



Welcome to [E-XFL.COM](https://www.e-xfl.com)

What is "[Embedded - Microcontrollers](#)"?

"[Embedded - Microcontrollers](#)" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "[Embedded - Microcontrollers](#)"

Details

Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	20MHz
Connectivity	UART/USART
Peripherals	Brown-out Detect/Reset, POR, WDT
Number of I/O	12
Program Memory Size	7KB (4K x 14)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	16-VQFN Exposed Pad
Supplier Device Package	16-QFN (4x4)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f688-e-ml

PIC16F688

TABLE 1-1: PIC16F688 PINOUT DESCRIPTION

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

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

PIC16F688

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

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR/BOR	Page
Bank 1											
80h	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								xxxx xxxx	20, 117
81h	OPTION_REG	RAPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	14, 117
82h	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	19, 117
83h	STATUS	IRP	RP1	RP0	TO	PD	Z	DC	C	0001 1xxx	13, 117
84h	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	20, 117
85h	TRISA	—	—	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	--11 1111	33, 117
86h	—	Unimplemented								—	—
87h	TRISC	—	—	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	--11 1111	42, 117
88h	—	Unimplemented								—	—
89h	—	Unimplemented								—	—
8Ah	PCLATH	—	—	—	Write Buffer for upper 5 bits of Program Counter					---0 0000	19, 117
8Bh	INTCON	GIE	PEIE	T0IE	INTE	RAIE	T0IF	INTF	RAIF ⁽³⁾	0000 000x	15, 117
8Ch	PIE1	EEIE	ADIE	RCIE	C2IE	C1IE	OSFIE	TXIE	TMR1IE	0000 0000	16, 117
8Dh	—	Unimplemented								—	—
8Eh	PCON	—	—	ULPWUE	SBOREN	—	—	POR	BOR	--01 --q	18, 117
8Fh	OSCCON	—	IRCF2	IRCF1	IRCF0	OSTS	HTS	LTS	SCS	-110 x000	22, 118
90h	OSCTUNE	—	—	—	TUN4	TUN3	TUN2	TUN1	TUN0	---0 0000	26, 118
91h	ANSEL	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0	1111 1111	34, 118
92h	—	Unimplemented								—	—
93h	—	Unimplemented								—	—
94h	—	Unimplemented								—	—
95h	WPUA ⁽²⁾	—	—	WPUA5	WPUA4	—	WPUA2	WPUA1	WPUA0	--11 -111	35, 118
96h	IOCA	—	—	IOCA5	IOCA4	IOCA3	IOCA2	IOCA1	IOCA0	--00 0000	35, 118
97h	EEDATH	—	—	EEDATH5	EEDATH4	EEDATH3	EEDATH2	EEDATH1	EEDATH0	--00 0000	78, 118
98h	EEADRH	—	—	—	—	EEADRH3	EEADRH2	EEADRH1	EEADRH0	---- 0000	78, 118
99h	VRCON	VREN	—	VRR	—	VR3	VR2	VR1	VR0	0-0- 0000	63, 118
9Ah	EEDAT	EEDAT7	EEDAT6	EEDAT5	EEDAT4	EEDAT3	EEDAT2	EEDAT1	EEDAT0	0000 0000	78, 118
9Bh	EEADR	EEADR7	EEADR6	EEADR5	EEADR4	EEADR3	EEADR2	EEADR1	EEADR0	0000 0000	78, 118
9Ch	EECON1	EEPGD	—	—	—	WRERR	WREN	WR	RD	x--- x000	79, 118
9Dh	EECON2	EEPROM Control 2 Register (not a physical register)								---- ----	77, 118
9Eh	ADRESL	Least Significant 2 bits of the left shifted result or 8 bits of the right shifted result								xxxx xxxx	72, 118
9Fh	ADCON1	—	ADCS2	ADCS1	ADCS0	—	—	—	—	-000 ----	71, 118

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

- Note**
- 1: Other (non Power-up) Resets include MCLR Reset and Watchdog Timer Reset during normal operation.
 - 2: RA3 pull-up is enabled when pin is configured as MCLR in the Configuration Word register.
 - 3: MCLR and WDT Reset does not affect the previous value data latch. The RAIF bit will be cleared upon Reset but will set again if the mismatched exists.

3.6 Clock Switching

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

3.6.1 SYSTEM CLOCK SELECT (SCS) BIT

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

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

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

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

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

3.7 Two-Speed Clock Start-up Mode

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

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

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

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

3.7.1 TWO-SPEED START-UP MODE CONFIGURATION

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

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

Two-Speed Start-up mode is entered after:

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

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

3.7.2 TWO-SPEED START-UP SEQUENCE

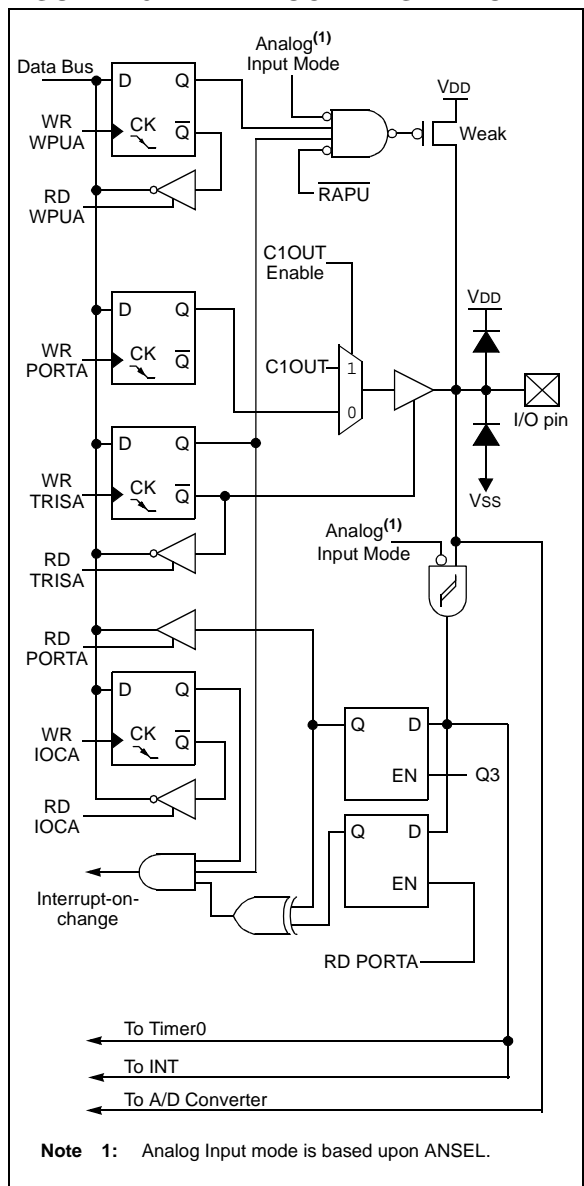
1. Wake-up from Power-on Reset or Sleep.
2. Instructions begin execution by the internal oscillator at the frequency set in the IRCF<2:0> bits of the OSCCON register.
3. OST enabled to count 1024 clock cycles.
4. OST timed out, wait for falling edge of the internal oscillator.
5. OSTS is set.
6. System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

4.2.5.3 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 Timer0
- an external edge triggered interrupt
- a digital output from the comparator

FIGURE 4-3: BLOCK DIAGRAM OF RA2

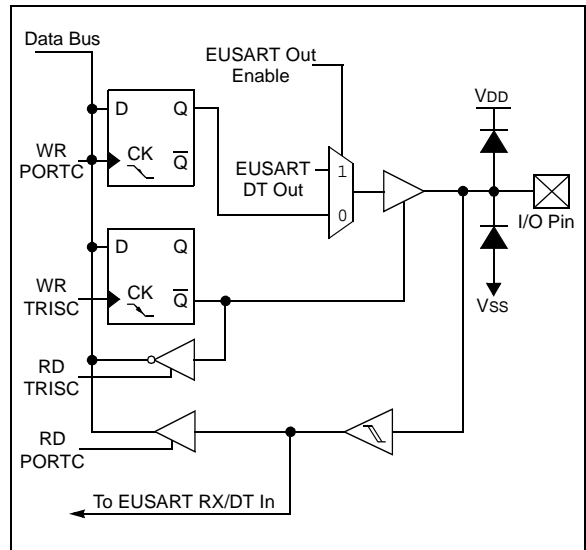


4.3.6 RC5/RX/DT

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



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

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

PIC16F688

FIGURE 7-8: COMPARATOR VOLTAGE REFERENCE BLOCK DIAGRAM

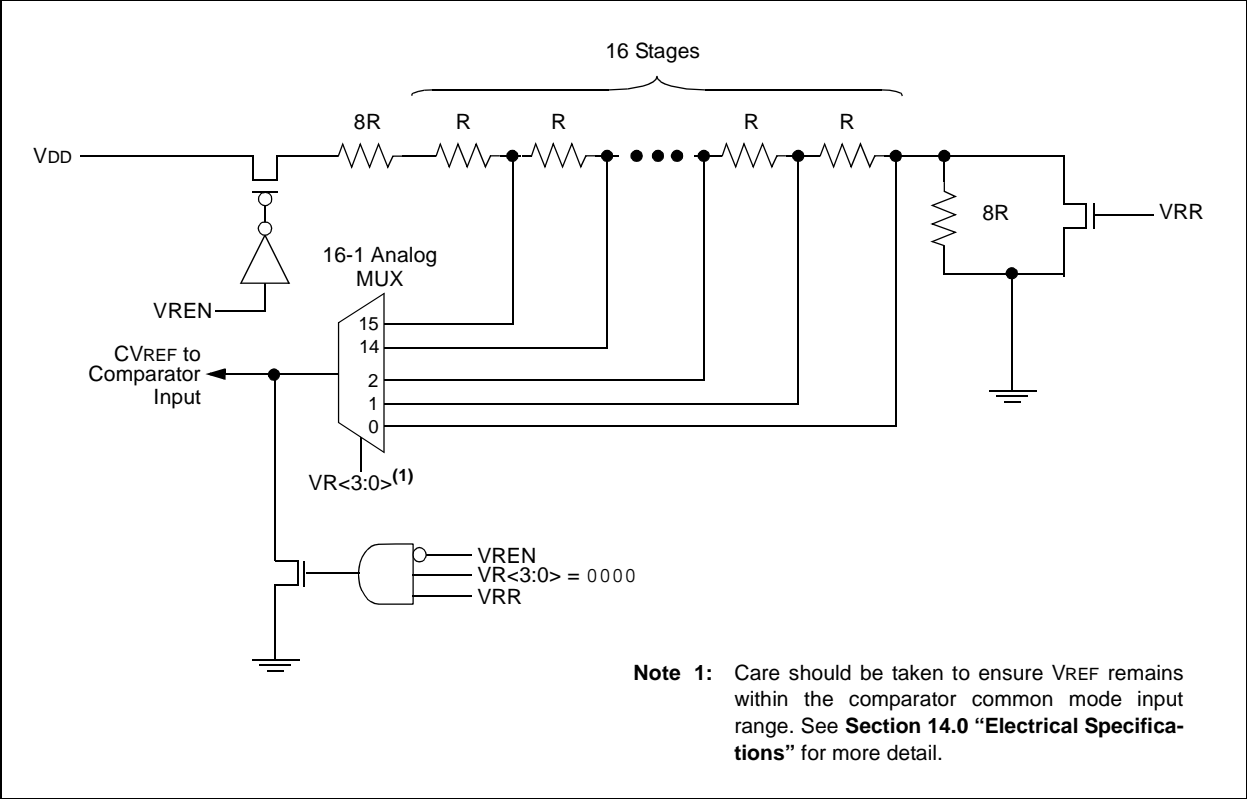


TABLE 7-2: SUMMARY OF REGISTERS ASSOCIATED WITH THE COMPARATOR AND VOLTAGE REFERENCE MODULES

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
ANSEL	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0	1111 1111	1111 1111
CMCON0	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	0000 0000	0000 0000
CMCON1	—	—	—	—	—	—	T1GSS	C2SYNC	---- --10	---- --10
INTCON	GIE	PEIE	T0IE	INTE	RAIE	T0IF	INTF	RAIF	0000 000x	0000 000x
PIE1	EEIE	ADIE	RCIE	C2IE	C1IE	OSFIE	TXIE	TMR1IE	0000 0000	0000 0000
PIR1	EEIF	ADIF	RCIF	C2IF	C1IF	OSFIF	TXIF	TMR1IF	0000 0000	0000 0000
PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	--x0 x000	--x0 x000
PORTC	—	—	RC5	RC4	RC3	RC2	RC1	RC0	--xx 0000	--xx 0000
TRISA	—	—	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	--11 1111	--11 1111
TRISC	—	—	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	--11 1111	--11 1111
VRCON	VREN	—	VRR	—	VR3	VR2	VR1	VR0	0-0- 0000	0-0- 0000

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

PIC16F688

8.1.5 INTERRUPTS

The ADC module allows for the ability to generate an interrupt upon completion of an Analog-to-Digital Conversion. The ADC interrupt flag is the ADIF bit in the PIR1 register. The ADC interrupt enable is the ADIE bit in the PIE1 register. The ADIF bit must be cleared in software.

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

This interrupt can be generated while the device is operating or while in Sleep. If the device is in Sleep, the interrupt will wake-up the device. Upon waking from Sleep, the next instruction following the `SLEEP` instruction is always executed. If the user is attempting to wake-up from Sleep and resume in-line code execution, the global interrupt must be disabled. If the global interrupt is enabled, execution will switch to the Interrupt Service Routine.

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

8.1.6 RESULT FORMATTING

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

Figure 8-4 shows the two output formats.

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

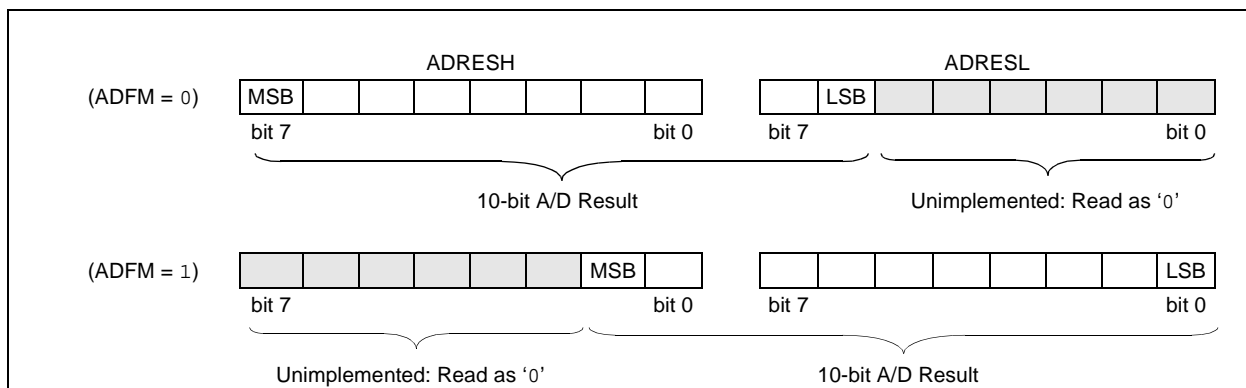


TABLE 8-2: SUMMARY OF ASSOCIATED ADC REGISTERS

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

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

10.3 EUSART Baud Rate Generator (BRG)

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

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

Table 10-3 contains the formulas for determining the baud rate. Example 10-1 provides a sample calculation for determining the baud rate and baud rate error.

Typical baud rates and error values for various asynchronous modes have been computed for your convenience and are shown in Table 10-3. It may be advantageous to use the high baud rate (BRGH = 1), or the 16-bit BRG (BRG16 = 1) to reduce the baud rate error. The 16-bit BRG mode is used to achieve slow baud rates for fast oscillator frequencies.

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

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

EXAMPLE 10-1: CALCULATING BAUD RATE ERROR

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

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

Solving for SPBRGH:SPBRG:

$$\begin{aligned} X &= \frac{\frac{F_{\text{OSC}}}{\text{Desired Baud Rate}}}{64} - 1 \\ &= \frac{\frac{16000000}{9600}}{64} - 1 \\ &= [25.042] = 25 \end{aligned}$$

$$\begin{aligned} \text{Calculated Baud Rate} &= \frac{16000000}{64(25 + 1)} \\ &= 9615 \end{aligned}$$

$$\begin{aligned} \text{Error} &= \frac{\text{Calc. Baud Rate} - \text{Desired Baud Rate}}{\text{Desired Baud Rate}} \\ &= \frac{(9615 - 9600)}{9600} = 0.16\% \end{aligned}$$

TABLE 10-3: BAUD RATE FORMULAS

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

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

TABLE 10-4: REGISTERS ASSOCIATED WITH THE BAUD RATE GENERATOR

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
BAUDCTL	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	01-0 0-00	01-0 0-00
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
SPBRG	BRG7	BRG6	BRG5	BRG4	BRG3	BRG2	BRG1	BRG0	0000 0000	0000 0000
SPBRGH	BRG15	BRG14	BRG13	BRG12	BRG11	BRG10	BRG9	BRG8	0000 0000	0000 0000
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010	0000 0010

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

PIC16F688

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

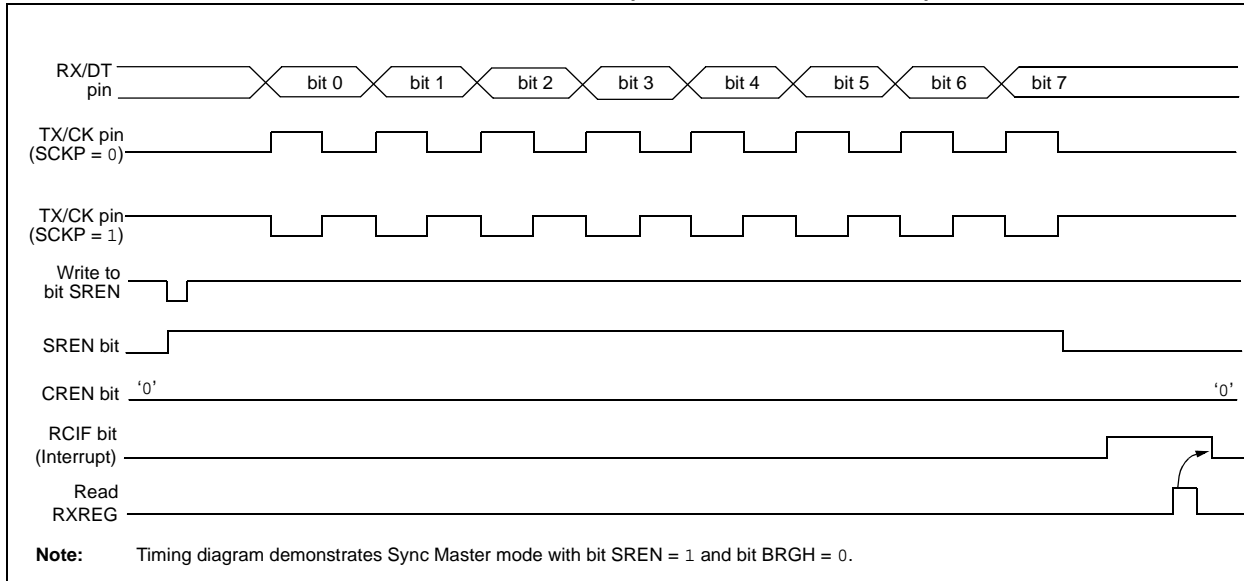


TABLE 10-8: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER RECEPTION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
BAUDCTL	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	01-0 0-00	01-0 0-00
INTCON	GIE	PEIE	T0IE	INTE	RAIE	T0IF	INTF	RAIF	0000 000x	0000 000x
PIE1	EEIE	ADIE	RCIE	C2IE	C1IE	OSFIE	TXIE	TMR1IE	0000 0000	0000 0000
PIR1	EEIF	ADIF	RCIF	C2IF	C1IF	OSFIF	TXIF	TMR1IF	0000 0000	0000 0000
RCREG	EUSART Receive Data Register								0000 0000	0000 0000
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
SPBRG	BRG7	BRG6	BRG5	BRG4	BRG3	BRG2	BRG1	BRG0	0000 0000	0000 0000
SPBRGH	BRG15	BRG14	BRG13	BRG12	BRG11	BRG10	BRG9	BRG8	0000 0000	0000 0000
TRISC	—	—	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	--11 1111	--11 1111
TXREG	EUSART Transmit Data Register								0000 0000	0000 0000
TXSTA	CSRC	TX9	TXEN	SYNC	SEnDB	BRGH	TRMT	TX9D	0000 0010	0000 0010

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

PIC16F688

11.2 Reset

The PIC16F688 differentiates between various kinds of Reset:

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

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

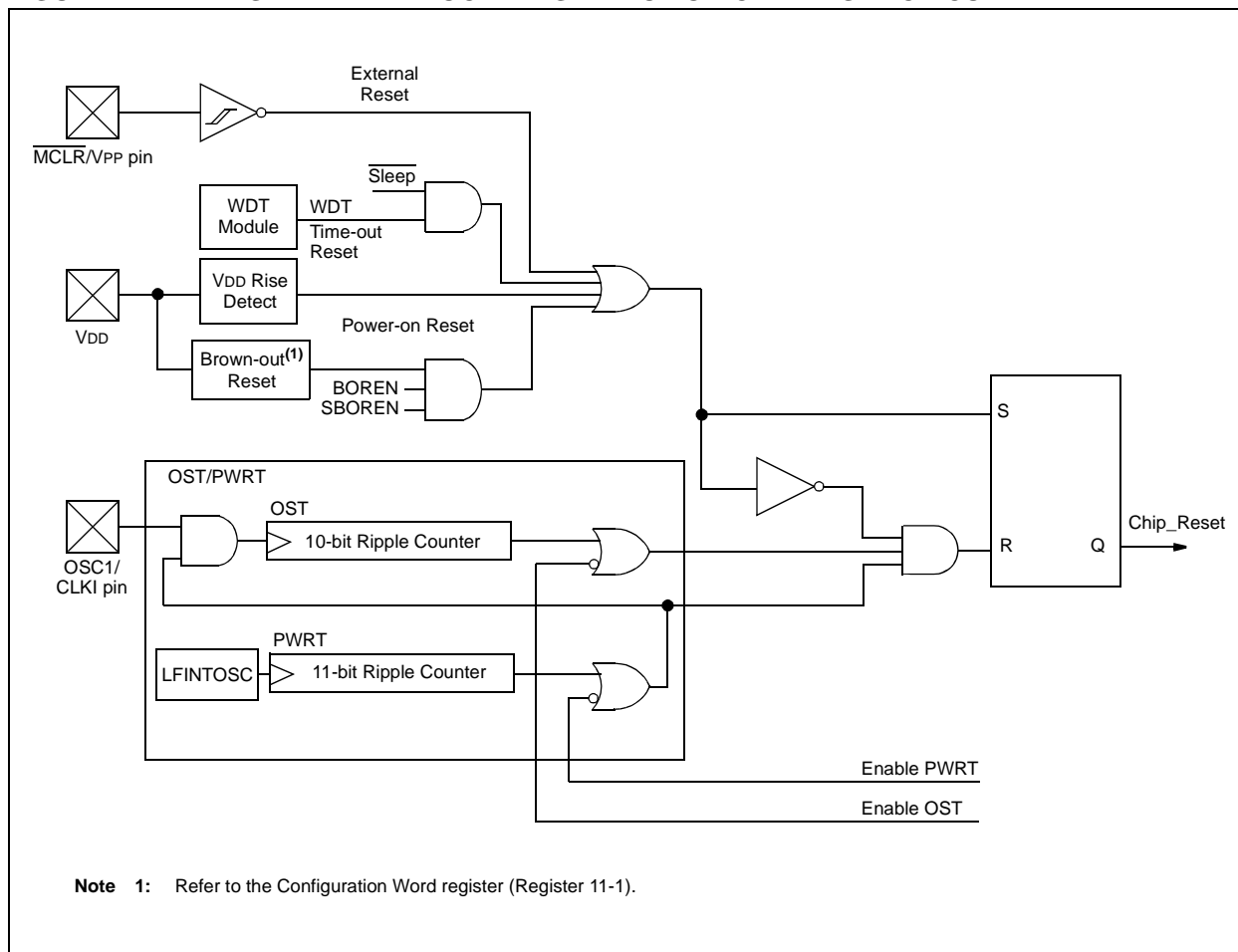
- Power-on Reset
- $\overline{\text{MCLR}}$ Reset
- $\overline{\text{MCLR}}$ Reset during Sleep
- WDT Reset
- Brown-out Reset (BOR)

They are not affected by a WDT wake-up since this is viewed as the resumption of normal operation. $\overline{\text{TO}}$ and $\overline{\text{PD}}$ bits are set or cleared differently in different Reset situations, as indicated in Table 11-2. These bits are used in software to determine the nature of the Reset. See Table 11-4 for a full description of Reset states of all registers.

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

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

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



PIC16F688

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

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

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

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

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

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

4: See Table 11-5 for Reset value for specific condition.

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

TABLE 11-5: INITIALIZATION CONDITION FOR SPECIAL REGISTERS

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

12.0 INSTRUCTION SET SUMMARY

The PIC16F688 instruction set is highly orthogonal and is comprised of three basic categories:

- **Byte-oriented** operations
- **Bit-oriented** operations
- **Literal and control** operations

Each PIC16 instruction is a 14-bit word divided into an **opcode**, which specifies the instruction type and one or more **operands**, which further specify the operation of the instruction. The formats for each of the categories is presented in Figure 12-1, while the various opcode fields are summarized in Table 12-1.

Table 12-2 lists the instructions recognized by the MPASM™ assembler.

For **byte-oriented** instructions, 'f' represents a file register designator and 'd' represents a destination designator. The file register designator specifies which file register is to be used by the instruction.

The destination designator specifies where the result of the operation is to be placed. If 'd' is zero, the result is placed in the W register. If 'd' is one, the result is placed in the file register specified in the instruction.

For **bit-oriented** instructions, 'b' represents a bit field designator, which selects the bit affected by the operation, while 'f' represents the address of the file in which the bit is located.

For **literal and control** operations, 'k' represents an 8-bit or 11-bit constant, or literal value.

One instruction cycle consists of four oscillator periods; for an oscillator frequency of 4 MHz, this gives a nominal instruction execution time of 1 μs. All instructions are executed within a single instruction cycle, unless a conditional test is true, or the program counter is changed as a result of an instruction. When this occurs, the execution takes two instruction cycles, with the second cycle executed as a NOP.

All instruction examples use the format '0xhh' to represent a hexadecimal number, where 'h' signifies a hexadecimal digit.

12.1 Read-Modify-Write Operations

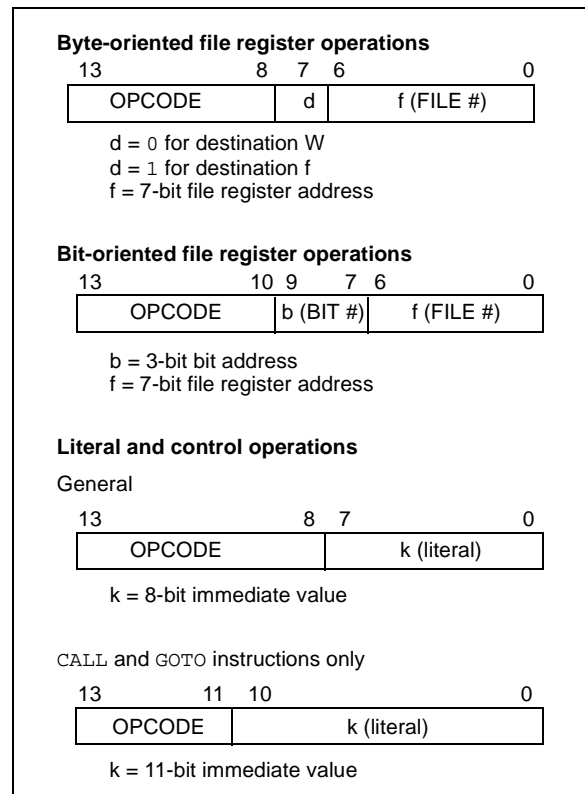
Any instruction that specifies a file register as part of the instruction performs a Read-Modify-Write (R-M-W) operation. The register is read, the data is modified, and the result is stored according to either the instruction, or the destination designator 'd'. A read operation is performed on a register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

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

TABLE 12-1: OPCODE FIELD DESCRIPTIONS

Field	Description
f	Register file address (0x00 to 0x7F)
W	Working register (accumulator)
b	Bit address within an 8-bit file register
k	Literal field, constant data or label
x	Don't care location (= 0 or 1). The assembler will generate code with x = 0. It is the recommended form of use for compatibility with all Microchip software tools.
d	Destination select; d = 0: store result in W, d = 1: store result in file register f. Default is d = 1.
PC	Program Counter
TO	Time-out bit
C	Carry bit
DC	Digit carry bit
Z	Zero bit
PD	Power-down bit

FIGURE 12-1: GENERAL FORMAT FOR INSTRUCTIONS



PIC16F688

BTFSS **Bit Test f, Skip if Set**

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

CLRWDT **Clear Watchdog Timer**

Syntax: [*label*] CLRWDT
Operands: None
Operation: 00h → WDT
 0 → WDT prescaler,
 1 → \overline{TO}
 1 → \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.

CALL **Call Subroutine**

Syntax: [*label*] CALL *k*
Operands: $0 \leq k \leq 2047$
Operation: (PC)+1 → TOS,
 k → PC<10:0>,
 (PCLATH<4:3>) → PC<12:11>
Status Affected: None
Description: Call Subroutine. First, return address (PC + 1) is pushed onto the stack. The eleven-bit immediate address is loaded into PC bits <10:0>. The upper bits of the PC are loaded from PCLATH. CALL is a two-cycle instruction.

COMF **Complement f**

Syntax: [*label*] COMF *f*,*d*
Operands: $0 \leq f \leq 127$
 d ∈ [0,1]
Operation: (\bar{f}) → (destination)
Status Affected: Z
Description: The contents of register '*f*' are complemented. If '*d*' is '0', the result is stored in W. If '*d*' is '1', the result is stored back in register '*f*'.

CLRF **Clear f**

Syntax: [*label*] CLRF *f*
Operands: $0 \leq f \leq 127$
Operation: 00h → (*f*)
 1 → Z
Status Affected: Z
Description: The contents of register '*f*' are cleared and the Z bit is set.

DECF **Decrement f**

Syntax: [*label*] DECF *f*,*d*
Operands: $0 \leq f \leq 127$
 d ∈ [0,1]
Operation: (*f*) - 1 → (destination)
Status Affected: Z
Description: Decrement register '*f*'. If '*d*' is '0', the result is stored in the W register. If '*d*' is '1', the result is stored back in register '*f*'.

CLRW **Clear W**

Syntax: [*label*] CLRW
Operands: None
Operation: 00h → (W)
 1 → Z
Status Affected: Z
Description: W register is cleared. Zero bit (Z) is set.

13.2 MPASM Assembler

The MPASM Assembler is a full-featured, universal macro assembler for all PIC MCUs.

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

The MPASM Assembler features include:

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

13.3 MPLAB C18 and MPLAB C30 C Compilers

The MPLAB C18 and MPLAB C30 Code Development Systems are complete ANSI C compilers for Microchip's PIC18 and PIC24 families of microcontrollers and the dsPIC30 and dsPIC33 family of digital signal controllers. These compilers provide powerful integration capabilities, superior code optimization and ease of use not found with other compilers.

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

13.4 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler and the MPLAB C18 C Compiler. It can link relocatable objects from precompiled libraries, using directives from a linker script.

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

The object linker/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 ASM30 Assembler, Linker and Librarian

MPLAB ASM30 Assembler produces relocatable machine code from symbolic assembly language for dsPIC30F devices. MPLAB C30 C Compiler uses the assembler to produce its object file. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. Notable features of the assembler include:

- Support for the entire dsPIC30F instruction set
- Support for fixed-point and floating-point data
- Command line interface
- Rich directive set
- Flexible macro language
- MPLAB IDE compatibility

13.6 MPLAB SIM Software Simulator

The MPLAB SIM Software Simulator allows code development in a PC-hosted environment by simulating the PIC MCUs and dsPIC® DSCs on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a comprehensive stimulus controller. Registers can be logged to files for further run-time analysis. The trace buffer and logic analyzer display extend the power of the simulator to record and track program execution, actions on I/O, most peripherals and internal registers.

The MPLAB SIM Software Simulator fully supports symbolic debugging using the MPLAB C18 and MPLAB C30 C Compilers, and the MPASM and MPLAB ASM30 Assemblers. The software simulator offers the flexibility to develop and debug code outside of the hardware laboratory environment, making it an excellent, economical software development tool.

FIGURE 15-12: MAXIMUM I_{DD} vs. F_{osc} OVER V_{DD} (HFINTOSC MODE)

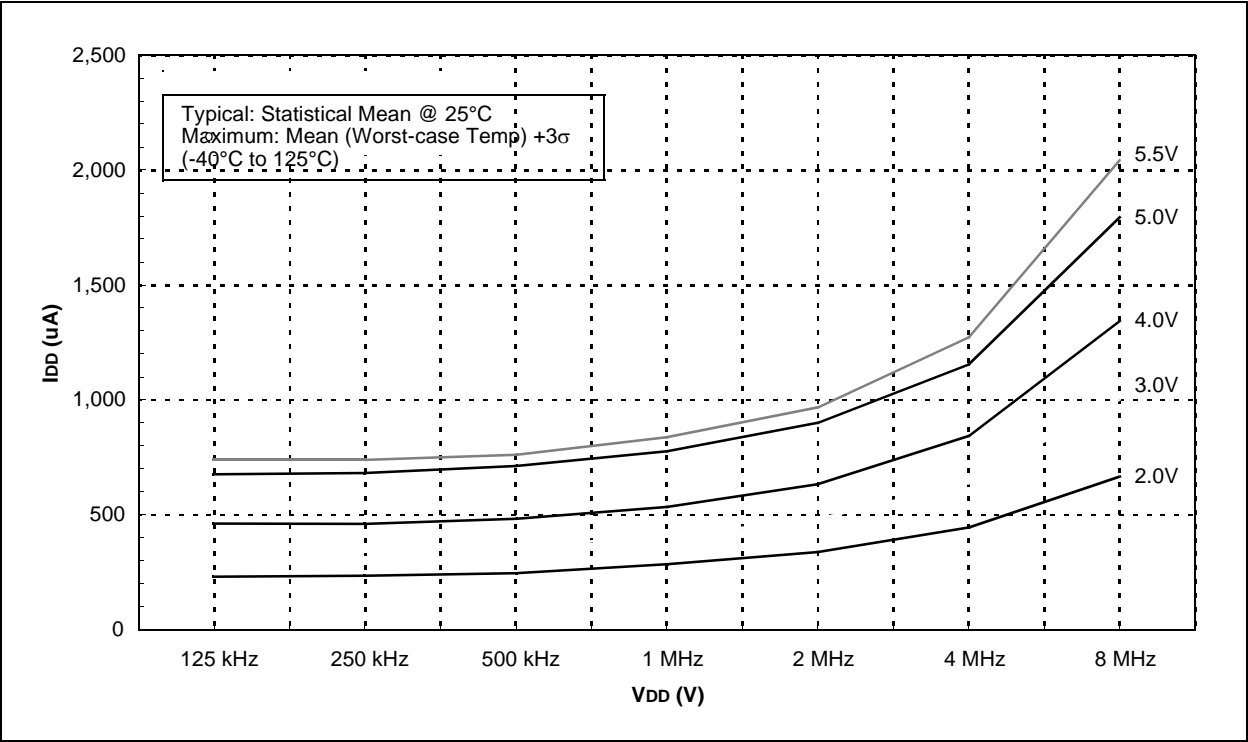


FIGURE 15-13: TYPICAL I_{PD} vs. V_{DD} (SLEEP MODE, ALL PERIPHERALS DISABLED)

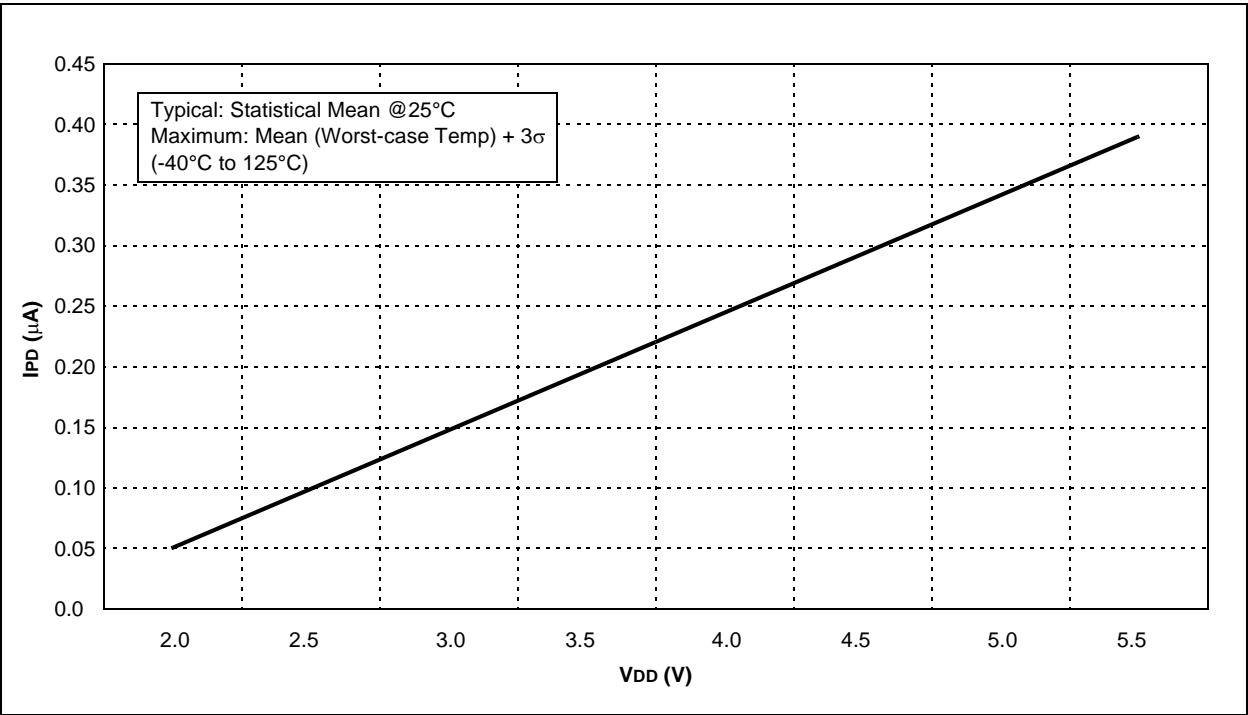


FIGURE 15-14: MAXIMUM I_{PD} vs. V_{DD} (SLEEP MODE, ALL PERIPHERALS DISABLED)

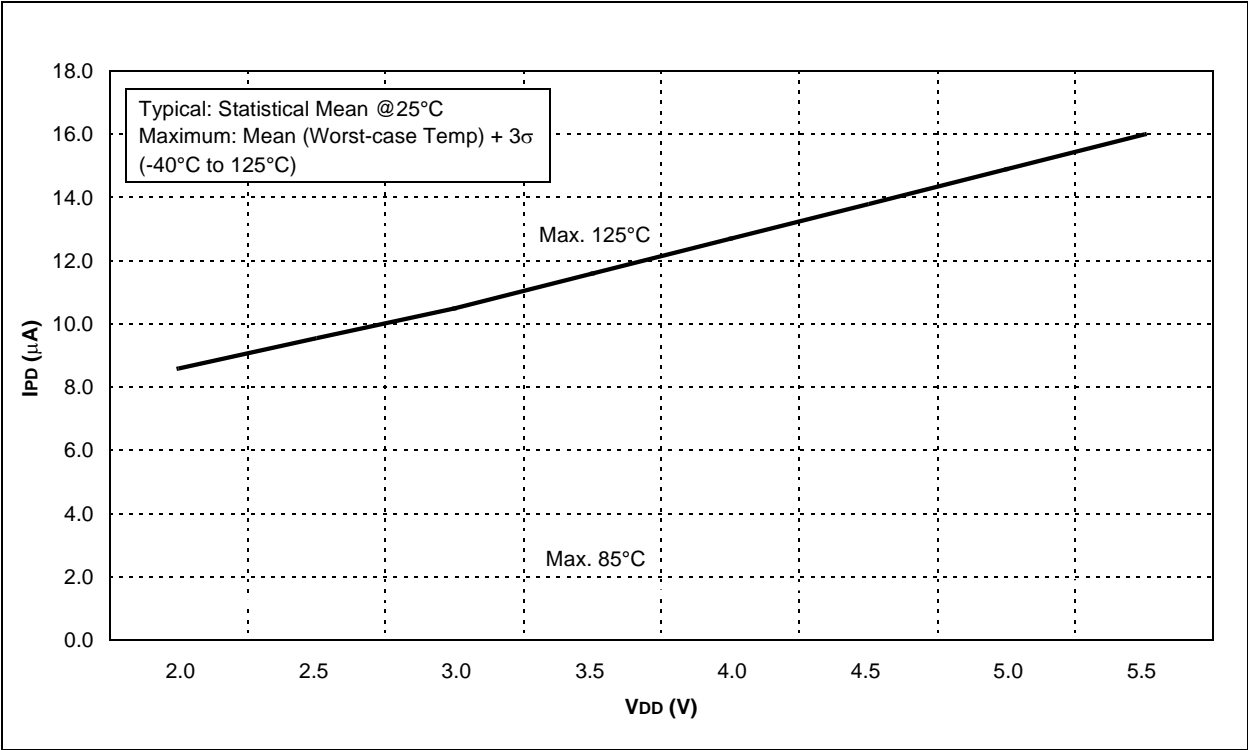


FIGURE 15-15: COMPARATOR I_{PD} vs. V_{DD} (BOTH COMPARATORS ENABLED)

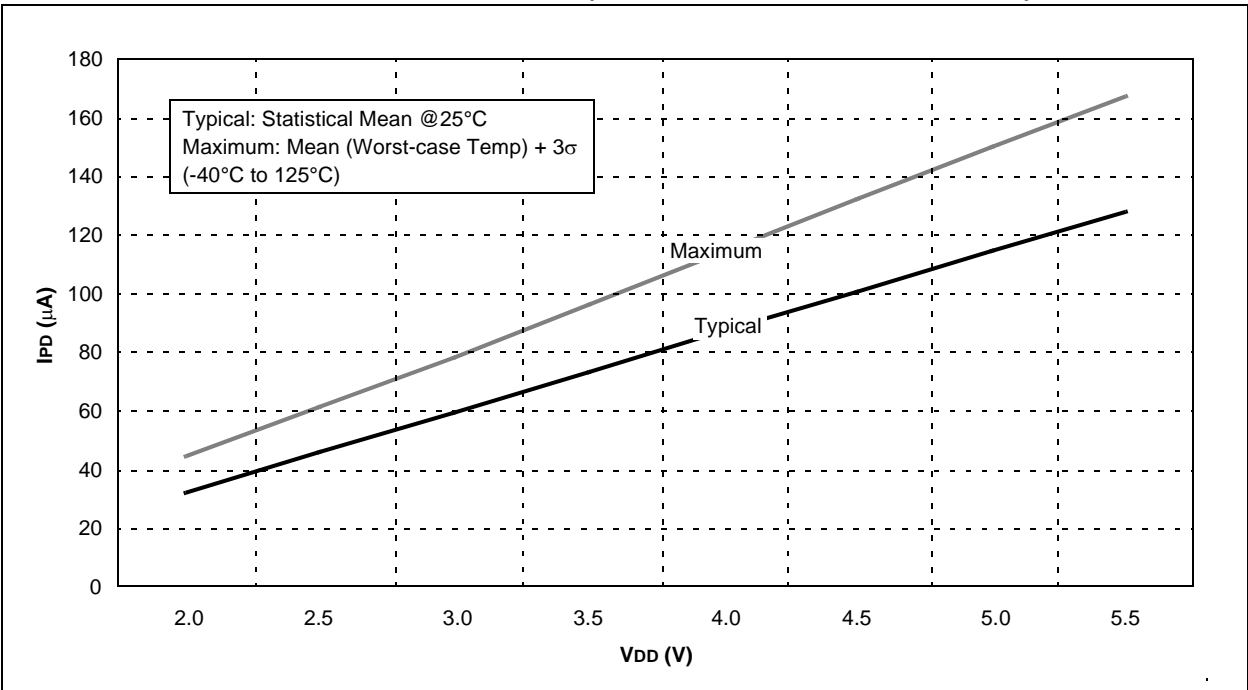


FIGURE 15-28: SCHMITT TRIGGER INPUT THRESHOLD VIN vs. VDD OVER TEMPERATURE

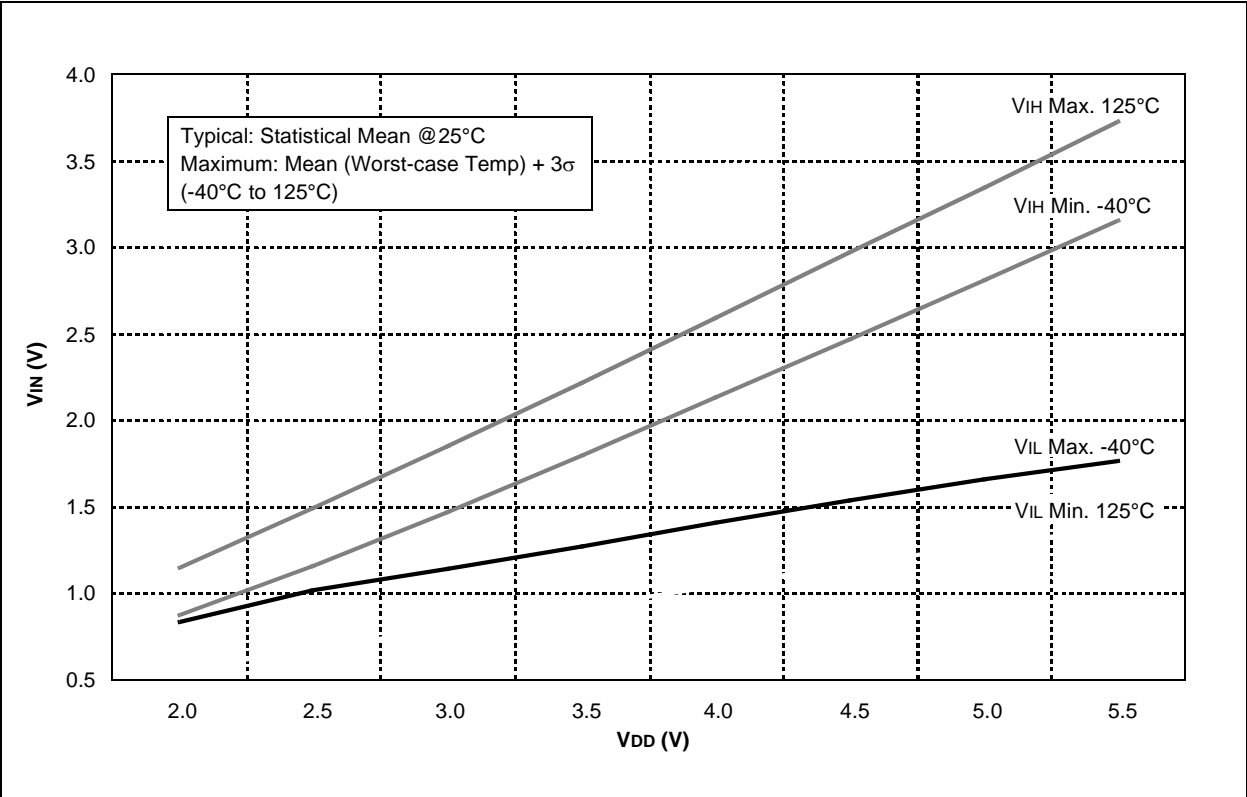
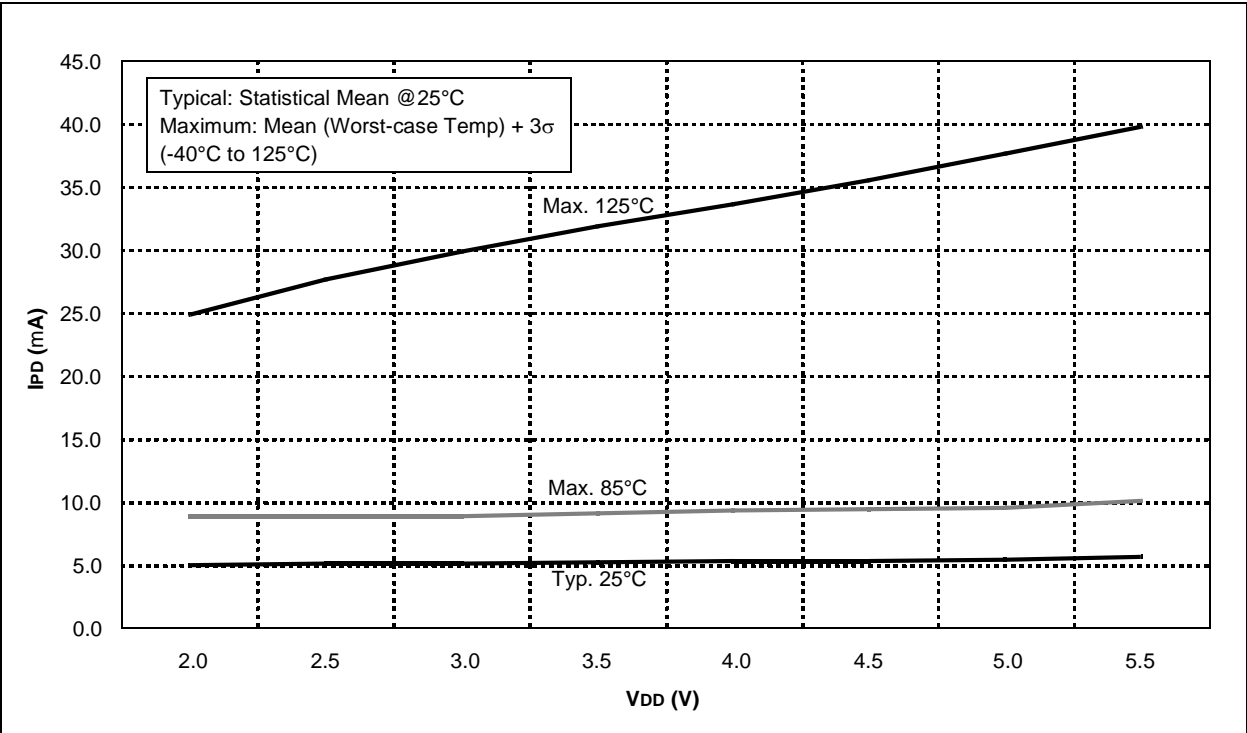


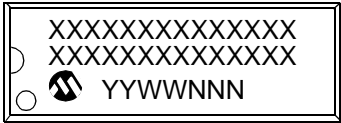
FIGURE 15-29: T1OSC IPD vs. VDD OVER TEMPERATURE (32 kHz)



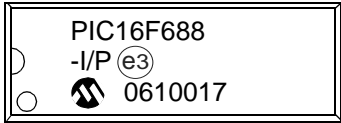
16.0 PACKAGING INFORMATION

16.1 Package Marking Information

14-Lead PDIP (Skinny DIP)



Example



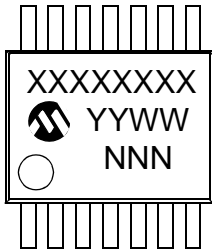
14-Lead SOIC (3.90 mm)



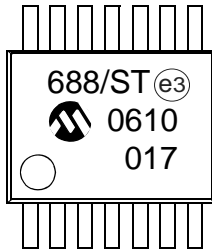
Example



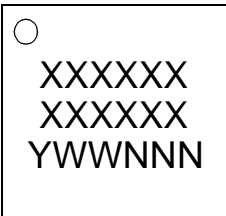
14-Lead TSSOP



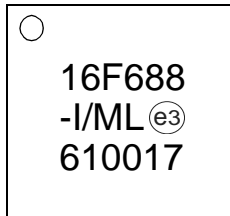
Example



16-Lead QFN



Example



Legend:

XX...X

Y

YY

WW

NNN

e3

*

Customer-specific information

Year code (last digit of calendar year)

Year code (last 2 digits of calendar year)

Week code (week of January 1 is week '01')

Alphanumeric traceability code

Pb-free JEDEC designator for Matte Tin (Sn)

This package is Pb-free. The Pb-free JEDEC designator (

e3

) can be found on the outer packaging for this package.

Note:

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

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