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.

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

Microchip received ISO/TS-16949:2002 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona and Mountain View, California in October 2003. The Company's quality system processes and procedures are for its PICmicro® 8-bit MCUs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

PIC16F688

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 ±1%
  - Software selectable frequency range of 8 MHz to 31 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 Detect (BOD) 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 pull-up/input 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:
  - 1 nA @ 2.0V, typical
- Operating Current:
  - 8.5 μA @ 32 kHz, 2.0V, typical
  - 100 μA @ 1 MHz, 2.0V, typical
- Watchdog Timer Current:
  - 1 μA @ 2.0V, typical

Peripheral Features

- 12 I/O pins with individual direction control:
  - High-current source/sink for direct LED drive
  - Interrupt-on-pin change
  - Individually programmable weak pull-ups
  - Ultra Low-Power Wake-up
- Analog comparator module with:
  - Two analog comparators
  - Programmable on-chip voltage reference (CV_{REF}) module (% of V_{DD})
  - 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 Gate Input mode
  - 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

Pin Diagram

14-pin PDIP, SOIC, TSSOP

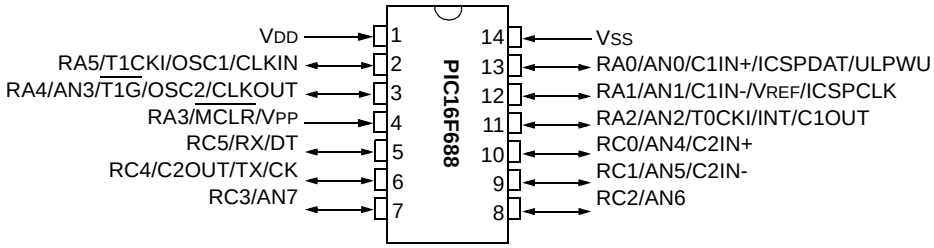


Table of Contents

TO OUR VALUED CUSTOMERS

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

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via E-mail at **docerrors@mail.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)
- The Microchip Corporate Literature Center; U.S. FAX: (480) 792-7277

When contacting a sales office or the literature center, 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/cn** to receive the most current information on all of our products.

NOTES:

1.0 DEVICE OVERVIEW

This document contains device specific information for the PIC16F688. Additional information may be found in the "*PICmicro® Mid-Range MCU Family Reference Manual*" (DS33023), downloaded from the Microchip web site. The Reference Manual should be considered a complementary document to this data sheet and is highly recommended reading for a better understanding of the device architecture and operation of the peripheral modules.

The PIC16F688 is covered by this data sheet. It is available in 14-pin PDIP, SOIC and TSSOP 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

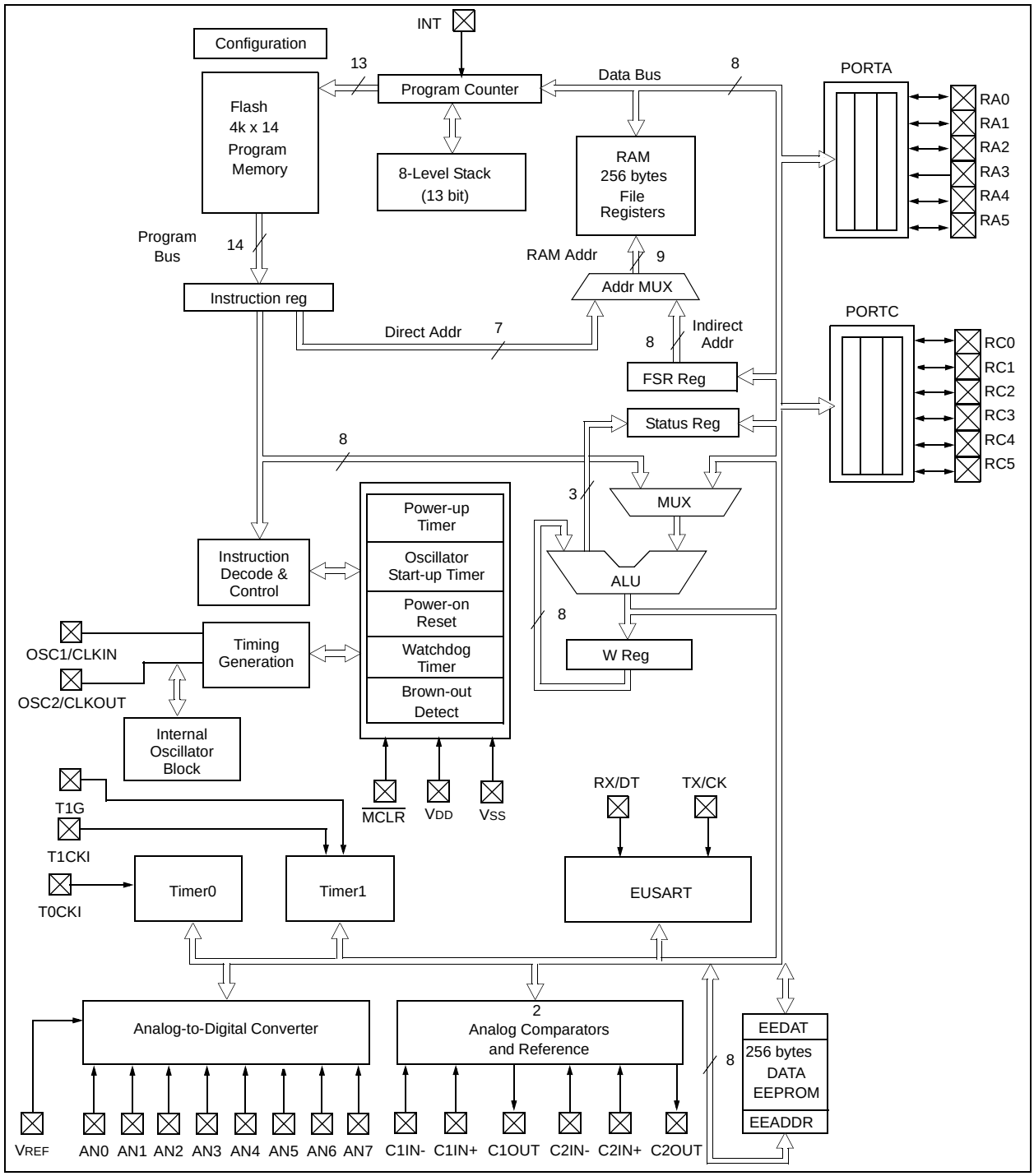


TABLE 1-1: PIC16F688 PINOUT DESCRIPTION

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/$\overline{\text{MCLR}}$/VPP	RA3	TTL	—	PORTA input with interrupt-on-change
	$\overline{\text{MCLR}}$	ST	—	Master Clear w/internal pull-up
	VPP	HV	—	Programming voltage
RA4/AN3/$\overline{\text{T1G}}$/OSC2/CLKOUT	RA4	TTL	CMOS	PORTA I/O w/prog pull-up and interrupt-on-change
	AN3	AN	—	A/D Channel 3 input
	$\overline{\text{T1G}}$	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	ST	CMOS	PORTC I/O
	C2OUT	—	CMOS	Comparator 2 output
	TX	—	CMOS	USART asynchronous output
	CK	ST	CMOS	USART asynchronous clock
RC5/RX/DT	RC5	ST	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

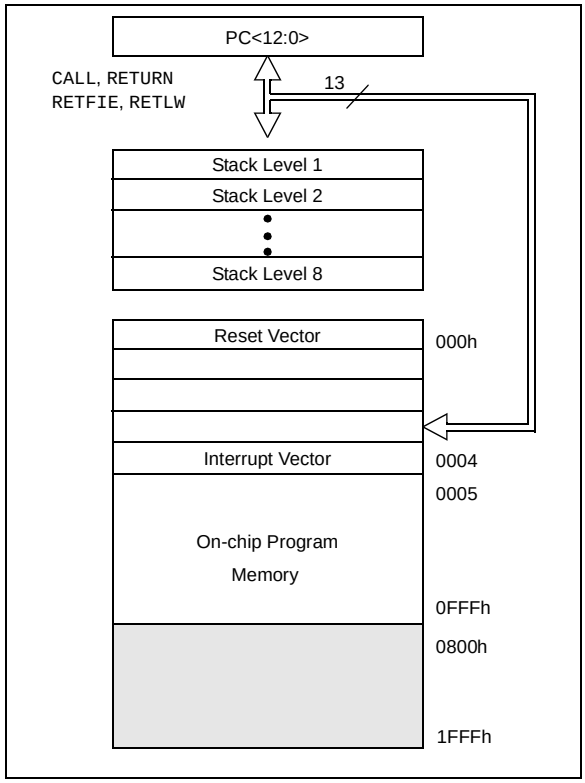
Legend: AN = Analog input or output CMOS = CMOS compatible input or output OC = Open collector output
TTL = TTL compatible input ST = Schmitt Trigger input with CMOS levels
HV = High Voltage XTAL = Crystal

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 wrap around 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.

RP0	RP1	(Status<6:5>)

= 00 : → Bank 0

= 01: → Bank 1

= 10: → Bank 2

= 11: → Bank 3

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.

FIGURE 2-2: PIC16F688 SPECIAL FUNCTION REGISTERS

	File Address		File Address		File Address		File Address
Indirect addr. (1)	00h	Indirect addr. (1)	80h	Indirect addr. (1)	100h	Indirect addr. (1)	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(1)	9Dh		11Dh		19Dh
ADRESH	1Eh	ADRESL	9Eh		11Eh		19Eh
ADCON0	1Fh	ADCON1	9Fh		11Fh		19Fh
General Purpose Register 96 Bytes	20h – 7Fh	General Purpose Register 80 Bytes	A0h – EFh	General Purpose Register 80 Bytes	120h – 17Fh		1A0h – 1FFh
		accesses Bank 0	F0h – FFh	accesses Bank 0		accesses Bank 0	
Bank 0		Bank1		Bank2		Bank3	

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/BOD Reset	Value on all other Resets(1)
Bank 0											
00h	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx
01h	TMR0	Timer0 Module's register								xxxx xxxx	uuuu uuuu
02h	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	0000 0000
03h	STATUS	IRP	RP1	RP0	$\overline{TO}$	PD	Z	DC	C	0001 1xxx	000q quuu
04h	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	uuuu uuuu
05h	PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	--xx xx00	--uu uuuu
06h	—	Unimplemented								—	—
07h	PORTC	—	—	RC5	RC4	RC3	RC2	RC1	RC0	--xx xx00	--uu uuuu
08h	—	Unimplemented								—	—
09h	—	Unimplemented								—	—
0Ah	PCLATH	—	—	—	Write Buffer for upper 5 bits of Program Counter					---0 0000	---0 0000
0Bh	INTCON	GIE	PEIE	T0IE	INTE	RAIE	T0IF	INTF	RAIF(2)	0000 0000	0000 0000
0Ch	PIR1	EEIF	ADIF	RCIF	C2IF	C1IF	OSFIF	TXIF	TMR1IF	0000 0000	0000 0000
0Dh	—	Unimplemented								—	—
0Eh	TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1								xxxx xxxx	uuuu uuuu
0Fh	TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1								xxxx xxxx	uuuu uuuu
10h	T1CON	T1GINV	TMR1GE	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{T1SYNC}$	TMR1CS	TMR1ON	0000 0000	uuuu uuuu
11h	BAUDCTL	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	01-0 0-00	01-0 0-00
12h	SPBRGH	USART Baud Rate High Generator								0000 0000	0000 0000
13h	SPBRG	USART Baud Rate Generator								0000 0000	0000 0000
14h	RCREG	USART Receive Register								0000 0000	0000 0000
15h	TXREG	USART Transmit Register								0000 0000	0000 0000
16h	TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010	0000 0010
17h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
18h	WDTCON	—	—	—	WDTPS3	WDTPS2	WDTPS1	WDTPS0	SWDTEN	---0 1000	---0 1000
19h	CMCON0	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	0000 0000	0000 0000
1Ah	CMCON1	—	—	—	—	—	—	T1GSS	C2SYNC	---- --10	---- --10
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	uuuu uuuu
1Fh	ADCON0	ADFM	VCFG	—	CHS2	CHS1	CHS0	GO/$\overline{DONE}$	ADON	00-0 0000	00-0 0000

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 $\overline{MCLR}$ Reset and Watchdog Timer Reset during normal operation.

2: $\overline{MCLR}$ 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 2-2: 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/BOD Reset	Value on all other Resets(1)
Bank 1											
80h	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx
81h	OPTION_REG	$\overline{RAPU}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
82h	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	0000 0000
83h	STATUS	IRP	RP1	RP0	$\overline{TO}$	PD	Z	DC	C	0001 1xxx	000q quuu
84h	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	uuuu uuuu
85h	TRISA	—	—	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	--11 1111	--11 1111
86h	—	Unimplemented								—	—
87h	TRISC	—	—	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	--11 1111	--11 1111
88h	—	Unimplemented								—	—
89h	—	Unimplemented								—	—
8Ah	PCLATH	—	—	—	Write Buffer for upper 5 bits of Program Counter					---0 0000	---0 0000
8Bh	INTCON	GIE	PEIE	T0IE	INTE	RAIE	T0IF	INTF	RAIF(3)	0000 0000	0000 0000
8Ch	PIE1	EEIE	ADIE	RCIE	C2IE	C1IE	OSFIE	TXIE	TMR1IE	0000 0000	0000 0000
8Dh	—	Unimplemented								—	—
8Eh	PCON	—	—	ULPWUE	SBODEN	—	—	$\overline{POR}$	$\overline{BOD}$	--01 --qq	--0u --uu
8Fh	OSCCON	—	IRCF2	IRCF1	IRCF0	OSTS	HTS	LTS	SCS	-110 x000	-110 x000
90h	OSCTUNE	—	—	—	TUN4	TUN3	TUN2	TUN1	TUN0	---0 0000	---u uuuu
91h	ANSEL	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0	1111 1111	1111 1111
92h	—	Unimplemented								—	—
93h	—	Unimplemented								—	—
94h	—	Unimplemented								—	—
95h	WPUA(2)	—	—	WPUA5	WPUA4	—	WPUA2	WPUA1	WPUA0	--11 -111	--11 -111
96h	IOCA	—	—	IOCA5	IOCA4	IOCA3	IOCA2	IOCA1	IOCA0	--00 0000	--00 0000
97h	EEDATH	—	—	EEPROM Data Register						--00 0000	0000 0000
98h	EEADRH	—	—	—	—	EEPROM Address Register				---- 0000	0000 0000
99h	VRCON	VREN	—	VRR	—	VR3	VR2	VR1	VR0	0-0- 0000	0-0- 0000
9Ah	EEDAT	EEDAT7	EEDAT6	EEDAT5	EEDAT4	EEDAT3	EEDAT2	EEDAT1	EEDAT0	0000 0000	0000 0000
9Bh	EEADR	EEADR7	EEADR6	EEADR5	EEADR4	EEADR3	EEADR2	EEADR1	EEADR0	0000 0000	0000 0000
9Ch	EECON1	EEPGD	—	—	—	WRERR	WREN	WR	RD	x--- x000	u--- q000
9Dh	EECON2	EEPROM Control 2 Register (not a physical register)								---- ----	---- ----
9Eh	ADRESL	Least Significant 2 bits of the left shifted result or 8 bits of the right shifted result								xxxx xxxx	uuuu uuuu
9Fh	ADCON1	—	ADCS2	ADCS1	ADCS0	—	—	—	—	-000 ----	-000 ----

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 $\overline{MCLR}$ Reset and Watchdog Timer Reset during normal operation.

2: RA3 pull-up is enabled when pin is configured as $\overline{MCLR}$ in the Configuration Word register.

3: $\overline{MCLR}$ 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/BOD Reset	Value on all other Resets(1)
Bank 2											
100h	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx
101h	TMR0	Timer0 Module's register								xxxx xxxx	uuuu uuuu
102h	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	0000 0000
103h	STATUS	IRP	RP1	RP0	$\overline{TO}$	PD	Z	DC	C	0001 1xxx	000q quuu
104h	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	uuuu uuuu
105h	PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	--xx xx00	--uu uuuu
106h	—	Unimplemented								—	—
107h	PORTC	—	—	RC5	RC4	RC3	RC2	RC1	RC0	--xx xx00	--uu uuuu
108h	—	Unimplemented								—	—
109h	—	Unimplemented								—	—
10Ah	PCLATH	—	—	—	Write Buffer for upper 5 bits of Program Counter					---0 0000	---0 0000
10Bh	INTCON	GIE	PEIE	T0IE	INTE	RAIE	T0IF	INTF	RAIF(2)	0000 0000	0000 0000
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 $\overline{MCLR}$ Reset and Watchdog Timer Reset during normal operation.

2: $\overline{MCLR}$ 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 2-4: 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/BOD Reset	Value on all other Resets(1)
Bank 3											
180h	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx
181h	OPTION_REG	$\overline{RAPU}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
182h	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	0000 0000
183h	STATUS	IRP	RP1	RP0	$\overline{TO}$	PD	Z	DC	C	0001 1xxx	000q quuu
184h	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	uuuu uuuu
185h	TRISA	—	—	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	--11 1111	--11 1111
186h	—	Unimplemented								—	—
187h	TRISC	—	—	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	--11 1111	--11 1111
188h	—	Unimplemented								—	—
189h	—	Unimplemented								—	—
18Ah	PCLATH	—	—	—	Write Buffer for upper 5 bits of Program Counter					---0 0000	---0 0000
18Bh	INTCON	GIE	PEIE	T0IE	INTE	RAIE	T0IF	INTF	RAIF(2)	0000 0000	0000 0000
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 $\overline{MCLR}$ Reset and Watchdog Timer Reset during normal operation.

2: $\overline{MCLR}$ 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 $\overline{TO}$ and 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 (Status<7:6>) 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 $\overline{Borrow}$ and $\overline{Digit\ Borrow}$ out bit, respectively, in subtraction. See the `SUBLW` and `SUBWF` instructions for examples.

REGISTER 2-1: STATUS – STATUS REGISTER (ADDRESS: 03h, 83h, 103h, OR 183h)

R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
IRP	RP1	RP0	$\overline{TO}$	PD	Z	DC	C
bit 7							bit 0

bit 7 **IRP:** Register Bank Select bit (used for indirect addressing)
`1` = Bank 2, 3 (100h-1FFh)
`0` = Bank 0, 1 (00h-FFh)

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

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

bit 3 **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/$\overline{borrow}$ bit (`ADDWF`, `ADDLW`, `SUBLW`, `SUBWF` instructions)
For $\overline{borrow}$, the polarity is reversed.
`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/$\overline{borrow}$ bit (`ADDWF`, `ADDLW`, `SUBLW`, `SUBWF` instructions)
`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 $\overline{borrow}$, 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 or low-order bit of the source register.

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

2.2.2.2 Option Register

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

- TMR0/WDT prescaler
- External RA2/INT interrupt
- TMR0
- Weak pull-ups on PORTA

Note:	To achieve a 1:1 prescaler assignment for TMR0, assign the prescaler to the WDT by setting PSA bit to '1' (Option<3>). See **Section 5.4 "Prescaler"**.

REGISTER 2-2: OPTION_REG – OPTION REGISTER (ADDRESS: 81h OR 181h)

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
$\overline{\text{RAPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7							bit 0

bit 7 **$\overline{\text{RAPU}}$:** 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:** TMR0 Clock Source Select bit
1 = Transition on RA2/T0CKI pin
0 = Internal instruction cycle clock (CLKOUT)

bit 4 **T0SE:** TMR0 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	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

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

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 (INTCON<7>). User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

REGISTER 2-3: INTCON – INTERRUPT CONTROL REGISTER (ADDRESS: 0Bh OR 8Bh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
GIE	PEIE	T0IE	INTE	RAIE	T0IF	INTF	RAIF
bit 7							bit 0

bit 7 **GIE:** Global Interrupt Enable bit
1 = Enables all unmasked interrupts
0 = Disables all interrupts

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

bit 5 **T0IE:** TMR0 Overflow Interrupt Enable bit
1 = Enables the TMR0 interrupt
0 = Disables the TMR0 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**(1)**
1 = Enables the PORTA change interrupt
0 = Disables the PORTA change interrupt

bit 2 **T0IF:** TMR0 Overflow Interrupt Flag bit**(2)**
1 = TMR0 register has overflowed (must be cleared in software)
0 = TMR0 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: T0IF bit is set when Timer0 rolls over. Timer0 is unchanged on Reset and should be initialized before clearing T0IF bit.

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

2.2.2.4 PIE1 Register

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

Note: Bit PEIE (INTCON<6>) must be set to enable any peripheral interrupt.

REGISTER 2-4: PIE1 – PERIPHERAL INTERRUPT ENABLE REGISTER 1 (ADDRESS: 8Ch)

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

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 Interrupt Enable bit
1 = Enables the A/D converter interrupt
0 = Disables the A/D converter interrupt

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

bit 4 **C2IE:** Comparator 2 Interrupt Enable bit
1 = Enables the Comparator 2 interrupt
0 = Disables the Comparator 2 interrupt

bit 3 **C1IE:** Comparator 1 Interrupt Enable bit
1 = Enables the Comparator 1 interrupt
0 = Disables the Comparator 1 interrupt

bit 2 **OSFIE:** Oscillator Fail Interrupt Enable bit
1 = Enables the oscillator fail interrupt
0 = Disables the oscillator fail interrupt

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

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

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

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 (INTCON<7>). 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 (ADDRESS: 0Ch)

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

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 Interrupt Flag bit
1 = A/D conversion complete
0 = A/D conversion has not completed or has not been started

bit 5 **RCIF**: USART Receive Interrupt Flag bit
1 = The USART receive buffer is full
0 = The USART receive buffer is empty

bit 4 **C2IF**: Comparator 2 Interrupt Flag bit
1 = Comparator 2 output has changed (must be cleared in software)
0 = Comparator 2 output has not changed

bit 3 **C1IF**: Comparator 1 Interrupt Flag bit
1 = Comparator 1 output has changed (must be cleared in software)
0 = Comparator 1 output 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**: USART Transmit Interrupt Flag bit
1 = The USART transmit buffer is empty
0 = The USART transmit buffer is full

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

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

2.2.2.6 PCON Register

The Power Control (PCON) register (See Table 12-2) contains flag bits to differentiate between a:

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

The PCON register also controls the Ultra Low-Power Wake-up and software enable of the BOD.

The PCON register bits are shown in Register 2-6.

REGISTER 2-6: PCON – POWER CONTROL REGISTER (ADDRESS: 8Eh)

U-0	U-0	R/W-0	R/W-1	U-0	U-0	R/W-0	R/W-x
—	—	ULPWUE	SBODEN	—	—	$\overline{POR}$	$\overline{BOD}$
bit 7							bit 0

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 **SBODEN**: Software BOD Enable bit(1)
1 = BOD enabled
0 = BOD disabled

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

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

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

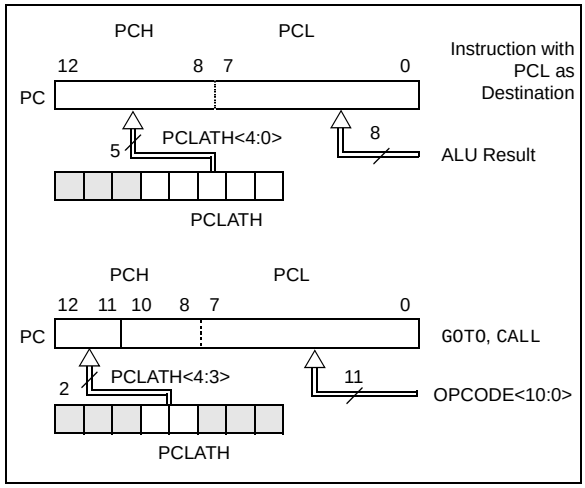
Note 1: BODEN<1:0> = 01 in the Configuration Word register for this bit to control the $\overline{BOD}$.

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

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.

2.4 Indirect Addressing, INDF and FSR Registers

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

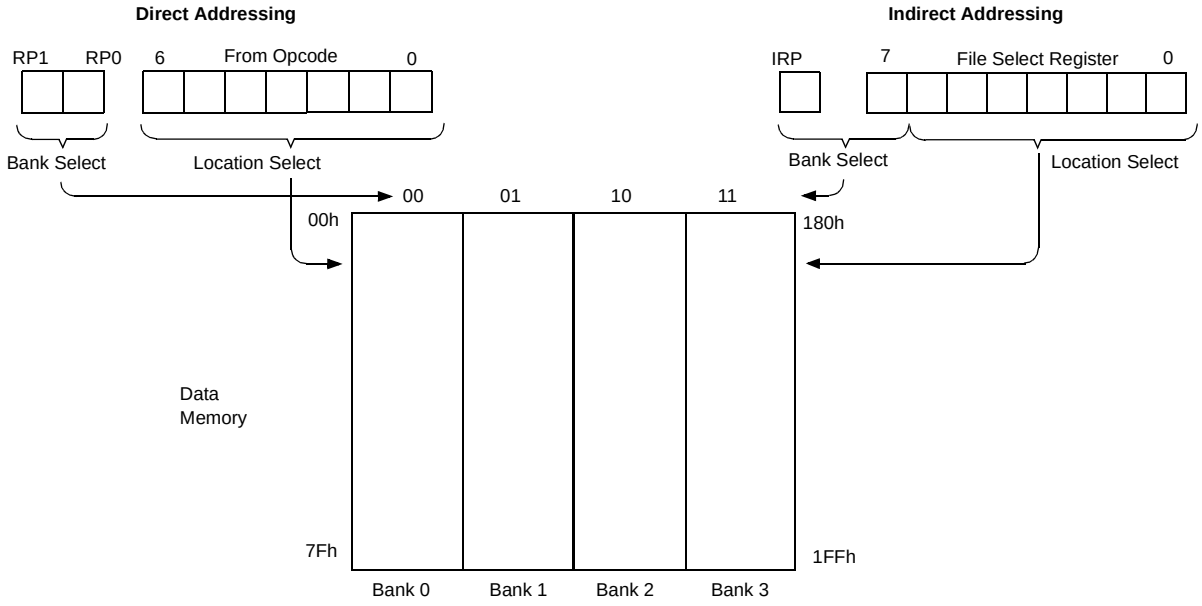
Indirect addressing is possible by using the INDF register. Any instruction using the INDF register actually accesses data pointed to by the File Select Register (FSR). Reading INDF itself indirectly will produce 00h. Writing to the INDF register indirectly results in a no operation (although Status bits may be affected). An effective 9-bit address is obtained by concatenating the 8-bit FSR register and the IRP bit (Status<7>), 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 CLRF   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



Note: For memory map detail, see Figure 2-2.

3.0 CLOCK SOURCES

3.1 Overview

The PIC16F688 has a wide variety of clock sources and selection features to 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 PIC16F688 clock sources.

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:

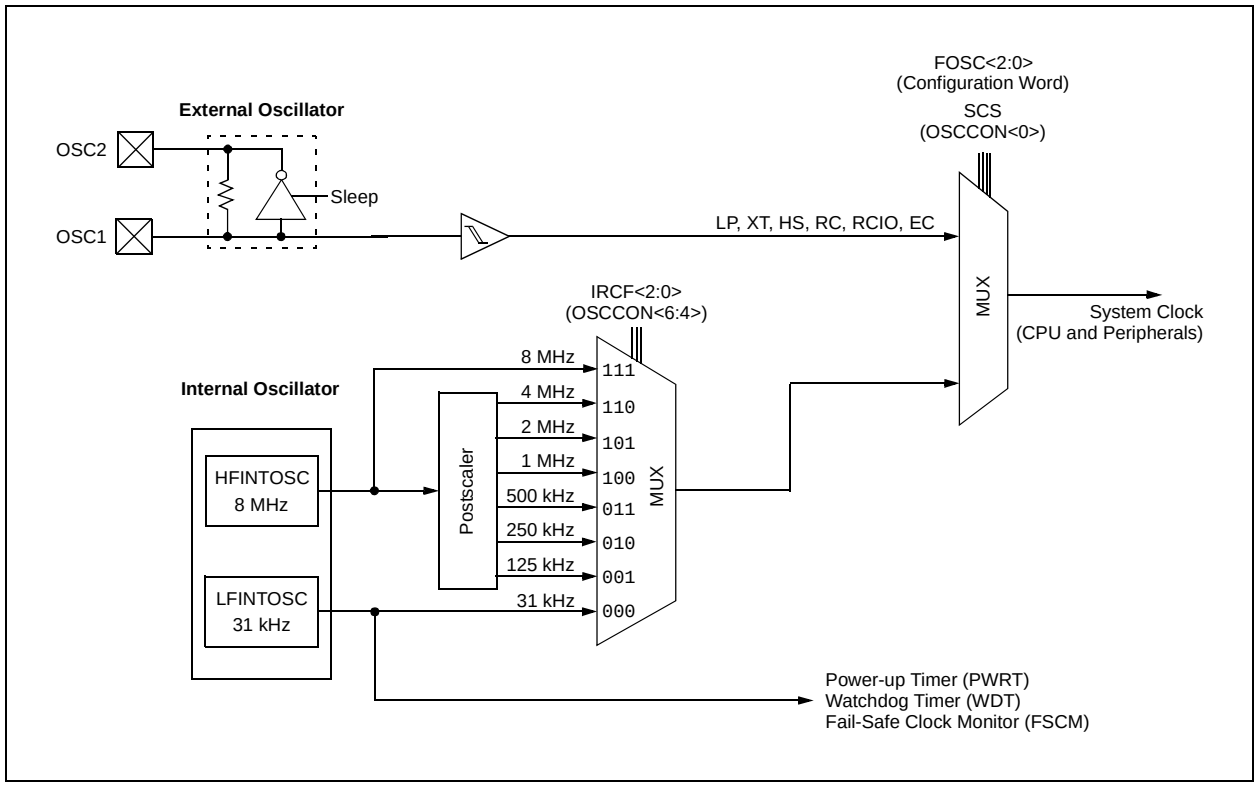
- Selectable system clock source between external or internal via software.
- Two-Speed Clock Start-up mode, which minimizes latency between external oscillator start-up and code execution.
- Fail-Safe Clock Monitor (FSCM) designed to detect a failure of the external clock source (LP, XT, HS, EC or RC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

The PIC16F688 can be configured in one of eight clock modes.

1. EC – External clock with I/O on RA4.
2. LP – Low gain Crystal or Ceramic Resonator Oscillator 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 RA4.
6. RCIO – External Resistor-Capacitor with I/O on RA4.
7. INTRC – Internal oscillator with FOSC/4 output on RA4 and I/O on RA5.
8. INTRCIO – Internal oscillator with I/O on RA4 and RA5.

Clock source modes are configured by the FOSC<2:0> bits in the Configuration Word register (see **Section 11.0 “Special Features Of The CPU”**). The internal clock can be generated by two oscillators. The HFINTOSC is a high-frequency calibrated oscillator. The LFINTOSC is a low-frequency uncalibrated oscillator.

FIGURE 3-1: PIC16F688 CLOCK SOURCE BLOCK DIAGRAM



3.2 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 PIC16F688. The PIC16F688 has two internal oscillators: the 8 MHz High-Frequency Internal Oscillator (HFINTOSC) and 31 kHz Low-Frequency Internal Oscillator (LFINTOSC).

The system clock can be selected between external or internal clock sources via the System Clock Selection (SCS) bit (see **Section 3.5 "Clock Switching"**).

3.3 External Clock Modes

3.3.1 OSCILLATOR START-UP TIMER (OST)

If the PIC16F688 is configured for LP, XT or HS modes, the Oscillator Start-up Timer (OST) counts 1024 oscillations from the OSC1 pin, following a Power-on Reset (POR) and 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 PIC16F688. 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 3.6 "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–8 MHz	5 µs-10 µs (approx.) CPU Start-up(1)
Sleep/POR	EC, RC	DC – 20 MHz	
LFINTOSC (31 kHz)	EC, RC	DC – 20 MHz	
Sleep/POR	LP, XT, HS	31 kHz–20 MHz	1024 Clock Cycles (OST)
LFINTOSC (31 kHz)	HFINTOSC	125 kHz–8 MHz	1 µs (approx.)

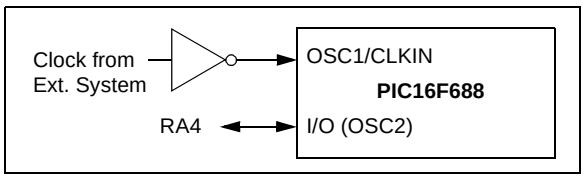
Note 1: The 5 µs to 10 µs start-up delay is based on a 1 MHz system clock.

3.3.2 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 pin and the RA5 pin 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 PIC16F688 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



3.3.3 LP, XT, HS MODES

The LP, XT and HS modes support the use of quartz crystal resonators or ceramic resonators connected to the OSC1 and OSC2 pins (Figure 3-1). The mode selects a low, medium or high gain setting of the internal inverter-amplifier to support various resonator types and speed.

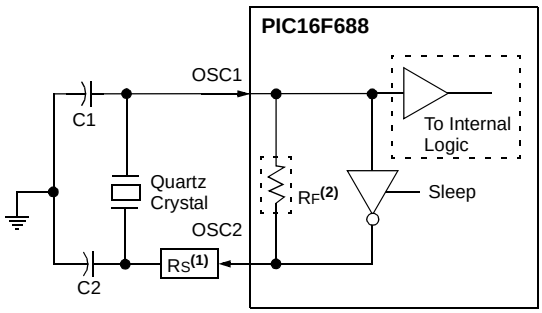
LP Oscillator mode selects the lowest gain setting of the internal inverter-amplifier. LP mode current consumption is the least of the three modes. This mode is best suited to drive resonators with a low drive level specification, for example, tuning fork type crystals.

XT Oscillator mode selects the intermediate gain setting of the internal inverter-amplifier. XT mode current consumption is the medium of the three modes. This mode is best suited to drive resonators with a medium drive level specification, for example, low-frequency/AT-cut quartz crystal resonators.

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, for example, high-frequency/AT-cut quartz crystal resonators or ceramic resonators.

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)



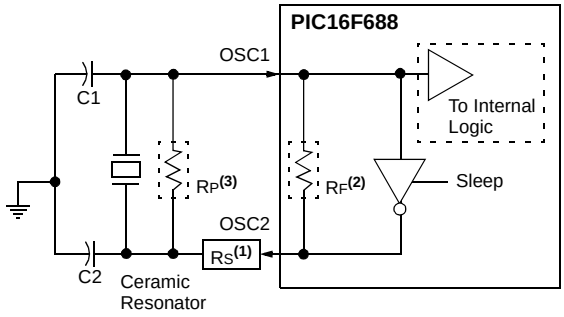
Note 1: A series resistor (RS) may be required for quartz crystals with low drive level.

2: The value of RF varies with the oscillator mode selected (typically between 2 MΩ to 10 MΩ).

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.

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



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

2: The value of RF varies with the oscillator mode selected (typically between 2 MΩ to 10 MΩ).

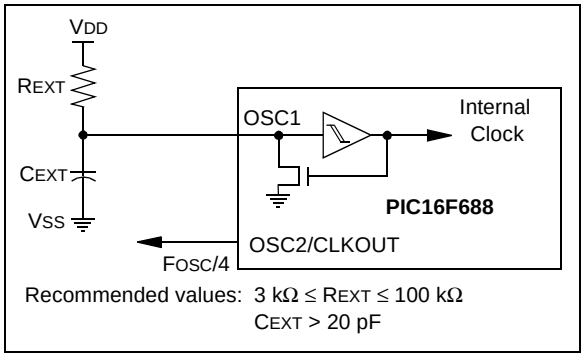
3: An additional parallel feedback resistor (RP) may be required for proper ceramic resonator operation (typical value 1 MΩ).

3.3.4 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.

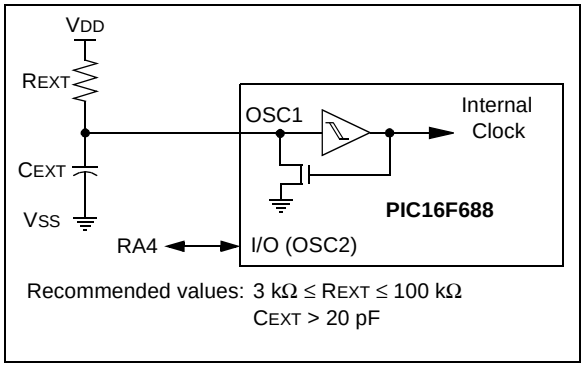
In RC mode, the RC circuit connects to the OSC1 pin. The OSC2/CLKOUT pin outputs the RC oscillator frequency divided by 4. This signal may be used to provide a clock for external circuitry, synchronization, calibration, test or other application requirements. Figure 3-5 shows the RC mode connections.

FIGURE 3-5: RC MODE



In RCIO mode, the RC circuit is connected to the OSC1 pin. The OSC2 pin becomes an additional general purpose I/O pin. The I/O pin becomes bit 4 of PORTA (RA4). Figure 3-6 shows the RCIO mode connections.

FIGURE 3-6: RCIO MODE



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

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

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

3.4 Internal Clock Modes

The PIC16F688 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 ±12% via software using the OSCTUNE register (Register 3-1).
2. The **LFINTOSC** (Low-Frequency Internal Oscillator) is uncalibrated and operates at approximately 31 kHz.

The system clock speed can be selected via software using the Internal Oscillator Frequency Select (IRCF) bits.

The system clock can be selected between external or internal clock sources via the System Clock Selection (SCS) bit (see **Section 3.5 "Clock Switching"**).

3.4.1 INTRC AND INTRCIO MODES

The INTRC and INTRCIO modes configure the internal oscillators as the system clock source when the device is programmed using the Oscillator Selection (FOSC) bits in the Configuration Word register (Register 11-1).

In **INTRC** mode, the OSC1 pin is available for general purpose I/O. The OSC2/CLKOUT pin 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 **INTRCIO** mode, the OSC1 and OSC2 pins are available for general purpose I/O.

3.4.2 HFINTOSC

The High-Frequency Internal Oscillator (HFINTOSC) is a factory calibrated 8 MHz internal clock source. The frequency of the HFINTOSC can be altered approximately ±12% via software using the OSCTUNE register (Register 3-1).

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 bits (see **Section 3.4.4 "Frequency Select Bits (IRCF)"**).

The HFINTOSC is enabled by selecting any frequency between 8 MHz and 125 kHz (IRCF ≠ 000) as the system clock source (SCS = 1), or when Two-Speed Start-up is enabled (IESO = 1 and IRCF ≠ 000).

The HF Internal Oscillator (HTS) bit (OSCCON<2>) indicates whether the HFINTOSC is stable or not.

3.4.2.1 OSCTUNE Register

The HFINTOSC is factory calibrated but can be adjusted in software by writing to the OSCTUNE register (Register 3-1).

The OSCTUNE register has a tuning range of ±12%. The default value of the OSCTUNE register is '0'. The value is a 5-bit two's complement number. Due to process variation, the monotonicity and frequency step cannot be specified.

When the OSCTUNE register is modified, the HFINTOSC frequency will begin shifting to the new frequency. The HFINTOSC clock will stabilize within 1 ms. 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 3-1: OSCTUNE – OSCILLATOR TUNING RESISTOR (ADDRESS: 90h)

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

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

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

3.4.3 LFINTOSC

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

The output of the LFINTOSC connects to a postscaler and multiplexer (see Figure 3-1). 31 kHz can be selected via software using the IRCF bits (see **Section 3.4.4 "Frequency Select Bits (IRCF)"**). 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 = 000) as the system clock source (SCS = 1), or when any of the following are enabled:

- Two-Speed Start-up (IESO = 1 and IRCF = 000)
- Power-up Timer (PWRT)
- Watchdog Timer (WDT)
- Fail-Safe Clock Monitor (FSCM)

The LF Internal Oscillator (LTS) bit (OSCCON<1>) indicates whether the LFINTOSC is stable or not.

3.4.4 FREQUENCY SELECT BITS (IRCF)

The output of the 8 MHz HFINTOSC and 31 kHz LFINTOSC connect to a postscaler and multiplexer (see Figure 3-1). The Internal Oscillator Frequency select bits, IRCF<2:0> (OSCCON<6:4>), 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

Note:	Following any Reset, the IRCF bits 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.

3.4.5 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. If this is the case, there is a 10 μs delay after the IRCF bits are modified before the frequency selection takes place. The LTS/HTS bits will reflect the current active status of the LFINTOSC and the HFINTOSC oscillators. The timing of a frequency selection is as follows:

1. IRCF bits are modified.
2. If the new clock is shut down, a 10 μs 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. HTS/LTS bits are updated as required.
6. Clock switch is complete.

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 the new frequencies are derived from the HFINTOSC via the postscaler and multiplexer.

3.5 Clock Switching

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

3.5.1 SYSTEM CLOCK SELECT (SCS) BIT

The System Clock Select (SCS) bit (OSCCON<0>) selects the system clock source that is used for the CPU and peripherals.

- When SCS = 0, the system clock source is determined by configuration of the FOSC<2:0> bits in the Configuration Word register (CONFIG).
- When SCS = 1, the system clock source is chosen by the internal oscillator frequency selected by the IRCF bits. After a Reset, SCS 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. The user can monitor the OSTS (OSCCON<3>) to determine the current system clock source.

3.5.2 OSCILLATOR START-UP TIME-OUT STATUS BIT

The Oscillator Start-up Time-out Status (OSTS) bit (OSCCON<3>) indicates whether the system clock is running from the external clock source, as defined by the FOSC bits, or from internal clock source. In particular, OSTS indicates that the Oscillator Start-up Timer (OST) has timed out for LP, XT or HS modes.

3.6 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 (OSCCON<3>) to remain clear.

When the PIC16F688 is configured for LP, XT or HS modes, the Oscillator Start-up Timer (OST) is enabled (see **Section 3.3.1 "Oscillator Start-up Timer (OST)"**). The OST timer 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 (OSCCON<3>) is set, program execution switches to the external oscillator.

3.6.1 TWO-SPEED START-UP MODE CONFIGURATION

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

- IESO = 1 (CONFIG<10>) Internal/External Switch Over bit.
- SCS = 0.
- FOSC configured for LP, XT or HS mode.

Two-Speed Start-up mode is entered after:

- Power-on Reset (POR) and, if enabled, after 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.

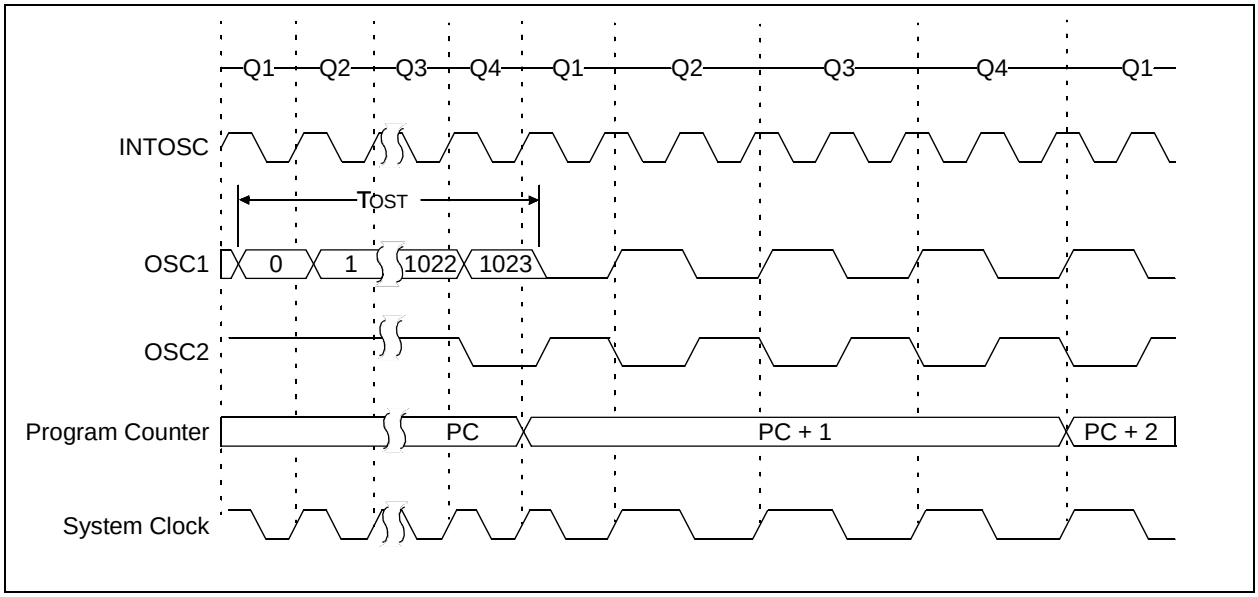
3.6.2 TWO-SPEED START-UP SEQUENCE

1. Wake-up from Power-on Reset or Sleep.
2. Instructions begin execution by the internal oscillator at the frequency set in the IRCF bits (OSCCON<6:4>).
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.

3.6.3 CHECKING EXTERNAL/INTERNAL CLOCK STATUS

Checking the state of the OSTS bit (OSCCON<3>) will confirm if the PIC16F688 is running from the external clock source as defined by the FOSC bits in the Configuration Word (CONFIG) or the internal oscillator.

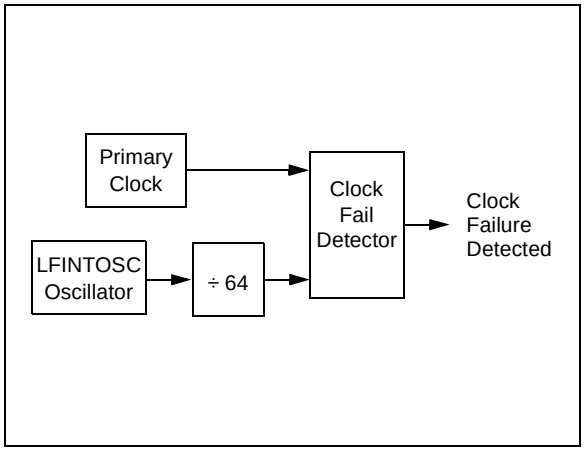
FIGURE 3-7: TWO-SPEED START-UP



3.7 Fail-Safe Clock Monitor

The Fail-Safe Clock Monitor (FSCM) is designed to allow the device to continue to operate in the event of an oscillator failure. The FSCM can detect oscillator failure at any point after the device has exited a Reset or Sleep condition and the Oscillator Start-up Timer (OST) has expired.

FIGURE 3-8: FSCM BLOCK DIAGRAM



The FSCM function is enabled by setting the FCMEN bit in the Configuration Word (CONFIG). It is applicable to all external clock options (LP, XT, HS, EC, RC or IO modes).

In the event of an external clock failure, the FSCM will set the OSFIF bit (PIR1<2>) and generate an oscillator fail interrupt if the OSFIE bit (PIE1<2>) is set. The device will then switch the system clock to the internal oscillator. The system clock will continue to come from the internal oscillator unless the external clock recovers and the Fail-Safe condition is exited.

The frequency of the internal oscillator will depend upon the value contained in the IRCF bits (OSCCON<6:4>). Upon entering the Fail-Safe condition, the OSTS bit (OSCCON<3>) is automatically cleared to reflect that the internal oscillator is active and the WDT is cleared. The SCS bit (OSCCON<0>) is not updated. Enabling FSCM does not affect the LTS bit.

The FSCM sample clock is generated by dividing the INTRC clock by 64. This will allow enough time between FSCM sample clocks for a system clock edge to occur. Figure 3-8 shows the FSCM block diagram.

On the rising edge of the sample clock, the monitoring latch (CM = 0) will be cleared. On a falling edge of the primary system clock, the monitoring latch will be set (CM = 1). In the event that a falling edge of the sample clock occurs, and the monitoring latch is not set, a clock failure has been detected. The assigned internal oscillator is enabled when FSCM is enabled as reflected by the IRCF.

Note:	Two-Speed Start-up is automatically enabled when the Fail-Safe Clock Monitor mode is enabled.

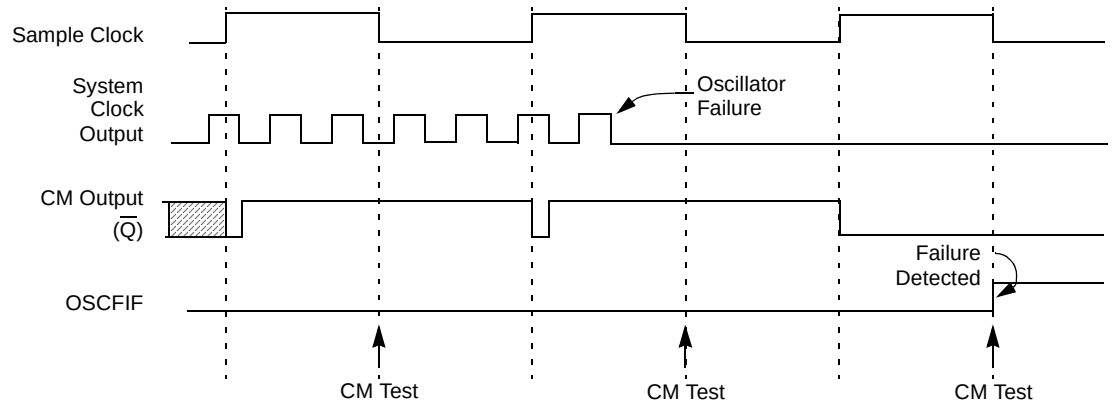
Note:	Primary clocks with a frequency ≤ ~488 Hz will be considered failed by the FSCM. A slow starting oscillator can cause an FSCM interrupt.

3.7.1 FAIL-SAFE CONDITION CLEARING

The Fail-Safe condition is cleared after a Reset, the execution of a SLEEP instruction, or a modification of the SCS bit. While in Fail-Safe condition, the PIC16F688 uses the internal oscillator as the system clock source. The IRCF bits (OSCCON<6:4>) can be modified to adjust the internal oscillator frequency without exiting the Fail-Safe condition.

The Fail-Safe condition must be cleared before the OSFIF flag can be cleared.

FIGURE 3-9: FSCM TIMING DIAGRAM



Note: The system clock is normally at a much higher frequency than the sample clock. The relative frequencies in this example have been chosen for clarity.

3.7.2 RESET OR WAKE-UP FROM SLEEP

The FSCM is designed to detect oscillator failure at any point after the device has exited a Reset or Sleep condition and the Oscillator Start-up Timer (OST) has expired. If the external clock is EC or RC mode, monitoring will begin immediately following these events.

For LP, XT or HS mode the external oscillator may require a start-up time considerably longer than the FSCM sample clock time, a false clock failure may be detected (see Figure 3-9). To prevent this, the internal oscillator is automatically configured as the system clock and functions until the external clock is stable (the OST has timed out). This is identical to Two-Speed Start-up mode. Once the external oscillator is stable, the LFINTOSC returns to its role as the FSCM source.

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 (OSCCON<3>) to verify the oscillator start-up and system clock switchover has successfully completed.

REGISTER 3-2: OSCCON – OSCILLATOR CONTROL REGISTER (ADDRESS: 8Fh)

U-0	R/W-1	R/W-1	R/W-0	R-1	R-0	R-0	R/W-0
—	IRCF2	IRCF1	IRCF0	OSTS(1)	HTS	LTS	SCS
bit 7							bit 0

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

bit 6-4 **IRCF<2:0>:** Internal Oscillator Frequency Select bits
000 = 31 kHz
001 = 125 kHz
010 = 250 kHz
011 = 500 kHz
100 = 1 MHz
101 = 2 MHz
110 = 4 MHz
111 = 8 MHz

bit 3 **OSTS:** Oscillator Start-up Time-out Status bit
1 = Device is running from the external system clock defined by FOSC<2:0>
0 = Device is running from the internal system clock (HFINTOSC or LFINTOSC)

bit 2 **HTS:** HFINTOSC (High Frequency – 8 MHz to 125 kHz) Status bit
1 = HFINTOSC is stable
0 = HFINTOSC is not stable

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

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.

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

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

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOD	Value on all other Resets
0Ch	PIR1	EEIF	ADIF	RCIF	C2IF	C1IF	OSFIF	TXIF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	EEIE	ADIE	RCIE	C2IE	C1IE	OSFIE	TXIE	TMR1IE	0000 0000	0000 0000
8Fh	OSCCON	—	IRCF2	IRCF1	IRCF0	OSTS(2)	HTS	LTS	SCS	-110 x000	-110 x000
90h	OSCTUNE	—	—	—	TUN4	TUN3	TUN2	TUN1	TUN0	---0 0000	---u uuuu
2007h(1)	CONFIG	$\overline{CPD}$	$\overline{CP}$	MCLRE	$\overline{PWRTE}$	WDTE	FOSC2	FOSC1	FOSC0	—	—

Legend: x = unknown, u = unchanged, — = unimplemented locations read as '0'. Shaded cells are not used by oscillators.
Note 1: See Register 11-1 for operation of all Configuration Word bits.
2: See Register 3-2 for details.

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.

Note:	Additional information on I/O ports may be found in the "*PICmicro® Mid-Range MCU Family Reference Manual*" (DS33023).

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 Hi-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 (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'.

EXAMPLE 4-1: INITIALIZING PORTA

```
BCF    STATUS,RP0      ;Bank 0
CLRF   PORTA           ;Init PORTA
MOVLW  07h             ;Set RA<2:0> to
MOVWF  CMCON0          ;digital I/O
BSF    STATUS,RP0      ;Bank 1
CLRF   ANSEL           ;digital I/O
MOVLW  0Ch             ;Set RA<3:2> as inputs
MOVWF  TRISA           ;and set RA<5:4,1:0>
                       ;as outputs
BCF    STATUS,RP0      ;Bank 0
```

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 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 4-3. 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 $\overline{RAPU}$ bit (Option<7>). A weak pull-up is automatically enabled for RA3 when configured as $\overline{MCLR}$ and disabled when RA3 is an I/O. There is no software control of the $\overline{MCLR}$ pull-up.

REGISTER 4-1: PORTA – PORTA REGISTER (ADDRESS: 05h OR 105h)

U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-0	R/W-0
—	—	RA5	RA4	RA3	RA2	RA1	RA0
bit 7							bit 0

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

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

1 = Port pin is > V_{IH}

0 = Port pin is < V_{IL}

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

REGISTER 4-2: TRISA – PORTA TRI-STATE REGISTER (ADDRESS: 85h OR 185h)

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

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'.

2: TRISA<5:4> always reads '1' in XT, HS and LP OSC modes.

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

REGISTER 4-3: WPUA – WEAK PULL-UP REGISTER (ADDRESS: 95h)

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

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

bit 5-4 **WPUA<5:4>:** Weak Pull-up Register bits

1 = Pull-up enabled
0 = Pull-up disabled

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

bit 2-0 **WPUA<2:0>:** Weak Pull-up Register 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.

2: The weak pull-up device is automatically disabled if the pin is in output mode (TRISA = 0).

3: The RA3 pull-up is enabled when configured as $\overline{\text{MCLR}}$ and disabled as an I/O in the Configuration Word.

4: WPUA<5:4> always reads '1' in XT, HS and LP OSC modes.

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

4.2.2 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 4-4. 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 $\overline{MCLR}$ nor BOD 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 4-4: IOCA – INTERRUPT-ON-CHANGE PORTA REGISTER (ADDRESS: 96h)

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

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.

2: IOCA<5:4> always reads '1' in XT, HS and LP OSC modes.

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

4.2.3 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 (PCON<5>). 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 (INTCON<7>), the device will either jump to the interrupt vector (0004h) or execute the next instruction when the interrupt event occurs. See **Section 4.2.2 "INTERRUPT-ON-CHANGE"** and **Section 11.5.3 "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.

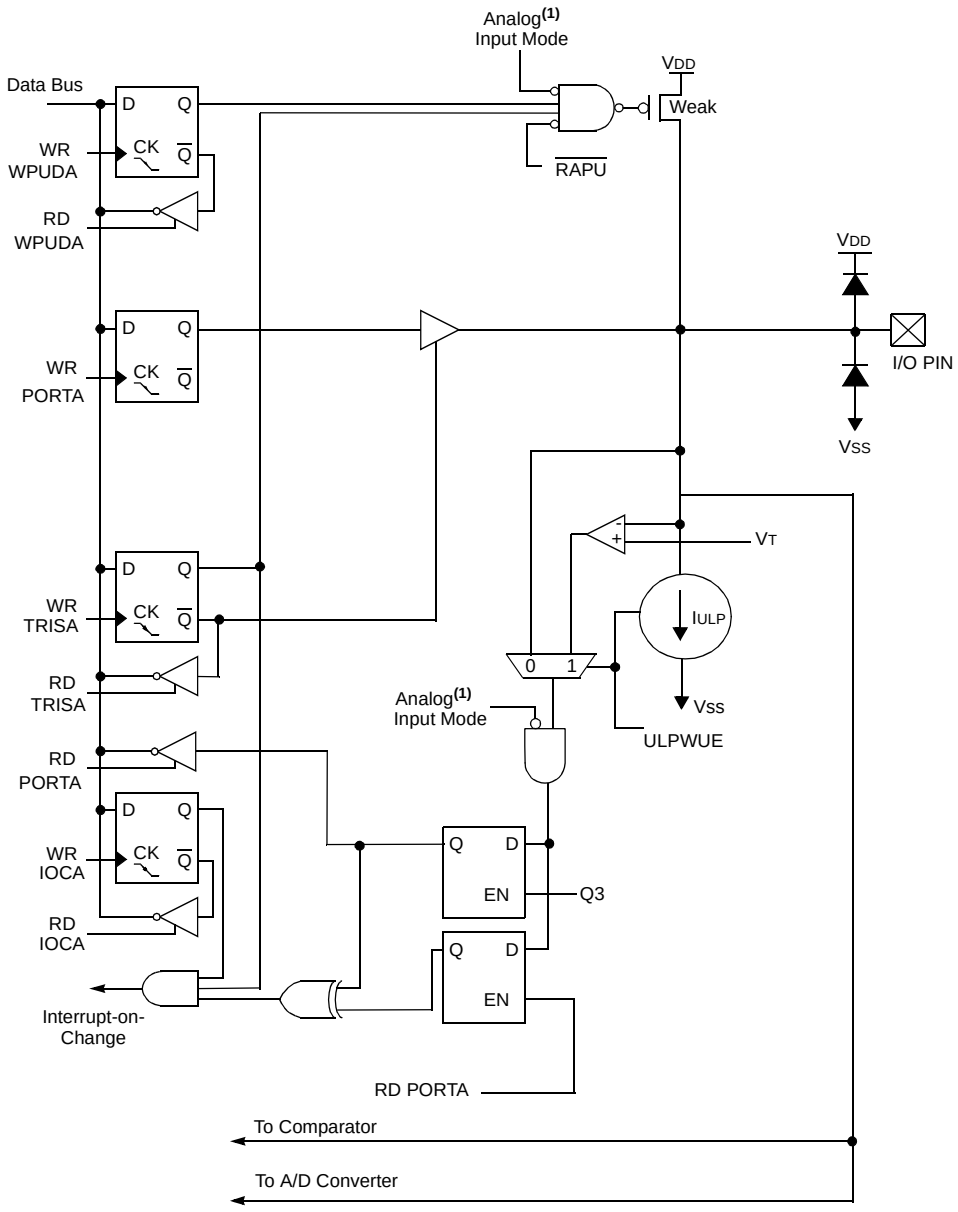
The series resistor provides overcurrent protection for the RA0 pin and can allow for software calibration of the time-out. (see Figure 4-1). A timer can be used to measure the charge time and discharge time of the capacitor. The charge time can then be adjusted to provide the desired interrupt delay. 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

```
BCF     STATUS,RP0      ;Bank 0
BSF     PORTA,0         ;Set RA0 data latch
MOVLW   H'7'            ;Turn off
MOVWF   CMCON0          ; comparators
BSF     STATUS,RP0      ;Bank 1
BCF     ANSEL,0         ;RA0 to digital I/O
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
```


FIGURE 4-1: BLOCK DIAGRAM OF RA0



Note 1: Comparator mode and ANSEL determines analog input mode.

4.2.4 PIN DESCRIPTIONS AND DIAGRAMS

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

4.2.4.1 RA0/AN0/C1IN+/ICSPDAT/ULPWU

Figure • shows the diagram for this pin. The RA0 pin is configurable to function as one of the following:

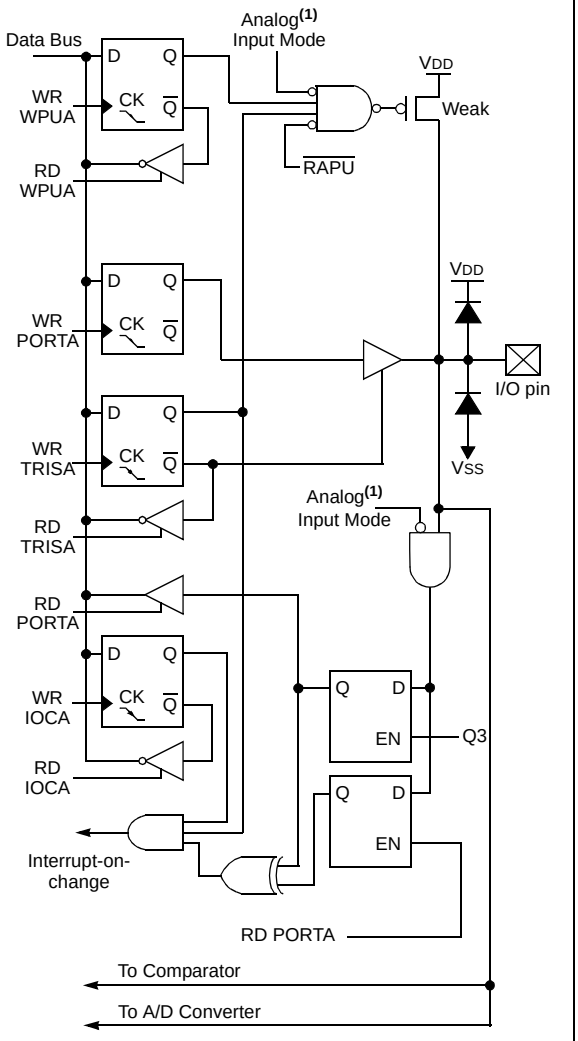
- a general purpose I/O
- an analog input for the A/D
- an analog input to the comparator
- an analog input to the Ultra Low-Power Wake-up
- In-Circuit Serial Programming™ data

4.2.4.2 RA1/AN1/C1IN-/VREF/ICSPCLK

Figure • shows the diagram for this pin. The RA1 pin is configurable to function as one of the following:

- a general purpose I/O
- an analog input for the A/D
- an analog input to the comparator
- a voltage reference input for the A/D
- In-Circuit Serial Programming clock

FIGURE 4-2: BLOCK DIAGRAM OF RA1



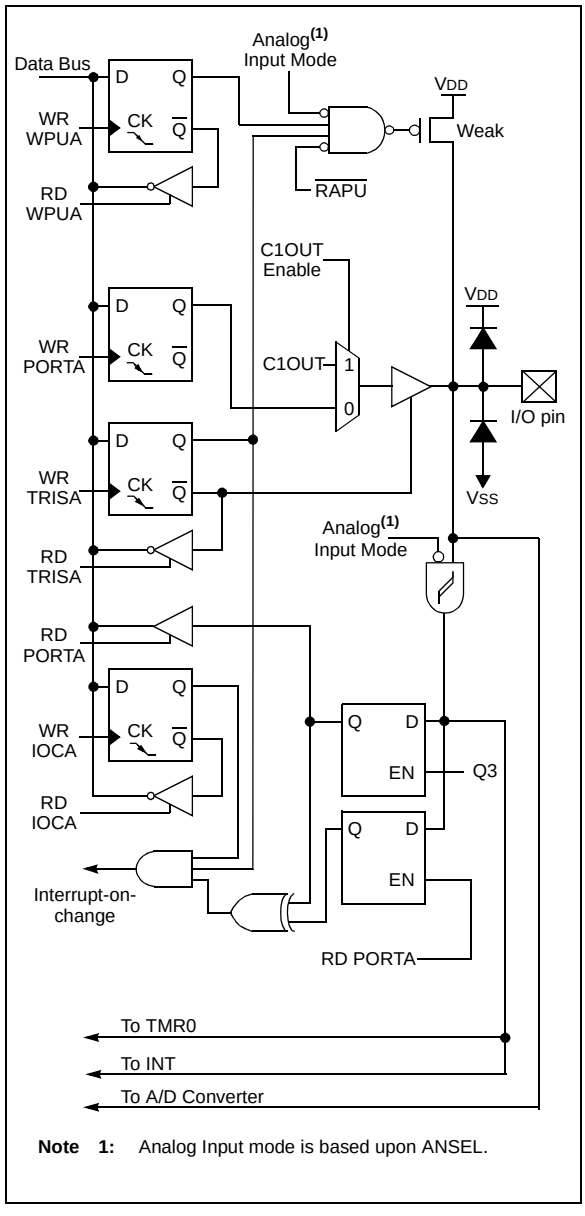
Note 1: Comparator mode and ANSEL determines analog input mode.

4.2.4.3 RA2/AN2/T0CKI/INT/C1OUT

Figure 4-3 shows the diagram for this pin. The RA2 pin is configurable to function as one of the following:

- a general purpose I/O
- an analog input for the A/D
- the clock input for TMR0
- an external edge triggered interrupt
- a digital output from the comparator

FIGURE 4-3: BLOCK DIAGRAM OF RA2



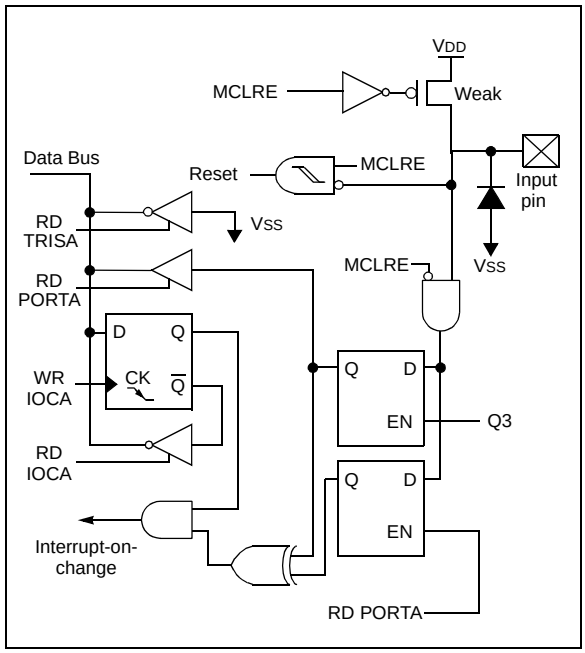
Note 1: Analog Input mode is based upon ANSEL.

4.2.4.4 RA3/$\overline{\text{MCLR}}$/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

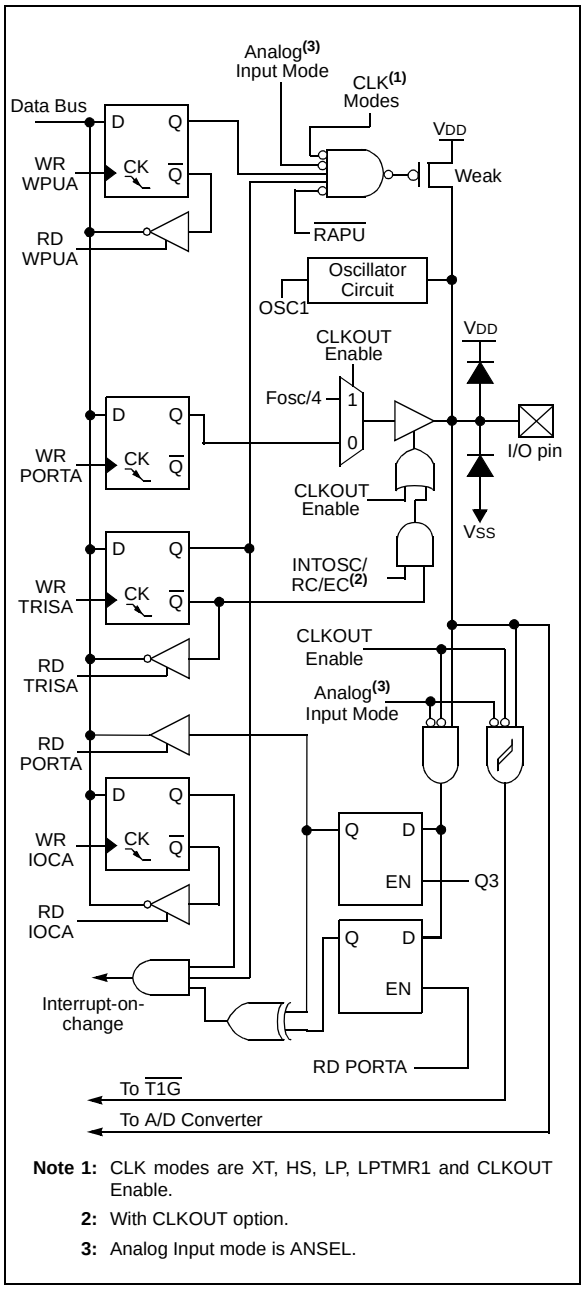


4.2.4.5 RA4/AN3/$\overline{T1G}$/OSC2/CLKOUT

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 TMR1 gate input
- a crystal/resonator connection
- a clock output

FIGURE 4-5: BLOCK DIAGRAM OF RA4



Note 1: CLK modes are XT, HS, LP, LPTMR1 and CLKOUT Enable.

2: With CLKOUT option.

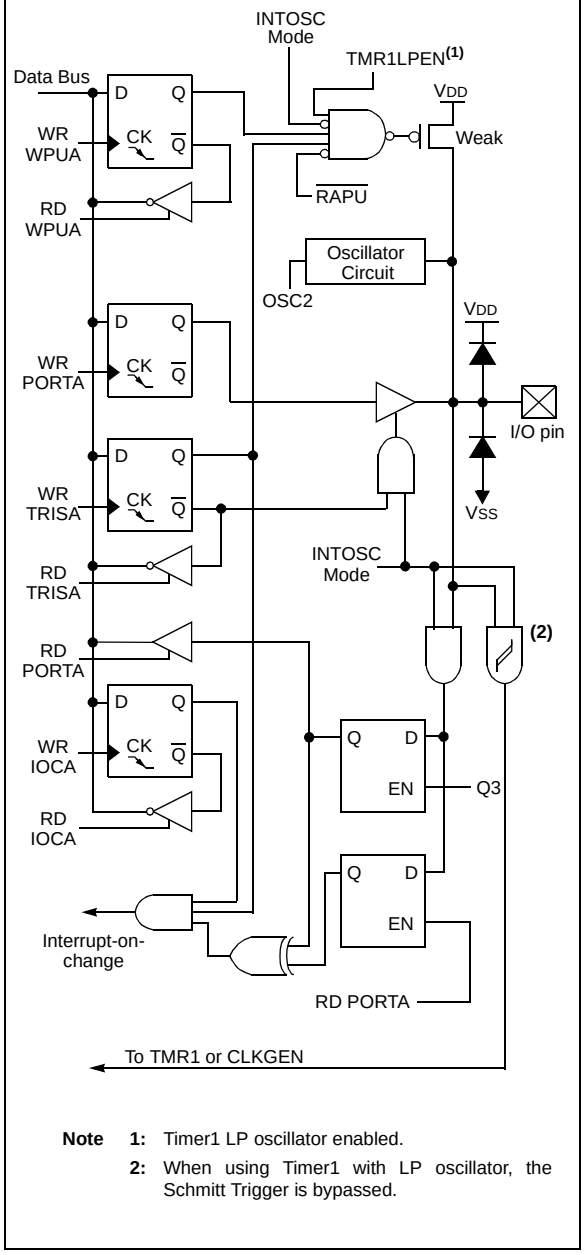
3: Analog Input mode is ANSEL.

4.2.4.6 RA5/T1CKI/OSC1/CLKIN

Figure 4-6 shows the diagram for this pin. The RA5 pin is configurable to function as one of the following:

- a general purpose I/O
- a TMR1 clock input
- a crystal/resonator connection
- a clock input

FIGURE 4-6: BLOCK DIAGRAM OF RA5



Note 1: Timer1 LP oscillator enabled.

2: When using Timer1 with LP oscillator, the Schmitt Trigger is bypassed.

TABLE 4-1: SUMMARY OF REGISTERS ASSOCIATED WITH PORTA

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOD	Value on all other Resets
05h/105h	PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	--xx xx00	--uu uu00
0Bh/8Bh	INTCON	GIE	PEIE	T0IE	INTE	RAIE	T0IF	INTF	RAIF	0000 0000	0000 0000
19h	CMCON0	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	0000 0000	0000 0000
81h	OPTION_REG	$\overline{\text{RAPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
85h/185h	TRISA	—	—	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	--11 1111	--11 1111
91h	ANSEL	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0	1111 1111	1111 1111
95h	WPUA	—	—	WPUA5	WPUA4	—	WPUA2	WPUA1	WPUA0	--11 -111	--11 -111
96h	IOCA	—	—	IOCA5	IOCA4	IOCA3	IOCA2	IOCA1	IOCA0	--00 0000	--00 0000

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

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, refer to the appropriate section in this data sheet.

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'.

EXAMPLE 4-3: INITIALIZING PORTC

```
BCF     STATUS,RP0      ;Bank 0
CLRF    PORTC           ;Init PORTC
MOVLW   07h             ;Set RC<4,1:0> to
MOVWF   CMCON0          ;digital I/O
BSF     STATUS,RP0      ;Bank 1
CLRF    ANSEL           ;digital I/O
MOVLW   0Ch             ;Set RC<3:2> as inputs
MOVWF   TRISC           ;and set RC<5:4,1:0>
                        ;as outputs
BCF     STATUS,RP0      ;Bank 0
```

4.3.1 RC0/AN4/C2IN+

The RC0 is configurable to function as one of the following:

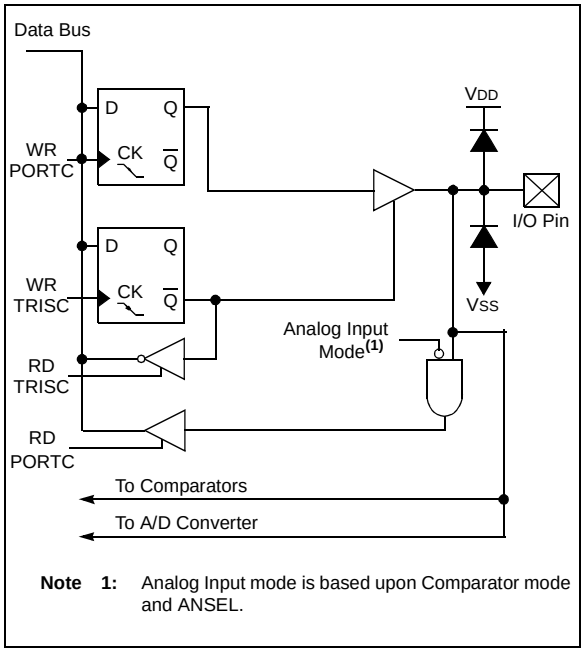
- a general purpose I/O
- an analog input for the A/D Converter
- an analog input to the comparator

4.3.2 RC1/AN5/C2IN-

The RC1 is configurable to function as one of the following:

- a general purpose I/O
- an analog input for the A/D Converter
- an analog input to the comparator

FIGURE 4-7: BLOCK DIAGRAM OF RC0 AND RC1



Note 1: Analog Input mode is based upon Comparator mode and ANSEL.

4.3.3 RC2/AN6

The RC2 is configurable to function as one of the following:

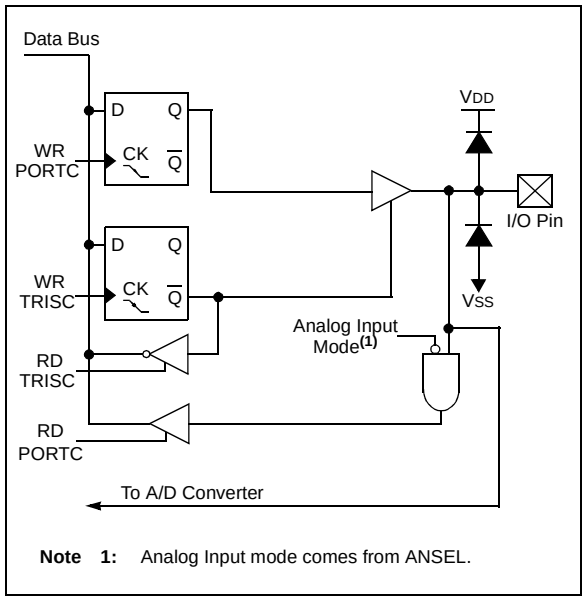
- a general purpose I/O
- an analog input for the A/D Converter

4.3.4 RC3/AN7

The RC3 is configurable to function as one of the following:

- a general purpose I/O
- an analog input for the A/D Converter

FIGURE 4-8: BLOCK DIAGRAM OF RC2 AND RC3



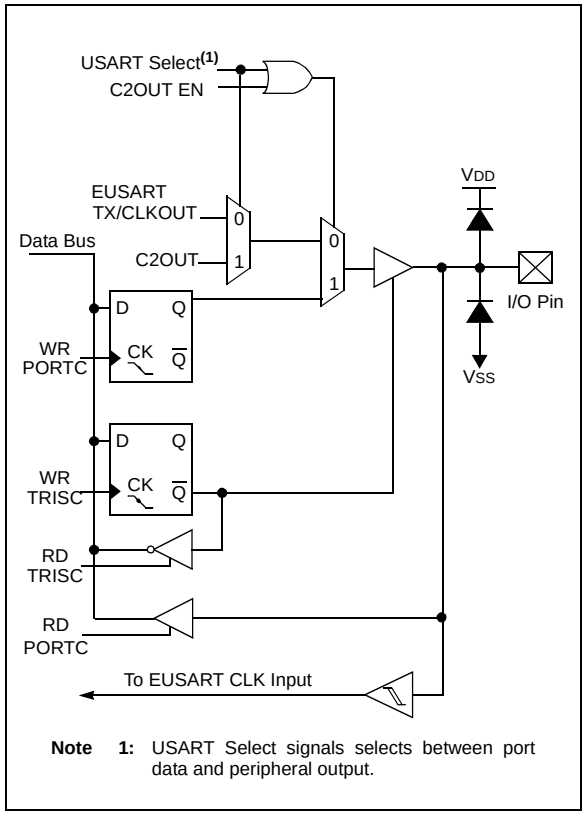
Note 1: Analog Input mode comes from ANSEL.

4.3.5 RC4/C2OUT/TX/CK

The RC4 is configurable to function as one of the following:

- a general purpose I/O
- a digital output from the comparator
- a digital I/O for the EUSART

FIGURE 4-9: BLOCK DIAGRAM OF RC4



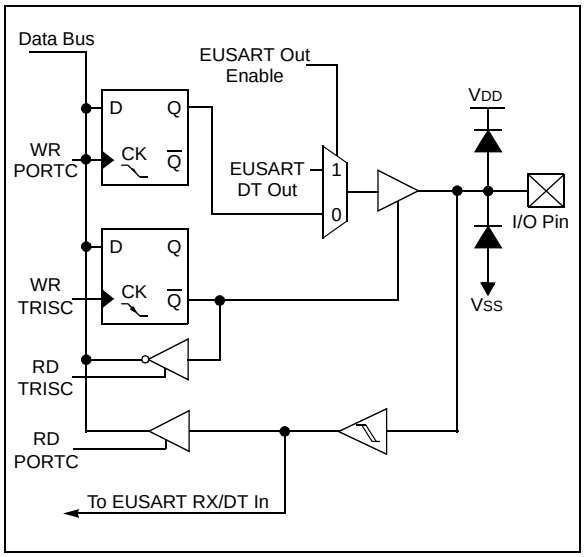
Note 1: USART Select signals selects between port data and peripheral output.

4.3.6 RC5/RX/DT

The RC5 is configurable to function as one of the following:

- a general purpose I/O
- a digital I/O for the EUSART

FIGURE 4-10: BLOCK DIAGRAM OF RC5 PIN



REGISTER 4-5: PORTC – PORTC REGISTER (ADDRESS: 07h OR 107h)

U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-0	R/W-0
—	—	RC5	RC4	RC3	RC2	RC1	RC0
bit 7							bit 0

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

bit 5-0: **PORTC<5:0>**: General Purpose I/O Pin bits

1 = Port pin is >V_{IH}

0 = Port pin is <V_{IL}

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

REGISTER 4-6: TRISC – PORTC TRI-STATE REGISTER (ADDRESS: 87h OR 187h)

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

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

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

TABLE 4-2: SUMMARY OF REGISTERS ASSOCIATED WITH PORTC

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOD	Value on all other Resets
07h	PORTC	—	—	RC5	RC4	RC3	RC2	RC1	RC0	--xx xx00	--uu uu00
19h	CMCON0	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	0000 0000	0000 0000
87h	TRISC	—	—	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	--11 1111	--11 1111
91h	ANSEL	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0	1111 1111	1111 1111

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

NOTES:

5.0 TIMER0 MODULE

The Timer0 module timer/counter has the following features:

- 8-bit timer/counter
- Readable and writable
- 8-bit software programmable prescaler
- Internal or external clock select
- Interrupt on overflow from FFh to 00h
- Edge select for external clock

Figure 5-1 is a block diagram of the Timer0 module and the prescaler shared with the WDT.

Note:	Additional information on the Timer0 module is available in the "*PICmicro® Mid-Range MCU Family Reference Manual*" (DS33023).

5.1 Timer0 Operation

Timer mode is selected by clearing the T0CS bit (OPTION_REG<5>). In Timer mode, the Timer0 module will increment every instruction cycle (without prescaler). If TMR0 is written, the increment is inhibited for the following two instruction cycles. The user can work around this by writing an adjusted value to the TMR0 register.

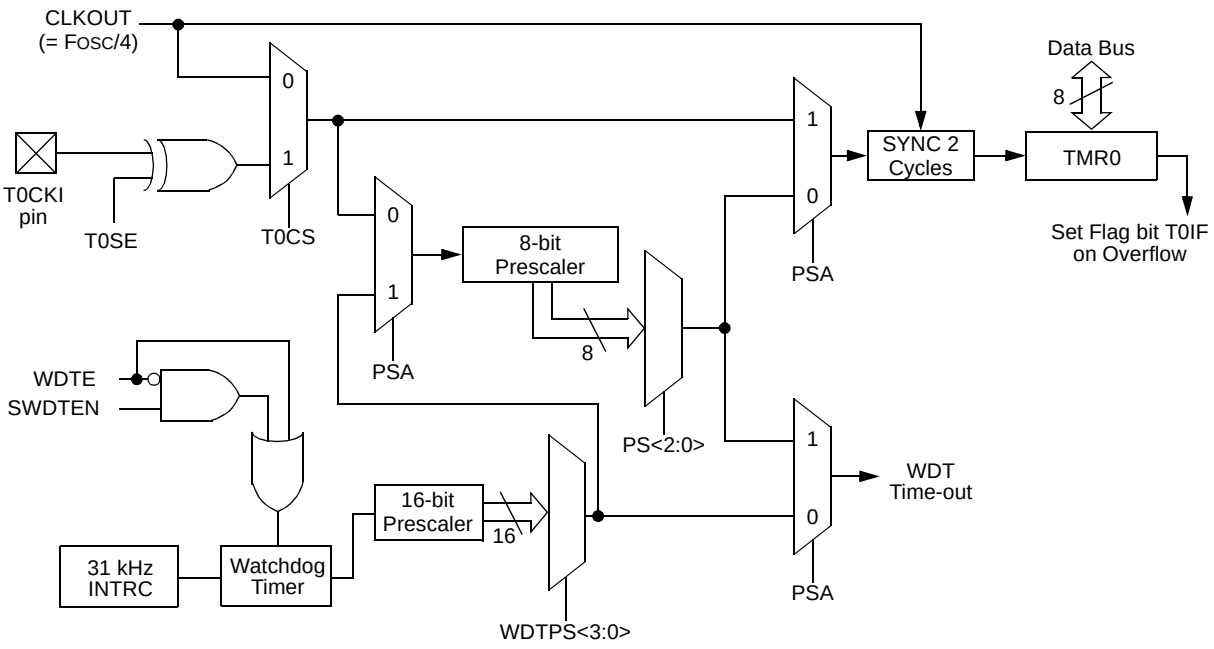
Counter mode is selected by setting the T0CS bit (OPTION_REG<5>). In this mode, the Timer0 module will increment either on every rising or falling edge of pin RA2/T0CKI. The incrementing edge is determined by the source edge (T0SE) control bit (OPTION_REG<4>). Clearing the T0SE bit selects the rising edge.

Note:	Counter mode has specific external clock requirements. Additional information on these requirements is available in the *"PICmicro® Mid-Range MCU Family Reference Manual"* (DS33023).

5.2 Timer0 Interrupt

A Timer0 interrupt is generated when the TMR0 register timer/counter overflows from FFh to 00h. This overflow sets the T0IF bit (INTCON<2>). The interrupt can be masked by clearing the T0IE bit (INTCON<5>). The T0IF bit must be cleared in software by the Timer0 module Interrupt Service Routine before re-enabling this interrupt. The Timer0 interrupt cannot wake the processor from Sleep, since the timer is shut off during Sleep.

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



Note 1: T0SE, T0CS, PSA, PS<2:0> are bits in the Option register, WDTPS<3:0> are bits in the WDTCON register.

5.3 Using Timer0 with an External Clock

When no prescaler is used, the external clock input is the same as the prescaler output. The synchronization of T0CKI, with the internal phase clocks, is accomplished by sampling the prescaler output on the Q2 and Q4 cycles of the internal phase clocks. Therefore, it is necessary for T0CKI to be high for at least 2 TOSC (and a small RC delay of 20 ns) and low for at least 2 TOSC (and a small RC delay of 20 ns). Refer to the electrical specification of the desired device.

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'.

REGISTER 5-1: OPTION_REG – OPTION REGISTER (ADDRESS: 81h OR 181h)

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
$\overline{\text{RAPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7							bit 0

bit 7 **$\overline{\text{RAPU}}$:** PORTA Pull-up Enable bit
1 = PORTA pull-ups are disabled
0 = PORTA pull-ups are enabled by individual port latch values in WPUA register

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:** TMR0 Clock Source Select bit
1 = Transition on RA2/T0CKI pin
0 = Internal instruction cycle clock (CLKOUT)

bit 4 **T0SE:** TMR0 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	TMR0 Rate	WDT Rate(1)
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 for the PIC16F688. See **Section 11.7 "Watchdog Timer (WDT)"** for more information.

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

5.4 Prescaler

An 8-bit counter is available as a prescaler for the Timer0 module, or as a postscaler for the Watchdog Timer. For simplicity, this counter will be referred to as “prescaler” throughout this data sheet. The prescaler assignment is controlled in software by the control bit PSA (OPTION_REG<3>). Clearing the PSA bit will assign the prescaler to Timer0. Prescale values are selectable via the PS<2:0> bits (OPTION_REG<2:0>).

The prescaler is not readable or writable. When assigned to the Timer0 module, all instructions writing to the TMR0 register (e.g., `CLRF 1, MOVWF 1, BSF 1, x....etc.`) will clear the prescaler. When assigned to WDT, a `CLRWDT` instruction will clear the prescaler along with the Watchdog Timer.

5.4.1 SWITCHING PRESCALER ASSIGNMENT

The prescaler assignment is fully under software control (i.e., it can be changed “on-the-fly” during program execution). To avoid an unintended device Reset, the following instruction sequence (Example 5-1 and Example 5-2) must be executed when changing the prescaler assignment from Timer0 to WDT.

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

```
BCF     STATUS,RP0      ;Bank 0
CLRWDT                  ;Clear WDT
CLRF    TMR0            ;Clear TMR0 and
                        ; prescaler
BSF     STATUS,RP0      ;Bank 1

MOVLW   b’00101111’     ;Required if desired
MOVWF   OPTION_REG      ; PS2:PS0 is
CLRWDT                  ; 000 or 001
                        ;
MOVLW   b’00101xxx’     ;Set postscaler to
MOVWF   OPTION_REG      ; desired WDT rate
BCF     STATUS,RP0      ;Bank 0
```

To change prescaler from the WDT to the TMR0 module, use the sequence shown in Example 5-2. This precaution must be taken even if the WDT is disabled.

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

```
CLRWDT                  ;Clear WDT and
                        ; prescaler
BSF     STATUS,RP0      ;Bank 1

MOVLW   b’xxxx0xxx’     ;Select TMR0,
                        ; prescale, and
                        ; clock source
MOVWF   OPTION_REG      ;
BCF     STATUS,RP0      ;Bank 0
```

TABLE 5-1: REGISTERS ASSOCIATED WITH TIMER0

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOD	Value on all other Resets
01h	TMR0	Timer0 Module register								xxxx xxxx	uuuu uuuu
0Bh/8Bh	INTCON	GIE	PEIE	T0IE	INTE	RAIE	T0IF	INTF	RAIF	0000 0000	0000 0000
81h	OPTION_REG	$\overline{\text{RAPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
85h/185h	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.

NOTES:

6.0 TIMER1 MODULE WITH GATE CONTROL

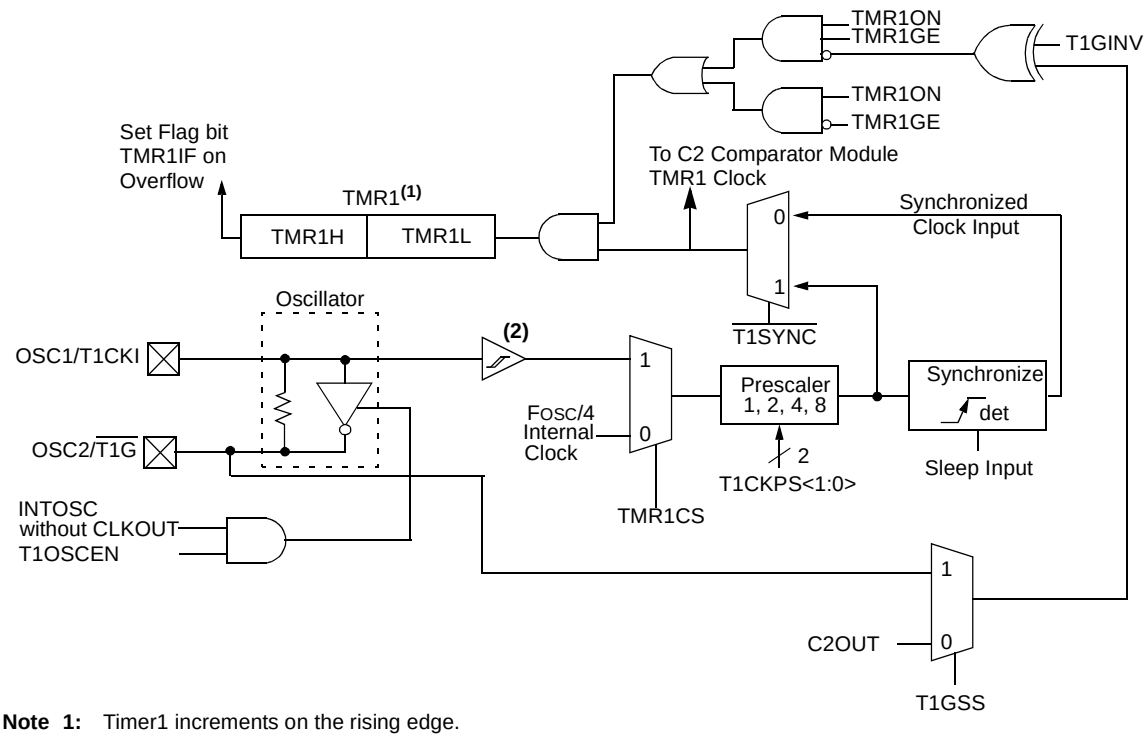
The PIC16F688 has a 16-bit timer. Figure 5-1 shows the basic block diagram of the Timer1 module. Timer1 has the following features:

- 16-bit timer/counter (TMR1H:TMR1L)
- Readable and writable
- Internal or external clock selection
- Synchronous or asynchronous operation
- Interrupt-on-overflow from FFFFh to 0000h
- Wake-up upon overflow (Asynchronous mode)
- Optional external enable input
  - Selectable gate source: $\overline{T1G}$ or C2 output (T1GSS)
  - Selectable gate polarity (T1GINV)
- Optional LP oscillator

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

Note:	Additional information on timer modules is available in the *"PICmicro® Mid-Range MCU Family Reference Manual"* (DS33023).

FIGURE 6-1: TIMER1 ON THE PIC16F688 BLOCK DIAGRAM



Note 1: Timer1 increments on the rising edge.

2: ST Buffer is low power type when using LP oscillator or high-speed type when using T1CKI.

6.1 Timer1 Modes of Operation

Timer1 can operate in one of three modes:

- 16-bit timer with prescaler
- 16-bit synchronous counter
- 16-bit asynchronous counter

In Timer mode, Timer1 is incremented on every instruction cycle. In Counter mode, 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.

In Counter and Timer modules, the counter/timer clock can be gated by the Timer1 gate, which can be selected as either the T1G pin or Comparator 2 output.

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.

6.2 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 (PIR1<0>) is set. To enable the interrupt on rollover, you must set these bits:

- Timer1 Interrupt Enable bit (PIE1<0>)
- PEIE bit (INTCON<6>)
- GIE bit (INTCON<7>)

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

Note:	The TMR1H:TTMR1L register pair and the TMR1IF bit should be cleared before enabling interrupts.

6.3 Timer1 Prescaler

Timer1 has four prescaler options allowing 1, 2, 4 or 8 divisions of the clock input. The T1CKPS bits (T1CON<5:4>) 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.

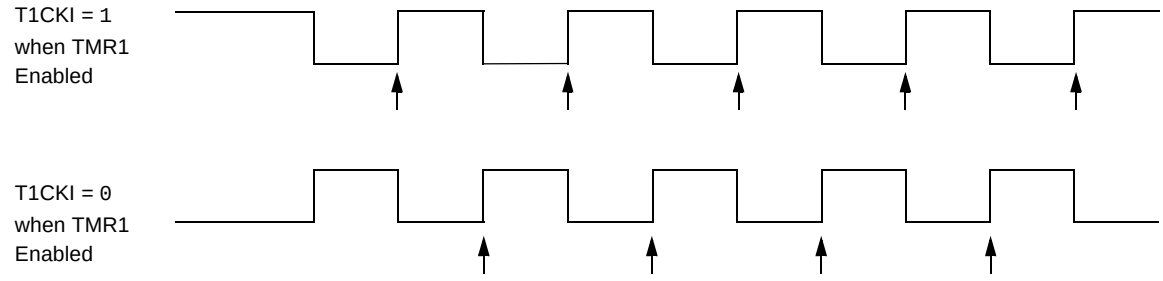
6.4 Timer1 Gate

Timer1 gate source is software configurable to be the T1G pin or the output of Comparator 2. This allows the device to directly time external events using T1G or analog events using Comparator 2. See CMCON1 (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).

Note:	TMR1GE bit (T1CON<6>) must be set to use either 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 (T1CON<7>), whether it originates from the T1G pin or Comparator 2 output. This configures Timer1 to measure either the active-high or active-low time between events.

FIGURE 6-2: TIMER1 INCREMENTING EDGE



Note 1: Arrows indicate counter increments.

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

REGISTER 6-1: T1CON – TIMER1 CONTROL REGISTER (ADDRESS: 10h)

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	TMR1GE	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON
bit 7							bit 0

bit 7 **T1GINV:** Timer1 Gate Invert bit(1)
1 = Timer1 gate is inverted
0 = Timer1 gate is not inverted

bit 6 **TMR1GE:** Timer1 Gate Enable bit(2)
If TMR1ON = 0:
This bit is ignored.
If TMR1ON = 1:
1 = Timer1 is on if Timer1 gate is not 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.

bit 2 **$\overline{\text{T1SYNC}}$:** 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 $\overline{\text{T1G}}$ pin or C2OUT, as selected by the T1GSS bit (CMCON1<1>), as a Timer1 gate source.

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

6.5 Timer1 Operation in Asynchronous Counter Mode

If control bit $\overline{\text{T1SYNC}}$ (T1CON<2>) 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:	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'.

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 timer register.

Reading the 16-bit value requires some care. Examples in the "*PICmicro® Mid-Range MCU Family Reference Manual*" (DS33023) show how to read and write Timer1 when it is running in Asynchronous mode.

6.6 Timer1 Oscillator

A crystal oscillator circuit is built-in between pins OSC1 (input) and OSC2 (amplifier output). It is enabled by setting control bit, T1OSCEN (T1CON<3>). The oscillator is a low power oscillator rated up to 32 kHz. It will continue to run during Sleep. It is primarily intended for a 32 kHz crystal. Table 3-1 shows the capacitor selection for the Timer1 oscillator.

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. As with the system LP oscillator, 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 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.

6.7 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:

- Timer1 must be on (T1CON<0>)
- TMR1IE bit (PIE1<0>) must be set
- PEIE bit (INTCON<6>) must be set

The device will wake-up on an overflow. If the GIE bit (INTCON<7>) is set, the device will wake-up and jump to the Interrupt Service Routine (0004h) on an overflow. If the GIE bit is clear, execution will continue with the next instruction.

TABLE 6-1: REGISTERS ASSOCIATED WITH TIMER1

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOD	Value on all other Resets
0Bh/ 8Bh	INTCON	GIE	PEIE	T0IE	INTE	RAIE	T0IF	INTF	RAIF	0000 0000	0000 0000
0Ch	PIR1	EEIF	ADIF	RCIF	C2IF	C1IF	OSFIF	TXIF	TMR1IF	0000 0000	0000 0000
0Eh	TMR1L	Holding register for the Least Significant Byte of the 16-bit TMR1 register								xxxx xxxx	uuuu uuuu
0Fh	TMR1H	Holding register for the Most Significant Byte of the 16-bit TMR1 register								xxxx xxxx	uuuu uuuu
10h	T1CON	T1GINV	TMR1GE	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON	0000 0000	uuuu uuuu
1Ah	CMCON1	—	—	—	—	—	—	T1GSS	C2SYNC	---- --10	---- --10
8Ch	PIE1	EEIE	ADIE	RCIE	C2IE	C1IE	OSFIE	TXIE	TMR1IE	0000 0000	0000 0000

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

7.0 COMPARATOR MODULE

The comparator module contains two analog comparators. The inputs to the comparators are multiplexed with I/O port pins RA0, RA1, RC0 and RC1, while the outputs are multiplexed to pins RA2 and RC4. An on-chip Comparator Voltage Reference (CV_{REF}) can also be applied to the inputs of the comparators.

The CMCON0 register (Register 7-1) controls the comparator input and output multiplexers. A block diagram of the various comparator configurations is shown in Figure 7-3.

REGISTER 7-1: CMCON0 – COMPARATOR CONFIGURATION REGISTER (ADDRESS: 19h)

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

bit 7 **C2OUT**: Comparator 2 Output bit
When C2INV = 0:
1 = C2 V_{IN}+ > C2 V_{IN}-
0 = C2 V_{IN}+ < C2 V_{IN}-
When C2INV = 1:
1 = C2 V_{IN}+ < C2 V_{IN}-
0 = C2 V_{IN}+ > C2 V_{IN}-

bit 6 **C1OUT**: Comparator 1 Output bit
When C1INV = 0:
1 = C1 V_{IN}+ > C1 V_{IN}-
0 = C1 V_{IN}+ < C1 V_{IN}-
When C1INV = 1:
1 = C1 V_{IN}+ < C1 V_{IN}-
0 = C1 V_{IN}+ > C1 V_{IN}-

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 = C1 V_{IN}- connects to RA0/AN0
C2 V_{IN}- connects to RC0/AN4
0 = C1 V_{IN}- connects to RA1/AN1
C2 V_{IN}- connects to RC1/AN5
When CM<2:0> = 001:
1 = C1 V_{IN}- connects to RA0/AN0
0 = C1 V_{IN}- connects to RA1/AN1

bit 2 **CM<2:0>**: Comparator Mode bits
Figure 7-3 shows the Comparator modes and CM<2:0> bit settings

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

7.1 Comparator Operation

A single comparator is shown in Figure 7-1 along with the relationship between the analog input levels and the digital output. When the analog input at VIN+ is less than the analog input VIN-, the output of the comparator is a digital low level. When the analog input at VIN+ is greater than the analog input VIN-, the output of the comparator is a digital high level. The shaded areas of the output of the comparator in Figure 7-1 represent the uncertainty due to input offsets and response time.

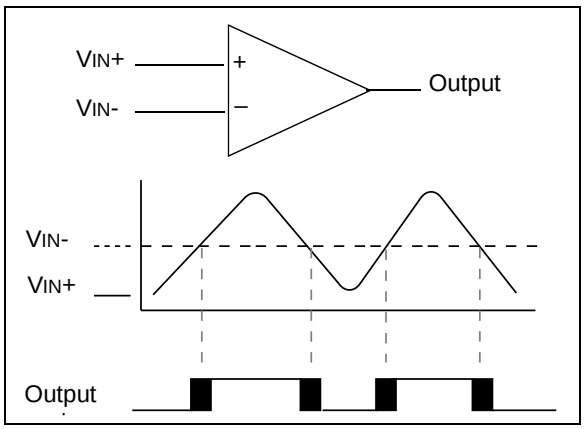
Note:	To use CIN+ and CIN- pins as analog inputs, the appropriate bits must be programmed in the CMCON0 (19h) register.

The polarity of the comparator output can be inverted by setting the CxINV bits (CMCON0<5:4>). 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	CINV	CxOUT
VIN- > VIN+	0	0
VIN- < VIN+	0	1
VIN- > VIN+	1	1
VIN- < VIN+	1	0

FIGURE 7-1: SINGLE COMPARATOR

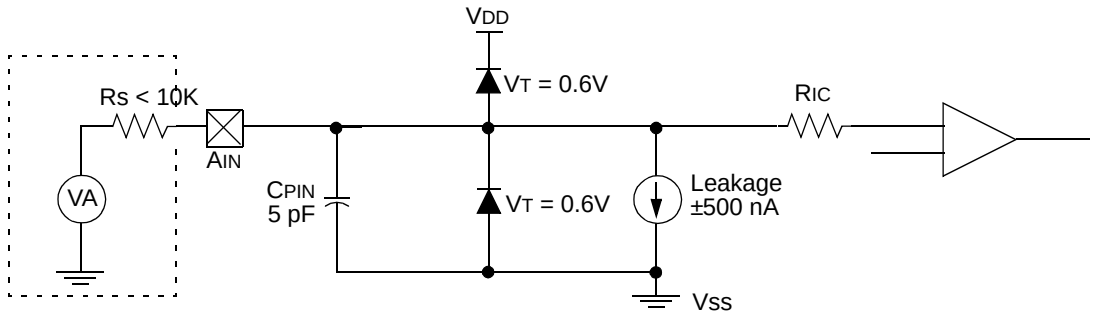


7.2 Analog Input Connection Considerations

A simplified circuit for an analog input is shown in Figure 7-2. Since the analog pins are connected to a digital output, they have reverse biased 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. Any external component connected to an analog input pin, such as a capacitor or a Zener diode, should have very little leakage current.

Note 1:	When reading the PORT register, all pins configured as analog inputs will read as a '0'. Pins configured as digital inputs will convert as analog inputs 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 7-2: ANALOG INPUT MODEL



Legend: CPIN = Input Capacitance
VT = Threshold Voltage
ILEAKAGE = Leakage Current at the pin due to various junctions
RIC = Interconnect Resistance
RS = Source Impedance
VA = Analog Voltage

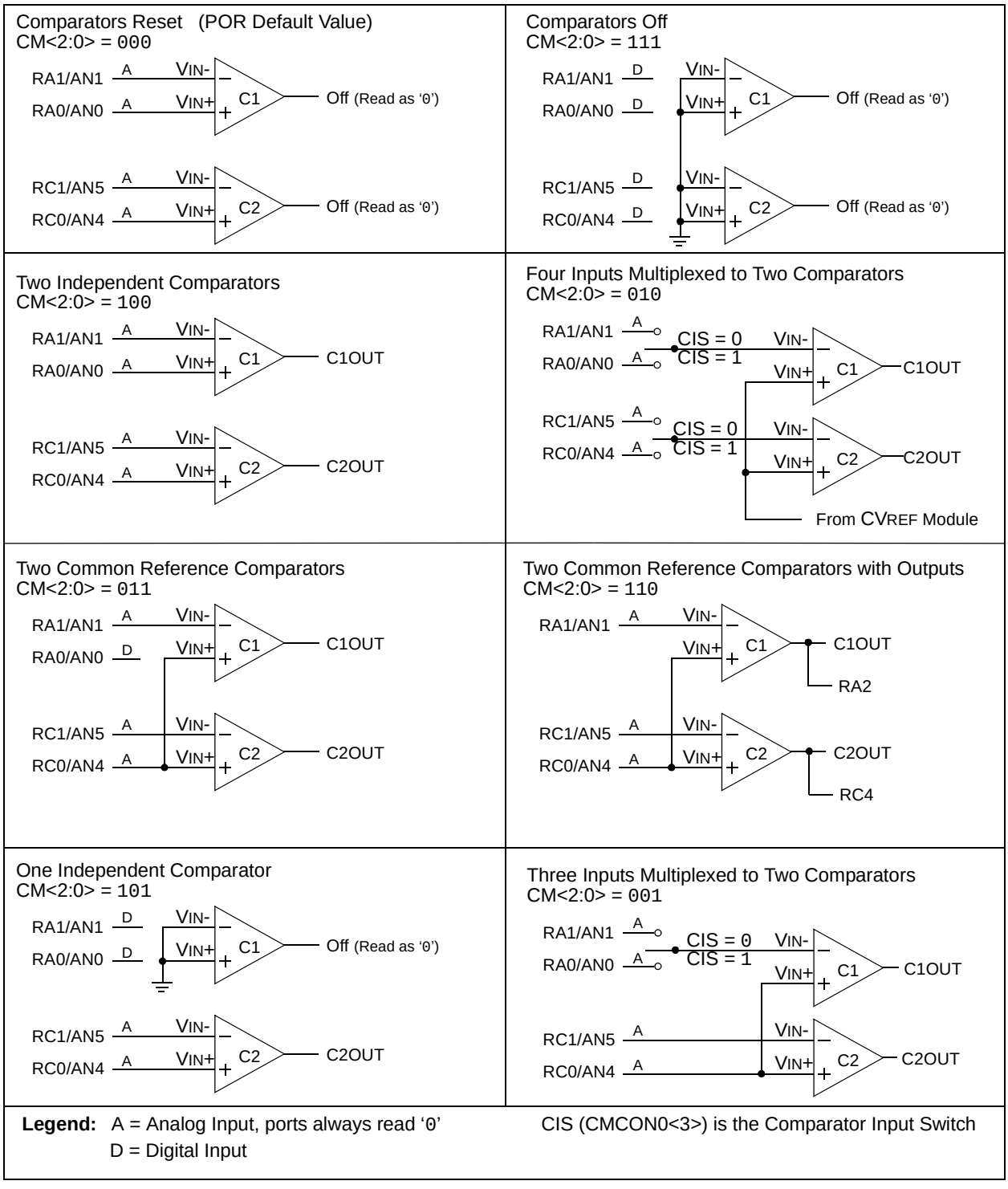
7.3 Comparator Configuration

There are eight modes of operation for the comparators. The CMCON0 register is used to select these modes. Figure 7-3 shows the eight possible modes.

If the Comparator mode is changed, the comparator output level may not be valid for the specified mode change delay shown in **Section 14.0 "Electrical Specifications"**.

Note:	Comparator interrupts should be disabled during a Comparator mode change. Otherwise, a false interrupt may occur.

FIGURE 7-3: COMPARATOR I/O OPERATING MODES



Legend: A = Analog Input, ports always read '0'
D = Digital Input
CIS (CMCON0<3>) is the Comparator Input Switch

FIGURE 7-4: MODIFIED COMPARATOR C1 OUTPUT BLOCK DIAGRAM

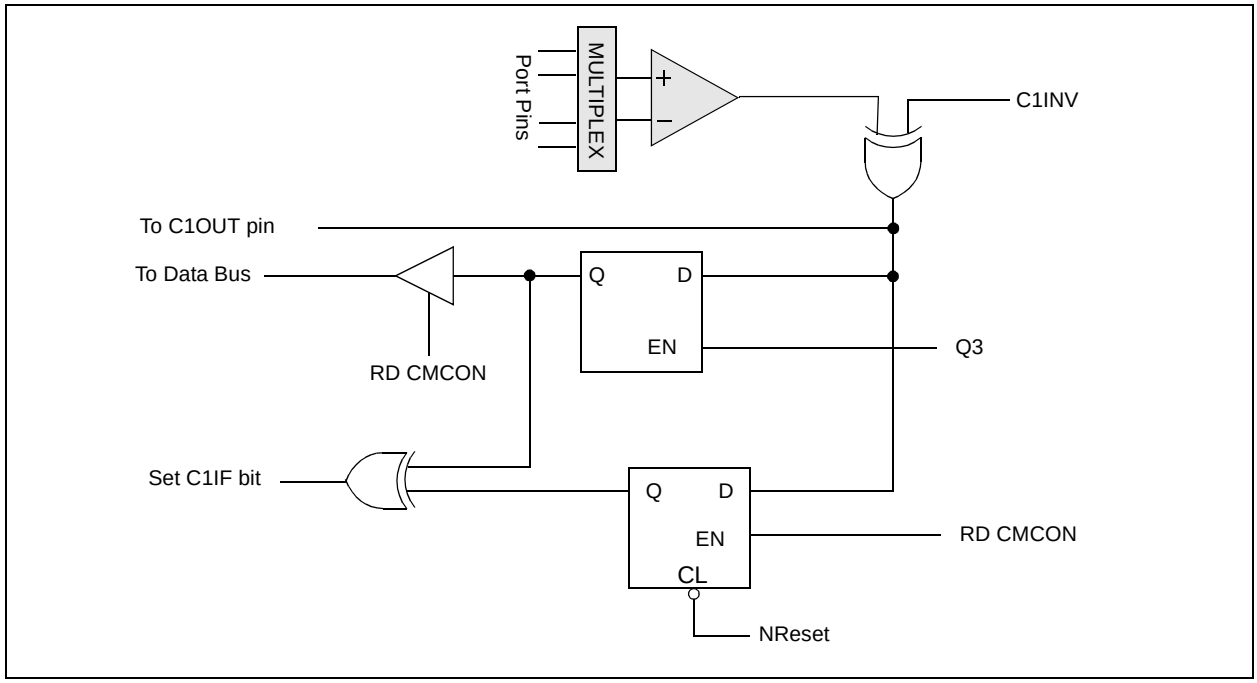
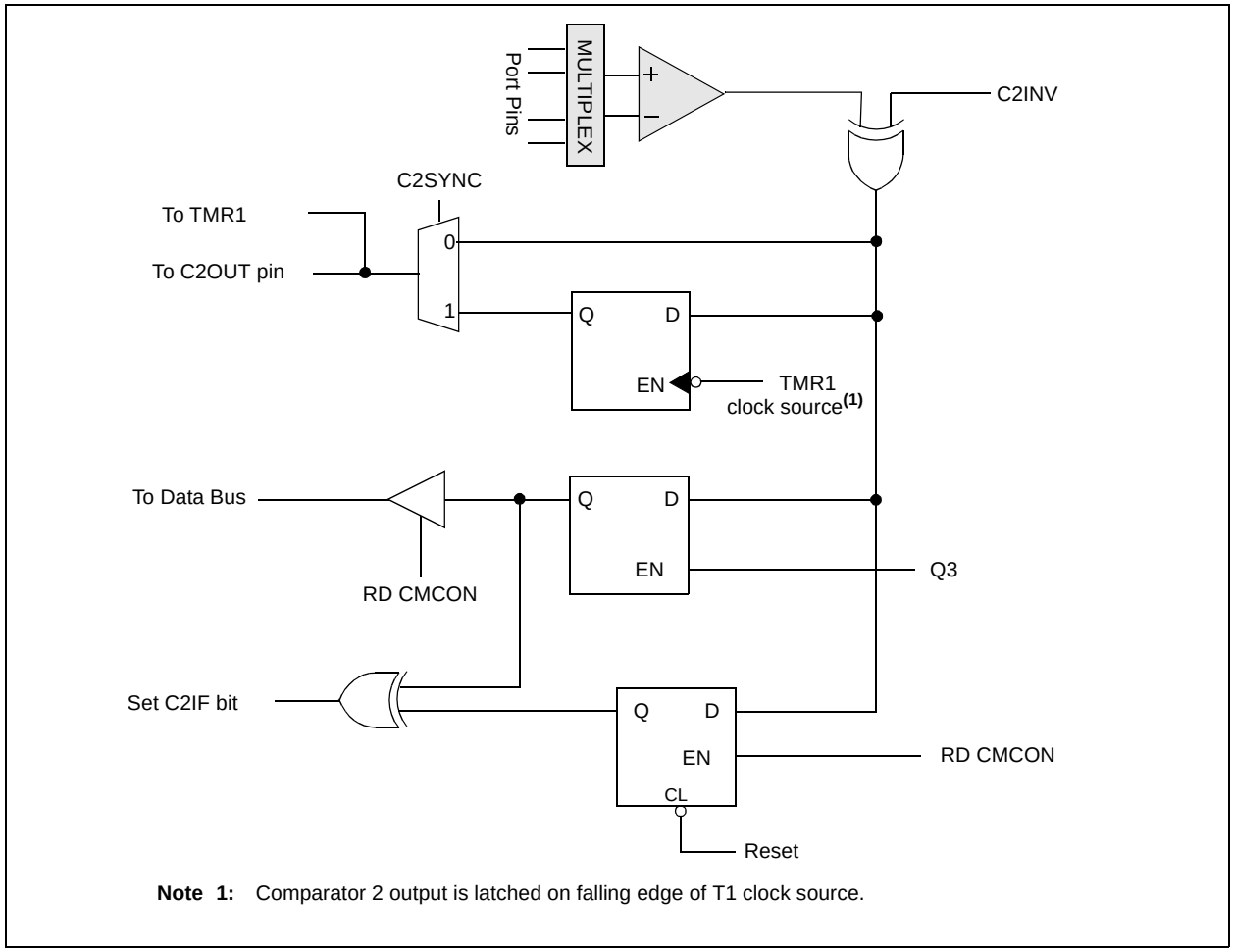


FIGURE 7-5: COMPARATOR C2 OUTPUT BLOCK DIAGRAM



Note 1: Comparator 2 output is latched on falling edge of T1 clock source.

REGISTER 7-2: CMCON1 – COMPARATOR CONFIGURATION REGISTER (ADDRESS: 1Ah)

U-0	U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0
—	—	—	—	—	—	T1GSS	C2SYNC
bit 7							bit 0

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

bit 1 **T1GSS:** Timer1 Gate Source Select bit
1 = Timer1 gate source is $\overline{T1G}$ pin (RA4 must be configured as digital input)
0 = Timer1 gate source is Comparator 2 Output

bit 0 **C2SYNC:** Comparator 2 Synchronize bit
1 = C2 output synchronized with falling edge of Timer1 clock
0 = C2 output not synchronized with Timer1 clock

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

7.4 Comparator Outputs

The comparator outputs are read through the CMCON0 register. These bits are read-only. The comparator outputs may also be directly output to the RA2 and RC4 I/O pins. When enabled, multiplexors in the output path of the RA2 and RC4 pins will switch and the output of each pin will be the unsynchronized output of the comparator. The uncertainty of each of the comparators is related to the input offset voltage and the response time given in the specifications. Figure 7-4 and Figure 7-5 show the output block diagram for Comparator 1 and 2.

The TRIS bits will still function as an output enable/ disable for the RA2 and RC4 pins while in this mode.

The polarity of the comparator outputs can be changed using the C1INV and C2INV bits (CMCON0<5:4>).

Timer1 gate source can be configured to use the $\overline{T1G}$ pin or Comparator 2 output as selected by the T1GSS bit (CMCON1<1>). This feature can be used to time the duration or interval of analog events. The output of Comparator 2 can also be synchronized with Timer1 by setting the C2SYNC bit (CMCON1<0>). When enabled, the output of Comparator 2 is latched on the falling edge of Timer1 clock source. If a prescaler is used with Timer1, Comparator 2 is latched after the prescaler. To prevent a race condition, the Comparator 2 output is latched on the falling edge of the Timer1 clock source and Timer1 increments on the rising edge of its clock source. See (Figure 7-5), Comparator 2 Block Diagram and (Figure 6-1), Timer1 Block Diagram for more information.

It is recommended to synchronize Comparator 2 with Timer1 by setting the C2SYNC bit when Comparator 2 is used as the Timer1 gate source. This ensures Timer1 does not miss an increment if Comparator 2 changes during an increment.

7.5 Comparator Interrupts

The comparator interrupt flags are set whenever there is a change in the output value of its respective comparator. Software will need to maintain information about the status of the output bits, as read from CMCON0<7:6>, to determine the actual change that has occurred. The CxIF bits, PIR1<4:3>, are the Comparator Interrupt flags. 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 bits (PIE1<4:3>) and the PEIE bit (INTCON<6>) must be set to enable the interrupts. In addition, the GIE bit must also be set. If any of these bits are cleared, the interrupt is not enabled, though the CxIF bits 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.
b) Clear flag bit CxIF

A mismatch condition will continue to set flag bit CxIF. Reading CMCON0 will end the mismatch condition and allow flag bits CxIF to be cleared.

Note:	If a change in the CMCON0 register (CxOUT) should occur when a read operation is being executed (start of the Q2 cycle), then the CxIF (PIR1<3>) interrupt flag may not get set.

7.6 Comparator Reference

The comparator module also allows the selection of an internally generated voltage reference for one of the comparator inputs. The VRCON register, Register 7-3, controls the voltage reference module shown in Figure 7-6.

7.6.1 CONFIGURING THE VOLTAGE REFERENCE

The voltage reference can output 32 distinct voltage levels, 16 in a high range and 16 in a low range.

The following equation determines the output voltages:

EQUATION 7-1:

VRR = 1 *(low range): CVREF = (VR3:VR0/24) x VDD*

VRR = 0 *(high range):*

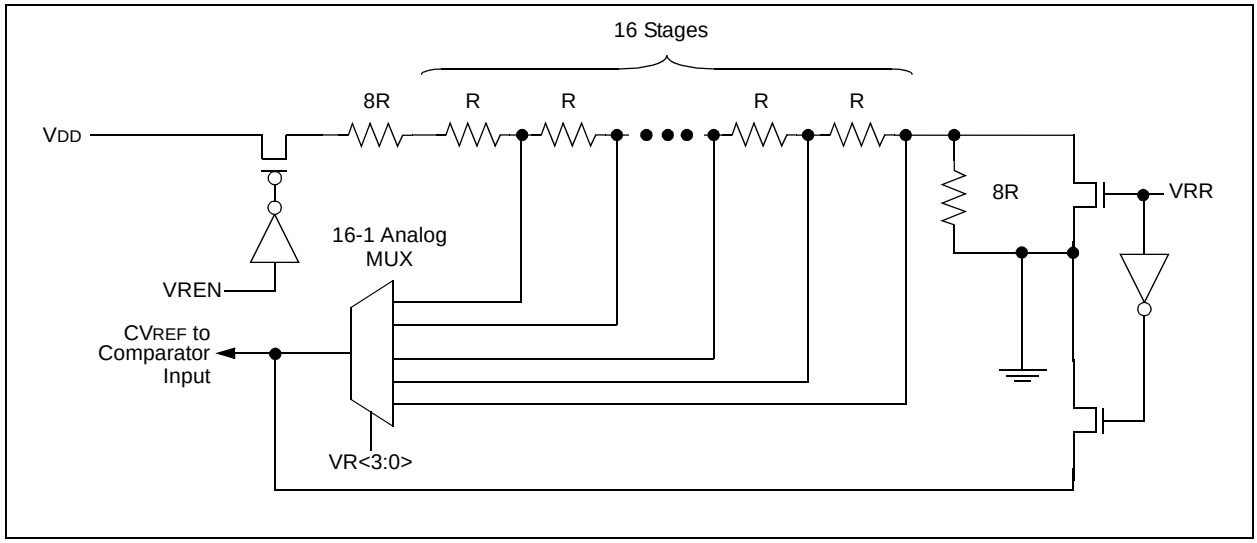
CVREF = (VDD/4) + (VR3:VR0 x VDD/32)

7.6.2 VOLTAGE REFERENCE ACCURACY/ERROR

The full range of VSS to VDD cannot be realized due to the construction of the module. The transistors on the top and bottom of the resistor ladder network (Figure 7-6) keep CVREF from approaching VSS or VDD. The exception is when the module is disabled by clearing the VREN bit (VRCON<7>). When disabled, the reference voltage is VSS when VR<3:0> = 0000. This allows the comparators to detect a zero-crossing and not consume CVREF module current.

The voltage reference is VDD derived and therefore, the CVREF output changes with fluctuations in VDD. The tested absolute accuracy of the comparator voltage reference can be found in **Section 14.0 "Electrical Specifications"**.

FIGURE 7-6: COMPARATOR VOLTAGE REFERENCE BLOCK DIAGRAM



7.7 Comparator Response Time

Response time is the minimum time, after selecting a new reference voltage or input source, before the comparator output is ensured to have a valid level. If the internal reference is changed, the maximum delay of the internal voltage reference must be considered when using the comparator outputs. Otherwise, the maximum delay of the comparators should be used (Table 14-9).

7.8 Operation During Sleep

The comparators and voltage reference, if enabled before entering Sleep mode, remain active during Sleep. This results in higher Sleep currents than shown in the power-down specifications. The additional current consumed by the comparator and the voltage reference is shown separately in the specifications. To minimize power consumption while in Sleep mode, turn off the comparator, CM<2:0> = 111, and voltage reference, VRCON<7> = 0.

While the comparator is enabled during Sleep, an interrupt will wake-up the device. If the GIE bit (INTCON<7>) is set, the device will jump to the interrupt vector (0004h), and if clear, continues execution with the next instruction. If the device wakes up from Sleep, the contents of the CMCON0, CMCON1 and VRCON registers are not affected.

7.9 Effects of a Reset

A device Reset forces the CMCON0, CMCON1 and VRCON registers to their Reset states. This forces the comparator module to be in the Comparator Reset mode, CM<2:0> = 000 and the voltage reference to its off state. Thus, all potential inputs are analog inputs with the comparator and voltage reference disabled to consume the smallest current possible.

REGISTER 7-3: VRCON – VOLTAGE REFERENCE CONTROL REGISTER (ADDRESS: 99h)

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

bit 7 **VREN:** CVREF Enable bit
1 = CVREF circuit powered on
0 = CVREF circuit powered down, no IDD drain and CVREF = VSS.

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 $0 \leq$ VR<3:0> ≤ 15
When VRR = 1: CVREF = (VR<3:0>/24) * VDD
When VRR = 0: CVREF = VDD/4 + (VR<3:0>/32) * VDD

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

TABLE 7-2: REGISTERS ASSOCIATED WITH COMPARATOR MODULE

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOD	Value on all other Resets
0Bh/8Bh	INTCON	GIE	PEIE	T0IE	INTE	RAIE	T0IF	INTF	RAIF	0000 0000	0000 0000
0Ch	PIR1	EEIF	ADIF	RCIF	C2IF	C1IF	OSFIF	TXIF	TMR1IF	0000 0000	0000 0000
19h	CMCON0	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	0000 0000	0000 0000
1Ah	CMCON1	—	—	—	—	—	—	T1GSS	C2SYNC	---- --10	---- --10
85h/185h	TRISA	—	—	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	--11 1111	--11 1111
87h/187h	TRISC	—	—	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	--11 1111	--11 1111
8Ch	PIE1	EEIE	ADIE	RCIE	C2IE	C1IE	OSFIE	TXIE	TMR1IE	0000 0000	0000 0000
99h	VRCON	VREN	—	VRR	—	VR3	VR2	VR1	VR0	0-0- 0000	0-0- 0000

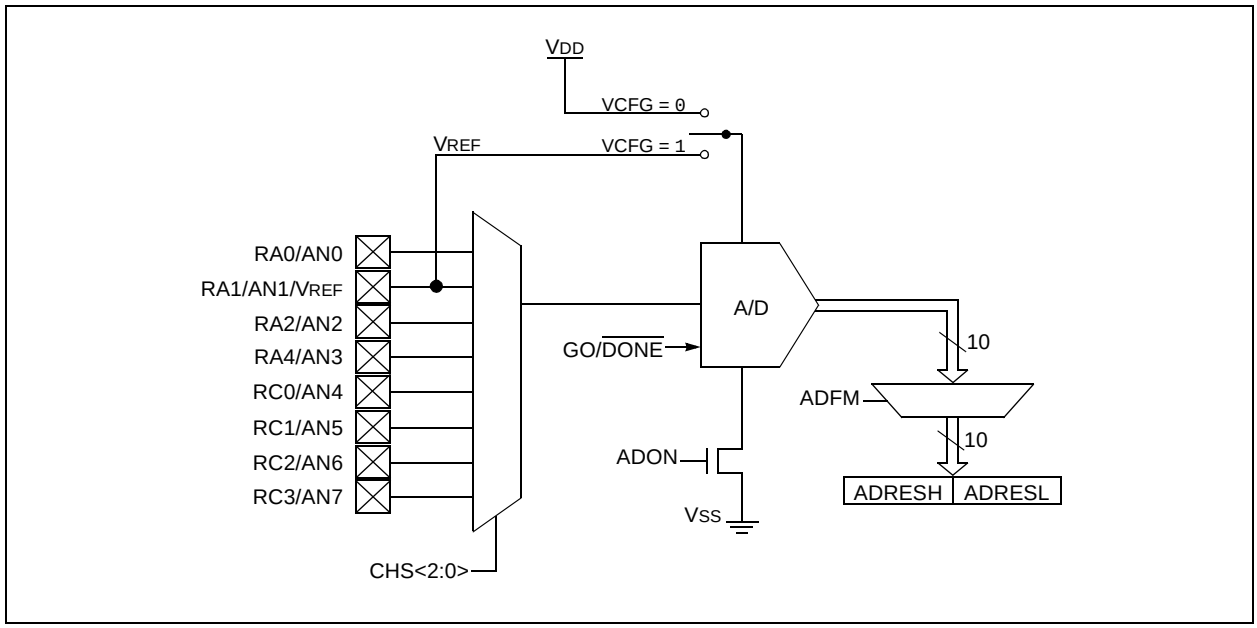
Legend: x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used by the comparator or Comparator Voltage Reference module.

NOTES:

8.0 ANALOG-TO-DIGITAL CONVERTER (A/D) MODULE

The analog-to-digital converter (A/D) allows conversion of an analog input signal to a 10-bit binary representation of that signal. The PIC16F688 has eight analog inputs, multiplexed into one sample and hold circuit. The output of the sample and hold is connected to the input of the converter. The converter generates a binary result via successive approximation and stores the result in a 10-bit register. The voltage reference used in the conversion is software selectable to either VDD or a voltage applied by the VREF pin. Figure 8-1 shows the block diagram of the A/D on the PIC16F688.

FIGURE 8-1: A/D BLOCK DIAGRAM



8.1 A/D Configuration and Operation

There are three registers available to control the functionality of the A/D module:

1. ANSEL (Register 8-1)
2. ADCON0 (Register 8-2)
3. ADCON1 (Register 8-3)

8.1.1 ANALOG PORT PINS

The ANS<7:0> bits (ANSEL<7:0>) and the TRIS bits control the operation of the A/D port pins. Set the corresponding TRIS bits to set the pin output driver to its high-impedance state. Likewise, set the corresponding ANSEL bit to disable the digital input buffer.

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

There are eight analog channels on the PIC16F688, AN0 through AN7. The CHS<2:0> bits (ADCON0<4:2>) control which channel is connected to the sample and hold circuit.

8.1.3 VOLTAGE REFERENCE

There are two options for the voltage reference to the A/D converter: either VDD is used, or an analog voltage applied to VREF is used. The VCFG bit (ADCON0<6>) controls the voltage reference selection. If VCFG is set, then the voltage on the VREF pin is the reference; otherwise, VDD is the reference.

8.1.4 CONVERSION CLOCK

The A/D conversion cycle requires 11 TAD. The source of the conversion clock is software selectable via the ADCS bits (ADCON1<6:4>). There are seven possible clock options:

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

For correct conversion, the A/D conversion clock (1/TAD) must be selected to ensure a minimum TAD of 1.6 µs. Table 8-1 shows a few TAD calculations for selected frequencies.

TABLE 8-1: TAD VS. DEVICE OPERATING FREQUENCIES

A/D Clock Source (TAD)		Device Frequency			
Operation	**ADCS<2:0>**	**20 MHz**	**5 MHz**	**4 MHz**	**1.25 MHz**
2 TOSC	000	100 ns(2)	400 ns(2)	500 ns(2)	1.6 µs
4 TOSC	100	200 ns(2)	800 ns(2)	1.0 µs(2)	3.2 µs
8 TOSC	001	400 ns(2)	1.6 µs	2.0 µs	6.4 µs
16 TOSC	101	800 ns(2)	3.2 µs	4.0 µs	12.8 µs(3)
32 TOSC	010	1.6 µs	6.4 µs	8.0 µs(3)	25.6 µs(3)
64 TOSC	110	3.2 µs	12.8 µs(3)	16.0 µs(3)	51.2 µs(3)
A/D RC	x11	2-6 µs(1,4)	2-6 µs(1,4)	2-6 µs(1,4)	2-6 µs(1,4)

Legend: Shaded cells are outside of recommended range.

Note 1: The A/D RC 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 A/D RC clock source is only recommended if the conversion will be performed during Sleep.

8.1.5 STARTING A CONVERSION

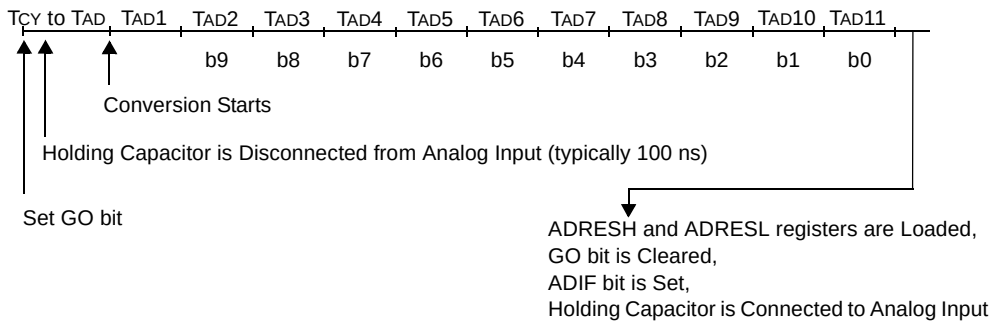
The A/D conversion is initiated by setting the GO/$\overline{\text{DONE}}$ bit (ADCON0<1>). When the conversion is complete, the A/D module:

- Clears the GO/$\overline{\text{DONE}}$ bit
- Sets the ADIF flag (PIR1<6>)
- Generates an interrupt (if enabled)

If the conversion must be aborted, the GO/$\overline{\text{DONE}}$ bit can be cleared in software. The ADRESH:ADRESL registers will not be updated with the partially complete A/D conversion sample. Instead, the ADRESH:ADRESL registers will retain the value of the previous conversion. After an aborted conversion, a 2 TAD delay is required before another acquisition can be initiated. Following the delay, an input acquisition is automatically started on the selected channel.

Note:	The GO/$\overline{\text{DONE}}$ bit should not be set in the same instruction that turns on the A/D.

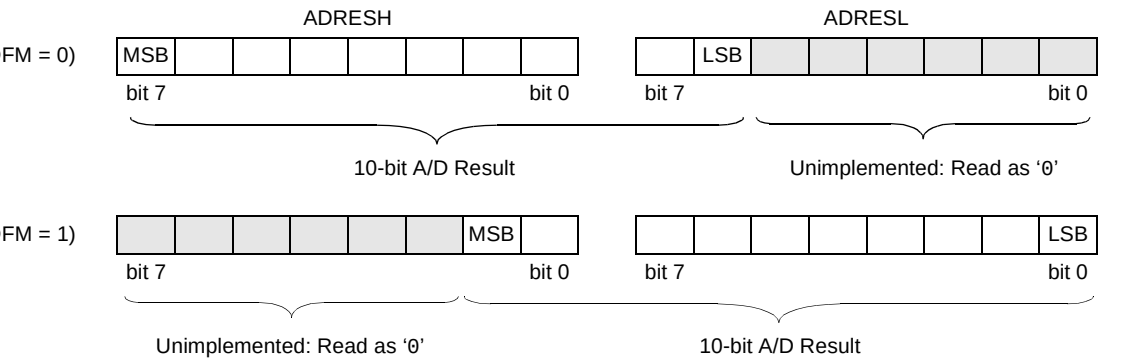
FIGURE 8-2: A/D CONVERSION TAD CYCLES



8.1.6 CONVERSION OUTPUT

The A/D conversion can be supplied in two formats: left or right shifted. The ADFM bit (ADCON0<7>) controls the output format. Figure 8-3 shows the output formats.

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



REGISTER 8-1: ANSEL – ANALOG SELECT REGISTER (ADDRESS: 91h)

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

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.**(1)**

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.

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

REGISTER 8-2: ADCON0 – A/D CONTROL REGISTER (ADDRESS: 1Fh)

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/$\overline{\text{DONE}}$	ADON
bit 7							bit 0

bit 7 **ADFM:** A/D Result Formed Select bit
1 = Right justified
0 = Left justified

bit 6 **VCFG:** Voltage Reference bit
1 = V_{REF} pin
0 = V_{DD}

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

bit 4-2 **CHS<2:0>:** Analog Channel Select bits
000 = Channel 00 (AN0)
001 = Channel 01 (AN1)
010 = Channel 02 (AN2)
011 = Channel 03 (AN3)
100 = Channel 04 (AN4)
101 = Channel 05 (AN5)
110 = Channel 06 (AN6)
111 = Channel 07 (AN7)

bit 1 **GO/$\overline{\text{DONE}}$:** A/D Conversion Status bit
1 = A/D conversion cycle in progress. Setting this bit starts an A/D conversion cycle.
This bit is automatically cleared by hardware when the A/D conversion has completed.
0 = A/D conversion completed/not in progress

bit 0 **ADON:** A/D Conversion Status bit
1 = A/D converter module is operating
0 = A/D converter is shut off and consumes no operating current

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

REGISTER 8-3: ADCON1 – A/D CONTROL REGISTER 1 (ADDRESS: 9Fh)

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

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

bit 6-4 **ADCS<2:0>:** A/D Conversion Clock Select bits
000 = $F_{OSC}/2$
001 = $F_{OSC}/8$
010 = $F_{OSC}/32$
x11 = F_{RC} (clock derived from a dedicated internal oscillator = 500 kHz max)
100 = $F_{OSC}/4$
101 = $F_{OSC}/16$
110 = $F_{OSC}/64$

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

8.1.7 CONFIGURING THE A/D

After the A/D module has been configured as desired, the selected channel must be acquired before the conversion is started. The analog input channels must have their corresponding TRIS bits selected as inputs.

To determine sample time, see **Section 14.0 "Electrical Specifications"**. After this sample time has elapsed, the A/D conversion can be started.

These steps should be followed for an A/D conversion:

1. Configure the A/D module:
   - Configure analog/digital I/O (ANSEL)
   - Configure voltage reference (ADCON0)
   - Select A/D input channel (ADCON0)
   - Select A/D conversion clock (ADCON1)
   - Turn on A/D module (ADCON0)
2. Configure A/D interrupt (if desired):
   - Clear ADIF bit (PIR1<6>)
   - Set ADIE bit (PIE1<6>)
   - Set PEIE and GIE bits (INTCON<7:6>)
3. Wait the required acquisition time.
4. Start conversion:
   - Set GO/$\overline{\text{DONE}}$ bit (ADCON0<0>)
5. Wait for A/D conversion to complete, by either:
   - Polling for the GO/$\overline{\text{DONE}}$ bit to be cleared (with interrupts disabled); OR
   - Waiting for the A/D interrupt
6. Read A/D Result register pair (ADRESH:ADRESL), clear bit ADIF if required.
7. For next conversion, go to step 1 or step 2 as required. The A/D conversion time per bit is defined as T_{AD}. A minimum wait of 2 T_{AD} is required before the next acquisition starts.

EXAMPLE 8-1: A/D CONVERSION

```
;This code block configures the A/D
;for polling, Vdd reference, R/C clock
;and RA0 input.
;
;Conversion start & wait for complete
;polling code included.
;
 BSF    STATUS,RP0     ;Bank 1
 MOVLW  B'01110000'    ;A/D RC clock
 MOVWF  ADCON1
 BSF    TRISA,0        ;Set RA0 to input
 BSF    ANSEL,0        ;Set RA0 to analog
 BCF    STATUS,RP0     ;Bank 0
 MOVLW  B'10000001'    ;Right, Vdd Vref, AN0
 MOVWF  ADCON0
 CALL   SampleTime     ;Wait min sample time
 BSF    ADCON0,GO      ;Start conversion
 BTFSC  ADCON0,GO      ;Is conversion done?
 GOTO   $-1            ;No, test again
 MOVF   ADRESH,W       ;Read upper 2 bits
 MOVWF  RESULTHI
 BSF    STATUS,RP0     ;Bank 1
 MOVF   ADRESL,W       ;Read lower 8 bits
 MOVWF  RESULTLO
```

8.2 A/D Acquisition Requirements

For the A/D converter 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 impedance is decreased, the acquisition time may be decreased. After the analog input channel is selected (changed), this 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 A/D). The 1/2 LSb error is the maximum error allowed for the A/D to meet its specified resolution.

To calculate the minimum acquisition time, TACQ, see the "*PICmicro® Mid-Range MCU Family Reference Manual*" (DS33023).

EQUATION 8-1: ACQUISITION TIME

TACQ = Amplifier Settling Time + Hold Capacitor Charging Time + Temperature Coefficient

= TAMP + TC + TCOFF

= 2 µs + TC + [(Temperature -25°C)(0.05 µs/°C)]

TC = CHOLD (RIC + RSS + RS) In(1/2047)

= -120 pF(1 kΩ + 7 kΩ + 10 kΩ) In(0.0004885)

= 16.47 µs

TACQ = 2 µs + 16.47 µs + [(50°C -25°C)(0.05 µs/°C)]

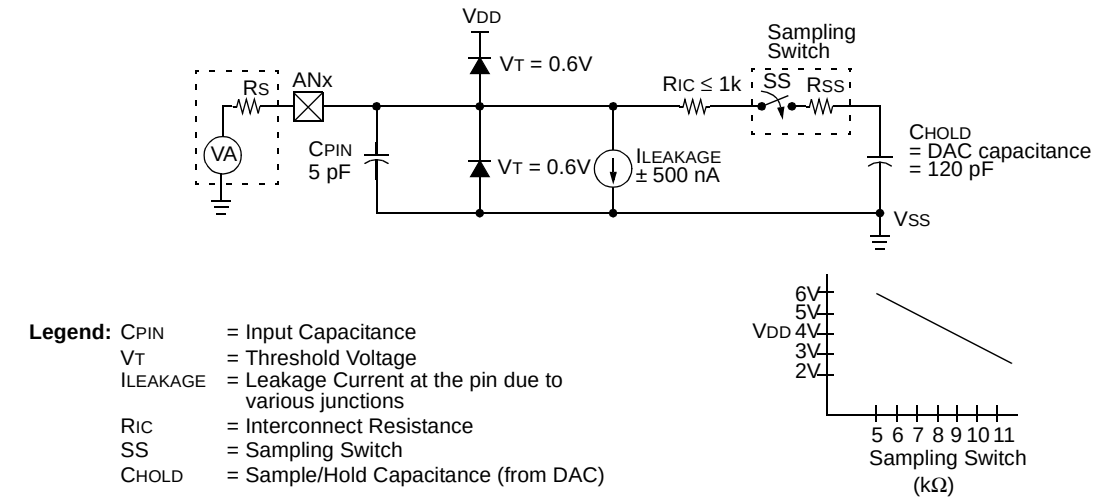
= 19.72 µs

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

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

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

FIGURE 8-4: ANALOG INPUT MODEL



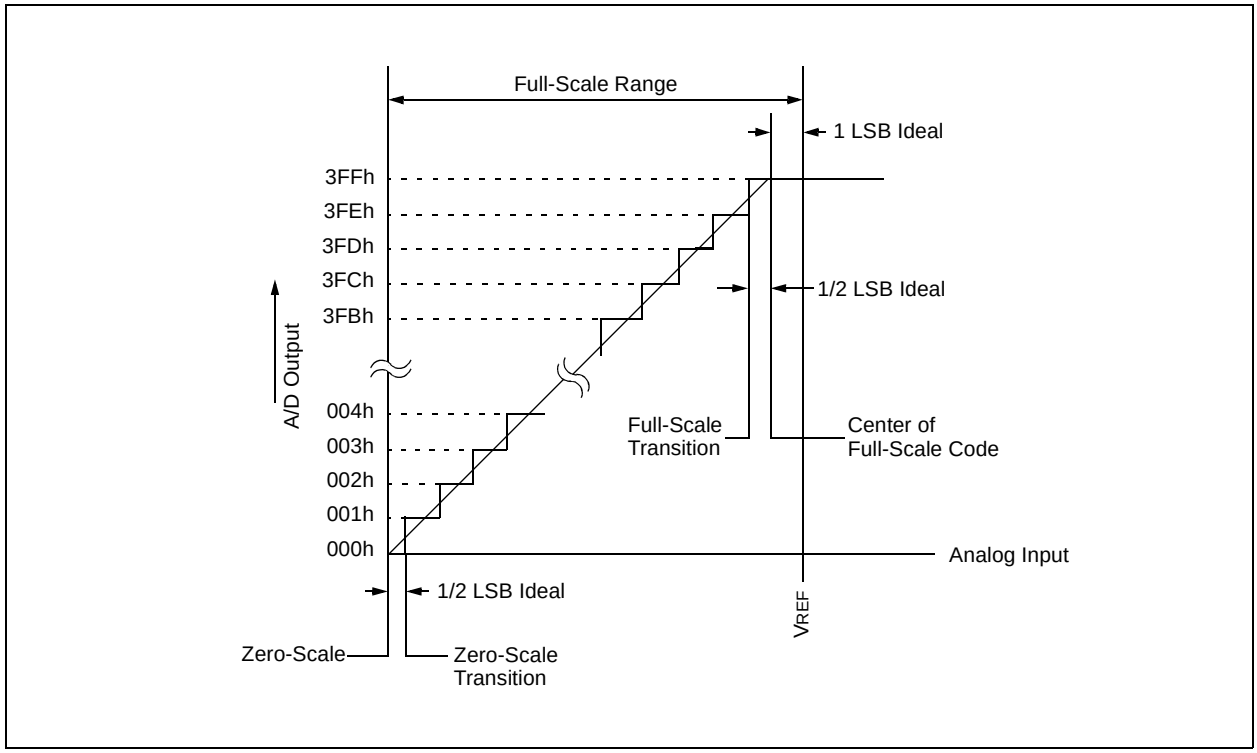
Legend: CPIN = Input Capacitance
VT = Threshold Voltage
ILEAKAGE = Leakage Current at the pin due to various junctions
RIC = Interconnect Resistance
SS = Sampling Switch
CHOLD = Sample/Hold Capacitance (from DAC)

8.3 A/D Operation During Sleep

The A/D converter module can operate during Sleep. This requires the A/D clock source to be set to the internal oscillator. When the RC clock source is selected, the A/D waits one instruction before starting the conversion. This allows the SLEEP instruction to be executed, thus eliminating much of the switching noise from the conversion. When the conversion is complete, the GO/DONE bit is cleared and the result is loaded into the ADRESH:ADRESL registers. If the A/D interrupt is enabled, the device awakens from Sleep. If the GIE bit (INTCON<7>) is set, the program counter is set to the interrupt vector (0004h), if GIE is clear, the next instruction is executed. If the A/D interrupt is not enabled, the A/D module is turned off, although the ADON bit remains set.

When the A/D clock source is something other than RC, a SLEEP instruction causes the present conversion to be aborted, and the A/D module is turned off. The ADON bit remains set.

FIGURE 8-5: A/D TRANSFER FUNCTION



8.4 Effects of Reset

A device Reset forces all registers to their Reset state. Thus, the A/D module is turned off and any pending conversion is aborted. The ADRESH:ADRESL registers are unchanged.

TABLE 8-2: SUMMARY OF A/D REGISTERS

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOD	Value on all other Resets
05h/ 105h	PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	--xx xxxx	--uu uuuu
07h/ 107h	PORTC	—	—	RC5	RC4	RC3	RC2	RC1	RC0	--xx xxxx	--uu uuuu
0Bh/ 8Bh	INTCON	GIE	PEIE	T0IE	INTE	RAIE	T0IF	INTF	RAIF	0000 0000	0000 0000
0Ch	PIR1	EEIF	ADIF	RCIF	C2IF	C1IF	OSFIF	TXIF	TMR1IF	0000 0000	0000 0000
1Eh	ADRESH	Most Significant 8 bits of the left shifted A/D result or 2 bits of the right shifted result								xxxx xxxx	uuuu uuuu
1Fh	ADCON0	ADFM	VCFG	—	CHS2	CHS1	CHS0	GO/$\overline{\text{DONE}}$	ADON	00-0 0000	00-0 0000
85h/ 185h	TRISA	—	—	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	--11 1111	--11 1111
87h/ 187h	TRISC	—	—	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	--11 1111	--11 1111
8Ch	PIE1	EEIE	ADIE	RCIE	C2IE	C1IE	OSFIE	TXIE	TMR1IE	0000 0000	0000 0000
91h	ANSEL	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0	1111 1111	1111 1111
9Eh	ADRESL	Least Significant 2 bits of the left shifted A/D result or 8 bits of the right shifted result								xxxx xxxx	uuuu uuuu
9Fh	ADCON1	—	ADCS2	ADCS1	ADCS0	—	—	—	—	-000 ----	-000 ----

Legend: x = unknown, u = unchanged, - = unimplemented read as '0'. Shaded cells are not used for A/D module.

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 VDD 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.

Additional information on the data EEPROM is available in the "*PICmicro® Mid-Range MCU Family Reference Manual*" (DS33023).

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 $\overline{\text{MCLR}}$ 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 (PIR1<7>), 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.

REGISTER 9-1: EEDAT – EEPROM DATA REGISTER (ADDRESS: 9Ah)

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

bit 7-0 **EEDATn**: Byte Value to Write to or Read From Data EEPROM bits

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

REGISTER 9-2: EEADR – EEPROM ADDRESS REGISTER (ADDRESS: 9Bh)

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

bit 7-0 **EEADR**: Specifies One of 256 Locations for EEPROM Read/Write Operation bits

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

REGISTER 9-3: EECON1 – EEPROM CONTROL REGISTER 1 (ADDRESS: 9Ch)

R/W-0	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

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 BOD detect)
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 an memory read (Read takes one cycle. RD is cleared in hardware. The RD bit can only be set, not cleared, in software.)
0 = Does not initiate an memory read

Legend:			
S = Bit can only be set			
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

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 (EECON1<7>), and then set control bit RD (EECON1<0>). 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

```
BSF     STATUS, RP0   ;
BCF     STATUS, RP1   ; Bank 1
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

```
        BSF     STATUS, RP0   ;
        BCF     STATUS, RP1   ; Bank 1
        MOVLW   DATA_EE_ADDR  ;
        MOVWF   EEADR         ; Data Memory Address to write
        MOVLW   DATA_EE_DATA  ;
        MOVWF   EEDAT        ; Data Memory Value to write
        BCF     EECON1, EEPGD ; Point to DATA memory
        BSF     EECON1, WREN  ; Enable writes

Required BCF    INTCON, GIE   ; Disable INTs.
Sequence 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
```

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 (EECON1<7>), and then set control bit RD (EECON1<0>). 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

```
    BSF     STATUS, RP0      ;
    BCF     STATUS, RP1      ; Bank 1
    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
    BSF     EECON1, EEPGD    ; Point to PROGRAM memory
    BSF     EECON1, RD       ; EE Read
;
                             ; First instruction after BSF EECON1,RD executes normally
    NOP
    NOP                      ; Any instructions here are ignored as program
                             ; memory is read in second cycle after BSF EECON1,RD
;
    MOVF    EEDAT, W         ; W = LS Byte of Program EEDAT
    MOVF    EEDATH, W        ; W = MS Byte of Program EEDAT
```


FIGURE 9-1: FLASH PROGRAM MEMORY READ CYCLE EXECUTION

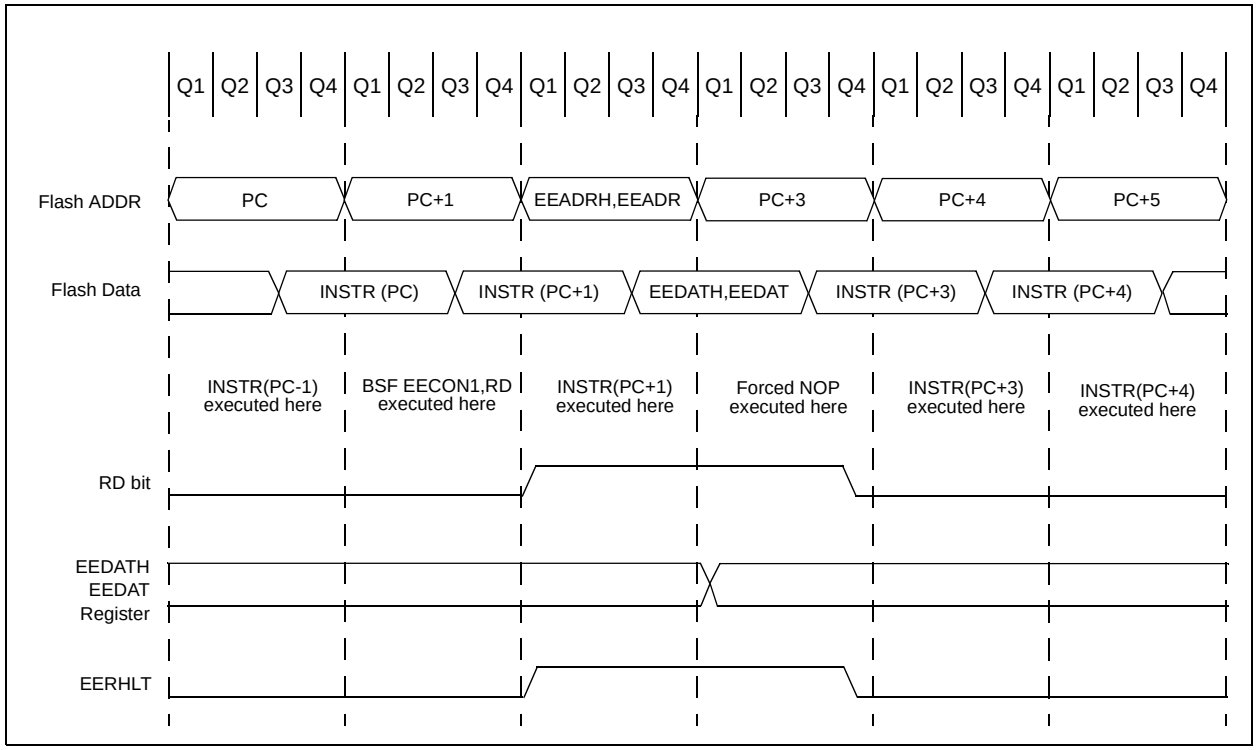


TABLE 9-1: REGISTERS/BITS ASSOCIATED WITH DATA EEPROM

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOD	Value on all other Resets
0Bh/8Bh	INTCON	GIE	PEIE	T0IE	INTE	RAIE	T0IF	INTF	RAIF	0000 0000	0000 0000
0Ch	PIR1	EEIF	ADIF	RCIF	C2IF	C1IF	OSFIF	TXIF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	EEIE	ADIE	RCIE	C2IE	C1IE	OSFIE	TXIE	TMR1IE	0000 0000	0000 0000
97h	EEDATH	EEPROM Data register, high byte								--00 0000	0000 0000
98h	EEADRH	EEPROM Address register, high byte								---- 0000	0000 0000
9Ah	EEDAT	EEDAT7	EEDAT6	EEDAT5	EEDAT4	EEDAT3	EEDAT2	EEDAT1	EEDAT0	0000 0000	0000 0000
9Bh	EEADR	EEADR7	EEADR6	EEADR5	EEADR4	EEADR3	EEADR2	EEADR1	EEADR0	0000 0000	0000 0000
9Ch	EECON1	EEPGD	—	—	—	WRERR	WREN	WR	RD	---- x000	---- q000
9Dh	EECON2(1)	—	—	—	—	—	—	—	—	—	—

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

Note 1: EECON2 is not a physical register.

NOTES:

10.0 ENHANCED UNIVERSAL ASYNCHRONOUS RECEIVER TRANSMITTER (EUSART)

The Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART) module is the serial I/O module available for PIC16F688 . (EUSART is also known as a Serial Communications Interface or SCI). The EUSART can be configured as a full-duplex asynchronous system that can communicate with peripheral devices, such as CRT terminals and personal computers. It can also be configured as a half-duplex synchronous system that can communicate with peripheral devices, such as A/D or D/A integrated circuits, serial EEPROMs, etc.

The EUSART module implements additional features, including automatic baud rate detection and calibration, automatic wake-up on Break reception and 13-bit Break character transmit. These make it ideally suited for use in Local Interconnect Network (LIN) bus systems.

The USART can be configured in the following modes:

- Asynchronous (full-duplex) with:
  - Auto-wake-up on Break
  - Auto baud calibration
  - 13-bit Break character transmission
- Synchronous – Master (half-duplex) with selectable clock polarity
- Synchronous – Slave (half-duplex) with selectable clock polarity

In order to configure pins RC4/C2OUT/TX/CK and RC5/RX/DT as the Universal Synchronous Asynchronous Receiver Transmitter:

- SPEN (RCSTA<7>) bit must be set (= 1),
- TRISC<5> bit must be set (= 1), and
- TRISC<4> bit must be set (= 1).

Note:	The USART control will automatically reconfigure the pin from input to output as needed.

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 are detailed in on the following pages in Register 10-1, Register 10-2 and Register 10-3, respectively.

10.1 Clock Accuracy With Asynchronous Operation

The factory calibrates the internal oscillator block output (INTOSC) for 8 MHz. However, this frequency may drift as V_{DD} 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 back to 8 MHz. Adjusting the value in the OSCTUNE register allows for fine resolution changes to the system clock source (see **Section 3.4 “Internal Clock Modes”** for more information).

The other method adjusts the value in the baud rate generator. 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 (ADDRESS: 16h)

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	SYNC	SENDB	BRGH	TRMT	TX9D
bit 7							bit 0

bit 7 **CSRC:** Clock Source Select bit

Asynchronous mode:

Don't care

Synchronous mode:

1 = Master mode (clock generated internally from BRG)
0 = Slave mode (clock from external source)

bit 6 **TX9:** 9-bit Transmit Enable bit

1 = Selects 9-bit transmission
0 = Selects 8-bit transmission

bit 5 **TXEN:** Transmit Enable bit

1 = Transmit enabled
0 = Transmit disabled

Note: SREN/CREN overrides TXEN in Sync mode.

bit 4 **SYNC:** USART 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:** 9th bit of Transmit Data

Can be address/data bit or a parity bit.

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

REGISTER 10-2: RCSTA – RECEIVE STATUS AND CONTROL REGISTER (ADDRESS: 17h)

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

bit 7 **SPEN:** Serial Port Enable bit

1 = Serial port enabled (configures RX/DT and TX/CK pins as serial port pins)
0 = Serial port disabled (held in Reset)

bit 6 **RX9:** 9-bit Receive Enable bit

1 = Selects 9-bit reception
0 = Selects 8-bit reception

bit 5 **SREN:** Single Receive Enable bit

Asynchronous mode:

Don't care
Synchronous mode - Master:

1 = Enables single receive
0 = Disables single receive
This bit is cleared after reception is complete.
Synchronous mode - Slave:

Don't care

bit 4 **CREN:** Continuous Receive Enable bit

Asynchronous mode:

1 = Enables receiver
0 = Disables receiver
Synchronous mode:

1 = Enables continuous receive until enable bit CREN is cleared (CREN overrides SREN)
0 = Disables continuous receive

bit 3 **ADDEN:** Address Detect Enable bit

Asynchronous mode 9-bit (RX9 = 1):

1 = Enables address detection, enable interrupt and load the receive buffer when RSR<8> is set
0 = Disables address detection, all bytes are received and ninth bit can be used as parity bit
Asynchronous mode 8-bit (RX9 = 0):

Don't care

bit 2 **FERR:** Framing Error bit

1 = Framing error (can be updated by reading RCREG register and receive next valid byte)
0 = No framing error

bit 1 **OERR:** Overrun Error bit

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

bit 0 **RX9D:** 9th bit of Received Data

This can be address/data bit or a parity bit and must be calculated by user firmware.

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

REGISTER 10-3: BAUDCTL – BAUD RATE CONTROL REGISTER (ADDRESS: 11h)

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

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 pin $\overline{TX}$
0 = Transmit pin TX
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

1 = Next falling RX/DT edge will generate interrupt (automatically cleared on next rising edge after falling edge)
0 = RX/DT edges do not generate interrupts

bit 0 **ABDEN**: Auto Baud Detect Enable bit

Asynchronous mode:
1 = Auto Baud mode is enabled (clears when auto baud is complete)
0 = Auto Baud mode is disabled
Synchronous mode:
Don't care

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

10.2 USART Baud Rate Generator (BRG)

The BRG is a dedicated 8-bit or 16-bit generator, that supports both the Asynchronous and Synchronous modes of the USART. By default, the BRG operates in 8-bit mode; setting the BRG16 bit (BAUDCTL<3>) selects 16-bit mode.

The SPBRGH:SPBRG register pair controls the period of a free running timer. In Asynchronous mode, bits BRGH (TXSTA<2>) and BRG16 also control the baud rate. In Synchronous mode, bit BRGH is ignored. Table 10-1 shows the formula for computation of the baud rate for different USART modes, which only apply in Master mode (internally generated clock).

Given the desired baud rate and FOSC, the nearest integer value for the SPBRGH:SPBRG registers can be calculated using the formulas in Table 10-1. From this, the error in baud rate can be determined. An example calculation is shown in Example 10-1. Typical baud rates and error values for the various asynchronous modes are shown in Table 10-2. It may be advantageous to use the high baud rate (BRGH = 1), or the 16-bit BRG to reduce the baud rate error, or achieve a slow baud rate for a fast oscillator frequency.

Writing a new value to the SPBRGH:SPBRG registers causes the BRG timer to be reset (or cleared). This ensures 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 and make sure that the receive operation is IDLE before changing the system clock.

10.2.1 SAMPLING

The data on the RC5/RX/DT pin is sampled three times by a majority detect circuit to determine if a high or a low level is present at the RX pin.

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:

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

Solving for SPBRGH:SPBRG:

$$X = \frac{\frac{FOSC}{Desired\ Baud\ Rate}}{64} - 1$$

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

$$= [25.042] = 25$$

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

$$= 9615$$

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

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

Note: When BRGH = 1 and BRG16 = 1 then SPBRGH:SPBRG values ≤ 4 are invalid.

TABLE 10-1: BAUD RATE FORMULAS

Configuration Bits			BRG/USART 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 10-2: REGISTERS ASSOCIATED WITH BAUD RATE GENERATOR

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOD	Value on all other Resets
11h	BAUDCTL	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	-1-1 0-00	-1-1 0-00
12h	SPBRGH	Baud Rate Generator register, high byte								0000 0000	0000 0000
13h	SPBRG	Baud Rate Generator register, low byte								0000 0000	0000 0000
16h	TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 -010	0000 -010
17h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x

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

TABLE 10-3: BAUD RATES FOR ASYNCHRONOUS MODES

BAUD RATE (K)	SYNC = 0, BRGH = 0, BRG16 = 0								
	FOSC = 20.000 MHz			FOSC = 10.000 MHz			FOSC = 8.000 MHz		
	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)
0.3	—	—	—	—	—	—	—	—	—
1.2	1.221	1.73	255	1.202	0.16	129	1201	-0.16	103
2.4	2.404	0.16	129	2.404	0.16	64	2403	-0.16	51
9.6	9.766	1.73	31	9.766	1.73	15	9615	-0.16	12
19.2	19.531	1.73	15	19.531	1.73	7	—	—	—
57.6	62.500	8.51	4	52.083	-9.58	2	—	—	—
115.2	104.167	-9.58	2	78.125	-32.18	1	—	—	—

BAUD RATE (K)	SYNC = 0, BRGH = 0, BRG16 = 0								
	FOSC = 4.000 MHz			FOSC = 2.000 MHz			FOSC = 1.000 MHz		
	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)
0.3	0.300	0.16	207	300	-0.16	103	300	-0.16	51
1.2	1.202	0.16	51	1201	-0.16	25	1201	-0.16	12
2.4	2.404	0.16	25	2403	-0.16	12	—	—	—
9.6	8.929	-6.99	6	—	—	—	—	—	—
19.2	20.833	8.51	2	—	—	—	—	—	—
57.6	62.500	8.51	0	—	—	—	—	—	—
115.2	62.500	-45.75	0	—	—	—	—	—	—

BAUD RATE (K)	SYNC = 0, BRGH = 1, BRG16 = 0								
	FOSC = 20.000 MHz			FOSC = 10.000 MHz			FOSC = 8.000 MHz		
	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)
2.4	—	—	—	2.441	1.73	255	2403	-0.16	207
9.6	9.615	0.16	129	9.615	0.16	64	9615	-0.16	51
19.2	19.231	0.16	64	19.531	1.73	31	19230	-0.16	25
57.6	56.818	-1.36	21	56.818	-1.36	10	55555	3.55	8
115.2	113.636	-1.36	10	125.000	8.51	4	—	—	—

TABLE 10-3: BAUD RATES FOR ASYNCHRONOUS MODES (CONTINUED)

BAUD RATE (K)	SYNC = 0, BRGH = 1, BRG16 = 0								
	FOSC = 4.000 MHz			FOSC = 2.000 MHz			FOSC = 1.000 MHz		
	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)
0.3	—	—	—	—	—	—	300	-0.16	207
1.2	1.202	0.16	207	1201	-0.16	103	1201	-0.16	51
2.4	2.404	0.16	103	2403	-0.16	51	2403	-0.16	25
9.6	9.615	0.16	25	9615	-0.16	12	—	—	—
19.2	19.231	0.16	12	—	—	—	—	—	—
57.6	62.500	8.51	3	—	—	—	—	—	—
115.2	125.000	8.51	1	—	—	—	—	—	—

BAUD RATE (K)	SYNC = 0, BRGH = 0, BRG16 = 1								
	FOSC = 20.000 MHz			FOSC = 10.000 MHz			FOSC = 8.000 MHz		
	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)
0.3	0.300	0.02	4165	0.300	0.02	2082	300	-0.04	1665
1.2	1.200	-0.03	1041	1.200	-0.03	520	1201	-0.16	415
2.4	2.399	-0.03	520	2.404	0.16	259	2403	-0.16	207
9.6	9.615	0.16	129	9.615	0.16	64	9615	-0.16	51
19.2	19.231	0.16	64	19.531	1.73	31	19230	-0.16	25
57.6	56.818	-1.36	21	56.818	-1.36	10	55555	3.55	8
115.2	113.636	-1.36	10	125.000	8.51	4	—	—	—

BAUD RATE (K)	SYNC = 0, BRGH = 0, BRG16 = 1								
	FOSC = 4.000 MHz			FOSC = 2.000 MHz			FOSC = 1.000 MHz		
	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)
0.3	0.300	0.04	832	300	-0.16	415	300	-0.16	207
1.2	1.202	0.16	207	1201	-0.16	103	1201	-0.16	51
2.4	2.404	0.16	103	2403	-0.16	51	2403	-0.16	25
9.6	9.615	0.16	25	9615	-0.16	12	—	—	—
19.2	19.231	0.16	12	—	—	—	—	—	—
57.6	62.500	8.51	3	—	—	—	—	—	—
115.2	125.000	8.51	1	—	—	—	—	—	—

BAUD RATE (K)	SYNC = 0, BRGH = 1, BRG16 = 1 or SYNC = 1, BRG16 = 1								
	FOSC = 20.000 MHz			FOSC = 10.000 MHz			FOSC = 8.000 MHz		
	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)
0.3	0.300	0.00	16665	0.300	0.00	8332	300	-0.01	6665
1.2	1.200	0.02	4165	1.200	0.02	2082	1200	-0.04	1665
2.4	2.400	0.02	2082	2.402	0.06	1040	2400	-0.04	832
9.6	9.596	-0.03	520	9.615	0.16	259	9615	-0.16	207
19.2	19.231	0.16	259	19.231	0.16	129	19230	-0.16	103
57.6	57.471	-0.22	86	58.140	0.94	42	57142	0.79	34
115.2	116.279	0.94	42	113.636	-1.36	21	117647	-2.12	16

TABLE 10-3: BAUD RATES FOR ASYNCHRONOUS MODES (CONTINUED)

BAUD RATE (K)	SYNC = 0, BRGH = 1, BRG16 = 1 or SYNC = 1, BRG16 = 1								
	Fosc = 4.000 MHz			Fosc = 2.000 MHz			Fosc = 1.000 MHz		
	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)
0.3	0.300	0.01	3332	300	-0.04	1665	300	-0.04	832
1.2	1.200	0.04	832	1201	-0.16	415	1201	-0.16	207
2.4	2.404	0.16	415	2403	-0.16	207	2403	-0.16	103
9.6	9.615	0.16	103	9615	-0.16	51	9615	-0.16	25
19.2	19.231	0.16	51	19230	-0.16	25	19230	-0.16	12
57.6	58.824	2.12	16	55555	3.55	8	—	—	—
115.2	111.111	-3.55	8	—	—	—	—	—	—

10.2.2 AUTO BAUD RATE DETECT

The EUSART module supports the automatic detection and calibration of baud rate. This feature is active only in Asynchronous mode and while the WUE bit is clear.

The automatic baud rate measurement sequence (Figure 10-1) begins whenever a Start bit is received and the ABDEN bit is set. The calculation is self-averaging.

In the Auto Baud Rate 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. In ABD mode, the internal baud rate generator is used as a counter to time the bit period of the incoming serial byte stream.

Once the ABDEN bit is set, the state machine will clear the BRG and look for a Start bit. The Auto Baud Detect must receive a byte with the value 55h (ASCII “U”, which is also the LIN bus sync character), in order to calculate the proper bit rate. The measurement takes over both a low and a high bit time in order to minimize any effects caused by asymmetry of the incoming signal. After a Start bit, the SPBRG begins counting up using the preselected clock source on the first rising edge of RX. After eight bits on the RX pin, or the fifth rising edge, an accumulated value totalling the proper BRG period is left in the SPBRGH:SPBRG registers. Once the 5th edge is seen (should correspond to the Stop bit), the ABDEN bit is automatically cleared.

While calibrating the baud rate period, the BRG registers are clocked at 1/8th the pre-configured clock rate. Note that the BRG clock will be configured by the BRG16 and BRGH bits. Independent of the BRG16 bit setting, both the SPBRG and SPBRGH will be used as a 16-bit counter. This allows the user to verify that no carry occurred for 8-bit modes, by checking for 00h in the SPBRGH register. Refer to Table 10-4 for counter clock rates to the BRG.

While the ABD sequence takes place, the USART state machine is held in IDLE. The RCIF interrupt is set once the fifth rising edge on RX is detected. The value in the RCREG needs to be read to clear the RCIF interrupt. RCREG content should be discarded.

Note 1: If the WUE bit is set with the ABDEN bit, auto baud rate detection will occur on the byte *following* the Break character (see **Section 10.3.4 “Auto-Wake-up on SYNC Break Character”**).

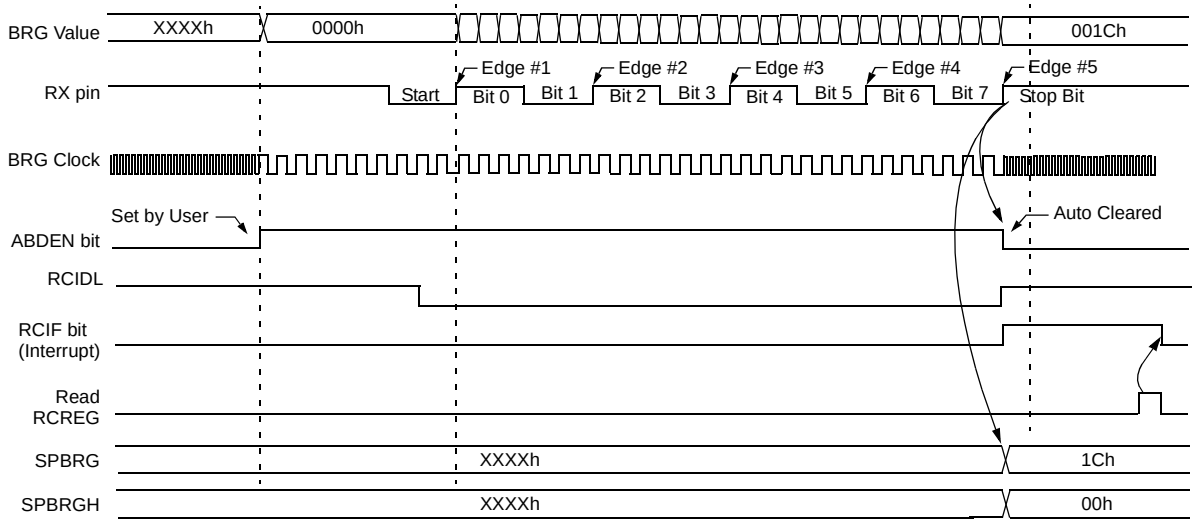
2: It is up to the user to determine that the incoming character baud rate is within the range of the selected BRG clock source. Some combinations of oscillator frequency and USART baud rates are not possible due to bit error rates. Overall system timing and communication baud rates must be taken into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

TABLE 10-4: BRG COUNTER CLOCK RATES

BRG16	BRGH	BRG Counter Clock
0	0	Fosc/512
0	1	Fosc/128
1	0	Fosc/128
1	1	Fosc/32

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

FIGURE 10-1: AUTOMATIC BAUD RATE CALCULATION



Note 1: The ABD sequence requires the USART module to be configured in Asynchronous mode and WUE = 0.

10.3 USART Asynchronous Mode

The Asynchronous mode of operation is selected by clearing the SYNC bit (TXSTA<4>). In this mode, the USART uses standard non-return-to-zero (NRZ) format (one Start bit, eight or nine data bits and one Stop bit). The most common data format is 8 bits. An on-chip dedicated 8-bit/16-bit baud rate generator can be used to derive standard baud rate frequencies from the oscillator.

The USART transmits and receives the LSb first. The USART's transmitter and receiver are functionally independent, but use the same data format and baud rate. The baud rate generator produces a clock, either x16 or x64 of the bit shift rate, depending on the BRGH and BRG16 bits (TXSTA<2> and BAUDCTL<3>). Parity is not supported by the hardware, but can be implemented in software and stored as the 9th data bit.

Asynchronous mode is available in all times. It is available in Sleep mode only when auto-wake-up on Sync Break is enabled. The baud rate generator values may need to be adjusted if the clocks are changed.

When operating in Asynchronous mode, the USART module consists of the following important elements:

- Baud Rate Generator
- Sampling Circuit
- Asynchronous Transmitter
- Asynchronous Receiver
- Auto-wake-up on Sync Break Character
- 13-bit Break Character Transmit
- Auto Baud Rate Detection

10.3.1 USART ASYNCHRONOUS TRANSMITTER

The USART transmitter block diagram is shown in Figure 10-2. The heart of the transmitter is the Transmit (serial) Shift Register (TSR). The shift register obtains its data from the read/write transmit buffer, TXREG. The TXREG register is loaded with data in software. The TSR register is not loaded until the Stop bit has been transmitted from the previous load. As soon as the Stop bit is transmitted, the TSR is loaded with new data from the TXREG register (if available).

Once the TXREG register transfers the data to the TSR register (occurs in one T_{CY}), the TXREG register is empty and flag bit TXIF (PIR1<4>) is set. This interrupt can be enabled/disabled by setting/clearing enable bit TXIE (PIE1<4>). Flag bit TXIF will be set, regardless of the state of enable bit TXIE and cannot be cleared in software. Flag bit TXIF is not cleared immediately upon loading the transmit buffer register TXREG. TXIF becomes valid in the second instruction cycle following the load instruction. Polling TXIF immediately following a load of TXREG will return invalid results.

While flag bit TXIF indicates the status of the TXREG register, another bit, TRMT (TXSTA<1>), shows the status of the TSR register. Status bit TRMT is a read only bit, which is set when the TSR register is empty. No interrupt logic is tied to this bit, so the user has to poll this bit in order to determine if the TSR register is empty.

Note 1:	The TSR register is not mapped in data memory, so it is not available to the user.
2:	Flag bit TXIF is set when enable bit TXEN is set.

To set up an Asynchronous Transmission:

1. Initialize the SPBRGH:SPBRG registers for the appropriate baud rate. Set or clear the BRGH and BRG16 bits, as required, to achieve the desired baud rate.
2. Enable the asynchronous serial port by clearing bit SYNC and setting bit SPEN.
3. If interrupts are desired, set enable bit TXIE.
4. If 9-bit transmission is desired, set transmit bit TX9. Can be used as address/data bit.
5. Enable the transmission by setting bit TXEN, which will also set bit TXIF.
6. If 9-bit transmission is selected, the ninth bit should be loaded in bit TX9D.
7. Load data to the TXREG register (starts transmission).

If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.

FIGURE 10-2: USART TRANSMIT BLOCK DIAGRAM

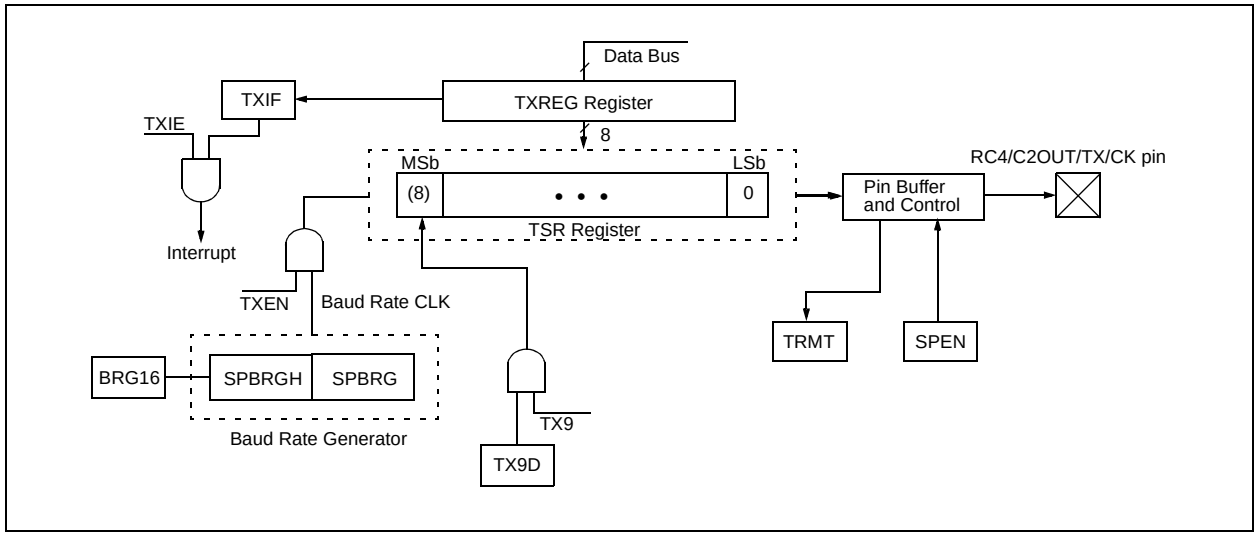


FIGURE 10-3: ASYNCHRONOUS TRANSMISSION

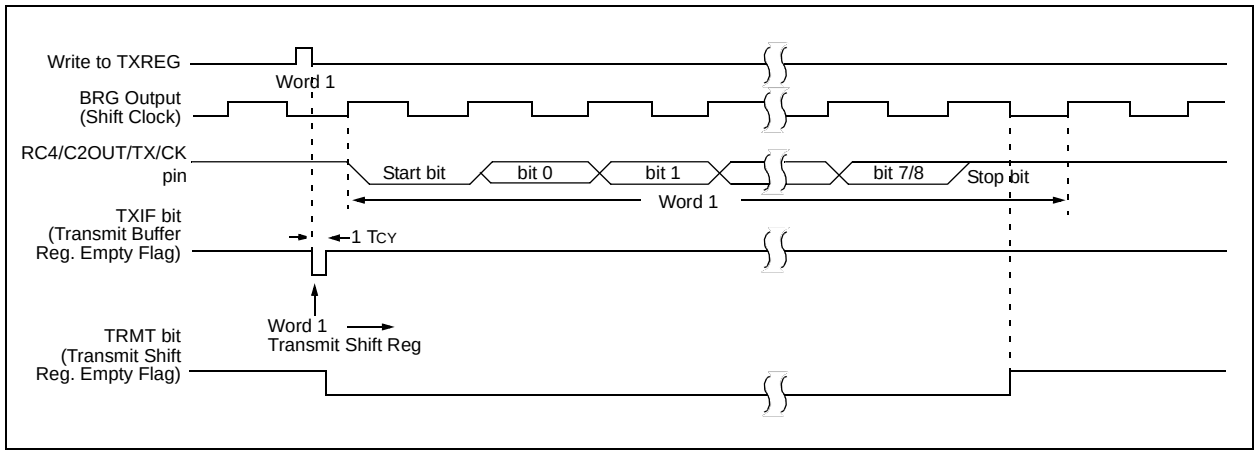
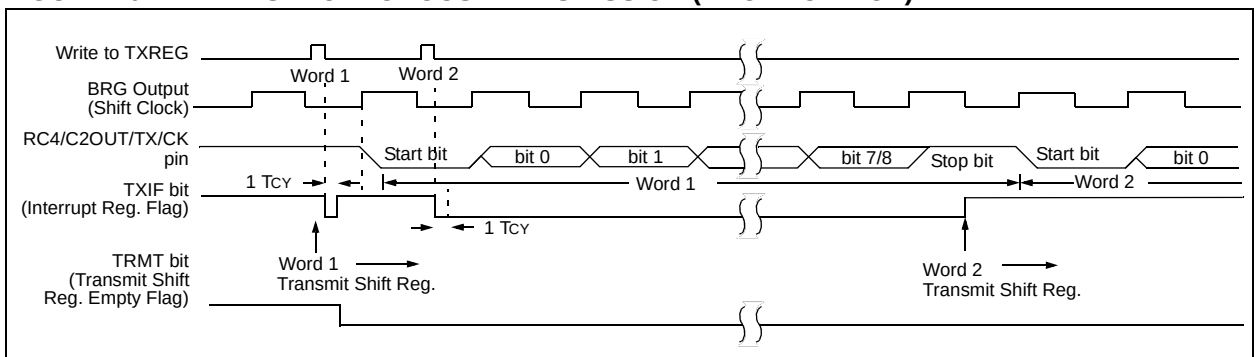


FIGURE 10-4: ASYNCHRONOUS TRANSMISSION (BACK TO BACK)



Note: This timing diagram shows two consecutive transmissions.

TABLE 10-5: REGISTERS ASSOCIATED WITH ASYNCHRONOUS TRANSMISSION

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOD	Value on all other Resets
0Ch	PIR1	EEIF	ADIF	RCIF	C2IF	C1IF	OSFIF	TXIF	TMR1IF	0000 0000	0000 0000
11h	BAUDCTL	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	00-0 0-00	00-0 0-00
12h	SPBRGH	USART Baud Rate High Generator								0000 0000	0000 0000
13h	SPBRG	USART Baud Rate Generator								0000 0000	0000 0000
14h	RCREG	USART Receive Register								0000 0000	0000 0000
15h	TXREG	USART Transmit Register								0000 0000	0000 0000
16h	TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010	0000 0010
17h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000X	0000 000X
8Ch	PIE1	EEIE	ADIE	RCIE	C2IE	C1IE	OSFIE	TXIE	TMR1IE	0000 0000	0000 0000

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

10.3.2 USART ASYNCHRONOUS RECEIVER

The receiver block diagram is shown in Figure 10-5. The data is received on the RC5/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 main receive serial shifter operates at the bit rate or at FOSC. This mode would typically be used in RS-232 systems.

To set up an Asynchronous Reception:

1. Initialize the SPBRGH:SPBRG registers for the appropriate baud rate. Set or clear the BRGH and BRG16 bits, as required, to achieve the desired baud rate.
2. Enable the asynchronous serial port by clearing bit SYNC and setting bit SPEN.
3. If interrupts are desired, set enable bit RCIE.
4. If 9-bit reception is desired, set bit RX9.
5. Enable the reception by setting bit CREN.
6. Flag bit RCIF will be set when reception is complete and an interrupt will be generated if enable bit RCIE was set.
7. Read the RCSTA register to get the 9th bit (if enabled) and determine if any error occurred during reception.
8. Read the 8-bit received data by reading the RCREG register.
9. If any error occurred, clear the error by clearing enable bit CREN.
10. If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.

10.3.3 SETTING UP 9-BIT MODE WITH ADDRESS DETECT

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 registers for the appropriate baud rate. Set or clear the BRGH and BRG16 bits, as required, to achieve the desired baud rate.
2. Enable the asynchronous serial port by clearing the SYNC bit and setting the SPEN bit.
3. If interrupts are required, set the RCEN bit and select the desired priority level with the RCIP bit.
4. Set the RX9 bit to enable 9-bit reception.
5. Set the ADDEN bit to enable address detect.
6. Enable reception by setting the CREN bit.
7. The RCIF bit will be set when reception is complete. The interrupt will be acknowledged if the RCIE and GIE bits are set.
8. Read the RCSTA register to determine if any error occurred during reception, as well as read bit 9 of data (if applicable).
9. Read RCREG to determine if the device is being addressed.
10. If any error occurred, clear the CREN bit.
11. If the device has been addressed, clear the ADDEN bit to allow all received data into the receive buffer and interrupt the CPU.

FIGURE 10-5: USART RECEIVE BLOCK DIAGRAM

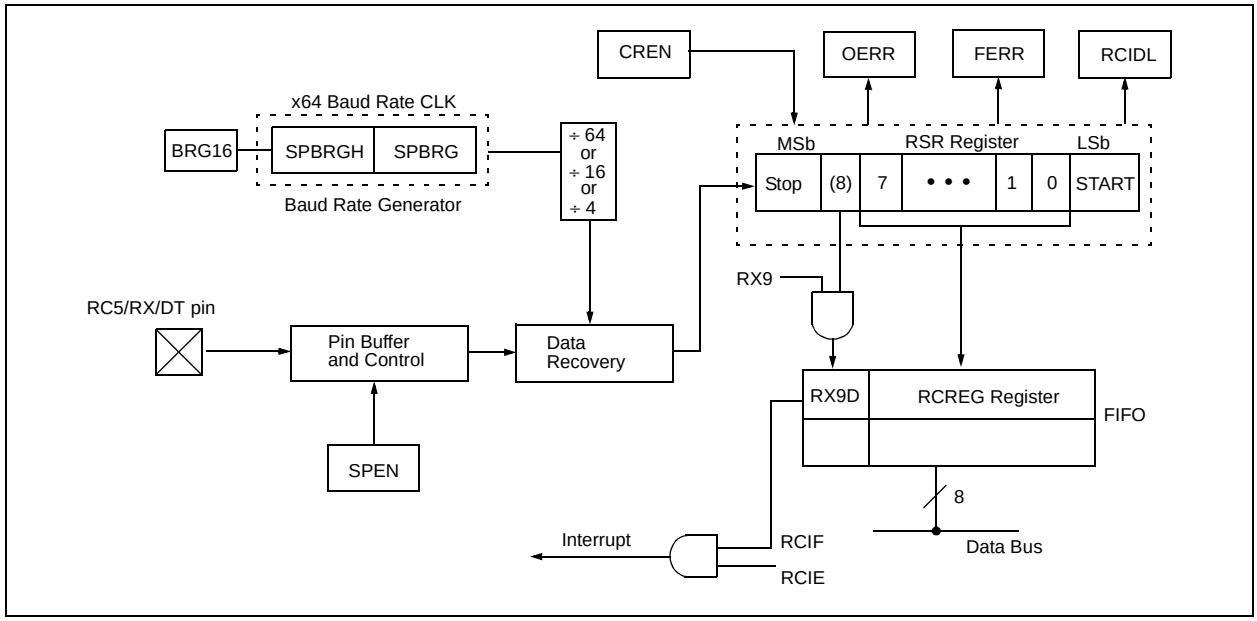
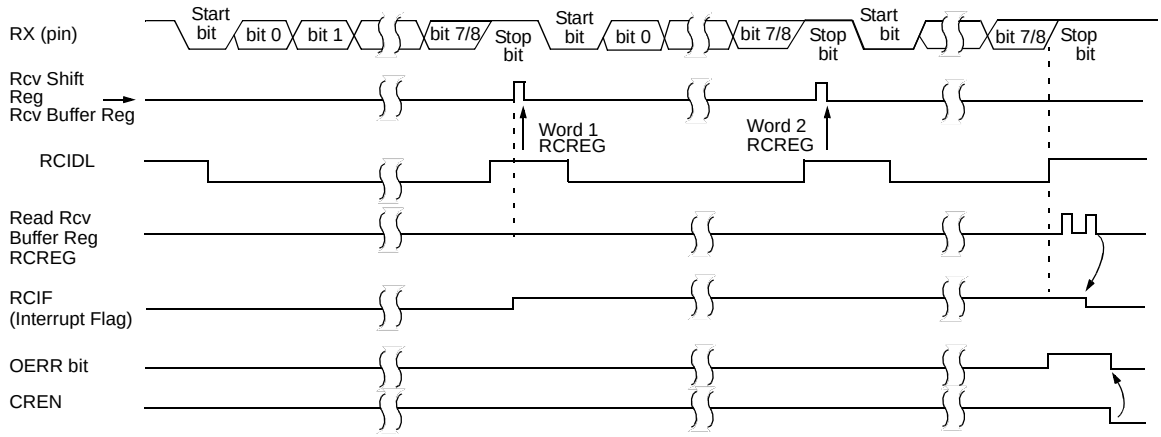


FIGURE 10-6: ASYNCHRONOUS RECEPTION



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

TABLE 10-6: REGISTERS ASSOCIATED WITH ASYNCHRONOUS RECEPTION

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOD	Value on all other Resets
0Ch	PIR1	EEIF	ADIF	RCIF	C2IF	C1IF	OSFIF	TXIF	TMR1IF	0000 0000	0000 0000
11h	BAUDCTL	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	00-0 0-00	00-0 0-00
12h	SPBRGH	USART Baud Rate High Generator								0000 0000	0000 0000
13h	SPBRG	USART Baud Rate Generator								0000 0000	0000 0000
14h	RCREG	USART Receive Register								0000 0000	0000 0000
15h	TXREG	USART Transmit Register								0000 0000	0000 0000
16h	TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010	0000 0010
17h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000X	0000 000X
8Ch	PIE1	EEIE	ADIE	RCIE	C2IE	C1IE	OSFIE	TXIE	TMR1IE	0000 0000	0000 0000

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

10.3.4 AUTO-WAKE-UP ON SYNC BREAK CHARACTER

The auto-wake-up feature allows the controller to wake-up due to activity on the RX/DT line, despite the baud clock being turned off. This allows communications systems to save power by only responding to direct requests.

Setting the WUE bit (BAUDCTL<1>) enables the auto-wake-up feature. When the auto-wake-up feature is enabled, the next falling edge on the RX/DT line will trigger an RCIF interrupt. The WUE bit will automatically clear after the rising RX/DT edge after triggering a falling edge. Receiving a RCIF interrupt after setting the WUE bit signals to the user that the wake-up event has occurred. See Figure 10-7 and Figure 10-8 for timing details of the auto-wake-up process.

10.3.4.1 Special Considerations Using Auto- Wake-up

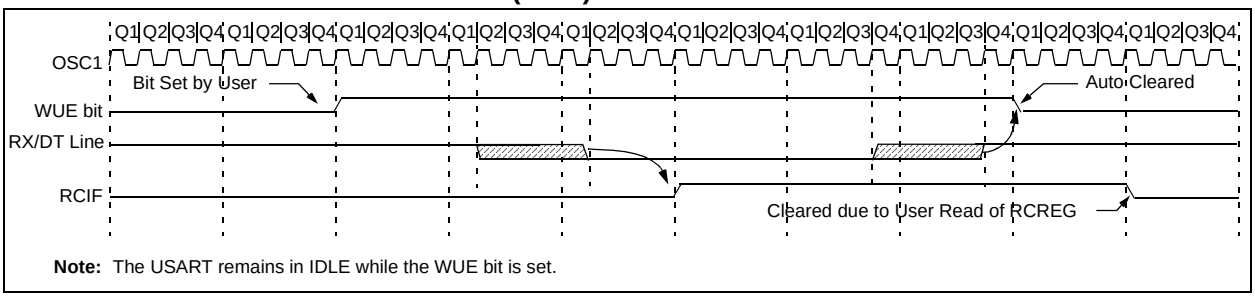
The auto-wake-up function is edge sensitive. To prevent data errors or framing errors, the data following the Break should be all ‘0’s until the baud clock is stable. If the LP, XT or HS oscillators are used, the oscillator start-up time will affect the amount of time the application must wait before receiving valid data. Special care should be taken when using the two-speed start-up or the fail-safe clock monitor because the application will start running from the internal oscillator before the primary oscillator is ready.

Because the auto-wake-up feature uses the RCIF flag to signify the wake-up event, the application should discard the data read from RCREG when servicing the RCIF flag after setting the WUE bit.

When entering Sleep with auto-wake-up enabled, the following procedure should be used.

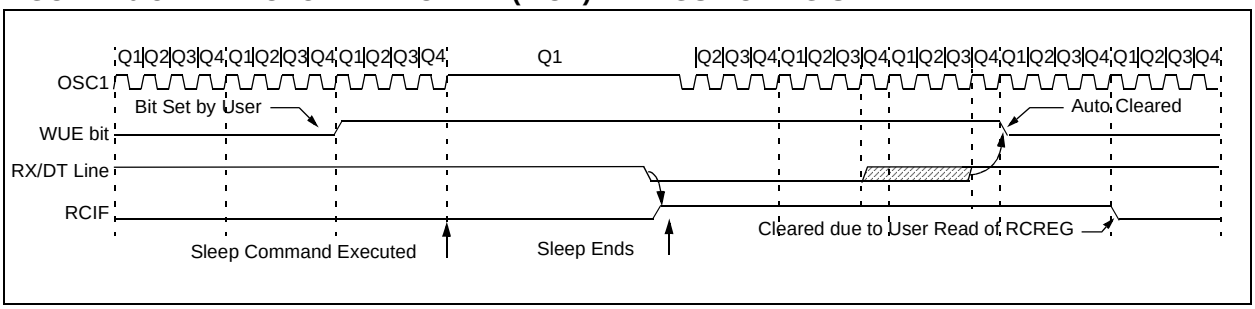
1. Clear all interrupt flags including RCIF.
2. Check RCIDL to ensure no receive is currently in progress.
3. No characters are being received so the WUE bit can be set.
4. Sleep.

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



Note: The USART remains in IDLE while the WUE bit is set.

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



10.3.5 BREAK CHARACTER SEQUENCE

The EUSART module has the capability of sending the special Break character sequences that are required by the LIN bus standard. The Break character transmit consists of a Start bit, followed by 12 '0' bits and a Stop bit. The frame Break character is sent whenever the SENDB and TXEN bits (TXSTA<3> and TXSTA<5>) are set, while the Transmit Shift register is loaded with data. Note that 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).

Note that the data value written to the TXREG for the Break character is ignored. The write simply serves the purpose of initiating the proper sequence.

The TRMT bit 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.

10.3.5.1 Break and Sync Transmit Sequence

The following sequence will send 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 USART for the desired mode.
2. Set the TXEN and SENDB bits to setup the Break character.
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. The Sync character now transmits in the Pre-Configured mode.

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

10.3.6 RECEIVING A BREAK CHARACTER

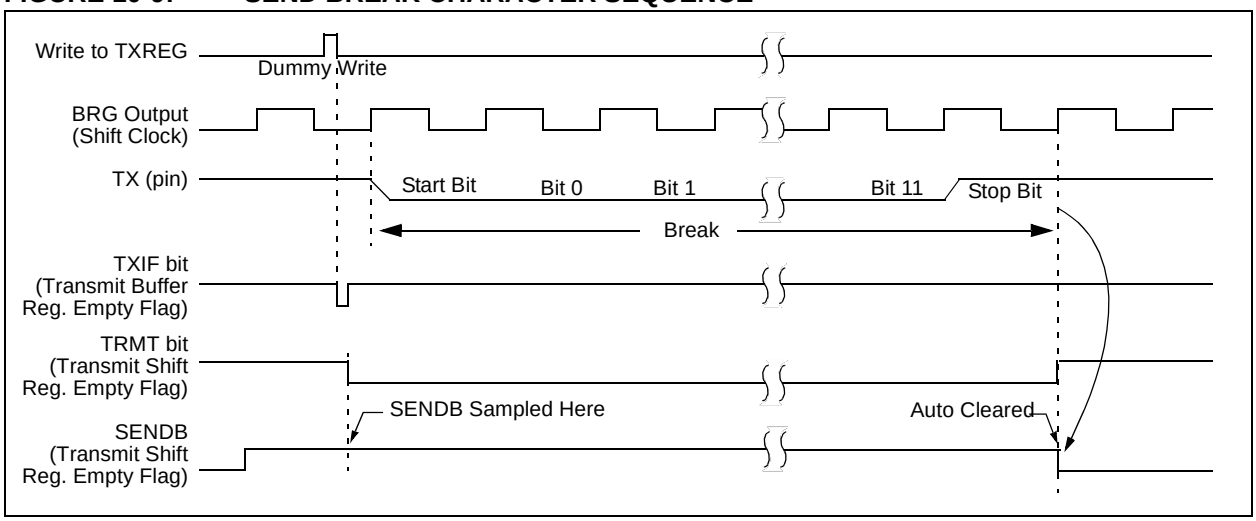
The Enhanced USART module can receive a Break character in two ways.

The first method forces to configure the baud rate at a frequency of 9/13 the typical speed. This allows for the Stop bit transition to be at the correct sampling location (13 bits for Break versus Start bit and 8 data bits for typical data).

The second method uses the auto-wake-up feature described in **Section 10.3.4 "Auto-Wake-up on SYNC Break Character"**. By enabling this feature, the USART 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 Rate Detect feature. For both methods, the user can set the ABD bit before placing the USART in its Sleep mode.

FIGURE 10-9: SEND BREAK CHARACTER SEQUENCE



10.4 USART Synchronous Master Mode

The Synchronous Master mode is entered by setting the CSRC bit (TXSTA<7>). In this mode, the data is transmitted in a half-duplex manner (i.e., transmission and reception do not occur at the same time). When transmitting data, the reception is inhibited and vice versa. Synchronous mode is entered by setting bit SYNC (TXSTA<4>). In addition, enable bit SPEN (RCSTA<7>) is set in order to configure the RC4/C2OUT/TX/CK and RC5/RX/DT I/O pins to CK (clock) and DT (data) lines, respectively.

The Master mode indicates that the processor transmits the master clock on the CK line. Clock polarity is selected with the SCKP bit (BAUDCTL<5>); setting SCKP sets the IDLE state on CK as high, while clearing the bit, sets the IDLE state low. This option is provided to support Microwire® devices with this module.

10.4.1 USART SYNCHRONOUS MASTER TRANSMISSION

The USART transmitter block diagram is shown in Figure 10-2. The heart of the transmitter is the Transmit (serial) Shift Register (TSR). The shift register obtains its data from the read/write transmit buffer register TXREG. The TXREG register is loaded with data in software. The TSR register is not loaded until the last bit has been transmitted from the previous load. As soon as the last bit is transmitted, the TSR is loaded with new data from the TXREG (if available).

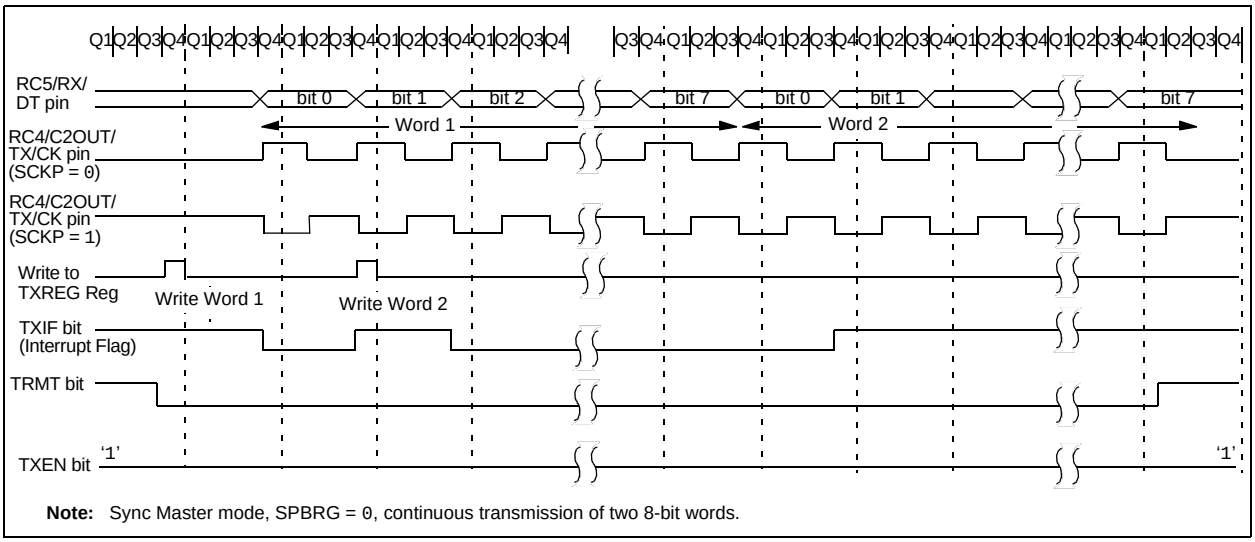
Once the TXREG register transfers the data to the TSR register (occurs in one TCYCLE), the TXREG is empty and interrupt bit TXIF (PIR1<4>) is set. The interrupt can be enabled/disabled by setting/clearing enable bit TXIE (PIE1<4>). Flag bit TXIF will be set, regardless of the state of enable bit TXIE, and cannot be cleared in software. It will reset only when new data is loaded into the TXREG register.

While flag bit TXIF indicates the status of the TXREG register, another bit, TRMT (TXSTA<1>), shows the status of the TSR register. TRMT is a read-only bit, which is set when the TSR is empty. No interrupt logic is tied to this bit, so the user has to poll this bit in order to determine if the TSR register is empty. The TSR is not mapped in data memory, so it is not available to the user.

To set up a Synchronous Master Transmission:

1. Initialize the SPBRGH:SPBRG registers 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. If interrupts are desired, set enable bit TXIE.
4. If 9-bit transmission is desired, set bit TX9.
5. Enable the transmission by setting bit TXEN.
6. If 9-bit transmission is selected, the ninth bit should be loaded in bit TX9D.
7. Start transmission by loading data to the TXREG register.
8. If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.

FIGURE 10-10: SYNCHRONOUS TRANSMISSION



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

FIGURE 10-11: SYNCHRONOUS TRANSMISSION (THROUGH TXEN)

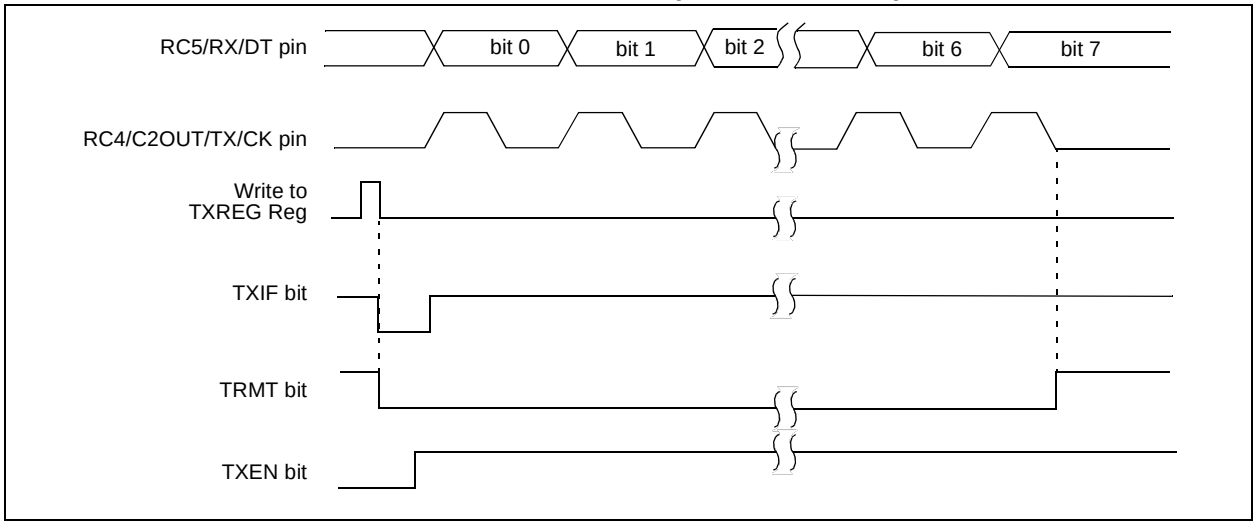


TABLE 10-7: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER TRANSMISSION

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOD	Value on all other Resets
0Ch	PIR1	EEIF	ADIF	RCIF	C2IF	C1IF	OSFIF	TXIF	TMR1IF	0000 0000	0000 0000
11h	BAUDCTL	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	00-0 0-00	00-0 0-00
12h	SPBRGH	USART Baud Rate High Generator								0000 0000	0000 0000
13h	SPBRG	USART Baud Rate Generator								0000 0000	0000 0000
14h	RCREG	USART Receive Register								0000 0000	0000 0000
15h	TXREG	USART Transmit Register								0000 0000	0000 0000
16h	TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010	0000 0010
17h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000X	0000 000X
8Ch	PIE1	EEIE	ADIE	RCIE	C2IE	C1IE	OSFIE	TXIE	TMR1IE	0000 0000	0000 0000

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

10.4.2 USART SYNCHRONOUS MASTER RECEPTION

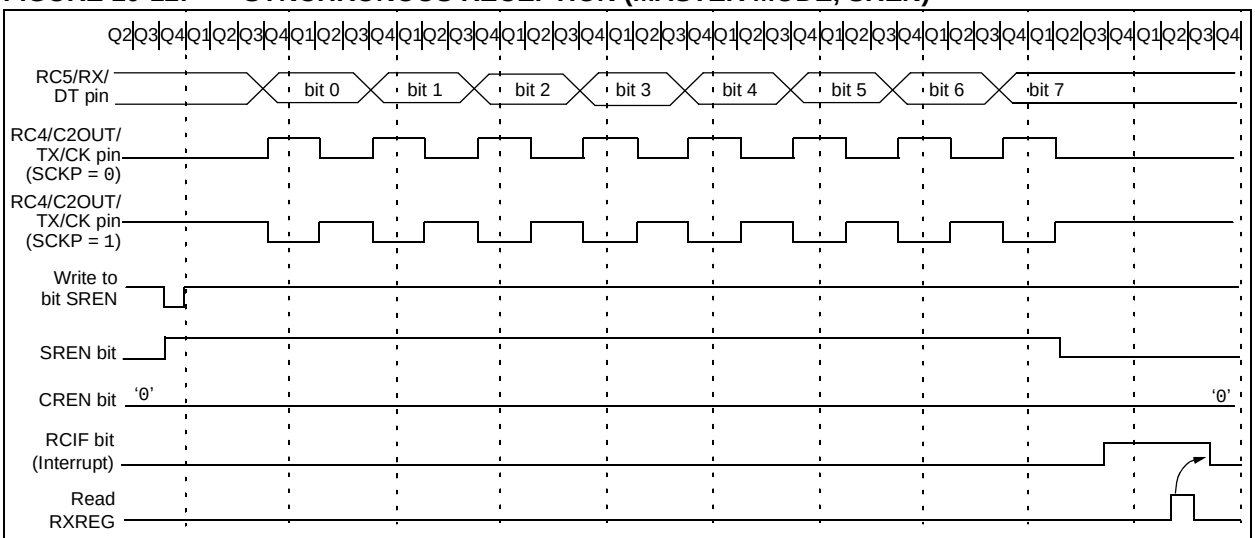
Once Synchronous mode is selected, reception is enabled by setting either the Single Receive Enable bit SREN (RCSTA<5>), or the Continuous Receive Enable bit, CREN (RCSTA<4>). Data is sampled on the RC5/RX/DT pin on the falling edge of the clock.

If enable bit SREN is set, only a single word is received. If enable bit CREN is set, the reception is continuous until CREN is cleared. If both bits are set, then CREN takes precedence.

To set up a Synchronous Master Reception:

1. Initialize the SPBRGH:SPBRG registers 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 interrupts are desired, set enable bit RCIE.
5. If 9-bit reception is desired, set bit RX9.
6. If a single reception is required, set bit SREN. For continuous reception, set bit CREN.
7. Interrupt flag bit RCIF will be set when reception is complete and an interrupt will be generated if the enable bit RCIE was set.
8. Read the RCSTA register to get the 9th 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 any error occurred, clear the error by clearing bit CREN.
11. If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.

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



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

TABLE 10-8: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER RECEPTION

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOD	Value on all other Resets
0Ch	PIR1	EEIF	ADIF	RCIF	C2IF	C1IF	OSFIF	TXIF	TMR1IF	0000 0000	0000 0000
11h	BAUDCTL	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	00-0 0-00	00-0 0-00
12h	SPBRGH	USART Baud Rate High Generator								0000 0000	0000 0000
13h	SPBRG	USART Baud Rate Generator								0000 0000	0000 0000
14h	RCREG	USART Receive Register								0000 0000	0000 0000
15h	TXREG	USART Transmit Register								0000 0000	0000 0000
16h	TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010	0000 0010
17h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000X	0000 000X
8Ch	PIE1	EEIE	ADIE	RCIE	C2IE	C1IE	OSFIE	TXIE	TMR1IE	0000 0000	0000 0000

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

10.5 USART Synchronous Slave Mode

Synchronous Slave mode is entered by clearing bit CSRC (TXSTA<7>). This mode differs from the Synchronous Master mode in that the shift clock is supplied externally at the RC4/C2OUT/TX/CK pin (instead of being supplied internally in Master mode). This allows the device to transfer or receive data while in any Low-power mode.

10.5.1 USART SYNCHRONOUS SLAVE TRANSMIT

The operation of the Synchronous Master and Slave modes are identical, 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:

a) The first word will immediately transfer to the TSR register and transmit.
b) The second word will remain in TXREG register.
c) Flag bit TXIF will not be set.
d) When the first word has been shifted out of TSR, the TXREG register will transfer the second word to the TSR and flag bit TXIF will now be set.
e) If enable bit TXIE is set, the interrupt will wake the chip from Sleep. If the global interrupt is enabled, the program will branch to the interrupt vector.

To set up a Synchronous Slave Transmission:

1. Enable the synchronous slave serial port by setting bits SYNC and SPEN and clearing bit CSRC.
2. Clear bits CREN and SREN.
3. If interrupts are desired, set enable bit TXIE.
4. If 9-bit transmission is desired, set bit TX9.
5. Enable the transmission by setting enable bit TXEN.
6. If 9-bit transmission is selected, the ninth bit should be loaded in bit TX9D.
7. Start transmission by loading data to the TXREG register.
8. If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.

TABLE 10-9: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE TRANSMISSION

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOD	Value on all other Resets
0Ch	PIR1	EEIF	ADIF	RCIF	C2IF	C1IF	OSFIF	TXIF	TMR1IF	0000 0000	0000 0000
11h	BAUDCTL	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	00-0 0-00	00-0 0-00
12h	SPBRGH	USART Baud Rate High Generator								0000 0000	0000 0000
13h	SPBRG	USART Baud Rate Generator								0000 0000	0000 0000
14h	RCREG	USART Receive Register								0000 0000	0000 0000
15h	TXREG	USART Transmit Register								0000 0000	0000 0000
16h	TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010	0000 0010
17h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000X	0000 000X
8Ch	PIE1	EEIE	ADIE	RCIE	C2IE	C1IE	OSFIE	TXIE	TMR1IE	0000 0000	0000 0000

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

10.5.2 USART SYNCHRONOUS SLAVE RECEPTION

The operation of the Synchronous Master and Slave modes is identical, except in the case of Sleep, or any IDLE mode and bit SREN, which is a “don't care” in Slave mode.

If receive is enabled by setting the CREN bit prior to entering Sleep, then a word may be received. 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 chip from Sleep. If the global interrupt is enabled, the program will branch to the interrupt vector.

To set up a Synchronous Slave Reception:

1. Enable the synchronous master serial port by setting bits SYNC and SPEN and clearing bit CSRC.
2. If interrupts are desired, set enable bit RCIE.
3. If 9-bit reception is desired, set bit RX9.
4. To enable reception, set enable bit CREN.
5. Flag bit RCIF will be set when reception is complete. An interrupt will be generated if enable bit RCIE was set.
6. Read the RCSTA register to get the 9th bit (if enabled) and determine if any error occurred during reception.
7. Read the 8-bit received data by reading the RCREG register.
8. If any error occurred, clear the error by clearing bit CREN.
9. If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.

TABLE 10-10: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE RECEPTION

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOD	Value on all other Resets
0Ch	PIR1	EEIF	ADIF	RCIF	C2IF	C1IF	OSFIF	TXIF	TMR1IF	0000 0000	0000 0000
11h	BAUDCTL	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	00-0 0-00	00-0 0-00
12h	SPBRGH	USART Baud Rate High Generator								0000 0000	0000 0000
13h	SPBRG	USART Baud Rate Generator								0000 0000	0000 0000
14h	RCREG	USART Receive Register								0000 0000	0000 0000
15h	TXREG	USART Transmit Register								0000 0000	0000 0000
16h	TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010	0000 0010
17h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000X	0000 000X
8Ch	PIE1	EEIE	ADIE	RCIE	C2IE	C1IE	OSFIE	TXIE	TMR1IE	0000 0000	0000 0000

Legend: x = unknown, - = unimplemented locations read as ‘0’. Shaded cells are not used for Asynchronous 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 Detect (BOD)
- 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).

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 (ADDRESS: 2007h)

—	—	FCMEN	IESO	BODEN1	BODEN0	$\overline{CPD}$	$\overline{CP}$	MCLRE	$\overline{PWRTE}$	WDTE	FOSC2	FOSC1	FOSC0
bit 13													bit 0

bit 13-12 **Unimplemented**: Read as '1'

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 **BODEN<1:0>:** Brown-out Detect Selection bits(1)
- 11 = BOD enabled
- 10 = BOD enabled during operation and disabled in Sleep
- 01 = BOD controlled by SBODEN bit (PCON<4>)
- 00 = BOD disabled

bit 7 **$\overline{CPD}$:** Data Code Protection bit(2)
- 1 = Data memory code protection is disabled
- 0 = Data memory code protection is enabled

bit 6 **$\overline{CP}$:** Code Protection bit(3)
- 1 = Program memory code protection is disabled
- 0 = Program memory code protection is enabled

bit 5 **MCLRE:** RA3/$\overline{MCLR}$ pin function select bit(4)
- 1 = RA3/$\overline{MCLR}$ pin function is $\overline{MCLR}$
- 0 = RA3/$\overline{MCLR}$ pin function is digital input, $\overline{MCLR}$ internally tied to V_{DD}

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

bit 3 **WDTE:** Watchdog Timer Enable bit
- 1 = WDT enabled
- 0 = WDT disabled and can be enabled by SWDTEN bit (WDTCON<0>)

bit 2-0 **FOSC<2:0>:** Oscillator Selection bits
- 111 = RC oscillator: CLKOUT function on RA4/OSC2/CLKOUT pin, RC on RA5/OSC1/CLKIN
- 110 = RCIO oscillator: I/O function on RA4/OSC2/CLKOUT pin, RC on RA5/OSC1/CLKIN
- 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 Detect 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 $\overline{MCLR}$ is asserted in INTOSC or RC mode, the internal clock oscillator is disabled.

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

11.2 Calibration Bits

The Brown-out Detect (BOD), Power-on Reset (POR) and 8 MHz internal oscillator (HFINTOSC) are factory calibrated. These calibration values are stored in the Calibration Word, as shown in Register 11-2 and are mapped in program memory location 2008h.

The Calibration Word is not erased when the device is erased when using the procedure described in the "*PIC12F6XX/16F6XX Memory Programming Specification"* (DS41204). Therefore, it is not necessary to store and reprogram these values when the device is erased.

Note:	Address 2008h 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 11-2: CALIB – CALIBRATION WORD (ADDRESS: 2008h)

—	FCAL6	FCAL5	FCAL4	FCAL3	FCAL2	FCAL1	FCAL0	—	POR1	POR0	BOD2	BOD1	BOD0
bit 13													bit 0

bit 13 **Unimplemented:** Read as '0'

bit 12-6 **FCAL<6:0>:** Internal Oscillator Calibration bits
0111111 = Maximum frequency
.
.
0000001
0000000 = Center frequency
1111111
.
.
1000000 = Minimum frequency

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

bit 4-3 **POR<1:0>:** POR Calibration bits
00 = Lowest POR voltage
11 = Highest POR voltage

bit 2-0 **BOD<2:0>:** BOD Calibration bits
000 = Reserved
001 = Lowest BOD voltage
111 = Highest BOD voltage

Note 1: This location does not participate in bulk erase operations if the "PIC12F6XX/16F6XX *Memory Programming Specification"* (DS41204) procedure is used.

2: Calibration bits are reserved for factory calibration. These values can and will change across the entire range; therefore, specific values and available adjustment range can not be specified.

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

11.3 Reset

The PIC16F688 differentiates between various kinds of Reset:

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

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:

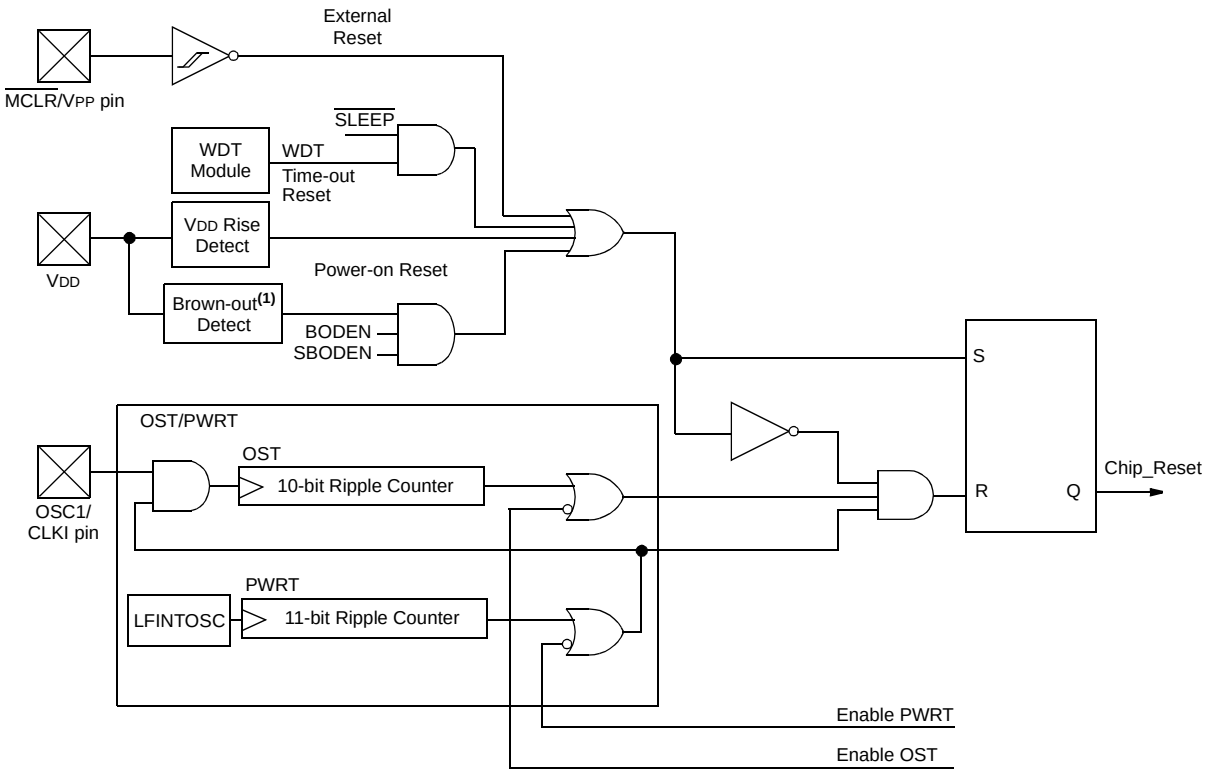
- Power-on Reset
- $\overline{\text{MCLR}}$ Reset
- $\overline{\text{MCLR}}$ Reset during Sleep
- WDT Reset
- Brown-out Detect (BOD)

They are not affected by a WDT wake-up since this is viewed as the resumption of normal operation. $\overline{\text{TO}}$ and 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



Note 1: Refer to the Configuration Word register (Register 11-1).

11.4 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 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 BOD is enabled, the maximum rise time specification does not apply. The BOD circuitry will keep the device in Reset until VDD reaches VBOD (see **Section 11.4.4 "Brown-Out Detect (BOD)"**).

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 µs.

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).

11.4.1 MCLR

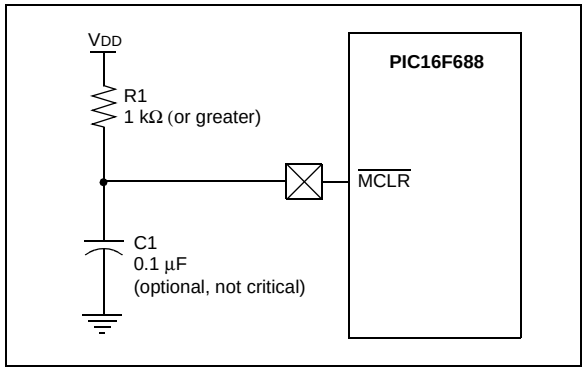
PIC16F688 has a noise filter in the MCLR Reset path. The filter will detect and ignore small pulses.

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

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

An internal MCLR option is enabled by clearing the MCLRE bit in the Configuration Word register. When cleared, MCLR is internally tied to VDD and an internal weak pull-up is enabled for the MCLR pin. In-Circuit Serial Programming is not affected by selecting the internal MCLR option.

FIGURE 11-2: RECOMMENDED MCLR CIRCUIT



11.4.2 POWER-ON RESET (POR)

The on-chip POR circuit holds the chip in Reset until VDD has reached a high enough level for proper operation. To take advantage of the POR, simply connect the 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 BOD is enabled, the maximum rise time specification does not apply. The BOD circuitry will keep the device in Reset until VDD reaches VBOD (see **Section 11.4.4 "Brown-Out Detect (BOD)"**).

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 µs.

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).

11.4.3 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 Detect. The Power-up Timer operates from the 31 kHz LFINTOSC oscillator. For more information, see **Section 3.4 "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 Detect 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"**).

11.4.4 BROWN-OUT DETECT (BOD)

The BODEN0 and BODEN1 bits in the Configuration Word register selects one of four BOD modes. Two modes have been added to allow software or hardware control of the BOD enable. When BODEN<1:0> = 01, the SBODEN bit (PCON<4>) enables/disables the BOD allowing it to be controlled in software. By selecting BODEN<1:0>, the BOD is automatically disabled in Sleep to conserve power and enabled on wake-up. In this mode, the SBODEN 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 Detect, 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 $\overline{\text{PWRTE}}$ bit in the Configuration Word.

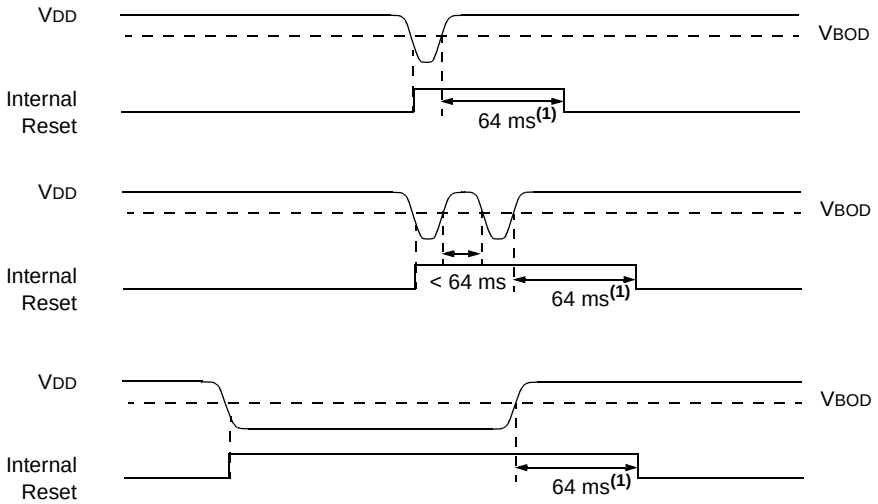
If VDD drops below VBOD while the Power-up Timer is running, the chip will go back into a Brown-out Detect and the Power-up Timer will be re-initialized. Once VDD rises above VBOD, the Power-up Timer will execute a 64 ms Reset.

11.4.5 BOD CALIBRATION

The PIC16F688 stores the BOD calibration values in fuses located in the Calibration Word (2008h). The Calibration Word is not erased when using the specified bulk erase sequence in the *PIC12F6XX/16F6XX Memory Programming Specification* (DS41204) and thus, does not require reprogramming.

Note:	Address 2008h 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.

FIGURE 11-3: BROWN-OUT SITUATIONS



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

11.4.6 TIME-OUT SEQUENCE

On power-up, the time-out sequence is as follows: first, PWRT time-out is invoked after POR has expired, then OST is activated after the PWRT time-out has expired. The total time-out will vary based on oscillator configuration and PWRTE bit status. For example, in EC mode with PWRTE bit erased (PWRT disabled), there will be no time-out at all. Figure 11.4, 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 3.6.2 "Two-Speed Start-up Sequence"** and **Section 3.7 "Fail-Safe Clock Monitor"**).

Since the time-outs occur from the POR pulse, if MCLR is kept low long enough, the time-outs will expire. Then, bringing 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.

11.4.7 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 BOD (Brown-out). BOD is unknown on Power-on Reset. It must then be set by the user and checked on subsequent Resets to see if BOD = 0, indicating that a Brown-out has occurred. The BOD status bit is a "don't care" and is not necessarily predictable if the brown-out circuit is disabled (BODEN<1:0> = 00 in the Configuration Word register).

Bit 1 is 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 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 4.2.3 "Ultra Low-Power Wake-up"** and **Section 11.4.4 "Brown-Out Detect (BOD)"**.

TABLE 11-1: TIME-OUT IN VARIOUS SITUATIONS

Oscillator Configuration	Power-up		Brown-out Detect		Wake-up from Sleep
	PWRTE = 0	PWRTE = 1	PWRTE = 0	PWRTE = 1	
XT, HS, LP	TPWRT + 1024 • TOSC	1024 • TOSC	TPWRT + 1024 • TOSC	1024 • TOSC	1024 • TOSC
RC, EC, INTOSC	TPWRT	—	TPWRT	—	—

TABLE 11-2: PCON BITS AND THEIR SIGNIFICANCE

POR	BOD	TO	PD	Condition
0	u	1	1	Power-on Reset
1	0	1	1	Brown-out Detect
u	u	0	u	WDT Reset
u	u	0	0	WDT Wake-up
u	u	u	u	MCLR Reset during normal operation
u	u	1	0	MCLR Reset during Sleep

Legend: u = unchanged, x = unknown

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

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOD	Value on all other Resets(1)
03h	STATUS	IRP	RP1	RPO	TO	PD	Z	DC	C	0001 1xxx	000q quuu
8Eh	PCON	—	—	ULPWUE	SBODEN	—	—	POR	BOD	--01 --qq	--0u --uu

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

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

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

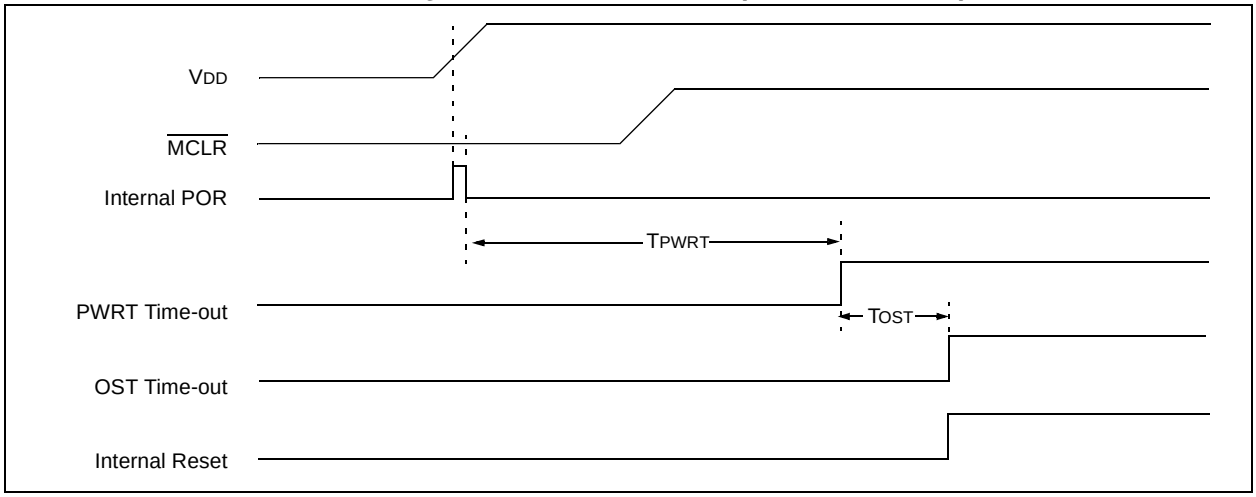


FIGURE 11-5: TIME-OUT SEQUENCE ON POWER-UP (DELAYED $\overline{MCLR}$)

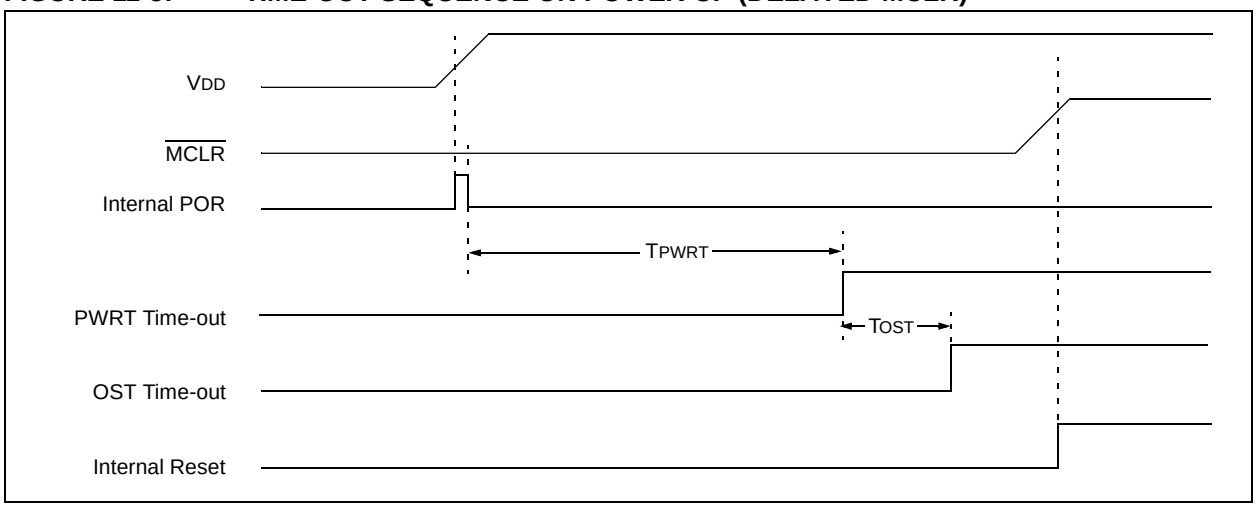


FIGURE 11-6: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ WITH VDD)

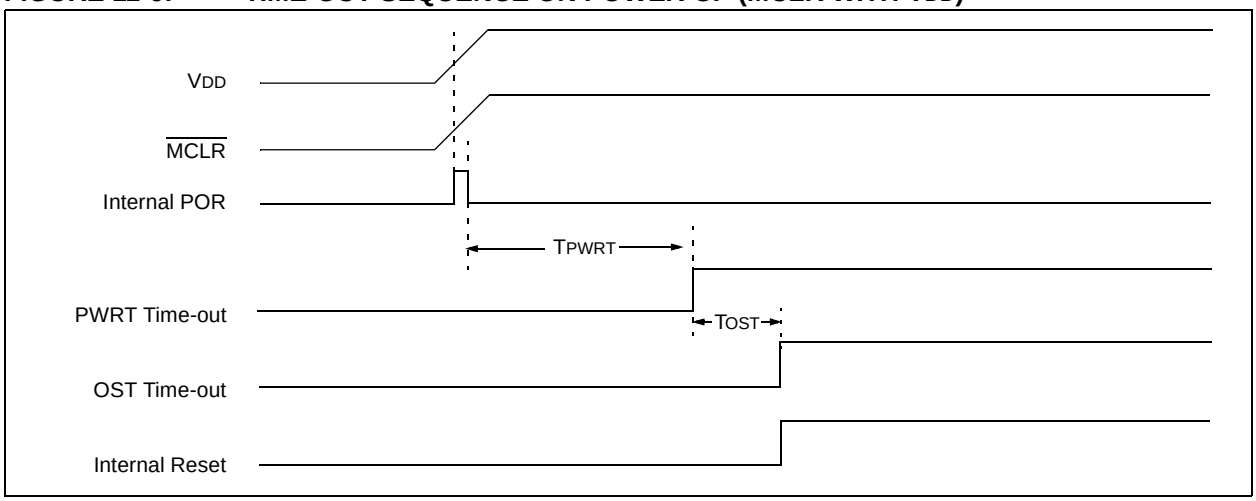


TABLE 11-4: INITIALIZATION CONDITION FOR REGISTERS

Register	Address	Power-on Reset	$\overline{\text{MCLR}}$ Reset WDT Reset Brown-out Detect(1)	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	xxxx xxxx	uuuu uuuu
TMR0	01h/101h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCL	02h/82h/102h/182h	0000 0000	0000 0000	PC + 1(3)
STATUS	03h/83h/103h/183h	0001 1xxx	000q quuu(4)	uuuq quuu(4)
FSR	04h/84h/104h/184h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTA	05h/105h	--xx xx00	--00 0000	--uu uuuu
PORTC	07h/107h	--xx xx00	--00 0000	--uu uuuu
PCLATH	0Ah/8Ah/10Ah/18Ah	---0 0000	---0 0000	---u uuuu
INTCON	0Bh/8Bh/10Bh/18Bh	0000 0000	0000 0000	uuuu uuuu(2)
PIR1	0Ch	0000 0000	0000 0000	uuuu uuuu(2)
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(1,5)	--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.

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

Register	Address	Power-on Reset	• $\overline{\text{MCLR}}$ Reset • WDT Reset • Brown-out Detect(1)	• Wake-up from Sleep through interrupt • Wake-up from Sleep through WDT time-out
OSCCON	8Fh	-110 x000	-110 x000	-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	0000 0000	uuuu uuuu
EEADRH	98h	---- 0000	0000 0000	uuuu 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
$\overline{\text{MCLR}}$ Reset during normal operation	000h	000u uuuu	--0u --uu
$\overline{\text{MCLR}}$ 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 Detect	000h	0001 1uuu	--01 --10
Interrupt Wake-up from Sleep	PC + 1(1)	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.

11.5 Interrupts

The PIC16F688 has 11 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

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 (INTCON<7>), 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
- 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.

11.5.1 RA2/INT INTERRUPT

External interrupt on RA2/INT pin is edge-triggered; either rising if the INTEDG bit (OPTION<6>) is set, or falling, if the INTEDG bit is clear. When a valid edge appears on the RA2/INT pin, the INTF bit (INTCON<1>) is set. This interrupt can be disabled by clearing the INTE control bit (INTCON<4>). 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 11.8 "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'.

11.5.2 TMR0 INTERRUPT

An overflow (FFh → 00h) in the TMR0 register will set the T0IF (INTCON<2>) bit. The interrupt can be enabled/disabled by setting/clearing T0IE (INTCON<5>) bit. See **Section 5.0 "Timer0 Module"** for operation of the Timer0 module.

11.5.3 PORTA INTERRUPT

An input change on PORTA change sets the RAIF (INTCON<0>) bit. The interrupt can be enabled/disabled by setting/clearing the RAIE (INTCON<3>) bit. 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

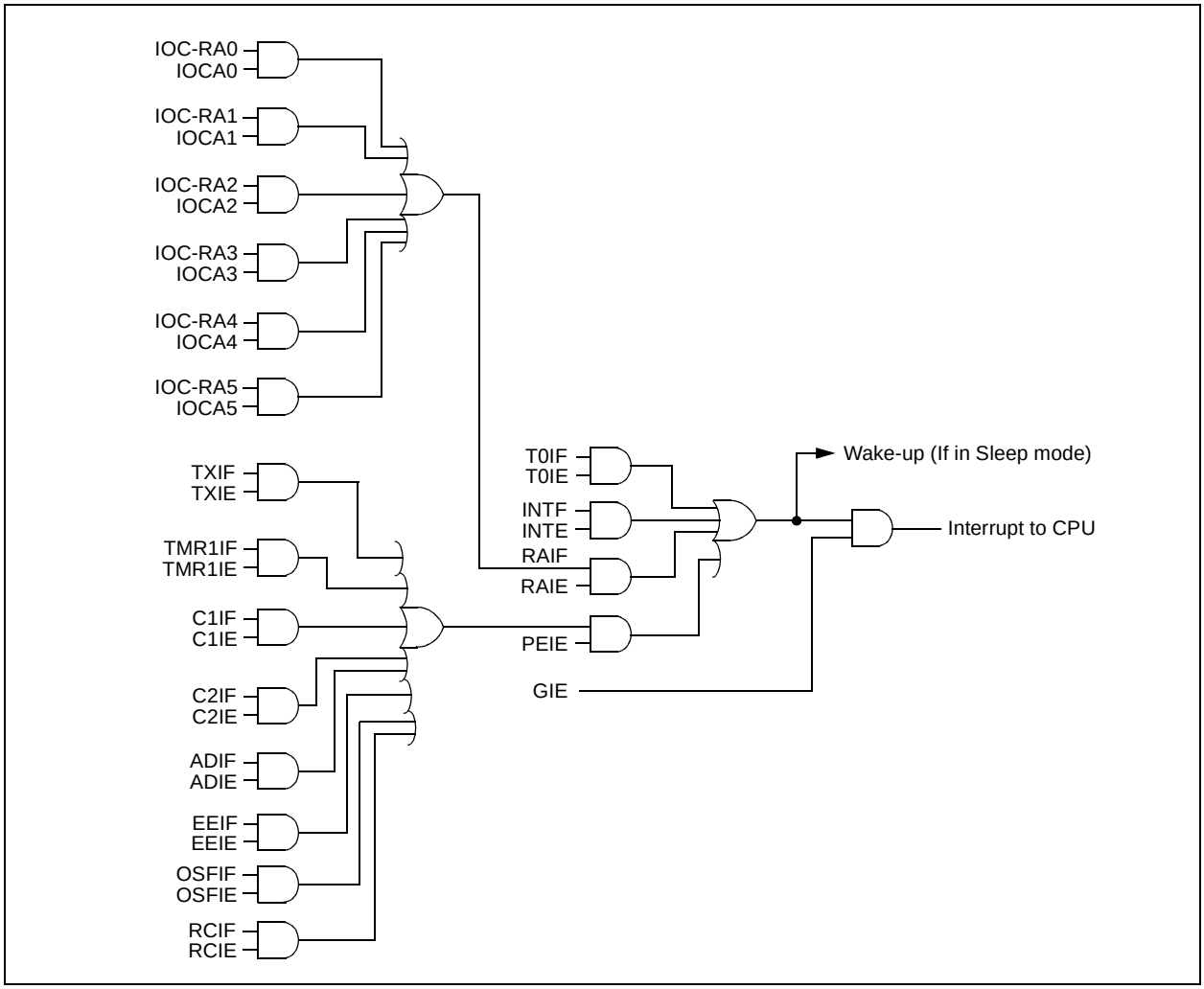
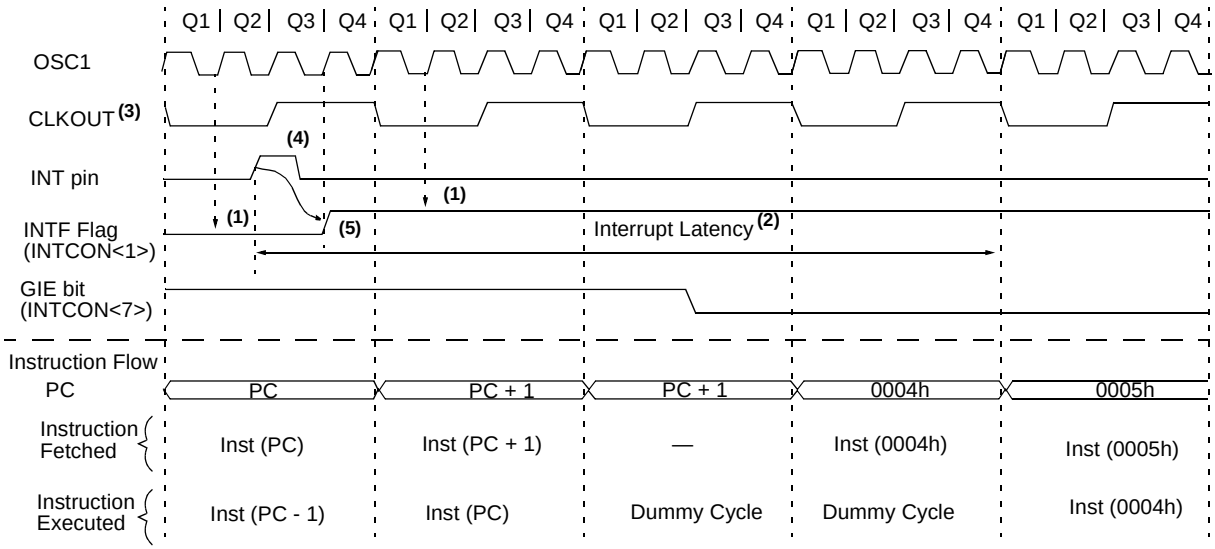


FIGURE 11-8: INT PIN INTERRUPT TIMING



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

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

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

4: For minimum width of INT pulse, refer to AC specifications in **Section 14.0 "Electrical Specifications"**.

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

TABLE 11-6: SUMMARY OF INTERRUPT REGISTERS

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOD	Value on all other Resets
0Bh, 8Bh	INTCON	GIE	PEIE	T0IE	INTE	RAIE	T0IF	INTF	RAIF	0000 0000	0000 0000
0Ch	PIR1	EEIF	ADIF	RCIF	C2IF	C1IF	OSFIF	TXIF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	EEIE	ADIE	RCIE	C2IE	C1IE	OSFIE	TXIE	TMR1IE	0000 0000	0000 0000

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

11.6 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
CLRF    STATUS            ;bank 0, regardless of current bank, Clears IRP,RP1,RP0
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
```


11.7 Watchdog Timer (WDT)

For PIC16F688, the WDT has been modified from previous 16F devices. The new WDT is code and functionally compatible with previous 16F WDT modules and adds a 16-bit prescaler to the WDT. This allows the user to have a scaler value for the WDT and TMR0 at the same time. In addition, the WDT time-out value can be extended to 268 seconds. WDT is cleared under certain conditions described in Table 11-7.

11.7.1 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 16F 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.

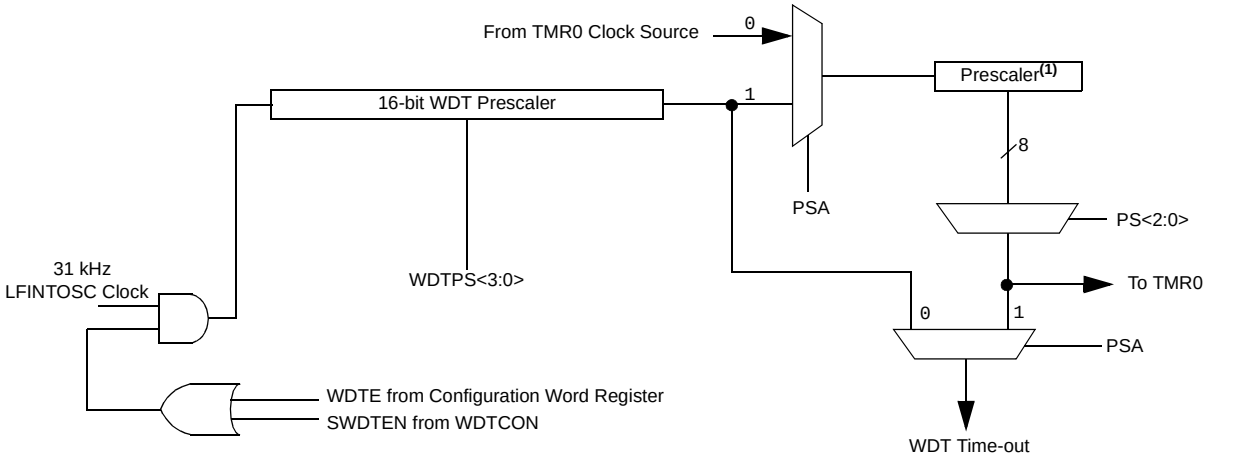
11.7.2 WDT CONTROL

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 (WDTCON<0>) 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 (OPTION_REG) have the same function as in previous versions of the 16F family of microcontrollers. See **Section 5.0 "Timer0 Module"** for more information.

FIGURE 11-9: WATCHDOG TIMER BLOCK DIAGRAM



Note 1: This is the shared Timer0/WDT prescaler. See **Section 5.4 "Prescaler"** for more information.

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

REGISTER 11-3: WDTCON – WATCHDOG TIMER CONTROL REGISTER (ADDRESS: 18h)

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

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(1)

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.

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

TABLE 11-8: SUMMARY OF WATCHDOG TIMER REGISTERS

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
18h	WDTCON	—	—	—	WDTPS3	WDTPS2	WSTPS1	WDTPS0	SWDTEN
81h	OPTION_REG	$\overline{RAPU}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
2007h(1)	CONFIG	$\overline{CPD}$	$\overline{CP}$	MCLRE	$\overline{PWRTE}$	WDTE	FOSC2	FOSC1	FOSC0

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

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

11.8 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.
- 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 T0CKI 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.

11.8.1 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 PD bits in the Status register can be used to determine the cause of device Reset. The PD bit, which is set on power-up, is cleared when Sleep is invoked. $\overline{TO}$ bit is cleared if WDT wake-up occurred.

The following peripheral interrupts can wake the device from Sleep:

1. TMR1 Interrupt. Timer1 must be operating as an asynchronous counter.
2. EUSART Receive Interrupt.
3. ULPWU Interrupt.
4. A/D conversion (when A/D clock source is RC).
5. EEPROM write operation completion.
6. Comparator output changes state.
7. Interrupt-on-change.
8. External Interrupt from INT pin.

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.

11.8.2 WAKE-UP USING INTERRUPTS

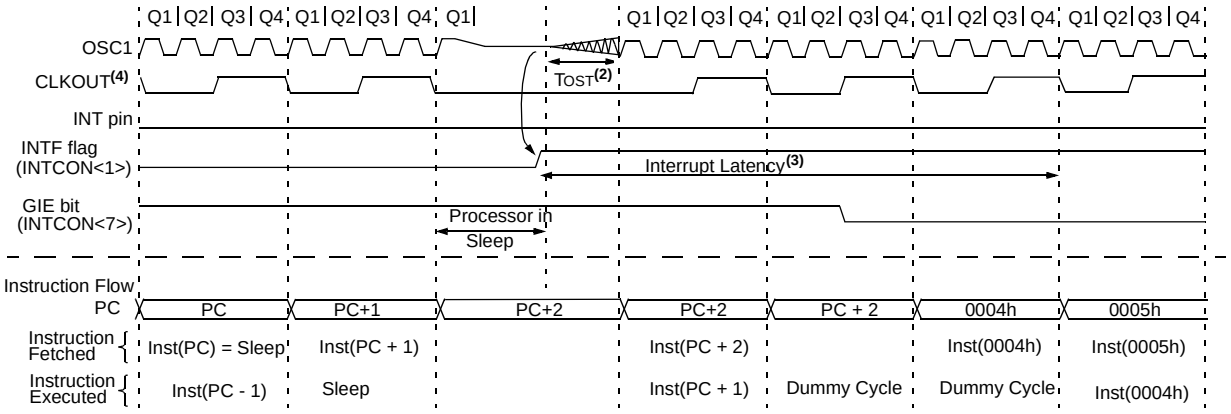
When global interrupts are disabled (GIE cleared) and any interrupt source has both its interrupt enable bit and interrupt flag bit set, one of the following will occur:

- 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 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 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 PD bit. If the PD bit is set, the SLEEP instruction was executed as a NOP.

To ensure that the WDT is cleared, a CLRWDT instruction should be executed before a SLEEP instruction.

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



Note 1: XT, HS or LP Oscillator mode assumed.
2: TOST = 1024 TOSC (drawing not to scale). This delay does not apply to EC and RC Oscillator modes.
3: GIE = 1 assumed. In this case after wake-up, the processor jumps to 0004h. If GIE = 0, execution will continue in-line.
4: CLKOUT is not available in XT, HS, LP or EC Oscillator modes, but shown here for timing reference.

11.9 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.

11.10 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.

11.11 In-Circuit Serial Programming

The PIC16F688 microcontrollers can be serially programmed while in the end application circuit. This is simply done with two lines for clock and data and three other lines for:

- power
- ground
- programming voltage

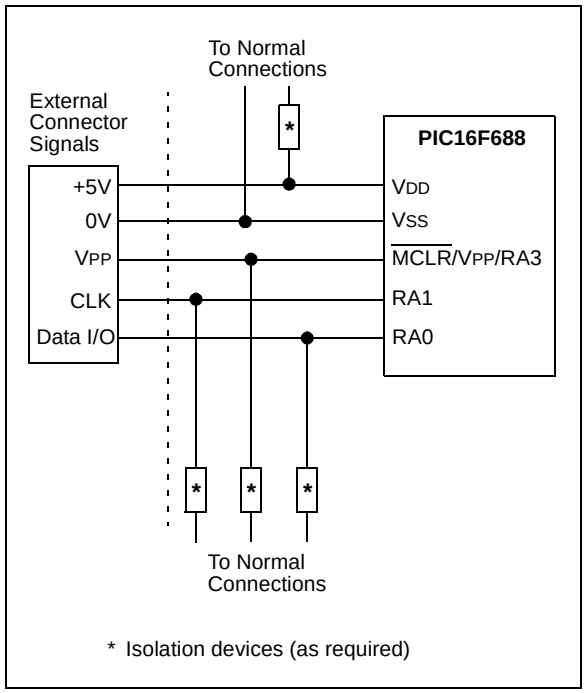
This allows customers to manufacture boards with unprogrammed devices and then program the microcontroller just before shipping the product. This also allows the most recent firmware or a custom firmware to be programmed.

The device is placed into a Program/Verify mode by holding the RA0 and RA1 pins low, while raising the $\overline{\text{MCLR}}$ (V_{PP}) pin from V_{IL} to V_{IHH}. 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 this mode.

After Reset, to place the device into Program/Verify mode, the Program Counter (PC) is at location 00h. A 6-bit command is then supplied to the device. Depending on the command, 14 bits of program data are then supplied to or from the device, depending on whether the command was a load or a read. For complete details of serial programming, please refer to the "*PIC12F6XX/16F6XX Memory Programming Specification*" (DS41204).

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

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



11.12 In-Circuit Debugger

Since in-circuit debugging requires access to the data and $\overline{MCLR}$ 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 $\overline{MCLR}$ 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.

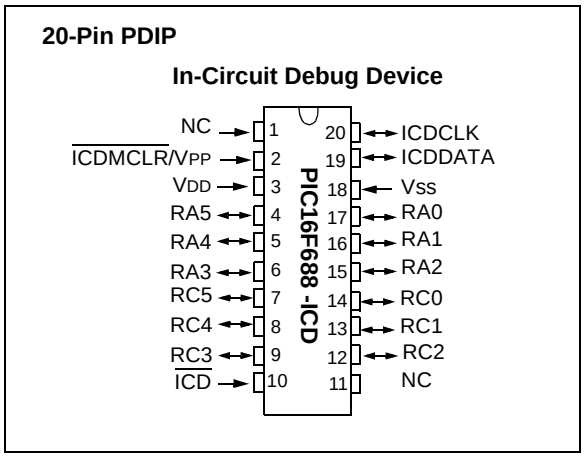
When the $\overline{ICD}$ pin on the PIC16F688 ICD device is held low, the In-Circuit Debugger functionality is enabled. This function allows simple debugging functions when used with MPLAB ICD 2. When the microcontroller has this feature enabled, some of the resources are not available for general use. 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*" (DS51292), available on Microchip's web site (www.microchip.com).

FIGURE 11-12: 20-PIN ICD PINOUT



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. A complete description of each instruction is also available in the "*PICmicro® Mid-Range MCU Family Reference Manual*" (DS33023).

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 normal 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.

Note:	To maintain upward compatibility with future products, do not use the OPTION and TRIS instructions.

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 GPIO instruction will read GPIO, clear all the data bits, then write the result back to GPIO. This example would have the unintended result of clearing the condition that set the GPIF 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
TO	Time-out bit
PD	Power-down bit

FIGURE 12-1: GENERAL FORMAT FOR INSTRUCTIONS

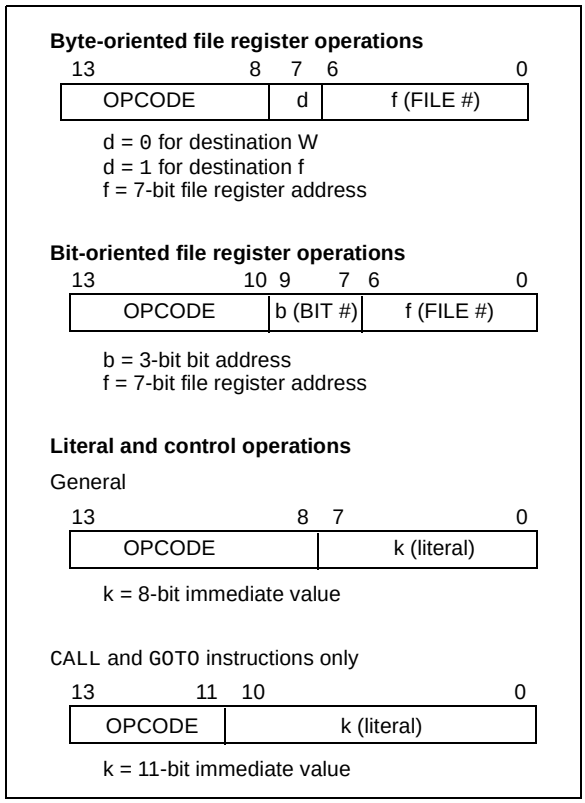


TABLE 12-2: PIC16F688 INSTRUCTION SET

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

Note 1: When an I/O register is modified as a function of itself (e.g., `MOVF 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 Program Counter (PC) is modified, or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.

Note: Additional information on the mid-range instruction set is available in the *PICmicro® Mid-Range MCU Family Reference Manual"* (DS33023).

12.2 Instruction Descriptions

ADDLW	Add Literal and W
Syntax:	[*label*] ADDLW k
Operands:	0 ≤ k ≤ 255
Operation:	(W) + k → (W)
Status Affected:	C, DC, Z
Description:	The contents of the W register are added to the eight-bit literal 'k' and the result is placed in the W register.

ADDWF	Add W and f
Syntax:	[*label*] ADDWF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(W) + (f) → (destination)
Status Affected:	C, DC, Z
Description:	Add the contents of the W register with register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'.

ANDLW	AND Literal with W
Syntax:	[*label*] ANDLW k
Operands:	0 ≤ k ≤ 255
Operation:	(W) .AND. (k) → (W)
Status Affected:	Z
Description:	The contents of W register are AND'ed with the eight-bit literal 'k'. The result is placed in the W register.

ANDWF	AND W with f
Syntax:	[*label*] ANDWF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(W) .AND. (f) → (destination)
Status Affected:	Z
Description:	AND the W register with register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'.

BCF	Bit Clear f
Syntax:	[*label*] BCF f,b
Operands:	0 ≤ f ≤ 127 0 ≤ b ≤ 7
Operation:	0 → (f<b>)
Status Affected:	None
Description:	Bit 'b' in register 'f' is cleared.

BSF	Bit Set f
Syntax:	[*label*] BSF f,b
Operands:	0 ≤ f ≤ 127 0 ≤ b ≤ 7
Operation:	1 → (f<b>)
Status Affected:	None
Description:	Bit 'b' in register 'f' is set.

BTFSS	Bit Test f, Skip if Set
Syntax:	[*label*] BTFSS f,b
Operands:	0 ≤ f ≤ 127 0 ≤ b < 7
Operation:	skip if (f<b>) = 1
Status Affected:	None
Description:	If bit 'b' in register 'f' is '0', the next instruction is executed. If bit 'b' is '1', then the next instruction is discarded and a NOP is executed instead, making this a 2-cycle instruction.

BTFSC **Bit Test, Skip if Clear**

Syntax: [*label*] BTFSC f,b

Operands: $0 \le f \le 127$
$0 \le b \le 7$

Operation: skip if (f<b>) = 0

Status Affected: None

Description: If bit 'b' in register 'f' is '1', the next instruction is executed.
If bit 'b', in register 'f', is '0', the next instruction is discarded, and a NOP is executed instead, making this a 2-cycle instruction.

CALL **Call Subroutine**

Syntax: [*label*] CALL k

Operands: $0 \le k \le 2047$

Operation: (PC)+ 1→ TOS,
k → PC<10:0>,
(PCLATH<4:3>) → PC<12:11>

Status Affected: None

Description: Call Subroutine. First, return address (PC + 1) is pushed onto the stack. The eleven-bit immediate address is loaded into PC bits <10:0>. The upper bits of the PC are loaded from PCLATH. CALL is a two-cycle instruction.

CLRF **Clear f**

Syntax: [*label*] CLRF f

Operands: $0 \le f \le 127$

Operation: 00h → (f)
1 → Z

Status Affected: Z

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

CLRW **Clear W**

Syntax: [*label*] CLRW

Operands: None

Operation: 00h → (W)
1 → Z

Status Affected: Z

Description: W register is cleared. Zero bit (Z) is set.

CLRWDT **Clear Watchdog Timer**

Syntax: [*label*] CLRWDT

Operands: None

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

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

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

COMF **Complement f**

Syntax: [*label*] COMF f,d

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

Operation: $(\bar{f}) \rightarrow$ (destination)

Status Affected: Z

Description: The contents of register 'f' are complemented. If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f'.

DECF **Decrement f**

Syntax: [*label*] DECF f,d

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

Operation: (f) - 1 → (destination)

Status Affected: Z

Description: Decrement register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'.

DECFSZ **Decrement f, Skip if 0**

Syntax: [*label*] DECFSZ f,d

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

Operation: (f) - 1 → (destination);
skip if result = 0

Status Affected: None

Description: The contents of register 'f' are decremented. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'.
If the result is '1', the next instruction is executed. If the result is '0', then a NOP is executed instead, making it a 2-cycle instruction.

GOTO **Unconditional Branch**

Syntax: [*label*] GOTO k

Operands: $0 \le k \le 2047$

Operation: k → PC<10:0>
PCLATH<4:3> → PC<12:11>

Status Affected: None

Description: GOTO is an unconditional branch. The eleven-bit immediate value is loaded into PC bits <10:0>. The upper bits of PC are loaded from PCLATH<4:3>. GOTO is a two-cycle instruction.

INCF **Increment f**

Syntax: [*label*] INCF f,d

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

Operation: (f) + 1 → (destination)

Status Affected: Z

Description: The contents of register 'f' are incremented. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'.

INCFSZ **Increment f, Skip if 0**

Syntax: [*label*] INCFSZ f,d

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

Operation: (f) + 1 → (destination),
skip if result = 0

Status Affected: None

Description: The contents of register 'f' are incremented. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'.
If the result is '1', the next instruction is executed. If the result is '0', a NOP is executed instead, making it a 2-cycle instruction.

MOVF **Move f**

Syntax: [*label*] MOVF f,d

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

Operation: (f) → (dest)

Status Affected: Z

Encoding:

00	1000	dfff	ffff

Description: The contents of register f is moved to a destination dependent upon the status of d. If d = 0, destination is W register. If d = 1, the destination is file register f itself. d = 1 is useful to test a file register since status flag Z is affected.

Words: 1

Cycles: 1

Example

```
MOVF    FSR, 0
```

After Instruction
W = value in FSR register
Z = 1

MOVWF **Move W to f**

Syntax: [*label*] MOVWF f

Operands: $0 \le f \le 127$

Operation: $(W) \rightarrow (f)$

Status Affected: None

Encoding:

00	0000	1fff	ffff

Description: Move data from W register to register ‘f’.

Words: 1

Cycles: 1

Example
```
MOVWF   OPTION
```
Before Instruction
OPTION = 0xFF
W = 0x4F
After Instruction
OPTION = 0x4F
W = 0x4F

IORLW **Inclusive OR Literal with W**

Syntax: [*label*] IORLW k

Operands: $0 \le k \le 255$

Operation: (W) .OR. $k \rightarrow (W)$

Status Affected: Z

Description: The contents of the W register are OR’ed with the eight-bit literal ‘k’. The result is placed in the W register.

IORWF **Inclusive OR W with f**

Syntax: [*label*] IORWF f,d

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

Operation: (W) .OR. $(f) \rightarrow$ (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’.

MOVLW **Move Literal to W**

Syntax: [*label*] MOVLW k

Operands: $0 \le k \le 255$

Operation: $k \rightarrow (W)$

Status Affected: None

Encoding:

11	00xx	kkkk	kkkk

Description: The eight bit literal ‘k’ is loaded into W register. The don’t cares will assemble as 0’s.

Words: 1

Cycles: 1

Example
```
MOVLW   0x5A
```
After Instruction
W = 0x5A

NOP **No Operation**

Syntax: [*label*] NOP

Operands: None

Operation: No operation

Status Affected: None

Encoding:

00	0000	0xx0	0000

Description: No operation.

Words: 1

Cycles: 1

Example
```
NOP
```


RETFIE **Return from Interrupt**

Syntax: [*label*] RETFIE

Operands: None

Operation: TOS → PC,
1 → GIE

Status Affected: None

Encoding:

00	0000	0000	1001

Description: Return from Interrupt. Stack is POPed and Top-of-Stack (TOS) is loaded in the PC. Interrupts are enabled by setting Global Interrupt Enable bit, GIE (INTCON<7>). This is a two-cycle instruction.

Words: 1

Cycles: 2

Example

```
RETFIE
```

After Interrupt
PC = TOS
GIE = 1

RETLW **Return with Literal in W**

Syntax: [*label*] RETLW k

Operands: 0 ≤ k ≤ 255

Operation: k → (W);
TOS → PC

Status Affected: None

Encoding:

11	01xx	kkkk	kkkk

Description: The W register is loaded with the eight bit literal 'k'. The program counter is loaded from the top of the stack (the return address). This is a two-cycle instruction.

Words: 1

Cycles: 2

Example

```
CALL TABLE;W contains table
          ;offset value
•         ;W now has table value
•
•
ADDWF PC  ;W = offset
RETLW k1  ;Begin table
RETLW k2  ;
•
•
•
RETLW kn  ; End of table
```

Before Instruction
W = 0x07
After Instruction
W = value of k8

RETURN **Return from Subroutine**

Syntax: [*label*] RETURN

Operands: None

Operation: TOS → PC

Status Affected: None

Description: Return from subroutine. The stack is POPed and the top of the stack (TOS) is loaded into the program counter. This is a two-cycle instruction.

RLF **Rotate Left f through Carry**

Syntax: [*label*] RLF f,d

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

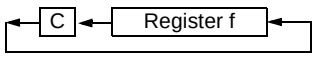
Operation: See description below

Status Affected: C

Encoding:

00	1101	dfff	ffff

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    =   1111 0110
W       =   1100 1100
C       =   1
```

RRF **Rotate Right f through Carry**

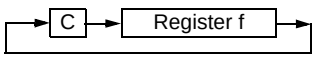
Syntax: [*label*] RRF f,d

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

Operation: See description below

Status Affected: C

Description: The contents of register ‘f’ are rotated one bit to the right through the Carry Flag. If ‘d’ is ‘0’, the result is placed in the W register. If ‘d’ is ‘1’, the result is placed back in register ‘f’.



SLEEP

Syntax: [*label*] SLEEP

Operands: None

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

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

Description: The power-down Status bit, $\overline{\text{PD}}$ is cleared. Time-out Status bit, $\overline{\text{TO}}$ is set. Watchdog Timer and its prescaler are cleared.
The processor is put into Sleep mode with the oscillator stopped.

SUBLW **Subtract W from Literal**

Syntax: [*label*] SUBLW k

Operands: $0 \le k \le 255$

Operation: k - (W) → (W)

Status Affected: C, DC, Z

Description: The W register is subtracted (2’s complement method) from the eight-bit literal ‘k’. The result is placed in the W register.

SUBWF **Subtract W from f**

Syntax: [*label*] SUBWF f,d

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

Operation: (f) - (W) → (destination)

Status Affected: C, DC, Z

Description: Subtract (2’s complement method) W register from register ‘f’. If ‘d’ is ‘0’, the result is stored in the W register. If ‘d’ is ‘1’, the result is stored back in register ‘f’.

SWAPF	Swap Nibbles in f
Syntax:	[*label*] SWAPF f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	(f<3:0>) → (destination<7:4>), (f<7:4>) → (destination<3:0>)
Status Affected:	None
Description:	The upper and lower nibbles of register 'f' are exchanged. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed in register 'f'.

XORLW	Exclusive OR Literal with W
Syntax:	[*label*] XORLW k
Operands:	$0 \leq k \leq 255$
Operation:	(W) .XOR. k → (W)
Status Affected:	Z
Description:	The contents of the W register are XOR'ed with the eight-bit literal 'k'. The result is placed in the W register.

XORWF	Exclusive OR W with f
Syntax:	[*label*] XORWF f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	(W) .XOR. (f) → (destination)
Status Affected:	Z
Description:	Exclusive OR the contents of the W register with register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'.

NOTES:

13.0 DEVELOPMENT SUPPORT

The PICmicro® microcontrollers are supported with a full range of hardware and software development tools:

- Integrated Development Environment
  - MPLAB® IDE Software
- Assemblers/Compilers/Linkers
  - MPASM™ Assembler
  - MPLAB C17 and MPLAB C18 C Compilers
  - MPLINK™ Object Linker/ MPLIB™ Object Librarian
  - MPLAB C30 C Compiler
  - MPLAB ASM30 Assembler/Linker/Library
- Simulators
  - MPLAB SIM Software Simulator
  - MPLAB dsPIC30 Software Simulator
- Emulators
  - MPLAB ICE 2000 In-Circuit Emulator
  - MPLAB ICE 4000 In-Circuit Emulator
- In-Circuit Debugger
  - MPLAB ICD 2
- Device Programmers
  - PRO MATE® II Universal Device Programmer
  - PICSTART® Plus Development Programmer
  - MPLAB PM3 Device Programmer
- Low-Cost Demonstration Boards
  - PICDEM™ 1 Demonstration Board
  - PICDEM.net™ Demonstration Board
  - PICDEM 2 Plus Demonstration Board
  - PICDEM 3 Demonstration Board
  - PICDEM 4 Demonstration Board
  - PICDEM 17 Demonstration Board
  - PICDEM 18R Demonstration Board
  - PICDEM LIN Demonstration Board
  - PICDEM USB Demonstration Board
- Evaluation Kits
  - KEELOQ®
  - PICDEM MSC
  - microID®
  - CAN
  - PowerSmart®
  - Analog

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® based application that contains:

- An interface to 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
- Mouse over variable inspection
- Extensive on-line help

The MPLAB IDE allows you to:

- Edit your source files (either assembly or C)
- One touch assemble (or compile) and download to PICmicro 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 increasing flexibility and power.

13.2 MPASM Assembler

The MPASM assembler is a full-featured, universal macro assembler for all PICmicro 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 C17 and MPLAB C18 C Compilers

The MPLAB C17 and MPLAB C18 Code Development Systems are complete ANSI C compilers for Microchip's PIC17CXXX and PIC18CXXX family of microcontrollers. 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 C17 and MPLAB C18 C compilers. It can link relocatable objects from precompiled libraries, using directives from a linker script.

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

The object linker/library features include:

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

13.5 MPLAB C30 C Compiler

The MPLAB C30 C compiler is a full-featured, ANSI compliant, optimizing compiler that translates standard ANSI C programs into dsPIC30F assembly language source. The compiler also supports many command line options and language extensions to take full advantage of the dsPIC30F device hardware capabilities and afford fine control of the compiler code generator.

MPLAB C30 is distributed with a complete ANSI C standard library. All library functions have been validated and conform to the ANSI C library standard. The library includes functions for string manipulation, dynamic memory allocation, data conversion, time-keeping and math functions (trigonometric, exponential and hyperbolic). The compiler provides symbolic information for high-level source debugging with the MPLAB IDE.

13.6 MPLAB ASM30 Assembler, Linker and Librarian

MPLAB ASM30 assembler produces relocatable machine code from symbolic assembly language for dsPIC30F devices. MPLAB C30 compiler uses the assembler to produce it's 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

13.7 MPLAB SIM Software Simulator

The MPLAB SIM software simulator allows code development in a PC hosted environment by simulating the PICmicro series microcontrollers on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a file, or user defined key press, to any pin. The execution can be performed in Single-Step, Execute Until Break or Trace mode.

The MPLAB SIM simulator fully supports symbolic debugging using the MPLAB C17 and MPLAB C18 C Compilers, as well as the MPASM assembler. The software simulator offers the flexibility to develop and debug code outside of the laboratory environment, making it an excellent, economical software development tool.

13.8 MPLAB SIM30 Software Simulator

The MPLAB SIM30 software simulator allows code development in a PC hosted environment by simulating the dsPIC30F series microcontrollers on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a file, or user defined key press, to any of the pins.

The MPLAB SIM30 simulator fully supports symbolic debugging using the MPLAB C30 C Compiler and MPLAB ASM30 assembler. The simulator runs in either a Command Line mode for automated tasks, or from MPLAB IDE. This high-speed simulator is designed to debug, analyze and optimize time intensive DSP routines.

13.9 MPLAB ICE 2000 High-Performance Universal In-Circuit Emulator

The MPLAB ICE 2000 universal in-circuit emulator is intended to provide the product development engineer with a complete microcontroller design tool set for PICmicro 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 universal architecture of the MPLAB ICE in-circuit emulator allows expansion to support new PICmicro 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.

13.10 MPLAB ICE 4000 High-Performance Universal In-Circuit Emulator

The MPLAB ICE 4000 universal in-circuit emulator is intended to provide the product development engineer with a complete microcontroller design tool set for high-end PICmicro microcontrollers. Software control of the MPLAB ICE in-circuit emulator is provided by the MPLAB Integrated Development Environment, which allows editing, building, downloading and source debugging from a single environment.

The MPLAB ICD 4000 is a premium emulator system, providing the features of MPLAB ICE 2000, but with increased emulation memory and high-speed performance for dsPIC30F and PIC18XXXX devices. Its advanced emulator features include complex triggering and timing, up to 2 Mb of emulation memory and the ability to view variables in real-time.

The MPLAB ICE 4000 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.

13.11 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 PICmicro MCUs and can be used to develop for these and other PICmicro microcontrollers. 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, 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 PICmicro devices.

13.12 PRO MATE II Universal Device Programmer

The PRO MATE II is a universal, CE compliant device programmer with programmable voltage verification at V_{DDMIN} and V_{DDMAX} for maximum reliability. It features an LCD display for instructions and error messages and a modular detachable socket assembly to support various package types. In Stand-Alone mode, the PRO MATE II device programmer can read, verify and program PICmicro devices without a PC connection. It can also set code protection in this mode.

13.13 MPLAB PM3 Device Programmer

The MPLAB PM3 is a universal, CE compliant device programmer with programmable voltage verification at V_{DDMIN} and V_{DDMAX} 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 PICmicro devices without a PC connection. It can also set code protection in this mode. MPLAB PM3 connects to the host PC via an RS-232 or USB cable. 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.

13.14 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 PICmicro devices 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.

13.15 PICDEM 1 PICmicro Demonstration Board

The PICDEM 1 demonstration board demonstrates the capabilities of the PIC16C5X (PIC16C54 to PIC16C58A), PIC16C61, PIC16C62X, PIC16C71, PIC16C8X, PIC17C42, PIC17C43 and PIC17C44. All necessary hardware and software is included to run basic demo programs. The sample microcontrollers provided with the PICDEM 1 demonstration board can be programmed with a PRO MATE II device programmer or a PICSTART Plus development programmer. The PICDEM 1 demonstration board can be connected to the MPLAB ICE in-circuit emulator for testing. A prototype area extends the circuitry for additional application components. Features include an RS-232 interface, a potentiometer for simulated analog input, push button switches and eight LEDs.

13.16 PICDEM.net Internet/Ethernet Demonstration Board

The PICDEM.net demonstration board is an Internet/Ethernet demonstration board using the PIC18F452 microcontroller and TCP/IP firmware. The board supports any 40-pin DIP device that conforms to the standard pinout used by the PIC16F877 or PIC18C452. This kit features a user friendly TCP/IP stack, web server with HTML, a 24L256 Serial EEPROM for Xmodem download to web pages into Serial EEPROM, ICSP/MPLAB ICD 2 interface connector, an Ethernet interface, RS-232 interface and a 16 x 2 LCD display. Also included is the book and CD-ROM *"TCP/IP Lean, Web Servers for Embedded Systems,"* by Jeremy Bentham

13.17 PICDEM 2 Plus Demonstration Board

The PICDEM 2 Plus demonstration board supports many 18, 28 and 40-pin microcontrollers, including PIC16F87X and PIC18FXX2 devices. All the necessary hardware and software is included to run the demonstration programs. The sample microcontrollers provided with the PICDEM 2 demonstration board can be programmed with a PRO MATE II device programmer, PICSTART Plus development programmer, or MPLAB ICD 2 with a Universal Programmer Adapter. The MPLAB ICD 2 and MPLAB ICE in-circuit emulators may also be used with the PICDEM 2 demonstration board to test firmware. A prototype area extends the circuitry for additional application components. Some of the features include an RS-232 interface, a 2 x 16 LCD display, a piezo speaker, an on-board temperature sensor, four LEDs and sample PIC18F452 and PIC16F877 Flash microcontrollers.

13.18 PICDEM 3 PIC16C92X Demonstration Board

The PICDEM 3 demonstration board supports the PIC16C923 and PIC16C924 in the PLCC package. All the necessary hardware and software is included to run the demonstration programs.

13.19 PICDEM 4 8/14/18-Pin Demonstration Board

The PICDEM 4 can be used to demonstrate the capabilities of the 8, 14 and 18-pin PIC16XXXX and PIC18XXXX MCUs, including the PIC16F818/819, PIC16F87/88, PIC16F62XA and the PIC18F1320 family of microcontrollers. PICDEM 4 is intended to showcase the many features of these low pin count parts, including LIN and Motor Control using ECCP. Special provisions are made for low-power operation with the supercapacitor circuit and jumpers allow on-board hardware to be disabled to eliminate current draw in this mode. Included on the demo board are provisions for Crystal, RC or Canned Oscillator modes, a five volt regulator for use with a nine volt wall adapter or battery, DB-9 RS-232 interface, ICD connector for programming via ICSP and development with MPLAB ICD 2, 2 x 16 liquid crystal display, PCB footprints for H-Bridge motor driver, LIN transceiver and EEPROM. Also included are: header for expansion, eight LEDs, four potentiometers, three push buttons and a prototyping area. Included with the kit is a PIC16F627A and a PIC18F1320. Tutorial firmware is included along with the User's Guide.

13.20 PICDEM 17 Demonstration Board

The PICDEM 17 demonstration board is an evaluation board that demonstrates the capabilities of several Microchip microcontrollers, including PIC17C752, PIC17C756A, PIC17C762 and PIC17C766. A programmed sample is included. The PRO MATE II device programmer, or the PICSTART Plus development programmer, can be used to reprogram the device for user tailored application development. The PICDEM 17 demonstration board supports program download and execution from external on-board Flash memory. A generous prototype area is available for user hardware expansion.

13.21 PICDEM 18R PIC18C601/801 Demonstration Board

The PICDEM 18R demonstration board serves to assist development of the PIC18C601/801 family of Microchip microcontrollers. It provides hardware implementation of both 8-bit Multiplexed/Demultiplexed and 16-bit Memory modes. The board includes 2 Mb external Flash memory and 128 Kb SRAM memory, as well as serial EEPROM, allowing access to the wide range of memory types supported by the PIC18C601/801.

13.22 PICDEM LIN PIC16C43X Demonstration Board

The powerful LIN hardware and software kit includes a series of boards and three PICmicro microcontrollers. The small footprint PIC16C432 and PIC16C433 are used as slaves in the LIN communication and feature on-board LIN transceivers. A PIC16F874 Flash microcontroller serves as the master. All three microcontrollers are programmed with firmware to provide LIN bus communication.

13.23 PICkit™ 1 Flash Starter Kit

A complete “development system in a box”, the PICkit Flash Starter Kit includes a convenient multi-section board for programming, evaluation and development of 8/14-pin Flash PIC® microcontrollers. Powered via USB, the board operates under a simple Windows GUI. The PICkit 1 Starter Kit includes the User’s Guide (on CD ROM), PICkit 1 tutorial software and code for various applications. Also included are MPLAB® IDE (Integrated Development Environment) software, software and hardware “Tips 'n Tricks for 8-pin Flash PIC® Microcontrollers” Handbook and a USB interface cable. Supports all current 8/14-pin Flash PIC microcontrollers, as well as many future planned devices.

13.24 PICDEM USB PIC16C7X5 Demonstration Board

The PICDEM USB Demonstration Board shows off the capabilities of the PIC16C745 and PIC16C765 USB microcontrollers. This board provides the basis for future USB products.

13.25 Evaluation and Programming Tools

In addition to the PICDEM series of circuits, Microchip has a line of evaluation kits and demonstration software for these products.

- KEELOQ evaluation and programming tools for Microchip’s HCS Secure Data Products
- CAN developers kit for automotive network applications
- Analog design boards and filter design software
- PowerSmart battery charging evaluation/ calibration kits
- IrDA® development kit
- microID development and rfLab™ development software
- SEEVAL® designer kit for memory evaluation and endurance calculations
- PICDEM MSC demo boards for Switching mode power supply, high-power IR driver, delta sigma ADC and flow rate sensor

Check the Microchip web page and the latest Product Selector Guide for the complete list of demonstration and evaluation kits.

NOTES:

14.0 ELECTRICAL SPECIFICATIONS

Absolute Maximum Ratings(†)

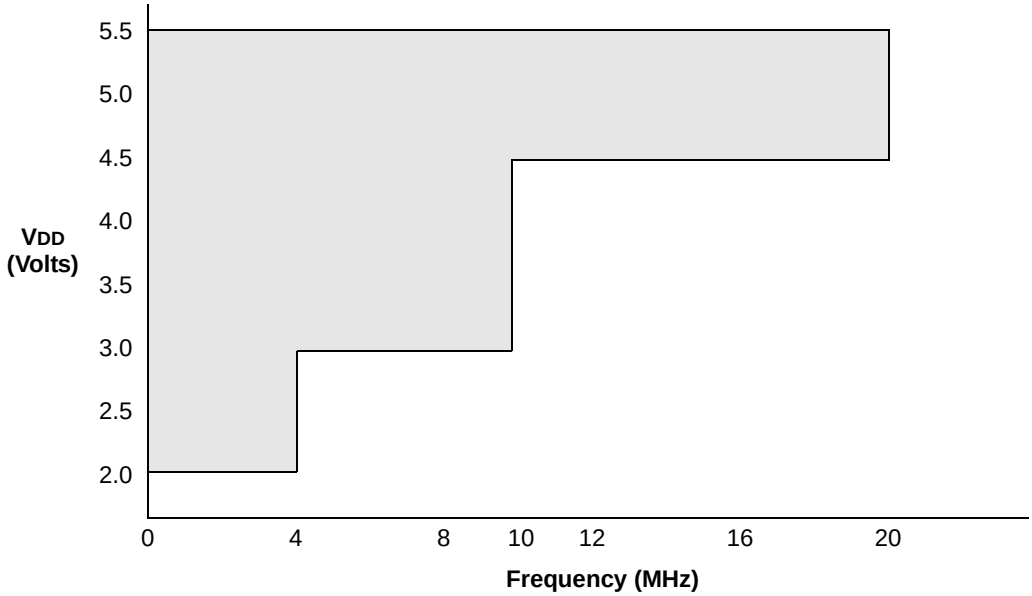
Ambient temperature under bias ..... -40° to +125°C
Storage temperature ..... -65°C to +150°C
Voltage on V_{DD} with respect to V_{SS} ..... -0.3V to +6.5V
Voltage on $\overline{MCLR}$ with respect to Vss ..... -0.3V to +13.5V
Voltage on all other pins with respect to V_{SS} ..... -0.3V to (V_{DD} + 0.3V)
Total power dissipation(1) ..... 800 mW
Maximum current out of V_{SS} pin ..... 300 mA
Maximum current into V_{DD} pin ..... 250 mA
Input clamp current, I_{IK} (V_I < 0 or V_I > V_{DD}) ..... ± 20 mA
Output clamp current, I_{OK} (Vo < 0 or Vo > V_{DD}) ..... ± 20 mA
Maximum output current sunk by any I/O pin ..... 25 mA
Maximum output current sourced by any I/O pin ..... 25 mA
Maximum current sunk by PORTA and PORTC (combined) ..... 200 mA
Maximum current sourced PORTA and PORTC (combined) ..... 200 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 to maximum rating conditions for extended periods may affect device reliability.

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

FIGURE 14-1: PIC16F688 VOLTAGE-FREQUENCY GRAPH, -40°C ≤ TA ≤ +125°C



Note 1: The shaded region indicates the permissible combinations of voltage and frequency.

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**
	VDD	**Supply Voltage**					
D001			2.0	—	5.5	V	FOSC < = 4 MHz:
D001C			3.0	—	5.5	V	FOSC < = 10 MHz
D001D			4.5	—	5.5	V	FOSC < = 20 MHz
D002	VDR	**RAM Data Retention Voltage(1)**	1.5*	—	—	V	Device in Sleep mode
D003	VPOR	**VDD Start Voltage** to ensure internal Power-on Reset signal	—	VSS	—	V	See **Section 11.4.2 "Power-On Reset (POR)"** for details.
D004	SVDD	**VDD Rise Rate** to ensure internal Power-on Reset signal	0.05*	—	—	V/ms	See **Section 11.4.2 "Power-On Reset (POR)"** for details.
D005	VBOD	**Brown-out Detect**	—	2.1	—	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: This is the limit to which VDD can be lowered in Sleep mode without losing RAM data.

14.2 DC Characteristics: PIC16F688-I (Industrial)

DC Characteristics		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial					
Param No.	Device Characteristics	Min	Typ†	Max	Units	Conditions	
						VDD	Note
D010	**Supply Current (IDD)(1, 2)**	—	9	TBD	µA	2.0	FOSC = 32 kHz LP Oscillator mode
		—	18	TBD	µA	3.0	
		—	35	TBD	µA	5.0	
D011		—	110	TBD	µA	2.0	FOSC = 1 MHz XT Oscillator mode
		—	190	TBD	µA	3.0	
		—	330	TBD	µA	5.0	
D012		—	220	TBD	µA	2.0	FOSC = 4 MHz XT Oscillator mode
		—	370	TBD	µA	3.0	
		—	0.6	TBD	mA	5.0	
D013		—	70	TBD	µA	2.0	FOSC = 1 MHz EC Oscillator mode
		—	140	TBD	µA	3.0	
		—	260	TBD	µA	5.0	
D014		—	180	TBD	µA	2.0	FOSC = 4 MHz EC Oscillator mode
		—	320	TBD	µA	3.0	
		—	580	TBD	µA	5.0	
D015		—	TBD	TBD	µA	2.0	FOSC = 31 kHz INTRC mode
		—	TBD	TBD	µA	3.0	
		—	TBD	TBD	mA	5.0	
D016		—	340	TBD	µA	2.0	FOSC = 4 MHz INTOSC mode
		—	500	TBD	µA	3.0	
		—	0.8	TBD	mA	5.0	
D017		—	180	TBD	µA	2.0	FOSC = 4 MHz EXTRC mode
		—	320	TBD	µA	3.0	
		—	580	TBD	µA	5.0	
D018		—	2.1	TBD	mA	4.5	FOSC = 20 MHz HS Oscillator mode
		—	2.4	TBD	mA	5.0	

Legend: TBD = To Be Determined

† 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; $\overline{MCLR}$ = VDD; WDT disabled.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors, such as I/O pin loading and switching rate, oscillator type, internal code execution pattern and temperature, also have an impact on the current consumption.

14.3 DC Characteristics: PIC16F688-I (Industrial)

DC Characteristics		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial					
Param No.	Device Characteristics	Min	Typ†	Max	Units	Conditions	
						VDD	Note
D020	**Power-down Base Current (IPD)(4)**	—	0.99	TBD	nA	2.0	WDT, BOD, Comparators, VREF and T1OSC disabled
		—	1.2	TBD	nA	3.0	
		—	2.9	TBD	nA	5.0	
D021		—	0.3	TBD	µA	2.0	WDT Current
		—	1.8	TBD	µA	3.0	
		—	8.4	TBD	µA	5.0	
D022		—	58	TBD	µA	3.0	BOD Current
		—	109	TBD	µA	5.0	
D023		—	3.3	TBD	µA	2.0	Comparator Current(3)
		—	6.1	TBD	µA	3.0	
		—	11.5	TBD	µA	5.0	
D024		—	58	TBD	µA	2.0	CVREF Current
		—	85	TBD	µA	3.0	
		—	138	TBD	µA	5.0	
D025		—	4.0	TBD	µA	2.0	T1OSC Current
		—	4.6	TBD	µA	3.0	
		—	6.0	TBD	µA	5.0	
D026		—	1.2	TBD	nA	3.0	A/D Current
		—	0.0022	TBD	µA	5.0	

Legend: TBD = To Be Determined

† 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 3: The peripheral current is the sum of the base IDD or IPD and the additional current consumed when this peripheral is enabled. The peripheral Δ current can be determined by subtracting the base IDD or IPD current from this limit. Max values should be used when calculating total current consumption.

4: 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 hi-impedance state and tied to VDD.

14.4 DC Characteristics: PIC16F688-E (Extended)

DC Characteristics		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +125°C for extended					
Param No.	Device Characteristics	Min	Typ†	Max	Units	Conditions	
						VDD	Note
D010E	**Supply Current (IDD)**	—	9	TBD	µA	2.0	FOSC = 32 kHz LP Oscillator mode
		—	18	TBD	µA	3.0	
		—	35	TBD	µA	5.0	
D011E		—	110	TBD	µA	2.0	FOSC = 1 MHz XT Oscillator mode
		—	190	TBD	µA	3.0	
		—	330	TBD	µA	5.0	
D012E		—	220	TBD	µA	2.0	FOSC = 4 MHz XT Oscillator mode
		—	370	TBD	µA	3.0	
		—	0.6	TBD	mA	5.0	
D013E		—	70	TBD	µA	2.0	FOSC = 1 MHz EC Oscillator mode
		—	140	TBD	µA	3.0	
		—	260	TBD	µA	5.0	
D014E		—	180	TBD	µA	2.0	FOSC = 4 MHz EC Oscillator mode
		—	320	TBD	µA	3.0	
		—	580	TBD	µA	5.0	
D015E		—	TBD	TBD	µA	2.0	FOSC = 31 kHz INTRC mode
		—	TBD	TBD	µA	3.0	
		—	TBD	TBD	mA	5.0	
D016E		—	340	TBD	µA	2.0	FOSC = 4 MHz INTOSC mode
		—	500	TBD	µA	3.0	
		—	0.8	TBD	mA	5.0	
D017E		—	180	TBD	µA	2.0	FOSC = 4 MHz EXTRC mode
		—	320	TBD	µA	3.0	
		—	580	TBD	µA	5.0	
D018E		—	2.1	TBD	mA	4.5	FOSC = 20 MHz HS Oscillator mode
		—	2.4	TBD	mA	5.0	

Legend: TBD = To Be Determined

† 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; $\overline{MCLR}$ = VDD; WDT disabled.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors, such as I/O pin loading and switching rate, oscillator type, internal code execution pattern and temperature, also have an impact on the current consumption.

14.5 DC Characteristics: PIC16F688-E (Extended)

DC Characteristics		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +125°C for extended					
Param No.	Device Characteristics	Min	Typ†	Max	Units	Conditions	
						VDD	Note
D020E	**Power-down Base Current (IPD)(4)**	—	0.00099	TBD	µA	2.0	WDT, BOD, Comparators, VREF and T1OSC disabled
		—	0.0012	TBD	µA	3.0	
		—	0.0029	TBD	µA	5.0	
D021E		—	0.3	TBD	µA	2.0	WDT Current
		—	1.8	TBD	µA	3.0	
		—	8.4	TBD	µA	5.0	
D022E		—	58	TBD	µA	3.0	BOD Current
		—	109	TBD	µA	5.0	
D023E		—	3.3	TBD	µA	2.0	Comparator Current(3)
		—	6.1	TBD	µA	3.0	
		—	11.5	TBD	µA	5.0	
D024E		—	58	TBD	µA	2.0	CVREF Current
		—	85	TBD	µA	3.0	
		—	138	TBD	µA	5.0	
D025E		—	4.0	TBD	µA	2.0	T1OSC Current
		—	4.6	TBD	µA	3.0	
		—	6.0	TBD	µA	5.0	
D026E		—	0.0012	TBD	µA	3.0	A/D Current(3)
		—	0.0022	TBD	µA	5.0	

Legend: TBD = To Be Determined

† 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 3: The peripheral current is the sum of the base IDD or IPD and the additional current consumed when this peripheral is enabled. The peripheral Δ current can be determined by subtracting the base IDD or IPD current from this limit. Max values should be used when calculating total current consumption.

4: 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.

14.6 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**
	VIL	**Input Low Voltage**					
		I/O port:					
D030		with TTL buffer	Vss	—	0.8	V	4.5V ≤ VDD ≤ 5.5V
D030A			Vss	—	0.15 VDD	V	Otherwise
D031		with Schmitt Trigger buffer	Vss	—	0.2 VDD	V	Entire range
D032		$\overline{MCLR}$, OSC1 (RC mode)	Vss	—	0.2 VDD	V	
D033		OSC1 (XT and LP modes)(1)	Vss	—	0.3	V	
D033A		OSC1 (HS mode)(1)	Vss	—	0.3 VDD	V	
	VIH	**Input High Voltage**					
		I/O port:		—			
D040		with TTL buffer	2.0	—	VDD	V	4.5V ≤ VDD ≤ 5.5V
D040A			(0.25 VDD + 0.8)	—	VDD	V	Otherwise
D041		with Schmitt Trigger buffer	0.8 VDD	—	VDD	V	Entire range
D042		$\overline{MCLR}$	0.8 VDD	—	VDD	V	
D043		OSC1 (XT and LP modes)	1.6	—	VDD	V	**(Note 1)**
D043A		OSC1 (HS mode)	0.7 VDD	—	VDD	V	**(Note 1)**
D043B		OSC1 (RC mode)	0.9 VDD	—	VDD	V	
D070	IPUR	**PORTA Weak Pull-up Current**	50*	250	400*	µA	VDD = 5.0V, VPIN = VSS
	IIL	**Input Leakage Current(2)**					
D060		I/O port	—	± 0.1	± 1	µA	VSS ≤ VPIN ≤ VDD, Pin at high-impedance
D061		$\overline{MCLR}$(3)	—	± 0.1	± 5	µA	VSS ≤ VPIN ≤ VDD
D063		OSC1	—	± 0.1	± 5	µA	VSS ≤ VPIN ≤ VDD, XT, HS and LP osc configuration
	VOL	**Output Low Voltage**					
D080		I/O port	—	—	0.6	V	IOL = 8.5 mA, VDD = 4.5V (Ind.)
D083		OSC2/CLKOUT (RC mode)	—	—	0.6	V	IOL = 1.6 mA, VDD = 4.5V (Ind.) IOL = 1.2 mA, VDD = 4.5V (Ext.)
	VOH	**Output High Voltage**					
D090		I/O port	VDD - 0.7	—	—	V	IOH = -3.0 mA, VDD = 4.5V (Ind.)
D092		OSC2/CLKOUT (RC mode)	VDD - 0.7	—	—	V	IOH = -1.3 mA, VDD = 4.5V (Ind.) IOH = -1.0 mA, VDD = 4.5V (Ext.)

* 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 $\overline{MCLR}$ pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

14.7 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**
D100	IULP	**Ultra Low-Power Wake-up Current**	—	200	—	nA	
		Capacitive Loading Specs on Output Pins					
D100	COSC2	OSC2 pin	—	—	15*	pF	In XT, HS and LP modes when external clock is used to drive OSC1
D101	CIO	All I/O pins	—	—	50*	pF	
		Data EEPROM Memory					
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(4)	1M	10M	—	E/W	-40°C ≤ TA ≤ +85°C
		Program Flash Memory					
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 4: See **Section 9.0 "Data EEPROM And Flash Program Memory Control"** for additional information.

14.8 Timing Parameter Symbology

The timing parameter symbols have been created with one of the following formats:

1. TppS2ppS
2. TppS

T			
F	Frequency	T	Time

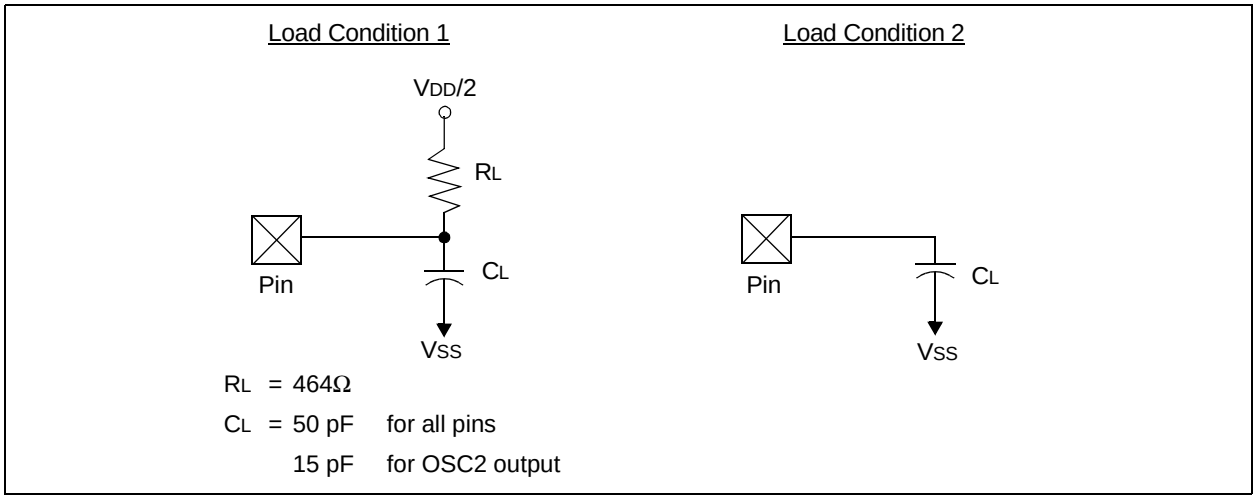
Lowercase letters (pp) and their meanings:

pp			
cc	RC	osc	OSC1
ck	CLKOUT	rd	$\overline{RD}$
cs	$\overline{CS}$	rw	$\overline{RD}$ or $\overline{WR}$
di	SDI	sc	SCK
do	SDO	ss	$\overline{SS}$
dt	Data in	t0	T0CKI
io	I/O port	t1	T1CKI
mc	$\overline{MCLR}$	wr	$\overline{WR}$

Uppercase letters and their meanings:

S			
F	Fall	P	Period
H	High	R	Rise
I	Invalid (High-impedance)	V	Valid
L	Low	Z	High-impedance

FIGURE 14-2: LOAD CONDITIONS



14.9 AC Characteristics: PIC16F688 (Industrial, Extended)

FIGURE 14-3: EXTERNAL CLOCK TIMING

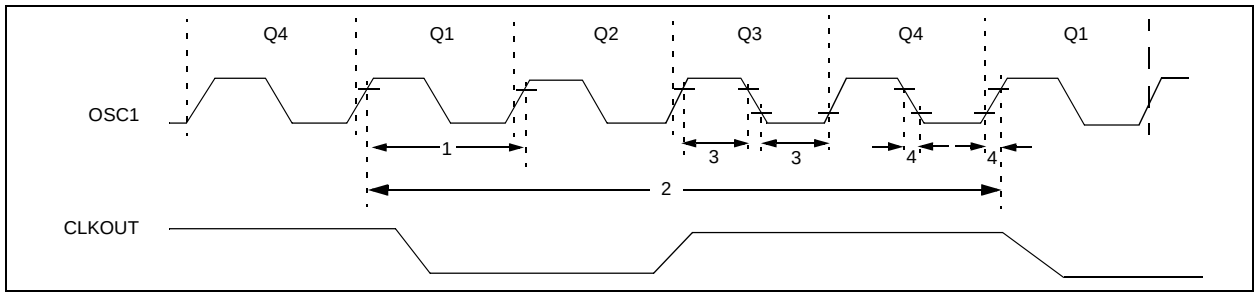


TABLE 14-1: EXTERNAL CLOCK TIMING REQUIREMENTS

Standard Operating Conditions (unless otherwise stated)
Operating Temperature -40°C ≤ TA ≤ +125°C

Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
	FOSC	External CLKIN Frequency(1)	DC	—	37	kHz	LP Oscillator mode
			DC	—	4	MHz	XT Oscillator mode
			DC	—	20	MHz	HS Oscillator mode
			DC	—	20	MHz	EC Oscillator mode
		Oscillator Frequency(1)	5	—	37	kHz	LP Oscillator mode
			—	4	—	MHz	INTOSC mode
			DC	—	4	MHz	RC Oscillator mode
			0.1	—	4	MHz	XT Oscillator mode
			1	—	20	MHz	HS Oscillator mode
1	TOSC	External CLKIN Period(1)	27	—	∞	µs	LP Oscillator mode
			50	—	∞	ns	HS Oscillator mode
			50	—	∞	ns	EC Oscillator mode
			250	—	∞	ns	XT Oscillator mode
		Oscillator Period(1)	27		200	µs	LP Oscillator mode
			—	250	—	ns	INTOSC mode
			250	—	—	ns	RC Oscillator mode
			250	—	10,000	ns	XT Oscillator mode
			50	—	1,000	ns	HS Oscillator mode
2	TCY	Instruction Cycle Time(1)	200	TCY	DC	ns	TCY = 4/FOSC
3	TosL, TosH	External CLKIN (OSC1) High External CLKIN Low	2*	—	—	µs	LP oscillator, TOSC L/H duty cycle
			20*	—	—	ns	HS oscillator, TOSC L/H duty cycle
			100 *	—	—	ns	XT oscillator, TOSC L/H duty cycle
4	TosR, TosF	External CLKIN Rise External CLKIN Fall	—	—	50*	ns	LP oscillator
			—	—	25*	ns	XT oscillator
			—	—	15*	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.

TABLE 14-2: PRECISION INTERNAL OSCILLATOR PARAMETERS

Standard Operating Conditions (unless otherwise stated) Operating Temperature-40°C ≤ TA ≤ +125°C								
Param No.	**Sym**	**Characteristic**	**Freq Tolerance**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
F10	FOSC	Internal Calibrated INTOSC Frequency(1)	±1%	—	8.00	TBD	MHz	VDD and Temperature TBD
			±2%	—	8.00	TBD	MHz	2.5V ≤ VDD ≤ 5.5V 0°C ≤ TA ≤ +85°C
			±5%	—	8.00	TBD	MHz	2.0V ≤ VDD ≤ 5.5V -40°C ≤ TA ≤ +85°C (Ind.) -40°C ≤ TA ≤ +125°C (Ext.)
F14	TIOSC ST	Oscillator Wake-up from Sleep Start-up Time*	—	—	TBD	TBD	µs	VDD = 2.0V, -40°C to +85°C
			—	—	TBD	TBD	µs	VDD = 3.0V, -40°C to +85°C
			—	—	TBD	TBD	µs	VDD = 5.0V, -40°C to +85°C

Legend: TBD = To Be Determined

* 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: To ensure these oscillator frequency tolerances, VDD and VSS must be capacitively decoupled as close to the device as possible. 0.1 uF and 0.01 uF values in parallel are recommended.

FIGURE 14-4: CLKOUT AND I/O TIMING

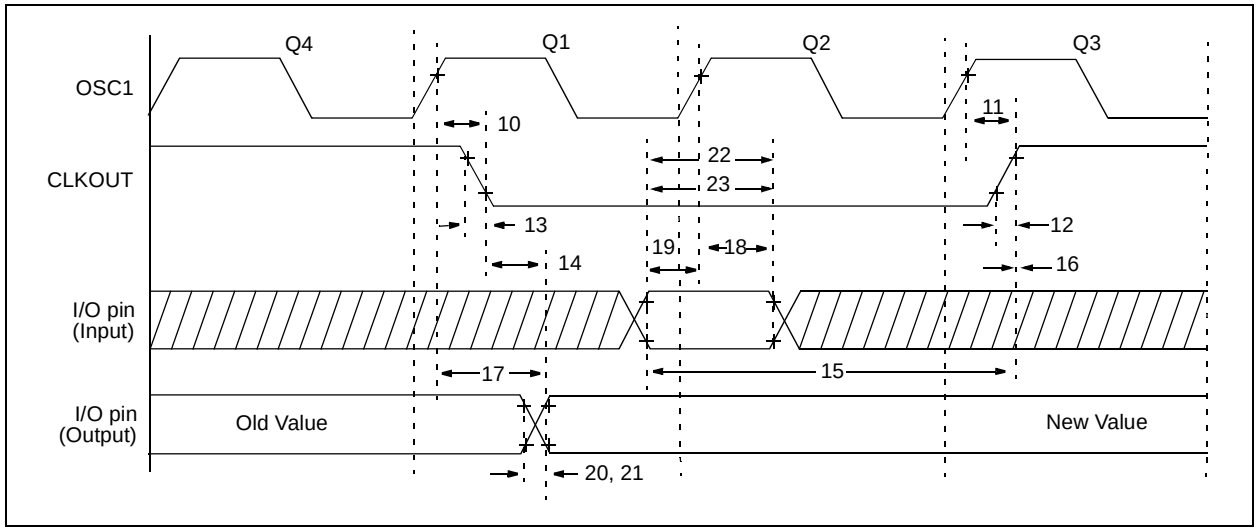


TABLE 14-3: CLKOUT AND I/O TIMING REQUIREMENTS

Standard Operating Conditions (unless otherwise stated)
Operating Temperature-40°C ≤ TA ≤ +125°C

Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
10	TosH2ckL	OSC1↑ to CLOUT↓	—	75	200	ns	**(Note 1)**
11	TosH2ckH	OSC1↑ to CLOUT↑	—	75	200	ns	**(Note 1)**
12	TckR	CLKOUT Rise Time	—	35	100	ns	**(Note 1)**
13	TckF	CLKOUT Fall Time	—	35	100	ns	**(Note 1)**
14	TckL2ioV	CLKOUT↓ to Port Out Valid	—	—	20	ns	**(Note 1)**
15	TioV2ckH	Port In Valid before CLKOUT↑	TOSC + 200 ns	—	—	ns	**(Note 1)**
16	TckH2ioI	Port In Hold after CLKOUT↑	0	—	—	ns	**(Note 1)**
17	TosH2ioV	OSC1↑ (Q1 cycle) to Port Out Valid	—	50	150*	ns	
			—	—	300	ns	
18	TosH2ioI	OSC1↑ (Q2 cycle) to Port Input Invalid (I/O in hold time)	100	—	—	ns	
19	TioV2osH	Port Input Valid to OSC1↑ (I/O in setup time)	0	—	—	ns	
20	TioR	Port Output Rise Time	—	10	40	ns	
21	TioF	Port Output Fall Time	—	10	40	ns	
22	Tinp	INT Pin High or Low Time	25	—	—	ns	
23	Trbp	PORTA change INT high or low 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.

FIGURE 14-5: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING

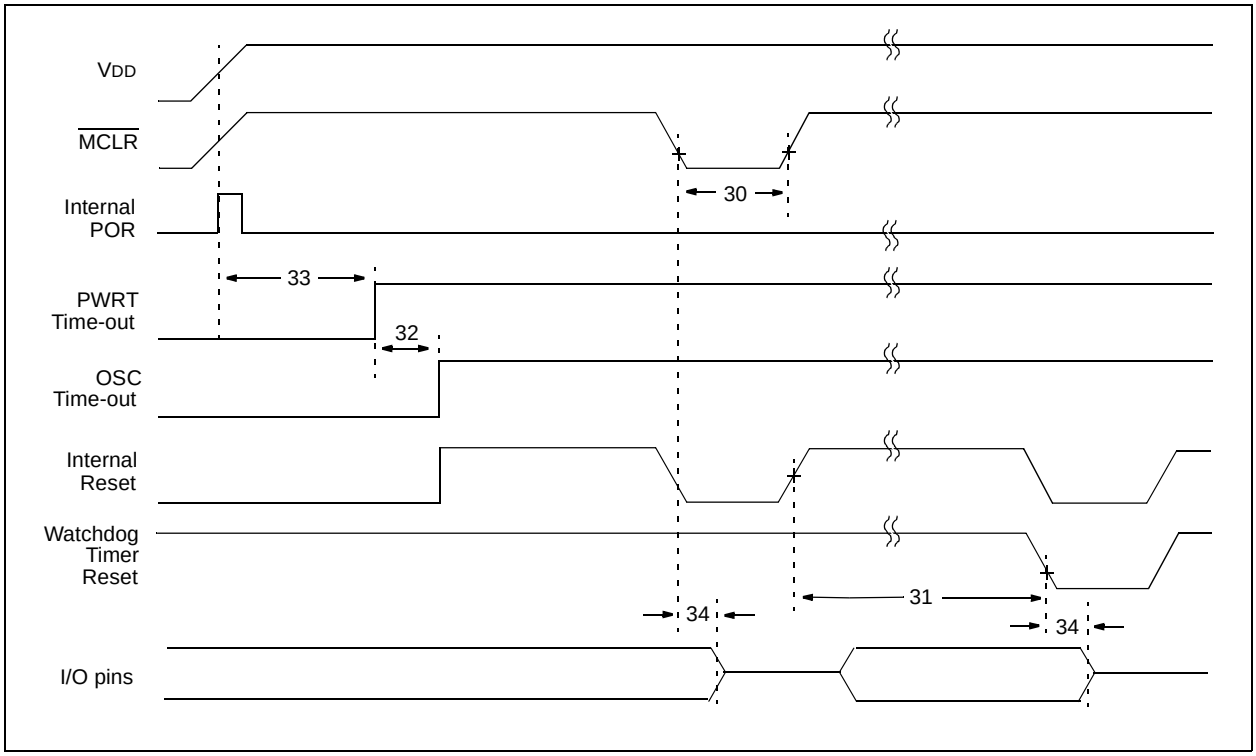
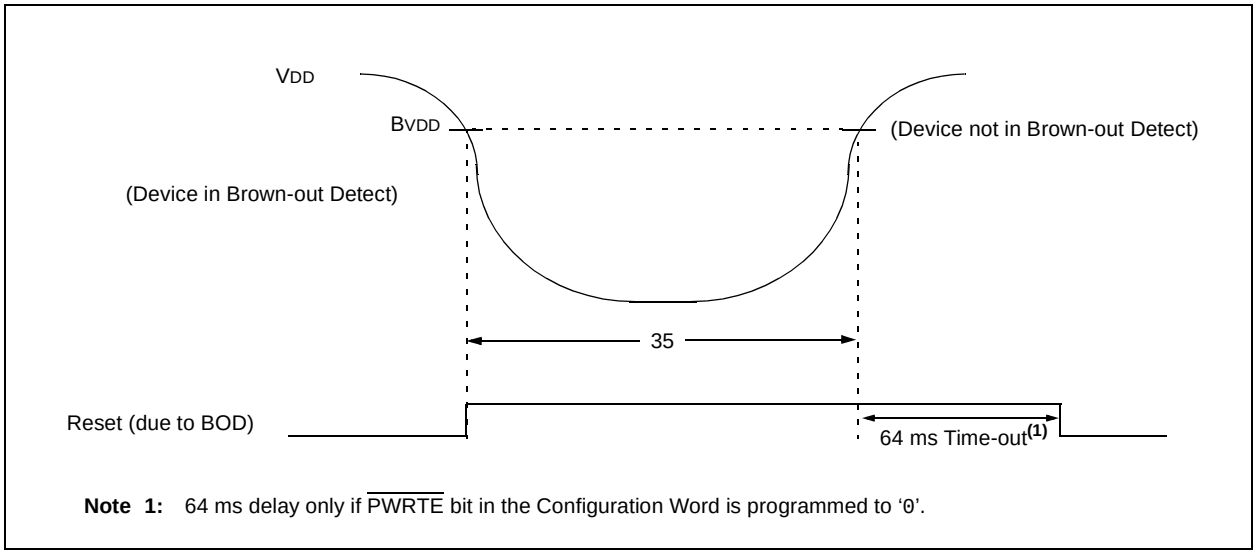


FIGURE 14-6: BROWN-OUT DETECT TIMING AND CHARACTERISTICS



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

TABLE 14-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER, POWER-UP TIMER AND BROWN-OUT DETECT REQUIREMENTS

Standard Operating Conditions (unless otherwise stated) Operating Temperature-40°C ≤ TA ≤ +125°C							
Param No.	**Sym**	**Characteristic**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
30	TMCL	$\overline{\text{MCLR}}$ Pulse Width (low)	2 11	— 18	— 24	µs ms	VDD = 5V, -40°C to +85°C Extended temperature
31	TWDT	Watchdog Timer Time-out Period (No Prescaler)	10 10	17 17	25 30	ms ms	VDD = 5V, -40°C to +85°C Extended temperature
32	TOST	Oscillation Start-up Timer Period	—	1024TOSC	—	—	TOSC = OSC1 period
33*	TPWRT	Power-up Timer Period	28* TBD	64 TBD	132* TBD	ms ms	VDD = 5V, -40°C to +85°C Extended Temperature
34	TIOZ	I/O High-impedance from $\overline{\text{MCLR}}$ Low or Watchdog Timer Reset	—	—	2.0	µs	
	BVDD	Brown-out Detect Voltage	2.025	—	2.175	V	
35	TBOD	Brown-out Detect Pulse Width	100*	—	—	µs	VDD ≤ BVDD (D005)

Legend: TBD = To Be Determined

* 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.

FIGURE 14-7: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS

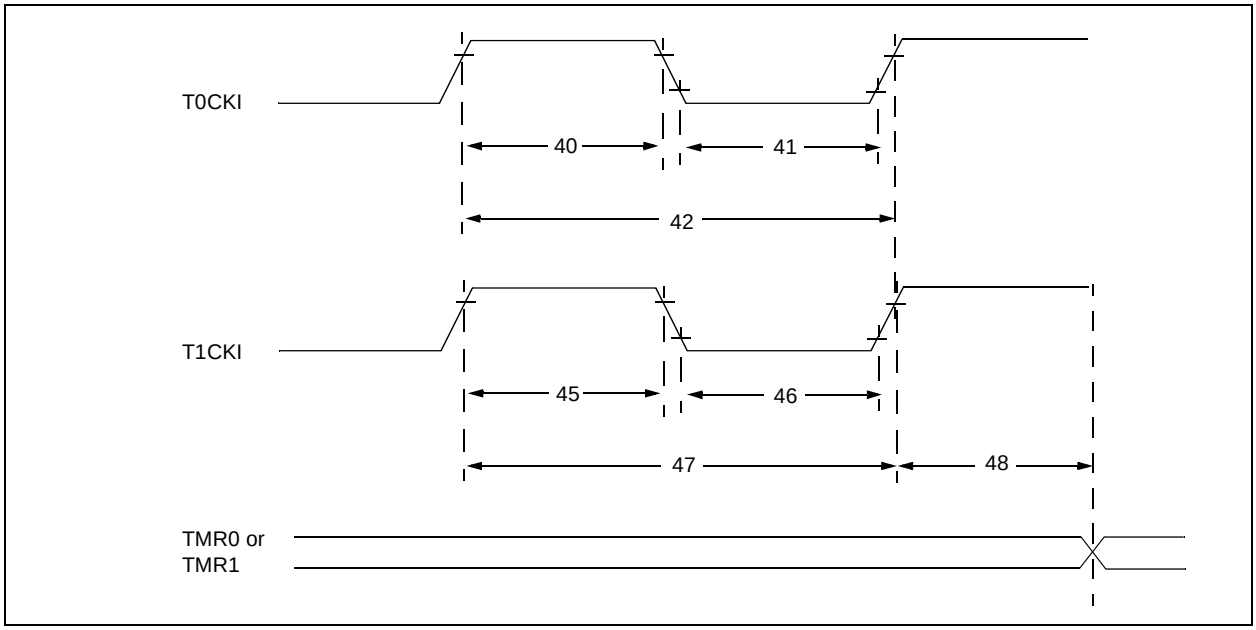


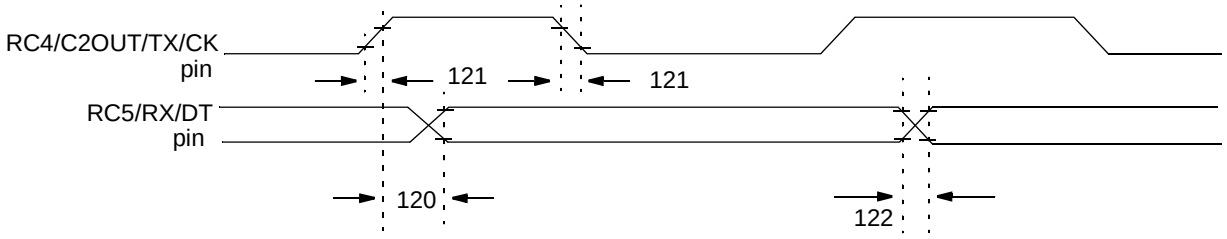
TABLE 14-5: TIMER0 AND TIMER1 EXTERNAL CLOCK REQUIREMENTS

Param No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
40*	Tt0H	T0CKI High Pulse Width	No Prescaler	0.5 TCY + 20	—	—	ns	
			With Prescaler	10	—	—	ns	
41*	Tt0L	T0CKI Low Pulse Width	No Prescaler	0.5 TCY + 20	—	—	ns	
			With Prescaler	10	—	—	ns	
42*	Tt0P	T0CKI Period		Greater of: 20 or TCY + 40 N	—	—	ns	N = prescale value (2, 4, ..., 256)
45*	Tt1H	T1CKI High Time	Synchronous, No Prescaler	0.5 TCY + 20	—	—	ns	
			Synchronous, with Prescaler	15	—	—	ns	
			Asynchronous	30	—	—	ns	
46*	Tt1L	T1CKI Low Time	Synchronous, No Prescaler	0.5 TCY + 20	—	—	ns	
			Synchronous, with Prescaler	15	—	—	ns	
			Asynchronous	30	—	—	ns	
47*	Tt1P	T1CKI Input Period	Synchronous	Greater of: 30 or TCY + 40 N	—	—	ns	N = prescale value (1, 2, 4, 8)
			Asynchronous	60	—	—	ns	
	Ft1	Timer1 oscillator input frequency range (oscillator enabled by setting bit T1OSCEN)		DC	—	200*	kHz	
48	TCKEZtmr1	Delay from external clock edge to timer increment		2 TOSC*	—	7 TOSC*	—	

* 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.

FIGURE 14-8: USART SYNCHRONOUS TRANSMISSION (MASTER/SLAVE) TIMING



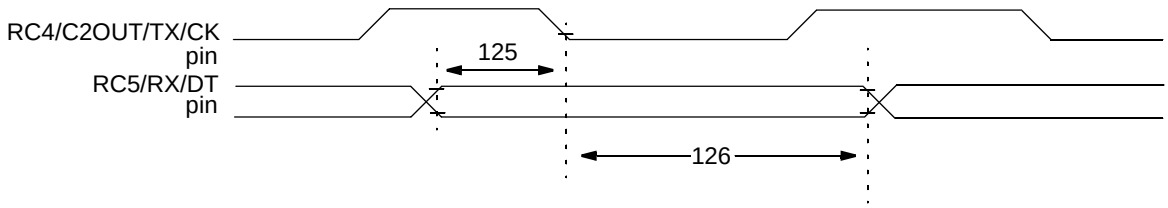
Note: Refer to Figure 14-2 for load conditions.

TABLE 14-6: USART SYNCHRONOUS TRANSMISSION REQUIREMENTS

Standard Operating Conditions (unless otherwise stated)
Operating Temperature-40°C ≤ TA ≤ +125°C

Param. No.	Symbol	Characteristic		Min	Max	Units	Conditions
120	TckH2dtV	SYNC XMIT (Master & Slave) Clock high to data-out valid	PIC16F688	—	40	ns	
			PIC16LF688	—	100	ns	
121	Tckrf	Clock out rise time and fall time (Master mode)	PIC16F688	—	20	ns	
			PIC16LF688	—	50	ns	
122	Tdtrf	Data-out rise time and fall time	PIC16F688	—	20	ns	
			PIC16LF688	—	50	ns	

FIGURE 14-9: USART SYNCHRONOUS RECEIVE (MASTER/SLAVE) TIMING



Note: Refer to Figure 14-2 for load conditions.

TABLE 14-7: USART SYNCHRONOUS RECEIVE REQUIREMENTS

Standard Operating Conditions (unless otherwise stated)
Operating Temperature-40°C ≤ TA ≤ +125°C

Param. No.	Symbol	Characteristic	Min	Max	Units	Conditions
125	TdtV2ckl	SYNC RCV (Master & Slave) Data-hold before CK ↓ (DT hold time)	10	—	ns	
126	TckL2dtl	Data-hold after CK ↓ (DT hold time)	15	—	ns	

TABLE 14-8: COMPARATOR SPECIFICATIONS

Comparator Specifications		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +125°C				
Sym	**Characteristics**	**Min**	**Typ**	**Max**	**Units**	**Comments**
VOS	Input Offset Voltage	—	± 5.0	± 10	mV	
VCM	Input Common Mode Voltage	0	—	VDD - 1.5	V	
CMRR	Common Mode Rejection Ratio	+55*	—	—	db	
TRT	Response Time(1)	—	150	400*	ns	
TMC2COV	Comparator Mode Change to Output Valid	—	—	10*	µs	

* These parameters are characterized but not tested.

Note 1: Response time measured with one comparator input at (VDD - 1.5)/2 while the other input transitions from VSS to VDD – 1.5V.

TABLE 14-9: COMPARATOR VOLTAGE REFERENCE SPECIFICATIONS

Voltage Reference Specifications		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +125°C				
Sym.	**Characteristics**	**Min**	**Typ**	**Max**	**Units**	**Comments**
	Resolution	— —	VDD/24* VDD/32	— —	LSb LSb	Low Range (VRR = 1) High Range (VRR = 0)
	Absolute Accuracy	— —	— —	± 1/4* ± 1/2*	LSb LSb	Low Range (VRR = 1) High Range (VRR = 0)
	Unit Resistor Value (R)	—	2K*	—	Ω	
	Settling Time(1)	—	—	10*	µs	

* These parameters are characterized but not tested.

Note 1: Settling time measured while VRR = 1 and VR<3:0> transitions from '0000' to '1111'.

TABLE 14-10: PIC16F688 A/D CONVERTER CHARACTERISTICS:

Standard Operating Conditions (unless otherwise stated) Operating Temperature-40°C ≤ TA ≤ +125°C							
Param No.	**Sym**	**Characteristic**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
A01	NR	Resolution	—	—	10 bits	bit	
A02	EABS	Total Absolute Error*(1)	—	—	±1	LSb	VREF = 5.0V
A03	EIL	Integral Error	—	—	±1	LSb	VREF = 5.0V
A04	EDL	Differential Error	—	—	±1	LSb	No missing codes to 10 bits VREF = 5.0V
A05	EFS	Full-scale Range	2.2*	—	5.5*	V	
A06	EOFF	Offset Error	—	—	±1	LSb	VREF = 5.0V
A07	EGN	Gain Error	—	—	±1	LSb	VREF = 5.0V
A10	—	Monotonicity	—	guaranteed(2)	—	—	VSS ≤ VAIN ≤ VREF+
A20 A20A	VREF	Reference Voltage	2.2 2.5	—	— VDD + 0.3	V	Absolute minimum to ensure 10-bit accuracy
A25	VAIN	Analog Input Voltage	VSS	—	VREF	V	
A30	ZAIN	Recommended Impedance of Analog Voltage Source	—	—	10	kΩ	
A50	IREF	VREF Input Current*(3)	10	—	1000	μA	During VAIN acquisition. Based on differential of VHOLD to VAIN.
			—	—	10	μA	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.

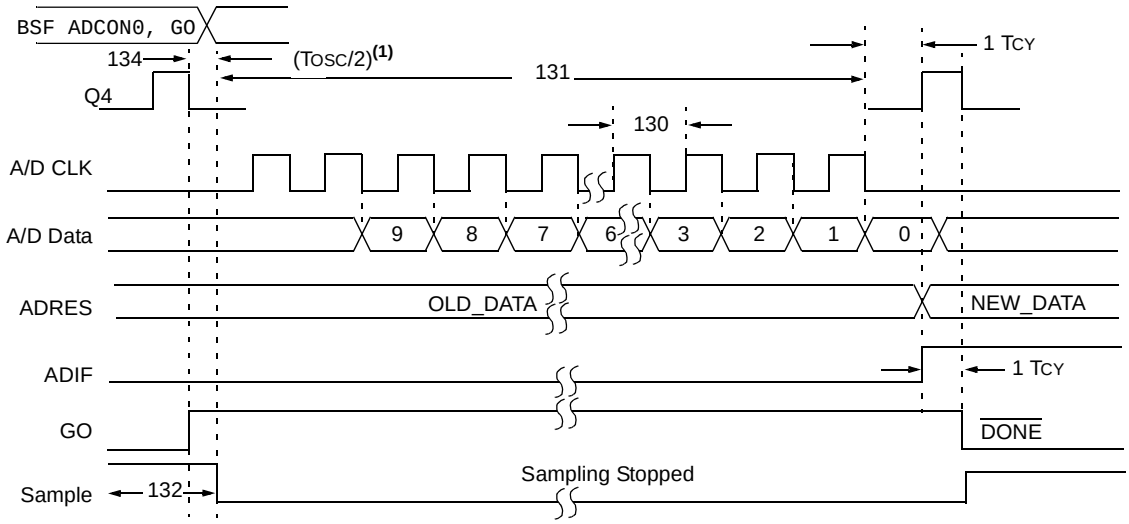
Note 1: Total Absolute Error includes integral, differential, offset and gain errors.

2: The A/D conversion result never decreases with an increase in the input voltage and has no missing codes.

3: VREF current is from external VREF or VDD pin, whichever is selected as reference input.

4: When A/D is off, it will not consume any current other than leakage current. The power-down current _specification includes any such leakage from the A/D module.

FIGURE 14-10: PIC16F688 A/D CONVERSION TIMING (NORMAL MODE)



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

TABLE 14-11: PIC16F688 A/D CONVERSION REQUIREMENTS

Standard Operating Conditions (unless otherwise stated) Operating Temperature-40°C ≤ TA ≤ +125°C							
Param No.	**Sym**	**Characteristic**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
130	TAD	A/D Clock Period	1.6	—	—	µs	TOSC-based, VREF ≥ 3.0V
			3.0*	—	—	µs	TOSC-based, VREF full range
130	TAD	A/D Internal RC Oscillator Period					ADCS<1:0> = 11 (RC mode)
			3.0*	6.0	9.0*	µs	At VDD = 2.5V
			2.0*	4.0	6.0*	µs	At VDD = 5.0V
131	TCNV	Conversion Time (not including Acquisition Time)(1)	—	11	—	TAD	Set GO bit to new data in A/D Result register
132	TACQ	Acquisition Time		11.5	—	µs	
			5*	—	—	µs	The minimum time is the amplifier settling time. This may be used if the "new" input voltage has not changed by more than 1 LSb (i.e., 4.1 mV @ 4.096V) from the last sampled voltage (as stored on CHOLD).
134	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.

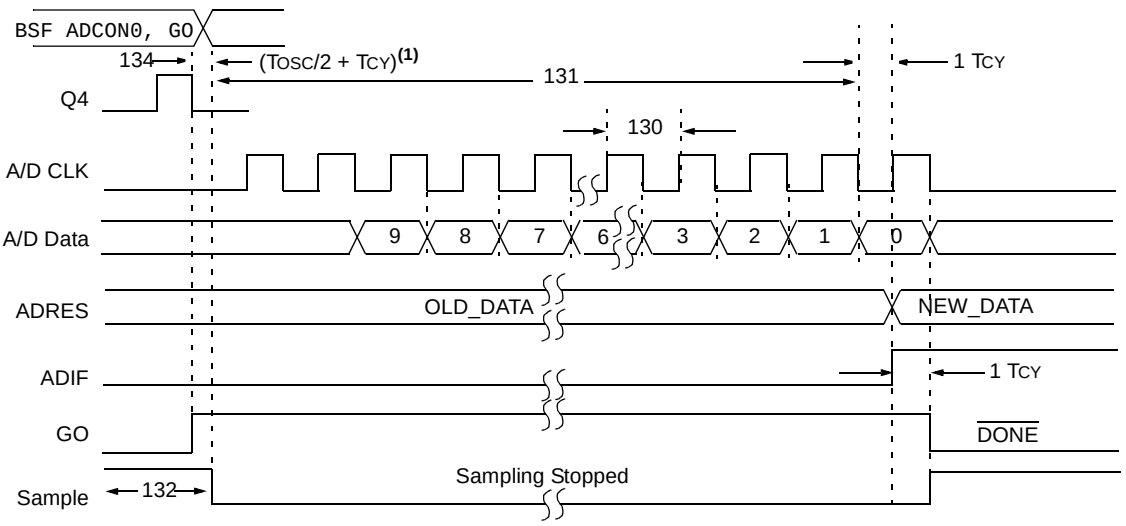
* These parameters are characterized but not tested.

† 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 Table 8-1 for minimum conditions.

FIGURE 14-11: PIC16F688 A/D CONVERSION TIMING (SLEEP MODE)



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

TABLE 14-12: PIC16F688 A/D CONVERSION REQUIREMENTS (SLEEP MODE)

Standard Operating Conditions (unless otherwise stated) Operating Temperature-40°C ≤ TA ≤ +125°C							
Param No.	**Sym**	**Characteristic**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
130	TAD	A/D Internal RC Oscillator Period	3.0* 2.0*	6.0 4.0	9.0* 6.0*	µs µs	ADCS<1:0> = 11 (RC mode) At VDD = 2.5V At VDD = 5.0V
131	Tcnv	Conversion Time (not including Acquisition Time)(1)	—	11	—	TAD	
132	TACQ	Acquisition Time	**(2)** 5*	11.5 —	— —	µs µs	The minimum time is the amplifier settling time. This may be used if the "new" input voltage has not changed by more than 1 LSb (i.e., 4.1 mV @ 4.096V) from the last sampled voltage (as stored on CHOLD).
134	TGO	Q4 to A/D Clock Start	—	TOSC/2 + TCY	—	—	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.

* 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: ADRES register may be read on the following TCY cycle.

2: See Table 8-1 for minimum conditions.

NOTES:

15.0 DC AND AC CHARACTERISTICS GRAPHS AND TABLES

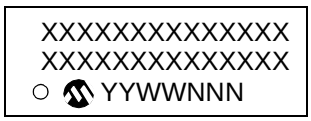
Graphs are not available at this time.

NOTES:

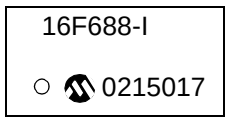
16.0 PACKAGING INFORMATION

16.1 Package Marking Information

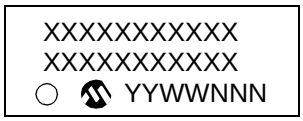
14-Lead PDIP (Skinny DIP)



Example

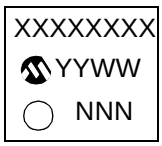


14-Lead SOIC

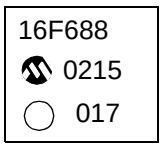


Example

14-Lead TSSOP



Example



Legend:	XX...X	Customer specific information*
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week ‘01’)
	NNN	Alphanumeric traceability code

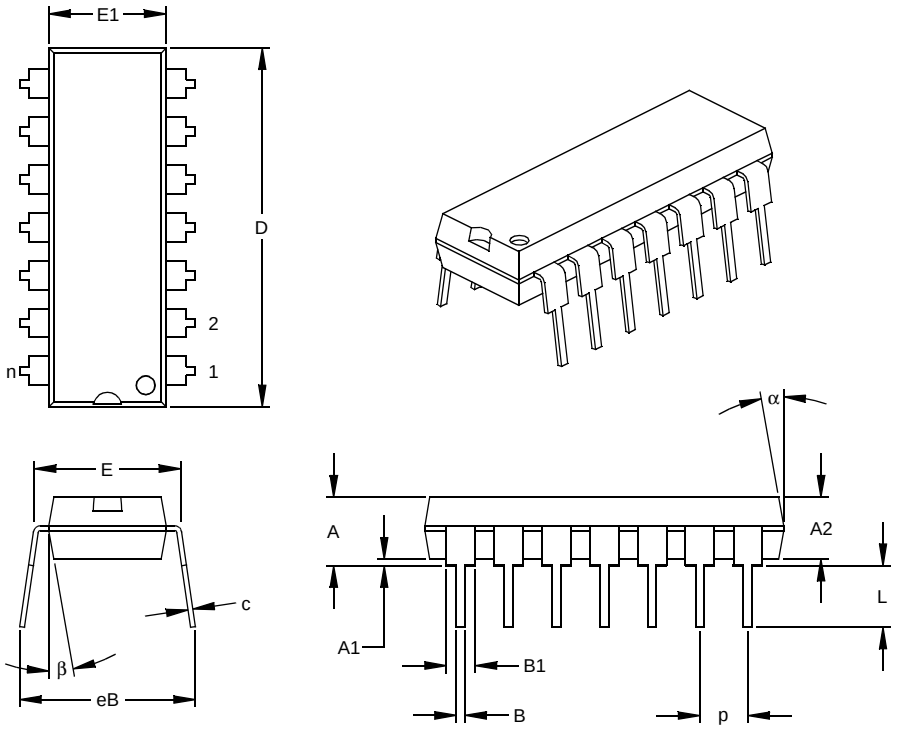
Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line thus limiting the number of available characters for customer specific information.

***** Standard PICmicro device marking consists of Microchip part number, year code, week code, and traceability code. For PICmicro device marking beyond this, certain price adders apply. Please check with your Microchip Sales Office. For QTP devices, any special marking adders are included in QTP price.

16.2 Package Details

The following sections give the technical details of the packages.

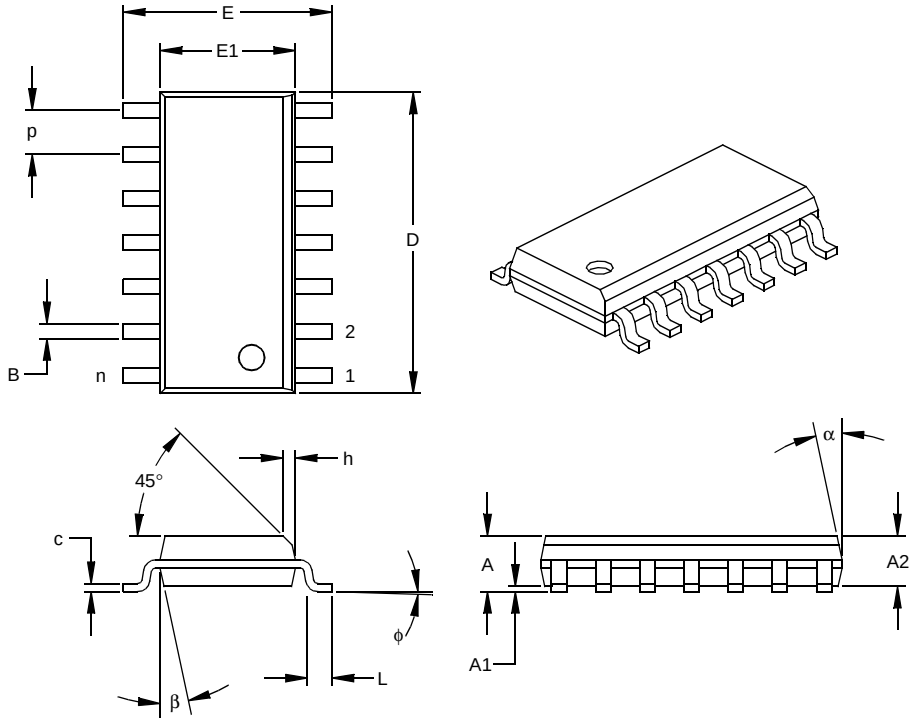
14-Lead Plastic Dual In-line (P) – 300 mil (PDIP)



Units		INCHES*			MILLIMETERS		
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		14			14	
Pitch	p		.100			2.54	
Top to Seating Plane	A	.140	.155	.170	3.56	3.94	4.32
Molded Package Thickness	A2	.115	.130	.145	2.92	3.30	3.68
Base to Seating Plane	A1	.015			0.38		
Shoulder to Shoulder Width	E	.300	.313	.325	7.62	7.94	8.26
Molded Package Width	E1	.240	.250	.260	6.10	6.35	6.60
Overall Length	D	.740	.750	.760	18.80	19.05	19.30
Tip to Seating Plane	L	.125	.130	.135	3.18	3.30	3.43
Lead Thickness	c	.008	.012	.015	0.20	0.29	0.38
Upper Lead Width	B1	.045	.058	.070	1.14	1.46	1.78
Lower Lead Width	B	.014	.018	.022	0.36	0.46	0.56
Overall Row Spacing §	eB	.310	.370	.430	7.87	9.40	10.92
Mold Draft Angle Top	α	5	10	15	5	10	15
Mold Draft Angle Bottom	β	5	10	15	5	10	15

* Controlling Parameter
§ Significant Characteristic
Notes:
Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" (0.254mm) per side.
JEDEC Equivalent: MS-001
Drawing No. C04-005

14-Lead Plastic Small Outline (SL) – Narrow, 150 mil (SOIC)

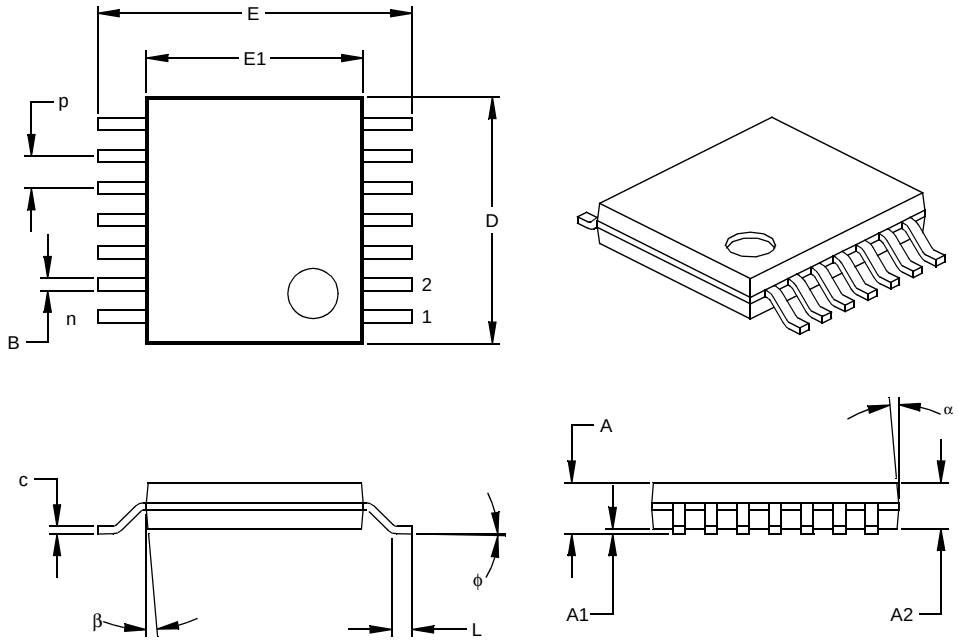


Units		INCHES*			MILLIMETERS		
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		14			14	
Pitch	p		.050			1.27	
Overall Height	A	.053	.061	.069	1.35	1.55	1.75
Molded Package Thickness	A2	.052	.056	.061	1.32	1.42	1.55
Standoff §	A1	.004	.007	.010	0.10	0.18	0.25
Overall Width	E	.228	.236	.244	5.79	5.99	6.20
Molded Package Width	E1	.150	.154	.157	3.81	3.90	3.99
Overall Length	D	.337	.342	.347	8.56	8.69	8.81
Chamfer Distance	h	.010	.015	.020	0.25	0.38	0.51
Foot Length	L	.016	.033	.050	0.41	0.84	1.27
Foot Angle	φ	0	4	8	0	4	8
Lead Thickness	c	.008	.009	.010	0.20	0.23	0.25
Lead Width	B	.014	.017	.020	0.36	0.42	0.51
Mold Draft Angle Top	α	0	12	15	0	12	15
Mold Draft Angle Bottom	β	0	12	15	0	12	15

* Controlling Parameter
§ Significant Characteristic

Notes:
Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010” (0.254mm) per side.
JEDEC Equivalent: MS-012
Drawing No. C04-065

14-Lead Plastic Thin Shrink Small Outline (ST) – 4.4 mm (TSSOP)



Units		INCHES			MILLIMETERS*		
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		14			14	
Pitch	p		.026			0.65	
Overall Height	A			.043			1.10
Molded Package Thickness	A2	.033	.035	.037	0.85	0.90	0.95
Standoff §	A1	.002	.004	.006	0.05	0.10	0.15
Overall Width	E	.246	.251	.256	6.25	6.38	6.50
Molded Package Width	E1	.169	.173	.177	4.30	4.40	4.50
Molded Package Length	D	.193	.197	.201	4.90	5.00	5.10
Foot Length	L	.020	.024	.028	0.50	0.60	0.70
Foot Angle	ϕ	0	4	8	0	4	8
Lead Thickness	c	.004	.006	.008	0.09	0.15	0.20
Lead Width	B	.007	.010	.012	0.19	0.25	0.30
Mold Draft Angle Top	α	0	5	10	0	5	10
Mold Draft Angle Bottom	β	0	5	10	0	5	10

* Controlling Parameter
§ Significant Characteristic

Notes:
Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .005” (0.127mm) per side.
JEDEC Equivalent: MO-153
Drawing No. C04-087

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.

APPENDIX B: MIGRATING FROM OTHER PICmicro® DEVICES

This discusses some of the issues in migrating from other PICmicro 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 Detect	Y	Y
Internal Pull-ups	RA0/1/2/4/5	RA0/1/2/4/5, $\overline{\text{MCLR}}$
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/BOD	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.

NOTES:

INDEX

U

V

W

ON-LINE SUPPORT

Microchip provides on-line support on the Microchip World Wide Web site.

The web site is used by Microchip as a means to make files and information easily available to customers. To view the site, the user must have access to the Internet and a web browser, such as Netscape® or Microsoft® Internet Explorer. Files are also available for FTP download from our FTP site.

Connecting to the Microchip Internet Web Site

The Microchip web site is available at the following URL:

www.microchip.com

The file transfer site is available by using an FTP service to connect to:

ftp://ftp.microchip.com

The web site and file transfer site provide a variety of services. Users may download files for the latest Development Tools, Data Sheets, Application Notes, User's Guides, Articles and Sample Programs. A variety of Microchip specific business information is also available, including listings of Microchip sales offices, distributors and factory representatives. Other data available for consideration is:

- Latest Microchip Press Releases
- Technical Support Section with Frequently Asked Questions
- Design Tips
- Device Errata
- Job Postings
- Microchip Consultant Program Member Listing
- Links to other useful web sites related to Microchip Products
- Conferences for products, Development Systems, technical information and more
- Listing of seminars and events

SYSTEMS INFORMATION AND UPGRADE HOT LINE

The Systems Information and Upgrade Line provides system users a listing of the latest versions of all of Microchip's development systems software products. Plus, this line provides information on how customers can receive the most current upgrade kits. The Hot Line Numbers are:

1-800-755-2345 for U.S. and most of Canada, and

1-480-792-7302 for the rest of the world.

042003

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 Total Pages Sent _____

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: DS41203B

Questions:

1. What are the best features of this document?

2. How does this document meet your hardware and software development needs?

3. Do you find the organization of this document easy to follow? If not, why?

4. What additions to the document do you think would enhance the structure and subject?

5. What deletions from the document could be made without affecting the overall usefulness?

6. Is there any incorrect or misleading information (what and where)?

7. How would you improve this document?

PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

PART NO.	X	/XX	XXX
Device	Temperature Range	Package	Pattern

Device	16F: Standard V_{DD} range 16FT: (Tape and Reel)	
Temperature Range	I =	-40°C to +85°C
	E =	-40°C to +125°C
Package	P =	PDIP
	SL =	SOIC (Gull wing, 150 mil body)
	ST =	TSSOP(4.4 mm)
Pattern	3-Digit Pattern Code for QTP (blank otherwise)	

Examples:

a) PIC16F688-E/P 301 = Extended Temp., PDIP package, 20 MHz, QTP pattern #301

b) PIC16F688-I/SO = Industrial Temp., SOIC package, 20 MHz

c)

* JW Devices are UV erasable and can be programmed to any device configuration. JW Devices meet the electrical requirement of each oscillator type.

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: 480-792-7627
Web Address: http://www.microchip.com

Atlanta
3780 Mansell Road, Suite 130
Alpharetta, GA 30022
Tel: 770-640-0034
Fax: 770-640-0307

Boston
2 Lan Drive, Suite 120
Westford, MA 01886
Tel: 978-692-3848
Fax: 978-692-3821

Chicago
333 Pierce Road, Suite 180
Itasca, IL 60143
Tel: 630-285-0071
Fax: 630-285-0075

Dallas
4570 Westgrove Drive, Suite 160
Addison, TX 75001
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Tri-Atria Office Building
32255 Northwestern Highway, Suite 190
Farmington Hills, MI 48334
Tel: 248-538-2250
Fax: 248-538-2260

Kokomo
2767 S. Albright Road
Kokomo, IN 46902
Tel: 765-864-8360
Fax: 765-864-8387

Los Angeles
18201 Von Karman, Suite 1090
Irvine, CA 92612
Tel: 949-263-1888
Fax: 949-263-1338

San Jose
1300 Terra Bella Avenue
Mountain View, CA 94043
Tel: 650-215-1444
Fax: 650-961-0286

Toronto
6285 Northam Drive, Suite 108
Mississauga, Ontario L4V 1X5, Canada
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Australia
Suite 22, 41 Rawson Street
Epping 2121, NSW
Australia
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Unit 706B
Wan Tai Bei Hai Bldg.
No. 6 Chaoyangmen Bei Str.
Beijing, 100027, China
Tel: 86-10-85282100
Fax: 86-10-85282104

China - Chengdu
Rm. 2401-2402, 24th Floor,
Ming Xing Financial Tower
No. 88 TIDU Street
Chengdu 610016, China
Tel: 86-28-86766200
Fax: 86-28-86766599

China - Fuzhou
Unit 28F, World Trade Plaza
No. 71 Wusi Road
Fuzhou 350001, China
Tel: 86-591-7503506
Fax: 86-591-7503521

China - Hong Kong SAR
Unit 901-6, Tower 2, Metroplaza
223 Hing Fong Road
Kwai Fong, N.T., Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

China - Shanghai
Room 701, Bldg. B
Far East International Plaza
No. 317 Xian Xia Road
Shanghai, 200051
Tel: 86-21-6275-5700
Fax: 86-21-6275-5060

China - Shenzhen
Rm. 1812, 18/F, Building A, United Plaza
No. 5022 Binhe Road, Futian District
Shenzhen 518033, China
Tel: 86-755-82901380
Fax: 86-755-8295-1393

China - Shunde
Room 401, Hongjian Building, No. 2
Fengxiangnan Road, Ronggui Town, Shunde
District, Foshan City, Guangdong 528303, China
Tel: 86-757-28395507 Fax: 86-757-28395571

China - Qingdao
Rm. B505A, Fullhope Plaza,
No. 12 Hong Kong Central Rd.
Qingdao 266071, China
Tel: 86-532-5027355 Fax: 86-532-5027205

India
Divyasree Chambers
1 Floor, Wing A (A3/A4)
No. 11, O'Shaugnessey Road
Bangalore, 560 025, India
Tel: 91-80-22290061 Fax: 91-80-22290062

Japan
Benex S-1 6F
3-18-20, Shinyokohama
Kohoku-Ku, Yokohama-shi
Kanagawa, 222-0033, Japan
Tel: 81-45-471- 6166 Fax: 81-45-471-6122

Korea
168-1, Youngbo Bldg. 3 Floor
Samsung-Dong, Kangnam-Ku
Seoul, Korea 135-882
Tel: 82-2-554-7200 Fax: 82-2-558-5932 or
82-2-558-5934

Singapore
200 Middle Road
#07-02 Prime Centre
Singapore, 188980
Tel: 65-6334-8870 Fax: 65-6334-8850

Taiwan
Kaohsiung Branch
30F - 1 No. 8
Min Chuan 2nd Road
Kaohsiung 806, Taiwan
Tel: 886-7-536-4818
Fax: 886-7-536-4803

Taiwan
Taiwan Branch
11F-3, No. 207
Tung Hua North Road
Taipei, 105, Taiwan
Tel: 886-2-2717-7175 Fax: 886-2-2545-0139

EUROPE

Austria
Durisolstrasse 2
A-4600 Wels
Austria
Tel: 43-7242-2244-399
Fax: 43-7242-2244-393

Denmark
Regus Business Centre
Lautrup hoj 1-3
Ballerup DK-2750 Denmark
Tel: 45-4420-9895 Fax: 45-4420-9910

France
Parc d'Activite du Moulin de Massy
43 Rue du Saule Trapu
Batiment A - ler Etage
91300 Massy, France
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany
Steinheilstrasse 10
D-85737 Ismaning, Germany
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Italy
Via Quasimodo, 12
20025 Legnano (MI)
Milan, Italy
Tel: 39-0331-742611
Fax: 39-0331-466781

Netherlands
P. A. De Biesbosch 14
NL-5152 SC Drunen, Netherlands
Tel: 31-416-690399
Fax: 31-416-690340

United Kingdom
505 Eskdale Road
Winnersh Triangle
Wokingham
Berkshire, England RG41 5TU
Tel: 44-118-921-5869
Fax: 44-118-921-5820

02/17/04