



# **PIC16F688**

## **Data Sheet**

14-Pin Flash-Based, 8-Bit  
CMOS Microcontrollers with  
nanoWatt 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.

---

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in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates.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

#### **Trademarks**

The Microchip name and logo, the Microchip logo, Accuron, dsPIC, KEELOQ, KEELOQ logo, microID, MPLAB, PIC, PICmicro, PICSTART, PRO MATE, PowerSmart, rfPIC, and SmartShunt are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.


AmpLab, FilterLab, Linear Active Thermistor, Migratable Memory, MXDEV, MXLAB, PS logo, SEEVAL, SmartSensor and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, ECAN, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, Mindi, MiWi, MPASM, MPLAB Certified logo, MPLIB, MPLINK, PICkit, PICDEM, PICDEM.net, PICLAB, PICtail, PowerCal, PowerInfo, PowerMate, PowerTool, REAL ICE, rFLAB, rfPICDEM, Select Mode, Smart Serial, SmartTel, Total Endurance, UNI/O, WiperLock and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

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

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

© 2007, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

 Printed on recycled paper.

*Microchip received ISO/TS-16949:2002 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona, Gresham, Oregon and Mountain View, California. The Company's quality system processes and procedures are for its PIC<sup>®</sup> MCUs and dsPIC<sup>®</sup> DSCs, KEELOQ<sup>®</sup> 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.*

**QUALITY MANAGEMENT SYSTEM**  
**CERTIFIED BY DNV**  
**== ISO/TS 16949:2002 ==**

## 14-Pin Flash-Based, 8-Bit CMOS Microcontrollers with nanoWatt Technology

### 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
- Interrupt capability
- 8-level deep hardware stack
- Direct, Indirect and Relative Addressing modes

### Special Microcontroller Features:

- Precision Internal Oscillator:
  - Factory calibrated to  $\pm 1\%$
  - Software selectable frequency range of 8 MHz to 125 kHz
  - Software tunable
  - Two-Speed Start-Up mode
  - Crystal fail detect for critical applications
  - Clock mode switching during operation for power savings
- Power-Saving Sleep mode
- Wide operating voltage range (2.0V-5.5V)
- Industrial and Extended temperature range
- Power-on Reset (POR)
- Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Brown-out Reset (BOR) with software control option
- Enhanced Low-Current Watchdog Timer (WDT) with on-chip oscillator (software selectable nominal 268 seconds with full prescaler) with software enable
- Multiplexed Master Clear with weak pull-up or input only pin
- Programmable code protection
- High-Endurance Flash/EEPROM cell:
  - 100,000 write Flash endurance
  - 1,000,000 write EEPROM endurance
  - Flash/Data EEPROM retention: > 40 years

### Low-Power Features:

- Standby Current:
  - 50 nA @ 2.0V, typical
- Operating Current:
  - 11  $\mu$ A @ 32 kHz, 2.0V, typical
  - 220  $\mu$ A @ 4 MHz, 2.0V, typical
- Watchdog Timer Current:
  - 1  $\mu$ A @ 2.0V, typical

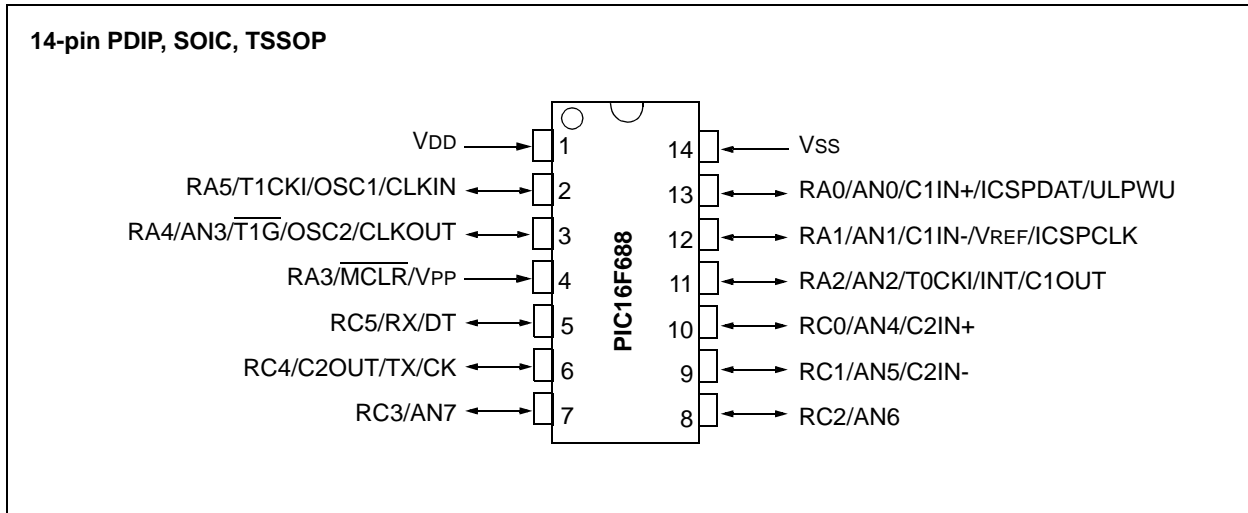
### Peripheral Features:

- 12 I/O pins with individual direction control:
  - High-current source/sink for direct LED drive
  - Interrupt-on-change pin
  - Individually programmable weak pull-ups
  - Ultra Low-Power Wake-up
- Analog Comparator module with:
  - Two analog comparators
  - Programmable On-chip Voltage Reference (CVREF) module (% of VDD)
  - Comparator inputs and outputs externally accessible
- A/D Converter:
  - 10-bit resolution and 8 channels
- Timer0: 8-bit timer/counter with 8-bit programmable prescaler
- Enhanced Timer1:
  - 16-bit timer/counter with prescaler
  - External Timer1 Gate (count enable)
  - Option to use OSC1 and OSC2 in LP mode as Timer1 oscillator if INTOSC mode selected
- Enhanced USART Module:
  - Supports RS-485, RS-232, and LIN 1.2
  - Auto-Baud Detect
  - Auto-wake-up on Start bit
- In-Circuit Serial Programming™ (ICSP™) via two pins

Device	Program Memory	Data Memory		I/O	10-bit A/D (ch)	Comparators	Timers 8/16-bit
	Flash (words)	SRAM (bytes)	EEPROM (bytes)				
PIC16F688	4096	256	256	12	8	2	1/1

# PIC16F688

## Pin Diagram (PDIP, SOIC, TSSOP)

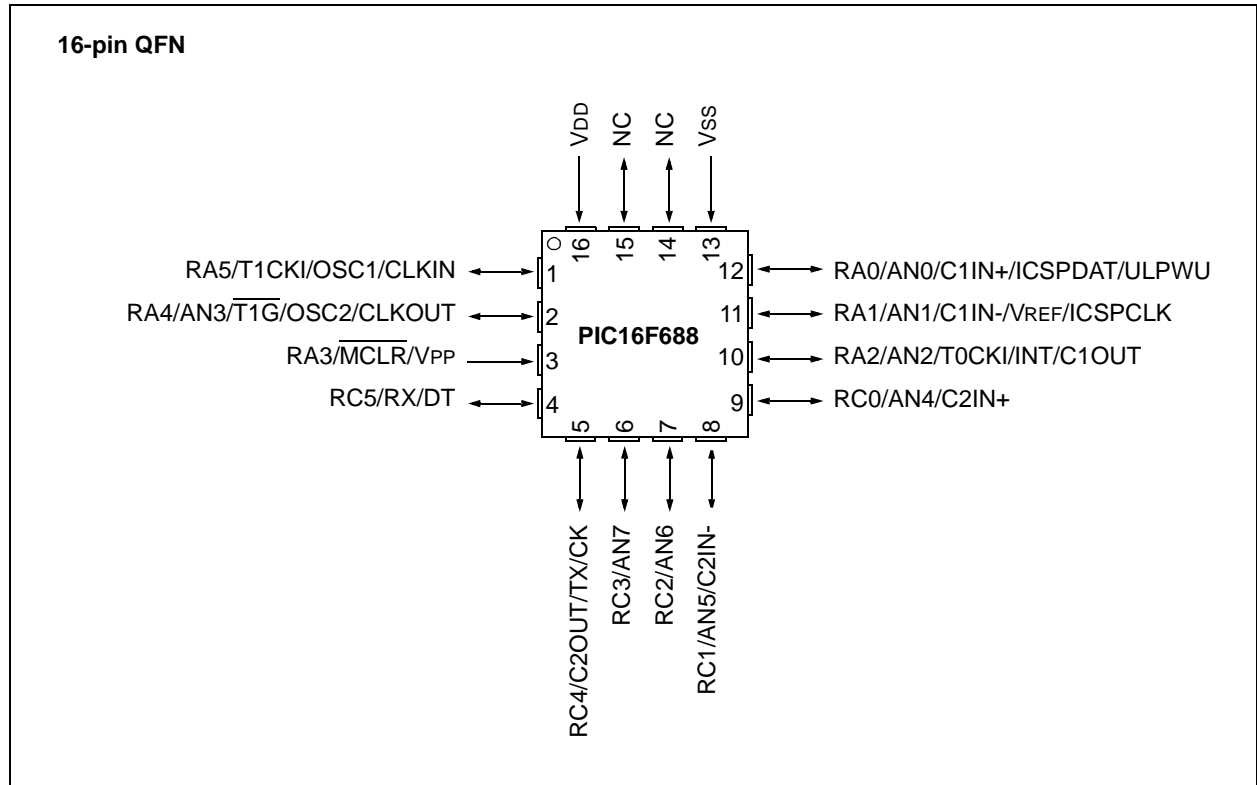


**TABLE 1: PIC16F688 14-PIN SUMMARY (PDIP, SOIC, TSSOP)**

I/O	Pin	Analog	Comparators	Timers	EUSART	Interrupt	Pull-up	Basic
RA0	13	AN0/ULPWU	C1IN+	—	—	IOC	Y	ICSPDAT
RA1	12	AN1	C1IN-	—	—	IOC	Y	VREF/ICSPCLK
RA2	11	AN2	C1OUT	T0CKI	—	IOC/INT	Y	—
RA3	4	—	—	—	—	IOC	Y <sup>(1)</sup>	MCLR/VPP
RA4	3	AN3	—	T1G	—	IOC	Y	OSC2/CLKOUT
RA5	2	—	—	T1CKI	—	IOC	Y	OSC1/CLKIN
RC0	10	AN4	C2IN+	—	—	—	—	—
RC1	9	AN5	C2IN-	—	—	—	—	—
RC2	8	AN6	—	—	—	—	—	—
RC3	7	AN7	—	—	—	—	—	—
RC4	6	—	C2OUT	—	TX/CK	—	—	—
RC5	5	—	—	—	RX/DT	—	—	—
—	1	—	—	—	—	—	—	VDD
—	14	—	—	—	—	—	—	Vss

**Note 1:** Pull-up activated only with external MCLR configuration.

## Pin Diagram (QFN)



**TABLE 2: PIC16F688 16-PIN SUMMARY (QFN)**

I/O	Pin	Analog	Comparators	Timers	EUSART	Interrupt	Pull-up	Basic
RA0	12	AN0/ULPWU	C1IN+	—	—	IOC	Y	ICSPDAT
RA1	11	AN1	C1IN-	—	—	IOC	Y	VREF/ICSPCLK
RA2	10	AN2	C1OUT	T0CKI	—	IOC/INT	Y	—
RA3	3	—	—	—	—	IOC	Y <sup>(1)</sup>	$\overline{MCLR}$ /VPP
RA4	2	AN3	—	$\overline{T1G}$	—	IOC	Y	OSC2/CLKOUT
RA5	1	—	—	T1CKI	—	IOC	Y	OSC1/CLKIN
RC0	9	AN4	C2IN+	—	—	—	—	—
RC1	8	AN5	C2IN-	—	—	—	—	—
RC2	7	AN6	—	—	—	—	—	—
RC3	6	AN7	—	—	—	—	—	—
RC4	5	—	C2OUT	—	TX/CK	—	—	—
RC5	4	—	—	—	RX/DT	—	—	—
—	16	—	—	—	—	—	—	VDD
—	13	—	—	—	—	—	—	VSS
—	14	—	—	—	—	—	—	NC
—	15	—	—	—	—	—	—	NC

**Note 1:** Pull-up activated only with external MCLR configuration.

# PIC16F688

## Table of Contents

1.0	Device Overview .....	5
2.0	Memory Organization .....	7
3.0	Clock Sources .....	21
4.0	I/O Ports .....	33
5.0	Timer0 Module .....	45
6.0	Timer1 Module with Gate Control.....	49
7.0	Comparator Module.....	55
8.0	Analog-to-Digital Converter (A/D) Module.....	65
9.0	Data EEPROM and Flash Program Memory Control .....	77
10.0	Enhanced Universal Asynchronous Receiver Transmitter (EUSART) .....	83
11.0	Special Features of the CPU .....	109
12.0	Instruction Set Summary .....	129
13.0	Development Support .....	139
14.0	Electrical Specifications.....	143
15.0	DC and AC Characteristics Graphs and Tables.....	163
16.0	Packaging Information.....	185
	Appendix A: Data Sheet Revision History.....	191
	Appendix B: Migrating from other PIC® Devices .....	191
	Index .....	193
	On-line Support .....	197
	Systems Information and Upgrade Hot Line .....	197
	Reader Response .....	198
	Product Identification System.....	199

## 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](mailto: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.

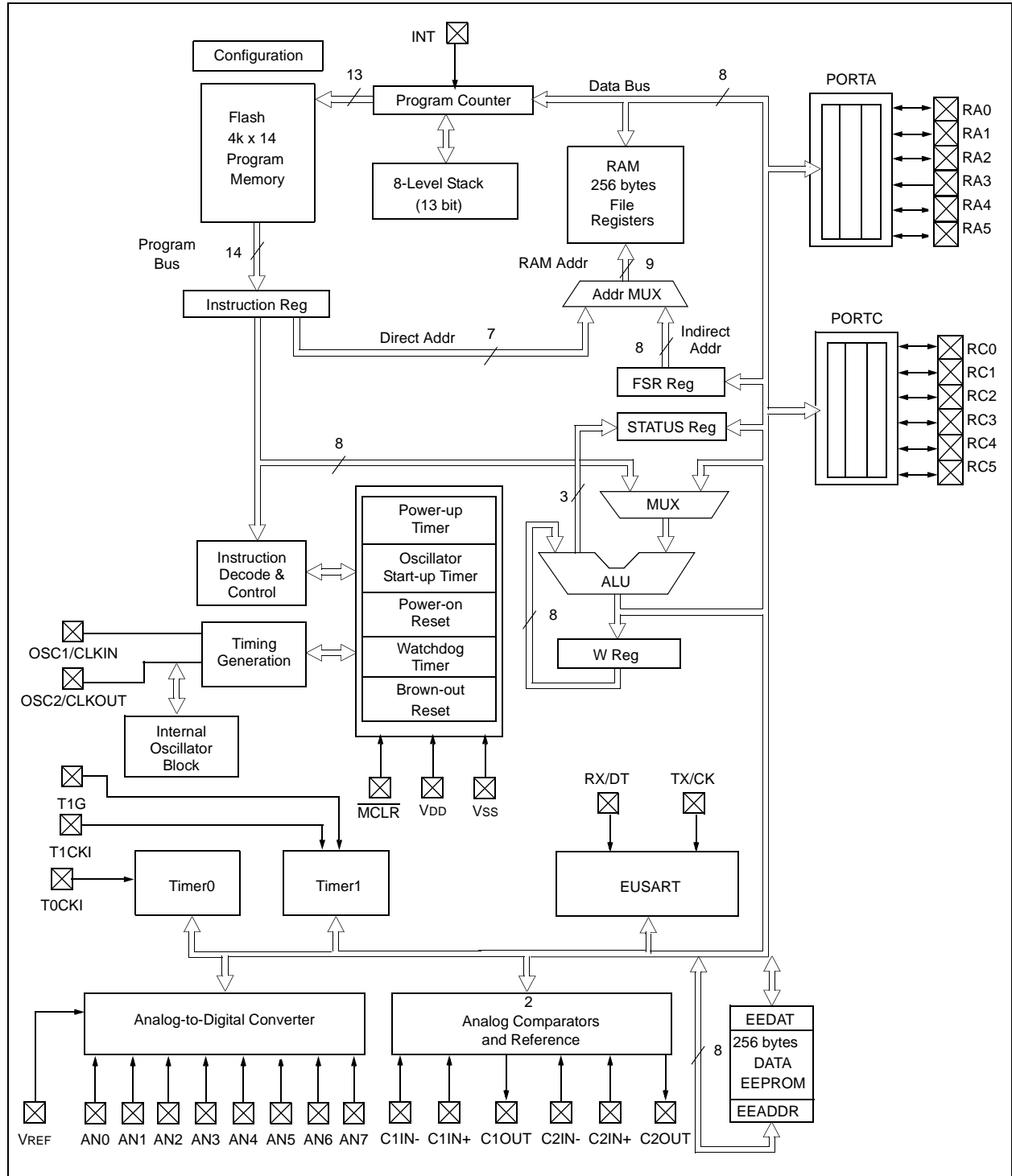
### Customer Notification System

Register on our web site at [www.microchip.com](http://www.microchip.com) to receive the most current information on all of our products.

## 1.0 DEVICE OVERVIEW

The PIC16F688 is covered by this data sheet. It is available in 14-pin PDIP, SOIC, TSSOP and QFN packages. Figure 1-1 shows a block diagram of the PIC16F688 device. Table 1-1 shows the pinout description.

**FIGURE 1-1: PIC16F688 BLOCK DIAGRAM**



# PIC16F688

**TABLE 1-1: PIC16F688 PINOUT DESCRIPTION**

Name	Function	Input Type	Output Type	Description
RA0/AN0/C1IN+/ICSPDAT/ULPWU	RA0	TTL	CMOS	PORTA I/O w/prog pull-up and interrupt-on-change
	AN0	AN	—	A/D Channel 0 input
	C1IN+	AN	—	Comparator 1 input
	ICSPDAT	TTL	CMOS	Serial Programming Data I/O
	ULPWU	AN	—	Ultra Low-Power Wake-up input
RA1/AN1/C1IN-/VREF/ICSPCLK	RA1	TTL	CMOS	PORTA I/O w/prog pull-up and interrupt-on-change
	AN1	AN	—	A/D Channel 1 input
	C1IN-	AN	—	Comparator 1 input
	VREF	AN	—	External Voltage Reference for A/D
	ICSPCLK	ST	—	Serial Programming Clock
RA2/AN2/T0CKI/INT/C1OUT	RA2	ST	CMOS	PORTA I/O w/prog pull-up and interrupt-on-change
	AN2	AN	—	A/D Channel 2 input
	T0CKI	ST	—	Timer0 clock input
	INT	ST	—	External Interrupt
	C1OUT	—	CMOS	Comparator 1 output
RA3/MCLR/VPP	RA3	TTL	—	PORTA input with interrupt-on-change
	MCLR	ST	—	Master Clear w/internal pull-up
	VPP	HV	—	Programming voltage
RA4/AN3/T1G/OSC2/CLKOUT	RA4	TTL	CMOS	PORTA I/O w/prog pull-up and interrupt-on-change
	AN3	AN	—	A/D Channel 3 input
	T1G	ST	—	Timer1 gate
	OSC2	—	XTAL	Crystal/Resonator
	CLKOUT	—	CMOS	Fosc/4 output
RA5/T1CKI/OSC1/CLKIN	RA5	TTL	CMOS	PORTA I/O w/prog pull-up and interrupt-on-change
	T1CKI	ST	—	Timer1 clock
	OSC1	XTAL	—	Crystal/Resonator
	CLKIN	ST	—	External clock input/RC oscillator connection
RC0/AN4/C2IN+	RC0	TTL	CMOS	PORTC I/O
	AN4	AN	—	A/D Channel 4 input
	C2IN+	AN	—	Comparator 2 input
RC1/AN5/C2IN-	RC1	TTL	CMOS	PORTC I/O
	AN5	AN	—	A/D Channel 5 input
	C2IN-	AN	—	Comparator 2 input
RC2/AN6	RC2	TTL	CMOS	PORTC I/O
	AN6	AN	—	A/D Channel 6 input
RC3/AN7	RC3	TTL	CMOS	PORTC I/O
	AN7	AN	—	A/D Channel 7 input
RC4/C2OUT/TX/CK	RC4	TTL	CMOS	PORTC I/O
	C2OUT	—	CMOS	Comparator 2 output
	TX	—	CMOS	USART asynchronous output
	CK	ST	CMOS	USART asynchronous clock
RC5/RX/DT	RC5	TTL	CMOS	Port C I/O
	RX	ST	CMOS	USART asynchronous input
	DT	ST	CMOS	USART asynchronous data
VSS	VSS	Power	—	Ground reference
VDD	VDD	Power	—	Positive supply

**Legend:** AN = Analog input or output      CMOS = CMOS compatible input or output      OC = Open collector output  
TTL = TTL compatible input      ST = Schmitt Trigger input with CMOS levels  
HV = High Voltage      XTAL = Crystal

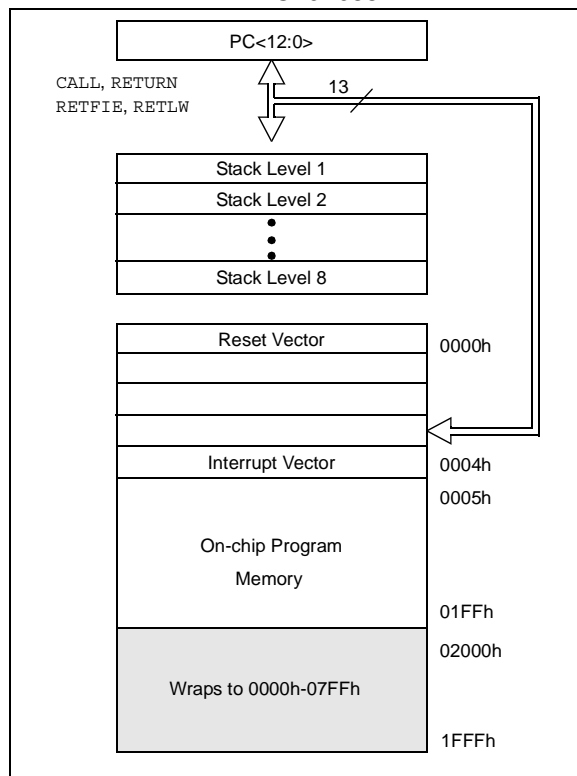


## 2.0 MEMORY ORGANIZATION

### 2.1 Program Memory Organization

The PIC16F688 has a 13-bit program counter capable of addressing a 4K x 14 program memory space. Only the first 4K x 14 (0000h-01FFF) for the PIC16F688 is physically implemented. Accessing a location above these boundaries will cause a wraparound within the first 4K x 14 space. The Reset vector is at 0000h and the interrupt vector is at 0004h (see Figure 2-1).

**FIGURE 2-1: PROGRAM MEMORY MAP AND STACK FOR THE PIC16F688**



### 2.2 Data Memory Organization

The data memory is partitioned into multiple banks, which contain the General Purpose Registers (GPR) and the Special Function Registers (SFR). 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 256 x 8 in the PIC16F688. Each register is accessed, either directly or indirectly, through the File Select Register (FSR) (see Section 2.4 "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 (see Tables 2-1, 2-2, 2-3 and 2-4). These registers are static RAM.

The special 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.

# PIC16F688

**FIGURE 2-2: PIC16F688 SPECIAL FUNCTION REGISTERS**

File Address	File Address	File Address	File Address
Indirect addr. <sup>(1)</sup> 00h	Indirect addr. <sup>(1)</sup> 80h	Indirect addr. <sup>(1)</sup> 100h	Indirect addr. <sup>(1)</sup> 180h
TMR0 01h	OPTION_REG 81h	TMR0 101h	OPTION_REG 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	PORTA 105h	TRISA 185h
06h	86h	106h	186h
PORTC 07h	TRISC 87h	PORTC 107h	TRISC 187h
08h	88h	108h	188h
09h	89h	109h	189h
PCLATH 0Ah	PCLATH 8Ah	PCLATH 10Ah	PCLATH 18Ah
INTCON 0Bh	INTCON 8Bh	INTCON 10Bh	INTCON 18Bh
PIR1 0Ch	PIE1 8Ch	10Ch	18Ch
0Dh	8Dh	10Dh	18Dh
TMR1L 0Eh	PCON 8Eh	10Eh	18Eh
TMR1H 0Fh	OSCCON 8Fh	10Fh	18Fh
T1CON 10h	OSCTUNE 90h	110h	190h
BAUDCTL 11h	ANSEL 91h	111h	191h
SPBRGH 12h	92h	112h	192h
SPBRG 13h	93h	113h	193h
RCREG 14h	94h	114h	194h
TXREG 15h	WPUA 95h	115h	195h
TXSTA 16h	IOCA 96h	116h	196h
RCSTA 17h	EEDATH 97h	117h	197h
WDTCON 18h	EEADRH 98h	118h	198h
CMCON0 19h	VRCON 99h	119h	199h
CMCON1 1Ah	EEDAT 9Ah	11Ah	19Ah
1Bh	EEADR 9Bh	11Bh	19Bh
1Ch	EECON1 9Ch	11Ch	19Ch
1Dh	EECON2 <sup>(1)</sup> 9Dh	11Dh	19Dh
ADRESH 1Eh	ADRESL 9Eh	11Eh	19Eh
ADCON0 1Fh	ADCON1 9Fh	11Fh	19Fh
20h	A0h	120h	1A0h
General Purpose Register	General Purpose Register	General Purpose Register	
96 Bytes	80 Bytes	80 Bytes	
7Fh	EFh	16Fh	1EFh
Bank 0	accesses Bank 0	accesses Bank 0	accesses Bank 0
	F0h	170h	1F0h
	FFh	17Fh	1FFh
	Bank 1	Bank 2	Bank 3

Unimplemented data memory locations, read as '0'.

**Note 1:** Not a physical register.

**TABLE 2-1: PIC16F688 SPECIAL REGISTERS SUMMARY BANK 0**

Addr	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	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								xxxx xxxx	20, 117
01h	TMR0	Timer0 Module's register								xxxx xxxx	45, 117
02h	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	19, 117
03h	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	13, 117
04h	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	20, 117
05h	PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	--x0 x000	33, 117
06h	—	Unimplemented								—	—
07h	PORTC	—	—	RC5	RC4	RC3	RC2	RC1	RC0	--xx 0000	42, 117
08h	—	Unimplemented								—	—
09h	—	Unimplemented								—	—
0Ah	PCLATH	—	—	—	Write Buffer for upper 5 bits of Program Counter					---0 0000	19, 117
0Bh	INTCON	GIE	PEIE	TOIE	INTE	RAIE	TOIF	INTF	RAIF <sup>(2)</sup>	0000 000x	15, 117
0Ch	PIR1	EEIF	ADIF	RCIF	C2IF	C1IF	OSFIF	TXIF	TMR1IF	0000 0000	17, 117
0Dh	—	Unimplemented								—	—
0Eh	TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1								xxxx xxxx	48, 117
0Fh	TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1								xxxx xxxx	48, 117
10h	T1CON	T1GINV	TMR1GE	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{T1SYNC}$	TMR1CS	TMR1ON	0000 0000	51, 117
11h	BAUDCTL	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	01-0 0-00	94, 117
12h	SPBRGH	USART Baud Rate High Generator								0000 0000	95, 117
13h	SPBRG	USART Baud Rate Generator								0000 0000	95, 117
14h	RCREG	USART Receive Register								0000 0000	87, 117
15h	TXREG	USART Transmit Register								0000 0000	87, 117
16h	TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010	92, 117
17h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	93, 117
18h	WDTCON	—	—	—	WDTPS3	WDTPS2	WDTPS1	WDTPS0	SWDTEN	---0 1000	124, 117
19h	CMCON0	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	0000 0000	61, 117
1Ah	CMCON1	—	—	—	—	—	—	T1GSS	C2SYNC	---- --10	62, 117
1Bh	—	Unimplemented								—	—
1Ch	—	Unimplemented								—	—
1Dh	—	Unimplemented								—	—
1Eh	ADRESH	Most Significant 8 bits of the left shifted A/D result or 2 bits of right shifted result								xxxx xxxx	72, 117
1Fh	ADCON0	ADFM	VCFG	—	CHS2	CHS1	CHS0	$\overline{GO/DONE}$	ADON	00-0 0000	71, 117

**Legend:** — = Unimplemented locations read as '0', u = unchanged, x = unknown, q = value depends on condition, shaded = unimplemented

**Note 1:** Other (non Power-up) Resets include MCLR Reset and Watchdog Timer Reset during normal operation.

**Note 2:** MCLR and WDT Reset does not affect the previous value data latch. The RAIF bit will be cleared upon Reset but will set again if the mismatched exists.

# PIC16F688

**TABLE 2-2: PIC16F688 SPECIAL FUNCTION REGISTERS SUMMARY BANK 1**

Addr	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	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								xxxx xxxx	20, 117
81h	OPTION_REG	RAPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	14, 117
82h	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	19, 117
83h	STATUS	IRP	RP1	RP0	TO	PD	Z	DC	C	0001 1xxx	13, 117
84h	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	20, 117
85h	TRISA	—	—	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	--11 1111	33, 117
86h	—	Unimplemented								—	—
87h	TRISC	—	—	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	--11 1111	42, 117
88h	—	Unimplemented								—	—
89h	—	Unimplemented								—	—
8Ah	PCLATH	—	—	—	Write Buffer for upper 5 bits of Program Counter				---	0000	19, 117
8Bh	INTCON	GIE	PEIE	T0IE	INTE	RAIE	T0IF	INTF	RAIF <sup>(3)</sup>	0000 000x	15, 117
8Ch	PIE1	EEIE	ADIE	RCIE	C2IE	C1IE	OSFIE	TXIE	TMR1IE	0000 0000	16, 117
8Dh	—	Unimplemented								—	—
8Eh	PCON	—	—	ULPWUE	SBOREN	—	—	POR	BOR	--01 --q <sub>q</sub>	18, 117
8Fh	OSCCON	—	IRCF2	IRCF1	IRCF0	OSTS	HTS	LTS	SCS	-110 x000	22, 118
90h	OSCTUNE	—	—	—	TUN4	TUN3	TUN2	TUN1	TUN0	---0 0000	26, 118
91h	ANSEL	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0	1111 1111	34, 118
92h	—	Unimplemented								—	—
93h	—	Unimplemented								—	—
94h	—	Unimplemented								—	—
95h	WPUA <sup>(2)</sup>	—	—	WPUA5	WPUA4	—	WPUA2	WPUA1	WPUA0	--11 -111	35, 118
96h	IOCA	—	—	IOCA5	IOCA4	IOCA3	IOCA2	IOCA1	IOCA0	--00 0000	35, 118
97h	EEDATH	—	—	EEDATH5	EEDATH4	EEDATH3	EEDATH2	EEDATH1	EEDATH0	--00 0000	78, 118
98h	EEADRH	—	—	—	—	EEADRH3	EEADRH2	EEADRH1	EEADRH0	---- 0000	78, 118
99h	VRCON	VREN	—	VRR	—	VR3	VR2	VR1	VR0	0-0- 0000	63, 118
9Ah	EEDAT	EEDAT7	EEDAT6	EEDAT5	EEDAT4	EEDAT3	EEDAT2	EEDAT1	EEDAT0	0000 0000	78, 118
9Bh	EEADR	EEADR7	EEADR6	EEADR5	EEADR4	EEADR3	EEADR2	EEADR1	EEADR0	0000 0000	78, 118
9Ch	EECON1	EEPGD	—	—	—	WRERR	WREN	WR	RD	x--- x000	79, 118
9Dh	EECON2	EEPROM Control 2 Register (not a physical register)								---- ----	77, 118
9Eh	ADRESL	Least Significant 2 bits of the left shifted result or 8 bits of the right shifted result								xxxx xxxx	72, 118
9Fh	ADCON1	—	ADCS2	ADCS1	ADCS0	—	—	—	—	-000 ----	71, 118

- Legend:** — = Unimplemented locations read as '0', u = unchanged, x = unknown, q = value depends on condition, shaded = unimplemented
- Note** 1: Other (non Power-up) Resets include MCLR Reset and Watchdog Timer Reset during normal operation.
- 2: RA3 pull-up is enabled when pin is configured as MCLR in the Configuration Word register.
- 3: MCLR and WDT Reset does not affect the previous value data latch. The RAIF bit will be cleared upon Reset but will set again if the mismatched exists.

**TABLE 2-3: PIC16F688 SPECIAL REGISTERS SUMMARY BANK 2**

Addr	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	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								xxxx xxxx	20, 117
101h	TMR0	Timer0 Module's register								xxxx xxxx	45, 117
102h	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	19, 117
103h	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	13, 117
104h	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	20, 117
105h	PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	--x0 x000	33, 117
106h	—	Unimplemented								—	—
107h	PORTC	—	—	RC5	RC4	RC3	RC2	RC1	RC0	--xx 0000	42, 117
108h	—	Unimplemented								—	—
109h	—	Unimplemented								—	—
10Ah	PCLATH	—	—	—	Write Buffer for upper 5 bits of Program Counter					---0 0000	19, 117
10Bh	INTCON	GIE	PEIE	TOIE	INTE	RAIE	TOIF	INTF	RAIF <sup>(2)</sup>	0000 000x	15, 117
10Ch	—	Unimplemented								—	—
10Dh	—	Unimplemented								—	—
10Eh	—	Unimplemented								—	—
10Fh	—	Unimplemented								—	—
110h	—	Unimplemented								—	—
111h	—	Unimplemented								—	—
112h	—	Unimplemented								—	—
113h	—	Unimplemented								—	—
114h	—	Unimplemented								—	—
115h	—	Unimplemented								—	—
116h	—	Unimplemented								—	—
117h	—	Unimplemented								—	—
118h	—	Unimplemented								—	—
119h	—	Unimplemented								—	—
11Ah	—	Unimplemented								—	—
11Bh	—	Unimplemented								—	—
11Ch	—	Unimplemented								—	—
11Dh	—	Unimplemented								—	—
11Eh	—	Unimplemented								—	—
11Fh	—	Unimplemented								—	—

**Legend:** — = Unimplemented locations read as '0', u = unchanged, x = unknown, q = value depends on condition, shaded = unimplemented

**Note 1:** Other (non Power-up) Resets include MCLR Reset and Watchdog Timer Reset during normal operation.

**2:** MCLR and WDT Reset does not affect the previous value data latch. The RAIF bit will be cleared upon Reset but will set again if the mismatched exists.

# PIC16F688

**TABLE 2-4: PIC16F688 SPECIAL FUNCTION REGISTERS SUMMARY BANK 3**

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR/BOR	Page
Bank 3											
180h	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								xxxx xxxx	20, 117
181h	OPTION_REG	RAPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	14, 117
182h	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	19, 117
183h	STATUS	IRP	RP1	RP0	T0	PD	Z	DC	C	0001 1xxx	13, 117
184h	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	20, 117
185h	TRISA	—	—	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	--11 1111	33, 117
186h	—	Unimplemented								—	—
187h	TRISC	—	—	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	--11 1111	42, 117
188h	—	Unimplemented								—	—
189h	—	Unimplemented								—	—
18Ah	PCLATH	—	—	—	Write Buffer for upper 5 bits of Program Counter					---0 0000	19, 117
18Bh	INTCON	GIE	PEIE	T0IE	INTE	RAIE	T0IF	INTF	RAIF <sup>(2)</sup>	0000 000x	15, 117
18Ch	—	Unimplemented								—	—
18Dh	—	Unimplemented								—	—
190h	—	Unimplemented								—	—
191h	—	Unimplemented								—	—
192h	—	Unimplemented								—	—
193h	—	Unimplemented								—	—
194h	—	Unimplemented								—	—
195h	—	Unimplemented								—	—
196h	—	Unimplemented								—	—
19Ah	—	Unimplemented								—	—
19Bh	—	Unimplemented								—	—
199h	—	Unimplemented								—	—
19Ah	—	Unimplemented								—	—
19Bh	—	Unimplemented								—	—
19Ch	—	Unimplemented								—	—
19Dh	—	Unimplemented								—	—
19Eh	—	Unimplemented								—	—
19Fh	—	Unimplemented								—	—

**Legend:** — = Unimplemented locations read as '0', u = unchanged, x = unknown, q = value depends on condition, shaded = unimplemented

**Note 1:** Other (non Power-up) Resets include MCLR Reset and Watchdog Timer Reset during normal operation.

**Note 2:** MCLR and WDT Reset does not affect the previous value data latch. The RAIF bit will be cleared upon Reset but will set again if the mismatched exists.

## 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  $\overline{\text{TO}}$  and  $\overline{\text{PD}}$  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 (see **Section 12.0 "Instruction Set Summary"**).

**Note 1:** Bits IRP and RP1 of the STATUS register are not used by the PIC16F688 and should be maintained as clear. Use of these bits is not recommended, since this may affect upward compatibility with future products.

**2:** The C and DC bits operate as a  $\overline{\text{Borrow}}$  and  $\overline{\text{Digit Borrow}}$  out bit, 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	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC <sup>(1)</sup>	C <sup>(1)</sup>
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  **$\overline{\text{TO}}$ :** Time-out bit
  - 1 = After power-up, `CLRWDT` instruction or `SLEEP` instruction
  - 0 = A WDT time-out occurred
- bit 3  **$\overline{\text{PD}}$ :** 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/ $\overline{\text{Borrow}}$  bit (`ADDWF`, `ADDLW`, `SUBLW`, `SUBWF` instructions)<sup>(1)</sup>
  - 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<sup>(1)</sup> (`ADDWF`, `ADDLW`, `SUBLW`, `SUBWF` instructions)<sup>(1)</sup>
  - 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.

# PIC16F688

## 2.2.2.2 OPTION Register

The OPTION register is a readable and writable register, which contains various control bits to configure:

- Timer0/WDT prescaler
- External RA2/INT interrupt
- Timer0
- Weak pull-ups on PORTA

**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'. See **Section 5.1.3 “Software Programmable 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
RAPU	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      **RAPU:** PORTA Pull-up Enable bit  
 1 = PORTA pull-ups are disabled  
 0 = PORTA pull-ups are enabled by individual PORT latch values
- bit 6      **INTEDG:** Interrupt Edge Select bit  
 1 = Interrupt on rising edge of RA2/INT pin  
 0 = Interrupt on falling edge of RA2/INT pin
- bit 5      **T0CS:** Timer0 Clock Source Select bit  
 1 = Transition on RA2/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 RA2/T0CKI pin  
 0 = Increment on low-to-high transition on RA2/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 INTCON Register

The INTCON register is a readable and writable register, which contains the various enable and flag bits for TMR0 register overflow, PORTA change and external RA2/INT 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 2-3: 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	TOIE	INTE	RAIE	TOIF	INTF	RAIF
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      **TOIE:** Timer0 Overflow Interrupt Enable bit  
1 = Enables the Timer0 interrupt  
0 = Disables the Timer0 interrupt
- bit 4      **INTE:** RA2/INT External Interrupt Enable bit  
1 = Enables the RA2/INT external interrupt  
0 = Disables the RA2/INT external interrupt
- bit 3      **RAIE:** PORTA Change Interrupt Enable bit<sup>(1)</sup>  
1 = Enables the PORTA change interrupt  
0 = Disables the PORTA change interrupt
- bit 2      **TOIF:** Timer0 Overflow Interrupt Flag bit<sup>(2)</sup>  
1 = Timer0 register has overflowed (must be cleared in software)  
0 = Timer0 register did not overflow
- bit 1      **INTF:** RA2/INT External Interrupt Flag bit  
1 = The RA2/INT external interrupt occurred (must be cleared in software)  
0 = The RA2/INT external interrupt did not occur
- bit 0      **RAIF:** PORTA Change Interrupt Flag bit  
1 = When at least one of the PORTA <5:0> pins changed state (must be cleared in software)  
0 = None of the PORTA <5:0> pins have changed state

- Note 1:** IOCA register must also be enabled.
- 2:** TOIF bit is set when TMR0 rolls over. TMR0 is unchanged on Reset and should be initialized before clearing TOIF bit.

# PIC16F688

## 2.2.2.4 PIE1 Register

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

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

### REGISTER 2-4: 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
EEIE	ADIE	RCIE	C2IE	C1IE	OSFIE	TXIE	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      **EEIE:** EE Write Complete Interrupt Enable bit  
           1 = Enables the EE write complete interrupt  
           0 = Disables the EE write 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:** EUSART Receive Interrupt Enable bit  
           1 = Enables the EUSART receive interrupt  
           0 = Disables the EUSART receive interrupt
- bit 4      **C2IE:** Comparator 2 Interrupt Enable bit  
           1 = Enables the Comparator C2 interrupt  
           0 = Disables the Comparator C2 interrupt
- bit 3      **C1IE:** Comparator 1 Interrupt Enable bit  
           1 = Enables the Comparator C1 interrupt  
           0 = Disables the Comparator C1 interrupt
- bit 2      **OSFIE:** Oscillator Fail Interrupt Enable bit  
           1 = Enables the oscillator fail interrupt  
           0 = Disables the oscillator fail interrupt
- bit 1      **TXIE:** EUSART Transmit Interrupt Enable bit  
           1 = Enables the EUSART transmit interrupt  
           0 = Disables the EUSART transmit interrupt
- bit 0      **TMR1IE:** Timer1 Overflow Interrupt Enable bit  
           1 = Enables the Timer1 overflow interrupt  
           0 = Disables the Timer1 overflow interrupt

## 2.2.2.5 PIR1 Register

The PIR1 register contains the interrupt flag bits, as shown in Register 2-5.

**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 bit of the INTCON register. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

**REGISTER 2-5: PIR1: PERIPHERAL INTERRUPT REQUEST REGISTER 1**

R/W-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0	R-0	R/W-0
EEIF	ADIF	RCIF	C2IF	C1IF	OSFIF	TXIF	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      **EEIF:** EEPROM Write Operation Interrupt Flag bit  
 1 = The write operation completed (must be cleared in software)  
 0 = The write operation has not completed or has not been started
- 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:** EUSART Receive Interrupt Flag bit  
 1 = The EUSART receive buffer is full (cleared by reading RCREG)  
 0 = The EUSART receive buffer is not full
- bit 4      **C2IF:** Comparator C2 Interrupt Flag bit  
 1 = Comparator output (C2OUT bit) has changed (must be cleared in software)  
 0 = Comparator output (C2OUT bit) has not changed
- bit 3      **C1IF:** Comparator C1 Interrupt Flag bit  
 1 = Comparator output (C1OUT bit) has changed (must be cleared in software)  
 0 = Comparator output (C1OUT bit) has not changed
- bit 2      **OSFIF:** Oscillator Fail Interrupt Flag bit  
 1 = System oscillator failed, clock input has changed to INTOSC (must be cleared in software)  
 0 = System clock operating
- bit 1      **TXIF:** EUSART Transmit Interrupt Flag bit  
 1 = The EUSART transmit buffer is empty (cleared by writing to TXREG)  
 0 = The EUSART transmit buffer is full
- 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

# PIC16F688

## 2.2.2.6 PCON Register

The Power Control (PCON) register (see Register 2-6) contains flag bits to differentiate between a:

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

The PCON register also controls the Ultra Low-Power Wake-up and software enable of the  $\overline{\text{BOR}}$ .

### REGISTER 2-6: PCON: POWER CONTROL REGISTER

U-0	U-0	R/W-0	R/W-1	U-0	U-0	R/W-0	R/W-x
—	—	ULPWUE	SBOREN <sup>(1)</sup>	—	—	$\overline{\text{POR}}$	$\overline{\text{BOR}}$
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

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

bit 5 **ULPWUE:** Ultra Low-Power Wake-up Enable bit

1 = Ultra low-power wake-up enabled

0 = Ultra low-power wake-up disabled

bit 4 **SBOREN:** Software BOR Enable bit<sup>(1)</sup>

1 = BOR enabled

0 = BOR disabled

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

bit 1  **$\overline{\text{POR}}$ :** 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  **$\overline{\text{BOR}}$ :** Brown-out Reset Status bit

1 = No Brown-out Reset occurred

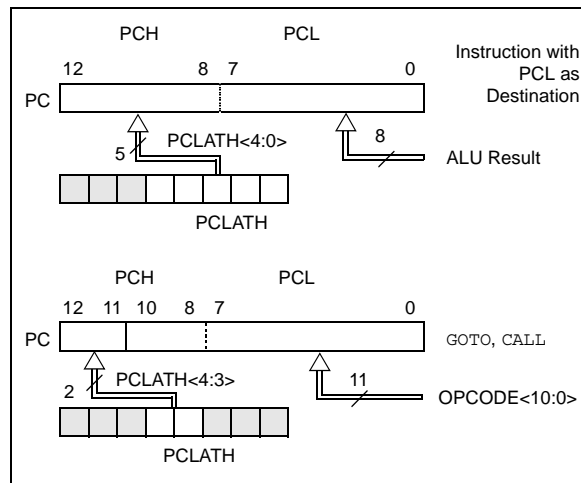
0 = A Brown-out Reset occurred (must be set in software after a Brown-out Reset occurs)

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

## 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 2-3 shows the two situations for the loading of the PC. The upper example in Figure 2-3 shows how the PC is loaded on a write to PCL (PCLATH<4:0> → PCH). The lower example in Figure 2-3 shows how the PC is loaded during a CALL or GOTO instruction (PCLATH<4:3> → PCH).

**FIGURE 2-3: 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 the Application Note AN556, "Implementing a Table Read" (DS00556).

### 2.3.2 STACK

The PIC16F688 family has an 8-level x 13-bit wide hardware stack (see Figure 2-1). 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.

# PIC16F688

## 2.4 Indirect Addressing, INDF and FSR Registers

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

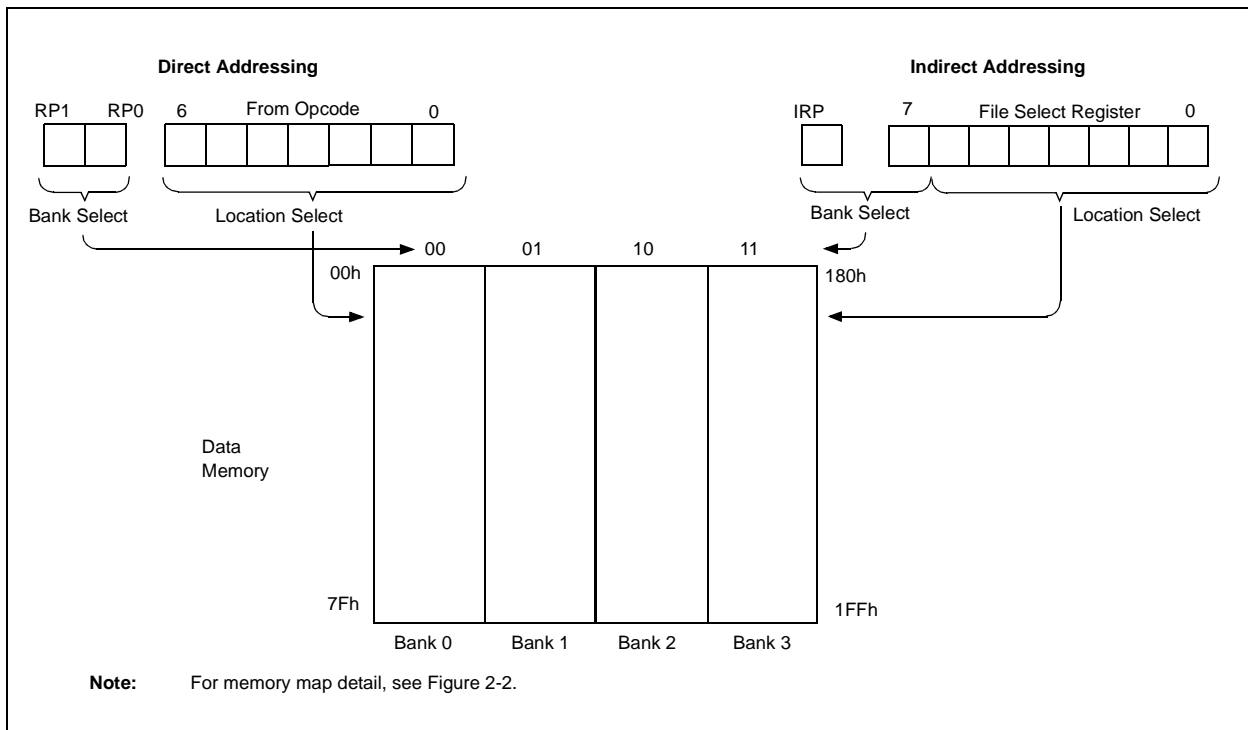
Indirect addressing is possible by using the INDF register. Any instruction using the INDF register actually accesses data pointed to by the File Select Register (FSR). Reading INDF itself indirectly will produce 00h. Writing to the INDF register indirectly results in a no operation (although Status 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 2-4.

A simple program to clear RAM location 20h-2Fh using indirect addressing is shown in Example 2-1.

### EXAMPLE 2-1: INDIRECT ADDRESSING

```
MOVLW 0x20 ;initialize pointer
MOVWF FSR ;to RAM
NEXT CLR F INDF ;clear INDF register
INCF FSR ;inc pointer
BTFSS FSR,4 ;all done?
GOTO NEXT ;no clear next
CONTINUE ;yes continue
```

FIGURE 2-4: DIRECT/INDIRECT ADDRESSING PIC16F688



## 3.0 OSCILLATOR MODULE (WITH FAIL-SAFE CLOCK MONITOR)

### 3.1 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 3-1 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 clock source can be configured from one of two internal oscillators, with a choice of speeds selectable via software. Additional clock features include:

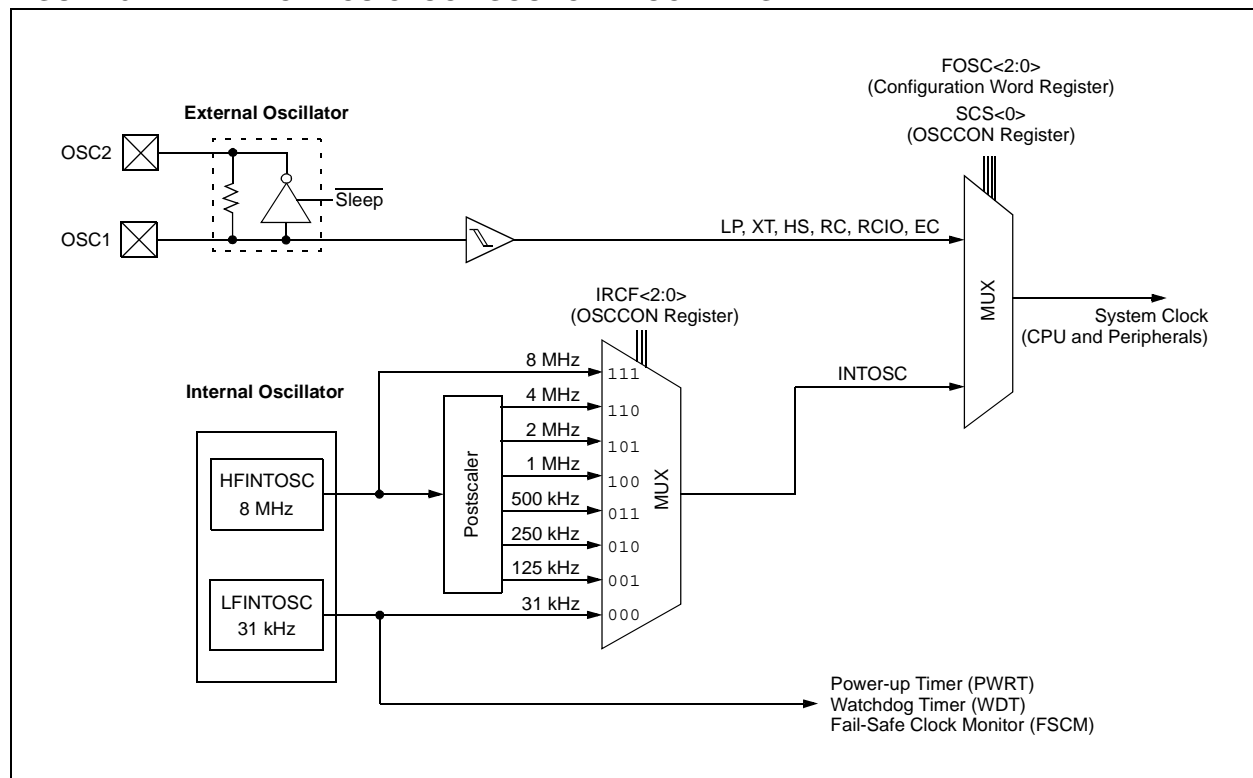
- Selectable system clock source between external or internal via software.
- Two-Speed Start-Up mode, which minimizes latency between external oscillator start-up and code execution.
- Fail-Safe Clock Monitor (FSCM) designed to detect a failure of the external clock source (LP, XT, HS, EC or RC modes) and switch automatically to the internal oscillator.

The Oscillator module can be configured in one of eight clock modes.

1. EC – External clock with I/O on OSC2/CLKOUT.
2. LP – 32 kHz Low-Power Crystal mode.
3. XT – Medium Gain Crystal or Ceramic Resonator Oscillator mode.
4. HS – High Gain Crystal or Ceramic Resonator mode.
5. RC – External Resistor-Capacitor (RC) with Fosc/4 output on OSC2/CLKOUT.
6. RCIO – External Resistor-Capacitor (RC) with I/O on OSC2/CLKOUT.
7. INTOSC – Internal oscillator with Fosc/4 output on OSC2 and I/O on OSC1/CLKIN.
8. INTOSCIO – Internal oscillator with I/O on OSC1/CLKIN and OSC2/CLKOUT.

Clock Source modes are configured by the FOSC<2:0> bits in the Configuration Word register (CONFIG). The internal clock can be generated from two internal oscillators. The HFINTOSC is a calibrated high-frequency oscillator. The LFINTOSC is an uncalibrated low-frequency oscillator.

**FIGURE 3-1: PIC® MCU CLOCK SOURCE BLOCK DIAGRAM**



# PIC16F688

## 3.2 Oscillator Control

The Oscillator Control (OSCCON) register (Figure 3-1) controls the system clock and frequency selection options. The OSCCON register contains the following bits:

- Frequency selection bits (IRCF)
- Frequency Status bits (HTS, LTS)
- System clock control bits (OSTS, SCS)

**REGISTER 3-1: OSCCON: OSCILLATOR CONTROL REGISTER**

U-0	R/W-1	R/W-1	R/W-0	R-1	R-0	R-0	R/W-0
—	IRCF2	IRCF1	IRCF0	OSTS <sup>(1)</sup>	HTS	LTS	SCS
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 **IRCF<2:0>:** Internal Oscillator Frequency Select bits

- 111 = 8 MHz
- 110 = 4 MHz (default)
- 101 = 2 MHz
- 100 = 1 MHz
- 011 = 500 kHz
- 010 = 250 kHz
- 001 = 125 kHz
- 000 = 31 kHz (LFINTOSC)

bit 3 **OSTS:** Oscillator Start-up Time-out Status bit<sup>(1)</sup>

- 1 = Device is running from the external clock defined by FOSC<2:0> of the Configuration Word
- 0 = Device is running from the internal oscillator (HFINTOSC or LFINTOSC)

bit 2 **HTS:** HFINTOSC Status bit (High Frequency – 8 MHz to 125 kHz)

- 1 = HFINTOSC is stable
- 0 = HFINTOSC is not stable

bit 1 **LTS:** LFINTOSC Stable bit (Low Frequency – 31 kHz)

- 1 = LFINTOSC is stable
- 0 = LFINTOSC is not stable

bit 0 **SCS:** System Clock Select bit

- 1 = Internal oscillator is used for system clock
- 0 = Clock source defined by FOSC<2:0> of the Configuration Word

**Note 1:** Bit resets to '0' with Two-Speed Start-up and LP, XT or HS selected as the Oscillator mode or Fail-Safe mode is enabled.



## 3.3 Clock Source Modes

Clock Source modes can be classified as external or internal.

- 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.
- Internal clock sources are contained internally within the Oscillator module. The Oscillator module has two internal oscillators: the 8 MHz High-Frequency Internal Oscillator (HFINTOSC) and the 31 kHz Low-Frequency Internal Oscillator (LFINTOSC).

The system clock can be selected between external or internal clock sources via the System Clock Select (SCS) bit of the OSCCON register. See **Section 3.6 “Clock Switching”** for additional information.

## 3.4 External Clock Modes

### 3.4.1 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 from OSC1. 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. When switching between clock sources, a delay is required to allow the new clock to stabilize. These oscillator delays are shown in Table 3-1.

In order to minimize latency between external oscillator start-up and code execution, the Two-Speed Clock Start-up mode can be selected (see **Section 3.7 “Two-Speed Clock Start-up Mode”**).

**TABLE 3-1: OSCILLATOR DELAY EXAMPLES**

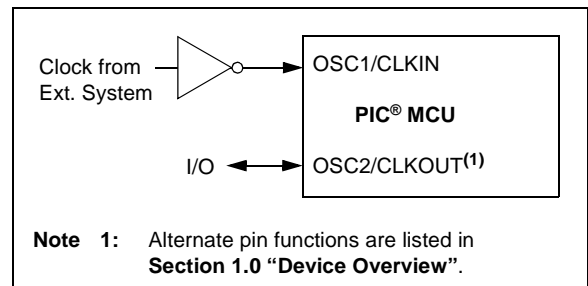
Switch From	Switch To	Frequency	Oscillator Delay
Sleep/POR	LFINTOSC HFINTOSC	31 kHz 125 kHz to 8 MHz	Oscillator Warm-Up Delay (TWARM)
Sleep/POR	EC, RC	DC – 20 MHz	2 instruction cycles
LFINTOSC (31 kHz)	EC, RC	DC – 20 MHz	1 cycle of each
Sleep/POR	LP, XT, HS	32 kHz to 20 MHz	1024 Clock Cycles (OST)
LFINTOSC (31 kHz)	HFINTOSC	125 kHz to 8 MHz	1 $\mu$ s (approx.)

### 3.4.2 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 3-2 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<sup>®</sup> 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 3-2: EXTERNAL CLOCK (EC) MODE OPERATION**



# PIC16F688

## 3.4.3 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 3-3). The mode selects a low, medium or high gain setting of the internal inverter-amplifier to support various resonator types and speed.

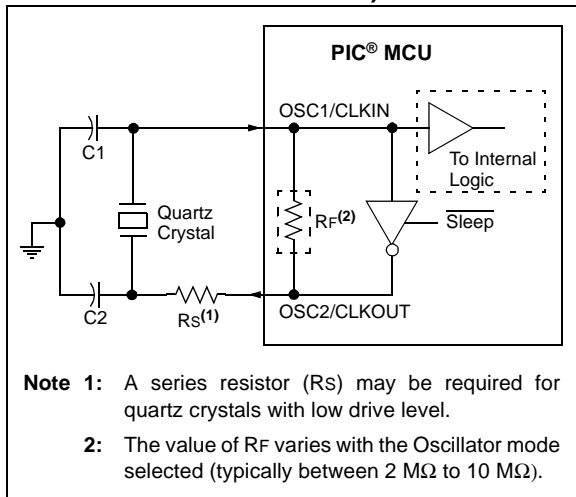
**LP** Oscillator mode selects the lowest gain setting of the internal inverter-amplifier. LP mode current consumption is the least of the three modes. This mode is best suited to drive resonators with a low drive level specification, for example, tuning fork type crystals. This mode is designed to drive only 32.768 kHz tuning-fork type crystals (watch 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 3-3 and Figure 3-4 show typical circuits for quartz crystal and ceramic resonators, respectively.

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



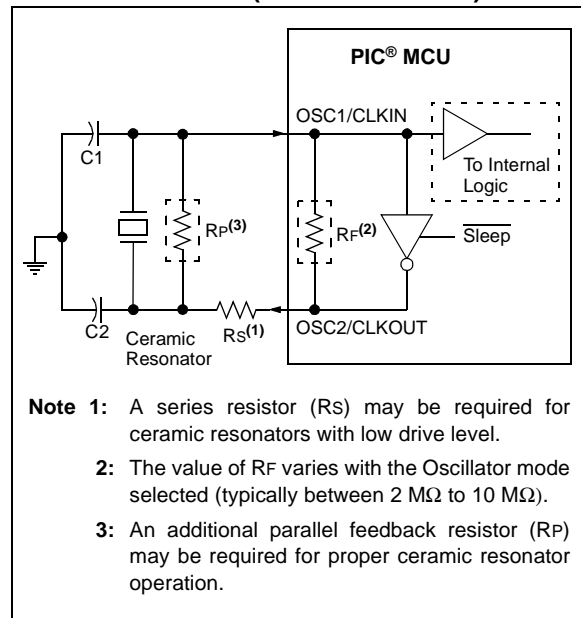
**Note 1:** Quartz crystal characteristics vary according to type, package and manufacturer. The user should consult the manufacturer data sheets for specifications and recommended application.

**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 rPIC<sup>®</sup> and PIC<sup>®</sup> Devices" (DS00826)
- AN849, "Basic PIC<sup>®</sup> Oscillator Design" (DS00849)
- AN943, "Practical PIC<sup>®</sup> Oscillator Analysis and Design" (DS00943)
- AN949, "Making Your Oscillator Work" (DS00949)

**FIGURE 3-4: CERAMIC RESONATOR OPERATION (XT OR HS MODE)**

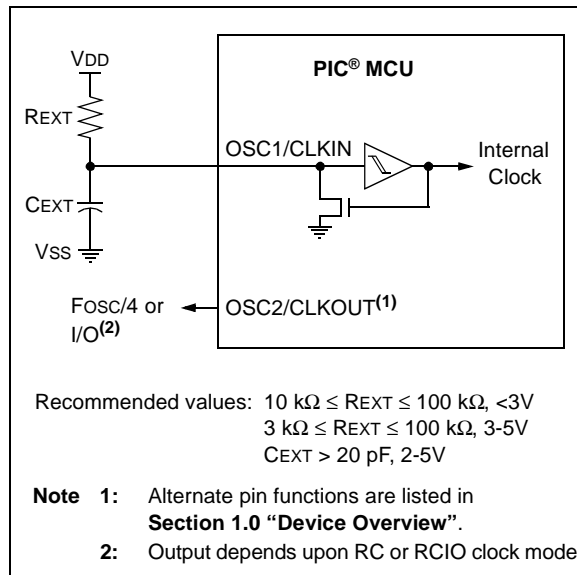


## 3.4.4 EXTERNAL RC MODES

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 3-5 shows the external RC mode connections.

**FIGURE 3-5: EXTERNAL RC MODES**



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 ( $R_{EXT}$ ) and capacitor ( $C_{EXT}$ ) 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.

## 3.5 Internal Clock Modes

The Oscillator module has two independent, internal oscillators that can be configured or selected as the system clock source.

1. The **HFINTOSC** (High-Frequency Internal Oscillator) is factory calibrated and operates at 8 MHz. The frequency of the HFINTOSC can be user-adjusted via software using the OSCTUNE register (Register 3-2).
2. The **LFINTOSC** (Low-Frequency Internal Oscillator) is uncalibrated and operates at 31 kHz.

The system clock speed can be selected via software using the Internal Oscillator Frequency Select bits  $IRCF\langle 2:0 \rangle$  of the OSCCON register.

The system clock can be selected between external or internal clock sources via the System Clock Selection (SCS) bit of the OSCCON register. See **Section 3.6 "Clock Switching"** for more information.

### 3.5.1 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  $FOSC\langle 2:0 \rangle$  bits in the Configuration Word register (CONFIG). See **Section 11.0 "Special Features of the CPU"** 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.

### 3.5.2 HFINTOSC

The High-Frequency Internal Oscillator (HFINTOSC) is a factory calibrated 8 MHz internal clock source. The frequency of the HFINTOSC can be altered via software using the OSCTUNE register (Register 3-2).

The output of the HFINTOSC connects to a postscaler and multiplexer (see Figure 3-1). One of seven frequencies can be selected via software using the  $IRCF\langle 2:0 \rangle$  bits of the OSCCON register. See **Section 3.5.4 "Frequency Select Bits (IRCF)"** for more information.

The HFINTOSC is enabled by selecting any frequency between 8 MHz and 125 kHz by setting the  $IRCF\langle 2:0 \rangle$  bits of the OSCCON register  $\neq 000$ . Then, set the System Clock Source (SCS) bit of the OSCCON register to '1' or enable Two-Speed Start-up by setting the IESO bit in the Configuration Word register (CONFIG) to '1'.

The HF Internal Oscillator (HTS) bit of the OSCCON register indicates whether the HFINTOSC is stable or not.

# PIC16F688

## 3.5.2.1 OSCTUNE Register

The HFINTOSC is factory calibrated but can be adjusted in software by writing to the OSCTUNE register (Register 3-2).

The default value of the OSCTUNE register is '0'. The value is a 5-bit two's complement number.

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

OSCTUNE does not affect the LFINTOSC frequency. Operation of features that depend on the LFINTOSC clock source frequency, such as the Power-up Timer (PWRT), Watchdog Timer (WDT), Fail-Safe Clock Monitor (FSCM) and peripherals, are *not* affected by the change in frequency.

### REGISTER 3-2: OSCTUNE: OSCILLATOR TUNING REGISTER

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	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 7-5      **Unimplemented:** Read as '0'

bit 4-0      **TUN<4:0>:** Frequency Tuning bits

01111 = Maximum frequency

01110 =

•

•

•

00001 =

00000 = Oscillator module is running at the calibrated frequency.

11111 =

•

•

•

10000 = Minimum frequency

### 3.5.3 LFINTOSC

The Low-Frequency Internal Oscillator (LFINTOSC) is an uncalibrated 31 kHz internal clock source.

The output of the LFINTOSC connects to a postscaler and multiplexer (see Figure 3-1). Select 31 kHz, via software, using the IRCF<2:0> bits of the OSCCON register. See **Section 3.5.4 “Frequency Select Bits (IRCF)”** for more information. The LFINTOSC is also the frequency for the Power-up Timer (PWRT), Watchdog Timer (WDT) and Fail-Safe Clock Monitor (FSCM).

The LFINTOSC is enabled by selecting 31 kHz (IRCF<2:0> bits of the OSCCON register = 000) as the system clock source (SCS bit of the OSCCON register = 1), or when any of the following are enabled:

- Two-Speed Start-up IESO bit of the Configuration Word register = 1 and IRCF<2:0> bits of the OSCCON register = 000
- Power-up Timer (PWRT)
- Watchdog Timer (WDT)
- Fail-Safe Clock Monitor (FSCM)

The LF Internal Oscillator (LTS) bit of the OSCCON register indicates whether the LFINTOSC is stable or not.

### 3.5.4 FREQUENCY SELECT BITS (IRCF)

The output of the 8 MHz HFINTOSC and 31 kHz LFINTOSC connects to a postscaler and multiplexer (see Figure 3-1). The Internal Oscillator Frequency Select bits IRCF<2:0> of the OSCCON register select the frequency output of the internal oscillators. One of eight frequencies can be selected via software:

- 8 MHz
- 4 MHz (Default after Reset)
- 2 MHz
- 1 MHz
- 500 kHz
- 250 kHz
- 125 kHz
- 31 kHz (LFINTOSC)

**Note:** Following any Reset, the IRCF<2:0> bits of the OSCCON register are set to '110' and the frequency selection is set to 4 MHz. The user can modify the IRCF bits to select a different frequency.

### 3.5.5 HF AND LF INTOSC CLOCK SWITCH TIMING

When switching between the LFINTOSC and the HFINTOSC, the new oscillator may already be shut down to save power (see Figure 3-6). If this is the case, there is a delay after the IRCF<2:0> bits of the OSCCON register are modified before the frequency selection takes place. The LTS and HTS bits of the OSCCON register will reflect the current active status of the LFINTOSC and HFINTOSC oscillators. The timing of a frequency selection is as follows:

1. IRCF<2:0> bits of the OSCCON register are modified.
2. If the new clock is shut down, a clock start-up delay is started.
3. Clock switch circuitry waits for a falling edge of the current clock.
4. CLKOUT is held low and the clock switch circuitry waits for a rising edge in the new clock.
5. CLKOUT is now connected with the new clock. LTS and HTS bits of the OSCCON register are updated as required.
6. Clock switch is complete.

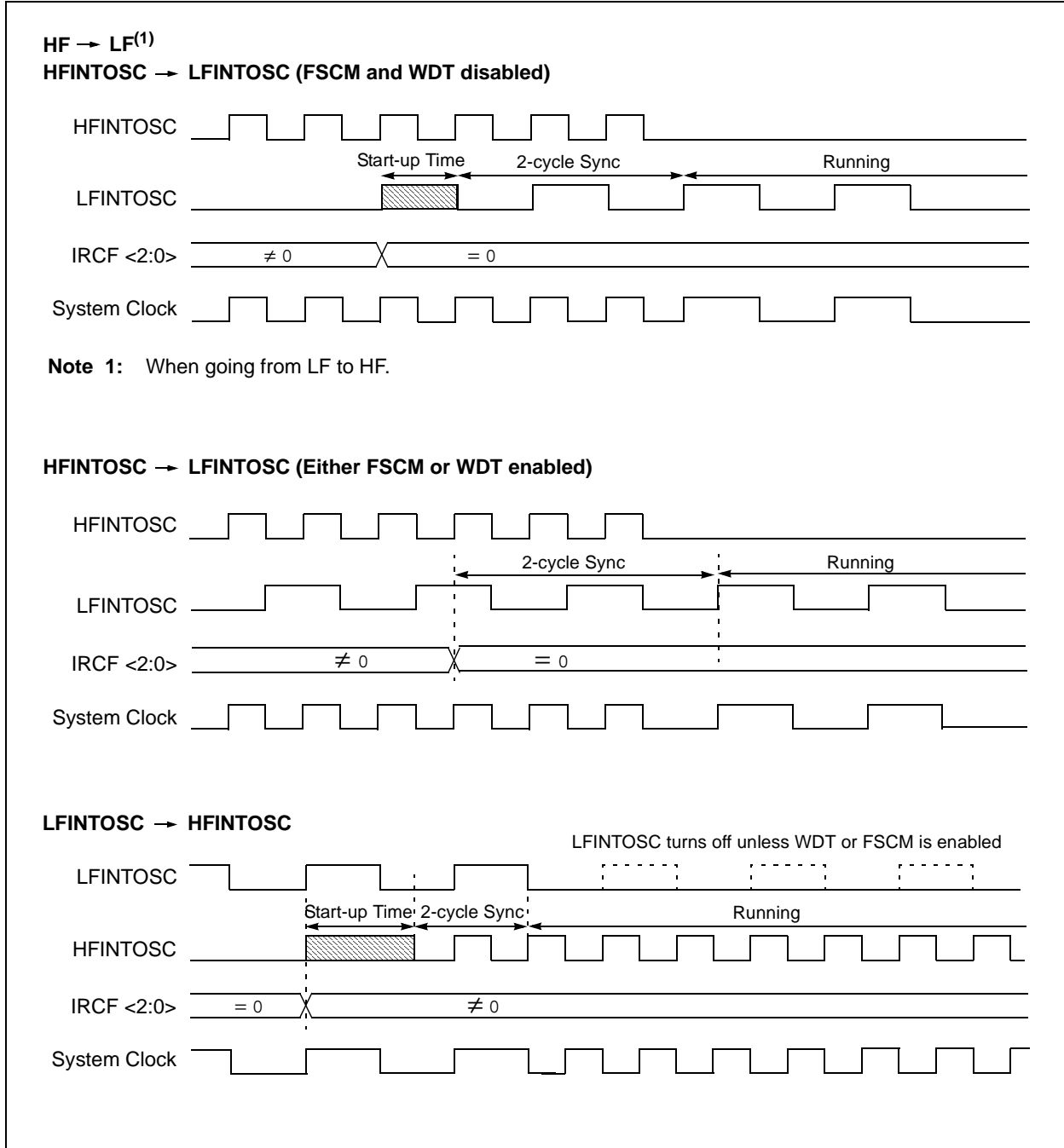
See Figure 3-1 for more details.

If the internal oscillator speed selected is between 8 MHz and 125 kHz, there is no start-up delay before the new frequency is selected. This is because the old and new frequencies are derived from the HFINTOSC via the postscaler and multiplexer.

Start-up delay specifications are located in the **Section 14.0 “Electrical Specifications”**, under the AC Specifications (Oscillator Module).

# PIC16F688

**FIGURE 3-6: INTERNAL OSCILLATOR SWITCH TIMING**



## 3.6 Clock Switching

The system clock source can be switched between external and internal clock sources via software using the System Clock Select (SCS) bit of the OSCCON register.

### 3.6.1 SYSTEM CLOCK SELECT (SCS) BIT

The System Clock Select (SCS) bit of the OSCCON register selects the system clock source that is used for the CPU and peripherals.

- When the SCS bit of the OSCCON register = 0, the system clock source is determined by configuration of the FOSC<2:0> bits in the Configuration Word register (CONFIG).
- When the SCS bit of the OSCCON register = 1, the system clock source is chosen by the internal oscillator frequency selected by the IRCF<2:0> bits of the OSCCON register. After a Reset, the SCS bit of the OSCCON register is always cleared.

**Note:** Any automatic clock switch, which may occur from Two-Speed Start-up or Fail-Safe Clock Monitor, does not update the SCS bit of the OSCCON register. The user can monitor the OSTs bit of the OSCCON register to determine the current system clock source.

### 3.6.2 OSCILLATOR START-UP TIME-OUT STATUS (OSTS) BIT

The Oscillator Start-up Time-out Status (OSTS) bit of the OSCCON register indicates whether the system clock is running from the external clock source, as defined by the FOSC<2:0> bits in the Configuration Word register (CONFIG), or from the internal clock source. In particular, OSTs indicates that the Oscillator Start-up Timer (OST) has timed out for LP, XT or HS modes.

## 3.7 Two-Speed Clock Start-up Mode

Two-Speed Start-up mode provides additional power savings by minimizing the latency between external oscillator start-up and code execution. In applications that make heavy use of the Sleep mode, Two-Speed Start-up will remove the external oscillator start-up time from the time spent awake and can reduce the overall power consumption of the device.

This mode allows the application to wake-up from Sleep, perform a few instructions using the INTOSC as the clock source and go back to Sleep without waiting for the primary oscillator to become stable.

**Note:** Executing a SLEEP instruction will abort the oscillator start-up time and will cause the OSTs bit of the OSCCON register to remain clear.

When the Oscillator module is configured for LP, XT or HS modes, the Oscillator Start-up Timer (OST) is enabled (see **Section 3.4.1 “Oscillator Start-up Timer (OST)”**). The OST will suspend program execution until 1024 oscillations are counted. Two-Speed Start-up mode minimizes the delay in code execution by operating from the internal oscillator as the OST is counting. When the OST count reaches 1024 and the OSTs bit of the OSCCON register is set, program execution switches to the external oscillator.

### 3.7.1 TWO-SPEED START-UP MODE CONFIGURATION

Two-Speed Start-up mode is configured by the following settings:

- IESO (of the Configuration Word register) = 1; Internal/External Switchover bit (Two-Speed Start-up mode enabled).
- SCS (of the OSCCON register) = 0.
- FOSC<2:0> bits in the Configuration Word register (CONFIG) configured for LP, XT or HS mode.

Two-Speed Start-up mode is entered after:

- Power-on Reset (POR) and, if enabled, after Power-up Timer (PWRT) has expired, or
- Wake-up from Sleep.

If the external clock oscillator is configured to be anything other than LP, XT or HS mode, then Two-Speed Start-up is disabled. This is because the external clock oscillator does not require any stabilization time after POR or an exit from Sleep.

### 3.7.2 TWO-SPEED START-UP SEQUENCE

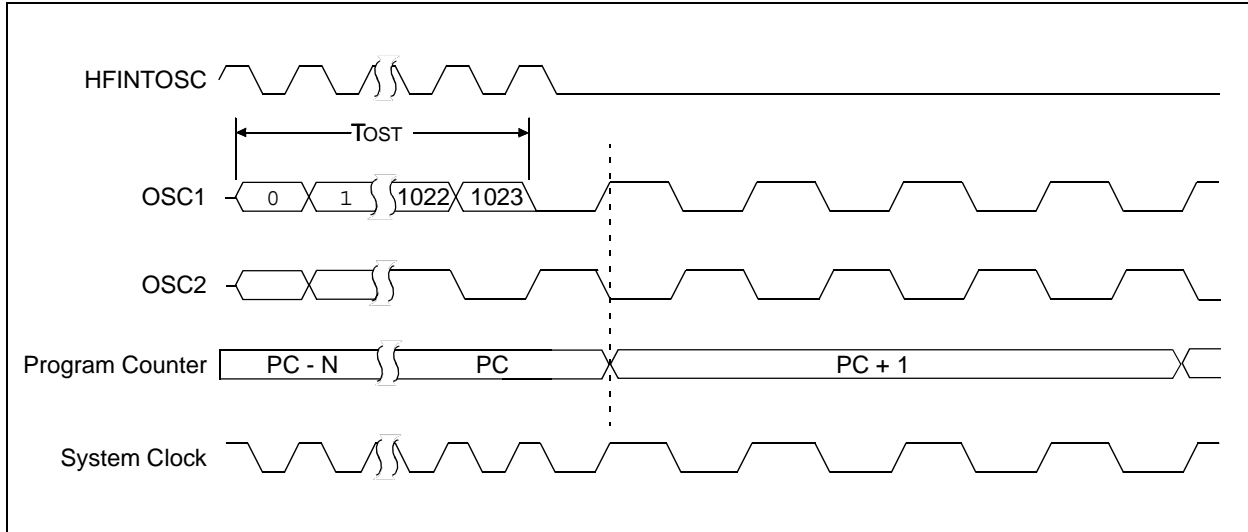
1. Wake-up from Power-on Reset or Sleep.
2. Instructions begin execution by the internal oscillator at the frequency set in the IRCF<2:0> bits of the OSCCON register.
3. OST enabled to count 1024 clock cycles.
4. OST timed out, wait for falling edge of the internal oscillator.
5. OSTs is set.
6. System clock held low until the next falling edge of new clock (LP, XT or HS mode).
7. System clock is switched to external clock source.

# PIC16F688

## 3.7.3 CHECKING TWO-SPEED CLOCK STATUS

Checking the state of the OSTS bit of the OSCCON register will confirm if the microcontroller is running from the external clock source, as defined by the FOSC<2:0> bits in the Configuration Word register (CONFIG), or the internal oscillator.

**FIGURE 3-7: TWO-SPEED START-UP**

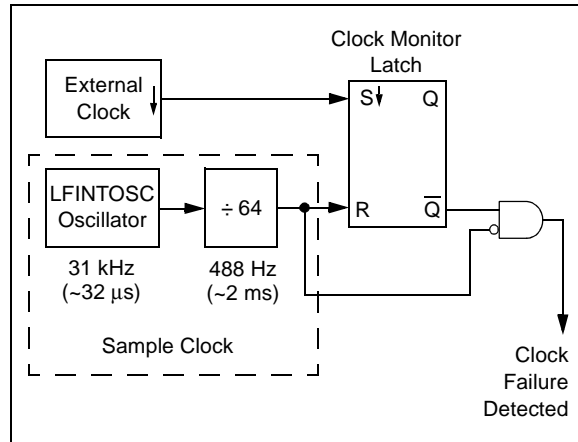




## 3.8 Fail-Safe Clock Monitor

The Fail-Safe Clock Monitor (FSCM) allows the device to continue operating should the external oscillator fail. The FSCM can detect oscillator failure any time after the Oscillator Start-up Timer (OST) has expired. The FSCM is enabled by setting the FCMEN bit in the Configuration Word register (CONFIG). The FSCM is applicable to all external oscillator modes (LP, XT, HS, EC, RC and RCIO).

**FIGURE 3-8: FSCM BLOCK DIAGRAM**



### 3.8.1 FAIL-SAFE DETECTION

The FSCM module detects a failed oscillator by comparing the external oscillator to the FSCM sample clock. The sample clock is generated by dividing the LFINTOSC by 64. See Figure 3-8. Inside the fail detector block is a latch. The external clock sets the latch on each falling edge of the external clock. The sample clock clears the latch on each rising edge of the sample clock. A failure is detected when an entire half-cycle of the sample clock elapses before the primary clock goes low.

### 3.8.2 FAIL-SAFE OPERATION

When the external clock fails, the FSCM switches the device clock to an internal clock source and sets the bit flag OSFIF of the PIR2 register. Setting this flag will generate an interrupt if the OSFIE bit of the PIE2 register is also set. The device firmware can then take steps to mitigate the problems that may arise from a failed clock. The system clock will continue to be sourced from the internal clock source until the device firmware successfully restarts the external oscillator and switches back to external operation.

The internal clock source chosen by the FSCM is determined by the IRCF<2:0> bits of the OSCCON register. This allows the internal oscillator to be configured before a failure occurs.

### 3.8.3 FAIL-SAFE CONDITION CLEARING

The Fail-Safe condition is cleared after a Reset, executing a SLEEP instruction or toggling the SCS bit of the OSCCON register. When the SCS bit is toggled, the OST is restarted. While the OST is running, the device continues to operate from the INTOSC selected in OSCCON. When the OST times out, the Fail-Safe condition is cleared and the device will be operating from the external clock source. The Fail-Safe condition must be cleared before the OSFIF flag can be cleared.

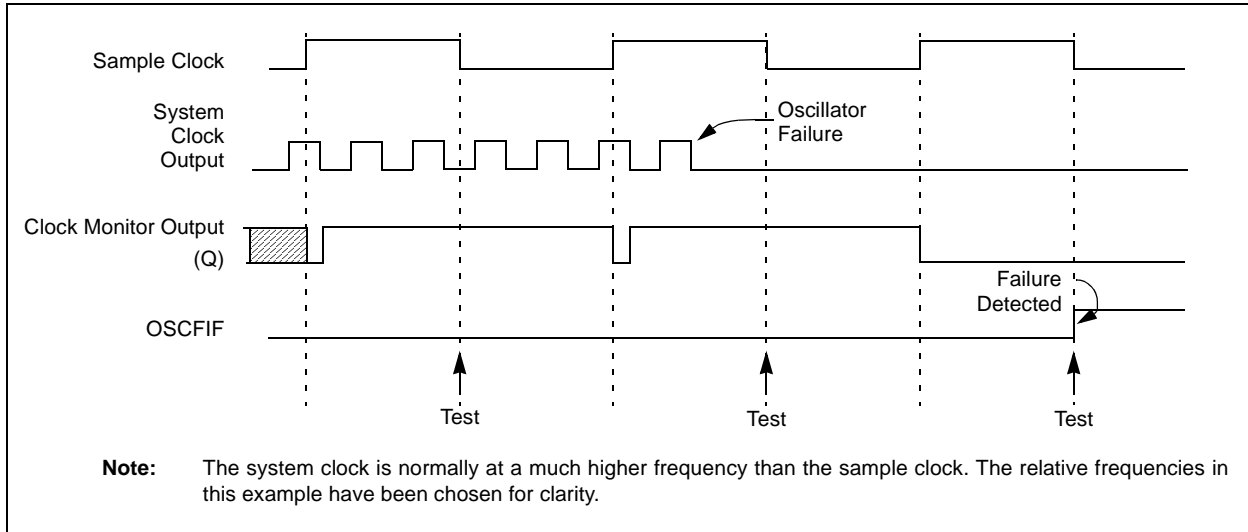
### 3.8.4 RESET OR WAKE-UP FROM SLEEP

The FSCM is designed to detect an oscillator failure after the Oscillator Start-up Timer (OST) has expired. The OST is used after waking up from Sleep and after any type of Reset. The OST is not used with the EC or RC Clock modes so that the FSCM will be active as soon as the Reset or wake-up has completed. When the FSCM is enabled, the Two-Speed Start-up is also enabled. Therefore, the device will always be executing code while the OST is operating.

**Note:** Due to the wide range of oscillator start-up times, the Fail-Safe circuit is not active during oscillator start-up (i.e., after exiting Reset or Sleep). After an appropriate amount of time, the user should check the OSTS bit of the OSCCON register to verify the oscillator start-up and that the system clock switchover has successfully completed.

# PIC16F688

**FIGURE 3-9: FSCM TIMING DIAGRAM**



**TABLE 3-2: SUMMARY OF REGISTERS ASSOCIATED WITH CLOCK SOURCES**

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 <sup>(1)</sup>
CONFIG <sup>(2)</sup>	CPD	CP	MCLRE	PWRTE	WDTE	FOSC2	FOSC1	FOSC0	—	—
INTCON	GIE	PEIE	TOIE	INTE	RAIE	TOIF	INTF	RAIF	0000 000x	0000 000x
OSCCON	—	IRCF2	IRCF1	IRCF0	OSTS	HTS	LTS	SCS	-110 x000	-110 x000
OSCTUNE	—	—	—	TUN4	TUN3	TUN2	TUN1	TUN0	---0 0000	---u uuuu
PIE1	EEIE	ADIE	RCIE	C2IE	C1IE	OSFIE	TXIE	TMR1IE	0000 0000	0000 0000
PIR1	EEIF	ADIF	RCIF	C2IF	C1IF	OSFIF	TXIF	TMR1IF	0000 0000	0000 0000

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

**Note 1:** Other (non Power-up) Resets include MCLR Reset and Watchdog Timer Reset during normal operation.

**2:** See Configuration Word register (CONFIG) for operation of all register bits.

## 4.0 I/O PORTS

There are as many as twelve 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.

### 4.1 PORTA and the TRISA Registers

PORTA is a 6-bit wide, bidirectional port. The corresponding data direction register is TRISA. Setting a TRISA bit (= 1) will make the corresponding PORTA pin an input (i.e., put the corresponding output driver in a High-Impedance mode). Clearing a TRISA bit (= 0) will make the corresponding PORTA pin an output (i.e., put the contents of the output latch on the selected pin). The exception is RA3, which is input only and its TRISA bit will always read as '1'. Example 4-1 shows how to initialize PORTA.

Reading the PORTA register 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. RA3 reads '0' when MCLRE = 1.

The TRISA register controls the direction of the PORTA pins, even when they are being used as analog inputs. The user must 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 ANSEL and CMCON0 registers must be initialized to configure an analog channel as a digital input. Pins configured as analog inputs will read '0'.

#### EXAMPLE 4-1: INITIALIZING PORTA

```
BANKSEL PORTA      ;
CLRF  PORTA        ;Init PORTA
MOVLW 07h          ;Set RA<2:0> to
MOVWF  CMCON0      ;digital I/O
BANKSEL ANSEL      ;
CLRF  ANSEL        ;digital I/O
MOVLW 0Ch          ;Set RA<3:2> as inputs
MOVWF  TRISA       ;and set RA<5:4,1:0>
                  ;as outputs
```

#### REGISTER 4-1: PORTA: PORTA REGISTER

U-0	U-0	R/W-x	R/W-0	R-x	R/W-0	R/W-0	R/W-0
—	—	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 7-6                      **Unimplemented:** Read as '0'  
 bit 5-0                      **RA<5:0>:** PORTA I/O Pin bit  
                                  1 = Port pin is > V<sub>IH</sub>  
                                  0 = Port pin is < V<sub>IL</sub>

#### REGISTER 4-2: TRISA: PORTA TRI-STATE REGISTER

U-0	U-0	R/W-1	R/W-1	R-1	R/W-1	R/W-1	R/W-1
—	—	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 7-6                      **Unimplemented:** Read as '0'  
 bit 5-0                      **TRISA<5:0>:** PORTA Tri-State Control bits  
                                  1 = PORTA pin configured as an input (tri-stated)  
                                  0 = PORTA pin configured as an output

**Note 1:** TRISA<3> always reads '1'.  
**Note 2:** TRISA<5:4> always reads '1' in XT, HS and LP Oscillator modes.

# PIC16F688

## 4.2 Additional Pin Functions

Every PORTA pin on the PIC16F688 has an interrupt-on-change option and a weak pull-up option. PORTA also provides an Ultra Low-Power Wake-up option. The next three sections describe these functions.

### 4.2.1 ANSEL REGISTER

The ANSEL register is used to configure the Input mode of an I/O pin to analog. Refer to Register 4-3. Setting the appropriate ANSEL 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 ANSEL 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.

### 4.2.2 WEAK PULL-UPS

Each of the PORTA pins, except RA3, has an individually configurable internal weak pull-up. Control bits WPUAx enable or disable each pull-up. Refer to Register 4-4. Each weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are disabled on a Power-on Reset by the RAPU bit of the OPTION register. A weak pull-up is automatically enabled for RA3 when configured as MCLR and disabled when RA3 is an I/O. There is no software control of the MCLR pull-up.

### 4.2.3 INTERRUPT-ON-CHANGE

Each of the PORTA pins is individually configurable as an interrupt-on-change pin. Control bits IOCAx enable or disable the interrupt function for each pin. Refer to Register 4-5. The interrupt-on-change is disabled on a Power-on Reset.

For enabled interrupt-on-change pins, the values are compared with the old value latched on the last read of PORTA. The 'mismatch' outputs of the last read are OR'd together to set the PORTA Change Interrupt Flag bit (RAIF) 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 PORTA. This will end the mismatch condition, then
- b) Clear the flag bit RAIF.

A mismatch condition will continue to set flag bit RAIF. Reading PORTA will end the mismatch condition and allow flag bit RAIF to be cleared. The latch holding the last read value is not affected by a MCLR nor BOR Reset. After these Resets, the RAIF flag will continue to be set if a mismatch is present.

**Note:** If a change on the I/O pin should occur when the read operation is being executed (start of the Q2 cycle), then the RAIF interrupt flag may not get set.

### REGISTER 4-3: ANSEL: ANALOG SELECT 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
ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0
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      **ANS<7:0>**: Analog Select bits  
 Analog select between analog or digital function on pins AN<7:0>, respectively.  
 1 = Analog input. Pin is assigned as analog input<sup>(1)</sup>.  
 0 = Digital I/O. Pin is assigned to port or special function.

**Note 1:** Setting a pin to an analog input automatically disables the digital input circuitry, weak pull-ups and interrupt-on-change, if available. The corresponding TRIS bit must be set to Input mode in order to allow external control of the voltage on the pin.

## REGISTER 4-4: WPUA: WEAK PULL-UP PORTA REGISTER

U-0	U-0	R/W-1	R/W-1	U-0	R/W-1	R/W-1	R/W-1
—	—	WPUA5	WPUA4	—	WPUA2	WPUA1	WPUA0
bit 7							bit 0

### Legend:

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

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

bit 5-4 **WPUA<5:4>:** Weak Pull-up Control bits  
 1 = Pull-up enabled  
 0 = Pull-up disabled

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

bit 2-0 **WPUA<2:0>:** Weak Pull-up Control bits  
 1 = Pull-up enabled  
 0 = Pull-up disabled

- Note 1:** Global  $\overline{\text{RAPU}}$  must be enabled for individual pull-ups to be enabled.  
**Note 2:** The weak pull-up device is automatically disabled if the pin is in Output mode ( $\text{TRISA} = 0$ ).  
**Note 3:** The RA3 pull-up is enabled when configured as  $\overline{\text{MCLR}}$  and disabled as an I/O in the Configuration Word.  
**Note 4:** WPUA<5:4> always reads '1' in XT, HS and LP OSC modes.

## REGISTER 4-5: IOCA: INTERRUPT-ON-CHANGE PORTA REGISTER

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	IOCA5	IOCA4	IOCA3	IOCA2	IOCA1	IOCA0
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 **IOCA<5:0>:** Interrupt-on-change PORTA Control bits  
 1 = Interrupt-on-change enabled  
 0 = Interrupt-on-change disabled

- Note 1:** Global Interrupt Enable (GIE) must be enabled for individual interrupts to be recognized.  
**Note 2:** IOCA<5:4> always reads '1' in XT, HS and LP OSC modes.

# PIC16F688

## 4.2.4 ULTRA LOW-POWER WAKE-UP

The Ultra Low-Power Wake-up (ULPWU) on RA0 allows a slow falling voltage to generate an interrupt-on-change on RA0 without excess current consumption. The mode is selected by setting the ULPWUE bit of the PCON register. This enables a small current sink which can be used to discharge a capacitor on RA0.

To use this feature, the RA0 pin is configured to output '1' to charge the capacitor, interrupt-on-change for RA0 is enabled, and RA0 is configured as an input. The ULPWUE bit is set to begin the discharge and a SLEEP instruction is performed. When the voltage on RA0 drops below  $V_{IL}$ , an interrupt will be generated which will cause the device to wake-up. Depending on the state of the GIE bit of the INTCON register, the device will either jump to the interrupt vector (0004h) or execute the next instruction when the interrupt event occurs. See **Section 4.2.3 "INTERRUPT-ON-CHANGE"** and **Section 11.3.3 "PORTA Interrupt"** for more information.

This feature provides a low-power technique for periodically waking up the device from Sleep. The time-out is dependent on the discharge time of the RC circuit on RA0. See Example 4-2 for initializing the Ultra Low-Power Wake-up module.

The series resistor provides overcurrent protection for the RA0 pin and can allow for software calibration of the time-out. (see Figure 4-1). A timer can be used to measure the charge time and discharge time of the capacitor. The charge time can then be adjusted to provide the desired interrupt delay. This technique will compensate for the affects of temperature, voltage and component accuracy. The Ultra Low-Power Wake-up peripheral can also be configured as a simple programmable low voltage detect or temperature sensor.

**Note:** For more information, refer to Application Note AN879, "Using the Microchip Ultra Low-Power Wake-up Module" (DS00879).

## EXAMPLE 4-2: ULTRA LOW-POWER WAKE-UP INITIALIZATION

```
BANKSEL PORTA      ;
BSF      PORTA,0   ;Set RA0 data latch
MOVLW   H'7'      ;Turn off
MOVWF   CMCON0    ; comparators
BANKSEL ANSEL      ;
BCF     ANSEL,0    ;RA0 to digital I/O
BANKSEL TRISA      ;
BCF     TRISA,0    ;Output high to
CALL   CapDelay   ; charge capacitor
BSF    PCON,ULPWUE ;Enable ULP Wake-up
BSF    IOCA,0     ;Select RA0 IOC
BSF    TRISA,0    ;RA0 to input
MOVLW  B'10001000' ;Enable interrupt
MOVWF  INTCON     ; and clear flag
SLEEP                          ;Wait for IOC
NOP                                  ;
```





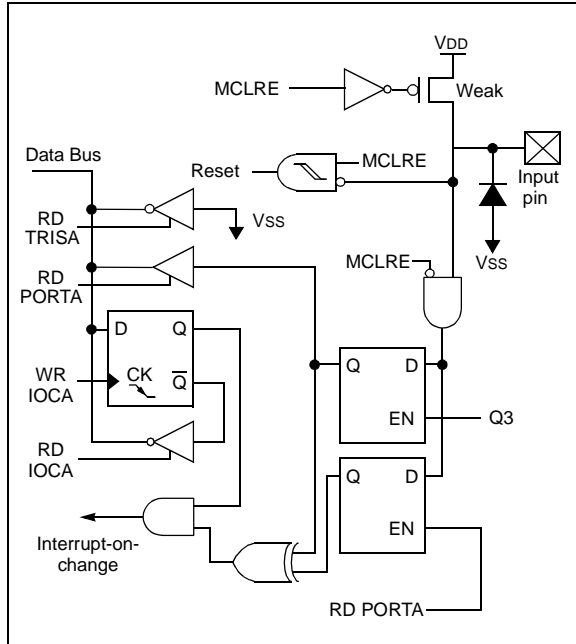


## 4.2.5.4 RA3/MCLR/VPP

Figure 4-4 shows the diagram for this pin. The RA3 pin is configurable to function as one of the following:

- a general purpose input
- as Master Clear Reset with weak pull-up

**FIGURE 4-4: BLOCK DIAGRAM OF RA3**

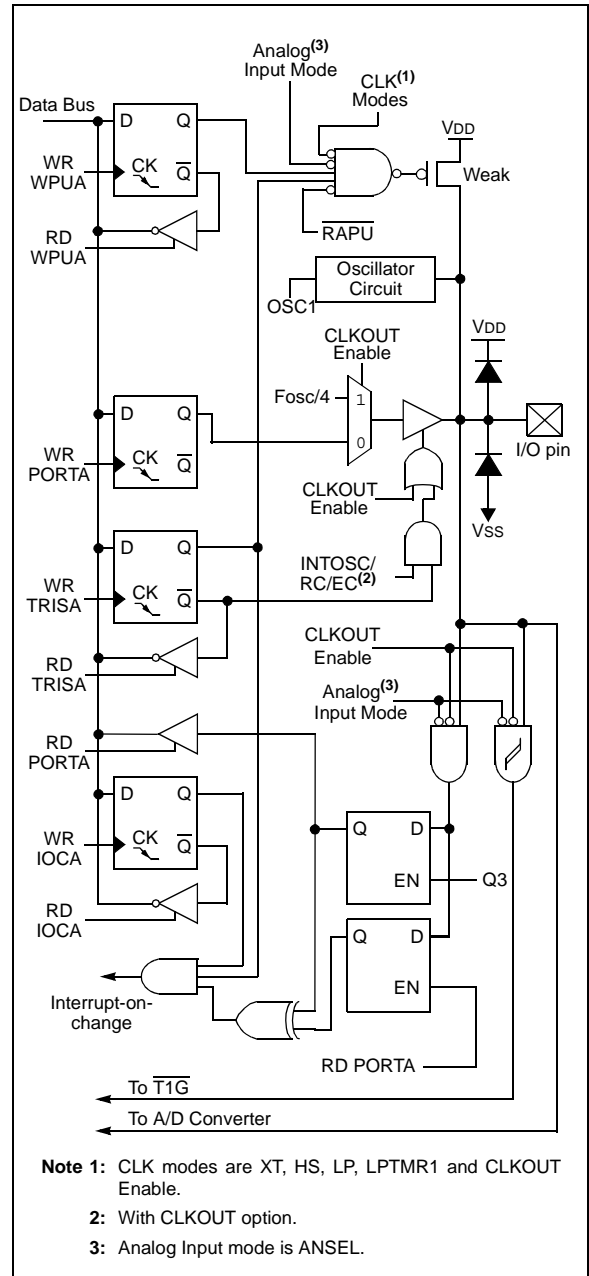


## 4.2.5.5 RA4/AN3/T1G/OSC2/CLKOUT

Figure 4-5 shows the diagram for this pin. The RA4 pin is configurable to function as one of the following:

- a general purpose I/O
- an analog input for the A/D
- a Timer1 gate input
- a crystal/resonator connection
- a clock output

**FIGURE 4-5: BLOCK DIAGRAM OF RA4**





**TABLE 4-1: SUMMARY OF REGISTERS ASSOCIATED WITH PORTA**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
ANSEL	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0	1111 1111	1111 1111
CMCON0	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	0000 0000	0000 0000
PCON	—	—	ULPWUE	SBOREN	—	—	POR	BOR	--01 --qq	--0u --uu
INTCON	GIE	PEIE	T0IE	INTE	RAIE	T0IF	INTF	RAIF	0000 000x	0000 000x
IOCA	—	—	IOCA5	IOCA4	IOCA3	IOCA2	IOCA1	IOCA0	--00 0000	--00 0000
OPTION_REG	RAPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	--x0 x000	--x0 x000
TRISA	—	—	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	--11 1111	--11 1111
WPUA	—	—	WPUA5	WPUA4	—	WPUA2	WPUA1	WPUA0	--11 -111	--11 -111

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

# PIC16F688

## 4.3 PORTC

PORTC is a general purpose I/O port consisting of 6 bidirectional pins. The pins can be configured for either digital I/O or analog input to A/D converter or comparator. For specific information about individual functions such as the EUSART or the A/D converter, refer to the appropriate section in this data sheet.

**Note:** The ANSEL and CMCON0 registers must be initialized to configure an analog channel as a digital input. Pins configured as analog inputs will read '0'.

### EXAMPLE 4-3: INITIALIZING PORTC

```
BANKSEL PORTC      ;
CLRF   PORTC       ;Init PORTC
MOVLW  07h         ;Set RC<4,1:0> to
MOVWF  CMCON0      ;digital I/O
BANKSEL ANSEL      ;
CLRF   ANSEL       ;digital I/O
MOVLW  0Ch         ;Set RC<3:2> as inputs
MOVWF  TRISC       ;and set RC<5:4,1:0>
                          ;as outputs
```

### REGISTER 4-6: PORTC: PORTC REGISTER

U-0	U-0	R/W-x	R/W-x	R/W-0	R/W-0	R/W-0	R/W-0
—	—	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 7-6                      **Unimplemented:** Read as '0'  
 bit 5-0                      **RC<5:0>:** PORTC I/O Pin bit  
                                     1 = PORTC pin is > VIH  
                                     0 = PORTC pin is < VIL

### REGISTER 4-7: TRISC: PORTC TRI-STATE REGISTER

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	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 7-6                      **Unimplemented:** Read as '0'  
 bit 5-0                      **TRISC<5:0>:** PORTC Tri-State Control bits  
                                     1 = PORTC pin configured as an input (tri-stated)  
                                     0 = PORTC pin configured as an output





## 5.0 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

Figure 5-1 is a block diagram of the Timer0 module.

## 5.1 Timer0 Operation

When used as a timer, the Timer0 module can be used as either an 8-bit timer or an 8-bit counter.

### 5.1.1 8-BIT TIMER MODE

When used as a timer, the Timer0 module will increment every instruction cycle (without prescaler). Timer mode is selected by clearing the T0CS bit of the OPTION register to '0'.

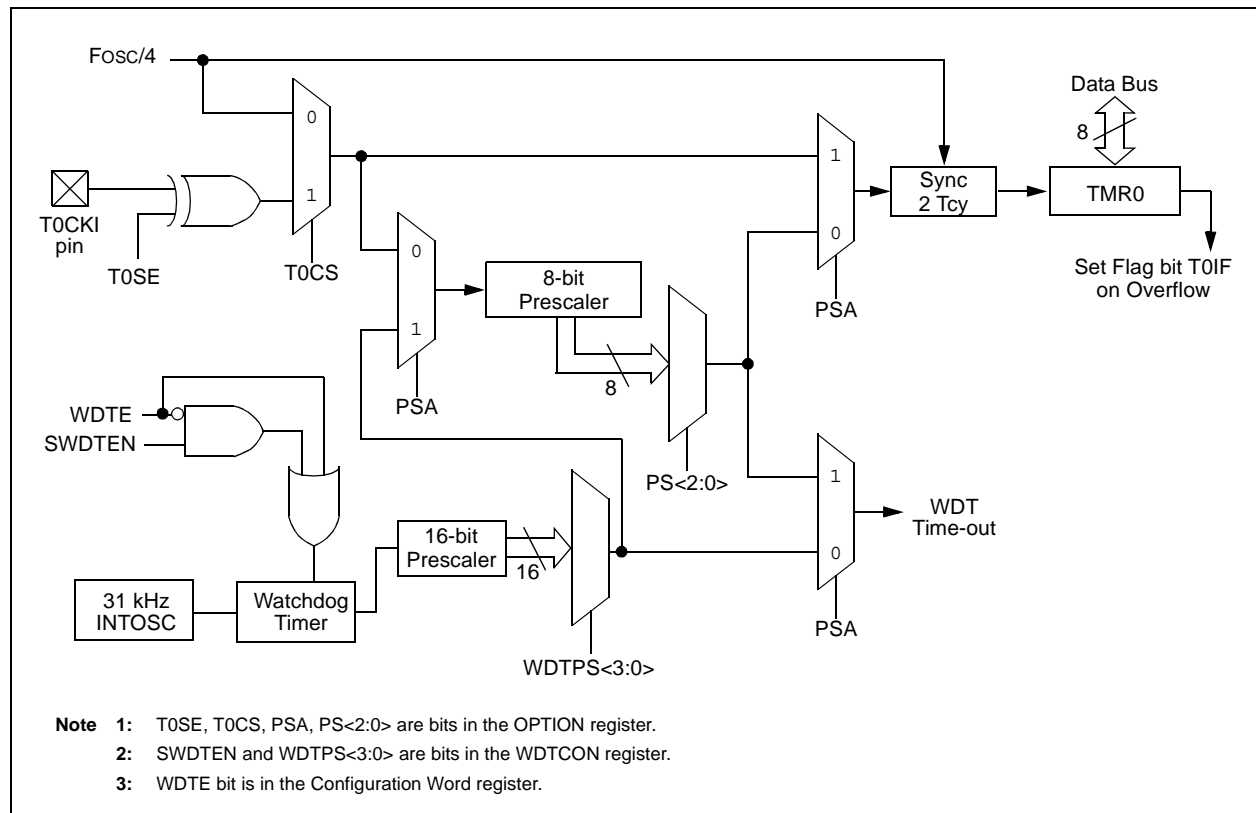
When TMR0 is written, the increment is inhibited for two instruction cycles immediately following the write.

**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.

### 5.1.2 8-BIT COUNTER MODE

When used as a counter, the Timer0 module will increment on every rising or falling edge of the T0CKI pin. The incrementing edge is determined by the T0SE bit of the OPTION register. Counter mode is selected by setting the T0CS bit of the OPTION register to '1'.

**FIGURE 5-1: BLOCK DIAGRAM OF THE TIMER0/WDT PRESCALER**



# PIC16F688

## 5.1.3 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 prescale 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.

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

### 5.1.3.1 Switching Prescaler Between Timer0 and WDT Modules

As a result of having the prescaler assigned to either Timer0 or the WDT, it is possible to generate an unintended device Reset when switching prescaler values. When changing the prescaler assignment from Timer0 to the WDT module, the instruction sequence shown in Example 5-1, must be executed.

#### EXAMPLE 5-1: CHANGING PRESCALER (TIMER0 → WDT)

```
BANKSEL  TMR0          ;
CLRWDT   TMR0          ;Clear WDT
CLRF     TMR0          ;Clear TMR0 and
                          ;prescaler

BANKSEL  OPTION_REG    ;
BSF      OPTION_REG,PSA ;Select WDT
CLRWDT   TMR0          ;
                          ;

MOVLW    b'11111000'   ;Mask prescaler
ANDWF    OPTION_REG,W  ;bits
IORLW    b'00000101'   ;Set WDT prescaler
MOVWF    OPTION_REG    ;to 1:32
```

When changing the prescaler assignment from the WDT to the Timer0 module, the following instruction sequence must be executed (see Example 5-2).

#### EXAMPLE 5-2: CHANGING PRESCALER (WDT → TIMER0)

```
CLRWDT   TMR0          ;Clear WDT and
                          ;prescaler

BANKSEL  OPTION_REG    ;
MOVLW    b'11110000'   ;Mask TMR0 select and
ANDWF    OPTION_REG,W  ;prescaler bits
IORLW    b'00000011'   ;Set prescale to 1:16
MOVWF    OPTION_REG    ;
```

## 5.1.4 TIMER0 INTERRUPT

Timer0 will generate an interrupt when the TMR0 register overflows from FFh to 00h. The TOIF 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 TOIF bit must be cleared in software. The Timer0 interrupt enable is the TOIE bit of the INTCON register.

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

## 5.1.5 USING TIMER0 WITH AN EXTERNAL CLOCK

When Timer0 is in Counter mode, the synchronization of the T0CKI input and the Timer0 register is accomplished by sampling the prescaler output on the Q2 and Q4 cycles of the internal phase clocks. Therefore, the high and low periods of the external clock source must meet the timing requirements as shown in Section 14.0 "Electrical Specifications".



## REGISTER 5-1: 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
<u>RAPU</u>	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                      **RAPU**: PORTA Pull-up Enable bit  
 1 = PORTA pull-ups are disabled  
 0 = PORTA 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  
 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

**Note 1:** A dedicated 16-bit WDT postscaler is available. See Section 11.5 "Watchdog Timer (WDT)" for more information.

**TABLE 5-1: SUMMARY OF REGISTERS ASSOCIATED WITH TIMER0**

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
TMR0	Timer0 Module Register								xxxx xxxx	uuuu uuuu
INTCON	GIE	PEIE	T0IE	INTE	RAIE	T0IF	INTF	RAIF	0000 000x	0000 000x
OPTION_REG	<u>RAPU</u>	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
TRISA	—	—	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	--11 1111	--11 1111

**Legend:** — = Unimplemented locations, read as '0', u = unchanged, x = unknown. Shaded cells are not used by the Timer0 module.

# PIC16F688

## 6.0 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
- Optional LP oscillator
- Synchronous or asynchronous operation
- Timer1 gate (count enable) via comparator or  $\overline{T1G}$  pin
- Interrupt on overflow
- Wake-up on overflow (external clock, Asynchronous mode only)

Figure 6-1 is a block diagram of the Timer1 module.

## 6.1 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.

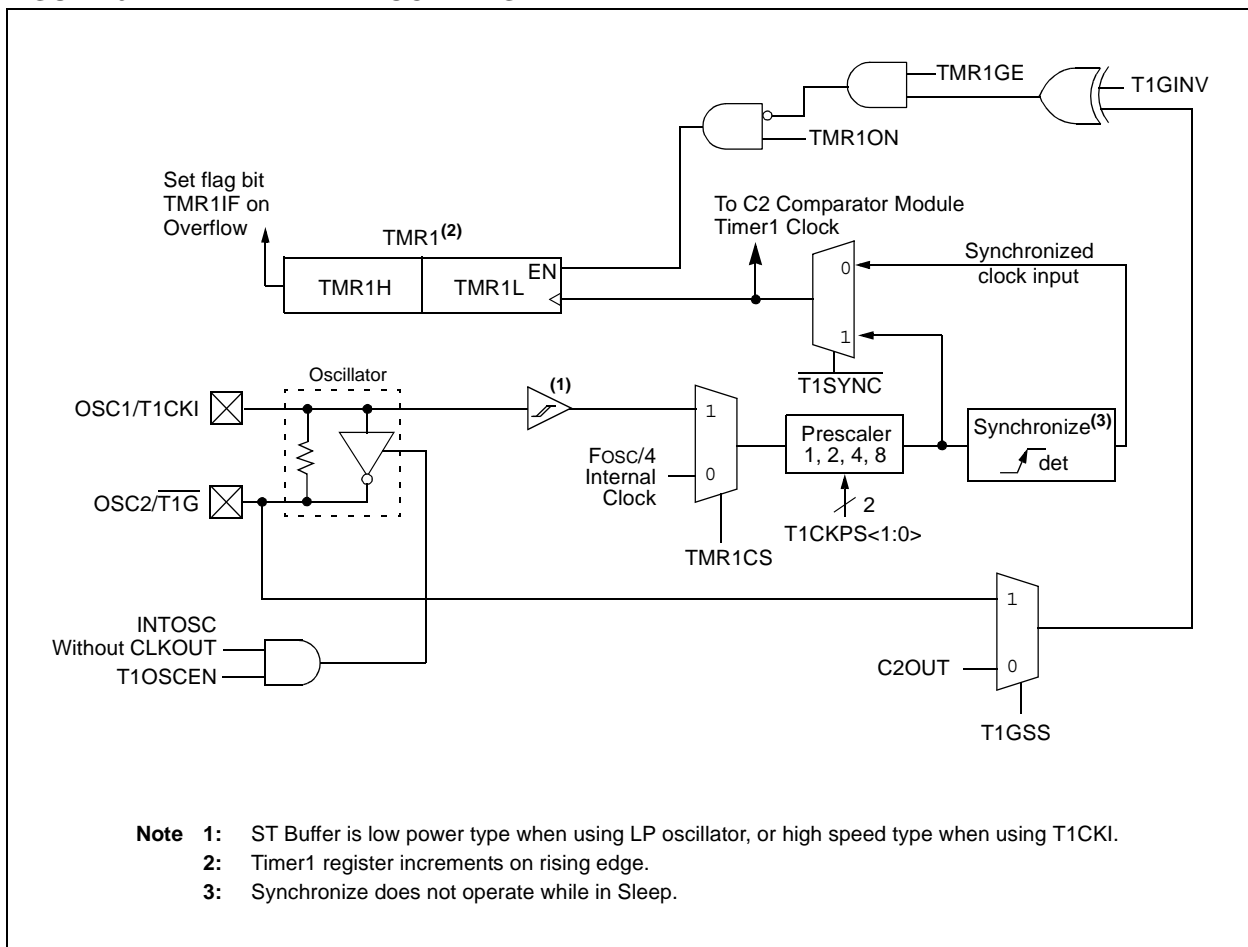
When used with an internal clock source, the module is a timer. When used with an external clock source, the module can be used as either a timer or counter.

## 6.2 Clock Source Selection

The TMR1CS bit of the T1CON register is used to select the clock source. When TMR1CS = 0, the clock source is Fosc/4. When TMR1CS = 1, the clock source is supplied externally.

Clock Source	TMR1CS	Clock Source
Fosc/4	0	Fosc/4
T1CKI pin	1	T1CKI pin

FIGURE 6-1: TIMER1 BLOCK DIAGRAM



## 6.2.1 INTERNAL CLOCK SOURCE

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

## 6.2.2 EXTERNAL CLOCK SOURCE

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

When counting, Timer1 is incremented on the rising edge of the external clock input T1CKI. In addition, the Counter mode clock can be synchronized to the microcontroller system clock or run asynchronously.

If an external clock oscillator is needed (and the microcontroller is using the INTOSC without CLKOUT), Timer1 can use the LP oscillator as a clock source.

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

## 6.3 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.

## 6.4 Timer1 Oscillator

A low-power 32.768 kHz crystal oscillator is built-in between pins OSC1 (input) and OSC2 (amplifier output). The oscillator is enabled by setting the T1OSCEN control bit of the T1CON register. The oscillator will continue to run during Sleep.

The Timer1 oscillator is shared with the system LP oscillator. Thus, Timer1 can use this mode only when the primary system clock is derived from the internal oscillator or when in LP oscillator mode. The user must provide a software time delay to ensure proper oscillator start-up.

TRISA5 and TRISA4 bits are set when the Timer1 oscillator is enabled. RA5 and RA4 bits read as '0' and TRISA5 and TRISA4 bits read as '1'.

**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.

## 6.5 Timer1 Operation in Asynchronous Counter Mode

If control bit  $\overline{T1SYNC}$  of the T1CON register is set, the external clock input is not synchronized. The timer continues to increment asynchronous to the internal phase clocks. 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 6.5.1 "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 a single spurious increment.

### 6.5.1 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.

## 6.6 Timer1 Gate

Timer1 gate source is software configurable to be the  $\overline{T1G}$  pin or the output of Comparator 2. This allows the device to directly time external events using  $\overline{T1G}$  or analog events using Comparator 2. See the CMCON1 register (Register 7-2) for selecting the Timer1 gate source. This feature can simplify the software for a Delta-Sigma A/D converter and many other applications. For more information on Delta-Sigma A/D converters, see the Microchip web site ([www.microchip.com](http://www.microchip.com)).

**Note:** TMR1GE bit of the T1CON register must be set to use either  $\overline{T1G}$  or C2OUT as the Timer1 gate source. See Register 7-2 for more information on selecting the Timer1 gate source.

Timer1 gate can be inverted using the T1GINV bit of the T1CON register, whether it originates from the  $\overline{T1G}$  pin or Comparator 2 output. This configures Timer1 to measure either the active-high or active-low time between events.

# PIC16F688

## 6.7 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:

- Timer1 interrupt enable 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.

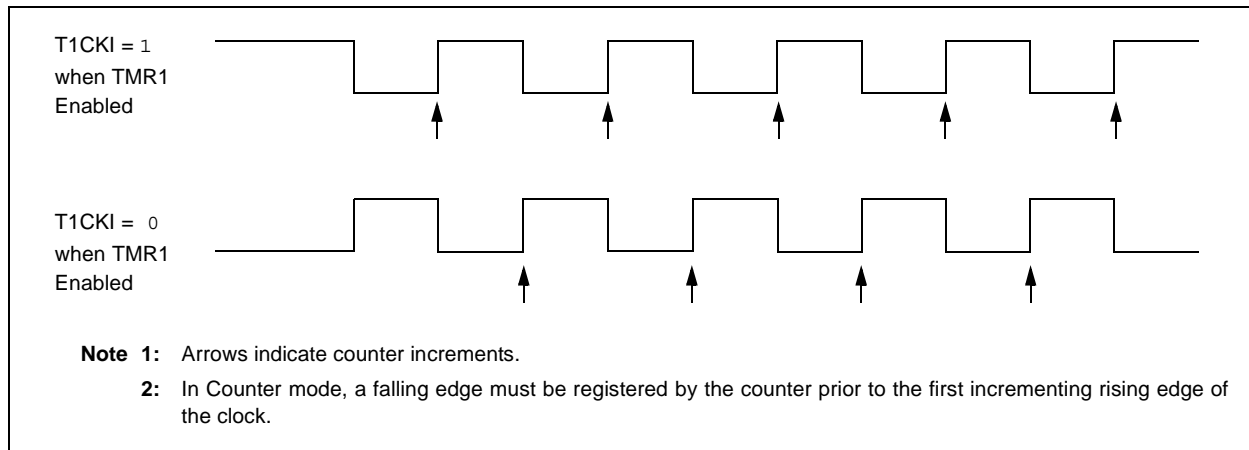
## 6.8 Timer1 Operation During Sleep

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

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

**FIGURE 6-2: TIMER1 INCREMENTING EDGE**



## 6.9 Timer1 Control Register

The Timer1 Control register (T1CON), shown in Register 6-1, is used to control Timer1 and select the various features of the Timer1 module.

**REGISTER 6-1: T1CON: TIMER 1 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-0
T1GINV <sup>(1)</sup>	TMR1GE <sup>(2)</sup>	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{T1SYNC}$	TMR1CS	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      **T1GINV:** Timer1 Gate Invert bit<sup>(1)</sup>  
             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 6      **TMR1GE:** Timer1 Gate Enable bit<sup>(2)</sup>  
             If TMR1ON = 0:  
             This bit is ignored  
             If TMR1ON = 1:  
             1 = Timer1 is on if Timer1 gate is active  
             0 = Timer1 is on
- 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  
             If INTOSC without CLKOUT oscillator is active:  
             1 = LP oscillator is enabled for Timer1 clock  
             0 = LP oscillator is off  
             Else:  
             This bit is ignored. LP oscillator is disabled.
- bit 2       **$\overline{T1SYNC}$ :** Timer1 External Clock Input Synchronization Control bit  
             TMR1CS = 1:  
             1 = Do not synchronize external clock input  
             0 = Synchronize external clock input  
             TMR1CS = 0:  
             This bit is ignored. Timer1 uses the internal clock
- bit 1      **TMR1CS:** Timer1 Clock Source Select bit  
             1 = External clock from T1CKI pin (on the rising edge)  
             0 = Internal clock (FOSC/4)
- bit 0      **TMR1ON:** Timer1 On bit  
             1 = Enables Timer1  
             0 = Stops Timer1

- Note 1:** T1GINV bit inverts the Timer1 gate logic, regardless of source.
- 2:** TMR1GE bit must be set to use either  $\overline{T1G}$  pin or C2OUT, as selected by the T1GSS bit of the CM2CON1 register, as a Timer1 gate source.

# PIC16F688

**TABLE 6-1: SUMMARY OF REGISTERS ASSOCIATED WITH TIMER1**

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
CMCON1	—	—	—	—	—	—	T1GSS	C2SYNC	---- --10	00-- --10
INTCON	GIE	PEIE	T0IE	INTE	RAIE	T0IF	INTF	RAIF	0000 000x	0000 000x
PIE1	EEIE	ADIE	RCIE	C2IE	C1IE	OSFIE	TXIE	TMR1IE	0000 0000	0000 0000
PIR1	EEIF	ADIF	RCIF	C2IF	C1IF	OSFIF	TXIF	TMR1IF	0000 0000	0000 0000
TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	uuuu uuuu
TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	uuuu uuuu
T1CON	T1GINV	TMR1GE	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	0000 0000	uuuu uuuu

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

## 7.0 COMPARATOR MODULE

Comparators are used to interface analog circuits to a digital circuit by comparing two analog voltages and providing a digital indication of their relative magnitudes. The comparators are very useful mixed signal building blocks because they provide analog functionality independent of the program execution. The Analog Comparator module includes the following features:

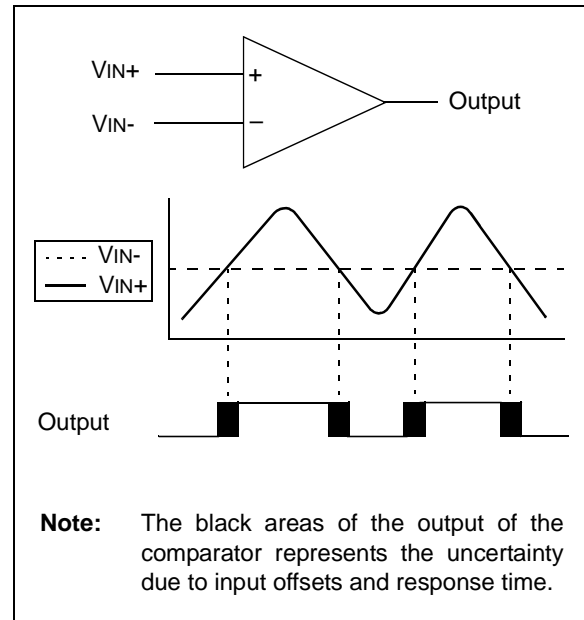
- Dual comparators
- Multiple comparator configurations
- Comparator outputs are available internally/externally
- Programmable output polarity
- Interrupt-on-change
- Wake-up from Sleep
- Timer1 gate (count enable)
- Output synchronization to Timer1 clock input
- Programmable voltage reference

**Note:** Only Comparator C2 can be linked to Timer1.

## 7.1 Comparator Overview

A comparator is shown in Figure 7-1 along with the relationship between the analog input levels and the digital output. When the analog voltage at  $V_{IN+}$  is less than the analog voltage at  $V_{IN-}$ , the output of the comparator is a digital low level. When the analog voltage at  $V_{IN+}$  is greater than the analog voltage at  $V_{IN-}$ , the output of the comparator is a digital high level.

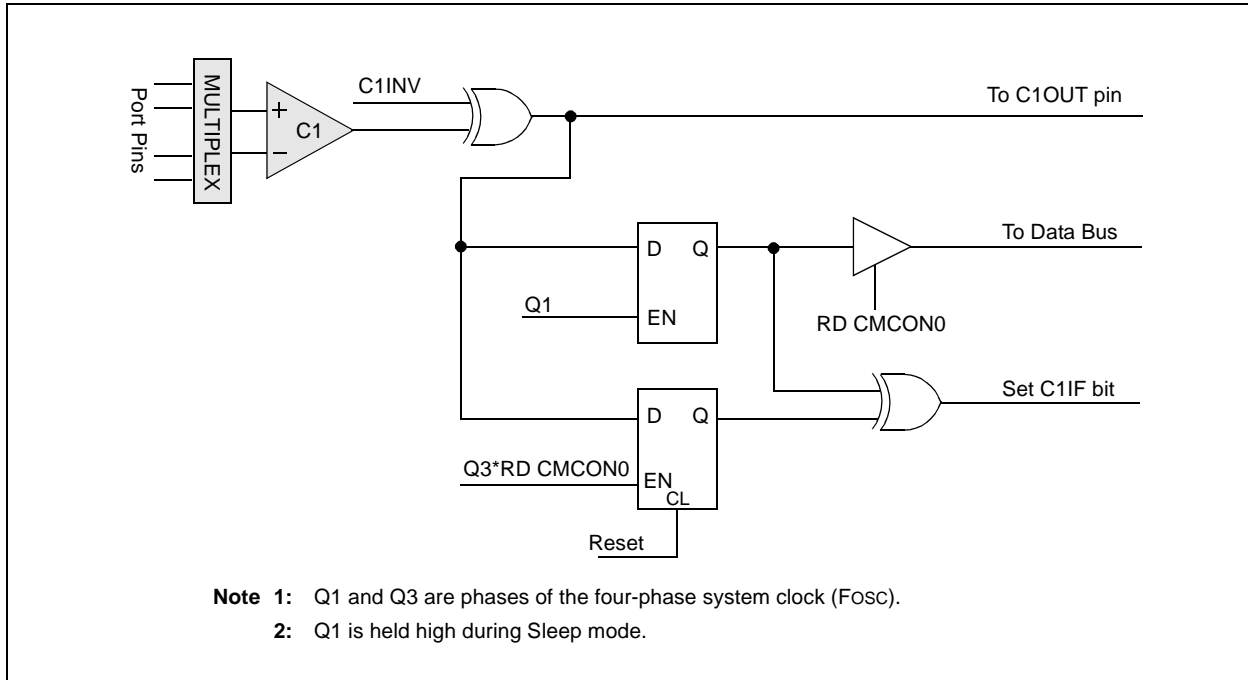
**FIGURE 7-1: SINGLE COMPARATOR**



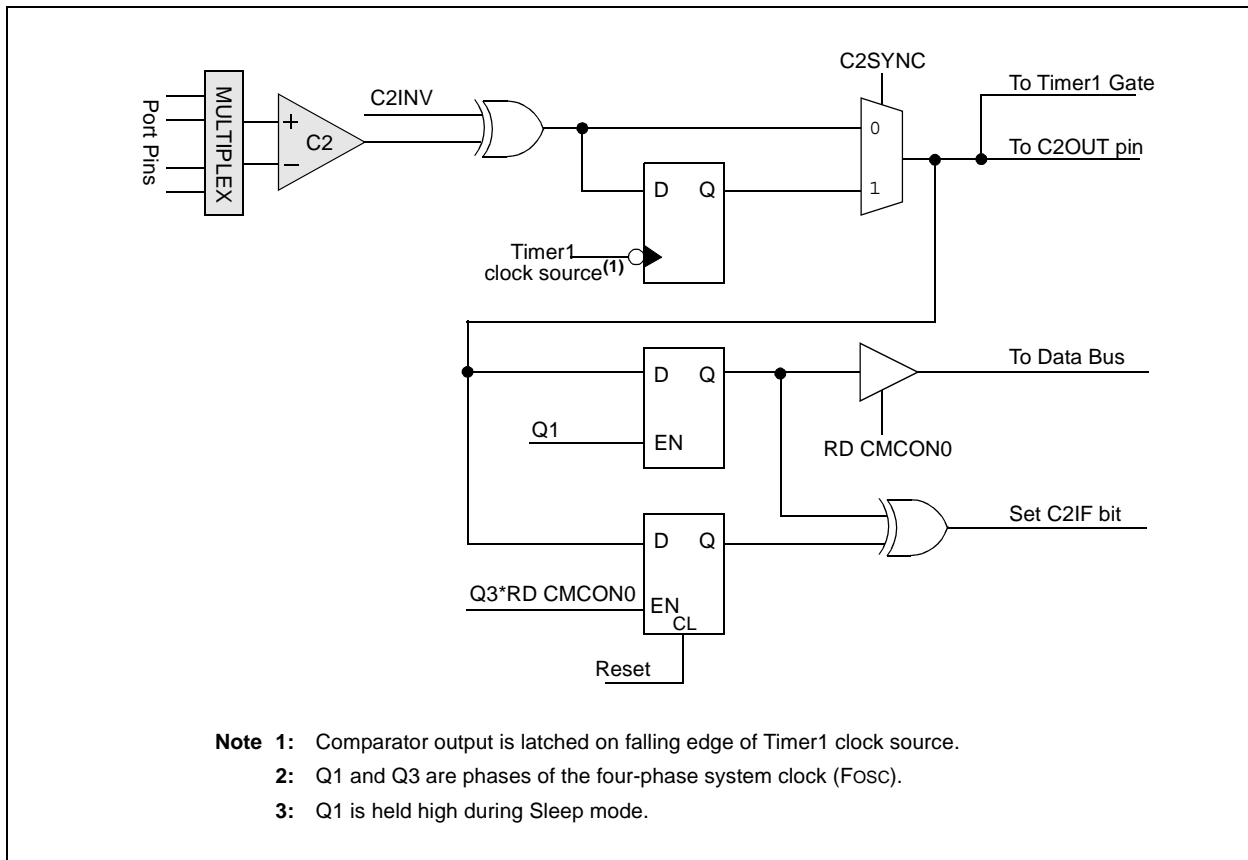
This device contains two comparators as shown in Figure 7-2 and Figure 7-3. The comparators are not independently configurable.

# PIC16F688

**FIGURE 7-2: COMPARATOR C1 OUTPUT BLOCK DIAGRAM**



**FIGURE 7-3: COMPARATOR C2 OUTPUT BLOCK DIAGRAM**





## 7.1.1 ANALOG INPUT CONNECTION CONSIDERATIONS

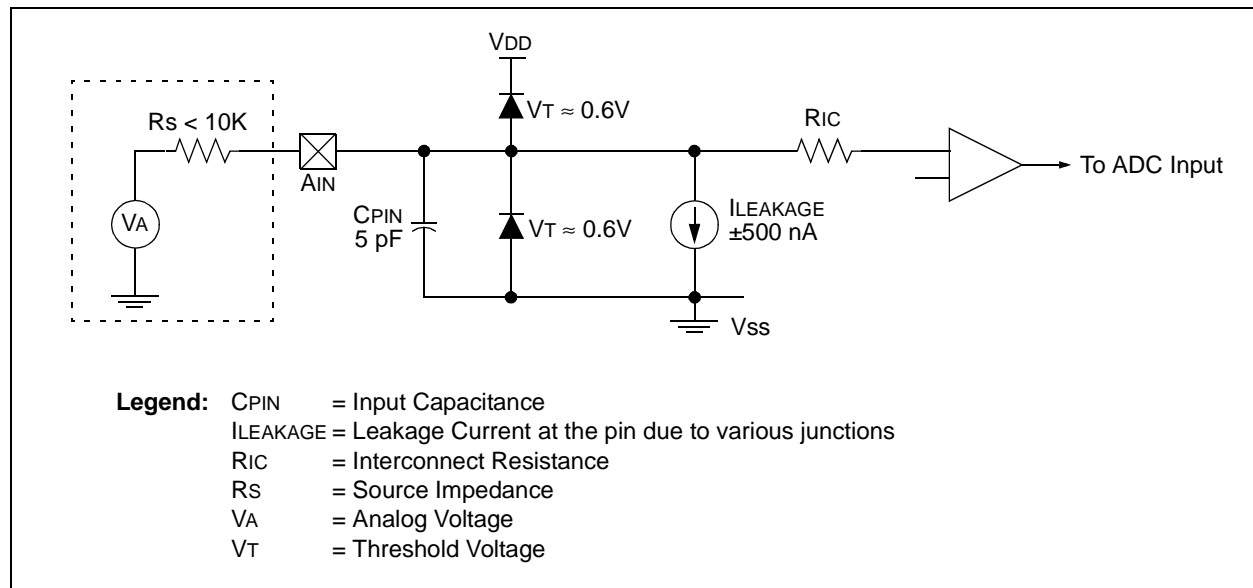
A simplified circuit for an analog input is shown in Figure 7-4. Since the analog input pins share their connection with a digital input, they have reverse biased ESD protection diodes to VDD and VSS. The analog input, therefore, must be between VSS and VDD. If the input voltage deviates from this range by more than 0.6V in either direction, one of the diodes is forward biased and a latch-up may occur.

A maximum source impedance of 10 kΩ is recommended for the analog sources. Also, any external component connected to an analog input pin, such as a capacitor or a Zener diode, should have very little leakage current to minimize inaccuracies introduced.

**Note 1:** When reading a PORT register, all pins configured as analog inputs will read as a '0'. Pins configured as digital inputs will convert as an analog input, according to the input specification.

**2:** Analog levels on any pin defined as a digital input, may cause the input buffer to consume more current than is specified.

**FIGURE 7-4: ANALOG INPUT MODEL**



# PIC16F688

---

## 7.2 Comparator Configuration

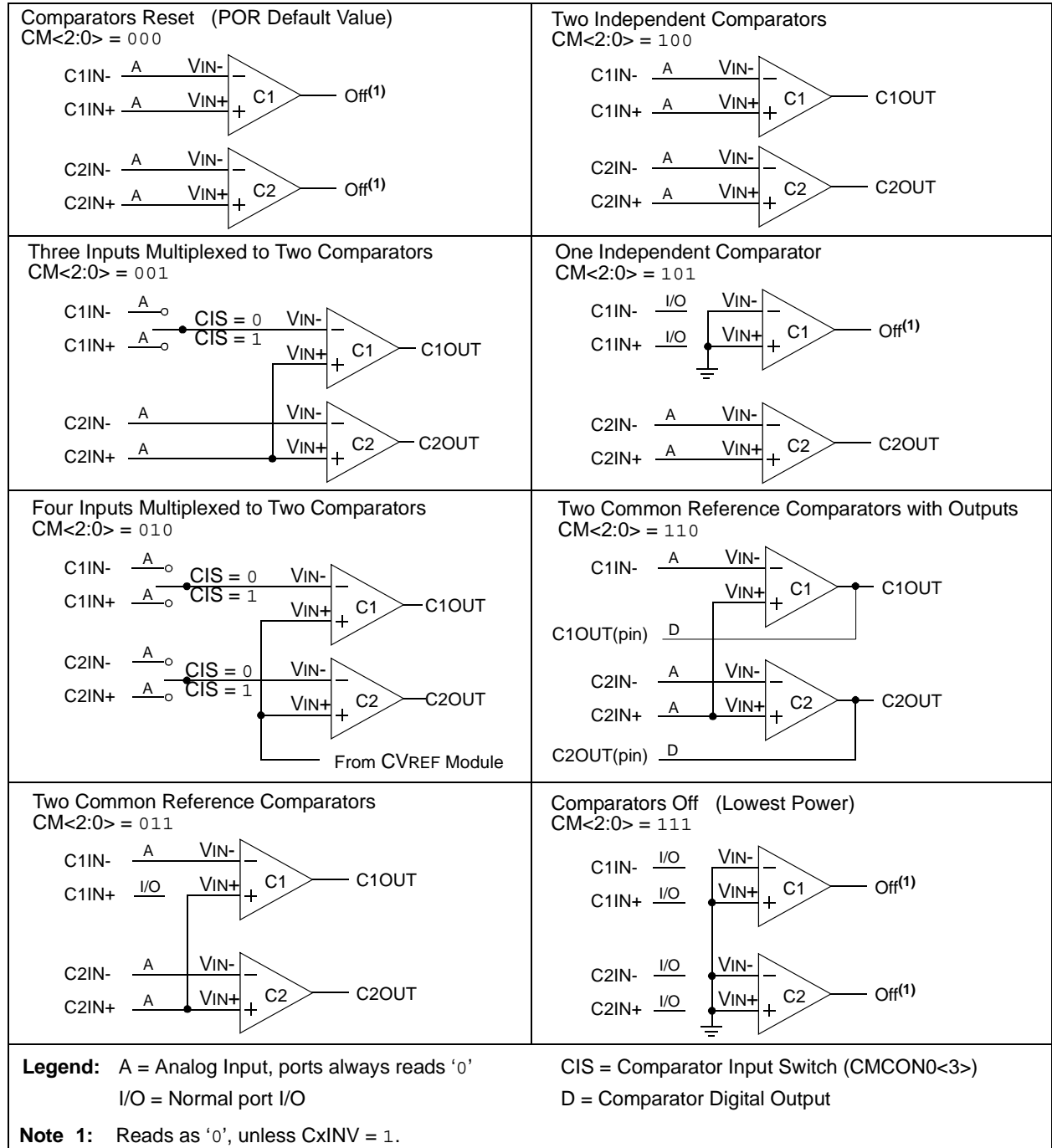
There are eight modes of operation for the comparator. The CM<2:0> bits of the CMCON0 register are used to select these modes as shown in Figure 7-5. I/O lines change as a function of the mode and are designated as follows:

- Analog function (A): digital input buffer is disabled
- Digital function (D): comparator digital output, overrides port function
- Normal port function (I/O): independent of comparator

The port pins denoted as “A” will read as a ‘0’ regardless of the state of the I/O pin or the I/O control TRIS bit. Pins used as analog inputs should also have the corresponding TRIS bit set to ‘1’ to disable the digital output driver. Pins denoted as “D” should have the corresponding TRIS bit set to ‘0’ to enable the digital output driver.

<p><b>Note:</b> Comparator interrupts should be disabled during a Comparator mode change to prevent unintended interrupts.</p>
--

**FIGURE 7-5: COMPARATOR I/O OPERATING MODES**



# PIC16F688

---

## 7.3 Comparator Control

The CMCON0 register (Register 7-1) provides access to the following comparator features:

- Mode selection
- Output state
- Output polarity
- Input switch

### 7.3.1 COMPARATOR OUTPUT STATE

Each comparator state can always be read internally via the associated CxOUT bit of the CMCON0 register. The comparator outputs are directed to the CxOUT pins when  $CM\langle 2:0 \rangle = 110$ . When this mode is selected, the TRIS bits for the associated CxOUT pins must be cleared to enable the output drivers.

### 7.3.2 COMPARATOR OUTPUT POLARITY

Inverting the output of a comparator is functionally equivalent to swapping the comparator inputs. The polarity of a comparator output can be inverted by setting the CxINV bits of the CMCON0 register. Clearing CxINV results in a non-inverted output. A complete table showing the output state versus input conditions and the polarity bit is shown in Table 7-1.

**TABLE 7-1: OUTPUT STATE VS. INPUT CONDITIONS**

Input Conditions	CxINV	CxOUT
$V_{IN-} > V_{IN+}$	0	0
$V_{IN-} < V_{IN+}$	0	1
$V_{IN-} > V_{IN+}$	1	1
$V_{IN-} < V_{IN+}$	1	0

**Note:** CxOUT refers to both the register bit and output pin.

### 7.3.3 COMPARATOR INPUT SWITCH

The inverting input of the comparators may be switched between two analog pins in the following modes:

- CM<2:0> = 001 (Comparator C1 only)
- CM<2:0> = 010 (Comparators C1 and C2)

In the above modes, both pins remain in analog mode regardless of which pin is selected as the input. The CIS bit of the CMCON0 register controls the comparator input switch.

### 7.4 Comparator Response Time

The comparator output is indeterminate for a period of time after the change of an input source or the selection of a new reference voltage. This period is referred to as the response time. The response time of the comparator differs from the settling time of the voltage reference. Therefore, both of these times must be considered when determining the total response time to a comparator input change. See the Comparator and Voltage Reference specifications in **Section 14.0 “Electrical Specifications”** for more details.

### 7.5 Comparator Interrupt Operation

The comparator interrupt flag is set whenever there is a change in the output value of the comparator. Changes are recognized by means of a mismatch circuit which consists of two latches and an exclusive-or gate (see Figure 7-2 and Figure 7-3). One latch is updated with the comparator output level when the CMCON0 register is read. This latch retains the value until the next read of the CMCON0 register or the occurrence of a Reset. The other latch of the mismatch circuit is updated on every Q1 system clock. A mismatch condition will occur when a comparator output change is clocked through the second latch on the Q1 clock cycle. The mismatch condition will persist, holding the CxIF bit of the PIR1 register true, until either the CMCON0 register is read or the comparator output returns to the previous state.

**Note:** A write operation to the CMCON0 register will also clear the mismatch condition because all writes include a read operation at the beginning of the write cycle.

Software will need to maintain information about the status of the comparator output to determine the actual change that has occurred.

The CxIF bit of the PIR1 register is the comparator interrupt flag. This bit must be reset in software by clearing it to '0'. Since it is also possible to write a '1' to this register, a simulated interrupt may be initiated.

The CxIE bit of the PIE1 register and the PEIE and GIE bits of the INTCON register must all be set to enable comparator interrupts. If any of these bits are cleared, the interrupt is not enabled, although the CxIF bit of the PIR1 register will still be set if an interrupt condition occurs.

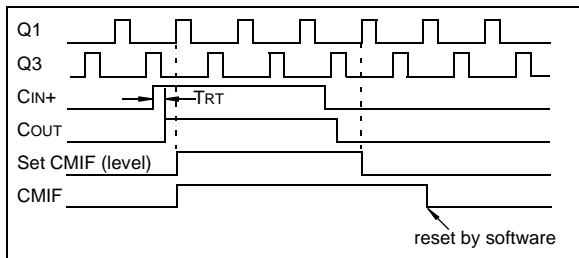
The user, in the Interrupt Service Routine, can clear the interrupt in the following manner:

- a) Any read or write of CMCON0. This will end the mismatch condition. See Figures 7-6 and 7-7
- b) Clear the CxIF interrupt flag.

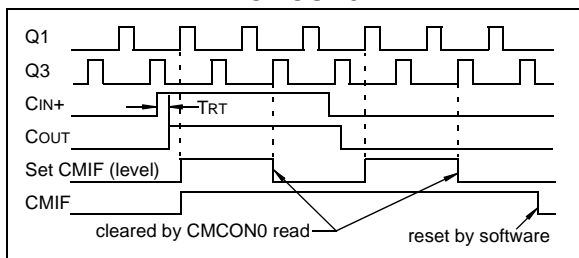
A persistent mismatch condition will preclude clearing the CxIF interrupt flag. Reading CMCON0 will end the mismatch condition and allow the CxIF bit to be cleared.

# PIC16F688

**FIGURE 7-6: COMPARATOR INTERRUPT TIMING W/O CMCON0 READ**



**FIGURE 7-7: COMPARATOR INTERRUPT TIMING WITH CMCON0 READ**



**Note 1:** If a change in the CM1CON0 register (CxOUT) occurs when a read operation is being executed (start of the Q2 cycle), then the CxIF Interrupt Flag bit of the PIR1 register may not get set.

**2:** When either comparator is first enabled, bias circuitry in the Comparator module may cause an invalid output from the comparator until the bias circuitry is stable. Allow about 1  $\mu$ s for bias settling then clear the mismatch condition and interrupt flags before enabling comparator interrupts.

## 7.6 Operation During Sleep

The comparator, if enabled before entering Sleep mode, remains active during Sleep. The additional current consumed by the comparator is shown separately in **Section 14.0 “Electrical Specifications”**. If the comparator is not used to wake the device, power consumption can be minimized while in Sleep mode by turning off the comparator. The comparator is turned off by selecting mode  $CM\langle 2:0 \rangle = 000$  or  $CM\langle 2:0 \rangle = 111$  of the CMCON0 register.

A change to the comparator output can wake-up the device from Sleep. To enable the comparator to wake the device from Sleep, the CxIE bit of the PIE1 register and the PEIE bit of the INTCON register must be set. The instruction following the Sleep instruction always executes following a wake from Sleep. If the GIE bit of the INTCON register is also set, the device will then execute the Interrupt Service Routine.

## 7.7 Effects of a Reset

A device Reset forces the CMCON0 and CMCON1 registers to their Reset states. This forces the Comparator module to be in the Comparator Reset mode ( $CM\langle 2:0 \rangle = 000$ ). Thus, all comparator inputs are analog inputs with the comparator disabled to consume the smallest current possible.

## REGISTER 7-1: CMCON0: COMPARATOR CONFIGURATION REGISTER

R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0
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      **C2OUT:** Comparator 2 Output bit  
When C2INV = 0:  
 1 = C2 VIN+ > C2 VIN-  
 0 = C2 VIN+ < C2 VIN-  
When C2INV = 1:  
 1 = C2 VIN+ < C2 VIN-  
 0 = C2 VIN+ > C2 VIN-
- bit 6      **C1OUT:** Comparator 1 Output bit  
When C1INV = 0:  
 1 = C1 VIN+ > C1 VIN-  
 0 = C1 VIN+ < C1 VIN-  
When C1INV = 1:  
 1 = C1 VIN+ < C1 VIN-  
 0 = C1 VIN+ > C1 VIN-
- bit 5      **C2INV:** Comparator 2 Output Inversion bit  
 1 = C2 output inverted  
 0 = C2 output not inverted
- bit 4      **C1INV:** Comparator 1 Output Inversion bit  
 1 = C1 Output inverted  
 0 = C1 Output not inverted
- bit 3      **CIS:** Comparator Input Switch bit  
When CM<2:0> = 010:  
 1 = C1IN+ connects to C1 VIN-  
       C2IN+ connects to C2 VIN-  
 0 = C1IN- connects to C1 VIN-  
       C2IN- connects to C2 VIN-  
When CM<2:0> = 001:  
 1 = C1IN+ connects to C1 VIN-  
 0 = C1IN- connects to C1 VIN-
- bit 2-0    **CM<2:0>:** Comparator Mode bits (See Figure 7-5)  
 000 = Comparators off. CxIN pins are configured as analog  
 001 = Three inputs multiplexed to two comparators  
 010 = Four inputs multiplexed to two comparators  
 011 = Two common reference comparators  
 100 = Two independent comparators  
 101 = One independent comparator  
 110 = Two common reference comparators with outputs  
 111 = Comparators off. CxIN pins are configured as digital I/O

# PIC16F688

## 7.8 Comparator C2 Gating Timer1

This feature can be used to time the duration or interval of analog events. Clearing the T1GSS bit of the CMCON1 register will enable Timer1 to increment based on the output of Comparator C2. This requires that Timer1 is on and gating is enabled. See **Section 6.0 “Timer1 Module with Gate Control”** for details.

It is recommended to synchronize Comparator C2 with Timer1 by setting the C2SYNC bit when the comparator is used as the Timer1 gate source. This ensures Timer1 does not miss an increment if the comparator changes during an increment.

## 7.9 Synchronizing Comparator C2 Output to Timer1

The output of Comparator C2 can be synchronized with Timer1 by setting the C2SYNC bit of the CMCON1 register. When enabled, the comparator output is latched on the falling edge of the Timer1 clock source. If a prescaler is used with Timer1, the comparator output is latched after the prescaling function. To prevent a race condition, the comparator output is latched on the falling edge of the Timer1 clock source and Timer1 increments on the rising edge of its clock source. Reference the comparator block diagrams (Figure 7-2 and Figure 7-3) and the Timer1 Block Diagram (Figure 6-1) for more information.

### REGISTER 7-2: CMCON1: COMPARATOR CONFIGURATION REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0
—	—	—	—	—	—	T1GSS	C2SYNC
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 **T1GSS:** Timer1 Gate Source Select bit<sup>(1)</sup>

1 = Timer1 gate source is  $\overline{T1G}$  pin (pin should be configured as digital input)

0 = Timer1 gate source is Comparator C2 output

bit 0 **C2SYNC:** Comparator C2 Output Synchronization bit<sup>(2)</sup>

1 = Output is synchronized with falling edge of Timer1 clock

0 = Output is asynchronous

**Note 1:** Refer to **Section 6.6 “Timer1 Gate”**.

**2:** Refer to Figure 7-3.



## 7.10 Comparator Voltage Reference

The Comparator Voltage Reference module provides an internally generated voltage reference for the comparators. The following features are available:

- Independent from Comparator operation
- Two 16-level voltage ranges
- Output clamped to V<sub>SS</sub>
- Ratiometric with V<sub>DD</sub>

The VRCON register (Figure 7-3) controls the Voltage Reference module shown in Figure 7-8.

### 7.10.1 INDEPENDENT OPERATION

The comparator voltage reference is independent of the comparator configuration. Setting the VREN bit of the VRCON register will enable the voltage reference.

### 7.10.2 OUTPUT VOLTAGE SELECTION

The CVREF voltage reference has 2 ranges with 16 voltage levels in each range. Range selection is controlled by the VRR bit of the VRCON register. The 16 levels are set with the VR<3:0> bits of the VRCON register.

The CVREF output voltage is determined by the following equations:

### EQUATION 7-1: CVREF OUTPUT VOLTAGE

$$\begin{aligned}
 &VRR = 1 \text{ (low range):} \\
 &CVREF = (VR<3:0>/24) \times VDD \\
 &VRR = 0 \text{ (high range):} \\
 &CVREF = (VDD/4) + (VR<3:0> \times VDD/32)
 \end{aligned}$$

The full range of V<sub>SS</sub> to V<sub>DD</sub> cannot be realized due to the construction of the module. See Figure 7-8.

### 7.10.3 OUTPUT CLAMPED TO V<sub>SS</sub>

The CVREF output voltage can be set to V<sub>SS</sub> with no power consumption by configuring VRCON as follows:

- VREN = 0
- VRR = 1
- VR<3:0> = 0000

This allows the comparator to detect a zero-crossing while not consuming additional CVREF module current.

### 7.10.4 OUTPUT RATIOMETRIC TO V<sub>DD</sub>

The comparator voltage reference is V<sub>DD</sub> derived and therefore, the CVREF output changes with fluctuations in V<sub>DD</sub>. The tested absolute accuracy of the Comparator Voltage Reference can be found in **Section 14.0 “Electrical Specifications”**.

### REGISTER 7-3: VRCON: VOLTAGE REFERENCE CONTROL REGISTER

R/W-0	U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
VREN	—	VRR	—	VR3	VR2	VR1	VR0
bit 7							bit 0

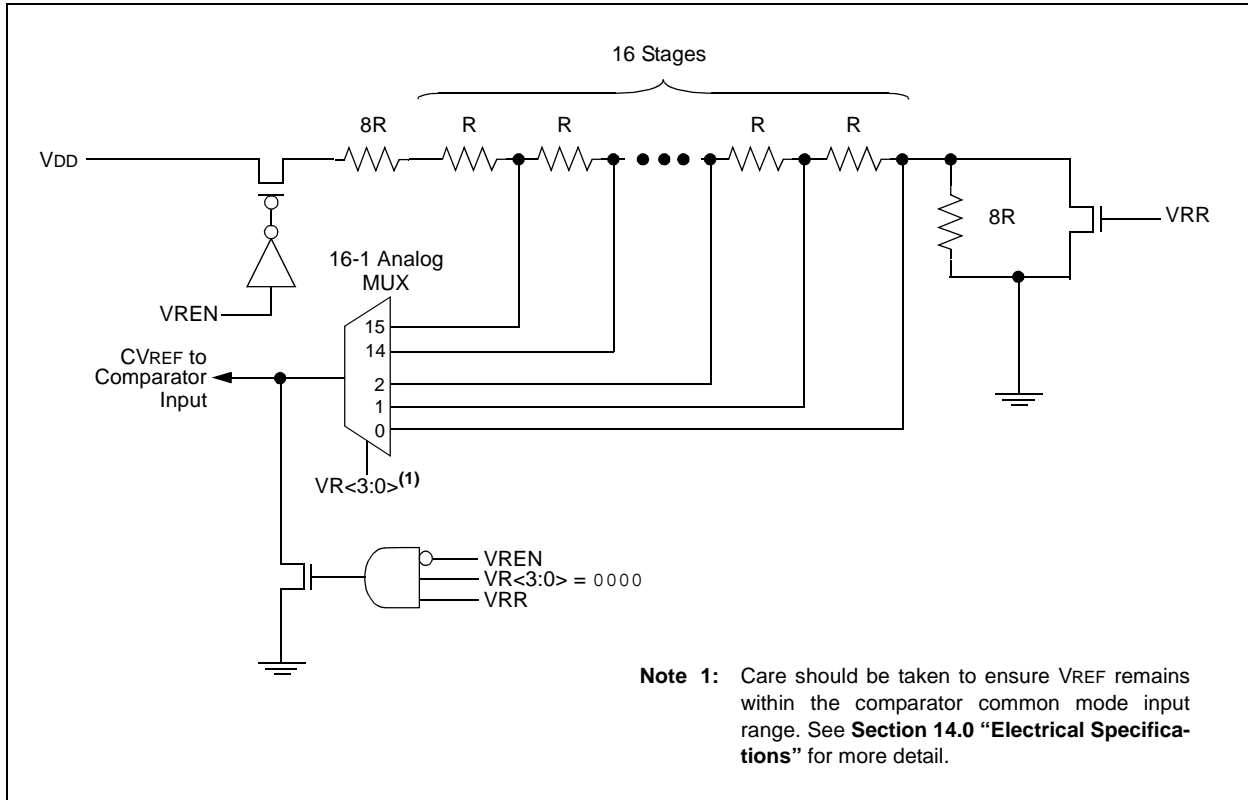
#### Legend:

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            **VREN:** CVREF Enable bit  
 1 = CVREF circuit powered on  
 0 = CVREF circuit powered down, no IDD drain and CVREF = V<sub>SS</sub>.
- bit 6            **Unimplemented:** Read as '0'
- bit 5            **VRR:** CVREF Range Selection bit  
 1 = Low range  
 0 = High range
- bit 4            **Unimplemented:** Read as '0'
- bit 3-0        **VR<3:0>:** CVREF Value Selection bits (0 ≤ VR<3:0> ≤ 15)  
 When VRR = 1: CVREF = (VR<3:0>/24) \* VDD  
 When VRR = 0: CVREF = VDD/4 + (VR<3:0>/32) \* VDD

# PIC16F688

**FIGURE 7-8: COMPARATOR VOLTAGE REFERENCE BLOCK DIAGRAM**



**TABLE 7-2: SUMMARY OF REGISTERS ASSOCIATED WITH THE COMPARATOR AND VOLTAGE REFERENCE MODULES**

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
ANSEL	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0	1111 1111	1111 1111
CMCON0	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	0000 0000	0000 0000
CMCON1	—	—	—	—	—	—	T1GSS	C2SYNC	---- --10	---- --10
INTCON	GIE	PEIE	T0IE	INTE	RAIE	T0IF	INTF	RAIF	0000 000x	0000 000x
PIE1	EEIE	ADIE	RCIE	C2IE	C1IE	OSFIE	TXIE	TMR1IE	0000 0000	0000 0000
PIR1	EEIF	ADIF	RCIF	C2IF	C1IF	OSFIF	TXIF	TMR1IF	0000 0000	0000 0000
PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	--x0 x000	--x0 x000
PORTC	—	—	RC5	RC4	RC3	RC2	RC1	RC0	--xx 0000	--xx 0000
TRISA	—	—	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	--11 1111	--11 1111
TRISC	—	—	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	--11 1111	--11 1111
VRCON	VREN	—	VRR	—	VR3	VR2	VR1	VR0	0-0- 0000	0-0- 0000

**Legend:** x = unknown, u = unchanged, — = unimplemented, read as '0'. Shaded cells are not used for comparator.

## 8.0 ANALOG-TO-DIGITAL CONVERTER (ADC) MODULE

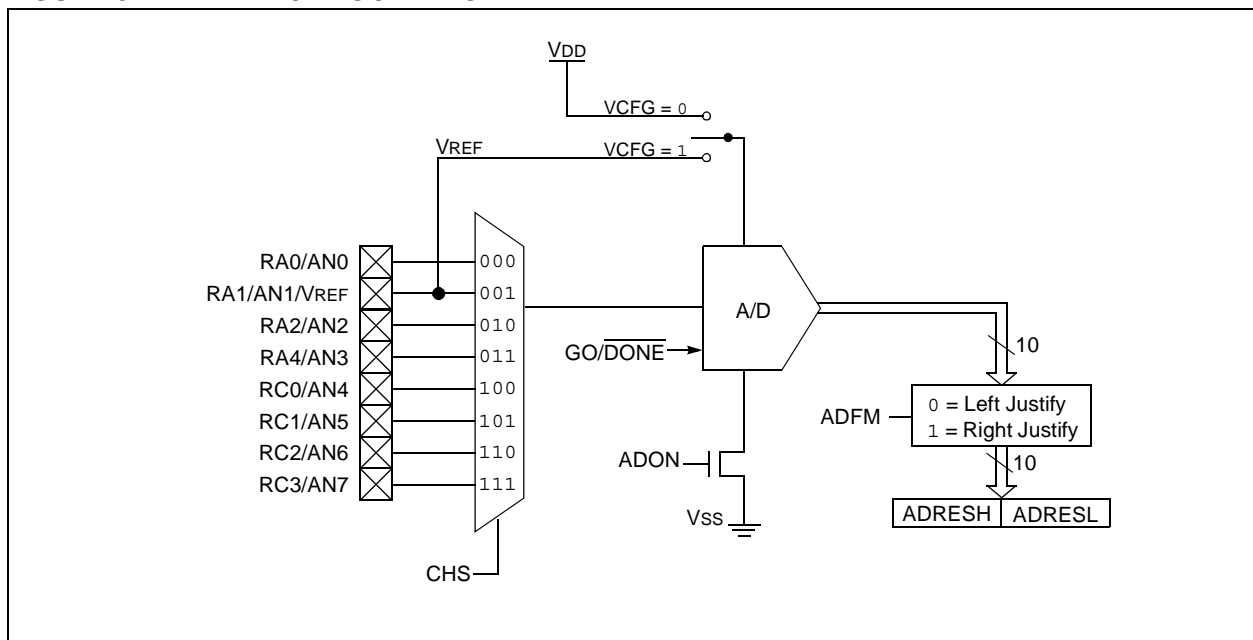
The Analog-to-Digital Converter (ADC) allows conversion of an analog input signal to a 10-bit binary representation of that signal. This device uses analog inputs, which are multiplexed into a single sample and hold circuit. The output of the sample and hold is connected to the input of the converter. The converter generates a 10-bit binary result via successive approximation and stores the conversion result into the ADC result registers (ADRESL and ADRESH).

The ADC voltage reference is software selectable to either  $V_{DD}$  or a voltage applied to the external reference pins.

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

Figure 8-1 shows the block diagram of the ADC.

**FIGURE 8-1: ADC BLOCK DIAGRAM**



# PIC16F688

---

## 8.1 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

### 8.1.1 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. See the corresponding Port section 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.

### 8.1.2 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 8.2 “ADC Operation”** for more information.

### 8.1.3 ADC VOLTAGE REFERENCE

The VCFG bit of the ADCON0 register provides control of the positive voltage reference. The positive voltage reference can be either  $V_{DD}$  or an external voltage source. The negative voltage reference is always connected to the ground reference.

### 8.1.4 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:

- $F_{osc}/2$
- $F_{osc}/4$
- $F_{osc}/8$
- $F_{osc}/16$
- $F_{osc}/32$
- $F_{osc}/64$
- FRC (dedicated internal oscillator)

The time to complete one bit conversion is defined as TAD. One full 10-bit conversion requires 11 TAD periods as shown in Figure 8-3.

For correct conversion, the appropriate TAD specification must be met. See A/D conversion requirements in **Section 14.0 “Electrical Specifications”** for more information. Table 8-1 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 8-1: ADC CLOCK PERIOD (TAD) Vs. DEVICE OPERATING FREQUENCIES (VDD ≥ 3.0V)**

ADC Clock Period (TAD)		Device Frequency (Fosc)			
ADC Clock Source	ADCS<2:0>	20 MHz	8 MHz	4 MHz	1 MHz
Fosc/2	000	100 ns <sup>(2)</sup>	250 ns <sup>(2)</sup>	500 ns <sup>(2)</sup>	2.0 μs
Fosc/4	100	200 ns <sup>(2)</sup>	500 ns <sup>(2)</sup>	1.0 μs <sup>(2)</sup>	4.0 μs
Fosc/8	001	400 ns <sup>(2)</sup>	1.0 μs <sup>(2)</sup>	2.0 μs	8.0 μs <sup>(3)</sup>
Fosc/16	101	800 ns <sup>(2)</sup>	2.0 μs	4.0 μs	16.0 μs <sup>(3)</sup>
Fosc/32	010	1.6 μs	4.0 μs	8.0 μs <sup>(3)</sup>	32.0 μs <sup>(3)</sup>
Fosc/64	110	3.2 μs	8.0 μs <sup>(3)</sup>	16.0 μs <sup>(3)</sup>	64.0 μs <sup>(3)</sup>
FRC	x11	2-6 μs <sup>(1,4)</sup>	2-6 μs <sup>(1,4)</sup>	2-6 μs <sup>(1,4)</sup>	2-6 μs <sup>(1,4)</sup>

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

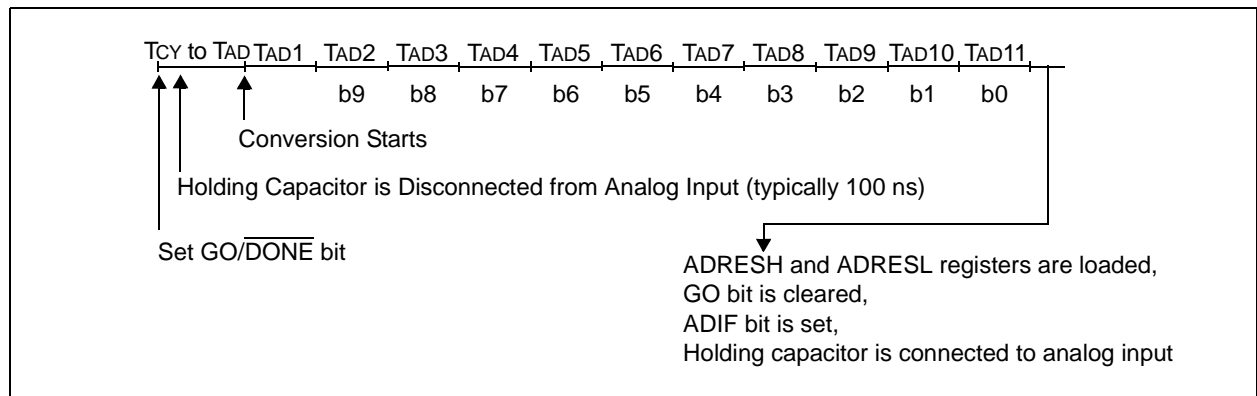
**Note 1:** The FRC source has a typical TAD time of 4 μs for VDD > 3.0V.

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

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

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

**FIGURE 8-2: ANALOG-TO-DIGITAL CONVERSION TAD CYCLES**



# PIC16F688

## 8.1.5 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:** The ADIF bit is set at the completion of every conversion, regardless of whether or not the ADC interrupt is enabled.

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 global interrupt must be disabled. If the global interrupt is enabled, execution will switch to the Interrupt Service Routine.

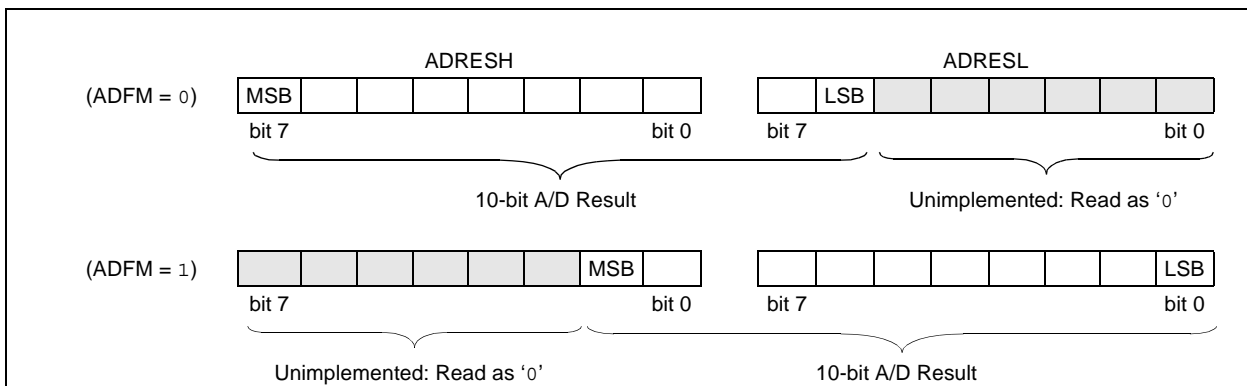
Please see **Section 8.1.5 “Interrupts”** for more information.

## 8.1.6 RESULT FORMATTING

The 10-bit A/D Conversion result can be supplied in two formats, left justified or right justified. The ADFM bit of the ADCON0 register controls the output format.

Figure 8-4 shows the two output formats.

**FIGURE 8-3: 10-BIT A/D CONVERSION RESULT FORMAT**



## 8.2 ADC Operation

### 8.2.1 STARTING A CONVERSION

To enable the ADC module, the ADON bit of the ADCON0 register must be set to a '1'. Setting the GO/DONE bit of the ADCON0 register to a '1' will start the Analog-to-Digital Conversion.

**Note:** The GO/DONE bit should not be set in the same instruction that turns on the ADC. Refer to **Section 8.2.5 "A/D Conversion Procedure"**.

### 8.2.2 COMPLETION OF A CONVERSION

When the conversion is complete, the ADC module will:

- Clear the GO/DONE bit
- Set the ADIF flag bit
- Update the ADRESH:ADRESL registers with new conversion result

### 8.2.3 TERMINATING A CONVERSION

If a conversion must be terminated before completion, the GO/DONE bit can be cleared in software. The ADRESH:ADRESL registers will not be updated with the partially complete Analog-to-Digital Conversion sample. Instead, the ADRESH:ADRESL register pair will retain the value of the previous conversion. Additionally, a 2 TAD delay is required before another acquisition can be initiated. Following this delay, an input acquisition is automatically started on the selected channel.

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

### 8.2.4 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.

### 8.2.5 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 (See TRIS register)
  - Configure pin as analog
2. Configure the ADC module:
  - Select ADC conversion clock
  - Configure voltage reference
  - Select ADC input channel
  - Select result format
  - Turn on ADC module
3. Configure ADC interrupt (optional):
  - Clear ADC interrupt flag
  - Enable ADC interrupt
  - Enable peripheral interrupt
  - Enable global interrupt<sup>(1)</sup>
4. Wait the required acquisition time<sup>(2)</sup>.
5. Start conversion by setting the GO/DONE bit.
6. Wait for ADC conversion to complete by one of the following:
  - Polling the GO/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:** See **Section 8.3 "A/D Acquisition Requirements"**.

# PIC16F688

---

## EXAMPLE 8-1: 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
MOVWF  ADCON1      ;
BANKSEL TRISA      ;
BSF    TRISA,0     ;Set RA0 to input
BANKSEL ANSEL      ;
BSF    ANSEL,0     ;Set RA0 to analog
BANKSEL ADCON0     ;
MOVLW  B'10000001' ;Right justify,
MOVWF  ADCON0     ;Vdd Vref, AN0, On
CALL   SampleTime ;Acquisition delay
BSF    ADCON0,GO  ;Start conversion
BTFSC  ADCON0,GO  ;Is conversion done?
GOTO   $-1        ;No, test again
BANKSEL ADRESH     ;
MOVF   ADRESH,W   ;Read upper 2 bits
MOVWF  RESULTHI   ;store in GPR space
BANKSEL ADRESL     ;
MOVF   ADRESL,W   ;Read lower 8 bits
MOVWF  RESULTLO   ;Store in GPR space
```



## 8.2.6 ADC REGISTER DEFINITIONS

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

### REGISTER 8-1: ADCON0: A/D CONTROL REGISTER 0

R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	VCFG	—	CHS2	CHS1	CHS0	GO/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      **ADFM:** A/D Conversion Result Format Select bit  
 1 = Right justified  
 0 = Left justified
- bit 6      **VCFG:** Voltage Reference bit  
 1 = VREF pin  
 0 = VDD
- bit 5      **Unimplemented:** Read as '0'
- bit 4-2    **CHS<2:0>:** Analog Channel Select bits  
 000 = AN0  
 001 = AN1  
 010 = AN2  
 011 = AN3  
 100 = AN4  
 101 = AN5  
 110 = AN6  
 111 = AN7
- bit 1      **GO/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 8-2: ADCON1: A/D CONTROL REGISTER 1

U-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	ADCS2	ADCS1	ADCS0	—	—	—	—
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  
 x11 = FRC (clock derived from a dedicated internal oscillator = 500 kHz max)  
 100 = Fosc/4  
 101 = Fosc/16  
 110 = Fosc/64
- bit 3-0    **Unimplemented:** Read as '0'

# PIC16F688

## REGISTER 8-3: ADRESH: ADC RESULT REGISTER HIGH (ADRESH) ADFM = 0

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
ADRES9	ADRES8	ADRES7	ADRES6	ADRES5	ADRES4	ADRES3	ADRES2
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<9:2>**: ADC Result Register bits  
 Upper 8 bits of 10-bit conversion result

## REGISTER 8-4: ADRESL: ADC RESULT REGISTER LOW (ADRESL) ADFM = 0

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
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 7-6                      **ADRES<1:0>**: ADC Result Register bits  
 Lower 2 bits of 10-bit conversion result

bit 5-0                      **Reserved**: Do not use.

## REGISTER 8-5: ADRESH: ADC RESULT REGISTER HIGH (ADRESH) ADFM = 1

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	—	—	—	—	ADRES9	ADRES8
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                      **Reserved**: Do not use.

bit 1-0                      **ADRES<9:8>**: ADC Result Register bits  
 Upper 2 bits of 10-bit conversion result

## REGISTER 8-6: ADRESL: ADC RESULT REGISTER LOW (ADRESL) ADFM = 1

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  
 Lower 8 bits of 10-bit conversion result

## 8.3 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 8-4. 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), see Figure 8-4. **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 8-1 may be used. This equation assumes that 1/2 LSB error is used (1024 steps for the ADC). The 1/2 LSB error is the maximum error allowed for the ADC to meet its specified resolution.

### EQUATION 8-1: ACQUISITION TIME EXAMPLE

*Assumptions: Temperature = 50°C and external impedance of 10kΩ 5.0V VDD*

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

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

$$V_{APPLIED} \left( 1 - \frac{1}{2047} \right) = V_{CHOLD} \quad ;[1] \text{ } V_{CHOLD} \text{ charged to within } 1/2 \text{ lsb}$$

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

$$V_{APPLIED} \left( 1 - e^{-\frac{T_C}{RC}} \right) = V_{APPLIED} \left( 1 - \frac{1}{2047} \right) \quad ;\text{combining [1] and [2]}$$

*Solving for TC:*

$$\begin{aligned} T_C &= -CHOLD(RIC + RSS + RS) \ln(1/2047) \\ &= -10pF(1k\Omega + 7k\Omega + 10k\Omega) \ln(0.0004885) \\ &= 1.37\mu s \end{aligned}$$

*Therefore:*

$$\begin{aligned} T_{ACQ} &= 2\mu s + 1.37\mu s + [(50^\circ C - 25^\circ C)(0.05\mu s/^\circ C)] \\ &= 4.67\mu s \end{aligned}$$

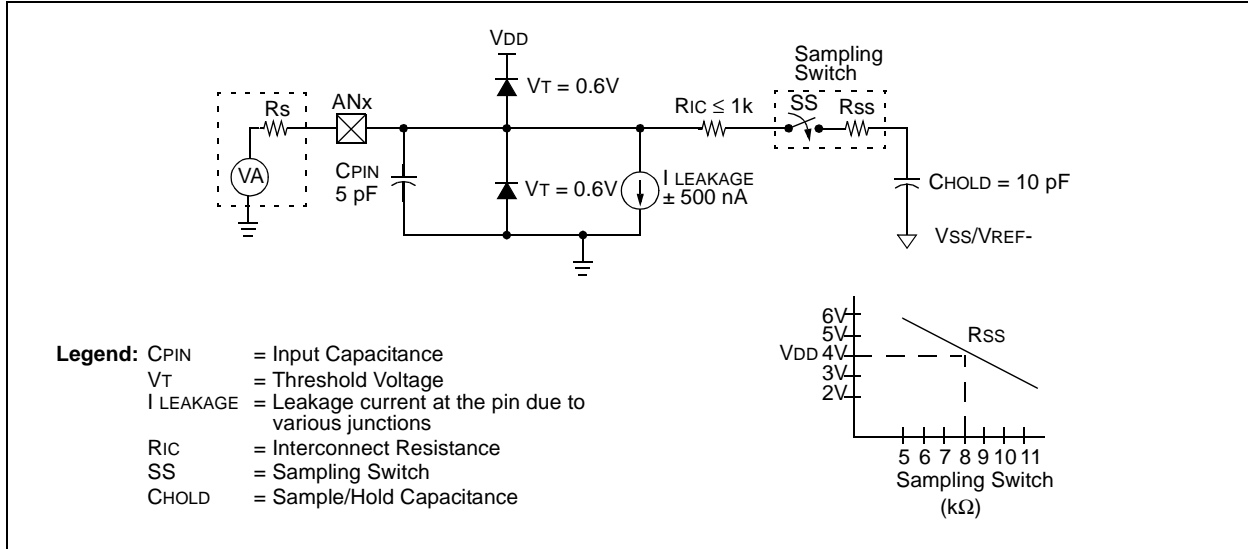
**Note 1:** The reference voltage (VREF) has no effect on the equation, since it cancels itself out.

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

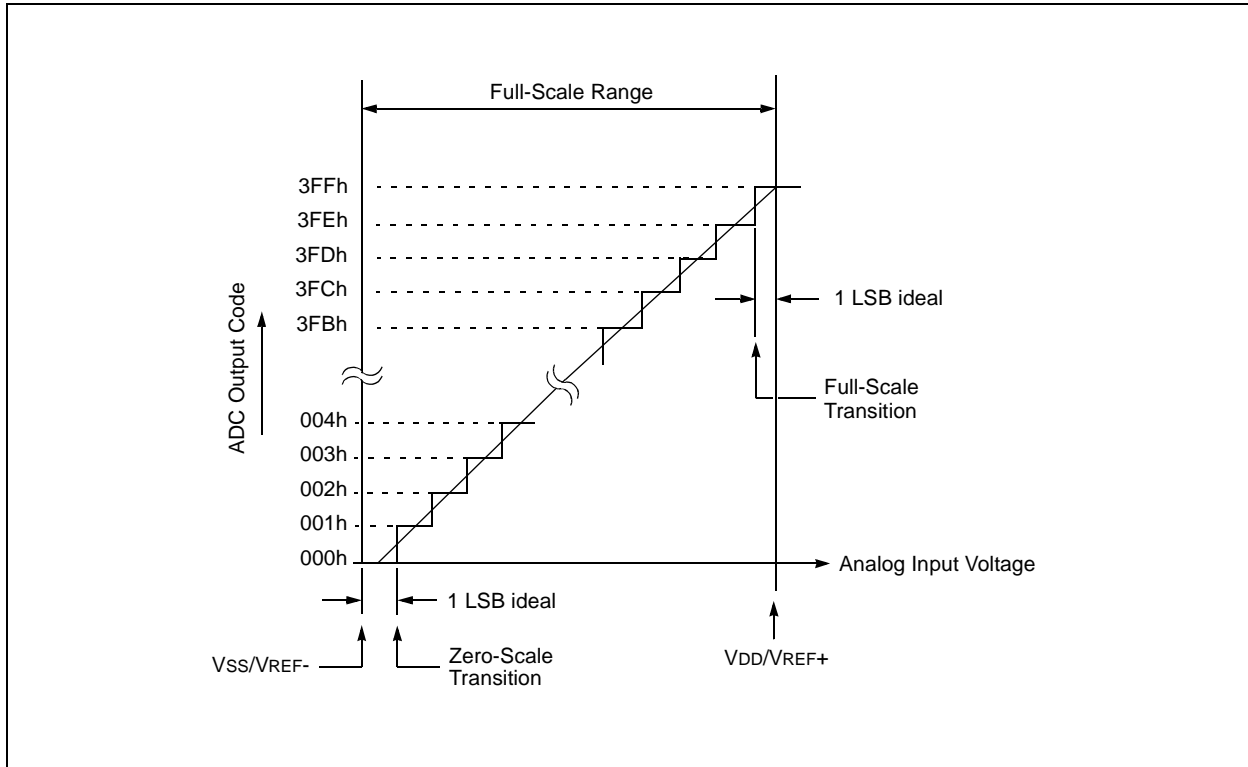
**3:** The maximum recommended impedance for analog sources is 10 kΩ. This is required to meet the pin leakage specification.

# PIC16F688

**FIGURE 8-4: ANALOG INPUT MODEL**



**FIGURE 8-5: ADC TRANSFER FUNCTION**



**TABLE 8-2: SUMMARY OF ASSOCIATED ADC REGISTERS**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
ADCON0	ADFM	VCFG	—	CHS2	CHS1	CHS0	GO/DONE	ADON	00-0 0000	00-0 0000
ADCON1	—	ADCS2	ADCS1	ADCS0	—	—	—	—	-000 ----	-000 ----
ANSEL	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0	1111 1111	1111 1111
ADRESH	A/D Result Register High Byte								xxxx xxxx	uuuu uuuu
ADRESL	A/D Result Register Low Byte								xxxx xxxx	uuuu uuuu
INTCON	GIE	PEIE	T0IE	INTE	RAIE	T0IF	INTF	RAIF	0000 000x	0000 000x
PIE1	EEIE	ADIE	RCIE	C2IE	C1IE	OSFIE	TXIE	TMR1IE	0000 0000	0000 0000
PIR1	EEIF	ADIF	RCIF	C2IF	C1IF	OSFIF	TXIF	TMR1IF	0000 0000	0000 0000
PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	--x0 x000	--x0 x000
PORTC	—	—	RC5	RC4	RC3	RC2	RC1	RC0	--xx 0000	--xx 0000
TRISA	—	—	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	--11 1111	--11 1111
TRISC	—	—	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	--11 1111	--11 1111

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

# PIC16F688

---

NOTES:

## 9.0 DATA EEPROM AND FLASH PROGRAM MEMORY CONTROL

Data EEPROM memory is readable and writable and the Flash program memory is readable during normal operation (full  $V_{DD}$  range). These memories are not directly mapped in the register file space. Instead, they are indirectly addressed through the Special Function Registers. There are six SFRs used to access these memories:

- EECON1
- EECON2
- EEDAT
- EEDATH
- EEADR
- EEADRH

When interfacing the data memory block, EEDAT holds the 8-bit data for read/write, and EEADR holds the address of the EE data location being accessed. This device has 256 bytes of data EEPROM with an address range from 0h to 0FFh.

When interfacing the program memory block, the EEDAT and EEDATH registers form a 2-byte word that holds the 14-bit data for read/write, and the EEADR and EEADRH registers form a 2-byte word that holds the 12-bit address of the EEPROM location being accessed. This device has 4K words of program EEPROM with an address range from 0h to 0FFFh. The program memory allows one word reads.

The EEPROM data memory allows byte read and write. A byte write automatically erases the location and writes the new data (erase before write).

The write time is controlled by an on-chip timer. The write/erase voltages are generated by an on-chip charge pump rated to operate over the voltage range of the device for byte or word operations.

When the device is code-protected, the CPU may continue to read and write the data EEPROM memory and read the program memory. When code-protected, the device programmer can no longer access data or program memory.

## 9.1 EEADR and EEADRH Registers

The EEADR and EEADRH registers can address up to a maximum of 256 bytes of data EEPROM or up to a maximum of 4K words of program EEPROM.

When selecting a program address value, the MSB of the address is written to the EEADRH register and the LSB is written to the EEADR register. When selecting a data address value, only the LSB of the address is written to the EEADR register.

### 9.1.1 EECON1 AND EECON2 REGISTERS

EECON1 is the control register for EE memory accesses.

Control bit EEPGD determines if the access will be a program or data memory access. When clear, as it is when reset, any subsequent operations will operate on the data memory. When set, any subsequent operations will operate on the program memory. Program memory can only be read.

Control bits RD and WR initiate read and write, respectively. These bits cannot be cleared, only set, in software. They are cleared in hardware at completion of the read or write operation. The inability to clear the WR bit in software prevents the accidental, premature termination of a write operation.

The WREN bit, when set, will allow a write operation to data EEPROM. On power-up, the WREN bit is clear. The WRERR bit is set when a write operation is interrupted by a  $\overline{MCLR}$  or a WDT Time-out Reset during normal operation. In these situations, following Reset, the user can check the WRERR bit and rewrite the location. The data and address will be unchanged in the EEDAT and EEADR registers.

Interrupt flag bit EEIF of the PIR1 register is set when write is complete. It must be cleared in the software.

EECON2 is not a physical register. Reading EECON2 will read all '0's. The EECON2 register is used exclusively in the data EEPROM write sequence.

# PIC16F688

## REGISTER 9-1: EEDAT: EEPROM DATA 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
EEDAT7	EEDAT6	EEDAT5	EEDAT4	EEDAT3	EEDAT2	EEDAT1	EEDAT0
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                      **EEDATn**: Byte Value to Write to or Read from Data EEPROM bits

## REGISTER 9-2: EEADR: EEPROM 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
EEADR7	EEADR6	EEADR5	EEADR4	EEADR3	EEADR2	EEADR1	EEADR0
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                      **EEADR<7:0>**: 8 Least Significant Address bits for EEPROM Read/Write Operation<sup>(1)</sup> or Read from program memory

## REGISTER 9-3: EEDATH: EEPROM DATA HIGH BYTE REGISTER

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	EEDATH5	EEDATH4	EEDATH3	EEDATH2	EEDATH1	EEDATH0
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                      **EEDATH<5:0>**: 6 Most Significant Data bits from program memory

## REGISTER 9-4: EEADRH: EEPROM ADDRESS HIGH BYTE REGISTER

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	EEADRH3	EEADRH2	EEADRH1	EEADRH0
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 3-0                      **EEADRH<3:0>**: Specifies the 4 Most Significant Address bits or high bits for program memory reads



## REGISTER 9-5: EECON1: EEPROM CONTROL REGISTER

R/W-x	U-0	U-0	U-0	R/W-x	R/W-0	R/S-0	R/S-0
EEPGD	—	—	—	WRERR	WREN	WR	RD
bit 7							bit 0

### Legend:

S = Bit can only be set

R = Readable bit

-n = Value at POR

W = Writable bit

'1' = Bit is set

U = Unimplemented bit, read as '0'

'0' = Bit is cleared

x = Bit is unknown

- bit 7      **EEPGD:** Program/Data EEPROM Select bit  
           1 = Accesses program memory  
           0 = Accesses data memory
- bit 6-4    **Unimplemented:** Read as '0'
- bit 3      **WRERR:** EEPROM Error Flag bit  
           1 = A write operation is prematurely terminated (any  $\overline{\text{MCLR}}$  Reset, any WDT Reset during normal operation or BOR Reset)  
           0 = The write operation completed
- bit 2      **WREN:** EEPROM Write Enable bit  
           1 = Allows write cycles  
           0 = Inhibits write to the data EEPROM
- bit 1      **WR:** Write Control bit  
           **EEPGD = 1:**  
           This bit is ignored  
           **EEPGD = 0:**  
           1 = Initiates a write cycle (The bit is cleared by hardware once write is complete. The WR bit can only be set, not cleared, in software.)  
           0 = Write cycle to the data EEPROM is complete
- bit 0      **RD:** Read Control bit  
           1 = Initiates a memory read (the RD is cleared in hardware and can only be set, not cleared, in software.)  
           0 = Does not initiate a memory read

# PIC16F688

## 9.1.2 READING THE DATA EEPROM MEMORY

To read a data memory location, the user must write the address to the EEADR register, clear the EEPGD control bit of the EECON1 register, and then set control bit RD of the EECON1 register. The data is available in the very next cycle, in the EEDAT register; therefore, it can be read in the next instruction. EEDAT will hold this value until another read or until it is written to by the user (during a write operation).

### EXAMPLE 9-1: DATA EEPROM READ

```
BANKSEL EEADR      ;
MOVLW  DATA_EE_ADDR  ;
MOVWF  EEADR        ;Data Memory
                    ;Address to read
BCF    EECON1, EEPGD ;Point to DATA
                    ;memory
BSF    EECON1, RD    ;EE Read
MOVF  EEDAT, W      ;W = EEDAT
```

## 9.1.3 WRITING TO THE DATA EEPROM MEMORY

To write an EEPROM data location, the user must first write the address to the EEADR register and the data to the EEDAT register. Then the user must follow a specific sequence to initiate the write for each byte.

The write will not initiate if the above sequence is not followed exactly (write 55h to EECON2, write AAh to EECON2, then set WR bit) for each byte. Interrupts should be disabled during this code segment.

Additionally, the WREN bit in EECON1 must be set to enable write. This mechanism prevents accidental writes to data EEPROM due to errant (unexpected) code execution (i.e., lost programs). The user should keep the WREN bit clear at all times, except when updating EEPROM. The WREN bit is not cleared by hardware.

After a write sequence has been initiated, clearing the WREN bit will not affect this write cycle. The WR bit will be inhibited from being set unless the WREN bit is set.

At the completion of the write cycle, the WR bit is cleared in hardware and the EE Write Complete Interrupt Flag bit (EEIF) is set. The user can either enable this interrupt or poll this bit. EEIF must be cleared by software.

### EXAMPLE 9-2: DATA EEPROM WRITE

```
BANKSEL EEADR      ;
MOVLW  DATA_EE_ADDR  ;
MOVWF  EEADR        ;Data Memory Address to write
MOVLW  DATA_EE_DATA  ;
MOVWF  EEDAT        ;Data Memory Value to write
BANKSEL EECON1     ;
BCF    EECON1, EEPGD ;Point to DATA memory
BSF    EECON1, WREN  ;Enable writes

BCF    INTCON, GIE  ;Disable INTs.
BTFSC  INTCON, GIE  ;SEE AN576
GOTO   $-2
MOVLW  55h          ;
MOVWF  EECON2      ;Write 55h
MOVLW  AAh          ;
MOVWF  EECON2      ;Write AAh
BSF    EECON1, WR   ;Set WR bit to begin write
BSF    INTCON, GIE  ;Enable INTs.

SLEEP      ;Wait for interrupt to signal write complete
BCF    EECON1, WREN ;Disable writes
```

Required Sequence

## 9.1.4 READING THE FLASH PROGRAM MEMORY

To read a program memory location, the user must write two bytes of the address to the EEADR and EEADRH registers, set the EEPGD control bit of the EECON1 register, and then set control bit RD of the EECON1 register. Once the read control bit is set, the program memory Flash controller will use the second instruction cycle to read the data. This causes the second instruction immediately following the “BSF EECON1, RD” instruction to be ignored. The data is available in the very next cycle, in the EEDAT and EEDATH registers; therefore, it can be read as two bytes in the following instructions.

EEDAT and EEDATH registers will hold this value until another read or until it is written to by the user (during a write operation).

**Note 1:** The two instructions following a program memory read are required to be NOP's. This prevents the user from executing a two-cycle instruction on the next instruction after the RD bit is set.

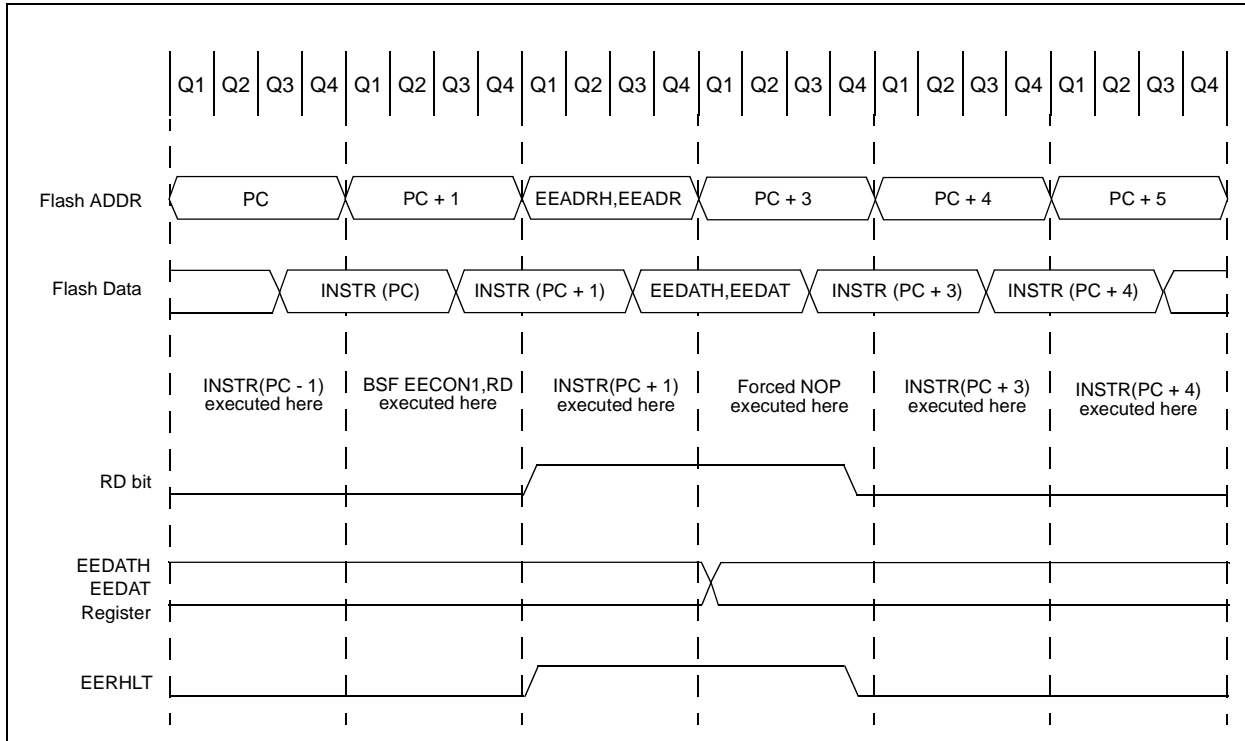
**2:** If the WR bit is set when EEPGD = 1, it will be immediately reset to '0' and no operation will take place.

### EXAMPLE 9-3: FLASH PROGRAM READ

Required Sequence	BANKSEL	EEADR	;
	MOVLW	MS_PROG_EE_ADDR	;
	MOVWF	EEADRH	;MS Byte of Program Address to read
	MOVLW	LS_PROG_EE_ADDR	;
	MOVWF	EEADR	;LS Byte of Program Address to read
	BANKSEL	EECON1	;
	BSF	EECON1, EEPGD	;Point to PROGRAM memory
	BSF	EECON1, RD	;EE Read
			;First instruction after BSF EECON1,RD executes normally
			;Any instructions here are ignored as program memory is read in second cycle after BSF EECON1,RD
		;	
	BANKSEL	EEDAT	;
	MOVF	EEDAT, W	;W = LS Byte of Program Memory
	MOVWF	LOWPMBYTE	;
	MOVF	EEDATH, W	;W = MS Byte of Program EEDAT
	MOVWF	HIGHPMBYTE	;
	BCF	STATUS, RP1	;Bank 0

# PIC16F688

**FIGURE 9-1: FLASH PROGRAM MEMORY READ CYCLE EXECUTION**



**TABLE 9-1: SUMMARY OF REGISTERS ASSOCIATED WITH DATA EEPROM**

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
EECON1	EEPGD	—	—	—	WRERR	WREN	WR	RD	x--- x000	0--- q000
EECON2	EEPROM Control Register 2 (not a physical register)								---- ----	---- ----
EEADR	EEADR7	EEADR6	EEADR5	EEADR4	EEADR3	EEADR2	EEADR1	EEADR0	0000 0000	0000 0000
EEADRH	—	—	—	—	EEADRH3	EEADRH2	EEADRH1	EEADRH0	---- 0000	---- 0000
EEDAT	EEDAT7	EEDAT6	EEDAT5	EEDAT4	EEDAT3	EEDAT2	EEDAT1	EEDAT0	0000 0000	0000 0000
EEDATH	—	—	EEDATH5	EEDATH4	EEDATH3	EEDATH2	EEDATH1	EEDATH0	--00 0000	--00 0000
INTCON	GIE	PEIE	T0IE	INTE	RABIE	T0IF	INTF	RABIF	0000 000x	0000 000x
PIE1	EEIE	ADIE	RCIE	C2IE	C1IE	OSFIE	TXIE	TMR1IE	0000 0000	0000 0000
PIR1	EEIF	ADIF	RCIF	C2IF	C1IF	OSFIF	TXIF	TMR1IF	0000 0000	0000 0000

**Legend:** x = unknown, u = unchanged, — = unimplemented read as '0', q = value depends upon condition. Shaded cells are not used by data EEPROM module.

## 10.0 ENHANCED UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER (EUSART)

The Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART) 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 EUSART, 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 EUSART module includes the following capabilities:

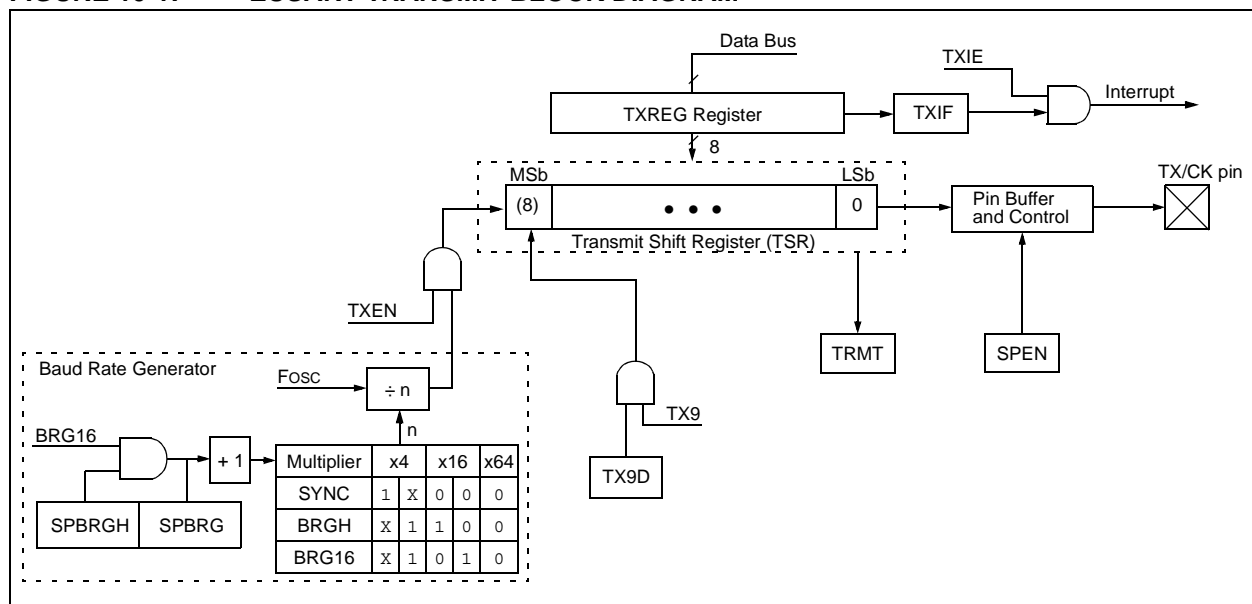
- Full-duplex asynchronous transmit and receive
- Two-character input buffer
- One-character output buffer
- Programmable 8-bit or 9-bit character length
- Address detection in 9-bit mode
- Input buffer overrun error detection
- Received character framing error detection
- Half-duplex synchronous master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

The EUSART module implements the following additional features, making it ideally suited for use in Local Interconnect Network (LIN) bus systems:

- Automatic detection and calibration of the baud rate
- Wake-up on Break reception
- 13-bit Break character transmit

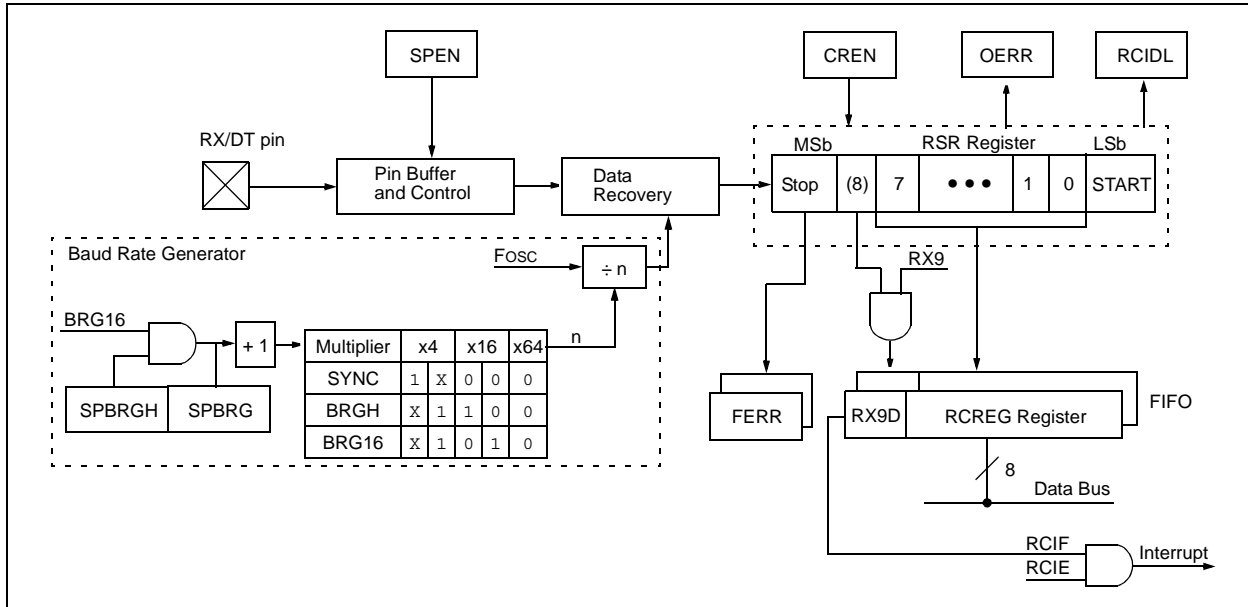
Block diagrams of the EUSART transmitter and receiver are shown in Figure 10-1 and Figure 10-2.

**FIGURE 10-1: EUSART TRANSMIT BLOCK DIAGRAM**



# PIC16F688

**FIGURE 10-2: EUSART RECEIVE BLOCK DIAGRAM**



The operation of the EUSART module is controlled through three registers:

- Transmit Status and Control (TXSTA)
- Receive Status and Control (RCSTA)
- Baud Rate Control (BAUDCTL)

These registers are detailed in Register 10-1, Register 10-2 and Register 10-3, respectively.

## 10.1 EUSART Asynchronous Mode

The EUSART transmits and receives data using the standard non-return-to-zero (NRZ) format. NRZ is implemented with two levels: a  $V_{OH}$  mark state which represents a '1' data bit, and a  $V_{OL}$  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  $1/(\text{Baud Rate})$ . An on-chip dedicated 8-bit/16-bit Baud Rate Generator is used to derive standard baud rate frequencies from the system oscillator. See Table 10-5 for examples of baud rate configurations.

The EUSART transmits and receives the LSb first. The EUSART'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.

### 10.1.1 EUSART ASYNCHRONOUS TRANSMITTER

The EUSART transmitter block diagram is shown in Figure 10-1. 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.

#### 10.1.1.1 Enabling the Transmitter

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

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

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

Setting the TXEN bit of the TXSTA register enables the transmitter circuitry of the EUSART. Clearing the SYNC bit of the TXSTA register configures the EUSART for asynchronous operation. Setting the SPEN bit of the RCSTA register enables the EUSART and automatically configures the TX/CK I/O pin as an output. If the TX/CK pin is shared with an analog peripheral the analog I/O function must be disabled by clearing the corresponding ANSEL bit.

- 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 EUSART 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.

#### 10.1.1.2 Transmitting Data

A transmission is initiated by writing a character to the TXREG register. If this is the first character, or the previous character has been completely flushed from the TSR, the data in the TXREG is immediately transferred to the TSR register. If the TSR still contains all or part of a previous character, the new character data is held in the TXREG until the Stop bit of the previous character has been transmitted. The pending character in the TXREG is then transferred to the TSR in one  $T_{CY}$  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.

#### 10.1.1.3 Transmit Interrupt Flag

The TXIF interrupt flag bit of the PIR1 register is set whenever the EUSART 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.

# PIC16F688

## 10.1.1.4 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.

## 10.1.1.5 Transmitting 9-Bit Characters

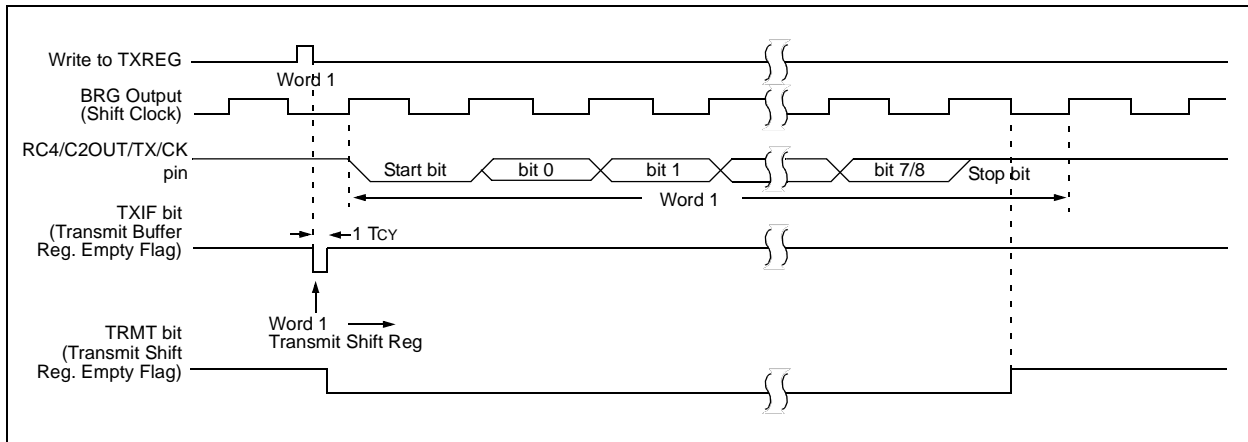
The EUSART supports 9-bit character transmissions. When the TX9 bit of the TXSTA register is set the EUSART will shift 9 bits out for each character transmitted. The TX9D bit of the TXSTA register is the ninth, and Most Significant, data bit. When transmitting 9-bit data, the TX9D data bit must be written 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. See **Section 10.1.2.7 “Address Detection”** for more information on the Address mode.

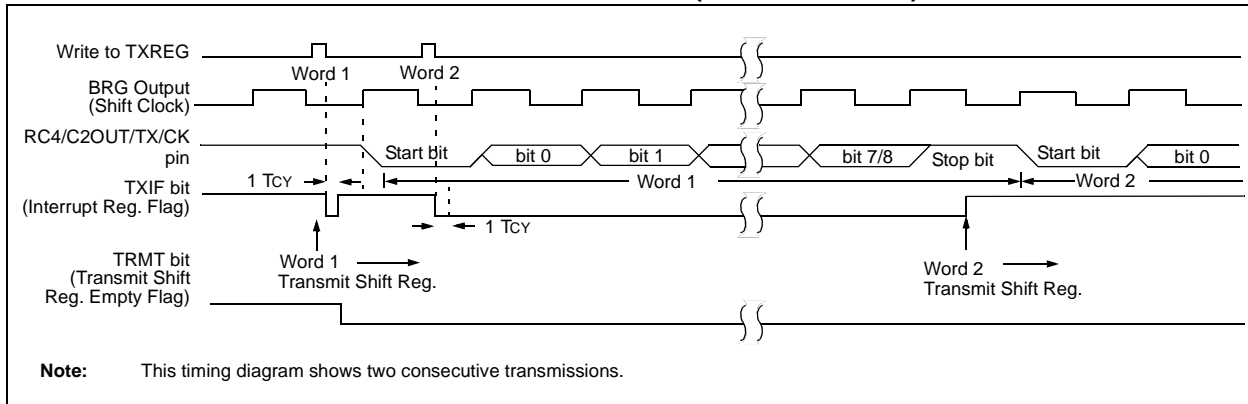
## 10.1.1.6 Asynchronous Transmission Set-up:

1. Initialize the SPBRGH, SPBRG register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see **Section 10.3 “EUSART 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. 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 10-3: ASYNCHRONOUS TRANSMISSION**



**FIGURE 10-4: ASYNCHRONOUS TRANSMISSION (BACK-TO-BACK)**





**TABLE 10-1: 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
BAUDCTL	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	01-0 0-00	01-0 0-00
INTCON	GIE	PEIE	T0IE	INTE	RAIE	T0IF	INTF	RAIF	0000 000x	0000 000x
PIE1	EEIE	ADIE	RCIE	C2IE	C1IE	OSFIE	TXIE	TMR1IE	0000 0000	0000 0000
PIR1	EEIF	ADIF	RCIF	C2IF	C1IF	OSFIF	TXIF	TMR1IF	0000 0000	0000 0000
RCREG	EUSART 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
SPBRGH	BRG15	BRG14	BRG13	BRG12	BRG11	BRG10	BRG9	BRG8	0000 0000	0000 0000
TRISC	—	—	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	--11 1111	--11 1111
TXREG	EUSART Transmit Data Register								0000 0000	0000 0000
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010	0000 0010

**Legend:** x = unknown, – = unimplemented read as '0'. Shaded cells are not used for Asynchronous Transmission.

# PIC16F688

## 10.1.2 EUSART ASYNCHRONOUS RECEIVER

The Asynchronous mode would typically be used in RS-232 systems. The receiver block diagram is shown in Figure 10-2. 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 EUSART receiver. The FIFO and RSR registers are not directly accessible by software. Access to the received data is via the RCREG register.

### 10.1.2.1 Enabling the Receiver

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

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

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

Setting the CREN bit of the RCSTA register enables the receiver circuitry of the EUSART. Clearing the SYNC bit of the TXSTA register configures the EUSART for asynchronous operation. Setting the SPEN bit of the RCSTA register enables the EUSART and automatically configures the RX/DT I/O pin as an input. If the RX/DT pin is shared with an analog peripheral the analog I/O function must be disabled by clearing the corresponding ANSEL bit.

**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 EUSART 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.

### 10.1.2.2 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. See **Section 10.1.2.4 "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 EUSART 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. See **Section 10.1.2.5 "Receive Overrun Error"** for more information on overrun errors.

### 10.1.2.3 Receive Interrupts

The RCIF interrupt flag bit of the PIR1 register is set whenever the EUSART 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 the following bits:

- RCIE 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 will be set when there is an unread character in the FIFO, regardless of the state of interrupt enable bits.

## 10.1.2.4 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 EUSART. Clearing the CREN bit of the RCSTA register does not affect the FERR bit. A framing error by itself does not generate an interrupt.

<p><b>Note:</b> If all receive characters in the receive FIFO have framing errors, repeated reads of the RCREG will not clear the FERR bit.</p>
---

## 10.1.2.5 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 resetting the EUSART by clearing the SPEN bit of the RCSTA register.

## 10.1.2.6 Receiving 9-bit Characters

The EUSART supports 9-bit character reception. When the RX9 bit of the RCSTA register is set the EUSART 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.

## 10.1.2.7 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. 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.

# PIC16F688

## 10.1.2.8 Asynchronous Reception Set-up:

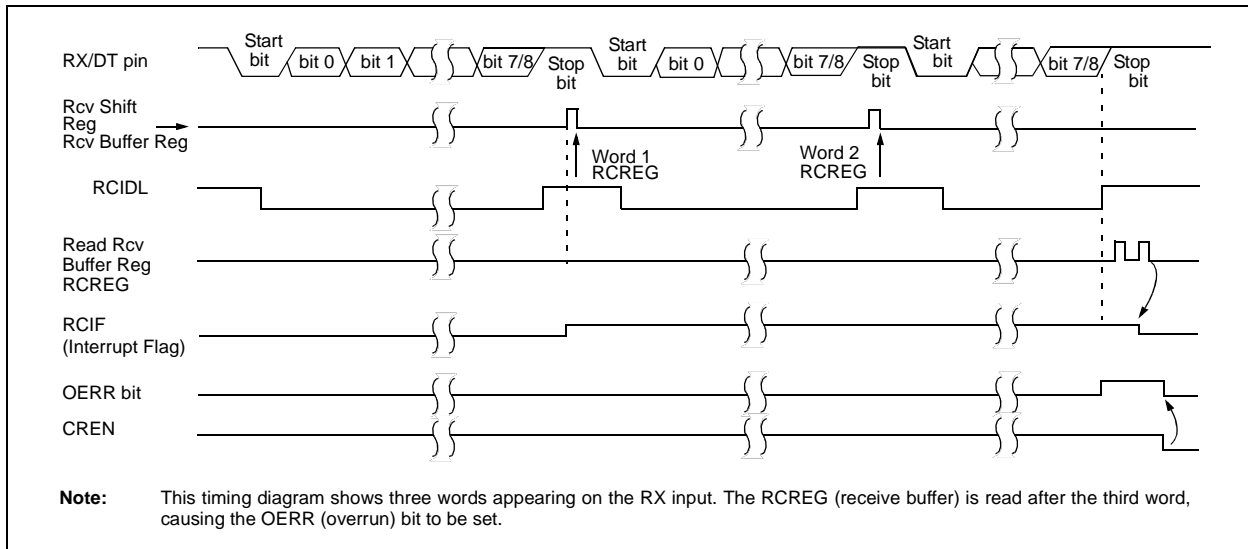
1. Initialize the SPBRGH, SPBRG register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see **Section 10.3 “EUSART Baud Rate Generator (BRG)”**).
2. Enable the serial port by setting the SPEN bit. The SYNC bit must be clear for asynchronous operation.
3. If interrupts are desired, set the RCIE interrupt enable bit and set 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 will be set when a character is transferred from the RSR to the receive buffer. An interrupt will be generated if the RCIE interrupt enable bit 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.

## 10.1.2.9 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 SPBRGH, SPBRG register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see **Section 10.3 “EUSART 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 interrupt enable bit and set 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 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 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 10-5: ASYNCHRONOUS RECEPTION**



**TABLE 10-2: 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
BAUDCTL	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	01-0 0-00	01-0 0-00
INTCON	GIE	PEIE	T0IE	INTE	RAIE	T0IF	INTF	RAIF	0000 000x	0000 000x
PIE1	EEIE	ADIE	RCIE	C2IE	C1IE	OSFIE	TXIE	TMR1IE	0000 0000	0000 0000
PIR1	EEIF	ADIF	RCIF	C2IF	C1IF	OSFIF	TXIF	TMR1IF	0000 0000	0000 0000
RCREG	EUSART 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
SPBRGH	BRG15	BRG14	BRG13	BRG12	BRG11	BRG10	BRG9	BRG8	0000 0000	0000 0000
TRISC	—	—	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	--11 1111	--11 1111
TXREG	EUSART Transmit Data Register								0000 0000	0000 0000
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010	0000 0010

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

# PIC16F688

## 10.2 Clock Accuracy with Asynchronous Operation

The factory calibrates the internal oscillator block output (INTOSC). However, the INTOSC frequency may drift as VDD or temperature changes, and this directly affects the asynchronous baud rate. Two methods may be used to adjust the baud rate clock, but both require a reference clock source of some kind.

The first (preferred) method uses the OSCTUNE register to adjust the INTOSC output. Adjusting the value in the OSCTUNE register allows for fine resolution changes to the system clock source. See **Section 3.5 “Internal Clock Modes”** for more information.

The other method adjusts the value in the Baud Rate Generator. This can be done automatically with the Auto-Baud Detect feature (see **Section 10.3.1 “Auto-Baud Detect”**). There may not be fine enough resolution when adjusting the Baud Rate Generator to compensate for a gradual change in the peripheral clock frequency.

### REGISTER 10-1: TXSTA: TRANSMIT STATUS AND CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN <sup>(1)</sup>	SYNC	SENDB	BRGH	TRMT	TX9D
bit 7						bit 0	

<b>Legend:</b>			
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<sup>(1)</sup>  
 1 = Transmit enabled  
 0 = Transmit disabled
- bit 4      **SYNC:** EUSART Mode Select bit  
 1 = Synchronous mode  
 0 = Asynchronous mode
- bit 3      **SENDB:** Send Break Character bit  
Asynchronous mode:  
 1 = Send Sync Break on next transmission (cleared by hardware upon completion)  
 0 = Sync Break transmission completed  
Synchronous mode:  
 Don't care
- 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 Sync mode.

## REGISTER 10-2: RCSTA: RECEIVE STATUS AND CONTROL REGISTER<sup>(1)</sup>

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	<p><b>SPEN:</b> Serial Port Enable bit</p> <p>1 = Serial port enabled (configures RX/DT and TX/CK pins as serial port pins)</p> <p>0 = Serial port disabled (held in Reset)</p>
bit 6	<p><b>RX9:</b> 9-bit Receive Enable bit</p> <p>1 = Selects 9-bit reception</p> <p>0 = Selects 8-bit reception</p>
bit 5	<p><b>SREN:</b> Single Receive Enable bit</p> <p><u>Asynchronous mode:</u></p> <p>Don't care</p> <p><u>Synchronous mode – Master:</u></p> <p>1 = Enables single receive</p> <p>0 = Disables single receive</p> <p>This bit is cleared after reception is complete.</p> <p><u>Synchronous mode – Slave</u></p> <p>Don't care</p>
bit 4	<p><b>CREN:</b> Continuous Receive Enable bit</p> <p><u>Asynchronous mode:</u></p> <p>1 = Enables receiver</p> <p>0 = Disables receiver</p> <p><u>Synchronous mode:</u></p> <p>1 = Enables continuous receive until enable bit CREN is cleared (CREN overrides SREN)</p> <p>0 = Disables continuous receive</p>
bit 3	<p><b>ADDEN:</b> Address Detect Enable bit</p> <p><u>Asynchronous mode 9-bit (RX9 = 1):</u></p> <p>1 = Enables address detection, enable interrupt and load the receive buffer when RSR&lt;8&gt; is set</p> <p>0 = Disables address detection, all bytes are received and ninth bit can be used as parity bit</p> <p><u>Asynchronous mode 8-bit (RX9 = 0):</u></p> <p>Don't care</p>
bit 2	<p><b>FERR:</b> Framing Error bit</p> <p>1 = Framing error (can be updated by reading RCREG register and receive next valid byte)</p> <p>0 = No framing error</p>
bit 1	<p><b>OERR:</b> Overrun Error bit</p> <p>1 = Overrun error (can be cleared by clearing bit CREN)</p> <p>0 = No overrun error</p>
bit 0	<p><b>RX9D:</b> Ninth bit of Received Data</p> <p>This can be address/data bit or a parity bit and must be calculated by user firmware.</p>

# PIC16F688

## REGISTER 10-3: BAUDCTL: BAUD RATE CONTROL REGISTER

R-0	R-1	U-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN
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            **ABDOVF:** Auto-Baud Detect Overflow bit  
Asynchronous mode:  
 1 = Auto-baud timer overflowed  
 0 = Auto-baud timer did not overflow  
Synchronous mode:  
 Don't care
- bit 6            **RCIDL:** Receive Idle Flag bit  
Asynchronous mode:  
 1 = Receiver is Idle  
 0 = Start bit has been received and the receiver is receiving  
Synchronous mode:  
 Don't care
- bit 5            **Unimplemented:** Read as '0'
- bit 4            **SCKP:** Synchronous Clock Polarity Select bit  
Asynchronous mode:  
 1 = Transmit inverted data to the RB7/TX/CK pin  
 0 = Transmit non-inverted data to the RB7/TX/CK pin  
Synchronous mode:  
 1 = Data is clocked on rising edge of the clock  
 0 = Data is clocked on falling edge of the clock
- bit 3            **BRG16:** 16-bit Baud Rate Generator bit  
 1 = 16-bit Baud Rate Generator is used  
 0 = 8-bit Baud Rate Generator is used
- bit 2            **Unimplemented:** Read as '0'
- bit 1            **WUE:** Wake-up Enable bit  
Asynchronous mode:  
 1 = Receiver is waiting for a falling edge. No character will be received byte RCIF will be set. WUE will automatically clear after RCIF is set.  
 0 = Receiver is operating normally  
Synchronous mode:  
 Don't care
- bit 0            **ABDEN:** Auto-Baud Detect Enable bit  
Asynchronous mode:  
 1 = Auto-Baud Detect mode is enabled (clears when auto-baud is complete)  
 0 = Auto-Baud Detect mode is disabled  
Synchronous mode:  
 Don't care



## 10.3 EUSART Baud Rate Generator (BRG)

The Baud Rate Generator (BRG) is an 8-bit or 16-bit timer that is dedicated to the support of both the asynchronous and synchronous EUSART operation. By default, the BRG operates in 8-bit mode. Setting the BRG16 bit of the BAUDCTL register selects 16-bit mode.

The SPBRGH, SPBRG register pair determines the period of the free running baud rate timer. In Asynchronous mode the multiplier of the baud rate period is determined by both the BRGH bit of the TXSTA register and the BRG16 bit of the BAUDCTL register. In Synchronous mode, the BRGH bit is ignored.

Table 10-3 contains the formulas for determining the baud rate. Example 10-1 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 10-3. It may be advantageous to use the high baud rate (BRGH = 1), or the 16-bit BRG (BRG16 = 1) to reduce the baud rate error. The 16-bit BRG mode is used to achieve slow baud rates for fast oscillator frequencies.

Writing a new value to the SPBRGH, SPBRG register pair 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.

If the system clock is changed during an active receive operation, a receive error or data loss may result. To avoid this problem, check the status of the RCIDL bit to make sure that the receive operation is Idle before changing the system clock.

### EXAMPLE 10-1: CALCULATING BAUD RATE ERROR

For a device with Fosc of 16 MHz, desired baud rate of 9600, Asynchronous mode, 8-bit BRG:

$$\text{Desired Baud Rate} = \frac{F_{OSC}}{64(SPBRGH:SPBRG) + 1}$$

Solving for SPBRGH:SPBRG:

$$X = \frac{F_{OSC}}{\text{Desired Baud Rate} \cdot 64} - 1$$

$$= \frac{16000000}{9600 \cdot 64} - 1$$

$$= [25.042] = 25$$

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

$$= 9615$$

$$\text{Error} = \frac{\text{Calc. Baud Rate} - \text{Desired Baud Rate}}{\text{Desired Baud Rate}}$$

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

**TABLE 10-3: BAUD RATE FORMULAS**

Configuration Bits			BRG/EUSART Mode	Baud Rate Formula
SYNC	BRG16	BRGH		
0	0	0	8-bit/Asynchronous	Fosc/[64 (n+1)]
0	0	1	8-bit/Asynchronous	Fosc/[16 (n+1)]
0	1	0	16-bit/Asynchronous	
0	1	1	16-bit/Asynchronous	Fosc/[4 (n+1)]
1	0	x	8-bit/Synchronous	
1	1	x	16-bit/Synchronous	

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

**TABLE 10-4: 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
BAUDCTL	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	01-0 0-00	01-0 0-00
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
SPBRGH	BRG15	BRG14	BRG13	BRG12	BRG11	BRG10	BRG9	BRG8	0000 0000	0000 0000
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010	0000 0010

**Legend:** x = unknown, — = unimplemented read as '0'. Shaded cells are not used for the Baud Rate Generator.

# PIC16F688

**TABLE 10-5: BAUD RATES FOR ASYNCHRONOUS MODES**

BAUD RATE	SYNC = 0, BRGH = 0, BRG16 = 0											
	Fosc = 20.000 MHz			Fosc = 18.432 MHz			Fosc = 11.0592 MHz			Fosc = 8.000 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	—	—	—	—	—	—	—	—	—	—	—	—
1200	1221	1.73	255	1200	0.00	239	1200	0.00	143	1202	0.16	103
2400	2404	0.16	129	2400	0.00	119	2400	0.00	71	2404	0.16	51
9600	9470	-1.36	32	9600	0.00	29	9600	0.00	17	9615	0.16	12
10417	10417	0.00	29	10286	-1.26	27	10165	-2.42	16	10417	0.00	11
19.2k	19.53k	1.73	15	19.20k	0.00	14	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, BRG16 = 0											
	Fosc = 4.000 MHz			Fosc = 3.6864 MHz			Fosc = 2.000 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	103	300	0.16	51
1200	1202	0.16	51	1200	0.00	47	1202	0.16	25	1202	0.16	12
2400	2404	0.16	25	2400	0.00	23	2404	0.16	12	—	—	—
9600	—	—	—	9600	0.00	5	—	—	—	—	—	—
10417	10417	0.00	5	—	—	—	10417	0.00	2	—	—	—
19.2k	—	—	—	19.20k	0.00	2	—	—	—	—	—	—
57.6k	—	—	—	57.60k	0.00	0	—	—	—	—	—	—
115.2k	—	—	—	—	—	—	—	—	—	—	—	—

BAUD RATE	SYNC = 0, BRGH = 1, BRG16 = 0											
	Fosc = 20.000 MHz			Fosc = 18.432 MHz			Fosc = 11.0592 MHz			Fosc = 8.000 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	—	—	—	—	—	—	—	—	—	—	—	—
1200	—	—	—	—	—	—	—	—	—	—	—	—
2400	—	—	—	—	—	—	—	—	—	2404	0.16	207
9600	9615	0.16	129	9600	0.00	119	9600	0.00	71	9615	0.16	51
10417	10417	0.00	119	10378	-0.37	110	10473	0.53	65	10417	0.00	47
19.2k	19.23k	0.16	64	19.20k	0.00	59	19.20k	0.00	35	19231	0.16	25
57.6k	56.82k	-1.36	21	57.60k	0.00	19	57.60k	0.00	11	55556	-3.55	8
115.2k	113.64k	-1.36	10	115.2k	0.00	9	115.2k	0.00	5	—	—	—

**TABLE 10-5: BAUD RATES FOR ASYNCHRONOUS MODES (CONTINUED)**

BAUD RATE	SYNC = 0, BRGH = 1, BRG16 = 0											
	Fosc = 4.000 MHz			Fosc = 3.6864 MHz			Fosc = 2.000 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	103	1202	0.16	51
2400	2404	0.16	103	2400	0.00	95	2404	0.16	51	2404	0.16	25
9600	9615	0.16	25	9600	0.00	23	9615	0.16	12	—	—	—
10417	10417	0.00	23	10473	0.53	21	10417	0.00	11	10417	0.00	5
19.2k	19.23k	0.16	12	19.2k	0.00	11	—	—	—	—	—	—
57.6k	—	—	—	57.60k	0.00	3	—	—	—	—	—	—
115.2k	—	—	—	115.2k	0.00	1	—	—	—	—	—	—

BAUD RATE	SYNC = 0, BRGH = 0, BRG16 = 1											
	Fosc = 20.000 MHz			Fosc = 18.432 MHz			Fosc = 11.0592 MHz			Fosc = 8.000 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	-0.01	4166	300.0	0.00	3839	300.0	0.00	2303	299.9	-0.02	1666
1200	1200	-0.03	1041	1200	0.00	959	1200	0.00	575	1199	-0.08	416
2400	2399	-0.03	520	2400	0.00	479	2400	0.00	287	2404	0.16	207
9600	9615	0.16	129	9600	0.00	119	9600	0.00	71	9615	0.16	51
10417	10417	0.00	119	10378	-0.37	110	10473	0.53	65	10417	0.00	47
19.2k	19.23k	0.16	64	19.20k	0.00	59	19.20k	0.00	35	19.23k	0.16	25
57.6k	56.818	-1.36	21	57.60k	0.00	19	57.60k	0.00	11	55556	-3.55	8
115.2k	113.636	-1.36	10	115.2k	0.00	9	115.2k	0.00	5	—	—	—

BAUD RATE	SYNC = 0, BRGH = 0, BRG16 = 1											
	Fosc = 4.000 MHz			Fosc = 3.6864 MHz			Fosc = 2.000 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	300.1	0.04	832	300.0	0.00	767	299.8	-0.108	416	300.5	0.16	207
1200	1202	0.16	207	1200	0.00	191	1202	0.16	103	1202	0.16	51
2400	2404	0.16	103	2400	0.00	95	2404	0.16	51	2404	0.16	25
9600	9615	0.16	25	9600	0.00	23	9615	0.16	12	—	—	—
10417	10417	0.00	23	10473	0.53	21	10417	0.00	11	10417	0.00	5
19.2k	19.23k	0.16	12	19.20k	0.00	11	—	—	—	—	—	—
57.6k	—	—	—	57.60k	0.00	3	—	—	—	—	—	—
115.2k	—	—	—	115.2k	0.00	1	—	—	—	—	—	—

# PIC16F688

**TABLE 10-5: BAUD RATES FOR ASYNCHRONOUS MODES (CONTINUED)**

BAUD RATE	SYNC = 0, BRGH = 1, BRG16 = 1 or SYNC = 1, BRG16 = 1											
	Fosc = 20.000 MHz			Fosc = 18.432 MHz			Fosc = 11.0592 MHz			Fosc = 8.000 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	0.00	16665	300.0	0.00	15359	300.0	0.00	9215	300.0	0.00	6666
1200	1200	-0.01	4166	1200	0.00	3839	1200	0.00	2303	1200	-0.02	1666
2400	2400	0.02	2082	2400	0.00	1919	2400	0.00	1151	2401	0.04	832
9600	9597	-0.03	520	9600	0.00	479	9600	0.00	287	9615	0.16	207
10417	10417	0.00	479	10425	0.08	441	10433	0.16	264	10417	0	191
19.2k	19.23k	0.16	259	19.20k	0.00	239	19.20k	0.00	143	19.23k	0.16	103
57.6k	57.47k	-0.22	86	57.60k	0.00	79	57.60k	0.00	47	57.14k	-0.79	34
115.2k	116.3k	0.94	42	115.2k	0.00	39	115.2k	0.00	23	117.6k	2.12	16

BAUD RATE	SYNC = 0, BRGH = 1, BRG16 = 1 or SYNC = 1, BRG16 = 1											
	Fosc = 4.000 MHz			Fosc = 3.6864 MHz			Fosc = 2.000 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	0.01	3332	300.0	0.00	3071	299.9	-0.02	1666	300.1	0.04	832
1200	1200	0.04	832	1200	0.00	767	1199	-0.08	416	1202	0.16	207
2400	2398	0.08	416	2400	0.00	383	2404	0.16	207	2404	0.16	103
9600	9615	0.16	103	9600	0.00	95	9615	0.16	51	9615	0.16	25
10417	10417	0.00	95	10473	0.53	87	10417	0.00	47	10417	0.00	23
19.2k	19.23k	0.16	51	19.20k	0.00	47	19.23k	0.16	25	19.23k	0.16	12
57.6k	58.82k	2.12	16	57.60k	0.00	15	55.56k	-3.55	8	—	—	—
115.2k	111.1k	-3.55	8	115.2k	0.00	7	—	—	—	—	—	—

## 10.3.1 AUTO-BAUD DETECT

The EUSART module supports automatic detection and calibration of the baud rate.

In the Auto-Baud Detect (ABD) mode, the clock to the BRG is reversed. Rather than the BRG clocking the incoming RX signal, the RX signal is timing the BRG. The Baud Rate Generator is used to time the period of a received 55h (ASCII "U") which is the Sync character for the LIN bus. The unique feature of this character is that it has five rising edges including the Stop bit edge.

Setting the ABDEN bit of the BAUDCTL register starts the auto-boot sequence (Figure 10-6). While the ABD sequence takes place, the EUSART state machine is held in Idle. On the first rising edge of the receive line, after the Start bit, the SPBRG begins counting up using the BRG counter clock as shown in Table 10-6. The fifth rising edge will occur on the RX pin at the end of the eighth bit period. At that time, an accumulated value totaling the proper BRG period is left in SPBRGH, SPBRG register pair, the ABDEN bit is automatically cleared and the RCIF interrupt flag is set. The value in the RCREG needs to be read to clear the RCIF interrupt. RCREG content should be discarded. When calibrating for modes that do not use the SPBRGH register the user can verify that the SPBRG register did not overflow by checking for 00h in the SPBRGH register.

The BRG auto-baud clock is determined by the BRG16 and BRGH bits as shown in Table 10-6. During ABD, both the SPBRGH and SPBRG registers are used as a 16-bit counter, independent of the BRG16 bit setting. While calibrating the baud rate period, the SPBRGH

and SPBRG registers are clocked at 1/8th the BRG base clock rate. The resulting byte measurement is the average bit time when clocked at full speed.

**Note 1:** If the WUE bit is set with the ABDEN bit, auto-baud detection will occur on the byte following the Break character (see Section 10.3.2 "Auto-Wake-up on Break").

**2:** It is up to the user to determine that the incoming character baud rate is within the range of the selected BRG clock source. Some combinations of oscillator frequency and EUSART baud rates are not possible due to bit error rates. Overall system timing and communication baud rates must be taken into consideration when using the Auto-Baud Detect feature.

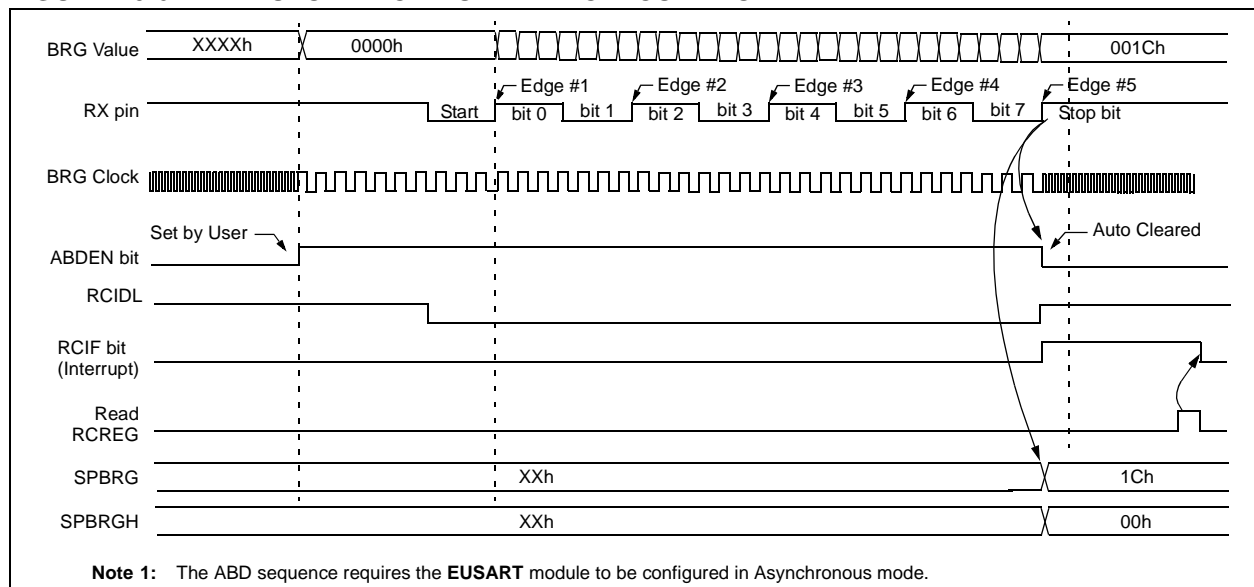
**3:** During the auto-baud process, the auto-baud counter starts counting at 1. Upon completion of the auto-baud sequence, to achieve maximum accuracy, subtract 1 from the SPBRGH:SPBRG register pair.

**TABLE 10-6: BRG COUNTER CLOCK RATES**

BRG16	BRGH	BRG Base Clock	BRG ABD Clock
0	0	Fosc/64	Fosc/512
0	1	Fosc/16	Fosc/128
1	0	Fosc/16	Fosc/128
1	1	Fosc/4	Fosc/32

**Note:** During the ABD sequence, SPBRG and SPBRGH registers are both used as a 16-bit counter, independent of BRG16 setting.

**FIGURE 10-6: AUTOMATIC BAUD RATE CALCULATION**



# PIC16F688

## 10.3.2 AUTO-WAKE-UP ON BREAK

During Sleep mode, all clocks to the EUSART are suspended. Because of this, the Baud Rate Generator is inactive and a proper character reception cannot be performed. The Auto-Wake-up feature allows the controller to wake-up due to activity on the RX/DT line. This feature is available only in Asynchronous mode.

The Auto-Wake-up feature is enabled by setting the WUE bit of the BAUDCTL register. Once set, the normal receive sequence on RX/DT is disabled, and the EUSART remains in an Idle state, monitoring for a wake-up event independent of the CPU mode. A wake-up event consists of a high-to-low transition on the RX/DT line. (This coincides with the start of a Sync Break or a wake-up signal character for the LIN protocol.)

The EUSART module generates an RCIF interrupt coincident with the wake-up event. The interrupt is generated synchronously to the Q clocks in normal CPU operating modes (Figure 10-7), and asynchronously if the device is in Sleep mode (Figure 10-8). The interrupt condition is cleared by reading the RCREG register.

The WUE bit is automatically cleared by the low-to-high transition on the RX line at the end of the Break. This signals to the user that the Break event is over. At this point, the EUSART module is in Idle mode waiting to receive the next character.

### 10.3.2.1 Special Considerations

#### Break Character

To avoid character errors or character fragments during a wake-up event, the wake-up character must be all zeros.

When the wake-up is enabled the function works independent of the low time on the data stream. If the WUE bit is set and a valid non-zero character is received, the low time from the Start bit to the first rising edge will be interpreted as the wake-up event. The remaining bits in the character will be received as a fragmented character and subsequent characters can result in framing or overrun errors.

Therefore, the initial character in the transmission must be all '0's. This must be 10 or more bit times, 13-bit times recommended for LIN bus, or any number of bit times for standard RS-232 devices.

#### Oscillator Start-up Time

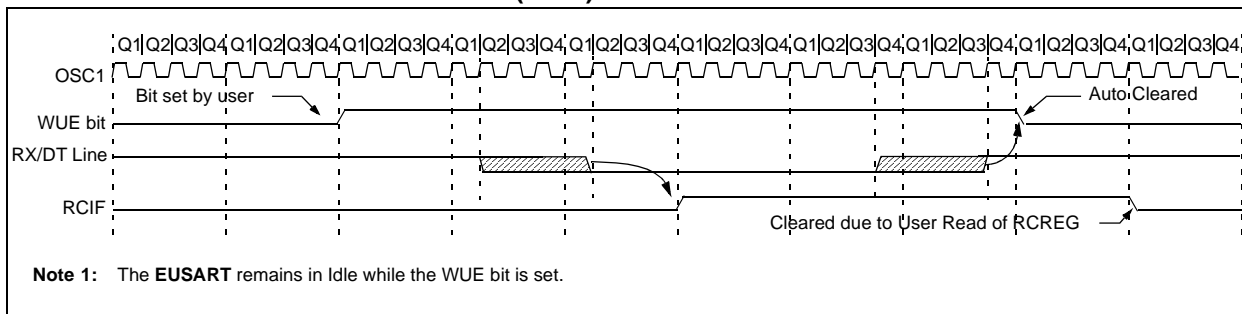
Oscillator start-up time must be considered, especially in applications using oscillators with longer start-up intervals (i.e., LP, XT or HS mode). The Sync Break (or wake-up signal) character must be of sufficient length, and be followed by a sufficient interval, to allow enough time for the selected oscillator to start and provide proper initialization of the EUSART.

#### WUE Bit

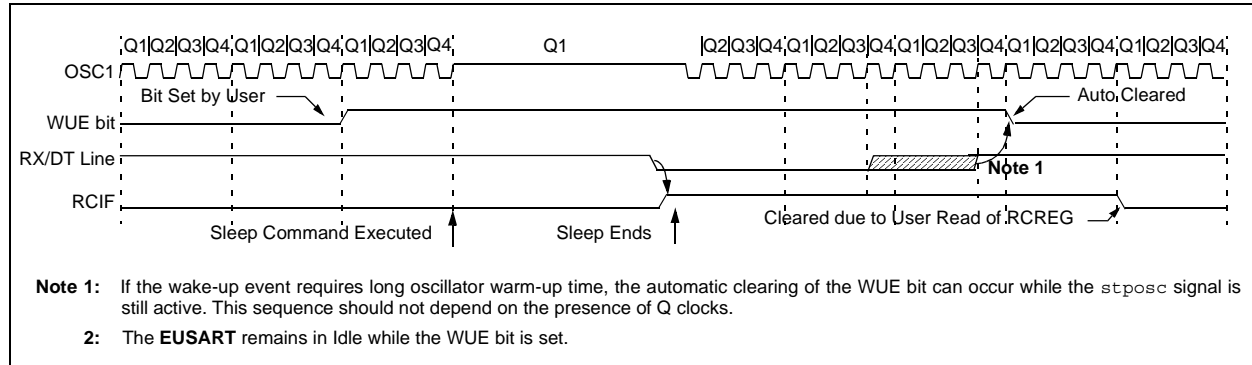
The wake-up event causes a receive interrupt by setting the RCIF bit. The WUE bit is cleared in hardware by a rising edge on RX/DT. The interrupt condition is then cleared in software by reading the RCREG register and discarding its contents.

To ensure that no actual data is lost, check the RCIDL bit to verify that a receive operation is not in process before setting the WUE bit. If a receive operation is not occurring, the WUE bit may then be set just prior to entering the Sleep mode.

**FIGURE 10-7: AUTO-WAKE-UP BIT (WUE) TIMING DURING NORMAL OPERATION**



**FIGURE 10-8: AUTO-WAKE-UP BIT (WUE) TIMINGS DURING SLEEP**



### 10.3.3 BREAK CHARACTER SEQUENCE

The EUSART module has the capability of sending the special Break character sequences that are required by the LIN bus standard. A Break character consists of a Start bit, followed by 12 '0' bits and a Stop bit.

To send a Break character, set the SENDB and TXEN bits of the TXSTA register. The Break character transmission is then initiated by a write to the TXREG. The value of data written to TXREG will be ignored and all '0's will be transmitted.

The SENDB bit is automatically reset by hardware after the corresponding Stop bit is sent. This allows the user to preload the transmit FIFO with the next transmit byte following the Break character (typically, the Sync character in the LIN specification).

The TRMT bit of the TXSTA register indicates when the transmit operation is active or Idle, just as it does during normal transmission. See Figure 10-9 for the timing of the Break character sequence.

#### 10.3.3.1 Break and Sync Transmit Sequence

The following sequence will start a message frame header made up of a Break, followed by an auto-baud Sync byte. This sequence is typical of a LIN bus master.

1. Configure the EUSART for the desired mode.
2. Set the TXEN and SENDB bits to enable the Break sequence.
3. Load the TXREG with a dummy character to initiate transmission (the value is ignored).
4. Write '55h' to TXREG to load the Sync character into the transmit FIFO buffer.
5. After the Break has been sent, the SENDB bit is reset by hardware and the Sync character is then transmitted.

When the TXREG becomes empty, as indicated by the TXIF, the next data byte can be written to TXREG.

### 10.3.4 RECEIVING A BREAK CHARACTER

The Enhanced EUSART module can receive a Break character in two ways.

The first method to detect a Break character uses the FERR bit of the RCSTA register and the Received data as indicated by RCREG. The Baud Rate Generator is assumed to have been initialized to the expected baud rate.

A Break character has been received when;

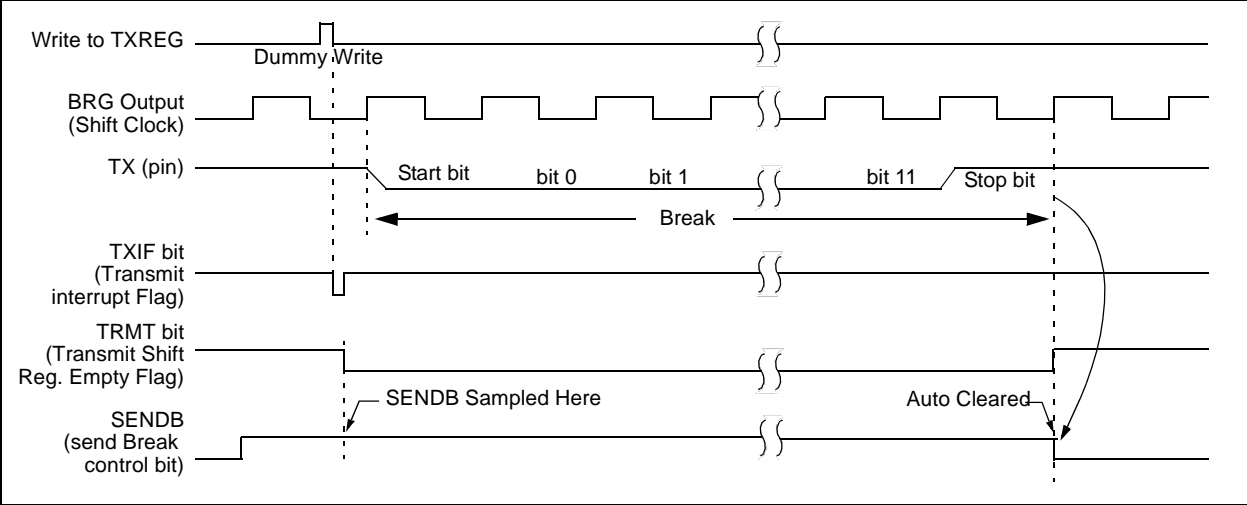
- RCIF bit is set
- FERR bit is set
- RCREG = 00h

The second method uses the Auto-Wake-up feature described in **Section 10.3.2 "Auto-Wake-up on Break"**. By enabling this feature, the EUSART will sample the next two transitions on RX/DT, cause an RCIF interrupt, and receive the next data byte followed by another interrupt.

Note that following a Break character, the user will typically want to enable the Auto-Baud Detect feature. For both methods, the user can set the ABDEN bit of the BAUDCTL register before placing the EUSART in Sleep mode.

# PIC16F688

**FIGURE 10-9: SEND BREAK CHARACTER SEQUENCE**





## 10.4 EUSART 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 EUSART can operate as either a master or slave device.

Start and Stop bits are not used in synchronous transmissions.

### 10.4.1 SYNCHRONOUS MASTER MODE

The following bits are used to configure the EUSART 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 EUSART. If the RX/DT or TX/CK pins are shared with an analog peripheral the analog I/O functions must be disabled by clearing the corresponding ANSEL bits.

#### 10.4.1.1 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 is automatically configured as an output when the EUSART is configured for synchronous transmit 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.

#### 10.4.1.2 Clock Polarity

A clock polarity option is provided for Microwire compatibility. Clock polarity is selected with the SCKP bit of the BAUDCTL register. Setting the SCKP bit sets

the clock Idle state as high. When the SCKP bit is set, the data changes on the falling edge of each clock. Clearing the SCKP bit sets the Idle state as low. When the SCKP bit is cleared, the data changes on the rising edge of each clock.

#### 10.4.1.3 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 EUSART 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.

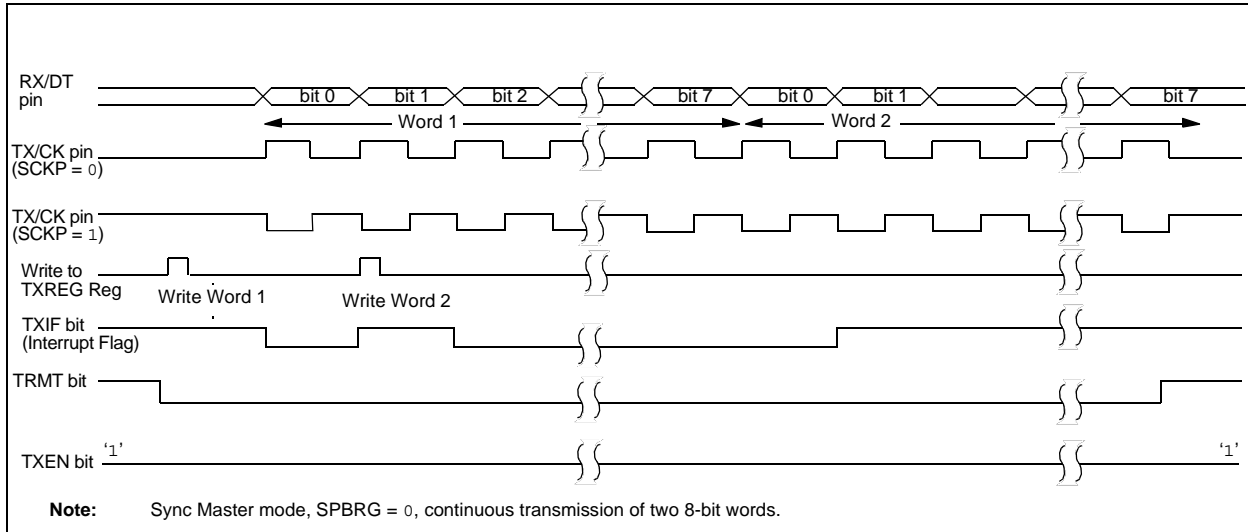
**Note:** The TSR register is not mapped in data memory, so it is not available to the user.

#### 10.4.1.4 Synchronous Master Transmission Set-up:

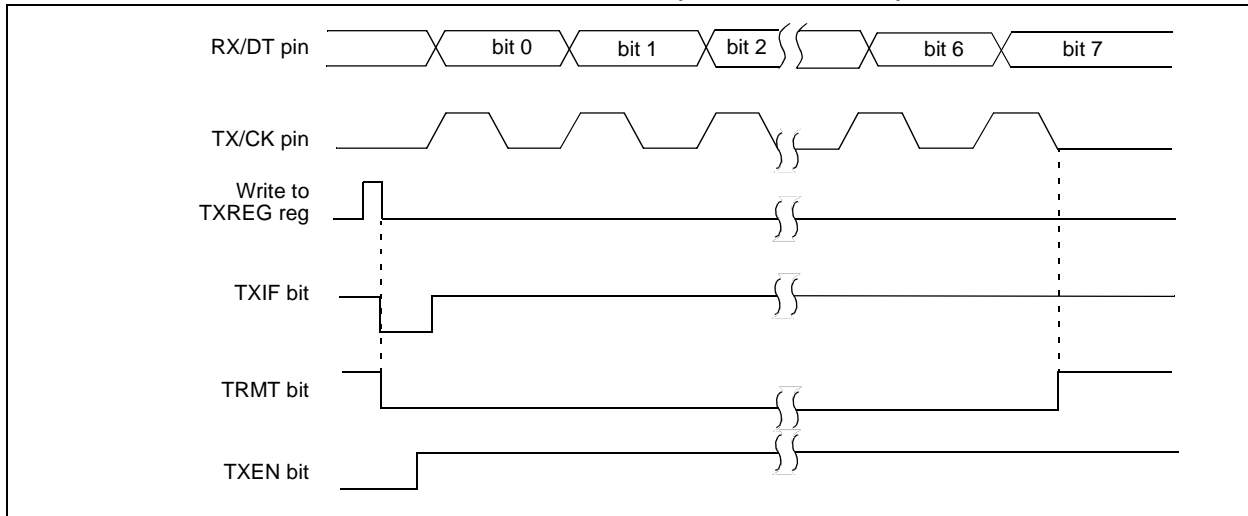
1. Initialize the SPBRGH, SPBRG register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see **Section 10.3 “EUSART Baud Rate Generator (BRG)”**).
2. Enable the synchronous master serial port by setting bits SYNC, SPEN and CSRC.
3. Disable Receive mode by clearing bits SREN and CREN.
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, GIE and PEIE interrupt enable bits.
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.

# PIC16F688

**FIGURE 10-10: SYNCHRONOUS TRANSMISSION**



**FIGURE 10-11: SYNCHRONOUS TRANSMISSION (THROUGH TXEN)**



**TABLE 10-7: 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
BAUDCTL	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	01-0 0-00	01-0 0-00
INTCON	GIE	PEIE	T0IE	INTE	RAIE	T0IF	INTF	RAIF	0000 000x	0000 000x
PIE1	EEIE	ADIE	RCIE	C2IE	C1IE	OSFIE	TXIE	TMR1IE	0000 0000	0000 0000
PIR1	EEIF	ADIF	RCIF	C2IF	C1IF	OSFIF	TXIF	TMR1IF	0000 0000	0000 0000
RCREG	EUSART 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
SPBRGH	BRG15	BRG14	BRG13	BRG12	BRG11	BRG10	BRG9	BRG8	0000 0000	0000 0000
TRISC	—	—	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	--11 1111	--11 1111
TXREG	EUSART Transmit Data Register								0000 0000	0000 0000
TXSTA	CSRC	TX9	TXEN	SYNC	SENB	BRGH	TRMT	TX9D	0000 0010	0000 0010

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

## 10.4.1.5 Synchronous Master Reception

Data is received at the RX/DT pin. The RX/DT and TX/CK pin output drivers are automatically disabled when the EUSART 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 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.

## 10.4.1.6 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 or by clearing the SPEN bit which resets the EUSART.

## 10.4.1.7 Receiving 9-bit Characters

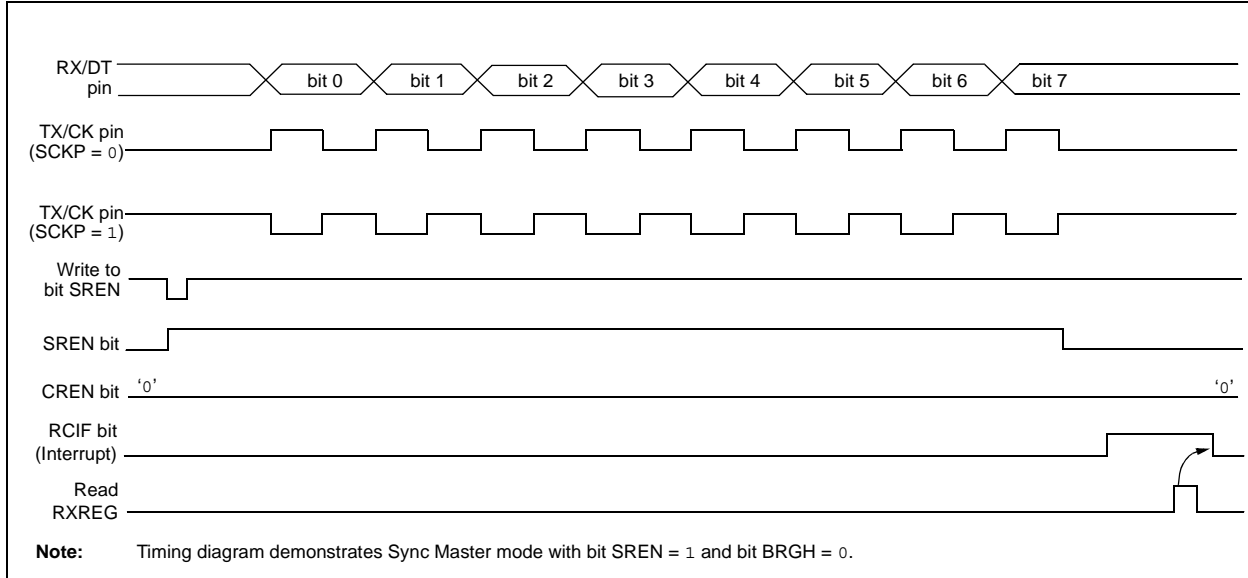
The EUSART supports 9-bit character reception. When the RX9 bit of the RCSTA register is set the EUSART will shift 9-bits into the RSR for each character received. The RX9D bit of the RCSTA register is the ninth, and Most Significant, data bit of the top unread character in the receive FIFO. When reading 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.

## 10.4.1.8 Synchronous Master Reception Setup:

1. Initialize the SPBRGH, SPBRG register pair for the appropriate baud rate. Set or clear the BRGH and BRG16 bits, 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 using interrupts, set the GIE and PEIE bits of the INTCON register and set RCIE.
5. If 9-bit reception is desired, set bit RX9.
6. Start reception by setting the SREN bit or for continuous reception, set the CREN bit.
7. Interrupt flag bit RCIF will be set when reception of a character is complete. An interrupt will be generated if the enable bit RCIE was set.
8. Read the RCSTA register to get the ninth bit (if enabled) and determine if any error occurred during reception.
9. Read the 8-bit received data by reading the RCREG register.
10. 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 EUSART.

# PIC16F688

**FIGURE 10-12: SYNCHRONOUS RECEPTION (MASTER MODE, SREN)**



**TABLE 10-8: 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
BAUDCTL	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	01-0 0-00	01-0 0-00
INTCON	GIE	PEIE	T0IE	INTE	RAIE	T0IF	INTF	RAIF	0000 000x	0000 000x
PIE1	EEIE	ADIE	RCIE	C2IE	C1IE	OSFIE	TXIE	TMR1IE	0000 0000	0000 0000
PIR1	EEIF	ADIF	RCIF	C2IF	C1IF	OSFIF	TXIF	TMR1IF	0000 0000	0000 0000
RCREG	EUSART 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
SPBRGH	BRG15	BRG14	BRG13	BRG12	BRG11	BRG10	BRG9	BRG8	0000 0000	0000 0000
TRISC	—	—	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	--11 1111	--11 1111
TXREG	EUSART Transmit Data Register								0000 0000	0000 0000
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010	0000 0010

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

## 10.4.2 SYNCHRONOUS SLAVE MODE

The following bits are used to configure the EUSART 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 EUSART. If the RX/DT or TX/CK pins are shared with an analog peripheral the analog I/O functions must be disabled by clearing the corresponding ANSEL bits.

### 10.4.2.1 EUSART Synchronous Slave Transmit

The operation of the Synchronous Master and Slave modes are identical (see **Section 10.4.1.3 “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.

### 10.4.2.2 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. If 9-bit transmission is selected, insert the Most Significant bit into the TX9D bit.
7. Start transmission by writing the Least Significant 8 bits to the TXREG register.

**TABLE 10-9: 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
BAUDCTL	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	01-0 0-00	01-0 0-00
INTCON	GIE	PEIE	T0IE	INTE	RAIE	T0IF	INTF	RAIF	0000 000x	0000 000x
PIE1	EEIE	ADIE	RCIE	C2IE	C1IE	OSFIE	TXIE	TMR1IE	0000 0000	0000 0000
PIR1	EEIF	ADIF	RCIF	C2IF	C1IF	OSFIF	TXIF	TMR1IF	0000 0000	0000 0000
RCREG	EUSART 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
SPBRGH	BRG15	BRG14	BRG13	BRG12	BRG11	BRG10	BRG9	BRG8	0000 0000	0000 0000
TRISC	—	—	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	--11 1111	--11 1111
TXREG	EUSART Transmit Data Register								0000 0000	0000 0000
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010	0000 0010

**Legend:** x = unknown, — = unimplemented read as '0'. Shaded cells are not used for Synchronous Slave Transmission.

# PIC16F688

## 10.4.2.3 EUSART Synchronous Slave Reception

The operation of the Synchronous Master and Slave modes is identical (**Section 10.4.1.5 “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 enable bit 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.

## 10.4.2.4 Synchronous Slave Reception Set-up:

1. Set the SYNC and SPEN bits and clear the CSRC bit.
2. If using interrupts, ensure that the GIE and PEIE bits of the INTCON register are set and set the RCIE bit.
3. If 9-bit reception is desired, set the RX9 bit.
4. Set the CREN bit to enable reception.
5. The RCIF bit will be set when reception is complete. An interrupt will be generated if the RCIE bit was set.
6. If 9-bit mode is enabled, retrieve the Most Significant bit from the RX9D bit of the RCSTA register.
7. Retrieve the 8 Least Significant bits from the receive FIFO by reading the RCREG register.
8. 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 EUSART.

**TABLE 10-10: 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
BAUDCTL	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	01-0 0-00	01-0 0-00
INTCON	GIE	PEIE	T0IE	INTE	RAIE	T0IF	INTF	RAIF	0000 000x	0000 000x
PIE1	EEIE	ADIE	RCIE	C2IE	C1IE	OSFIE	TXIE	TMR1IE	0000 0000	0000 0000
PIR1	EEIF	ADIF	RCIF	C2IF	C1IF	OSFIF	TXIF	TMR1IF	0000 0000	0000 0000
RCREG	EUSART 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
SPBRGH	BRG15	BRG14	BRG13	BRG12	BRG11	BRG10	BRG9	BRG8	0000 0000	0000 0000
TRISC	—	—	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	--11 1111	--11 1111
TXREG	EUSART Transmit Data Register								0000 0000	0000 0000
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010	0000 0010

**Legend:** x = unknown, – = unimplemented read as ‘0’. Shaded cells are not used for Synchronous Slave Reception.

## 11.0 SPECIAL FEATURES OF THE CPU

The PIC16F688 has a host of features intended to maximize system reliability, minimize cost through elimination of external components, provide power-saving features and offer code protection.

These features are:

- Reset
  - Power-on Reset (POR)
  - Power-up Timer (PWRT)
  - Oscillator Start-up Timer (OST)
  - Brown-out Reset (BOR)
- Interrupts
- Watchdog Timer (WDT)
- Oscillator Selection
- Sleep
- Code Protection
- ID Locations
- In-Circuit Serial Programming™

The PIC16F688 has two timers that offer necessary delays on power-up. One is the Oscillator Start-up Timer (OST), intended to keep the chip in Reset until the crystal oscillator is stable. The other is the Power-up Timer (PWRT), which provides a fixed delay of 64 ms (nominal) on power-up only, designed to keep the part in Reset while the power supply stabilizes. There is also circuitry to reset the device if a brown-out occurs, which can use the Power-up Timer to provide at least a 64 ms Reset. With these three functions-on-chip, most applications need no external Reset circuitry.

The Sleep mode is designed to offer a very low-current Power-Down mode. The user can wake-up from Sleep through:

- External Reset
- Watchdog Timer Wake-up
- An interrupt

Several oscillator options are also made available to allow the part to fit the application. The INTOSC option saves system cost while the LP crystal option saves power. A set of Configuration bits are used to select various options (see Register 11-1).

# PIC16F688

---

## 11.1 Configuration Bits

The Configuration bits can be programmed (read as '0'), or left unprogrammed (read as '1') to select various device configurations as shown in Register 11-1. These bits are mapped in program memory location 2007h.

**Note:** Address 2007h is beyond the user program memory space. It belongs to the special configuration memory space (2000h-3FFFh), which can be accessed only during programming. See "PIC12F6XX/16F6XX *Memory Programming Specification*" (DS41204) for more information.



## REGISTER 11-1: CONFIG: CONFIGURATION WORD REGISTER

Reserved	Reserved	Reserved	Reserved	FCMEN	IESO	BOREN <sup>(1)</sup>	BOREN0 <sup>(1)</sup>
bit 15				bit 8			

$\overline{\text{CPD}}^{\text{(2)}}$	$\overline{\text{CP}}^{\text{(3)}}$	MCLRE <sup>(4)</sup>	$\overline{\text{PWRTE}}$	WDTE	FOSC2	FOSC1	FOSC0
bit 7				bit 0			

### Legend:

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

- bit 15-12      **Reserved:** Reserved bits. Do Not Use.
- bit 11      **FCMEN:** Fail-Safe Clock Monitor Enabled bit  
 1 = Fail-Safe Clock Monitor is enabled  
 0 = Fail-Safe Clock Monitor is disabled
- bit 10      **IESO:** Internal External Switchover bit  
 1 = Internal External Switchover mode is enabled  
 0 = Internal External Switchover mode is disabled
- bit 9-8      **BOREN<1:0>:** Brown-out Reset Selection bits<sup>(1)</sup>  
 11 = BOR enabled  
 10 = BOR enabled during operation and disabled in Sleep  
 01 = BOR controlled by SBOREN bit of the PCON register  
 00 = BOR disabled
- bit 7      **CPD:** Data Code Protection bit<sup>(2)</sup>  
 1 = Data memory code protection is disabled  
 0 = Data memory code protection is enabled
- bit 6      **CP:** Code Protection bit<sup>(2)</sup>  
 1 = Program memory code protection is disabled  
 0 = Program memory code protection is enabled
- bit 5      **MCLRE:** MCLR Pin Function Select bit<sup>(3)</sup>  
 1 = MCLR pin function is MCLR  
 0 = MCLR pin function is digital input, MCLR internally tied to V<sub>DD</sub>
- bit 4      **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
- bit 2-0      **FOSC<2:0>:** Oscillator Selection bits  
 111 = EXTRC oscillator: External RC on RA5/OSC1/CLKIN, CLKOUT function on RA4/OSC2/CLKOUT pin  
 110 = EXTRCIO oscillator: External RC on RA5/OSC1/CLKIN, I/O function on RA4/OSC2/CLKOUT pin  
 101 = INTOSC oscillator: CLKOUT function on RA4/OSC2/CLKOUT pin, I/O function on RA5/OSC1/CLKIN  
 100 = INTOSCIO oscillator: I/O function on RA4/OSC2/CLKOUT pin, I/O function on RA5/OSC1/CLKIN  
 011 = EC: I/O function on RA4/OSC2/CLKOUT pin, CLKIN on RA5/OSC1/CLKIN  
 010 = HS oscillator: High-speed crystal/resonator on RA4/OSC2/CLKOUT and RA5/OSC1/CLKIN  
 001 = XT oscillator: Crystal/resonator on RA4/OSC2/CLKOUT and RA5/OSC1/CLKIN  
 000 = LP oscillator: Low-power crystal on RA4/OSC2/CLKOUT and RA5/OSC1/CLKIN

- Note** 1: Enabling Brown-out Reset does not automatically enable Power-up Timer.  
 2: The entire data EEPROM will be erased when the code protection is turned off.  
 3: The entire program memory will be erased when the code protection is turned off.  
 4: When MCLR is asserted in INTOSC or RC mode, the internal clock oscillator is disabled.

# PIC16F688

## 11.2 Reset

The PIC16F688 differentiates between various kinds of Reset:

- Power-on Reset (POR)
- WDT Reset during normal operation
- WDT Reset during Sleep
- $\overline{\text{MCLR}}$  Reset during normal operation
- $\overline{\text{MCLR}}$  Reset during Sleep
- 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:

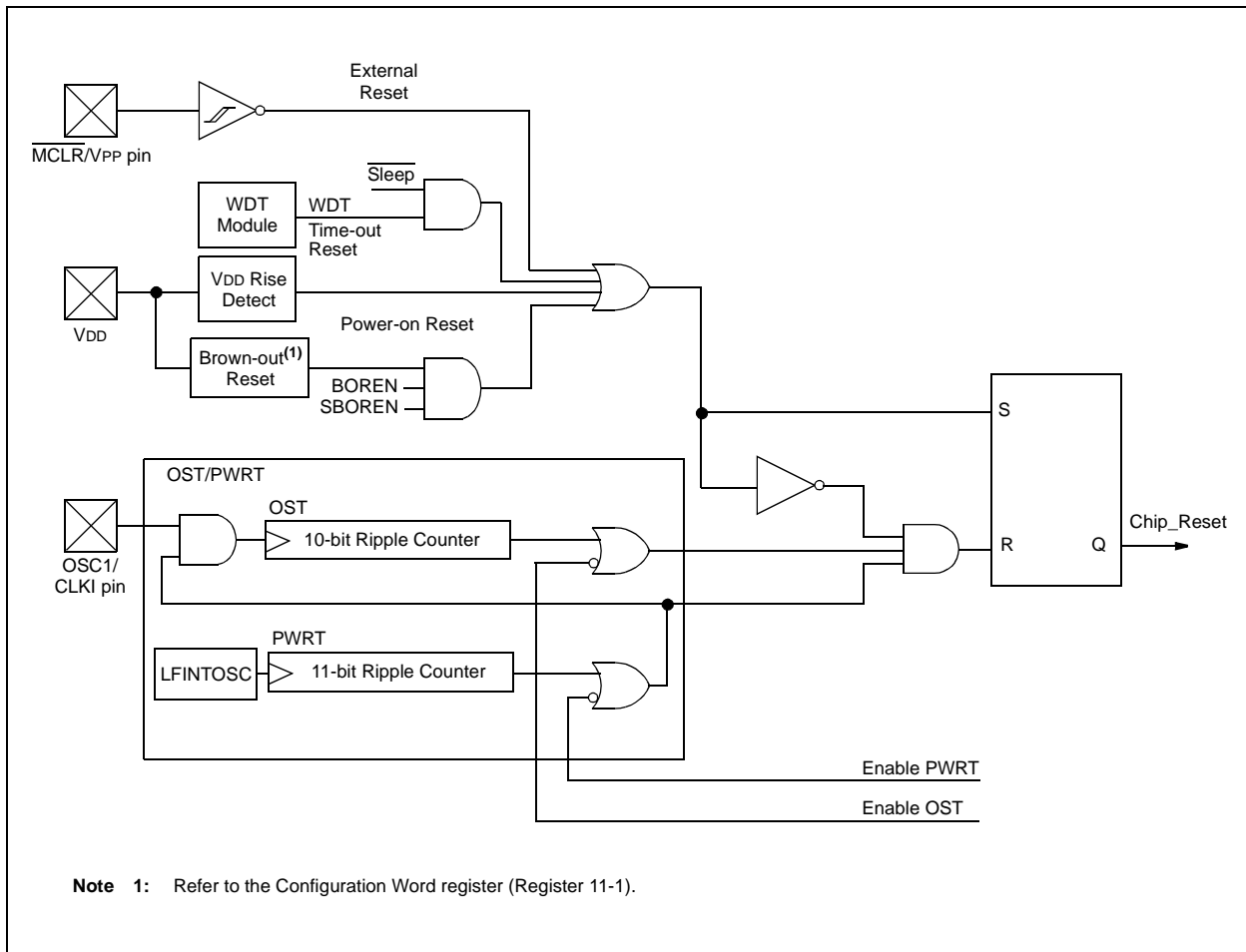
- Power-on Reset
- $\overline{\text{MCLR}}$  Reset
- $\overline{\text{MCLR}}$  Reset during Sleep
- WDT Reset
- Brown-out Reset (BOR)

They 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 11-2. These bits are used in software to determine the nature of the Reset. See Table 11-4 for a full description of Reset states of all registers.

A simplified block diagram of the On-Chip Reset Circuit is shown in Figure 11-1.

The  $\overline{\text{MCLR}}$  Reset path has a noise filter to detect and ignore small pulses. See **Section 14.0 "Electrical Specifications"** for pulse width specifications.

**FIGURE 11-1: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT**



## 11.2.1 POWER-ON RESET

The on-chip POR circuit holds the chip in Reset until VDD has reached a high enough level for proper operation. To take advantage of the POR, simply connect the  $\overline{\text{MCLR}}$  pin through a resistor to VDD. This will eliminate external RC components usually needed to create Power-on Reset. A maximum rise time for VDD is required. See **Section 14.0 “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 VBOD (see **Section 11.2.4 “Brown-Out Reset (BOR)”**).

**Note:** The POR circuit does not produce an internal Reset when VDD declines. To re-enable the POR, VDD must reach VSS for a minimum of 100  $\mu\text{s}$ .

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).

## 11.2.2 $\overline{\text{MCLR}}$

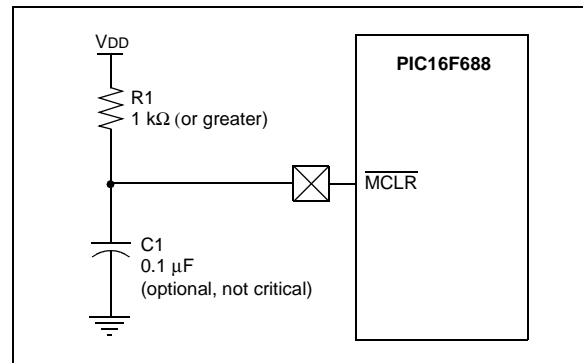
PIC16F688 has a noise filter in the  $\overline{\text{MCLR}}$  Reset path. The filter will detect and ignore small pulses.

It should be noted that a WDT Reset does not drive  $\overline{\text{MCLR}}$  pin low.

The behavior of the ESD protection on the  $\overline{\text{MCLR}}$  pin has been altered from early devices of this family. Voltages applied to the pin that exceed its specification can result in both  $\overline{\text{MCLR}}$  Resets and excessive current beyond the device specification during the ESD event. For this reason, Microchip recommends that the  $\overline{\text{MCLR}}$  pin no longer be tied directly to VDD. The use of an RC network, as shown in Figure 11-2, is suggested.

An internal  $\overline{\text{MCLR}}$  option is enabled by clearing the MCLRE bit in the Configuration Word register. When  $\overline{\text{MCLRE}} = 0$ , the Reset signal to the chip is generated internally. When the  $\overline{\text{MCLRE}} = 1$ , the RA3/ $\overline{\text{MCLR}}$  pin becomes an external Reset input. In this mode, the RA3/ $\overline{\text{MCLR}}$  pin has a weak pull-up to VDD.

**FIGURE 11-2: RECOMMENDED  $\overline{\text{MCLR}}$  CIRCUIT**



## 11.2.3 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 31 kHz LFINTOSC oscillator. For more information, see **Section 3.5 “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,  $\overline{\text{PWRTE}}$ , 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 14.0 “Electrical Specifications”**).

**Note:** Voltage spikes below VSS at the  $\overline{\text{MCLR}}$  pin, inducing currents greater than 80 mA, may cause latch-up. Thus, a series resistor of 50-100  $\Omega$  should be used when applying a “low” level to the  $\overline{\text{MCLR}}$  pin, rather than pulling this pin directly to VSS.

# PIC16F688

## 11.2.4 BROWN-OUT RESET (BOR)

The BOREN0 and BOREN1 bits in the Configuration Word register selects one of four BOR modes. Two modes have been added to allow software or hardware control of the BOR enable. When BOREN<1:0> = 01, the SBOREN bit of the PCON register enables/disables the BOR, allowing it to be controlled in software. By selecting BOREN<1:0>, the BOR is automatically disabled in Sleep to conserve power and enabled on wake-up. In this mode, the SBOREN bit is disabled. See Register 11-1 for the Configuration Word definition.

If VDD falls below VBOD for greater than parameter (TBOD) (see **Section 14.0 “Electrical Specifications”**), the Brown-out situation will reset the device.

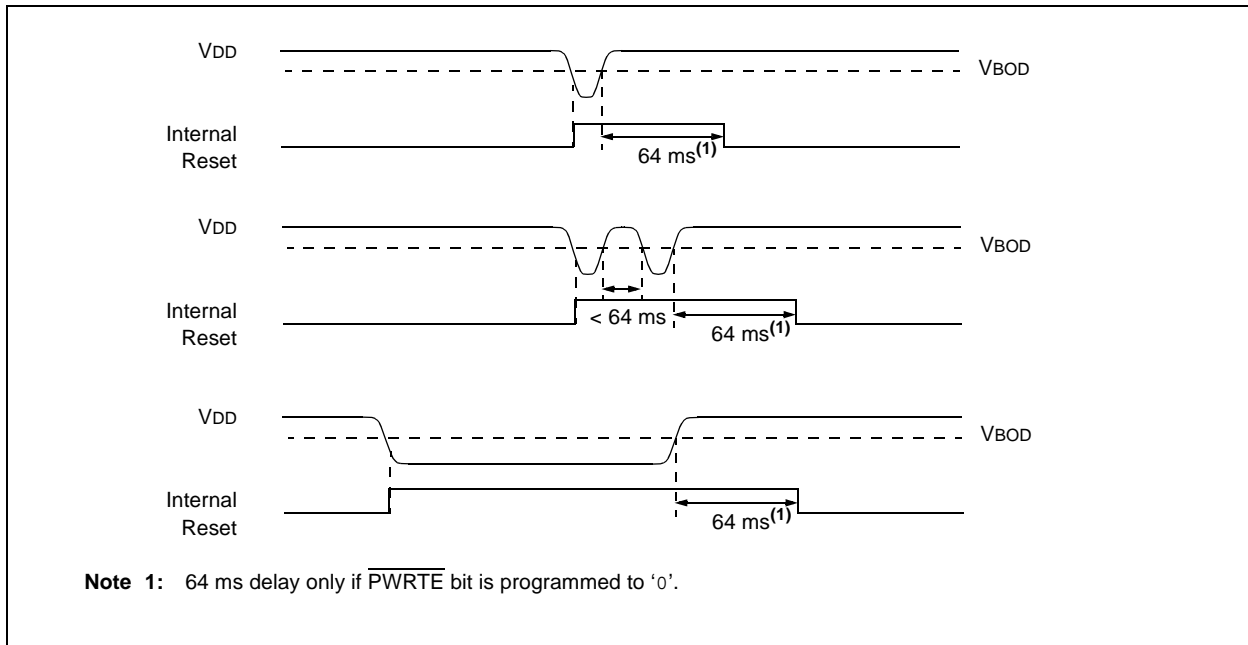
This will occur regardless of VDD slew rate. A Reset is not insured to occur if VDD falls below VBOD for less than parameter (TBOD).

On any Reset (Power-on, Brown-out Reset, Watchdog Timer, etc.), the chip will remain in Reset until VDD rises above VBOD (see Figure 11-3). The Power-up Timer will now be invoked, if enabled and will keep the chip in Reset an additional 64 ms.

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

If VDD drops below VBOD 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 VBOD, the Power-up Timer will execute a 64 ms Reset.

**FIGURE 11-3: BROWN-OUT SITUATIONS**



## 11.2.5 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  $\overline{\text{PWRT}}\text{E}$  bit status. For example, in EC mode with  $\overline{\text{PWRT}}\text{E}$  bit erased (PWRT disabled), there will be no time-out at all. Figure 11.2.1, Figure 11-5 and Figure 11-6 depict time-out sequences. The device can execute code from the INTOSC while OST is active by enabling Two-Speed Start-up or Fail-Safe Monitor (see Section 3.7.2 “Two-Speed Start-up Sequence” and Section 3.8 “Fail-Safe Clock Monitor”).

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 11-5). This is useful for testing purposes or to synchronize more than one PIC16F688 device operating in parallel.

Table 11-5 shows the Reset conditions for some special registers, while Table 11-4 shows the Reset conditions for all the registers.

## 11.2.6 POWER CONTROL (PCON) REGISTER

The Power Control (PCON) register (address 8Eh) has two Status bits to indicate what type of Reset that last occurred.

Bit 0 is  $\overline{\text{BOR}}$  (Brown-out).  $\overline{\text{BOR}}$  is unknown on Power-on Reset. It must then be set by the user and checked on subsequent Resets to see if  $\text{BOR} = 0$ , 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 ( $\text{BOREN}\langle 1:0 \rangle = 00$  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., VDD may have gone too low).

For more information, see Section 4.2.4 “Ultra Low-Power Wake-up” and Section 11.2.4 “Brown-Out Reset (BOR)”.

**TABLE 11-1: TIME-OUT IN VARIOUS SITUATIONS**

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

**TABLE 11-2: PCON 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

**TABLE 11-3: SUMMARY OF REGISTERS ASSOCIATED WITH BROWN-OUT RESET**

Name	Bit 9	Bit 8	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 <sup>(1)</sup>
CONFIG <sup>(2)</sup>	BOREN1	BOREN0	$\overline{\text{CPD}}$	$\overline{\text{CP}}$	MCLRRE	$\overline{\text{PWRT}}\text{E}$	WDTE	FOSC2	FOSC1	FOSC0	—	—
PCON			—	—	ULPWUE	SBOREN	—	—	$\overline{\text{POR}}$	$\overline{\text{BOR}}$	--01 --qq	--0u --uu
STATUS			IRP	RP1	RP0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	0001 1xxx	000q quuu

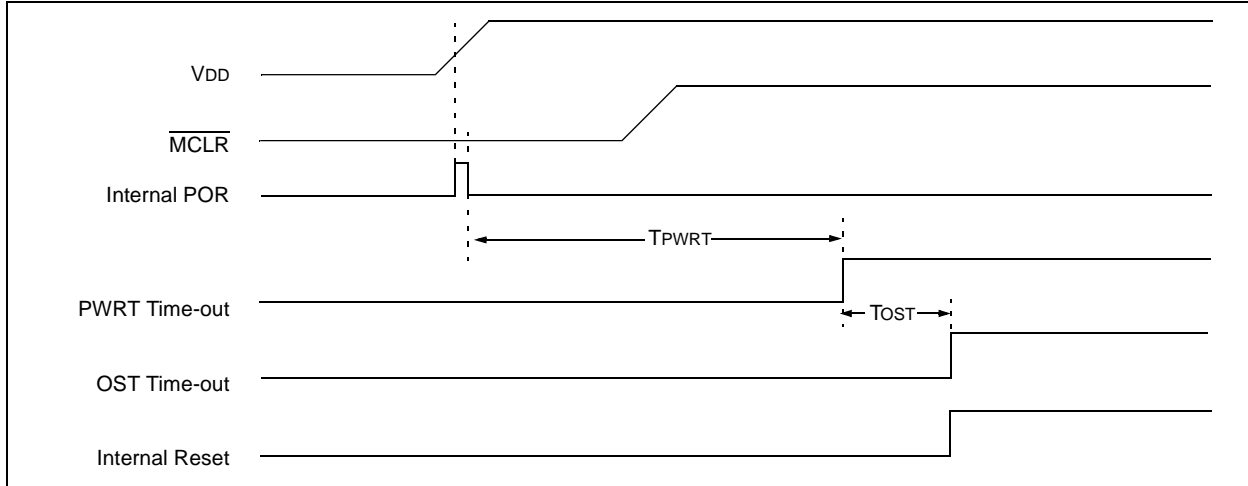
**Legend:** u = unchanged, x = unknown, — = unimplemented bit, reads as ‘0’, q = value depends on condition. Shaded cells are not used by BOR.

**Note 1:** Other (non Power-up) Resets include MCLR Reset and Watchdog Timer Reset during normal operation.

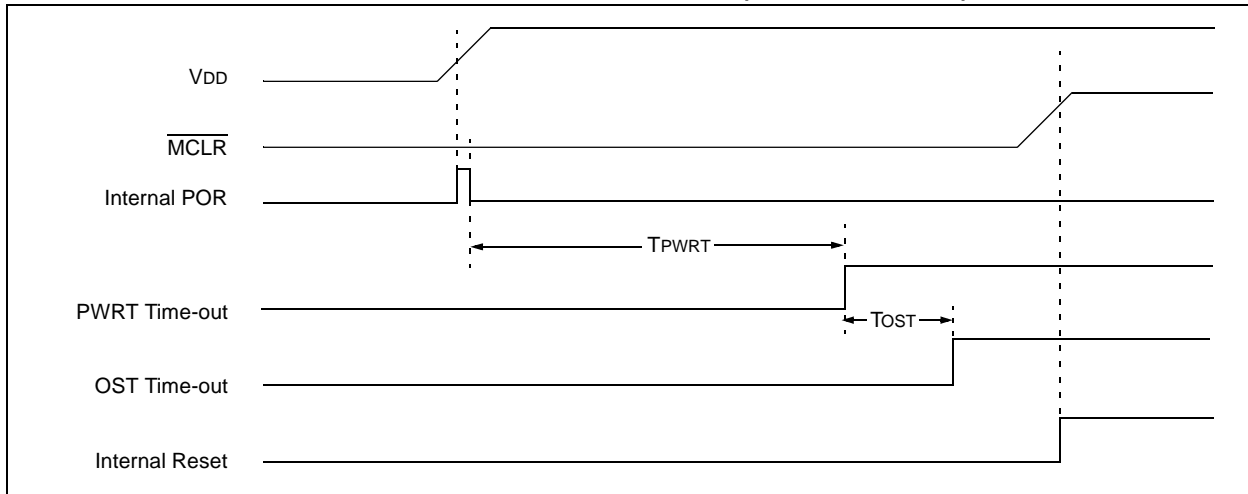
**Note 2:** See Configuration Word register (Register 11-1) for operation of all register bits.

# PIC16F688

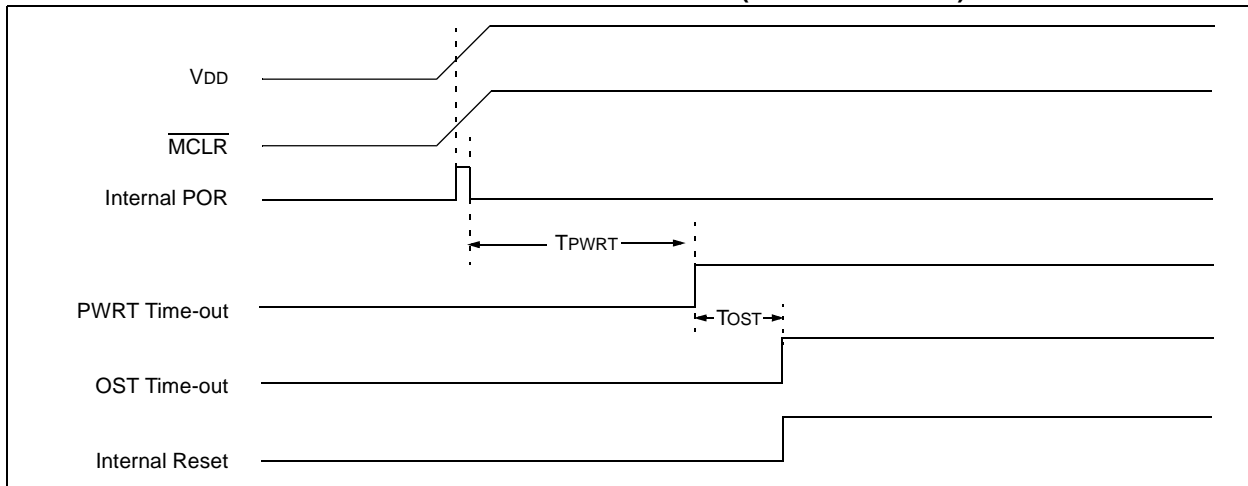
**FIGURE 11-4: TIME-OUT SEQUENCE ON POWER-UP (DELAYED  $\overline{\text{MCLR}}$ )**



**FIGURE 11-5: TIME-OUT SEQUENCE ON POWER-UP (DELAYED  $\overline{\text{MCLR}}$ )**



**FIGURE 11-6: TIME-OUT SEQUENCE ON POWER-UP ( $\overline{\text{MCLR}}$  WITH VDD)**



**TABLE 11-4: INITIALIZATION CONDITION FOR REGISTERS**

Register	Address	Power-on Reset	MCLR Reset WDT Reset Brown-out Reset <sup>(1)</sup>	Wake-up from Sleep through Interrupt Wake-up from Sleep through WDT Time-out
W	—	xxxx xxxx	uuuu uuuu	uuuu uuuu
INDF	00h/80h/100h/180h	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR0	01h/101h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCL	02h/82h/102h/182h	0000 0000	0000 0000	PC + 1 <sup>(3)</sup>
STATUS	03h/83h/103h/183h	0001 1xxx	000q quuu <sup>(4)</sup>	uuuq quuu <sup>(4)</sup>
FSR	04h/84h/104h/184h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTA	05h/105h	--x0 x000	--00 0000	--uu uuuu
PORTC	07h/107h	--xx 0000	--00 0000	--uu uuuu
PCLATH	0Ah/8Ah/10Ah/18Ah	---0 0000	---0 0000	---u uuuu
INTCON	0Bh/8Bh/10Bh/18Bh	0000 000x	0000 000x	uuuu uuuu <sup>(2)</sup>
PIR1	0Ch	0000 0000	0000 0000	uuuu uuuu <sup>(2)</sup>
TMR1L	0Eh	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR1H	0Fh	xxxx xxxx	uuuu uuuu	uuuu uuuu
T1CON	10h	0000 0000	uuuu uuuu	-uuu uuuu
BAUDCTL	11h	01-0 0-00	01-0 0-00	uu-u u-uu
SPBRGH	12h	-000 0000	-000 0000	-uuu uuuu
SPBRG	13h	0000 0000	0000 0000	uuuu uuuu
RCREG	14h	0000 0000	0000 0000	uuuu uuuu
TXREG	15h	0000 0000	0000 0000	uuuu uuuu
TXSTA	16h	0000 0010	0000 0010	uuuu uuuu
RCSTA	17h	000x 000x	000x 000x	uuuu uuuu
WDTCON	18h	---0 1000	---0 1000	---u uuuu
CMCON0	19h	0000 0000	0000 0000	uuuu uuuu
CMCON1	1Ah	---- --10	---- --10	---- --uu
ADRESH	1Eh	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADCON0	1Fh	00-0 0000	00-0 0000	uu-u uuuu
OPTION_REG	81h/181h	1111 1111	1111 1111	uuuu uuuu
TRISA	85h/185h	--11 1111	--11 1111	--uu uuuu
TRISC	87h/187h	--11 1111	--11 1111	--uu uuuu
PIE1	8Ch	0000 0000	0000 0000	uuuu uuuu
PCON	8Eh	--01 --0x	--0u --uu <sup>(1,5)</sup>	--uu --uu

**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 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 11-5 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.

# PIC16F688

**TABLE 11-4: INITIALIZATION CONDITION FOR REGISTERS (CONTINUED)**

Register	Address	Power-on Reset	<ul style="list-style-type: none"> <li>• MCLR Reset</li> <li>• WDT Reset</li> <li>• Brown-out Reset<sup>(1)</sup></li> </ul>	<ul style="list-style-type: none"> <li>• Wake-up from Sleep through interrupt</li> <li>• Wake-up from Sleep through WDT time-out</li> </ul>
OSCCON	8Fh	-110 q000	-110 q000	-uuu uuuu
OSCTUNE	90h	---0 0000	---u uuuu	---u uuuu
ANSEL	91h	1111 1111	1111 1111	uuuu uuuu
WPUA	95h	--11 -111	--11 -111	uuuu uuuu
IOCA	96h	--00 0000	--00 0000	--uu uuuu
EEDATH	97h	--00 0000	--00 0000	--uu uuuu
EEADRH	98h	---- 0000	---- 0000	---- uuuu
VRCON	99h	0-0- 0000	0-0- 0000	u-u- uuuu
EEDAT	9Ah	0000 0000	0000 0000	uuuu uuuu
EEADR	9Bh	0000 0000	0000 0000	uuuu uuuu
EECON1	9Ch	x--- x000	u--- q000	u--- uuuu
EECON2	9Dh	---- ----	---- ----	---- ----
ADRESL	9Eh	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADCON1	9Fh	-000 ----	-000 ----	-uuu ----

**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 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 11-5 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 11-5: INITIALIZATION CONDITION FOR SPECIAL REGISTERS**

Condition	Program Counter	Status Register	PCON Register
Power-on Reset	000h	0001 1xxx	--01 --0x
MCLR Reset during normal operation	000h	000u uuuu	--0u --uu
MCLR Reset during Sleep	000h	0001 0uuu	--0u --uu
WDT Reset	000h	0000 uuuu	--0u --uu
WDT Wake-up	PC + 1	uuu0 0uuu	--uu --uu
Brown-out Reset	000h	0001 1uuu	--01 --10
Interrupt Wake-up from Sleep	PC + 1 <sup>(1)</sup>	uuu1 0uuu	--uu --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.



## 11.3 Interrupts

The PIC16F688 has multiple sources of interrupt:

- External Interrupt RA2/INT
- TMR0 Overflow Interrupt
- PORTA Change Interrupts
- 2 Comparator Interrupts
- A/D Interrupt
- Timer1 Overflow Interrupt
- EEPROM Data Write Interrupt
- Fail-Safe Clock Monitor Interrupt
- EUSART Receive and Transmit interrupts

The Interrupt Control (INTCON) register and Peripheral Interrupt Request 1 (PIR1) register record individual interrupt requests in flag bits. The INTCON register also has individual and global interrupt enable bits.

A Global Interrupt Enable bit, GIE bit of the INTCON register, enables (if set) all unmasked interrupts, or disables (if cleared) all interrupts. Individual interrupts can be disabled through their corresponding enable bits in the INTCON register and PIE1 register. GIE is cleared on Reset.

The Return from Interrupt instruction, `RETFIE`, exits the interrupt routine, as well as sets the GIE bit, which re-enables unmasked interrupts.

The following interrupt flags are contained in the INTCON register:

- INT Pin Interrupt
- PORTA Change Interrupt
- TMR0 Overflow Interrupt

The peripheral interrupt flags are contained in the special register, PIR1. The corresponding interrupt enable bit is contained in special register, PIE1.

The following interrupt flags are contained in the PIR1 register:

- EEPROM Data Write Interrupt
- A/D Interrupt
- EUSART Receive and Transmit Interrupts
- 2 Comparator Interrupts
- Timer1 Overflow Interrupt
- Fail-Safe Clock Monitor Interrupt

When an interrupt is serviced:

- The GIE is cleared to disable any further interrupt.
- The return address is pushed onto the stack.
- The PC is loaded with 0004h.

For external interrupt events, such as the INT pin or PORTA change interrupt, the interrupt latency will be three or four instruction cycles. The exact latency depends upon when the interrupt event occurs (see Figure 11-8). The latency is the same for one or two-cycle instructions. Once in the Interrupt Service Routine, the source(s) of the interrupt can be determined by polling the interrupt flag bits. The interrupt flag bit(s) must be cleared in software before re-enabling interrupts to avoid multiple interrupt requests.

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

**2:** When an instruction that clears the GIE bit is executed, any interrupts that were pending for execution in the next cycle are ignored. The interrupts, which were ignored, are still pending to be serviced when the GIE bit is set again.

For additional information on Timer1, A/D or data EEPROM modules, refer to the respective peripheral section.

# PIC16F688

## 11.3.1 RA2/INT INTERRUPT

External interrupt on RA2/INT pin is edge-triggered; either rising if the INTEDG bit of the OPTION register is set, or falling if the INTEDG bit is clear. When a valid edge appears on the RA2/INT pin, the INTF bit of the INTCON register is set. This interrupt can be disabled by clearing the INTE control bit of the INTCON register. The INTF bit must be cleared in software in the Interrupt Service Routine before re-enabling this interrupt. The RA2/INT interrupt can wake-up the processor from Sleep if the INTE bit was set prior to going into Sleep. The status of the GIE bit decides whether or not the processor branches to the interrupt vector following wake-up (0004h). See **Section 11.6 “Power-Down Mode (Sleep)”** for details on Sleep and Figure 11-10 for timing of wake-up from Sleep through RA2/INT interrupt.

**Note:** The ANSEL (91h) and CMCON0 (19h) registers must be initialized to configure an analog channel as a digital input. Pins configured as analog inputs will read '0'.

## 11.3.2 TIMER0 INTERRUPT

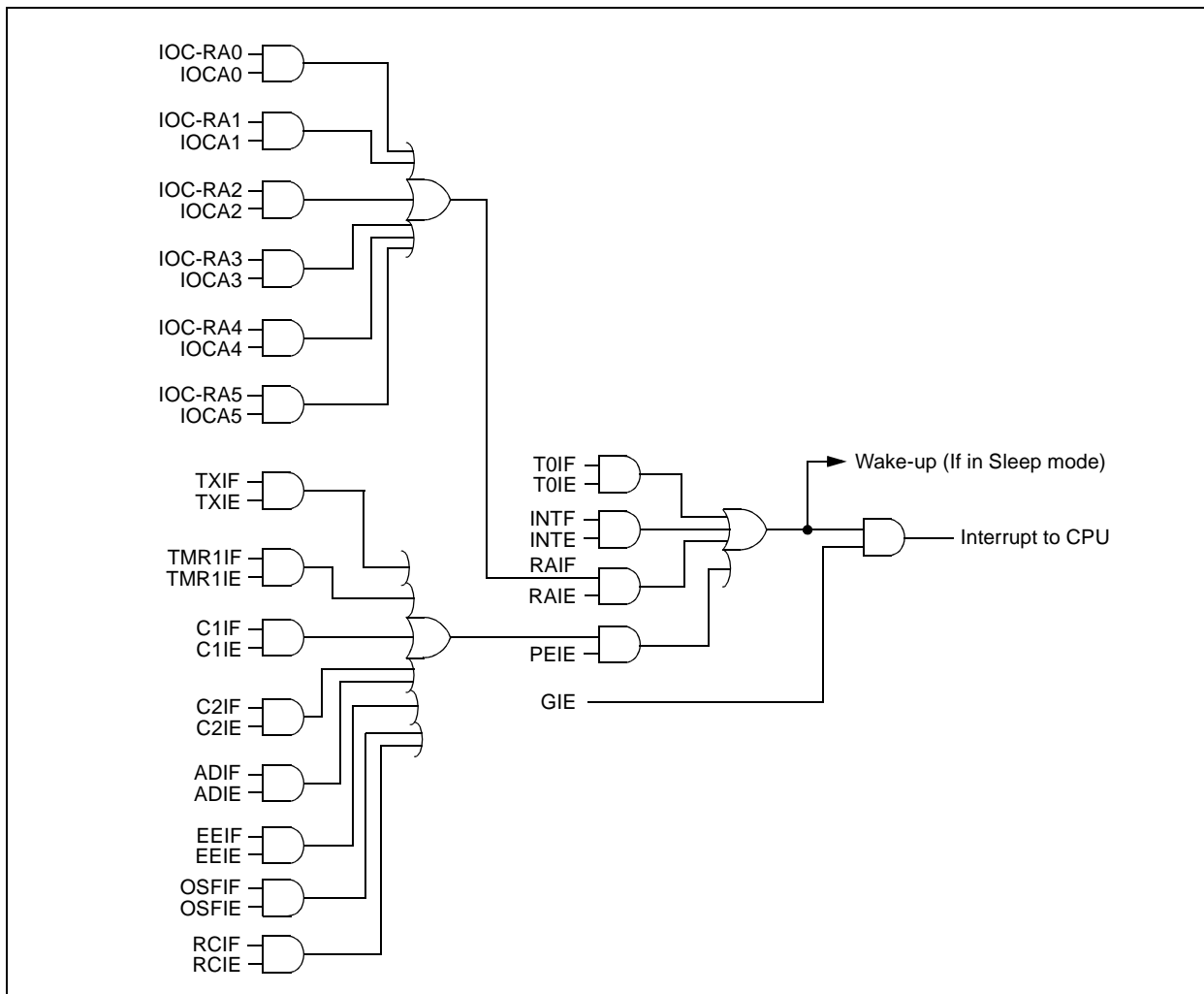
An overflow (FFh → 00h) in the TMR0 register will set the TOIF of the INTCON register bit. The interrupt can be enabled/disabled by setting/clearing TOIE bit of the INTCON register. See **Section 5.0 “Timer0 Module”** for operation of the Timer0 module.

## 11.3.3 PORTA INTERRUPT

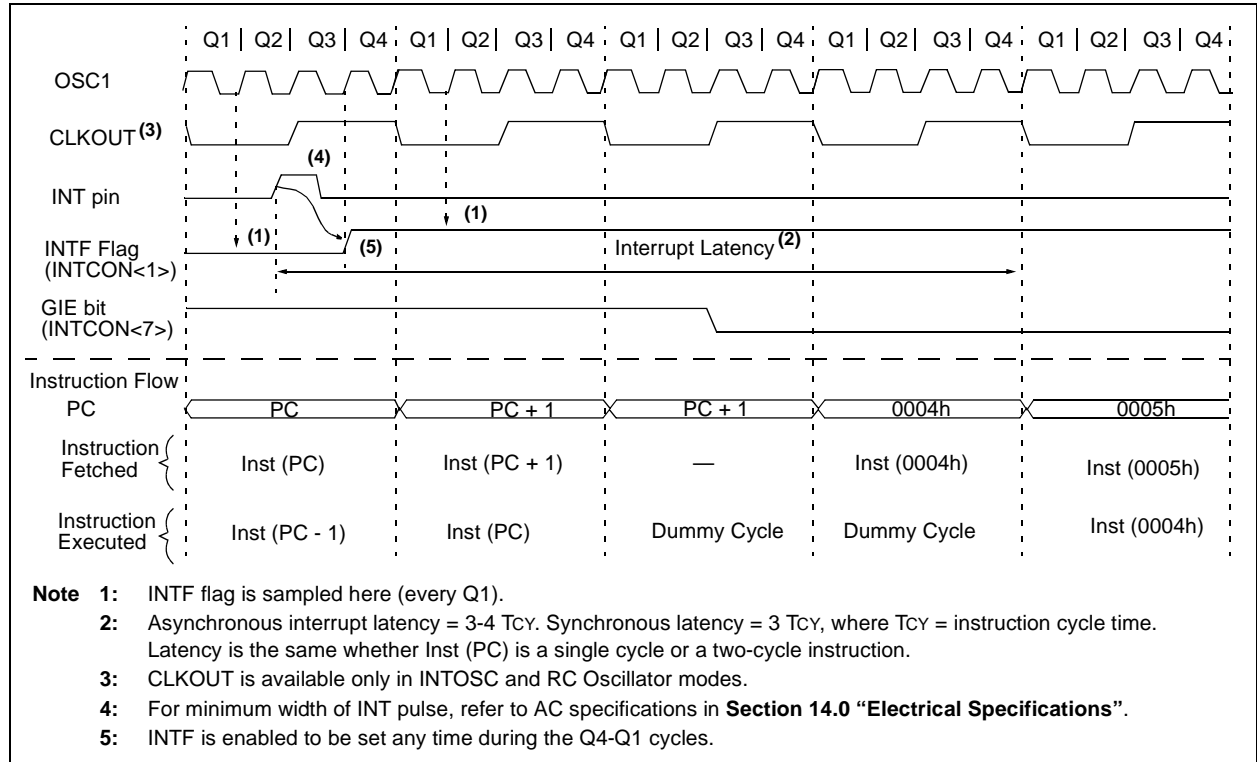
An input change on PORTA change sets the RAIF bit of the INTCON register. The interrupt can be enabled/disabled by setting/clearing the RAIE bit of the INTCON register. Plus, individual pins can be configured through the IOCA register.

**Note:** If a change on the I/O pin should occur when the read operation is being executed (start of the Q2 cycle), then the RAIF interrupt flag may not get set.

**FIGURE 11-7: INTERRUPT LOGIC**



**FIGURE 11-8: INT PIN INTERRUPT TIMING**



**TABLE 11-6: SUMMARY OF REGISTERS ASSOCIATED WITH INTERRUPTS**

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	TOIE	INTE	RAIE	TOIF	INTF	RAIF	0000 000x	0000 000x
PIE1	EEIE	ADIE	RCIE	C2IE	C1IE	OSFIE	TXIE	TMR1IE	0000 0000	0000 0000
PIR1	EEIF	ADIF	RCIF	C2IF	C1IF	OSFIF	TXIF	TMR1IF	0000 0000	0000 0000

**Legend:** x = unknown, u = unchanged, — = unimplemented read as '0', α = value depends upon condition.  
 Shaded cells are not used by the Interrupt module.

# PIC16F688

---

## 11.4 Context Saving During Interrupts

During an interrupt, only the return PC value is saved on the stack. Typically, users may wish to save key registers during an interrupt (e.g., W and Status registers). This must be implemented in software.

Since the lower 16 bytes of all banks are common in the PIC16F688 (see Figure 2-2), temporary holding registers, W\_TEMP and STATUS\_TEMP, should be placed in here. These 16 locations do not require banking and therefore, make it easier to context save and restore. The same code shown in Example 11-1 can be used to:

- Store the W register
- Store the Status register
- Execute the ISR code
- Restore the Status (and Bank Select Bit register)
- Restore the W register

**Note:** The PIC16F688 normally does not require saving the PCLATH. However, if computed GOTO's are used in the ISR and the main code, the PCLATH must be saved and restored in the ISR.

### EXAMPLE 11-1: SAVING STATUS AND W REGISTERS IN RAM

```
MOVWF  W_TEMP           ;Copy W to TEMP register
SWAPF  STATUS,W         ;Swap status to be saved into W
                               ;Swaps are used because they do not affect the status bits
MOVWF  STATUS_TEMP     ;Save status to bank zero STATUS_TEMP register
:
:(ISR)                   ;Insert user code here
:
SWAPF  STATUS_TEMP,W   ;Swap STATUS_TEMP register into W
                               ;(sets bank to original state)
MOVWF  STATUS          ;Move W into STATUS register
SWAPF  W_TEMP,F        ;Swap W_TEMP
SWAPF  W_TEMP,W        ;Swap W_TEMP into W
```

## 11.5 Watchdog Timer (WDT)

The WDT has the following features:

- Operates from the LFINTOSC (31 kHz)
- Contains a 16-bit prescaler
- Shares an 8-bit prescaler with Timer0
- Time-out period is from 1 ms to 268 seconds
- Configuration bit and software controlled

WDT is cleared under certain conditions described in Table 11-7.

### 11.5.1 WDT OSCILLATOR

The WDT derives its time base from the 31 kHz LFINTOSC. The LTS bit does not reflect that the LFINTOSC is enabled.

The value of WDTCON is '---0 1000' on all Resets. This gives a nominal time base of 16 ms, which is compatible with the time base generated with previous PIC16F688 microcontroller versions.

**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).

A new prescaler has been added to the path between the INTRC and the multiplexers used to select the path for the WDT. This prescaler is 16 bits and can be programmed to divide the INTRC by 32 to 65536, giving the WDT a nominal range of 1 ms to 268s.

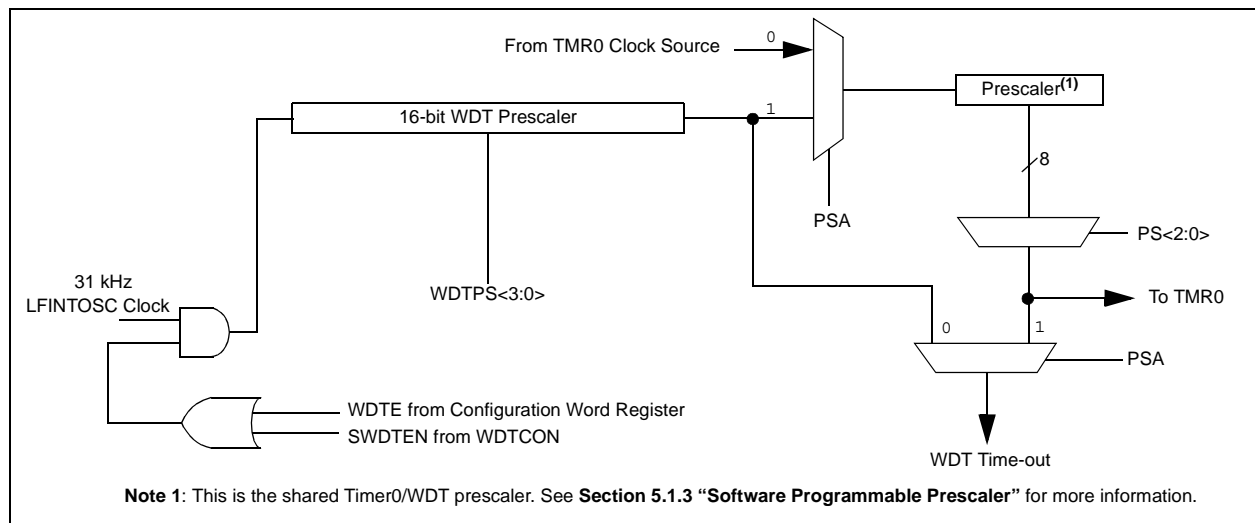
### 11.5.2 WDT CONTROL

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

When the WDTE bit in the Configuration Word register is set, the SWDTEN bit of the WDTCON register has no effect. If WDTE is clear, then the SWDTEN bit can be used to enable and disable the WDT. Setting the bit will enable it and clearing the bit will disable it.

The PSA and PS<2:0> bits of the OPTION register have the same function as in previous versions of the PIC16F688 family of microcontrollers. See **Section 5.0 "Timer0 Module"** for more information.

**FIGURE 11-9: WATCHDOG TIMER BLOCK DIAGRAM**



**TABLE 11-7: WDT STATUS**

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

# PIC16F688

## REGISTER 11-2: WDTCON: WATCHDOG TIMER CONTROL REGISTER

U-0	U-0	U-0	R/W-0	R/W-1	R/W-0	R/W-0	R/W-0
—	—	—	WDTPS3	WDTPS2	WDTPS1	WDTPS0	SWDTEN
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 4-1                      **WDTPS<3:0>:** Watchdog Timer Period Select bits

Bit Value = Prescale Rate

- 0000 = 1:32
- 0001 = 1:64
- 0010 = 1:128
- 0011 = 1:256
- 0100 = 1:512 (Reset value)
- 0101 = 1:1024
- 0110 = 1:2048
- 0111 = 1:4096
- 1000 = 1:8192
- 1001 = 1:16384
- 1010 = 1:32768
- 1011 = 1:65536
- 1100 = Reserved
- 1101 = Reserved
- 1110 = Reserved
- 1111 = Reserved

bit 0                      **SWDTEN:** Software Enable or Disable the Watchdog Timer<sup>(1)</sup>

- 1 = WDT is turned on
- 0 = WDT is turned off (Reset value)

**Note 1:** If WDTE Configuration bit = 1, then WDT is always enabled, irrespective of this control bit. If WDTE Configuration bit = 0, then it is possible to turn WDT on/off with this control bit.

## TABLE 11-8: SUMMARY OF REGISTERS ASSOCIATED WITH WATCHDOG TIMER

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
WDTCON	—	—	—	WDTPS3	WDTPS2	WDTPS1	WDTPS0	SWDTEN	---0 1000	---0 1000
OPTION_REG	RAPU	INTEDG	TOCS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
CONFIG	CPD	CP	MCLRE	PWRTE	WDTE	FOSC2	FOSC1	FOSC0	—	—

**Legend:** Shaded cells are not used by the Watchdog Timer.

**Note 1:** See Register 11.0 for operation of all Configuration Word register bits.

## 11.6 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 in the Status register is cleared.
- $\overline{TO}$  bit is set.
- Oscillator driver is turned off.
- 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  $V_{DD}$  or  $V_{SS}$ , with no external circuitry drawing current from the I/O pin, and the comparators and  $CV_{REF}$  should be disabled. I/O pins that are high-impedance inputs should be pulled high or low externally to avoid switching currents caused by floating inputs. The  $T_{OCLK}$  input should also be at  $V_{DD}$  or  $V_{SS}$  for lowest current consumption. The contribution from on-chip pull-ups on  $PORTA$  should be considered.

The  $\overline{MCLR}$  pin must be at a logic high level.

**Note:** It should be noted that a Reset generated by a WDT time-out does not drive  $\overline{MCLR}$  pin low.

### 11.6.1 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  $RA2/INT$  pin,  $PORTA$  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. Timer1 interrupt. Timer1 must be operating as an asynchronous counter.
2. A/D conversion (when A/D clock source is FRC).
3. EEPROM write operation completion.
4. Comparator output changes state.
5. Interrupt-on-change.
6. External Interrupt from  $INT$  pin.
7. EUSART Receive Interrupt.
8. ULPUWU Interrupt.

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.

### 11.6.2 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:

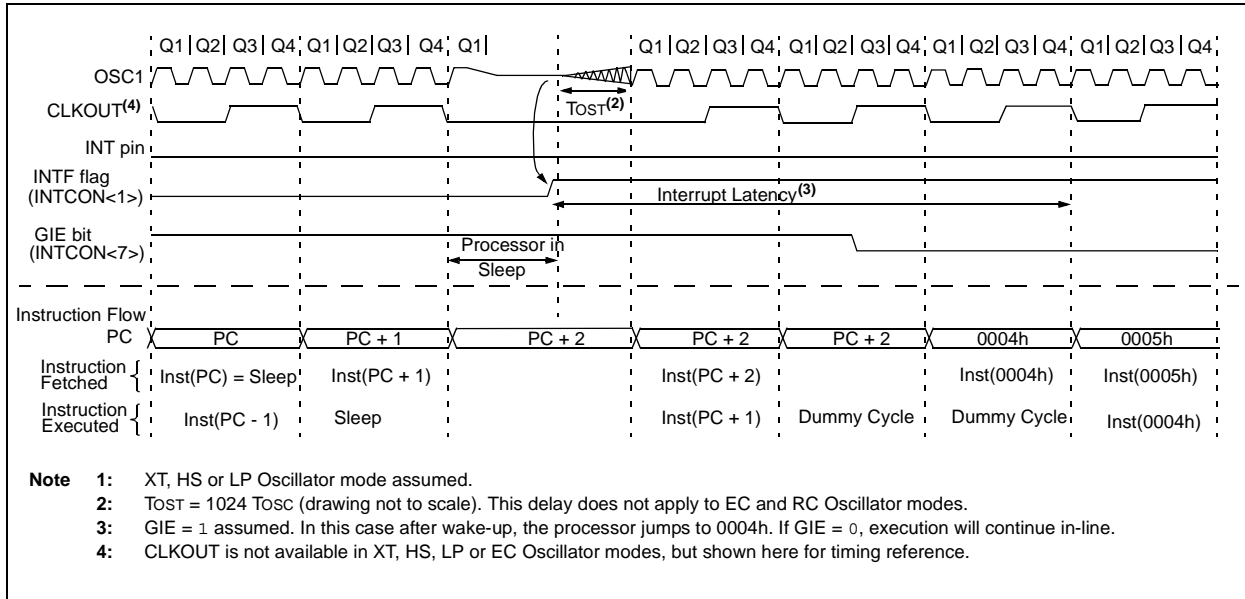
- If the interrupt occurs **before** the execution of a `SLEEP` instruction, the `SLEEP` instruction will complete as a `NOP`. Therefore, the WDT and WDT prescaler and postscaler (if enabled) will not be cleared, the  $\overline{TO}$  bit will not be set and the  $\overline{PD}$  bit will not be cleared.
- 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.

# PIC16F688

**FIGURE 11-10: WAKE-UP FROM SLEEP THROUGH INTERRUPT**



## 11.7 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 data EEPROM and Flash program memory will be erased when the code protection is turned off. See the "PIC12F6XX/16F6XX Memory Programming Specification" (DS41204) for more information.

## 11.8 ID Locations

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 used.



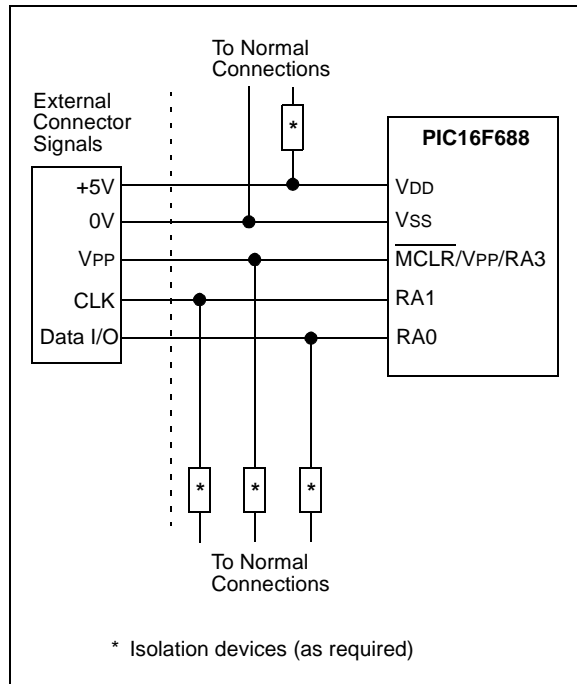
## 11.9 In-Circuit Serial Programming

This allows customers to manufacture boards with unprogrammed devices and then program the microcontroller just before shipping the product. This also allows the most recent firmware or a custom firmware to be programmed.

The device is placed into a Program/Verify mode by holding the RA0 and RA1 pins low, while raising the MCLR (VPP) pin from VIL to VIH. See the "PIC12F6XX/16F6XX Memory Programming Specification" (DS41204) for more information. RA0 becomes the programming data and RA1 becomes the programming clock. Both RA0 and RA1 are Schmitt Trigger inputs in Program/Verify mode.

A typical In-Circuit Serial Programming connection is shown in Figure 11-11.

**FIGURE 11-11: TYPICAL IN-CIRCUIT SERIAL PROGRAMMING CONNECTION**



## 11.10 In-Circuit Debugger

Since in-circuit debugging requires access to the data and MCLR pins, MPLAB® ICD 2 development with an 14-pin device is not practical. A special 20-pin PIC16F688 ICD device is used with MPLAB ICD 2 to provide separate clock, data and MCLR pins and frees all normally available pins to the user.

A special debugging adapter allows the ICD device to be used in place of a PIC16F688 device. The debugging adapter is the only source of the ICD device.

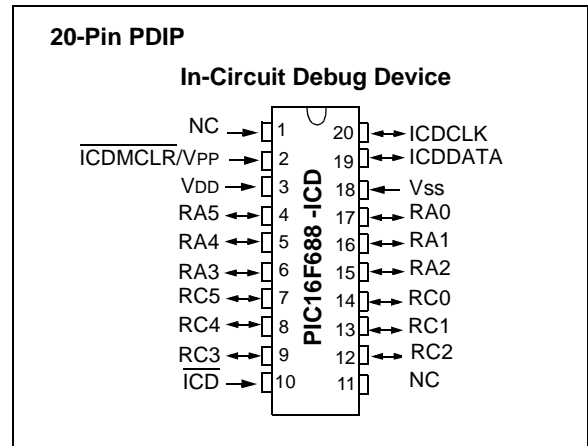
When the ICD pin on the PIC16F688 ICD device is held low, the In-Circuit Debugger functionality is enabled. This function allows simple debugging functions when used with MPLAB ICD 2. When the microcontroller has this feature enabled, some of the resources are not available for general use. Table 11-9 shows which features are consumed by the background debugger:

**TABLE 11-9: DEBUGGER RESOURCES**

Resource	Description
I/O pins	ICDCLK, ICDDATA
Stack	1 level
Program Memory	Address 0h must be NOP 700h-7FFh

For more information, see "MPLAB® ICD 2 In-Circuit Debugger User's Guide" (DS51331), available on Microchip's web site ([www.microchip.com](http://www.microchip.com)).

**FIGURE 11-12: 20-PIN ICD PINOUT**



# PIC16F688

---

NOTES:

## 12.0 INSTRUCTION SET SUMMARY

The PIC16F688 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 12-1, while the various opcode fields are summarized in Table 12-1.

Table 12-2 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.

### 12.1 Read-Modify-Write Operations

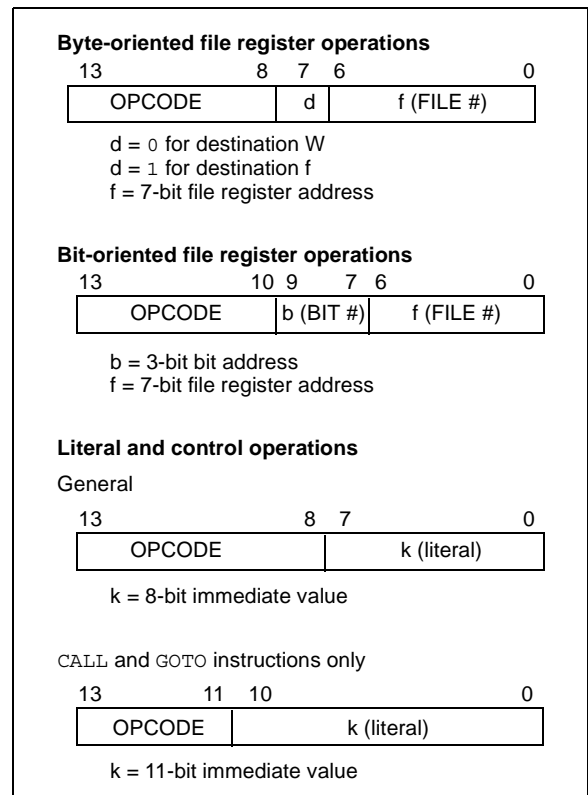
Any instruction that specifies a file register as part of the instruction performs a Read-Modify-Write (R-M-W) operation. The register is read, the data is modified, and the result is stored according to either the instruction, or the destination designator 'd'. A read operation is performed on a register even if the instruction writes to that register.

For example, a `CLRF PORTA` instruction will read PORTA, clear all the data bits, then write the result back to PORTA. This example would have the unintended consequence of clearing the condition that set the RAIF flag.

**TABLE 12-1: 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 12-1: GENERAL FORMAT FOR INSTRUCTIONS**



# PIC16F688

**TABLE 12-2: PIC16F684 INSTRUCTION SET**

Mnemonic, Operands	Description	Cycles	14-Bit Opcode				Status Affected	Notes	
			MSb	LSb					
<b>BYTE-ORIENTED FILE REGISTER OPERATIONS</b>									
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	1fff	ffff	Z	2
CLRWF	–	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	1fff	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
<b>BIT-ORIENTED FILE REGISTER OPERATIONS</b>									
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
<b>LITERAL AND CONTROL OPERATIONS</b>									
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		
CLRWD <sub>T</sub>	–	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 GPIO, 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`.

## 12.2 Instruction Descriptions

### **ADDLW**      **Add literal and W**

**Syntax:**      [ *label* ] ADDLW    *k*

**Operands:**     $0 \leq k \leq 255$

**Operation:**     $(W) + k \rightarrow (W)$

**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.

### **BCF**            **Bit Clear f**

**Syntax:**      [ *label* ] BCF    *f*,*b*

**Operands:**     $0 \leq f \leq 127$   
 $0 \leq b \leq 7$

**Operation:**     $0 \rightarrow (f<b>)$

**Status Affected:**    None

**Description:**    Bit 'b' in register 'f' is cleared.

### **ADDWF**        **Add W and f**

**Syntax:**      [ *label* ] ADDWF   *f*,*d*

**Operands:**     $0 \leq f \leq 127$   
 $d \in [0,1]$

**Operation:**     $(W) + (f) \rightarrow (\text{destination})$

**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'.

### **BSF**            **Bit Set f**

**Syntax:**      [ *label* ] BSF    *f*,*b*

**Operands:**     $0 \leq f \leq 127$   
 $0 \leq b \leq 7$

**Operation:**     $1 \rightarrow (f<b>)$

**Status Affected:**    None

**Description:**    Bit 'b' in register 'f' is set.

### **ANDLW**        **AND literal with W**

**Syntax:**      [ *label* ] ANDLW   *k*

**Operands:**     $0 \leq k \leq 255$

**Operation:**     $(W) .\text{AND.} (k) \rightarrow (W)$

**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.

### **BTFSC**         **Bit Test f, Skip if Clear**

**Syntax:**      [ *label* ] BTFSC   *f*,*b*

**Operands:**     $0 \leq f \leq 127$   
 $0 \leq b \leq 7$

**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 two-cycle instruction.

### **ANDWF**        **AND W with f**

**Syntax:**      [ *label* ] ANDWF   *f*,*d*

**Operands:**     $0 \leq f \leq 127$   
 $d \in [0,1]$

**Operation:**     $(W) .\text{AND.} (f) \rightarrow (\text{destination})$

**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'.

# PIC16F688

---

---

## **BTFSF**      **Bit Test f, Skip if Set**

---

Syntax:      [ *label* ] BTFSF f,b

Operands:     $0 \leq f \leq 127$   
               $0 \leq b < 7$

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 two-cycle instruction.

## **CLRWDT**      **Clear Watchdog Timer**

---

Syntax:      [ *label* ] CLRWDT

Operands:    None

Operation:    00h → WDT  
              0 → WDT prescaler,  
              1 →  $\overline{TO}$   
              1 → PD

Status Affected:  $\overline{TO}$ , PD

Description:    CLRWDT instruction resets the Watchdog Timer. It also resets the prescaler of the WDT.  
                  Status bits  $\overline{TO}$  and PD are set.

## **CALL**        **Call Subroutine**

---

Syntax:      [ *label* ] CALL k

Operands:     $0 \leq k \leq 2047$

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.

## **COMF**        **Complement f**

---

Syntax:      [ *label* ] COMF f,d

Operands:     $0 \leq f \leq 127$   
              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'.

## **CLRF**        **Clear f**

---

Syntax:      [ *label* ] CLRF f

Operands:     $0 \leq f \leq 127$

Operation:    00h → (f)  
              1 → Z

Status Affected: Z

Description:    The contents of register 'f' are cleared and the Z bit is set.

## **DECF**        **Decrement f**

---

Syntax:      [ *label* ] DECF f,d

Operands:     $0 \leq f \leq 127$   
              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'.

## **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.

---

**DECFSZ          Decrement f, Skip if 0**


---

Syntax:            [ *label* ] DECFSZ f,d

Operands:         $0 \leq f \leq 127$   
 $d \in [0,1]$

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 two-cycle instruction.

---

**INCFSZ          Increment f, Skip if 0**


---

Syntax:            [ *label* ] INCFSZ f,d

Operands:         $0 \leq f \leq 127$   
 $d \in [0,1]$

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 two-cycle instruction.

---

**GOTO            Unconditional Branch**


---

Syntax:            [ *label* ] GOTO k

Operands:         $0 \leq k \leq 2047$

Operation:         $k \rightarrow PC<10:0>$   
 $PCLATH<4:3> \rightarrow PC<12:11>$

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.

---

**IORLW          Inclusive OR literal with W**


---

Syntax:            [ *label* ] IORLW k

Operands:         $0 \leq k \leq 255$

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.

---

**INCF            Increment f**


---

Syntax:            [ *label* ] INCF f,d

Operands:         $0 \leq f \leq 127$   
 $d \in [0,1]$

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'.

---

**IORWF          Inclusive OR W with f**


---

Syntax:            [ *label* ] IORWF f,d

Operands:         $0 \leq f \leq 127$   
 $d \in [0,1]$

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'.

# PIC16F688

---

---

<b>MOVF</b>	<b>Move f</b>
Syntax:	[ <i>label</i> ] MOVF f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
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

<b>MOVLW</b>	<b>Move literal to W</b>
Syntax:	[ <i>label</i> ] MOVLW k
Operands:	$0 \leq k \leq 255$
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

<b>MOVWF</b>	<b>Move W to f</b>
Syntax:	[ <i>label</i> ] MOVWF f
Operands:	$0 \leq f \leq 127$
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

<b>NOP</b>	<b>No Operation</b>
Syntax:	[ <i>label</i> ] NOP
Operands:	None
Operation:	No operation
Status Affected:	None
Description:	No operation.
Words:	1
Cycles:	1
Example:	NOP



<b>RETFIE</b>	<b>Return from Interrupt</b>
Syntax:	[ <i>label</i> ] 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
<u>Example:</u>	RETFIE  After Interrupt PC = TOS GIE = 1

<b>RETLW</b>	<b>Return with literal in W</b>
Syntax:	[ <i>label</i> ] 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
<u>Example:</u>	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

<b>RETURN</b>	<b>Return from Subroutine</b>
Syntax:	[ <i>label</i> ] 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.

# PIC16F688

## RLF Rotate Left f through Carry

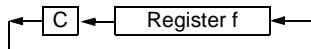
**Syntax:** `[label] RLF f,d`

**Operands:**  $0 \leq f \leq 127$   
 $d \in [0,1]$

**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
    
```

## 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.

## RRF Rotate Right f through Carry

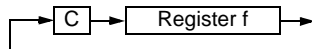
**Syntax:** `[label] RRF f,d`

**Operands:**  $0 \leq f \leq 127$   
 $d \in [0,1]$

**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'.



## SUBLW Subtract W from literal

**Syntax:** `[label] SUBLW k`

**Operands:**  $0 \leq k \leq 255$

**Operation:**  $k - (W) \rightarrow (W)$

**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	$W \leq k$
DC = 0	$W\langle 3:0 \rangle > k\langle 3:0 \rangle$
DC = 1	$W\langle 3:0 \rangle \leq k\langle 3:0 \rangle$

## **SUBWF**      **Subtract W from f**

**Syntax:**      [ *label* ] SUBWF f,d

**Operands:**     $0 \leq f \leq 127$   
 $d \in [0,1]$

**Operation:**     $(f) - (W) \rightarrow (\text{destination})$

**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	$W \leq f$
DC = 0	$W\langle 3:0 \rangle > f\langle 3:0 \rangle$
DC = 1	$W\langle 3:0 \rangle \leq f\langle 3:0 \rangle$

## **XORLW**      **Exclusive OR literal with W**

**Syntax:**      [ *label* ] XORLW k

**Operands:**     $0 \leq k \leq 255$

**Operation:**     $(W) .XOR. k \rightarrow (W)$

**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.

## **SWAPF**      **Swap Nibbles in f**

**Syntax:**      [ *label* ] SWAPF f,d

**Operands:**     $0 \leq f \leq 127$   
 $d \in [0,1]$

**Operation:**     $(f\langle 3:0 \rangle) \rightarrow (\text{destination}\langle 7:4 \rangle)$ ,  
 $(f\langle 7:4 \rangle) \rightarrow (\text{destination}\langle 3:0 \rangle)$

**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'.

## **XORWF**      **Exclusive OR W with f**

**Syntax:**      [ *label* ] XORWF f,d

**Operands:**     $0 \leq f \leq 127$   
 $d \in [0,1]$

**Operation:**     $(W) .XOR. (f) \rightarrow (\text{destination})$

**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'.

# PIC16F688

---

NOTES:

## 13.0 DEVELOPMENT SUPPORT

The PIC<sup>®</sup> microcontrollers are supported with a full range of hardware and software development tools:

- Integrated Development Environment
  - MPLAB<sup>®</sup> IDE Software
- Assemblers/Compilers/Linkers
  - MPASM<sup>™</sup> Assembler
  - MPLAB C18 and MPLAB C30 C Compilers
  - MPLINK<sup>™</sup> Object Linker/  
MPLIB<sup>™</sup> Object Librarian
  - MPLAB ASM30 Assembler/Linker/Library
- Simulators
  - MPLAB SIM Software Simulator
- Emulators
  - MPLAB ICE 2000 In-Circuit Emulator
  - MPLAB REAL ICE<sup>™</sup> In-Circuit Emulator
- In-Circuit Debugger
  - MPLAB ICD 2
- Device Programmers
  - PICSTART<sup>®</sup> Plus Development Programmer
  - MPLAB PM3 Device Programmer
  - PICKit<sup>™</sup> 2 Development Programmer
- Low-Cost Demonstration and Development Boards and Evaluation Kits

## 13.1 MPLAB Integrated Development Environment Software

The MPLAB IDE software brings an ease of software development previously unseen in the 8/16-bit microcontroller market. The MPLAB IDE is a Windows<sup>®</sup> operating system-based application that contains:

- A single graphical interface to all debugging tools
  - Simulator
  - Programmer (sold separately)
  - 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
- Visual device initializer for easy register initialization
- 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 HI-TECH Software C Compilers and IAR C Compilers

The MPLAB IDE allows you to:

- Edit your source files (either assembly or C)
- One touch assemble (or compile) and download to PIC MCU emulator and simulator tools (automatically updates all project information)
- Debug using:
  - Source files (assembly or C)
  - Mixed assembly and C
  - 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.

# PIC16F688

---

## 13.2 MPASM Assembler

The MPASM Assembler is a full-featured, universal macro assembler for all PIC 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

## 13.3 MPLAB C18 and MPLAB C30 C Compilers

The MPLAB C18 and MPLAB C30 Code Development Systems are complete ANSI C compilers for Microchip's PIC18 and PIC24 families of microcontrollers and the dsPIC30 and dsPIC33 family of digital signal controllers. These compilers provide powerful integration capabilities, superior code optimization and ease of use not found with other compilers.

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

## 13.4 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/librarian features include:

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

## 13.5 MPLAB ASM30 Assembler, Linker and Librarian

MPLAB ASM30 Assembler produces relocatable machine code from symbolic assembly language for dsPIC30F devices. MPLAB C30 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 dsPIC30F instruction set
- Support for fixed-point and floating-point data
- Command line interface
- Rich directive set
- Flexible macro language
- MPLAB IDE compatibility

## 13.6 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 C18 and MPLAB C30 C Compilers, and the MPASM and MPLAB ASM30 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.

## 13.7 MPLAB ICE 2000 High-Performance In-Circuit Emulator

The MPLAB ICE 2000 In-Circuit Emulator is intended to provide the product development engineer with a complete microcontroller design tool set for PIC microcontrollers. Software control of the MPLAB ICE 2000 In-Circuit Emulator is advanced by the MPLAB Integrated Development Environment, which allows editing, building, downloading and source debugging from a single environment.

The MPLAB ICE 2000 is a full-featured emulator system with enhanced trace, trigger and data monitoring features. Interchangeable processor modules allow the system to be easily reconfigured for emulation of different processors. The architecture of the MPLAB ICE 2000 In-Circuit Emulator allows expansion to support new PIC microcontrollers.

The MPLAB ICE 2000 In-Circuit Emulator system has been designed as a real-time emulation system with advanced features that are typically found on more expensive development tools. The PC platform and Microsoft® Windows® 32-bit operating system were chosen to best make these features available in a simple, unified application.

## 13.8 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® and dsPIC® Flash microcontrollers with the easy-to-use, powerful graphical user interface of the MPLAB Integrated Development Environment (IDE), included with each kit.

The MPLAB REAL ICE probe 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 the popular MPLAB ICD 2 system (RJ11) or with the new high speed, noise tolerant, low-voltage differential signal (LVDS) interconnection (CAT5).

MPLAB REAL ICE is field upgradeable through future firmware downloads in MPLAB IDE. In upcoming releases of MPLAB IDE, new devices will be supported, and new features will be added, such as software breakpoints and assembly code trace. MPLAB REAL ICE offers significant advantages over competitive emulators including low-cost, full-speed emulation, real-time variable watches, trace analysis, complex breakpoints, a ruggedized probe interface and long (up to three meters) interconnection cables.

## 13.9 MPLAB ICD 2 In-Circuit Debugger

Microchip's In-Circuit Debugger, MPLAB ICD 2, is a powerful, low-cost, run-time development tool, connecting to the host PC via an RS-232 or high-speed USB interface. This tool is based on the Flash PIC MCUs and can be used to develop for these and other PIC MCUs and dsPIC DSCs. The MPLAB ICD 2 utilizes the in-circuit debugging capability built into the Flash devices. This feature, along with Microchip's In-Circuit Serial Programming™ (ICSP™) protocol, offers cost-effective, in-circuit Flash debugging from the graphical user interface of the MPLAB Integrated Development Environment. This enables a designer to develop and debug source code by setting breakpoints, single stepping and watching variables, and CPU status and peripheral registers. Running at full speed enables testing hardware and applications in real time. MPLAB ICD 2 also serves as a development programmer for selected PIC devices.

## 13.10 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 SD/MMC card for file storage and secure data applications.

## 13.11 PICSTART Plus Development Programmer

The PICSTART Plus Development Programmer is an easy-to-use, low-cost, prototype programmer. It connects to the PC via a COM (RS-232) port. MPLAB Integrated Development Environment software makes using the programmer simple and efficient. The PICSTART Plus Development Programmer supports most PIC devices in DIP packages up to 40 pins. Larger pin count devices, such as the PIC16C92X and PIC17C76X, may be supported with an adapter socket. The PICSTART Plus Development Programmer is CE compliant.

## 13.12 PICkit 2 Development Programmer

The PICkit™ 2 Development Programmer is a low-cost programmer and selected Flash device debugger with an easy-to-use interface for programming many of Microchip's baseline, mid-range and PIC18F families of Flash memory microcontrollers. The PICkit 2 Starter Kit includes a prototyping development board, twelve sequential lessons, software and HI-TECH's PICC™ Lite C compiler, and is designed to help get up to speed quickly using PIC® microcontrollers. The kit provides everything needed to program, evaluate and develop applications using Microchip's powerful, mid-range Flash memory family of microcontrollers.

## 13.13 Demonstration, Development and Evaluation Boards

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.

Check the Microchip web page ([www.microchip.com](http://www.microchip.com)) and the latest "Product Selector Guide" (DS00148) for the complete list of demonstration, development and evaluation kits.



## 14.0 ELECTRICAL SPECIFICATIONS

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

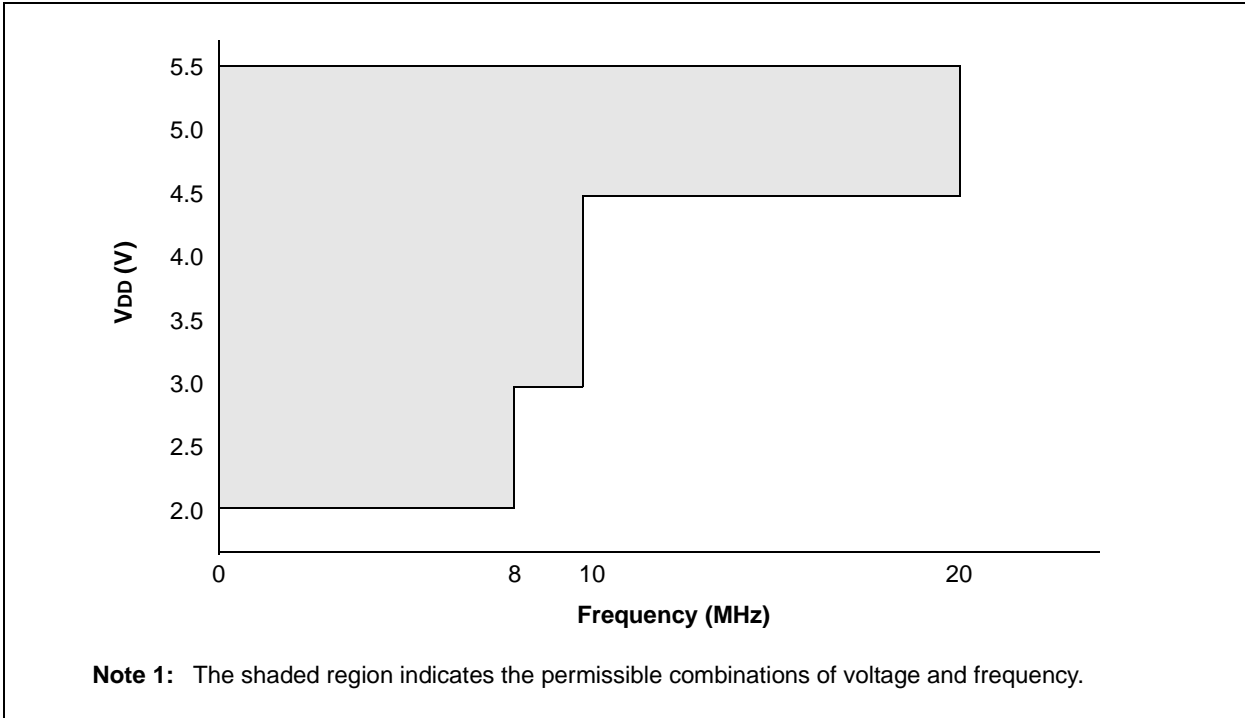
Ambient temperature under bias .....	-40° to +125°C
Storage temperature .....	-65°C to +150°C
Voltage on VDD with respect to VSS .....	-0.3V to +6.5V
Voltage on $\overline{\text{MCLR}}$ with respect to Vss .....	-0.3V to +13.5V
Voltage on all other pins with respect to Vss .....	-0.3V to (VDD + 0.3V)
Total power dissipation <sup>(1)</sup> .....	800 mW
Maximum current out of VSS pin .....	95 mA
Maximum current into VDD pin .....	95 mA
Input clamp current, I <sub>IK</sub> (V <sub>I</sub> < 0 or V <sub>I</sub> > VDD).....	± 20 mA
Output clamp current, I <sub>OK</sub> (V <sub>O</sub> < 0 or V <sub>O</sub> > VDD).....	± 20 mA
Maximum output current sunk by any I/O pin.....	25 mA
Maximum output current sourced by any I/O pin .....	25 mA
Maximum current sunk by PORTA and PORTC (combined) .....	90 mA
Maximum current sourced PORTA and PORTC (combined).....	90 mA

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

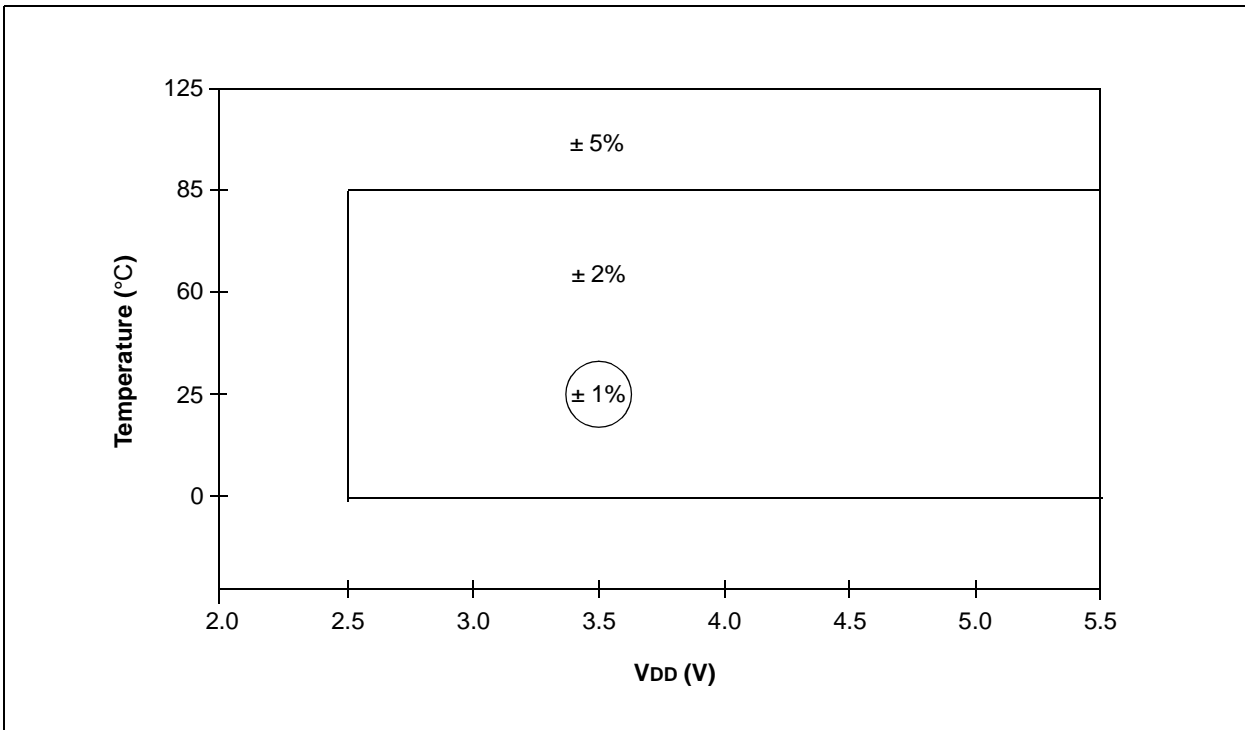
† NOTICE: Stresses above those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure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

# PIC16F688

**FIGURE 14-1: PIC16F688 VOLTAGE-FREQUENCY GRAPH,  $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$**



**FIGURE 14-2: HFINTOSC FREQUENCY ACCURACY OVER DEVICE V<sub>DD</sub> AND TEMPERATURE**



## 14.1 DC Characteristics: PIC16F688 -I (Industrial) PIC16F688 -E (Extended)

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
D001 D001C D001D	VDD	<b>Supply Voltage</b>	2.0 2.0 3.0 4.5	— — — —	5.5 5.5 5.5 5.5	V V V V	Fosc ≤ 8 MHz: HFINTOSC, EC Fosc ≤ 4 MHz Fosc ≤ 10 MHz Fosc ≤ 20 MHz
D002*	VDR	<b>RAM Data Retention Voltage<sup>(1)</sup></b>	1.5	—	—	V	Device in Sleep mode
D003	VPOR	<b>VDD Start Voltage</b> to ensure internal Power-on Reset signal	—	VSS	—	V	See <b>Section 11.2.1 “Power-On Reset”</b> for details.
D004*	SVDD	<b>VDD Rise Rate</b> to ensure internal Power-on Reset signal	0.05	—	—	V/ms	See <b>Section 11.2.1 “Power-On Reset”</b> for details.

\* These parameters are characterized but not tested.

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

**Note 1:** This is the limit to which VDD can be lowered in Sleep mode without losing RAM data.

# PIC16F688

## 14.2 DC Characteristics: PIC16F688 -I (Industrial) PIC16F688 -E (Extended)

DC CHARACTERISTICS		Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for extended					
Param No.	Device Characteristics	Min	Typ†	Max	Units	Conditions	
						VDD	Note
D010	Supply Current (IDD) <sup>(1, 2)</sup>	—	16	23	μA	2.0	Fosc = 32 kHz
		—	27	38	μA	3.0	LP Oscillator mode
		—	47	75	μA	5.0	
D011*		—	180	250	μA	2.0	Fosc = 1 MHz
		—	290	400	μA	3.0	XT Oscillator mode
		—	490	650	μA	5.0	
D012		—	280	380	μA	2.0	Fosc = 4 MHz
		—	480	670	μA	3.0	XT Oscillator mode
		—	0.9	1.4	mA	5.0	
D013*		—	130	220	μA	2.0	Fosc = 1 MHz
		—	215	360	μA	3.0	EC Oscillator mode
		—	360	520	μA	5.0	
D014		—	220	340	μA	2.0	Fosc = 4 MHz
		—	375	550	μA	3.0	EC Oscillator mode
		—	0.65	1.0	mA	5.0	
D015		—	8	20	μA	2.0	Fosc = 31 kHz
		—	16	40	μA	3.0	LFINTOSC mode
		—	31	65	μA	5.0	
D016*		—	320	400	μA	2.0	Fosc = 4 MHz
		—	490	640	μA	3.0	HFINTOSC mode
		—	0.87	1.2	mA	5.0	
D017		—	0.5	0.7	mA	2.0	Fosc = 8 MHz
		—	0.78	1	mA	3.0	HFINTOSC mode
		—	1.43	1.8	mA	5.0	
D018		—	340	580	μA	2.0	Fosc = 4 MHz
		—	550	950	μA	3.0	EXTRC mode <sup>(3)</sup>
		—	0.92	1.6	mA	5.0	
D019		—	2.9	3.7	mA	4.5	Fosc = 20 MHz
		—	3.1	3.8	mA	5.0	HS Oscillator mode

\* These parameters are characterized but not tested.

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

- 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; 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  $I_R = V_{DD}/2R_{EXT}$  (mA) with REXT in kΩ.

## 14.3 DC Characteristics: PIC16F688-I (Industrial)

DC CHARACTERISTICS		Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial					
Param No.	Device Characteristics	Min	Typ†	Max	Units	Conditions	
						VDD	Note
D020	Power-down Base Current(IPD) <sup>(2)</sup>	—	0.05	1.2	$\mu\text{A}$	2.0	WDT, BOR, Comparators, VREF and T1OSC disabled
		—	0.15	1.5	$\mu\text{A}$	3.0	
		—	0.35	1.8	$\mu\text{A}$	5.0	
		—	150	500	nA	3.0	$-40^{\circ}\text{C} \leq T_A \leq +25^{\circ}\text{C}$
D021		—	1.0	2.2	$\mu\text{A}$	2.0	WDT Current <sup>(1)</sup>
		—	2.0	4.0	$\mu\text{A}$	3.0	
		—	3.0	7.0	$\mu\text{A}$	5.0	
D022		—	42	60	$\mu\text{A}$	3.0	BOR Current <sup>(1)</sup>
		—	85	122	$\mu\text{A}$	5.0	
D023		—	32	45	$\mu\text{A}$	2.0	Comparator Current <sup>(1)</sup> , both comparators enabled
		—	60	78	$\mu\text{A}$	3.0	
		—	120	160	$\mu\text{A}$	5.0	
D024		—	30	36	$\mu\text{A}$	2.0	CVREF Current <sup>(1)</sup> (high range)
		—	45	55	$\mu\text{A}$	3.0	
		—	75	95	$\mu\text{A}$	5.0	
D025*		—	39	47	$\mu\text{A}$	2.0	CVREF Current <sup>(1)</sup> (low range)
		—	59	72	$\mu\text{A}$	3.0	
		—	98	124	$\mu\text{A}$	5.0	
D026		—	4.5	7.0	$\mu\text{A}$	2.0	T1OSC Current <sup>(1)</sup> , 32.768 kHz
		—	5.0	8.0	$\mu\text{A}$	3.0	
		—	6.0	12	$\mu\text{A}$	5.0	
D027		—	0.30	1.6	$\mu\text{A}$	3.0	A/D Current <sup>(1)</sup> , no conversion in progress
		—	0.36	1.9	$\mu\text{A}$	5.0	

\* These parameters are characterized but not tested.

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

- 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  $\Delta$  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.

# PIC16F688

## 14.4 DC Characteristics: PIC16F688-E (Extended)

DC CHARACTERISTICS		Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for extended					
Param No.	Device Characteristics	Min	Typ†	Max	Units	Conditions	
						VDD	Note
D020E	Power-down Base Current (IPD) <sup>(2)</sup>	—	0.05	9	$\mu\text{A}$	2.0	WDT, BOR, Comparators, VREF and T1OSC disabled
		—	0.15	11	$\mu\text{A}$	3.0	
		—	0.35	15	$\mu\text{A}$	5.0	
D021E		—	1	28	$\mu\text{A}$	2.0	WDT Current <sup>(1)</sup>
		—	2	30	$\mu\text{A}$	3.0	
		—	3	35	$\mu\text{A}$	5.0	
D022E		—	42	65	$\mu\text{A}$	3.0	BOR Current <sup>(1)</sup>
		—	85	127	$\mu\text{A}$	5.0	
D023E		—	32	45	$\mu\text{A}$	2.0	Comparator Current <sup>(1)</sup> , both comparators enabled
		—	60	78	$\mu\text{A}$	3.0	
		—	120	160	$\mu\text{A}$	5.0	
D024E		—	30	70	$\mu\text{A}$	2.0	CVREF Current <sup>(1)</sup> (high range)
		—	45	90	$\mu\text{A}$	3.0	
		—	75	120	$\mu\text{A}$	5.0	
D025E*		—	39	91	$\mu\text{A}$	2.0	CVREF Current <sup>(1)</sup> (low range)
		—	59	117	$\mu\text{A}$	3.0	
		—	98	156	$\mu\text{A}$	5.0	
D026E		—	4.5	25	$\mu\text{A}$	2.0	T1OSC Current <sup>(1)</sup> , 32.768 kHz
		—	5	30	$\mu\text{A}$	3.0	
		—	6	40	$\mu\text{A}$	5.0	
D027E		—	0.30	12	$\mu\text{A}$	3.0	A/D Current <sup>(1)</sup> , no conversion in progress
		—	0.36	16	$\mu\text{A}$	5.0	

\* These parameters are characterized but not tested.

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

- 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  $\Delta$  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.

## 14.5 DC Characteristics: PIC16F688 -I (Industrial) PIC16F688 -E (Extended)

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	
D030	V <sub>IL</sub>	<b>Input Low Voltage</b> I/O Port: with TTL buffer	V <sub>SS</sub>	—	0.8	V	4.5V ≤ V <sub>DD</sub> ≤ 5.5V	
D030A			V <sub>SS</sub>	—	0.15 V <sub>DD</sub>	V	2.0V ≤ V <sub>DD</sub> ≤ 4.5V	
D031			with Schmitt Trigger buffer	V <sub>SS</sub>	—	0.2 V <sub>DD</sub>	V	2.0V ≤ V <sub>DD</sub> ≤ 5.5V
D032			MCLR, OSC1 (RC mode) <sup>(1)</sup>	V <sub>SS</sub>	—	0.2 V <sub>DD</sub>	V	
D033			OSC1 (XT and LP modes)	V <sub>SS</sub>	—	0.3	V	
D033A			OSC1 (HS mode)	V <sub>SS</sub>	—	0.3 V <sub>DD</sub>	V	
D040	V <sub>IH</sub>	<b>Input High Voltage</b> I/O ports: with TTL buffer	2.0	—	V <sub>DD</sub>	V	4.5V ≤ V <sub>DD</sub> ≤ 5.5V	
D040A			0.25 V <sub>DD</sub> + 0.8	—	V <sub>DD</sub>	V	2.0V ≤ V <sub>DD</sub> ≤ 4.5V	
D041			with Schmitt Trigger buffer	0.8 V <sub>DD</sub>	—	V <sub>DD</sub>	V	2.0V ≤ V <sub>DD</sub> ≤ 5.5V
D042			MCLR	0.8 V <sub>DD</sub>	—	V <sub>DD</sub>	V	
D043			OSC1 (XT and LP modes)	1.6	—	V <sub>DD</sub>	V	
D043A			OSC1 (HS mode)	0.7 V <sub>DD</sub>	—	V <sub>DD</sub>	V	
D043B			OSC1 (RC mode)	0.9 V <sub>DD</sub>	—	V <sub>DD</sub>	V	<b>(Note 1)</b>
D060	I <sub>IL</sub>	<b>Input Leakage Current<sup>(2)</sup></b> I/O ports	—	± 0.1	± 1	μA	V <sub>SS</sub> ≤ V <sub>PIN</sub> ≤ V <sub>DD</sub> , Pin at high-impedance	
D061			MCLR <sup>(3)</sup>	—	± 0.1	± 5	μA	V <sub>SS</sub> ≤ V <sub>PIN</sub> ≤ V <sub>DD</sub>
D063			OSC1	—	± 0.1	± 5	μA	V <sub>SS</sub> ≤ V <sub>PIN</sub> ≤ V <sub>DD</sub> , XT, HS and LP oscillator configuration
D070*	I <sub>PUR</sub>	PORTA Weak Pull-up Current	50	250	400	μA	V <sub>DD</sub> = 5.0V, V <sub>PIN</sub> = V <sub>SS</sub>	
D080	V <sub>OL</sub>	<b>Output Low Voltage<sup>(5)</sup></b> I/O ports	—	—	0.6	V	I <sub>OL</sub> = 8.5 mA, V <sub>DD</sub> = 4.5V (Ind.)	
D090			V <sub>OH</sub>	<b>Output High Voltage<sup>(5)</sup></b> I/O ports	V <sub>DD</sub> - 0.7	—	—	V

\* These parameters are characterized but not tested.

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

**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 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:** See Section 9.0 "Data EEPROM and Flash Program Memory Control" for additional information.

**5:** Including OSC2 in CLKOUT mode.

# PIC16F688

## 14.5 DC Characteristics: PIC16F688 -I (Industrial) PIC16F688 -E (Extended) (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
D100	IULP	Ultra Low-Power Wake-Up Current	—	200	—	nA	See Application Note AN879, "Using the Microchip Ultra Low-Power Wake-up Module" (DS00879)
		<b>Capacitive Loading Specs on Output Pins</b>					
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	
		<b>Data EEPROM Memory</b>					
D120	ED	Byte Endurance	100K	1M	—	E/W	-40°C ≤ TA ≤ +85°C
D120A	ED	Byte Endurance	10K	100K	—	E/W	+85°C ≤ TA ≤ +125°C
D121	VDRW	VDD for Read/Write	VMIN	—	5.5	V	Using EECON1 to read/write VMIN = Minimum operating voltage
D122	TDEW	Erase/Write Cycle Time	—	5	6	ms	
D123	TRETD	Characteristic Retention	40	—	—	Year	Provided no other specifications are violated
D124	TREF	Number of Total Erase/Write Cycles before Refresh <sup>(4)</sup>	1M	10M	—	E/W	-40°C ≤ TA ≤ +85°C
		<b>Program Flash Memory</b>					
D130	EP	Cell Endurance	10K	100K	—	E/W	-40°C ≤ TA ≤ +85°C
D130A	ED	Cell Endurance	1K	10K	—	E/W	+85°C ≤ TA ≤ +125°C
D131	VPR	VDD for Read	VMIN	—	5.5	V	VMIN = Minimum operating voltage
D132	VPEW	VDD for Erase/Write	4.5	—	5.5	V	
D133	TPEW	Erase/Write cycle time	—	2	2.5	ms	
D134	TRETD	Characteristic Retention	40	—	—	Year	Provided no other specifications are violated

\* These parameters are characterized but not tested.

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

- 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 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:** See Section 9.0 "Data EEPROM and Flash Program Memory Control" for additional information.
- 5:** Including OSC2 in CLKOUT mode.



## 14.6 Thermal Considerations

Standard Operating Conditions (unless otherwise stated)					
Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$					
Param No.	Sym	Characteristic	Typ	Units	Conditions
TH01	$\theta_{JA}$	Thermal Resistance Junction to Ambient	69.8	C/W	14-pin PDIP package
			85.0	C/W	14-pin SOIC package
			100.4	C/W	14-pin TSSOP package
			46.3	C/W	16-pin QFN 4x0.9mm package
TH02	$\theta_{JC}$	Thermal Resistance Junction to Case	32.5	C/W	14-pin PDIP package
			31.0	C/W	14-pin SOIC package
			31.7	C/W	14-pin TSSOP package
			2.6	C/W	16-pin QFN 4x0.9mm package
TH03	$T_J$	Junction Temperature	150	C	For derated power calculations
TH04	PD	Power Dissipation	—	W	$PD = P_{INTERNAL} + P_{I/O}$
TH05	$P_{INTERNAL}$	Internal Power Dissipation	—	W	$P_{INTERNAL} = I_{DD} \times V_{DD}$ <b>(NOTE 1)</b>
TH06	$P_{I/O}$	I/O Power Dissipation	—	W	$P_{I/O} = \Sigma (I_{OL} \times V_{OL}) + \Sigma (I_{OH} \times (V_{DD} - V_{OH}))$
TH07	$P_{DER}$	Derated Power	—	W	$P_{DER} = (T_J - T_A) / \theta_{JA}$ <b>(NOTE 2, 3)</b>

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

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

**3:** Maximum allowable power dissipation is the lower value of either the absolute maximum total power dissipation or derated power ( $P_{DER}$ ).

# PIC16F688

## 14.7 Timing Parameter Symbology

The timing parameter symbols have been created with one of the following formats:

1. TppS2ppS
2. TppS

<b>T</b>			
F	Frequency	T	Time

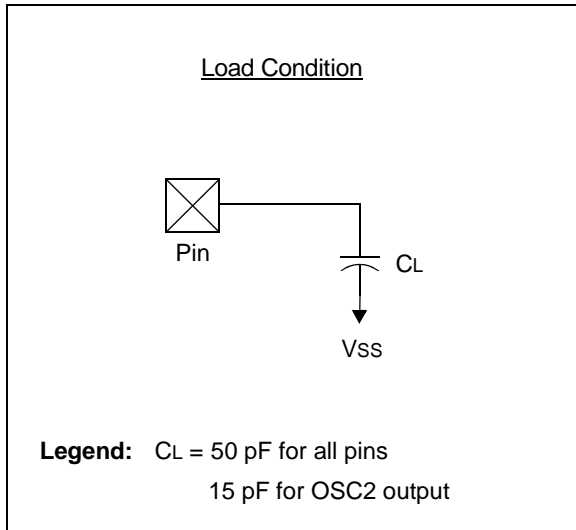
Lowercase letters (pp) and their meanings:

<b>pp</b>			
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:

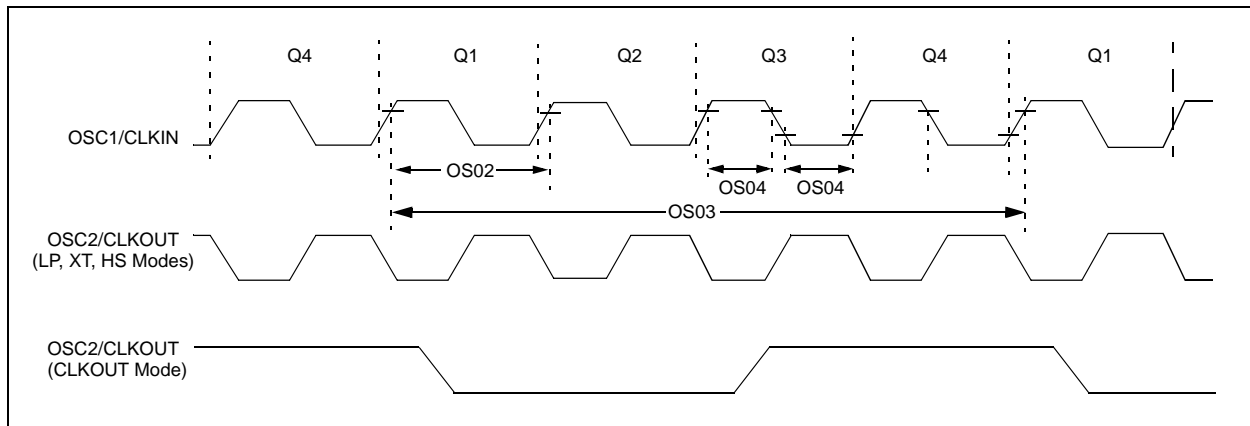
<b>S</b>			
F	Fall	P	Period
H	High	R	Rise
I	Invalid (High-impedance)	V	Valid
L	Low	Z	High-impedance

**FIGURE 14-3: LOAD CONDITIONS**



## 14.8 AC Characteristics: PIC16F688 (Industrial, Extended)

**FIGURE 14-4: CLOCK TIMING**



**TABLE 14-1: CLOCK OSCILLATOR TIMING REQUIREMENTS**

Standard Operating Conditions (unless otherwise stated)							
Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$							
Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
OS01	Fosc	External CLKIN Frequency <sup>(1)</sup>	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 <sup>(1)</sup>	—	32.768	—	kHz	LP Oscillator mode
			0.1	—	4	MHz	XT Oscillator mode
			1	—	20	MHz	HS Oscillator mode
			DC	—	4	MHz	RC Oscillator mode
OS02	Tosc	External CLKIN Period <sup>(1)</sup>	27	—	•	μs	LP Oscillator mode
			250	—	•	ns	XT Oscillator mode
			50	—	•	ns	HS Oscillator mode
			50	—	•	ns	EC Oscillator mode
		Oscillator Period <sup>(1)</sup>	—	30.5	—	μs	LP Oscillator mode
			250	—	10,000	ns	XT Oscillator mode
			50	—	1,000	ns	HS Oscillator mode
			250	—	—	ns	RC Oscillator mode
OS03	Tcy	Instruction Cycle Time <sup>(1)</sup>	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 5V, 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.

# PIC16F688

**TABLE 14-2: OSCILLATOR PARAMETERS**

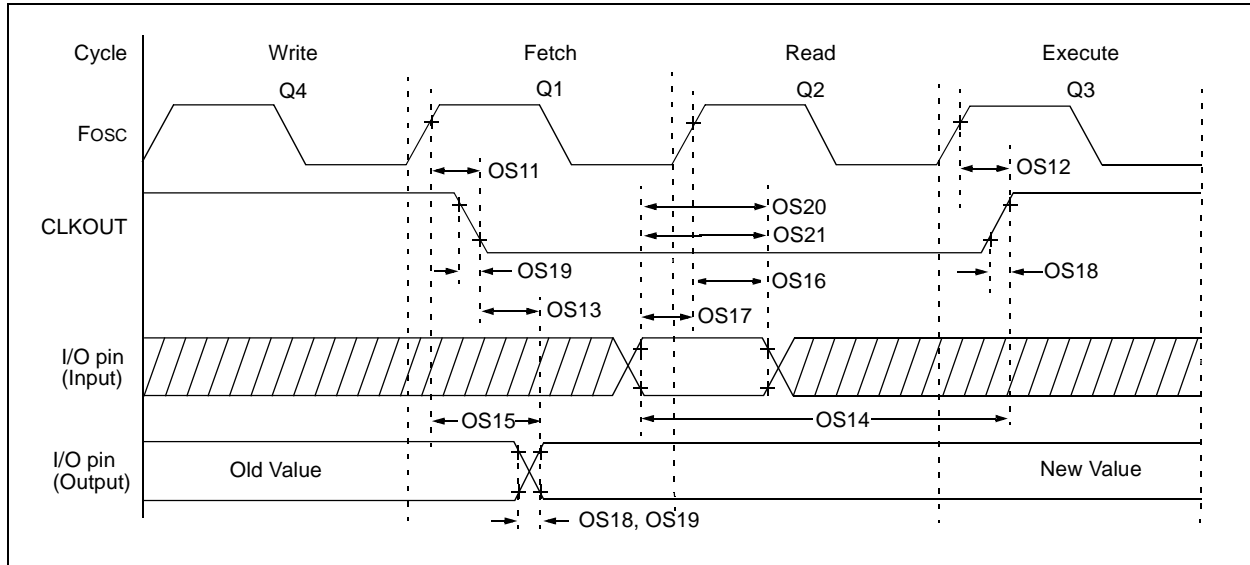
Standard Operating Conditions (unless otherwise stated)								
Operating Temperature $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$								
Param No.	Sym	Characteristic	Freq Tolerance	Min	Typ†	Max	Units	Conditions
OS06	TWARM	Internal Oscillator Switch when running <sup>(3)</sup>	—	—	—	2	Tosc	Slowest clock
OS07	Tsc	Fail-Safe Sample Clock Period <sup>(1)</sup>	—	—	21	—	ms	LFINTOSC/64
OS08	HFosc	Internal Calibrated HFINTOSC Frequency <sup>(2)</sup>	$\pm 1\%$	7.92	8.0	8.08	MHz	$V_{DD} = 3.5\text{V}$ , $25^{\circ}\text{C}$
			$\pm 2\%$	7.84	8.0	8.16	MHz	$2.5\text{V} \leq V_{DD} \leq 5.5\text{V}$ , $0^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$
			$\pm 5\%$	7.60	8.0	8.40	MHz	$2.0\text{V} \leq V_{DD} \leq 5.5\text{V}$ , $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (Ind.), $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ (Ext.)
OS09*	LFosc	Internal Uncalibrated LFINTOSC Frequency	—	15	31	45	kHz	
OS10*	Tiosc ST	HFINTOSC Oscillator Wake-up from Sleep Start-up Time	—	5.5	12	24	$\mu\text{s}$	$V_{DD} = 2.0\text{V}$ , $-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$
			—	3.5	7	14	$\mu\text{s}$	$V_{DD} = 3.0\text{V}$ , $-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$
			—	3	6	11	$\mu\text{s}$	$V_{DD} = 5.0\text{V}$ , $-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$

\* These parameters are characterized but not tested.

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

- 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  $\mu\text{F}$  and 0.01  $\mu\text{F}$  values in parallel are recommended.
- 3:** By design.

**FIGURE 14-5: CLKOUT AND I/O TIMING**



**TABLE 14-3: CLKOUT AND I/O TIMING PARAMETERS**

Standard Operating Conditions (unless otherwise stated)							
Operating Temperature $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$							
Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
OS11	TosH2ckL	Fosc $\uparrow$ to CLKOUT $\downarrow$ <sup>(1)</sup>	—	—	70	ns	VDD = 5.0V
OS12	TosH2ckH	Fosc $\uparrow$ to CLKOUT $\uparrow$ <sup>(1)</sup>	—	—	72	ns	VDD = 5.0V
OS13	TckL2ioV	CLKOUT $\downarrow$ to Port out valid <sup>(1)</sup>	—	—	20	ns	
OS14	TioV2ckH	Port input valid before CLKOUT $\uparrow$ <sup>(1)</sup>	Tosc + 200 ns	—	—	ns	
OS15*	TosH2ioV	Fosc $\uparrow$ (Q1 cycle) to Port out valid	—	50	70	ns	VDD = 5.0V
OS16	TosH2ioI	Fosc $\uparrow$ (Q2 cycle) to Port input invalid (I/O in hold time)	50	—	—	ns	VDD = 5.0V
OS17	TioV2osH	Port input valid to Fosc $\uparrow$ (Q2 cycle) (I/O in setup time)	20	—	—	ns	
OS18	TioR	Port output rise time <sup>(2)</sup>	—	15 40	72 32	ns	VDD = 2.0V VDD = 5.0V
OS19	TioF	Port output fall time <sup>(2)</sup>	—	28 15	55 30	ns	VDD = 2.0V VDD = 5.0V
OS20*	TINP	INT pin input high or low time	25	—	—	ns	
OS21*	TRAP	PORTA interrupt-on-change new input level time	TCY	—	—	ns	

\* These parameters are characterized but not tested.

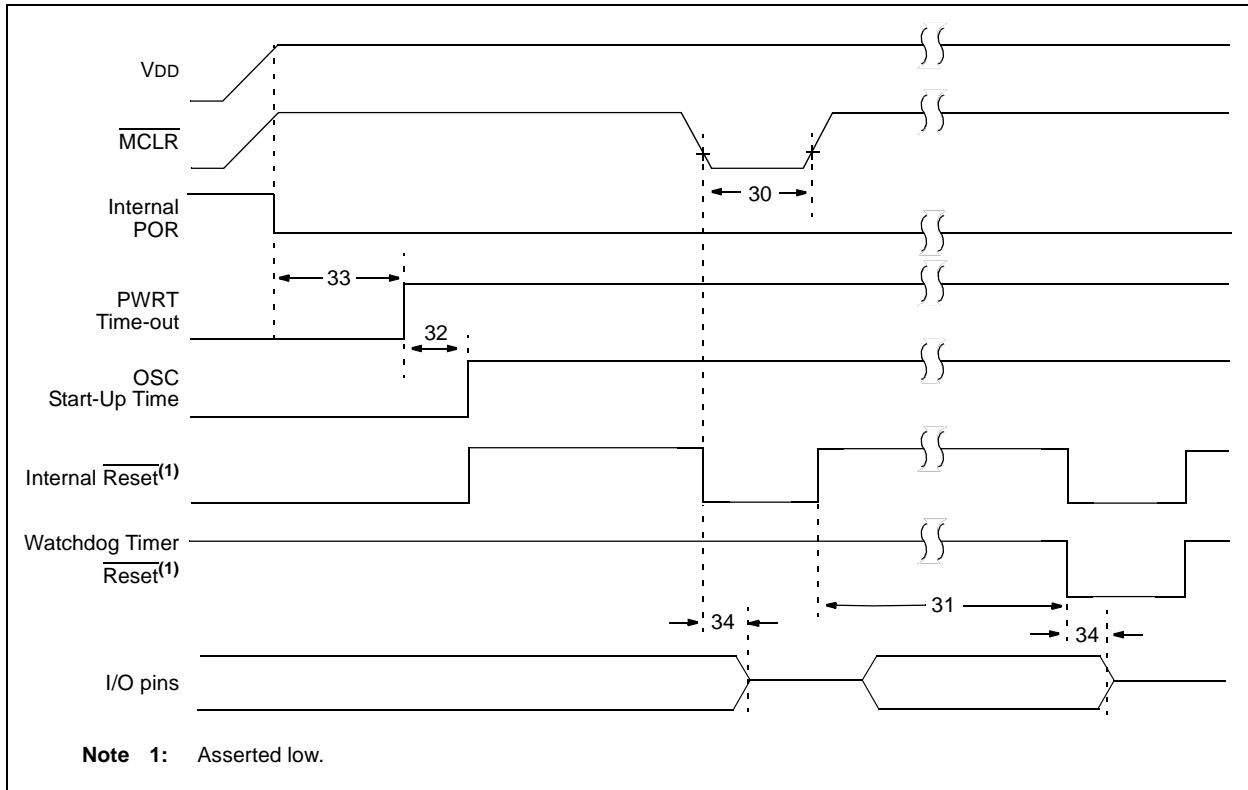
† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated.

**Note 1:** Measurements are taken in RC mode where CLKOUT output is 4 x TOSC.

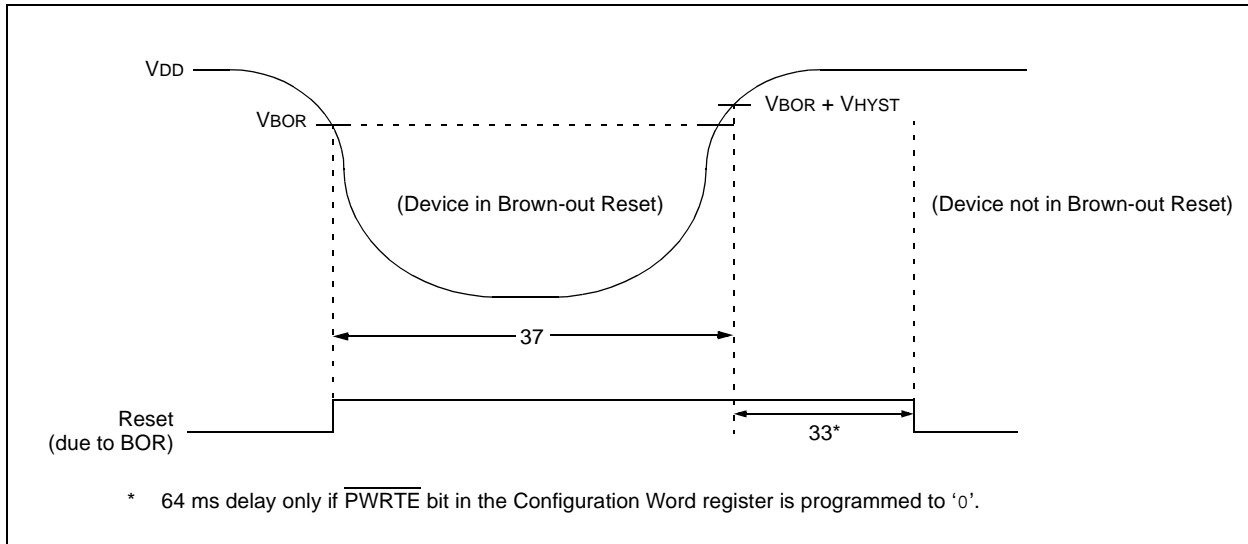
**2:** Includes OSC2 in CLKOUT mode.

# PIC16F688

**FIGURE 14-6: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING**



**FIGURE 14-7: BROWN-OUT RESET TIMING AND CHARACTERISTICS**



\* 64 ms delay only if  $\overline{\text{PWRTE}}$  bit in the Configuration Word register is programmed to '0'.

**TABLE 14-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER, POWER-UP TIMER AND BROWN-OUT RESET PARAMETERS**

Standard Operating Conditions (unless otherwise stated)							
Operating Temperature $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$							
Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
30	TMCL	MCLR Pulse Width (low)	2	—	—	$\mu\text{s}$	$V_{DD} = 5\text{V}$ , $-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$ $V_{DD} = 5\text{V}$
			5	—	—	$\mu\text{s}$	
31	TWDT	Watchdog Timer Time-out Period (No Prescaler)	10	16	29	ms	$V_{DD} = 5\text{V}$ , $-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$ $V_{DD} = 5\text{V}$
			10	16	31	ms	
32	TOST	Oscillation Start-up Timer Period <sup>(1, 2)</sup>	—	1024	—	TOSC	<b>(NOTE 3)</b>
33*	TPWRT	Power-up Timer Period	40	65	140	ms	
34*	TIOZ	I/O High-impedance from MCLR Low or Watchdog Timer Reset	—	—	2.0	$\mu\text{s}$	
35	VBOR	Brown-out Reset Voltage	2.0	—	2.2	V	<b>(NOTE 4)</b>
36*	VHYST	Brown-out Reset Hysteresis	—	50	—	mV	
37*	TBOR	Brown-out Reset Minimum Detection Period	100	—	—	$\mu\text{s}$	$V_{DD} \leq V_{BOR}$

\* These parameters are characterized but not tested.

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

**Note 1:** Instruction cycle period (T<sub>cy</sub>) 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.

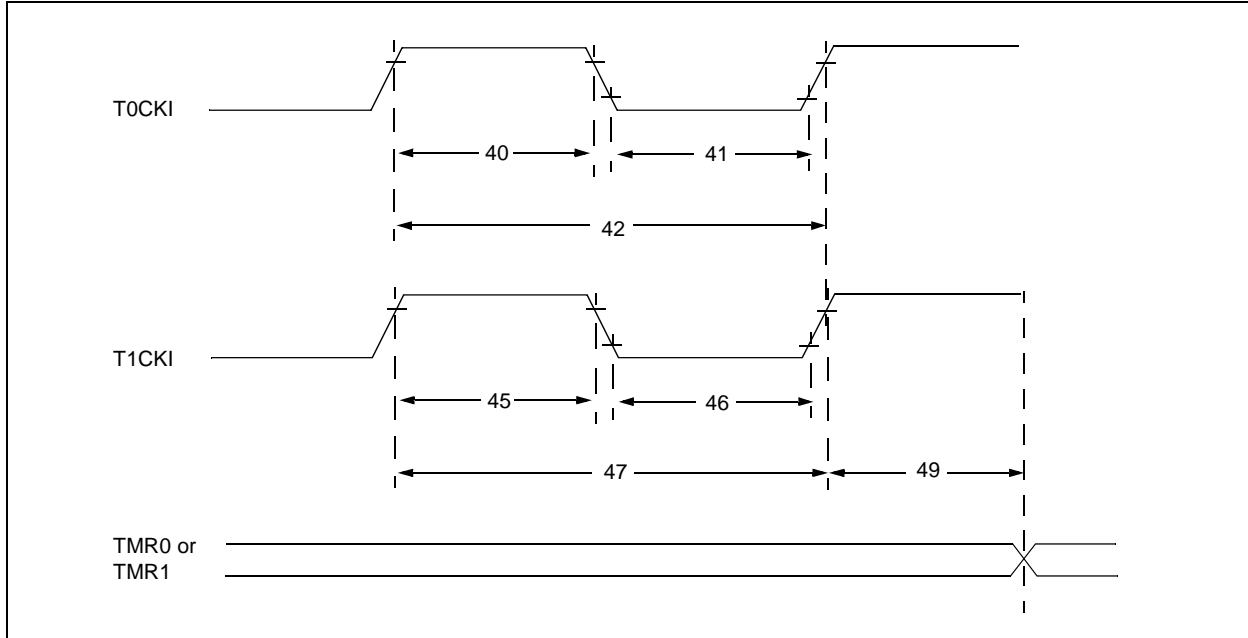
2: By design.

3: Period of the slower clock.

4: To ensure these voltage tolerances, V<sub>DD</sub> and V<sub>SS</sub> must be capacitively decoupled as close to the device as possible. 0.1  $\mu\text{F}$  and 0.01  $\mu\text{F}$  values in parallel are recommended.

# PIC16F688

**FIGURE 14-8: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS**



**TABLE 14-5: TIMER0 AND TIMER1 EXTERNAL CLOCK REQUIREMENTS**

Standard Operating Conditions (unless otherwise stated)								
Operating Temperature $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$								
Param No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
40*	Tt0H	T0CKI High Pulse Width	No Prescaler	$0.5 T_{CY} + 20$	—	—	ns	
			With Prescaler	10	—	—	ns	
41*	Tt0L	T0CKI Low Pulse Width	No Prescaler	$0.5 T_{CY} + 20$	—	—	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	$0.5 T_{CY} + 20$	—	—	ns	
			Synchronous, with Prescaler	15	—	—	ns	
			Asynchronous	30	—	—	ns	
46*	Tt1L	T1CKI Low Time	Synchronous, No Prescaler	$0.5 T_{CY} + 20$	—	—	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)		—	32.768	—	kHz	
49*	TCKEZTMR1	Delay from External Clock Edge to Timer Increment		$2 T_{osc}$	—	$7 T_{osc}$	—	Timers in Sync mode

\* These parameters are characterized but not tested.

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



**TABLE 14-6: COMPARATOR SPECIFICATIONS**

Standard Operating Conditions (unless otherwise stated)								
Operating Temperature $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$								
Param No.	Sym	Characteristics		Min	Typ†	Max	Units	Comments
CM01	VOS	Input Offset Voltage		—	$\pm 5.0$	$\pm 10$	mV	$(V_{DD} - 1.5)/2$
CM02	VCM	Input Common Mode Voltage		0	—	$V_{DD} - 1.5$	V	
CM03*	CMRR	Common Mode Rejection Ratio		+55	—	—	dB	
CM04*	TRT	Response Time	Falling	—	150	600	ns	<b>(NOTE 1)</b>
			Rising	—	200	1000	ns	
CM05*	TMC2COV	Comparator Mode Change to Output Valid		—	—	10	$\mu\text{s}$	

\* These parameters are characterized but not tested.

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

**Note 1:** Response time is measured with one comparator input at  $(V_{DD} - 1.5)/2 - 100\text{ mV}$  to  $(V_{DD} - 1.5)/2 + 20\text{ mV}$ .

**TABLE 14-7: COMPARATOR VOLTAGE REFERENCE (CVREF) SPECIFICATIONS**

Standard Operating Conditions (unless otherwise stated)							
Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$							
Param No.	Sym	Characteristics	Min	Typ†	Max	Units	Comments
CV01*	CLSB	Step Size <sup>(2)</sup>	—	$V_{DD}/24$	—	V	Low Range (VRR = 1)
			—	$V_{DD}/32$	—	V	High Range (VRR = 0)
CV02*	CACC	Absolute Accuracy	—	—	$\pm 1/2$	LSb	Low Range (VRR = 1)
			—	—	$\pm 1/2$	LSb	High Range (VRR = 0)
CV03*	CR	Unit Resistor Value (R)	—	2k	—	$\Omega$	
CV04*	CST	Settling Time <sup>(1)</sup>	—	—	10	$\mu\text{s}$	

\* These parameters are characterized but not tested.

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

**Note 1:** Settling time measured while  $VRR = 1$  and  $VR<3:0>$  transitions from ‘0000’ to ‘1111’.

**2:** See **Section 7.10 “Comparator Voltage Reference”** for more information.

# PIC16F688

**TABLE 14-8: PIC16F688 A/D CONVERTER (ADC) CHARACTERISTICS**

Standard Operating Conditions (unless otherwise stated)							
Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$							
Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
AD01	NR	Resolution	—	—	10 bits	bit	
AD02	EIL	Integral Error	—	—	$\pm 1$	LSb	$V_{REF} = 5.12\text{V}$
AD03	EDL	Differential Error	—	—	$\pm 1$	LSb	No missing codes to 10 bits $V_{REF} = 5.12\text{V}$
AD04	E <sub>OFF</sub>	Offset Error	—	—	$\pm 1$	LSb	$V_{REF} = 5.12\text{V}$
AD07	E <sub>GN</sub>	Gain Error	—	—	$\pm 1$	LSb	$V_{REF} = 5.12\text{V}$
AD06 AD06A	V <sub>REF</sub>	Reference Voltage <sup>(1)</sup>	2.2 2.7	—	— V <sub>DD</sub>	V	Absolute minimum to ensure 1 LSb accuracy
AD07	V <sub>AIN</sub>	Full-Scale Range	V <sub>SS</sub>	—	V <sub>REF</sub>	V	
AD08	Z <sub>AIN</sub>	Recommended Impedance of Analog Voltage Source	—	—	10	k $\Omega$	
AD09*	I <sub>REF</sub>	V <sub>REF</sub> Input Current <sup>(1)</sup>	10	—	1000	$\mu\text{A}$	During V <sub>AIN</sub> acquisition. Based on differential of V <sub>HOLD</sub> to V <sub>AIN</sub> .
			—	—	50	$\mu\text{A}$	During A/D conversion cycle.

\* These parameters are characterized but not tested.

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

**Note 1:** ADC V<sub>REF</sub> is from external V<sub>REF</sub> or V<sub>DD</sub> pin, whichever is selected as reference input.

**TABLE 14-9: PIC16F688 A/D CONVERSION REQUIREMENTS**

Standard Operating Conditions (unless otherwise stated)							
Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$							
Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
AD130*	TAD	A/D Clock Period	1.6	—	9.0	$\mu\text{s}$	TOSC-based, $V_{\text{REF}} \geq 3.0\text{V}$
			3.0	—	9.0	$\mu\text{s}$	TOSC-based, $V_{\text{REF}}$ full range
		A/D Internal RC Oscillator Period	3.0	6.0	9.0	$\mu\text{s}$	ADCS<1:0> = 11 (ADRC mode) At $V_{\text{DD}} = 2.5\text{V}$
			1.6	4.0	6.0	$\mu\text{s}$	At $V_{\text{DD}} = 5.0\text{V}$
AD131	TCNV	Conversion Time (not including Acquisition Time) <sup>(1)</sup>	—	11	—	TAD	Set GO/DONE bit to new data in A/D Result register
AD132*	TACQ	Acquisition Time	—	11.5	—	$\mu\text{s}$	
AD133*	TAMP	Amplifier Settling Time	—	—	5	$\mu\text{s}$	
AD134	TGO	Q4 to A/D Clock Start	—	Tosc/2	—	—	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.
			—	Tosc/2 + TCY	—	—	

\* These parameters are characterized but not tested.

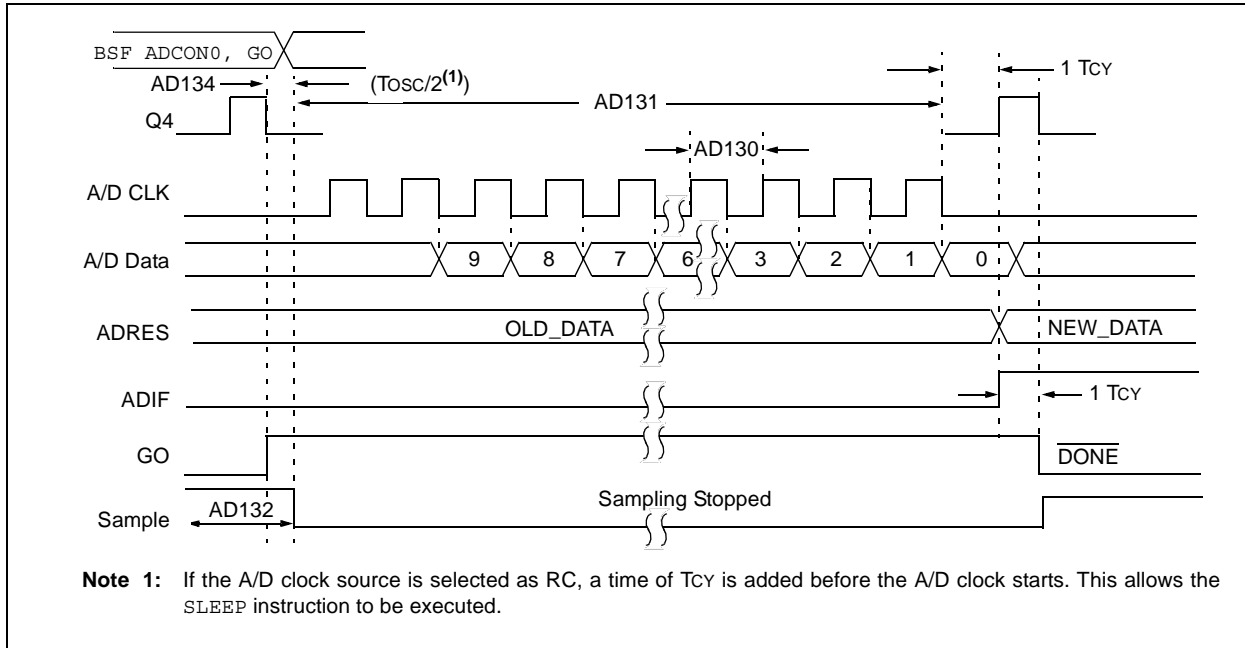
† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** ADRESH and ADRESL registers may be read on the following TCY cycle.

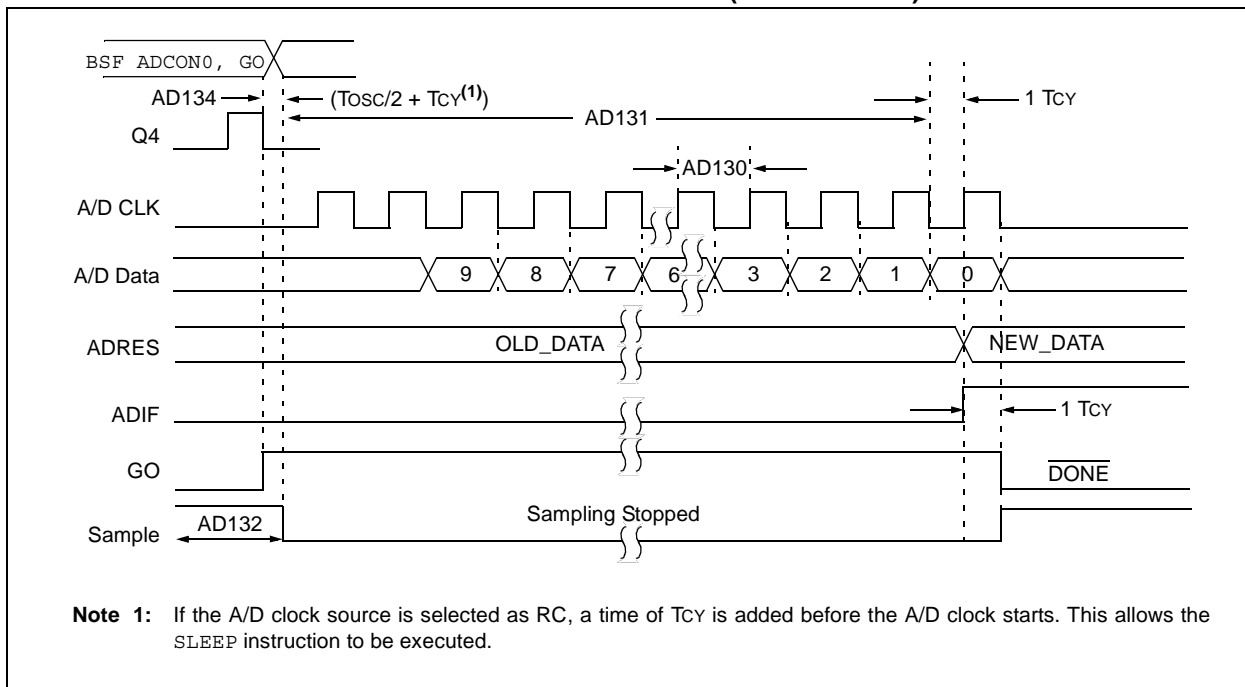
**2:** See **Section 8.3 "A/D Acquisition Requirements"** for minimum conditions.

# PIC16F688

**FIGURE 14-9: PIC16F688 A/D CONVERSION TIMING (NORMAL MODE)**



**FIGURE 14-10: PIC16F688 A/D CONVERSION TIMING (SLEEP MODE)**



## 15.0 DC AND AC CHARACTERISTICS GRAPHS AND TABLES

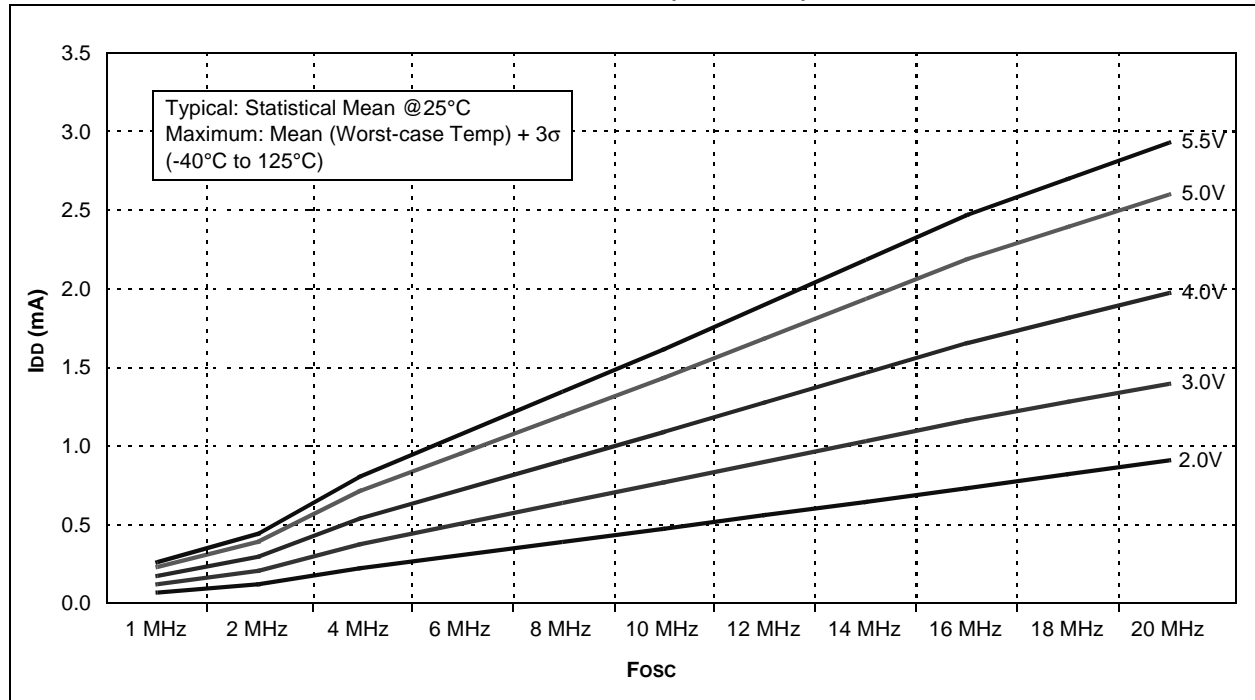
The graphs and tables provided in this section are for **design guidance** and are **not tested**.

In some graphs or tables, the data presented are **outside specified operating range** (i.e., outside specified  $V_{DD}$  range). This is for **information only** and devices are ensured to operate properly only within the specified range.

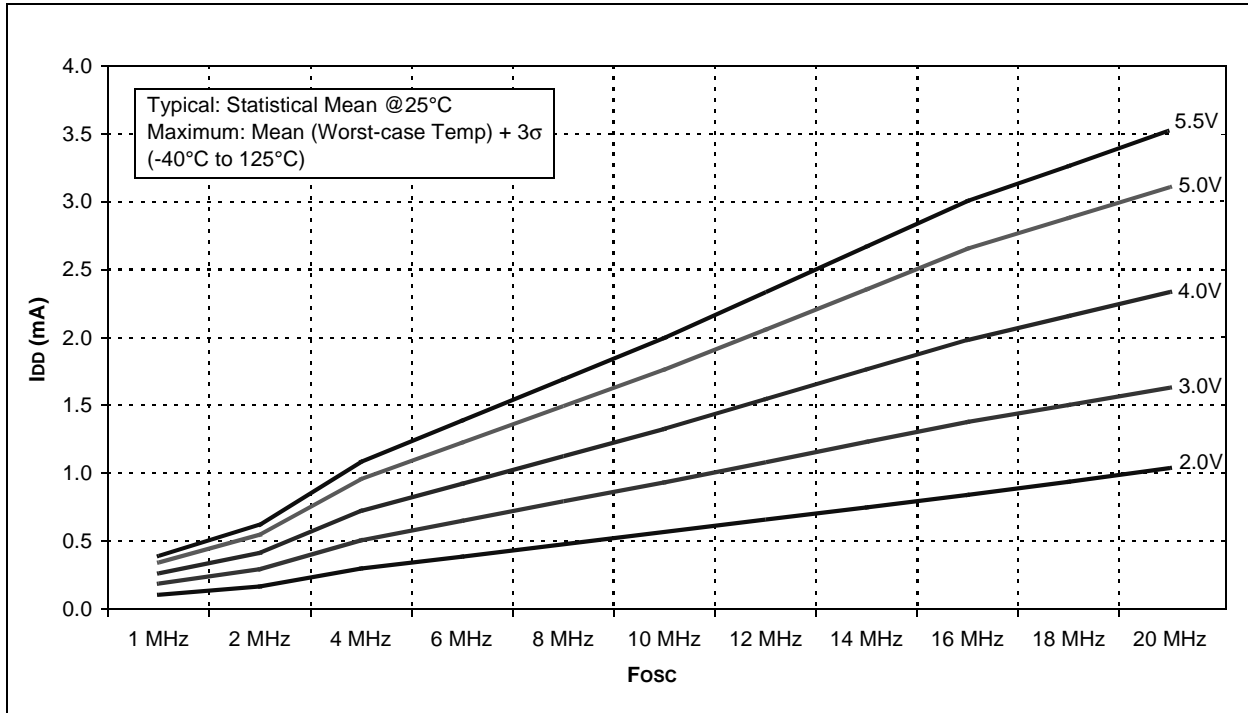
**Note:** The graphs and tables provided following this note are a statistical summary based on a limited number of samples and are provided for informational purposes only. The performance characteristics listed herein are not tested or guaranteed. In some graphs or tables, the data presented may be 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 + 3 $\sigma$ ) or (mean - 3 $\sigma$ ) respectively, where  $\sigma$  is a standard deviation, over each temperature range.

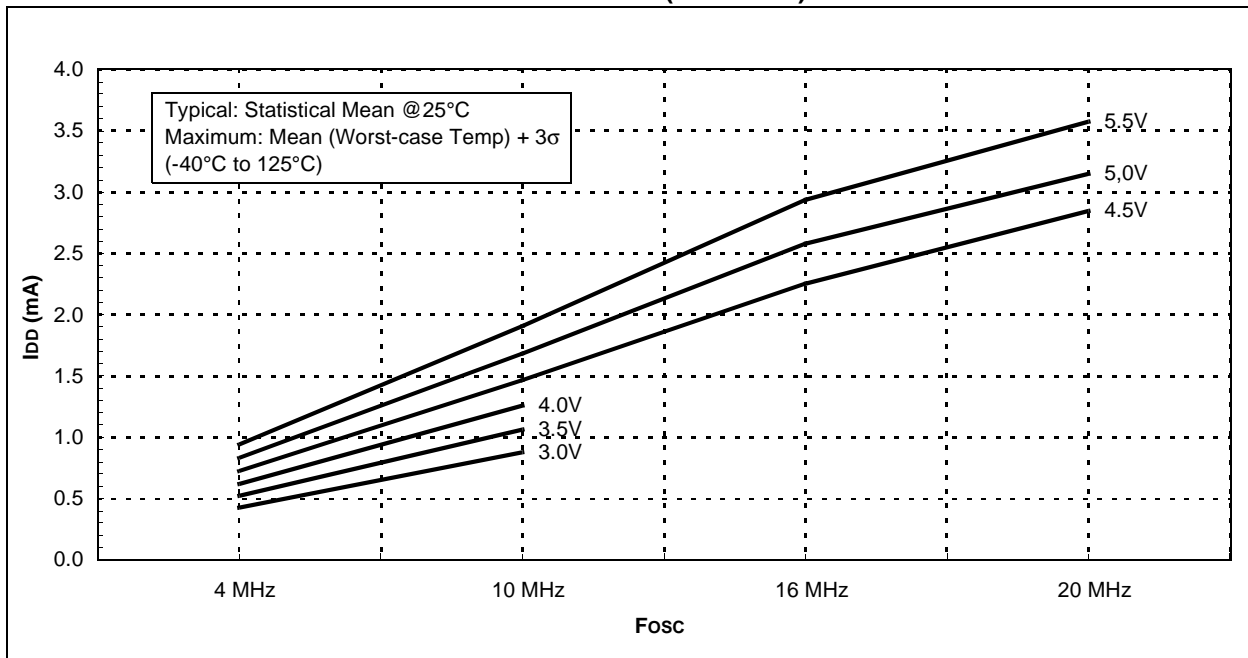
**FIGURE 15-1: TYPICAL  $I_{DD}$  vs.  $F_{osc}$  OVER  $V_{DD}$  (EC MODE)**



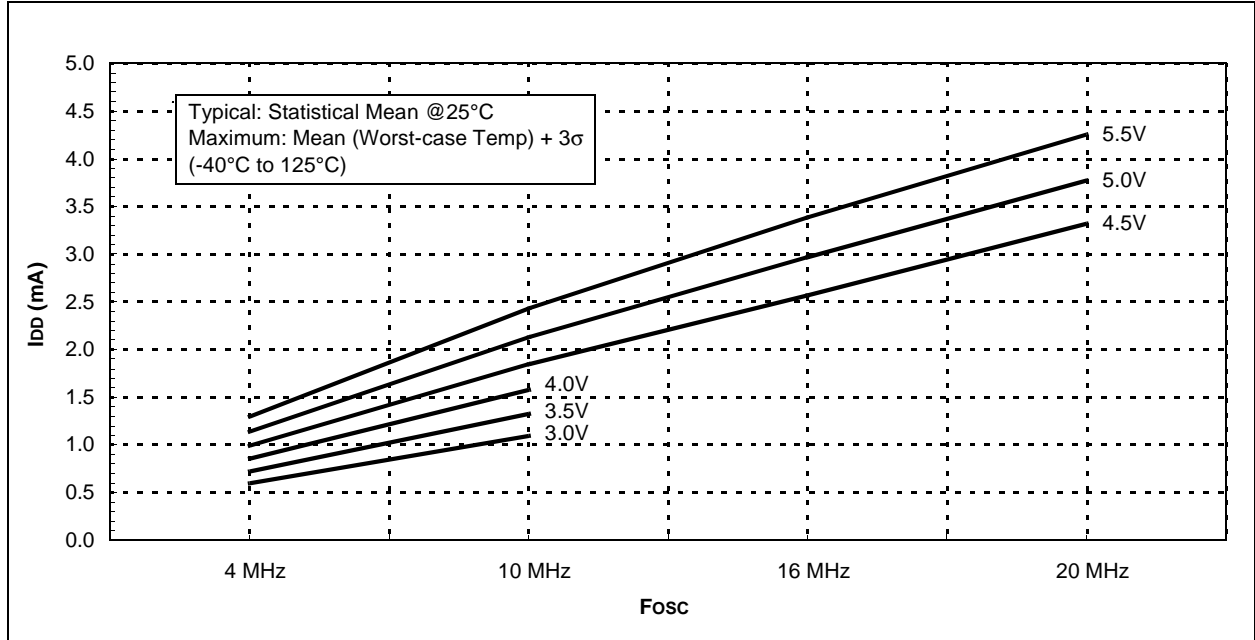
**FIGURE 15-2: MAXIMUM  $I_{DD}$  vs.  $F_{osc}$  OVER  $V_{DD}$  (EC MODE)**



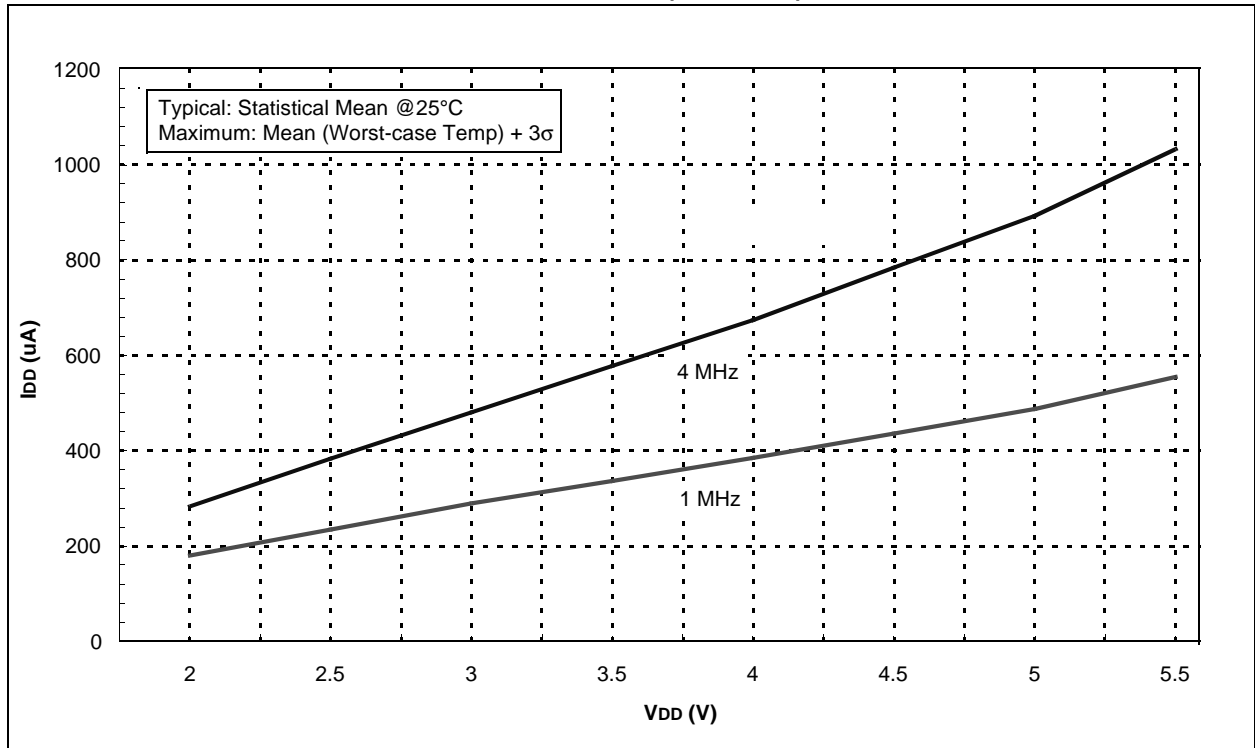
**FIGURE 15-3: TYPICAL  $I_{DD}$  vs.  $F_{osc}$  OVER  $V_{DD}$  (HS MODE)**



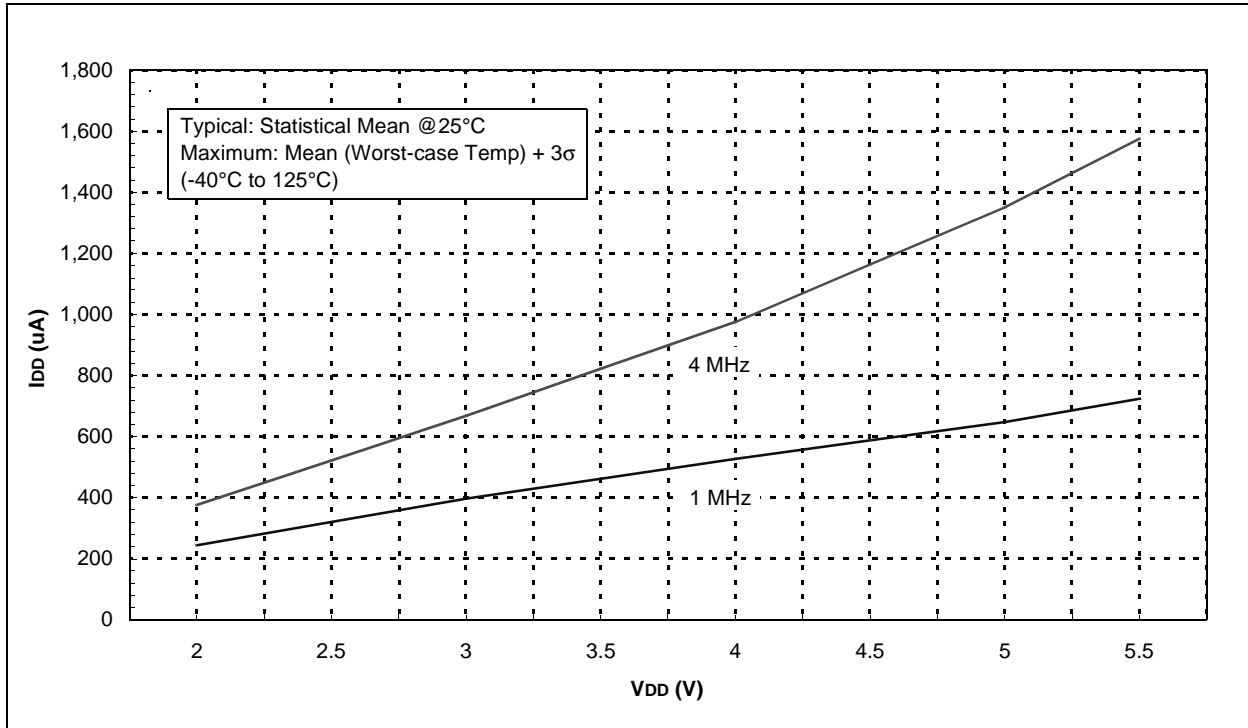
**FIGURE 15-4: MAXIMUM I<sub>DD</sub> vs. F<sub>OSC</sub> OVER V<sub>DD</sub> (HS MODE)**



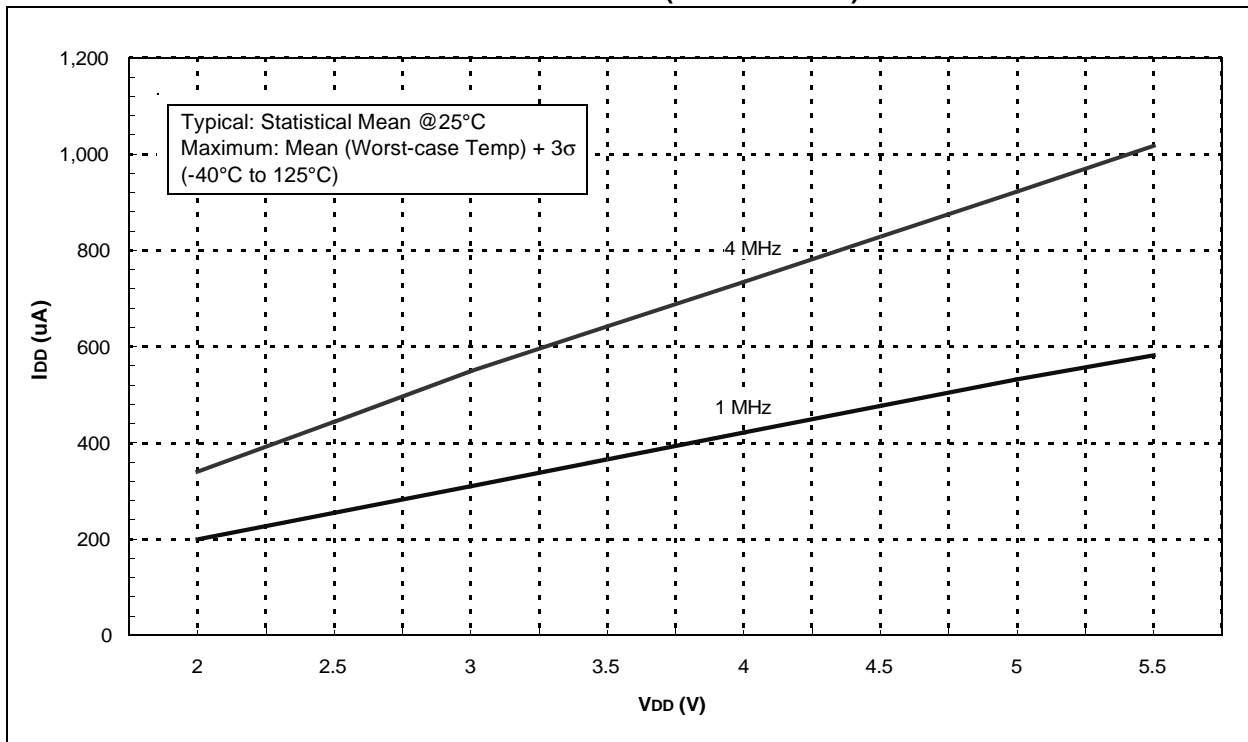
**FIGURE 15-5: TYPICAL I<sub>DD</sub> vs. V<sub>DD</sub> OVER F<sub>OSC</sub> (XT MODE)**



**FIGURE 15-6: MAXIMUM  $I_{DD}$  vs.  $V_{DD}$  OVER  $F_{osc}$  (XT MODE)**

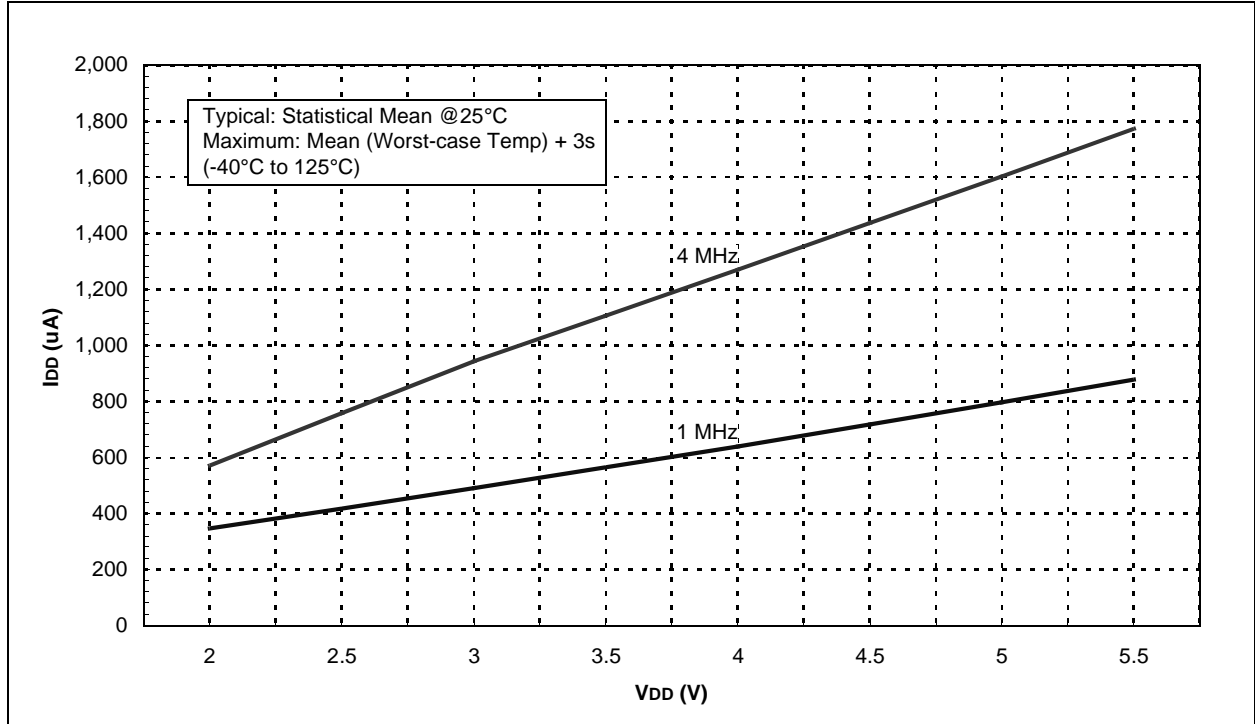


**FIGURE 15-7: TYPICAL  $I_{DD}$  vs.  $V_{DD}$  OVER  $F_{osc}$  (EXTRC MODE)**

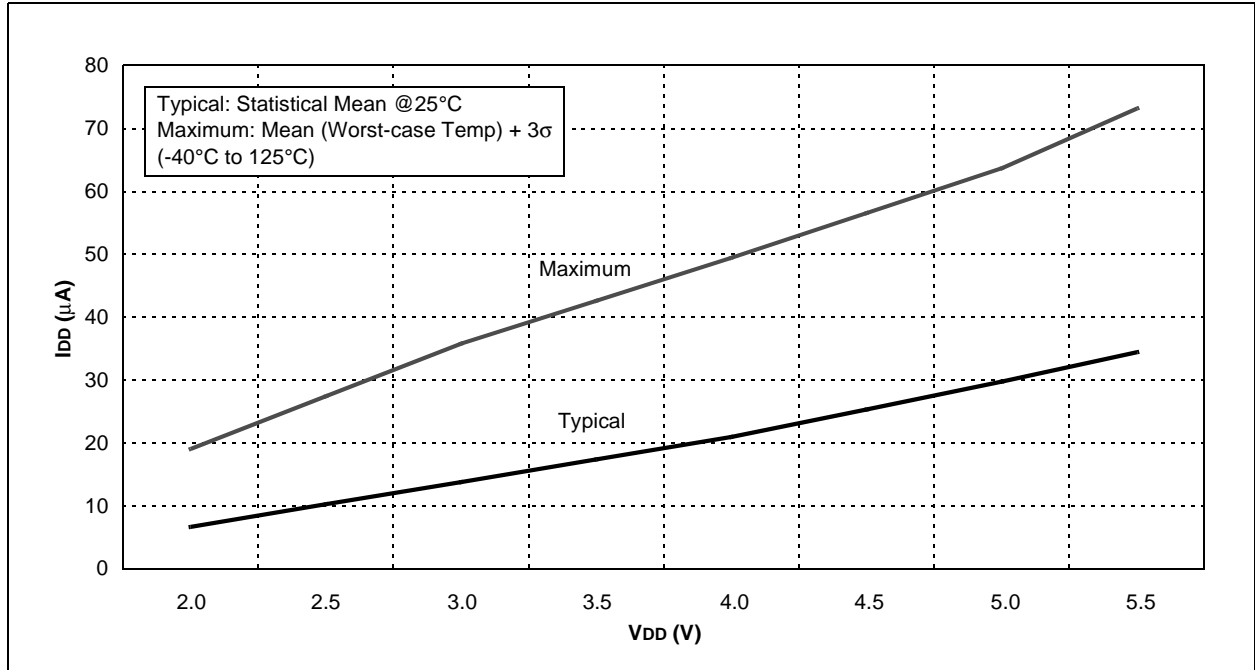




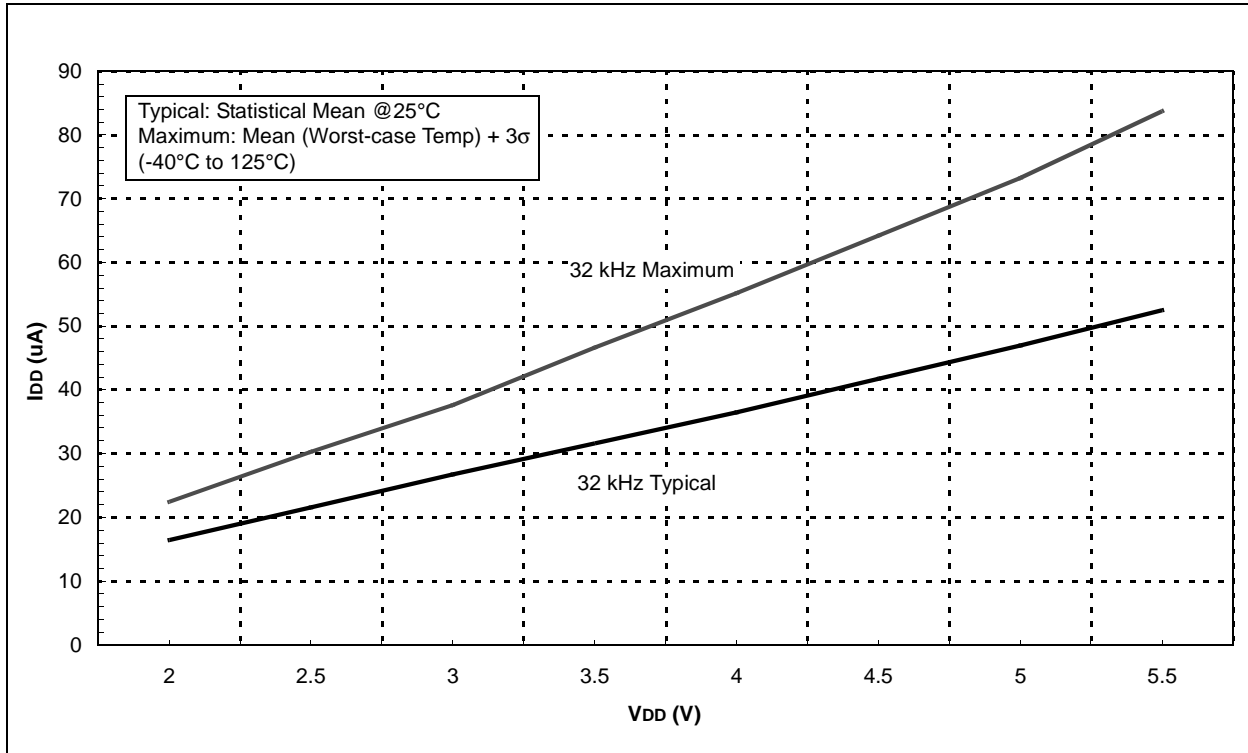
**FIGURE 15-8: MAXIMUM I<sub>DD</sub> vs. V<sub>DD</sub> (EXTRC MODE)**



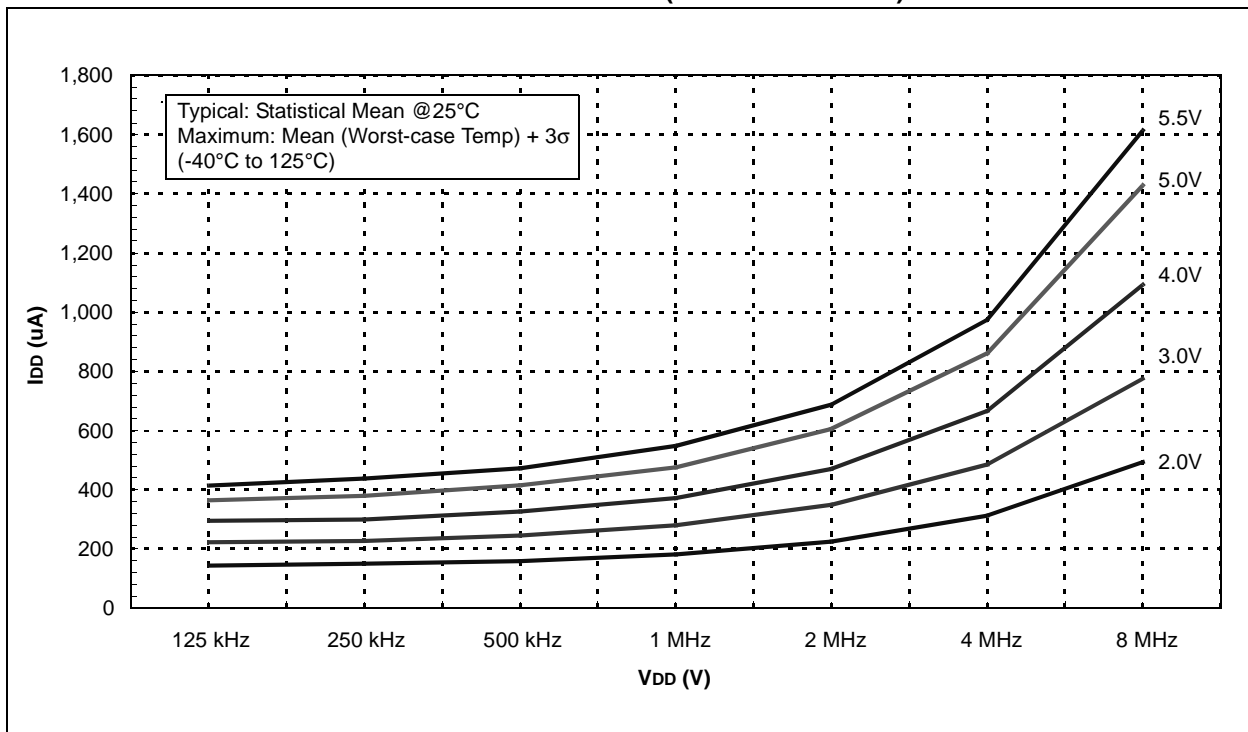
**FIGURE 15-9: I<sub>DD</sub> vs. V<sub>DD</sub> OVER F<sub>osc</sub> (LFINTOSC MODE, 31 kHz)**



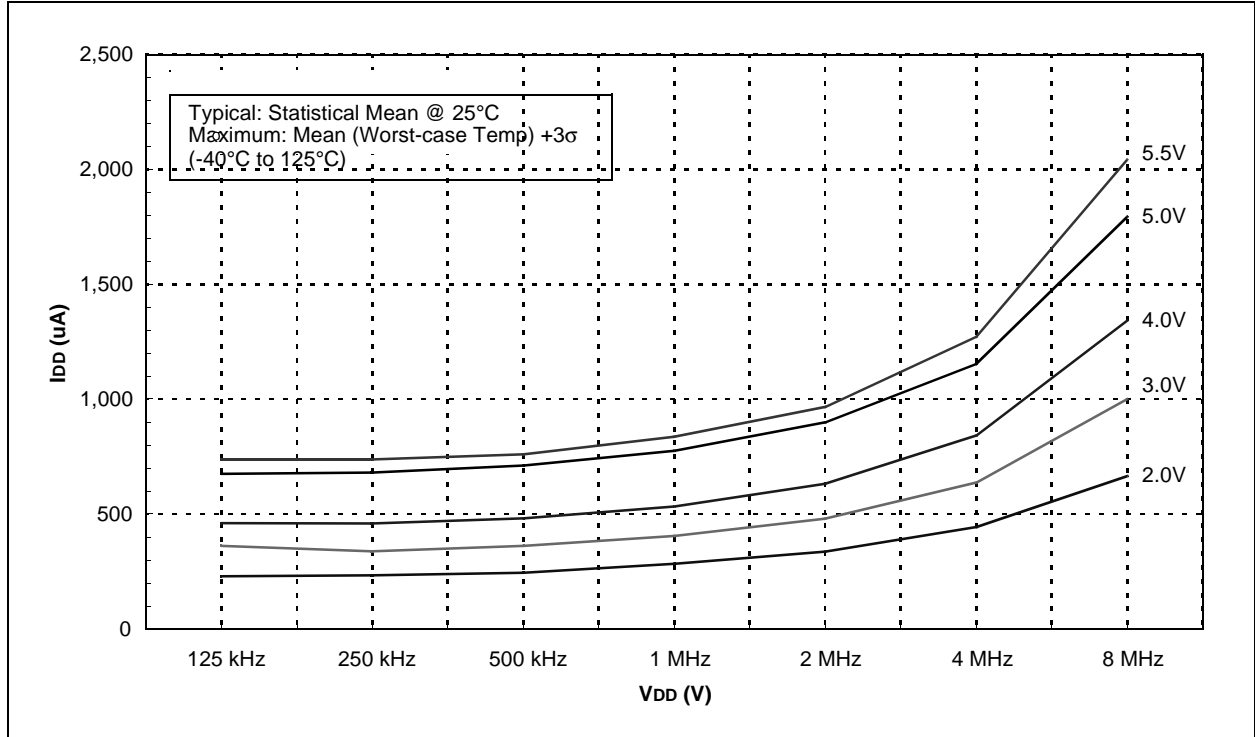
**FIGURE 15-10: I<sub>DD</sub> vs. V<sub>DD</sub> (LP MODE)**



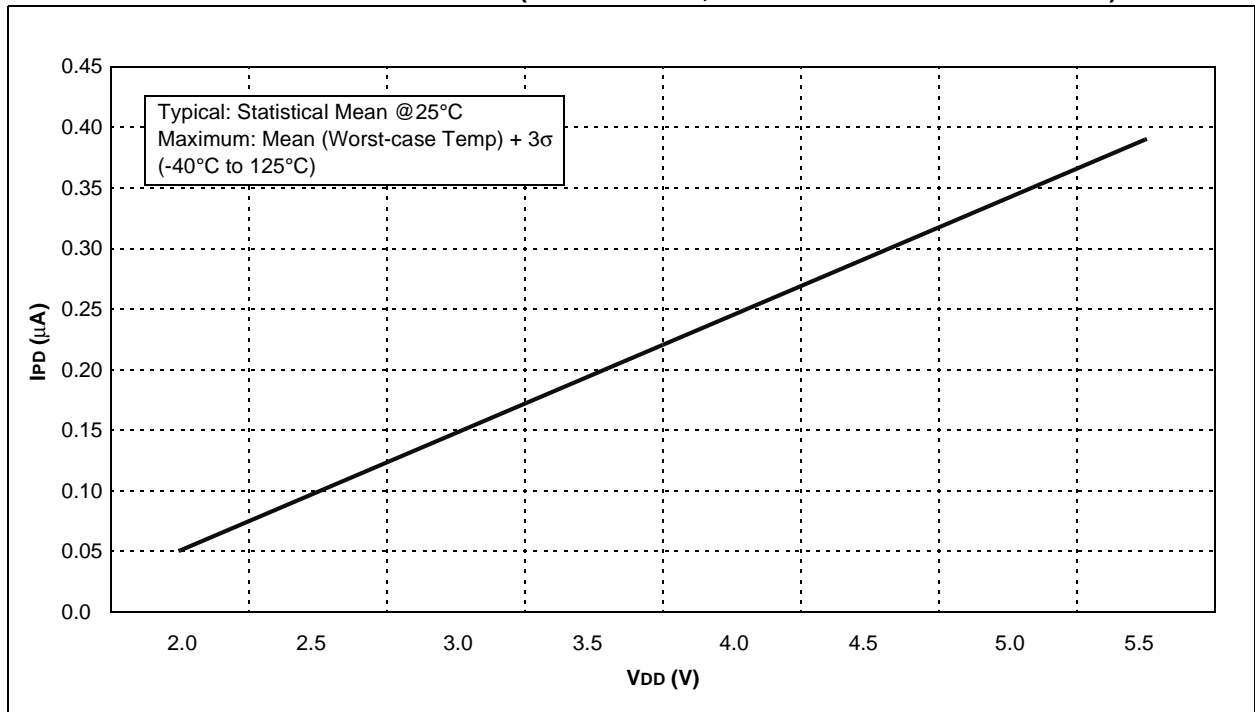
**FIGURE 15-11: TYPICAL I<sub>DD</sub> vs. F<sub>OSC</sub> OVER V<sub>DD</sub> (HFINTOSC MODE)**



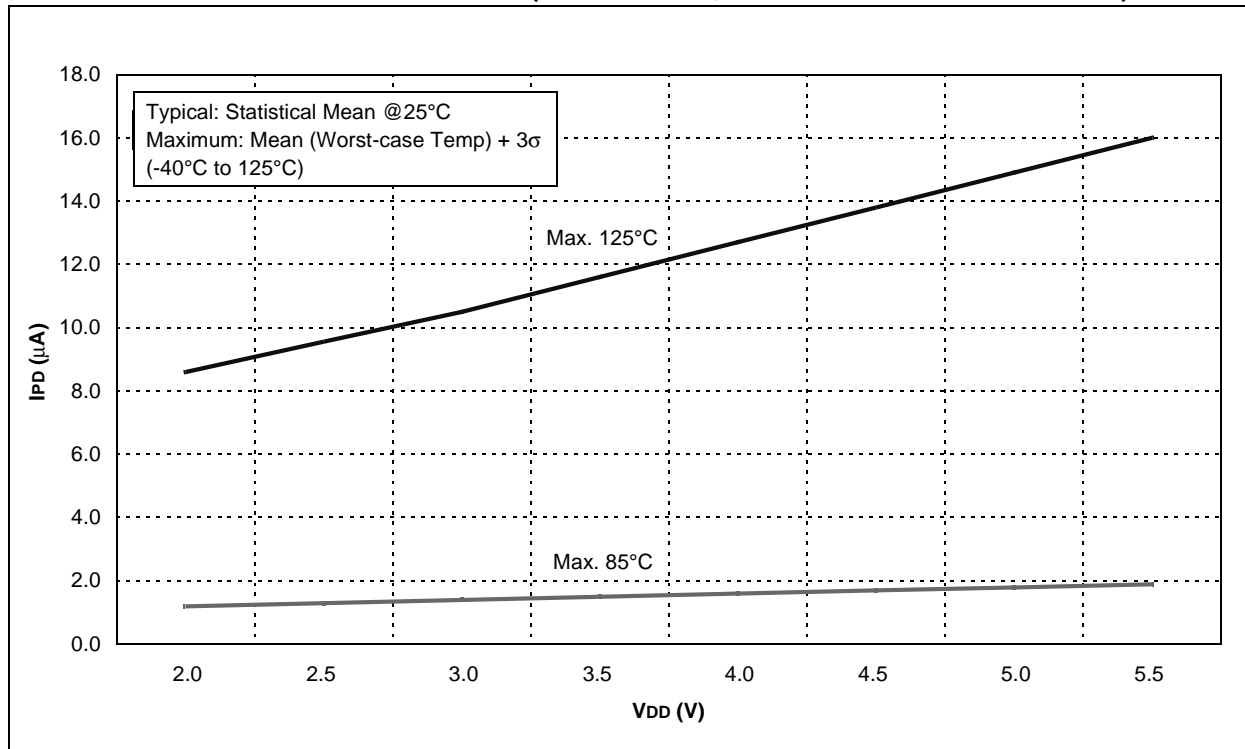
**FIGURE 15-12: MAXIMUM I<sub>DD</sub> vs. F<sub>OSC</sub> OVER V<sub>DD</sub> (HFINTOSC MODE)**



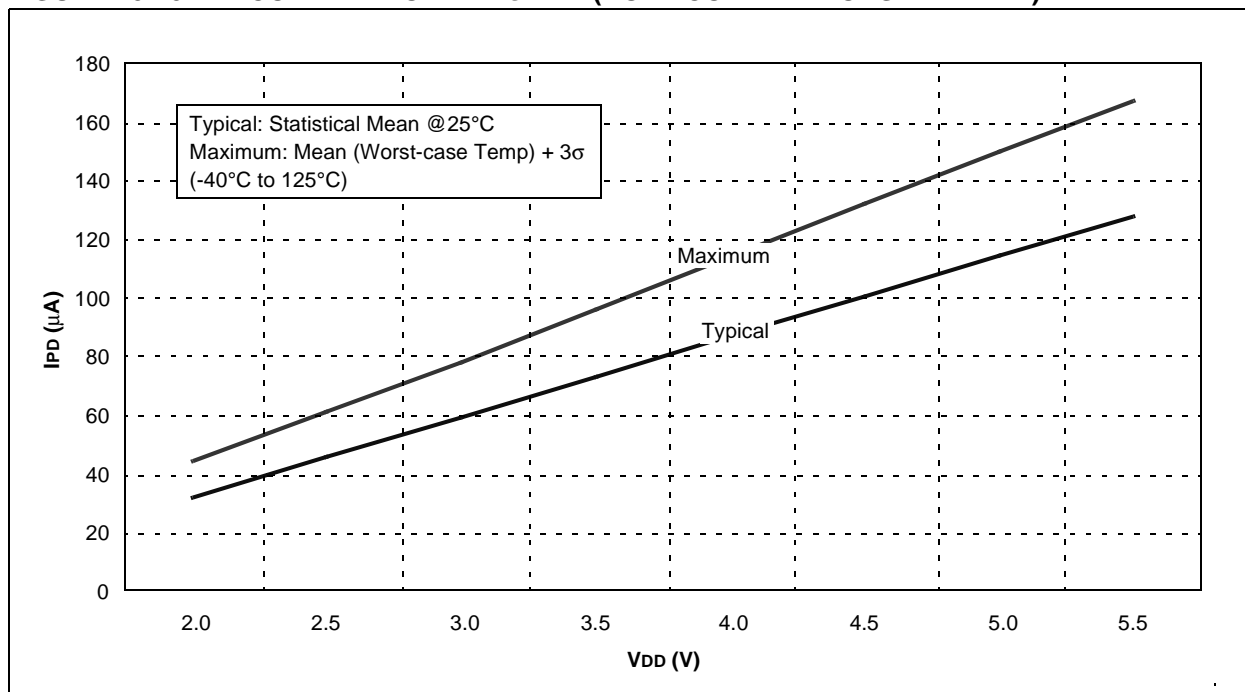
**FIGURE 15-13: TYPICAL I<sub>PD</sub> vs. V<sub>DD</sub> (SLEEP MODE, ALL PERIPHERALS DISABLED)**



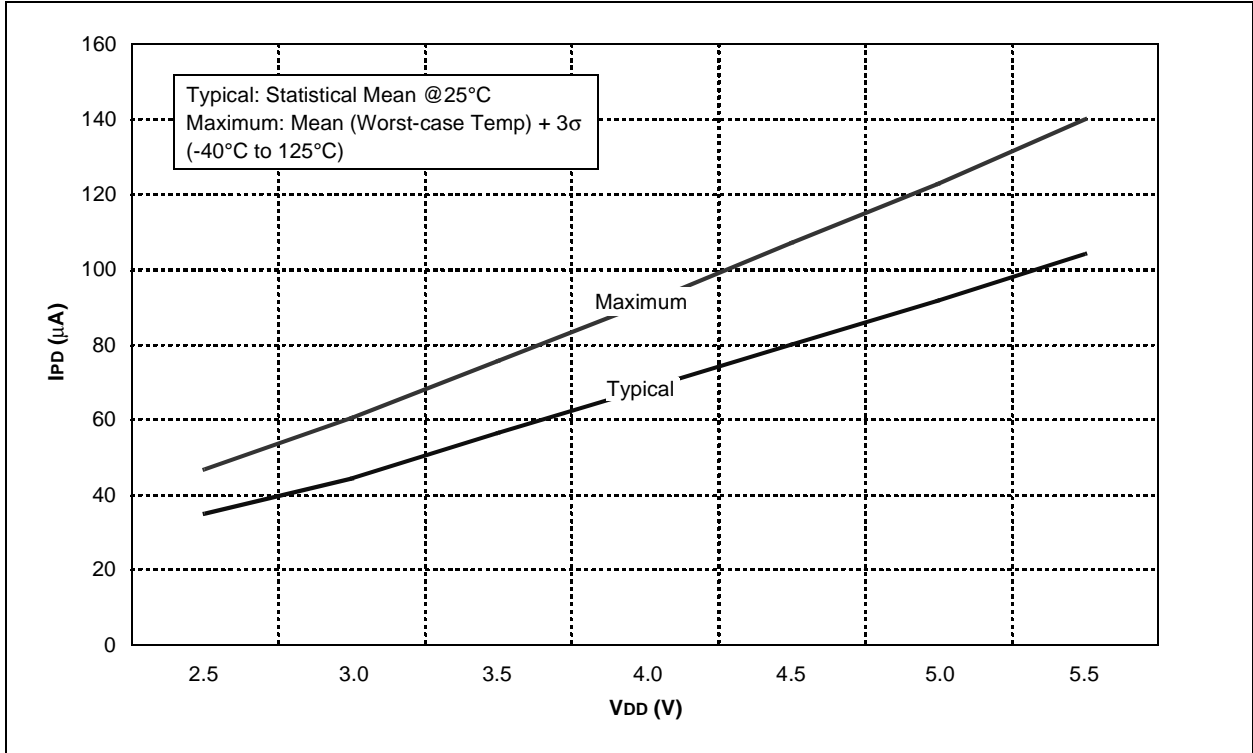
**FIGURE 15-14: MAXIMUM IPD vs. VDD (SLEEP MODE, ALL PERIPHERALS DISABLED)**



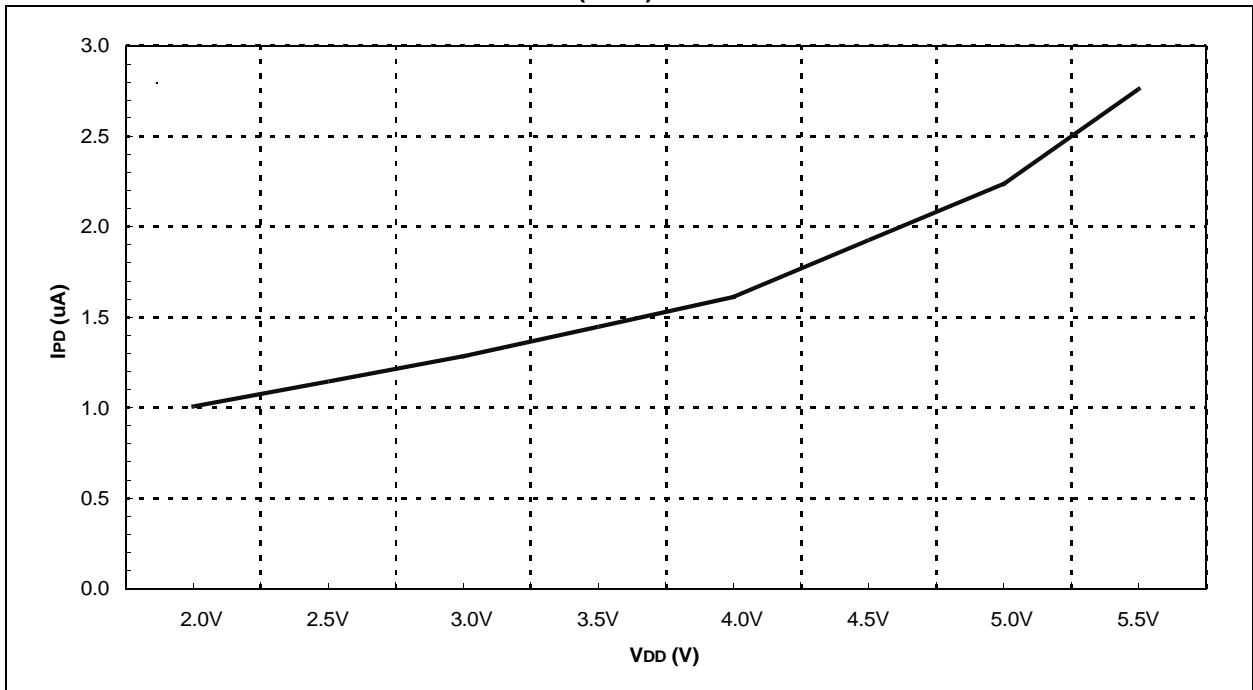
**FIGURE 15-15: COMPARATOR IPD vs. VDD (BOTH COMPARATORS ENABLED)**



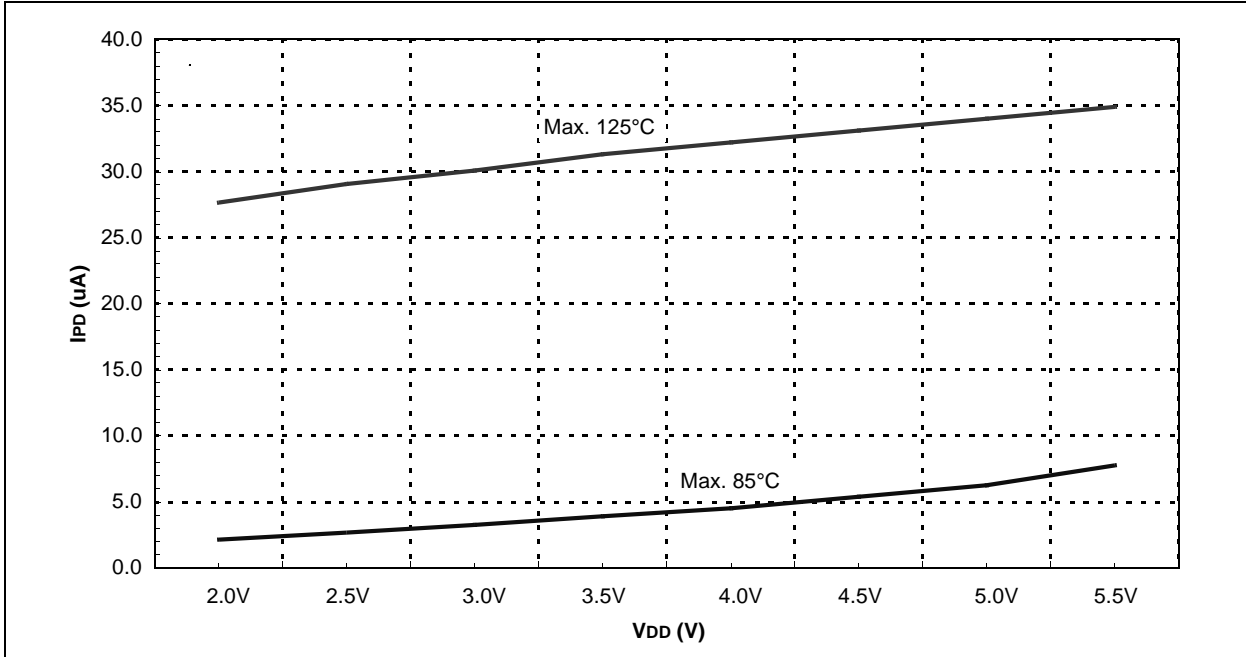
**FIGURE 15-16: BOR IPD vs. VDD OVER TEMPERATURE**



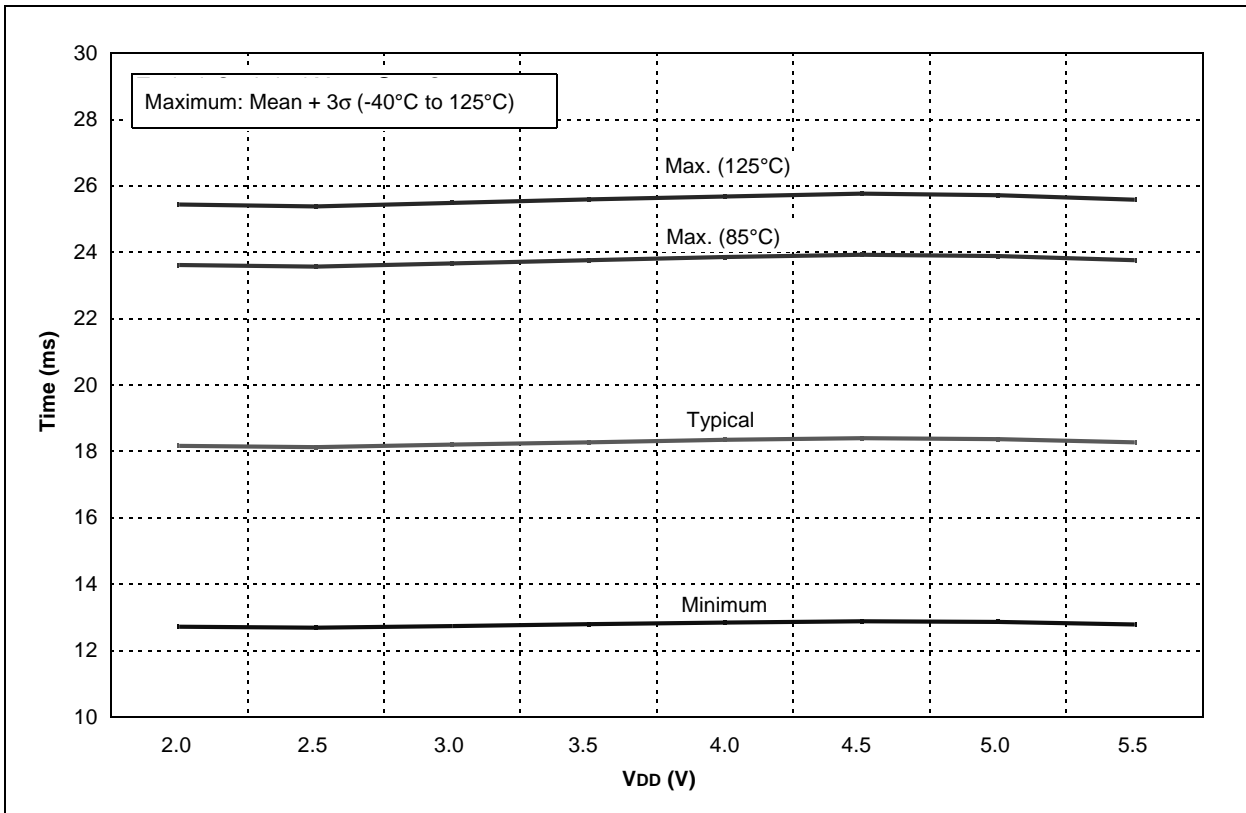
**FIGURE 15-17: TYPICAL WDT IPD vs. VDD (25°C)**



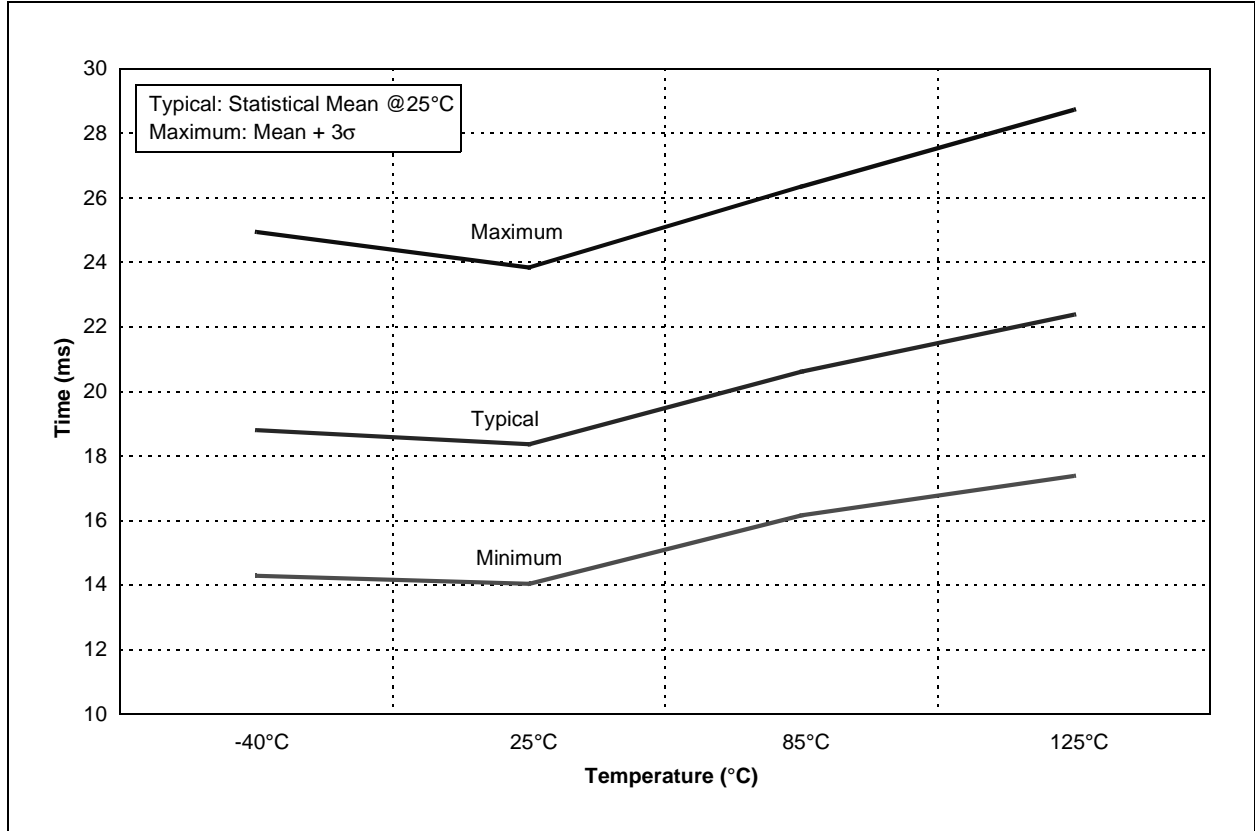
**FIGURE 15-18: MAXIMUM WDT IPD vs. VDD OVER TEMPERATURE**



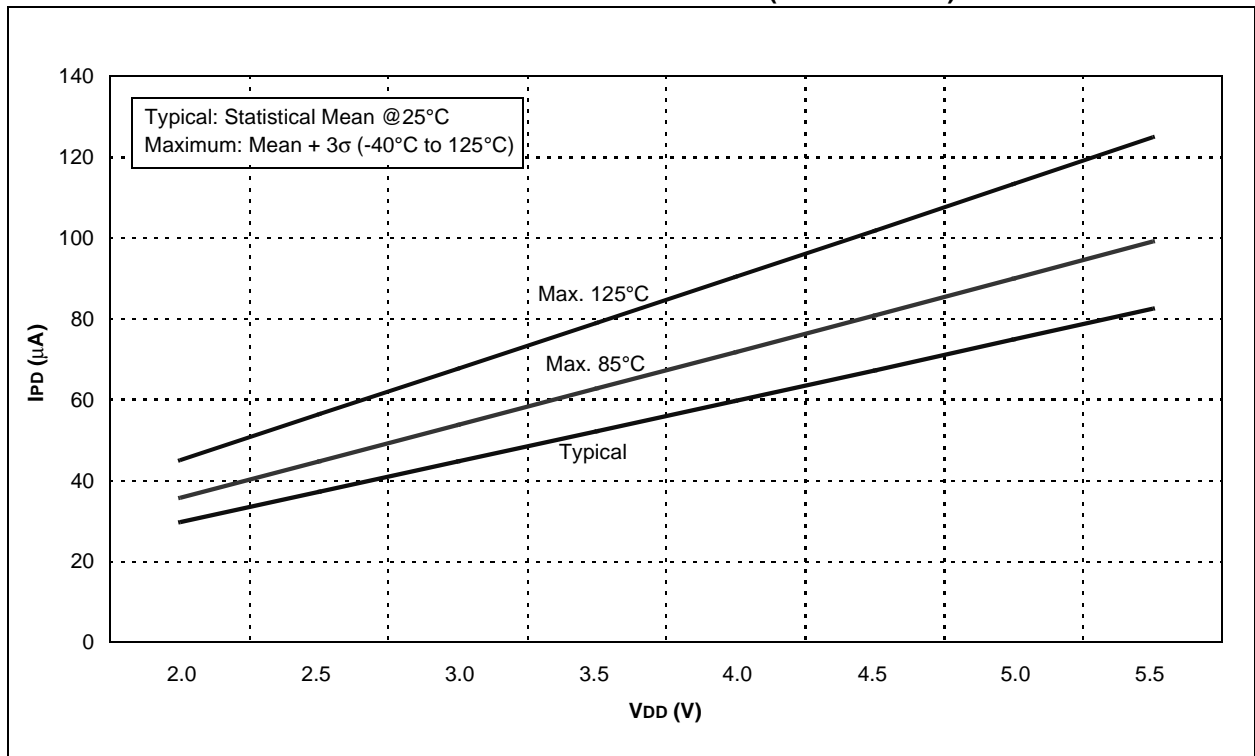
**FIGURE 15-19: WDT PERIOD vs. VDD OVER TEMPERATURE**



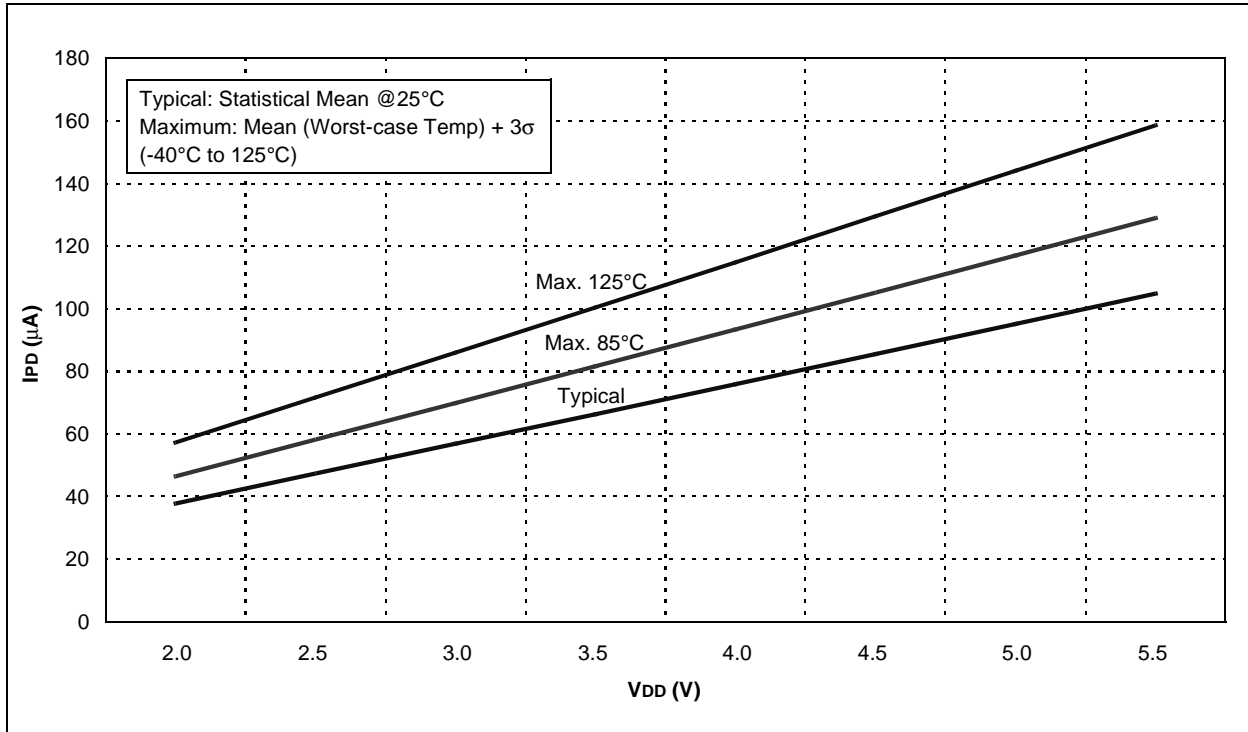
**FIGURE 15-20: WDT PERIOD vs. TEMPERATURE**



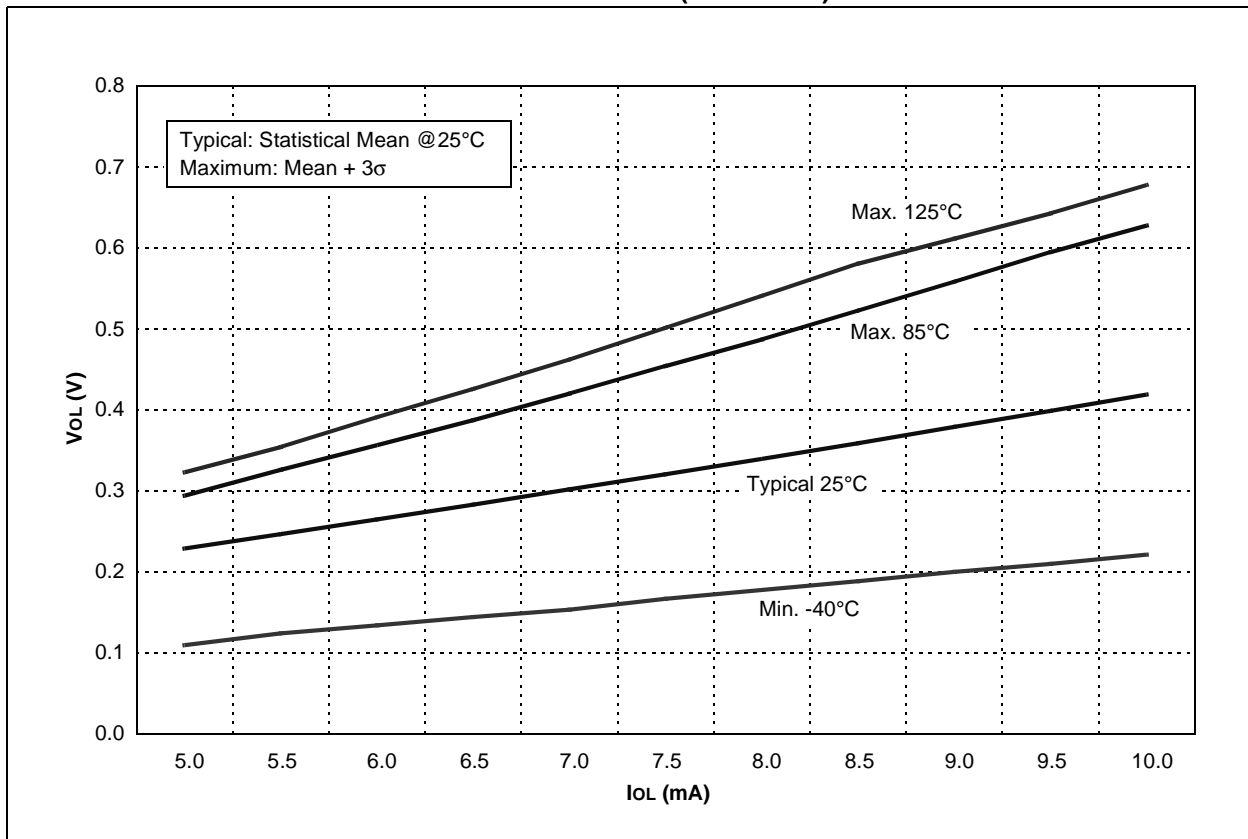
**FIGURE 15-21: CVREF IPD vs. VDD OVER TEMPERATURE (HIGH RANGE)**



**FIGURE 15-22: CVREF IPD vs. VDD OVER TEMPERATURE (LOW RANGE)**

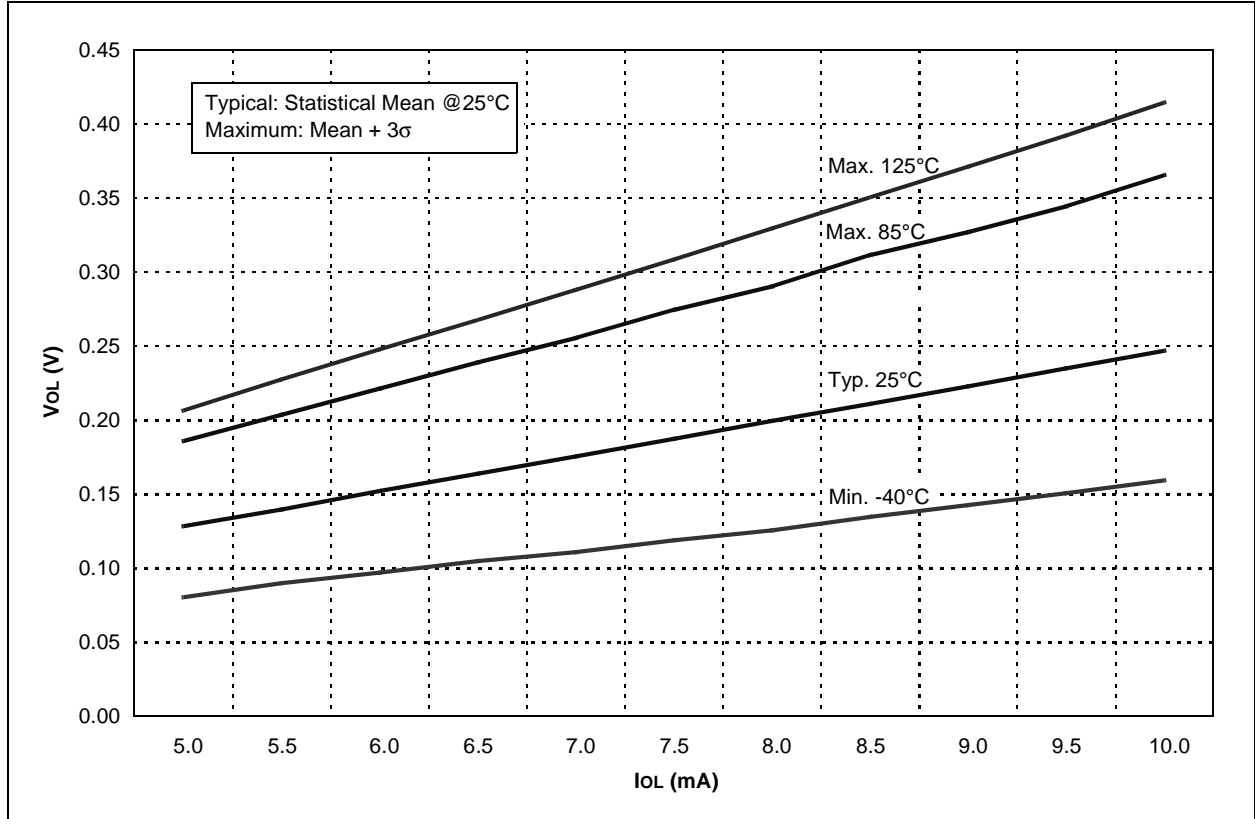


**FIGURE 15-23: VOL vs. IOL OVER TEMPERATURE (VDD = 3.0V)**

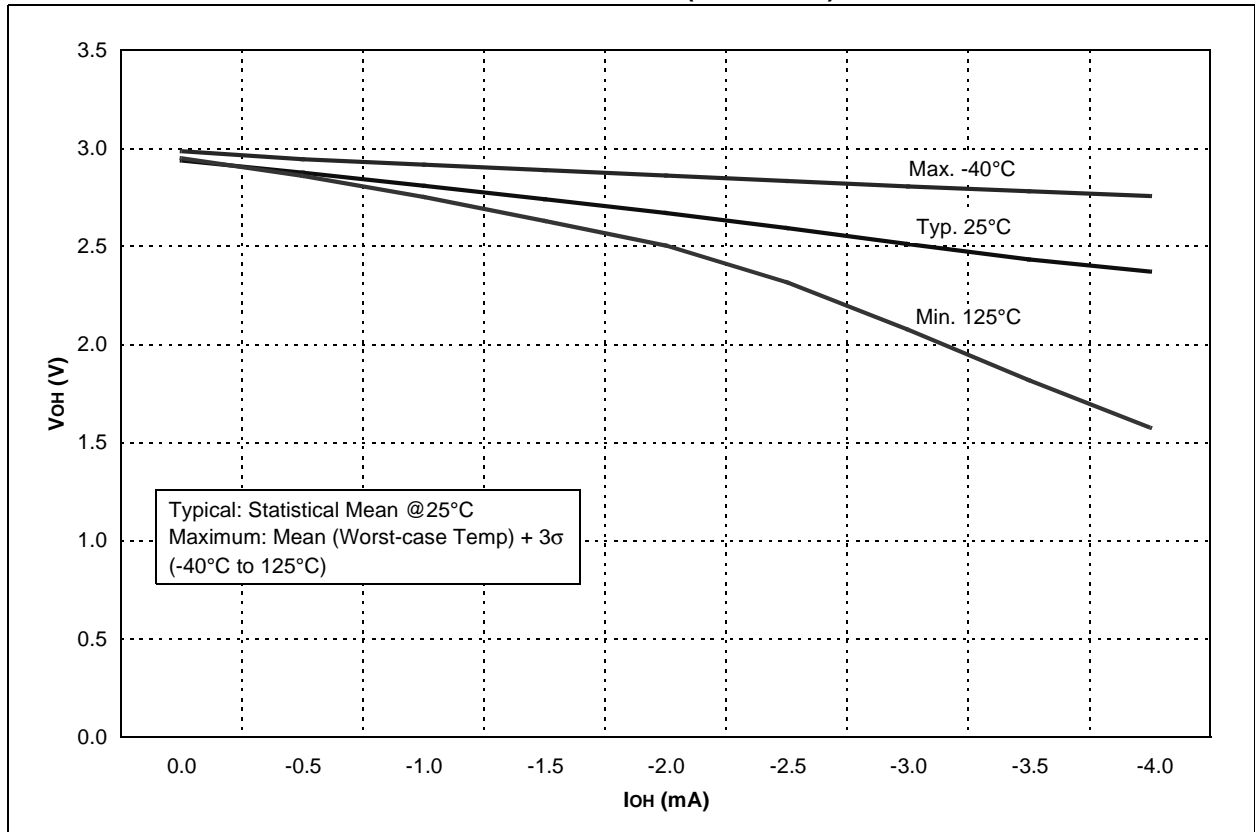




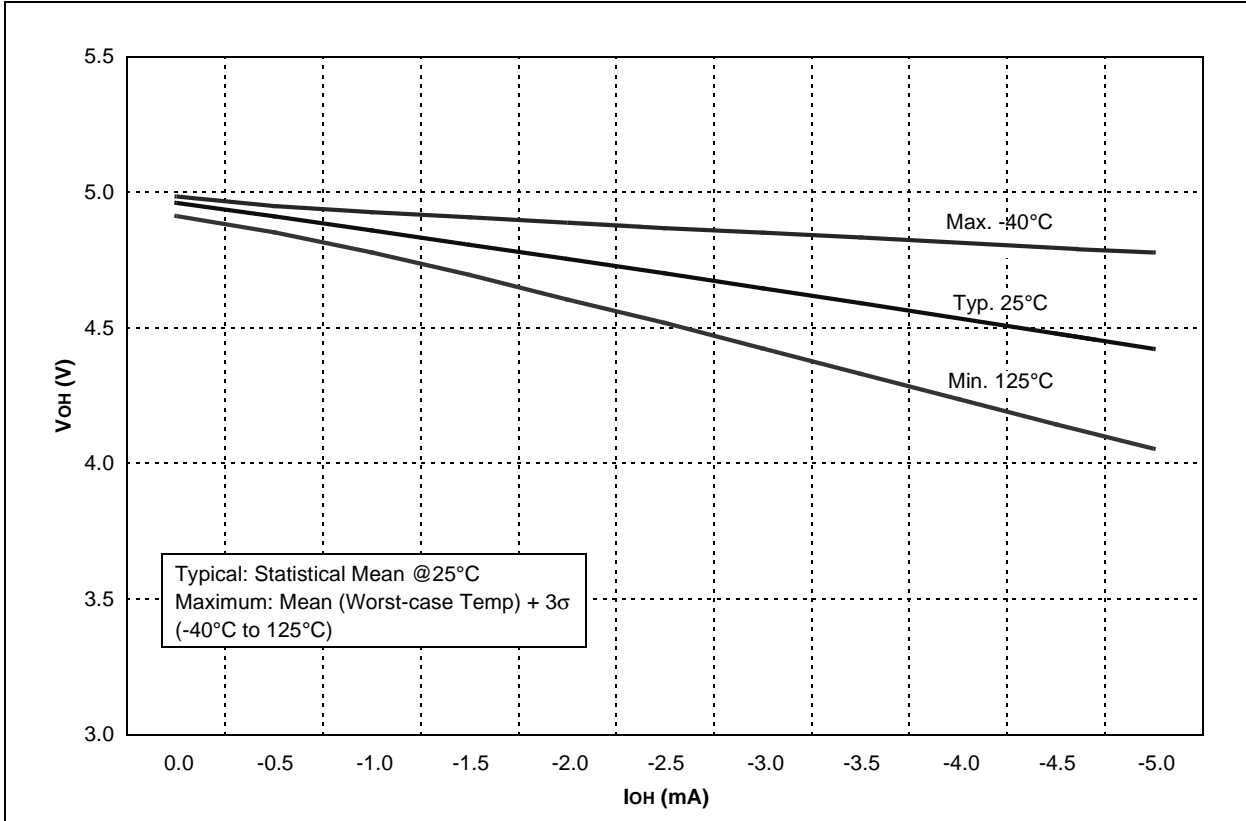
**FIGURE 15-24:  $V_{OL}$  vs.  $I_{OL}$  OVER TEMPERATURE ( $V_{DD} = 5.0V$ )**



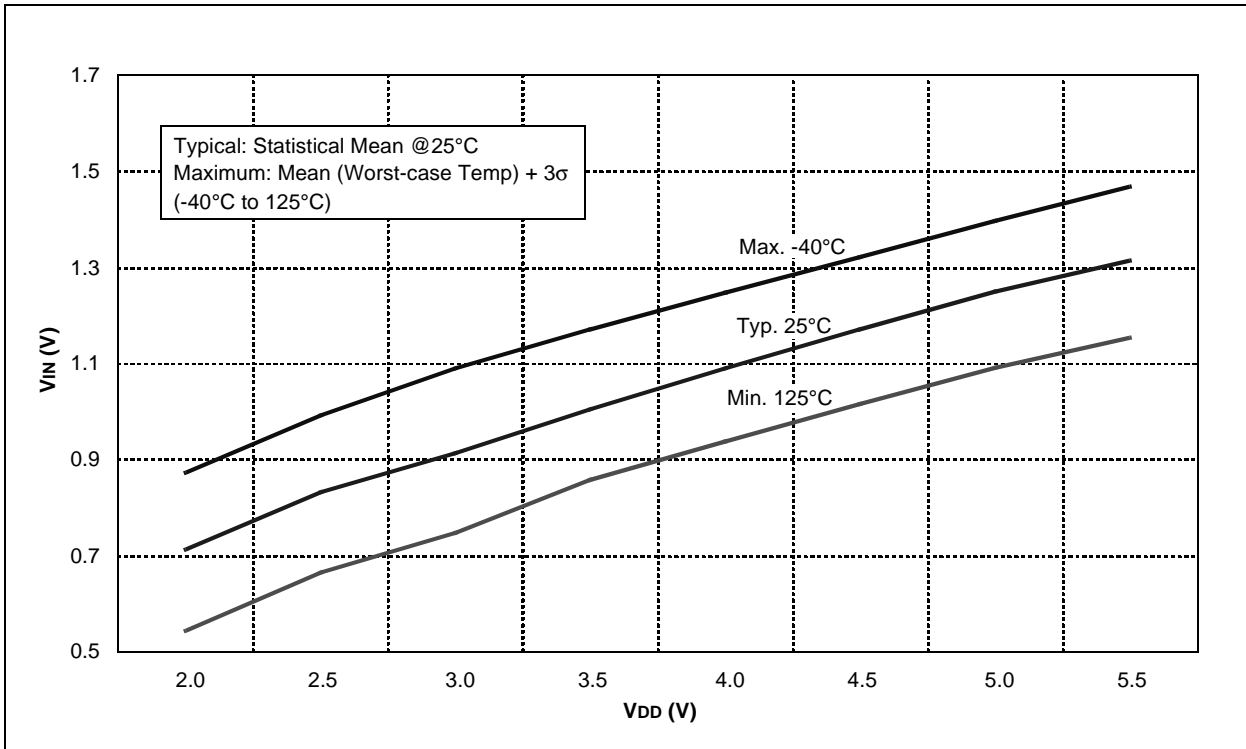
**FIGURE 15-25:  $V_{OH}$  vs.  $I_{OH}$  OVER TEMPERATURE ( $V_{DD} = 3.0V$ )**



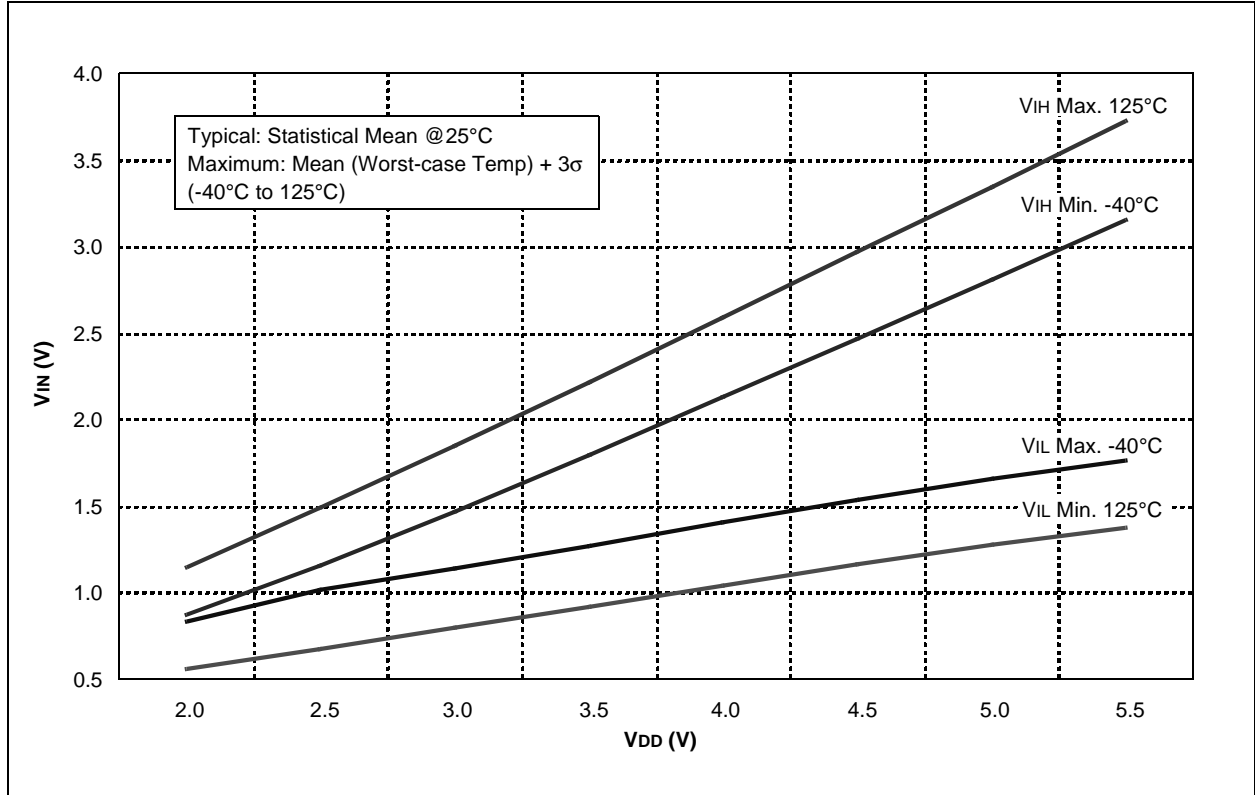
**FIGURE 15-26:  $V_{OH}$  vs.  $I_{OH}$  OVER TEMPERATURE ( $V_{DD} = 5.0V$ )**



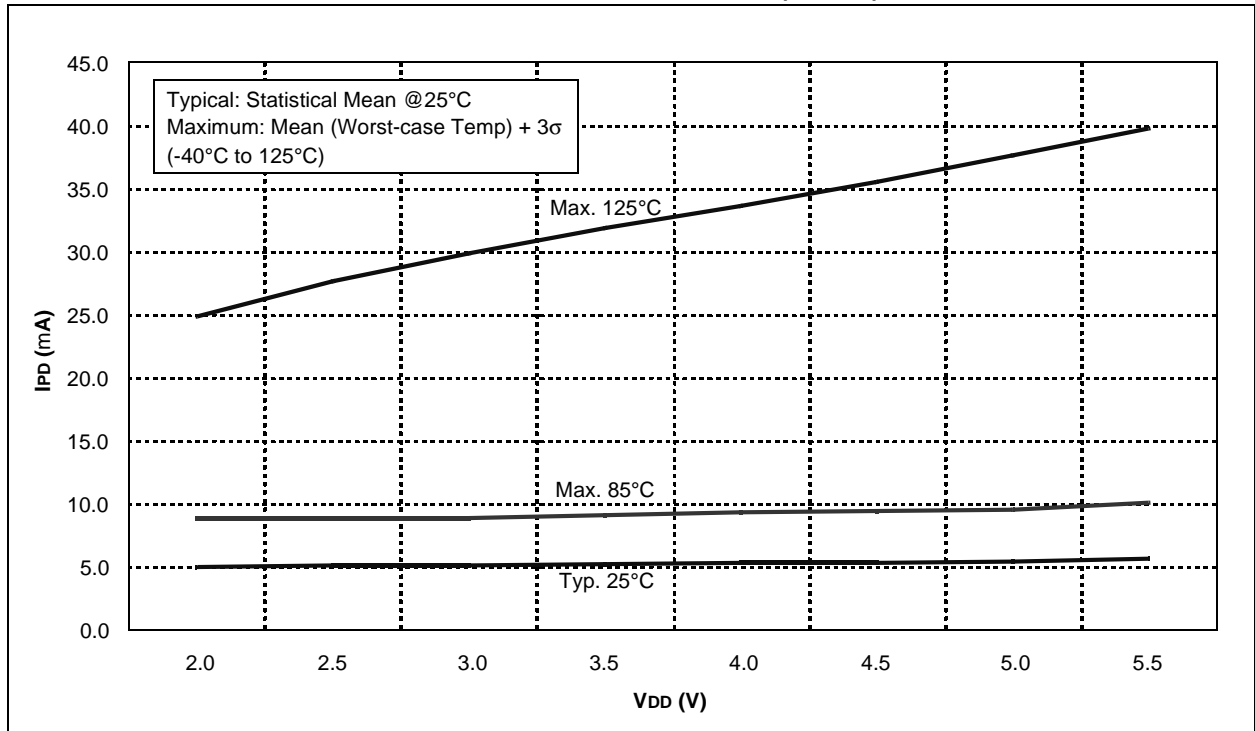
**FIGURE 15-27: TTL INPUT THRESHOLD  $V_{IN}$  vs.  $V_{DD}$  OVER TEMPERATURE**



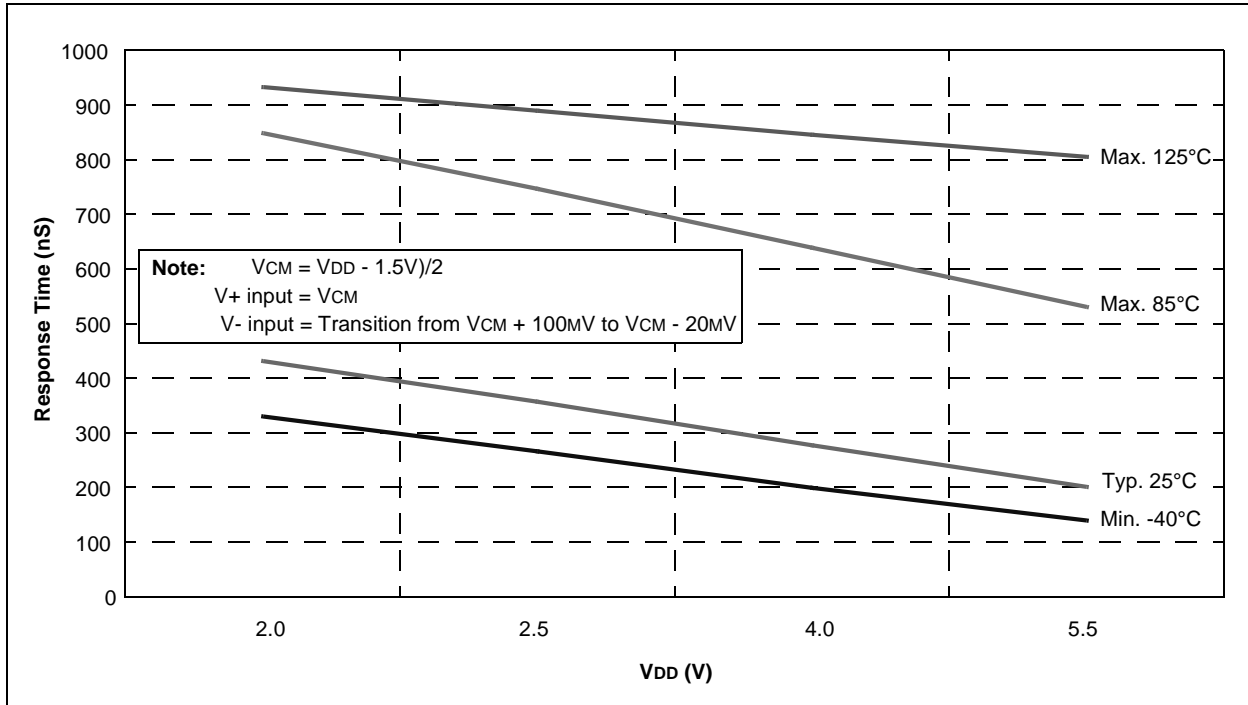
**FIGURE 15-28: SCHMITT TRIGGER INPUT THRESHOLD  $V_{IN}$  vs.  $V_{DD}$  OVER TEMPERATURE**



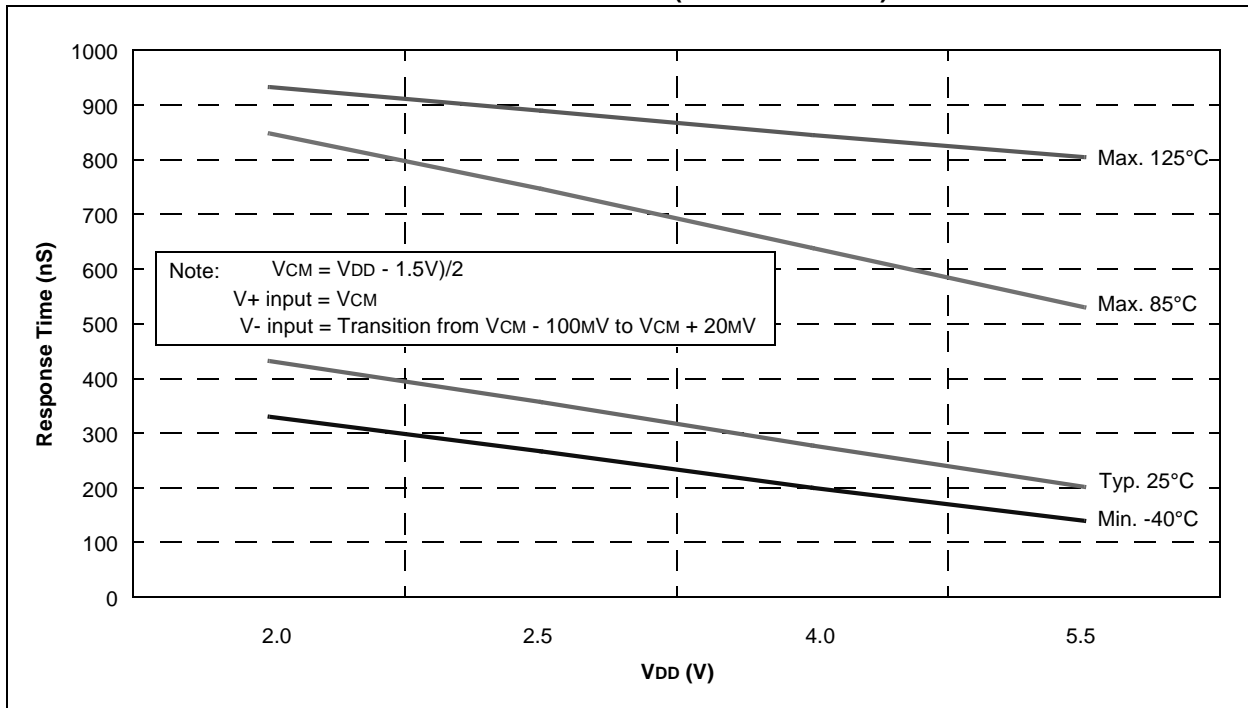
**FIGURE 15-29: T1OSC  $I_{PD}$  vs.  $V_{DD}$  OVER TEMPERATURE (32 kHz)**



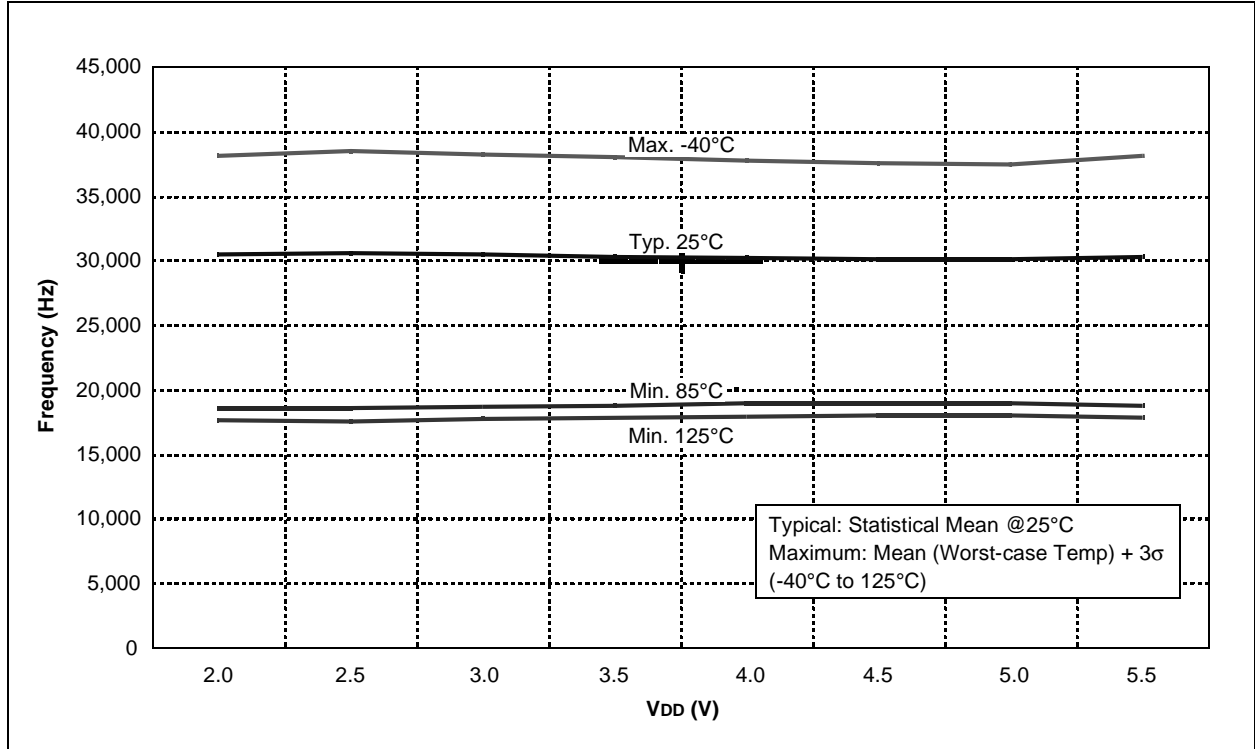
**FIGURE 15-30: COMPARATOR RESPONSE TIME (RISING EDGE)**



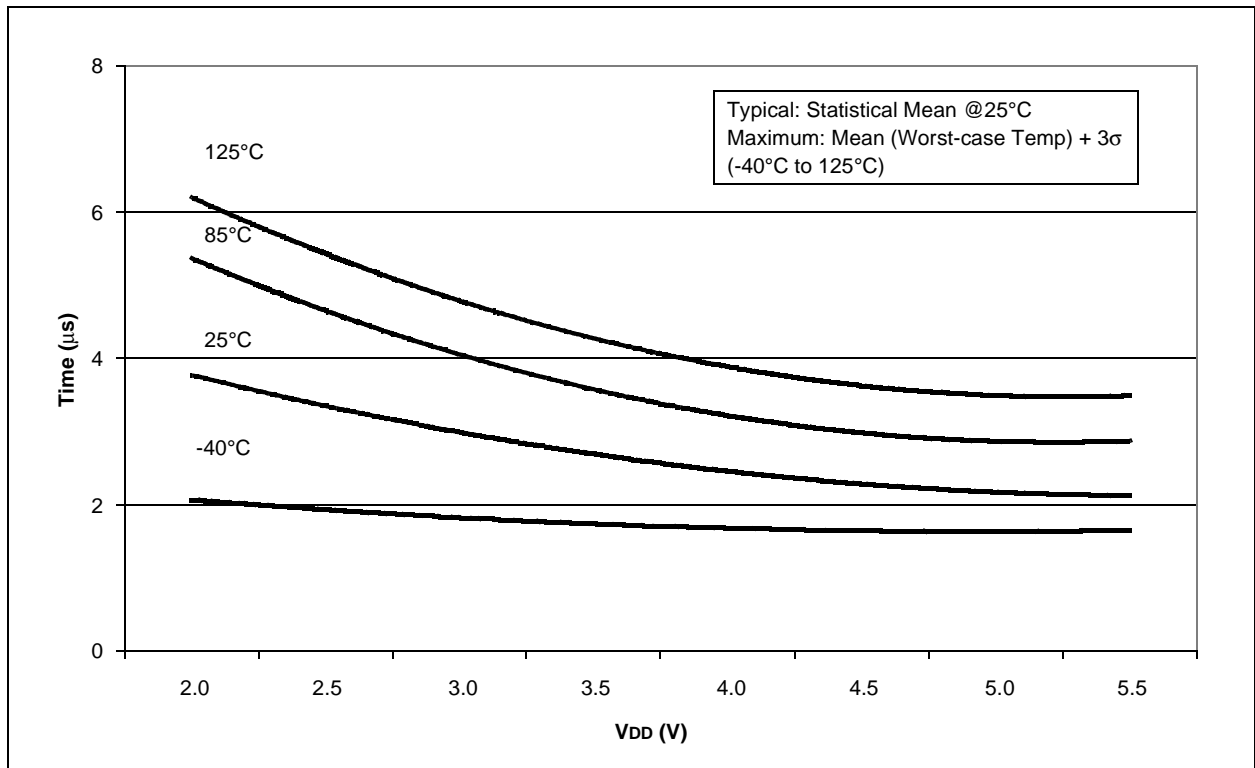
**FIGURE 15-31: COMPARATOR RESPONSE TIME (FALLING EDGE)**



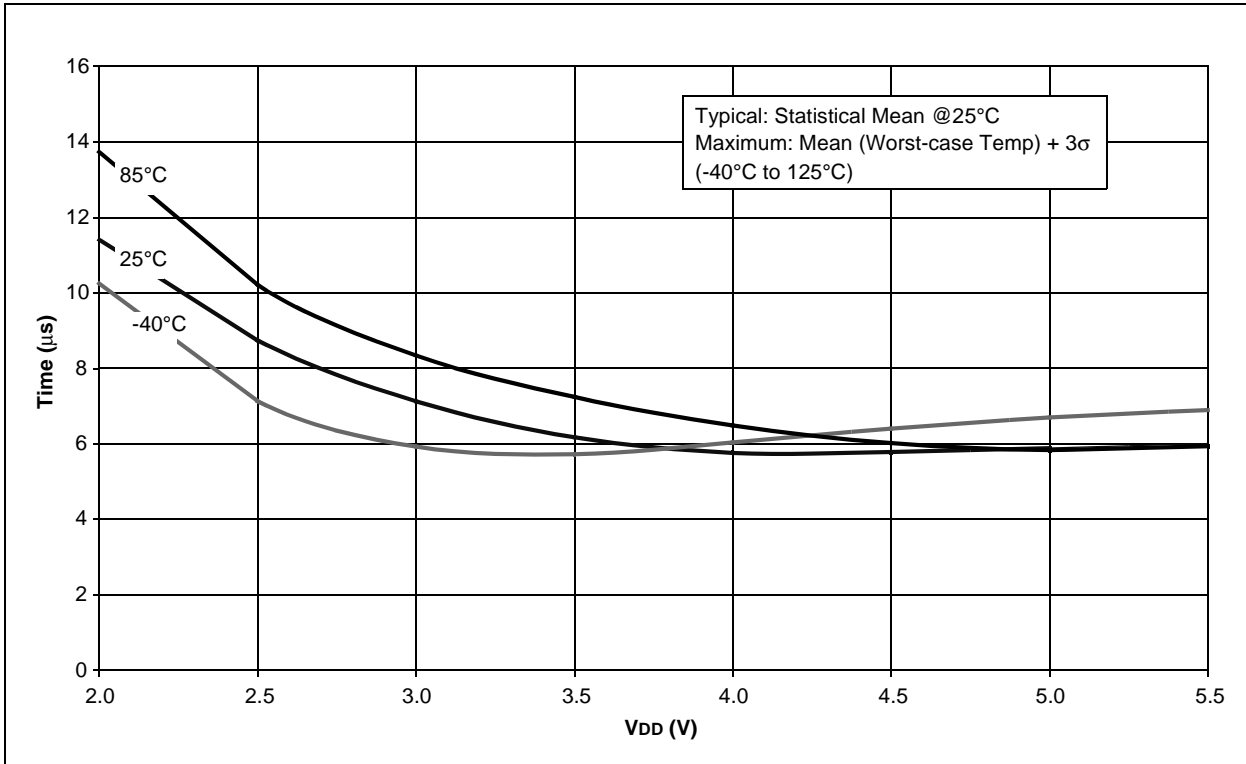
**FIGURE 15-32: LFINTOSC FREQUENCY vs. V<sub>DD</sub> OVER TEMPERATURE (31 kHz)**



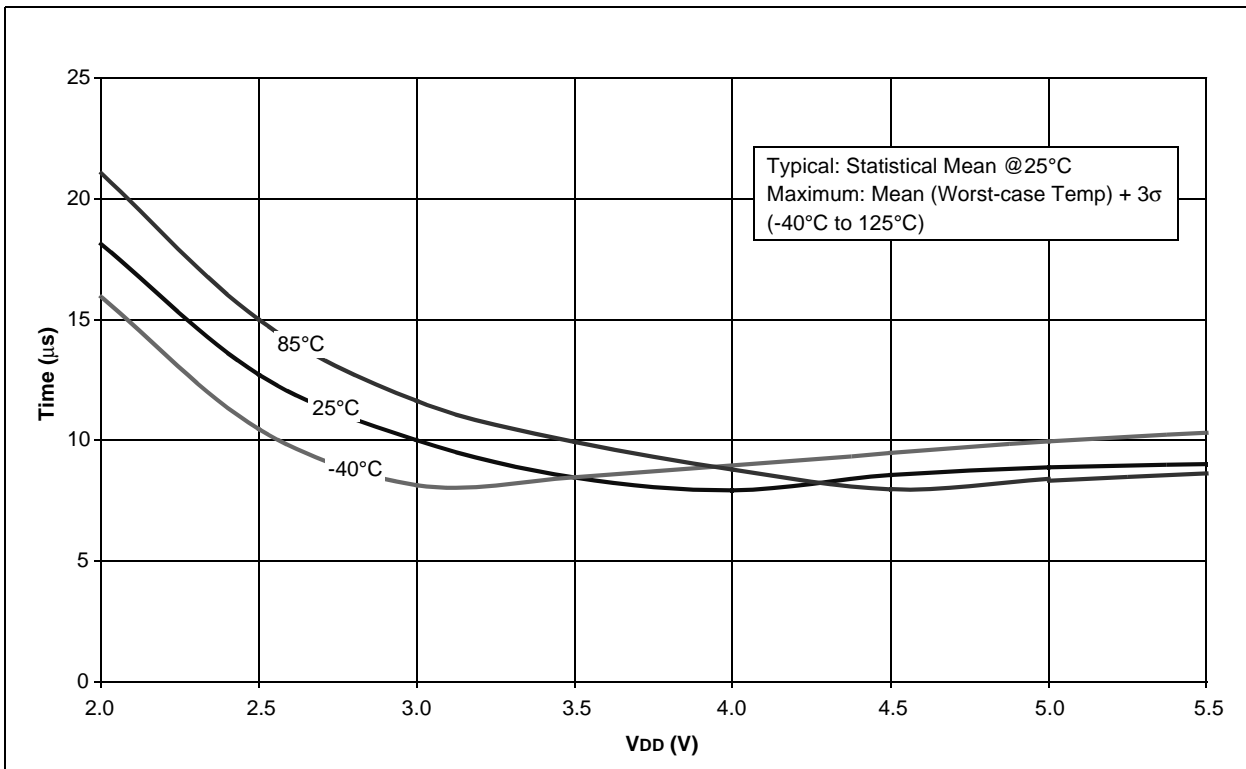
**FIGURE 15-33: ADC CLOCK PERIOD vs. V<sub>DD</sub> OVER TEMPERATURE**



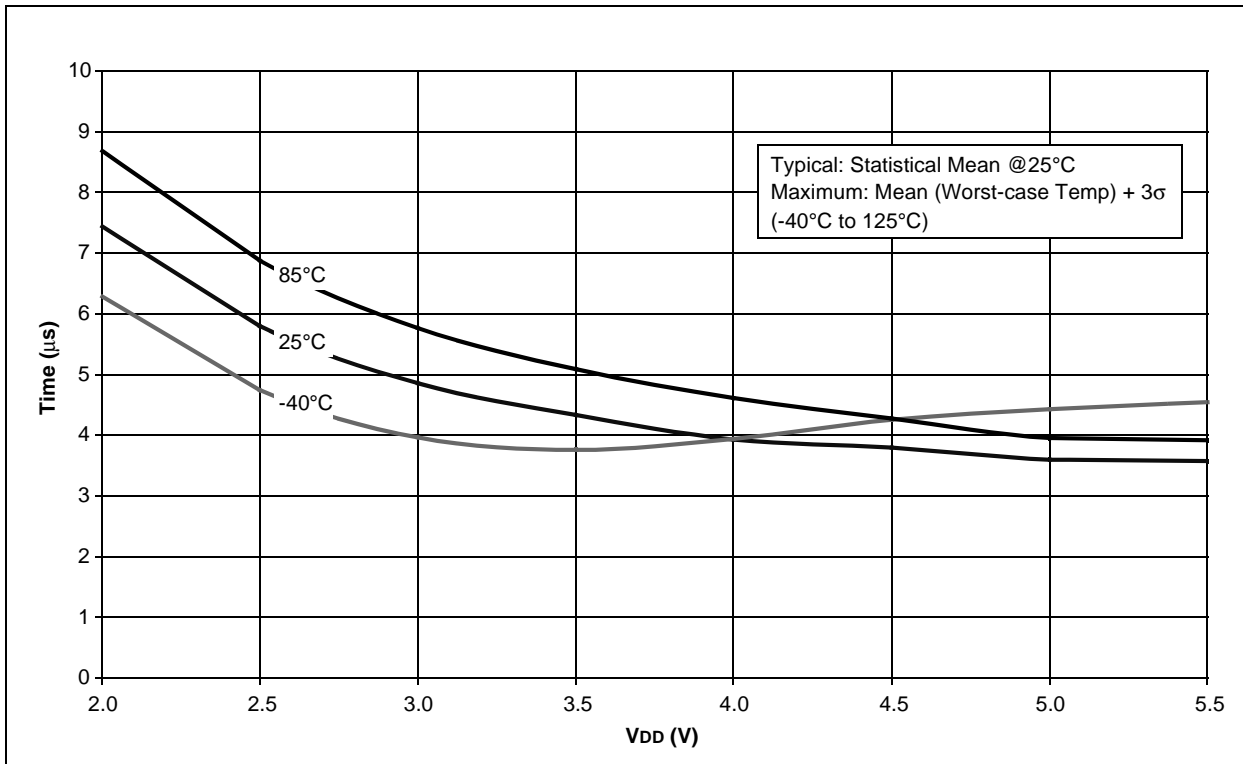
**FIGURE 15-34: TYPICAL HFINTOSC START-UP TIMES vs. V<sub>DD</sub> OVER TEMPERATURE**



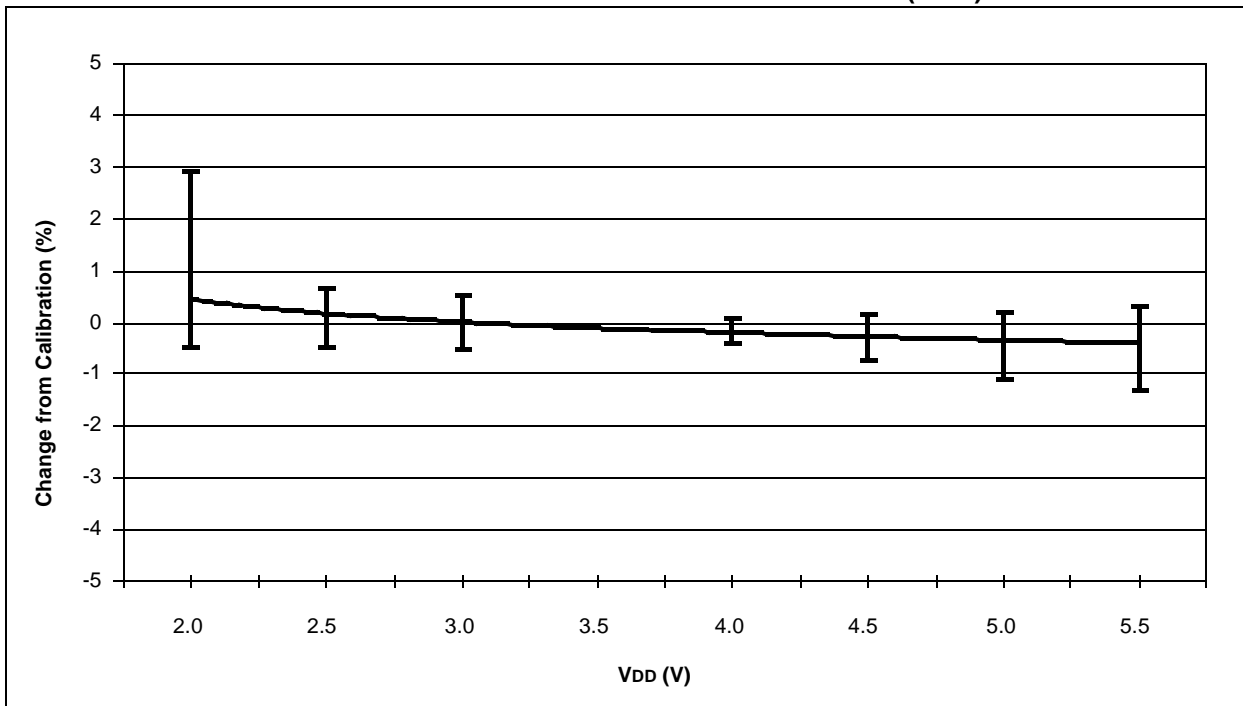
**FIGURE 15-35: MAXIMUM HFINTOSC START-UP TIMES vs. V<sub>DD</sub> OVER TEMPERATURE**



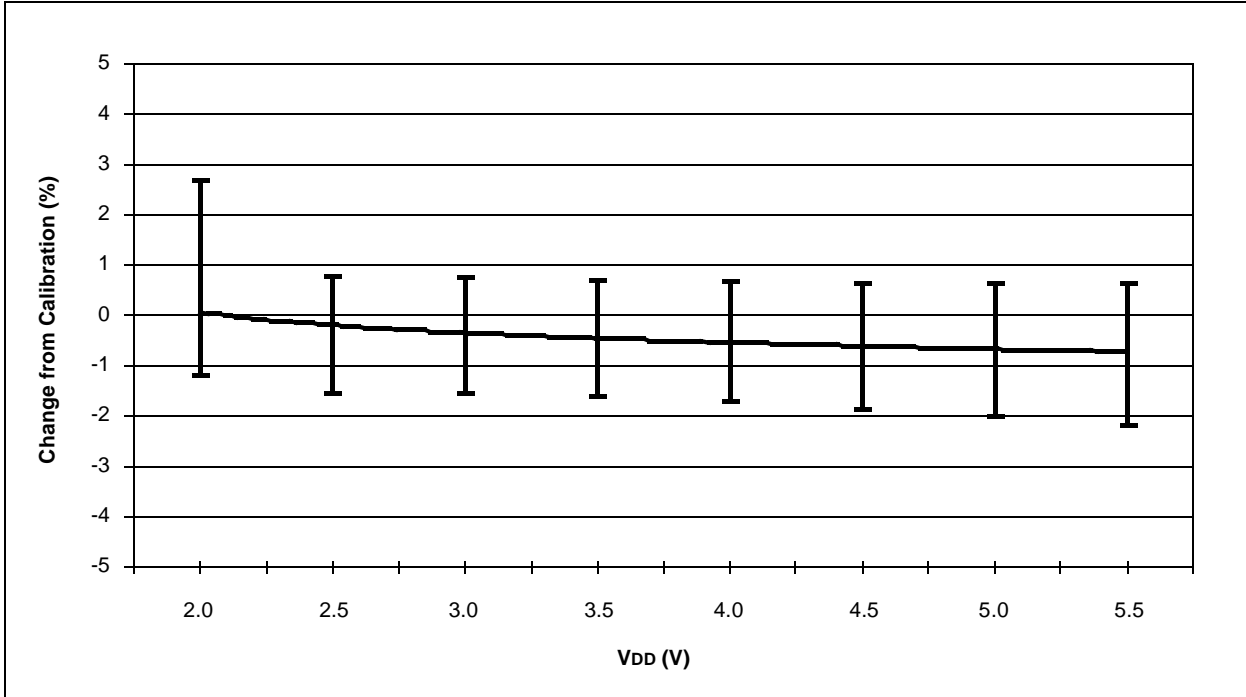
**FIGURE 15-36: MINIMUM HFINTOSC START-UP TIMES vs. V<sub>DD</sub> OVER TEMPERATURE**



**FIGURE 15-37: TYPICAL HFINTOSC FREQUENCY CHANGE vs. V<sub>DD</sub> (25°C)**



**FIGURE 15-38: TYPICAL HFINTOSC FREQUENCY CHANGE OVER DEVICE V<sub>DD</sub> (85°C)**



**FIGURE 15-39: TYPICAL HFINTOSC FREQUENCY CHANGE vs. V<sub>DD</sub> (125°C)**

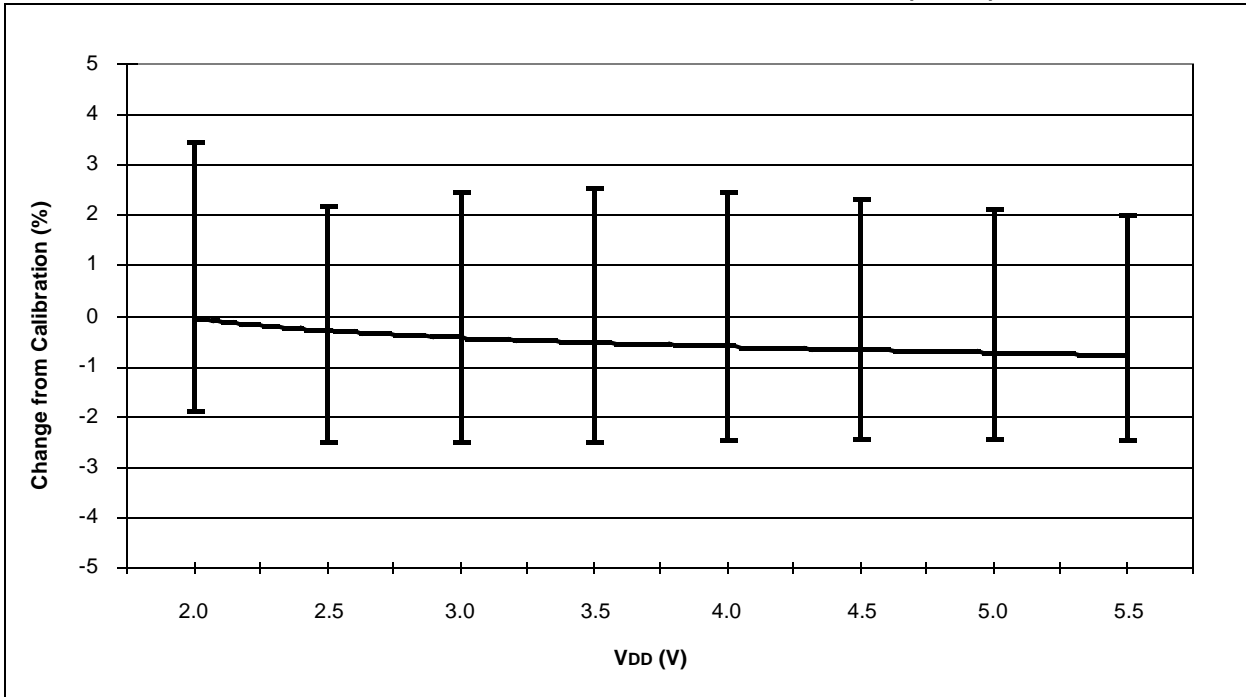
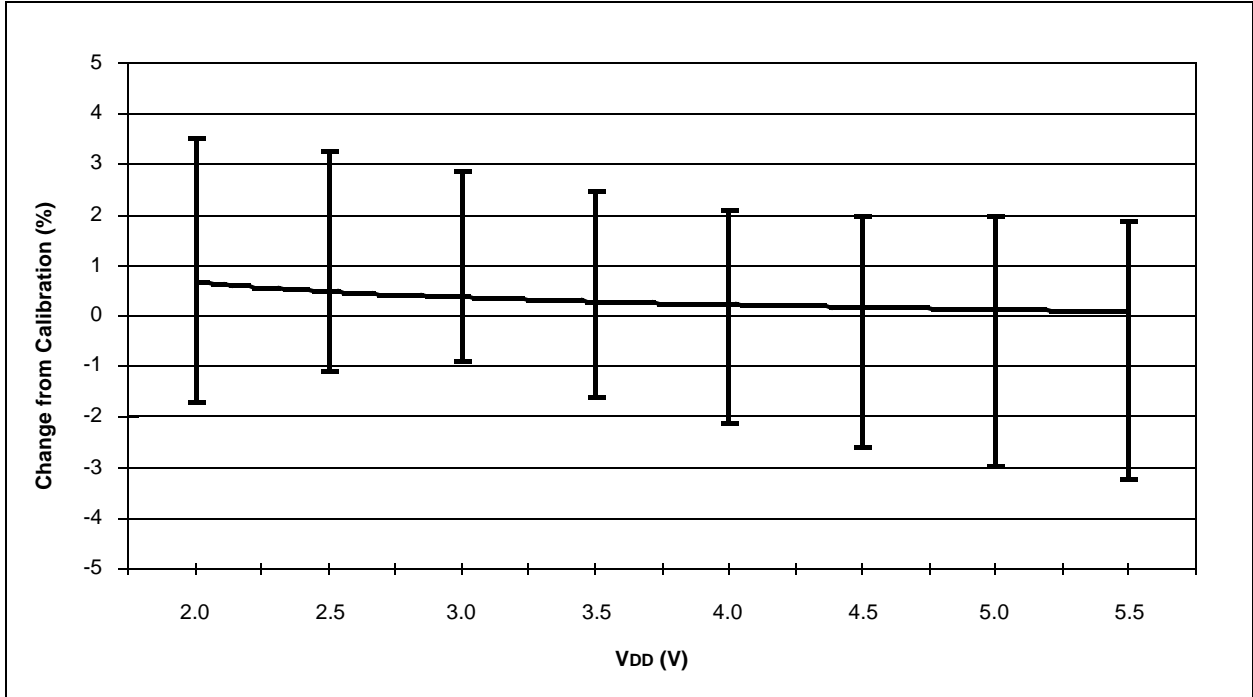




FIGURE 15-40: TYPICAL HFINTOSC FREQUENCY CHANGE vs. VDD (-40°C)



NOTES:

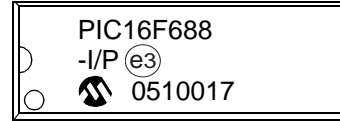
## 16.0 PACKAGING INFORMATION

### 16.1 Package Marking Information

14-Lead PDIP (Skinny DIP)



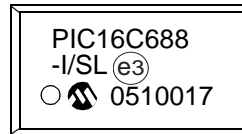
Example



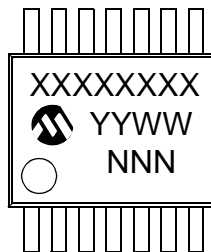
14-Lead SOIC (3.90 mm)



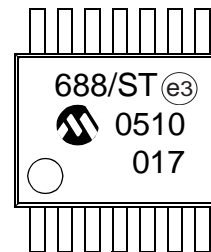
Example



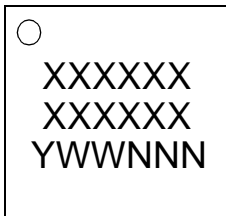
14-Lead TSSOP



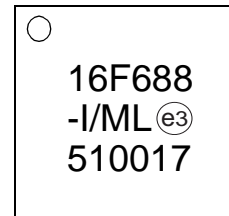
Example



16-Lead QFN



Example



<b>Legend:</b>	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 PIC<sup>®</sup> device marking consists of Microchip part number, year code, week code, and traceability code. For PIC 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.

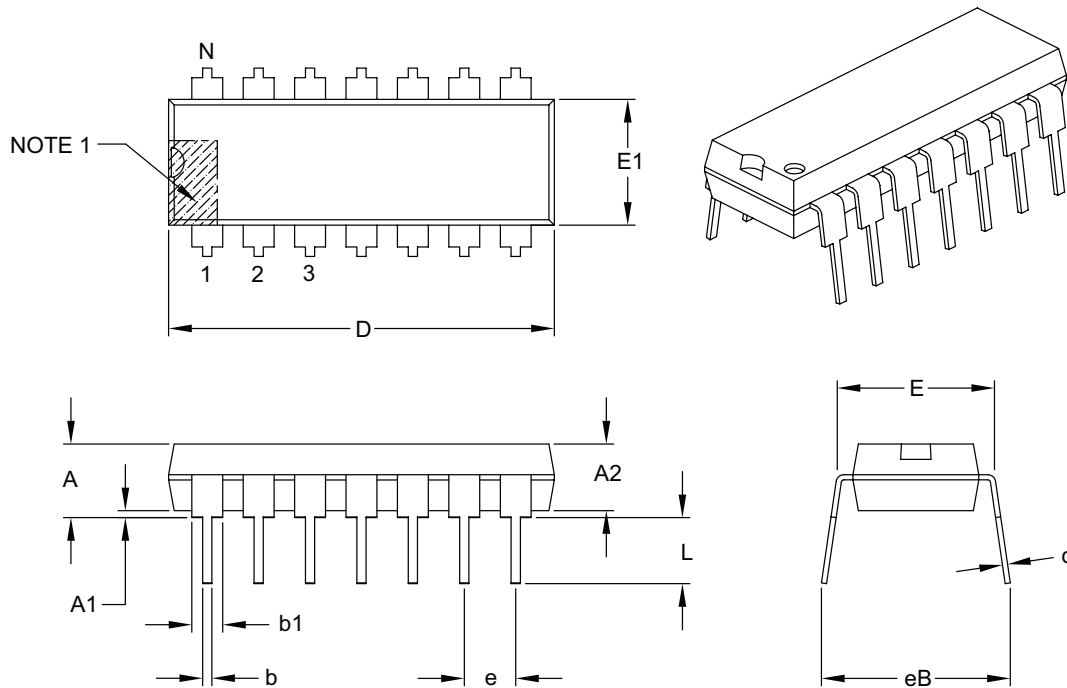
# PIC16F688

## 16.2 Package Details

The following sections give the technical details of the packages.

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

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



Dimension Limits	Units	INCHES		
		MIN	NOM	MAX
Number of Pins	N	14		
Pitch	e	.100 BSC		
Top to Seating Plane	A	–	–	.210
Molded Package Thickness	A2	.115	.130	.195
Base to Seating Plane	A1	.015	–	–
Shoulder to Shoulder Width	E	.290	.310	.325
Molded Package Width	E1	.240	.250	.280
Overall Length	D	.735	.750	.775
Tip to Seating Plane	L	.115	.130	.150
Lead Thickness	c	.008	.010	.015
Upper Lead Width	b1	.045	.060	.070
Lower Lead Width	b	.014	.018	.022
Overall Row Spacing §	eB	–	–	.430

**Notes:**

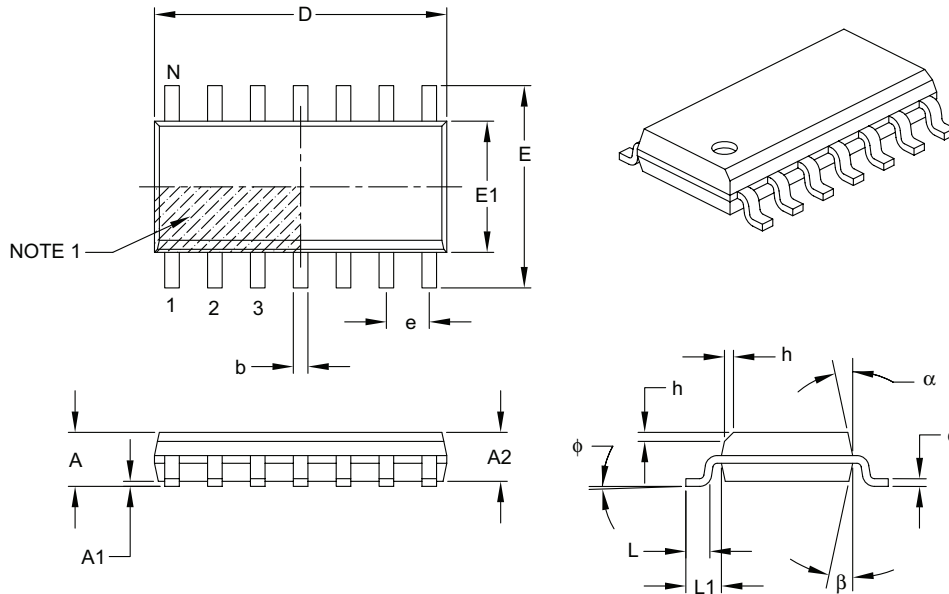
1. Pin 1 visual index feature may vary, but must be located with the hatched area.
2. § Significant Characteristic.
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

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

Microchip Technology Drawing C04-005B

## 14-Lead Plastic Small Outline (SL) – Narrow, 3.90 mm Body [SOIC]

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



Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Number of Pins	N	14		
Pitch	e	1.27 BSC		
Overall Height	A	–	–	1.75
Molded Package Thickness	A2	1.25	–	–
Standoff §	A1	0.10	–	0.25
Overall Width	E	6.00 BSC		
Molded Package Width	E1	3.90 BSC		
Overall Length	D	8.65 BSC		
Chamfer (optional)	h	0.25	–	0.50
Foot Length	L	0.40	–	1.27
Footprint	L1	1.04 REF		
Foot Angle	$\phi$	0°	–	8°
Lead Thickness	c	0.17	–	0.25
Lead Width	b	0.31	–	0.51
Mold Draft Angle Top	$\alpha$	5°	–	15°
Mold Draft Angle Bottom	$\beta$	5°	–	15°

**Notes:**

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- § Significant Characteristic.
- Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.15 mm per side.
- Dimensioning and tolerancing per ASME Y14.5M.

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

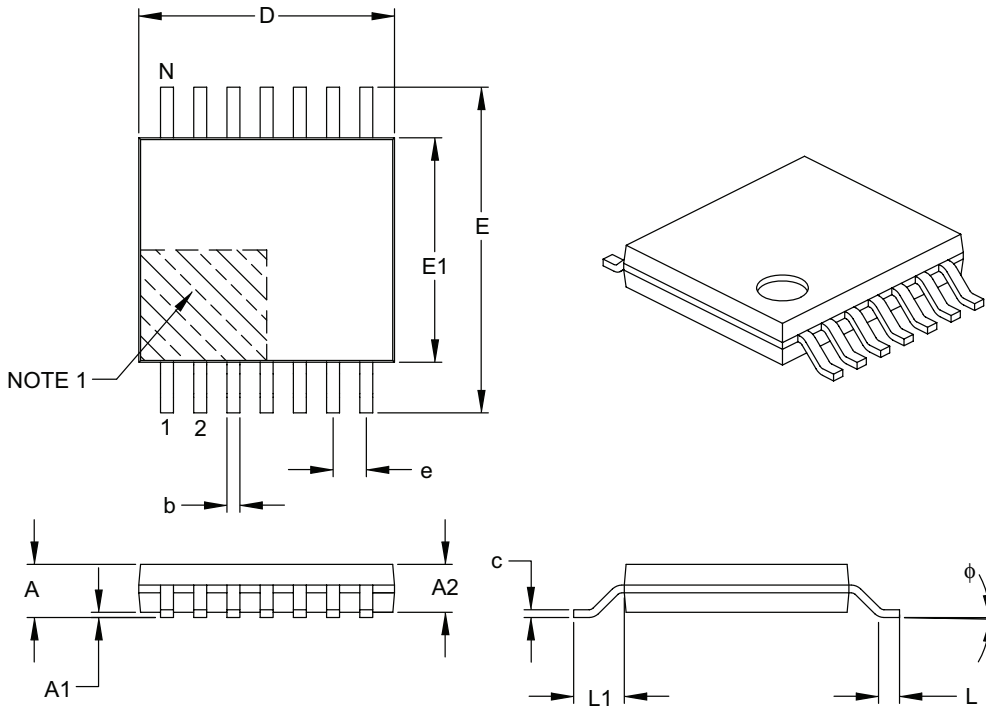
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-065B

# PIC16F688

## 14-Lead Plastic Thin Shrink Small Outline (ST) – 4.4 mm Body [TSSOP]

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



Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Number of Pins	N	14		
Pitch	e	0.65 BSC		
Overall Height	A	–	–	1.20
Molded Package Thickness	A2	0.80	1.00	1.05
Standoff	A1	0.05	–	0.15
Overall Width	E	6.40 BSC		
Molded Package Width	E1	4.30	4.40	4.50
Molded Package Length	D	4.90	5.00	5.10
Foot Length	L	0.45	0.60	0.75
Footprint	L1	1.00 REF		
Foot Angle	$\phi$	0°	–	8°
Lead Thickness	c	0.09	–	0.20
Lead Width	b	0.19	–	0.30

### Notes:

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.15 mm per side.
- Dimensioning and tolerancing per ASME Y14.5M.

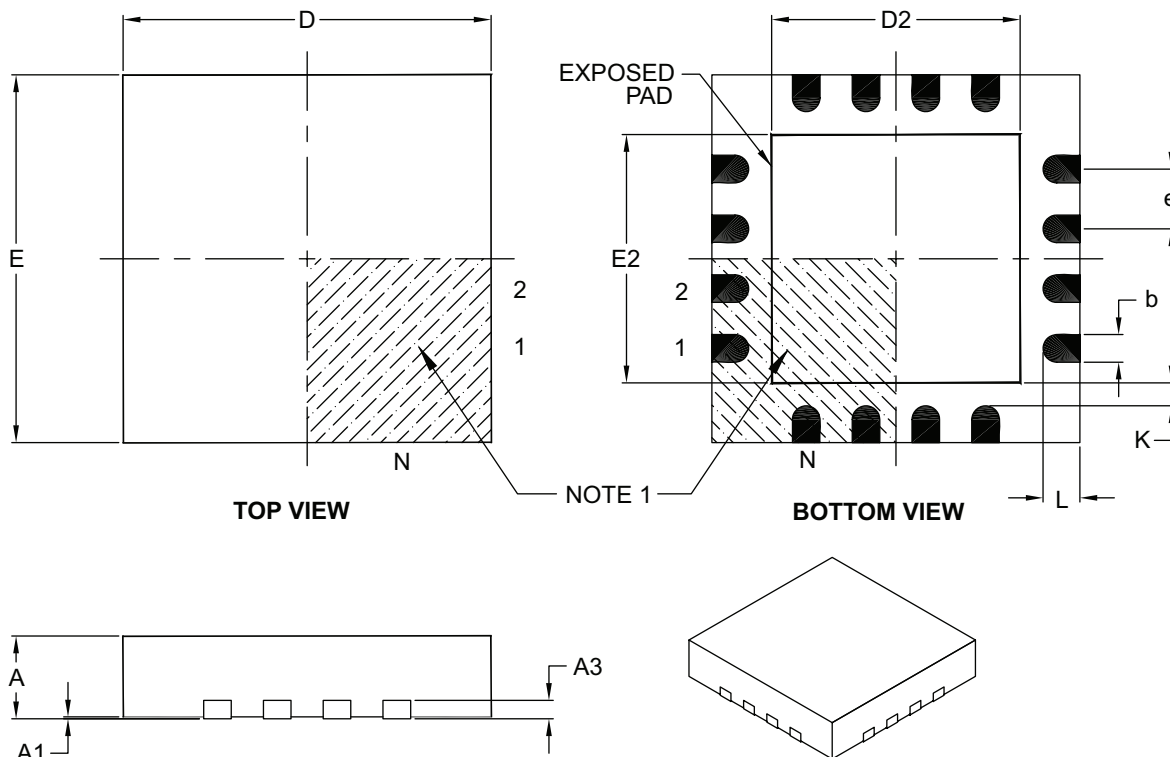
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-087B

## 16-Lead Plastic Quad Flat, No Lead Package (ML) – 4x4x0.9 mm Body [QFN]

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



Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Number of Pins	N	16		
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	4.00 BSC		
Exposed Pad Width	E2	2.50	2.65	2.80
Overall Length	D	4.00 BSC		
Exposed Pad Length	D2	2.50	2.65	2.80
Contact Width	b	0.25	0.30	0.35
Contact Length	L	0.30	0.40	0.50
Contact-to-Exposed Pad	K	0.20	–	–

**Notes:**

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- Package is saw singulated.
- 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 C04-127B

# PIC16F688

---

NOTES:



## APPENDIX A: DATA SHEET REVISION HISTORY

### Revision A

This is a new data sheet.

### Revision B

Rewrites of the Oscillator and Special Features of the CPU Sections. General corrections to Figures and formatting.

### Revision C

Revised Electrical Section and added Char Data. Added Golden Chapters.

### Revision D

Replaced Package Drawings; Revised Product ID (SL Package to 3.90 mm); Replaced PICmicro with PIC; Replaced Dev. Tool Section.

## APPENDIX B: MIGRATING FROM OTHER PIC® DEVICES

This discusses some of the issues in migrating from other PIC devices to the PIC16F6XX family of devices.

### B.1 PIC16F676 to PIC16F688

**TABLE B-1: FEATURE COMPARISON**

Feature	PIC16F676	PIC16F688
Max Operating Speed	20 MHz	20 MHz
Max Program Memory (Words)	1024	4K
SRAM (Bytes)	64	256
A/D Resolution	10-bit	10-bit
Data EEPROM (bytes)	128	256
Timers (8/16-bit)	1/1	1/1
Oscillator Modes	8	8
Brown-out Reset	Y	Y
Internal Pull-ups	RA0/1/2/4/5	RA0/1/2/4/5, MCLR
Interrupt-on-change	RA0/1/2/3/4/5	RA0/1/2/3/4/5
Comparator	1	2
EUSART	N	Y
Ultra Low-Power Wake-up	N	Y
Extended WDT	N	Y
Software Control Option of WDT/BOR	N	Y
INTOSC Frequencies	4 MHz	32 kHz - 8 MHz
Clock Switching	N	Y

**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.

# PIC16F688

---

NOTES:

## INDEX

### A

A/D	
Specifications	160, 161
Absolute Maximum Ratings	143
AC Characteristics	
Industrial and Extended	153
Load Conditions	152
ADC	65
Acquisition Requirements	73
Associated registers	75
Block Diagram	65
Calculating Acquisition Time	73
Channel Selection	66
Configuration	66
Configuring Interrupt	69
Conversion Clock	66
Conversion Procedure	69
Internal Sampling Switch (Rss) Impedance	73
Interrupts	68
Operation	69
Operation During Sleep	69
Port Configuration	66
Reference Voltage (VREF)	66
Result Formatting	68
Source Impedance	73
Starting an A/D Conversion	68
ADCON0 Register	71
ADCON1 Register	71
ADRESH Register (ADFM = 0)	72
ADRESH Register (ADFM = 1)	72
ADRESL Register (ADFM = 0)	72
ADRESL Register (ADFM = 1)	72
Analog Front-end (AFE)	
Power-On Reset	113
Analog Input Connection Considerations	55
Analog-to-Digital Converter. See ADC	
ANSEL Register	34
Assembler	
MPASM Assembler	140

### B

BAUDCTL Register	94
Block Diagrams	
ADC	65
ADC Transfer Function	74
Analog Input Model	55, 74
Clock Source	21
Comparator 1	54
Comparator 2	54
Comparator Modes	57
Crystal Operation	24
EUSART Receive	84
EUSART Transmit	83
External RC Mode	25
Fail-Safe Clock Monitor (FSCM)	31
In-Circuit Serial Programming Connections	127
Interrupt Logic	120
MCLR Circuit	113
On-Chip Reset Circuit	112
PIC16F688	5
RA1 Pins	38
RA2 Pin	38
RA3 Pin	39
RA4 Pin	39

RA5 Pin	40
RC0 and RC1 Pins	43
RC2 and RC3 Pins	43
RC4 Pin	44
RC5 Pin	44
Resonator Operation	24
Timer1	48
TMR0/WDT Prescaler	45
Watchdog Timer (WDT)	123
Break Character (12-bit) Transmit and Receive	101
Brown-out Reset (BOR)	114
Associated	115
Specifications	157
Timing and Characteristics	156

### C

C Compilers	
MPLAB C18	140
MPLAB C30	140
Clock Accuracy with Asynchronous Operation	92
Clock Sources	
External Modes	23
EC	23
HS	24
LP	24
OST	23
RC	25
XT	24
Internal Modes	25
Frequency Selection	27
HFINTOSC	25
INTOSC	25
INTOSCIO	25
LFINTOSC	27
Clock Switching	29
CMCON0 Register	61
CMCON1 Register	62
Code Examples	
A/D Conversion	70
Assigning Prescaler to Timer0	46
Assigning Prescaler to WDT	46
Indirect Addressing	20
Initializing PORTA	33
Initializing PORTC	42
Saving Status and W Registers in RAM	122
Ultra Low-Power Wake-up Initialization	36
Code Protection	126
Comparator	53
C2OUT as T1 Gate	62
Configurations	56
Interrupts	59
Operation	53, 58
Operation During Sleep	60
Response Time	59
Synchronizing COUT w/Timer1	62
Comparator Module	
Associated registers	64
Comparator Voltage Reference (CVREF)	
Response Time	59
Comparator Voltage Reference (CVREF)	63
Effects of a Reset	60
Specifications	159
Comparators	
C2OUT as T1 Gate	49

# PIC16F688

Effects of a Reset.....	60	Transmission .....	107
Specifications.....	159	<b>F</b>	
CONFIG Register.....	111	Fail-Safe Clock Monitor .....	31
Configuration Bits.....	110	Fail-Safe Condition Clearing.....	31
CPU Features .....	109	Fail-Safe Detection .....	31
Customer Change Notification Service .....	197	Fail-Safe Operation.....	31
Customer Notification Service.....	197	Reset or Wake-up from Sleep .....	31
Customer Support.....	197	Firmware Instructions .....	129
<b>D</b>		Flash Program Memory .....	77
Data EEPROM Memory.....	77	Fuses. See Configuration Bits	
Associated Registers .....	82	<b>G</b>	
Reading.....	80	General Purpose Register File .....	7
Writing.....	80	<b>I</b>	
Data Memory.....	7	I/O Ports.....	33
DC and AC Characteristics		ID Locations.....	126
Graphs and Tables .....	163	In-Circuit Debugger.....	127
DC Characteristics		In-Circuit Serial Programming (ICSP).....	127
Extended and Industrial .....	149	Indirect Addressing, INDF and FSR Registers .....	20
Industrial and Extended .....	145	Instruction Format.....	129
Development Support .....	139	Instruction Set.....	129
Device Overview .....	5	ADDLW.....	131
<b>E</b>		ADDWF.....	131
EEADR Register .....	78	ANDLW.....	131
EEADR Registers.....	77	ANDWF.....	131
EEADRH Registers.....	77	BCF .....	131
EECON1 Register .....	77, 79	BSF .....	131
EECON2 Register .....	77	BTFSC .....	131
EEDAT Register .....	78	BTFSS .....	132
EEDATH Register .....	78	CALL.....	132
Electrical Specifications .....	143	CLRF .....	132
Enhanced Universal Synchronous Asynchronous Receiver		CLRW .....	132
Transmitter (EUSART).....	83	CLRWDT .....	132
Errata .....	4	COMF .....	132
EUSART.....	83	DECF .....	132
Associated Registers		DECFSZ .....	133
Baud Rate Generator.....	95	GOTO .....	133
Asynchronous Mode .....	85	INCF .....	133
12-bit Break Transmit and Receive.....	101	INCFSZ.....	133
Associated Registers		IORLW.....	133
Receive .....	91	IORWF.....	133
Transmit.....	87	MOVF .....	134
Auto-Wake-up on Break.....	100	MOVLW .....	134
Baud Rate Generator (BRG).....	95	MOVWF .....	134
Clock Accuracy .....	92	NOP .....	134
Receiver.....	88	RETFIE .....	135
Setting up 9-bit Mode with Address Detect.....	90	RETLW .....	135
Transmitter.....	85	RETURN.....	135
Baud Rate Generator (BRG)		RLF .....	136
Auto Baud Rate Detect .....	99	RRF .....	136
Baud Rate Error, Calculating .....	95	SLEEP .....	136
Baud Rates, Asynchronous Modes.....	96	SUBLW .....	136
Formulas .....	95	SUBWF .....	137
High Baud Rate Select (BRGH Bit).....	95	SWAPF .....	137
Synchronous Master Mode .....	103, 107	XORLW .....	137
Associated Registers		XORWF .....	137
Receive .....	106	Summary Table .....	130
Transmit.....	104	INTCON Register.....	15
Reception.....	105	Internal Oscillator Block	
Transmission.....	103	INTOSC	
Synchronous Slave Mode		Specifications .....	154, 155
Associated Registers		Internal Sampling Switch (Rss) Impedance.....	73
Receive .....	108	Internet Address .....	197
Transmit.....	107	Interrupts.....	119
Reception.....	108		

ADC .....	69	Stack.....	19
Associated Registers .....	121	PCON Register.....	18, 115
Comparator.....	59	PICSTART Plus Development Programmer.....	142
Context Saving.....	122	PIE1 Register .....	16
Interrupt-on-Change.....	34	Pin Diagram.....	2, 3
PORTA Interrupt-on-Change .....	120	Pinout Description	
RA2/INT .....	120	PIC16F688 .....	6
Timer0.....	120	PIR1 Register .....	17
TMR1 .....	50	PORTA .....	33
INTOSC Specifications .....	154, 155	Additional Pin Functions .....	34
IOCA Register.....	35	ANSEL Register .....	34
<b>L</b>		Interrupt-on-Change .....	34
Load Conditions .....	152	Ultra Low-Power Wake-up.....	34, 36
<b>M</b>		Weak Pull-up .....	34
MCLR .....	113	Associated registers .....	41
Internal .....	113	Pin Descriptions and Diagrams .....	37
Memory Organization.....	7	RA0.....	37
Data .....	7	RA1.....	38
Program .....	7	RA2.....	38
Microchip Internet Web Site.....	197	RA4.....	39
Migrating from other PIC Devices .....	191	RA5.....	40
MPLAB ASM30 Assembler, Linker, Librarian .....	140	Specifications .....	155
MPLAB ICD 2 In-Circuit Debugger .....	141	PORTA Register.....	33
MPLAB ICE 2000 High-Performance Universal		PORTC .....	42
In-Circuit Emulator .....	141	Associated registers .....	44
MPLAB ICE 4000 High-Performance Universal		PA/PB/PC/PD. See Enhanced Universal	
In-Circuit Emulator .....	141	Asynchronous Receiver Transmitter (EUSART) .....	42
MPLAB Integrated Development Environment Software ..	139	Specifications .....	155
MPLAB PM3 Device Programmer .....	141	PORTC Register.....	42
MPLINK Object Linker/MPLIB Object Librarian .....	140	Power-Down Mode (Sleep).....	125
<b>O</b>		Power-up Timer (PWRT) .....	113
OPCODE Field Descriptions .....	129	Specifications .....	157
OPTION Register.....	14	Precision Internal Oscillator Parameters .....	155
OPTION_REG Register.....	47	Prescaler	
OSCCON Register.....	10, 22	Shared WDT/Timer0.....	46
Oscillator		Switching Prescaler Assignment .....	46
Associated registers.....	32, 52	Product Identification System .....	199
Oscillator Module .....	21	Program Memory .....	7
EC.....	21	Map and Stack.....	7
HFINTOSC.....	21	Programming, Device Instructions.....	129
HS .....	21	<b>R</b>	
INTOSC .....	21	RA3/MCLR/VPP .....	39
INTOSCIO.....	21	RREG.....	90
LFINTOSC .....	21	RCSTA Register.....	93
LP.....	21	Reader Response.....	198
RC.....	21	Read-Modify-Write Operations .....	129
RCIO.....	21	Register	
XT .....	21	RREG Register .....	99
Oscillator Parameters .....	154	Registers	
Oscillator Specifications.....	153	ADCON0 (ADC Control 0) .....	71
Oscillator Start-up Timer (OST)		ADCON1 (ADC Control 1) .....	71
Specifications.....	157	ADRESH (ADC Result High) with ADFM = 0) .....	72
Oscillator Switching		ADRESH (ADC Result High) with ADFM = 1) .....	72
Fail-Safe Clock Monitor.....	31	ADRESL (ADC Result Low) with ADFM = 0).....	72
Two-Speed Clock Start-up.....	29	ADRESL (ADC Result Low) with ADFM = 1).....	72
OSCTUNE Register .....	26	ANSEL (Analog Select) .....	34
<b>P</b>		BAUDCTL (Baud Rate Control).....	94
Packaging .....	185	CMCON0 (Comparator Control 0) .....	61
Marking .....	185	CMCON1 (Comparator Control 1) .....	62
PDIP Details.....	186	CONFIG (Configuration Word) .....	111
PCL and PCLATH.....	19	EEADR (EEPROM Address) .....	78
Computed GOTO.....	19	EECON1 (EEPROM Control 1) .....	79
		EEDAT (EEPROM Data) .....	78
		EEDATH (EEPROM Data High Byte).....	78
		INTCON (Interrupt Control) .....	15

# PIC16F688

IOCA (Interrupt-on-Change PORTA) .....	35	A/D Conversion (Sleep Mode) .....	162
OPTION_REG (OPTION) .....	14	Asynchronous Reception .....	90
OPTION_REG (Option) .....	47	Asynchronous Transmission .....	86
OSCCON (Oscillator Control) .....	22	Asynchronous Transmission (Back to Back) .....	86
OSCTUNE (Oscillator Tuning) .....	26	Auto Wake-up Bit (WUE) During Normal Operation ..	100
PCON (Power Control Register) .....	18	Auto Wake-up Bit (WUE) During Sleep .....	101
PCON (Power Control) .....	115	Automatic Baud Rate Calculator .....	99
PIE1 (Peripheral Interrupt Enable 1) .....	16	Brown-out Reset (BOR) .....	156
PIR1 (Peripheral Interrupt Register 1) .....	17	Brown-out Reset Situations .....	114
PORTA .....	33	CLKOUT and I/O .....	155
PORTC .....	42	Clock Timing .....	153
RCSTA (Receive Status and Control) .....	93	Comparator Output .....	53
Reset Values .....	117	Fail-Safe Clock Monitor (FSCM) .....	32
Reset Values (Special Registers) .....	118	INT Pin Interrupt .....	121
Special Function Register Map .....	8	Internal Oscillator Switch Timing .....	28
Special Register Summary .....	9	Reset, WDT, OST and Power-up Timer .....	156
STATUS .....	13	Send Break Character Sequence .....	102
T1CON .....	51	Synchronous Reception (Master Mode, SREN) .....	106
TRISA (Tri-State PORTA) .....	33	Synchronous Transmission .....	104
TRISC (Tri-State PORTC) .....	42	Synchronous Transmission (Through TXEN) .....	104
TXSTA (Transmit Status and Control) .....	92	Time-out Sequence .....	116
VRCON (Voltage Reference Control) .....	63	Case 3 .....	116
WDTCON (Watchdog Timer Control) .....	124	Timer0 and Timer1 External Clock .....	158
WPUA (Weak Pull-Up PORTA) .....	35	Timer1 Incrementing Edge .....	50
Reset .....	112	Two Speed Start-up .....	30
Revision History .....	191	Wake-up from Interrupt .....	126
<b>S</b>		Timing Parameter Symbology .....	152
Software Simulator (MPLAB SIM) .....	140	TRISA .....	33
SPBRG .....	95	TRISA Register .....	33
SPBRGH .....	95	TRISC Register .....	42
Special Function Registers .....	7	Two-Speed Clock Start-up Mode .....	29
STATUS Register .....	13	TXREG .....	85
<b>T</b>		TXSTA Register .....	92
T1CON Register .....	51	BRGH Bit .....	95
Thermal Considerations .....	151	<b>U</b>	
Time-out Sequence .....	115	Ultra Low-Power Wake-up .....	6, 34, 36
Timer0 .....	45	<b>V</b>	
Associated Registers .....	47	Voltage Reference. See Comparator Voltage	
External Clock .....	46	Reference (CVREF)	
Interrupt .....	47	Voltage References	
Operation .....	45, 48	Associated registers .....	64
Specifications .....	158	VREF. SEE ADC Reference Voltage	
T0CKI .....	46	<b>W</b>	
Timer1 .....	48	Wake-up on Break .....	100
Associated registers .....	52	Wake-up Using Interrupts .....	125
Asynchronous Counter Mode .....	49	Watchdog Timer (WDT) .....	123
Reading and Writing .....	49	Associated Registers .....	124
Interrupt .....	50	Clock Source .....	123
Modes of Operation .....	48	Modes .....	123
Operation During Sleep .....	50	Period .....	123
Oscillator .....	49	Specifications .....	157
Prescaler .....	49	WDTCON Register .....	9, 124
Specifications .....	158	WPUA Register .....	35
Timer1 Gate		WWW Address .....	197
Inverting Gate .....	49	WWW, On-Line Support .....	4
Selecting Source .....	49, 62		
Synchronizing COUT w/Timer1 .....	62		
TMR1H Register .....	48		
TMR1L Register .....	48		
Timers			
Timer1			
T1CON .....	51		
Timing Diagrams			
A/D Conversion .....	162		

## THE MICROCHIP WEB SITE

Microchip provides online support via our WWW site at [www.microchip.com](http://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](http://www.microchip.com), 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://support.microchip.com>**

# PIC16F688

---

---

## 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  
From: Name \_\_\_\_\_  
Company \_\_\_\_\_  
Address \_\_\_\_\_  
City / State / ZIP / Country \_\_\_\_\_  
Telephone: (\_\_\_\_\_) \_\_\_\_\_ - \_\_\_\_\_ FAX: (\_\_\_\_\_) \_\_\_\_\_ - \_\_\_\_\_

Application (optional):

Would you like a reply? \_\_\_Y \_\_\_N

Device: PIC16F688

Literature Number: DS41203D

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.

<u>PART NO.</u>	<u>X</u>	<u>/XX</u>	<u>XXX</u>
Device	Temperature Range	Package	Pattern
<b>Device:</b> PIC16F688, PIC16F688T <sup>(1)</sup> V <sub>DD</sub> range 2.0V to 5.5V	<b>Temperature Range:</b> I = -40°C to +85°C (Industrial) E = -40°C to +125°C (Extended)	<b>Package:</b> ML = Quad Flat No Leads (QFN) P = Plastic DIP SL = 16-lead Small Outline (3.90 mm) ST = Thin Shrink Small Outline (4.4 mm)	<b>Pattern:</b> QTP, SQTP <sup>SM</sup> or ROM Code; Special Requirements (blank otherwise)
<b>Examples:</b> a) PIC16F688-E/P 301 = Extended Temp., PDIP package, 20 MHz, QTP pattern #301 b) PIC16F688-I/SO = Industrial Temp., SOIC package, 20 MHz			
<b>Note 1:</b> T = In tape and reel TSSOP, SOIC and QFN packages only.			



## 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://support.microchip.com>  
Web Address:  
[www.microchip.com](http://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

**Dallas**  
Addison, TX  
Tel: 972-818-7423  
Fax: 972-818-2924

**Detroit**  
Farmington Hills, MI  
Tel: 248-538-2250  
Fax: 248-538-2260

**Kokomo**  
Kokomo, IN  
Tel: 765-864-8360  
Fax: 765-864-8387

**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: 86-10-8528-2100  
Fax: 86-10-8528-2104

**China - Chengdu**  
Tel: 86-28-8665-5511  
Fax: 86-28-8665-7889

**China - Fuzhou**  
Tel: 86-591-8750-3506  
Fax: 86-591-8750-3521

**China - Hong Kong SAR**  
Tel: 852-2401-1200  
Fax: 852-2401-3431

**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 - Shunde**  
Tel: 86-757-2839-5507  
Fax: 86-757-2839-5571

**China - Wuhan**  
Tel: 86-27-5980-5300  
Fax: 86-27-5980-5118

**China - Xian**  
Tel: 86-29-8833-7250  
Fax: 86-29-8833-7256

### ASIA/PACIFIC

**India - Bangalore**  
Tel: 91-80-4182-8400  
Fax: 91-80-4182-8422

**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 - Yokohama**  
Tel: 81-45-471-6166  
Fax: 81-45-471-6122

**Korea - Gumi**  
Tel: 82-54-473-4301  
Fax: 82-54-473-4302

**Korea - Seoul**  
Tel: 82-2-554-7200  
Fax: 82-2-558-5932 or  
82-2-558-5934

**Malaysia - Penang**  
Tel: 60-4-646-8870  
Fax: 60-4-646-5086

**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: 886-3-572-9526  
Fax: 886-3-572-6459

**Taiwan - Kaohsiung**  
Tel: 886-7-536-4818  
Fax: 886-7-536-4803

**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