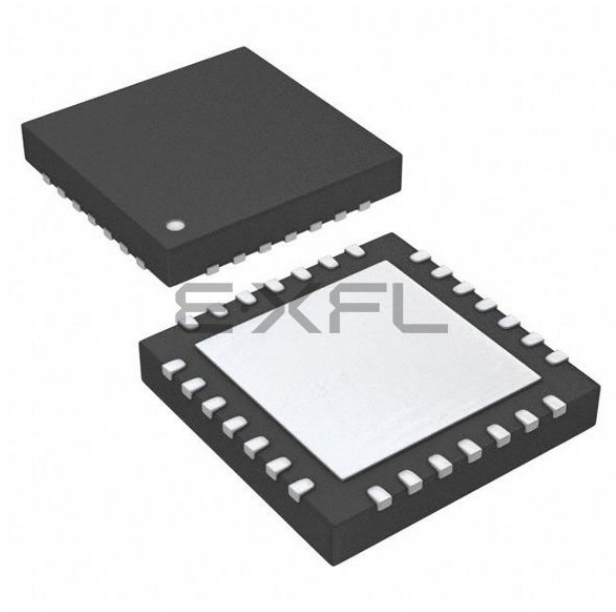


Welcome to E-XFL.COM

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.

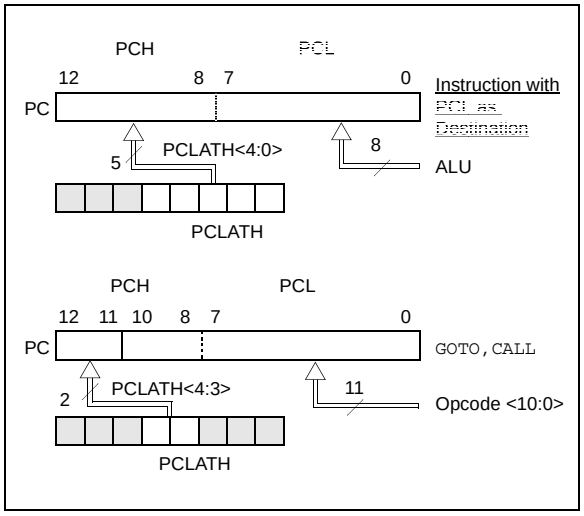
Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	10MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	16
Program Memory Size	7KB (4K x 14)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	368 x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 5.5V
Data Converters	A/D 7x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-VQFN Exposed Pad
Supplier Device Package	28-QFN (6x6)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lf88-i-ml

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 upper bits (PC<12:8>) are not readable but are indirectly writable through the PCLATH register. On any Reset, the upper bits of the PC will be cleared. Figure 2-4 shows the two situations for the loading of the PC. The upper example in the figure shows how the PC is loaded on a write to PCL (PCLATH<4:0> → PCH). The lower example in the figure shows how the PC is loaded during a CALL or GOTO instruction (PCLATH<4:3> → PCH).

FIGURE 2-4: 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 doing 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"*.

2.3.2 STACK

The PIC16F87/88 family has an 8-level deep x 13-bit wide hardware stack. The stack space is not part of either program or data space and the Stack Pointer is not readable or writable. The PC is PUSHed onto the stack when a CALL instruction is executed or an interrupt causes a branch. The stack is POPed in the event of a RETURN, RETLW or a RETFIE instruction execution. PCLATH is not affected by a PUSH or POP operation.

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

Note 1:	There are no status bits to indicate stack overflow or stack underflow conditions.
2:	There are no instructions/mnemonics called PUSH or POP. These are actions that occur from the execution of the CALL, RETURN, RETLW and RETFIE instructions, or the vectoring to an interrupt address.

2.4 Program Memory Paging

All PIC16F87/88 devices are capable of addressing a continuous 8K word block of program memory. The CALL and GOTO instructions provide only 11 bits of address to allow branching within any 2K program memory page. When doing a CALL or GOTO instruction, the upper 2 bits of the address are provided by PCLATH<4:3>. When doing a CALL or GOTO instruction, the user must ensure that the page select bits are programmed so that the desired program memory page is addressed. If a return from a CALL instruction (or interrupt) is executed, the entire 13-bit PC is popped off the stack. Therefore, manipulation of the PCLATH<4:3> bits is not required for the RETURN instructions (which POPs the address from the stack).

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

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

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

```
        ORG 0x500
        BCF PCLATH, 4
        BSF PCLATH, 3 ;Select page 1
                      ;(800h-FFFh)
        CALL SUB1_P1  ;Call subroutine in
        :             ;page 1 (800h-FFFh)
        :
        ORG 0x900     ;page 1 (800h-FFFh)
SUB1_P1
        :             ;called subroutine
                      ;page 1 (800h-FFFh)
        :
        RETURN        ;return to
                      ;Call subroutine
                      ;in page 0
                      ;(000h-7FFh)
```

TABLE 4-4: CLOCK SWITCHING MODES

Current System Clock	SCS Bits <1:0> Modified to:	Delay	OSTS Bit	IOFS Bit	T1RUN Bit	New System Clock	Comments
LP, XT, HS, T1OSC, EC, RC	10 (INTRC) FOSC<2:0> = LP, XT or HS	8 Clocks of INTRC	0	1(1)	0	INTRC or INTOSC or INTOSC Postscaler	The internal RC oscillator frequency is dependant upon the IRCF bits.
LP, XT, HS, INTRC, EC, RC	01 (T1OSC) FOSC<2:0> = LP, XT or HS	8 Clocks of T1OSC	0	N/A	1	T1OSC	T1OSCEN bit must be enabled.
INTRC T1OSC	00 FOSC<2:0> = EC or FOSC<2:0> = RC	8 Clocks of EC or RC	1	N/A	0	EC or RC	
INTRC T1OSC	00 FOSC<2:0> = LP, XT, HS	1024 Clocks (OST) + 8 Clocks of LP, XT, HS	1	N/A	0	LP, XT, HS	During the 1024 clocks, program execution is clocked from the secondary oscillator until the primary oscillator becomes stable.
LP, XT, HS	00 (Due to Reset) LP, XT, HS	1024 Clocks (OST)	1	N/A	0	LP, XT, HS	When a Reset occurs, there is no clock transition sequence. Instruction execution and/or peripheral operation is suspended unless Two-Speed Start-up mode is enabled, after which the INTRC will act as the system clock until the OST timer has expired.

Note 1: If the new clock source is the INTOSC or INTOSC postscaler, then the IOFS bit will be set 4 ms (approx.) after the clock change.

FIGURE 5-2: BLOCK DIAGRAM OF RA3/AN3/VREF+/C1OUT PIN

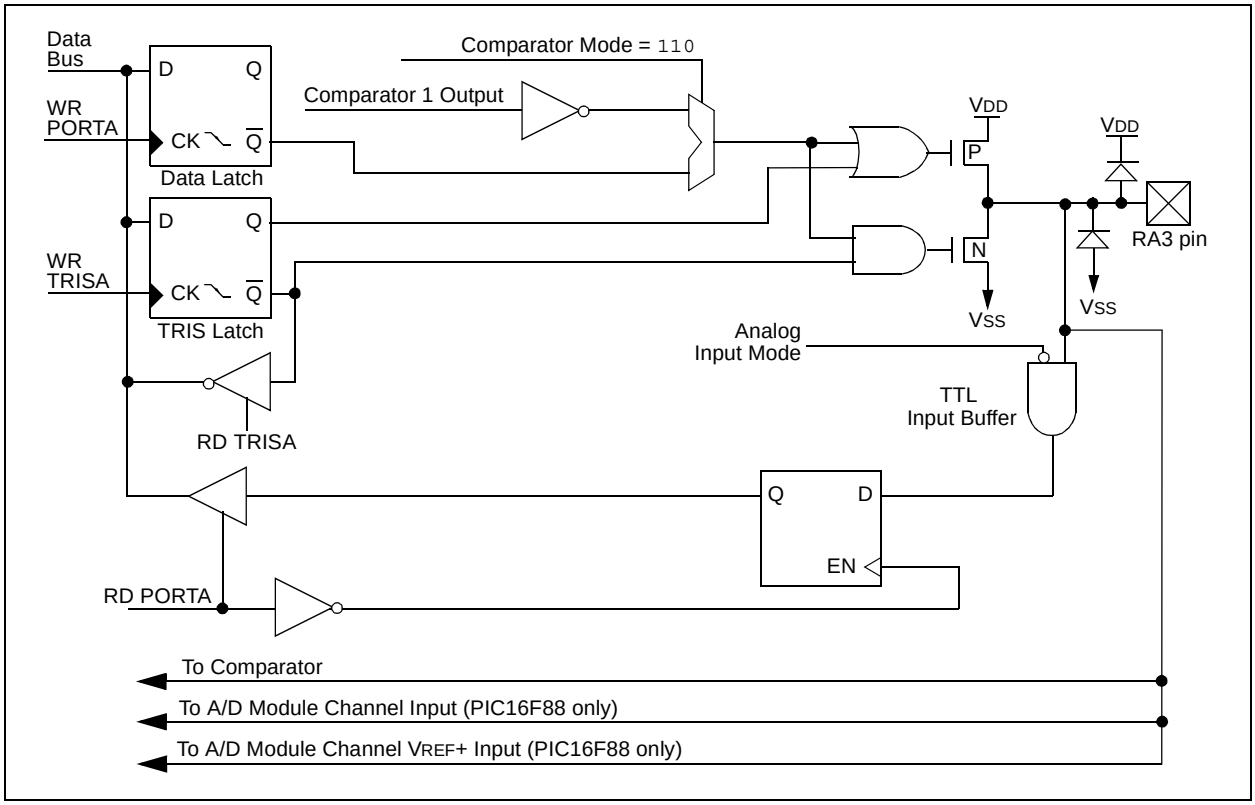
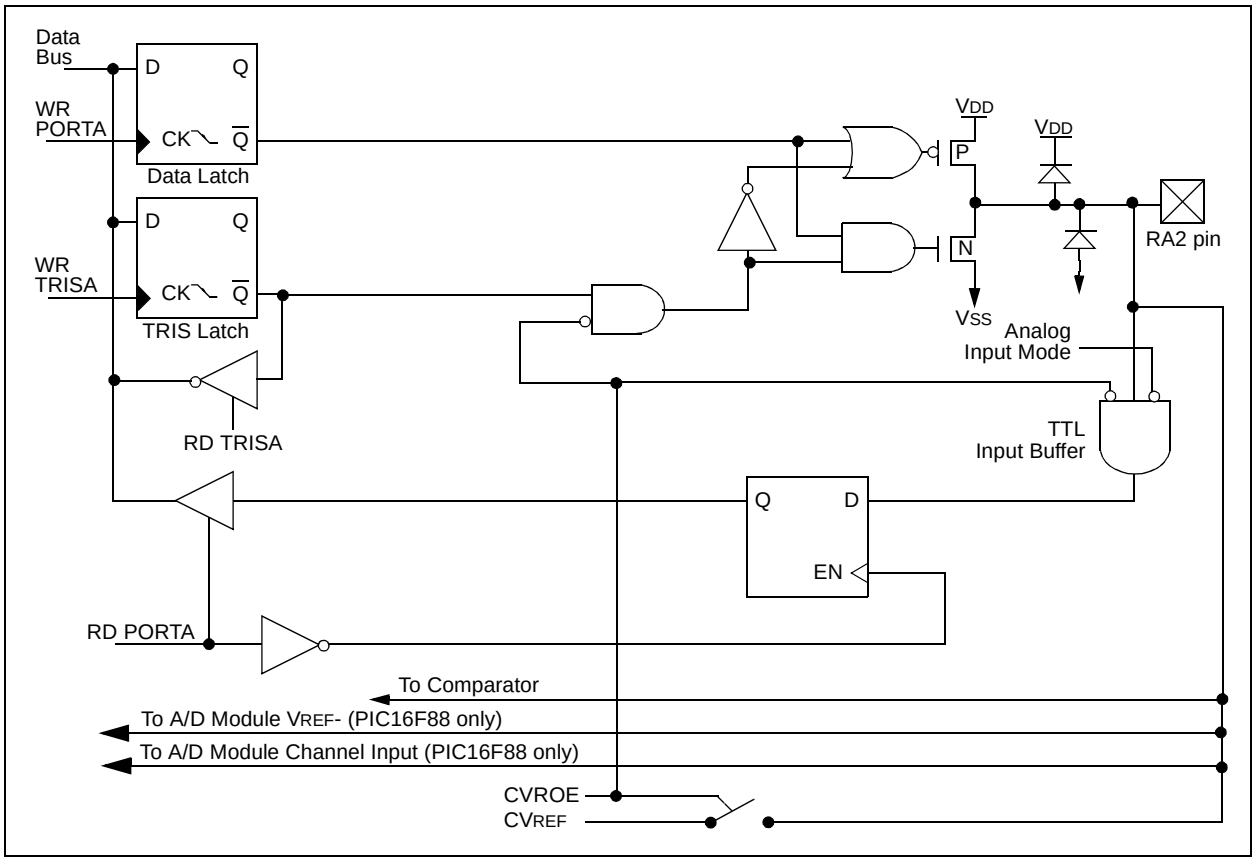


FIGURE 5-3: BLOCK DIAGRAM OF RA2/AN2/CVREF/VREF- PIN



6.0 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

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

Figure 6-1 is a block diagram of the Timer0 module and the prescaler shared with the WDT.

6.1 Timer0 Operation

Timer0 operation is controlled through the OPTION_REG register (see Register 2-2). Timer mode is selected by clearing bit T0CS (OPTION_REG<5>). In Timer mode, the Timer0 module will increment every instruction cycle (without prescaler). If the TMR0 register 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.

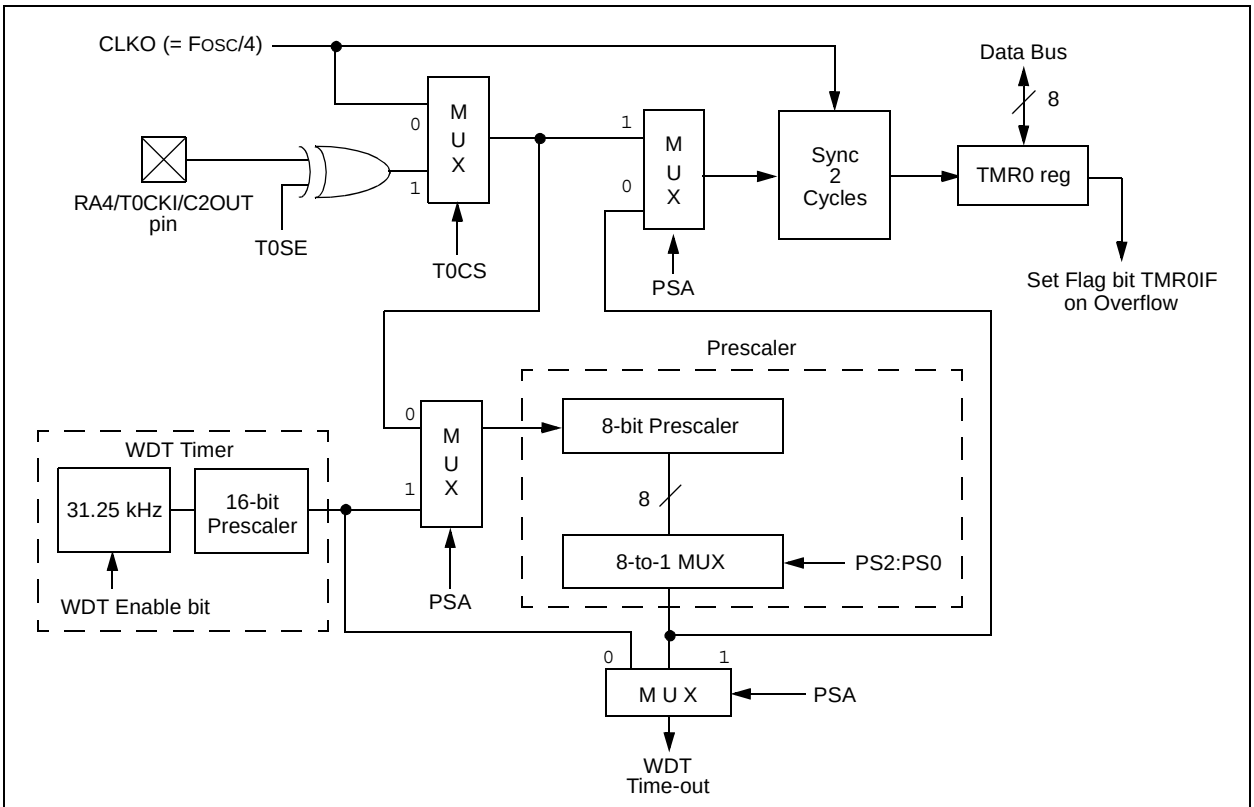
Counter mode is selected by setting bit T0CS (OPTION_REG<5>). In Counter mode, Timer0 will increment, either on every rising or falling edge of pin RA4/T0CKI/C2OUT. The incrementing edge is determined by the Timer0 Source Edge Select bit, T0SE (OPTION_REG<4>). Clearing bit T0SE selects the rising edge. Restrictions on the external clock input are discussed in detail in **Section 6.3 "Using Timer0 with an External Clock"**.

The prescaler is mutually, exclusively shared between the Timer0 module and the Watchdog Timer. The prescaler is not readable or writable. **Section 6.4 "Prescaler"** details the operation of the prescaler.

6.2 Timer0 Interrupt

The TMR0 interrupt is generated when the TMR0 register overflows from FFh to 00h. This overflow sets bit TMR0IF (INTCON<2>). The interrupt can be masked by clearing bit TMR0IE (INTCON<5>). Bit TMR0IF must be cleared in software by the Timer0 module Interrupt Service Routine before re-enabling this interrupt. The TMR0 interrupt cannot awaken the processor from Sleep, since the timer is shut off during Sleep.

FIGURE 6-1: BLOCK DIAGRAM OF THE TIMER0/WDT PRESCALER



Note: T0CS, T0SE, PSA and PS2:PS0 bits are (OPTION_REG<5:0>).

7.8 Resetting Timer1 Using a CCP Trigger Output

If the CCP1 module is configured in Compare mode to generate a "special event trigger" signal (CCP1M3:CCP1M0 = 1011), the signal will reset Timer1 and start an A/D conversion (if the A/D module is enabled).

Note:	The special event triggers from the CCP1 module will not set interrupt flag bit, TMR1IF (PIR1<0>).

Timer1 must be configured for either Timer or Synchronized Counter mode to take advantage of this feature. If Timer1 is running in Asynchronous Counter mode, this Reset operation may not work.

In the event that a write to Timer1 coincides with a special event trigger from CCP1, the write will take precedence.

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

7.9 Resetting Timer1 Register Pair (TMR1H, TMR1L)

TMR1H and TMR1L registers are not reset to 00h on a POR, or any other Reset, except by the CCP1 special event triggers.

T1CON register is reset to 00h on a Power-on Reset or a Brown-out Reset, which shuts off the timer and leaves a 1:1 prescale. In all other Resets, the register is unaffected.

7.10 Timer1 Prescaler

The prescaler counter is cleared on writes to the TMR1H or TMR1L registers.

7.11 Using Timer1 as a Real-Time Clock

Adding an external LP oscillator to Timer1 (such as the one described in **Section 7.6 "Timer1 Oscillator"**) gives users the option to include RTC functionality to their applications. This is accomplished with an inexpensive watch crystal to provide an accurate time base and several lines of application code to calculate the time. When operating in Sleep mode and using a battery or supercapacitor as a power source, it can completely eliminate the need for a separate RTC device and battery backup.

The application code routine, RTCisr, shown in Example 7-3, demonstrates a simple method to increment a counter at one-second intervals using an Interrupt Service Routine. Incrementing the TMR1 register pair to overflow triggers the interrupt and calls the routine, which increments the seconds counter by one; additional counters for minutes and hours are incremented as the previous counter overflows.

Since the register pair is 16 bits wide, counting up to overflow the register directly from a 32.768 kHz clock would take 2 seconds. To force the overflow at the required one-second intervals, it is necessary to preload it; the simplest method is to set the MSb of TMR1H with a BSF instruction. Note that the TMR1L register is never preloaded or altered; doing so may introduce cumulative error over many cycles.

For this method to be accurate, Timer1 must operate in Asynchronous mode and the Timer1 overflow interrupt must be enabled (PIE1<0> = 1), as shown in the routine, RTCinit. The Timer1 oscillator must also be enabled and running at all times.

10.3.2 MASTER MODE OPERATION

Master mode operation is supported in firmware using interrupt generation on the detection of the Start and Stop conditions. The Stop (P) and Start (S) bits are cleared from a Reset, or when the SSP module is disabled. The Stop (P) and Start (S) bits will toggle based on the Start and Stop conditions. Control of the I^2C bus may be taken when the P bit is set, or the bus is Idle and both the S and P bits are clear.

In Master mode operation, the SCL and SDA lines are manipulated in firmware by clearing the corresponding TRISB<4,1> bit(s). The output level is always low, irrespective of the value(s) in PORTB<4,1>. So, when transmitting data, a '1' data bit must have the TRISB<1> bit set (input) and a '0' data bit must have the TRISB<1> bit cleared (output). The same scenario is true for the SCL line with the TRISB<4> bit. Pull-up resistors must be provided externally to the SCL and SDA pins for proper operation of the I^2C module.

The following events will cause the SSP Interrupt Flag bit, SSPIF, to be set (SSP Interrupt if enabled):

- Start condition
- Stop condition
- Data transfer byte transmitted/received

Master mode operation can be done with either the Slave mode Idle (SSPM3:SSPM0 = 1011), or with the Slave mode active. When both Master mode operation and Slave modes are used, the software needs to differentiate the source(s) of the interrupt.

For more information on Master mode operation, see Application Note *AN554, "Software Implementation of I^2C™ Bus Master"*.

10.3.3 MULTI-MASTER MODE OPERATION

In Multi-Master mode operation, the interrupt generation on the detection of the Start and Stop conditions allows the determination of when the bus is free. The Stop (P) and Start (S) bits are cleared from a Reset, or when the SSP module is disabled. The Stop (P) and Start (S) bits will toggle based on the Start and Stop conditions. Control of the I^2C bus may be taken when bit P (SSPSTAT<4>) is set, or the bus is Idle and both the S and P bits clear. When the bus is busy, enabling the SSP interrupt will generate the interrupt when the Stop condition occurs.

In Multi-Master mode operation, the SDA line must be monitored to see if the signal level is the expected output level. This check only needs to be done when a high level is output. If a high level is expected and a low level is present, the device needs to release the SDA and SCL lines (set TRISB<4,1>). There are two stages where this arbitration can be lost:

- Address Transfer
- Data Transfer

When the slave logic is enabled, the slave device continues to receive. If arbitration was lost during the address transfer stage, communication to the device may be in progress. If addressed, an $\overline{ACK}$ pulse will be generated. If arbitration was lost during the data transfer stage, the device will need to retransfer the data at a later time.

For more information on Multi-Master mode operation, see Application Note *AN578, "Use of the SSP Module in the of I^2C™ Multi-Master Environment"*.

TABLE 10-3: REGISTERS ASSOCIATED WITH I^2C™ OPERATION

Address	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
0Bh, 8Bh, 10Bh,18Bh	INTCON	GIE	PEIE	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
0Ch	PIR1	—	ADIF(1)	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	-000 0000	-000 0000
8Ch	PIE1	—	ADIE(1)	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	-000 0000	-000 0000
13h	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								xxxx xxxx	uuuu uuuu
93h	SSPADD	Synchronous Serial Port (I^2C mode) Address Register								0000 0000	0000 0000
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000
94h	SSPSTAT	SMP(2)	CKE(2)	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF	0000 0000	0000 0000
86h	TRISB	PORTB Data Direction Register								1111 1111	1111 1111

Legend: x = unknown, u = unchanged, - = unimplemented locations read as '0'.
Shaded cells are not used by SSP module in SPI mode.

Note 1: This bit is only implemented on the PIC16F88. The bit will read '0' on the PIC16F87.

2: Maintain these bits clear in I^2C™ mode.

The ADRESH:ADRESL registers contain the result of the A/D conversion. When the A/D conversion is complete, the result is loaded into the A/D Result register pair, the GO/$\overline{\text{DONE}}$ bit (ADCON0<2>) is cleared and A/D Interrupt Flag bit, ADIF, is set. The block diagram of the A/D module is shown in Figure 12-1.

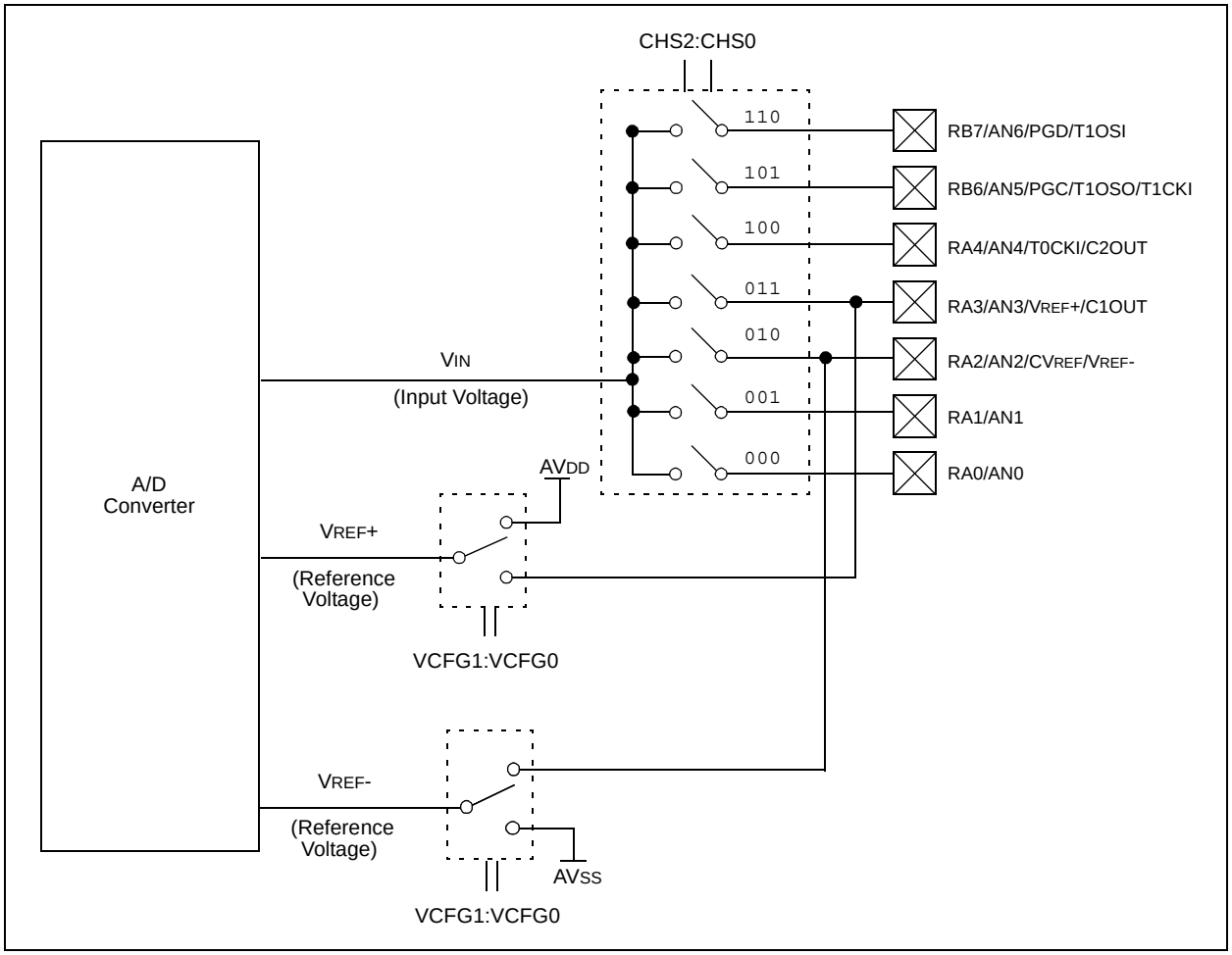
After the A/D module has been configured as desired, the selected channel must be acquired before the conversion is started. The analog input channels must have their corresponding TRIS bits selected as inputs.

To determine sample time, see **Section 12.1 "A/D Acquisition Requirements"**. After this sample time has elapsed, the A/D conversion can be started.

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

1. Configure the A/D module:
   - Configure analog/digital I/O (ANSEL)
   - Configure voltage reference (ADCON1)
   - Select A/D input channel (ADCON0)
   - Select A/D conversion clock (ADCON0)
   - Turn on A/D module (ADCON0)
2. Configure A/D interrupt (if desired):
   - Clear ADIF bit
   - Set ADIE bit
   - SET PEIE bit
   - Set GIE bit
3. Wait the required acquisition time.
4. Start conversion:
   - Set GO/$\overline{\text{DONE}}$ bit (ADCON0)
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 TAD. A minimum wait of 2 TAD is required before the next acquisition starts.

FIGURE 12-1: A/D BLOCK DIAGRAM



12.2 Selecting the A/D Conversion Clock

The A/D conversion time per bit is defined as TAD. The A/D conversion requires 9.0 TAD per 10-bit conversion. The source of the A/D conversion clock is software selectable. The seven possible options for TAD are:

- 2 TOSC
- 4 TOSC
- 8 TOSC
- 16 TOSC
- 32 TOSC
- 64 TOSC
- Internal A/D module RC oscillator (2-6 μs)

For correct A/D conversions, the A/D conversion clock (TAD) must be selected to ensure a minimum TAD time as small as possible, but no less than 1.6 μs and not greater than 6.4 μs.

Table 12-1 shows the resultant TAD times derived from the device operating frequencies and the A/D clock source selected.

12.3 Operation in Power-Managed Modes

The selection of the automatic acquisition time and A/D conversion clock is determined in part by the clock source and frequency while in a power-managed mode.

If the A/D is expected to operate while the device is in a power-managed mode, the ADCS2:ADCS0 bits in ADCON0 and ADCON1 should be updated in accordance with the power-managed mode clock that will be used. After the power-managed mode is entered (either of the power-managed Run modes), an A/D acquisition or conversion may be started. Once an acquisition or conversion is started, the device should continue to be clocked by the same power-managed mode clock source until the conversion has been completed.

If the power-managed mode clock frequency is less than 1 MHz, the A/D RC clock source should be selected.

TABLE 12-1: TAD VS. MAXIMUM DEVICE OPERATING FREQUENCIES – STANDARD DEVICES (C)

AD Clock Source (TAD)			Maximum Device Frequency
Operation	**ADCS<2>**	**ADCS<1:0>**	**Max.**
2 TOSC	0	00	1.25 MHz
4 TOSC	1	00	2.5 MHz
8 TOSC	0	01	5 MHz
16 TOSC	1	01	10 MHz
32 TOSC	0	10	20 MHz
64 TOSC	1	10	20 MHz
RC(1,2,3)	x	11	**(Note 1)**

Note 1: The RC source has a typical TAD time of 4 μs, but can vary between 2-6 μs.

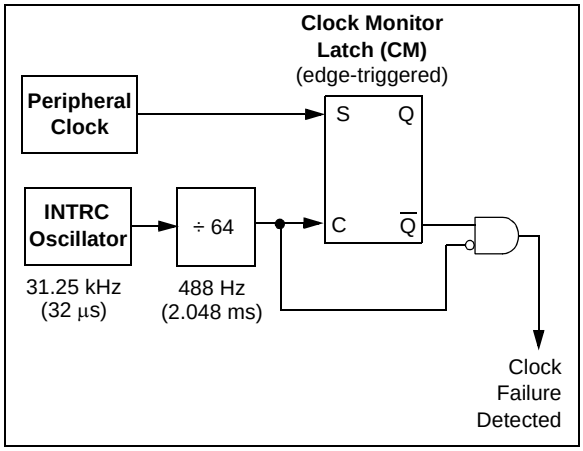
2: When the device frequencies are greater than 1 MHz, the RC A/D conversion clock source is only recommended for Sleep operation.

3: For extended voltage devices (LF), please refer to **Section 18.0 "Electrical Characteristics"**.

15.12.4 FAIL-SAFE OPTION

The Fail-Safe Clock Monitor (FSCM) is designed to allow the device to continue to operate even in the event of an oscillator failure.

FIGURE 15-10: FSCM BLOCK DIAGRAM



The FSCM function is enabled by setting the FCMEN bit in Configuration Word 2.

In the event of an oscillator failure, the FSCM will generate an oscillator fail interrupt and will switch the system clock over to the internal oscillator. The system will continue to come from the internal oscillator until the fail-safe condition is exited. The fail-safe condition is exited with either a Reset, the execution of a SLEEP instruction or a write to the OSCCON register.

The frequency of the internal oscillator will depend upon the value contained in the IRCF bits. Another clock source can be selected via the IRCF and the SCS bits of the OSCCON register.

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.

On the rising edge of the postscaled clock, the monitoring latch (CM = 0) will be cleared. On a falling edge of the primary or secondary system clock, the monitoring latch will be set (CM = 1). In the event that a falling edge of the postscaled clock occurs and the monitoring latch is not set, a clock failure has been detected.

While in Fail-Safe mode, a Reset will exit the fail-safe condition. If the primary clock source is configured for a crystal, the OST timer will wait for the 1024 clock cycles for the OST time-out and the device will continue running from the internal oscillator until the OST is complete. A SLEEP instruction, or a write to the SCS bits (where SCS bits do not = 00), can be performed to put the device into a low-power mode.

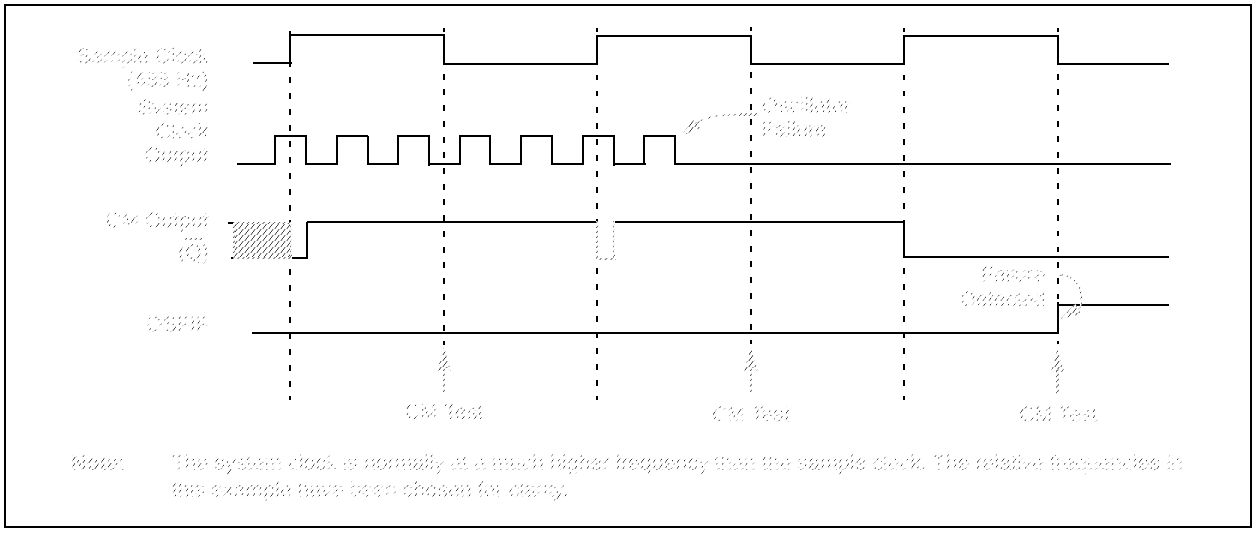
Note:	Two-Speed Start-up mode is automatically enabled when the fail-safe option is enabled.

If Reset occurs while in Fail-Safe mode and the primary clock source is EC or RC, then the device will immediately switch back to EC or RC mode.

15.12.4.1 Fail-Safe in Low-Power Mode

A write to the OSCCON register, or SLEEP instruction, will end the fail-safe condition. The system clock will default to the source selected by the SCS bits, which is either T1OSC, INTRC or none (Sleep mode). However, the FSCM will continue to monitor the system clock. If the secondary clock fails, the device will immediately switch to the internal oscillator clock. If OSFIE is set, an interrupt will be generated.

FIGURE 15-11: 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.

15.18 Low-Voltage ICSP Programming

The LVP bit of the Configuration Word enables Low-Voltage ICSP Programming. This mode allows the microcontroller to be programmed via ICSP using a VDD source in the operating voltage range. This only means that VPP does not have to be brought to VIHH, but can instead be left at the normal operating voltage. In this mode, the RB3/PGM pin is dedicated to the programming function and ceases to be a general purpose I/O pin.

If Low-Voltage Programming mode is not used, the LVP bit can be programmed to a '0' and RB3/PGM becomes a digital I/O pin. However, the LVP bit may only be programmed when Programming mode is entered with VIHH on MCLR. The LVP bit can only be changed when using high voltage on MCLR.

It should be noted that once the LVP bit is programmed to '0', only the High-Voltage Programming mode is available and only this mode can be used to program the device.

When using Low-Voltage ICSP, the part must be supplied at 4.5V to 5.5V if a bulk erase will be executed. This includes reprogramming of the code-protect bits from an ON state to an OFF state. For all other cases of Low-Voltage ICSP, the part may be programmed at the normal operating voltage. This means calibration values, unique user IDs or user code can be reprogrammed or added.

The following LVP steps assume the LVP bit is set in the Configuration register.

1. Apply VDD to the VDD pin.
2. Drive MCLR low.
3. Apply VDD to the RB3/PGM pin.
4. Apply VDD to the MCLR pin.
5. Follow with the associated programming steps.

Note 1: The High-Voltage Programming mode is always available, regardless of the state of the LVP bit, by applying VIHH to the MCLR pin.

2: While in Low-Voltage ICSP mode (LVP = 1), the RB3 pin can no longer be used as a general purpose I/O pin.

3: When using Low-Voltage ICSP Programming (LVP) and the pull-ups on PORTB are enabled, bit 3 in the TRISB register must be cleared to disable the pull-up on RB3 and ensure the proper operation of the device.

4: RB3 should not be allowed to float if LVP is enabled. An external pull-down device should be used to default the device to normal operating mode. If RB3 floats high, the PIC16F87/88 devices will enter Programming mode.

5: LVP mode is enabled by default on all devices shipped from Microchip. It can be disabled by clearing the LVP bit in the CONFIG1 register.

6: Disabling LVP will provide maximum compatibility to other PIC16CXXX devices.

16.2 Instruction Descriptions

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

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

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

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

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

BSF	Bit Set f
Syntax:	[*label*] BSF f,b
Operands:	$0 \le f \le 127$ $0 \le b \le 7$
Operation:	$1 \rightarrow$ (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' = 0, the next instruction is executed. If bit 'b' = 1, then the next instruction is discarded and a NOP is executed instead, making this a 2 TCY instruction.

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

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

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

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

CLRWDT	Clear Watchdog Timer
Syntax:	[*label*] CLRWDT
Operands:	None
Operation:	00h → WDT, 0 → WDT prescaler, 1 → $\overline{TO}$, 1 → $\overline{PD}$
Status Affected:	$\overline{TO}$, $\overline{PD}$
Description:	CLRWDT instruction resets the Watchdog Timer. It also resets the prescaler of the WDT. Status bits $\overline{TO}$ and $\overline{PD}$ are set.

18.1 DC Characteristics: Supply Voltage PIC16F87/88 (Industrial, Extended) PIC16LF87/88 (Industrial)

PIC16LF87/88 (Industrial)			**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial				
PIC16F87/88 (Industrial, Extended)			**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended				
Param No.	**Symbol**	**Characteristic**	**Min**	**Typ**	**Max**	**Units**	**Conditions**
	VDD	**Supply Voltage**					
D001		PIC16LF87/88	2.0	—	5.5	V	HS, XT, RC and LP Oscillator mode
D001		PIC16F87/88	4.0	—	5.5	V	
D002	VDR	**RAM Data Retention Voltage(1)**	1.5	—	—	V	
D003	VPOR	**VDD Start Voltage** to ensure internal Power-on Reset signal	—	—	0.7	V	See **Section 15.4 "Power-on Reset (POR)"** for details
D004	SVDD	**VDD Rise Rate** to ensure internal Power-on Reset signal	0.05	—	—	V/ms	See **Section 15.4 "Power-on Reset (POR)"** for details
	VBOR	**Brown-out Reset Voltage**					
D005		PIC16LF87/88	3.65	—	4.35	V	
D005		PIC16F87/88	3.65	—	4.35	V	FMAX = 14 MHz(2)

Legend: Shading of rows is to assist in readability of the table.

Note 1: This is the limit to which VDD can be lowered in Sleep mode, or during a device Reset, without losing RAM data.

2: When BOR is enabled, the device will operate correctly until the VBOR voltage trip point is reached.

18.2 DC Characteristics: Power-Down and Supply Current PIC16F87/88 (Industrial, Extended) PIC16LF87/88 (Industrial) (Continued)

PIC16LF87/88 (Industrial)		**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial					
PIC16F87/88 (Industrial, Extended)		**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended					
Param No.	**Device**	**Typ**	**Max**	**Units**	**Conditions**		
	Supply Current (IDD)(2,3)						
	PIC16LF87/88	8	20	µA	-40°C	VDD = 2.0V	FOSC = 31.25 kHz (**RC_RUN** mode, Internal RC Oscillator)
		7	15	µA	+25°C		
		7	15	µA	+85°C		
	PIC16LF87/88	16	30	µA	-40°C	VDD = 3.0V	
		14	25	µA	+25°C		
		14	25	µA	+85°C		
	All devices	32	40	µA	-40°C	VDD = 5.0V	
		29	35	µA	+25°C		
		29	35	µA	+85°C		
	Extended devices	35	45	µA	+125°C		
	PIC16LF87/88	132	160	µA	-40°C	VDD = 2.0V	FOSC = 1 MHz (**RC_RUN** mode, Internal RC Oscillator)
		126	155	µA	+25°C		
		126	155	µA	+85°C		
	PIC16LF87/88	260	310	µA	-40°C	VDD = 3.0V	
		230	300	µA	+25°C		
		230	300	µA	+85°C		
	All devices	560	690	µA	-40°C	VDD = 5.0V	
		500	650	µA	+25°C		
		500	650	µA	+85°C		
	Extended devices	570	710	µA	+125°C		

Legend: Shading of rows is to assist in readability of the table.

Note 1: 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 or VSS and all features that add delta current disabled (such as WDT, Timer1 Oscillator, BOR, etc.).

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

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;
$\overline{\text{MCLR}}$ = VDD; WDT enabled/disabled as specified.

3: For RC oscillator configurations, current through REXT is not included. The current through the resistor can be estimated by the formula Ir = VDD/2REXT (mA) with REXT in kΩ.

TABLE 18-1: COMPARATOR SPECIFICATIONS

Operating Conditions: 3.0V < V_{DD} < 5.5V, -40°C < T_A < +85°C, unless otherwise stated							
Param No.	**Sym**	**Characteristics**	**Min**	**Typ**	**Max**	**Units**	**Comments**
D300	V_{IOFF}	Input Offset Voltage	—	±5.0	±10	mV	
D301	V_{ICM}	Input Common Mode Voltage*	0	—	V_{DD} – 1.5	V	
D302	CMRR	Common Mode Rejection Ratio*	55	—	—	dB	
300 300A	T_{RESP}	Response Time(1)*	—	150	400 600	ns ns	PIC16F87/88 PIC16LF87/88
301	T_{MC2OV}	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 (V_{DD} – 1.5)/2 while the other input transitions from V_{SS} to V_{DD}.

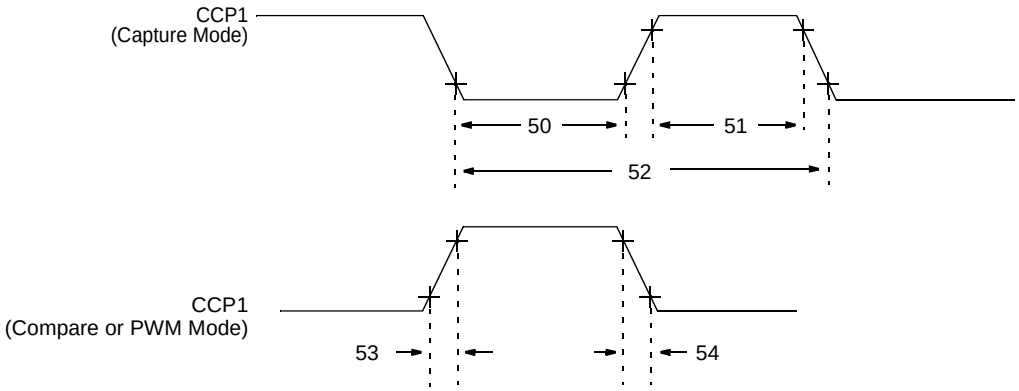
TABLE 18-2: VOLTAGE REFERENCE SPECIFICATIONS

Operating Conditions: 3.0V < V_{DD} < 5.5V, -40°C < T_A < +85°C, unless otherwise stated							
Spec No.	**Sym**	**Characteristics**	**Min**	**Typ**	**Max**	**Units**	**Comments**
D310	V_{RES}	Resolution	V_{DD}/24	—	V_{DD}/32	LSb	
D311	V_{RAA}	Absolute Accuracy	— —	— —	1/2 1/2	LSb LSb	Low Range (CVRR = 1) High Range (CVRR = 0)
D312	V_{RUR}	Unit Resistor Value (R)*	—	2k	—	Ω	
310	T_{SET}	Settling Time(1)*	—	—	10	µs	

* These parameters are characterized but not tested.

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

FIGURE 18-9: CAPTURE/COMPARE/PWM TIMINGS (CCP1)



Note: Refer to Figure 18-3 for load conditions.

TABLE 18-7: CAPTURE/COMPARE/PWM REQUIREMENTS (CCP1)

Param No.	Symbol	Characteristic			Min	Typ†	Max	Units	Conditions
50*	TccL	CCP1 Input Low Time	No Prescaler		0.5 TCY + 20	—	—	ns	
			With Prescaler	PIC16**F**87/88	10	—	—	ns	
				PIC16**LF**87/88	20	—	—	ns	
51*	TccH	CCP1 Input High Time	No Prescaler		0.5 TCY + 20	—	—	ns	
			With Prescaler	PIC16**F**87/88	10	—	—	ns	
				PIC16**LF**87/88	20	—	—	ns	
52*	TccP	CCP1 Input Period			$\frac{3\ T_{CY} + 40}{N}$	—	—	ns	N = prescale value (1, 4 or 16)
53*	TccR	CCP1 Output Rise Time		PIC16**F**87/88	—	10	25	ns	
				PIC16**LF**87/88	—	25	50	ns	
54*	TccF	CCP1 Output Fall Time		PIC16**F**87/88	—	10	25	ns	
				PIC16**LF**87/88	—	25	45	ns	

* These parameters are characterized but not tested.

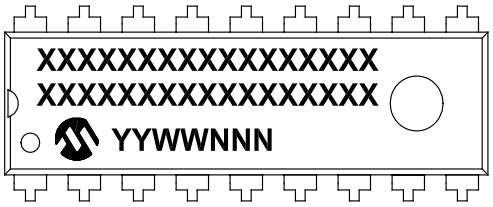
† Data in “Typ” column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

NOTES:

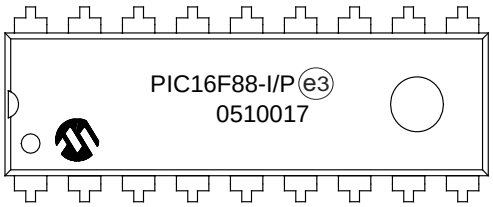
20.0 PACKAGING INFORMATION

20.1 Package Marking Information

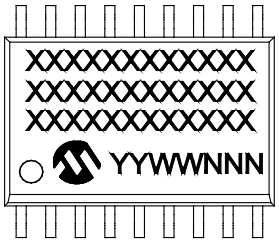
18-Lead PDIP (300 mil)



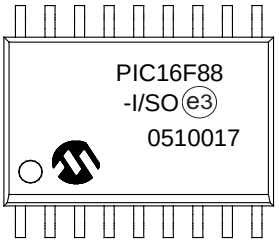
Example



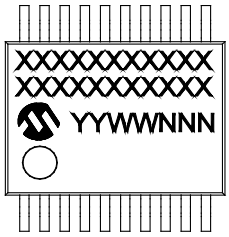
18-Lead SOIC (7.50 mm)



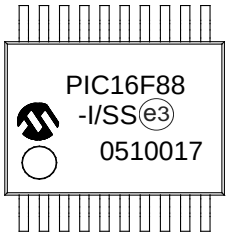
Example



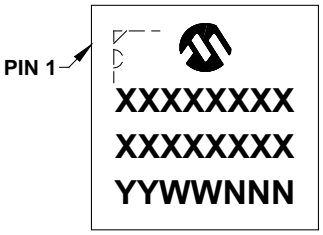
20-Lead SSOP (5.30 mm)



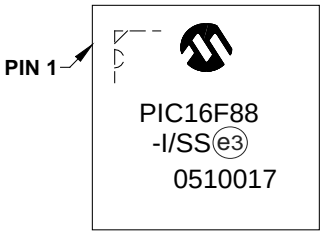
Example



28-Lead QFN (6x6 mm)



Example



Legend:	XX...X	Customer-specific information
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week '01')
	NNN	Alphanumeric traceability code
	(e3)	Pb-free JEDEC designator for Matte Tin (Sn)
	*	This package is Pb-free. The Pb-free JEDEC designator (e3) can be found on the outer packaging for this package.

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

T

THE MICROCHIP WEB SITE

Microchip provides online support via our WWW site at www.microchip.com. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at www.microchip.com, click on Customer Change Notification and follow the registration instructions.

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or field application engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at: http://support.microchip.com