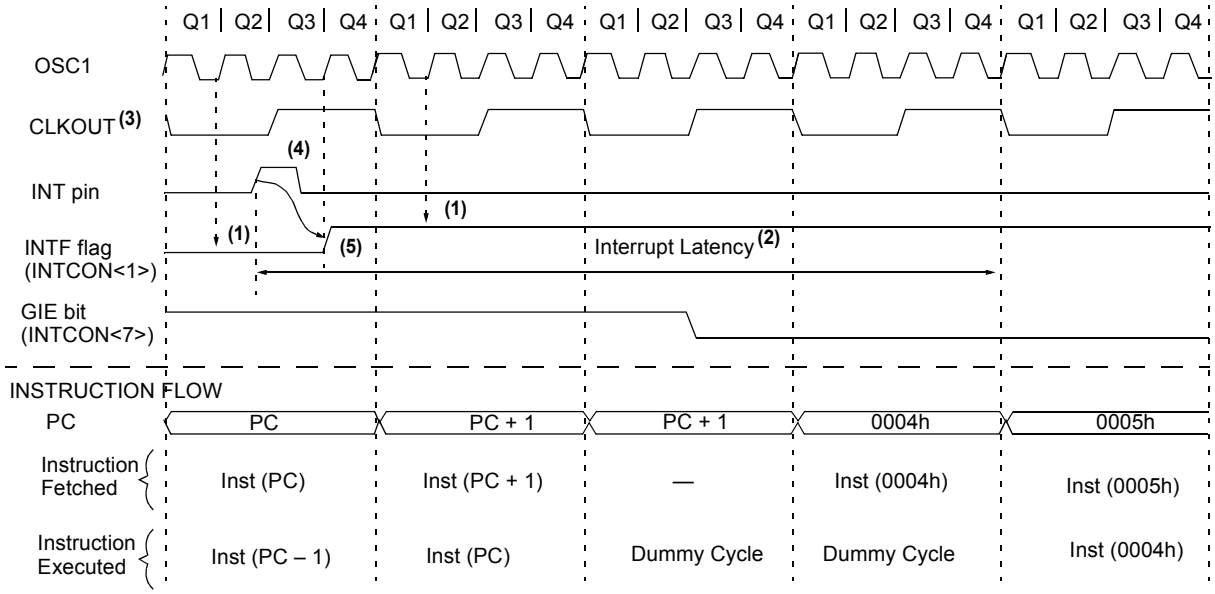
For additional information on a specific interrupt's operation, refer to its peripheral chapter.

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 3 instruction cycles. For asynchronous interrupts, the latency is 3 to 4 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 **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.

EXAMPLE 4-1: SAVING W, STATUS AND PCLATH REGISTERS IN RAM

```
MOVWFW_TEMP             ;Copy W to W_TEMP register
SWAPFSTATUS,W           ;Swap status to be saved into W
                        ;Swaps are used because they do not affect the status bits
BANKSELSTATUS_TEMP      ;Select regardless of current bank
MOVWFSTATUS_TEMP        ;Copy status to bank zero STATUS_TEMP register
MOVF    PCLATH,W        ;Copy PCLATH to W register
MOVWF   PCLATH_TEMP     ;Copy W register to PCLATH_TEMP
:
:(ISR)                  ;Insert user code here
:
BANKSELSTATUS_TEMP      ;Select regardless of current bank
MOVF    PCLATH_TEMP,W ;
MOVWF   PCLATH          ;Restore PCLATH
SWAPFSTATUS_TEMP,W      ;Swap STATUS_TEMP register into W
                        ;(sets bank to original state)
MOVWFSTATUS             ;Move W into STATUS register
SWAPFW_TEMP,F           ;Swap W_TEMP
SWAPFW_TEMP,W           ;Swap W_TEMP into W
```

4.5.1 INTCON REGISTER

The INTCON register is a readable and writable register, which contains the various enable and flag bits for TMR0 register overflow, PORTB change and external RB0/INT/SEG0 pin interrupts.

Note:	Interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the global enable bit, GIE of the INTCON register. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

REGISTER 4-1: INTCON: INTERRUPT CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE	PEIE	T0IE	INTE	RBIE(1)	T0IF(2)	INTF	RBIF
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 **GIE:** Global Interrupt Enable bit
1 = Enables all unmasked interrupts
0 = Disables all interrupts

bit 6 **PEIE:** Peripheral Interrupt Enable bit
1 = Enables all unmasked peripheral interrupts
0 = Disables all peripheral interrupts

bit 5 **T0IE:** Timer0 Overflow Interrupt Enable bit
1 = Enables the Timer0 interrupt
0 = Disables the Timer0 interrupt

bit 4 **INTE:** RB0/INT External Interrupt Enable bit
1 = Enables the RB0/INT external interrupt
0 = Disables the RB0/INT external interrupt

bit 3 **RBIE:** PORTB Change Interrupt Enable bit(1)
1 = Enables the PORTB change interrupt
0 = Disables the PORTB change interrupt

bit 2 **T0IF:** Timer0 Overflow Interrupt Flag bit(2)
1 = TMR0 register has overflowed (must be cleared in software)
0 = TMR0 register did not overflow

bit 1 **INTF:** RB0/INT External Interrupt Flag bit
1 = The RB0/INT external interrupt occurred (must be cleared in software)
0 = The RB0/INT external interrupt did not occur

bit 0 **RBIF:** PORTB Change Interrupt Flag bit
1 = When at least one of the PORTB general purpose I/O pins changed state (must be cleared in software)
0 = None of the PORTB general purpose I/O pins have changed state

Note 1: The appropriate bits in the IOCB register must also be set.
2: T0IF bit is set when Timer0 rolls over. Timer0 is unchanged on Reset and should be initialized before clearing T0IF bit.

4.5.2 PIE1 REGISTER

The PIE1 register contains the interrupt enable bits, as shown in Register 4-2.

Note: Bit PEIE of the INTCON register must be set to enable any peripheral interrupt.

REGISTER 4-2: PIE1: PERIPHERAL INTERRUPT ENABLE REGISTER 1

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
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 **TMR1GIE:** Timer1 Gate Interrupt Enable bit
1 = Enable the Timer1 gate acquisition complete interrupt
0 = Disable the Timer1 gate acquisition complete interrupt

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

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

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

bit 3 **SSPIE:** Synchronous Serial Port (SSP) Interrupt Enable bit
1 = Enables the SSP interrupt
0 = Disables the SSP interrupt

bit 2 **CCP1IE:** CCP1 Interrupt Enable bit
1 = Enables the CCP1 interrupt
0 = Disables the CCP1 interrupt

bit 1 **TMR2IE:** TMR2 to PR2 Match Interrupt Enable bit
1 = Enables the Timer2 to PR2 match interrupt
0 = Disables the Timer2 to PR2 match interrupt

bit 0 **TMR1IE:** Timer1 Overflow Interrupt Enable bit
1 = Enables the Timer1 overflow interrupt
0 = Disables the Timer1 overflow interrupt

4.5.3 PIE2 REGISTER

The PIE2 register contains the interrupt enable bits, as shown in Register 4-3.

Note: Bit PEIE of the INTCON register must be set to enable any peripheral interrupt.

REGISTER 4-3: PIE2: PERIPHERAL INTERRUPT ENABLE REGISTER 2

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	CCP2IE
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 7-1 **Unimplemented:** Read as ‘0’

bit 0 **CCP2IE:** CCP2 Interrupt Enable bit
1 = Enables the CCP2 interrupt
0 = Disables the CCP2 interrupt

4.5.4 PIR1 REGISTER

The PIR1 register contains the interrupt flag bits, as shown in Register 4-4.

Note:	Interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the Global Enable bit, GIE of the INTCON register. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

REGISTER 4-4: PIR1: PERIPHERAL INTERRUPT REQUEST REGISTER 1

R/W-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
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 **TMR1GIF:** Timer1 Gate Interrupt Flag bit
1 = Timer1 gate is inactive
0 = Timer1 gate is active

bit 6 **ADIF:** A/D Converter Interrupt Flag bit
1 = A/D conversion complete (must be cleared in software)
0 = A/D conversion has not completed or has not been started

bit 5 **RCIF:** USART Receive Interrupt Flag bit
1 = The USART receive buffer is full (cleared by reading RCREG)
0 = The USART receive buffer is not full

bit 4 **TXIF:** USART Transmit Interrupt Flag bit
1 = The USART transmit buffer is empty (cleared by writing to TXREG)
0 = The USART transmit buffer is full

bit 3 **SSPIF:** Synchronous Serial Port (SSP) Interrupt Flag bit
1 = The Transmission/Reception is complete (must be cleared in software)
0 = Waiting to Transmit/Receive

bit 2 **CCP1IF:** CCP1 Interrupt Flag bit
Capture mode:
1 = A TMR1 register capture occurred (must be cleared in software)
0 = No TMR1 register capture occurred
Compare mode:
1 = A TMR1 register compare match occurred (must be cleared in software)
0 = No TMR1 register compare match occurred
PWM mode:
Unused in this mode

bit 1 **TMR2IF:** Timer2 to PR2 Interrupt Flag bit
1 = A Timer2 to PR2 match occurred (must be cleared in software)
0 = No Timer2 to PR2 match occurred

bit 0 **TMR1IF:** Timer1 Overflow Interrupt Flag bit
1 = The TMR1 register overflowed (must be cleared in software)
0 = The TMR1 register did not overflow

4.5.5 PIR2 REGISTER

The PIR2 register contains the interrupt flag bits, as shown in Register 4-5.

Note: Interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the global enable bit, GIE of the INTCON register. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

REGISTER 4-5: PIR2: PERIPHERAL INTERRUPT REQUEST REGISTER 2

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	CCP2IF
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 7-1 **Unimplemented:** Read as '0'

bit 0 **CCP2IF:** CCP2 Interrupt Flag bit

Capture Mode:

1 = A TMR1 register capture occurred (must be cleared in software)
0 = No TMR1 register capture occurred

Compare Mode:

1 = A TMR1 register compare match occurred (must be cleared in software)
0 = No TMR1 register compare match occurred

PWM mode:

Unused in this mode

TABLE 4-1: SUMMARY OF REGISTERS ASSOCIATED WITH INTERRUPTS

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	40
OPTION_REG	$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	23
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	41
PIE2	—	—	—	—	—	—	—	CCP2IE	42
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	43
PIR2	—	—	—	—	—	—	—	CCP2IF	44

Legend: – = Unimplemented locations, read as '0', u = unchanged, x = unknown. Shaded cells are not used by the Capture, Compare and PWM.

5.0 LOW DROPOUT (LDO) VOLTAGE REGULATOR

The PIC16F722A/723A devices differ from the PIC16LF722A/723A devices due to an internal Low Dropout (LDO) voltage regulator. The PIC16F722A/723A contain an internal LDO, while the PIC16LF722A/723A do not.

The lithography of the die allows a maximum operating voltage of 3.6V on the internal digital logic. In order to continue to support 5.0V designs, a LDO voltage regulator is integrated on the die. The LDO voltage regulator allows for the internal digital logic to operate at 3.2V, while I/O's operate at 5.0V (V_{DD}).

The LDO voltage regulator requires an external bypass capacitor for stability. One of three pins, denoted as V_{CAP}, can be configured for the external bypass capacitor. It is recommended that the capacitor be a ceramic cap between 0.1 to 1.0 µF. The V_{CAP} pin is not intended to supply power to external loads. An external voltage regulator should be used if this functionality is required. In addition, external devices should not supply power to the V_{CAP} pin.

On power-up, the external capacitor will look like a large load on the LDO voltage regulator. To prevent erroneous operation, the device is held in Reset while a constant current source charges the external capacitor. After the cap is fully charged, the device is released from Reset. For more information, refer to **Section 23.0 "Electrical Specifications"**.

See Configuration Word 2 register (Register 8-2) for V_{CAP} enable bits.

NOTES:

6.0 I/O PORTS

There are as many as thirty-five general purpose I/O pins available. Depending on which peripherals are enabled, some or all of the pins may not be available as general purpose I/O. In general, when a peripheral is enabled, the associated pin may not be used as a general purpose I/O pin.

6.1 Alternate Pin Function

The Alternate Pin Function Control (APFCON) register is used to steer specific peripheral input and output functions between different pins. The APFCON register is shown in Register 6-1. For this device family, the following functions can be moved between different pins:

- $\overline{SS}$ (Slave Select)
- CCP2

REGISTER 6-1: APFCON: ALTERNATE PIN FUNCTION CONTROL REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	SSSEL	CCP2SEL
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 7-2 **Unimplemented:** Read as '0'.

bit 1 **SSSEL:** $\overline{SS}$ Input Pin Selection bit
0 = $\overline{SS}$ function is on RA5/AN4/CPS7/$\overline{SS}$/VCAP
1 = $\overline{SS}$ function is on RA0/AN0/$\overline{SS}$/VCAP

bit 0 **CCP2SEL:** CCP2 Input/Output Pin Selection bit
0 = CCP2 function is on RC1/T1OSI/CCP2
1 = CCP2 function is on RB3/CCP2

6.2 PORTA and the TRISA Registers

PORTA is a 8-bit wide, bidirectional port. The corresponding data direction register is TRISA (Register 6-3). Setting a TRISA bit (= 1) will make the corresponding PORTA pin an input (i.e., disable the output driver). Clearing a TRISA bit (= 0) will make the corresponding PORTA pin an output (i.e., enables output driver and puts the contents of the output latch on the selected pin). Example 6-1 shows how to initialize PORTA.

Reading the PORTA register (Register 6-2) reads the status of the pins, whereas writing to it will write to the PORT latch. All write operations are read-modify-write operations. Therefore, a write to a port implies that the port pins are read, this value is modified and then written to the PORT data latch.

The TRISA register (Register 6-3) controls the PORTA pin output drivers, even when they are being used as analog inputs. The user should ensure the bits in the TRISA register are maintained set when using them as analog inputs. I/O pins configured as analog input always read '0'.

Note:	The ANSELA register must be initialized to configure an analog channel as a digital input. Pins configured as analog inputs will read '0'.

EXAMPLE 6-1: INITIALIZING PORTA

```
BANKSEL PORTA       ;
CLRF    PORTA       ;Init PORTA
BANKSEL ANSELA      ;
CLRF    ANSELA      ;digital I/O
BANKSEL TRISA       ;
MOVLW   0Ch         ;Set RA<3:2> as inputs
MOVWF   TRISA       ;and set RA<7:4,1:0>
                    ;as outputs
```

REGISTER 6-2: PORTA: PORTA REGISTER

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
RA7	RA6	RA5	RA4	RA3	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 7-0 **RA<7:0>**: PORTA I/O Pin bit
1 = Port pin is > V_{IH}
0 = Port pin is < V_{IL}

REGISTER 6-3: TRISA: PORTA TRI-STATE REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	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 7-0 **TRISA<7:0>:** PORTA Tri-State Control bit
1 = PORTA pin configured as an input (tri-stated)
0 = PORTA pin configured as an output

6.2.1 ANSELA REGISTER

The ANSELA register (Register 6-4) is used to configure the Input mode of an I/O pin to analog. Setting the appropriate ANSELA bit high will cause all digital reads on the pin to be read as '0' and allow analog functions on the pin to operate correctly.

The state of the ANSELA bits has no affect on digital output functions. A pin with TRIS clear and ANSEL set will still operate as a digital output, but the Input mode will be analog. This can cause unexpected behavior when executing read-modify-write instructions on the affected port.

REGISTER 6-4: ANSELA: PORTA ANALOG SELECT REGISTER

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	ANSA5	ANSA4	ANSA3	ANSA2	ANSA1	ANSA0
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 **ANSA<5:0>**: Analog Select between Analog or Digital Function on pins RA<5:0>, respectively
0 = Digital I/O. Pin is assigned to port or digital special function.
1 = Analog input. Pin is assigned as analog input(1). Digital input buffer disabled.

Note 1: When setting a pin to an analog input, the corresponding TRIS bit must be set to Input mode in order to allow external control of the voltage on the pin.

6.2.2 PIN DESCRIPTIONS AND DIAGRAMS

Each PORTA pin is multiplexed with other functions. The pins and their combined functions are briefly described here. For specific information about individual functions such as the A/D Converter (ADC), refer to the appropriate section in this data sheet.

6.2.2.1 RA0/AN0/$\overline{SS}$/VCAP

Figure 6-1 shows the diagram for this pin. This pin is configurable to function as one of the following:

- General purpose I/O
- Analog input for the ADC
- Slave select input for the SSP(1)
- Voltage regulator capacitor pin (PIC16F722A/723A only)

Note 1: $\overline{SS}$ pin location may be selected as RA5 or RA0.

6.2.2.2 RA1/AN1

Figure 6-2 shows the diagram for this pin. This pin is configurable to function as one of the following:

- General purpose I/O
- Analog input for the ADC

6.2.2.3 RA2/AN2

Figure 6-2 shows the diagram for this pin. This pin is configurable to function as one of the following:

- General purpose I/O
- Analog input for the ADC

6.2.2.4 RA3/AN3/VREF

Figure 6-2 shows the diagram for this pin. This pin is configurable to function as one of the following:

- General purpose I/O
- Analog input for the ADC
- Voltage reference input for the ADC

6.2.2.5 RA4/CPS6/T0CKI

Figure 6-3 shows the diagram for this pin. This pin is configurable to function as one of the following:

- General purpose I/O
- Capacitive sensing input
- Clock input for Timer0

The Timer0 clock input function works independently of any TRIS register setting. Effectively, if TRISA4 = 0, the PORTA4 register bit will output to the pad and clock Timer0 at the same time.

6.2.2.6 RA5/AN4/CPS7/$\overline{SS}$/VCAP

Figure 6-4 shows the diagram for this pin. This pin is configurable to function as one of the following:

- General purpose I/O
- Analog input for the ADC
- Capacitive sensing input
- Slave select input for the SSP(1)
- Voltage regulator capacitor pin (PIC16F722A/723A only)

Note 1: $\overline{SS}$ pin location may be selected as RA5 or RA0.

6.2.2.7 RA6/OSC2/CLKOUT/VCAP

Figure 6-5 shows the diagram for this pin. This pin is configurable to function as one of the following:

- General purpose I/O
- Crystal/resonator connection
- Clock output
- Voltage regulator capacitor pin (PIC16F722A/723A only)

6.2.2.8 RA7/OSC1/CLKIN

Figure 6-6 shows the diagram for this pin. This pin is configurable to function as one of the following:

- General purpose I/O
- Crystal/resonator connection
- Clock input

FIGURE 6-1: BLOCK DIAGRAM OF RA0

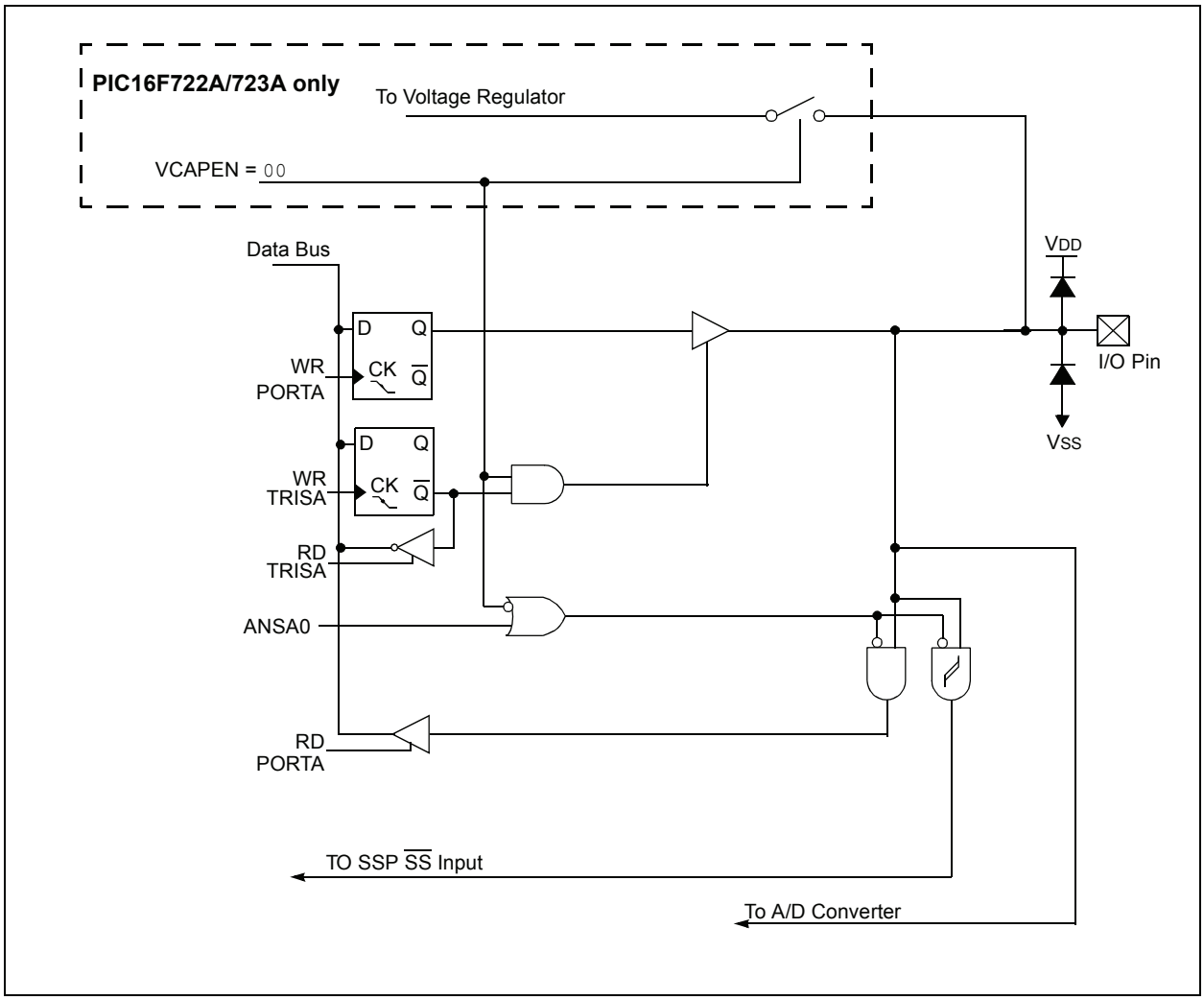


FIGURE 6-2: RA<3:1> BLOCK DIAGRAM

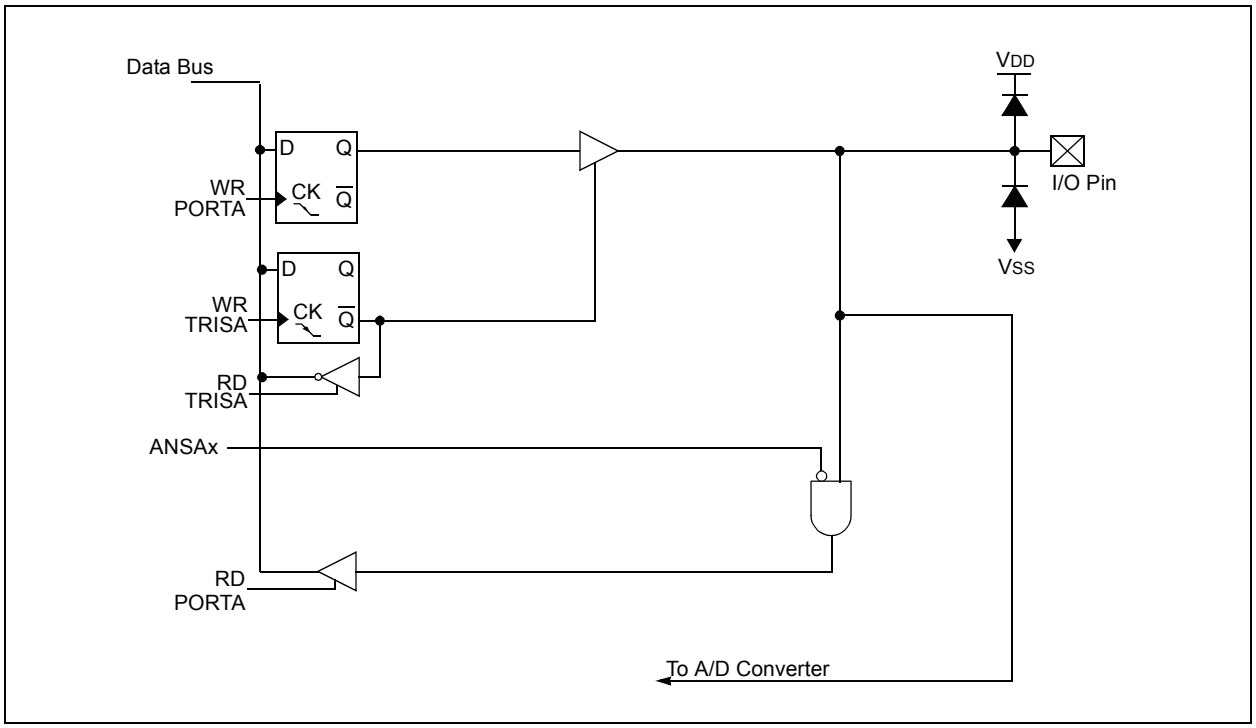


FIGURE 6-3: BLOCK DIAGRAM OF RA4

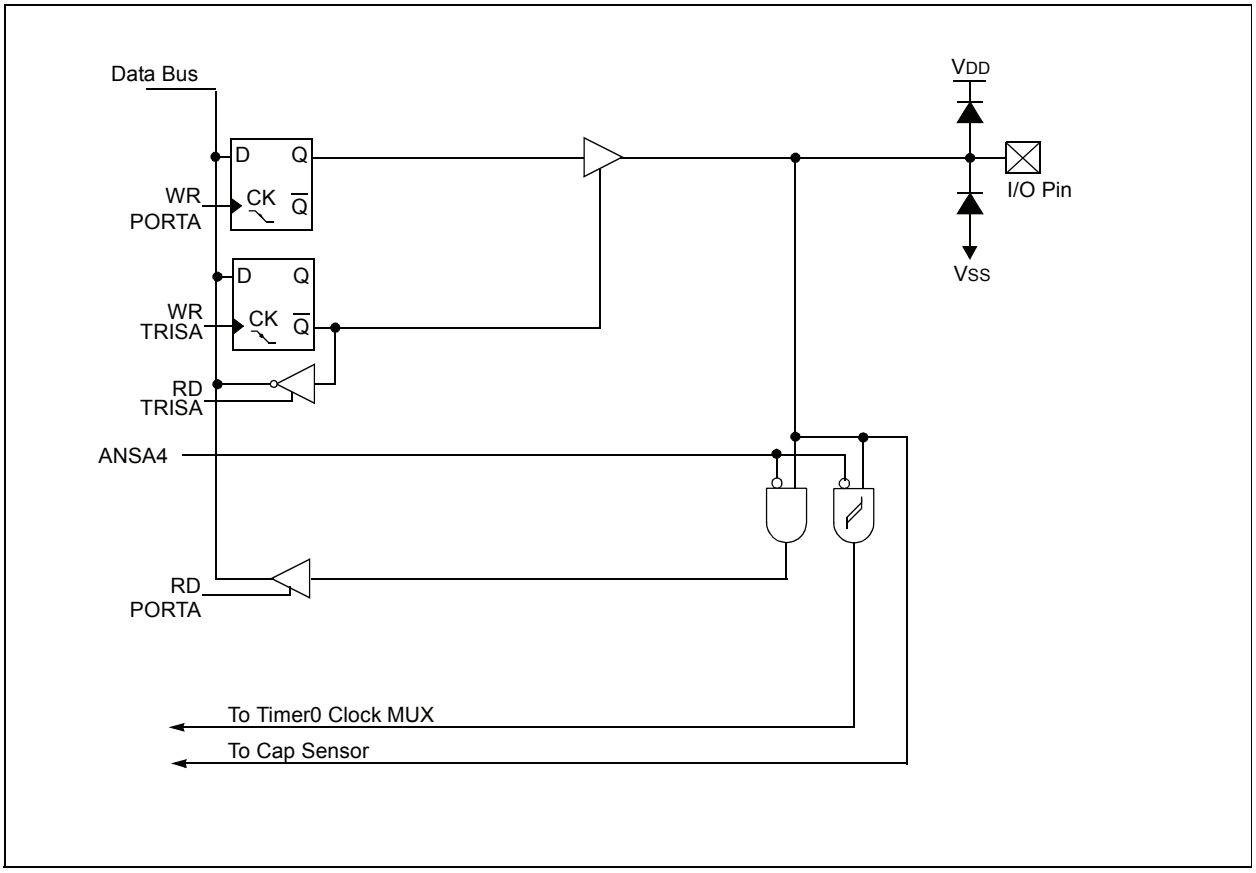


FIGURE 6-4: BLOCK DIAGRAM OF RA5

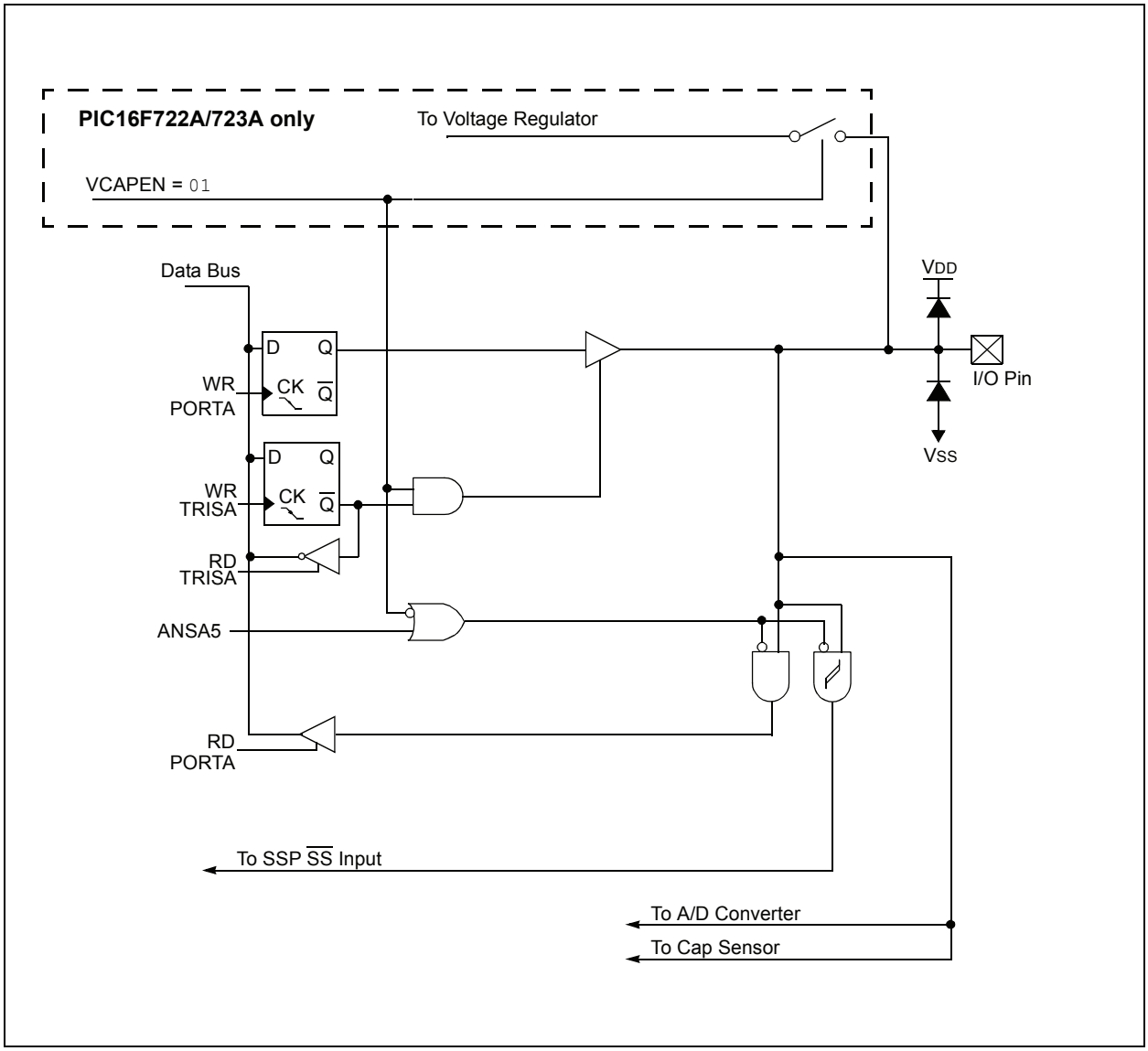
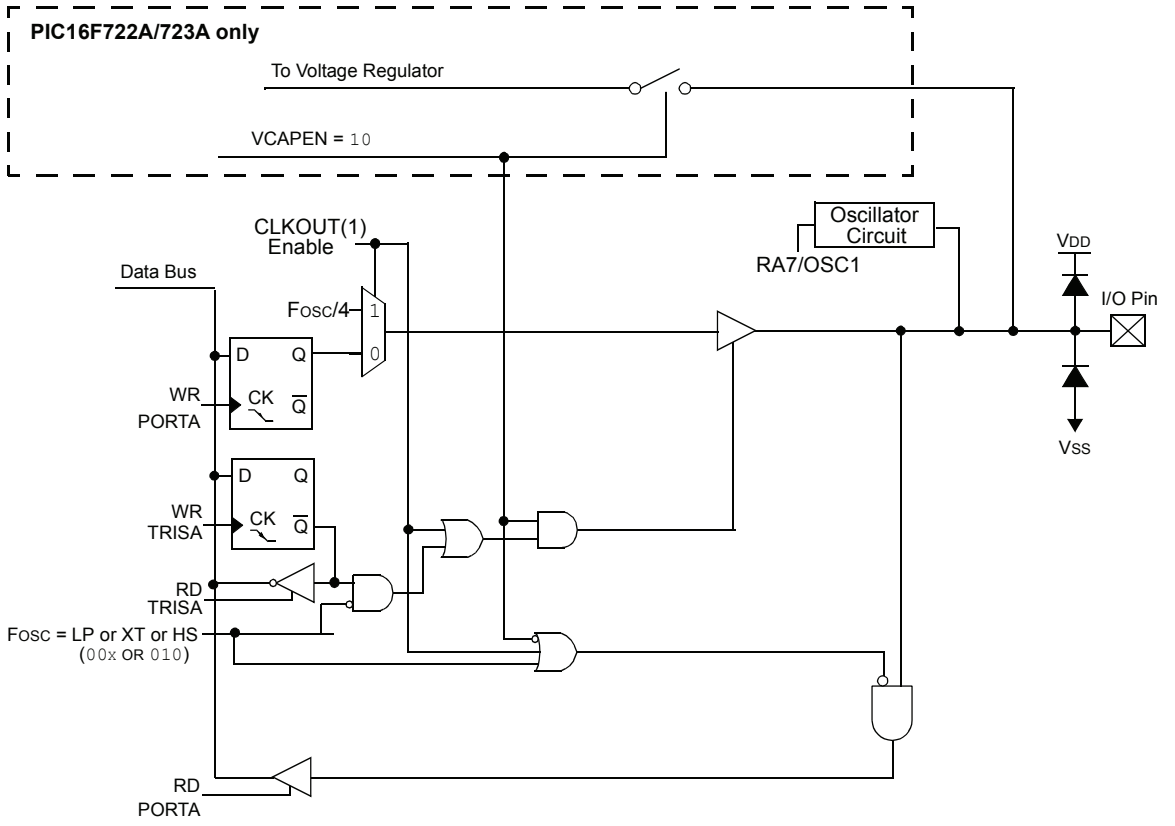


FIGURE 6-5: BLOCK DIAGRAM OF RA6



Note 1: CLKOUT Enable = 1 when FOSC = RC or INTOSC (No I/O Selected).

FIGURE 6-6: BLOCK DIAGRAM OF RA7

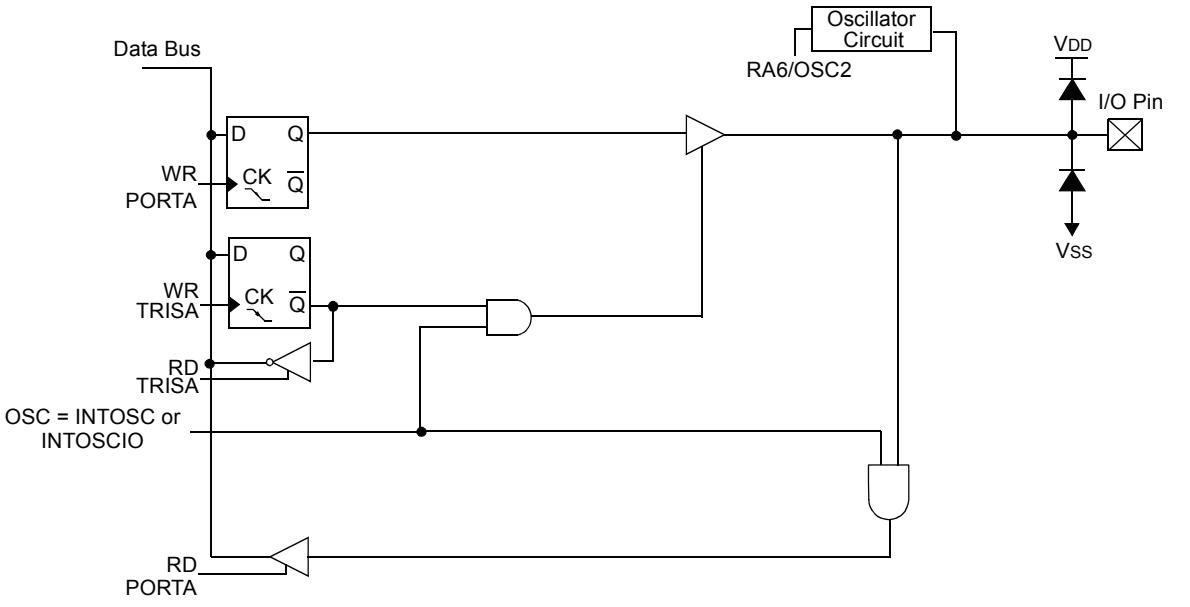


TABLE 6-1: SUMMARY OF REGISTERS ASSOCIATED WITH PORTA

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ADCON0	—	—	CHS3	CHS2	CHS1	CHS0	GO/$\overline{\text{DONE}}$	ADON	92
ADCON1	—	ADCS2	ADCS1	ADCS0	—	—	ADREF1	ADREF0	93
ANSELA	—	—	ANSA5	ANSA4	ANSA3	ANSA2	ANSA1	ANSA0	49
APFCON	—	—	—	—	—	—	SSSEL	CCP2SEL	47
CPSCON0	CPSON	—	—	—	CPSRNG1	CPSRNG0	CPSOUT	T0XCS	121
CPSCON1	—	—	—	—	CPSCH3	CPSCH2	CPSCH1	CPSCH0	122
OPTION_REG	$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	23
PORTA	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0	48
SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	162
TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	48

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

TABLE 6-2: SUMMARY OF CONFIGURATION WORD WITH CLOCK SOURCES

Name	Bits	Bit -/7	Bit -/6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	Register on Page
CONFIG2(1)	13:8	—	—	—	—	—	—	—	—	84
	7:0	—	—	VCAPEN1	VCAPEN0	WDTE	—	—	—	

Legend: — = unimplemented location, read as '0'. Shaded cells are not used by clock sources.

Note 1: PIC16F722A/723A only.

6.3 PORTB and TRISB Registers

PORTB is an 8-bit wide, bidirectional port. The corresponding data direction register is TRISB (Register 6-6). Setting a TRISB bit (= 1) will make the corresponding PORTB pin an input (i.e., put the corresponding output driver in a High-Impedance mode). Clearing a TRISB bit (= 0) will make the corresponding PORTB pin an output (i.e., enable the output driver and put the contents of the output latch on the selected pin). Example 6-2 shows how to initialize PORTB.

Reading the PORTB register (Register 6-5) reads the status of the pins, whereas writing to it will write to the PORT latch. All write operations are read-modify-write operations. Therefore, a write to a port implies that the port pins are read, this value is modified and then written to the PORT data latch.

The TRISB register (Register 6-6) controls the PORTB pin output drivers, even when they are being used as analog inputs. The user should ensure the bits in the TRISB register are maintained set when using them as analog inputs. I/O pins configured as analog input always read '0'. Example 6-2 shows how to initialize PORTB.

EXAMPLE 6-2: INITIALIZING PORTB

```
BANKSEL PORTB   ;
CLRF    PORTB   ;Init PORTB
BANKSEL ANSELB
CLRF    ANSELB  ;Make RB<7:0> digital
BANKSEL TRISB   ;
MOVLW   B'11110000';Set RB<7:4> as inputs
                ;and RB<3:0> as outputs
MOVWF   TRISB   ;
```

Note:	The ANSELB register must be initialized to configure an analog channel as a digital input. Pins configured as analog inputs will read '0'.

6.3.1 ANSELB REGISTER

The ANSELB register (Register 6-9) is used to configure the Input mode of an I/O pin to analog. Setting the appropriate ANSELB bit high will cause all digital reads on the pin to be read as '0' and allow analog functions on the pin to operate correctly.

The state of the ANSELB bits has no affect on digital output functions. A pin with TRIS clear and ANSELB set will still operate as a digital output, but the Input mode will be analog. This can cause unexpected behavior when executing read-modify-write instructions on the affected port.

6.3.2 WEAK PULL-UPS

Each of the PORTB pins has an individually configurable internal weak pull-up. Control bits WPUB<7:0> enable or disable each pull-up (see Register 6-7). Each weak pull-up is automatically turned off when the port pin is configured as an output. All pull-ups are disabled on a Power-on Reset by the $\overline{\text{RBPU}}$ bit of the OPTION register.

6.3.3 INTERRUPT-ON-CHANGE

All of the PORTB pins are individually configurable as an interrupt-on-change pin. Control bits IOCB<7:0> enable or disable the interrupt function for each pin. Refer to Register 6-8. The interrupt-on-change feature is disabled on a Power-on Reset.

For enabled interrupt-on-change pins, the present value is compared with the old value latched on the last read of PORTB to determine which bits have changed or mismatched the old value. The 'mismatch' outputs of the last read are OR'd together to set the PORTB Change Interrupt flag bit (RBIF) in the INTCON register.

This interrupt can wake the device from Sleep. The user, in the Interrupt Service Routine, clears the interrupt by:

a) Any read or write of PORTB. This will end the mismatch condition.
b) Clear the flag bit RBIF.

A mismatch condition will continue to set flag bit RBIF. Reading or writing PORTB will end the mismatch condition and allow flag bit RBIF to be cleared. The latch holding the last read value is not affected by a $\overline{\text{MCLR}}$ nor Brown-out Reset. After these Resets, the RBIF flag will continue to be set if a mismatch is present.

Note:	When a pin change occurs at the same time as a read operation on PORTB, the RBIF flag will always be set. If multiple PORTB pins are configured for the interrupt-on-change, the user may not be able to identify which pin changed state.

REGISTER 6-5: PORTB: PORTB REGISTER

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
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 7-0 **RB<7:0>**: PORTB I/O Pin bits
1 = Port pin is > VIH
0 = Port pin is < VIL

REGISTER 6-6: TRISB: PORTB TRI-STATE REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0
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 7-0 **TRISB<7:0>:** PORTB Tri-State Control bits
1 = PORTB pin configured as an input (tri-stated)
0 = PORTB pin configured as an output

REGISTER 6-7: WPUB: WEAK PULL-UP PORTB REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
WPUB7	WPUB6	WPUB5	WPUB4	WPUB3	WPUB2	WPUB1	WPUB0
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 7-0 **WPUB<7:0>**: Weak Pull-up Register bits
1 = Pull-up enabled
0 = Pull-up disabled

Note 1: Global $\overline{\text{RBPU}}$ bit of the OPTION register must be cleared for individual pull-ups to be enabled.
2: The weak pull-up device is automatically disabled if the pin is in configured as an output.

REGISTER 6-8: IOCB: INTERRUPT-ON-CHANGE PORTB REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
IOCB7	IOCB6	IOCB5	IOCB4	IOCB3	IOCB2	IOCB1	IOCB0
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 7-0 **IOCB<7:0>:** Interrupt-on-Change PORTB Control bits
1 = Interrupt-on-change enabled
0 = Interrupt-on-change disabled

REGISTER 6-9: ANSELB: PORTB ANALOG SELECT REGISTER

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0
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 **ANSB<5:0>**: Analog Select between Analog or Digital Function on Pins RB<5:0>, respectively
0 = Digital I/O. Pin is assigned to port or digital special function.
1 = Analog input. Pin is assigned as analog input(1). Digital input buffer disabled.

Note 1: When setting a pin to an analog input, the corresponding TRIS bit must be set to Input mode in order to allow external control of the voltage on the pin.

6.3.4 PIN DESCRIPTIONS AND DIAGRAMS

Each PORTB pin is multiplexed with other functions. The pins and their combined functions are briefly described here. For specific information about individual functions such as the SSP, I^2C or interrupts, refer to the appropriate section in this data sheet.

6.3.4.1 RB0/AN12/CPS0/INT

Figure 6-7 shows the diagram for this pin. This pin is configurable to function as one of the following:

- General purpose I/O
- Analog input for the ADC
- Capacitive sensing input
- External edge triggered interrupt

6.3.4.2 RB1/AN10/CPS1

Figure 6-8 shows the diagram for this pin. This pin is configurable to function as one of the following:

- General purpose I/O
- Analog input for the ADC
- Capacitive sensing input

6.3.4.3 RB2/AN8/CPS2

Figure 6-8 shows the diagram for this pin. This pin is configurable to function as one of the following:

- General purpose I/O
- Analog input for the ADC
- Capacitive sensing input

6.3.4.4 RB3/AN9/CPS3/CCP2

Figure 6-9 shows the diagram for this pin. This pin is configurable to function as one of the following:

- General purpose I/O
- Analog input for the ADC
- Capacitive sensing input
- Capture 2 input, Compare 2 output, and PWM2 output

Note:	CCP2 pin location may be selected as RB3 or RC1.

6.3.4.5 RB4/AN11/CPS4

Figure 6-8 shows the diagram for this pin. This pin is configurable to function as one of the following:

- General purpose I/O
- Analog input for the ADC
- Capacitive sensing input

6.3.4.6 RB5/AN13/CPS5/T1G

Figure 6-10 shows the diagram for this pin. This pin is configurable to function as one of the following:

- General purpose I/O
- Analog input for the ADC
- Capacitive sensing input
- Timer1 gate input

6.3.4.7 RB6/ICSPCLK

Figure 6-11 shows the diagram for this pin. This pin is configurable to function as one of the following:

- a general purpose I/O
- In-Circuit Serial Programming clock

6.3.4.8 RB7/ICSPDAT

Figure 6-12 shows the diagram for this pin. This pin is configurable to function as one of the following:

- General purpose I/O
- In-Circuit Serial Programming data

FIGURE 6-7: BLOCK DIAGRAM OF RB0

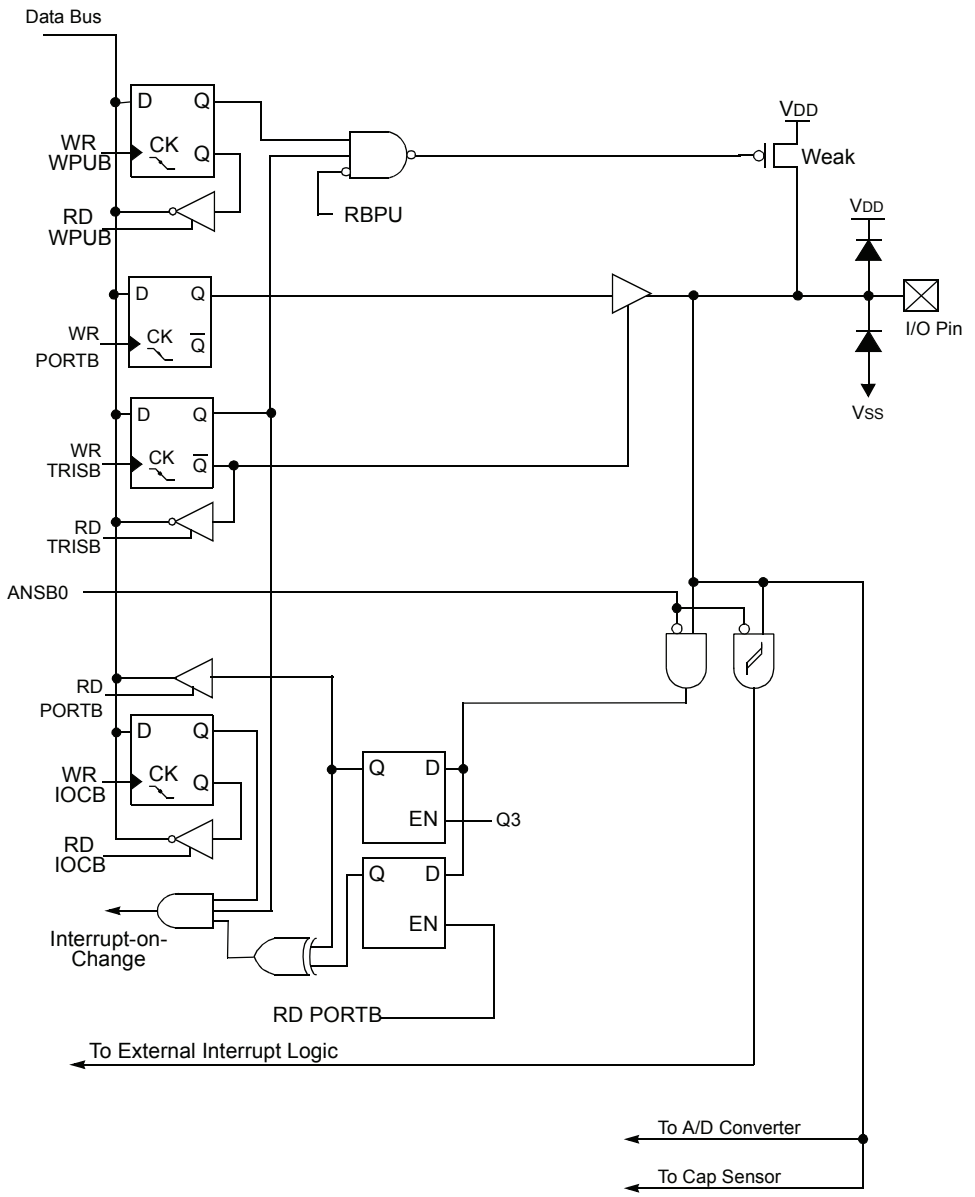


FIGURE 6-8: BLOCK DIAGRAM OF RB4, RB<2:1>

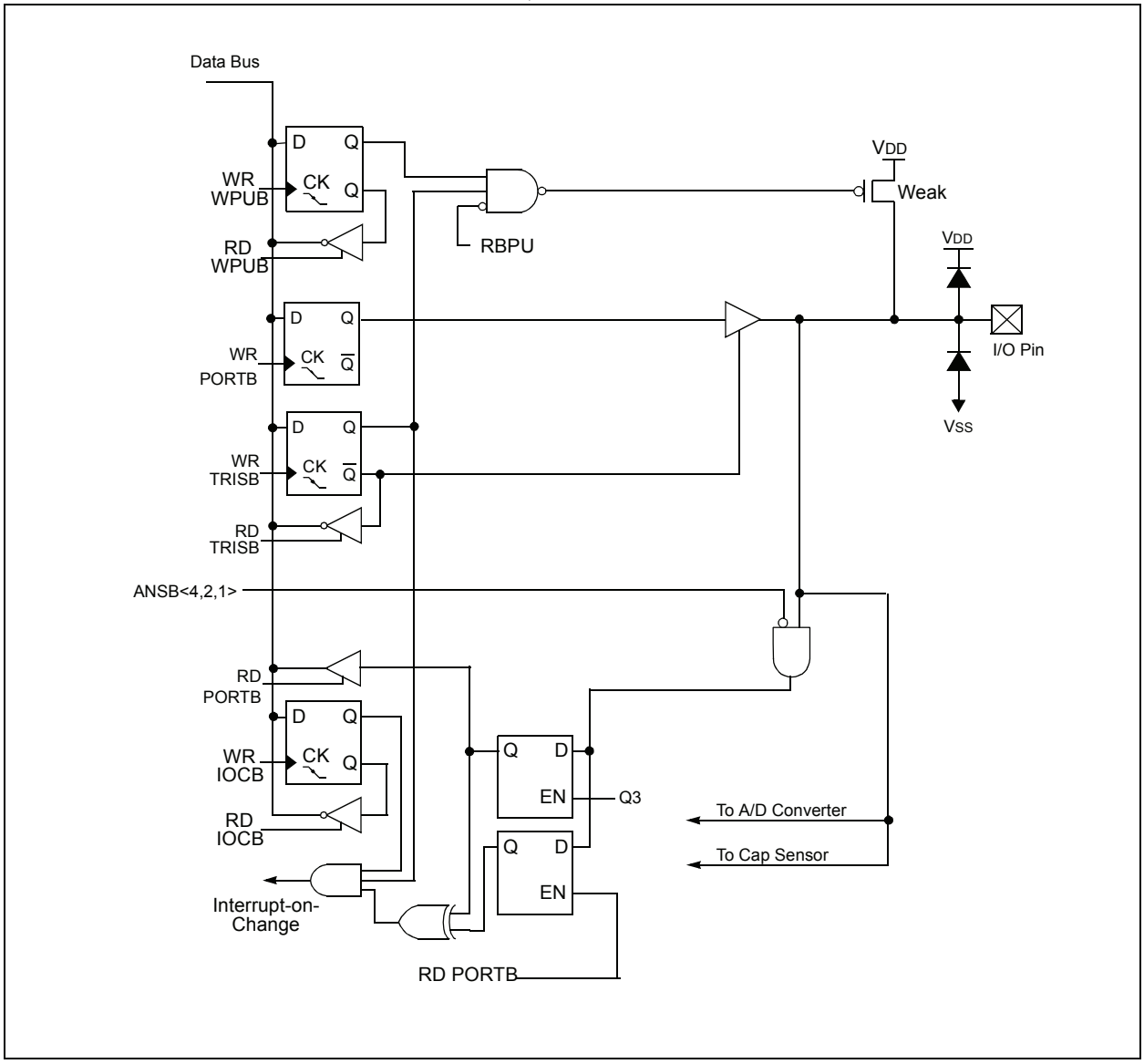
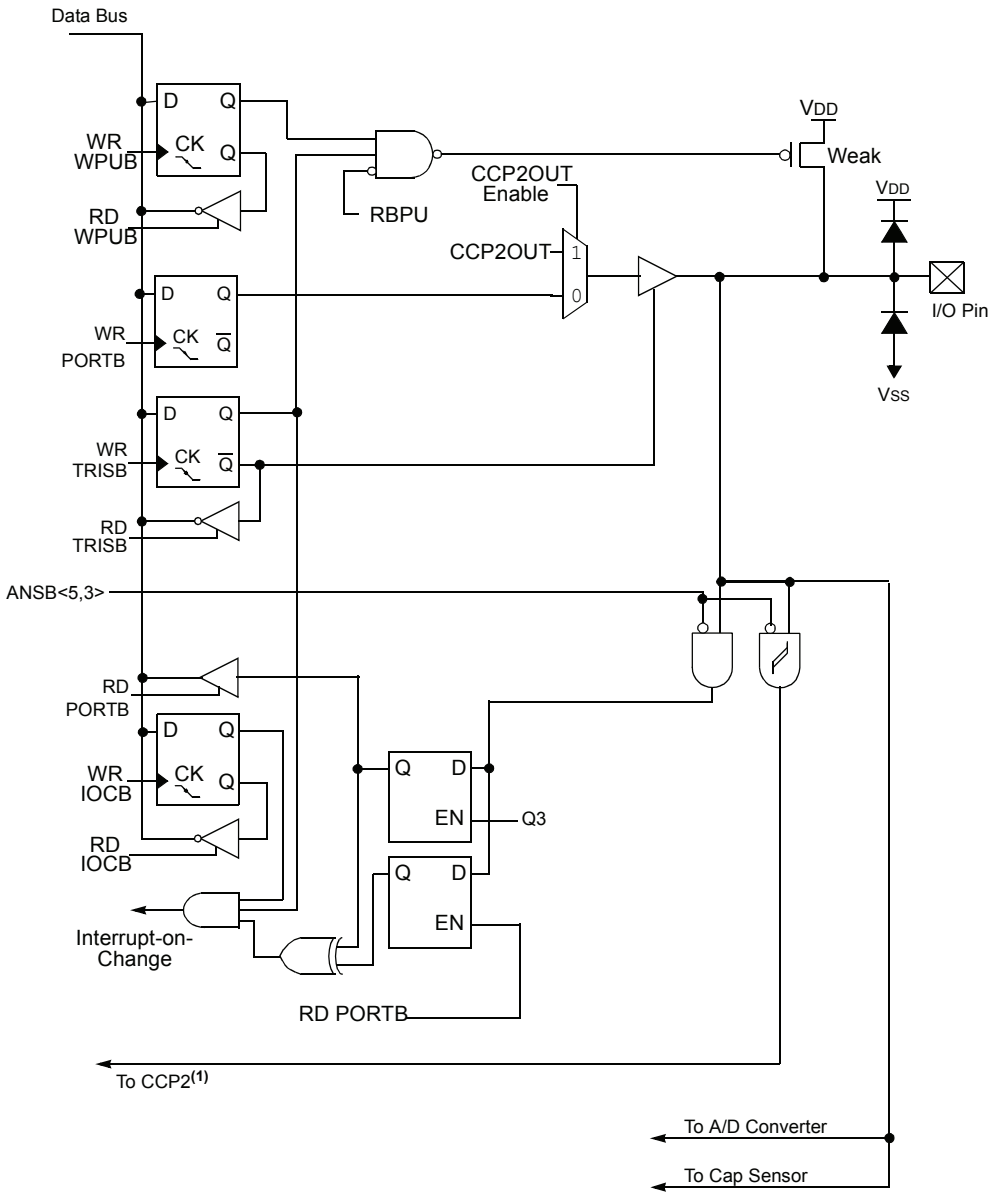


FIGURE 6-9: BLOCK DIAGRAM OF RB3



Note 1: CCP2 input is controlled by CCP2SEL in the APFCON register.

FIGURE 6-10: BLOCK DIAGRAM OF RB5

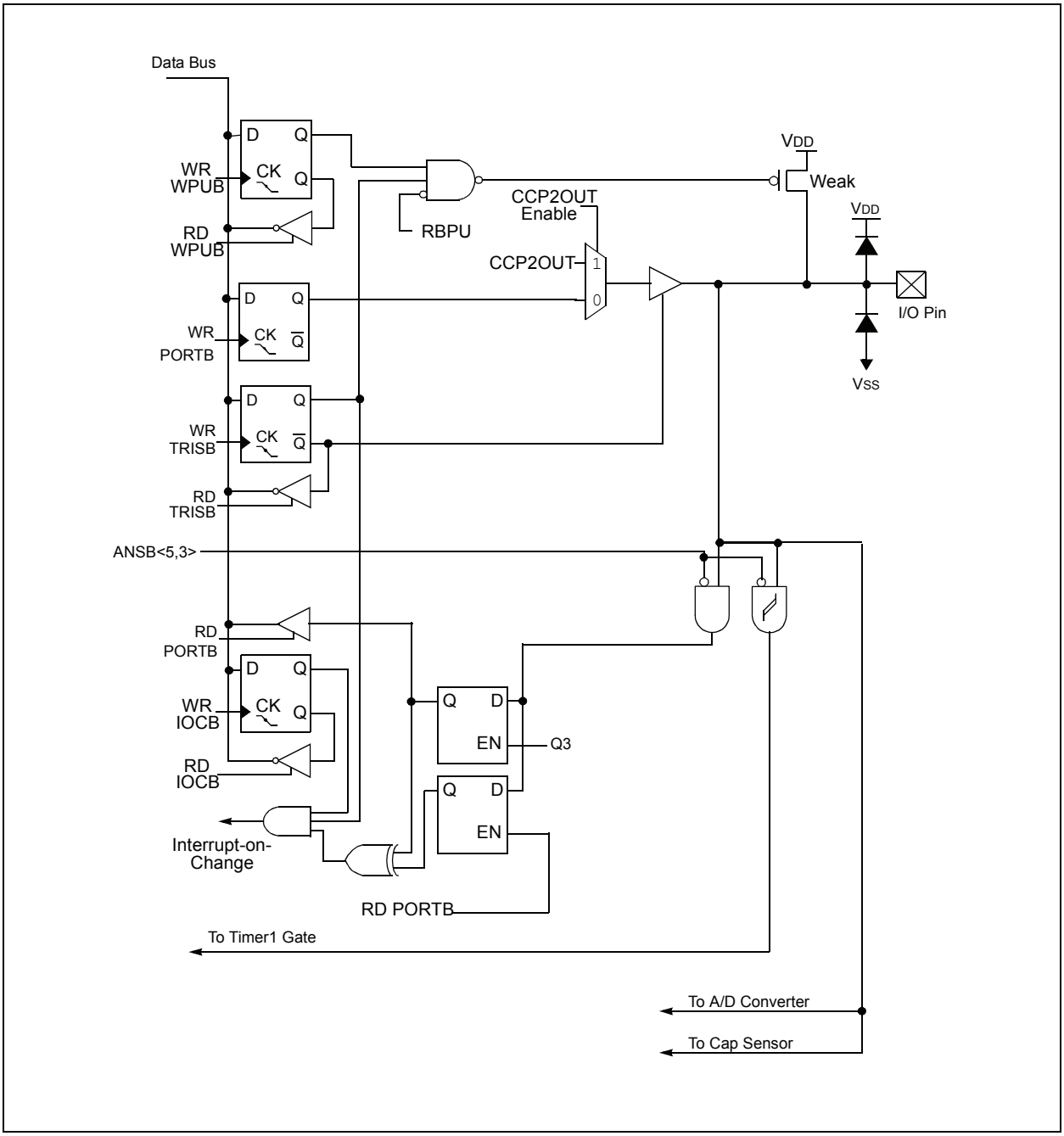


FIGURE 6-11: BLOCK DIAGRAM OF RB6

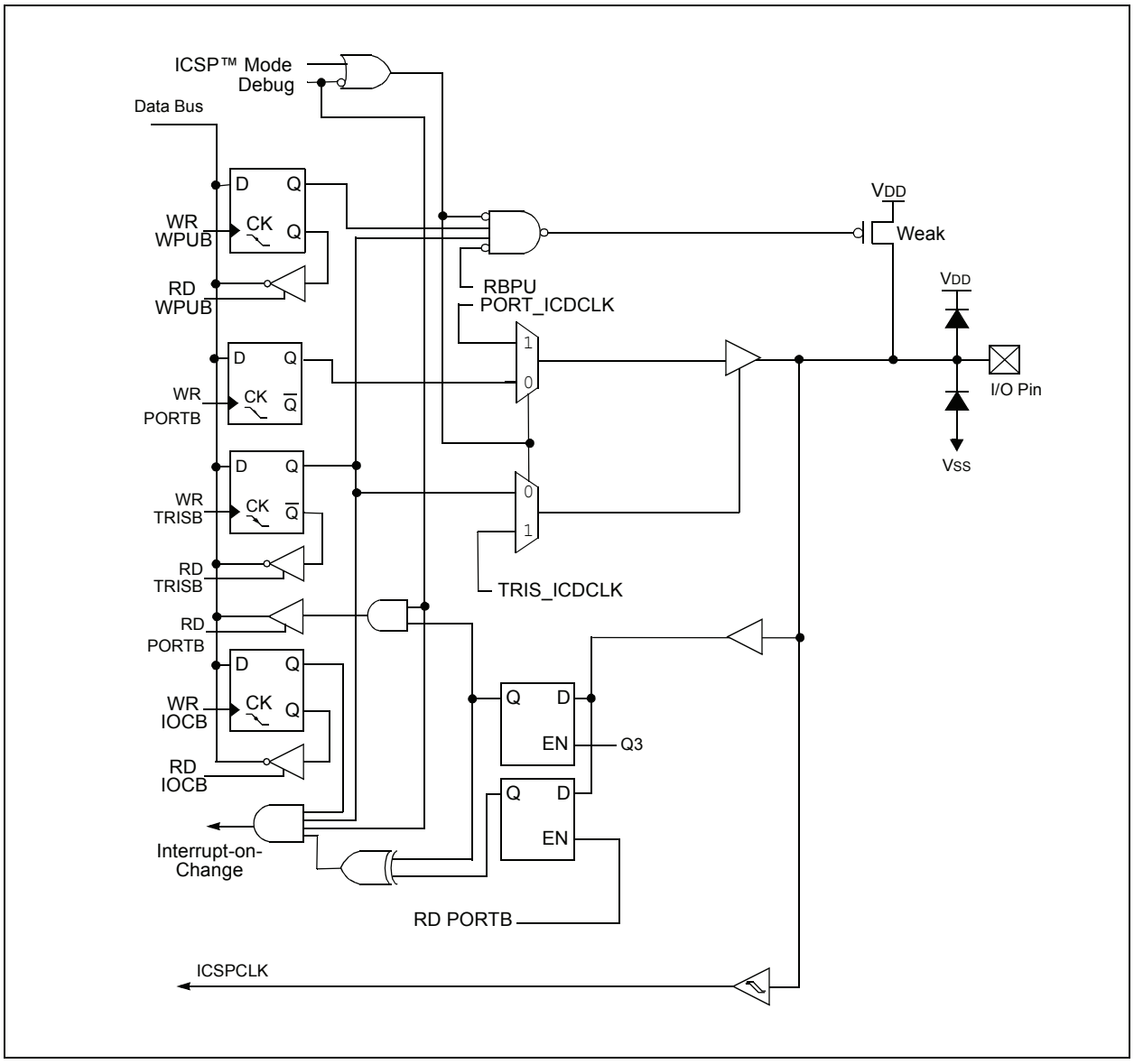


FIGURE 6-12: BLOCK DIAGRAM OF RB7

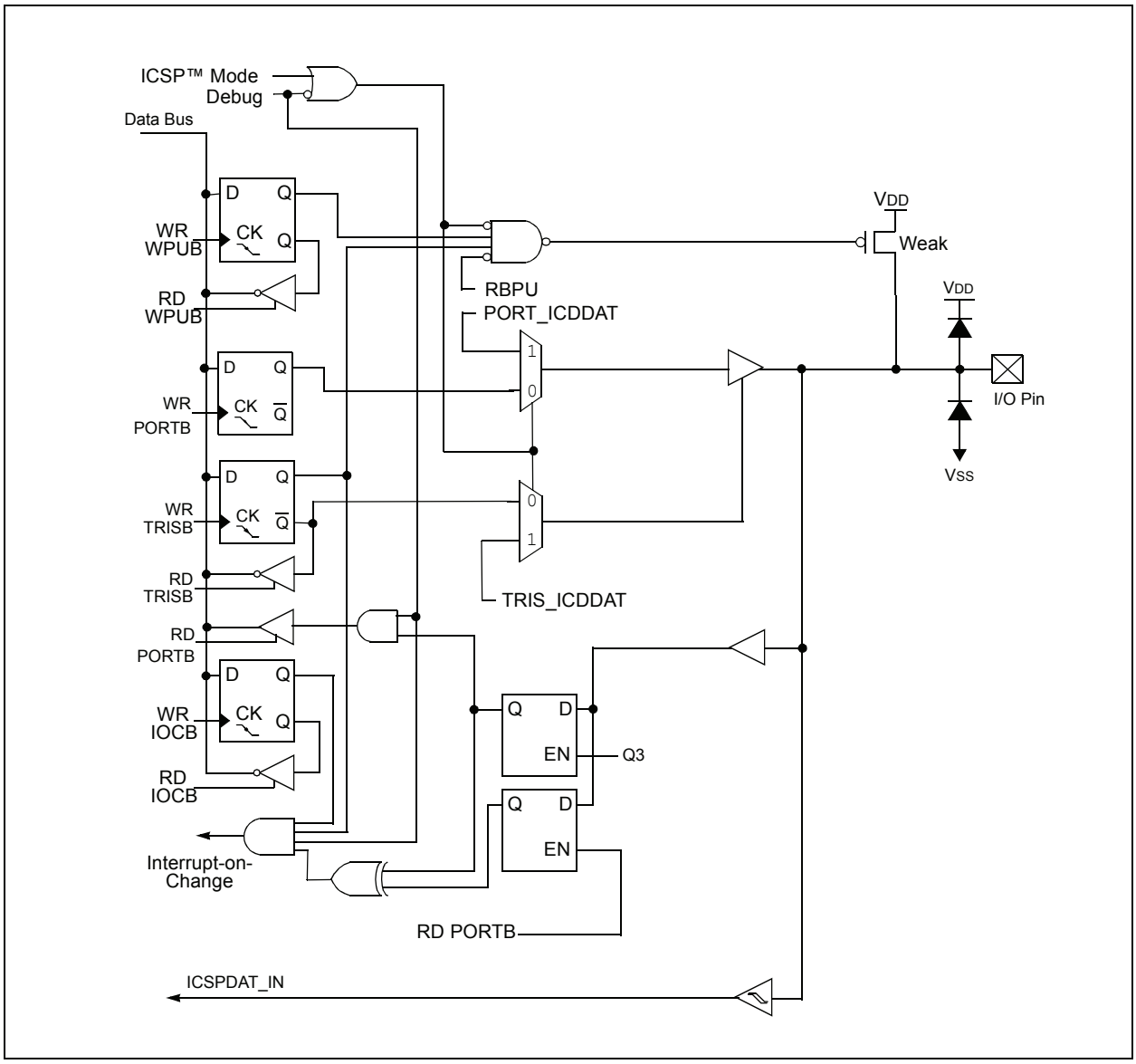


TABLE 6-3: SUMMARY OF REGISTERS ASSOCIATED WITH PORTB

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ADCON0	—	—	CHS3	CHS2	CHS1	CHS0	GO/$\overline{\text{DONE}}$	ADON	92
ANSELB	—	—	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0	58
APFCON	—	—	—	—	—	—	SSSEL	CCP2SEL	47
CCP2CON	—	—	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	124
CPSCON0	CPSON	—	—	—	CPSRNG1	CPSRNG0	CPSOUT	T0XCS	121
CPSCON1	—	—	—	—	CPSCH3	CPSCH2	CPSCH1	CPSCH0	122
INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	40
IOCB	IOCB7	IOCB6	IOCB5	IOCB4	IOCB3	IOCB2	IOCB1	IOCB0	58
OPTION_REG	$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	23
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	57
T1GCON	TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/$\overline{\text{DONE}}$	T1GVAL	T1GSS1	T1GSS0	113
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	57
WPUB	WPUB7	WPUB6	WPUB5	WPUB4	WPUB3	WPUB2	WPUB1	WPUB0	57

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

6.4 PORTC and TRISC Registers

PORTC is a 8-bit wide, bidirectional port. The corresponding data direction register is TRISC (Register 6-11). Setting a TRISC bit (= 1) will make the corresponding PORTC pin an input (i.e., put the corresponding output driver in a High-Impedance mode). Clearing a TRISC bit (= 0) will make the corresponding PORTC pin an output (i.e., enable the output driver and put the contents of the output latch on the selected pin). Example 6-3 shows how to initialize PORTC.

Reading the PORTC register (Register 6-10) reads the status of the pins, whereas writing to it will write to the PORT latch. All write operations are read-modify-write operations. Therefore, a write to a port implies that the port pins are read, this value is modified and then written to the PORT data latch.

The TRISC register (Register 6-11) controls the PORTC pin output drivers, even when they are being used as analog inputs. The user should ensure the bits in the TRISC register are maintained set when using them as analog inputs. I/O pins configured as analog input always read '0'.

EXAMPLE 6-3: INITIALIZING PORTC

```
BANKSEL PORTC          ;
CLRF    PORTC          ;Init PORTC
BANKSEL TRISC          ;
MOVLW   B'00001100'    ;Set RC<3:2> as inputs
MOVWF   TRISC          ;and set RC<7:4,1:0>
                       ;as outputs
```

The location of the CCP2 function is controlled by the CCP2SEL bit in the APFCON register (refer to Register 6-1).

REGISTER 6-10: PORTC: PORTC REGISTER

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0
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 7-0 **RC<7:0>**: PORTC General Purpose I/O Pin bits
1 = Port pin is > VIH
0 = Port pin is < VIL

REGISTER 6-11: TRISC: PORTC TRI-STATE REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0
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 7-0 **TRISC<7:0>:** PORTC Tri-State Control bits
1 = PORTC pin configured as an input (tri-stated)
0 = PORTC pin configured as an output

6.4.1 RC0/T1OSO/T1CKI

Figure 6-13 shows the diagram for this pin. This pin is configurable to function as one of the following:

- General purpose I/O
- Timer1 oscillator output
- Timer1 clock input

6.4.2 RC1/T1OSI/CCP2

Figure 6-14 shows the diagram for this pin. This pin is configurable to function as one of the following:

- General purpose I/O
- Timer1 oscillator input
- Capture 2 input, Compare 2 output, and PWM2 output

Note:	CCP2 pin location may be selected as RB3 or RC1.

6.4.3 RC2/CCP1

Figure 6-15 shows the diagram for this pin. This pin is configurable to function as one of the following:

- General purpose I/O
- Capture 1 input, Compare 1 output, and PWM1 output

6.4.4 RC3/SCK/SCL

Figure 6-16 shows the diagram for this pin. This pin is configurable to function as one of the following:

- General purpose I/O
- SPI clock
- I^2C™ clock

6.4.5 RC4/SDI/SDA

Figure 6-17 shows the diagram for this pin. This pin is configurable to function as one of the following:

- General purpose I/O
- SPI data input
- I^2C data I/O

6.4.6 RC5/SDO

Figure 6-18 shows the diagram for this pin. This pin is configurable to function as one of the following:

- General purpose I/O
- SPI data output

6.4.7 RC6/TX/CK

Figure 6-19 shows the diagram for this pin. This pin is configurable to function as one of the following:

- General purpose I/O
- Asynchronous serial output
- Synchronous clock I/O

6.4.8 RC7/RX/DT

Figure 6-20 shows the diagram for this pin. This pin is configurable to function as one of the following:

- General purpose I/O
- Asynchronous serial input
- Synchronous serial data I/O

FIGURE 6-13: BLOCK DIAGRAM OF RC0

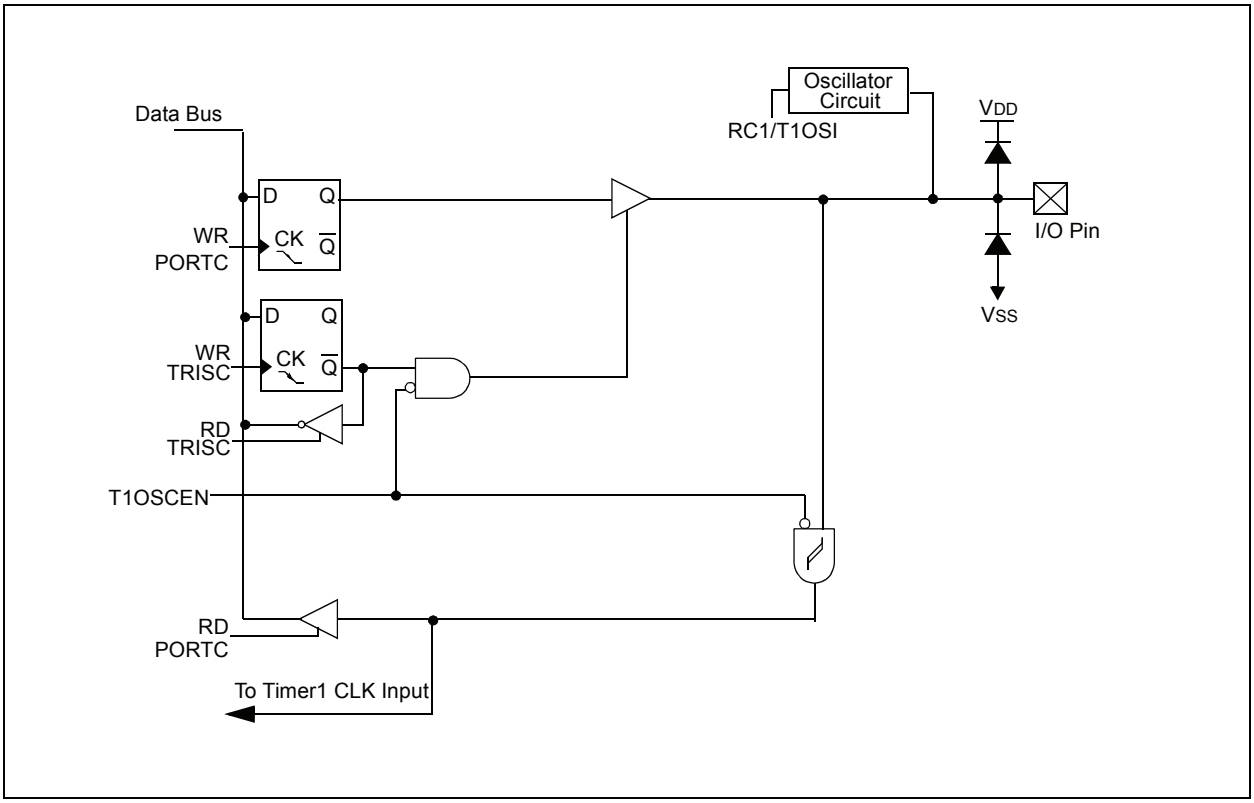
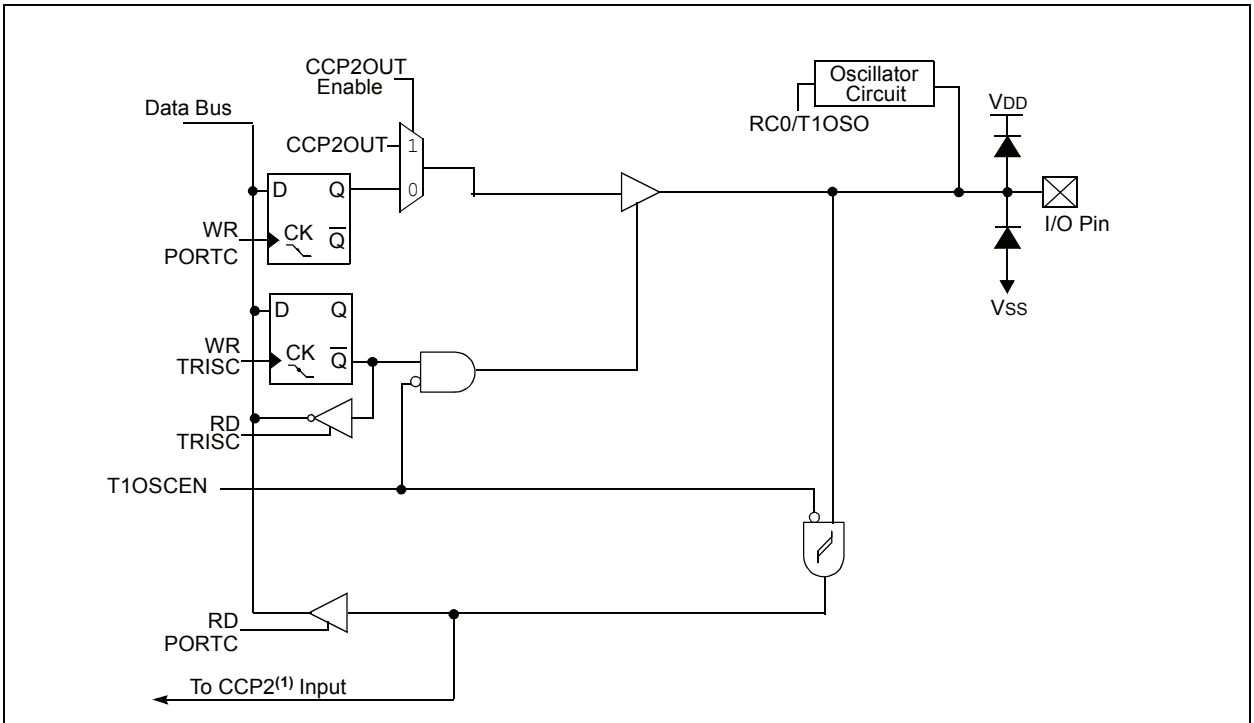


FIGURE 6-14: BLOCK DIAGRAM OF RC1



Note 1: CCP2 input is controlled by CCP2SEL in the APFCON register.

FIGURE 6-15: BLOCK DIAGRAM OF RC2

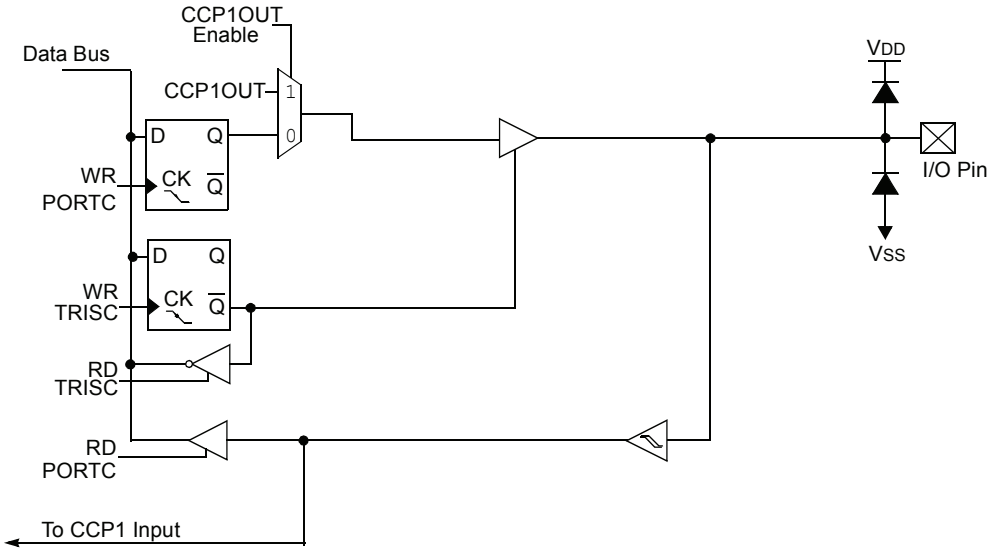
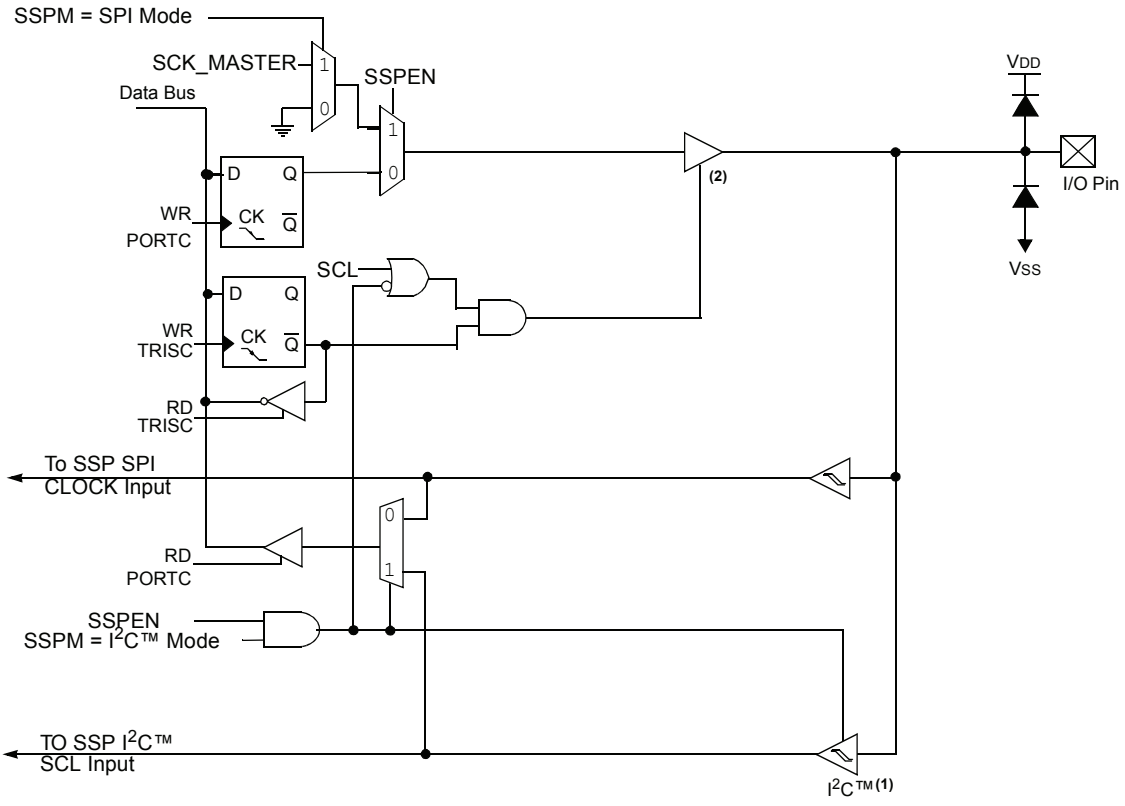
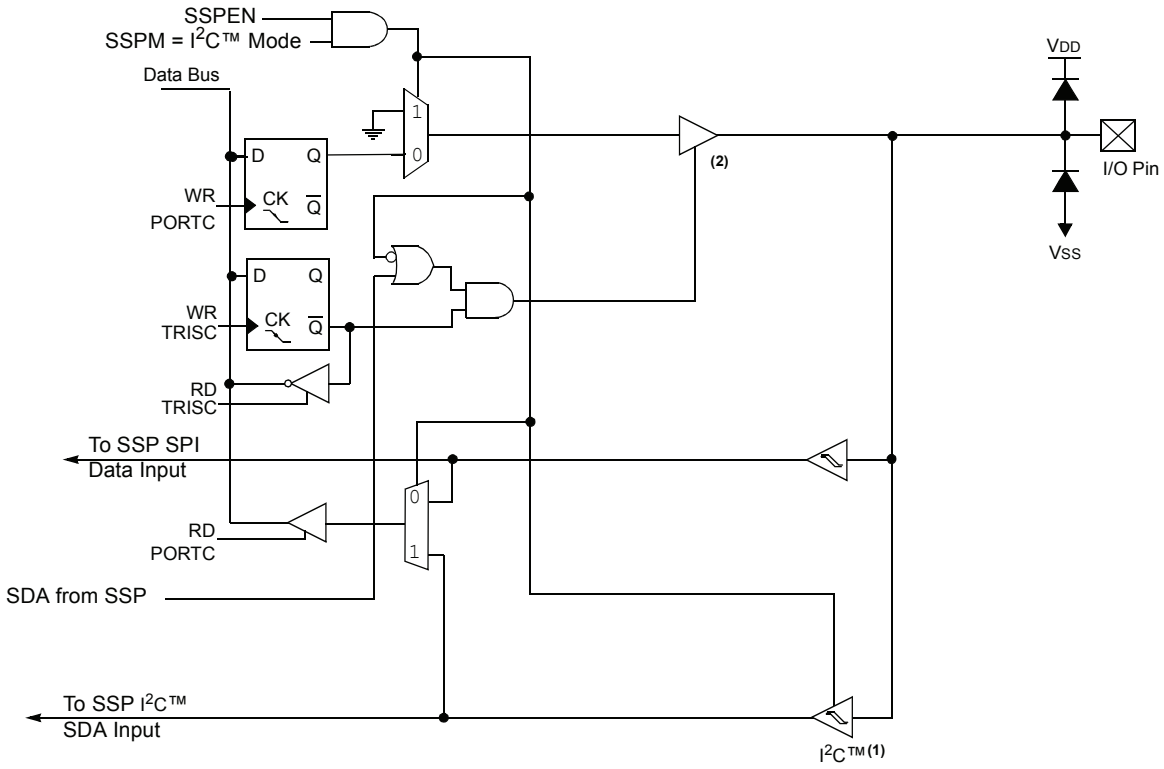


FIGURE 6-16: BLOCK DIAGRAM OF RC3



Note 1: I^2C™ Schmitt Trigger has special input levels.
2: I^2C™ Slew Rate limiting controlled by SMP bit of SSPSTAT register.

FIGURE 6-17: BLOCK DIAGRAM OF RC4



Note 1: I²C™ Schmitt Trigger has special input levels.
2: I²C™ Slew Rate limiting controlled by SMP bit of SSPSTAT register.

FIGURE 6-18: BLOCK DIAGRAM OF RC5

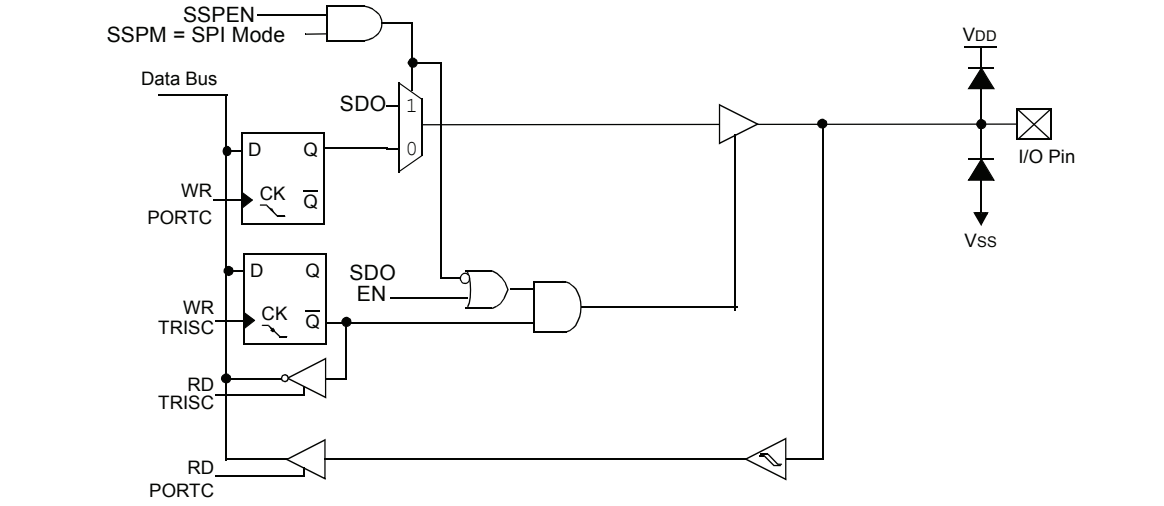


FIGURE 6-19: BLOCK DIAGRAM OF RC6

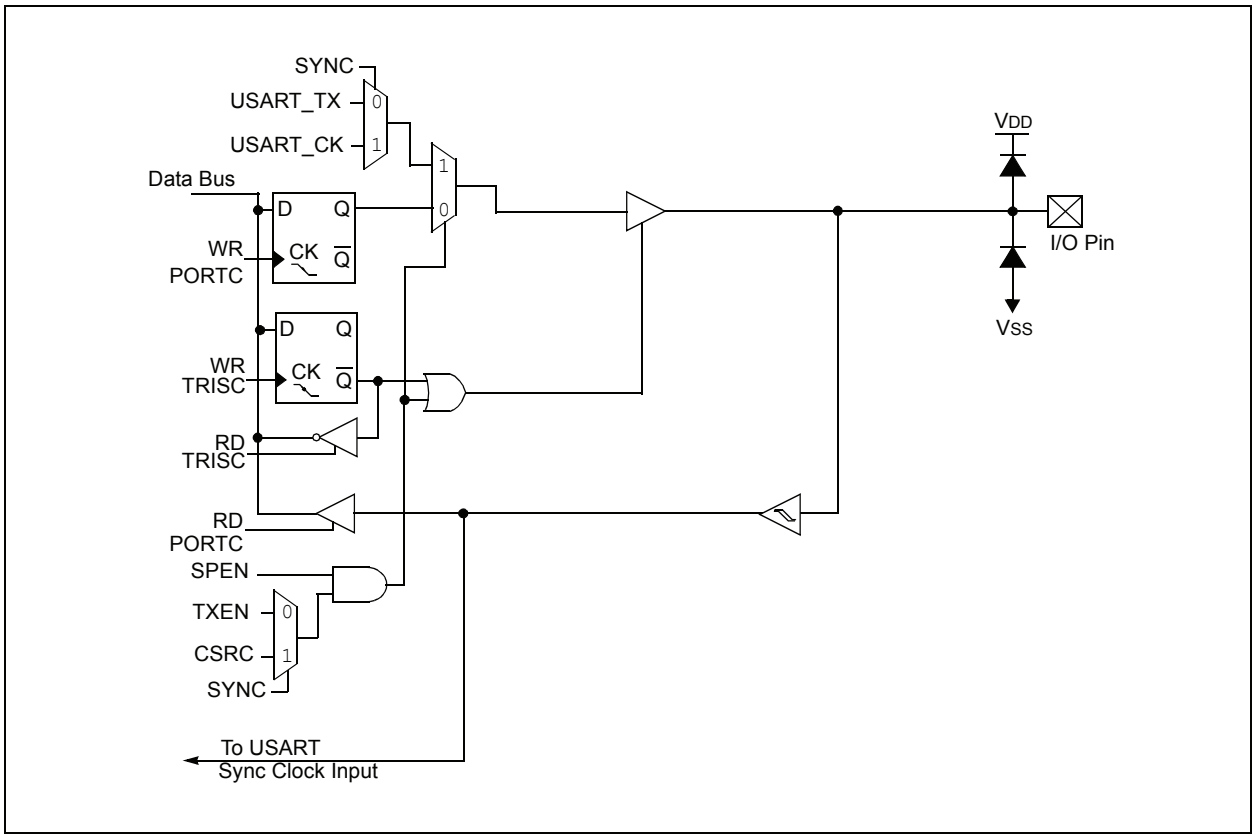


FIGURE 6-20: BLOCK DIAGRAM OF RC7

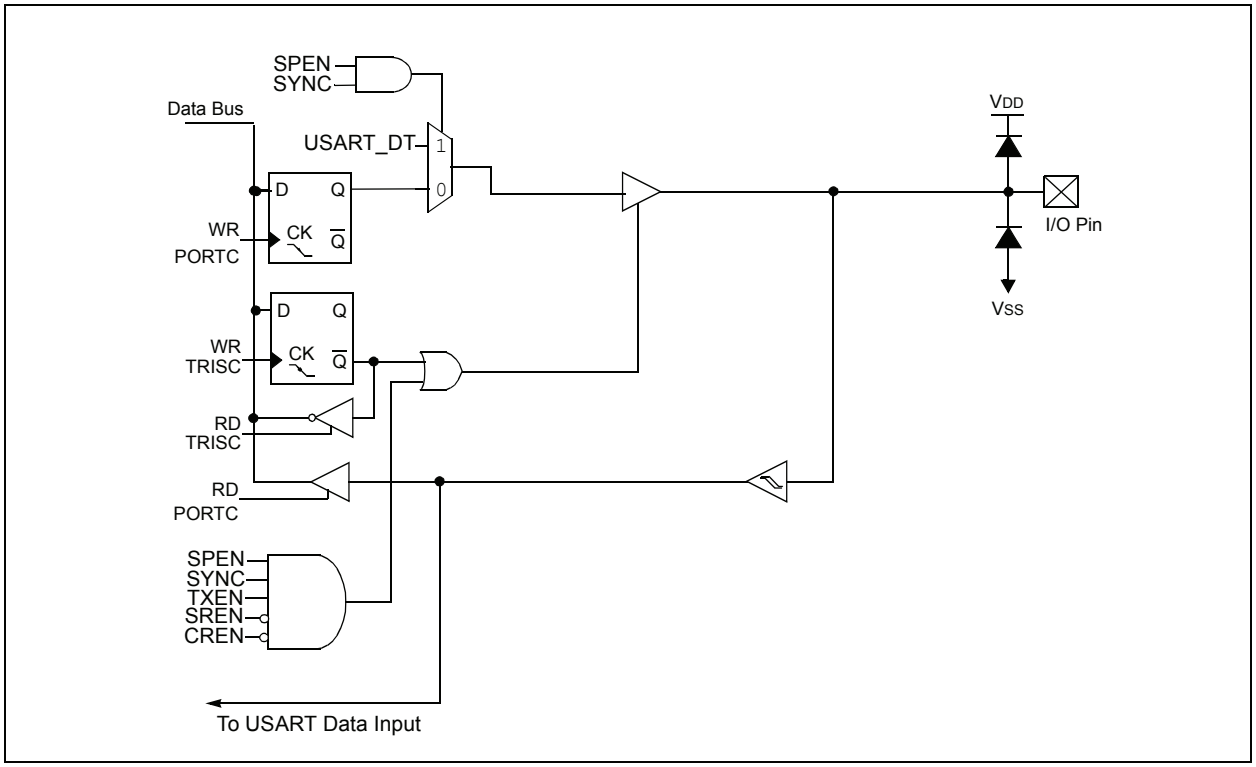


TABLE 6-4: SUMMARY OF REGISTERS ASSOCIATED WITH PORTC

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
APFCON	—	—	—	—	—	—	SSSEL	CCP2SEL	47
CCP1CON	—	—	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	124
CCP2CON	—	—	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	124
PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	67
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	143
SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	162
SSPSTAT	SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF	163
T1CON	TMR1CS1	TMR1CS0	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{T1SYNC}$	—	TMR1ON	112
TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	142
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	67

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

6.5 PORTE and TRISE Registers

PORTE(1) is an 1-bit wide, input only port. RE3 is input only and its TRIS bit will always read as '1'.

Reading the PORTE register (Register 6-12) reads the status of the pins. RE3 reads '0' when MCLRE = 1.

REGISTER 6-12: PORTE: PORTE REGISTER

U-0	U-0	U-0	U-0	R-x	U-0	U-0	U-0
—	—	—	—	RE3	—	—	—
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 7-4 **Unimplemented**: Read as '0'

bit 3 **RE3**: PORTE I/O Pin bits(1)
1 = Port pin is > VIH
0 = Port pin is < VIL

bit 2-0 **Unimplemented**: Read as '0'

REGISTER 6-13: TRISE: PORTE TRI-STATE REGISTER

U-0	U-0	U-0	U-0	R-1	U-0	U-0	U-0
—	—	—	—	TRISE3	—	—	—
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 7-4 **Unimplemented**: Read as '0'

bit 3 **TRISE3:** RE3 Port Tri-state Control bit
This bit is always '1' as RE3 is an input only

bit 2-0 **Unimplemented**: Read as '0'

TABLE 6-5: SUMMARY OF REGISTERS ASSOCIATED WITH PORTE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
PORTE	—	—	—	—	RE3	—	—	—	74
TRISE	—	—	—	—	TRISE3(1)	—	—	—	74

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

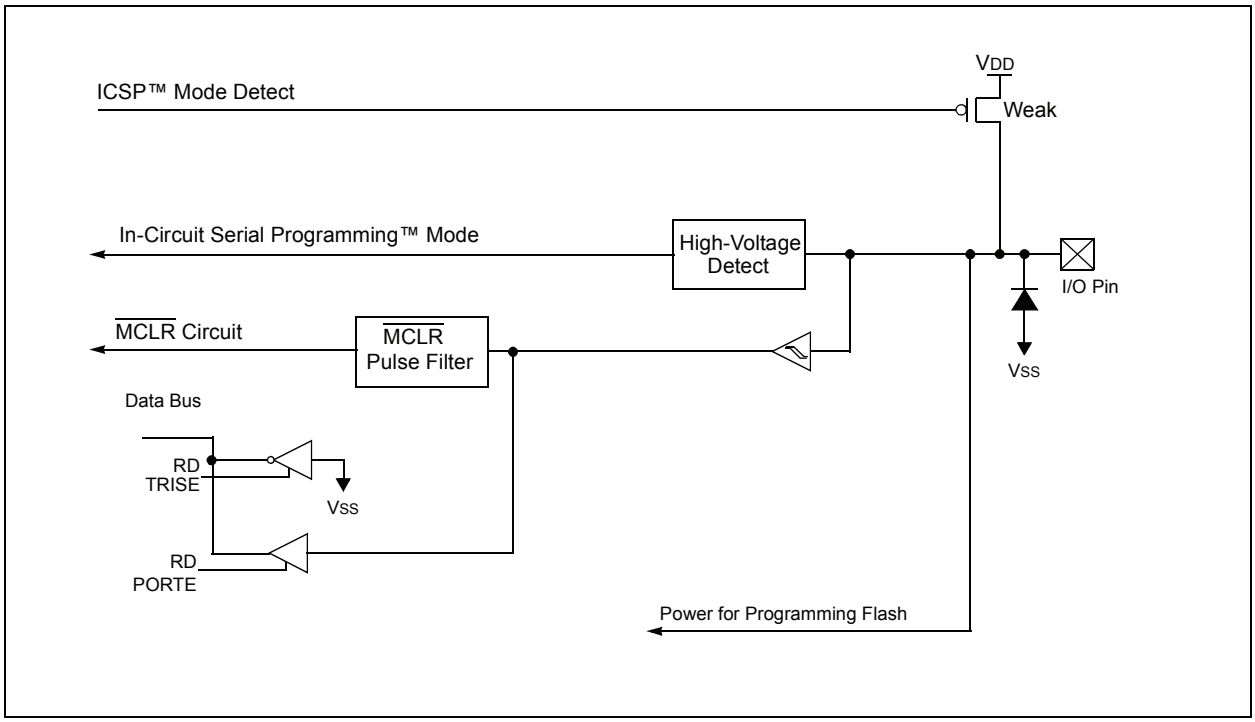
Note 1: This bit is always '1' as RE3 is input only.

6.5.1 RE3/$\overline{MCLR}$/VPP

Figure 6-21 shows the diagram for this pin. This pin is configurable to function as one of the following:

- General purpose input
- Master Clear Reset with weak pull-up
- Programming voltage reference input

FIGURE 6-21: BLOCK DIAGRAM OF RE3



NOTES:

7.0 OSCILLATOR MODULE

7.1 Overview

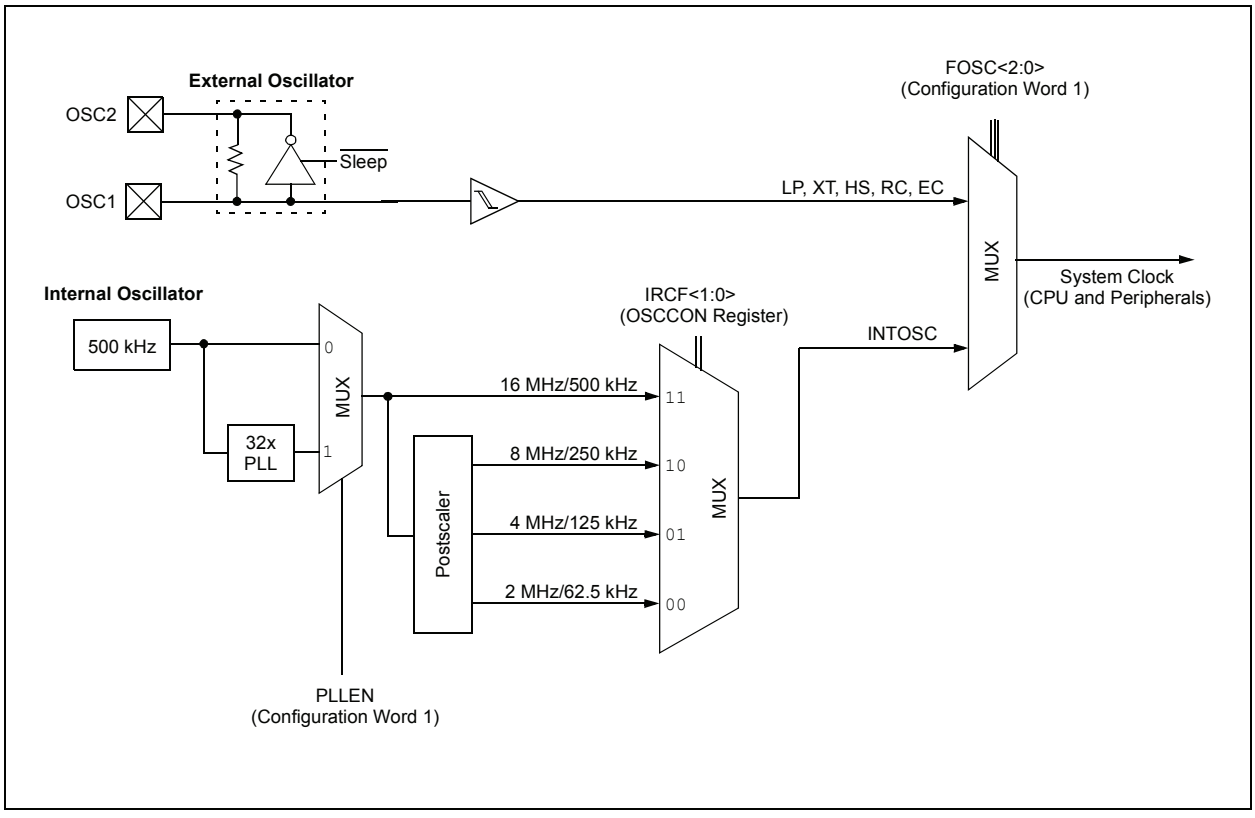
The oscillator module has a wide variety of clock sources and selection features that allow it to be used in a wide range of applications while maximizing performance and minimizing power consumption. Figure 7-1 illustrates a block diagram of the oscillator module.

Clock sources can be configured from external oscillators, quartz crystal resonators, ceramic resonators and Resistor-Capacitor (RC) circuits. In addition, the system can be configured to use an internal calibrated high-frequency oscillator as clock source, with a choice of selectable speeds via software.

Clock source modes are configured by the FOSC bits in Configuration Word 1 (CONFIG1). The oscillator module can be configured for one of eight modes of operation.

1. RC – External Resistor-Capacitor (RC) with FOSC/4 output on OSC2/CLKOUT.
2. RCIO – External Resistor-Capacitor (RC) with I/O on OSC2/CLKOUT.
3. INTOSC – Internal oscillator with FOSC/4 output on OSC2 and I/O on OSC1/CLKIN.
4. INTOSCIO – Internal oscillator with I/O on OSC1/CLKIN and OSC2/CLKOUT.
5. EC – External clock with I/O on OSC2/CLKOUT.
6. HS – High Gain Crystal or Ceramic Resonator mode.
7. XT – Medium Gain Crystal or Ceramic Resonator Oscillator mode.
8. LP – Low-Power Crystal mode.

FIGURE 7-1: SIMPLIFIED PIC® MCU CLOCK SOURCE BLOCK DIAGRAM



7.2 Clock Source Modes

Clock source modes can be classified as external or internal.

- Internal clock source (INTOSC) is contained within the oscillator module and derived from a 500 kHz high precision oscillator. The oscillator module has eight selectable output frequencies, with a maximum internal frequency of 16 MHz.
- External clock modes rely on external circuitry for the clock source. Examples are: oscillator modules (EC mode), quartz crystal resonators or ceramic resonators (LP, XT and HS modes) and Resistor-Capacitor (RC) mode circuits.

The system clock can be selected between external or internal clock sources via the FOSC bits of the Configuration Word 1.

7.3 Internal Clock Modes

The oscillator module has eight output frequencies derived from a 500 kHz high precision oscillator. The IRCF bits of the OSCCON register select the postscaler applied to the clock source dividing the frequency by 1, 2, 4 or 8. Setting the PLLEN bit of the Configuration Word 1 locks the internal clock source to 16 MHz before the postscaler is selected by the IRCF bits. The PLLEN bit must be set or cleared at the time of programming; therefore, only the upper or low four clock source frequencies are selectable in software.

7.3.1 INTOSC AND INTOSCIO MODES

The INTOSC and INTOSCIO modes configure the internal oscillators as the system clock source when the device is programmed using the oscillator selection or the FOSC<2:0> bits in the CONFIG1 register. See **Section 8.0 "Device Configuration"** for more information.

In INTOSC mode, OSC1/CLKIN is available for general purpose I/O. OSC2/CLKOUT outputs the selected internal oscillator frequency divided by 4. The CLKOUT signal may be used to provide a clock for external circuitry, synchronization, calibration, test or other application requirements.

In INTOSCIO mode, OSC1/CLKIN and OSC2/CLKOUT are available for general purpose I/O.

7.3.2 FREQUENCY SELECT BITS (IRCF)

The output of the 500 kHz INTOSC and 16 MHz INTOSC, with Phase Locked Loop enabled, connect to a postscaler and multiplexer (see Figure 7-1). The Internal Oscillator Frequency Select bits (IRCF) of the OSCCON register select the frequency output of the internal oscillator. Depending upon the PLLEN bit, one of four frequencies of two frequency sets can be selected via software:

If PLLEN = 1, frequency selection is as follows:

- 16 MHz
- 8 MHz (Default after Reset)
- 4 MHz
- 2 MHz

If PLLEN = 0, frequency selection is as follows:

- 500 kHz
- 250 kHz (Default after Reset)
- 125 kHz
- 62.5 kHz

Note:	Following any Reset, the IRCF<1:0> bits of the OSCCON register are set to '10' and the frequency selection is set to 8 MHz or 250 kHz. The user can modify the IRCF bits to select a different frequency.

There is no start-up delay before a new frequency selected in the IRCF bits takes effect. This is because the old and new frequencies are derived from INTOSC via the postscaler and multiplexer.

Start-up delay specifications are located in Table 23-2 in **Section 23.0 "Electrical Specifications"**.

7.4 Oscillator Control

The Oscillator Control (OSCCON) register (Figure 7-1) displays the status and allows frequency selection of the internal oscillator (INTOSC) system clock. The OSCCON register contains the following bits:

- Frequency selection bits (IRCF)
- Status Locked bits (ICSL)
- Status Stable bits (ICSS)

REGISTER 7-1: OSCCON: OSCILLATOR CONTROL REGISTER

U-0	U-0	R/W-1	R/W-0	R-q	R-q	U-0	U-0
—	—	IRCF1	IRCF0	ICSL	ICSS	—	—
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
q = Value depends on condition			

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

bit 5-4 **IRCF<1:0>:** Internal Oscillator Frequency Select bits
When PLLEN = 1 (16 MHz INTOSC)
11 = 16 MHz
10 = 8 MHz (POR value)
01 = 4 MHz
00 = 2 MHz
When PLLEN = 0 (500 kHz INTOSC)
11 = 500 kHz
10 = 250 kHz (POR value)
01 = 125 kHz
00 = 62.5 kHz

bit 3 **ICSL:** Internal Clock Oscillator Status Locked bit (2% Stable)
1 = 16 MHz/500 kHz Internal Oscillator (HFIOSC) is in lock
0 = 16 MHz/500 kHz Internal Oscillator (HFIOSC) has not yet locked

bit 2 **ICSS:** Internal Clock Oscillator Status Stable bit (0.5% Stable)
1 = 16 MHz/500 kHz Internal Oscillator (HFIOSC) has stabilized to its maximum accuracy
0 = 16 MHz/500 kHz Internal Oscillator (HFIOSC) has not yet reached its maximum accuracy

bit 1-0 **Unimplemented:** Read as '0'

7.5 Oscillator Tuning

The INTOSC is factory calibrated but can be adjusted in software by writing to the OSCTUNE register (Register 7-2).

The default value of the OSCTUNE register is '0'. The value is a 6-bit two's complement number.

When the OSCTUNE register is modified, the INTOSC frequency will begin shifting to the new frequency. Code execution continues during this shift. There is no indication that the shift has occurred.

REGISTER 7-2: OSCTUNE: OSCILLATOR TUNING REGISTER

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	TUN5	TUN4	TUN3	TUN2	TUN1	TUN0
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 **TUN<5:0>:** Frequency Tuning bits

01 1111 = Maximum frequency
01 1110 =
•
•
•
00 0001 =
00 0000 = Oscillator module is running at the factory-calibrated frequency.
11 1111 =
•
•
•
10 0000 = Minimum frequency

7.6 External Clock Modes

7.6.1 OSCILLATOR START-UP TIMER (OST)

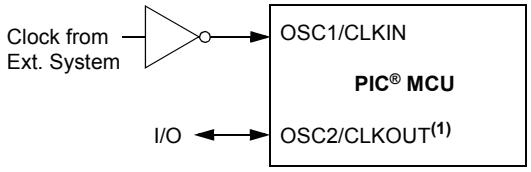
If the oscillator module is configured for LP, XT or HS modes, the Oscillator Start-up Timer (OST) counts 1024 oscillations on the OSC1 pin before the device is released from Reset. This occurs following a Power-on Reset (POR) and when the Power-up Timer (PWRT) has expired (if configured), or a wake-up from Sleep. During this time, the program counter does not increment and program execution is suspended. The OST ensures that the oscillator circuit, using a quartz crystal resonator or ceramic resonator, has started and is providing a stable system clock to the oscillator module.

7.6.2 EC MODE

The External Clock (EC) mode allows an externally generated logic level as the system clock source. When operating in this mode, an external clock source is connected to the OSC1 input and the OSC2 is available for general purpose I/O. Figure 7-2 shows the pin connections for EC mode.

The Oscillator Start-up Timer (OST) is disabled when EC mode is selected. Therefore, there is no delay in operation after a Power-on Reset (POR) or wake-up from Sleep. Because the PIC® MCU design is fully static, stopping the external clock input will have the effect of halting the device while leaving all data intact. Upon restarting the external clock, the device will resume operation as if no time had elapsed.

FIGURE 7-2: EXTERNAL CLOCK (EC) MODE OPERATION



Note 1: Alternate pin functions are described in **Section 6.1 "Alternate Pin Function"**.

7.6.3 LP, XT, HS MODES

The LP, XT and HS modes support the use of quartz crystal resonators or ceramic resonators connected to OSC1 and OSC2 (Figure 7-3). The mode selects a low, medium or high gain setting of the internal inverter-amplifier to support various resonator types and speed.

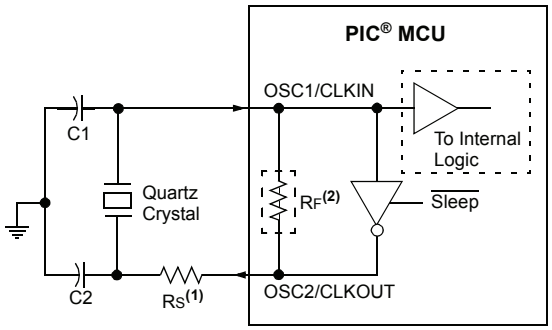
LP Oscillator mode selects the lowest gain setting of the internal inverter-amplifier. LP mode current consumption is the least of the three modes. This mode is best suited to drive resonators with a low drive level specification, for example, tuning fork type crystals.

XT Oscillator mode selects the intermediate gain setting of the internal inverter-amplifier. XT mode current consumption is the medium of the three modes. This mode is best suited to drive resonators with a medium drive level specification.

HS Oscillator mode selects the highest gain setting of the internal inverter-amplifier. HS mode current consumption is the highest of the three modes. This mode is best suited for resonators that require a high drive setting.

Figure 7-3 and Figure 7-4 show typical circuits for quartz crystal and ceramic resonators, respectively.

FIGURE 7-3: QUARTZ CRYSTAL OPERATION (LP, XT OR HS MODE)



Note 1: A series resistor (RS) may be required for quartz crystals with low drive level.

2: The value of RF varies with the Oscillator mode selected.

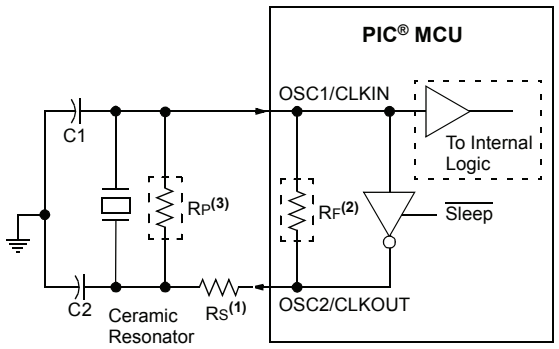
Note 1: Quartz crystal characteristics vary according to type, package and manufacturer. The user should consult the manufacturer data sheets for specifications and recommended application.

2: Always verify oscillator performance over the VDD and temperature range that is expected for the application.

3: For oscillator design assistance, reference the following Microchip Applications Notes:

- AN826, "*Crystal Oscillator Basics and Crystal Selection for rfPIC® and PIC® Devices*" (DS00826)
- AN849, "*Basic PIC® Oscillator Design*" (DS00849)
- AN943, "*Practical PIC® Oscillator Analysis and Design*" (DS00943)
- AN949, "*Making Your Oscillator Work*" (DS00949)

FIGURE 7-4: CERAMIC RESONATOR OPERATION (XT OR HS MODE)



Note 1: A series resistor (RS) may be required for ceramic resonators with low drive level.

2: The value of RF varies with the Oscillator mode selected.

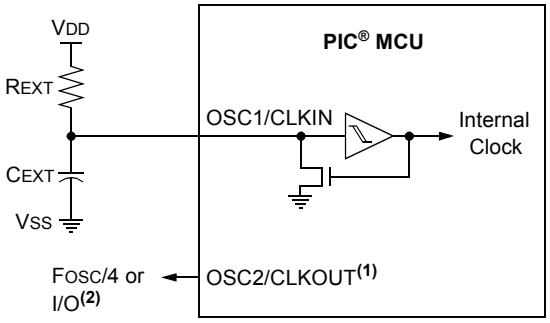
3: An additional parallel feedback resistor (RP) may be required for proper ceramic resonator operation.

7.6.4 EXTERNAL RC MODES

The external Resistor-Capacitor (RC) modes support the use of an external RC circuit. This allows the designer maximum flexibility in frequency choice while keeping costs to a minimum when clock accuracy is not required. There are two modes: RC and RCIO.

In RC mode, the RC circuit connects to OSC1. OSC2/CLKOUT outputs the RC oscillator frequency divided by 4. This signal may be used to provide a clock for external circuitry, synchronization, calibration, test or other application requirements. Figure 7-5 shows the external RC mode connections.

FIGURE 7-5: EXTERNAL RC MODES



Recommended values: 10 kΩ ≤ REXT ≤ 100 kΩ, <3V
3 kΩ ≤ REXT ≤ 100 kΩ, 3-5V
CEXT > 20 pF, 2-5V

Note 1: Alternate pin functions are described in **Section 6.1 "Alternate Pin Function"**.

2: Output depends upon RC or RCIO clock mode.

In RCIO mode, the RC circuit is connected to OSC1. OSC2 becomes an additional general purpose I/O pin.

The RC oscillator frequency is a function of the supply voltage, the resistor (REXT) and capacitor (CEXT) values and the operating temperature. Other factors affecting the oscillator frequency are:

- threshold voltage variation
- component tolerances
- packaging variations in capacitance

The user also needs to take into account variation due to tolerance of external RC components used.

TABLE 7-1: SUMMARY OF REGISTERS ASSOCIATED WITH CLOCK SOURCES

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
OSCCON	—	—	IRCF1	IRCF0	ICSL	ICSS	—	—	79
OSCTUNE	—	—	TUN5	TUN4	TUN3	TUN2	TUN1	TUN0	80

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

TABLE 7-2: SUMMARY OF CONFIGURATION WORD WITH CLOCK SOURCES

Name	Bits	Bit -/7	Bit -/6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	Register on Page
CONFIG1	13:8	—	—	$\overline{\text{DEBUG}}$	PLLEN	—	BORV	BOREN1	BOREN0	83
	7:0	—	$\overline{\text{CP}}$	MCLRE	$\overline{\text{PWRTE}}$	WDTE	FOSC<2:0>			

Legend: — = unimplemented location, read as '0'. Shaded cells are not used by clock sources.

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(4)	R/P-1	R/P-1	R/P-1
$\overline{\text{DEBUG}}$	PLLEN	—	BORV	BOREN1	BOREN0
bit 13					bit 8

U-1(4)	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
—	$\overline{\text{CP}}$	MCLRE	$\overline{\text{PWRTE}}$	WDTE	FOSC2	FOSC1	FOSC0
bit 7							bit 0

Legend:	P = Programmable bit		
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 13 **$\overline{\text{DEBUG}}$:** 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 **PLLEN:** INTOSC PLL Enable bit
0 = INTOSC frequency is 500 kHz
1 = INTOSC frequency is 16 MHz (32x)

bit 11 **Unimplemented:** Read as '1'

bit 10 **BORV:** Brown-out Reset Voltage selection bit
0 = Brown-out Reset Voltage (V_{BOR}) set to 2.5 V nominal
1 = Brown-out Reset Voltage (V_{BOR}) set to 1.9 V nominal

bit 9-8 **BOREN<1:0>:** Brown-out Reset Selection bits(1)
0x = BOR disabled (preconditioned state)
10 = BOR enabled during operation and disabled in Sleep
11 = BOR enabled

bit 7 **Unimplemented:** Read as '1'

bit 6 **$\overline{\text{CP}}$:** Code Protection bit(2)
1 = Program memory code protection is disabled
0 = Program memory code protection is enabled

bit 5 **MCLRE:** RE3/$\overline{\text{MCLR}}$ Pin Function Select bit(3)
1 = RE3/$\overline{\text{MCLR}}$ pin function is $\overline{\text{MCLR}}$
0 = RE3/$\overline{\text{MCLR}}$ pin function is digital input, $\overline{\text{MCLR}}$ internally tied to V_{DD}

bit 4 **$\overline{\text{PWRTE}}$:** Power-up Timer Enable bit
1 = PWRT disabled
0 = PWRT enabled

bit 3 **WDTE:** 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 $\overline{\text{MCLR}}$ is asserted in INTOSC or RC mode, the internal clock oscillator is disabled.
4: MPLAB® IDE masks unimplemented Configuration bits to '0'.

REGISTER 8-1: CONFIG1: CONFIGURATION WORD REGISTER 1 (CONTINUED)

bit 2-0 **FOSC<2:0>:** Oscillator Selection bits
111 = RC oscillator: CLKOUT function on RA6/OSC2/CLKOUT pin, RC on RA7/OSC1/CLKIN
110 = RCIO oscillator: I/O function on RA6/OSC2/CLKOUT pin, RC on RA7/OSC1/CLKIN
101 = INTOSC oscillator: CLKOUT function on RA6/OSC2/CLKOUT pin, I/O function on RA7/OSC1/CLKIN
100 = INTOSCIO oscillator: I/O function on RA6/OSC2/CLKOUT pin, I/O function on RA7/OSC1/CLKIN
011 = EC: I/O function on RA6/OSC2/CLKOUT pin, CLKIN on RA7/OSC1/CLKIN
010 = HS oscillator: High-speed crystal/resonator on RA6/OSC2/CLKOUT and RA7/OSC1/CLKIN
001 = XT oscillator: Crystal/resonator on RA6/OSC2/CLKOUT and RA7/OSC1/CLKIN
000 = LP oscillator: Low-power crystal on RA6/OSC2/CLKOUT and RA7/OSC1/CLKIN

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 $\overline{MCLR}$ is asserted in INTOSC or RC mode, the internal clock oscillator is disabled.
4: MPLAB® IDE masks unimplemented Configuration bits to '0'.

REGISTER 8-2: CONFIG2: CONFIGURATION WORD REGISTER 2

U-1(1)	U-1(1)	U-1(1)	U-1(1)	U-1(1)	U-1(1)
—	—	—	—	—	—
bit 13					bit 8

U-1(1)	U-1(1)	R/P-1	R/P-1	U-1(1)	U-1(1)	U-1(1)	U-1(1)
—	—	VCAPEN1	VCAPEN0	—	—	—	—
bit 7							bit 0

Legend:	P = Programmable bit		
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 13-6 **Unimplemented**: Read as '1'

bit 5-4 **VCAPEN<1:0>:** Voltage Regulator Capacitor Enable bits
For the PIC16LF722A/723A:
These bits are ignored. All VCAP pin functions are disabled.
For the PIC16F722A/723A:
00 = VCAP functionality is enabled on RA0
01 = VCAP functionality is enabled on RA5
10 = VCAP functionality is enabled on RA6
11 = All VCAP functions are disabled (not recommended)

bit 3-0 **Unimplemented**: Read as '1'

Note 1: MPLAB® IDE masks unimplemented Configuration bits to '0'.

8.2 Code Protection

If the code protection bit(s) have not been programmed, the on-chip program memory can be read out using ICSP™ for verification purposes.

Note:	The entire Flash program memory will be erased when the code protection is turned off. See the "*PIC16F72X/PIC16LF72X Memory Programming Specification*" (DS41332) for more information.

8.3 User ID

Four memory locations (2000h-2003h) are designated as ID locations where the user can store checksum or other code identification numbers. These locations are not accessible during normal execution, but are readable and writable during Program/Verify mode. Only the Least Significant 7 bits of the ID locations are reported when using MPLAB IDE. See the "*PIC16F72X/PIC16LF72X Memory Programming Specification*" (DS41332) for more information.

NOTES:

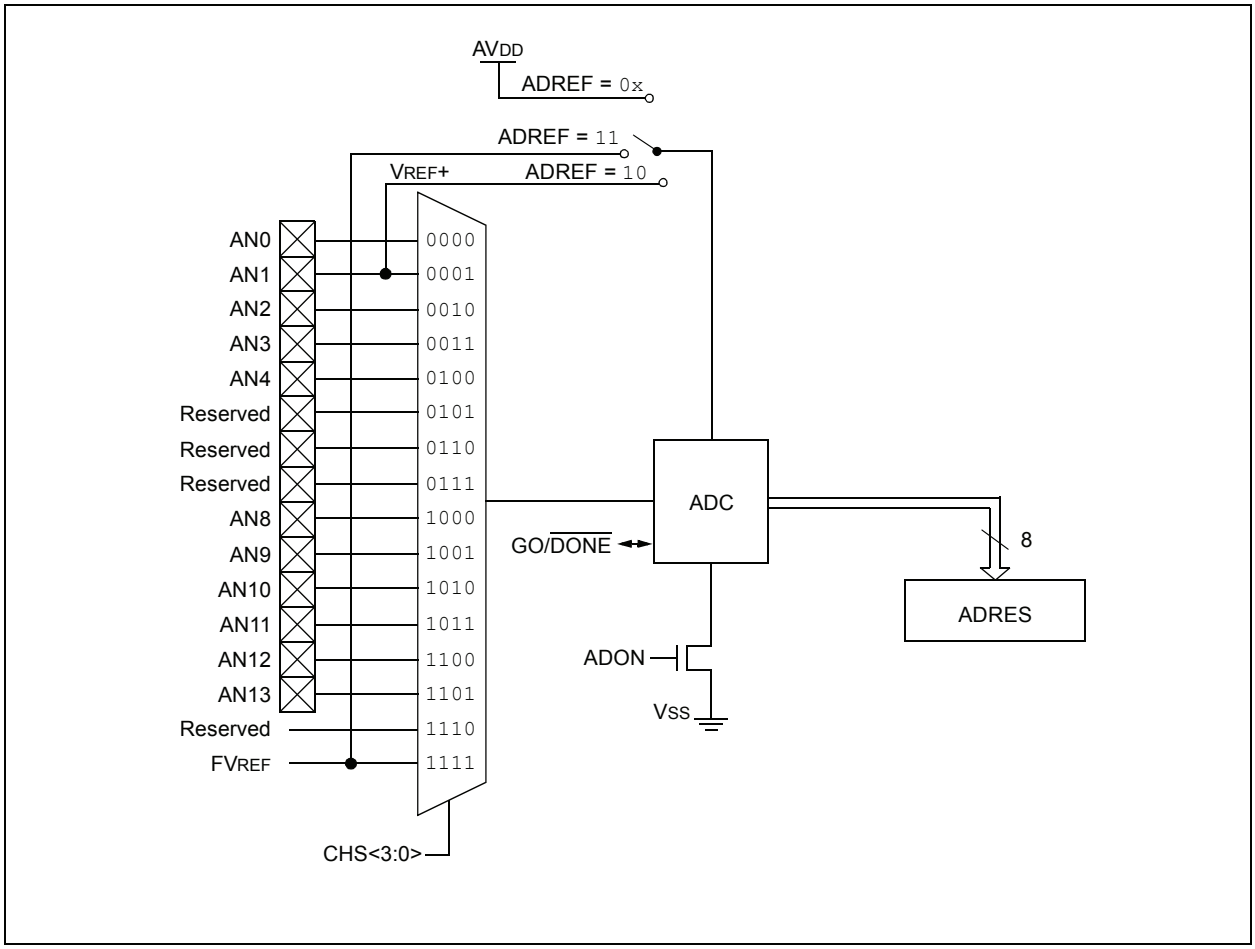
9.0 ANALOG-TO-DIGITAL CONVERTER (ADC) MODULE

The Analog-to-Digital Converter (ADC) allows conversion of an analog input signal to a 8-bit binary representation of that signal. This device uses analog inputs, which are multiplexed into a single sample and hold circuit. The output of the sample and hold is connected to the input of the converter. The converter generates a 8-bit binary result via successive approximation and stores the conversion result into the ADC result register (ADRES). Figure 9-1 shows the block diagram of the ADC.

The ADC voltage reference is software selectable to be either internally generated or externally supplied.

The ADC can generate an interrupt upon completion of a conversion. This interrupt can be used to wake-up the device from Sleep.

FIGURE 9-1: ADC BLOCK DIAGRAM



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

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.

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 (T_{AD}) VS. DEVICE OPERATING FREQUENCIES

ADC Clock Period (T_{AD})		Device Frequency (F_{OSC})				
ADC Clock Source	ADCS<2:0>	20 MHz	16 MHz	8 MHz	4 MHz	1 MHz
Fosc/2	000	100 ns(2)	125 ns(2)	250 ns(2)	500 ns(2)	2.0 μs
Fosc/4	100	200 ns(2)	250 ns(2)	500 ns(2)	1.0 μs	4.0 μs
Fosc/8	001	400 ns(2)	0.5 μs(2)	1.0 μs	2.0 μs	8.0 μs(3)
Fosc/16	101	800 ns	1.0 μs	2.0 μs	4.0 μs	16.0 μs(3)
Fosc/32	010	1.6 μs	2.0 μs	4.0 μs	8.0 μs(3)	32.0 μs(3)
Fosc/64	110	3.2 μs	4.0 μs	8.0 μs(3)	16.0 μs(3)	64.0 μs(3)
FRC	x11	1.0-6.0 μs(1,4)	1.0-6.0 μs(1,4)	1.0-6.0 μs(1,4)	1.0-6.0 μs(1,4)	1.0-6.0 μs(1,4)

Legend: Shaded cells are outside of recommended range.

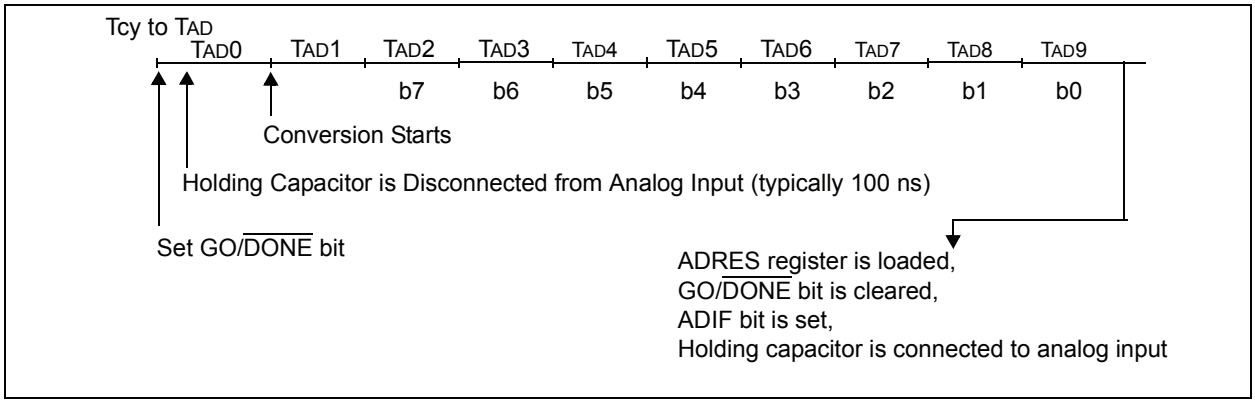
Note 1: The FRC source has a typical T_{AD} time of 1.6 μs for V_{DD}.

2: These values violate the minimum required T_{AD} 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.

FIGURE 9-2: ANALOG-TO-DIGITAL CONVERSION T_{AD} CYCLES



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.

9.2.6 A/D CONVERSION PROCEDURE

This is an example procedure for using the ADC to perform an Analog-to-Digital conversion:

1. Configure Port:
   - Disable pin output driver (Refer to the TRIS register)
   - Configure pin as analog (Refer to the ANSEL register)
2. Configure the ADC module:
   - Select ADC conversion clock
   - Configure voltage reference
   - Select ADC input channel
   - Turn on ADC module
3. Configure ADC interrupt (optional):
   - Clear ADC interrupt flag
   - Enable ADC interrupt
   - Enable peripheral interrupt
   - Enable global interrupt(1)
4. Wait the required acquisition time(2).
5. Start conversion by setting the GO/$\overline{\text{DONE}}$ bit.
6. Wait for ADC conversion to complete by one of the following:
   - Polling the GO/$\overline{\text{DONE}}$ bit
   - Waiting for the ADC interrupt (interrupts enabled)
7. Read ADC Result.
8. Clear the ADC interrupt flag (required if interrupt is enabled).

Note 1: The global interrupt can be disabled if the user is attempting to wake-up from Sleep and resume in-line code execution.

2: Refer to **Section 9.3 "A/D Acquisition Requirements"**.

EXAMPLE 9-1: A/D CONVERSION

```
;This code block configures the ADC
;for polling, Vdd reference, Frc clock
;and AN0 input.
;
;Conversion start & polling for completion
; are included.
;
BANKSEL     ADCON1      ;
MOVLW       B'01110000' ;ADC Frc clock,
                        ;VDD reference
MOVWF       ADCON1      ;
BANKSEL     TRISA       ;
BSF         TRISA,0     ;Set RA0 to input
BANKSEL     ANSELA      ;
BSF         ANSELA,0    ;Set RA0 to analog
BANKSEL     ADCON0      ;
MOVLW       B'00000001';AN0, On
MOVWF       ADCON0      ;
CALL        SampleTime  ;Acquisiton delay
BSF         ADCON0,GO   ;Start conversion
BTFSC       ADCON0,GO   ;Is conversion done?
GOTO        $-1         ;No, test again
BANKSEL     ADRES       ;
MOVF        ADRES,W     ;Read result
MOVWF       RESULT      ;store in GPR space
```

9.2.7 ADC REGISTER DEFINITIONS

The following registers are used to control the operation of the ADC.

REGISTER 9-1: ADCON0: A/D CONTROL REGISTER 0

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	CHS3	CHS2	CHS1	CHS0	GO/$\overline{\text{DONE}}$	ADON
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 5-2 **CHS<3:0>:** Analog Channel Select bits

0000 = AN0
0001 = AN1
0010 = AN2
0011 = AN3
0100 = AN4
0101 = Reserved
0110 = Reserved
0111 = Reserved
1000 = AN8
1001 = AN9
1010 = AN10
1011 = AN11
1100 = AN12
1101 = AN13
1110 = Reserved
1111 = Fixed Voltage Reference (FVREF)

bit 1 **GO/$\overline{\text{DONE}}$:** A/D Conversion Status bit

1 = A/D conversion cycle in progress. Setting this bit starts an A/D conversion cycle.
This bit is automatically cleared by hardware when the A/D conversion has completed.
0 = A/D conversion completed/not in progress

bit 0 **ADON:** ADC Enable bit

1 = ADC is enabled
0 = ADC is disabled and consumes no operating current

REGISTER 9-2: ADCON1: A/D CONTROL REGISTER 1

U-0	R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0
—	ADCS2	ADCS1	ADCS0	—	—	ADREF1	ADREF0
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 **Unimplemented:** Read as '0'

bit 6-4 **ADCS<2:0>:** A/D Conversion Clock Select bits

000 = FOSC/2
001 = FOSC/8
010 = FOSC/32
011 = FRC (clock supplied from a dedicated RC oscillator)
100 = FOSC/4
101 = FOSC/16
110 = FOSC/64
111 = FRC (clock supplied from a dedicated RC oscillator)

bit 3-2 **Unimplemented:** Read as '0'

bit 1-0 **ADREF<1:0>:** Voltage Reference Configuration bits

0x = VREF is connected to VDD
10 = VREF is connected to external VREF (RA3/AN3)
11 = VREF is connected to internal Fixed Voltage Reference

REGISTER 9-3: ADRES: ADC RESULT REGISTER

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
ADRES7	ADRES6	ADRES5	ADRES4	ADRES3	ADRES2	ADRES1	ADRES0
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 7-0 **ADRES<7:0>**: ADC Result Register bits
8-bit conversion result.

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 10kΩ 5.0V VDD

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

The value for TC 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 TC:

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

Therefore:

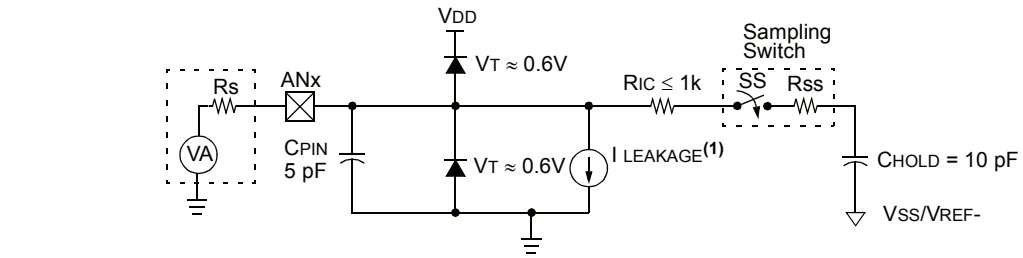
$$\begin{aligned} T_{ACQ} &= 2\mu s + 1.12\mu s + [(50°C - 25°C)(0.05\mu s/°C)] \\ &= 4.42\mu s \end{aligned}$$

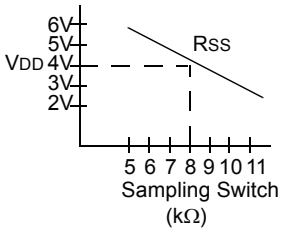
Note 1: The reference voltage (VREF) has no effect on the equation, since it cancels itself out.

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.

FIGURE 9-3: ANALOG INPUT MODEL





Legend:

CHOLD	=	Sample/Hold Capacitance
CPIN	=	Input Capacitance
I LEAKAGE	=	Leakage current at the pin due to various junctions
RIC	=	Interconnect Resistance
RSS	=	Resistance of Sampling Switch
SS	=	Sampling Switch
VT	=	Threshold Voltage

Note 1: Refer to **Section 23.0 "Electrical Specifications"**.

FIGURE 9-4: ADC TRANSFER FUNCTION

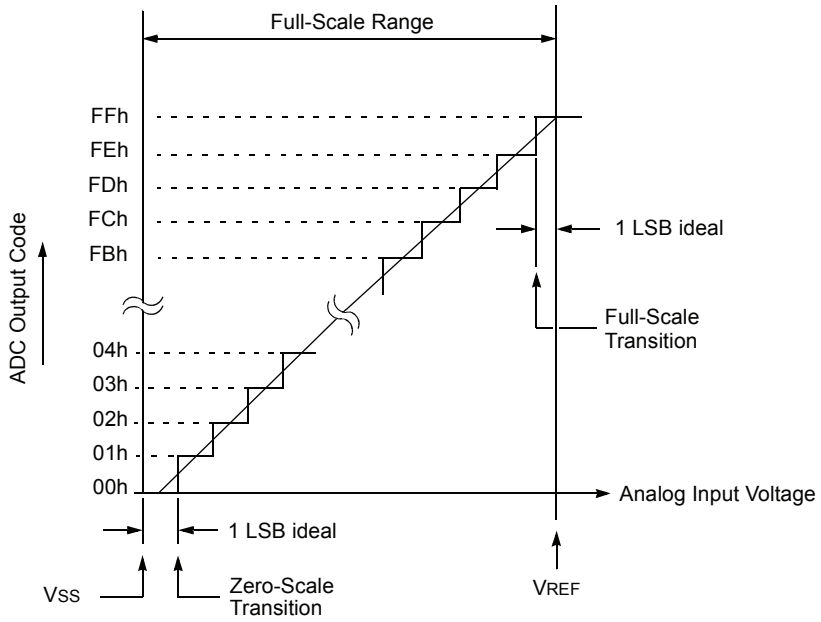


TABLE 9-2: SUMMARY OF ASSOCIATED ADC REGISTERS

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ADCON0	—	—	CHS3	CHS2	CHS1	CHS0	GO/$\overline{\text{DONE}}$	ADON	92
ADCON1	—	ADCS2	ADCS1	ADCS0	—	—	ADREF1	ADREF0	93
ANSELA	—	—	ANSA5	ANSA4	ANSA3	ANSA2	ANSA1	ANSA0	49
ANSELB	—	—	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0	58
ADRES	A/D Result Register Byte								93
CCP2CON	—	—	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	124
FVRCON	FVRRDY	FVREN	—	—	—	—	ADFVR1	ADFVR0	97
INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	40
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	41
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	43
TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	48
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	57

Legend: x = unknown, u = unchanged, – = unimplemented read as '0', q = value depends on condition. Shaded cells are not used for ADC module.

10.0 FIXED VOLTAGE REFERENCE

This device contains an internal voltage regulator. To provide a reference for the regulator, a band gap reference is provided. This band gap is also user accessible via an A/D converter channel.

User level band gap functions are controlled by the FVRCON register, which is shown in Register 10-1.

REGISTER 10-1: FVRCON: FIXED VOLTAGE REFERENCE REGISTER

R-q	R/W-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
FVRRDY	FVREN	—	—	—	—	ADFVR1	ADFVR0
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
q = Value depends on condition			

bit 7 **FVRRDY:** Fixed Voltage Reference Ready Flag bit
0 = Fixed Voltage Reference output is not active or stable
1 = Fixed Voltage Reference output is ready for use

bit 6 **FVREN(1):** Fixed Voltage Reference Enable bit
0 = Fixed Voltage Reference is disabled
1 = Fixed Voltage Reference is enabled

bit 5-2 **Unimplemented:** Read as '0'

bit 1-0 **ADFVR<1:0>:** A/D Converter Fixed Voltage Reference Selection bits
00 = A/D Converter Fixed Voltage Reference Peripheral output is off.
01 = A/D Converter Fixed Voltage Reference Peripheral output is 1x (1.024V)
10 = A/D Converter Fixed Voltage Reference Peripheral output is 2x (2.048V)(1)
11 = A/D Converter Fixed Voltage Reference Peripheral output is 4x (4.096V)(1)

Note 1: Fixed Voltage Reference output cannot exceed VDD.

NOTES:

11.0 TIMER0 MODULE

The Timer0 module is an 8-bit timer/counter with the following features:

- 8-bit timer/counter register (TMR0)
- 8-bit prescaler (shared with Watchdog Timer)
- Programmable internal or external clock source
- Programmable external clock edge selection
- Interrupt on overflow
- TMR0 can be used to gate Timer1

Figure 11-1 is a block diagram of the Timer0 module.

11.1 Timer0 Operation

The Timer0 module can be used as either an 8-bit timer or an 8-bit counter.

11.1.1 8-BIT TIMER MODE

The Timer0 module will increment every instruction cycle, if used without a prescaler. 8-Bit Timer mode is selected by clearing the T0CS bit of the OPTION register.

When TMR0 is written, the increment is inhibited for two instruction cycles immediately following the write.

Note:	The value written to the TMR0 register can be adjusted, in order to account for the two instruction cycle delay when TMR0 is written.

11.1.2 8-BIT COUNTER MODE

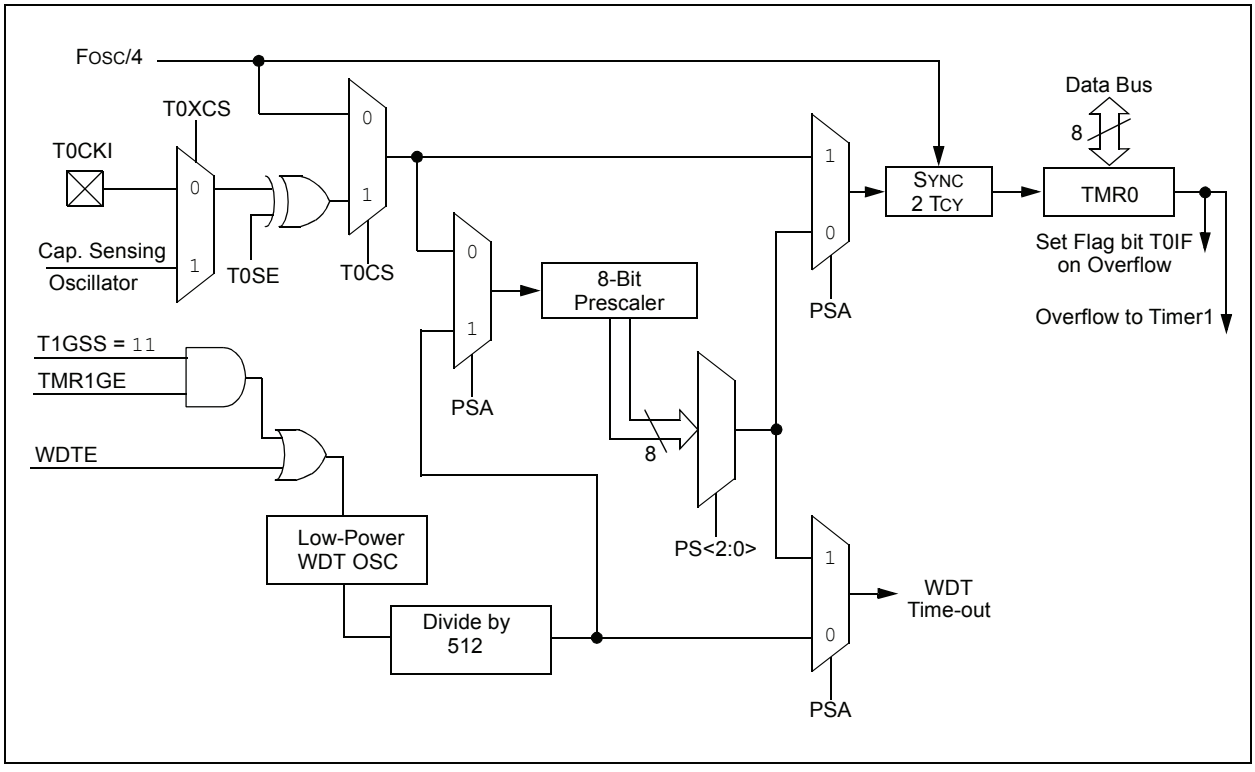
In 8-Bit Counter mode, the Timer0 module will increment on every rising or falling edge of the T0CKI pin or the Capacitive Sensing Oscillator (CPSOSC) signal.

8-Bit Counter mode using the T0CKI pin is selected by setting the T0CS bit in the OPTION register to '1' and resetting the T0XCS bit in the CPSCON0 register to '0'.

8-Bit Counter Mode using the Capacitive Sensing Oscillator (CPSOSC) signal is selected by setting the T0CS bit in the OPTION register to '1' and setting the T0XCS bit in the CPSCON0 register to '1'.

The rising or falling transition of the incrementing edge for either input source is determined by the T0SE bit in the OPTION register.

FIGURE 11-1: BLOCK DIAGRAM OF THE TIMER0/WDT PRESCALER



11.1.3 SOFTWARE PROGRAMMABLE PRESCALER

A single software programmable prescaler is available for use with either Timer0 or the Watchdog Timer (WDT), but not both simultaneously. The prescaler assignment is controlled by the PSA bit of the OPTION register. To assign the prescaler to Timer0, the PSA bit must be cleared to a '0'.

There are 8 prescaler options for the Timer0 module ranging from 1:2 to 1:256. The pres ca le values are selectable via the PS<2:0> bits of the OPTION register. In order to have a 1:1 prescaler value for the Timer0 module, the prescaler must be assigned to the WDT module.

The prescaler is not readable or writable. When assigned to the Timer0 module, all instructions writing to the TMR0 register will clear the prescaler.

Note:	When the prescaler is assigned to WDT, a CLRWDT instruction will clear the prescaler along with the WDT.

11.1.4 TIMER0 INTERRUPT

Timer0 will generate an interrupt when the TMR0 register overflows from PH to 00h. The T0IF interrupt flag bit of the INTCON register is set every time the TMR0 register overflows, regardless of whether or not the Timer0 interrupt is enabled. The T0IF bit can only be cleared in software. The Timer0 interrupt enable is the T0IE bit of the INTCON register.

Note:	The Timer0 interrupt cannot wake the processor from Sleep since the timer is frozen during Sleep.

11.1.5 8-BIT COUNTER MODE SYNCHRONIZATION

When in 8-Bit Counter mode, the incrementing edge on the T0CKI pin must be synchronized to the instruction clock. Synchronization can be accomplished by sampling the prescaler output on the Q2 and Q4 cycles of the instruction clock. The high and low periods of the external clocking source must meet the timing requirements as shown in **Section 23.0 "Electrical Specifications"**.

REGISTER 11-1: OPTION_REG: OPTION REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
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 **$\overline{\text{RBPU}}$:** PORTB Pull-up Enable bit
1 = PORTB pull-ups are disabled
0 = PORTB pull-ups are enabled by individual PORT latch values

bit 6 **INTEDG:** Interrupt Edge Select bit
1 = Interrupt on rising edge of INT pin
0 = Interrupt on falling edge of INT pin

bit 5 **T0CS:** TMR0 Clock Source Select bit
1 = Transition on T0CKI pin or CPSOSC signal
0 = Internal instruction cycle clock (FOSC/4)

bit 4 **T0SE:** TMR0 Source Edge Select bit
1 = Increment on high-to-low transition on T0CKI pin
0 = Increment on low-to-high transition on T0CKI pin

bit 3 **PSA:** Prescaler Assignment bit
1 = Prescaler is assigned to the WDT
0 = Prescaler is assigned to the Timer0 module

bit 2-0 **PS<2:0>:** Prescaler Rate Select bits

BIT VALUE	TMR0 RATE	WDT RATE
000	1 : 2	1 : 1
001	1 : 4	1 : 2
010	1 : 8	1 : 4
011	1 : 16	1 : 8
100	1 : 32	1 : 16
101	1 : 64	1 : 32
110	1 : 128	1 : 64
111	1 : 256	1 : 128

TABLE 11-1: SUMMARY OF REGISTERS ASSOCIATED WITH TIMER0

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
CPSCON0	CPSON	—	—	—	CPSRNG1	CPSRNG0	CPSOUT	T0XCS	121
INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	40
OPTION_REG	$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	101
TMR0	Timer0 Module Register								—
TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	48

Legend: – = Un implemented locations, read as '0', u = unchanged, x = unknown. Shaded cells are not used by the Timer0 module.

NOTES:

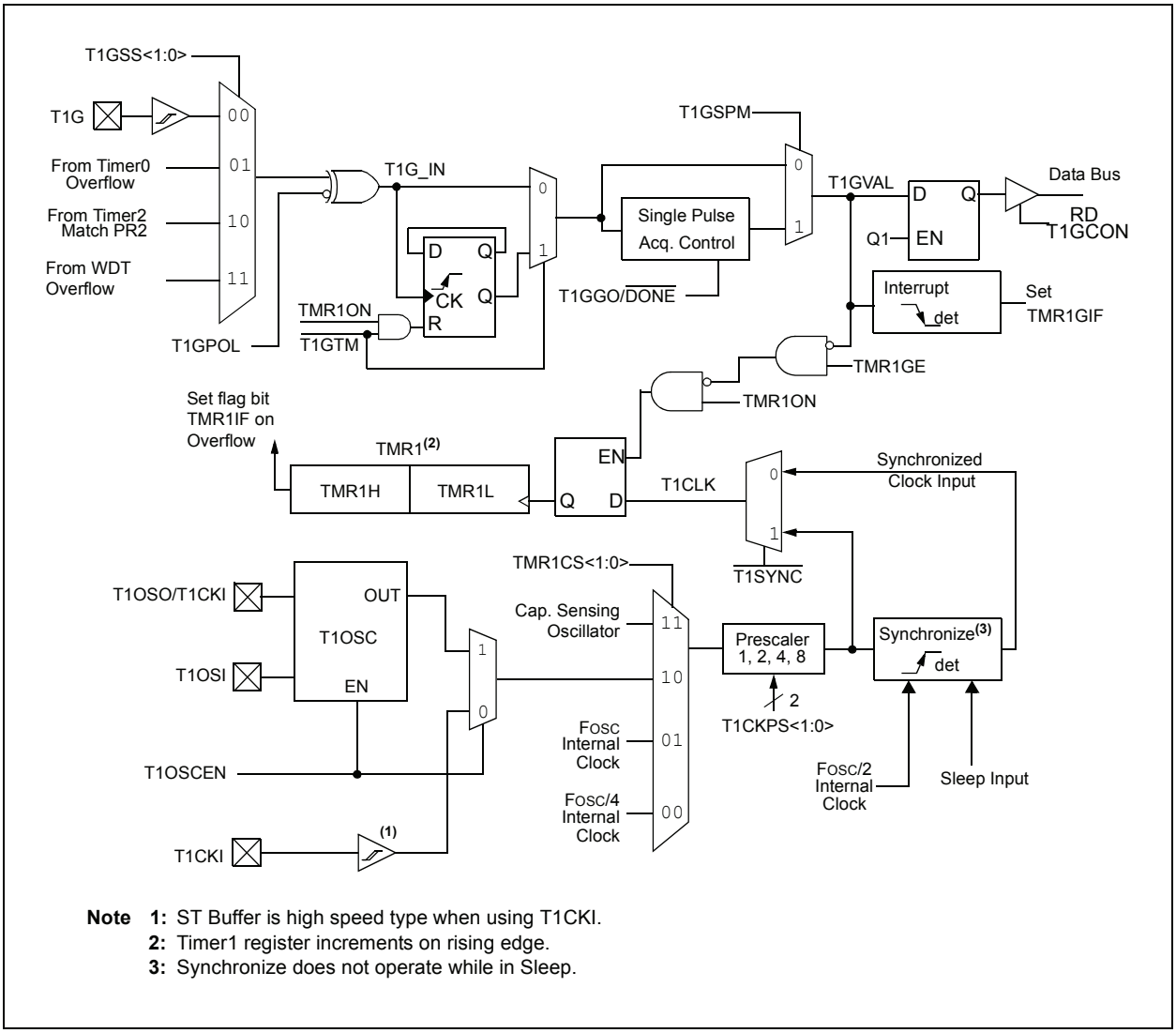
12.0 TIMER1 MODULE WITH GATE CONTROL

The Timer1 module is a 16-bit timer/counter with the following features:

- 16-bit timer/counter register pair (TMR1H:TMR1L)
- Programmable internal or external clock source
- 3-bit prescaler
- Dedicated LP oscillator circuit
- Synchronous or asynchronous operation
- Multiple Timer1 gate (count enable) sources
- Interrupt on overflow
- Wake-up on overflow (external clock, Asynchronous mode only)
- Time base for the Capture/Compare function
- Special Event Trigger (with CCP)
- Selectable Gate Source Polarity
- Gate Toggle Mode
- Gate Single-pulse Mode
- Gate Value Status
- Gate Event Interrupt

Figure 12-1 is a block diagram of the Timer1 module.

FIGURE 12-1: TIMER1 BLOCK DIAGRAM



Note 1: ST Buffer is high speed type when using T1CKI.
2: Timer1 register increments on rising edge.
3: Synchronize does not operate while in Sleep.

12.1 Timer1 Operation

The Timer1 module is a 16-bit incrementing counter which is accessed through the TMR1H:TMR1L register pair. Writes to TMR1H or TMR1L directly update the counter.

When used with an internal clock source, the module is a timer and increments on every instruction cycle. When used with an external clock source, the module can be used as either a timer or counter and increments on every selected edge of the external source.

Timer1 is enabled by configuring the TMR1ON and TMR1GE bits in the T1CON and T1GCON registers, respectively. Table 12-1 displays the Timer1 enable selections.

TABLE 12-1: TIMER1 ENABLE SELECTIONS

TMR1ON	TMR1GE	Timer1 Operation
0	0	Off
0	1	Off
1	0	Always On
1	1	Count Enabled

12.2 Clock Source Selection

The TMR1CS<1:0> and T1OSCEN bits of the T1CON register are used to select the clock source for Timer1. Table 12-2 displays the clock source selections.

12.2.1 INTERNAL CLOCK SOURCE

When the internal clock source is selected the TMR1H:TMR1L register pair will increment on multiples of FISC as determined by the Timer1 prescaler.

12.2.2 EXTERNAL CLOCK SOURCE

When the external clock source is selected, the Timer1 module may work as a timer or a counter.

When enabled to count, Timer1 is increment ed on the rising edge of the external clock input T1CKI or the capacitive sensing oscillator signal. Either of these external clock sources can be synchronized to the microcontroller system clock or they can run asynchronously.

When used as a timer with a clock oscillator, an external 32.768 kHz crystal can be used in conjunction with the dedicated internal oscillator circuit.

Note: In Counter mode, a falling edge must be registered by the counter prior to the first incrementing rising edge after any one or more of the following conditions:

- Timer1 enabled after POR
- Write to TMR1H or TMR1L
- Timer1 is disabled
- Timer1 is disabled (TMR1ON = 0) when T1CKI is high then Timer1 is enabled (TMR1ON=1) when T1CKI is low.

TABLE 12-2: CLOCK SOURCE SELECTIONS

TMR1CS1	TMR1CS0	T1OSCEN	Clock Source
0	1	x	System Clock (FOSC)
0	0	x	Instruction Clock (FOSC/4)
1	1	x	Capacitive Sensing Oscillator
1	0	0	External Clocking on T1CKI Pin
1	0	1	Oscillator Circuit on T1OSI/T1OSO Pins

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.

Note:	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 $\overline{\text{T1SYNC}}$ 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 12.5.1 "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.

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.

12.6 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.

12.6.1 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 (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 12-3 for timing details.

TABLE 12-3: TIMER1 GATE ENABLE SELECTIONS

T1CLK	T1GPOL	T1G	Timer1 Operation
↑	0	0	Counts
↑	0	1	Holds Count
↑	1	0	Holds Count
↑	1	1	Counts

12.6.2 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 12-4: 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)

12.6.2.1 T1G Pin Gate Operation

The T1G pin is one source for Timer1 gate control. It can be used to supply an external source to the Timer1 gate circuitry.

12.6.2.2 Timer0 Overflow Gate Operation

When Timer0 increments from FFh to 00h, a low-to-high pulse will automatically be generated and internally supplied to the Timer1 gate circuitry.

12.6.2.3 Timer2 Match Gate Operation

The TMR2 register will increment until it matches the value in the PR2 register. On the very next increment cycle, TMR2 will be reset to 00h. When this Reset occurs, a low-to-high pulse will automatically be generated and internally supplied to the Timer1 gate circuitry.

12.6.2.4 Watchdog Overflow Gate Operation

The Watchdog Timer oscillator, prescaler and counter will be automatically turned on when TMR1GE = 1 and T1GSS selects the WDT as a gate source for Timer1 (T1GSS = 11). TMR1ON does not factor into the oscillator, prescaler and counter enable. See Table 12-5.

The PSA and PS bits of the OPTION register still control what time-out interval is selected. Changing the prescaler during operation may result in a spurious capture.

Enabling the Watchdog Timer oscillator does not automatically enable a Watchdog Reset or Wake-up from Sleep upon counter overflow.

Note:	When using the WDT as a gate source for Timer1, operations that clear the Watchdog Timer (CLRWDT, SLEEP instructions) will affect the time interval being measured for capacitive sensing. This includes waking from Sleep. All other interrupts that might wake the device from Sleep should be disabled to prevent them from disturbing the measurement period.

As the gate signal coming from the WDT counter will generate different pulse widths depending on if the WDT is enabled, when the CLRWDT instruction is executed, and so on, Toggle mode must be used. A specific sequence is required to put the device into the correct state to capture the next WDT counter interval.

TABLE 12-5: WDT/TIMER1 GATE INTERACTION

WDTE	TMR1GE = 1 and T1GSS = 11	WDT Oscillator Enable	WDT Reset	Wake-up	WDT Available for T1G Source
1	N	Y	Y	Y	N
1	Y	Y	Y	Y	Y
0	Y	Y	N	N	Y
0	N	N	N	N	N

12.6.3 TIMER1 GATE TOGGLE MODE

When Timer1 Gate Toggle mode is enabled, it is possible to measure the full-cycle length of a Timer1 gate signal, as opposed to the duration of a single level pulse.

The Timer1 gate source is routed through a flip-flop that changes state on every incrementing edge of the signal. See Figure 12-4 for timing details.

Timer1 Gate Toggle mode is enabled by setting the T1GTM bit of the T1GCON register. When the T1GTM bit is cleared, the flip-flop is cleared and held clear. This is necessary in order to control which edge is measured.

Note: Enabling Toggle mode at the same time as changing the gate polarity may result in indeterminate operation.

12.6.4 TIMER1 GATE SINGLE-PULSE MODE

When Timer1 Gate Single-Pulse mode is enabled, it is possible to capture a single pulse gate event. Timer1 Gate Single-Pulse mode is first enabled by setting the T1GSPM bit in the T1GCON register. Next, the T1GGO/$\overline{\text{DONE}}$ bit in the T1GCON register must be set. The Timer1 will be fully enabled on the next incrementing edge. On the next trailing edge of the pulse, the T1GGO/$\overline{\text{DONE}}$ bit will automatically be cleared. No other gate events will be allowed to increment Timer1 until the T1GGO/$\overline{\text{DONE}}$ bit is once again set in software.

Clearing the T1GSPM bit of the T1GCON register will also clear the T1GGO/$\overline{\text{DONE}}$ bit. See Figure 12-5 for timing details.

Enabling the Toggle mode and the Single-Pulse mode simultaneously will permit both sections to work together. This allows the cycle times on the Timer1 gate source to be measured. See Figure 12-6 for timing details.

12.6.5 TIMER1 GATE VALUE STATUS

When Timer1 gate value status is utilized, it is possible to read the most current level of the gate control value. The value is stored in the T1GVAL bit in the T1GCON register. The T1GVAL bit is valid even when the Timer1 gate is not enabled (TMR1GE bit is cleared).

12.6.6 TIMER1 GATE EVENT INTERRUPT

When Timer1 gate event interrupt is enabled, it is possible to generate an interrupt upon the completion of a gate event. When the falling edge of T1GVAL occurs, the TMR1GIF flag bit in the PIR1 register will be set. If the TMR1GIE bit in the PIE1 register is set, then an interrupt will be recognized.

The TMR1GIF flag bit operates even when the Timer1 gate is not enabled (TMR1GE bit is cleared).

12.7 Timer1 Interrupt

The Timer1 register pair (TMR1H:TMR1L) increments to FFFFh and rolls over to 0000h. When Timer1 rolls over, the Timer1 interrupt flag bit of the PIR1 register is set. To enable the interrupt on rollover, you must set these bits:

- TMR1ON bit of the T1CON register
- TMR1IE bit of the PIE1 register
- PEIE bit of the INTCON register
- GIE bit of the INTCON register

The interrupt is cleared by clearing the TMR1IF bit in the Interrupt Service Routine.

Note:	The TMR1H:TMR1L register pair and the TMR1IF bit should be cleared before enabling interrupts.

12.8 Timer1 Operation During Sleep

Timer1 can only operate during Sleep when setup in Asynchronous Counter mode. In this mode, an external crystal or clock source can be used to increment the counter. To set up the timer to wake the device:

- TMR1ON bit of the T1CON register must be set
- TMR1IE bit of the PIE1 register must be set
- PEIE bit of the INTCON register must be set
- $\overline{\text{T1SYNC}}$ bit of the T1CON register must be set
- TMR1CS bits of the T1CON register must be configured
- T1OSCEN bit of the T1CON register must be configured
- TMR1GIE bit of the T1GCON register must be configured

The device will wake-up on an overflow and execute the next instructions. If the GIE bit of the INTCON register is set, the device will call the Interrupt Service Routine (0004h).

12.9 CCP Capture/Compare Time Base

The CCP module uses the TMR1H:TMR1L register pair as the time base when operating in Capture or Compare mode.

In Capture mode, the value in the TMR1H:TMR1L register pair is copied into the CCPR1H:CCPR1L register pair on a configured event.

In Compare mode, an event is triggered when the value CCPR1H:CCPR1L register pair matches the value in the TMR1H:TMR1L register pair. This event can be a Special Event Trigger.

For more information, see **Section 15.0 "Capture/Compare/PWM (CCP) Module"**.

12.10 CCP Special Event Trigger

When the CCP is configured to trigger a special event, the trigger will clear the TMR1H:TMR1L register pair. This special event does not cause a Timer1 interrupt. The CCP module may still be configured to generate a CCP interrupt.

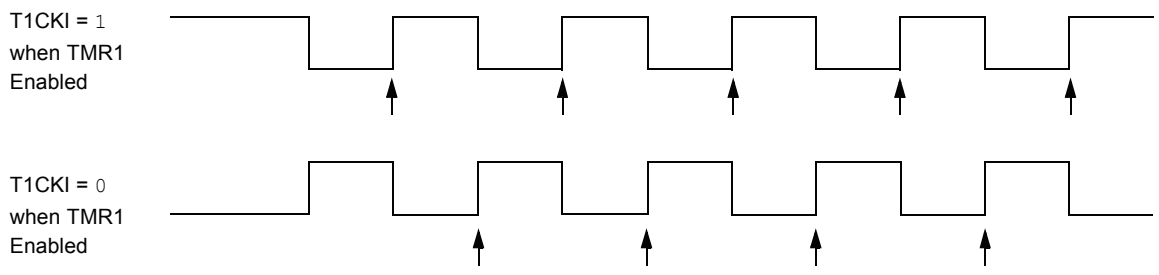
In this mode of operation, the CCPR1H:CCPR1L register pair becomes the period register for Timer1.

Timer1 should be synchronized to the FOSC/4 to utilize the Special Event Trigger. Asynchronous operation of Timer1 can cause a Special Event Trigger to be missed.

In the event that a write to TMR1H or TMR1L coincides with a Special Event Trigger from the CCP, the write will take precedence.

For more information, see **Section 9.2.5 "Special Event Trigger"**.

FIGURE 12-2: TIMER1 INCREMENTING EDGE



Note 1: Arrows indicate counter increments.

2: In Counter mode, a falling edge must be registered by the counter prior to the first incrementing rising edge of the clock.

FIGURE 12-3: TIMER1 GATE COUNT ENABLE MODE

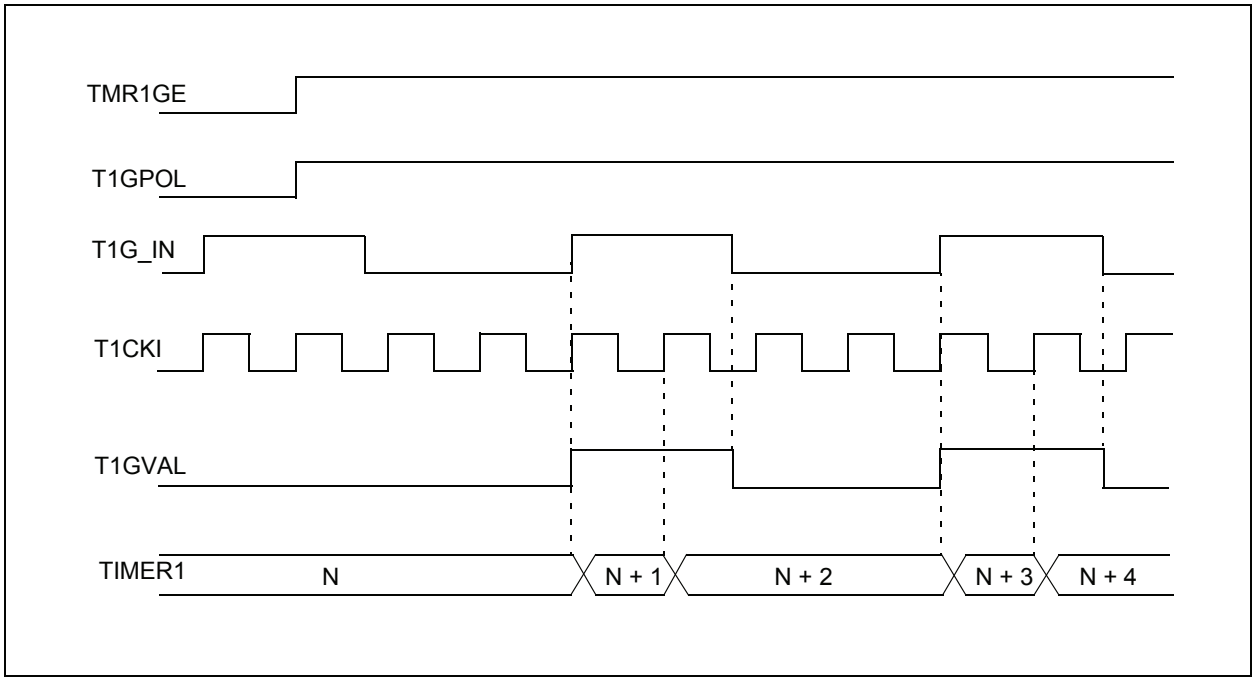


FIGURE 12-4: TIMER1 GATE TOGGLE MODE

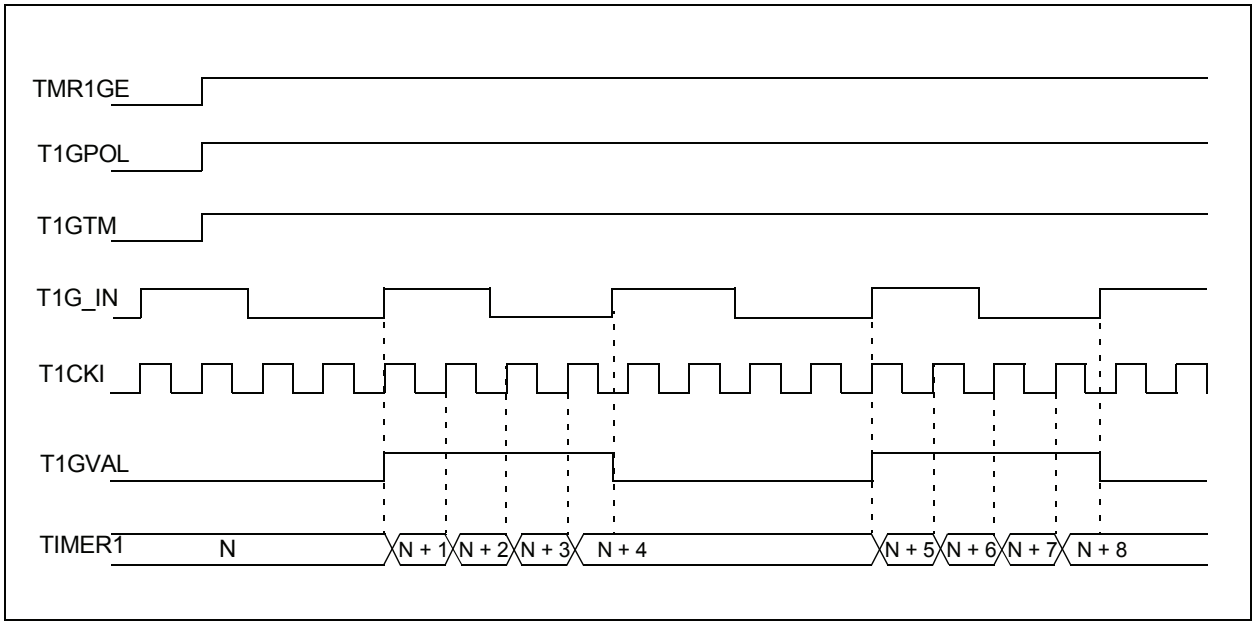


FIGURE 12-5: TIMER1 GATE SINGLE-PULSE MODE

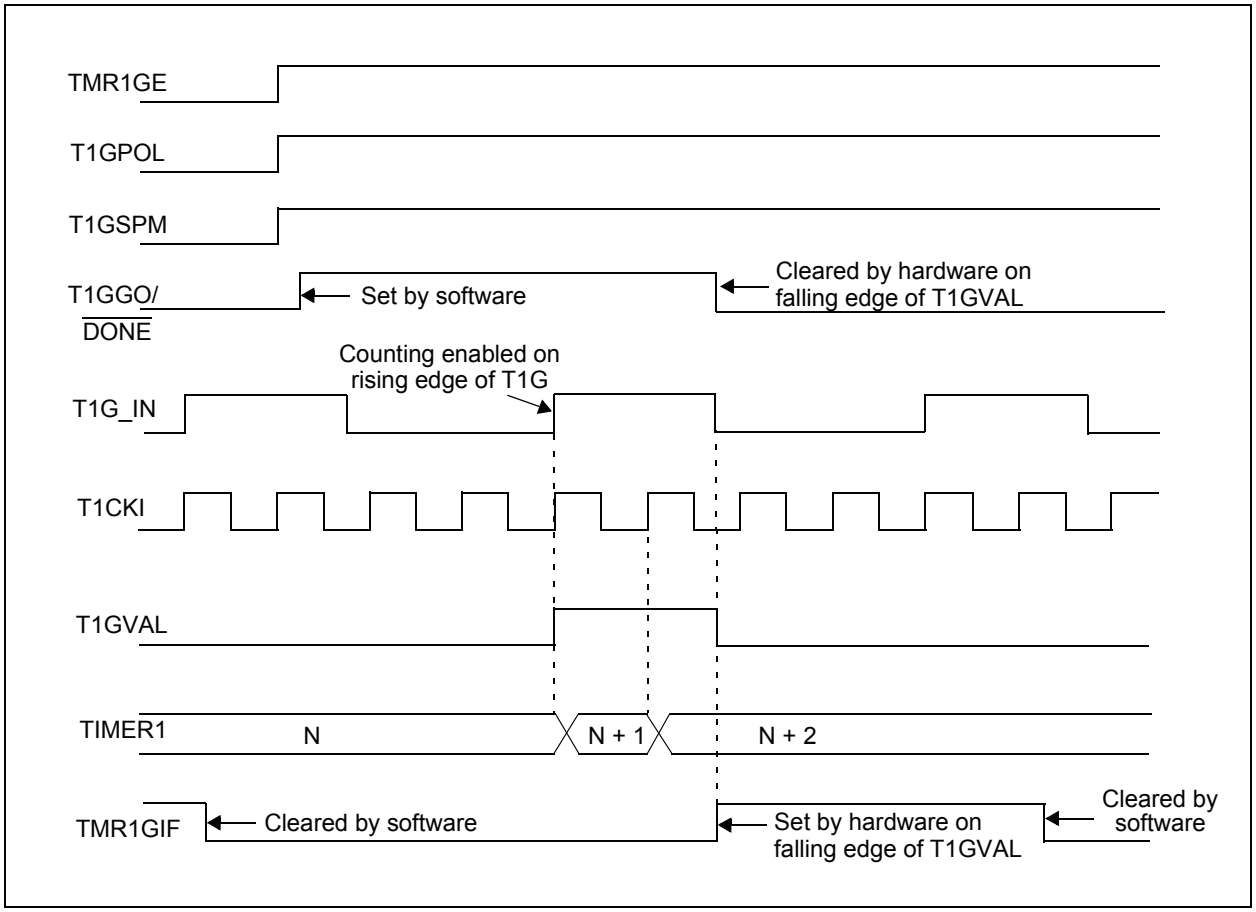
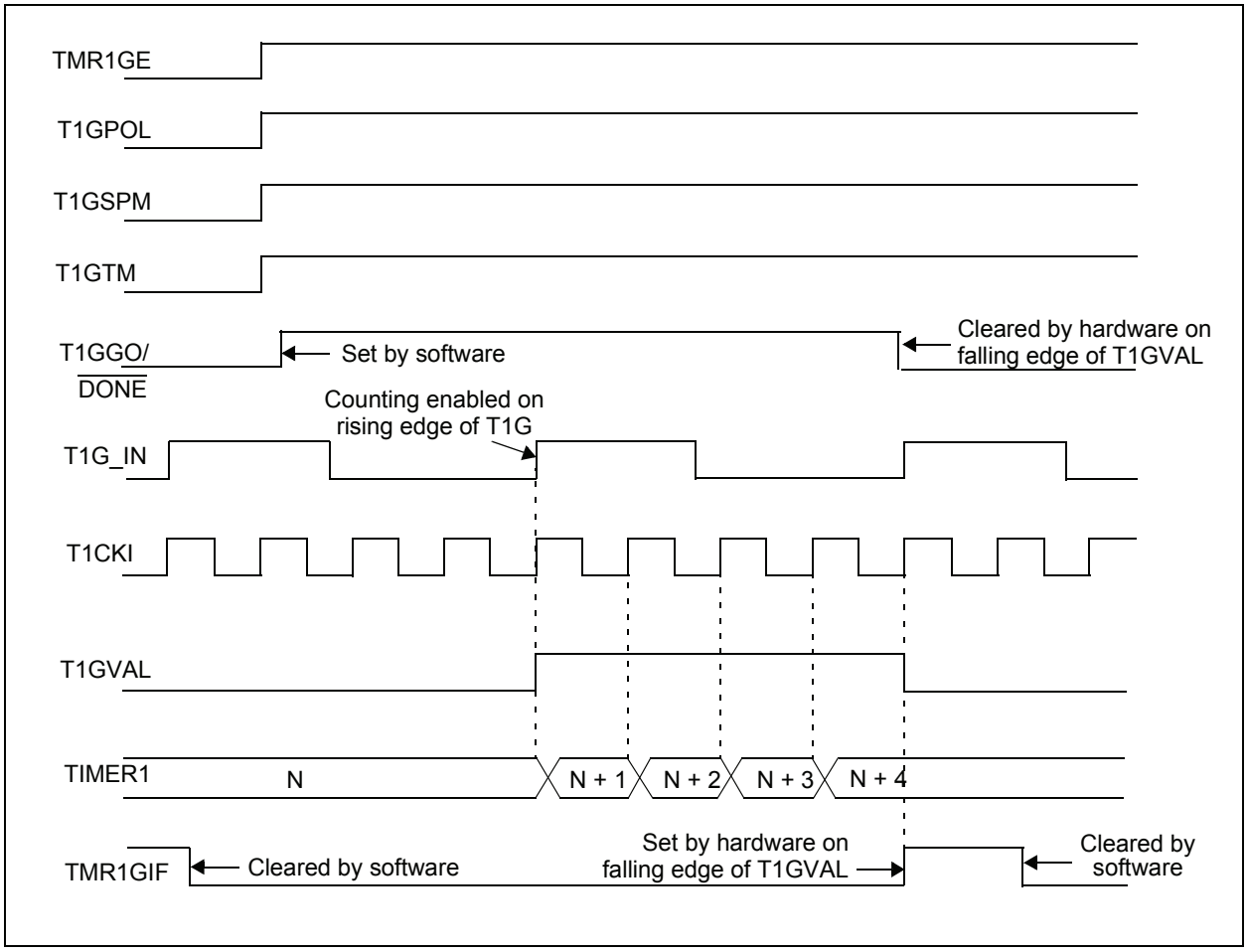


FIGURE 12-6: TIMER1 GATE SINGLE-PULSE AND TOGGLE COMBINED MODE



12.11 Timer1 Control Register

The Timer1 Control register (T1CON), shown in Register 12-1, is used to control Timer1 and select the various features of the Timer1 module.

REGISTER 12-1: T1CON: TIMER1 CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
TMR1CS1	TMR1CS0	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	—	TMR1ON
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 **TMR1CS<1:0>:** Timer1 Clock Source Select bits
- 11 = Timer1 clock source is Capacitive Sensing Oscillator (CAPOSC)
- 10 = Timer1 clock source is pin or oscillator:
  - If T1OSCEN = 0:
  - External clock from T1CKI pin (on the rising edge)
  - If T1OSCEN = 1:
  - Crystal oscillator on T1OSI/T1OSO pins
- 01 = Timer1 clock source is system clock (FOSC)
- 00 = Timer1 clock source is instruction clock (FOSC/4)

bit 5-4 **T1CKPS<1:0>:** Timer1 Input Clock Prescale Select bits
- 11 = 1:8 Prescale value
- 10 = 1:4 Prescale value
- 01 = 1:2 Prescale value
- 00 = 1:1 Prescale value

bit 3 **T1OSCEN:** LP Oscillator Enable Control bit
- 1 = Dedicated Timer1 oscillator circuit enabled
- 0 = Dedicated Timer1 oscillator circuit disabled

bit 2 **$\overline{\text{T1SYNC}}$:** Timer1 External Clock Input Synchronization Control bit

TMR1CS<1:0> = 1X
- 1 = Do not synchronize external clock input
- 0 = Synchronize external clock input with system clock (FOSC)

TMR1CS<1:0> = 0X

This bit is ignored. Timer1 uses the internal clock when TMR1CS<1:0> = 1X.

bit 1 **Unimplemented:** Read as '0'

bit 0 **TMR1ON:** Timer1 On bit
- 1 = Enables Timer1
- 0 = Stops Timer1 (Clears Timer1 gate flip-flop)

12.12 Timer1 Gate Control Register

The Timer1 Gate Control register (T1GCON), shown in Register 12-2, is used to control Timer1 gate.

REGISTER 12-2: T1GCON: TIMER1 GATE CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-x	R/W-0	R/W-0
TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/$\overline{\text{DONE}}$	T1GVAL	T1GSS1	T1GSS0
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 **TMR1GE:** Timer1 Gate Enable bit
If TMR1ON = 0:
This bit is ignored
If TMR1ON = 1:
1 = Timer1 counting is controlled by the Timer1 gate function
0 = Timer1 counts regardless of Timer1 gate function

bit 6 **T1GPOL:** Timer1 Gate Polarity bit
1 = Timer1 gate is active-high (Timer1 counts when gate is high)
0 = Timer1 gate is active-low (Timer1 counts when gate is low)

bit 5 **T1GTM:** Timer1 Gate Toggle Mode bit
1 = Timer1 Gate Toggle mode is enabled.
0 = Timer1 Gate Toggle mode is disabled and toggle flip flop is cleared
Timer1 gate flip-flop toggles on every rising edge.

bit 4 **T1GSPM:** Timer1 Gate Single Pulse Mode bit
1 = Timer1 Gate Single-Pulse mode is enabled and is controlling Timer1 gate
0 = Timer1 Gate Single-Pulse mode is disabled

bit 3 **T1GGO/$\overline{\text{DONE}}$:** Timer1 Gate Single-Pulse Acquisition Status bit
1 = Timer1 gate single-pulse acquisition is ready, waiting for an edge
0 = Timer1 gate single-pulse acquisition has completed or has not been started
This bit is automatically cleared when T1GSPM is cleared.

bit 2 **T1GVAL:** Timer1 Gate Current State bit
Indicates the current state of the Timer1 gate that could be provided to TMR1H:TMR1L.
Unaffected by Timer1 Gate Enable (TMR1GE).

bit 1-0 **T1GSS<1:0>:** Timer1 Gate Source Select bits
00 = Timer1 gate pin
01 = Timer0 Overflow output
10 = TMR2 Match PR2 output
11 = Watchdog Timer scaler overflow
Watchdog Timer oscillator is turned on if TMR1GE = 1, regardless of the state of TMR1ON

TABLE 12-6: SUMMARY OF REGISTERS ASSOCIATED WITH TIMER1

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELB	—	—	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0	58
CCP1CON	—	—	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	124
CCP2CON	—	—	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	124
INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	40
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	41
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	43
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	57
TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Register								108
TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Register								108
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	57
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	67
T1CON	TMR1CS1	TMR1CS0	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	—	TMR1ON	112
T1GCON	TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/$\overline{\text{DONE}}$	T1GVAL	T1GSS1	T1GSS0	113

Legend: x = unknown, u = unchanged, – = unimplemented, read as '0'. Shaded cells are not used by the Timer1 module.

13.0 TIMER2 MODULE

The Timer2 module is an 8-bit timer with the following features:

- 8-bit timer register (TMR2)
- 8-bit period register (PR2)
- Interrupt on TMR2 match with PR2
- Software programmable prescaler (1:1, 1:4, 1:16)
- Software programmable postscaler (1:1 to 1:16)

See Figure 13-1 for a block diagram of Timer2.

13.1 Timer2 Operation

The clock input to the Timer2 module is the system instruction clock (FOSC/4). The clock is fed into the Timer2 prescaler, which has prescale options of 1:1, 1:4 or 1:16. The output of the prescaler is then used to increment the TMR2 register.

The values of TMR2 and PR2 are constantly compared to determine when they match. TMR2 will increment from 00h until it matches the value in PR2. When a match occurs, two things happen:

- TMR2 is reset to 00h on the next increment cycle.
- The Timer2 postscaler is incremented.

The match output of the Timer2/PR2 comparator is then fed into the Timer2 postscaler. The postscaler has postscale options of 1:1 to 1:16 inclusive. The output of the Timer2 postscaler is used to set the TMR2IF interrupt flag bit in the PIR1 register.

The TMR2 and PR2 registers are both fully readable and writable. On any Reset, the TMR2 register is set to 00h and the PR2 register is set to FFh.

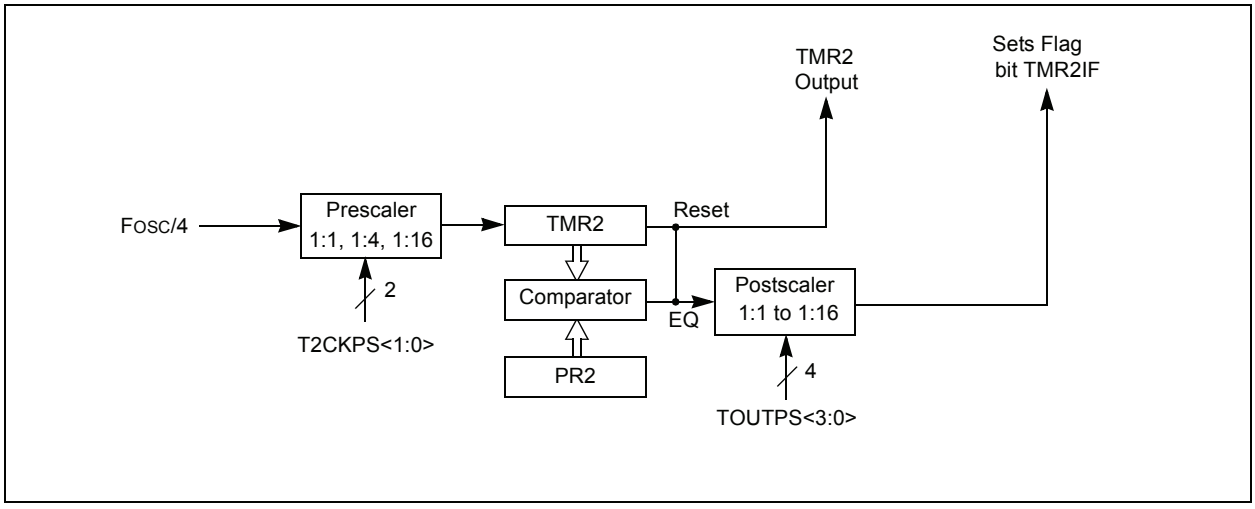
Timer2 is turned on by setting the TMR2ON bit in the T2CON register to a '1'. Timer2 is turned off by clearing the TMR2ON bit to a '0'.

The Timer2 prescaler is controlled by the T2CKPS bits in the T2CON register. The Timer2 postscaler is controlled by the TOUTPS bits in the T2CON register. The prescaler and postscaler counters are cleared when:

- A write to TMR2 occurs.
- A write to T2CON occurs.
- Any device Reset occurs (Power-on Reset, $\overline{\text{MCLR}}$ Reset, Watchdog Timer Reset, or Brown-out Reset).

Note: TMR2 is not cleared when T2CON is written.

FIGURE 13-1: TIMER2 BLOCK DIAGRAM



REGISTER 13-1: T2CON: TIMER2 CONTROL REGISTER

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0
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 **Unimplemented:** Read as '0'

bit 6-3 **TOUTPS<3:0>:** Timer2 Output Postscaler Select bits
- 0000 = 1:1 Postscaler
- 0001 = 1:2 Postscaler
- 0010 = 1:3 Postscaler
- 0011 = 1:4 Postscaler
- 0100 = 1:5 Postscaler
- 0101 = 1:6 Postscaler
- 0110 = 1:7 Postscaler
- 0111 = 1:8 Postscaler
- 1000 = 1:9 Postscaler
- 1001 = 1:10 Postscaler
- 1010 = 1:11 Postscaler
- 1011 = 1:12 Postscaler
- 1100 = 1:13 Postscaler
- 1101 = 1:14 Postscaler
- 1110 = 1:15 Postscaler
- 1111 = 1:16 Postscaler

bit 2 **TMR2ON:** Timer2 On bit
- 1 = Timer2 is on
- 0 = Timer2 is off

bit 1-0 **T2CKPS<1:0>:** Timer2 Clock Prescale Select bits
- 00 = Prescaler is 1
- 01 = Prescaler is 4
- 1x = Prescaler is 16

TABLE 13-1: SUMMARY OF REGISTERS ASSOCIATED WITH TIMER2

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	40
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	41
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	43
PR2	Timer2 Module Period Register								115
TMR2	Holding Register for the 8-bit TMR2 Register								115
T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	116

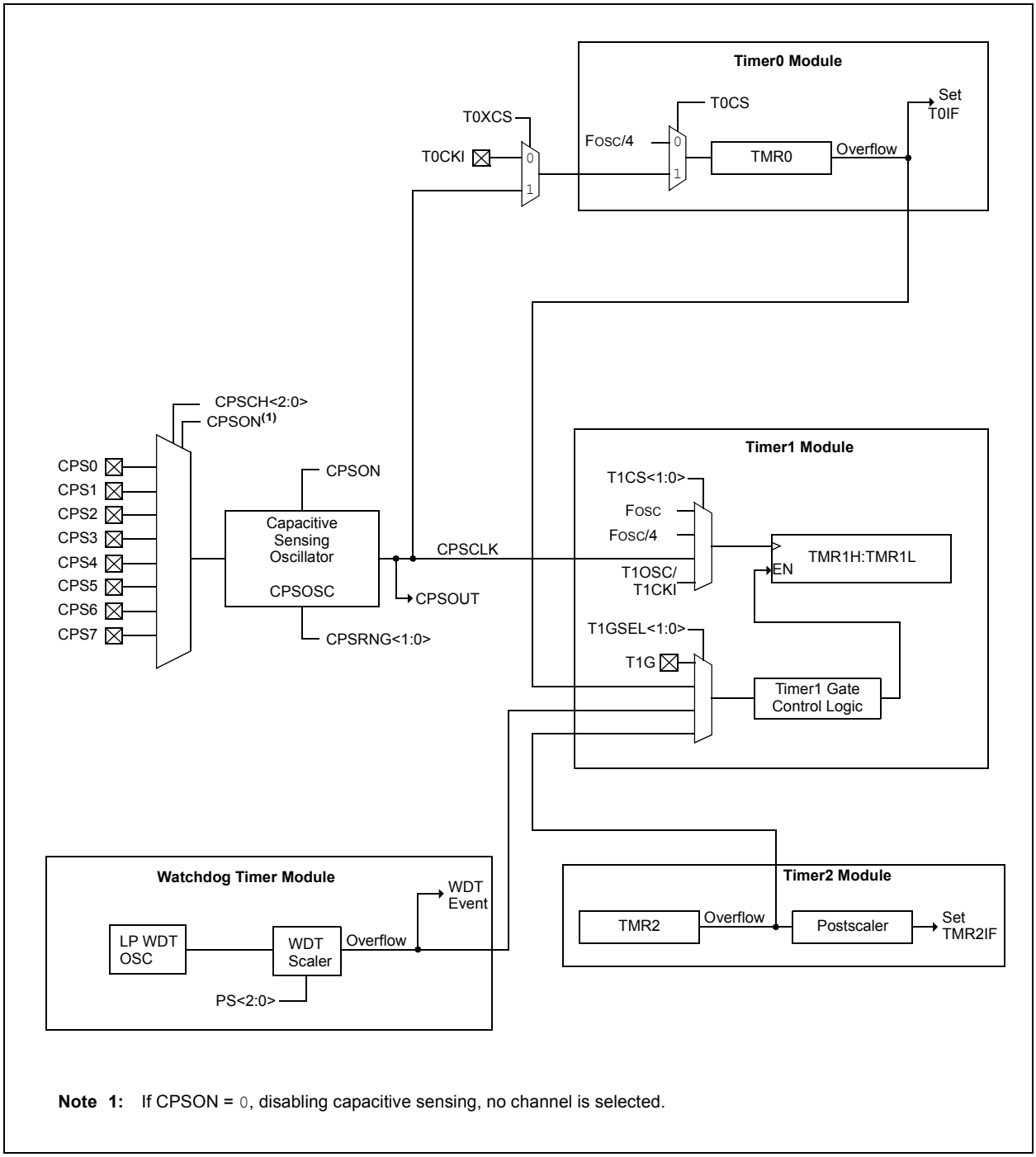
Legend: x = unknown, u = unchanged, – = unimplemented read as '0'. Shaded cells are not used for Timer2 module.

14.0 CAPACITIVE SENSING MODULE

The capacitive sensing module allows for an interaction with an end user without a mechanical interface. In a typical application, the capacitive sensing module is attached to a pad on a printed circuit board (PCB), which is electrically isolated from the end user. When the end user places their finger over the PCB pad, a capacitive load is added, causing a frequency shift in the capacitive sensing module. The capacitive sensing module requires software and at least one timer resource to determine the change in frequency. Key features of this module include:

- Analog MUX for monitoring multiple inputs
- Capacitive sensing oscillator
- Multiple timer resources
- Software control
- Operation during Sleep

FIGURE 14-1: CAPACITIVE SENSING BLOCK DIAGRAM



Note 1: If CPSON = 0, disabling capacitive sensing, no channel is selected.

14.1 Analog MUX

The capacitive sensing module can monitor up to 8 inputs. The capacitive sensing inputs are defined as CPS<7:0>. To determine if a frequency change has occurred the user must:

- Select the appropriate CPS pin by setting the CPSCH<2:0> bits of the CPSCON1 register
- Set the corresponding ANSEL bit
- Set the corresponding TRIS bit
- Run the software algorithm

Selection of the CPSx pin while the module is enabled will cause the capacitive sensing oscillator to be on the CPSx pin. Failure to set the corresponding ANSEL and TRIS bits can cause the capacitive sensing oscillator to stop, leading to false frequency readings.

14.2 Capacitive Sensing Oscillator

The capacitive sensing oscillator consists of a constant current source and a constant current sink, to produce a triangle waveform. The CPSOUT bit of the CPSCON0 register shows the status of the capacitive sensing oscillator, whether it is a sinking or sourcing current. The oscillator is designed to drive a capacitive load (single PCB pad) and at the same time, be a clock source to either Timer0 or Timer1. The oscillator has three different current settings as defined by CPSRNG<1:0> of the CPSCON0 register. The different current settings for the oscillator serve two purposes:

- Maximize the number of counts in a timer for a fixed time base
- Maximize the count differential in the timer during a change in frequency

14.3 Timer Resources

To measure the change in frequency of the capacitive sensing oscillator, a fixed time base is required. For the period of the fixed time base, the capacitive sensing oscillator is used to clock either Timer0 or Timer1. The frequency of the capacitive sensing oscillator is equal to the number of counts in the timer divided by the period of the fixed time base.

14.4 Fixed Time Base

To measure the frequency of the capacitive sensing oscillator, a fixed time base is required. Any timer resource or software loop can be used to establish the fixed time base. It is up to the end user to determine the method in which the fixed time base is generated.

Note:	The fixed time base can not be generated by the timer resource the capacitive sensing oscillator is clocking.

14.4.1 TIMER0

To select Timer0 as the timer resource for the capacitive sensing module:

- Set the T0XCS bit of the CPSCON0 register
- Clear the T0CS bit of the OPTION register

When Timer0 is chosen as the timer resource, the capacitive sensing oscillator will be the clock source for Timer0. Refer to **Section 11.0 "Timer0 Module"** for additional information.

14.4.2 TIMER1

To select Timer1 as the timer resource for the capacitive sensing module, set the TMR1CS<1:0> of the T1CON register to '11'. When Timer1 is chosen as the timer resource, the capacitive sensing oscillator will be the clock source for Timer1. Because the Timer1 module has a gate control, developing a time base for the frequency measurement can be simplified using either:

- The Timer0 overflow flag
- The Timer2 overflow flag
- The WDT overflow flag

It is recommended that one of these flags, in conjunction with the toggle mode of the Timer1 gate, is used to develop the fixed time base required by the software portion of the capacitive sensing module. Refer to **Section 12.0 "Timer1 Module with Gate Control"** for additional information.

TABLE 14-1: TIMER1 ENABLE FUNCTION

TMR1ON	TMR1GE	Timer1 Operation
0	0	Off
0	1	Off
1	0	On
1	1	Count Enabled by input

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)

14.6 Operation During Sleep

The capacitive sensing oscillator will continue to run as long as the module is enabled, independent of the part being in Sleep. In order for the software to determine if a frequency change has occurred, the part must be awake. However, the part does not have to be awake when the timer resource is acquiring counts. One way to acquire the Timer1 counts while in Sleep is to have Timer1 gated with the overflow of the Watchdog Timer. This can be accomplished using the following steps:

1. Configure the Watchdog Time-out overflow as the Timer1's gate source T1GSS<1:0> = 11.
2. Set Timer1 gate to toggle mode by setting the T1GTM bit of the T1GCON register.
3. Set the TMR1GE bit of the T1GCON register.
4. Set TMR1ON bit of the T1CON register.
5. Enable capacitive sensing module with the appropriate current settings and pin selection.
6. Clear Timer1.
7. Put the part to Sleep.
8. On the first WDT overflow, the capacitive sensing oscillator will begin to increment Timer1. Then put the part to Sleep.
9. On the second WDT overflow Timer1 will stop incrementing. Then run the software routine to determine if a frequency change has occurred.

Refer to **Section 12.0 "Timer1 Module with Gate Control"** for additional information.

Note 1:	When using the WDT to set the interval on Timer1, any other source that wakes the part up early will cause the WDT overflow to be delayed, affecting the value captured by Timer1.
2:	Timer0 does not operate when in Sleep, and therefore cannot be used for capacitive sense measurements in Sleep.

REGISTER 14-1: CPSCON0: CAPACITIVE SENSING CONTROL REGISTER 0

R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	R-0	R/W-0
CPSON	—	—	—	CPSRNG1	CPSRNG0	CPSOUT	T0XCS
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 **CPSON:** Capacitive Sensing Module Enable bit
1 = Capacitive sensing module is operating
0 = Capacitive sensing module is shut off and consumes no operating current

bit 6-4 **Unimplemented:** Read as '0'

bit 3-2 **CPSRNG<1:0>:** Capacitive Sensing Oscillator Range bits
00 = Oscillator is off.
01 = Oscillator is in low range. Charge/discharge current is nominally 0.1 µA.
10 = Oscillator is in medium range. Charge/discharge current is nominally 1.2 µA.
11 = Oscillator is in high range. Charge/discharge current is nominally 18 µA.

bit 1 **CPSOUT:** Capacitive Sensing Oscillator Status bit
1 = Oscillator is sourcing current (Current flowing out the pin)
0 = Oscillator is sinking current (Current flowing into the pin)

bit 0 **T0XCS:** Timer0 External Clock Source Select bit
If T0CS = 1
The T0XCS bit controls which clock external to the core/Timer0 module supplies Timer0:
1 = Timer0 Clock Source is the capacitive sensing oscillator
0 = Timer0 Clock Source is the T0CKI pin
If T0CS = 0
Timer0 clock source is controlled by the core/Timer0 module and is FOSC/4.

REGISTER 14-2: CPSCON1: CAPACITIVE SENSING CONTROL REGISTER 1

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	CPSCH2	CPSCH1	CPSCH0
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 7-3 **Unimplemented:** Read as '0'

bit 2-0 **CPSCH<2:0>**: Capacitive Sensing Channel Select bits

If CPSON = 0:

These bits are ignored. No channel is selected.

If CPSON = 1:

0000 = channel 0, (CPS0)
0001 = channel 1, (CPS1)
0010 = channel 2, (CPS2)
0011 = channel 3, (CPS3)
0100 = channel 4, (CPS4)
0101 = channel 5, (CPS5)
0110 = channel 6, (CPS6)
0111 = channel 7, (CPS7)

TABLE 14-2: SUMMARY OF REGISTERS ASSOCIATED WITH CAPACITIVE SENSING

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELA	—	—	ANSA5	ANSA4	ANSA3	ANSA2	ANSA1	ANSA0	49
ANSELB	—	—	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0	58
OPTION_REG	$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	23
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	41
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	43
T1CON	TMR1CS1	TMR1CS0	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	—	TMR1ON	112
T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	116
TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	48
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	57

Legend: – = Unimplemented locations, read as '0', u = unchanged, x = unknown. Shaded cells are not used by the capacitive sensing module.

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

TABLE 15-2: INTERACTION OF TWO CCP MODULES

CCP1 Mode	CCP2 Mode	Interaction
Capture	Capture	Same TMR1 time base
Capture	Compare	Same TMR1 time base(1, 2)
Compare	Compare	Same TMR1 time base(1, 2)
PWM	PWM	The PWMs will have the same frequency and update rate (TMR2 interrupt). The rising edges will be aligned.
PWM	Capture	None
PWM	Compare	None

Note 1: If CCP2 is configured as a Special Event Trigger, CCP1 will clear Timer1, affecting the value captured on the CCP2 pin.

2: If CCP1 is in Capture mode and CCP2 is configured as a Special Event Trigger, CCP2 will clear Timer1, affecting the value captured on the CCP1 pin.

Note:	CCPRx and CCPx throughout this document refer to CCPR1 or CCPR2 and CCP1 or CCP2, respectively.

REGISTER 15-1: CCPxCON: CCPx CONTROL REGISTER

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	DCxB1	DCxB0	CCPxM3	CCPxM2	CCPxM1	CCPxM0
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 **DCxB<1:0>**: 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 CCPRxL.

bit 3-0 **CCPxM<3:0>:** CCP Mode Select bits
0000 = Capture/Compare/PWM off (resets CCP module)
0001 = Unused (reserved)
0010 = Compare mode, toggle output on match (CCPxIF 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 (CCPxIF bit of the PIRx register is set)
1001 = Compare mode, clear output on match (CCPxIF bit of the PIRx register is set)
1010 = Compare mode, generate software interrupt on match (CCPxIF bit is set of the PIRx register, CCPx pin is unaffected)
1011 = Compare mode, trigger special event (CCPxIF bit of the PIRx register is set, TMR1 is reset and A/D conversion(1) is started if the ADC module is enabled. CCPx pin is unaffected.)
11xx = PWM mode.

Note 1: A/D conversion start feature is available only on CCP2.

15.1 Capture Mode

In Capture mode, CCPRxH:CCPRxL captures the 16-bit value of the TMR1 register when an event occurs on pin CCPx. An event is defined as one of the following and is configured by the CCPxM<3:0> bits of the CCPxCON register:

- Every falling edge
- Every rising edge
- Every 4th rising edge
- Every 16th rising edge

When a capture is made, the Interrupt Request Flag bit CCPxIF of the PIRx register is set. The interrupt flag must be cleared in software. If another capture occurs before the value in the CCPRxH, CCPRxL register pair is read, the old captured value is overwritten by the new captured value (refer to Figure 15-1).

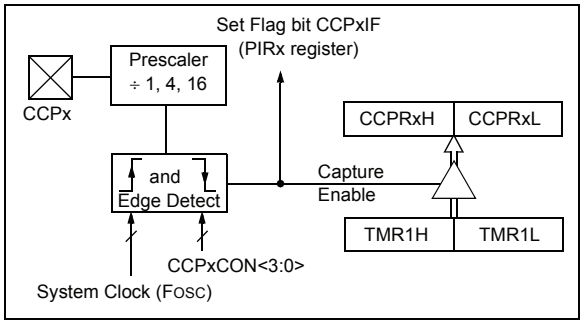
15.1.1 CCPx PIN CONFIGURATION

In Capture mode, the CCPx pin should be configured as an input by setting the associated TRIS control bit.

Either RC1 or RB3 can be selected as the CCP2 pin. Refer to **Section 6.1 "Alternate Pin Function"** for more information.

Note:	If the CCPx pin is configured as an output, a write to the port can cause a capture condition.

FIGURE 15-1: CAPTURE MODE OPERATION BLOCK DIAGRAM



15.1.2 TIMER1 MODE SELECTION

Timer1 must be running in Timer mode or Synchronized Counter mode for the CCP module to use the capture feature. In Asynchronous Counter mode or when Timer1 is clocked at FOSC, the capture operation may not work.

15.1.3 SOFTWARE INTERRUPT

When the Capture mode is changed, a false capture interrupt may be generated. The user should keep the CCPxIE interrupt enable bit of the PIEx register clear to avoid false interrupts. Additionally, the user should clear the CCPxIF interrupt flag bit of the PIRx register following any change in operating mode.

Note:	Clocking Timer1 from the system clock (FOSC) should not be used in Capture mode. In order for Capture mode to recognize the trigger event on the CCPx pin, Timer1 must be clocked from the instruction clock (FOSC/4) or from an external clock source.

15.1.4 CCP PRESCALER

There are four prescaler settings specified by the CCPxM<3:0> bits of the CCPxCON register. Whenever the CCP module is turned off, or the CCP module is not in Capture mode, the prescaler counter is cleared. Any Reset will clear the prescaler counter.

Switching from one capture prescaler to another does not clear the prescaler and may generate a false interrupt. To avoid this unexpected operation, turn the module off by clearing the CCPxCON register before changing the prescaler (refer to Example 15-1).

EXAMPLE 15-1: CHANGING BETWEEN CAPTURE PRESCALERS

```
BANKSEL CCP1CON     ;Set Bank bits to point
                    ;to CCP1CON
CLRF    CCP1CON     ;Turn CCP module off
MOVLW   NEW_CAPT_PS;Load the W reg with
                    ; the new prescaler
                    ; move value and CCP ON
MOVWF   CCP1CON     ;Load CCP1CON with this
                    ; value
```

15.1.5 CAPTURE DURING SLEEP

Capture mode depends upon the Timer1 module for proper operation. There are two options for driving the Timer1 module in Capture mode. It can be driven by the instruction clock (FOSC/4), or by an external clock source.

If Timer1 is clocked by FOSC/4, then Timer1 will not increment during Sleep. When the device wakes from Sleep, Timer1 will continue from its previous state.

If Timer1 is clocked by an external clock source, then Capture mode will operate as defined in **Section 15.1 "Capture Mode"**.

TABLE 15-3: SUMMARY OF REGISTERS ASSOCIATED WITH CAPTURE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELB	—	—	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0	58
APFCON	—	—	—	—	—	—	SSSEL	CCP2SEL	47
CCP1CON	—	—	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	124
CCP2CON	—	—	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	124
CCPRxL	Capture/Compare/PWM Register X Low Byte								125
CCPRxH	Capture/Compare/PWM Register X High Byte								125
INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	40
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	41
PIE2	—	—	—	—	—	—	—	CCP2IE	42
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	43
PIR2	—	—	—	—	—	—	—	CCP2IF	44
T1CON	TMR1CS1	TMR1CS0	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	—	TMR1ON	112
T1GCON	TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/$\overline{\text{DONE}}$	T1GVAL	T1GSS1	T1GSS0	113
TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Register								108
TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Register								108
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	57
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	67

Legend: – = Unimplemented locations, read as '0', u = unchanged, x = unknown. Shaded cells are not used by the Capture.

15.2 Compare Mode

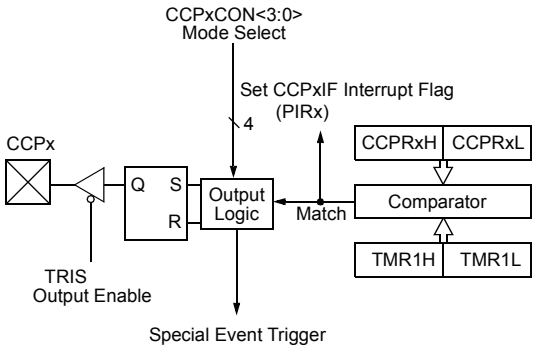
In Compare mode, the 16-bit CCPRx register value is constantly compared against the TMR1 register pair value. When a match occurs, the CCPx module may:

- Toggle the CCPx output
- Set the CCPx output
- Clear the CCPx output
- Generate a Special Event Trigger
- Generate a Software Interrupt

The action on the pin is based on the value of the CCPxM<3:0> control bits of the CCPxCON register.

All Compare modes can generate an interrupt.

FIGURE 15-2: COMPARE MODE OPERATION BLOCK DIAGRAM



Special Event Trigger will:

- Clear TMR1H and TMR1L registers.
- NOT set interrupt flag bit TMR1IF of the PIR1 register.
- Set the GO/$\overline{\text{DONE}}$ bit to start the ADC conversion (CCP2 only).

15.2.1 CCPx PIN CONFIGURATION

The user must configure the CCPx pin as an output by clearing the associated TRIS bit.

Either RC1 or RB3 can be selected as the CCP2 pin. Refer to **Section 6.1 "Alternate Pin Function"** for more information.

Note:	Clearing the CCPxCON register will force the CCPx compare output latch to the default low level. This is not the PORT I/O data latch.

15.2.2 TIMER1 MODE SELECTION

In Compare mode, Timer1 must be running in either Timer mode or Synchronized Counter mode. The compare operation may not work in Asynchronous Counter mode.

Note:	Clocking Timer1 from the system clock (FOSC) should not be used in Compare mode. For the Compare operation of the TMR1 register to the CCPRx register to occur, Timer1 must be clocked from the instruction clock (FOSC/4) or from an external clock source.

15.2.3 SOFTWARE INTERRUPT MODE

When Software Interrupt mode is chosen (CCPxM<3:0> = 1010), the CCPxIF bit in the PIRx register is set and the CCPx module does not assert control of the CCPx pin (refer to the CCPxCON register).

15.2.4 SPECIAL EVENT TRIGGER

When Special Event Trigger mode is chosen (CCPxM<3:0> = 1011), the CCPx module does the following:

- Resets Timer1
- Starts an ADC conversion if ADC is enabled (CCP2 only)

The CCPx module does not assert control of the CCPx pin in this mode (refer to the CCPxCON register).

The Special Event Trigger output of the CCP occurs immediately upon a match between the TMR1H, TMR1L register pair and the CCPRxH, CCPRxL register pair. The TMR1H, TMR1L register pair is not reset until the next rising edge of the Timer1 clock. This allows the CCPRxH, CCPRxL register pair to effectively provide a 16-bit programmable period register for Timer1.

Note 1:	The Special Event Trigger from the CCP module does not set interrupt flag bit TMR1IF of the PIR1 register.
2:	Removing the match condition by changing the contents of the CCPRxH and CCPRxL register pair, between the clock edge that generates the Special Event Trigger and the clock edge that generates the Timer1 Reset, will preclude the Reset from occurring.

15.2.5 COMPARE DURING SLEEP

The Compare mode is dependent upon the system clock (FOSC) for proper operation. Since FOSC is shut down during Sleep mode, the Compare mode will not function properly during Sleep.

TABLE 15-4: SUMMARY OF REGISTERS ASSOCIATED WITH COMPARE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ADCON0	—	—	CHS3	CHS2	CHS1	CHS0	GO/$\overline{\text{DONE}}$	ADON	92
ANSELB	—	—	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0	58
APFCON	—	—	—	—	—	—	SSSEL	CCP2SEL	47
CCP1CON	—	—	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	124
CCP2CON	—	—	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	124
CCPRxL	Capture/Compare/PWM Register X Low Byte								125
CCPRxH	Capture/Compare/PWM Register X High Byte								125
INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	40
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	41
PIE2	—	—	—	—	—	—	—	CCP2IE	42
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	43
PIR2	—	—	—	—	—	—	—	CCP2IF	44
T1CON	TMR1CS1	TMR1CS0	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	—	TMR1ON	112
T1GCON	TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/$\overline{\text{DONE}}$	T1GVAL	T1GSS1	T1GSS0	113
TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Register								108
TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Register								108
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	57
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	67

Legend: – = Unimplemented locations, read as '0', u = unchanged, x = unknown. Shaded cells are not used by the Compare.

15.3 PWM Mode

The PWM mode generates a Pulse-Width Modulated signal on the CCPx pin. The duty cycle, period and resolution are determined by the following registers:

- PR2
- T2CON
- CCPRxL
- CCPxCON

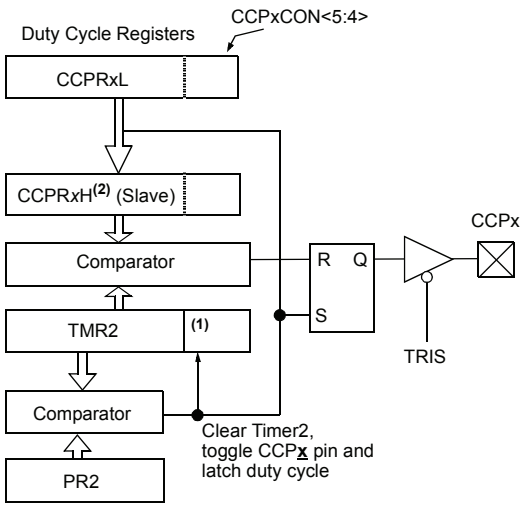
In Pulse-Width Modulation (PWM) mode, the CCP module produces up to a 10-bit resolution PWM output on the CCPx pin.

Figure 15-3 shows a simplified block diagram of PWM operation.

Figure 15-4 shows a typical waveform of the PWM signal.

For a step-by-step procedure on how to set up the CCP module for PWM operation, refer to **Section 15.3.8 "Setup for PWM Operation"**.

FIGURE 15-3: SIMPLIFIED PWM BLOCK DIAGRAM

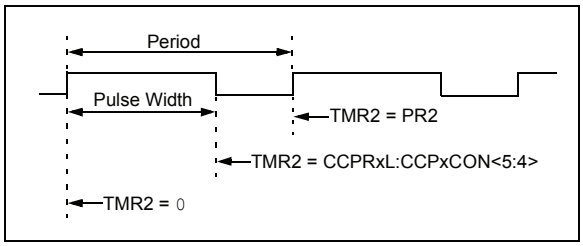


Note 1: The 8-bit timer TMR2 register is concatenated with the 2-bit internal system clock (FOSC), or 2 bits of the prescaler, to create the 10-bit time base.

2: In PWM mode, CCPRxH is a read-only register.

The PWM output (Figure 15-4) has a time base (period) and a time that the output stays high (duty cycle).

FIGURE 15-4: CCP PWM OUTPUT



15.3.1 CCPX PIN CONFIGURATION

In PWM mode, the CCPx pin is multiplexed with the PORT data latch. The user must configure the CCPx pin as an output by clearing the associated TRIS bit.

Either RC1 or RB3 can be selected as the CCP2 pin. Refer to **Section 6.1 "Alternate Pin Function"** for more information.

Note: Clearing the CCPxCON register will relinquish CCPx control of the CCPx pin.

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

$$PWM\ Period = [(PR2) + 1] \bullet 4 \bullet TOSC \bullet (TMR2\ Prescale\ Value)$$

Note: TOSC = 1/FOSC

When TMR2 is equal to PR2, the following three events occur on the next increment cycle:

- TMR2 is cleared
- The CCPx pin is set. (Exception: If the PWM duty cycle = 0%, the pin will not be set.)
- The PWM duty cycle is latched from CCPRxL into CCPRxH.

Note: The Timer2 postscaler (refer to **Section 13.1 "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: CCPRxL register and DCxB<1:0> bits of the CCPxCON register. The CCPRxL contains the eight MSbs and the DCxB<1:0> bits of the CCPxCON register contain the two LSbs. CCPRxL and DCxB<1:0> bits of the CCPxCON register can be written to at any time. The duty cycle value is not latched into CCPRxH until after the period completes (i.e., a match between PR2 and TMR2 registers occurs). While using the PWM, the CCPRxH 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

$$Pulse\ Width = (CCPRxL:CCPxCON<5:4>) \bullet TOSC \bullet (TMR2\ Prescale\ Value)$$

Note: TOSC = 1/FOSC

EQUATION 15-3: DUTY CYCLE RATIO

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

The CCPRxH 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 (FOSC), or 2 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 CCPRxH and 2-bit latch, then the CCPx pin is cleared (refer to Figure 15-3).

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 10 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.

TABLE 15-7: SUMMARY OF REGISTERS ASSOCIATED WITH PWM

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELB	—	—	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0	58
APFCON	—	—	—	—	—	—	SSSEL	CCP2SEL	47
CCP1CON	—	—	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	124
CCP2CON	—	—	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	124
CCPRxL	Capture/Compare/PWM Register X Low Byte								125
CCPRxH	Capture/Compare/PWM Register X High Byte								125
PR2	Timer2 Period Register								115
T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	116
TMR2	Timer2 Module Register								115
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	57
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	67

Legend: – = Unimplemented locations, read as '0', u = unchanged, x = unknown. Shaded cells are not used by the PWM.

16.0 ADDRESSABLE UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER (AUSART)

The Addressable Universal Synchronous Asynchronous Receiver Transmitter (AUSART) module is a serial I/O communications peripheral. It contains all the clock generators, shift registers and data buffers necessary to perform an input or output serial data transfer independent of device program execution. The AUSART, also known as a Serial Communications Interface (SCI), can be configured as a full-duplex asynchronous system or half-duplex synchronous system. Full-Duplex mode is useful for communications with peripheral systems, such as CRT terminals and personal computers. Half-Duplex Synchronous mode is intended for communications with peripheral devices, such as A/D or D/A integrated circuits, serial EEPROMs or other microcontrollers. These devices typically do not have internal clocks for baud rate generation and require the external clock signal provided by a master synchronous device.

The AUSART module includes the following capabilities:

- Full-duplex asynchronous transmit and receive
- Two-character input buffer
- One-character output buffer
- Programmable 8-bit or 9-bit character length
- Address detection in 9-bit mode
- Input buffer overrun error detection
- Received character framing error detection
- Half-duplex synchronous master
- Half-duplex synchronous slave
- Sleep operation

Block diagrams of the AUSART transmitter and receiver are shown in Figure 16-1 and Figure 16-2.

FIGURE 16-1: AUSART TRANSMIT BLOCK DIAGRAM

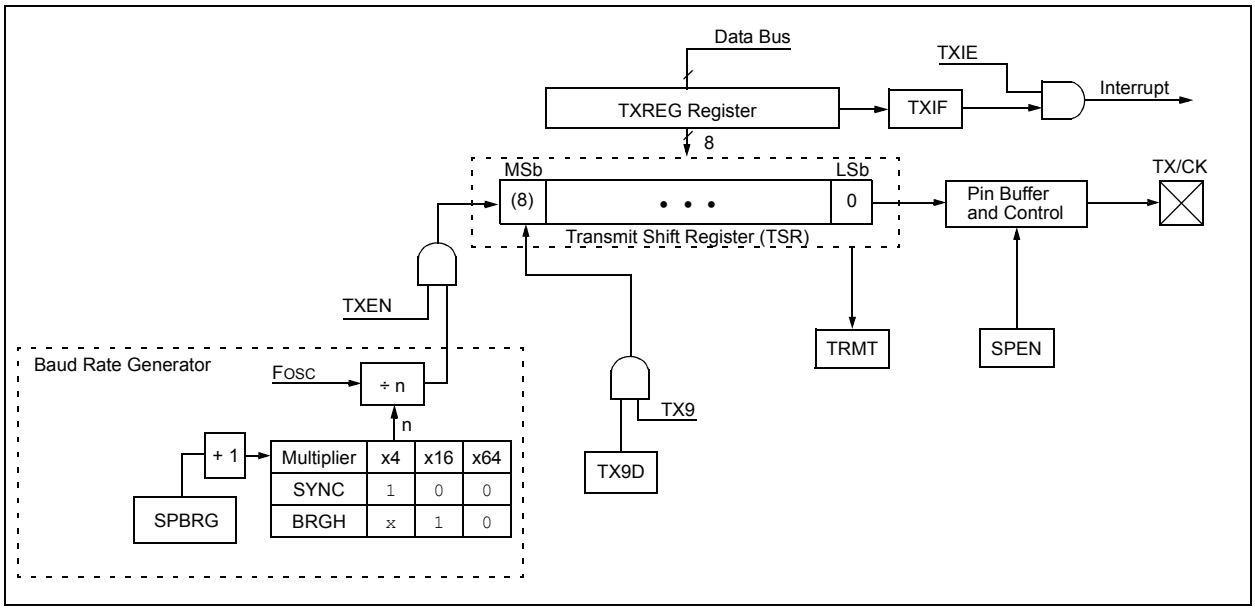
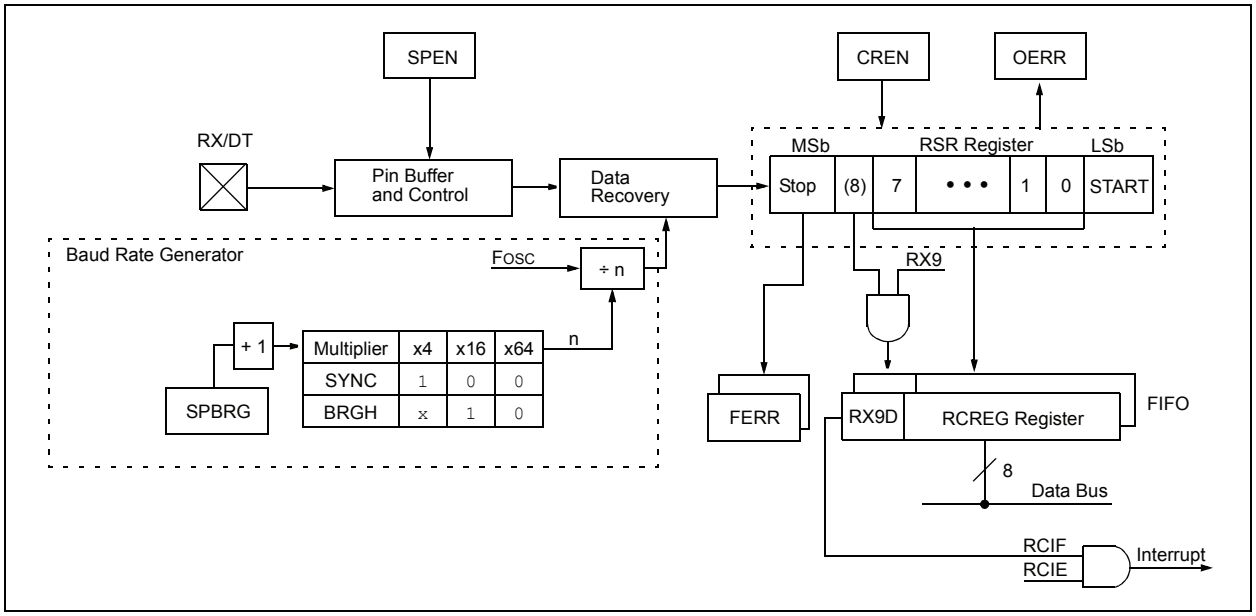


FIGURE 16-2: AUSART RECEIVE BLOCK DIAGRAM



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 VOH mark state which represents a '1' data bit, and a VOL 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 8 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 TCY 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 the TXREG. In other words, the TXIF bit is only clear when the TSR is busy with a character and a new character has been queued for transmission in the 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 the TXREG is empty, regardless of the state of 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 the TXREG.

16.1.1.4 TSR Status

The TRMT bit of the TXSTA register indicates the status of the TSR register. This is a read-only bit. The TRMT bit is set when the TSR register is empty and is cleared when a character is transferred to the TSR register from the TXREG. The TRMT bit remains clear until all bits have been shifted out of the TSR register. No interrupt logic is tied to this bit, so the user has to poll this bit to determine the TSR status.

Note:	The TSR register is not mapped in data memory, so it is not available to the user.

16.1.1.5 Transmitting 9-Bit Characters

The AUSART supports 9-bit character transmissions. When the TX9 bit of the TXSTA register is set, the AUSART will shift 9 bits out for each character transmitted. The TX9D bit of the TXSTA register is the ninth, and Most Significant, data bit. When transmitting 9-bit data, the TX9D data bit must be written before writing the 8 Least Significant bits into the TXREG. All nine bits of data will be transferred to the TSR shift register immediately after the TXREG is written.

A special 9-bit Address mode is available for use with multiple receivers. Refer to **Section 16.1.2.7 "Address Detection"** for more information on the Address mode.

16.1.1.6 Asynchronous Transmission Set-up:

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 asynchronous serial port by clearing the SYNC bit and setting the SPEN bit.
3. If 9-bit transmission is desired, set the TX9 control bit. A set ninth data bit will indicate that the 8 Least Significant data bits are an address when the receiver is set for address detection.
4. Enable the transmission by setting the TXEN control bit. This will cause the TXIF interrupt bit to be set.
5. If interrupts are desired, set the TXIE interrupt enable bit of the PIE1 register. An interrupt will occur immediately provided that the GIE and PEIE bits of the INTCON register are also set.
6. If 9-bit transmission is selected, the ninth bit should be loaded into the TX9D data bit.
7. Load 8-bit data into the TXREG register. This will start the transmission.

FIGURE 16-3: ASYNCHRONOUS TRANSMISSION

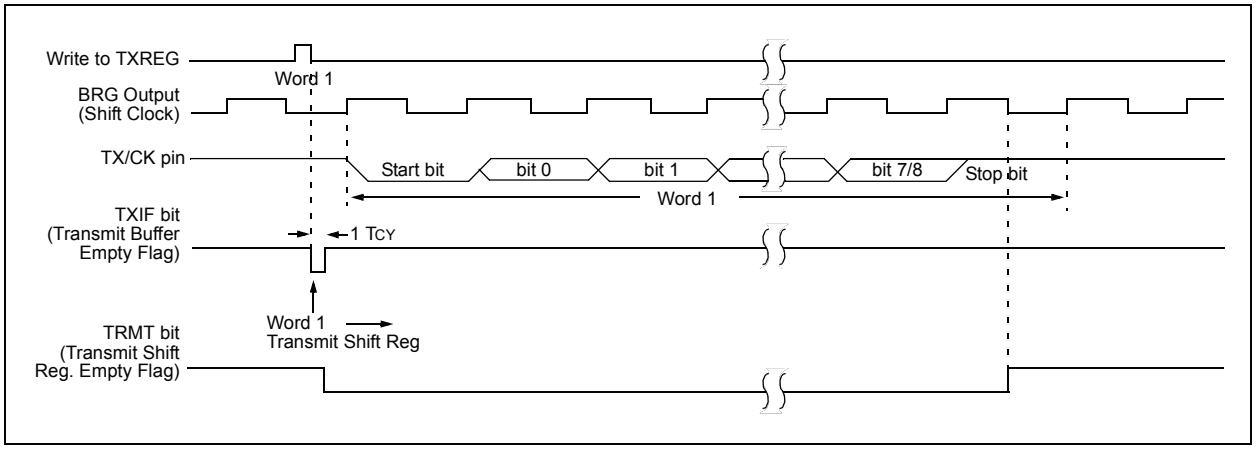
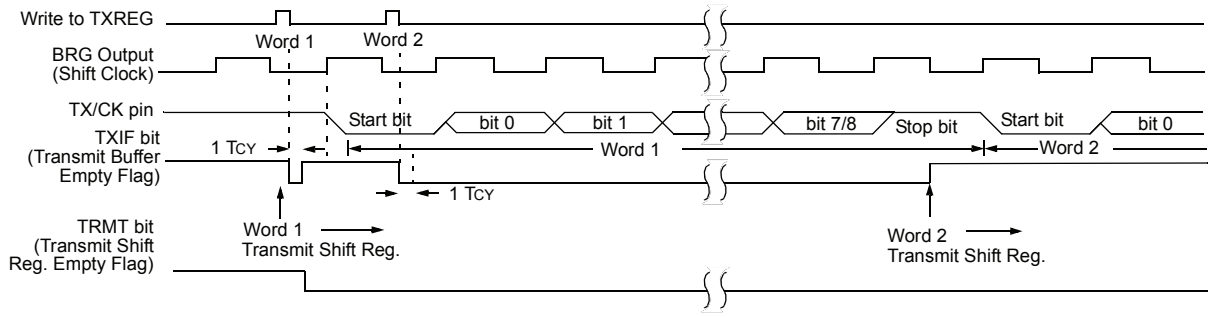


FIGURE 16-4: ASYNCHRONOUS TRANSMISSION (BACK-TO-BACK)



Note: This timing diagram shows two consecutive transmissions.

TABLE 16-1: REGISTERS ASSOCIATED WITH ASYNCHRONOUS TRANSMISSION

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
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
TXREG	AUSART Transmit Data Register								0000 0000	0000 0000
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 Transmission.

16.1.2 AUSART ASYNCHRONOUS RECEIVER

The Asynchronous mode is typically used in RS-232 systems. The receiver block diagram is shown in Figure 16-2. The data is received on the RX/DT pin and drives the data recovery block. The data recovery block is actually a high-speed shifter operating at 16 times the baud rate, whereas the serial Receive Shift Register (RSR) operates at the bit rate. When all 8 or 9 bits of the character have been shifted in, they are immediately transferred to a two character First-In First-Out (FIFO) memory. The FIFO buffering allows reception of two complete characters and the start of a third character before software must start servicing the AUSART receiver. The FIFO and RSR registers are not directly accessible by software. Access to the received data is via the RCREG register.

16.1.2.1 Enabling the Receiver

The AUSART receiver is enabled for asynchronous operation by configuring the following three control bits:

- CREN = 1
- SYNC = 0
- SPEN = 1

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

Setting the CREN bit of the RCSTA register enables the receiver 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 RX/DT I/O pin as an input.

Note:	When the SPEN bit is set the TX/CK I/O pin is automatically configured as an output, regardless of the state of the corresponding TRIS bit and whether or not the AUSART transmitter is enabled. The PORT latch is disconnected from the output driver so it is not possible to use the TX/CK pin as a general purpose output.

16.1.2.2 Receiving Data

The receiver data recovery circuit initiates character reception on the falling edge of the first bit. The first bit, also known as the Start bit, is always a zero. The data recovery circuit counts one-half bit time to the center of the Start bit and verifies that the bit is still a zero. If it is not a zero then the data recovery circuit aborts character reception, without generating an error, and resumes looking for the falling edge of the Start bit. If the Start bit zero verification succeeds then the data recovery circuit counts a full bit time to the center of the next bit. The bit is then sampled by a majority detect circuit and the resulting '0' or '1' is shifted into the RSR. This repeats until all data bits have been sampled and shifted into the RSR. One final bit time is measured and the level sampled. This is the Stop bit, which is always a '1'. If the data recovery circuit samples a '0' in the Stop bit position then a framing error is set for this character, otherwise the framing error is cleared for this character. Refer to **Section 16.1.2.4 "Receive Framing Error"** for more information on framing errors.

Immediately after all data bits and the Stop bit have been received, the character in the RSR is transferred to the AUSART receive FIFO and the RCIF interrupt flag bit of the PIR1 register is set. The top character in the FIFO is transferred out of the FIFO by reading the RCREG register.

Note:	If the receive FIFO is overrun, no additional characters will be received until the overrun condition is cleared. Refer to ***Section 16.1.2.5 "Receive Overrun Error"*** for more information on overrun errors.

16.1.2.3 Receive Interrupts

The RCIF interrupt flag bit of the PIR1 register is set whenever the AUSART receiver is enabled and there is an unread character in the receive FIFO. The RCIF interrupt flag bit is read-only, it cannot be set or cleared by software.

RCIF interrupts are enabled by setting all of the following bits:

- RCIE, Receive Interrupt Enable bit of the PIE1 register
- PEIE, Peripheral Interrupt Enable bit of the INTCON register
- GIE, Global Interrupt Enable bit of the INTCON register

The RCIF interrupt flag bit of the PIR1 register will be set when there is an unread character in the FIFO, regardless of the state of interrupt enable bits.

16.1.2.4 Receive Framing Error

Each character in the receive FIFO buffer has a corresponding framing error Status bit. A framing error indicates that a Stop bit was not seen at the expected time. The framing error status is accessed via the FERR bit of the RCSTA register. The FERR bit represents the status of the top unread character in the receive FIFO. Therefore, the FERR bit must be read before reading the RCREG.

The FERR bit is read-only and only applies to the top unread character in the receive FIFO. A framing error (FERR = 1) does not preclude reception of additional characters. It is not necessary to clear the FERR bit. Reading the next character from the FIFO buffer will advance the FIFO to the next character and the next corresponding framing error.

The FERR bit can be forced clear by clearing the SPEN bit of the RCSTA register which resets the AUSART. Clearing the CREN bit of the RCSTA register does not affect the FERR bit. A framing error by itself does not generate an interrupt.

Note:	If all receive characters in the receive FIFO have framing errors, repeated reads of the RCREG will not clear the FERR bit.

16.1.2.5 Receive Overrun Error

The receive FIFO buffer can hold two characters. An overrun error will be generated if a third character, in its entirety, is received before the FIFO is accessed. When this happens the OERR bit of the RCSTA register is set. The characters already in the FIFO buffer can be read but no additional characters will be received until the error is cleared. The error must be cleared by either clearing the CREN bit of the RCSTA register or by setting the AUSART by clearing the SPEN bit of the RCSTA register.

16.1.2.6 Receiving 9-bit Characters

The AUSART supports 9-bit character reception. When the RX9 bit of the RCSTA register is set the AUSART will shift 9 bits into the RSR for each character received. The RX9D bit of the RCSTA register is the ninth and Most Significant data bit of the top unread character in the receive FIFO. When reading 9-bit data from the receive FIFO buffer, the RX9D data bit must be read before reading the 8 Least Significant bits from the RCREG.

16.1.2.7 Address Detection

A special Address Detection mode is available for use when multiple receivers share the same transmission line, such as in RS-485 systems. Address detection is enabled by setting the ADDEN bit of the RCSTA register.

Address detection requires 9-bit character reception. When address detection is enabled, only characters with the ninth data bit set will be transferred to the receive FIFO buffer, thereby setting the RCIF interrupt bit of the PIR1 register. All other characters will be ignored.

Upon receiving an address character, user software determines if the address matches its own. Upon address match, user software must disable address detection by clearing the ADDEN bit before the next Stop bit occurs. When user software detects the end of the message, determined by the message protocol used, software places the receiver back into the Address Detection mode by setting the ADDEN bit.

16.1.2.8 Asynchronous Reception Set-up:

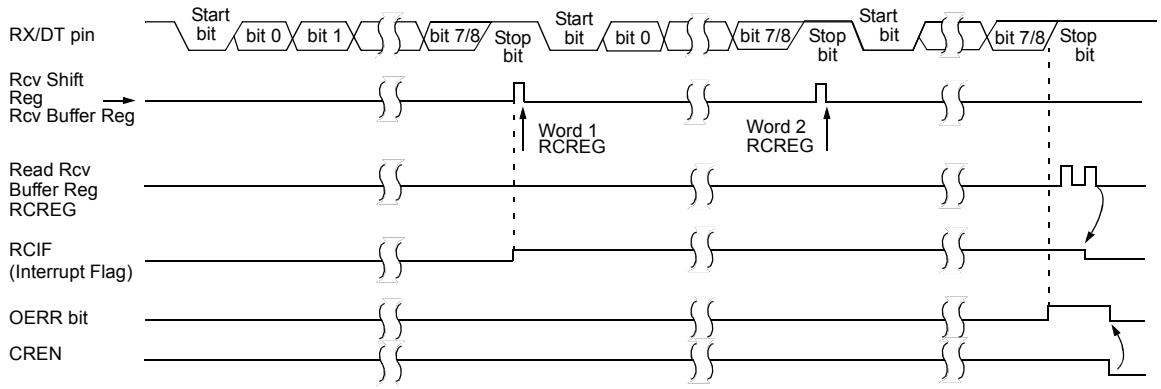
1. Initialize the SPBRG register and the BRGH bit to achieve the desired baud rate (refer to **Section 16.2 "AUSART Baud Rate Generator (BRG)"**).
2. Enable the serial port by setting the SPEN bit. The SYNC bit must be clear for asynchronous operation.
3. If interrupts are desired, set the RCIE bit of the PIE1 register and the GIE and PEIE bits of the INTCON register.
4. If 9-bit reception is desired, set the RX9 bit.
5. Enable reception by setting the CREN bit.
6. The RCIF interrupt flag bit of the PIR1 register will be set when a character is transferred from the RSR to the receive buffer. An interrupt will be generated if the RCIE bit of the PIE1 register was also set.
7. Read the RCSTA register to get the error flags and, if 9-bit data reception is enabled, the ninth data bit.
8. Get the received 8 Least Significant data bits from the receive buffer by reading the RCREG register.
9. If an overrun occurred, clear the OERR flag by clearing the CREN receiver enable bit.

16.1.2.9 9-bit Address Detection Mode Set-up

This mode would typically be used in RS-485 systems. To set up an Asynchronous Reception with Address Detect Enable:

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 serial port by setting the SPEN bit. The SYNC bit must be clear for asynchronous operation.
3. If interrupts are desired, set the RCIE bit of the PIE1 register and the GIE and PEIE bits of the INTCON register.
4. Enable 9-bit reception by setting the RX9 bit.
5. Enable address detection by setting the ADDEN bit.
6. Enable reception by setting the CREN bit.
7. The RCIF interrupt flag bit of the PIR1 register will be set when a character with the ninth bit set is transferred from the RSR to the receive buffer. An interrupt will be generated if the RCIE interrupt enable bit of the PIE1 register was also set.
8. Read the RCSTA register to get the error flags. The ninth data bit will always be set.
9. Get the received 8 Least Significant data bits from the receive buffer by reading the RCREG register. Software determines if this is the device's address.
10. If an overrun occurred, clear the OERR flag by clearing the CREN receiver enable bit.
11. If the device has been addressed, clear the ADDEN bit to allow all received data into the receive buffer and generate interrupts.

FIGURE 16-5: ASYNCHRONOUS RECEPTION



Note: This timing diagram shows three words appearing on the RX input. The RCREG (receive buffer) is read after the third word, causing the OERR (overrun) bit to be set.

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.

REGISTER 16-1: TXSTA: TRANSMIT STATUS AND CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN(1)	SYNC	—	BRGH	TRMT	TX9D
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 **CSRC:** Clock Source Select bit
Asynchronous mode:
Don't care
Synchronous mode:
1 = Master mode (clock generated internally from BRG)
0 = Slave mode (clock from external source)

bit 6 **TX9:** 9-bit Transmit Enable bit
1 = Selects 9-bit transmission
0 = Selects 8-bit transmission

bit 5 **TXEN:** Transmit Enable bit(1)
1 = Transmit enabled
0 = Transmit disabled

bit 4 **SYNC:** AUSART Mode Select bit
1 = Synchronous mode
0 = Asynchronous mode

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

bit 2 **BRGH:** High Baud Rate Select bit
Asynchronous mode:
1 = High speed
0 = Low speed
Synchronous mode:
Unused in this mode

bit 1 **TRMT:** Transmit Shift Register Status bit
1 = TSR empty
0 = TSR full

bit 0 **TX9D:** Ninth bit of Transmit Data
Can be address/data bit or a parity bit.

Note 1: SREN/CREN overrides TXEN in Synchronous mode.

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.

16.2 AUSART Baud Rate Generator (BRG)

The Baud Rate Generator (BRG) is an 8-bit timer that is dedicated to the support of both the asynchronous and synchronous AUSART operation.

The SPBRG register determines the period of the free running baud rate timer. In Asynchronous mode the multiplier of the baud rate period is determined by the BRGH bit of the TXSTA register. In Synchronous mode, the BRGH bit is ignored.

Table 16-3 contains the formulas for determining the baud rate. Example 16-1 provides a sample calculation for determining the baud rate and baud rate error.

Typical baud rates and error values for various asynchronous modes have been computed for your convenience and are shown in Table 16-3. It may be advantageous to use the high baud rate (BRGH = 1), to reduce the baud rate error.

Writing a new value to the SPBRG register causes the BRG timer to be reset (or cleared). This ensures that the BRG does not wait for a timer overflow before outputting the new baud rate.

EXAMPLE 16-1: CALCULATING BAUD RATE ERROR

For a device with FOSC of 16 MHz, desired baud rate of 9600, and Asynchronous mode with SYNC = 0 and BRGH = 0 (as seen in Table 16-3):

$$Desired\ Baud\ Rate = \frac{FOSC}{64(SPBRG + 1)}$$

Solving for SPBRG:

$$SPBRG = \left(\frac{FOSC}{64(Desired\ Baud\ Rate)}\right) - 1$$

$$= \left(\frac{16000000}{64(9600)}\right) - 1$$

$$= [25.042] = 25$$

$$Actual\ Baud\ Rate = \frac{16000000}{64(25 + 1)}$$

$$= 9615$$

$$\%\ Error = \left(\frac{Actual\ Baud\ Rate - Desired\ Baud\ Rate}{Desired\ Baud\ Rate}\right) 100$$

$$= \left(\frac{9615 - 9600}{9600}\right) 100 = 0.16\%$$

TABLE 16-3: BAUD RATE FORMULAS

Configuration Bits		AUSART Mode	Baud Rate Formula
SYNC	BRGH		
0	0	Asynchronous	FOSC/[64 (n+1)]
0	1	Asynchronous	FOSC/[16 (n+1)]
1	x	Synchronous	FOSC/[4 (n+1)]

Legend: x = Don't care, n = value of SPBRG register

TABLE 16-4: REGISTERS ASSOCIATED WITH THE BAUD RATE GENERATOR

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
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
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 the Baud Rate Generator.

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

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	—	—	—

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.

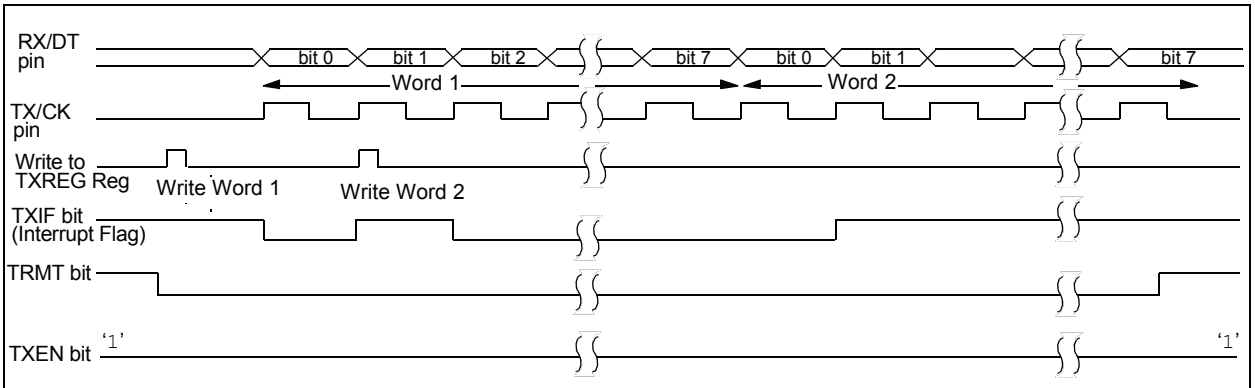
Each data bit changes on the leading edge of the master clock and remains valid until the subsequent leading clock edge.

Note:	The TSR register is not mapped in data memory, so it is not available to the user.

16.3.1.3 Synchronous Master Transmission Set-up:

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.

FIGURE 16-6: SYNCHRONOUS TRANSMISSION



Note: Synchronous Master mode, SPBRG = 0, continuous transmission of two 8-bit words.

FIGURE 16-7: SYNCHRONOUS TRANSMISSION (THROUGH TXEN)

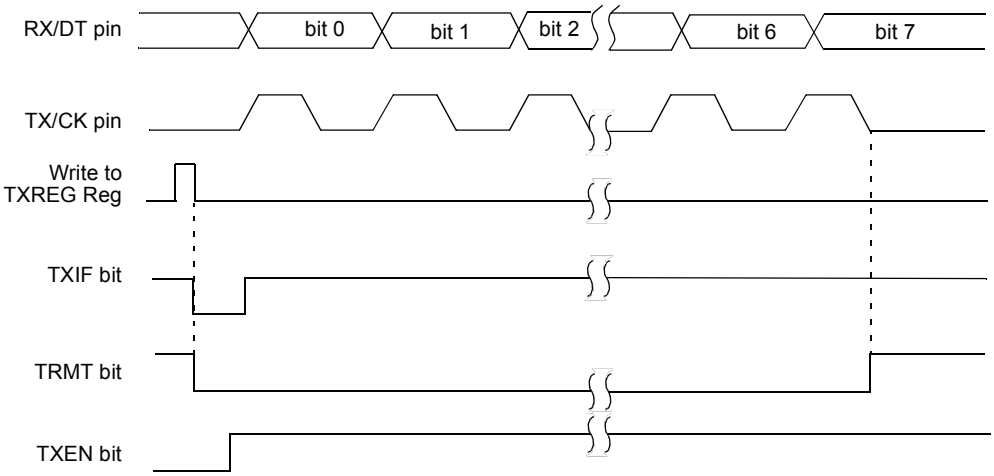


TABLE 16-6: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER TRANSMISSION

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
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
TXREG	AUSART Transmit Data Register								0000 0000	0000 0000
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 Transmission.

16.3.1.4 Synchronous Master Reception

Data is received at the RX/DT pin. The RX/DT pin output driver is automatically disabled when the AUSART is configured for synchronous master receive operation.

In Synchronous mode, reception is enabled by setting either the Single Receive Enable bit (SREN of the RCSTA register) or the Continuous Receive Enable bit (CREN of the RCSTA register).

When SREN is set and CREN is clear, only as many clock cycles are generated as there are data bits in a single character. The SREN bit is automatically cleared at the completion of one character. When CREN is set, clocks are continuously generated until CREN is cleared. If CREN is cleared in the middle of a character the CK clock stops immediately and the partial character is discarded. If SREN and CREN are both set, then SREN is cleared at the completion of the first character and CREN takes precedence.

To initiate reception, set either SREN or CREN. Data is sampled at the RX/DT pin on the trailing edge of the TX/CK clock pin and is shifted into the Receive Shift Register (RSR). When a complete character is received into the RSR, the RCIF bit of the PIR1 register is set and the character is automatically transferred to the two character receive FIFO. The Least Significant eight bits of the top character in the receive FIFO are available in RCREG. The RCIF bit remains set as long as there are un-read characters in the receive FIFO.

16.3.1.5 Slave Clock

Synchronous data transfers use a separate clock line, which is synchronous with the data. A device configured as a slave receives the clock on the TX/CK line. The TX/CK pin output driver is automatically disabled when the device is configured for synchronous slave 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 data bit is transferred for each clock cycle. Only as many clock cycles should be received as there are data bits.

16.3.1.6 Receive Overrun Error

The receive FIFO buffer can hold two characters. An overrun error will be generated if a third character, in its entirety, is received before RCREG is read to access the FIFO. When this happens the OERR bit of the RCSTA register is set. Previous data in the FIFO will not be overwritten. The two characters in the FIFO buffer can be read, however, no additional characters will be received until the error is cleared. The OERR bit can only be cleared by clearing the overrun condition. If the overrun error occurred when the SREN bit is set and CREN is clear then the error is cleared by reading RCREG. If the overrun occurred when the CREN bit is set then the error condition is cleared by either clearing the CREN bit of the RCSTA register.

16.3.1.7 Receiving 9-bit Characters

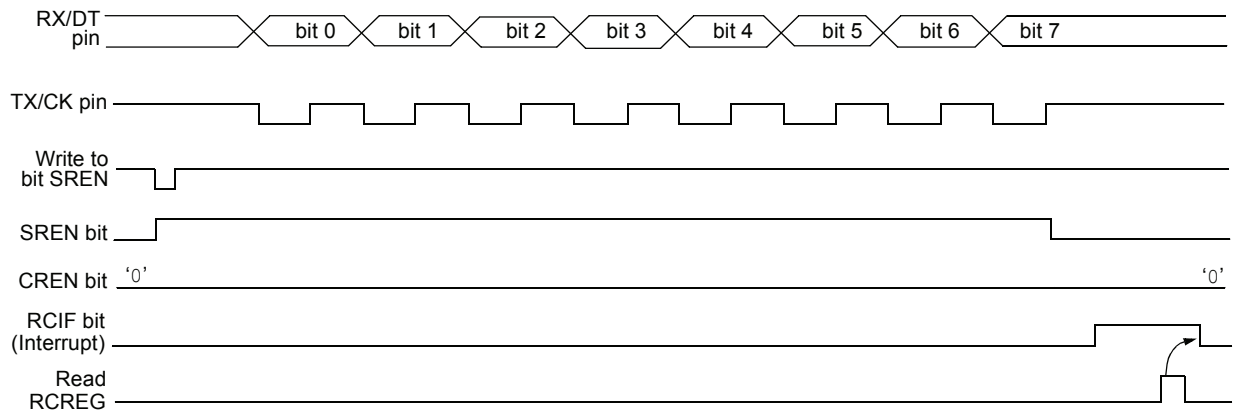
The AUSART supports 9-bit character reception. When the RX9 bit of the RCSTA register is set, the AUSART will shift 9-bits into the RSR for each character received. The RX9D bit of the RCSTA register is the ninth, and Most Significant, data bit of the top unread character in the receive FIFO. When reading 9-bit data from the receive FIFO buffer, the RX9D data bit must be read before reading the 8 Least Significant bits from the RCREG.

Address detection in Synchronous modes is not supported, therefore the ADDEN bit of the RCSTA register must be cleared.

16.3.1.8 Synchronous Master Reception Set-up:

1. Initialize the SPBRG register for the appropriate baud rate. Set or clear the BRGH bit, as required, to achieve the desired baud rate.
2. Enable the synchronous master serial port by setting bits SYNC, SPEN and CSRC.
3. Ensure bits CREN and SREN are clear.
4. If interrupts are desired, set the RCIE bit of the PIE1 register and the GIE and PEIE bits of the INTCON register.
5. If 9-bit reception is desired, set bit RX9.
6. Verify address detection is disabled by clearing the ADDEN bit of the RCSTA register.
7. Start reception by setting the SREN bit or for continuous reception, set the CREN bit.
8. Interrupt flag bit RCIF of the PIR1 register will be set when reception of a character is complete. An interrupt will be generated if the RCIE interrupt enable bit of the PIE1 register was set.
9. Read the RCSTA register to get the ninth bit (if enabled) and determine if any error occurred during reception.
10. Read the 8-bit received data by reading the RCREG register.
11. If an overrun error occurs, clear the error by either clearing the CREN bit of the RCSTA register or by clearing the SPEN bit, which resets the AUSART.

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.

16.3.2 SYNCHRONOUS SLAVE MODE

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

- SYNC = 1
- CSRC = 0
- 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. Clearing the CSRC bit of the TXSTA register configures the device as a slave. 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.

16.3.2.1 AUSART Synchronous Slave Transmit

The operation of the Synchronous Master and Slave modes are identical (refer to **Section 16.3.1.2 "Synchronous Master Transmission"**), except in the case of the Sleep mode.

If two words are written to the TXREG and then the SLEEP instruction is executed, the following will occur:

1. The first character will immediately transfer to the TSR register and transmit.
2. The second word will remain in TXREG register.
3. The TXIF bit will not be set.
4. After the first character has been shifted out of TSR, the TXREG register will transfer the second character to the TSR and the TXIF bit will now be set.
5. If the PEIE and TXIE bits are set, the interrupt will wake the device from Sleep and execute the next instruction. If the GIE bit is also set, the program will call the Interrupt Service Routine.

16.3.2.2 Synchronous Slave Transmission Set-up:

1. Set the SYNC and SPEN bits and clear the CSRC bit.
2. Clear the CREN and SREN bits.
3. If using interrupts, ensure that the GIE and PEIE bits of the INTCON register are set and set the TXIE bit.
4. If 9-bit transmission is desired, set the TX9 bit.
5. Enable transmission by setting the TXEN bit.
6. Verify address detection is disabled by clearing the ADDEN bit of the RCSTA register.
7. If 9-bit transmission is selected, insert the Most Significant bit into the TX9D bit.
8. Start transmission by writing the Least Significant 8 bits to the TXREG register.

TABLE 16-8: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE TRANSMISSION

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
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
TXREG	AUSART Transmit Data Register								0000 0000	0000 0000
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 Transmission.

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 Set-up:

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 8 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.

16.4 AUSART Operation During Sleep

The AUSART will remain active during Sleep only in the Synchronous Slave mode. All other modes require the system clock and therefore can not generate the necessary signals to run the Transmit or Receive Shift registers during Sleep.

Synchronous Slave mode uses an externally generated clock to run the Transmit and Receive Shift registers.

16.4.1 SYNCHRONOUS RECEIVE DURING SLEEP

To receive during Sleep, all the following conditions must be met before entering Sleep mode:

- RCSTA and TXSTA Control registers must be configured for Synchronous Slave Reception (refer to **Section 16.3.2.4 "Synchronous Slave Reception Set-up:"**).
- If interrupts are desired, set the RCIE bit of the PIE1 register and the PEIE bit of the INTCON register.
- The RCIF interrupt flag must be cleared by reading RCREG to unload any pending characters in the receive buffer.

Upon entering Sleep mode, the device will be ready to accept data and clocks on the RX/DT and TX/CK pins, respectively. When the data word has been completely clocked in by the external device, the RCIF interrupt flag bit of the PIR1 register will be set, thereby waking the processor from Sleep.

Upon waking from Sleep, the instruction following the `SLEEP` instruction will be executed. If the Global Interrupt Enable (GIE) bit of the INTCON register is also set, then the Interrupt Service Routine at address 0004h will be called.

16.4.2 SYNCHRONOUS TRANSMIT DURING SLEEP

To transmit during Sleep, all the following conditions must be met before entering Sleep mode:

- RCSTA and TXSTA Control registers must be configured for Synchronous Slave Transmission (refer to **Section 16.3.2.2 "Synchronous Slave Transmission Set-up:"**).
- The TXIF interrupt flag must be cleared by writing the output data to the TXREG, thereby filling the TSR and transmit buffer.
- If interrupts are desired, set the TXIE bit of the PIE1 register and the PEIE bit of the INTCON register.

Upon entering Sleep mode, the device will be ready to accept clocks on TX/CK pin and transmit data on the RX/DT pin. When the data word in the TSR has been completely clocked out by the external device, the pending byte in the TXREG will transfer to the TSR and the TXIF flag will be set, thereby waking the processor from Sleep. At this point, the TXREG is available to accept another character for transmission, which will clear the TXIF flag.

Upon waking from Sleep, the instruction following the `SLEEP` instruction will be executed. If the Global Interrupt Enable (GIE) bit is also set then the Interrupt Service Routine at address 0004h will be called.

NOTES:

17.0 SSP MODULE OVERVIEW

The Synchronous Serial Port (SSP) module is a serial interface useful for communicating with other peripherals or microcontroller devices. These peripheral devices may be serial EEPROMs, shift registers, display drivers, A/D converters, etc. The SSP module can operate in one of two modes:

- Serial Peripheral Interface (SPI)
- Inter-Integrated Circuit (I^2C™)

17.1 SPI Mode

The SPI mode allows 8 bits of data to be synchronously transmitted and received, simultaneously. The SSP module can be operated in one of two SPI modes:

- Master mode
- Slave mode

SPI is a full-duplex protocol, with all communication being bidirectional and initiated by a master device. All clocking is provided by the master device and all bits are transmitted, MSb first. Care must be taken to ensure that all devices on the SPI bus are setup to allow all controllers to send and receive data at the same time.

A typical SPI connection between microcontroller devices is shown in Figure 17-1. Addressing of more than one slave device is accomplished via multiple hardware slave select lines. External hardware and additional I/O pins must be used to support multiple slave select addressing. This prevents extra overhead in software for communication.

For SPI communication, typically three pins are used:

- Serial Data Out (SDO)
- Serial Data In (SDI)
- Serial Clock (SCK)

Additionally, a fourth pin may be used when in a Slave mode of operation:

- Slave Select ($\overline{SS}$)

FIGURE 17-1: TYPICAL SPI MASTER/SLAVE CONNECTION

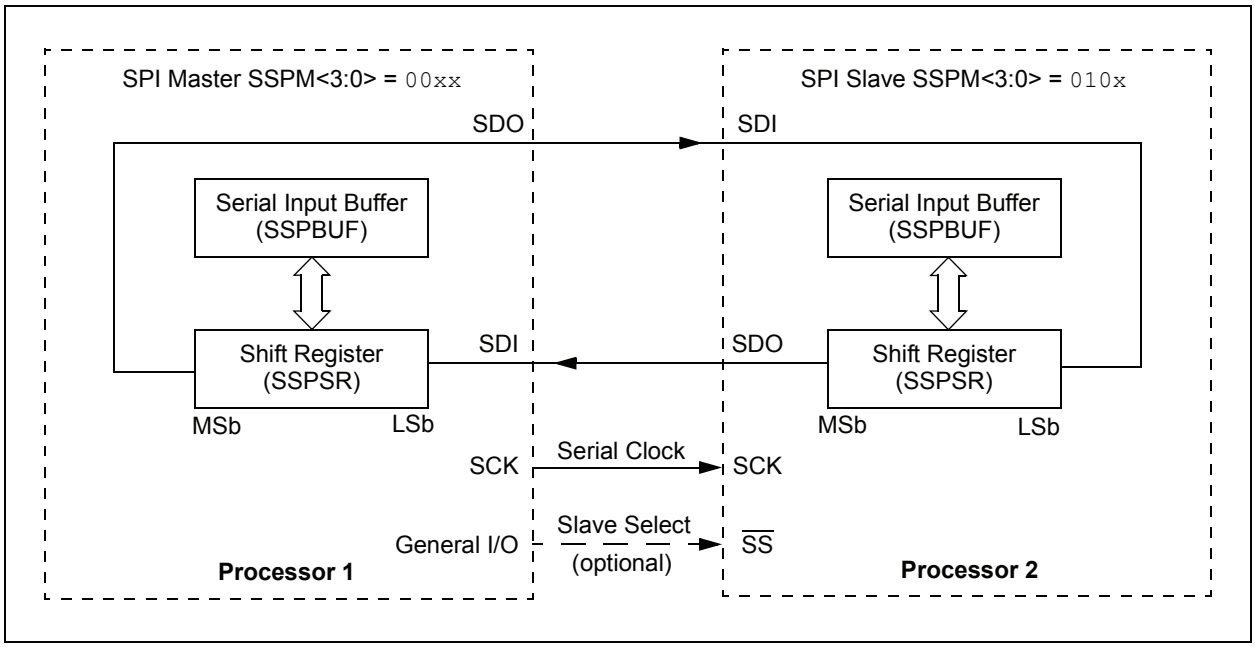
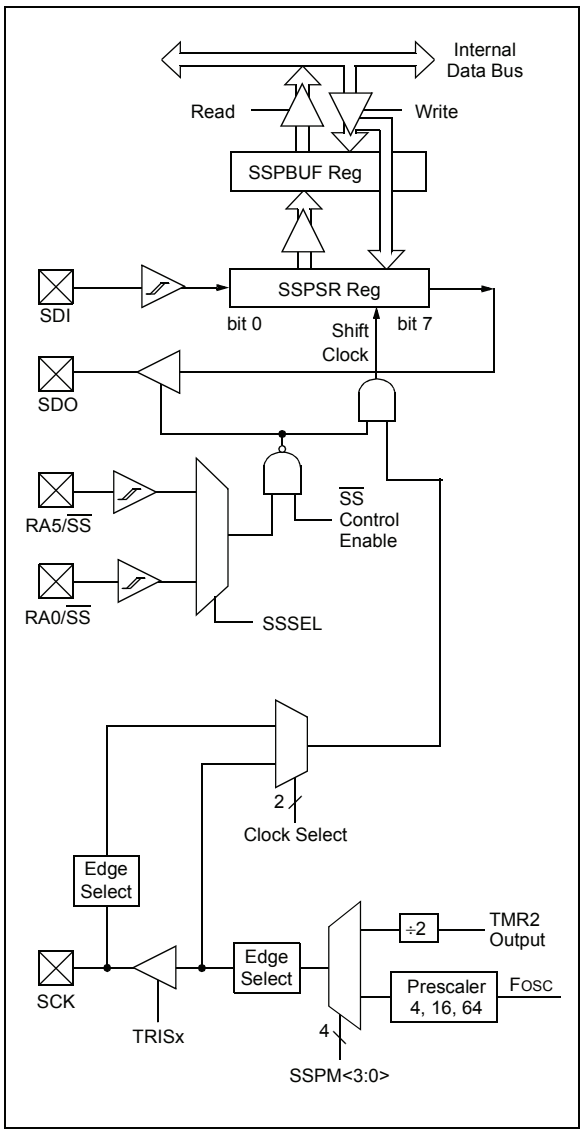


FIGURE 17-2: SPI MODE BLOCK DIAGRAM



17.1.1 MASTER MODE

In Master mode, data transfer can be initiated at any time because the master controls the SCK line. Master mode determines when the slave (Figure 17-1, Processor 2) transmits data via control of the SCK line.

17.1.1.1 Master Mode Operation

The SSP consists of a transmit/receive shift register (SSPSR) and a buffer register (SSPBUF). The SSPSR register shifts the data in and out of the device, MSb first. The SSPBUF register holds the data that is written out of the master until the received data is ready. Once the eight bits of data have been received, the byte is moved to the SSPBUF register. The Buffer Full Status bit, BF of the SSPSTAT register, and the SSP Interrupt Flag bit, SSPIF of the PIR1 register, are then 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.

When the application software is expecting to receive valid data, the SSPBUF should be read before the next byte of data is written to the SSPBUF. The BF bit of the SSPSTAT register is set when SSPBUF has been loaded with the received data (transmission is complete). When the SSPBUF is read, the BF bit is cleared. This data may be irrelevant if the SPI is only a transmitter. The SSP interrupt may be used to determine when the transmission/reception is complete and the SSPBUF must be read and/or written. If interrupts are not used, then software polling can be done to ensure that a write collision does not occur. Example 17-1 shows the loading of the SSPBUF (SSPSR) for data transmission.

Note:	The SSPSR is not directly readable or writable and can only be accessed by addressing the SSPBUF register.

17.1.1.2 Enabling Master I/O

To enable the serial port, the SSPEN bit of the SSPCON register, must be set. To reset or reconfigure SPI mode, clear the SSPEN bit, re-initialize the SSPCON register and then set the SSPEN bit. If a Master 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 output

17.1.1.3 Master Mode Setup

In Master mode, the data is transmitted/received as soon as the SSPBUF register is loaded with a byte value. If the master is only going to receive, SDO output could be disabled (programmed and used as an input). The SSPSR register will continue to shift in the signal present on the SDI pin at the programmed clock rate.

When initializing SPI Master mode operation, several options need to be specified. This is accomplished by programming the appropriate control bits in the SSPCON and SSPSTAT registers. These control bits allow the following to be specified:

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

In Master mode, the SPI clock rate (bit rate) is user selectable to be one of the following:

- FOSC/4 (or TCY)
- FOSC/16 (or 4 • TCY)
- FOSC/64 (or 16 • TCY)
- (Timer2 output)/2

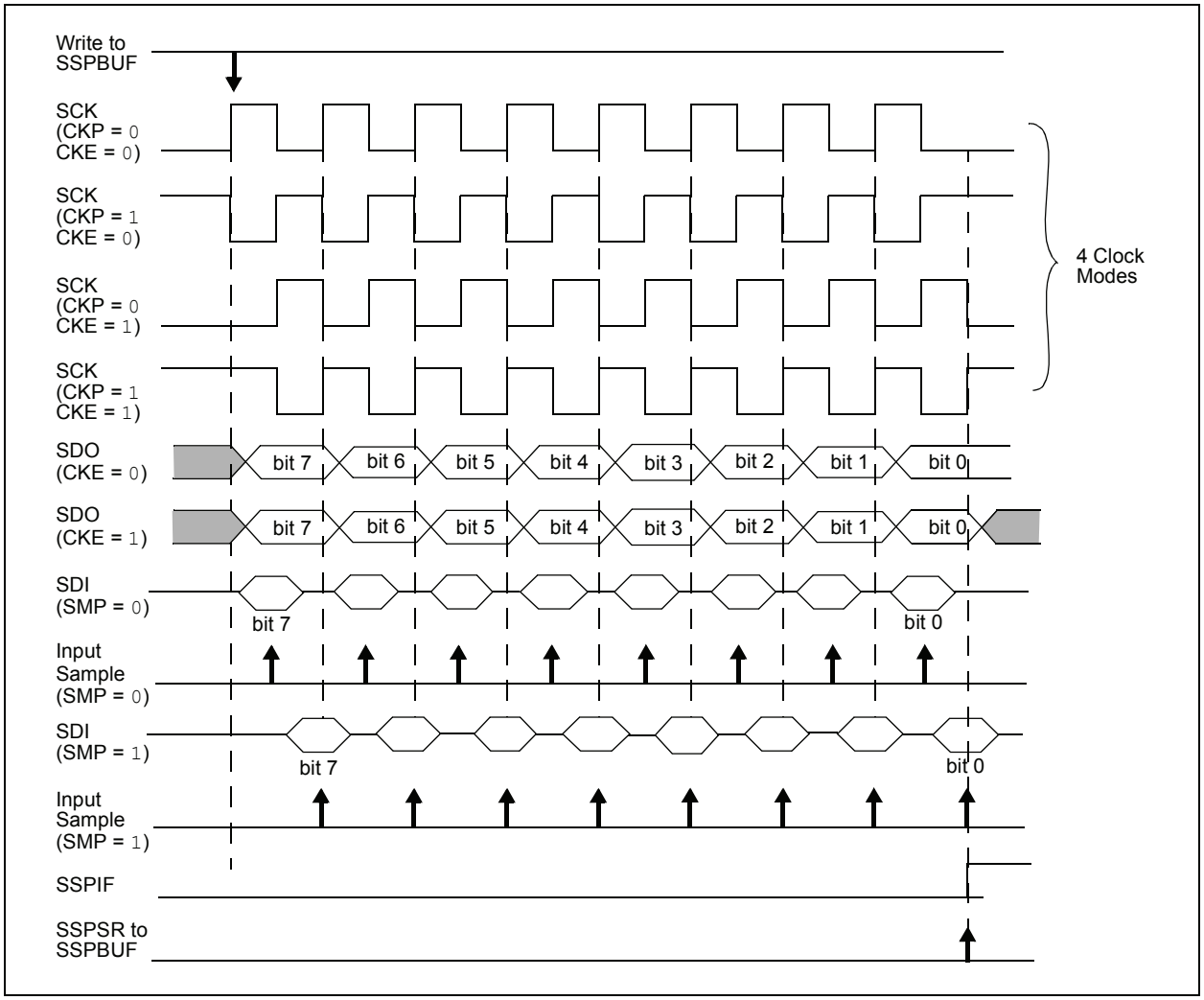
This allows a maximum data rate of 5 Mbps (at FOSC = 20 MHz).

Figure 17-3 shows the waveforms for Master mode. The clock polarity is selected by appropriately programming the CKP bit of the SSPCON register. When the CKE bit is set, the SDO data is valid before there is a clock edge on SCK. The sample time of the input data is shown based on the state of the SMP bit and can occur at the middle or end of the data output time. The time when the SSPBUF is loaded with the received data is shown.

17.1.1.4 Sleep in Master Mode

In Master mode, all module clocks are halted and the transmission/reception will remain in their current state, paused, until the device wakes from Sleep. After the device wakes up from Sleep, the module will continue to transmit/receive data.

FIGURE 17-3: SPI MASTER MODE WAVEFORM



EXAMPLE 17-1: LOADING THE SSPBUF (SSPSR) REGISTER

```
        BANKSEL   SSPSTAT      ;
LOOP    BTFSS     SSPSTAT, BF  ;Has data been received(transmit complete)?
        GOTO      LOOP         ;No
        BANKSEL   SSPBUF       ;
        MOVF      SSPBUF, W    ;WREG reg = contents of SSPBUF
        MOVWF     RXDATA       ;Save in user RAM, if data is meaningful
        MOVF      TXDATA, W    ;W reg = contents of TXDATA
        MOVWF     SSPBUF       ;New data to xmit
```

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 one of the following pins via the SSSEL bit in the APFCON register.

- RA5/AN4/$\overline{SS}$
- RA0/AN0/$\overline{SS}$

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.

FIGURE 17-4: SPI MODE WAVEFORM (SLAVE MODE WITH CKE = 0)

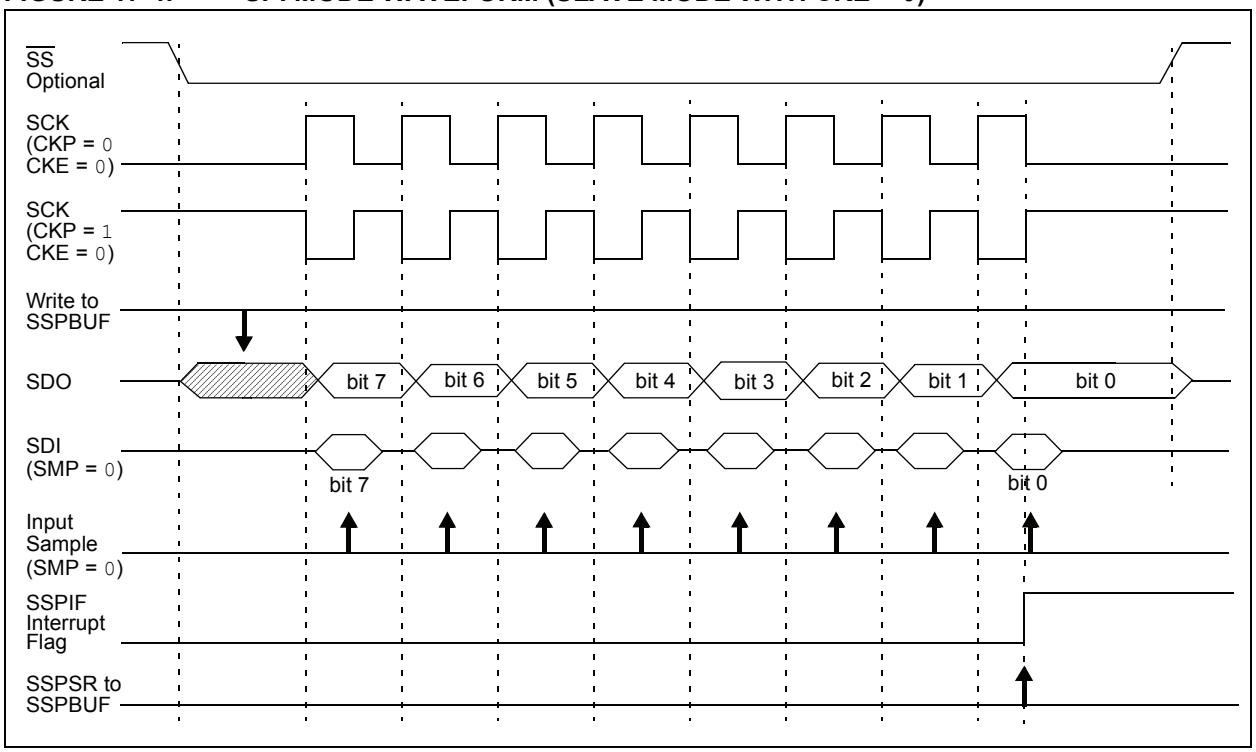
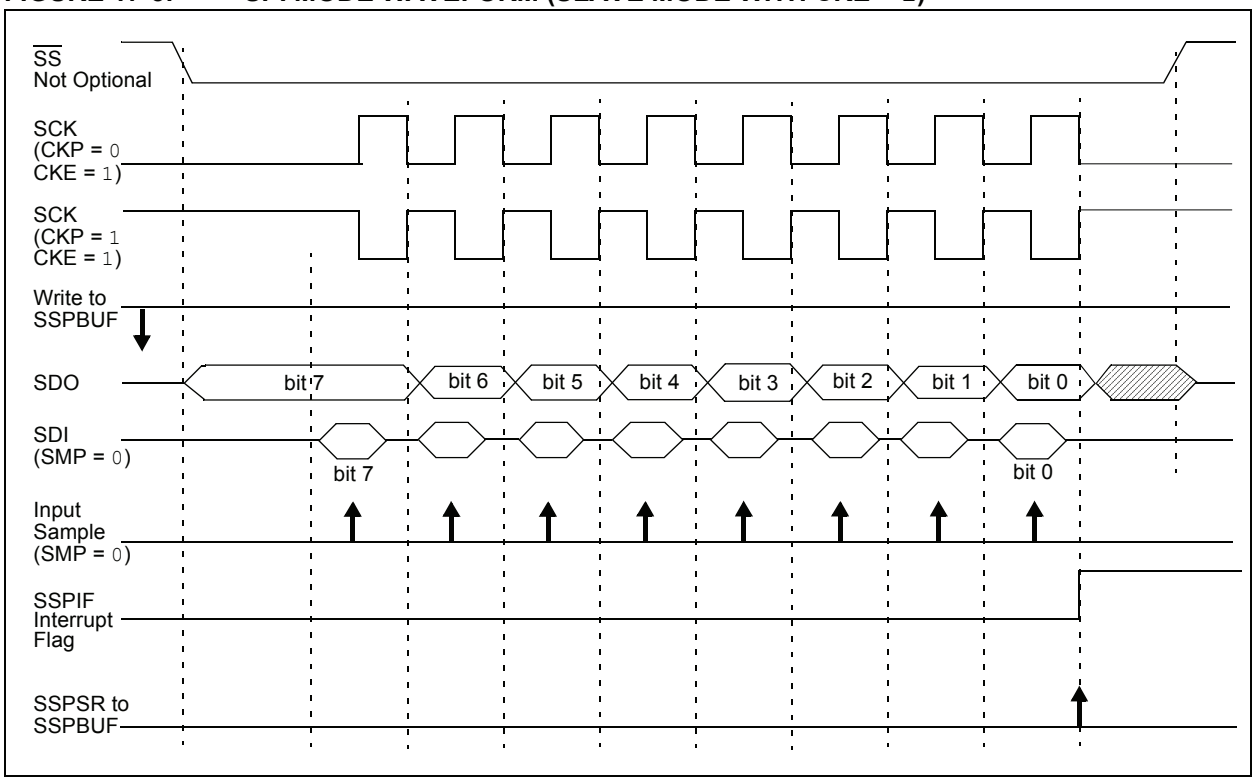


FIGURE 17-5: SPI MODE WAVEFORM (SLAVE MODE WITH CKE = 1)



17.1.2.4 Slave Select Operation

The $\overline{SS}$ pin allows Synchronous Slave mode operation. The SPI must be in Slave mode with $\overline{SS}$ pin control enabled (SSPM<3:0> = 0100). The associated TRIS bit for the $\overline{SS}$ pin must be set, making $\overline{SS}$ an input.

In Slave Select mode, when:

- $\overline{SS}$ = 0, The device operates as specified in **Section 17.1.2 "Slave Mode"**.
- $\overline{SS}$ = 1, The SPI module is held in Reset and the SDO pin will be tri-stated.

Note 1: When the SPI is in Slave mode with $\overline{SS}$ pin control enabled (SSPM<3:0> = 0100), the SPI module will reset if the $\overline{SS}$ pin is driven high.

2: If the SPI is used in Slave mode with CKE set, the $\overline{SS}$ pin control must be enabled.

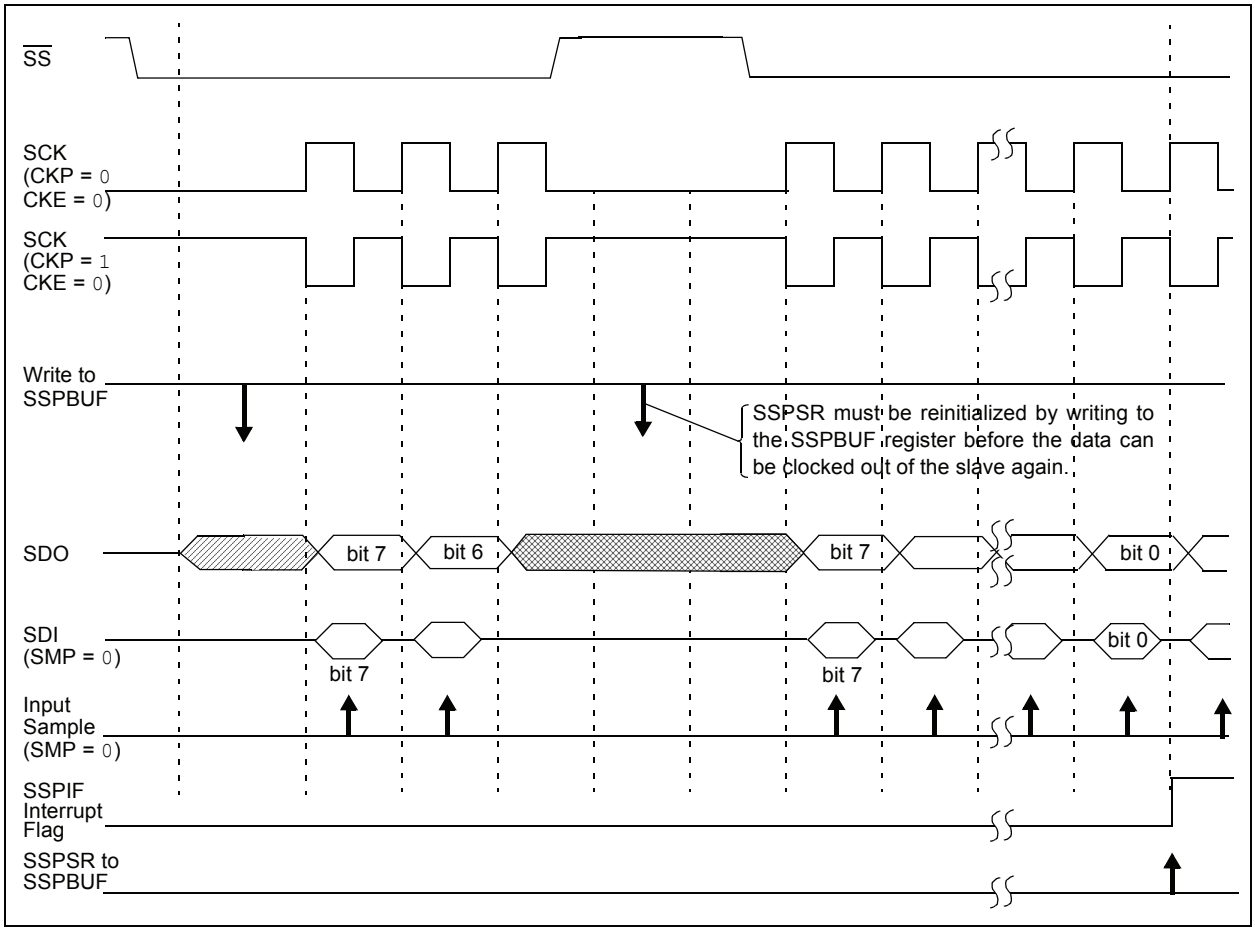
When the SPI module resets, the bit counter is cleared to '0'. This can be done by either forcing the $\overline{SS}$ pin to a high level or clearing the SSPEN bit. Figure 17-6 shows the timing waveform for such a synchronization event.

Note: SSPSR must be reinitialized by writing to the SSPBUF register before the data can be clocked out of the slave again.

17.1.2.5 Sleep in Slave Mode

While in Sleep mode, the slave can transmit/receive data. The SPI Transmit/Receive Shift register operates asynchronously to the device on the externally supplied clock source. This allows the device to be placed in Sleep mode and data to be shifted into the SPI Transmit/Receive Shift register. When all 8 bits have been received, the SSP Interrupt Flag bit will be set and if enabled, will wake the device from Sleep.

FIGURE 17-6: SLAVE SELECT SYNCHRONIZATION WAVEFORM



REGISTER 17-1: SSPCON: SYNC SERIAL PORT CONTROL REGISTER (SPI MODE)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0
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 **WCOL**: Write Collision Detect bit
- 1 = The SSPBUF register is written while it is still transmitting the previous word (must be cleared in software)
- 0 = No collision

bit 6 **SSPOV**: Receive Overflow Indicator bit
- 1 = A new byte is received while the SSPBUF register is still holding the previous data. In case of overflow, the data in SSPSR is lost. Overflow can only occur in Slave mode. The user must read the SSPBUF, even if only transmitting data, to avoid setting overflow. In Master mode, the overflow bit is not set since each new reception (and transmission) is initiated by writing to the SSPBUF register.
- 0 = No overflow

bit 5 **SSPEN**: Synchronous Serial Port Enable bit
- 1 = Enables serial port and configures SCK, SDO and SDI as serial port pins(1)
- 0 = Disables serial port and configures these pins as I/O port pins

bit 4 **CKP**: Clock Polarity Select bit
- 1 = Idle state for clock is a high level
- 0 = Idle state for clock is a low level

bit 3-0 **SSPM<3:0>**: Synchronous Serial Port Mode Select bits
- 0000 = SPI Master mode, clock = FOSC/4
- 0001 = SPI Master mode, clock = FOSC/16
- 0010 = SPI Master mode, clock = FOSC/64
- 0011 = SPI Master mode, clock = TMR2 output/2
- 0100 = SPI Slave mode, clock = SCK pin. $\overline{SS}$ pin control enabled.
- 0101 = SPI Slave mode, clock = SCK pin. $\overline{SS}$ pin control disabled. $\overline{SS}$ can be used as I/O pin.

Note 1: When enabled, these pins must be properly configured as input or output.

REGISTER 17-2: SSPSTAT: SYNC SERIAL PORT STATUS REGISTER (SPI MODE)

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF
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 **SMP**: SPI Data Input Sample Phase bit
SPI Master mode:
1 = Input data sampled at end of data output time
0 = Input data sampled at middle of data output time
SPI Slave mode:
SMP must be cleared when SPI is used in Slave mode

bit 6 **CKE**: SPI Clock Edge Select bit
SPI mode, CKP = 0:
1 = Data stable on rising edge of SCK
0 = Data stable on falling edge of SCK
SPI mode, CKP = 1:
1 = Data stable on falling edge of SCK
0 = Data stable on rising edge of SCK

bit 5 **D/$\overline{A}$:** Data/$\overline{Address}$ bit
Used in I^2C mode only.

bit 4 **P:** Stop bit
Used in I^2C mode only.

bit 3 **S:** Start bit
Used in I^2C mode only.

bit 2 **R/$\overline{W}$:** Read/$\overline{Write}$ Information bit
Used in I^2C mode only.

bit 1 **UA:** Update Address bit
Used in I^2C mode only.

bit 0 **BF**: Buffer Full Status bit
1 = Receive complete, SSPBUF is full
0 = Receive not complete, SSPBUF is empty

TABLE 17-1: SUMMARY OF REGISTERS ASSOCIATED WITH SPI OPERATION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELA	—	—	ANSA5	ANSA4	ANSA3	ANSA2	ANSA1	ANSA0	49
APFCON	—	—	—	—	—	—	SSSEL	CCP2SEL	47
INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	40
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	41
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	43
PR2	Timer2 Period Register								115
SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								157
SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	162
SSPSTAT	SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF	163
TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	48
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	67
T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	116

Legend: x = unknown, u = unchanged, – = unimplemented, read as '0'. Shaded cells are not used by the SSP in SPI mode.

17.2 I²C Mode

The SSP module, in I²C mode, implements all slave functions, except general call support. It provides interrupts on Start and Stop bits in hardware to facilitate firmware implementations of the master functions. The SSP module implements the I²C Standard mode specifications:

- I²C Slave mode (7-bit address)
- I²C Slave mode (10-bit address)
- Start and Stop bit interrupts enabled to support firmware Master mode
- Address masking

Two pins are used for data transfer; the SCL pin (clock line) and the SDA pin (data line). The user must configure the two pin's data direction bits as inputs in the appropriate TRIS register. Upon enabling I²C mode, the I²C slew rate limiters in the I/O pads are controlled by the SMP bit of SSPSTAT register. The SSP module functions are enabled by setting the SSPEN bit of SSPCON register.

Data is sampled on the rising edge and shifted out on the falling edge of the clock. This ensures that the SDA signal is valid during the SCL high time. The SCL clock input must have minimum high and low times for proper operation. Refer to **Section 23.0 "Electrical Specifications"**.

FIGURE 17-7: I²C™ MODE BLOCK DIAGRAM

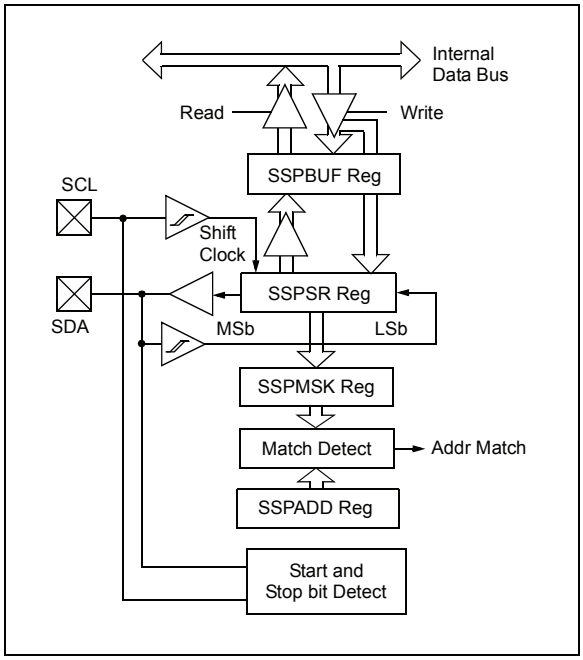
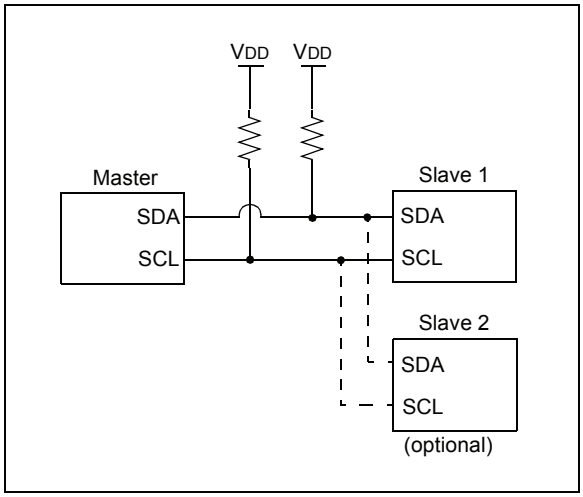


FIGURE 17-8: TYPICAL I²C™ CONNECTIONS



The SSP module has six registers for I²C operation. They are:

- SSP Control (SSPCON) register
- SSP Status (SSPSTAT) register
- Serial Receive/Transmit Buffer (SSPBUF) register
- SSP Shift Register (SSPSR), not directly accessible
- SSP Address (SSPADD) register
- SSP Address Mask (SSPMSK) register

17.2.1 HARDWARE SETUP

Selection of I²C mode, with the SSPEN bit of the SSPCON register set, forces the SCL and SDA pins to be open drain, provided these pins are programmed as inputs by setting the appropriate TRISC bits. The SSP module will override the input state with the output data, when required, such as for Acknowledge and slave-transmitter sequences.

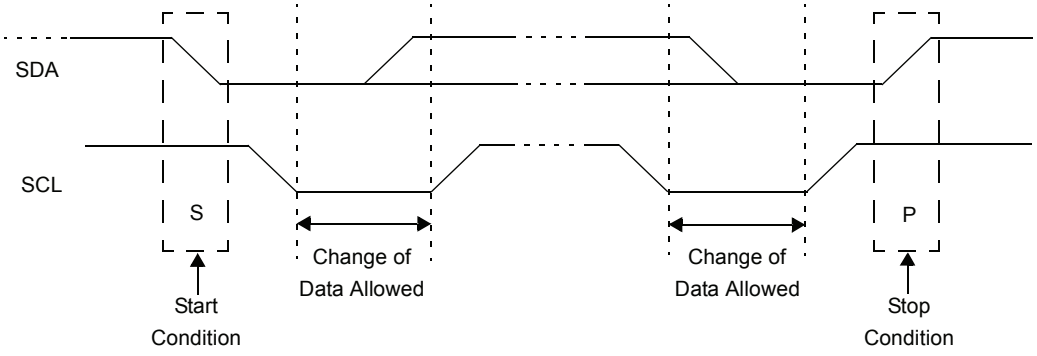
Note:	Pull-up resistors must be provided externally to the SCL and SDA pins for proper operation of the I²C module.

17.2.2 START AND STOP CONDITIONS

During times of no data transfer (Idle time), both the clock line (SCL) and the data line (SDA) are pulled high through external pull-up resistors. The Start and Stop conditions determine the start and stop of data transmission. The Start condition is defined as a high-to-low transition of the SDA line while SCL is high. The Stop condition is defined as a low-to-high transition of the SDA line while SCL is high.

Figure 17-9 shows the Start and Stop conditions. A master device generates these conditions for starting and terminating data transfer. Due to the definition of the Start and Stop conditions, when data is being transmitted, the SDA line can only change state when the SCL line is low.

FIGURE 17-9: START AND STOP CONDITIONS



17.2.3 ACKNOWLEDGE

After the valid reception of an address or data byte, the hardware automatically will generate the Acknowledge ($\overline{ACK}$) pulse and load the SSPBUF register with the received value currently in the SSPSR register. There are certain conditions that will cause the SSP module not to generate this $\overline{ACK}$ pulse. They include any or all of the following:

- The Buffer Full bit, BF of the SSPSTAT register, was set before the transfer was received.
- The SSP Overflow bit, SSPOV of the SSPCON register, was set before the transfer was received.
- The SSP module is being operated in Firmware Master mode.

In such a case, the SSPSR register value is not loaded into the SSPBUF, but bit SSPIF of the PIR1 register is set. Table 17-2 shows the results of when a data transfer byte is received, given the status of bits BF and SSPOV. Flag bit BF is cleared by reading the SSPBUF register, while bit SSPOV is cleared through software.

TABLE 17-2: DATA TRANSFER RECEIVED BYTE ACTIONS

Status Bits as Data Transfer is Received		SSPSR → SSPBUF	Generate $\overline{ACK}$ Pulse	Set bit SSPIF (SSP Interrupt occurs if enabled)
BF	SSPOV			
0	0	Yes	Yes	Yes
1	0	No	No	Yes
1	1	No	No	Yes
0	1	No	No	Yes

Note 1: Shaded cells show the conditions where the user software did not properly clear the overflow condition.

17.2.4 ADDRESSING

Once the SSP module has been enabled, it waits for a Start condition to occur. Following the Start condition, the 8 bits are shifted into the SSPSR register. All incoming bits are sampled with the rising edge of the clock line (SCL).

17.2.4.1 7-bit Addressing

In 7-bit Addressing mode (Figure 17-10), the value of register SSPSR<7:1> is compared to the value of register SSPADD<7:1>. The address is compared on the falling edge of the eighth clock (SCL) pulse. If the addresses match, and the BF and SSPOV bits are clear, the following events occur:

- The SSPSR register value is loaded into the SSPBUF register.
- The BF bit is set.
- An $\overline{\text{ACK}}$ pulse is generated.
- SSP interrupt flag bit, SSPIF of the PIR1 register, is set (interrupt is generated if enabled) on the falling edge of the ninth SCL pulse.

17.2.4.2 10-bit Addressing

In 10-bit Address mode, two address bytes need to be received by the slave (Figure 17-11). The five Most Significant bits (MSbs) of the first address byte specify if it is a 10-bit address. The R/$\overline{\text{W}}$ bit of the SSPSTAT register must specify a write so the slave device will receive the second address byte. For a 10-bit address, the first byte would equal '1111 0 A9 A8 0', where A9 and A8 are the two MSbs of the address.

The sequence of events for 10-bit address is as follows for reception:

1. Load SSPADD register with high byte of address.
2. Receive first (high) byte of address (bits SSPIF, BF and UA of the SSPSTAT register are set).
3. Read the SSPBUF register (clears bit BF).
4. Clear the SSPIF flag bit.
5. Update the SSPADD register with second (low) byte of address (clears UA bit and releases the SCL line).
6. Receive low byte of address (bits SSPIF, BF and UA are set).
7. Update the SSPADD register with the high byte of address. If match releases SCL line, this will clear bit UA.
8. Read the SSPBUF register (clears bit BF).
9. Clear flag bit SSPIF.

If data is requested by the master, once the slave has been addressed:

1. Receive repeated Start condition.
2. Receive repeat of high byte address with R/$\overline{\text{W}}$ = 1, indicating a read.
3. BF bit is set and the CKP bit is cleared, stopping SCL and indicating a read request.
4. SSPBUF is written, setting BF, with the data to send to the master device.
5. CKP is set in software, releasing the SCL line.

17.2.4.3 Address Masking

The Address Masking register (SSPMSK) is only accessible while the SSPM bits of the SSPCON register are set to '1001'. In this register, the user can select which bits of a received address the hardware will compare when determining an address match. Any bit that is set to a zero in the SSPMSK register, the corresponding bit in the received address byte and SSPADD register are ignored when determining an address match. By default, the register is set to all ones, requiring a complete match of a 7-bit address or the lower eight bits of a 10-bit address.

17.2.5 RECEPTION

When the R/$\overline{W}$ bit of the received address byte is clear, the master will write data to the slave. If an address match occurs, the received address is loaded into the SSPBUF register. An address byte overflow will occur if that loaded address is not read from the SSPBUF before the next complete byte is received.

An SSP interrupt is generated for each data transfer byte. The BF, R/$\overline{W}$ and D/$\overline{A}$ bits of the SSPSTAT register are used to determine the status of the last received byte.

FIGURE 17-10: I²C™ WAVEFORMS FOR RECEPTION (7-BIT ADDRESS)

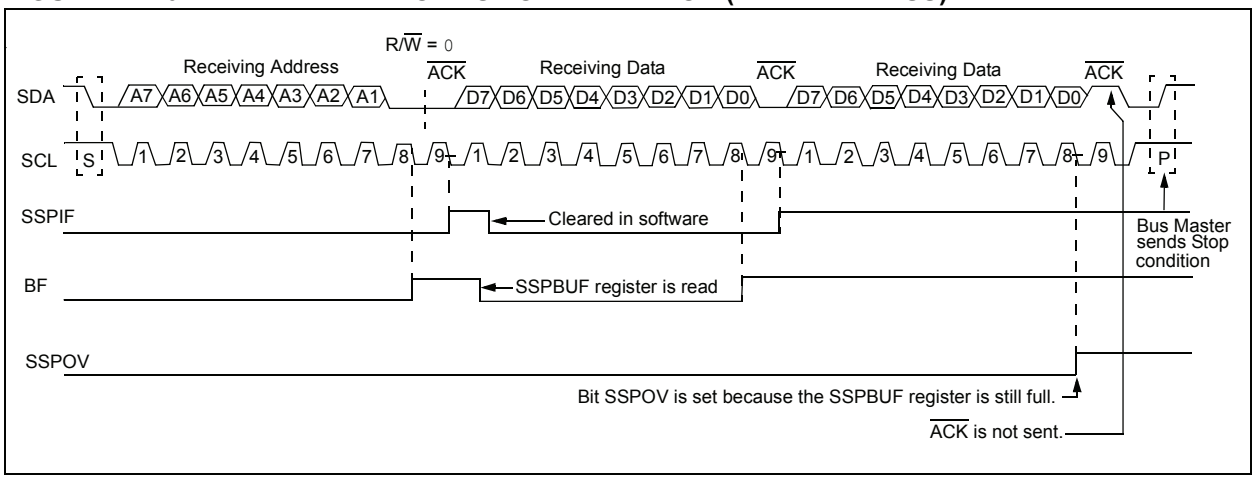
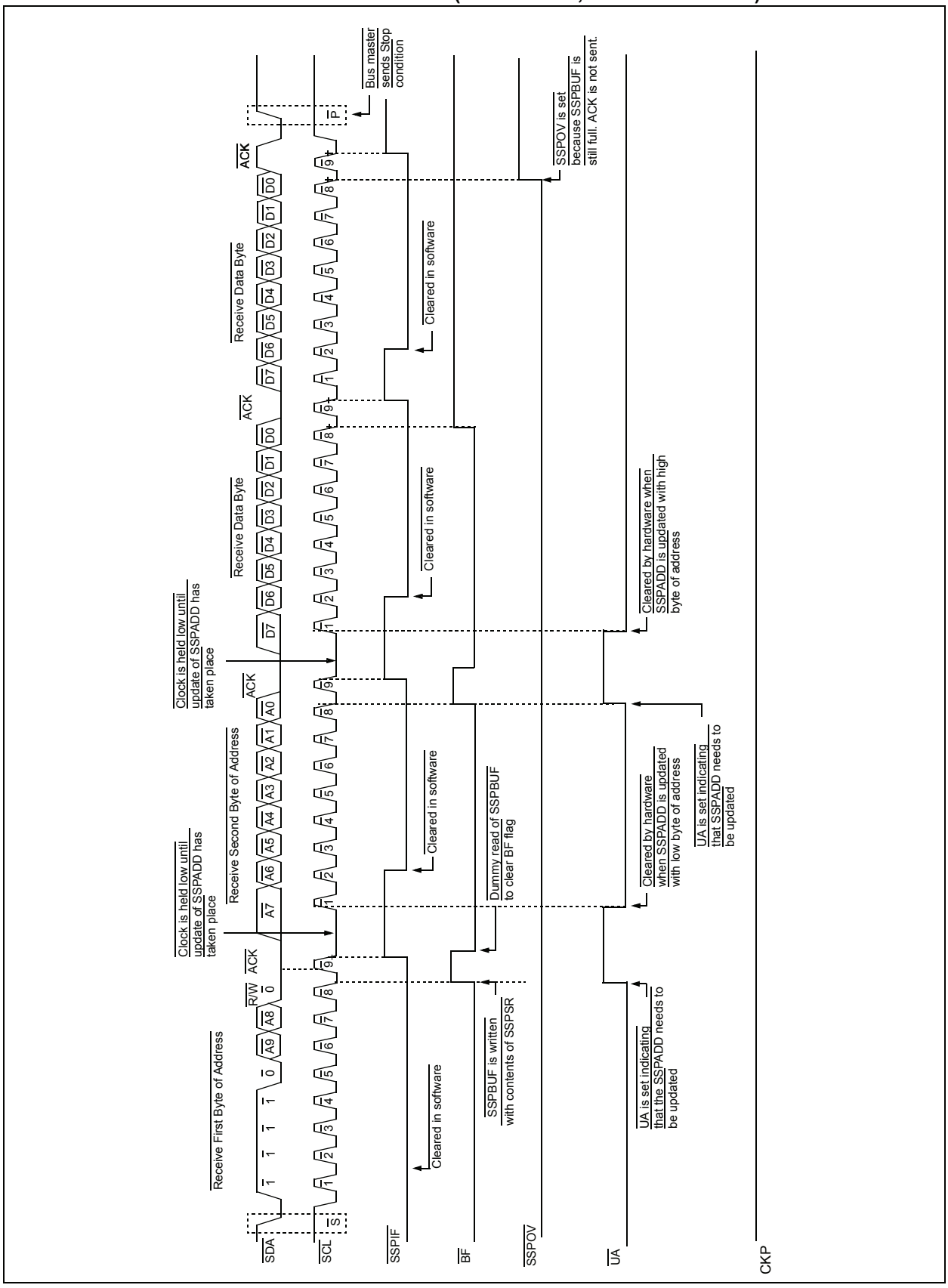


FIGURE 17-11: I²C™ SLAVE MODE TIMING (RECEPTION, 10-BIT ADDRESS)



17.2.6 TRANSMISSION

When the R/$\overline{W}$ bit of the received address byte is set and an address match occurs, the R/$\overline{W}$ bit of the SSPSTAT register is set and the slave will respond to the master by reading out data. After the address match, an $\overline{ACK}$ pulse is generated by the slave hardware and the SCL pin is held low (clock is automatically stretched) until the slave is ready to respond. See **Section 17.2.7 "Clock Stretching"**. The data the slave will transmit must be loaded into the SSPBUF register, which sets the BF bit. The SCL line is released by setting the CKP bit of the SSPCON register.

An SSP interrupt is generated for each transferred data byte. The SSPIF flag bit of the PIR1 register initiates an SSP interrupt, and must be cleared by software before the next byte is transmitted. The BF bit of the SSPSTAT register is cleared on the falling edge of the 8th received clock pulse. The SSPIF flag bit is set on the falling edge of the ninth clock pulse.

Following the 8th falling clock edge, control of the SDA line is released back to the master so that the master can acknowledge or not acknowledge the response. If the master sends a not acknowledge, the slave's transmission is complete and the slave must monitor for the next Start condition. If the master acknowledges, control of the bus is returned to the slave to transmit another byte of data. Just as with the previous byte, the clock is stretched by the slave, data must be loaded into the SSPBUF and CKP must be set to release the clock line (SCL).

FIGURE 17-12: I²C WAVEFORMS FOR TRANSMISSION (7-BIT ADDRESS)

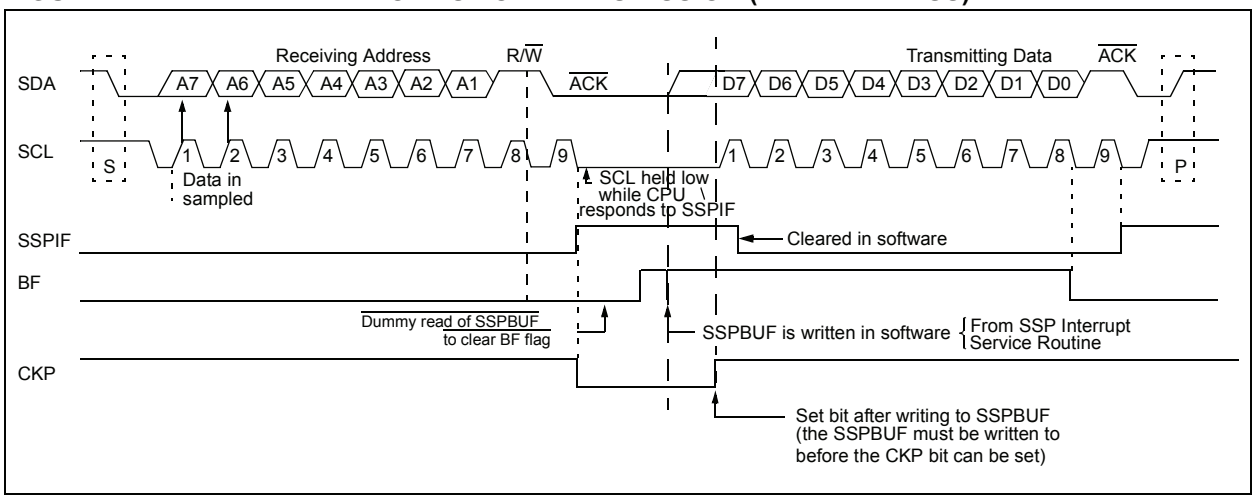
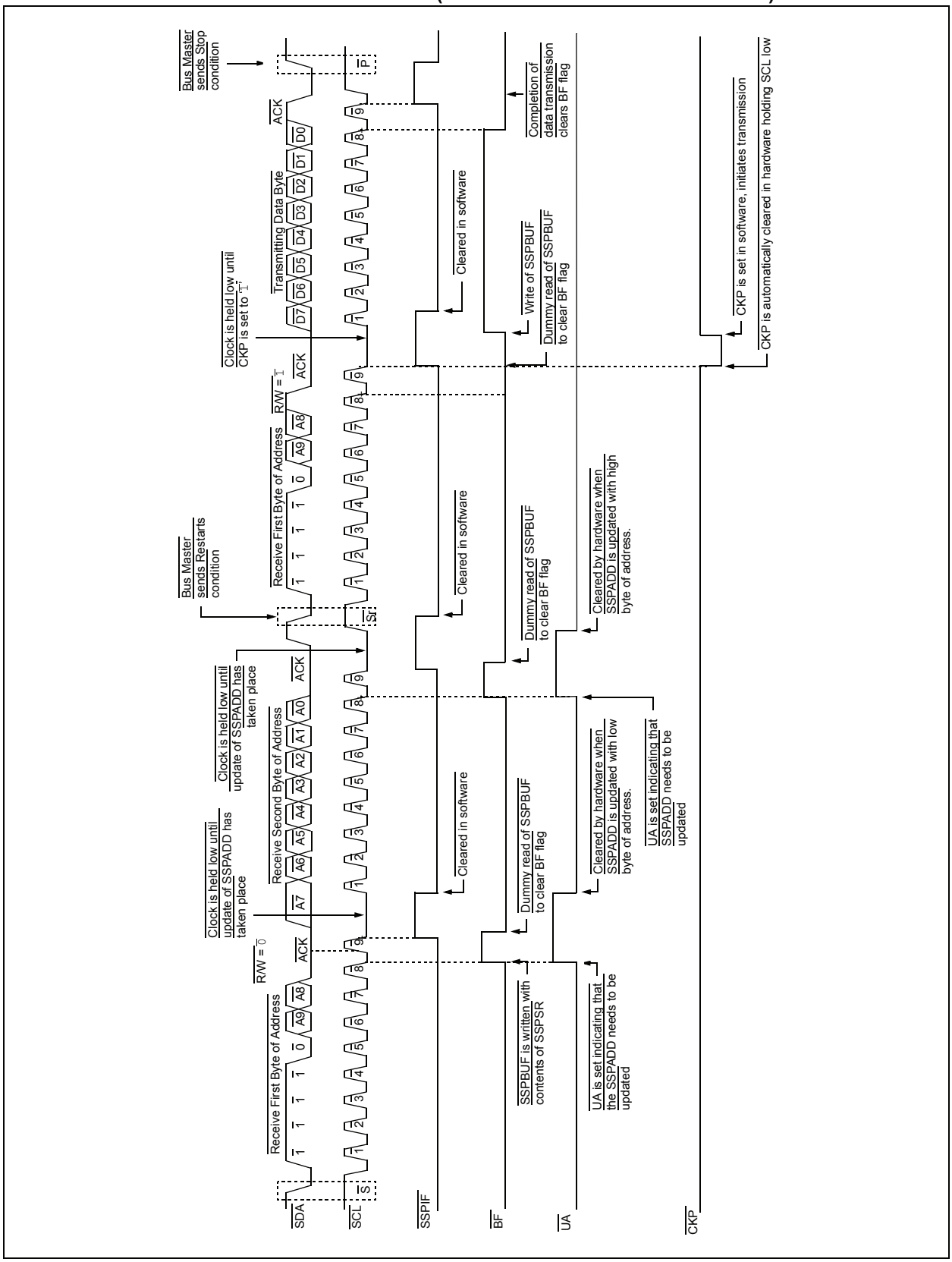


FIGURE 17-13: I²C SLAVE MODE TIMING (TRANSMISSION 10-BIT ADDRESS)



17.2.7 CLOCK STRETCHING

During any SCL low phase, any device on the I^2C bus may hold the SCL line low and delay, or pause, the transmission of data. This "stretching" of a transmission allows devices to slow down communication on the bus. The SCL line must be constantly sampled by the master to ensure that all devices on the bus have released SCL for more data.

Stretching usually occurs after an $\overline{ACK}$ bit of a transmission, delaying the first bit of the next byte. The SSP module hardware automatically stretches for two conditions:

- After a 10-bit address byte is received (update SSPADD register)
- Anytime the CKP bit of the SSPCON register is cleared by hardware

The module will hold SCL low until the CKP bit is set. This allows the user slave software to update SSPBUF with data that may not be readily available. In 10-bit addressing modes, the SSPADD register must be updated after receiving the first and second address bytes. The SSP module will hold the SCL line low until the SSPADD has a byte written to it. The UA bit of the SSPSTAT register will be set, along with SSPIF, indicating an address update is needed.

17.2.8 FIRMWARE MASTER MODE

Master mode of operation is supported in firmware using interrupt generation on the detection of the Start and Stop conditions. The Stop (P) and Start (S) bits of the SSPSTAT register are cleared from a Reset or when the SSP module is disabled (SSPEN cleared). The Stop (P) and Start (S) bits will toggle based on the Start and Stop conditions. Control of the I^2C bus may be taken when the P bit is set or the bus is Idle and both the S and P bits are clear.

In Firmware Master mode, the SCL and SDA lines are manipulated by setting/clearing the corresponding TRIS bit(s). The output level is always low, irrespective of the value(s) in the corresponding PORT register bit(s). When transmitting a '1', the TRIS bit must be set (input) and a '0', the TRIS bit must be clear (output).

The following events will cause the SSP Interrupt Flag bit, SSPIF, to be set (SSP Interrupt will occur if enabled):

- Start condition
- Stop condition
- Data transfer byte transmitted/received

Firmware Master mode of operation can be done with either the Slave mode Idle (SSPM<3:0> = 1011), or with either of the Slave modes in which interrupts are enabled. When both master and slave functionality is enabled, the software needs to differentiate the source(s) of the interrupt.

Refer to Application Note AN554, "*Software Implementation of I^2C™ Bus Master*" (DS00554) for more information.

17.2.9 MULTI-MASTER MODE

In Multi-Master mode, the interrupt generation on the detection of the Start and Stop conditions allow the determination of when the bus is free. The Stop (P) and Start (S) bits are cleared from a Reset or when the SSP module is disabled. The Stop (P) and Start (S) bits will toggle based on the Start and Stop conditions. Control of the I^2C bus may be taken when the P bit of the SSPSTAT register is set or when the bus is Idle, and both the S and P bits are clear. When the bus is busy, enabling the SSP Interrupt will generate the interrupt when the Stop condition occurs.

In Multi-Master operation, the SDA line must be monitored to see if the signal level is the expected output level. This check only needs to be done when a high level is output. If a high level is expected and a low level is present, the device needs to release the SDA and SCL lines (set TRIS bits). There are two stages where this arbitration of the bus can be lost. They are the address transfer and data transfer stages.

When the slave logic is enabled, the slave continues to receive. If arbitration was lost during the address transfer stage, communication to the device may be in progress. If addressed, an $\overline{ACK}$ pulse will be generated. If arbitration was lost during the data transfer stage, the device will need to re-transfer the data at a later time.

Refer to Application Note AN578, "*Use of the SSP Module in the I^2C™ Multi-Master Environment*" (DS00578) for more information.

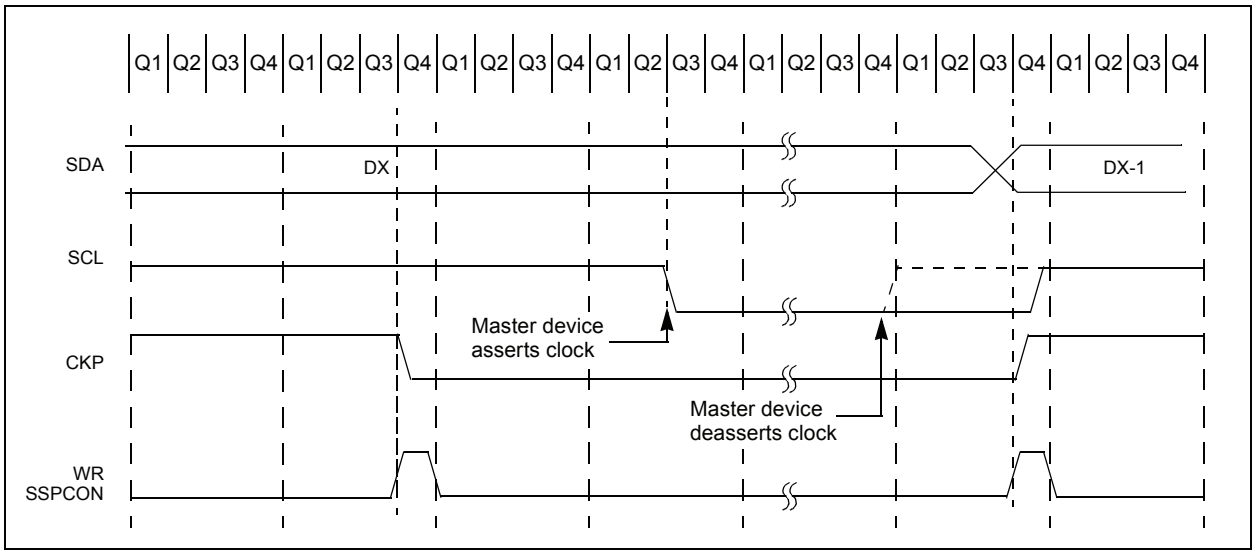
17.2.10 CLOCK SYNCHRONIZATION

When the CKP bit is cleared, the SCL output is held low once it is sampled low. Therefore, the CKP bit will not stretch the SCL line until an external I^2C master device has already asserted the SCL line low. The SCL output will remain low until the CKP bit is set and all other devices on the I^2C bus have released SCL. This ensures that a write to the CKP bit will not violate the minimum high time requirement for SCL (Figure 17-14).

17.2.11 SLEEP OPERATION

While in Sleep mode, the I^2C module can receive addresses of data, and when an address match or complete byte transfer occurs, wake the processor from Sleep (if SSP interrupt is enabled).

FIGURE 17-14: CLOCK SYNCHRONIZATION TIMING



REGISTER 17-3: SSPCON: SYNCHRONOUS SERIAL PORT CONTROL REGISTER (I²C MODE)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0
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 **WCOL**: Write Collision Detect bit
- 1 = The SSPBUF register is written while it is still transmitting the previous word (must be cleared in software)
- 0 = No collision

bit 6 **SSPOV**: Receive Overflow Indicator bit
- 1 = A byte is received while the SSPBUF register is still holding the previous byte. SSPOV is a "don't care" in Transmit mode. SSPOV must be cleared in software in either mode.
- 0 = No overflow

bit 5 **SSPEN**: Synchronous Serial Port Enable bit
- 1 = Enables the serial port and configures the SDA and SCL pins as serial port pins(2)
- 0 = Disables serial port and configures these pins as I/O port pins

bit 4 **CKP**: Clock Polarity Select bit
- 1 = Release control of SCL
- 0 = Holds clock low (clock stretch). (Used to ensure data setup time.)

bit 3-0 **SSPM<3:0>**: Synchronous Serial Port Mode Select bits
- 0110 = I²C Slave mode, 7-bit address
- 0111 = I²C Slave mode, 10-bit address
- 1000 = Reserved
- 1001 = Load SSPMSK register at SSPADD SFR Address(1)
- 1010 = Reserved
- 1011 = I²C Firmware Controlled Master mode (Slave Idle)
- 1100 = Reserved
- 1101 = Reserved
- 1110 = I²C Slave mode, 7-bit address with Start and Stop bit interrupts enabled
- 1111 = I²C Slave mode, 10-bit address with Start and Stop bit interrupts enabled

Note 1: When this mode is selected, any reads or writes to the SSPADD SFR address accesses the SSPMSK register.

2: When enabled, these pins must be properly configured as input or output using the associated TRIS bit.

REGISTER 17-4: SSPSTAT: SYNCHRONOUS SERIAL PORT STATUS REGISTER (I²C MODE)

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF
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 **SMP**: SPI Data Input Sample Phase bit
1 = Slew Rate Control (limiting) disabled. Operating in I²C Standard mode (100 kHz and 1 MHz).
0 = Slew Rate Control (limiting) enabled. Operating in I²C Fast mode (400 kHz).

bit 6 **CKE**: SPI Clock Edge Select bit
This bit must be maintained clear. Used in SPI mode only.

bit 5 **D/$\overline{A}$**: DATA/$\overline{ADDRESS}$ bit (I²C mode only)
1 = Indicates that the last byte received or transmitted was data
0 = Indicates that the last byte received or transmitted was address

bit 4 **P**: Stop bit
This bit is cleared when the SSP module is disabled, or when the Start bit is detected last.
1 = Indicates that a Stop bit has been detected last (this bit is '0' on Reset)
0 = Stop bit was not detected last

bit 3 **S**: Start bit
This bit is cleared when the SSP module is disabled, or when the Stop bit is detected last.
1 = Indicates that a Start bit has been detected last (this bit is '0' on Reset)
0 = Start bit was not detected last

bit 2 **R/$\overline{W}$**: READ/$\overline{WRITE}$ bit Information
This bit holds the R/$\overline{W}$ bit information following the last address match. This bit is only valid from the address match to the next Start bit, Stop bit or $\overline{ACK}$ bit.
1 = Read
0 = Write

bit 1 **UA**: Update Address bit (10-bit I²C mode only)
1 = Indicates that the user needs to update the address in the SSPADD register
0 = Address does not need to be updated

bit 0 **BF**: Buffer Full Status bit
Receive:
1 = Receive complete, SSPBUF is full
0 = Receive not complete, SSPBUF is empty
Transmit:
1 = Transmit in progress, SSPBUF is full
0 = Transmit complete, SSPBUF is empty

REGISTER 17-5: SSPMSK: SSP MASK REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0
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 7-1 **MSK<7:1>:** Mask bits
1 = The received address bit n is compared to SSPADD<n> to detect I²C address match
0 = The received address bit n is not used to detect I²C address match

bit 0 **MSK<0>:** Mask bit for I²C Slave Mode, 10-bit Address
I²C Slave Mode, 10-bit Address (SSPM<3:0> = 0111):
1 = The received address bit '0' is compared to SSPADD<0> to detect I²C address match
0 = The received address bit '0' is not used to detect I²C address match
All other SSP modes: this bit has no effect.

REGISTER 17-6: SSPADD: SSP I²C ADDRESS REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADD7	ADD6	ADD5	ADD4	ADD3	ADD2	ADD1	ADD0
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 7-0 **ADD<7:0>:** Address bits
Received address

TABLE 17-7: SUMMARY OF REGISTERS ASSOCIATED WITH I²C OPERATION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	40
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	43
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	41
SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								157
SSPADD	Synchronous Serial Port (I²C mode) Address Register								165
SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	174
SSPMSK(2)	Synchronous Serial Port (I²C mode) Address Mask Register								176
SSPSTAT	SMP(1)	CKE(1)	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF	175
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	67

Legend: x = unknown, u = unchanged, – = unimplemented locations read as '0'. Shaded cells are not used by SSP module in I²C mode.

Note 1: Maintain these bits clear in I²C mode.
2: Accessible only when SSPM<3:0> = 1001.

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

```
        BANKSEL PMADRL  ;
        MOVF    MS_PROG_ADDR, W;
        MOVWF   PMADRH  ;MS Byte of Program Address to read
        MOVF    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  ;
        MOVF    PMDATL, W;W = LS Byte of Program Memory Read
        MOVWF   LOWPMBYTE;
        MOVF    PMDATH, W;W = MS Byte of Program Memory Read
        MOVWF   HIGHPMBYTE;
```

Required Sequence

REGISTER 18-1: PMCON1: PROGRAM MEMORY CONTROL 1 REGISTER

R-1	U-0	U-0	U-0	U-0	U-0	U-0	R/S-0
Reserved	—	—I	—	—	—	—	RD
bit 7							bit 0

Legend:		S = Setable bit, cleared in hardware	
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 **Reserved:** Read as '1'. Maintain this bit set.

bit 6-1 **Unimplemented:** Read as '0'

bit 0 **RD:** Read Control bit

1 = Initiates an program memory read (The RD is cleared in hardware; the RD bit can only be set (not cleared) in software).

0 = Does not initiate a program memory read

REGISTER 18-2: PMDATH: PROGRAM MEMORY DATA HIGH REGISTER

U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	PMD13	PMD12	PMD11	PMD10	PMD9	PMD8
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 **PMD<13:8>:** The value of the program memory word pointed to by PMADRH and PMADRL after a Program Memory Read command.

REGISTER 18-3: PMDATL: PROGRAM MEMORY DATA LOW REGISTER

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
PMD7	PMD6	PMD5	PMD4	PMD3	PMD2	PMD1	PMD0
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 7-0 **PMD<7:0>:** The value of the program memory word pointed to by PMADRH and PMADRL after a Program Memory Read command.

REGISTER 18-4: PMADRH: PROGRAM MEMORY ADDRESS HIGH REGISTER

U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	—	PMA12	PMA11	PMA10	PMA9	PMA8
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 7-5 **Unimplemented:** Read as '0'

bit 4-0 **PMA<12:8>:** Program Memory Read Address bits

REGISTER 18-5: PMADRL: PROGRAM MEMORY ADDRESS LOW REGISTER

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
PMA7	PMA6	PMA5	PMA4	PMA3	PMA2	PMA1	PMA0
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 7-0 **PMA<7:0>:** Program Memory Read Address bits

TABLE 18-1: SUMMARY OF REGISTERS ASSOCIATED WITH PROGRAM MEMORY READ

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
PMCON1	Reserved	—	—	—	—	—	—	RD	178
PMADRH	—	—	—	Program Memory Read Address Register High Byte					179
PMADRL	Program Memory Read Address Register Low Byte								179
PMDATH	—	—	Program Memory Read Data Register High Byte						178
PMDATL	Program Memory Read Data Register Low Byte								178

Legend: x = unknown, u = unchanged, – = unimplemented, read as '0'. Shaded cells are not used by the Program Memory Read.

NOTES:

19.0 POWER-DOWN MODE (SLEEP)

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

If the Watchdog Timer is enabled:

- WDT will be cleared but keeps running.
- $\overline{PD}$ bit of the STATUS register is cleared.
- $\overline{TO}$ bit of the STATUS register is set.
- Oscillator driver is turned off.
- Timer1 oscillator is unaffected
- I/O ports maintain the status they had before SLEEP was executed (driving high, low or high-impedance).

For lowest current consumption in this mode, all I/O pins should be either at VDD or VSS, with no external circuitry drawing current from the I/O pin. I/O pins that are high-impedance inputs should be pulled high or low externally to avoid switching currents caused by floating inputs. The T0CKI input should also be at VDD or VSS for lowest current consumption. The contribution from on-chip pull-ups on PORTB should be considered.

The $\overline{MCLR}$ pin must be at a logic high level when external $\overline{MCLR}$ is enabled.

Note:	A Reset generated by a WDT time-out does not drive $\overline{MCLR}$ pin low.

19.1 Wake-up from Sleep

The device can wake-up from Sleep through one of the following events:

1. External Reset input on $\overline{MCLR}$ pin.
2. Watchdog Timer wake-up (if WDT was enabled).
3. Interrupt from RB0/INT pin, PORTB change or a peripheral interrupt.

The first event will cause a device Reset. The two latter events are considered a continuation of program execution. The $\overline{TO}$ and $\overline{PD}$ bits in the STATUS register can be used to determine the cause of device Reset. The $\overline{PD}$ bit, which is set on power-up, is cleared when Sleep is invoked. $\overline{TO}$ bit is cleared if WDT wake-up occurred.

The following peripheral interrupts can wake the device from Sleep:

1. TMR1 Interrupt. Timer1 must be operating as an asynchronous counter.
2. USART Receive Interrupt (Synchronous Slave mode only)
3. A/D conversion (when A/D clock source is RC)
4. Interrupt-on-change
5. External Interrupt from INT pin
6. Capture event on CCP1 or CCP2
7. SSP Interrupt in SPI or I^2C Slave mode

Other peripherals cannot generate interrupts since during Sleep, no on-chip clocks are present.

When the SLEEP instruction is being executed, the next instruction (PC + 1) is prefetched. For the device to wake-up through an interrupt event, the corresponding interrupt enable bit must be set (enabled). Wake-up is regardless of the state of the GIE bit. If the GIE bit is clear (disabled), the device continues execution at the instruction after the SLEEP instruction. If the GIE bit is set (enabled), the device executes the instruction after the SLEEP instruction, then branches to the interrupt address (0004h). In cases where the execution of the instruction following SLEEP is not desirable, the user should have a NOP after the SLEEP instruction.

Note:	If the global interrupts are disabled (GIE is cleared), but any interrupt source has both its interrupt enable bit and the corresponding interrupt flag bits set, the device will immediately wake-up from Sleep. The SLEEP instruction is completely executed.

The WDT is cleared when the device wakes up from Sleep, regardless of the source of wake-up.

19.2 Wake-up Using Interrupts

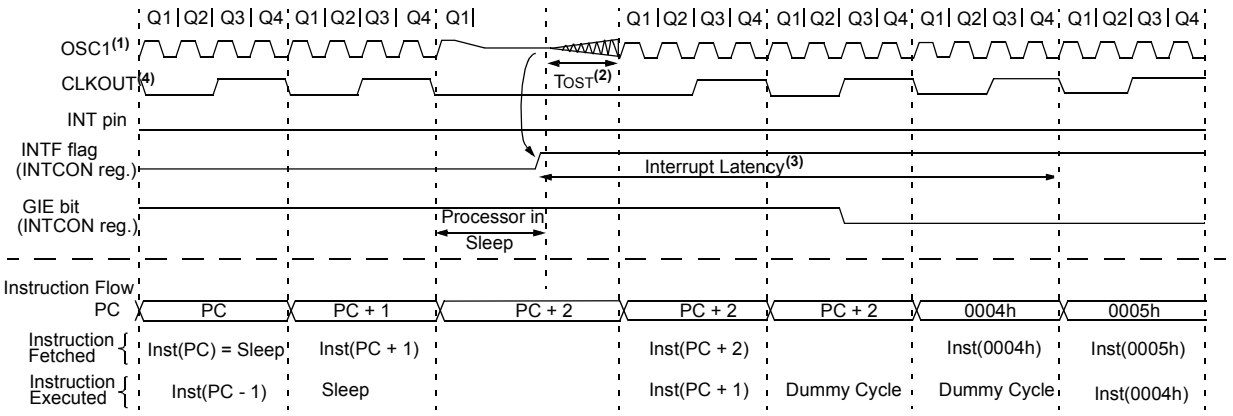
When global interrupts are disabled (GIE cleared) and any interrupt source has both its interrupt enable bit and interrupt flag bit set, one of the following will occur:

- If the interrupt occurs **before** the execution of a SLEEP instruction, the SLEEP instruction will complete as a NOP. Therefore, the WDT and WDT prescaler and postscaler (if enabled) will not be cleared, the $\overline{TO}$ bit will not be set and the $\overline{PD}$ bit will not be cleared.
- If the interrupt occurs **during or after** the execution of a SLEEP instruction, the device will immediately wake-up from Sleep. The SLEEP instruction will be completely executed before the wake-up. Therefore, the WDT and WDT prescaler and postscaler (if enabled) will be cleared, the $\overline{TO}$ bit will be set and the $\overline{PD}$ bit will be cleared.

Even if the flag bits were checked before executing a SLEEP instruction, it may be possible for flag bits to become set before the SLEEP instruction completes. To determine whether a SLEEP instruction executed, test the $\overline{PD}$ bit. If the $\overline{PD}$ bit is set, the SLEEP instruction was executed as a NOP.

To ensure that the WDT is cleared, a CLRWDT instruction should be executed before a SLEEP instruction.

FIGURE 19-1: WAKE-UP FROM SLEEP THROUGH INTERRUPT



Note 1: XT, HS or LP Oscillator mode assumed.
2: TOST = 1024 TOSC (drawing not to scale). This delay does not apply to EC and RC Oscillator modes.
3: GIE = 1 assumed. In this case after wake-up, the processor jumps to 0004h. If GIE = 0, execution will continue in-line.
4: CLKOUT is not available in XT, HS, LP or EC Oscillator modes, but shown here for timing reference.

TABLE 19-1: SUMMARY OF REGISTERS ASSOCIATED WITH POWER-DOWN MODE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
IOCB	IOCB7	IOCB6	IOCB5	IOCB4	IOCB3	IOCB2	IOCB1	IOCB0	58
INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	40
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	41
PIE2	—	—	—	—	—	—	—	CCP2IE	42
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	43
PIR2	—	—	—	—	—	—	—	CCP2IF	44

Legend: x = unknown, u = unchanged, – = unimplemented, read as '0'. Shaded cells are not used in Power-Down mode.

20.0 IN-CIRCUIT SERIAL PROGRAMMING™ (ICSP™)

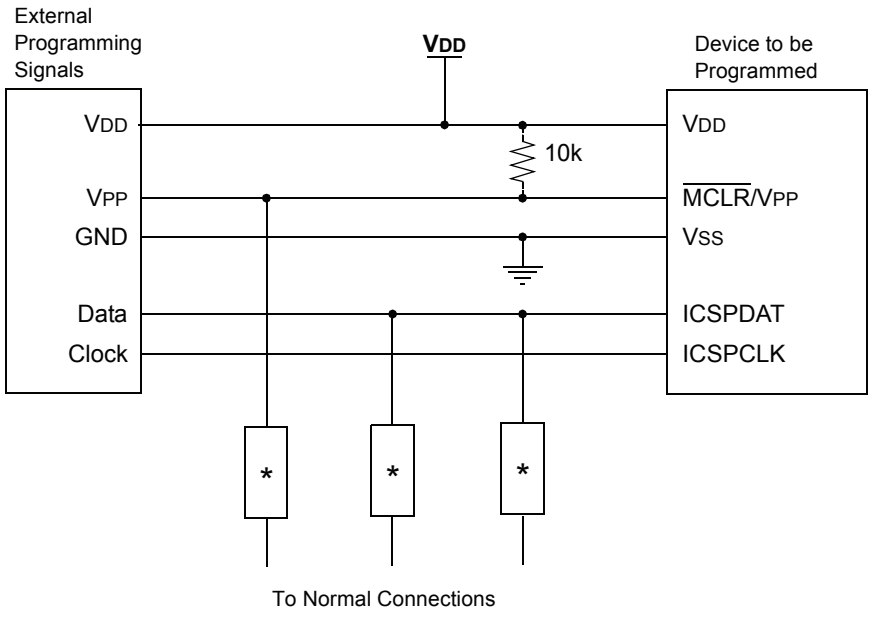
ICSP™ programming allows customers to manufacture circuit boards with unprogrammed devices. Programming can be done after the assembly process allowing the device to be programmed with the most recent firmware or a custom firmware. Five pins are needed for ICSP™ programming:

- 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 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 ISCPCLK pin is the clock input. For more information on ICSP™ refer to the *"PIC16F72X/PIC16LF72X 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



* Isolation devices (as required).

NOTES:

21.0 INSTRUCTION SET SUMMARY

The PIC16(L)F722A/723A instruction set is highly orthogonal and is comprised of three basic categories:

- **Byte-oriented** operations
- **Bit-oriented** operations
- **Literal and control** operations

Each PIC16 instruction is a 14-bit word divided into an **opcode,** which specifies the instruction type and one or more **operands,** which further specify the operation of the instruction. The formats for each of the categories is presented in Figure 21-1, while the various opcode fields are summarized in Table 21-1.

Table 21-2 lists the instructions recognized by the MPASM™ assembler.

For **byte-oriented** instructions, 'f' represents a file register designator and 'd' represents a destination designator. The file register designator specifies which file register is to be used by the instruction.

The destination designator specifies where the result of the operation is to be placed. If 'd' is zero, the result is placed in the W register. If 'd' is one, the result is placed in the file register specified in the instruction.

For **bit-oriented** instructions, 'b' represents a bit field designator, which selects the bit affected by the operation, while 'f' represents the address of the file in which the bit is located.

For **literal and control** operations, 'k' represents an 8-bit or 11-bit constant, or literal value.

One instruction cycle consists of four oscillator periods; for an oscillator frequency of 4 MHz, this gives a nominal instruction execution time of 1 μs. All instructions are executed within a single instruction cycle, unless a conditional test is true, or the program counter is changed as a result of an instruction. When this occurs, the execution takes two instruction cycles, with the second cycle executed as a NOP.

All instruction examples use the format '0xhh' to represent a hexadecimal number, where 'h' signifies a hexadecimal digit.

21.1 Read-Modify-Write Operations

Any instruction that specifies a file register as part of the instruction performs a Read-Modify-Write (R-M-W) operation. The register is read, the data is modified, and the result is stored according to either the instruction, or the destination designator 'd'. A read operation is performed on a register even if the instruction writes to that register.

For example, a CLRF PORTB instruction will read PORTB, clear all the data bits, then write the result back to PORTB. This example would have the unintended consequence of clearing the condition that set the RBIF flag.

TABLE 21-1: OPCODE FIELD DESCRIPTIONS

Field	Description
f	Register file address (0x00 to 0x7F)
W	Working register (accumulator)
b	Bit address within an 8-bit file register
k	Literal field, constant data or label
x	Don't care location (= 0 or 1). The assembler will generate code with x = 0. It is the recommended form of use for compatibility with all Microchip software tools.
d	Destination select; d = 0: store result in W, d = 1: store result in file register f. Default is d = 1.
PC	Program Counter
$\overline{TO}$	Time-out bit
C	Carry bit
DC	Digit carry bit
Z	Zero bit
$\overline{PD}$	Power-down bit

FIGURE 21-1: GENERAL FORMAT FOR INSTRUCTIONS

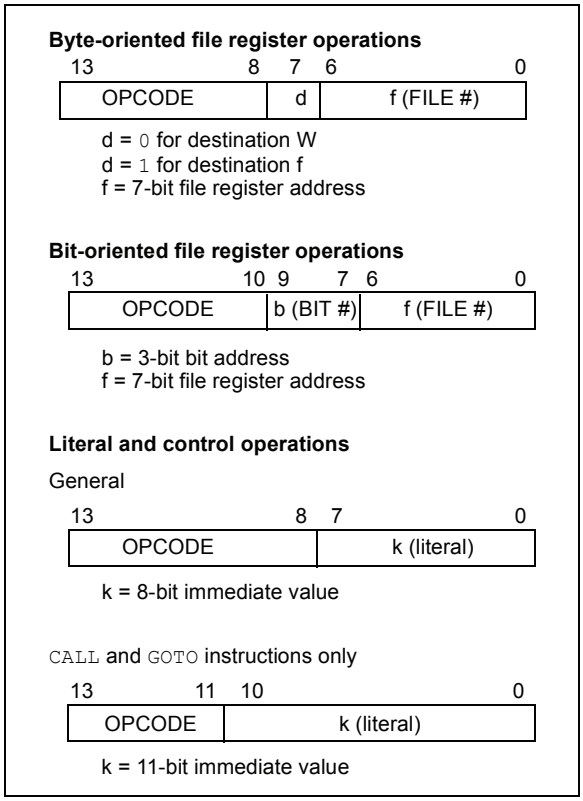


TABLE 21-2: PIC16(L)F722A/723A INSTRUCTION SET

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`.

21.2 Instruction Descriptions

ADDLW	Add literal and W
Syntax:	[*label*] ADDLW k
Operands:	0 ≤ k ≤ 255
Operation:	(W) + k → (W)
Status Affected:	C, DC, Z
Description:	The contents of the W register are added to the eight-bit literal 'k' and the result is placed in the W register.

ADDWF	Add W and f
Syntax:	[*label*] ADDWF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(W) + (f) → (destination)
Status Affected:	C, DC, Z
Description:	Add the contents of the W register with register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'.

ANDLW	AND literal with W
Syntax:	[*label*] ANDLW k
Operands:	0 ≤ k ≤ 255
Operation:	(W) .AND. (k) → (W)
Status Affected:	Z
Description:	The contents of W register are AND'ed with the eight-bit literal 'k'. The result is placed in the W register.

ANDWF	AND W with f
Syntax:	[*label*] ANDWF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(W) .AND. (f) → (destination)
Status Affected:	Z
Description:	AND the W register with register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'.

BCF	Bit Clear f
Syntax:	[*label*] BCF f,b
Operands:	0 ≤ f ≤ 127 0 ≤ b ≤ 7
Operation:	0 → (f<b>)
Status Affected:	None
Description:	Bit 'b' in register 'f' is cleared.

BSF	Bit Set f
Syntax:	[*label*] BSF f,b
Operands:	0 ≤ f ≤ 127 0 ≤ b ≤ 7
Operation:	1 → (f<b>)
Status Affected:	None
Description:	Bit 'b' in register 'f' is set.

BTFSC	Bit Test f, Skip if Clear
Syntax:	[*label*] BTFSC f,b
Operands:	0 ≤ f ≤ 127 0 ≤ b ≤ 7
Operation:	skip if (f<b>) = 0
Status Affected:	None
Description:	If bit 'b' in register 'f' is '1', the next instruction is executed. If bit 'b', in register 'f', is '0', the next instruction is discarded, and a NOP is executed instead, making this a 2-cycle instruction.

BTFSS	Bit Test f, Skip if Set
Syntax:	[*label*] BTFSS f,b
Operands:	0 ≤ f ≤ 127 0 ≤ b < 7
Operation:	skip if (f<b>) = 1
Status Affected:	None
Description:	If bit 'b' in register 'f' is '0', the next instruction is executed. If bit 'b' is '1', then the next instruction is discarded and a NOP is executed instead, making this a 2-cycle instruction.

CALL	Call Subroutine
Syntax:	[*label*] CALL k
Operands:	0 ≤ k ≤ 2047
Operation:	(PC)+ 1→ TOS, k → PC<10:0>, (PCLATH<4:3>) → PC<12:11>
Status Affected:	None
Description:	Call Subroutine. First, return address (PC + 1) is pushed onto the stack. The eleven-bit immediate address is loaded into PC bits <10:0>. The upper bits of the PC are loaded from PCLATH. CALL is a two-cycle instruction.

CLRF	Clear f
Syntax:	[*label*] CLRF f
Operands:	0 ≤ f ≤ 127
Operation:	00h → (f) 1 → Z
Status Affected:	Z
Description:	The contents of register 'f' are cleared and the Z bit is set.

CLRW	Clear W
Syntax:	[*label*] CLRW
Operands:	None
Operation:	00h → (W) 1 → Z
Status Affected:	Z
Description:	W register is cleared. Zero bit (Z) is set.

CLRWDT	Clear Watchdog Timer
Syntax:	[*label*] CLRWDT
Operands:	None
Operation:	00h → WDT 0 → WDT prescaler, 1 → $\overline{TO}$ 1 → $\overline{PD}$
Status Affected:	$\overline{TO}$, $\overline{PD}$
Description:	CLRWDT instruction resets the Watchdog Timer. It also resets the prescaler of the WDT. Status bits $\overline{TO}$ and $\overline{PD}$ are set.

COMF	Complement f
Syntax:	[*label*] COMF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	($\bar{f}$) → (destination)
Status Affected:	Z
Description:	The contents of register 'f' are complemented. If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f'.

DECF	Decrement f
Syntax:	[*label*] DECF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(f) - 1 → (destination)
Status Affected:	Z
Description:	Decrement register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'.

DECFSZ	Decrement f, Skip if 0
Syntax:	[*label*] DECFSZ f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(f) - 1 → (destination); skip if result = 0
Status Affected:	None
Description:	The contents of register 'f' are decremented. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'. If the result is '1', the next instruction is executed. If the result is '0', then a NOP is executed instead, making it a 2-cycle instruction.

GOTO	Unconditional Branch
Syntax:	[*label*] GOTO k
Operands:	$0 \le k \le 2047$
Operation:	k → PC<10:0> PCLATH<4:3> → PC<12:11>
Status Affected:	None
Description:	GOTO is an unconditional branch. The eleven-bit immediate value is loaded into PC bits <10:0>. The upper bits of PC are loaded from PCLATH<4:3>. GOTO is a two-cycle instruction.

INCF	Increment f
Syntax:	[*label*] INCF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(f) + 1 → (destination)
Status Affected:	Z
Description:	The contents of register 'f' are incremented. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'.

INCFSZ	Increment f, Skip if 0
Syntax:	[*label*] INCFSZ f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(f) + 1 → (destination), skip if result = 0
Status Affected:	None
Description:	The contents of register 'f' are incremented. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'. If the result is '1', the next instruction is executed. If the result is '0', a NOP is executed instead, making it a 2-cycle instruction.

IORLW	Inclusive OR literal with W
Syntax:	[*label*] IORLW k
Operands:	$0 \le k \le 255$
Operation:	(W) .OR. k → (W)
Status Affected:	Z
Description:	The contents of the W register are OR'ed with the eight-bit literal 'k'. The result is placed in the W register.

IORWF	Inclusive OR W with f
Syntax:	[*label*] IORWF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(W) .OR. (f) → (destination)
Status Affected:	Z
Description:	Inclusive OR the W register with register 'f'. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'.

MOVF	Move f
Syntax:	[*label*] MOVF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(f) → (dest)
Status Affected:	Z
Description:	The contents of register f is moved to a destination dependent upon the status of d. If d = 0, destination is W register. If d = 1, the destination is file register f itself. d = 1 is useful to test a file register since status flag Z is affected.
Words:	1
Cycles:	1
Example:	MOVF FSR, 0 After Instruction W = value in FSR register Z = 1

MOVWF	Move W to f
Syntax:	[*label*] MOVWF f
Operands:	$0 \le f \le 127$
Operation:	(W) → (f)
Status Affected:	None
Description:	Move data from W register to register 'f'.
Words:	1
Cycles:	1
Example:	MOVW OPTION F Before Instruction OPTION = 0xFF W = 0x4F After Instruction OPTION = 0x4F W = 0x4F

MOVLW	Move literal to W
Syntax:	[*label*] MOVLW k
Operands:	$0 \le k \le 255$
Operation:	k → (W)
Status Affected:	None
Description:	The eight-bit literal 'k' is loaded into W register. The "don't cares" will assemble as '0's.
Words:	1
Cycles:	1
Example:	MOVLW 0x5A After Instruction W = 0x5A

NOP	No Operation
Syntax:	[*label*] NOP
Operands:	None
Operation:	No operation
Status Affected:	None
Description:	No operation.
Words:	1
Cycles:	1
Example:	NOP

RETFIE	Return from Interrupt
Syntax:	[*label*] RETFIE
Operands:	None
Operation:	TOS → PC, 1 → GIE
Status Affected:	None
Description:	Return from Interrupt. Stack is POPed and Top-of-Stack (TOS) is loaded in the PC. Interrupts are enabled by setting Global Interrupt Enable bit, GIE (INTCON<7>). This is a two-cycle instruction.
Words:	1
Cycles:	2
Example:	RETFIE After Interrupt PC = TOS GIE = 1

RETLW	Return with literal in W
Syntax:	[*label*] RETLW k
Operands:	0 ≤ k ≤ 255
Operation:	k → (W); TOS → PC
Status Affected:	None
Description:	The W register is loaded with the eight bit literal ‘k’. The program counter is loaded from the top of the stack (the return address). This is a two-cycle instruction.
Words:	1
Cycles:	2
Example:	CALL TABLE;W contains table ;offset value TABLE ;W now has table value • • ADDWF PC ;W = offset RETLW k1 ;Begin table RETLW k2 ; • • • RETLW kn ; End of table Before Instruction W = 0x07 After Instruction W = value of k8

RETURN	Return from Subroutine
Syntax:	[*label*] RETURN
Operands:	None
Operation:	TOS → PC
Status Affected:	None
Description:	Return from subroutine. The stack is POPed and the top of the stack (TOS) is loaded into the program counter. This is a two-cycle instruction.

RLF **Rotate Left f through Carry**

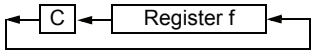
Syntax: [*label*] RLF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: See description below

Status Affected: C

Description: The contents of register 'f' are rotated one bit to the left through the Carry flag. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is stored back in register 'f'.



Words: 1

Cycles: 1

Example:

```
RLF     REG1,0
```

Before Instruction

```
REG1    =   1110 0110
C       =   0
```

After Instruction

```
REG1    =   1110 0110
W       =   1100 1100
C       =   1
```

RRF **Rotate Right f through Carry**

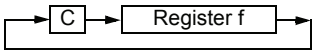
Syntax: [*label*] RRF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: See description below

Status Affected: C

Description: The contents of register 'f' are rotated one bit to the right through the Carry flag. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'.



SLEEP **Enter Sleep mode**

Syntax: [*label*] SLEEP

Operands: None

Operation: 00h → WDT,
0 → WDT prescaler,
1 → $\overline{TO}$,
0 → $\overline{PD}$

Status Affected: $\overline{TO}$, $\overline{PD}$

Description: The power-down Status bit, $\overline{PD}$ is cleared. Time-out Status bit, $\overline{TO}$ is set. Watchdog Timer and its prescaler are cleared.
The processor is put into Sleep mode with the oscillator stopped.

SUBLW **Subtract W from literal**

Syntax: [*label*] SUBLW k

Operands: $0 \le k \le 255$

Operation: k - (W) → (W)

Status Affected: C, DC, Z

Description: The W register is subtracted (2's complement method) from the eight-bit literal 'k'. The result is placed in the W register.

C = 0	W > k
C = 1	W ≤ k
DC = 0	W<3:0> > k<3:0>
DC = 1	W<3:0> ≤ k<3:0>

SUBWF **Subtract W from f**

Syntax: [*label*] SUBWF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (f) - (W) → (destination)

Status Affected: C, DC, Z

Description: Subtract (2's complement method) W register from register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f.

C = 0	W > f
C = 1	W ≤ f
DC = 0	W<3:0> > f<3:0>
DC = 1	W<3:0> ≤ f<3:0>

SWAPF **Swap Nibbles in f**

Syntax: [*label*] SWAPF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (f<3:0>) → (destination<7:4>),
(f<7:4>) → (destination<3:0>)

Status Affected: None

Description: The upper and lower nibbles of register 'f' are exchanged. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed in register 'f'.

XORLW **Exclusive OR literal with W**

Syntax: [*label*] XORLW k

Operands: $0 \le k \le 255$

Operation: (W) .XOR. k → (W)

Status Affected: Z

Description: The contents of the W register are XOR'ed with the eight-bit literal 'k'. The result is placed in the W register.

XORWF **Exclusive OR W with f**

Syntax: [*label*] XORWF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (W) .XOR. (f) → (destination)

Status Affected: Z

Description: Exclusive OR the contents of the W register with register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'.

NOTES:

22.0 DEVELOPMENT SUPPORT

The PIC® microcontrollers and dsPIC® digital signal controllers are supported with a full range of software and hardware development tools:

- Integrated Development Environment
  - MPLAB® IDE Software
- Compilers/Assemblers/Linkers
  - MPLAB C Compiler for Various Device Families
  - HI-TECH C® for Various Device Families
  - MPASM™ Assembler
  - MPLINK™ Object Linker/ MPLIB™ Object Librarian
  - MPLAB Assembler/Linker/Librarian for Various Device Families
- Simulators
  - MPLAB SIM Software Simulator
- Emulators
  - MPLAB REAL ICE™ In-Circuit Emulator
- In-Circuit Debuggers
  - MPLAB ICD 3
  - PICkit™ 3 Debug Express
- Device Programmers
  - PICkit™ 2 Programmer
  - MPLAB PM3 Device Programmer
- Low-Cost Demonstration/Development Boards, Evaluation Kits, and Starter Kits

22.1 MPLAB Integrated Development Environment Software

The MPLAB IDE software brings an ease of software development previously unseen in the 8/16/32-bit microcontroller market. The MPLAB IDE is a Windows® operating system-based application that contains:

- A single graphical interface to all debugging tools
  - Simulator
  - Programmer (sold separately)
  - In-Circuit Emulator (sold separately)
  - In-Circuit Debugger (sold separately)
- A full-featured editor with color-coded context
- A multiple project manager
- Customizable data windows with direct edit of contents
- High-level source code debugging
- Mouse over variable inspection
- Drag and drop variables from source to watch windows
- Extensive on-line help
- Integration of select third party tools, such as IAR C Compilers

The MPLAB IDE allows you to:

- Edit your source files (either C or assembly)
- One-touch compile or assemble, and download to emulator and simulator tools (automatically updates all project information)
- Debug using:
  - Source files (C or assembly)
  - Mixed C and assembly
  - Machine code

MPLAB IDE supports multiple debugging tools in a single development paradigm, from the cost-effective simulators, through low-cost in-circuit debuggers, to full-featured emulators. This eliminates the learning curve when upgrading to tools with increased flexibility and power.

22.2 MPLAB C Compilers for Various Device Families

The MPLAB C Compiler code development systems are complete ANSI C compilers for Microchip's PIC18, PIC24 and PIC32 families of microcontrollers and the dsPIC30 and dsPIC33 families of digital signal controllers. These compilers provide powerful integration capabilities, superior code optimization and ease of use.

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

22.3 HI-TECH C for Various Device Families

The HI-TECH C Compiler code development systems are complete ANSI C compilers for Microchip's PIC family of microcontrollers and the dsPIC family of digital signal controllers. These compilers provide powerful integration capabilities, omniscient code generation and ease of use.

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

The compilers include a macro assembler, linker, preprocessor, and one-step driver, and can run on multiple platforms.

22.4 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 IDE projects
- User-defined macros to streamline assembly code
- Conditional assembly for multi-purpose source files
- Directives that allow complete control over the assembly process

22.5 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler and the MPLAB C18 C Compiler. 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/library 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

22.6 MPLAB Assembler, Linker and Librarian for Various Device Families

MPLAB Assembler produces relocatable machine code from symbolic assembly language for PIC24, PIC32 and dsPIC devices. MPLAB C 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 IDE compatibility

22.7 MPLAB SIM Software Simulator

The MPLAB SIM Software Simulator allows code development in a PC-hosted environment by simulating the PIC MCUs and dsPIC® DSCs on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a comprehensive stimulus controller. Registers can be logged to files for further run-time analysis. The trace buffer and logic analyzer display extend the power of the simulator to record and track program execution, actions on I/O, most peripherals and internal registers.

The MPLAB SIM Software Simulator fully supports symbolic debugging using the MPLAB C Compilers, and the MPASM and MPLAB Assemblers. The software simulator offers the flexibility to develop and debug code outside of the hardware laboratory environment, making it an excellent, economical software development tool.

22.8 MPLAB REAL ICE In-Circuit Emulator System

MPLAB REAL ICE In-Circuit Emulator System is Microchip's next generation high-speed emulator for Microchip Flash DSC and MCU devices. It debugs and programs PIC® Flash MCUs and dsPIC® Flash DSCs with the easy-to-use, powerful graphical user interface of the MPLAB Integrated Development Environment (IDE), included with each kit.

The emulator is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with either a connector compatible with in-circuit debugger systems (RJ11) or with the new high-speed, noise tolerant, Low-Voltage Differential Signal (LVDS) interconnection (CAT5).

The emulator is field upgradable through future firmware downloads in MPLAB IDE. In upcoming releases of MPLAB IDE, new devices will be supported, and new features will be added. MPLAB REAL ICE offers significant advantages over competitive emulators including low-cost, full-speed emulation, run-time variable watches, trace analysis, complex breakpoints, a ruggedized probe interface and long (up to three meters) interconnection cables.

22.9 MPLAB ICD 3 In-Circuit Debugger System

MPLAB ICD 3 In-Circuit Debugger System is Microchip's most cost effective high-speed hardware debugger/programmer for Microchip Flash Digital Signal Controller (DSC) and microcontroller (MCU) devices. It debugs and programs PIC® Flash microcontrollers and dsPIC® DSCs with the powerful, yet easy-to-use graphical user interface of MPLAB Integrated Development Environment (IDE).

The MPLAB ICD 3 In-Circuit Debugger probe is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with a connector compatible with the MPLAB ICD 2 or MPLAB REAL ICE systems (RJ-11). MPLAB ICD 3 supports all MPLAB ICD 2 headers.

22.10 PICkit 3 In-Circuit Debugger/ Programmer and PICkit 3 Debug Express

The MPLAB PICkit 3 allows debugging and programming of PIC® and dsPIC® Flash microcontrollers at a most affordable price point using the powerful graphical user interface of the MPLAB Integrated Development Environment (IDE). The MPLAB PICkit 3 is connected to the design engineer's PC using a full speed USB interface and can be connected to the target via an Microchip debug (RJ-11) connector (compatible with MPLAB ICD 3 and MPLAB REAL ICE). The connector uses two device I/O pins and the reset line to implement in-circuit debugging and In-Circuit Serial Programming™.

The PICkit 3 Debug Express include the PICkit 3, demo board and microcontroller, hookup cables and CDROM with user's guide, lessons, tutorial, compiler and MPLAB IDE software.

22.11 PICkit 2 Development Programmer/Debugger and PICkit 2 Debug Express

The PICkit™ 2 Development Programmer/Debugger is a low-cost development tool with an easy to use interface for programming and debugging Microchip's Flash families of microcontrollers. The full featured Windows® programming interface supports baseline (PIC10F, PIC12F5xx, PIC16F5xx), midrange (PIC12F6xx, PIC16F), PIC18F, PIC24, dsPIC30, dsPIC33, and PIC32 families of 8-bit, 16-bit, and 32-bit microcontrollers, and many Microchip Serial EEPROM products. With Microchip's powerful MPLAB Integrated Development Environment (IDE) the PICkit™ 2 enables in-circuit debugging on most PIC® microcontrollers. In-Circuit-Debugging runs, halts and single steps the program while the PIC microcontroller is embedded in the application. When halted at a breakpoint, the file registers can be examined and modified.

The PICkit 2 Debug Express include the PICkit 2, demo board and microcontroller, hookup cables and CDROM with user's guide, lessons, tutorial, compiler and MPLAB IDE software.

22.12 MPLAB PM3 Device Programmer

The MPLAB PM3 Device Programmer is a universal, CE compliant device programmer with programmable voltage verification at VDDMIN and VDDMAX for maximum reliability. It features a large LCD display (128 x 64) for menus and error messages and a modular, detachable socket assembly to support various package types. The ICSP™ cable assembly is included as a standard item. In Stand-Alone mode, the MPLAB PM3 Device Programmer can read, verify and program PIC devices without a PC connection. It can also set code protection in this mode. The MPLAB PM3 connects to the host PC via an RS-232 or USB cable. The MPLAB PM3 has high-speed communications and optimized algorithms for quick programming of large memory devices and incorporates an MMC card for file storage and data applications.

22.13 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.

23.0 ELECTRICAL SPECIFICATIONS

Absolute Maximum Ratings(†)

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

23.1 DC Characteristics: PIC16(L)F722A/723A-I/E (Industrial, Extended)

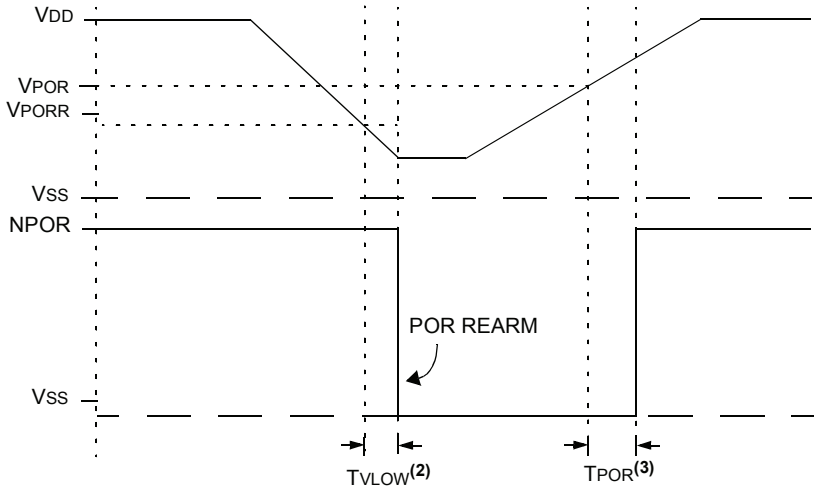
PIC16LF722A/723A			Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended				
PIC16F722A/723A			**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended				
Param. No.	**Sym.**	**Characteristic**	**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**
D001	VDD	**Supply Voltage**					
		PIC16LF722A/723A	1.8 1.8 2.3 2.5	— — — —	3.6 3.6 3.6 3.6	V V V V	FOSC ≤ 16 MHz: HFINTOSC, EC FOSC ≤ 4 MHz FOSC ≤ 20 MHz, EC FOSC ≤ 20 MHz, HS
D001		**PIC16F722A/723A**	1.8 1.8 2.3 2.5	— — — —	5.5 5.5 5.5 5.5	V V V V	FOSC ≤ 16 MHz: HFINTOSC, EC FOSC ≤ 4 MHz FOSC ≤ 20 MHz, EC FOSC ≤ 20 MHz, HS
D002*	VDR	**RAM Data Retention Voltage(1)**					
		PIC16LF722A/723A	1.5	—	—	V	Device in Sleep mode
D002*		**PIC16F722A/723A**	1.7	—	—	V	Device in Sleep mode
	VPOR*	**Power-on Reset Release Voltage**	—	1.6	—	V	
	VPORR*	**Power-on Reset Rearm Voltage**					
		PIC16LF722A/723A	—	0.8	—	V	Device in Sleep mode
		PIC16F722A/723A	—	1.7	—	V	Device in Sleep mode
D003	VFVR	**Fixed Voltage Reference Voltage, Initial Accuracy**	-5.5 -5.5 -5.5	— — —	5.5 5.5 5.5	% % %	VFVR = 1.024V, VDD ≥ 2.5V VFVR = 2.048V, VDD ≥ 2.5V VFVR = 4.096V, VDD ≥ 4.75V; -40 ≤ TA ≤ 85°C
			-6 -6 -6	— — —	6 6 6	% % %	VFVR = 1.024V, VDD ≥ 2.5V VFVR = 2.048V, VDD ≥ 2.5V VFVR = 4.096V, VDD ≥ 4.75V; -40 ≤ TA ≤ 125°C
D004*	SVDD	**VDD Rise Rate** to ensure internal Power-on Reset signal	0.05	—	—	V/ms	See **Section 3.2 "Power-on Reset (POR)"** for details.

* These parameters are characterized but not tested.

† Data in "Typ" column is at 3.3V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: This is the limit to which VDD can be lowered in Sleep mode without losing RAM data.

FIGURE 23-1: POR AND POR REARM WITH SLOW RISING VDD



Note 1: When NPOR is low, the device is held in Reset.
2: TPOR 1 μs typical.
3: TVLOW 2.7 μs typical.

23.2 DC Characteristics: PIC16(L)F722A/723A-I/E (Industrial, Extended)

PIC16LF722A/723A			Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended				
PIC16F722A/723A			Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended				
Param No.	Device Characteristics	Min.	Typ†	Max.	Units	Conditions	
						VDD	Note
	Supply Current (IDD)(1, 2)						
D009	LDO Regulator	—	350	—	µA	—	HS, EC OR INTOSC/INTOSCIO (8-16 MHz) Clock modes with all VCAP pins disabled
		—	50	—	µA	—	All VCAP pins disabled
		—	30	—	µA	—	VCAP enabled on RA0, RA5 or RA6
		—	5	—	µA	—	LP Clock mode and Sleep (requires FVR and BOR to be disabled)
D010		—	7.0	12	µA	1.8	FOSC = 32 kHz LP Oscillator mode **(Note 4)**, -40°C ≤ TA ≤ +85°C
		—	9.0	14	µA	3.0	
D010		—	11	20	µA	1.8	FOSC = 32 kHz LP Oscillator mode **(Note 4)**, -40°C ≤ TA ≤ +85°C
		—	14	22	µA	3.0	
		—	15	24	µA	5.0	
D011		—	7.0	12	µA	1.8	FOSC = 32 kHz LP Oscillator mode -40°C ≤ TA ≤ +125°C
		—	9.0	18	µA	3.0	
D011		—	11	21	µA	1.8	FOSC = 32 kHz LP Oscillator mode **(Note 4)** -40°C ≤ TA ≤ +125°C
		—	14	25	µA	3.0	
		—	15	27	µA	5.0	
D011		—	110	150	µA	1.8	FOSC = 1 MHz XT Oscillator mode
		—	150	215	µA	3.0	
D011		—	120	175	µA	1.8	FOSC = 1 MHz XT Oscillator mode **(Note 5)**
		—	180	250	µA	3.0	
		—	240	300	µA	5.0	
D012		—	230	300	µA	1.8	FOSC = 4 MHz XT Oscillator mode
		—	400	600	µA	3.0	
D012		—	250	350	µA	1.8	FOSC = 4 MHz XT Oscillator mode **(Note 5)**
		—	420	650	µA	3.0	
		—	500	750	µA	5.0	
D013		—	125	180	µA	1.8	FOSC = 1 MHz EC Oscillator mode
		—	230	270	µA	3.0	
D013		—	150	205	µA	1.8	FOSC = 1 MHz EC Oscillator mode **(Note 5)**
		—	225	320	µA	3.0	
		—	250	410	µA	5.0	

Note 1: The test conditions for all IDD measurements in active operation mode are: OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD; $\overline{MCLR}$ = VDD; WDT disabled.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors, such as I/O pin loading and switching rate, oscillator type, internal code execution pattern and temperature, also have an impact on the current consumption.

3: For RC oscillator configurations, current through REXT is not included. The current through the resistor can be extended by the formula IR = VDD/2REXT (mA) with REXT in kΩ.

4: FVR and BOR are disabled.

5: 0.1 µF capacitor on VCAP (RA0).

23.2 DC Characteristics: PIC16(L)F722A/723A-I/E (Industrial, Extended) (Continued)

PIC16LF722A/723A			Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended				
PIC16F722A/723A			**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended				
Param No.	**Device Characteristics**	**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**	
						VDD	**Note**
	Supply Current (IDD)(1, 2)						
D014		—	290	330	µA	1.8	FOSC = 4 MHz EC Oscillator mode
		—	460	500	µA	3.0	
D014		—	300	430	µA	1.8	FOSC = 4 MHz EC Oscillator mode **(Note 5)**
		—	450	655	µA	3.0	
		—	500	730	µA	5.0	
D015		—	100	130	µA	1.8	FOSC = 500 kHz MFINTOSC mode
		—	120	150	µA	3.0	
D015		—	115	195	µA	1.8	FOSC = 500 kHz MFINTOSC mode **(Note 5)**
		—	135	200	µA	3.0	
		—	150	220	µA	5.0	
D016		—	650	800	µA	1.8	FOSC = 8 MHz HFINTOSC mode
		—	1000	1200	µA	3.0	
D016		—	625	850	µA	1.8	FOSC = 8 MHz HFINTOSC mode **(Note 5)**
		—	1000	1200	µA	3.0	
		—	1100	1500	µA	5.0	
D017		—	1.0	1.2	mA	1.8	FOSC = 16 MHz HFINTOSC mode
		—	1.5	1.85	mA	3.0	
D017		—	1	1.2	mA	1.8	FOSC = 16 MHz HFINTOSC mode **(Note 5)**
		—	1.5	1.7	mA	3.0	
		—	1.7	2.1	mA	5.0	
D018		—	210	240	µA	1.8	FOSC = 4 MHz EXTRC mode **(Note 3, Note 5)**
		—	340	380	µA	3.0	
D018		—	225	320	µA	1.8	FOSC = 4 MHz EXTRC mode **(Note 3, Note 5)**
		—	360	445	µA	3.0	
		—	410	650	µA	5.0	
D019		—	1.6	1.9	mA	3.0	FOSC = 20 MHz HS Oscillator mode
		—	2.0	2.8	mA	3.6	
D019		—	1.6	2	mA	3.0	FOSC = 20 MHz HS Oscillator mode **(Note 5)**
		—	1.9	3.2	mA	5.0	

Note 1: The test conditions for all IDD measurements in active operation mode are: OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD; $\overline{MCLR}$ = VDD; WDT disabled.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors, such as I/O pin loading and switching rate, oscillator type, internal code execution pattern and temperature, also have an impact on the current consumption.

3: For RC oscillator configurations, current through REXT is not included. The current through the resistor can be extended by the formula IR = VDD/2REXT (mA) with REXT in kΩ.

4: FVR and BOR are disabled.

5: 0.1 µF capacitor on VCAP (RA0).

23.3 DC Characteristics: PIC16(L)F722A/723A-I/E (Power-Down)

PIC16LF722A/723A			**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended					
PIC16F722A/723A			**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended					
Param No.	**Device Characteristics**	**Min.**	**Typ†**	**Max. +85°C**	**Max. +125°C**	**Units**	**Conditions**	
							VDD	**Note**
	Power-down Base Current (IPD)(2)							
D020		—	0.02	0.7	3.9	µA	1.8	WDT, BOR, FVR, and T1OSC disabled, all Peripherals Inactive
		—	0.08	1.0	4.3	µA	3.0	
D020		—	4.3	10.2	17	µA	1.8	WDT, BOR, FVR, and T1OSC disabled, all Peripherals Inactive
		—	5	10.5	18	µA	3.0	
		—	5.5	11.8	21	µA	5.0	
D021		—	0.5	1.7	4.1	µA	1.8	LPWDT Current **(Note 1)**
		—	0.8	2.5	4.8	µA	3.0	
D021		—	6	13.5	16.4	µA	1.8	LPWDT Current **(Note 1)**
		—	6.5	14.5	16.8	µA	3.0	
		—	7.5	16	18.7	µA	5.0	
D021A		—	8.5	14	19	µA	1.8	FVR current **(Note 1. Note 3)**
		—	8.5	14	20	µA	3.0	
D021A		—	23	44	48	µA	1.8	FVR current **(Note 1, Note 3, Note 5)**
		—	25	45	55	µA	3.0	
		—	26	60	70	µA	5.0	
D022		—	—	—	—	µA	1.8	BOR Current **(Note 1, Note 3)**
		—	7.5	12	22	µA	3.0	
D022		—	—	—	—	µA	1.8	BOR Current **(Note 1, Note 3, Note 5)**
		—	23	42	49	µA	3.0	
		—	25	46	50	µA	5.0	
D026		—	0.6	2	—	µA	1.8	T1OSC Current **(Note 1)**
		—	1.8	3.0	—	µA	3.0	
D026		—	4.5	11.1	—	µA	1.8	T1OSC Current **(Note 1)**
		—	6	12.5	—	µA	3.0	
		—	7	13.5	—	µA	5.0	

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

Note 1: The peripheral current is the sum of the base IDD or IPD and the additional current consumed when this peripheral is enabled. The peripheral Δ current can be determined by subtracting the base IDD or IPD current from this limit. Max values should be used when calculating total current consumption.

2: The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to VDD.

3: Fixed Voltage Reference is automatically enabled whenever the BOR is enabled.

4: A/D oscillator source is FRC.

5: 0.1 µF capacitor on VCAP (RA0).

23.3 DC Characteristics: PIC16(L)F722A/723A-I/E (Power-Down) (Continued)

PIC16LF722A/723A			Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended					
PIC16F722A/723A			Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended					
Param No.	Device Characteristics	Min.	Typ†	Max. +85°C	Max. +125°C	Units	Conditions	
							VDD	Note
	Power-down Base Current (IPD)(2)							
D027		—	0.06	0.7	5.0	µA	1.8	A/D Current **(Note 1, Note 4)**, no conversion in progress
		—	0.08	1.0	5.5	µA	3.0	
D027		—	6	10.7	18	µA	1.8	A/D Current **(Note 1, Note 4)**, no conversion in progress
		—	7	10.6	20	µA	3.0	
		—	7.2	11.9	22	µA	5.0	
D027A		—	250	400	—	µA	1.8	A/D Current **(Note 1, Note 4)**, conversion in progress
		—	250	400	—	µA	3.0	
D027A		—	280	430	—	µA	1.8	A/D Current **(Note 1, Note 4, Note 5)**, conversion in progress
		—	280	430	—	µA	3.0	
		—	280	430	—	µA	5.0	
D028		—	2.2	3.2	14.4	µA	1.8	Cap Sense Low Power Oscillator mode
		—	3.3	4.4	15.6	µA	3.0	
D028		—	6.5	13	21	µA	1.8	Cap Sense Low Power Oscillator mode
		—	8	14	23	µA	3.0	
		—	8	14	25	µA	5.0	
D028A		—	4.2	6	17	µA	1.8	Cap Sense Medium Power Oscillator mode
		—	6	7	18	µA	3.0	
D028A		—	8.5	15.5	23	µA	1.8	Cap Sense Medium Power Oscillator mode
		—	11	17	24	µA	3.0	
		—	11	18	27	µA	5.0	
D028B		—	12	14	25	µA	1.8	Cap Sense High Power Oscillator mode
		—	32	35	44	µA	3.0	
D028B		—	16	20	31	µA	1.8	Cap Sense High Power Oscillator mode
		—	36	41	50	µA	3.0	
		—	42	49	58	µA	5.0	

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

Note 1: The peripheral current is the sum of the base IDD or IPD and the additional current consumed when this peripheral is enabled. The peripheral Δ current can be determined by subtracting the base IDD or IPD current from this limit. Max values should be used when calculating total current consumption.

2: The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to VDD.

3: Fixed Voltage Reference is automatically enabled whenever the BOR is enabled.

4: A/D oscillator source is FRC.

5: 0.1 µF capacitor on VCAP (RA0).

23.4 DC Characteristics: PIC16(L)F722A/723A-I/E

DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended				
Param No.	**Sym.**	**Characteristic**	**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**
	VIL	**Input Low Voltage**					
		I/O PORT:					
D030		with TTL buffer	—	—	0.8	V	4.5V ≤ VDD ≤ 5.5V
D030A			—	—	0.15 VDD	V	1.8V ≤ VDD ≤ 4.5V
D031		with Schmitt Trigger buffer	—	—	0.2 VDD	V	2.0V ≤ VDD ≤ 5.5V
		with I²C™ levels	—	—	0.3 VDD	V	
D032		$\overline{MCLR}$, OSC1 (RC mode)(1)	—	—	0.2 VDD	V	
D033A		OSC1 (HS mode)	—	—	0.3 VDD	V	
	VIH	**Input High Voltage**					
		I/O ports:		—	—		
D040		with TTL buffer	2.0	—	—	V	4.5V ≤ VDD ≤ 5.5V
D040A			0.25 VDD + 0.8	—	—	V	1.8V ≤ VDD ≤ 4.5V
D041		with Schmitt Trigger buffer	0.8 VDD	—	—	V	2.0V ≤ VDD ≤ 5.5V
		with I²C™ levels	0.7 VDD	—	—	V	
D042		$\overline{MCLR}$	0.8 VDD	—	—	V	
D043A		OSC1 (HS mode)	0.7 VDD	—	—	V	
D043B		OSC1 (RC mode)	0.9 VDD	—	—	V	**(Note 1)**
	IIL	**Input Leakage Current(2)**					
D060		I/O ports	—	± 5	± 125	nA	VSS ≤ VPIN ≤ VDD, Pin at high-impedance, 85°C
				± 5	± 1000	nA	125°C
D061		$\overline{MCLR}$(3)	—	± 50	± 200	nA	VSS ≤ VPIN ≤ VDD, 85°C
	IPUR	**PORTB Weak Pull-up Current**					
D070*			25 25	100 140	200 300	 μA	VDD = 3.3V, VPIN = VSS VDD = 5.0V, VPIN = VSS
	VOL	**Output Low Voltage(4)**					
D080		I/O ports	—	—	0.6	V	IOL = 8 mA, VDD = 5V IOL = 6 mA, VDD = 3.3V IOL = 1.8 mA, VDD = 1.8V
	VOH	**Output High Voltage(4)**					
D090		I/O ports	VDD - 0.7	—	—	V	IOH = 3.5 mA, VDD = 5V IOH = 3 mA, VDD = 3.3V IOH = 1 mA, VDD = 1.8V
		Capacitive Loading Specs on Output Pins					
D101*	COSC2	OSC2 pin	—	—	15	pF	In XT, HS and LP modes when external clock is used to drive OSC1
D101A*	CIO	All I/O pins	—	—	50	pF	

Legend: TBD = To Be Determined

* 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.

Note 1: In RC oscillator configuration, the OSC1/CLKIN pin is a Schmitt Trigger input. It is not recommended to use an external clock in RC mode.

2: Negative current is defined as current sourced by the pin.

3: The leakage current on the $\overline{MCLR}$ pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

4: Including OSC2 in CLKOUT mode.

23.4 DC Characteristics: PIC16(L)F722A/723A-I/E (Continued)

DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended				
Param No.	**Sym.**	**Characteristic**	**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**
		Program Flash Memory					
D130	EP	Cell Endurance	100	1k	—	E/W	Temperature during programming: 10°C ≤ TA ≤ 40°C
D131		VDD for Read	VMIN	—	—	V	
		Voltage on $\overline{\text{MCLR}}$/VPP during Erase/Program	8.0	—	9.0	V	Temperature during programming: 10°C ≤ TA ≤ 40°C
		VDD for Bulk Erase	2.7	3	—	V	Temperature during programming: 10°C ≤ TA ≤ 40°C
D132	VPEW	VDD for Write or Row Erase	2.7	—	—	V	VMIN = Minimum operating voltage VMAX = Maximum operating voltage
	IPPPGM	Current on $\overline{\text{MCLR}}$/VPP during Erase/Write	—	—	5.0	mA	Temperature during programming: 10°C ≤ TA ≤ 40°C
	IDDPGM	Current on VDD during Erase/ Write	—		5.0	mA	Temperature during programming: 10°C ≤ TA ≤ 40°C
D133	TPEW	Erase/Write cycle time	—		2.8	ms	Temperature during programming: 10°C ≤ TA ≤ 40°C
D134	TRETD	Characteristic Retention	40	—	—	Year	Provided no other specifications are violated
		VCAP Capacitor Charging					
D135		Charging current	—	200	—	µA	
D135A		Source/sink capability when charging complete	—	0.0	—	mA	

Legend: TBD = To Be Determined

* 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.

Note 1: In RC oscillator configuration, the OSC1/CLKIN pin is a Schmitt Trigger input. It is not recommended to use an external clock in RC mode.

2: Negative current is defined as current sourced by the pin.

3: The leakage current on the $\overline{\text{MCLR}}$ pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

4: Including OSC2 in CLKOUT mode.

23.5 Thermal Considerations

Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +125°C					
Param No.	**Sym.**	**Characteristic**	**Typ.**	**Units**	**Conditions**
TH01	θJA	Thermal Resistance Junction to Ambient	60.0	°C/W	28-pin SPDIP package
			69.7	°C/W	28-pin SOIC package
			71.0	°C/W	28-pin SSOP package
			52.5	°C/W	28-pin UQFN 4x4mm package
			30.0	°C/W	28-pin QFN 6x6mm package
TH02	θJC	Thermal Resistance Junction to Case	29.0	°C/W	28-pin SPDIP package
			18.9	°C/W	28-pin SOIC package
			24.0	°C/W	28-pin SSOP package
			16.7	°C/W	28-pin UQFN 4x4mm package
			5.0	°C/W	28-pin QFN 6x6mm package
TH03	TJMAX	Maximum Junction Temperature	150	°C	
TH04	PD	Power Dissipation	—	W	PD = PINTERNAL + PI/O
TH05	PINTERNAL	Internal Power Dissipation	—	W	PINTERNAL = IDD x VDD(1)
TH06	PI/O	I/O Power Dissipation	—	W	PI/O = Σ (IOL * VOL) + Σ (IOH * (VDD - VOH))
TH07	PDER	Derated Power	—	W	PDER = PDMAX (TJ - TA)/θJA(2)

Note 1: IDD is current to run the chip alone without driving any load on the output pins.
2: TA = Ambient Temperature
3: TJ = Junction Temperature

23.6 Timing Parameter Symbology

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

1. TppS2ppS
2. TppS

T			
F	Frequency	T	Time

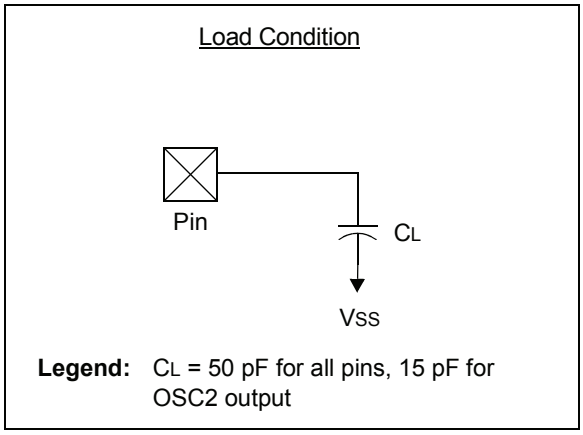
Lowercase letters (pp) and their meanings:

pp			
cc	CCP1	osc	OSC1
ck	CLKOUT	rd	$\overline{RD}$
cs	$\overline{CS}$	rw	$\overline{RD}$ or $\overline{WR}$
di	SDI	sc	SCK
do	SDO	ss	$\overline{SS}$
dt	Data in	t0	T0CKI
io	I/O PORT	t1	T1CKI
mc	$\overline{MCLR}$	wr	$\overline{WR}$

Uppercase letters and their meanings:

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

FIGURE 23-2: LOAD CONDITIONS



Legend: C_L = 50 pF for all pins, 15 pF for OSC2 output

23.7 AC Characteristics: PIC16F722A/723A-I/E

FIGURE 23-3: CLOCK TIMING

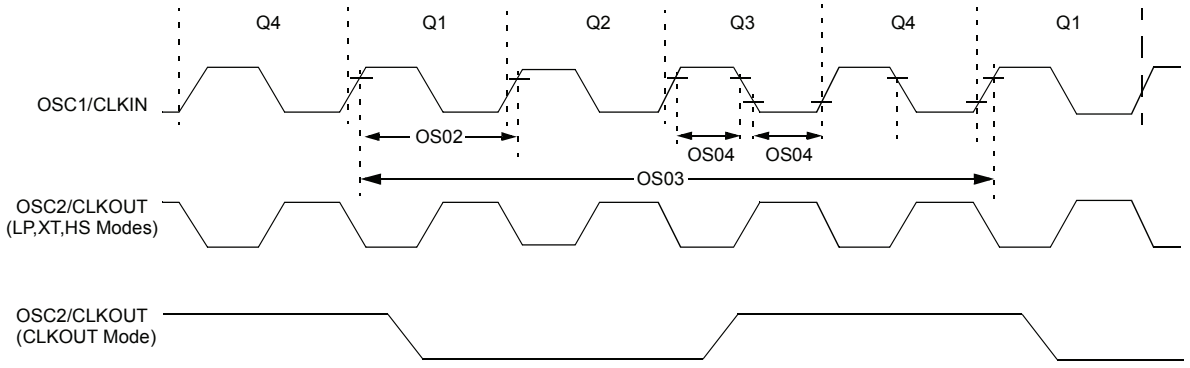
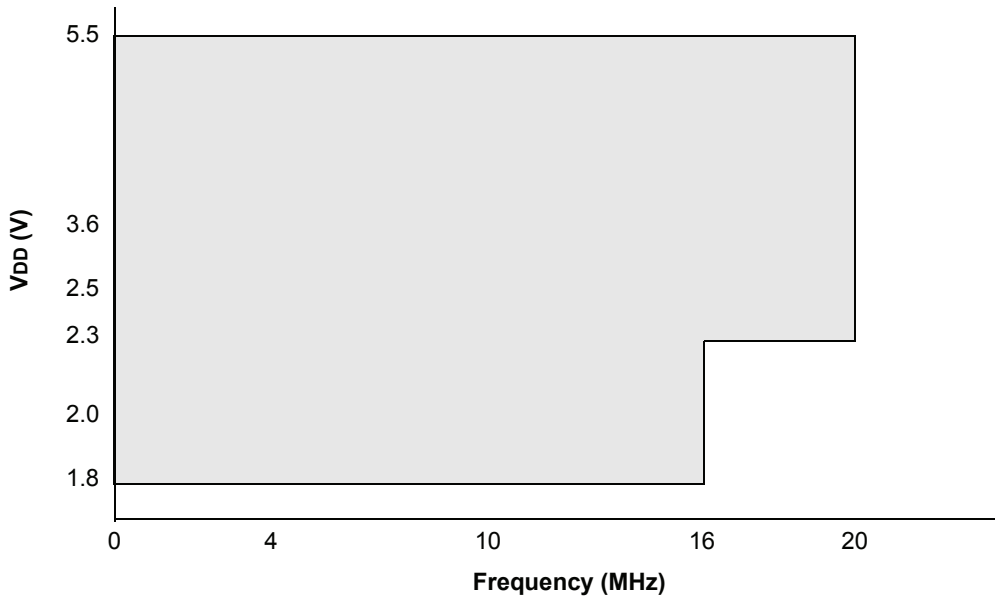
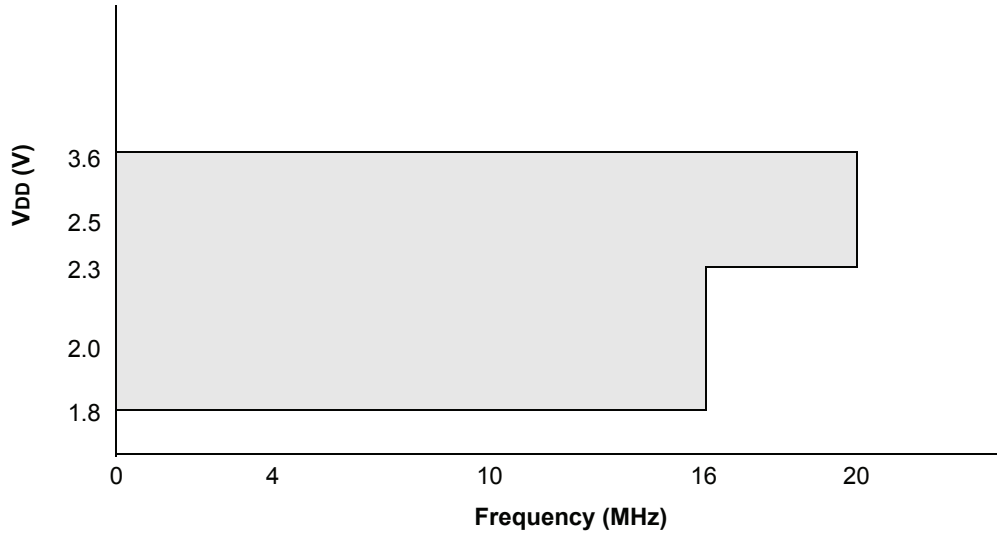


FIGURE 23-4: PIC16F722A/723A VOLTAGE FREQUENCY GRAPH, -40°C ≤ TA ≤ +125°C



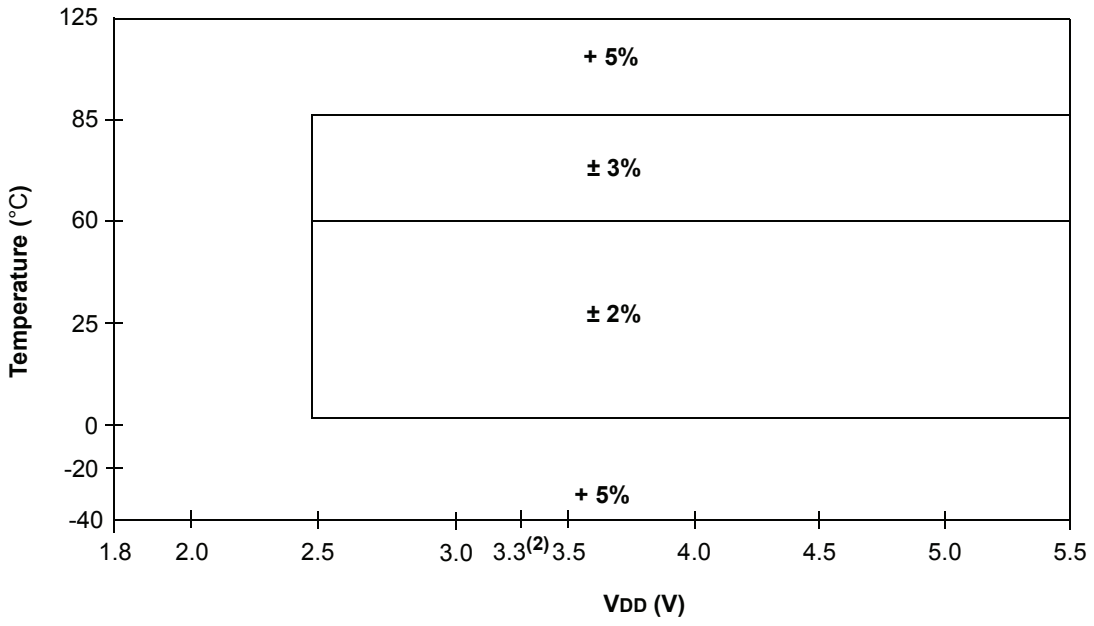
Note 1: The shaded region indicates the permissible combinations of voltage and frequency.
2: Refer to Table 23-1 for each Oscillator mode's supported frequencies.

FIGURE 23-5: PIC16LF722A/723A 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 23-6: HFINTOSC FREQUENCY ACCURACY OVER DEVICE VDD AND TEMPERATURE



Note 1: This chart covers both regulator enabled and regulator disabled states.
2: Regulator Nominal voltage.

TABLE 23-1: CLOCK OSCILLATOR TIMING REQUIREMENTS

Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +125°C							
Param No.	**Sym.**	**Characteristic**	**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**
OS01	FOSC	External CLKIN Frequency(1)	DC	—	37	kHz	LP Oscillator mode
			DC	—	4	MHz	XT Oscillator mode
			DC	—	20	MHz	HS Oscillator mode
			DC	—	20	MHz	EC Oscillator mode
		Oscillator Frequency(1)	—	32.768	—	kHz	LP Oscillator mode
			0.1	—	4	MHz	XT Oscillator mode
			1	—	20	MHz	HS Oscillator mode, VDD ≥ 2.7V
			DC	—	4	MHz	RC Oscillator mode
OS02	TOSC	External CLKIN Period(1)	27	—	∞	µs	LP Oscillator mode
			250	—	∞	ns	XT Oscillator mode
			50	—	∞	ns	HS Oscillator mode
			50	—	∞	ns	EC Oscillator mode
		Oscillator Period(1)	—	30.5	—	µs	LP Oscillator mode
			250	—	10,000	ns	XT Oscillator mode
			50	—	1,000	ns	HS Oscillator mode, VDD ≥ 2.7V
			250	—	—	ns	RC Oscillator mode
OS03	TCY	Instruction Cycle Time(1)	200	TCY	DC	ns	TCY = 4/FOSC
OS04*	TosH, TosL	External CLKIN High, External CLKIN Low	2	—	—	µs	LP oscillator
			100	—	—	ns	XT oscillator
			20	—	—	ns	HS oscillator
OS05*	TosR, TosF	External CLKIN Rise, External CLKIN Fall	0	—	∞	ns	LP oscillator
			0	—	∞	ns	XT oscillator
			0	—	∞	ns	HS oscillator

* 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.

Note 1: Instruction cycle period (TCY) equals four times the input oscillator time base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at "min" values with an external clock applied to OSC1 pin. When an external clock input is used, the "max" cycle time limit is "DC" (no clock) for all devices.

TABLE 23-2: OSCILLATOR PARAMETERS

Standard Operating Conditions (unless otherwise stated) Operating Temperature -40°C ≤ TA ≤ +125°C								
Param No.	**Sym.**	**Characteristic**	**Freq. Tolerance**	**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**
OS08	HFOSC	Internal Calibrated HFINTOSC Frequency(2)	±2%	—	16.0	—	MHz	0°C ≤ TA ≤ +85°C, VDD ≥ 2.5V
			±5%	—	16.0	—	MHz	-40°C ≤ TA ≤ +125°C
OS08A	MFOSC	Internal Calibrated MFINTOSC Frequency(2)	±2%	—	500	—	kHz	0°C ≤ TA ≤ +85°C VDD ≥ 2.5V
			±5%	—	500	10	kHz	-40°C ≤ TA ≤ +125°C
OS10*	TIOSC ST	HFINTOSC Wake-up from Sleep Start-up Time	—	—	5	8	μs	
		MFINTOSC Wake-up from Sleep Start-up Time	—	—	20	30	μs	

* 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.

Note 1: Instruction cycle period (TCY) equals four times the input oscillator time base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at "min" values with an external clock applied to the OSC1 pin. When an external clock input is used, the "max" cycle time limit is "DC" (no clock) for all devices.

2: To ensure these oscillator frequency tolerances, VDD and VSS must be capacitively decoupled as close to the device as possible. 0.1 μF and 0.01 μF values in parallel are recommended.

3: By design.

FIGURE 23-7: CLKOUT AND I/O TIMING

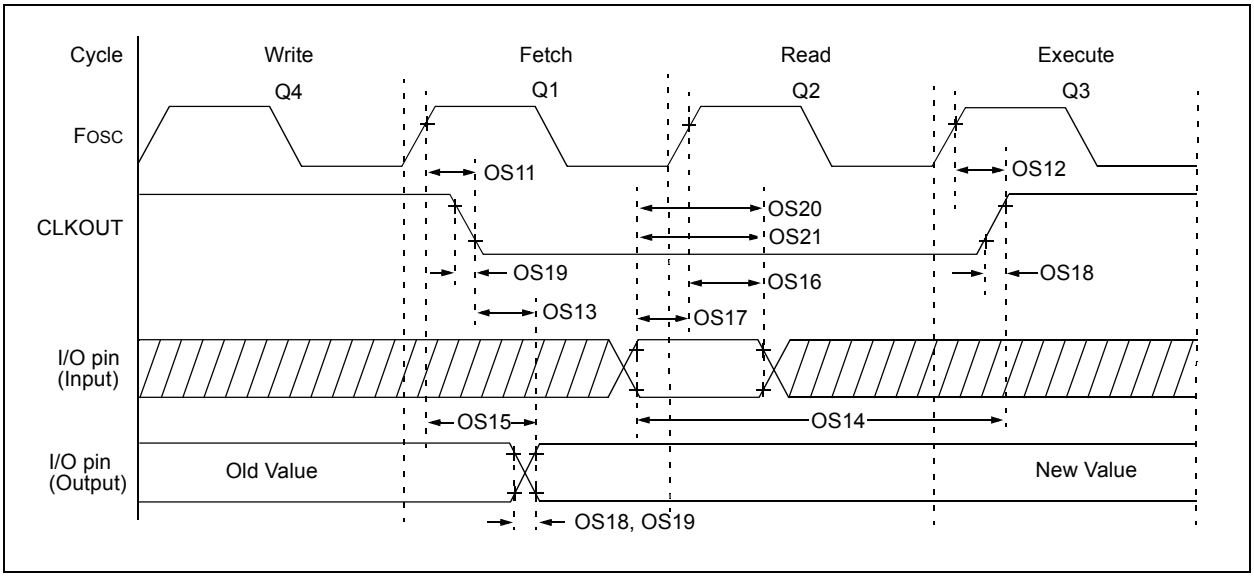


TABLE 23-3: CLKOUT AND I/O TIMING PARAMETERS

Standard Operating Conditions (unless otherwise stated) Operating Temperature -40°C ≤ TA ≤ +125°C							
Param No.	**Sym.**	**Characteristic**	**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**
OS11	TosH2ckL	Fosc↑ to CLKOUT↓ (1)	—	—	70	ns	VDD = 3.3-5.0V
OS12	TosH2ckH	Fosc↑ to CLKOUT↑ (1)	—	—	72	ns	VDD = 3.3-5.0V
OS13	TckL2ioV	CLKOUT↓ to Port out valid(1)	—	—	20	ns	
OS14	TioV2ckH	Port input valid before CLKOUT↑(1)	TOSC + 200 ns	—	—	ns	
OS15	TosH2ioV	Fosc↑ (Q1 cycle) to Port out valid	—	50	70*	ns	VDD = 3.3-5.0V
OS16	TosH2ioI	Fosc↑ (Q2 cycle) to Port input invalid (I/O in hold time)	50	—	—	ns	VDD = 3.3-5.0V
OS17	TioV2osH	Port input valid to Fosc↑ (Q2 cycle) (I/O in setup time)	20	—	—	ns	
OS18	TioR	Port output rise time(2)	— —	40 15	72 32	ns	VDD = 2.0V VDD = 3.3-5.0V
OS19	TioF	Port output fall time(2)	— —	28 15	55 30	ns	VDD = 2.0V VDD = 3.3-5.0V
OS20*	Tinp	INT pin input high or low time	25	—	—	ns	
OS21*	Trbp	PORTB interrupt-on-change new input level time	TCY	—	—	ns	

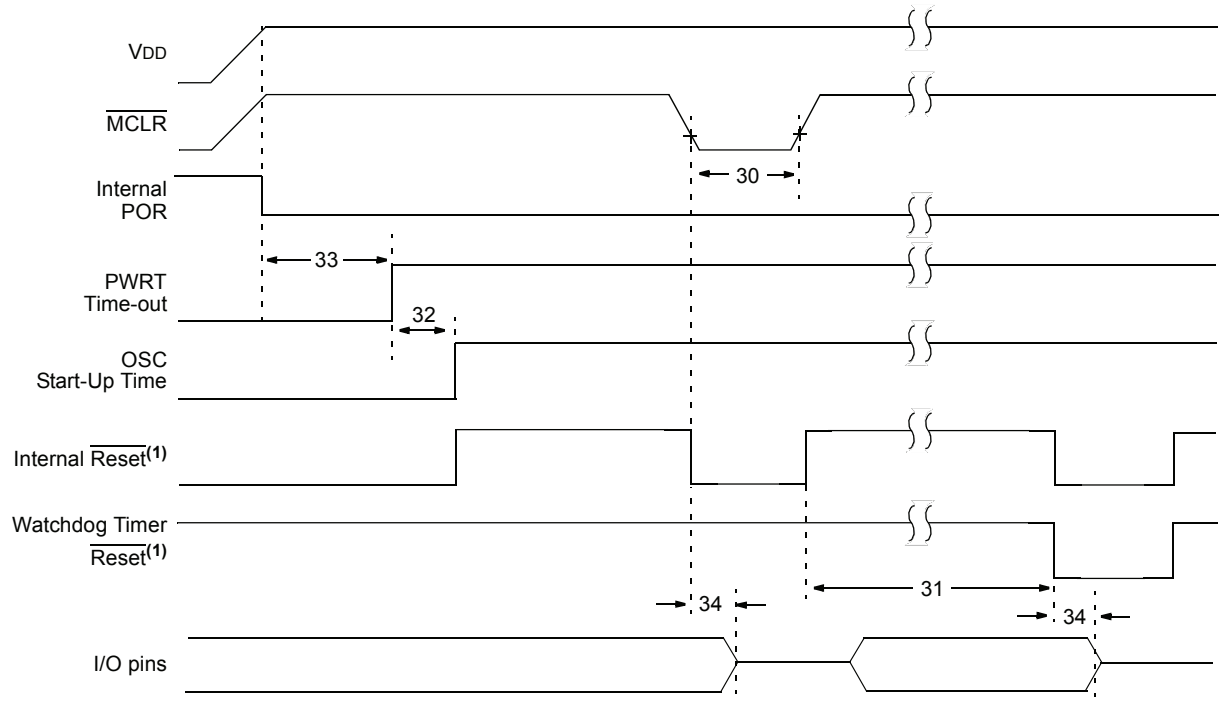
* These parameters are characterized but not tested.

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated.

Note 1: Measurements are taken in RC mode where CLKOUT output is 4 x TOSC.

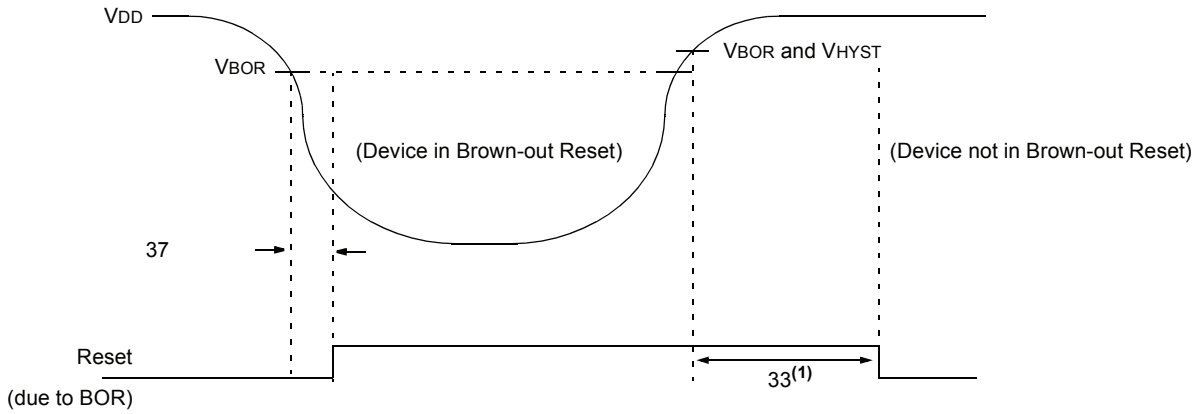
2: Includes OSC2 in CLKOUT mode.

FIGURE 23-8: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING



Note 1: Asserted low.

FIGURE 23-9: BROWN-OUT RESET TIMING AND CHARACTERISTICS



Note 1: 64 ms delay only if $\overline{\text{PWRTE}}$ bit in the Configuration Word register is programmed to '0'. 2 ms delay if $\overline{\text{PWRTE}}$ = 0 and VREGEN = 1.

TABLE 23-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER, POWER-UP TIMER, AND BROWN-OUT RESET PARAMETERS

Standard Operating Conditions (unless otherwise stated) Operating Temperature -40°C ≤ TA ≤ +125°C							
Param No.	**Sym.**	**Characteristic**	**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**
30	TMCL	$\overline{MCLR}$ Pulse Width (low)	2 5	— —	— —	μs μs	VDD = 3.3-5V, -40°C to +85°C VDD = 3.3-5V
31	TWDTLP	Low Power Watchdog Timer Time-out Period (No Prescaler)	10	18	27	ms	VDD = 3.3V-5V
32	TOST	Oscillator Start-up Timer Period(1), (2)	—	1024	—	Tosc	**(Note 3)**
33*	TPWRT	Power-up Timer Period, $\overline{PWRTE}$ = 0	40	65	140	ms	
34*	TIOZ	I/O high-impedance from $\overline{MCLR}$ Low or Watchdog Timer Reset	—	—	2.0	μs	
35	VBOR	Brown-out Reset Voltage	2.38 1.80	2.5 1.9	2.73 2.11	V	BORV=2.5V BORV=1.9V
36*	VHYST	Brown-out Reset Hysteresis	0	25	50	mV	-40°C to +85°C
37*	TBORDC	Brown-out Reset DC Response Time	1	3	5 10	μs	VDD ≤ VBOR, -40°C to +85°C VDD ≤ VBOR

* 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.

Note 1: Instruction cycle period (TCY) equals four times the input oscillator time base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at "min" values with an external clock applied to the OSC1 pin. When an external clock input is used, the "max" cycle time limit is "DC" (no clock) for all devices.

2: By design.

3: Period of the slower clock.

4: To ensure these voltage tolerances, VDD and VSS must be capacitively decoupled as close to the device as possible. 0.1 μF and 0.01 μF values in parallel are recommended.

FIGURE 23-10: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS

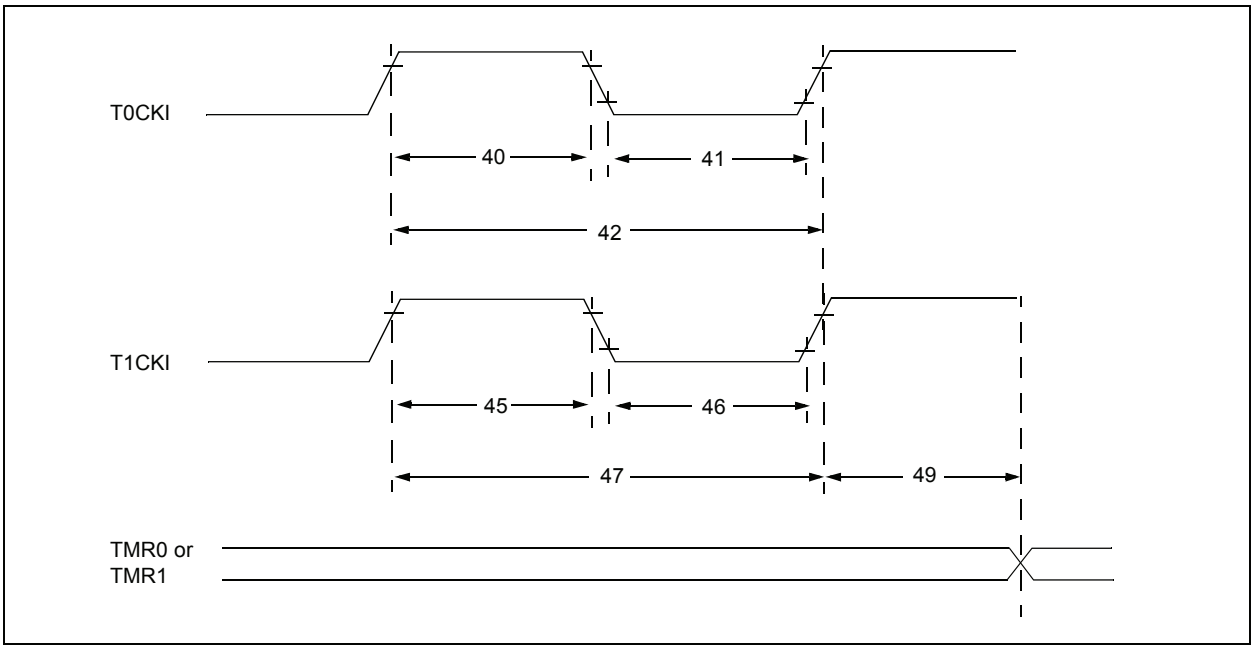


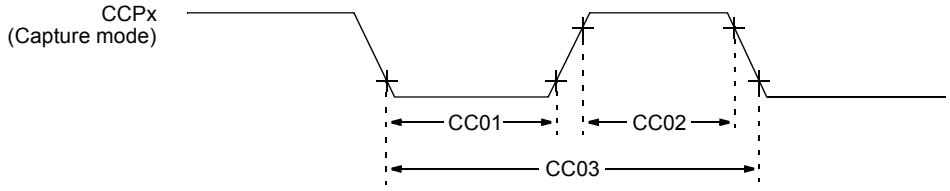
TABLE 23-5: TIMER0 AND TIMER1 EXTERNAL CLOCK REQUIREMENTS

Standard Operating Conditions (unless otherwise stated) Operating Temperature -40°C ≤ TA ≤ +125°C								
Param No.	**Sym.**	**Characteristic**		**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**
40*	TT0H	T0CKI High Pulse Width	No Prescaler	0.5 TCY + 20	—	—	ns	
			With Prescaler	10	—	—	ns	
41*	TT0L	T0CKI Low Pulse Width	No Prescaler	0.5 TCY + 20	—	—	ns	
			With Prescaler	10	—	—	ns	
42*	TT0P	T0CKI Period		Greater of: 20 or $\frac{T_{CY} + 40}{N}$	—	—	ns	N = prescale value (2, 4, ..., 256)
45*	TT1H	T1CKI High Time	Synchronous, No Prescaler	0.5 TCY + 20	—	—	ns	
			Synchronous, with Prescaler	15	—	—	ns	
			Asynchronous	30	—	—	ns	
46*	TT1L	T1CKI Low Time	Synchronous, No Prescaler	0.5 TCY + 20	—	—	ns	
			Synchronous, with Prescaler	15	—	—	ns	
			Asynchronous	30	—	—	ns	
47*	TT1P	T1CKI Input Period	Synchronous	Greater of: 30 or $\frac{T_{CY} + 40}{N}$	—	—	ns	N = prescale value (1, 2, 4, 8)
			Asynchronous	60	—	—	ns	
48	FT1	Timer1 Oscillator Input Frequency Range (oscillator enabled by setting bit T1OSCEN)		32.4	32.768	33.1	kHz	
49*	TCKEZTMR1	Delay from External Clock Edge to Timer Increment		2 TOSC	—	7 TOSC	—	Timers in Sync mode

* 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 23-11: CAPTURE/COMPARE/PWM TIMINGS (CCP)



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

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

Standard Operating Conditions (unless otherwise stated) Operating Temperature -40°C ≤ TA ≤ +125°C								
Param No.	**Sym.**	**Characteristic**		**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**
CC01*	TccL	CCPx Input Low Time	No Prescaler	0.5TCY + 20	—	—	ns	
			With Prescaler	20	—	—	ns	
CC02*	TccH	CCPx Input High Time	No Prescaler	0.5TCY + 20	—	—	ns	
			With Prescaler	20	—	—	ns	
CC03*	TccP	CCPx 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 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

TABLE 23-7: PIC16F722A/723A A/D CONVERTER (ADC) CHARACTERISTICS:

Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +125°C							
Param No.	**Sym.**	**Characteristic**	**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**
AD01	NR	Resolution	—	—	8	bit	
AD02	EIL	Integral Error	—	—	±1.7	LSb	VREF = 3.0V
AD03	EDL	Differential Error	—	—	±1	LSb	No missing codes VREF = 3.0V
AD04	EOFF	Offset Error	—	—	±2.2	LSb	VREF = 3.0V
AD05	EGN	Gain Error	—	—	±1.5	LSb	VREF = 3.0V
AD06	VREF	Reference Voltage(3)	1.8	—	VDD	V	
AD07	VAIN	Full-Scale Range	VSS	—	VREF	V	
AD08	ZAIN	Recommended Impedance of Analog Voltage Source	—	—	50	kΩ	Can go higher if external 0.01µF capacitor is present on input pin.

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

Note 1: Total Absolute Error includes integral, differential, offset and gain errors.

2: The A/D conversion result never decreases with an increase in the input voltage and has no missing codes.

3: When ADC is off, it will not consume any current other than leakage current. The power-down current specification includes any such leakage from the ADC module.

TABLE 23-8: PIC16F722A/723A A/D CONVERSION REQUIREMENTS

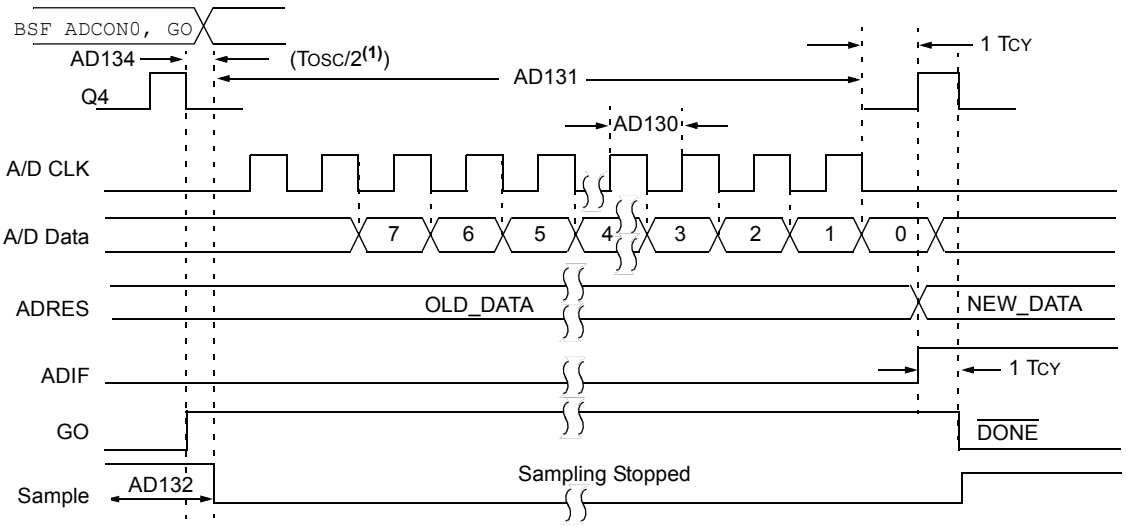
Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +125°C							
Param No.	**Sym.**	**Characteristic**	**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**
AD130*	TAD	A/D Clock Period	1.0	—	9.0	µs	TOSC-based
		A/D Internal RC Oscillator Period	1.0	2.0	6.0	µs	ADCS<1:0> = 11 (ADRC mode)
AD131	TCNV	Conversion Time (not including Acquisition Time)(1)	—	10.5	—	TAD	Set GO/$\overline{\text{DONE}}$ bit to conversion complete
AD132*	TACQ	Acquisition Time	—	1.0	—	µs	

* 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.

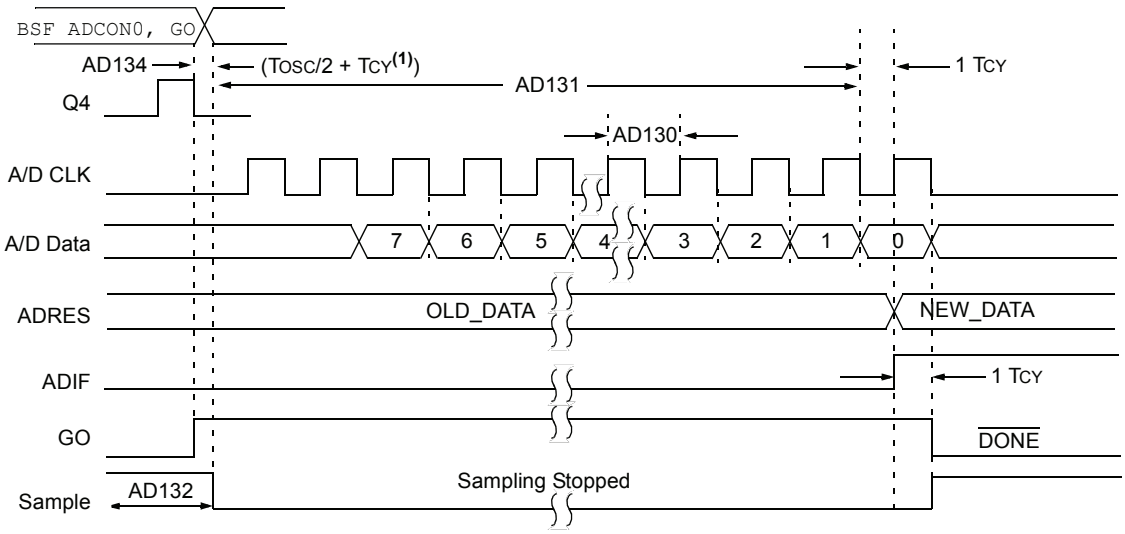
Note 1: The ADRES register may be read on the following TCY cycle.

FIGURE 23-12: PIC16F722A/723A A/D CONVERSION TIMING (NORMAL MODE)



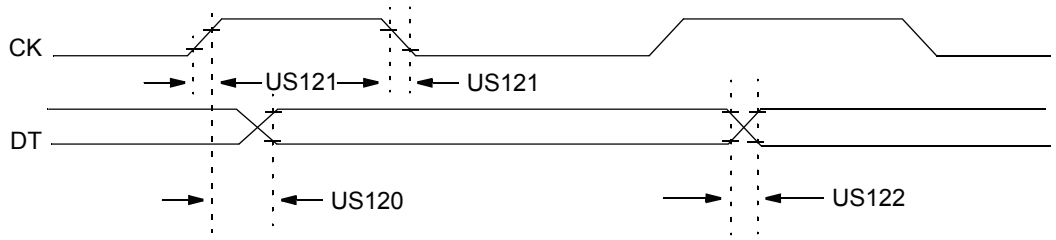
Note 1: If the A/D clock source is selected as RC, a time of TCY is added before the A/D clock starts. This allows the SLEEP instruction to be executed.

FIGURE 23-13: PIC16F722A/723A 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

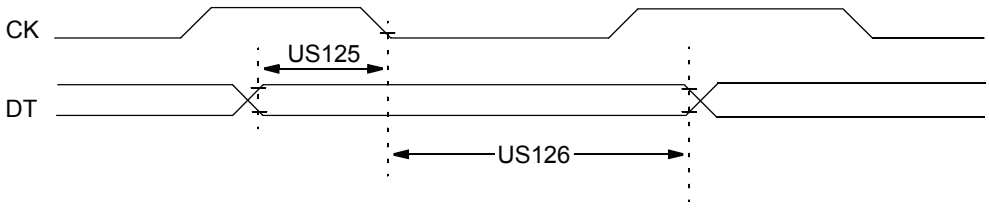


Note: Refer to Figure 23-2 for load conditions.

TABLE 23-9: USART SYNCHRONOUS TRANSMISSION REQUIREMENTS

Standard Operating Conditions (unless otherwise stated) Operating Temperature -40°C ≤ TA ≤ +125°C							
Param. No.	**Symbol**	**Characteristic**		**Min.**	**Max.**	**Units**	**Conditions**
US120	TCKH2DTV	SYNC XMIT (Master and Slave) 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

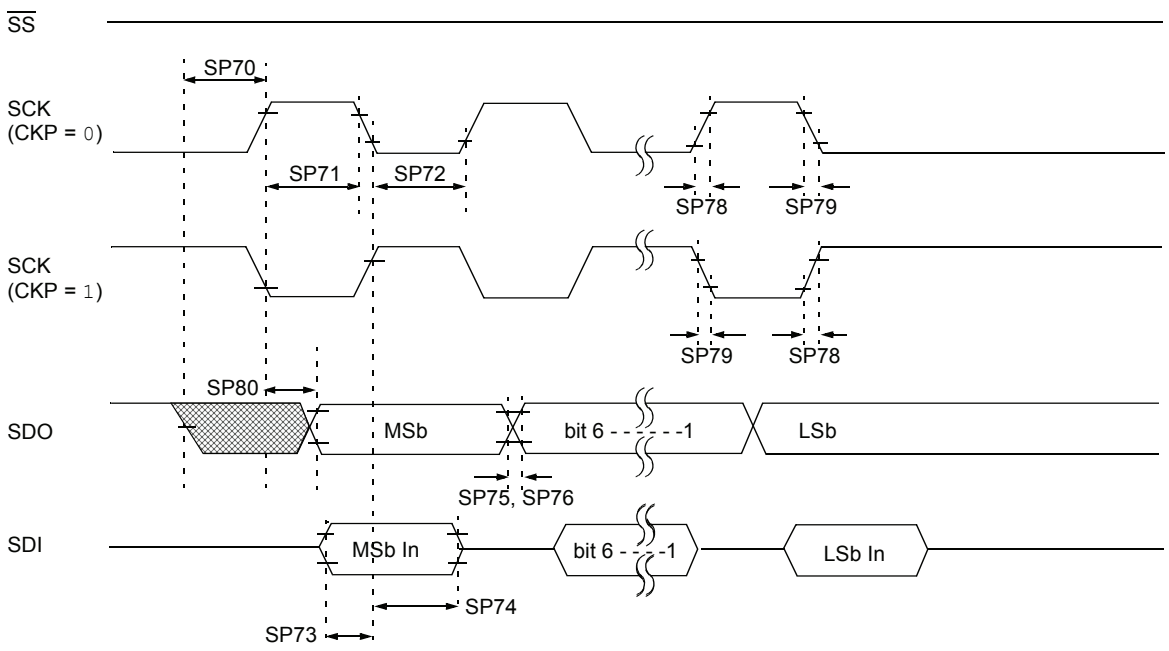


Note: Refer to Figure 23-2 for load conditions.

TABLE 23-10: USART SYNCHRONOUS RECEIVE REQUIREMENTS

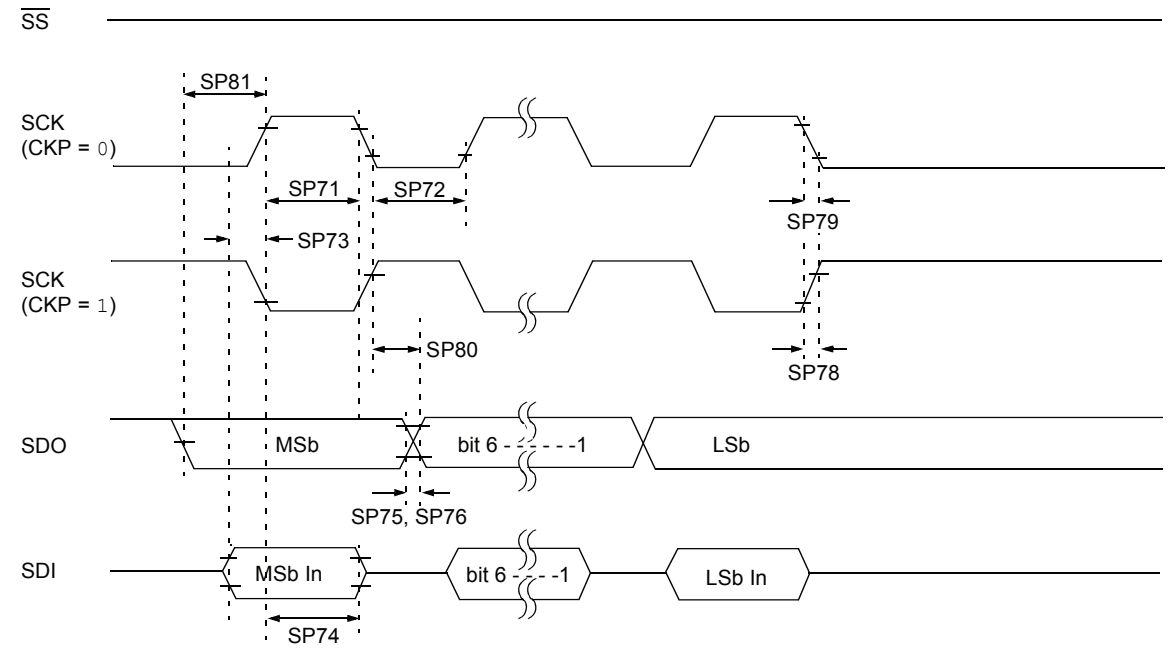
Standard Operating Conditions (unless otherwise stated) Operating Temperature -40°C ≤ TA ≤ +125°C						
Param. No.	**Symbol**	**Characteristic**	**Min.**	**Max.**	**Units**	**Conditions**
US125	TDTV2CKL	SYNC RCV (Master and Slave) Data-hold before CK ↓ (DT hold time)	10	—	ns	
US126	TCKL2DTL	Data-hold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

FIGURE 23-16: SPI MASTER MODE TIMING (CKE = 0, SMP = 0)



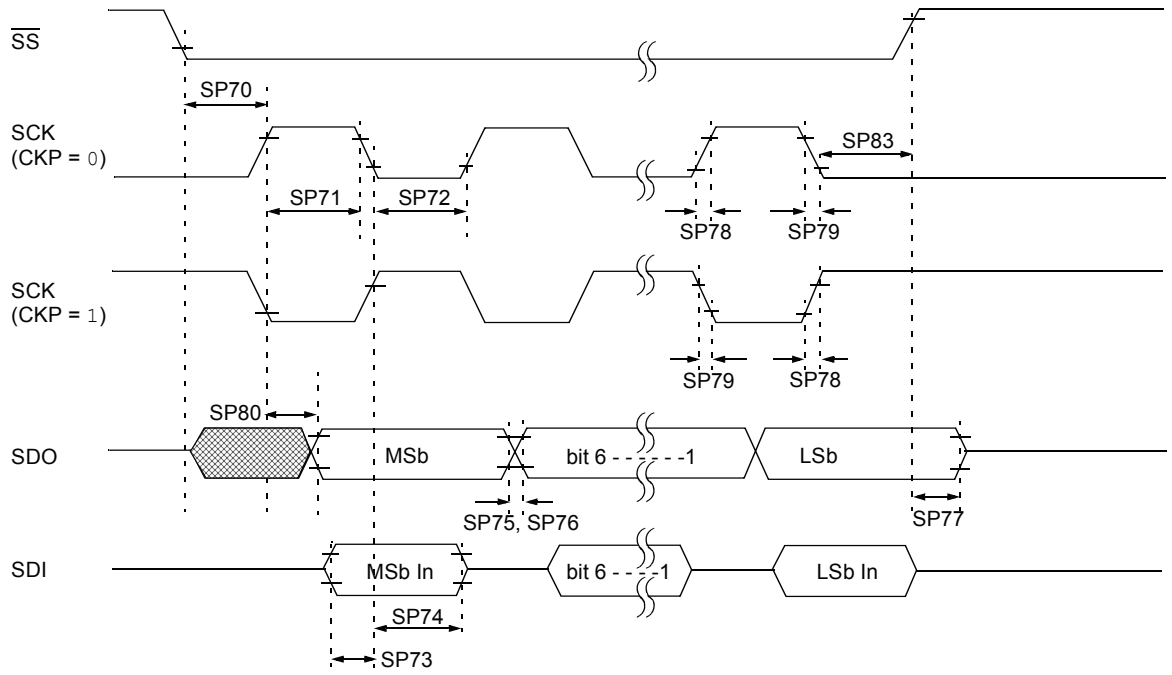
Note: Refer to Figure 23-2 for load conditions.

FIGURE 23-17: SPI MASTER MODE TIMING (CKE = 1, SMP = 1)



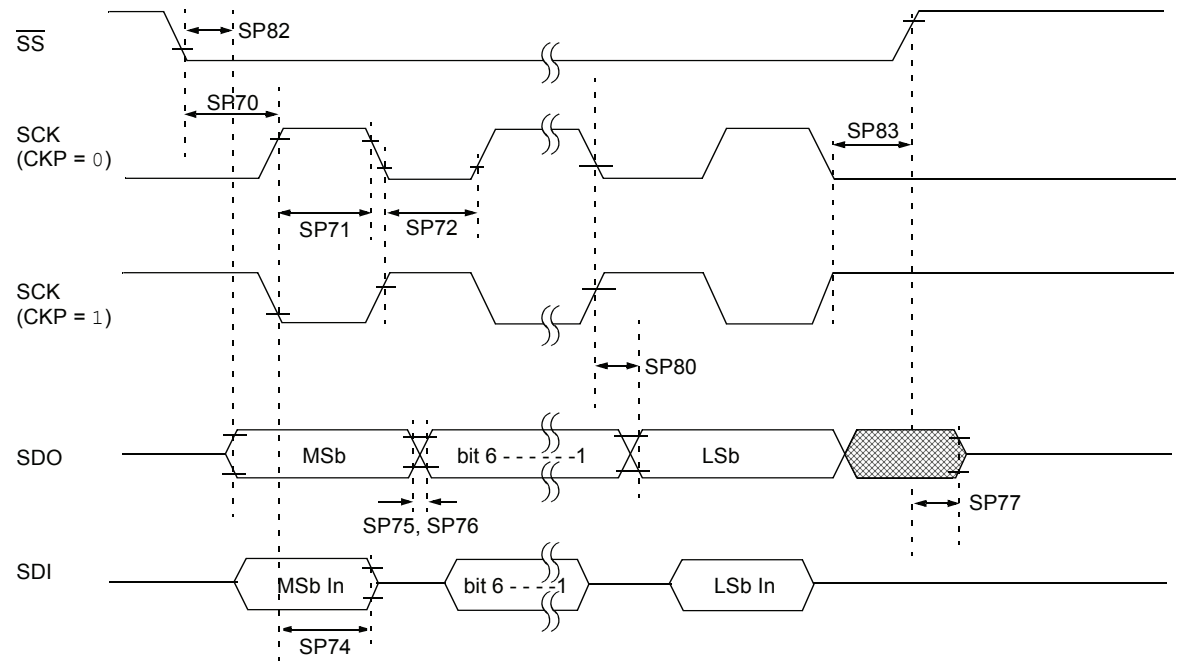
Note: Refer to Figure 23-2 for load conditions.

FIGURE 23-18: SPI SLAVE MODE TIMING (CKE = 0)



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

FIGURE 23-19: SPI SLAVE MODE TIMING (CKE = 1)



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

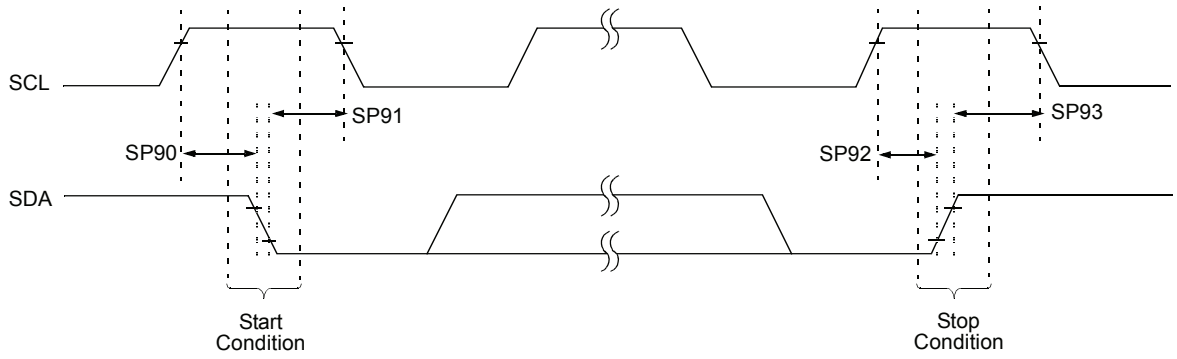
TABLE 23-11: SPI MODE REQUIREMENTS

Param No.	Symbol	Characteristic		Min.	Typ†	Max.	Units	Conditions
SP70*	TssL2scH, TssL2scL	$\overline{SS}$↓ 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	$\overline{SS}$↑ 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 $\overline{SS}$↓ edge		—	—	50	ns	
SP83*	TscH2ssH, TscL2ssH	$\overline{SS}$ ↑ after SCK edge		1.5 Tcy + 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: I²C™ BUS START/STOP BITS TIMING



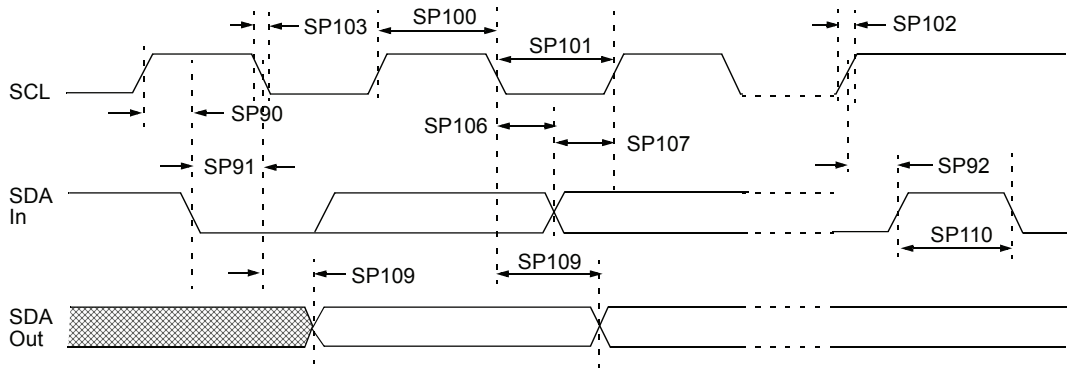
Note: Refer to Figure 23-2 for load conditions.

TABLE 23-12: I²C™ BUS START/STOP BITS REQUIREMENTS

Param No.	Symbol	Characteristic		Min.	Typ	Max.	Units	Conditions
SP90*	TSU:STA	Start condition Setup time	100 kHz mode	4700	—	—	ns	Only relevant for Repeated Start condition
			400 kHz mode	600	—	—		
SP91*	THD:STA	Start condition Hold time	100 kHz mode	4000	—	—	ns	After this period, the first clock pulse is generated
			400 kHz mode	600	—	—		
SP92*	TSU:STO	Stop condition Setup time	100 kHz mode	4700	—	—	ns	
			400 kHz mode	600	—	—		
SP93	THD:STO	Stop condition Hold time	100 kHz mode	4000	—	—	ns	
			400 kHz mode	600	—	—		

* These parameters are characterized but not tested.

FIGURE 23-21: I²C™ BUS DATA TIMING



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

TABLE 23-13: I²C™ BUS DATA REQUIREMENTS

Param. No.	Symbol	Characteristic		Min.	Max.	Units	Conditions
SP100*	THIGH	Clock high time	100 kHz mode	4.0	—	μs	Device must operate at a minimum of 1.5 MHz
			400 kHz mode	0.6	—	μs	Device must operate at a minimum of 10 MHz
			SSP Module	1.5TCY	—		
SP101*	TLOW	Clock low time	100 kHz mode	4.7	—	μs	Device must operate at a minimum of 1.5 MHz
			400 kHz mode	1.3	—	μs	Device must operate at a minimum of 10 MHz
			SSP Module	1.5TCY	—		
SP102*	TR	SDA and SCL rise time	100 kHz mode	—	1000	ns	
			400 kHz mode	20 + 0.1CB	300	ns	CB is specified to be from 10-400 pF
SP103*	TF	SDA and SCL fall time	100 kHz mode	—	250	ns	
			400 kHz mode	20 + 0.1CB	250	ns	CB is specified to be from 10-400 pF
SP106*	THD:DAT	Data input hold time	100 kHz mode	0	—	ns	
			400 kHz mode	0	0.9	μs	
SP107*	TSU:DAT	Data input setup time	100 kHz mode	250	—	ns	**(Note 2)**
			400 kHz mode	100	—	ns	
SP109*	TAA	Output valid from clock	100 kHz mode	—	3500	ns	**(Note 1)**
			400 kHz mode	—	—	ns	
SP110*	TBUF	Bus free time	100 kHz mode	4.7	—	μs	Time the bus must be free before a new transmission can start
			400 kHz mode	1.3	—	μs	
SP111	CB	Bus capacitive loading		—	400	pF	

* These parameters are characterized but not tested.

Note 1: As a transmitter, the device must provide this internal minimum delay time to bridge the undefined region (min. 300 ns) of the falling edge of SCL to avoid unintended generation of Start or Stop conditions.

2: A Fast mode (400 kHz) I²C™ bus device can be used in a Standard mode (100 kHz) I²C bus system, but the requirement TSU:DAT ≥ 250 ns must then be met. This will automatically be the case if the device does not stretch the low period of the SCL signal. If such a device does stretch the low period of the SCL signal, it must output the next data bit to the SDA line TR max. + TSU:DAT = 1000 + 250 = 1250 ns (according to the Standard mode I²C bus specification), before the SCL line is released.

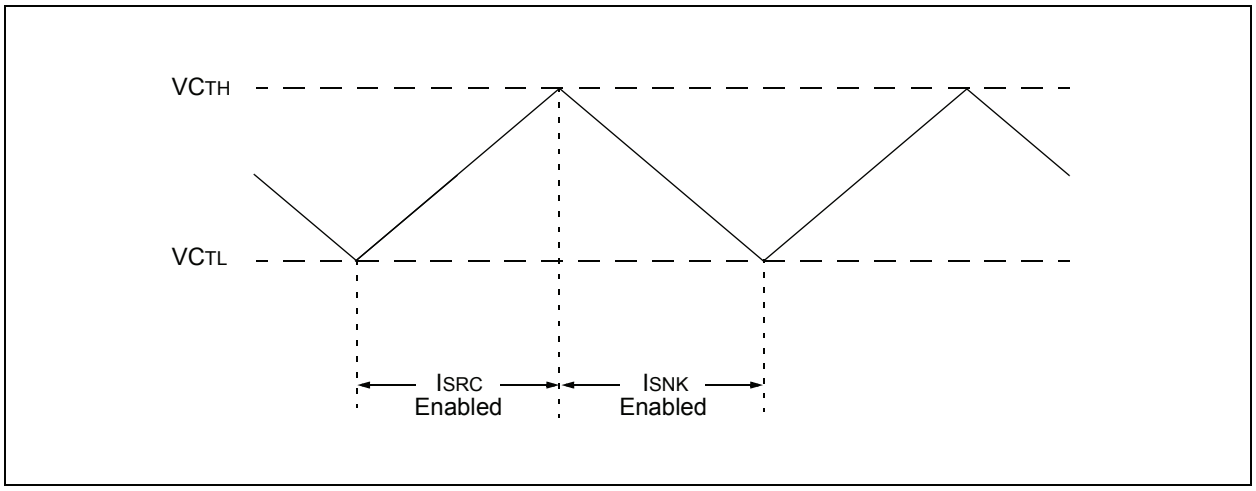
TABLE 23-14: CAP SENSE OSCILLATOR SPECIFICATIONS

Param. No.	Symbol	Characteristic		Min.	Typ†	Max.	Units	Conditions
CS01	ISRC	Current Source	High	—	-5.8	-6	μA	-40, -85°C
			Medium	—	-1.1	-3.2	μA	
			Low	—	-0.2	-0.9	μA	
CS02	ISNK	Current Sink	High	—	6.6	6	μA	-40, -85°C
			Medium	—	1.3	3.2	μA	
			Low	—	0.24	0.9	μA	
CS03	VCHYST	Cap Hysteresis	High	—	525	—	mV	VCTH-VCTL
			Medium	—	375	—	mV	
			Low	—	280	—	mV	

* 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 23-22: CAP SENSE OSCILLATOR



24.0 DC AND AC CHARACTERISTICS GRAPHS AND CHARTS

Note: The graphs and tables provided following this note are a statistical summary based on a limited number of samples and are provided for informational purposes only. The performance characteristics listed herein are not tested or guaranteed. In some graphs or tables, the data presented may be outside the specified operating range (e.g., outside specified power supply range) and therefore, outside the warranted range.

"Typical" represents the mean of the distribution at 25°C. "Maximum" or "minimum" represents (mean + 3σ) or (mean - 3σ) respectively, where σ is a standard deviation, over the whole temperature range.

FIGURE 24-1: PIC16F722A/723A MAXIMUM IDD vs. FOSC OVER VDD, EC MODE, VCAP = 0.1µF

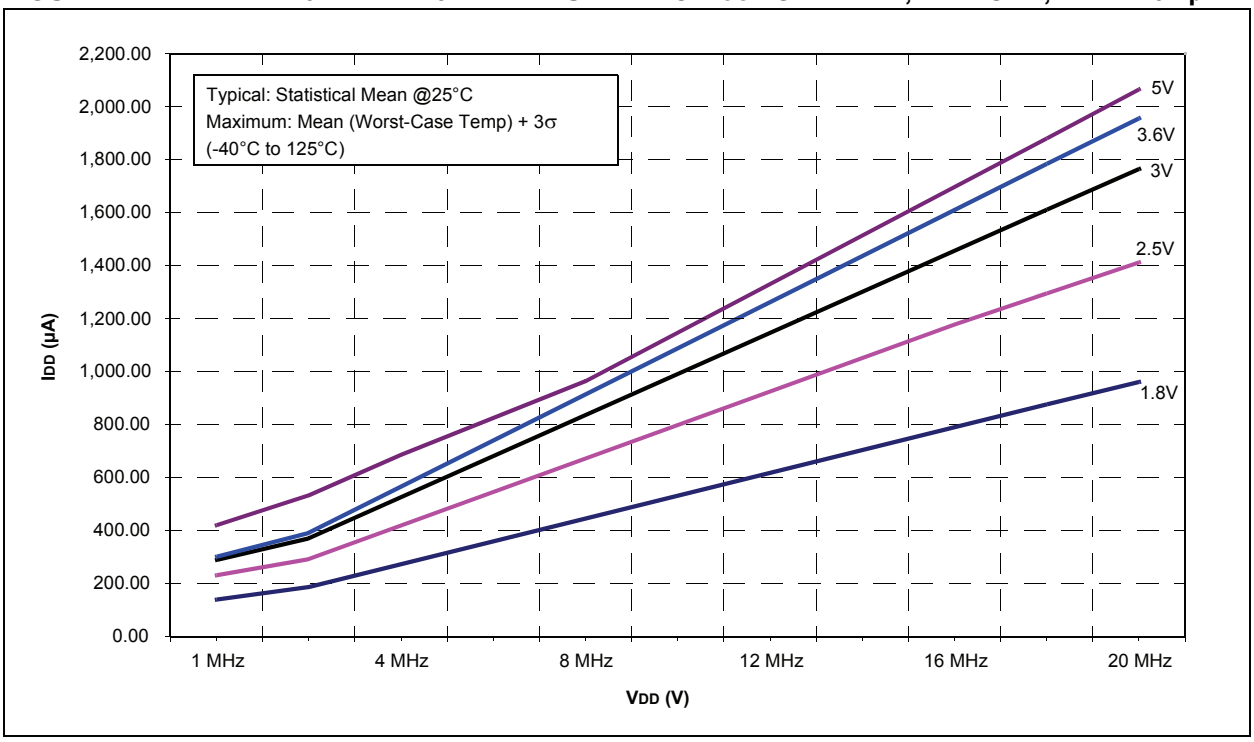


FIGURE 24-2: PIC16LF722A/723A MAXIMUM IDD vs. FOSC OVER VDD, EC MODE

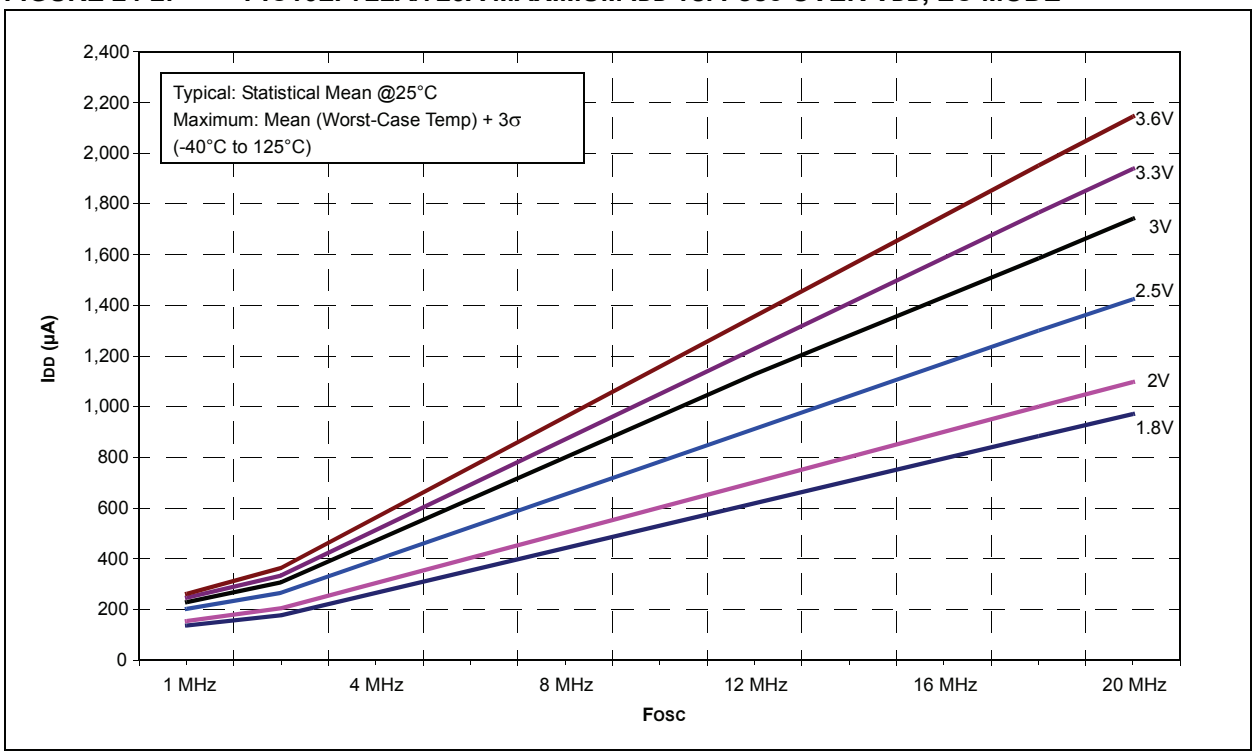


FIGURE 24-3: PIC16F722A/723A TYPICAL IDD vs. FOSC OVER VDD, EC MODE, VCAP = 0.1µF

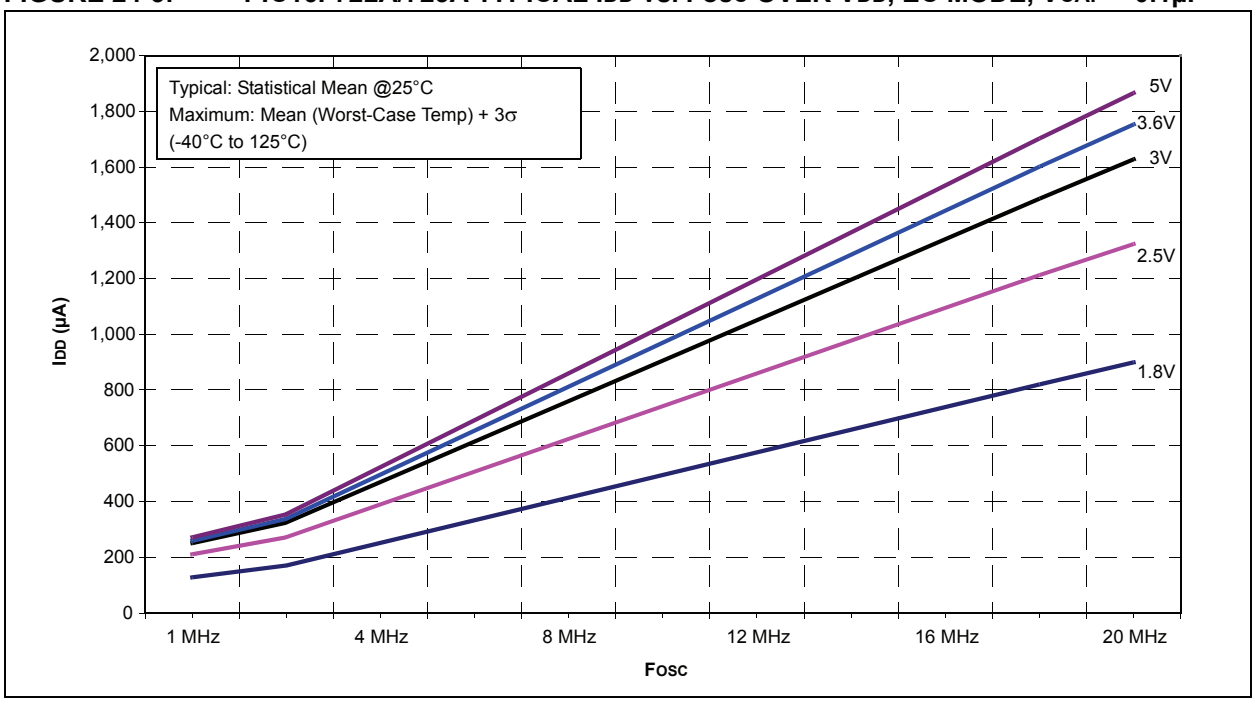


FIGURE 24-4: PIC16LF722A/723A TYPICAL IDD vs. FOSC OVER VDD, EC MODE

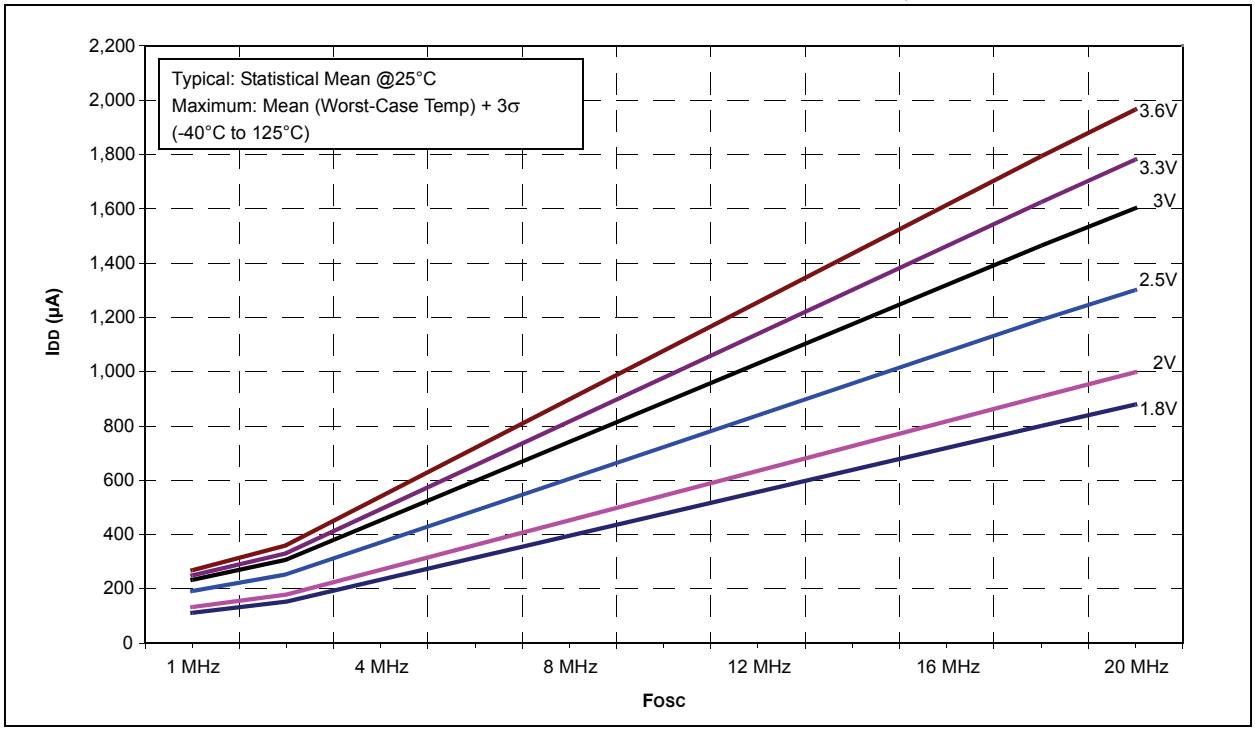


FIGURE 24-5: PIC16F722A/723A MAXIMUM IDD vs. VDD OVER FOSC, EXTRC MODE, VCAP = 0.1µF

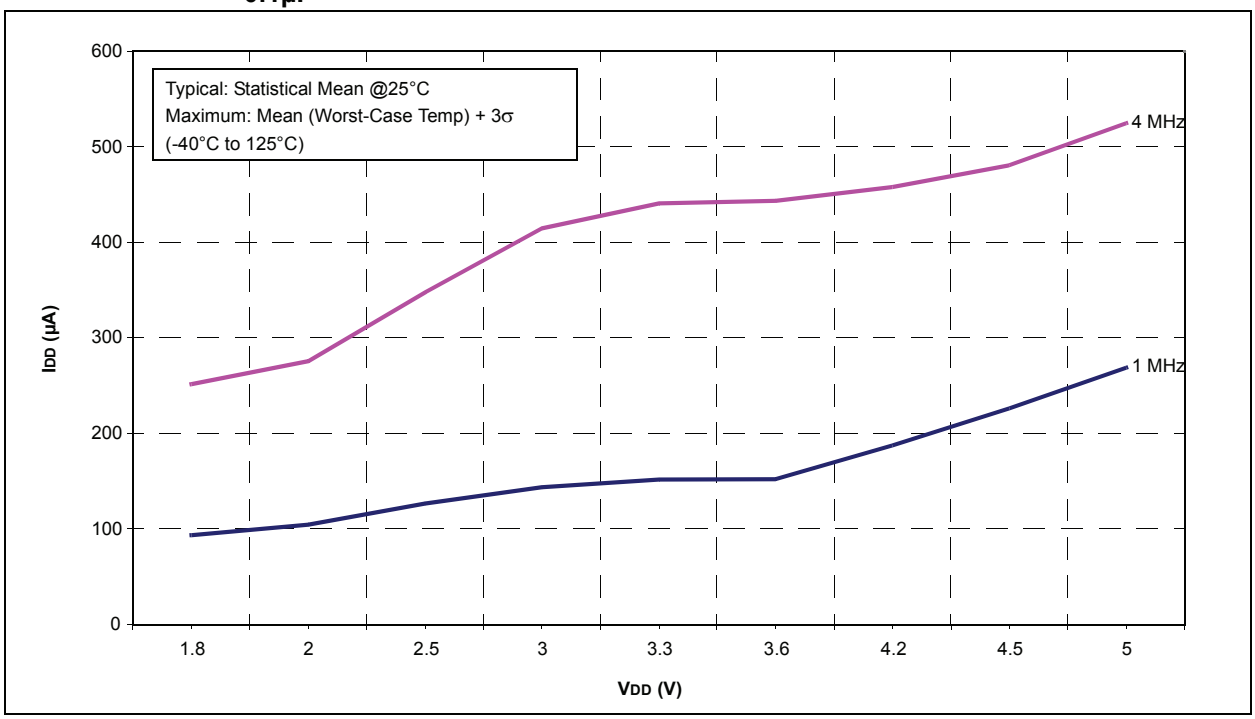


FIGURE 24-6: PIC16LF722A/723A MAXIMUM IDD vs. VDD OVER FOSC, EXTRC MODE

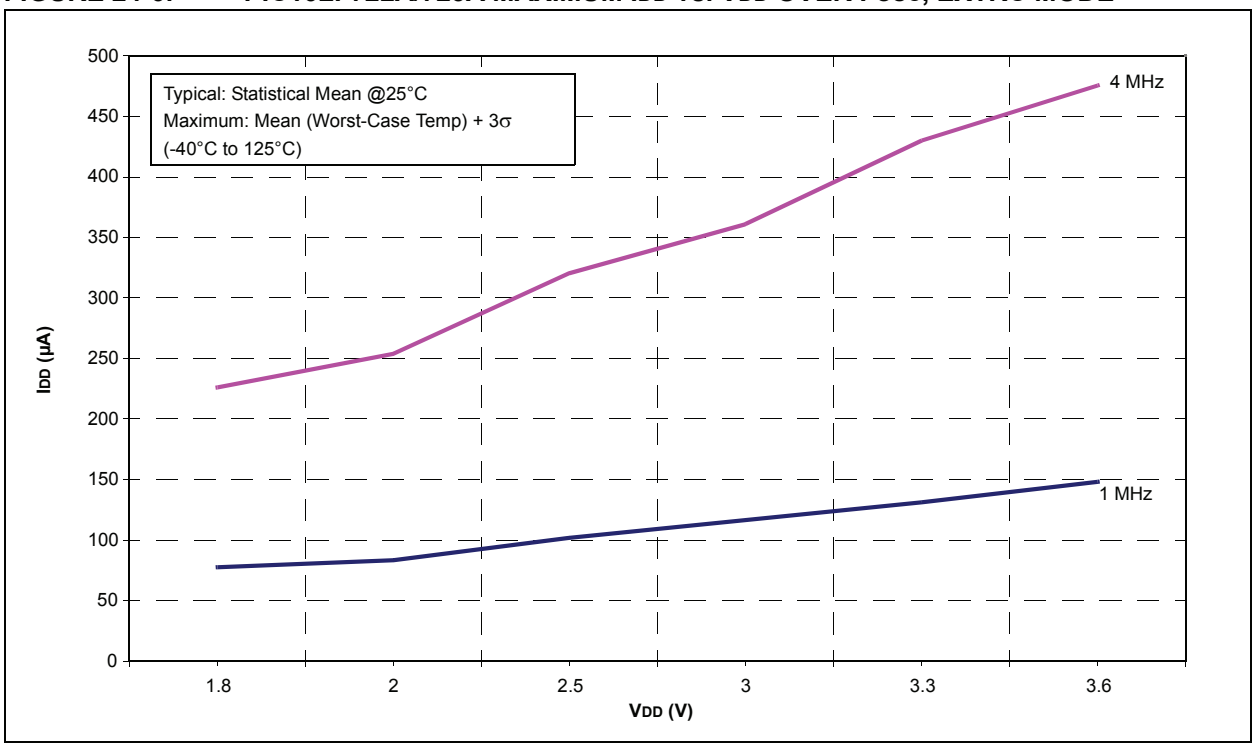


FIGURE 24-7: PIC16F722A/723A TYPICAL IDD vs. VDD OVER FOSC, EXTRC MODE, VCAP = 0.1µF

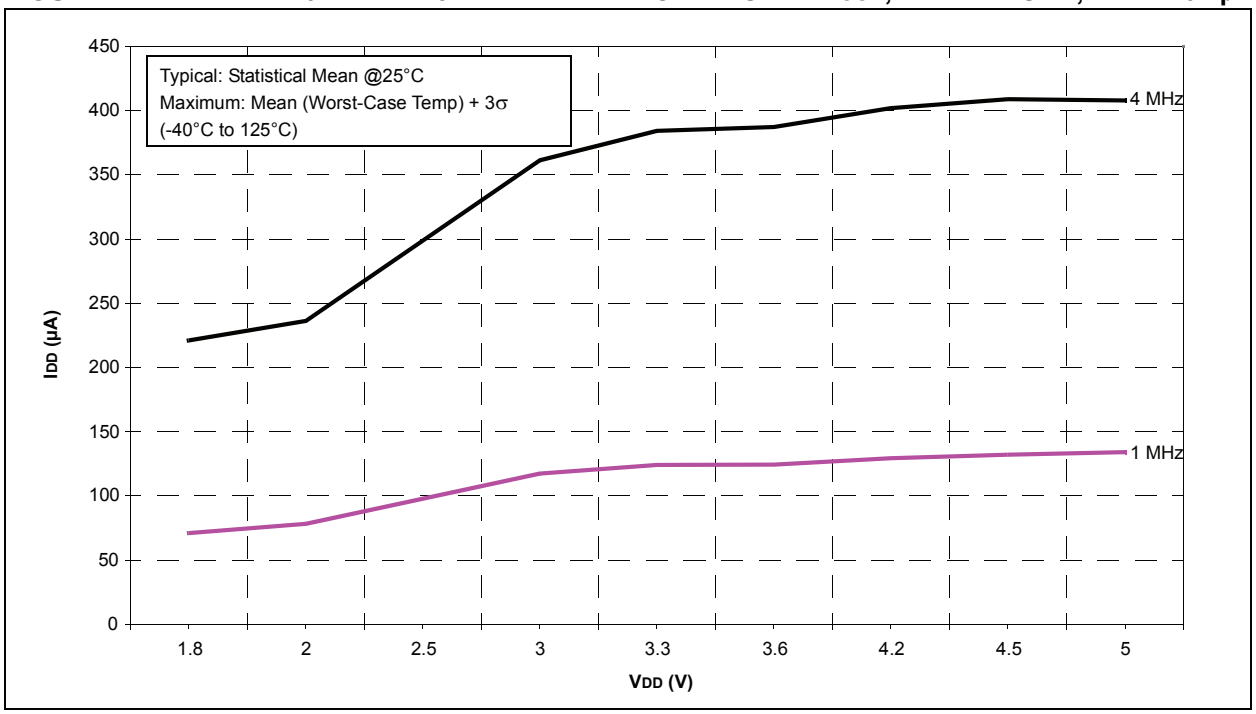


FIGURE 24-8: PIC16LF722A/723A TYPICAL IDD vs. VDD OVER FOSC, EXTRC MODE

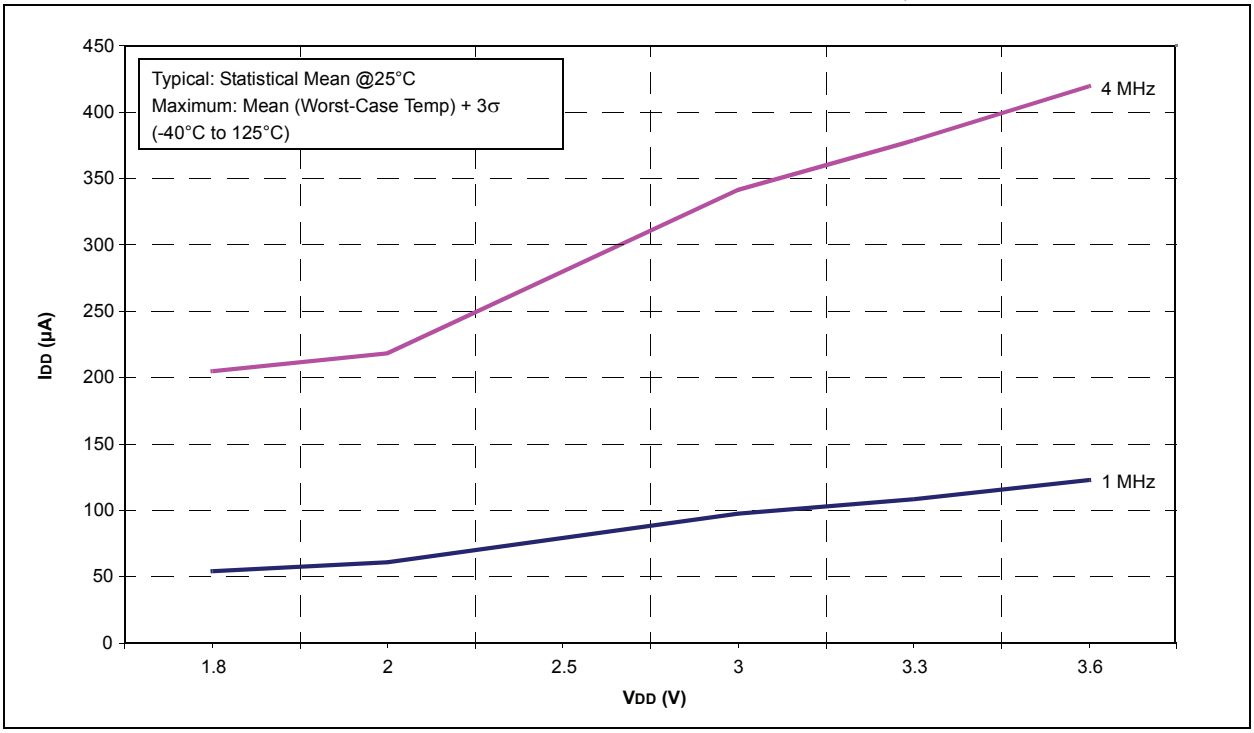


FIGURE 24-9: PIC16F722A/723A MAXIMUM IDD vs. FOSC OVER VDD, HS MODE, VCAP = 0.1µF

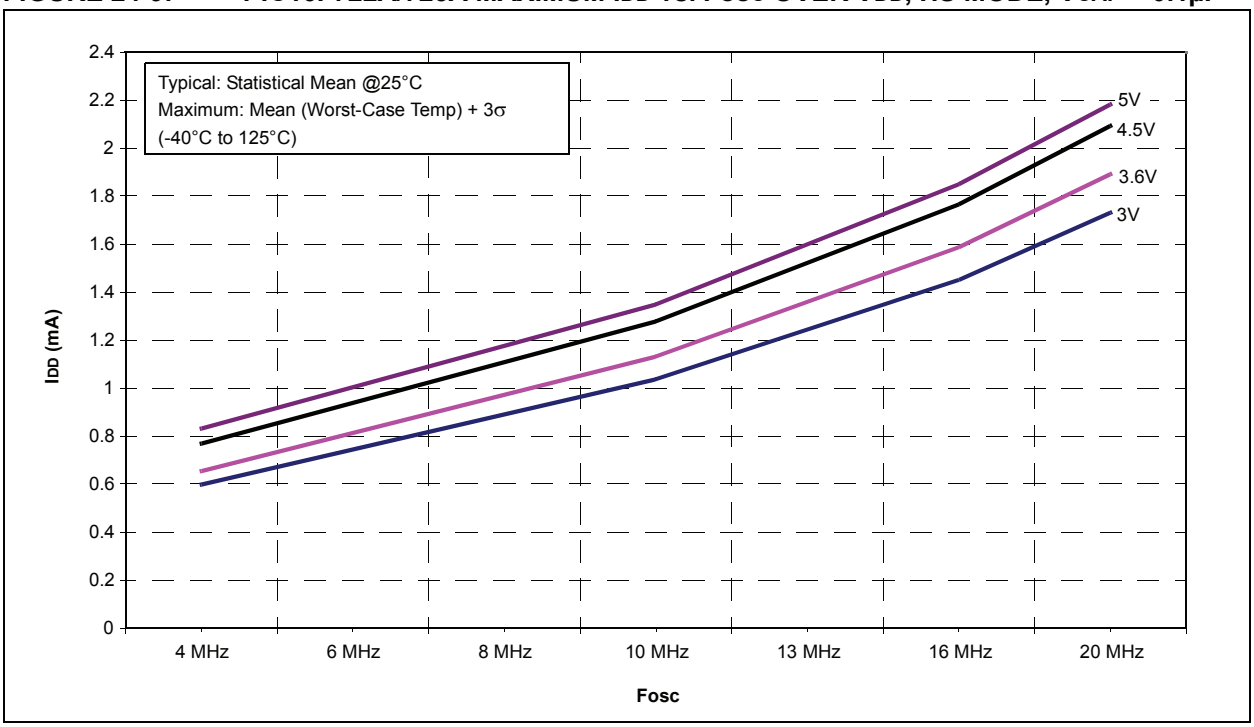


FIGURE 24-10: PIC16LF722A/723A MAXIMUM IDD vs. FOSC OVER VDD, HS MODE

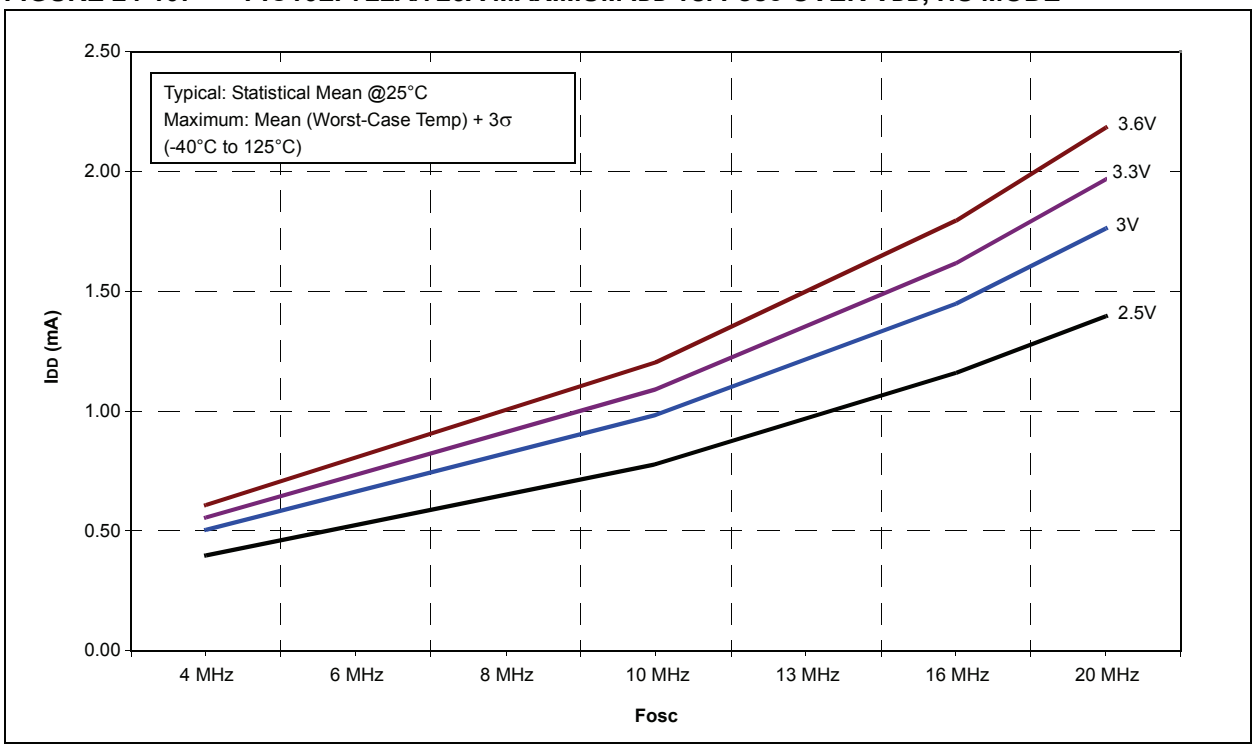


FIGURE 24-11: PIC16F722A/723A TYPICAL IDD vs. FOSC OVER VDD, HS MODE, VCAP = 0.1µF

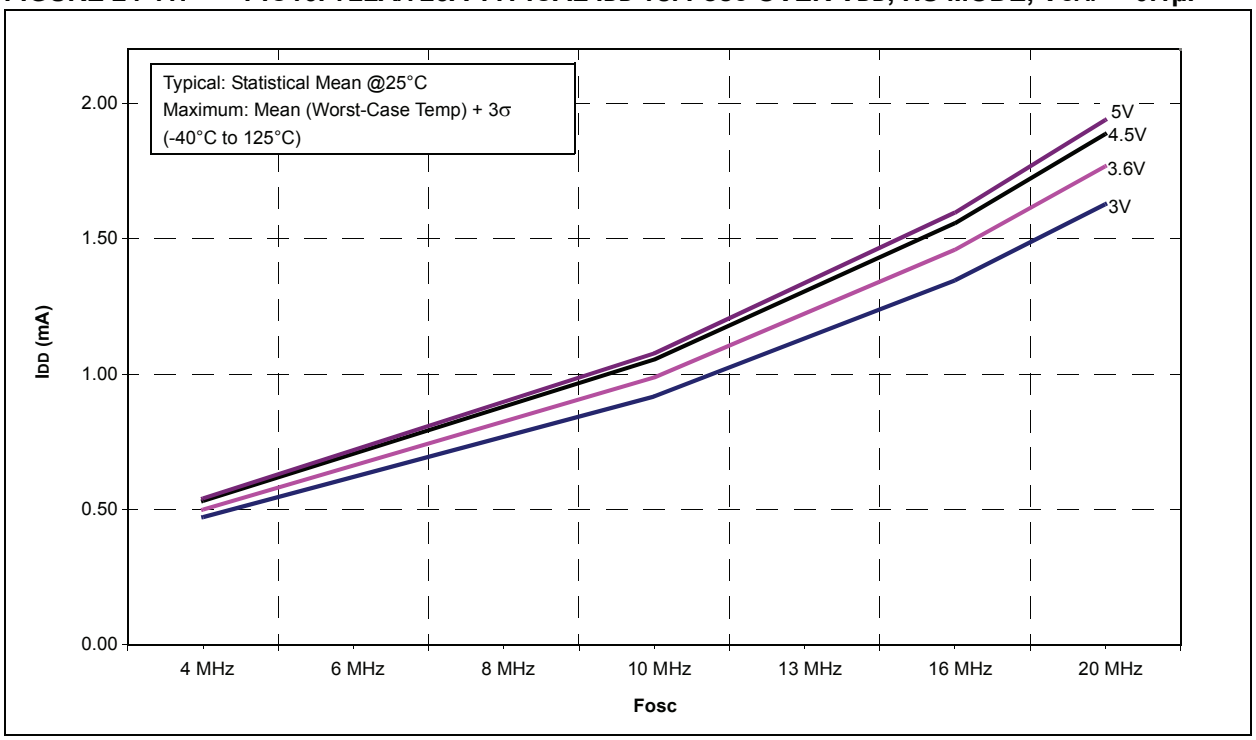


FIGURE 24-12: PIC16LF722A/723A TYPICAL IDD vs. FOSC OVER VDD, HS MODE

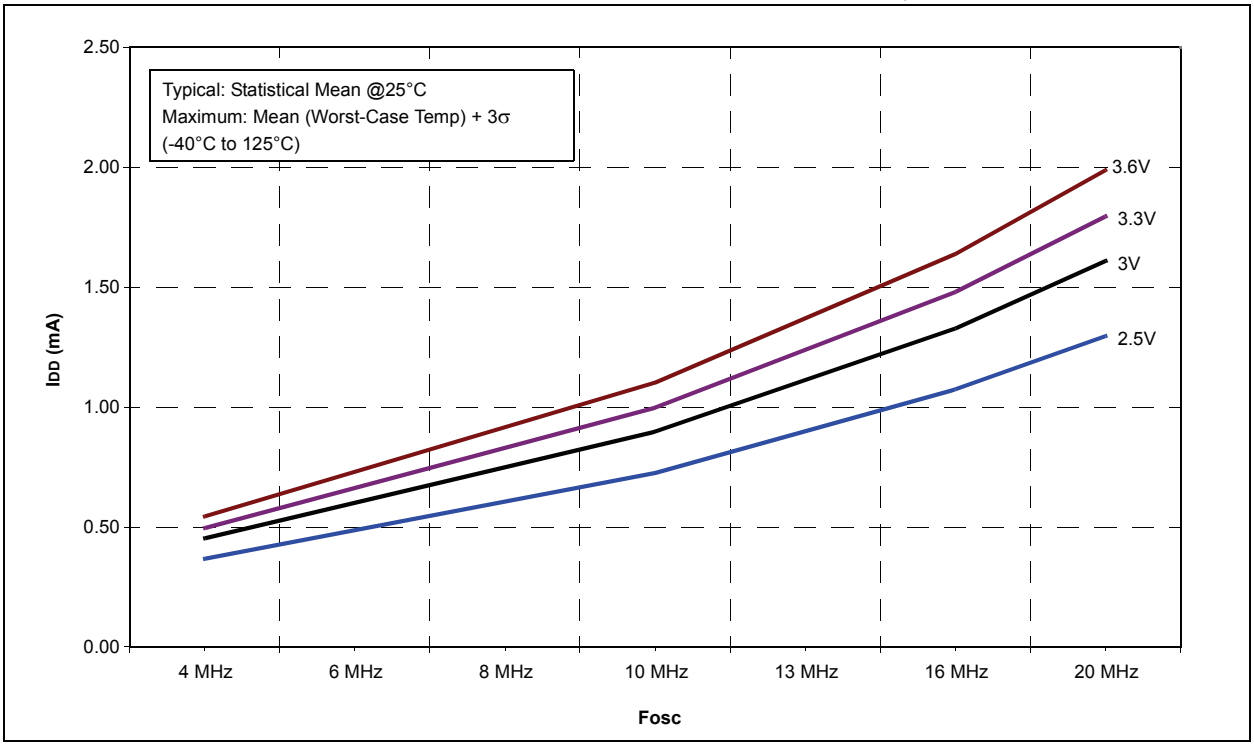


FIGURE 24-13: PIC16F722A/723A MAXIMUM IDD vs. VDD OVER FOSC, XT MODE, VCAP = 0.1µF

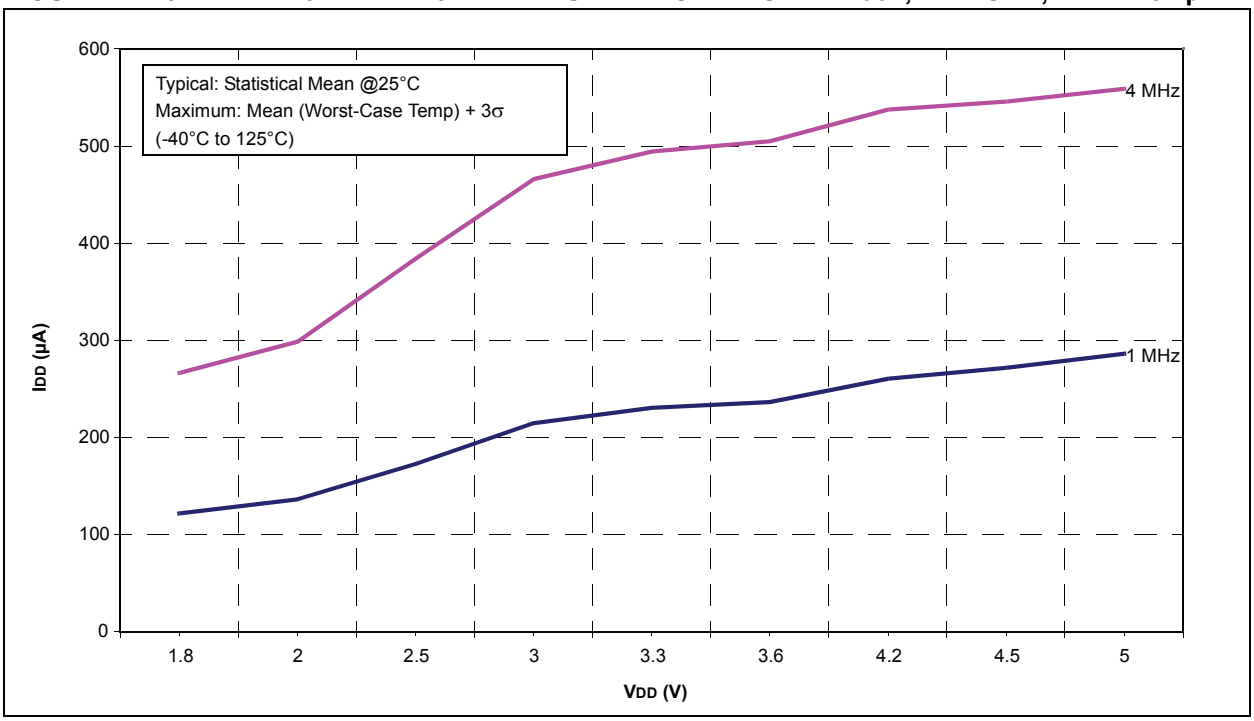


FIGURE 24-14: PIC16LF722A/723A MAXIMUM IDD vs. VDD OVER FOSC, XT MODE

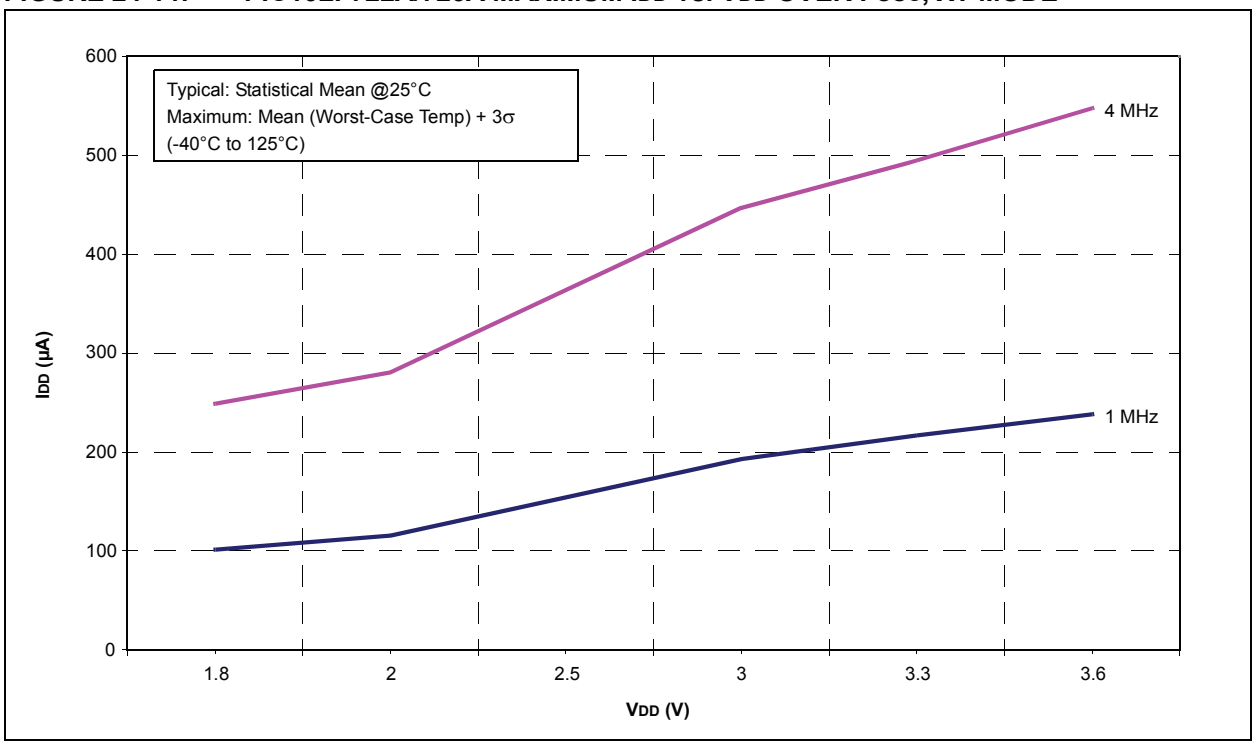


FIGURE 24-15: PIC16F722A/723A TYPICAL IDD vs. VDD OVER FOSC, XT MODE, VCAP = 0.1µF

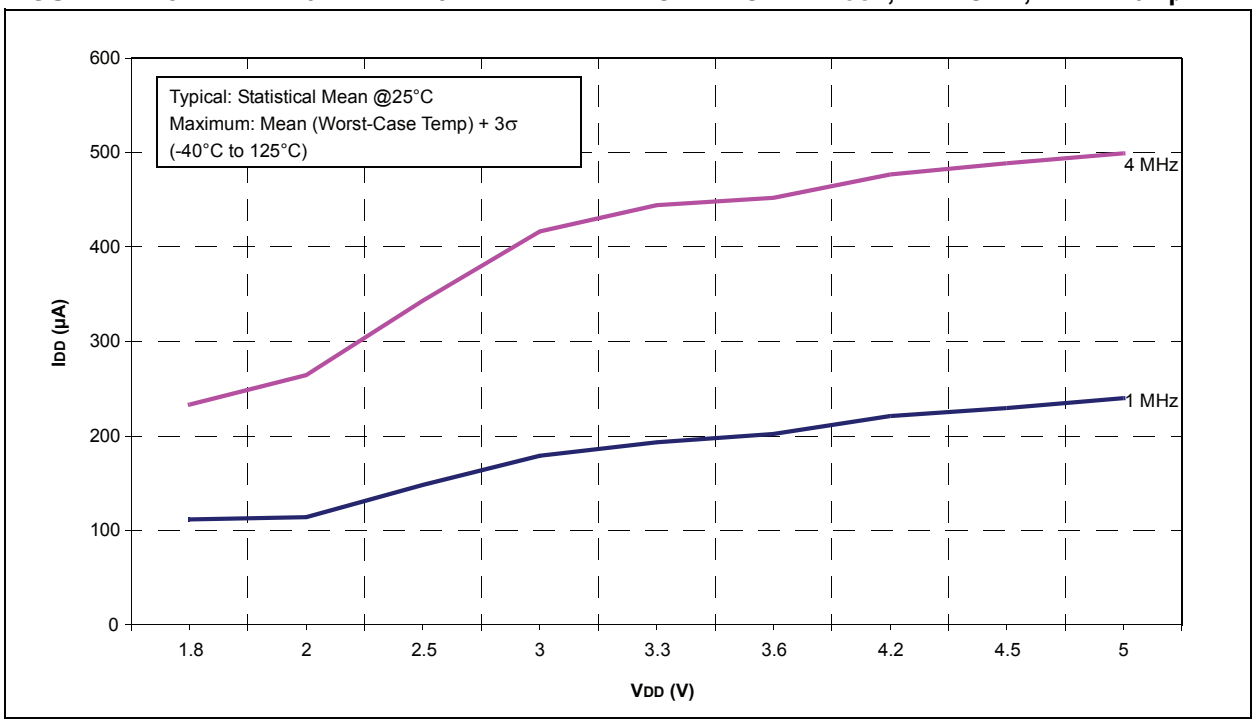


FIGURE 24-16: PIC16LF722A/723A TYPICAL IDD vs. VDD OVER FOSC, XT MODE

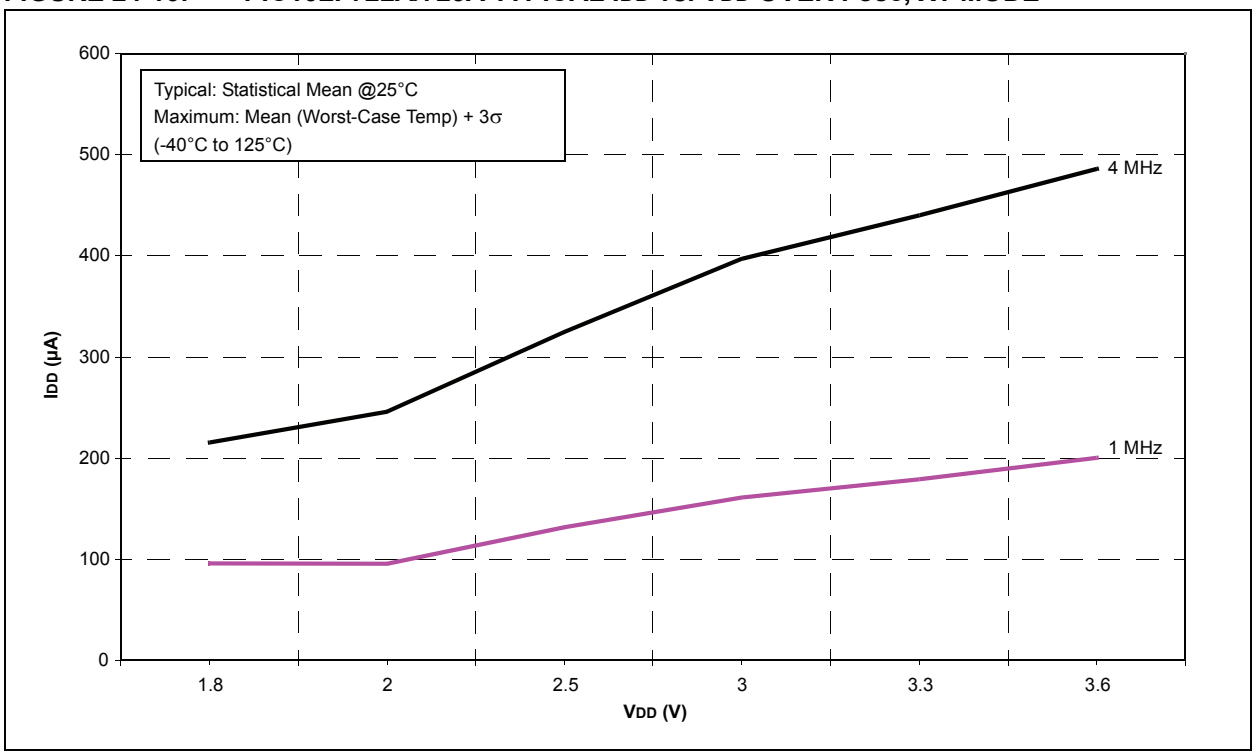


FIGURE 24-17: PIC16F722A/723A IDD vs. VDD, LP MODE, VCAP = 0.1µF

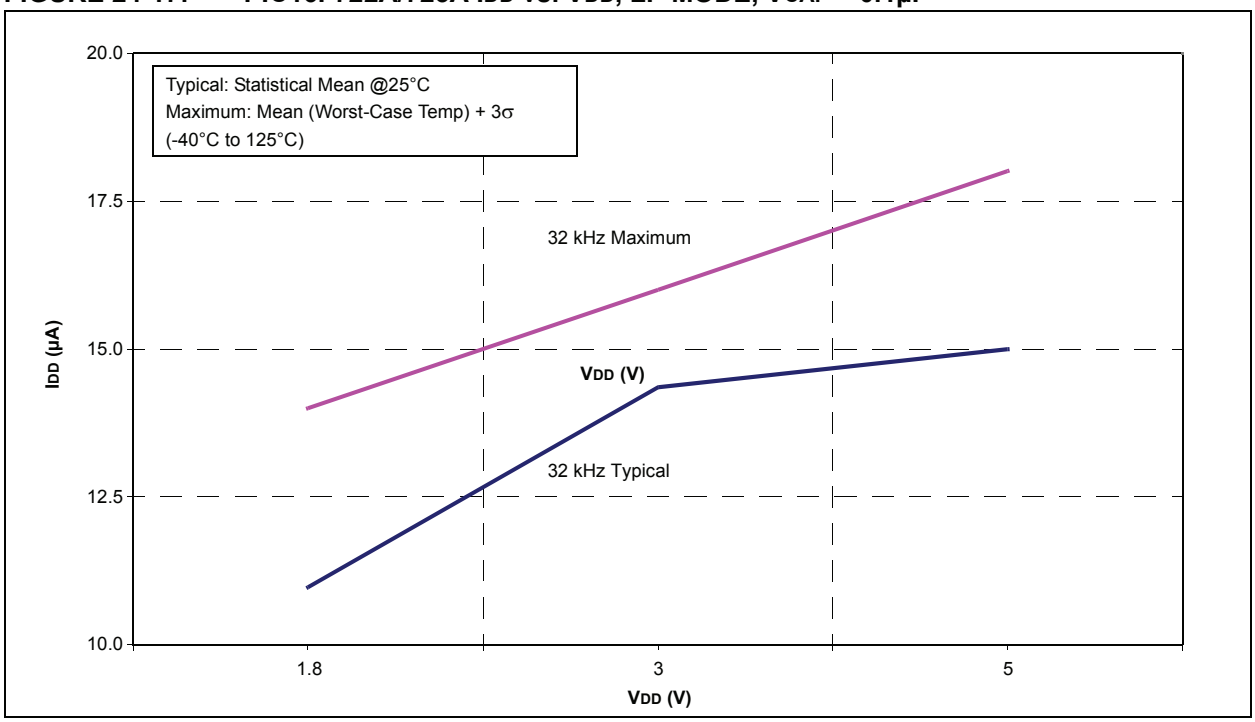


FIGURE 24-18: PIC16LF722A/723A IDD vs. VDD, LP MODE

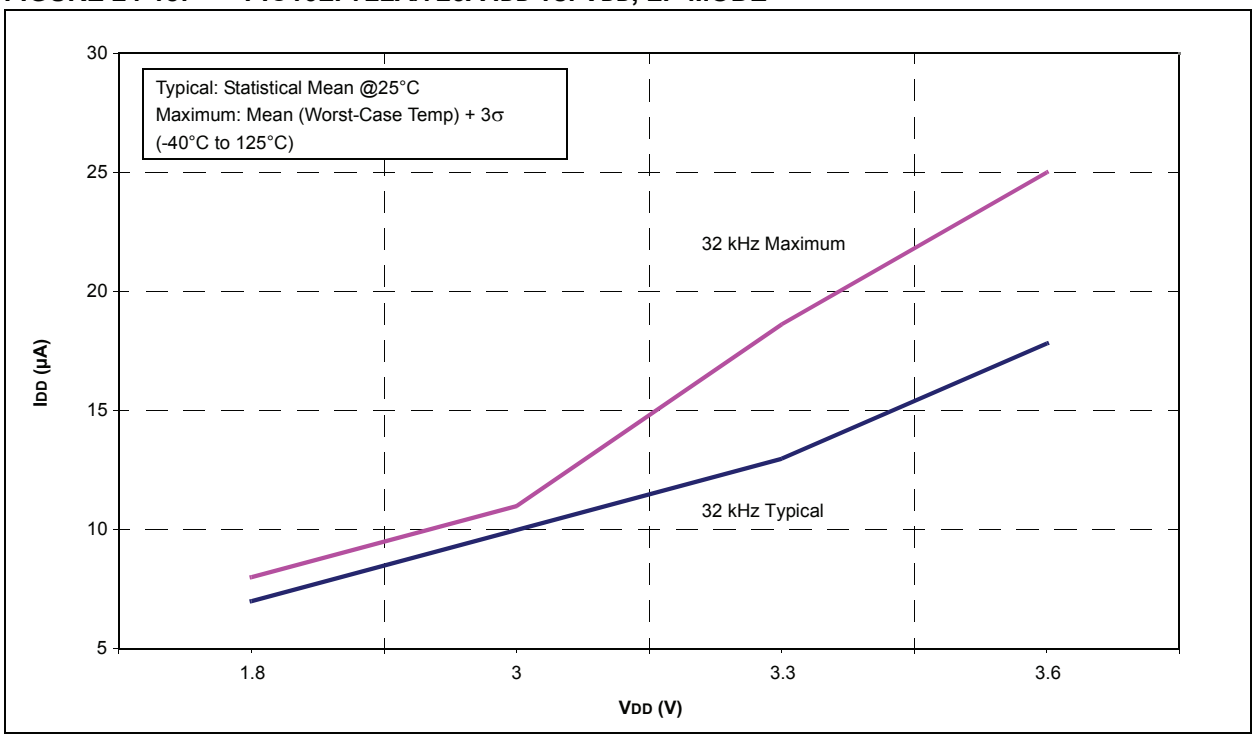


FIGURE 24-19: PIC16F722A/723A MAXIMUM IDD vs. FOSC OVER VDD, INTOSC MODE, VCAP = 0.1µF

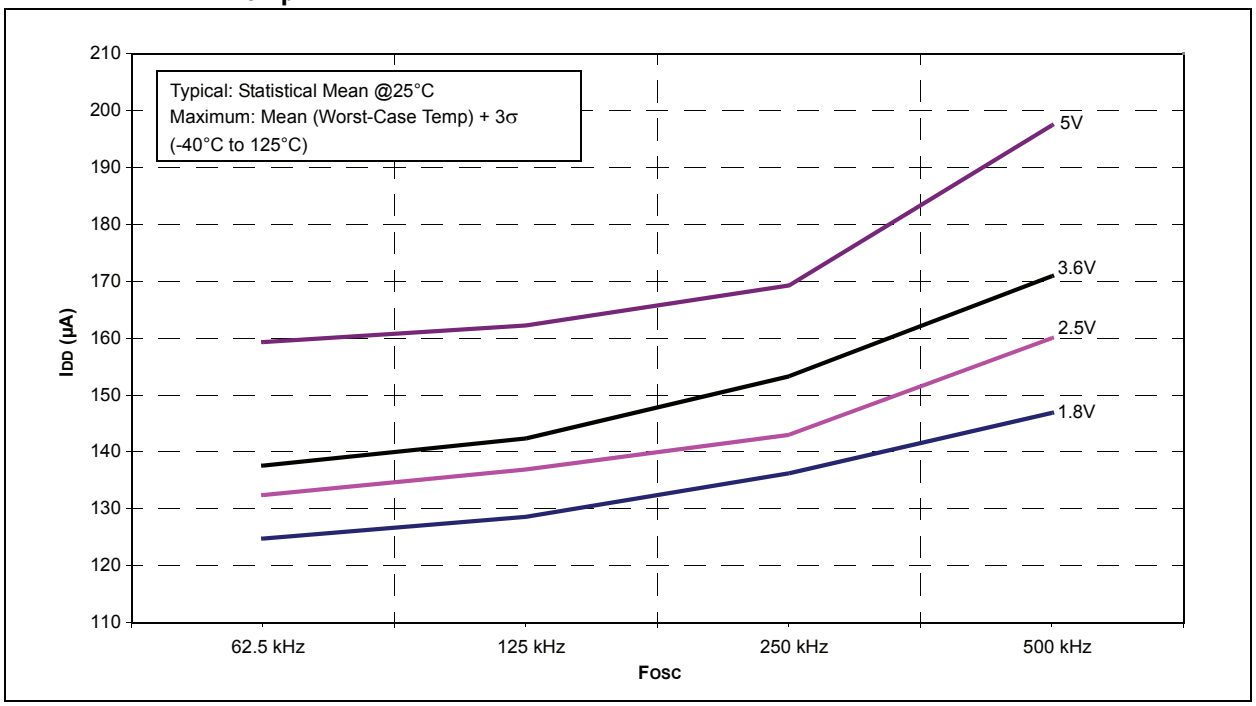


FIGURE 24-20: PIC16LF722A/723A MAXIMUM IDD vs. FOSC OVER VDD, INTOSC MODE

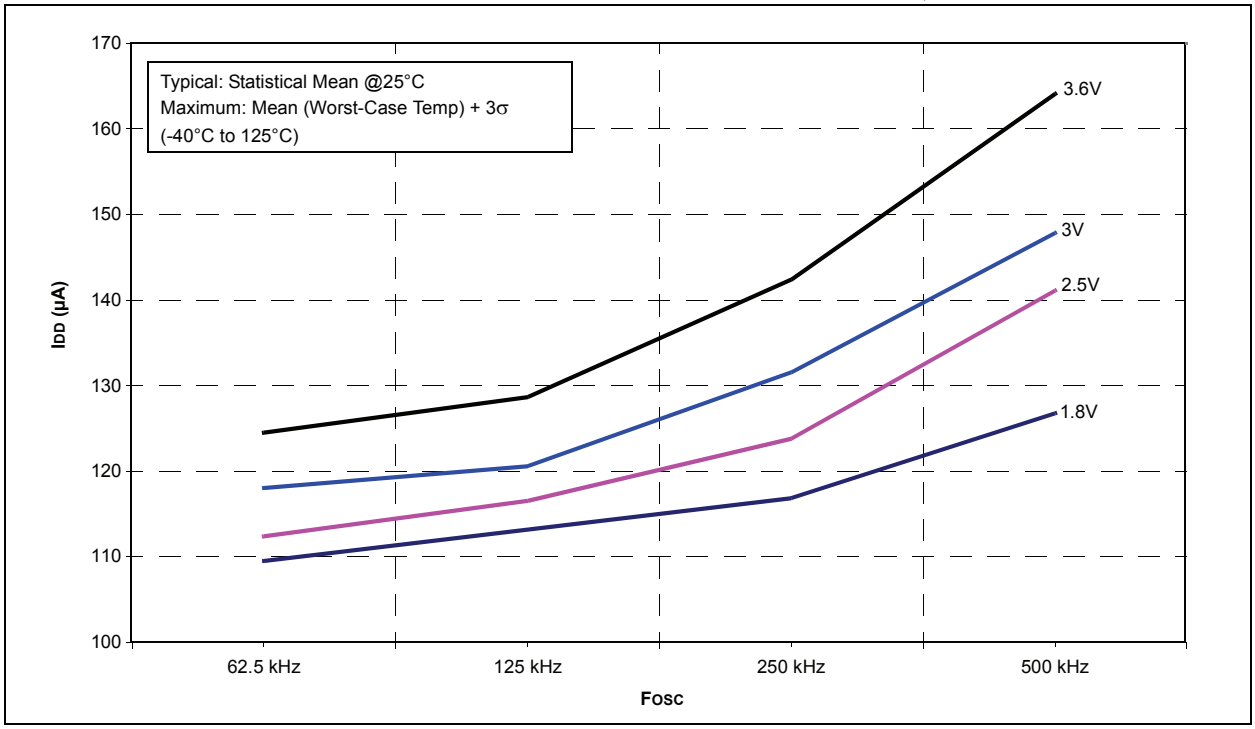


FIGURE 24-21: PIC16F722A/723A MAXIMUM IDD vs. FOSC OVER VDD, INTOSC MODE, VCAP = 0.1µF

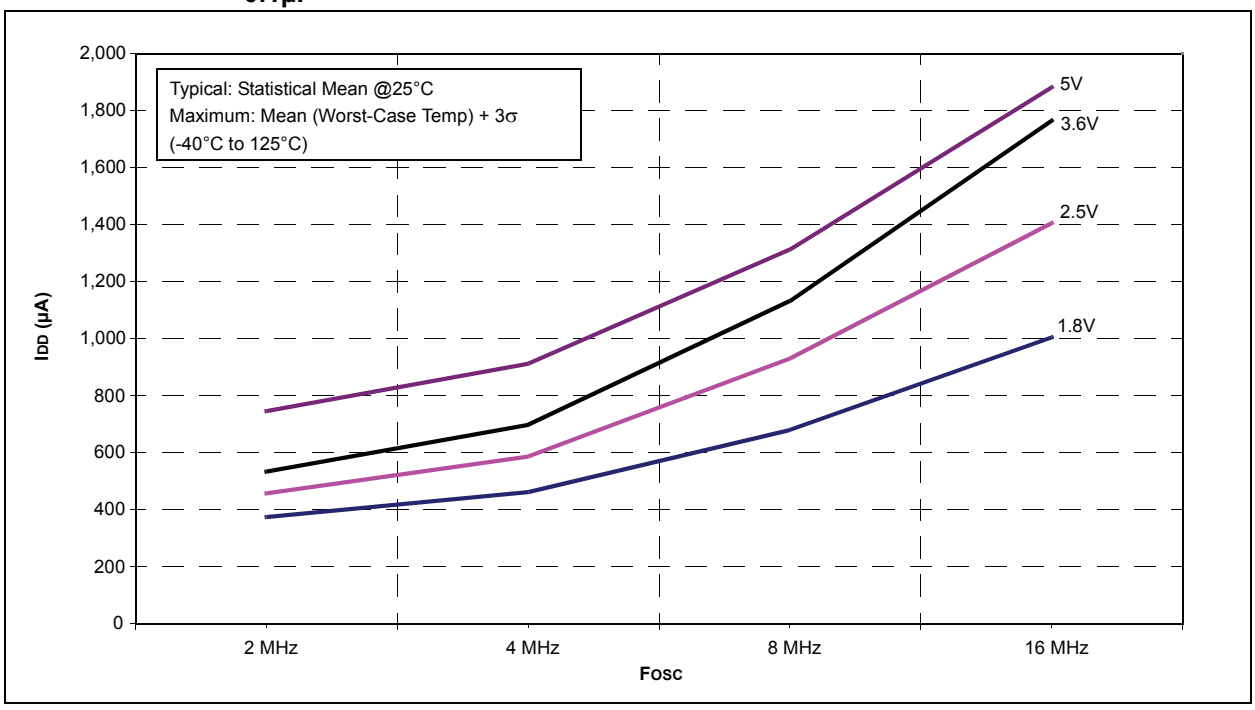


FIGURE 24-22: PIC16LF722A/723A MAXIMUM IDD vs. FOSC OVER VDD, INTOSC MODE

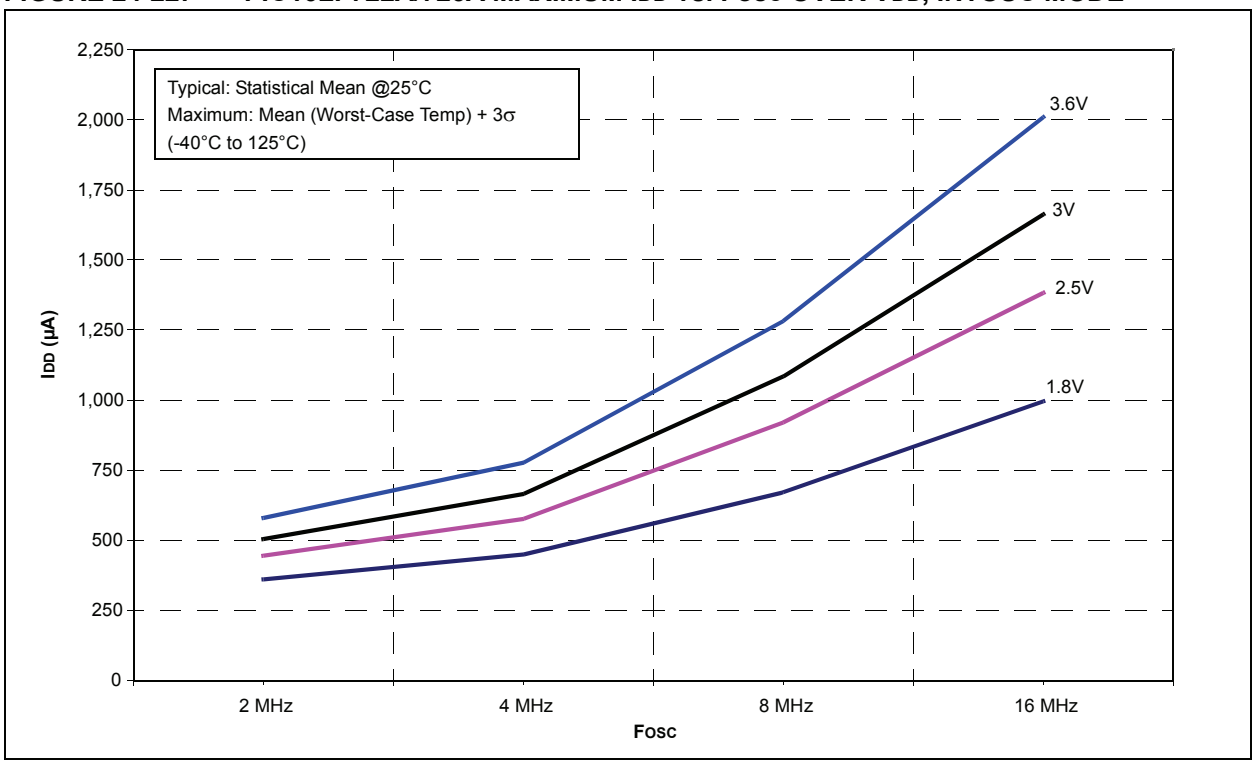


FIGURE 24-23: PIC16F722A/723A TYPICAL IDD vs. FOSC OVER VDD, INTOSC MODE, VCAP = 0.1µF

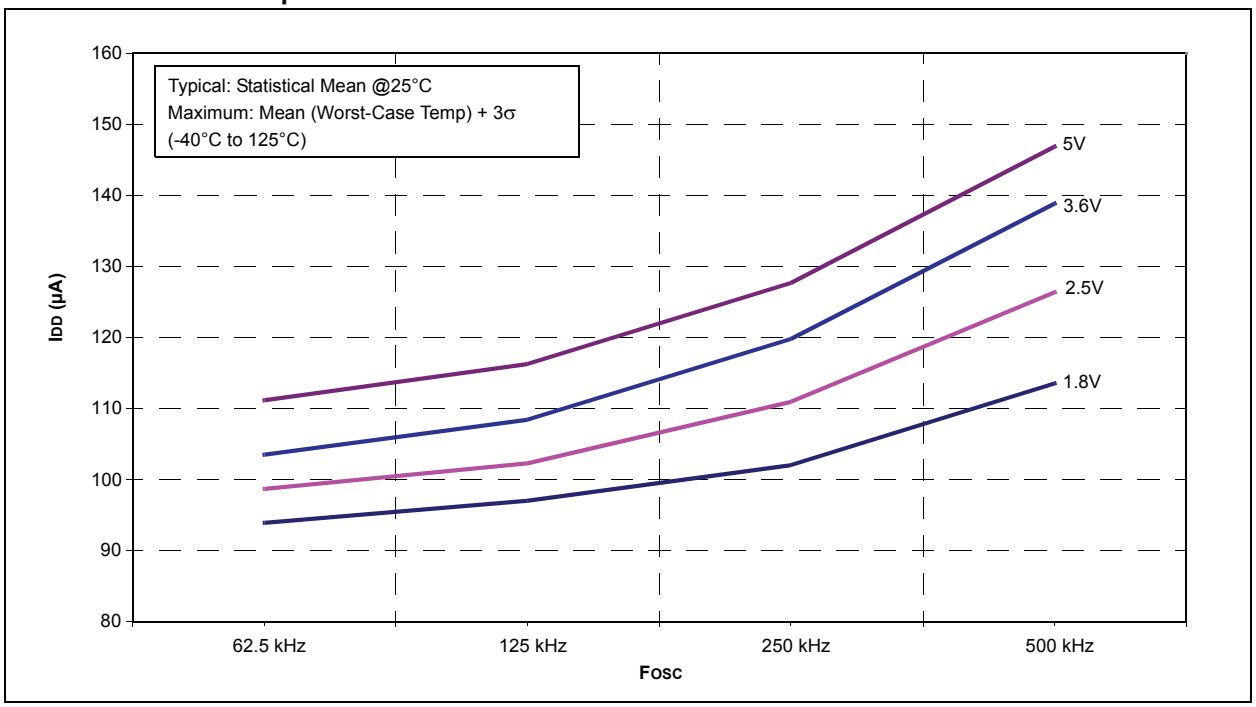


FIGURE 24-24: PIC16LF722A/723A TYPICAL IDD vs. FOSC OVER VDD, INTOSC MODE

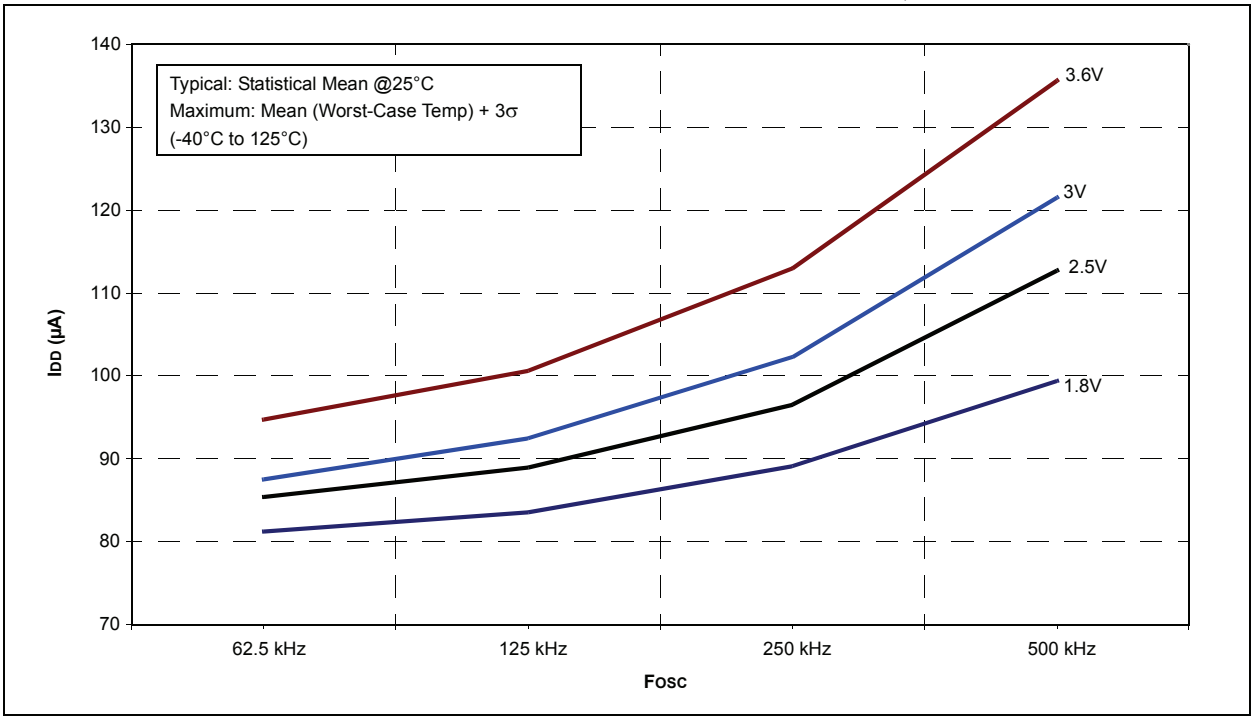


FIGURE 24-25: PIC16F722A/723A TYPICAL IDD vs. FOSC OVER VDD, INTOSC MODE, VCAP = 0.1µF

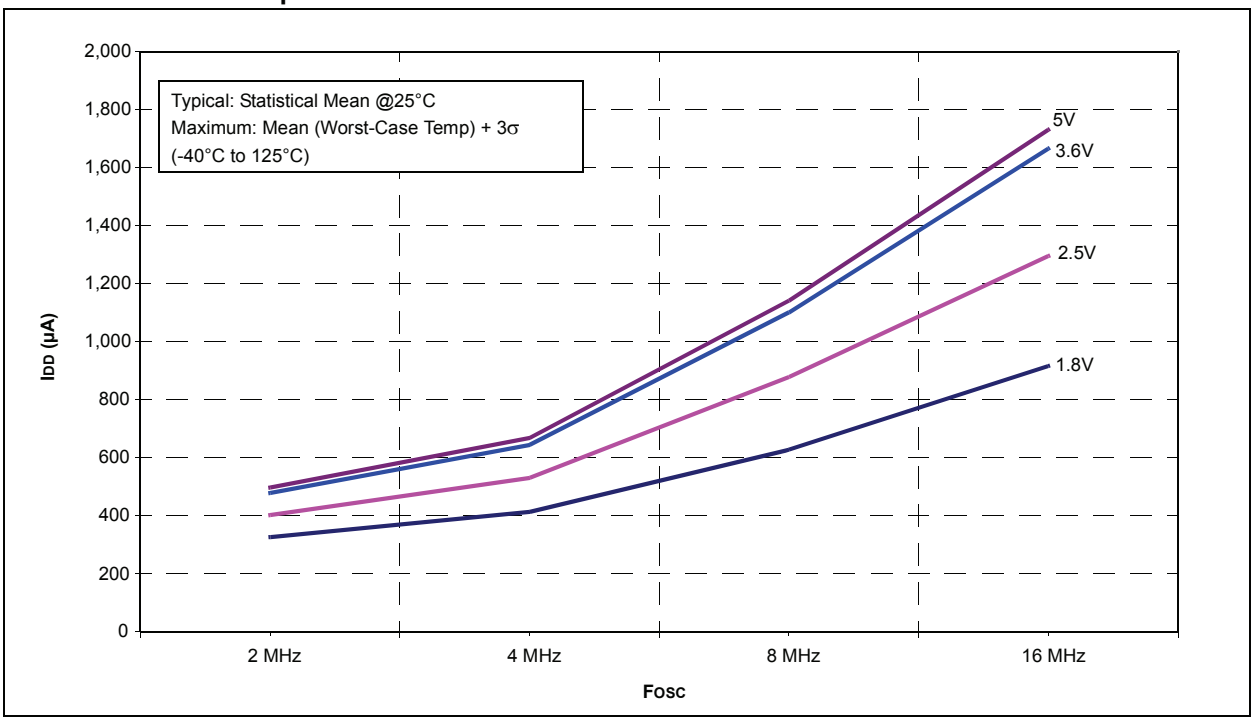


FIGURE 24-26: PIC16LF722A/723A TYPICAL IDD vs. FOSC OVER VDD, INTOSC MODE

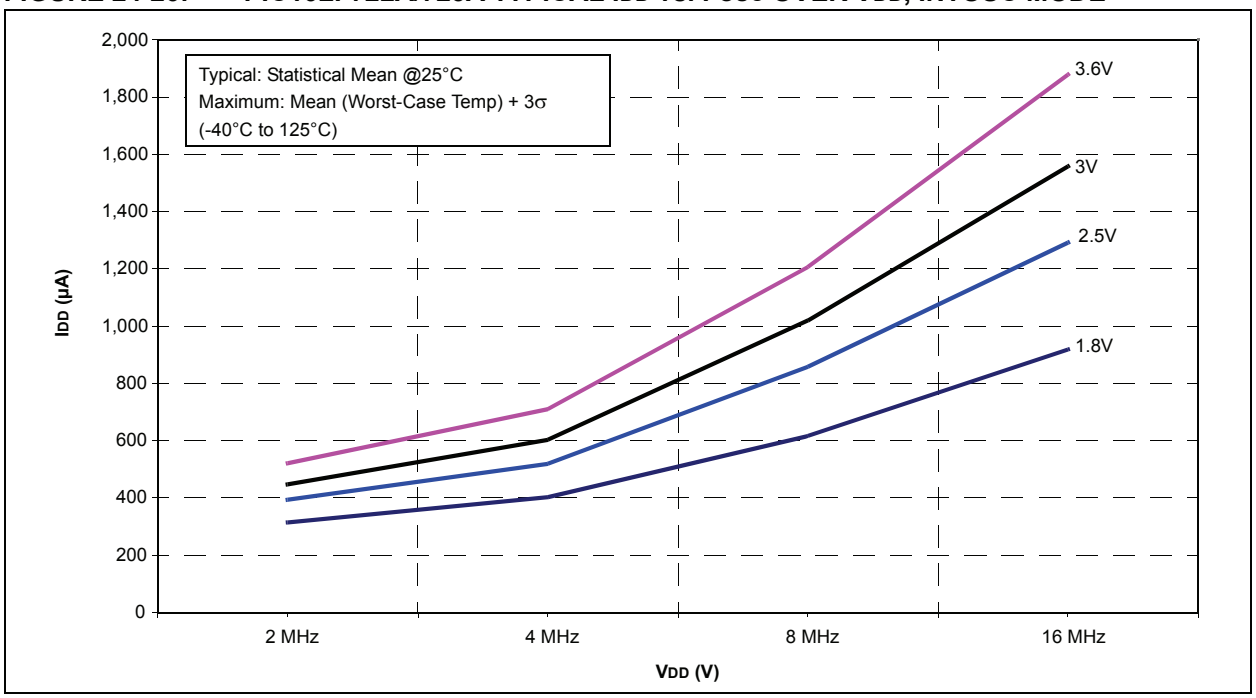


FIGURE 24-27: PIC16F722A/723A MAXIMUM BASE IPD vs. VDD, VCAP = 0.1µF

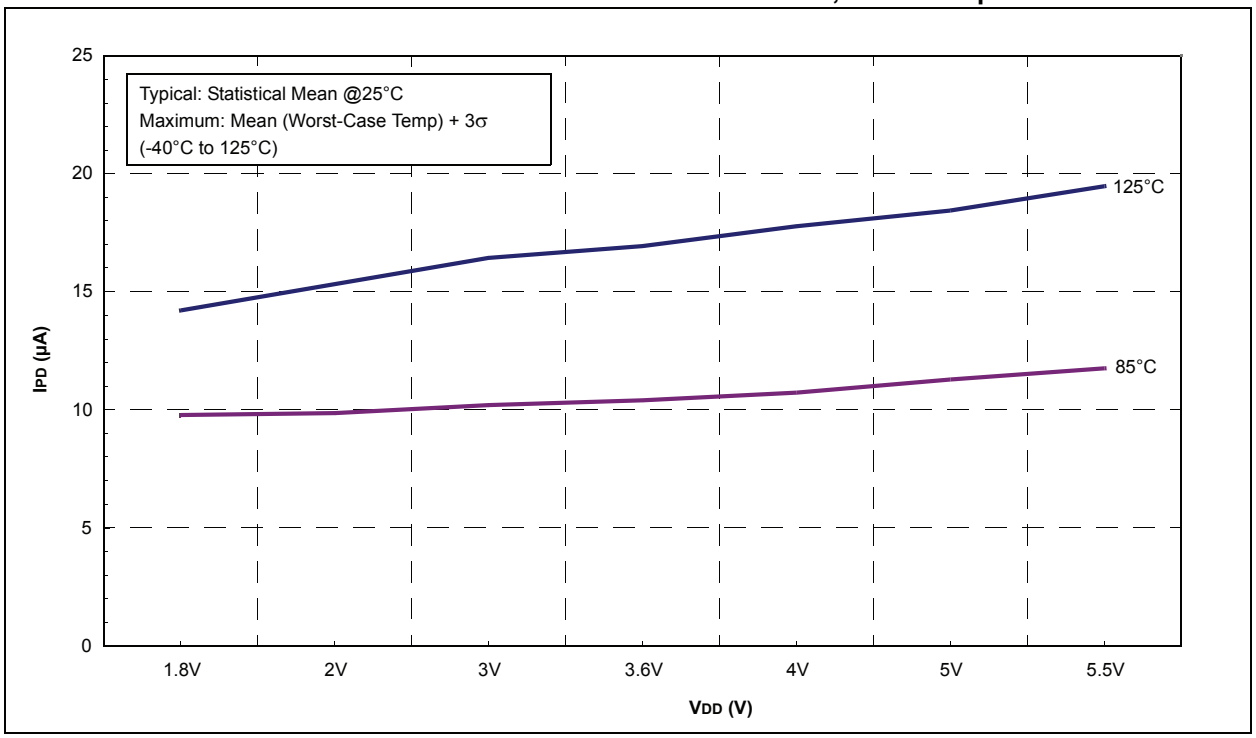


FIGURE 24-28: PIC16LF722A/723A MAXIMUM BASE IPD vs. VDD

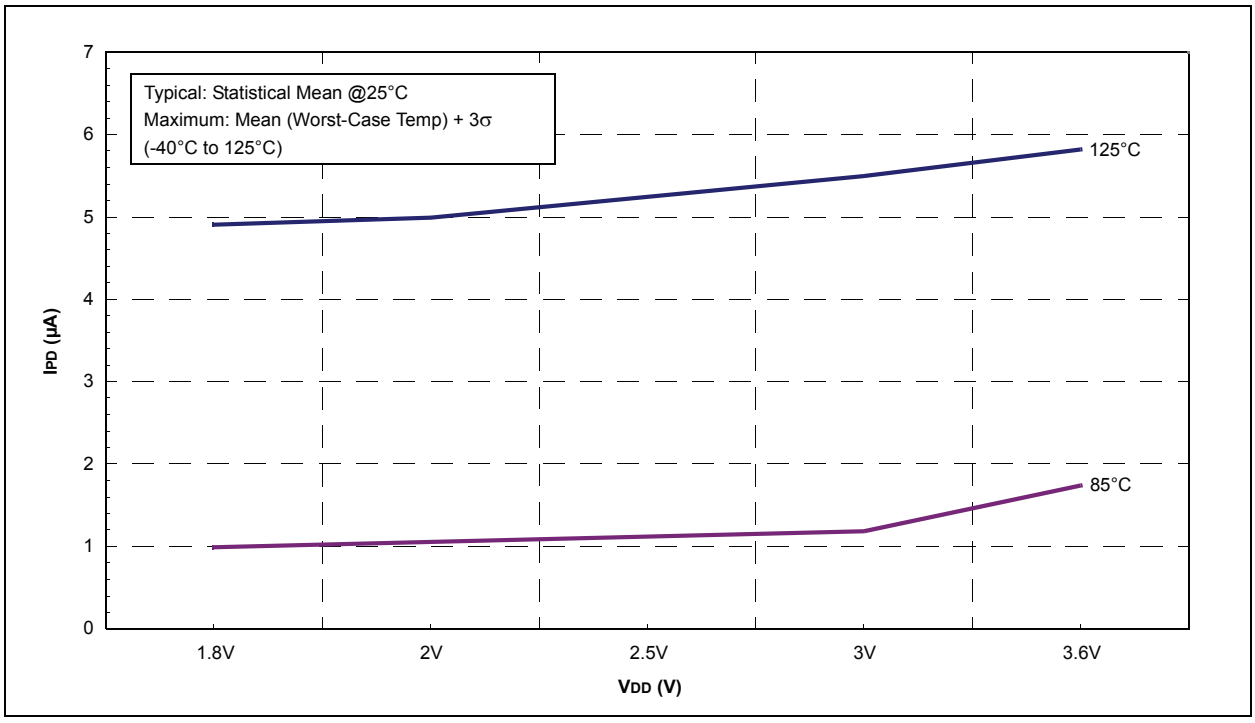


FIGURE 24-29: PIC16F722A/723A TYPICAL BASE IPD vs. VDD, VCAP = 0.1µF

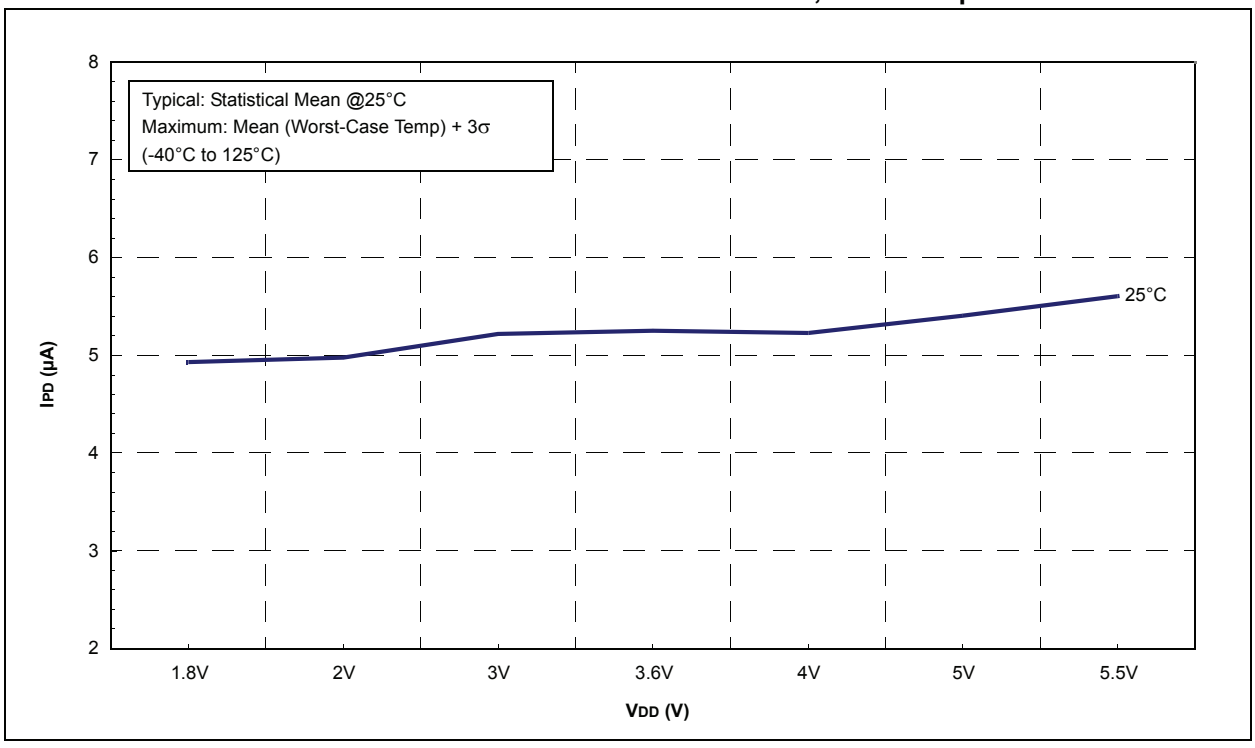


FIGURE 24-30: PIC16LF722A/723A TYPICAL BASE IPD vs. VDD

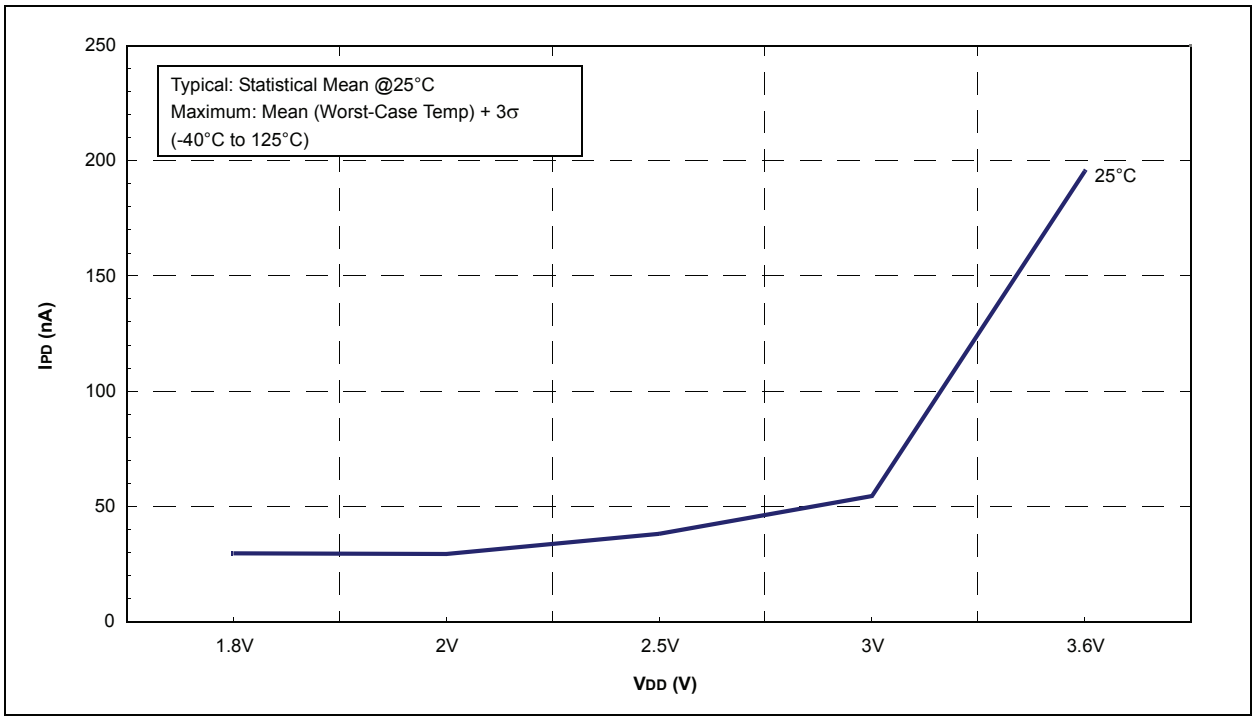


FIGURE 24-31: PIC16F722A/723A FIXED VOLTAGE REFERENCE IPD vs. VDD, VCAP = 0.1µF

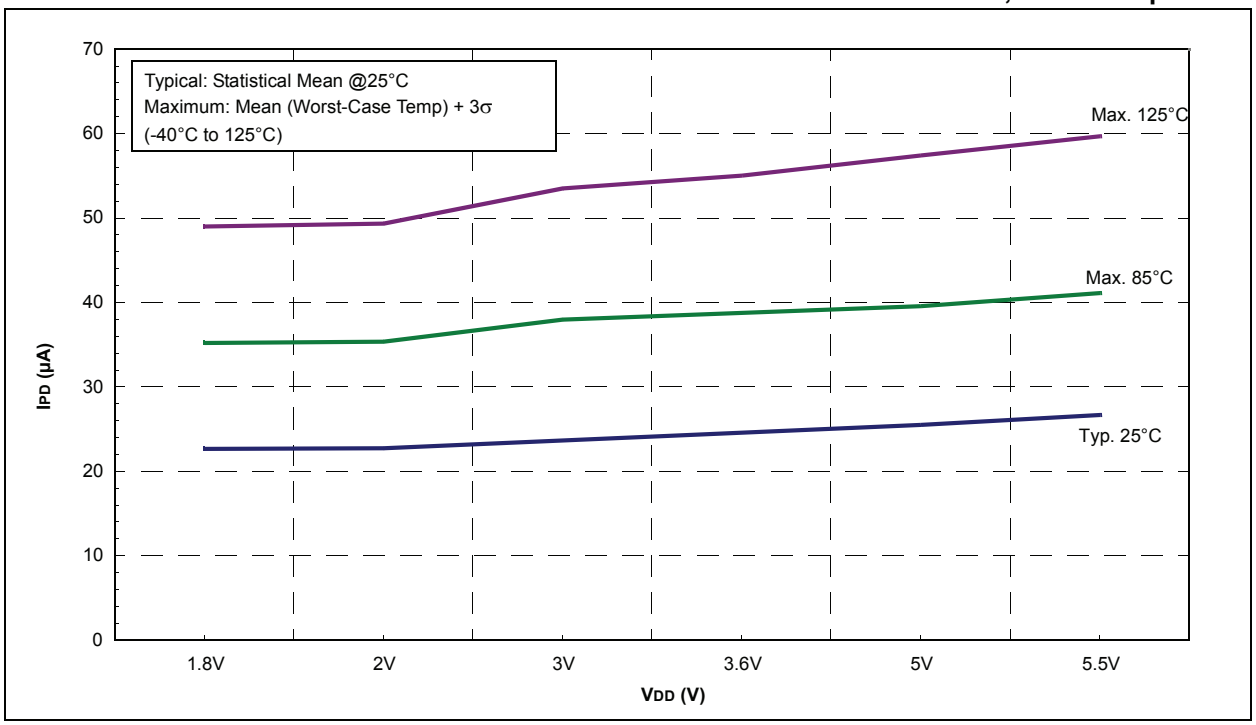


FIGURE 24-32: PIC16LF722A/723A FIXED VOLTAGE REFERENCE IPD vs. VDD

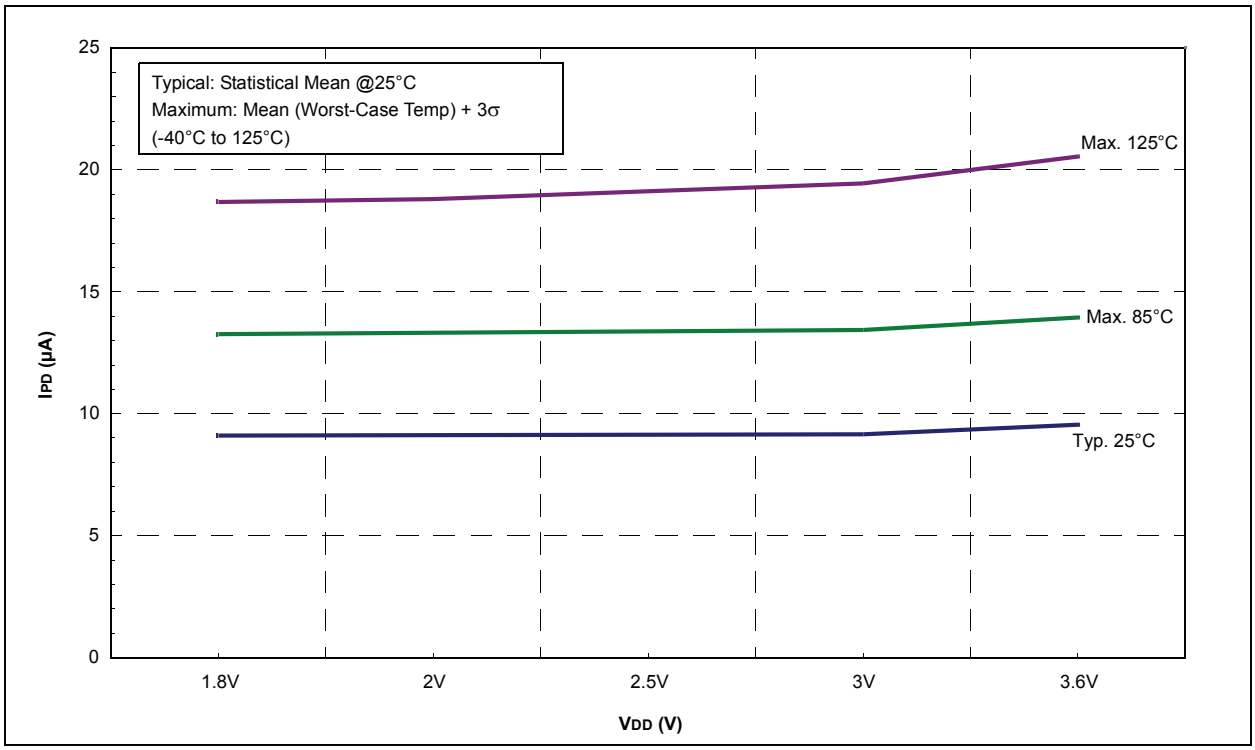


FIGURE 24-33: PIC16F722A/723A BOR IPD vs. VDD, VCAP = 0.1µF

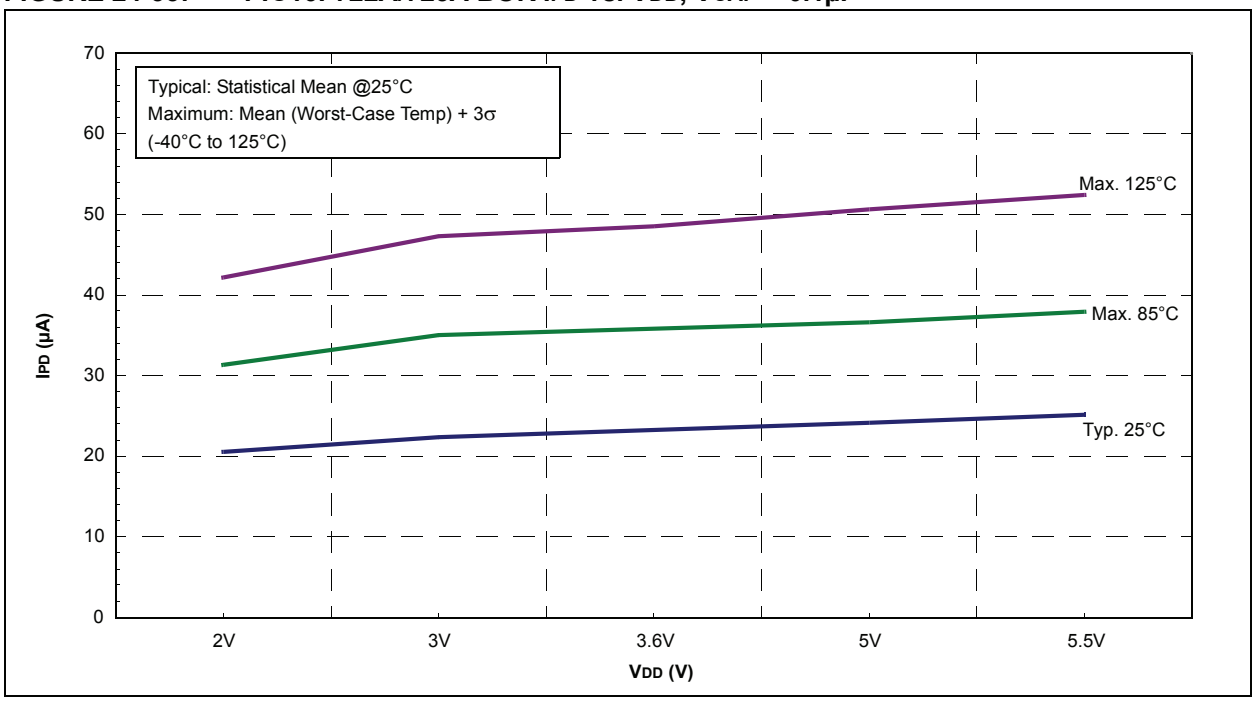


FIGURE 24-34: PIC16LF722A/723A BOR IPD vs. VDD

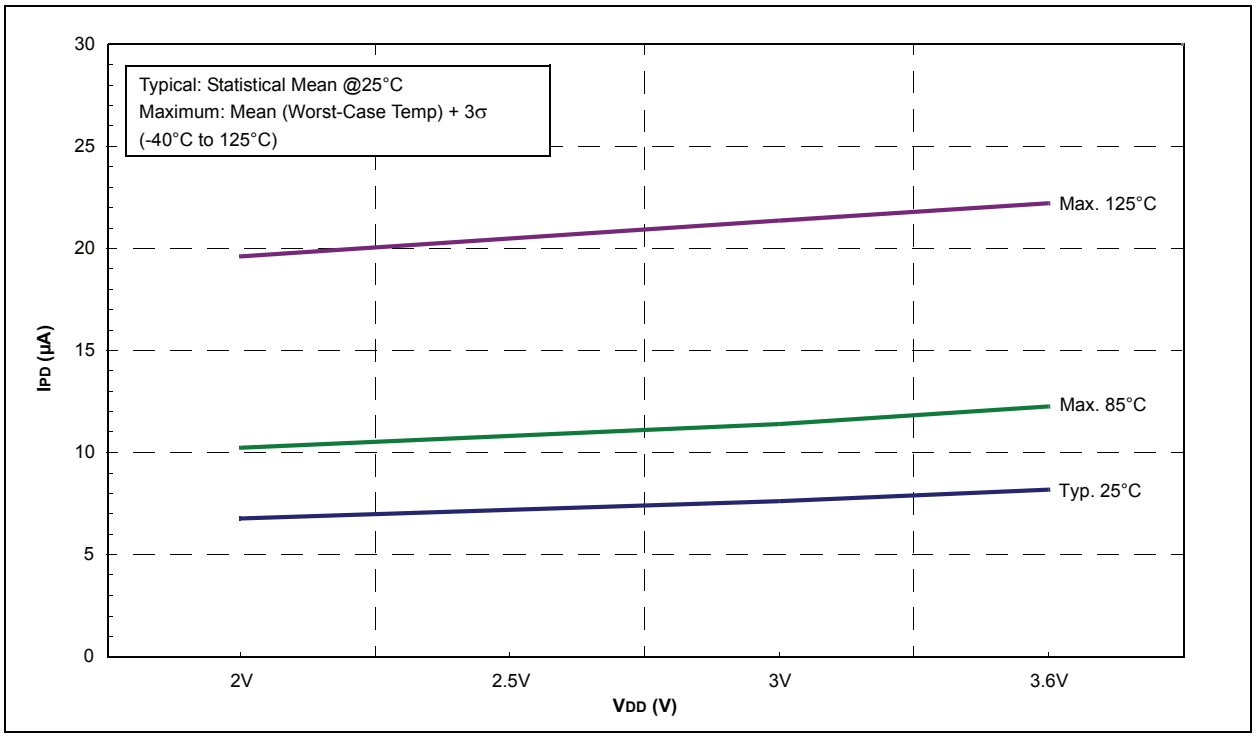


FIGURE 24-35: PIC16F722A/723A CAP SENSE HIGH POWER IPD vs. VDD, VCAP = 0.1µF

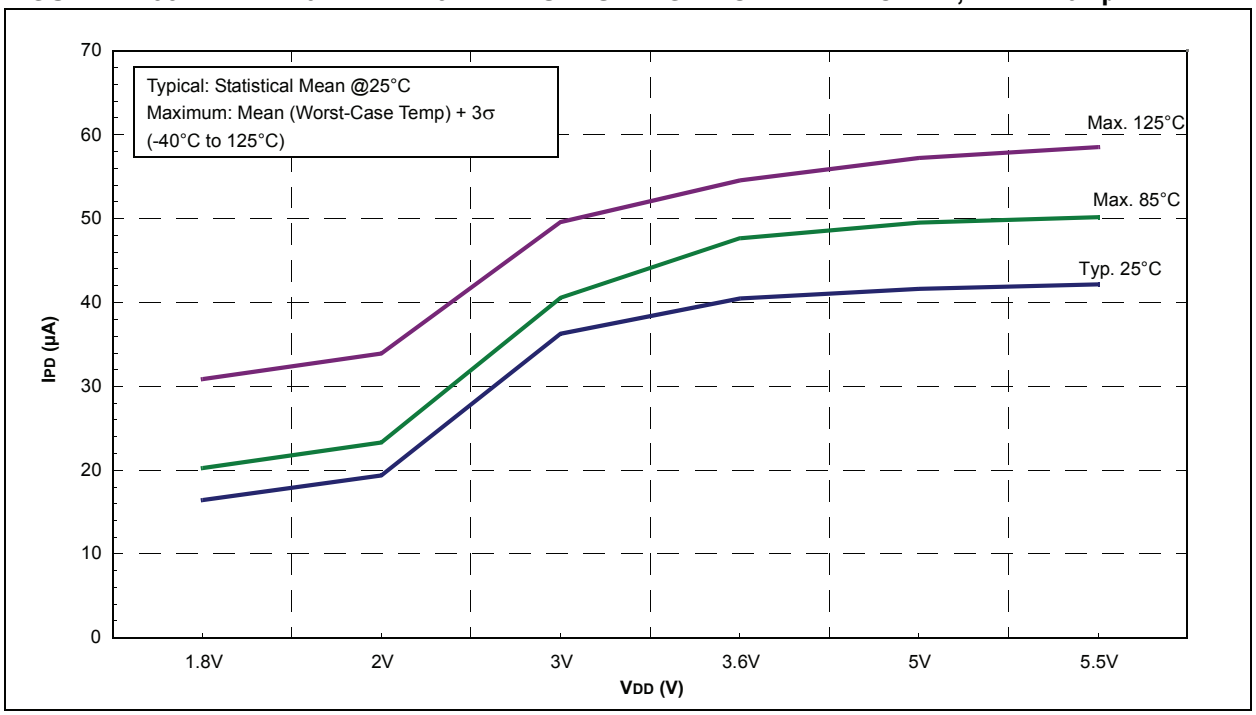


FIGURE 24-36: PIC16LF722A/723A CAP SENSE HIGH POWER IPD vs. VDD

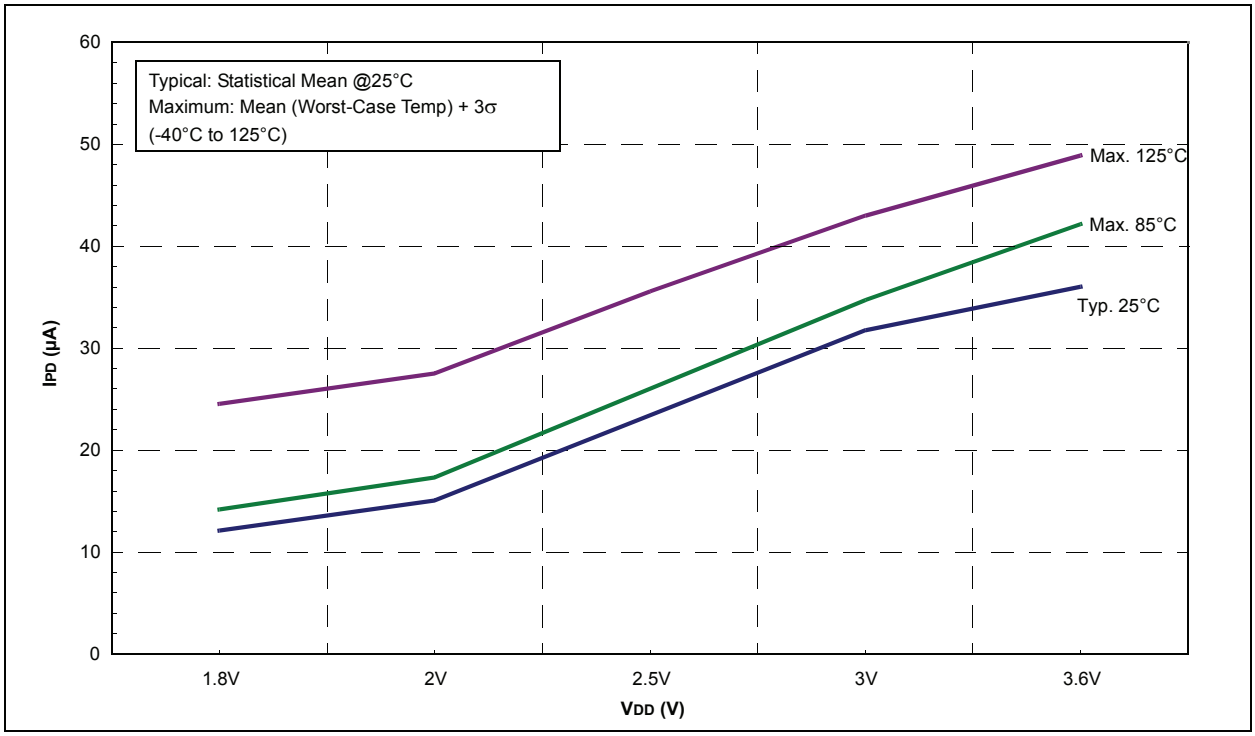


FIGURE 24-37: PIC16F722A/723A CAP SENSE MEDIUM POWER IPD vs. VDD, VCAP = 0.1µF

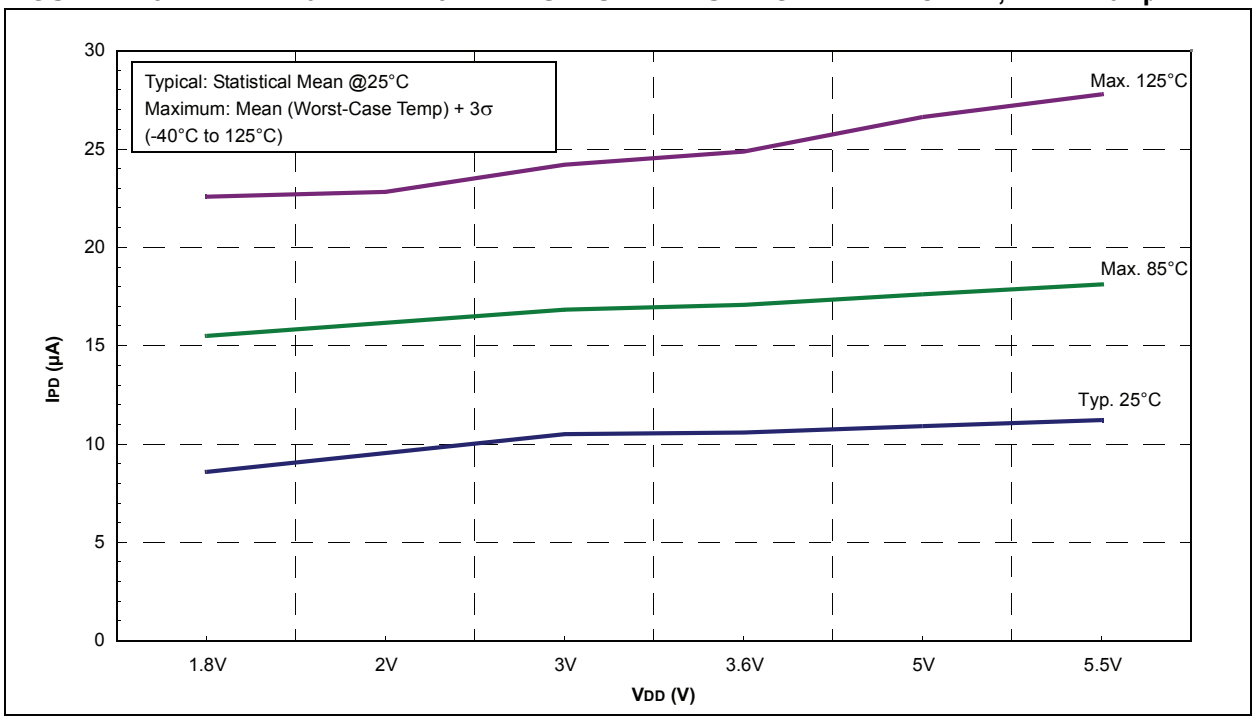


FIGURE 24-38: PIC16LF722A/723A CAP SENSE MEDIUM POWER IPD vs. VDD

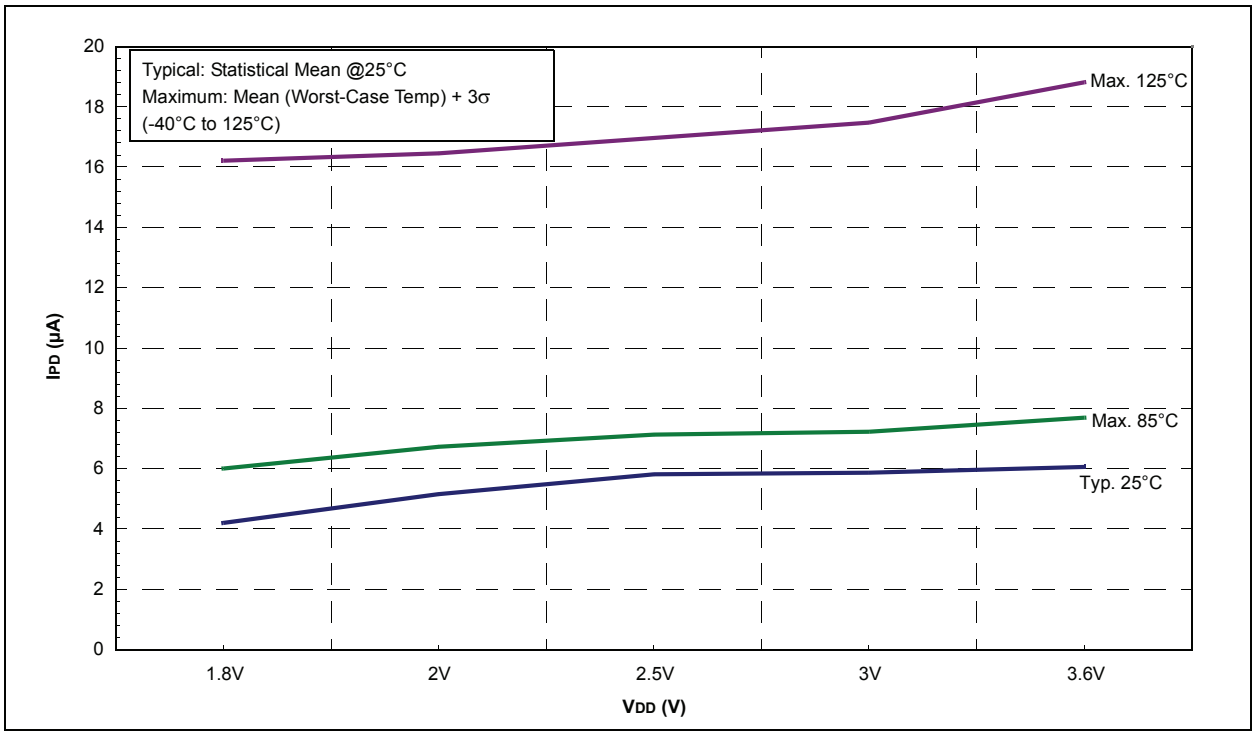


FIGURE 24-39: PIC16F722A/723A CAP SENSE LOW POWER IPD vs. VDD, VCAP = 0.1µF

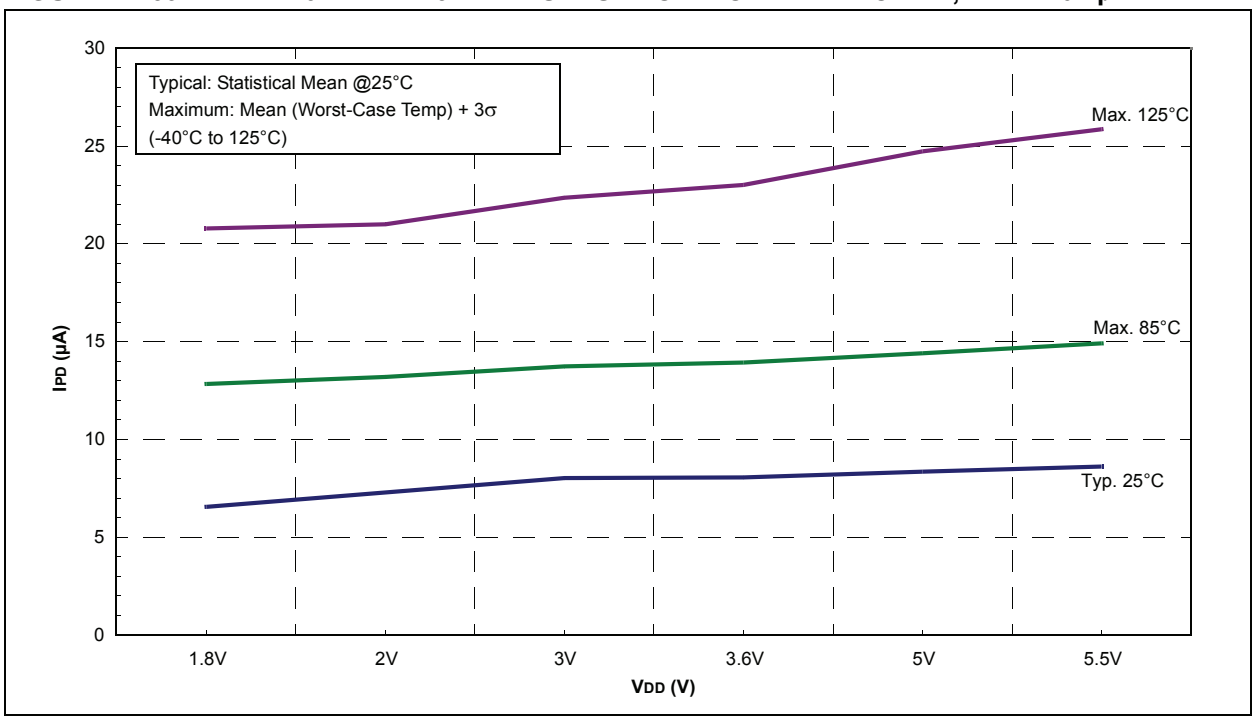


FIGURE 24-40: PIC16LF722A/723A CAP SENSE LOW POWER IPD vs. VDD

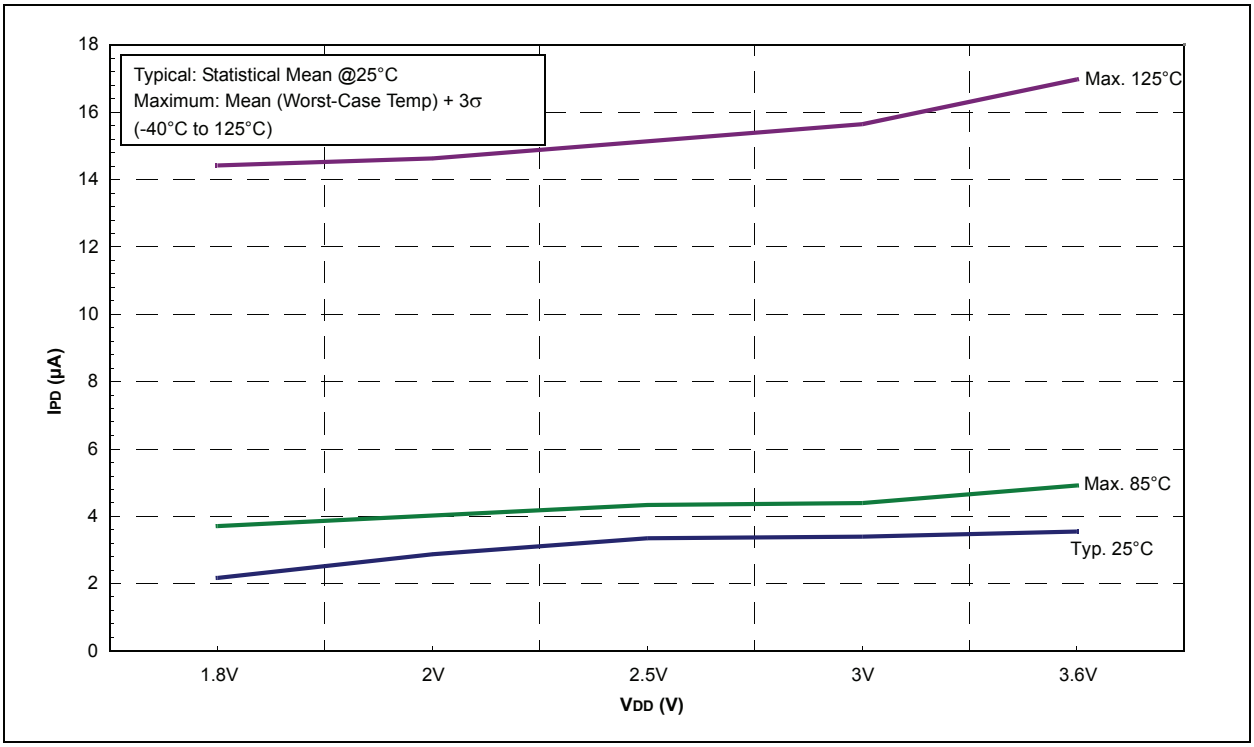


FIGURE 24-41: PIC16F722A/723A T1OSC 32 kHz IPD vs. VDD, VCAP = 0.1µF

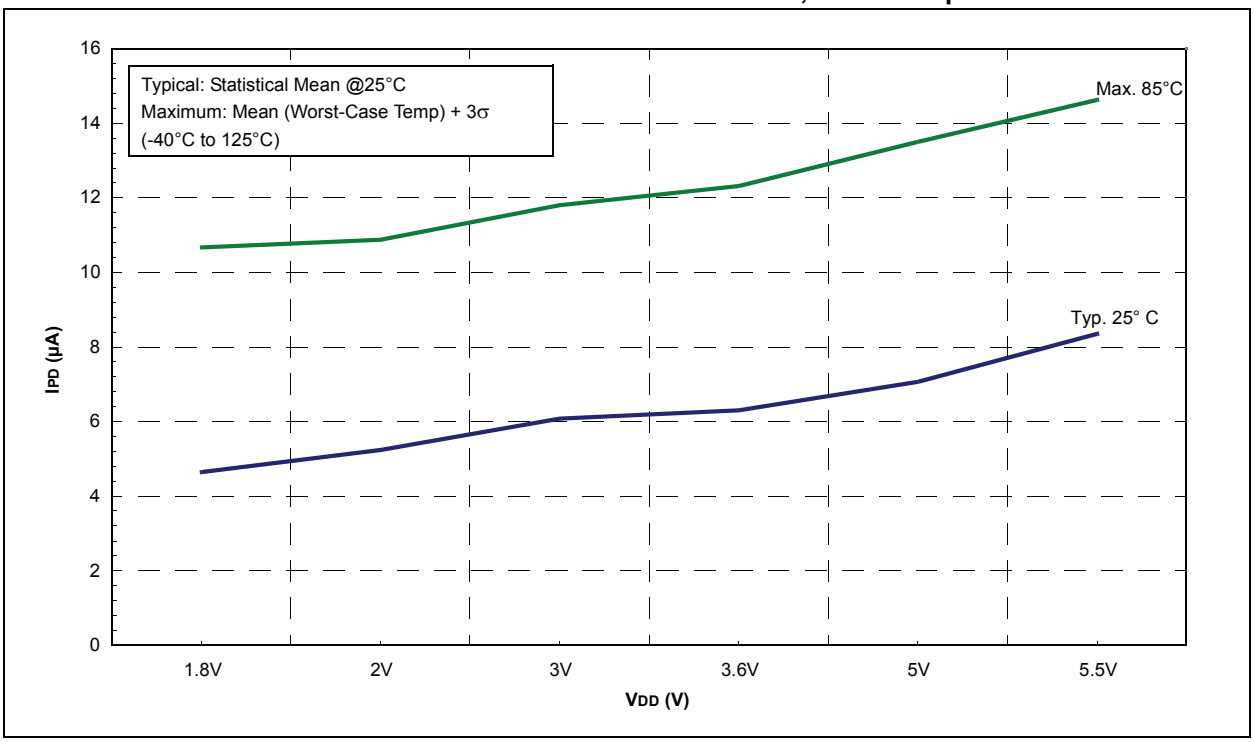


FIGURE 24-42: PIC16LF722A/723A T1OSC 32 kHz IPD vs. VDD

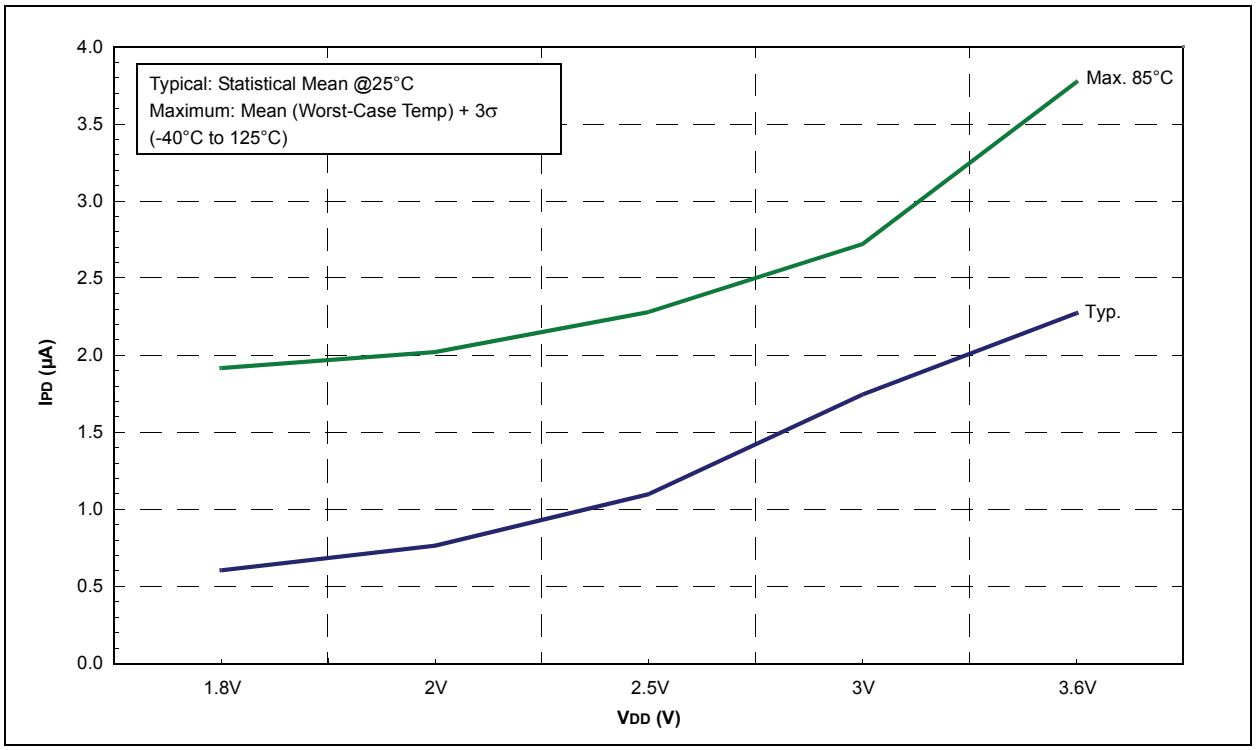


FIGURE 24-43: PIC16F722A/723A TYPICAL ADC IPD vs. VDD, VCAP = 0.1µF

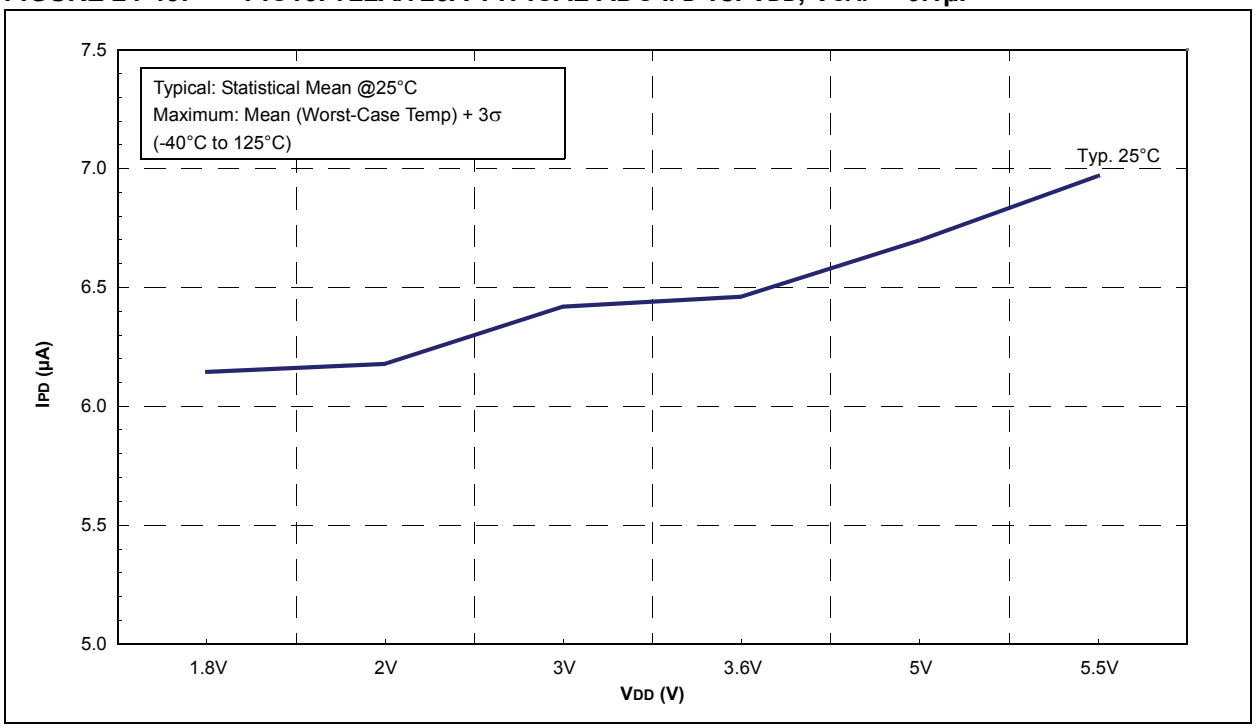


FIGURE 24-44: PIC16LF722A/723A TYPICAL ADC IPD vs. VDD

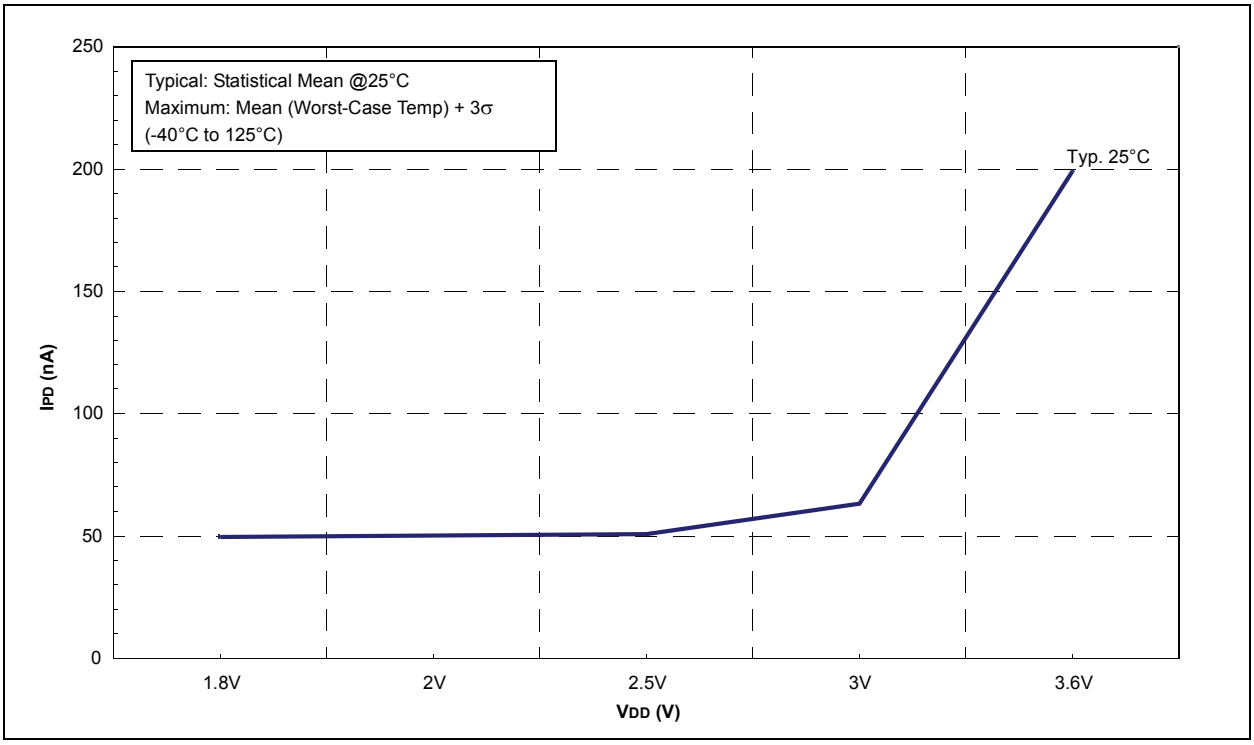


FIGURE 24-45: PIC16F722A/723A ADC IPD vs. VDD, VCAP = 0.1µF

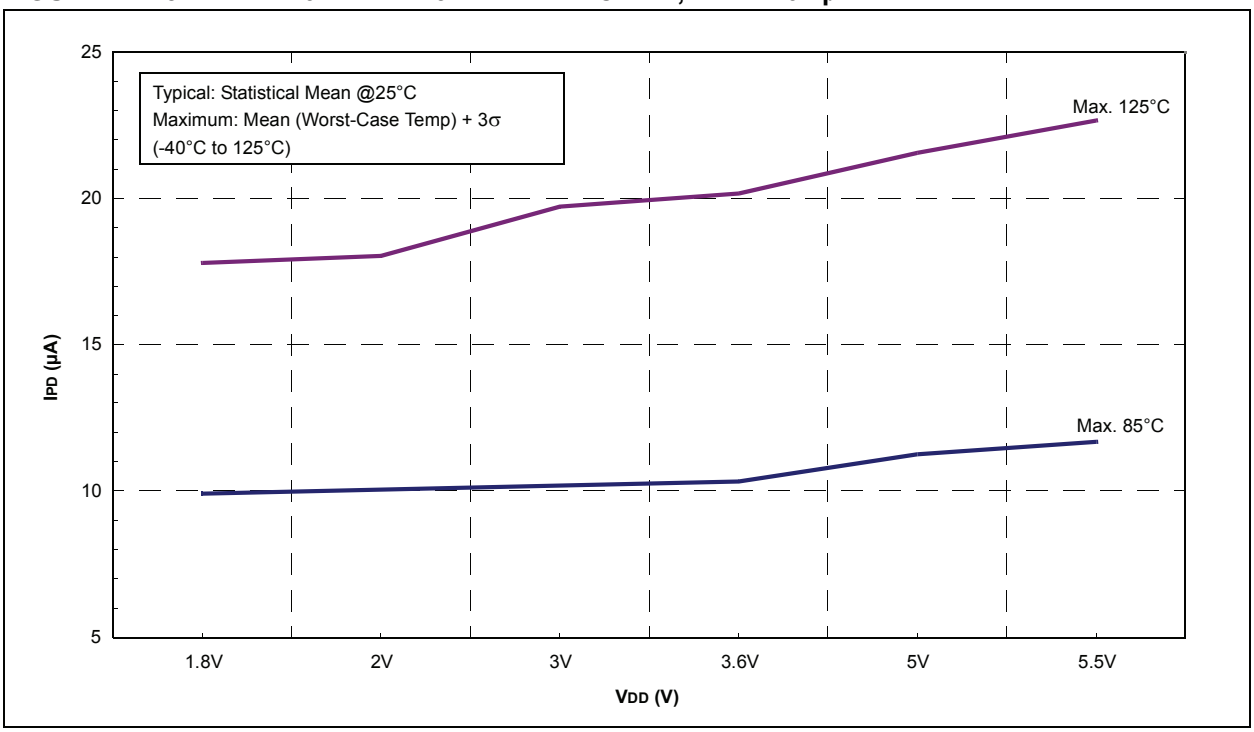


FIGURE 24-46: PIC16LF722A/723A ADC IPD vs. VDD

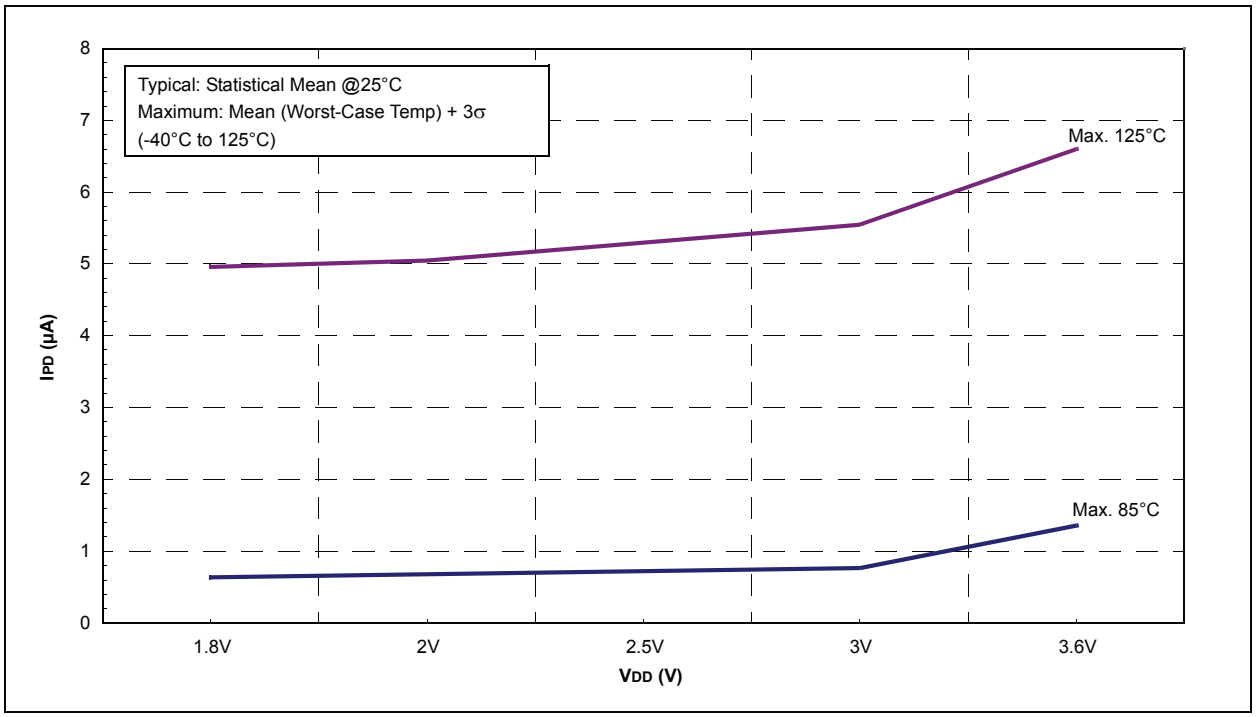


FIGURE 24-47: PIC16F722A/723A WDT IPD vs. VDD, VCAP = 0.1µF

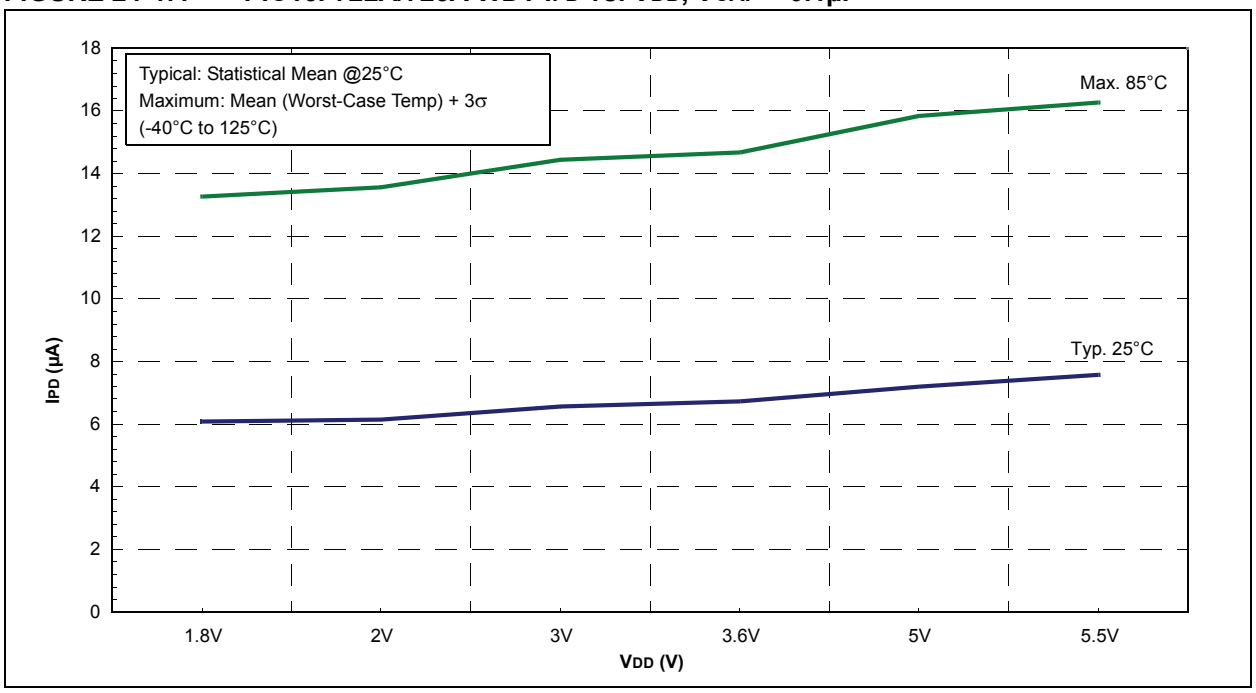


FIGURE 24-48: PIC16LF722A/723A WDT IPD vs. VDD

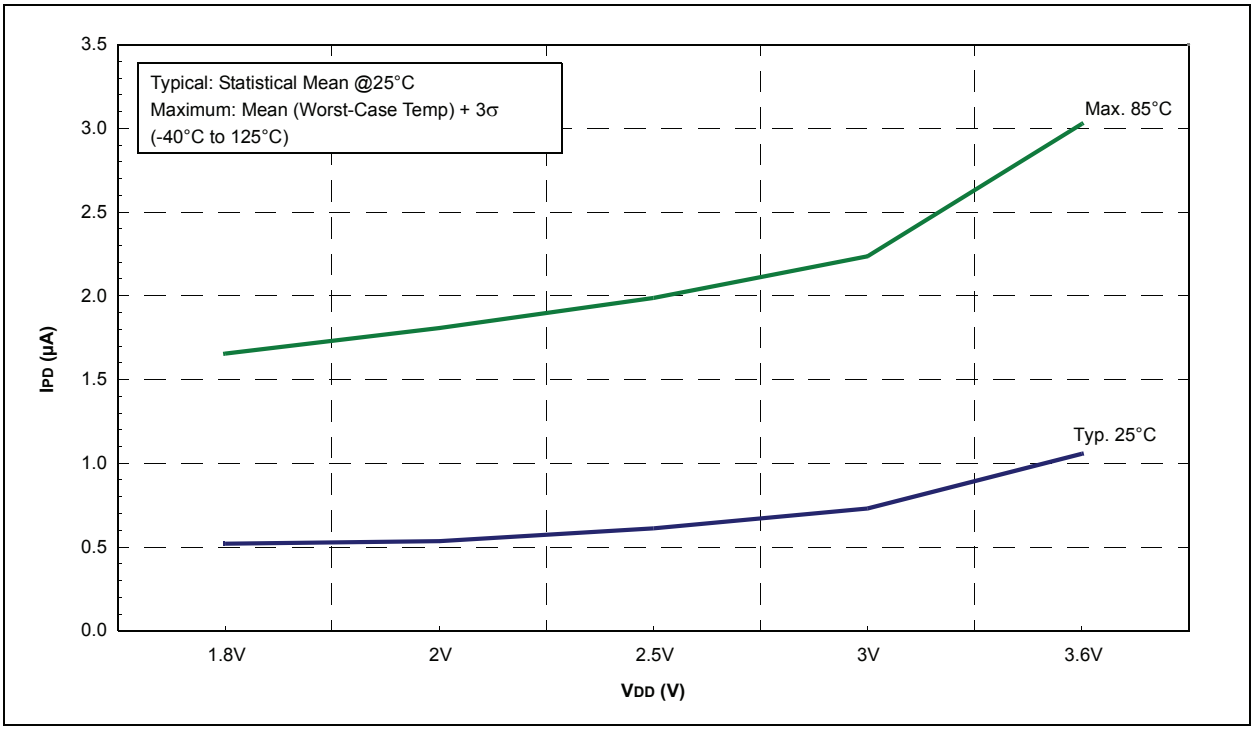


FIGURE 24-49: TTL INPUT THRESHOLD VIN vs. VDD OVER TEMPERATURE

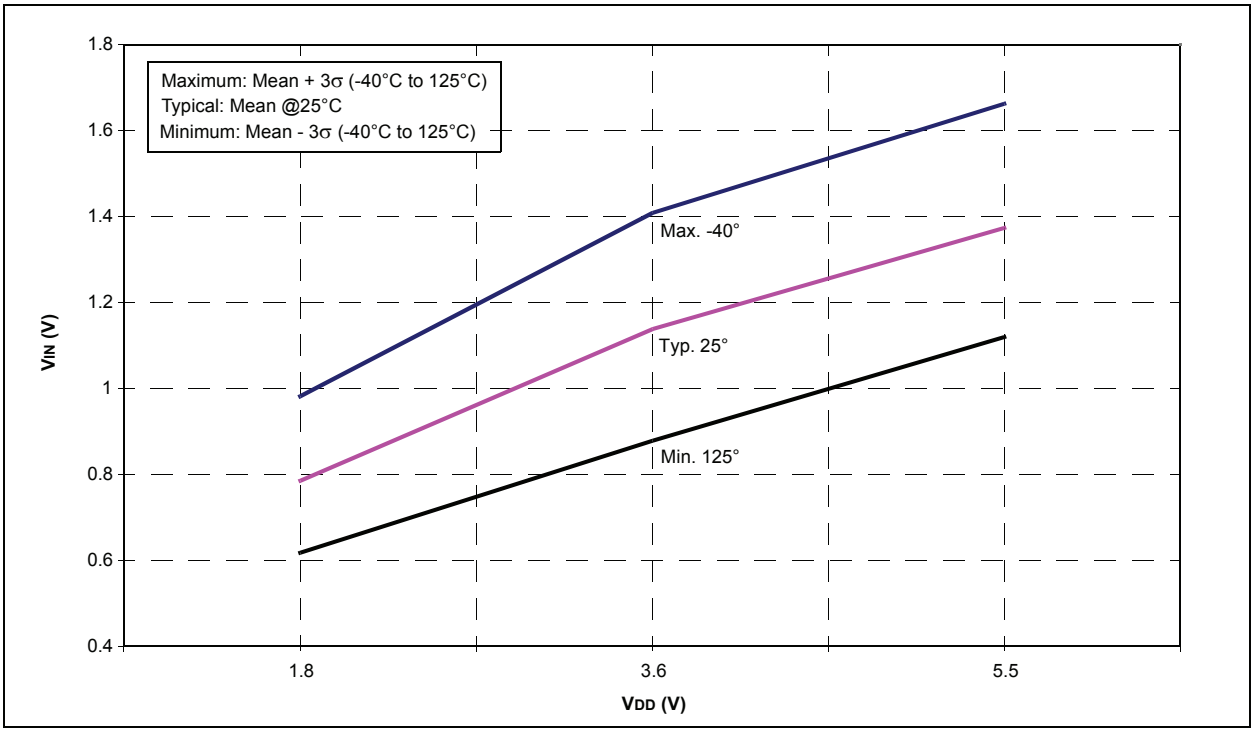


FIGURE 24-50: SCHMITT TRIGGER INPUT THRESHOLD VIN vs. VDD OVER TEMPERATURE

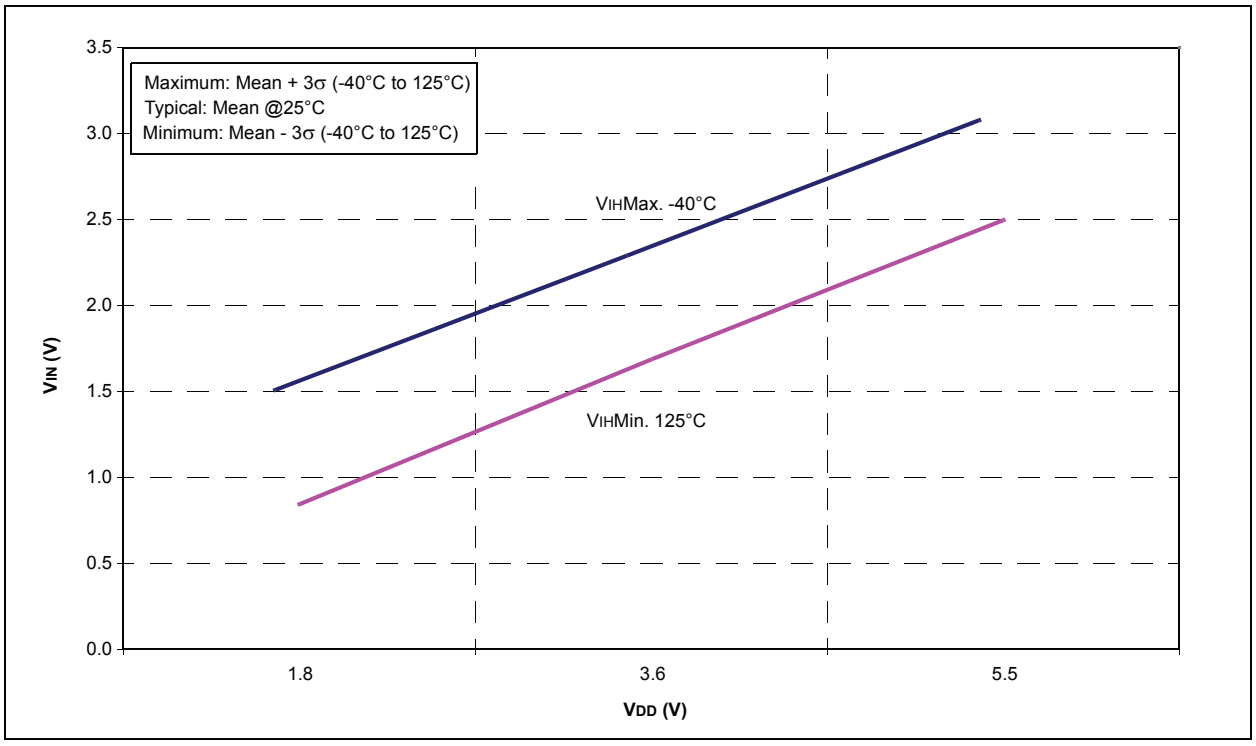


FIGURE 24-51: SCHMITT TRIGGER INPUT THRESHOLD VIN vs. VDD OVER TEMPERATURE

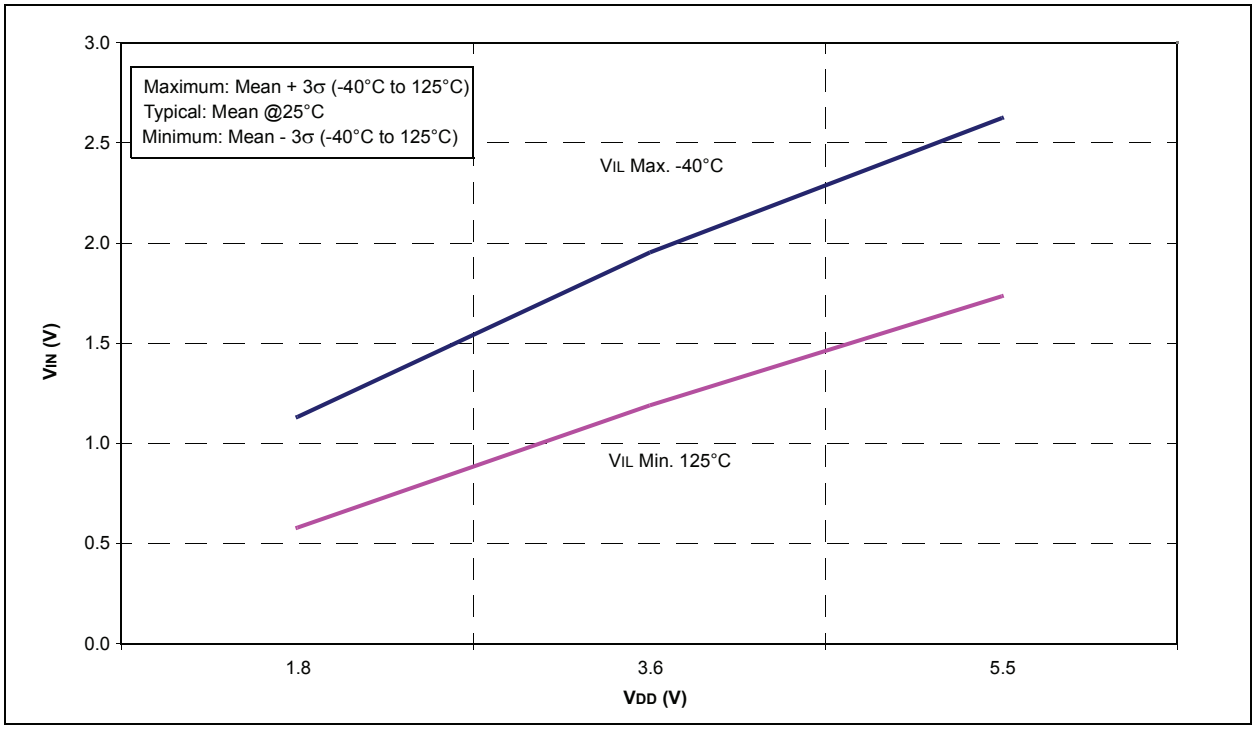


FIGURE 24-52: VOH vs. IOH OVER TEMPERATURE, VDD = 5.5V

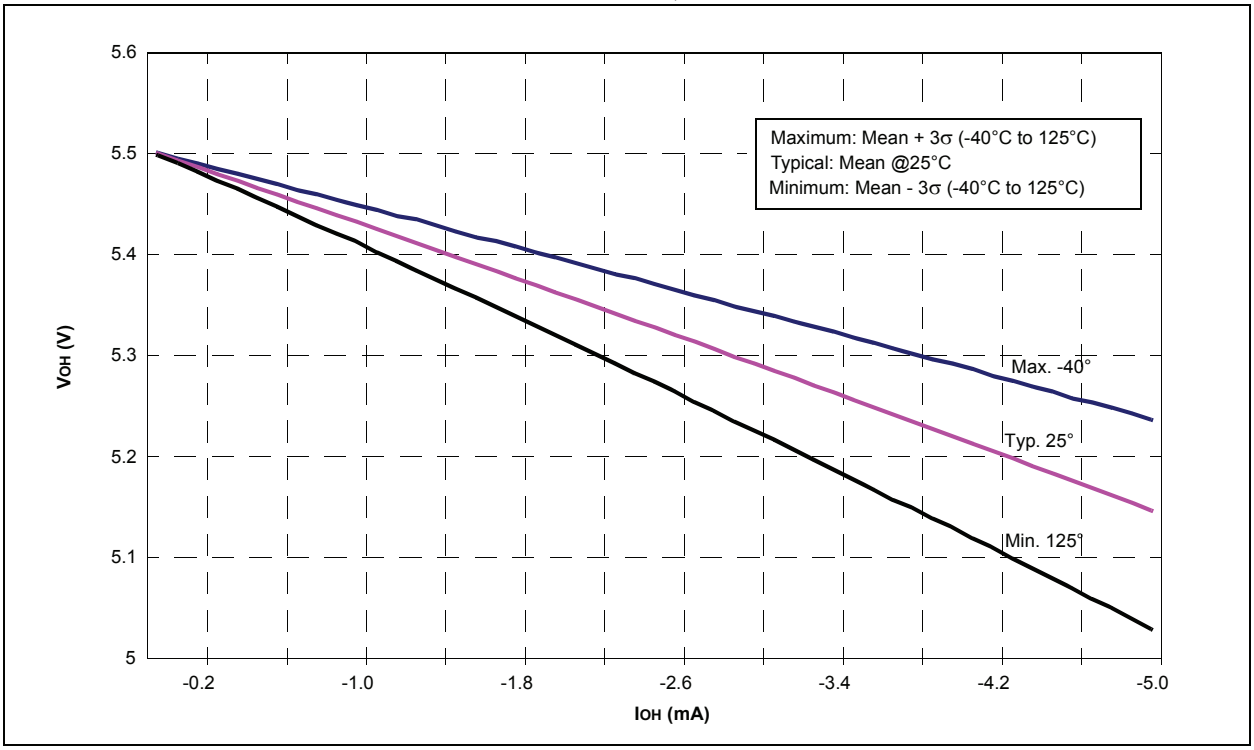


FIGURE 24-53: VOH vs. IOH OVER TEMPERATURE, VDD = 3.6V

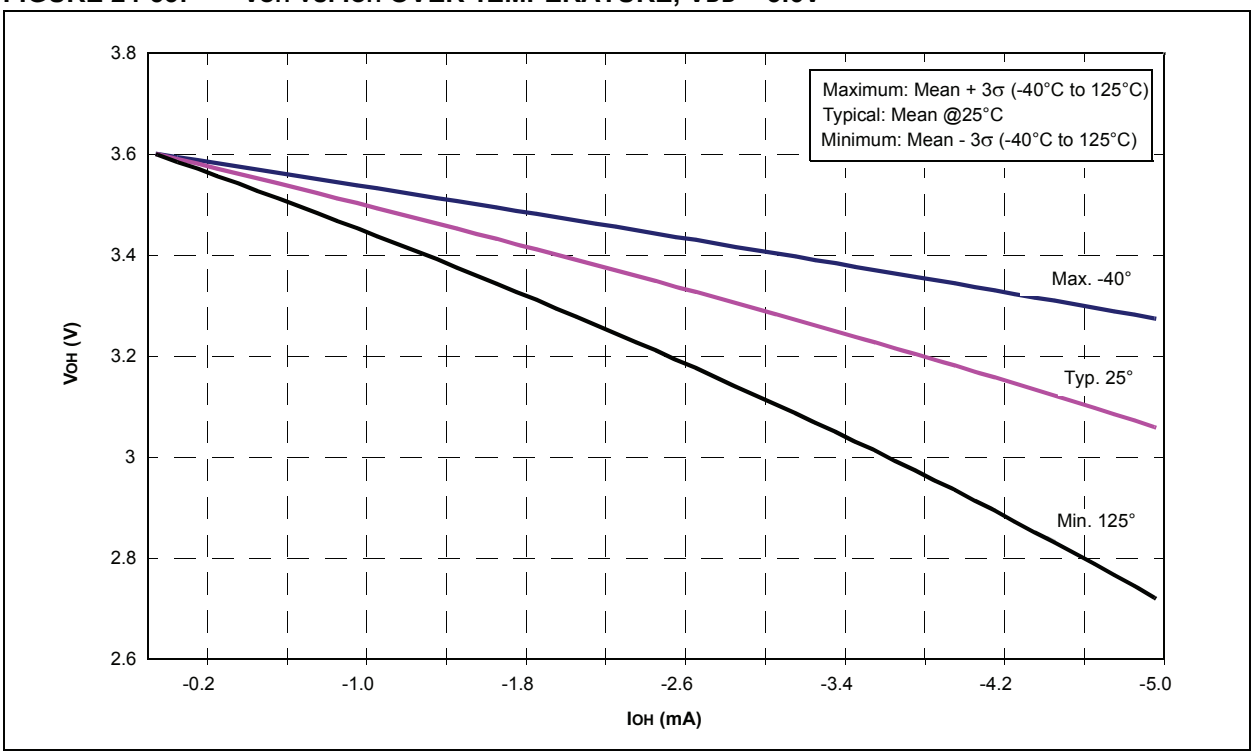


FIGURE 24-54: VOH vs. IOH OVER TEMPERATURE, VDD = 1.8V

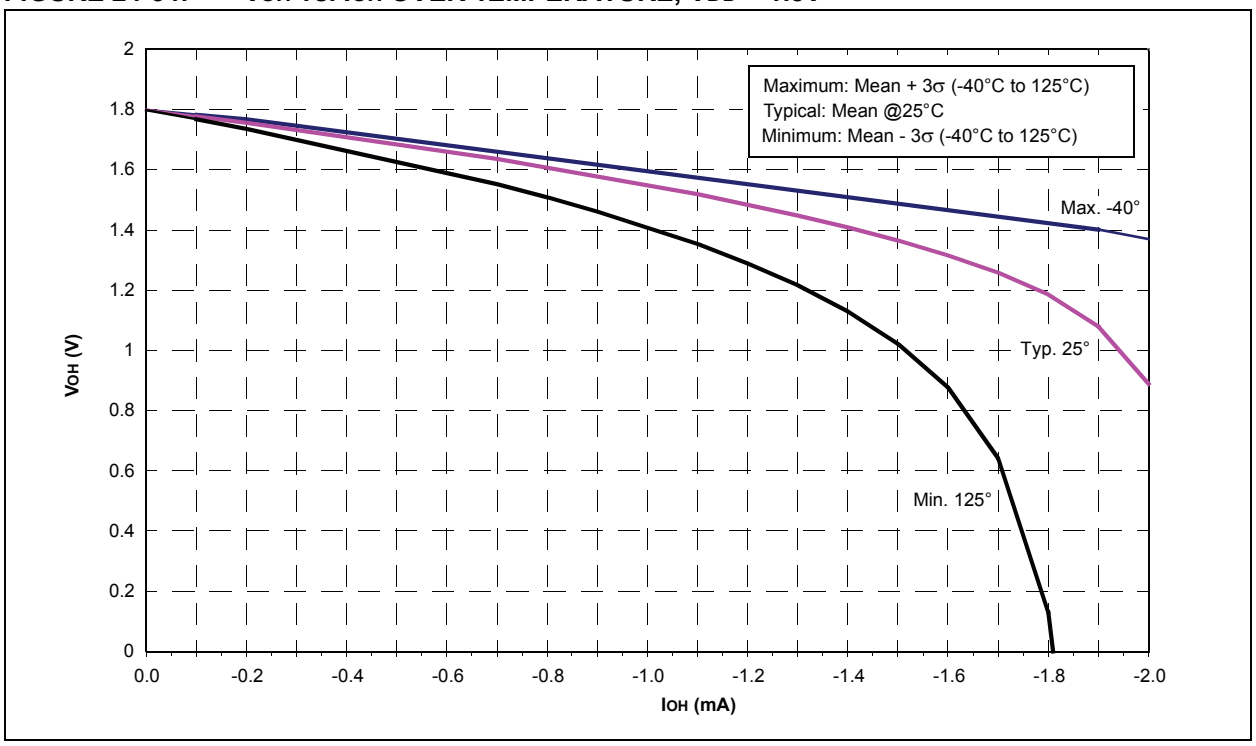


FIGURE 24-55: VOL vs. IOL OVER TEMPERATURE, VDD = 5.5V

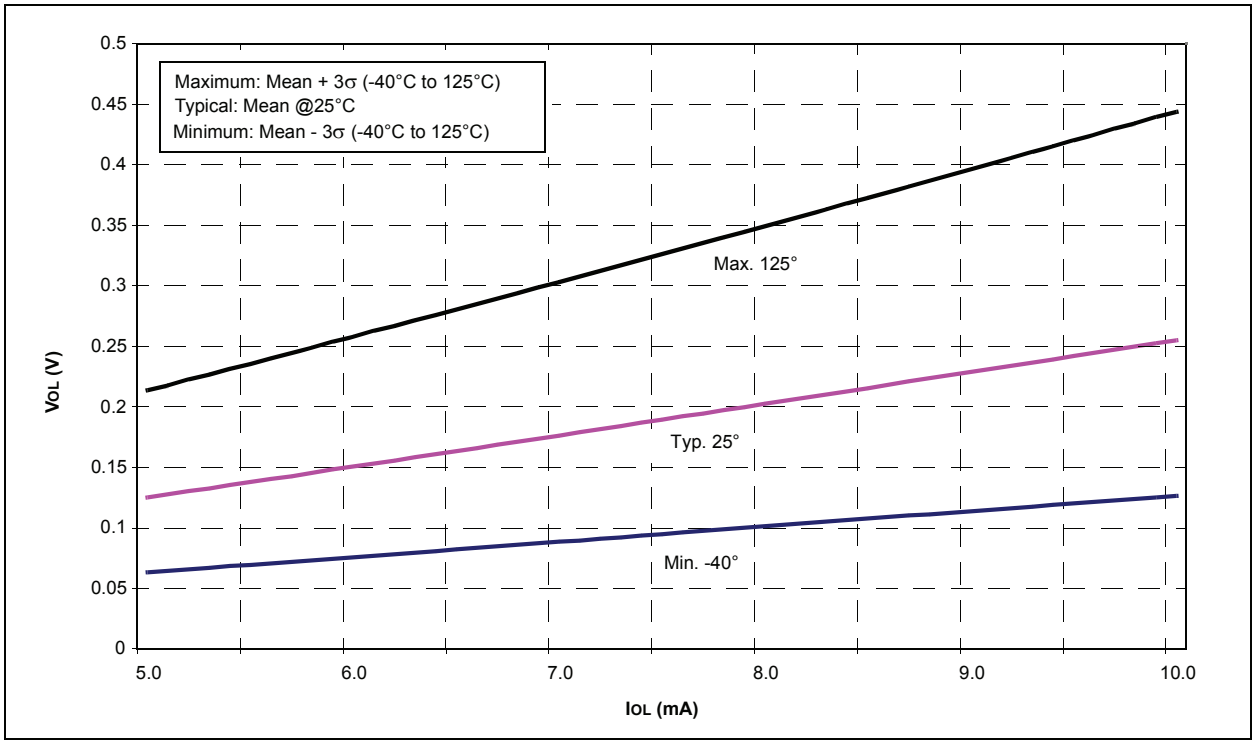


FIGURE 24-56: VOL vs. IOL OVER TEMPERATURE, VDD = 3.6

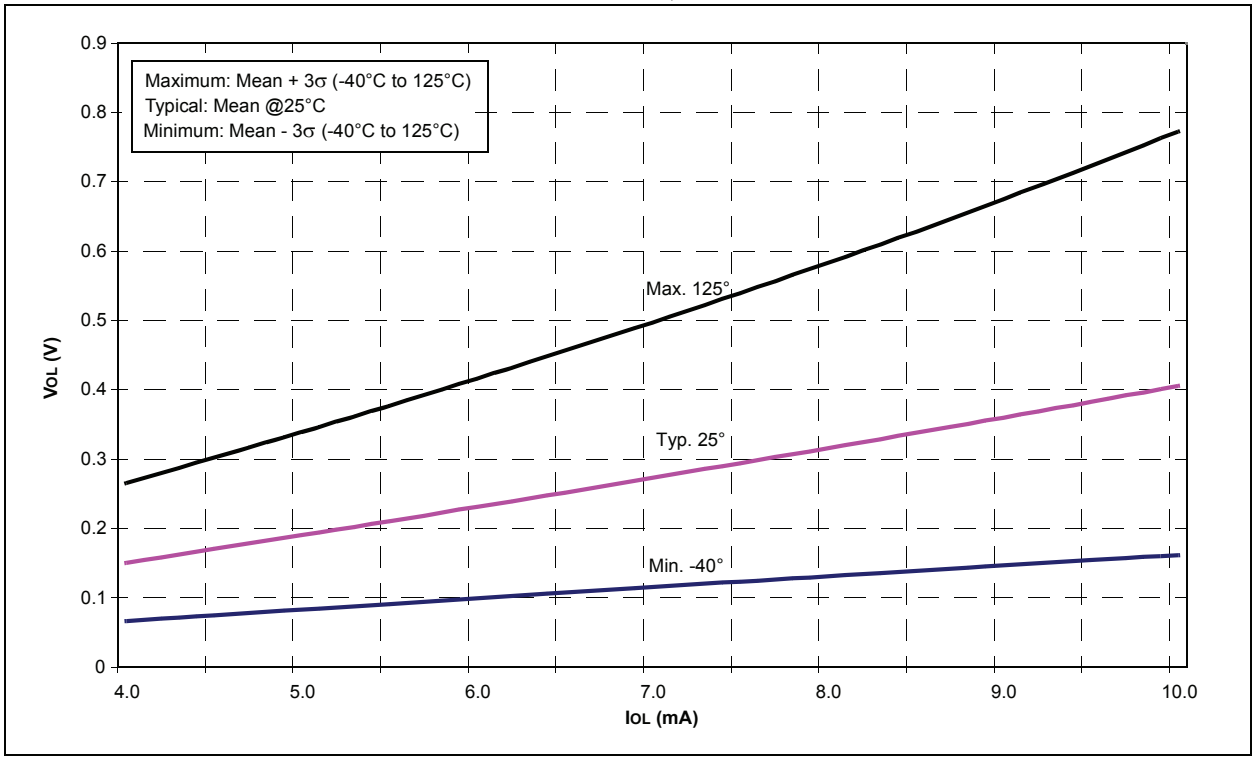


FIGURE 24-57: VOL vs. IOL OVER TEMPERATURE, VDD = 1.8V

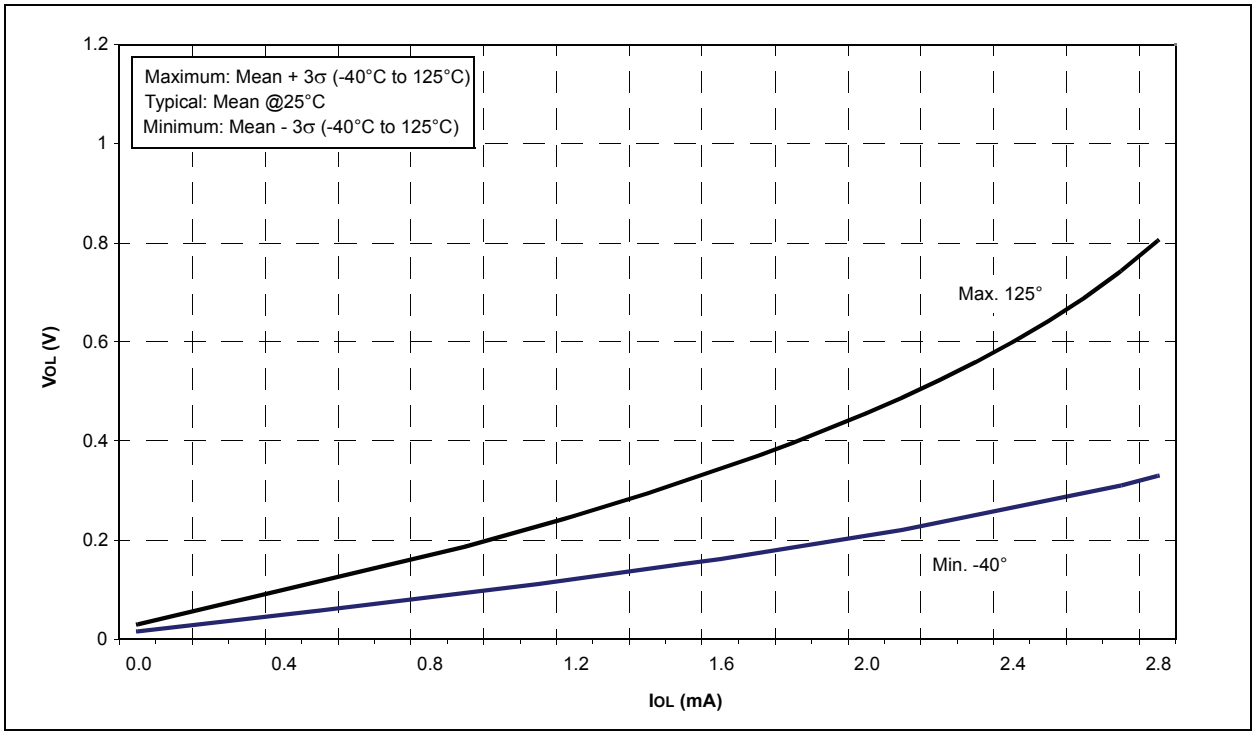


FIGURE 24-58: PIC16F722A/723A PWRT PERIOD

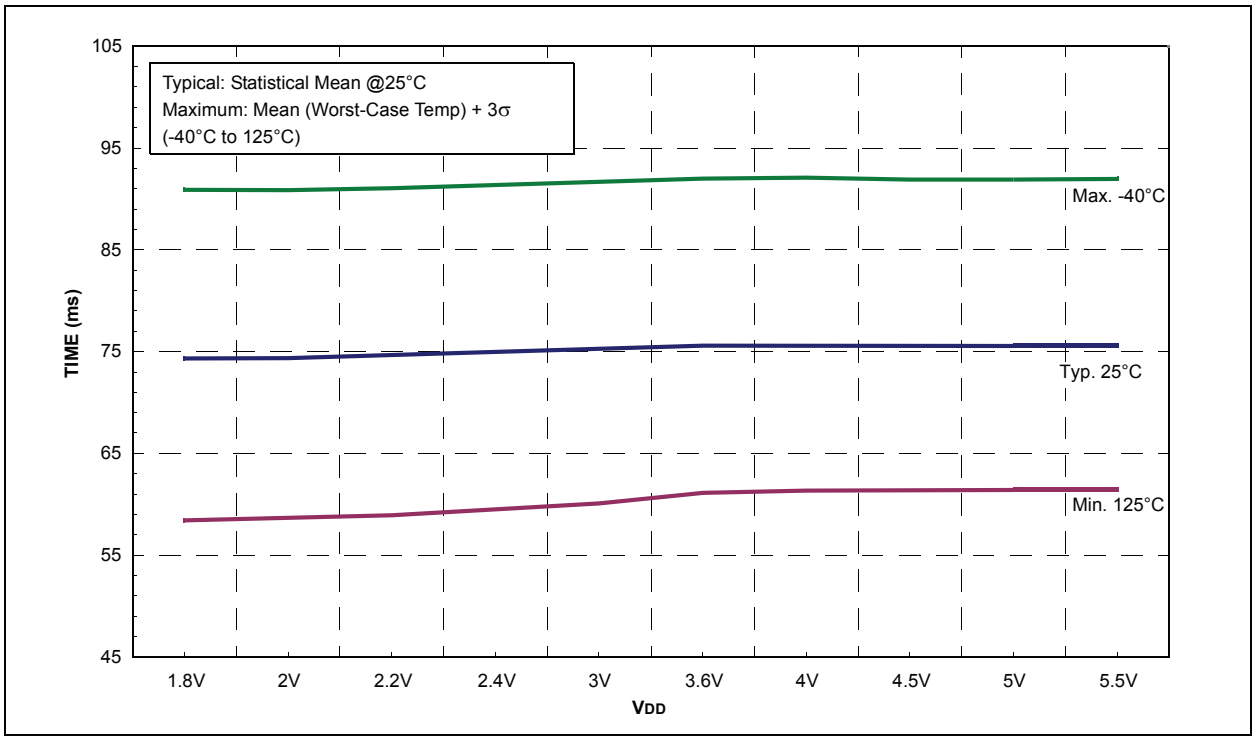


FIGURE 24-59: PIC16F722A/723A WDT TIME-OUT PERIOD

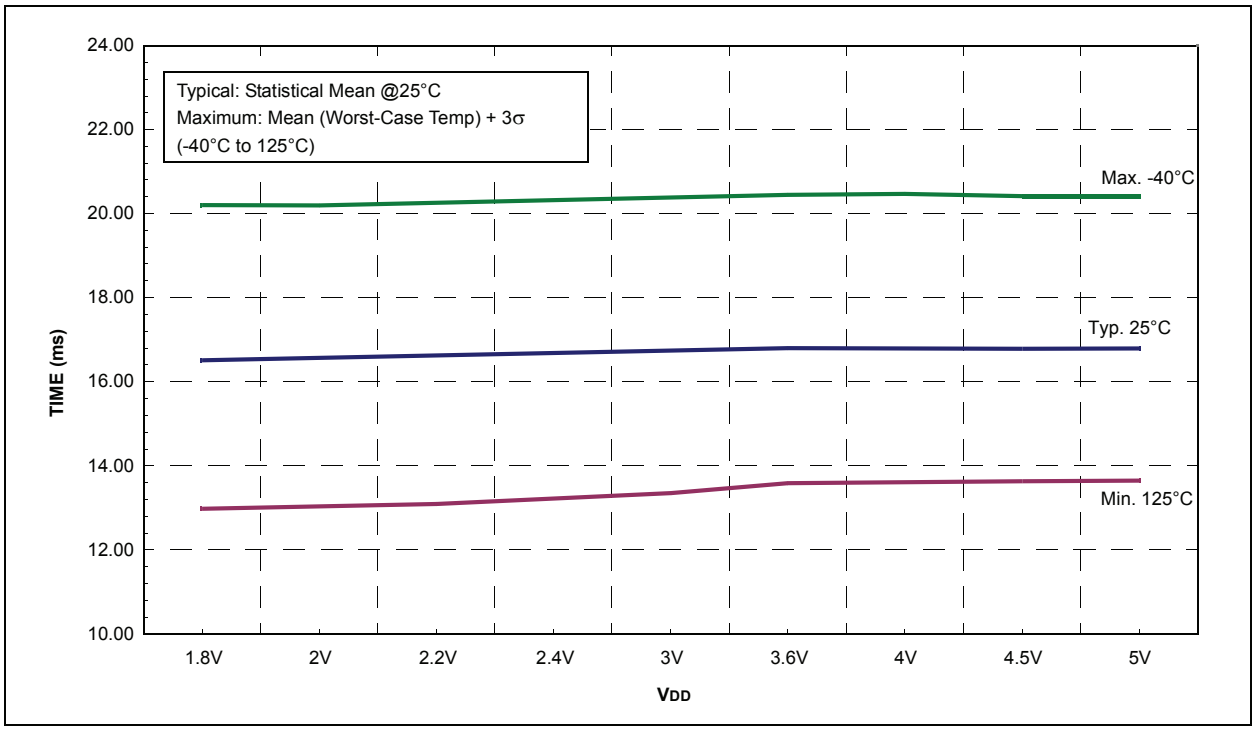


FIGURE 24-60: PIC16F722A/723A HFINTOSC WAKE-UP FROM SLEEP START-UP TIME

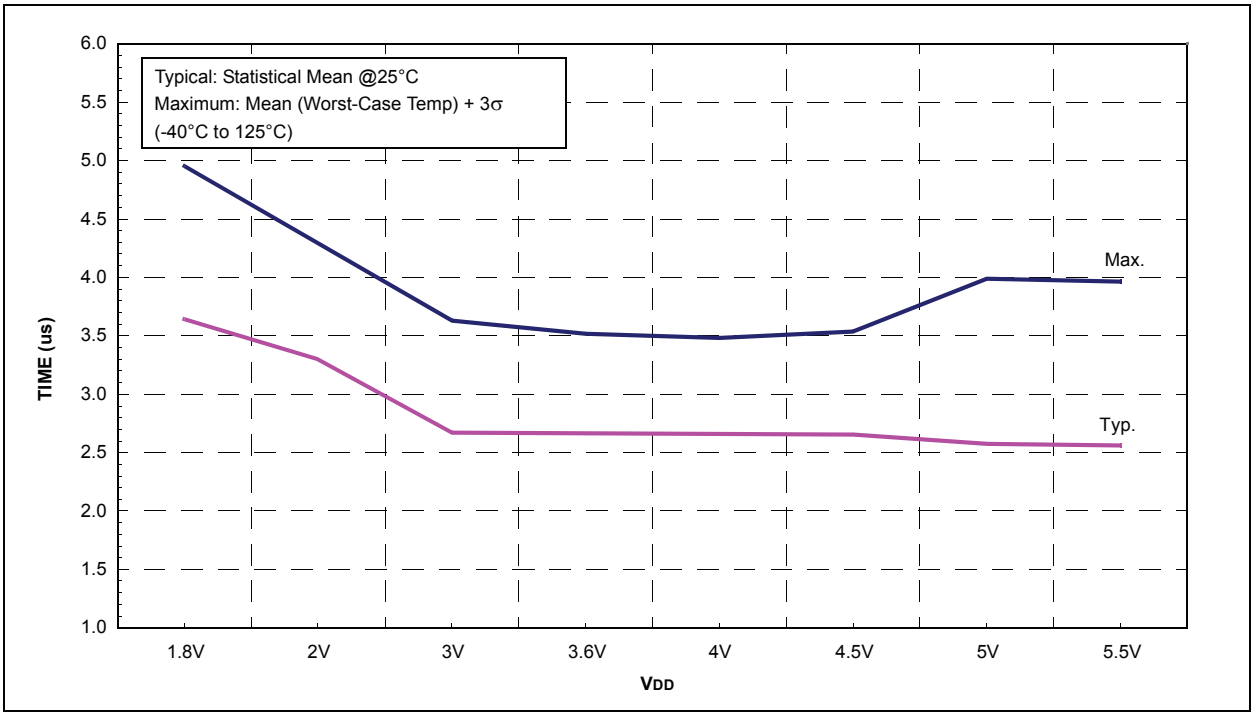


FIGURE 24-61: PIC16F722A/723A A/D INTERNAL RC OSCILLATOR PERIOD

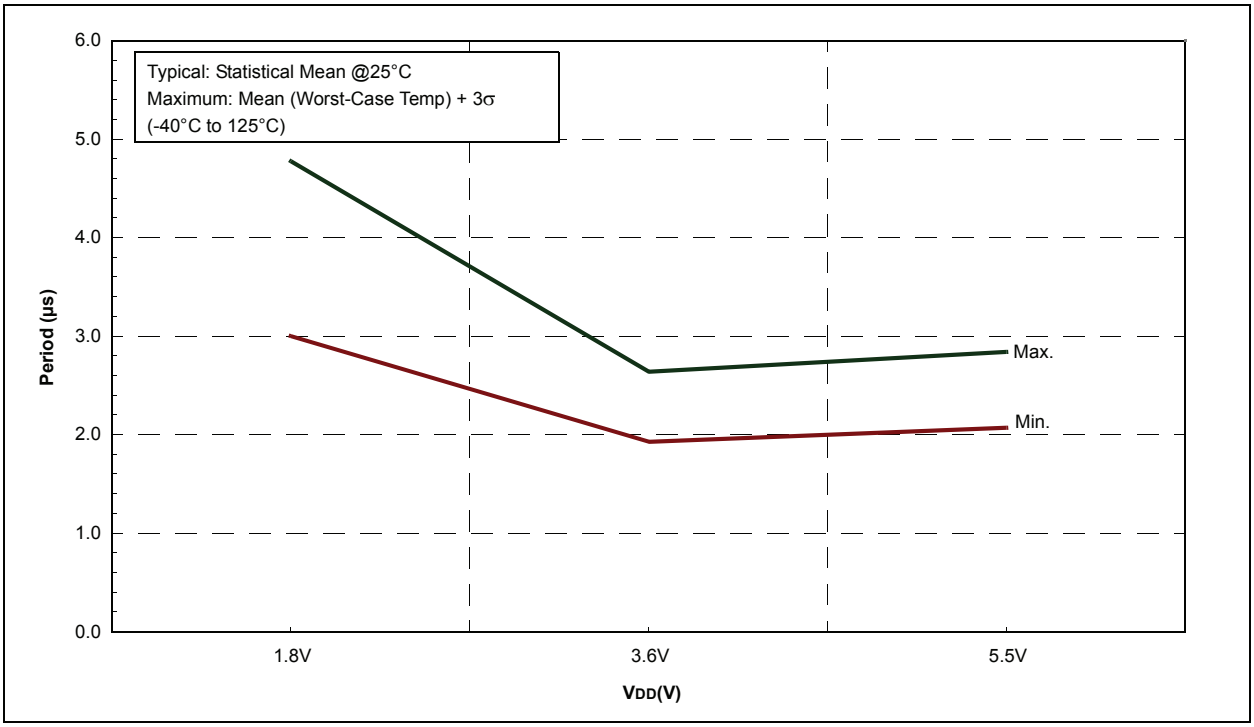


FIGURE 24-62: PIC16F722A/723A CAP SENSE OUTPUT CURRENT, POWER MODE = HIGH

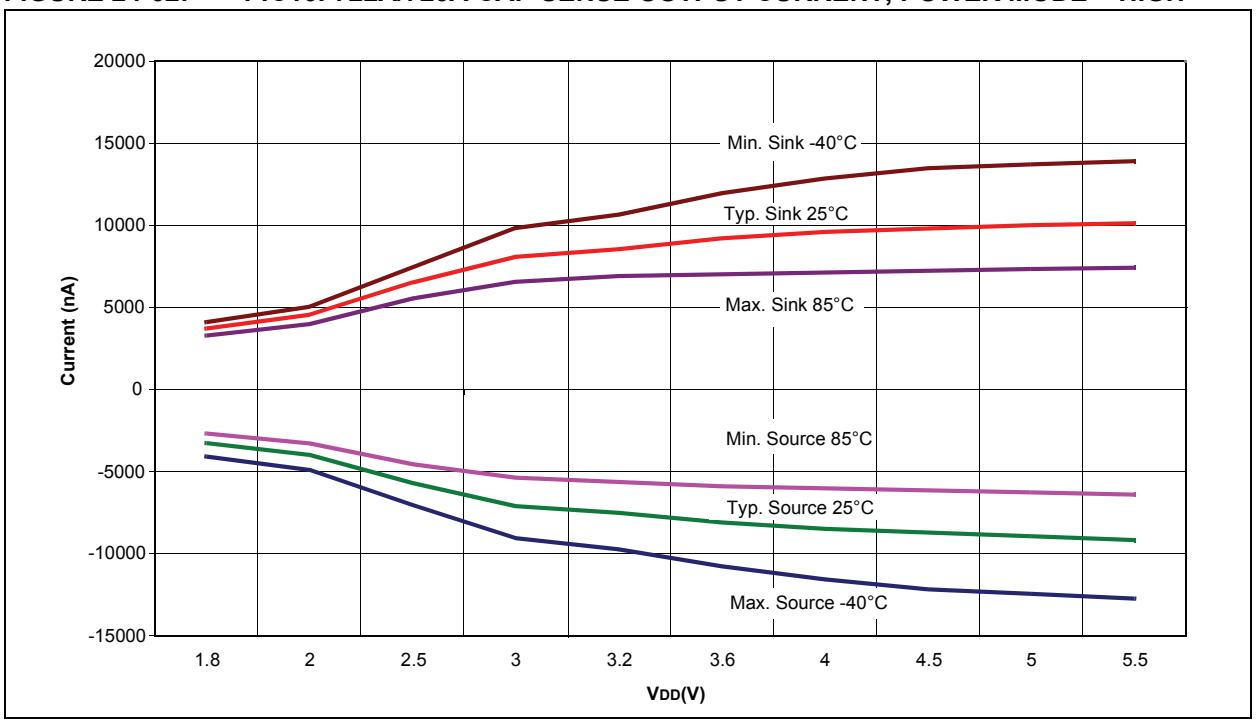


FIGURE 24-63: PIC16F722A/723A CAP SENSE OUTPUT CURRENT, POWER MODE = MEDIUM

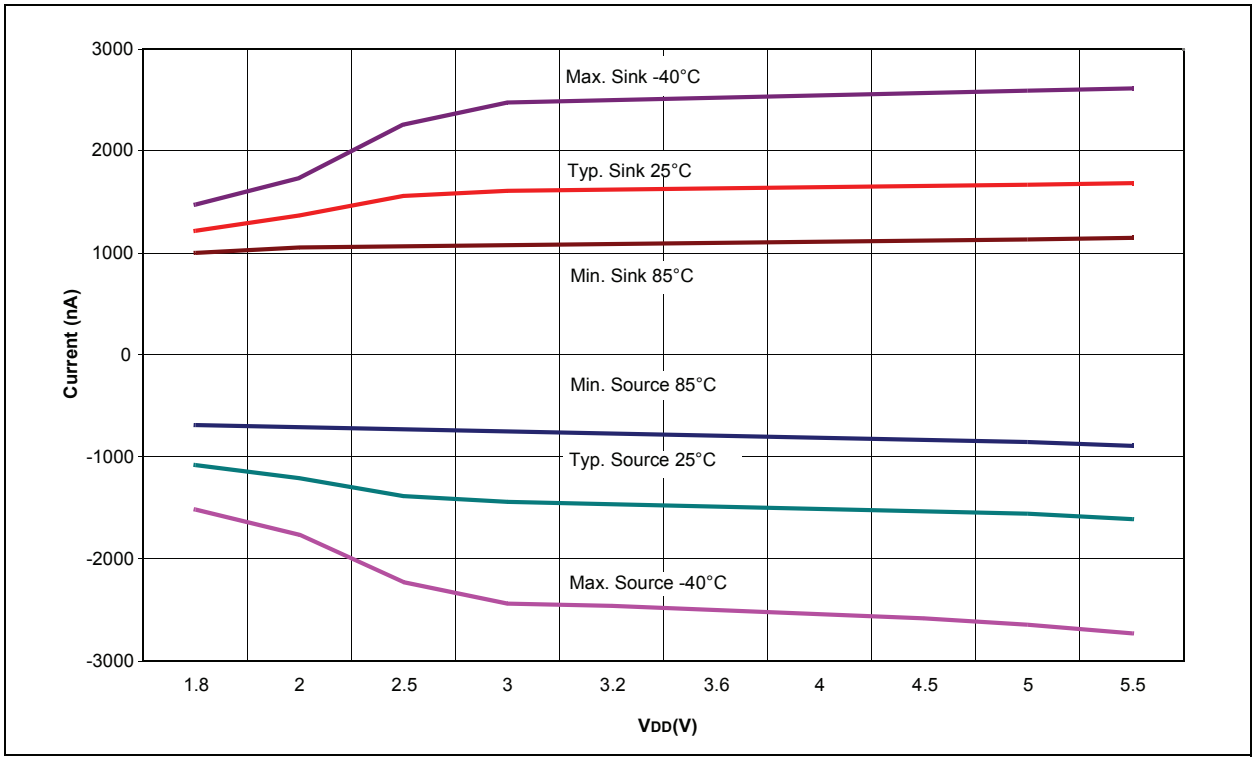


FIGURE 24-64: PIC16F722A/723A CAP SENSE OUTPUT CURRENT, POWER MODE = LOW

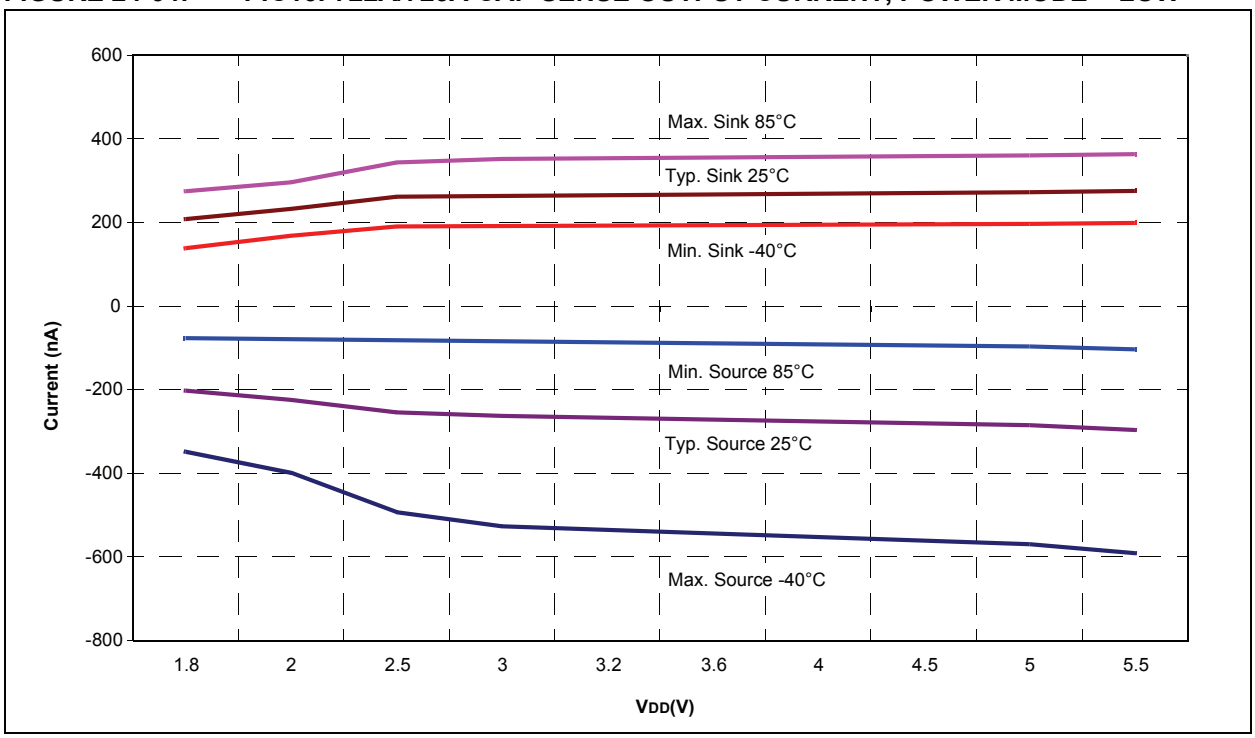


FIGURE 24-65: PIC16F722A/723A CAP SENSOR HYSTERESIS, POWER MODE = HIGH

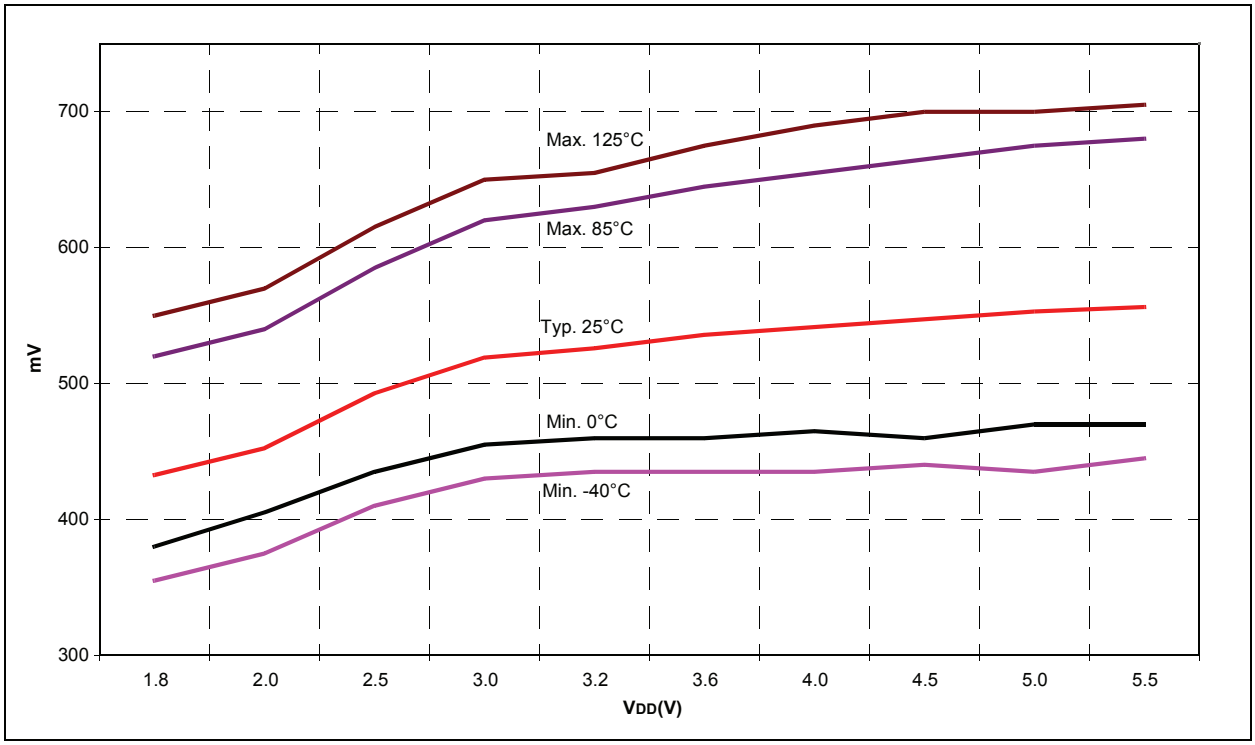


FIGURE 24-66: PIC16F722A/723A CAP SENSOR HYSTERESIS, POWER MODE = MEDIUM

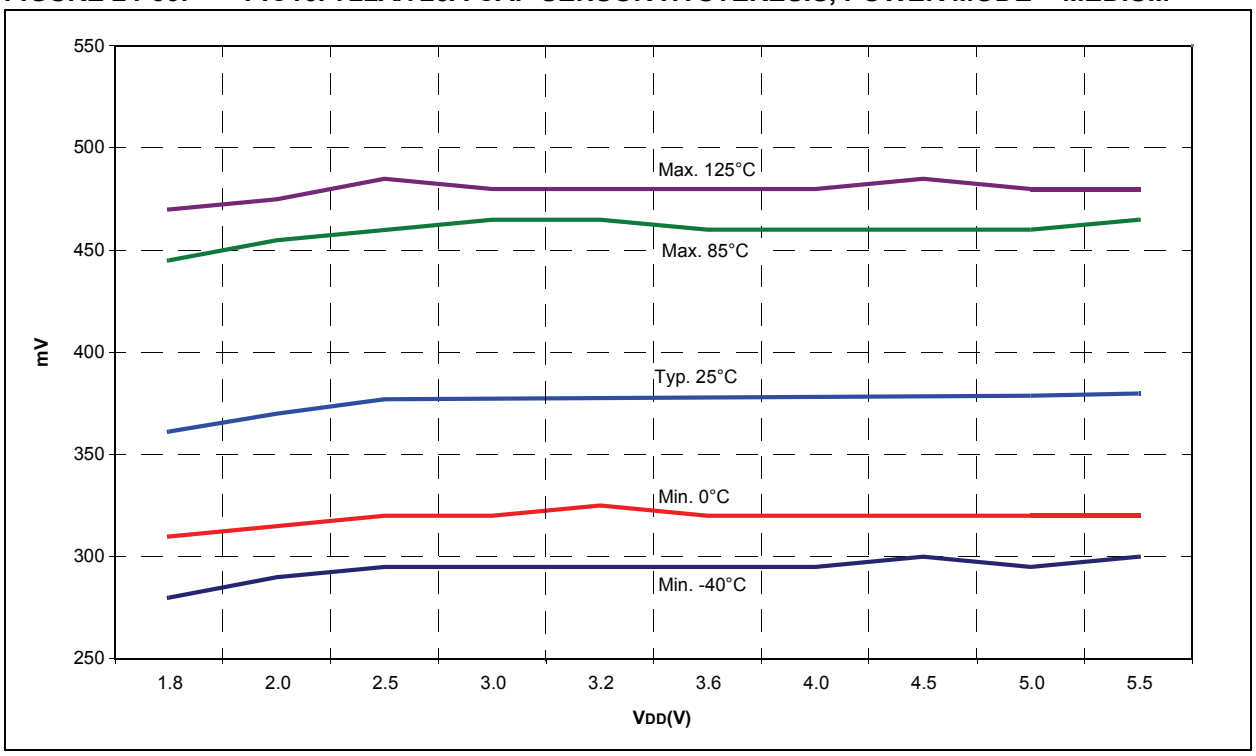


FIGURE 24-67: PIC16F722A/723A CAP SENSOR HYSTERESIS, POWER MODE = LOW

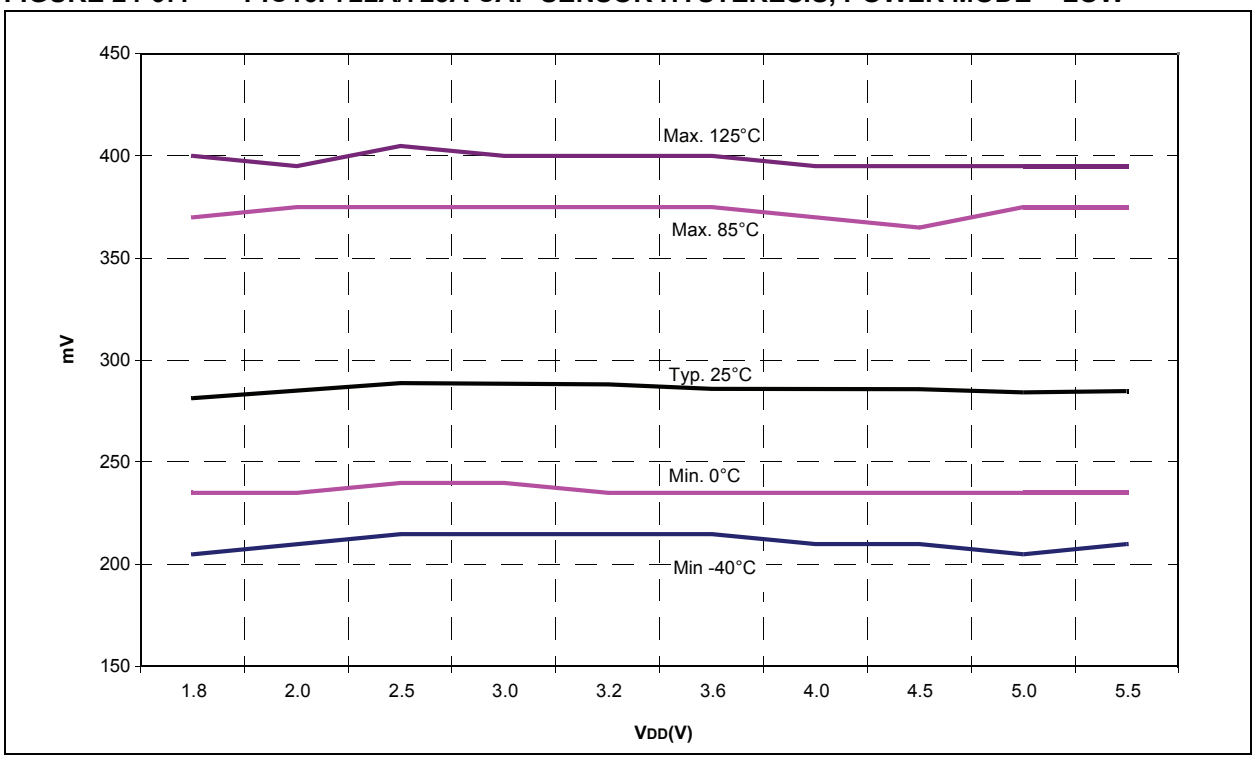


FIGURE 24-68: TYPICAL FVR (X1 AND X2) VS. SUPPLY VOLTAGE (V) NORMALIZED AT 3.0V

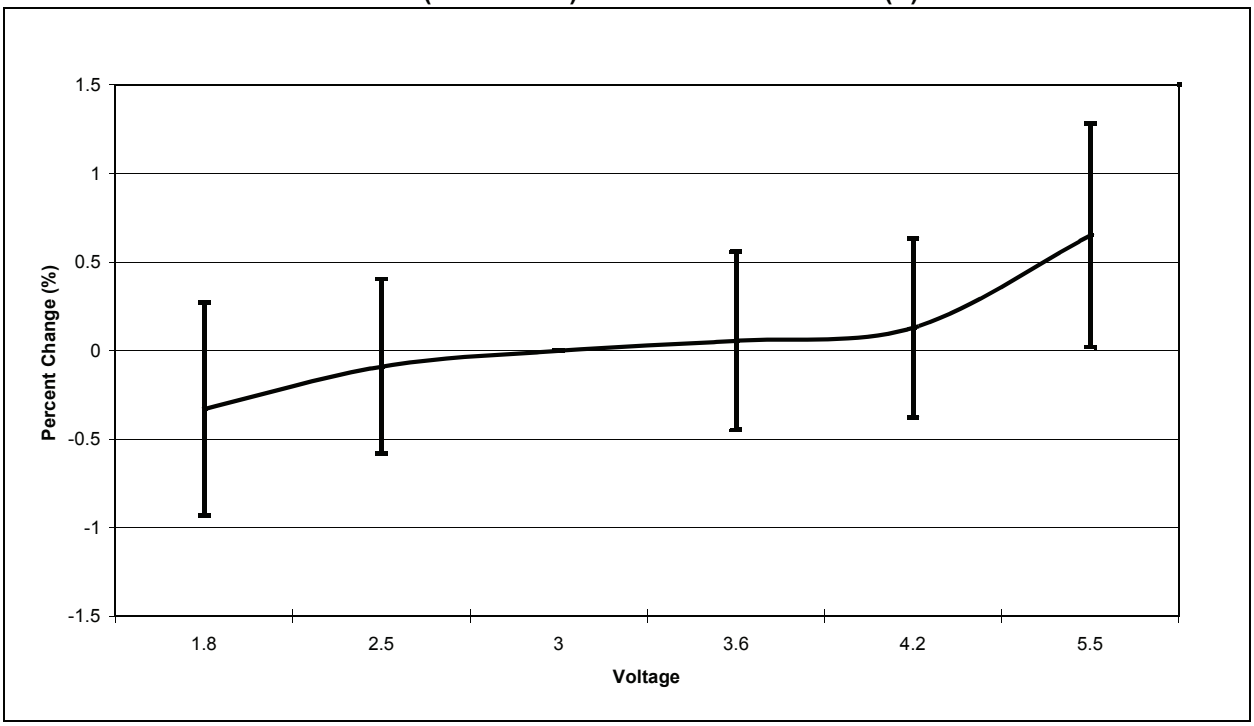
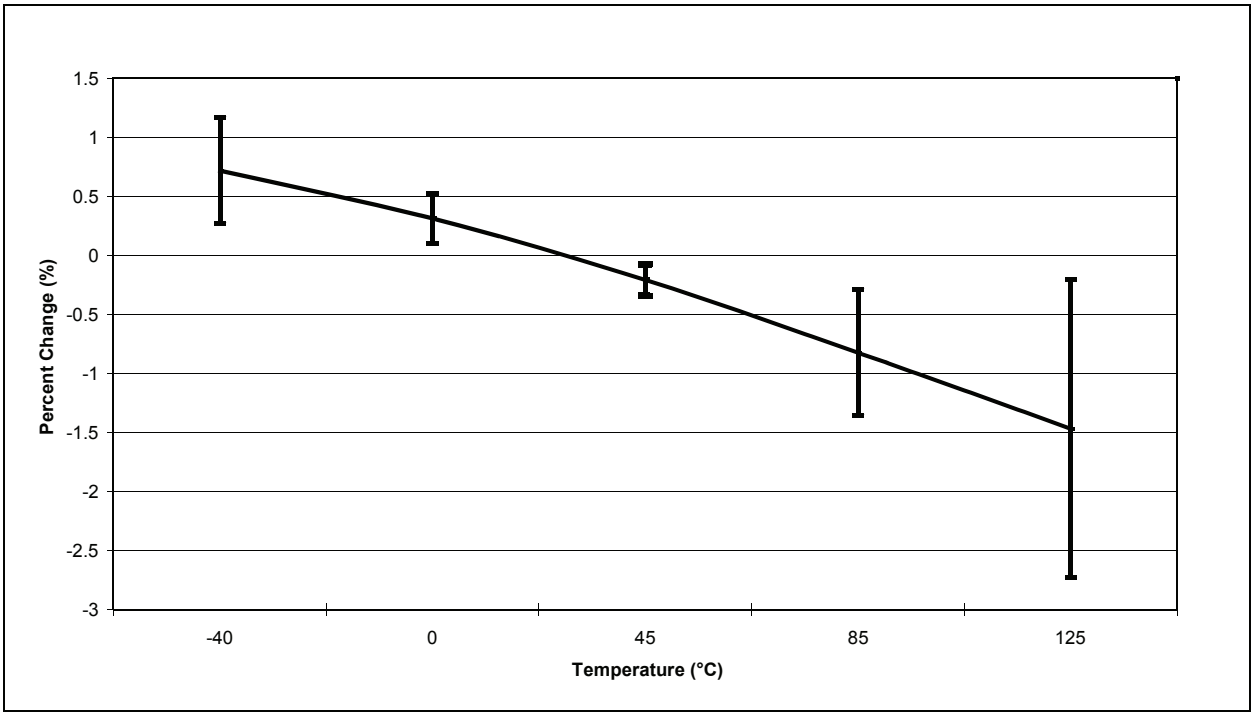


FIGURE 24-69: TYPICAL FVR CHANGE VS. TEMPERATURE NORMALIZED AT 25°C

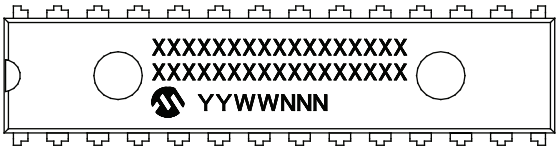


NOTES:

25.0 PACKAGING INFORMATION

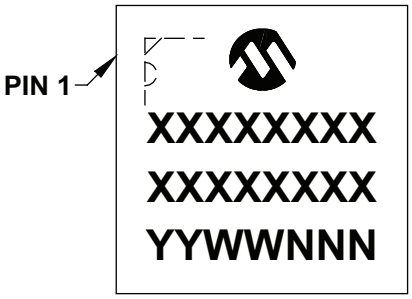
25.1 Package Marking Information

28-Lead SPDIP (.300")

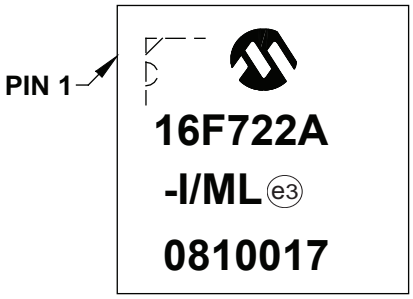


Example

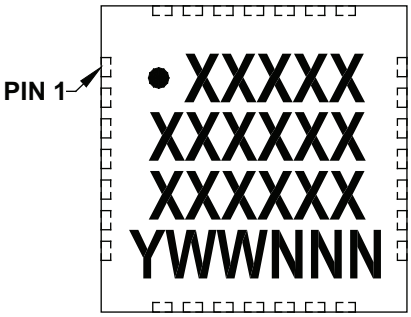
28-Lead QFN (6x6 mm)



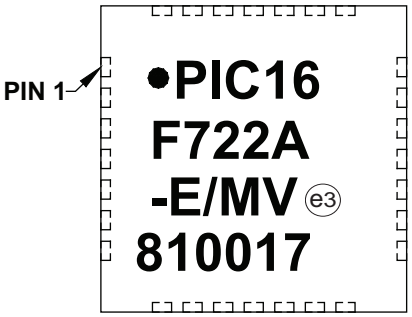
Example



28-Lead UQFN (4x4x0.5 mm)



Example



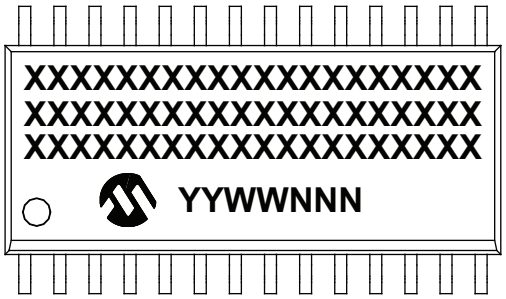
Legend:	XX...X	Customer-specific information
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week '01')
	NNN	Alphanumeric traceability code
	(e3)	Pb-free JEDEC designator for Matte Tin (Sn)
	*	This package is Pb-free. The Pb-free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

Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.

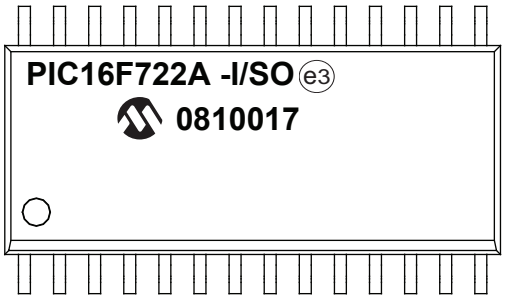
* Standard PICmicro® device marking consists of Microchip part number, year code, week code and traceability code. For PICmicro device marking beyond this, certain price adders apply. Please check with your Microchip Sales Office. For QTP devices, any special marking adders are included in QTP price.

25.1 Package Marking Information (Continued)

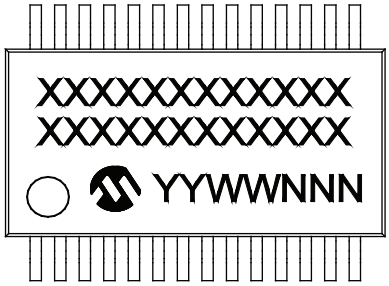
28-Lead SOIC (7.50 mm)



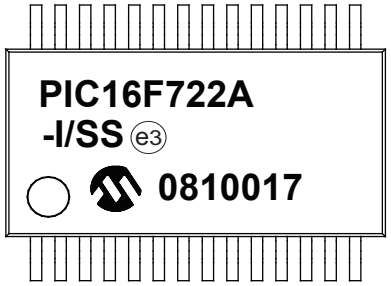
Example



28-Lead SSOP (5.30 mm)



Example



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

Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.

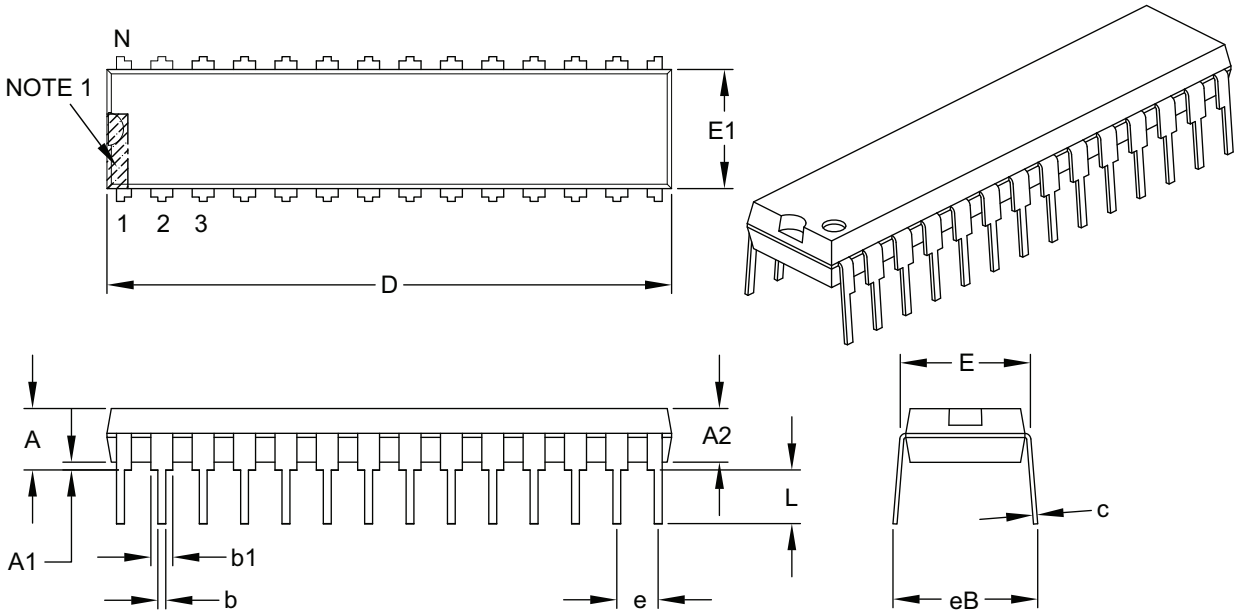
* Standard PICmicro® device marking consists of Microchip part number, year code, week code and traceability code. For PICmicro device marking beyond this, certain price adders apply. Please check with your Microchip Sales Office. For QTP devices, any special marking adders are included in QTP price.

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:

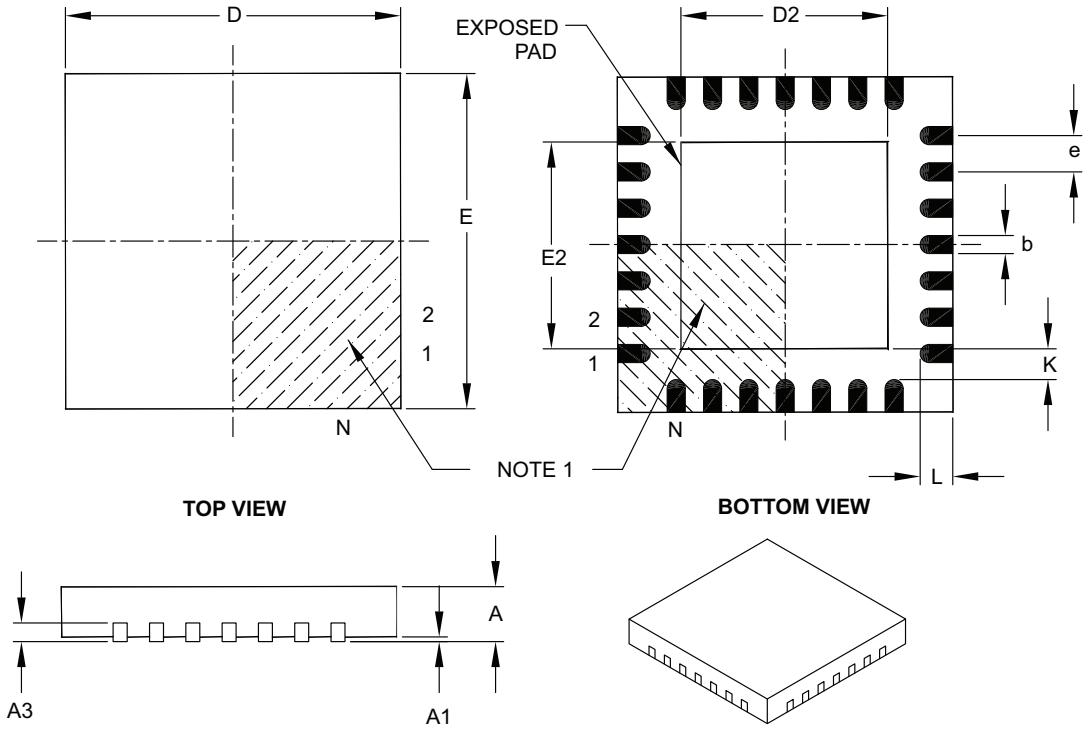
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic.
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

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

Microchip Technology Drawing C04-070B

28-Lead Plastic Quad Flat, No Lead Package (ML) – 6x6 mm Body [QFN] with 0.55 mm Contact Length

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



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		28	
Pitch	e		0.65 BSC	
Overall Height	A	0.80	0.90	1.00
Standoff	A1	0.00	0.02	0.05
Contact Thickness	A3		0.20 REF	
Overall Width	E		6.00 BSC	
Exposed Pad Width	E2	3.65	3.70	4.20
Overall Length	D		6.00 BSC	
Exposed Pad Length	D2	3.65	3.70	4.20
Contact Width	b	0.23	0.30	0.35
Contact Length	L	0.50	0.55	0.70
Contact-to-Exposed Pad	K	0.20	–	–

Notes:

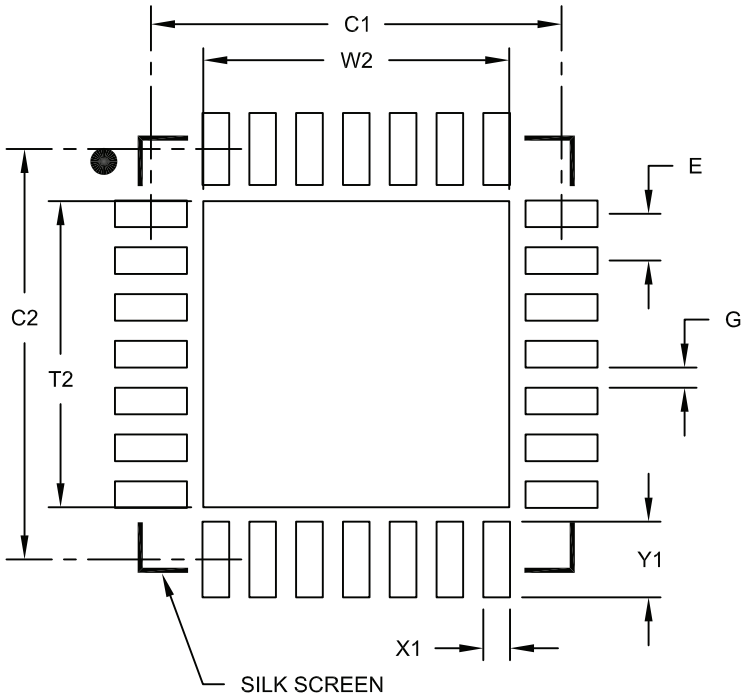
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package is saw singulated.
3. Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-105B

28-Lead Plastic Quad Flat, No Lead Package (ML) – 6x6 mm Body [QFN] with 0.55 mm Contact Length

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



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		0.65 BSC	
Optional Center Pad Width	W2			4.25
Optional Center Pad Length	T2			4.25
Contact Pad Spacing	C1		5.70	
Contact Pad Spacing	C2		5.70	
Contact Pad Width (X28)	X1			0.37
Contact Pad Length (X28)	Y1			1.00
Distance Between Pads	G	0.20		

Notes:

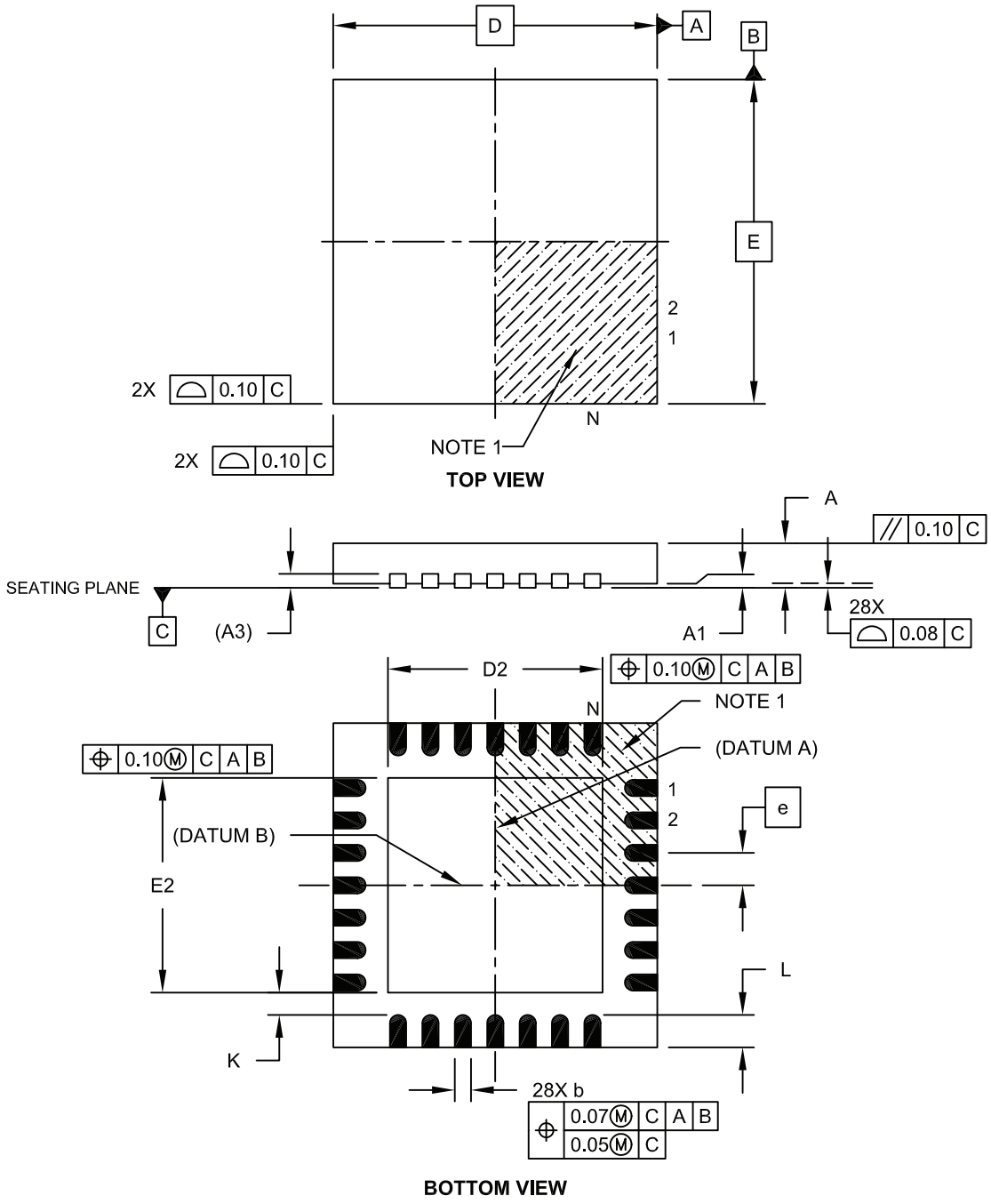
1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-2105A

28-Lead Plastic Ultra Thin Quad Flat, No Lead Package (MV) – 4x4x0.5 mm Body [UQFN]

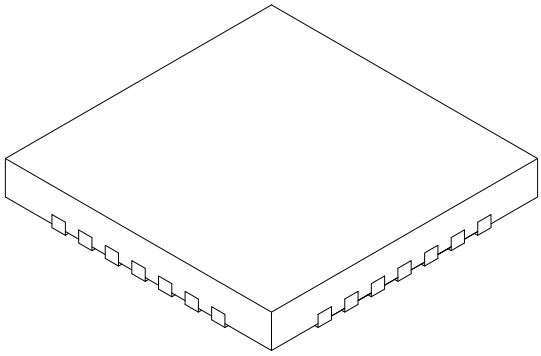
Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Microchip Technology Drawing C04-152A Sheet 1 of 2

28-Lead Plastic Ultra Thin Quad Flat, No Lead Package (MV) – 4x4x0.5 mm Body [UQFN]

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



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		28	
Pitch	e		0.40 BSC	
Overall Height	A	0.45	0.50	0.55
Standoff	A1	0.00	0.02	0.05
Contact Thickness	A3		0.127 REF	
Overall Width	E		4.00 BSC	
Exposed Pad Width	E2	2.55	2.65	2.75
Overall Length	D		4.00 BSC	
Exposed Pad Length	D2	2.55	2.65	2.75
Contact Width	b	0.15	0.20	0.25
Contact Length	L	0.30	0.40	0.50
Contact-to-Exposed Pad	K	0.20	-	-

Notes:

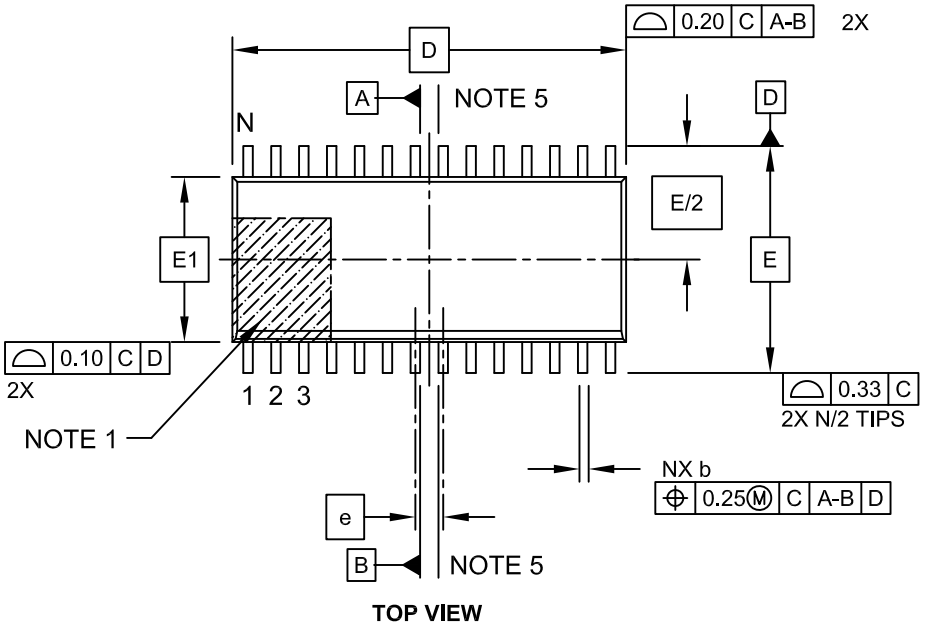
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package is saw singulated.
3. Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.
REF: Reference Dimension, usually without tolerance, for information purposes only.

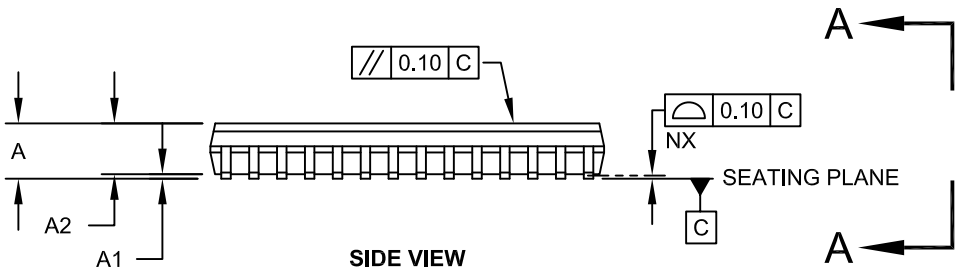
Microchip Technology Drawing C04-152A Sheet 2 of 2

28-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

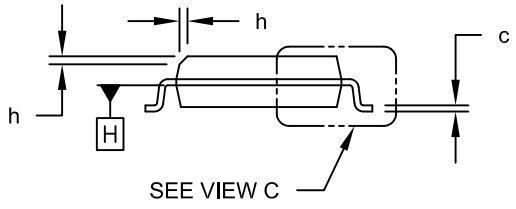
Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



TOP VIEW



SIDE VIEW

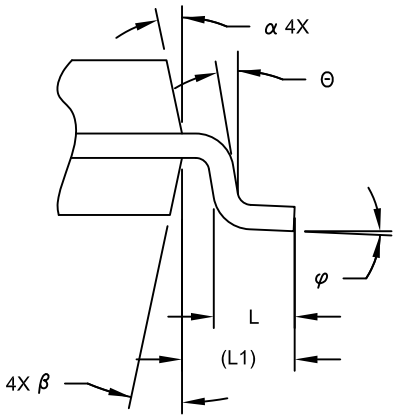


VIEW A-A

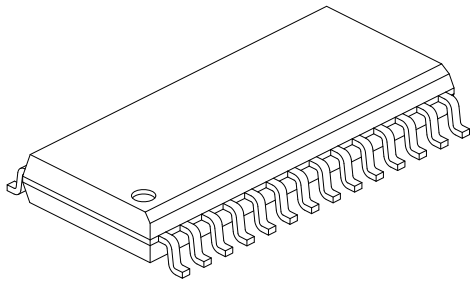
Microchip Technology Drawing C04-052C Sheet 1 of 2

28-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

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



VIEW C



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		28	
Pitch	e		1.27 BSC	
Overall Height	A	-	-	2.65
Molded Package Thickness	A2	2.05	-	-
Standoff §	A1	0.10	-	0.30
Overall Width	E		10.30 BSC	
Molded Package Width	E1		7.50 BSC	
Overall Length	D		17.90 BSC	
Chamfer (Optional)	h	0.25	-	0.75
Foot Length	L	0.40	-	1.27
Footprint	L1		1.40 REF	
Lead Angle	Θ	0°	-	-
Foot Angle	φ	0°	-	8°
Lead Thickness	c	0.18	-	0.33
Lead Width	b	0.31	-	0.51
Mold Draft Angle Top	α	5°	-	15°
Mold Draft Angle Bottom	β	5°	-	15°

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic
3. Dimension D does not include mold flash, protrusions or gate burrs, which shall not exceed 0.15 mm per end. Dimension E1 does not include interlead flash or protrusion, which shall not exceed 0.25 mm per side.
4. Dimensioning and tolerancing per ASME Y14.5M

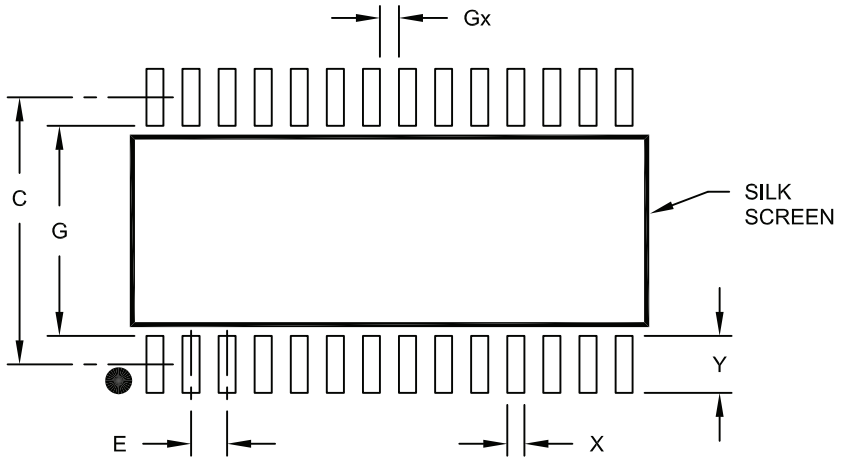
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

   REF: Reference Dimension, usually without tolerance, for information purposes only.
5. Datums A & B to be determined at Datum H.

Microchip Technology Drawing C04-052C Sheet 2 of 2

28-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

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



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E	1.27 BSC		
Contact Pad Spacing	C		9.40	
Contact Pad Width (X28)	X			0.60
Contact Pad Length (X28)	Y			2.00
Distance Between Pads	Gx	0.67		
Distance Between Pads	G	7.40		

Notes:

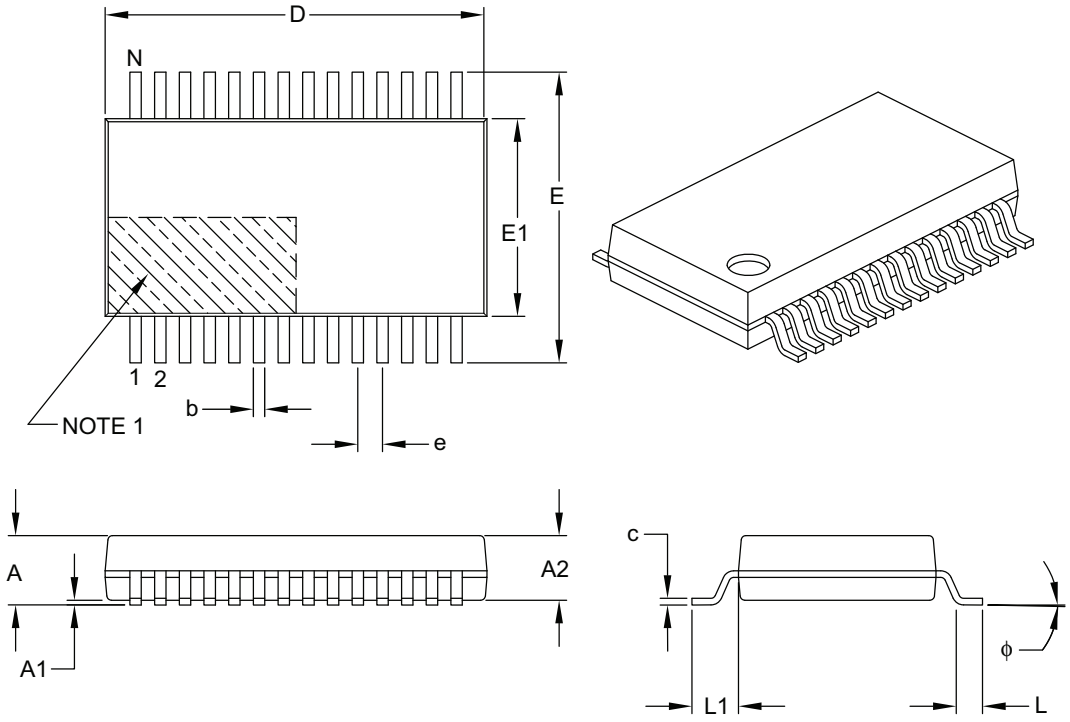
1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-2052A

28-Lead Plastic Shrink Small Outline (SS) – 5.30 mm Body [SSOP]

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



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		28	
Pitch	e		0.65 BSC	
Overall Height	A	–	–	2.00
Molded Package Thickness	A2	1.65	1.75	1.85
Standoff	A1	0.05	–	–
Overall Width	E	7.40	7.80	8.20
Molded Package Width	E1	5.00	5.30	5.60
Overall Length	D	9.90	10.20	10.50
Foot Length	L	0.55	0.75	0.95
Footprint	L1		1.25 REF	
Lead Thickness	c	0.09	–	0.25
Foot Angle	ϕ	0°	4°	8°
Lead Width	b	0.22	–	0.38

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.20 mm per side.
3. Dimensioning and tolerancing per ASME Y14.5M.

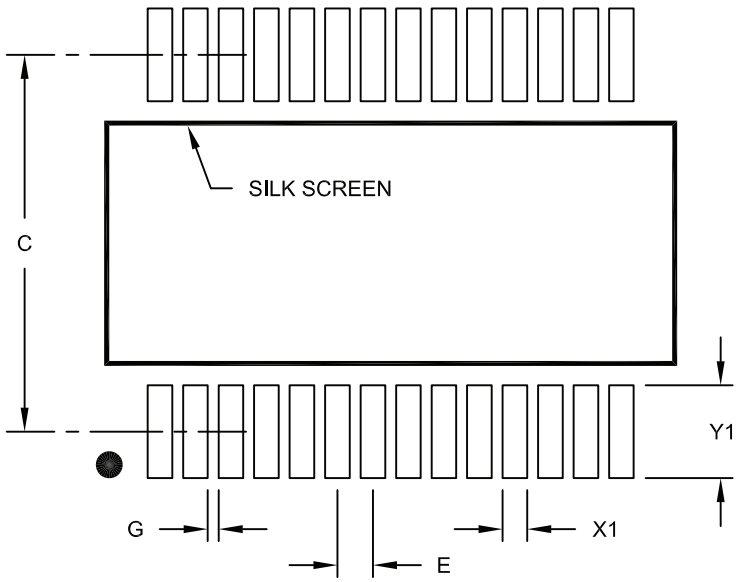
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-073B

28-Lead Plastic Shrink Small Outline (SS) - 5.30 mm Body [SSOP]

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



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		0.65 BSC	
Contact Pad Spacing	C		7.20	
Contact Pad Width (X28)	X1			0.45
Contact Pad Length (X28)	Y1			1.75
Distance Between Pads	G	0.20		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-2073A

APPENDIX A: DATA SHEET REVISION HISTORY

Revision A (April 2010)

Original release of this data sheet.

Revision B (January 2012)

Updated the data sheet to new format; Updated Figure 9-1 and Register 9-1; Updated the Packaging Information section; Updated the Product Identification System section; Other minor corrections.

APPENDIX B: MIGRATING FROM OTHER PIC® DEVICES

This discusses some of the issues in migrating from other PIC® devices to the PIC16F722A/723A family of devices.

Note:	This device has been designed to perform to the parameters of its data sheet. It has been tested to an electrical specification designed to determine its conformance with these parameters. Due to process differences in the manufacture of this device, this device may have different performance characteristics than its earlier version. These differences may cause this device to perform differently in your application than the earlier version of this device.

Note:	The user should verify that the device oscillator starts and performs as expected. Adjusting the loading capacitor values and/or the oscillator mode may be required.

B.1 PIC16F77 to PIC16F722A/723A

TABLE B-1: FEATURE COMPARISON

Feature	PIC16F77	PIC16F722A/723A
Max. Operating Speed	20 MHz	20 MHz
Max. Program Memory (Words)	8K	4K
Max. SRAM (Bytes)	368	192
A/D Resolution	8-bit	8-bit
Timers (8/16-bit)	2/1	2/1
Oscillator Modes	4	8
Brown-out Reset	Y	Y
Internal Pull-ups	RB<7:0>	RB<7:0>
Interrupt-on-change	RB<7:4>	RB<7:0>
Comparator	0	0
USART	Y	Y
Extended WDT	N	N
Software Control Option of WDT/BOR	N	N
INTOSC Frequencies	None	500 kHz - 16 MHz
Clock Switching	N	N

NOTES:

INDEX

T

U

V

W

NOTES:

THE MICROCHIP WEB SITE

Microchip provides online support via our WWW site at www.microchip.com. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at www.microchip.com. Under "Support", click on "Customer Change Notification" and follow the registration instructions.

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support
- Development Systems Information Line

Customers should contact their distributor, representative or field application engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at: http://microchip.com/support

READER RESPONSE

It is our intention to provide you with the best documentation possible to ensure successful use of your Microchip product. If you wish to provide your comments on organization, clarity, subject matter, and ways in which our documentation can better serve you, please FAX your comments to the Technical Publications Manager at (480) 792-4150.

Please list the following information, and use this outline to provide us with your comments about this document.

TO: Technical Publications Manager
RE: Reader Response

Total Pages Sent _____

From: Name _____

Company _____

Address _____

City / State / ZIP / Country _____

Telephone: (_____) _____ - _____ FAX: (_____) _____ - _____

Application (optional):

Would you like a reply? ___Y ___N

Device: PIC16(L)F722A/723A Literature Number: DS41417B

Questions:

1. What are the best features of this document?

2. How does this document meet your hardware and software development needs?

3. Do you find the organization of this document easy to follow? If not, why?

4. What additions to the document do you think would enhance the structure and subject?

5. What deletions from the document could be made without affecting the overall usefulness?

6. Is there any incorrect or misleading information (what and where)?

7. How would you improve this document?

PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

PART NO.	[X](1)	–	X	/XX	XXX
Device	Tape and Reel Option		Temperature Range	Package	Pattern

Device:	PIC16F722A, PIC16LF722A PIC16F723A, PIC16LF723A		
Tape and Reel Option:	Blank T	= =	Standard packaging (tube or tray) Tape and Reel(1)
Temperature Range:	I E	= =	-40°C to+85°C (Industrial) -40°C to+125°C (Extended)
Package:	MV ML SO SP SS	= = = = =	UQFN QFN SOIC SPDIP SSOP
Pattern:	3-Digit Pattern Code for QTP (blank otherwise)		

Examples:

a) PIC16F722A-E/SP 301 = Extended Temp., SPDIP package, QTP pattern #301

b) PIC16F722A-I/SO = Industrial Temp., SOIC package

Note 1: Tape and Reel identifier only appears in the catalog part number description. This identifier is used for ordering purposes and is not printed on the device package. Check with your Microchip Sales Office for package availability with the Tape and Reel option.

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://www.microchip.com/support
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Cleveland
Independence, OH
Tel: 216-447-0464
Fax: 216-447-0643

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Farmington Hills, MI
Tel: 248-538-2250
Fax: 248-538-2260

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

Santa Clara
Santa Clara, CA
Tel: 408-961-6444
Fax: 408-961-6445

Toronto
Mississauga, Ontario,
Canada
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Chongqing
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

China - Hangzhou
Tel: 86-571-2819-3187
Fax: 86-571-2819-3189

China - Hong Kong SAR
Tel: 852-2401-1200
Fax: 852-2401-3431

China - Nanjing
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8203-2660
Fax: 86-755-8203-1760

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xian
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

China - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

India - Pune
Tel: 91-20-2566-1512
Fax: 91-20-2566-1513

Japan - Osaka
Tel: 81-66-152-7160
Fax: 81-66-152-9310

Japan - Yokohama
Tel: 81-45-471- 6166
Fax: 81-45-471-6122

Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-5778-366
Fax: 886-3-5770-955

Taiwan - Kaohsiung
Tel: 886-7-536-4818
Fax: 886-7-330-9305

Taiwan - Taipei
Tel: 886-2-2500-6610
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

UK - Wokingham
Tel: 44-118-921-5869
Fax: 44-118-921-5820

11/29/11