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	41.667MHz
Connectivity	Ethernet, I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	55
Program Memory Size	128KB (64K x 16)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	3808 x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 3.6V
Data Converters	A/D 15x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	80-TQFP
Supplier Device Package	80-TQFP (12x12)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f87j60t-i-pt

TO OUR VALUED CUSTOMERS

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

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

Most Current Data Sheet

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

http://www.microchip.com

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

Errata

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

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

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

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

Customer Notification System

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

TABLE 6-5: REGISTER FILE SUMMARY (PIC18F97J60 FAMILY) (CONTINUED)

File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Values on POR, BOR	Details on Page:
ESTAT	—	BUFER	—	r	—	RXBUSY	TXABRT	PHYRDY	-0-0 -000	73, 228
EIE	—	PKTIE	DMAIE	LINKIE	TXIE	—	TXERIE	RXERIE	-000 0-00	73, 240
EDMACSH	DMA Checksum Register High Byte								0000 0000	73, 265
EDMACSL	DMA Checksum Register Low Byte								0000 0000	73, 265
EDMADSTH	—	—	—	DMA Destination Register High Byte					---0 0000	73, 265
EDMADSTL	DMA Destination Register Low Byte								0000 0000	73, 265
EDMANDH	—	—	—	DMA End Register High Byte					---0 0000	73, 265
EDMANDL	DMA End Register Low Byte								0000 0000	73, 265
EDMASTH	—	—	—	DMA Start Register High Byte					---0 0000	73, 265
EDMASTL	DMA Start Register Low Byte								0000 0000	73, 265
ERXWRPTH	—	—	—	Receive Buffer Write Pointer High Byte					---0 0000	73, 225
ERXWRPTL	Receive Buffer Write Pointer Low Byte								0000 0000	73, 225
ERXRDPTH	—	—	—	Receive Buffer Read Pointer High Byte					---0 0101	73, 225
ERXRDPTL	Receive Buffer Read Pointer Low Byte								1111 1010	73, 225
ERXNDH	—	—	—	Receive End Register High Byte					---1 1111	73, 225
ERXNDL	Receive End Register Low Byte								1111 1111	73, 225
ERXSTH	—	—	—	Receive Start Register High Byte					---0 0101	73, 225
ERXSTL	Receive Start Register Low Byte								1111 1010	73, 225
ETXNDH	—	—	—	Transmit End Register High Byte					---0 0000	74, 226
ETXNDL	Transmit End Register Low Byte								0000 0000	74, 226
ETXSTH	—	—	—	Transmit Start Register High Byte					---0 0000	74, 226
ETXSTL	Transmit Start Register Low Byte								0000 0000	74, 226
EWRPTH	—	—	—	Buffer Write Pointer High Byte					---0 0000	74, 223
EWRPTL	Buffer Write Pointer Low Byte								0000 0000	74, 223
EPKTCNT	Ethernet Packet Count Register								0000 0000	74, 252
ERXFCON	UCEN	ANDOR	CRCEN	PMEN	MPEN	HTEN	MCEN	BCEN	1010 0001	74, 260
EPMOH	—	—	—	Pattern Match Offset Register High Byte					---0 0000	74, 263
EPMOL	Pattern Match Offset Register Low Byte								0000 0000	74, 263
EPMCSH	Pattern Match Checksum Register High Byte								0000 0000	74, 263
EPMCSL	Pattern Match Checksum Register Low Byte								0000 0000	74, 263
EPMM7	Pattern Match Mask Register Byte 7								0000 0000	74, 263
EPMM6	Pattern Match Mask Register Byte 6								0000 0000	74, 263
EPMM5	Pattern Match Mask Register Byte 5								0000 0000	74, 263
EPMM4	Pattern Match Mask Register Byte 4								0000 0000	74, 263
EPMM3	Pattern Match Mask Register Byte 3								0000 0000	74, 263
EPMM2	Pattern Match Mask Register Byte 2								0000 0000	74, 263
EPMM1	Pattern Match Mask Register Byte 1								0000 0000	74, 263
EPMM0	Pattern Match Mask Register Byte 0								0000 0000	74, 263

Legend: x = unknown; u = unchanged; - = unimplemented, read as '0'; q = value depends on condition; r = reserved bit, do not modify. Shaded cells are unimplemented, read as '0'.

Note 1: Bit 7 and bit 6 are cleared by user software or by a POR.

2: Bit 21 of the PC is only available in Serial Programming modes.

3: Reset value is '0' when Two-Speed Start-up is enabled and '1' if disabled.

4: Alternate names and definitions for these bits when the MSSP module is operating in I^2C™ Slave mode.

5: These bits and/or registers are only available in 100-pin devices; otherwise, they are unimplemented and read as '0'. Reset values shown apply only to 100-pin devices.

6: These bits and/or registers are only available in 80-pin and 100-pin devices. In 64-pin devices, they are unimplemented and read as '0'. Reset values are shown for 100-pin devices.

7: In Microcontroller mode, the bits in this register are unwritable and read as '0'.

8: PLLEN is only available when either ECPLL or HSPLL Oscillator mode is selected; otherwise, read as '0'.

9: Implemented in 100-pin devices in Microcontroller mode only.

6.3.6 STATUS REGISTER

The STATUS register, shown in Register 6-3, contains the arithmetic status of the ALU. The STATUS register can be the operand for any instruction, as with any other register. If the STATUS register is the destination for an instruction that affects the Z, DC, C, OV or N bits, then the write to these five bits is disabled.

These bits are set or cleared according to the device logic. Therefore, the result of an instruction with the STATUS register as destination may be different than intended. For example, `CLRF STATUS` will set the Z bit but leave the other bits unchanged. The STATUS register then reads back as '`000u u1uu`'. It is recommended, therefore, that only `BCF`, `BSF`, `SWAPF`, `MOVFF` and `MOVWF` instructions are used to alter the STATUS register because these instructions do not affect the Z, C, DC, OV or N bits in the STATUS register.

For other instructions not affecting any Status bits, see the instruction set summaries in Table 26-2 and Table 26-3.

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

REGISTER 6-3: STATUS REGISTER

U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	—	N	OV	Z	DC(1)	C(2)
bit 7							bit 0

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

bit 7-5 **Unimplemented:** Read as '0'

bit 4 **N:** Negative bit
This bit is used for signed arithmetic (2's complement). It indicates whether the result was negative (ALU MSb = 1).
1 = Result was negative
0 = Result was positive

bit 3 **OV:** Overflow bit
This bit is used for signed arithmetic (2's complement). It indicates an overflow of the 7-bit magnitude which causes the sign bit (bit 7 of the result) to change state.
1 = Overflow occurred for signed arithmetic (in this arithmetic operation)
0 = No overflow occurred

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 non-zero

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

bit 0 **C:** Carry/$\overline{\text{Borrow}}$ bit(2)
For `ADDWF`, `ADDLW`, `SUBLW` and `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 $\overline{\text{Borrow}}$, the polarity is reversed. A subtraction is executed by adding the 2's complement of the second operand. For rotate (`RRF`, `RLF`) instructions, this bit is loaded with either bit 4 or bit 3 of the source register.

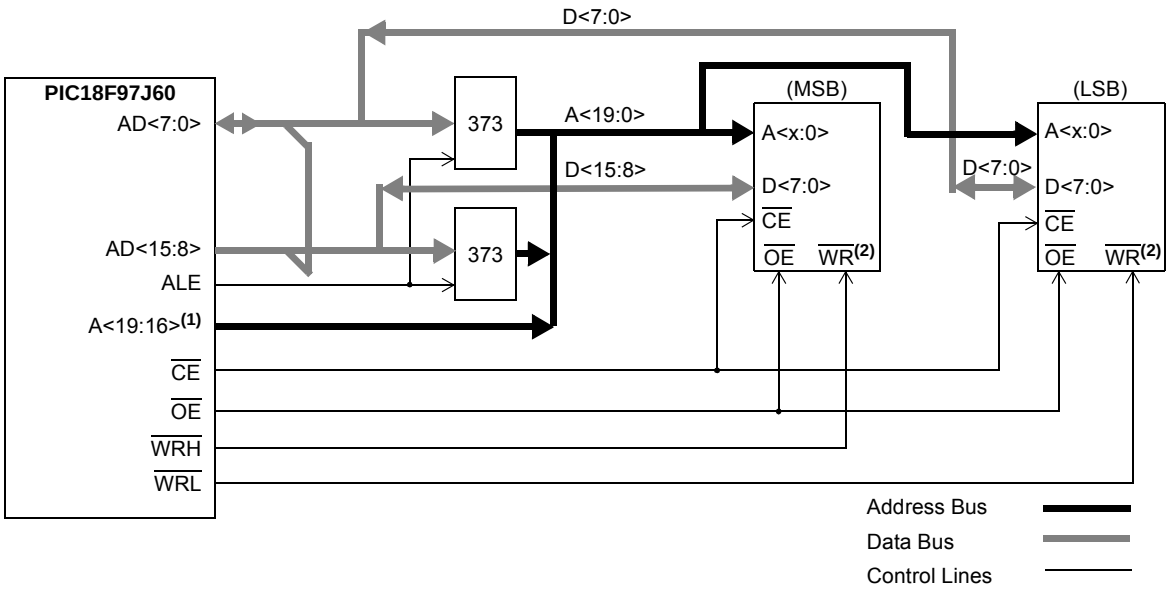
2: For $\overline{\text{Borrow}}$, the polarity is reversed. A subtraction is executed by adding the 2's complement of the second operand. For rotate (`RRF`, `RLF`) instructions, this bit is loaded with either the high or low-order bit of the source register.

8.6.1 16-BIT BYTE WRITE MODE

Figure 8-1 shows an example of 16-Bit Byte Write mode for PIC18F97J60 family devices. This mode is used for two separate 8-bit memories connected for 16-bit operation. This generally includes basic EPROM and Flash devices. It allows table writes to byte-wide external memories.

During a TBLWT instruction cycle, the TABLAT data is presented on the upper and lower bytes of the AD<15:0> bus. The appropriate $\overline{WRH}$ or $\overline{WRL}$ control line is strobed on the LSb of the TBLPTR.

FIGURE 8-1: 16-BIT BYTE WRITE MODE EXAMPLE



Note 1: The upper order address lines are used only for 20-bit address widths.

2: This signal only applies to table writes. See **Section 7.1 "Table Reads and Table Writes"**.

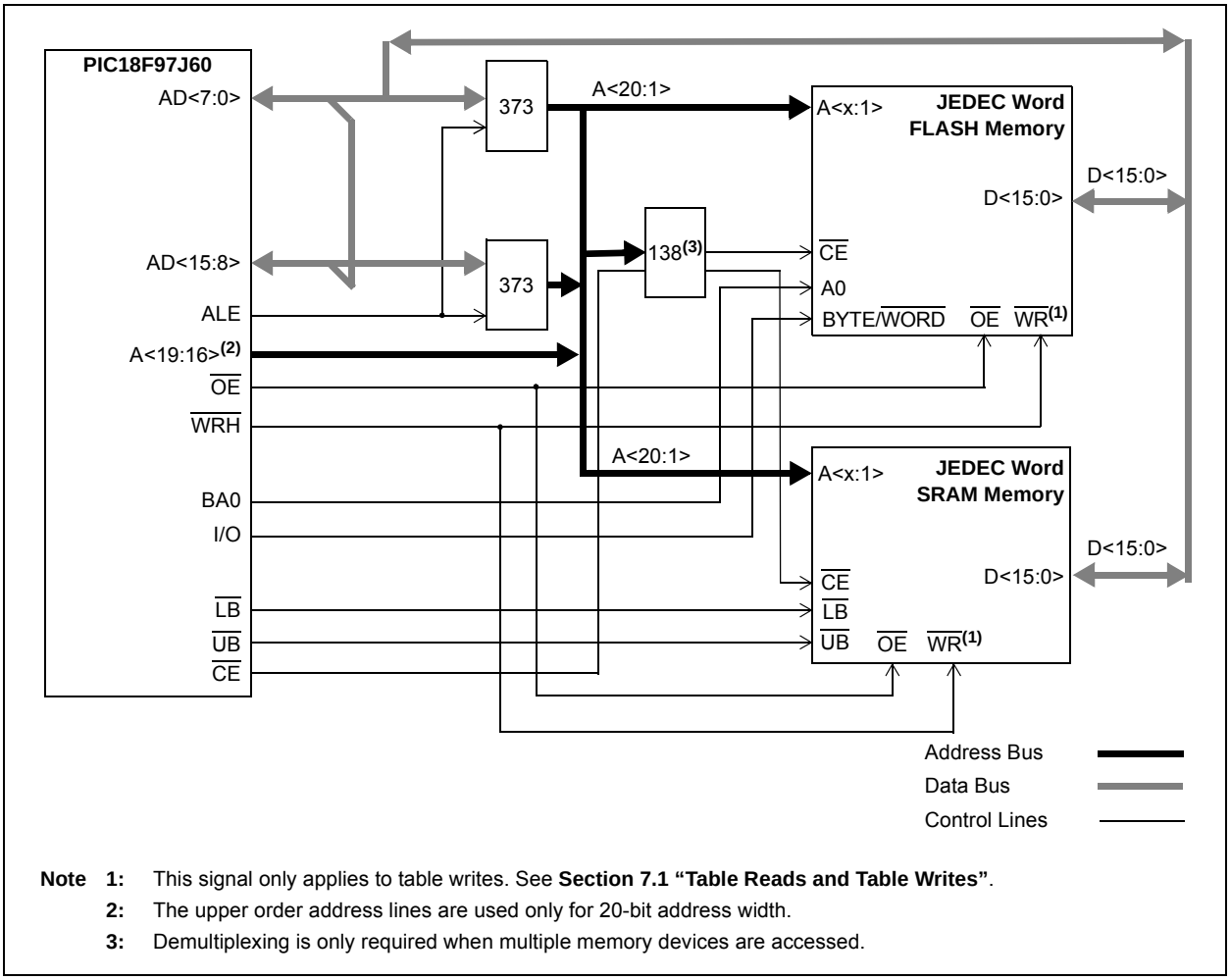
8.6.3 16-BIT BYTE SELECT MODE

Figure 8-3 shows an example of 16-Bit Byte Select mode. This mode allows table write operations to word-wide external memories with byte selection capability. This generally includes both word-wide Flash and SRAM devices.

During a TBLWT cycle, the TABLAT data is presented on the upper and lower byte of the AD<15:0> bus. The $\overline{WRH}$ signal is strobed for each write cycle; the $\overline{WRL}$ pin is not used. The BA0 or $\overline{UB}$/$\overline{LB}$ signals are used to select the byte to be written based on the Least Significant bit of the TBLPTR register.

Flash and SRAM devices use different control signal combinations to implement Byte Select mode. JEDEC standard Flash memories require that a controller I/O port pin be connected to the memory's BYTE/$\overline{WORD}$ pin to provide the select signal. They also use the BA0 signal from the controller as a byte address. JEDEC standard, static RAM memories, on the other hand, use the $\overline{UB}$ or $\overline{LB}$ signals to select the byte.

FIGURE 8-3: 16-BIT BYTE SELECT MODE EXAMPLE



Note 1: This signal only applies to table writes. See **Section 7.1 "Table Reads and Table Writes"**.

2: The upper order address lines are used only for 20-bit address width.

3: Demultiplexing is only required when multiple memory devices are accessed.

Example 9-3 shows the sequence to do a 16 x 16 unsigned multiplication. Equation 9-1 shows the algorithm that is used. The 32-bit result is stored in four registers (RES3:RES0).

EQUATION 9-1: 16 x 16 UNSIGNED MULTIPLICATION ALGORITHM

$$
\begin{aligned}
RES3:RES0 &= ARG1H:ARG1L \bullet ARG2H:ARG2L \\
&= (ARG1H \bullet ARG2H \bullet 2^{16}) + \\
&\quad (ARG1H \bullet ARG2L \bullet 2^{8}) + \\
&\quad (ARG1L \bullet ARG2H \bullet 2^{8}) + \\
&\quad (ARG1L \bullet ARG2L)
\end{aligned}
$$

EXAMPLE 9-3: 16 x 16 UNSIGNED MULTIPLY ROUTINE

```
    MOVF    ARG1L, W
    MULWF   ARG2L           ; ARG1L * ARG2L->
                            ; PRODH:PRODL
    MOVFF   PRODH, RES1     ;
    MOVFF   PRODL, RES0     ;
;
    MOVF    ARG1H, W
    MULWF   ARG2H           ; ARG1H * ARG2H->
                            ; PRODH:PRODL
    MOVFF   PRODH, RES3     ;
    MOVFF   PRODL, RES2     ;
;
    MOVF    ARG1L, W
    MULWF   ARG2H           ; ARG1L * ARG2H->
                            ; PRODH:PRODL
    MOVF    PRODL, W        ;
    ADDWF   RES1, F         ; Add cross
    MOVF    PRODH, W        ; products
    ADDWFC  RES2, F         ;
    CLRF    WREG            ;
    ADDWFC  RES3, F         ;
;
    MOVF    ARG1H, W        ;
    MULWF   ARG2L           ; ARG1H * ARG2L->
                            ; PRODH:PRODL
    MOVF    PRODL, W        ;
    ADDWF   RES1, F         ; Add cross
    MOVF    PRODH, W        ; products
    ADDWFC  RES2, F         ;
    CLRF    WREG            ;
    ADDWFC  RES3, F         ;
```

Example 9-4 shows the sequence to do a 16 x 16 signed multiply. Equation 9-2 shows the algorithm used. The 32-bit result is stored in four registers (RES3:RES0). To account for the sign bits of the arguments, the MSb for each argument pair is tested and the appropriate subtractions are done.

EQUATION 9-2: 16 x 16 SIGNED MULTIPLICATION ALGORITHM

$$
\begin{aligned}
RES3:RES0 &= ARG1H:ARG1L \bullet ARG2H:ARG2L \\
&= (ARG1H \bullet ARG2H \bullet 2^{16}) + \\
&\quad (ARG1H \bullet ARG2L \bullet 2^{8}) + \\
&\quad (ARG1L \bullet ARG2H \bullet 2^{8}) + \\
&\quad (ARG1L \bullet ARG2L) + \\
&\quad (-1 \bullet ARG2H\langle 7\rangle \bullet ARG1H:ARG1L \bullet 2^{16}) + \\
&\quad (-1 \bullet ARG1H\langle 7\rangle \bullet ARG2H:ARG2L \bullet 2^{16})
\end{aligned}
$$

EXAMPLE 9-4: 16 x 16 SIGNED MULTIPLY ROUTINE

```
    MOVF    ARG1L, W
    MULWF   ARG2L           ; ARG1L * ARG2L ->
                            ; PRODH:PRODL
    MOVFF   PRODH, RES1     ;
    MOVFF   PRODL, RES0     ;
;
    MOVF    ARG1H, W
    MULWF   ARG2H           ; ARG1H * ARG2H ->
                            ; PRODH:PRODL
    MOVFF   PRODH, RES3     ;
    MOVFF   PRODL, RES2     ;
;
    MOVF    ARG1L, W
    MULWF   ARG2H           ; ARG1L * ARG2H ->
                            ; PRODH:PRODL
    MOVF    PRODL, W        ;
    ADDWF   RES1, F         ; Add cross
    MOVF    PRODH, W        ; products
    ADDWFC  RES2, F         ;
    CLRF    WREG            ;
    ADDWFC  RES3, F         ;
;
    MOVF    ARG1H, W        ;
    MULWF   ARG2L           ; ARG1H * ARG2L ->
                            ; PRODH:PRODL
    MOVF    PRODL, W        ;
    ADDWF   RES1, F         ; Add cross
    MOVF    PRODH, W        ; products
    ADDWFC  RES2, F         ;
    CLRF    WREG            ;
    ADDWFC  RES3, F         ;
;
    BTFSS   ARG2H, 7        ; ARG2H:ARG2L neg?
    BRA     SIGN_ARG1       ; no, check ARG1
    MOVF    ARG1L, W        ;
    SUBWF   RES2            ;
    MOVF    ARG1H, W        ;
    SUBWFB  RES3
;
SIGN_ARG1
    BTFSS   ARG1H, 7        ; ARG1H:ARG1L neg?
    BRA     CONT_CODE       ; no, done
    MOVF    ARG2L, W        ;
    SUBWF   RES2            ;
    MOVF    ARG2H, W        ;
    SUBWFB  RES3
;
CONT_CODE
   :
```

REGISTER 10-3: INTCON3: INTERRUPT CONTROL REGISTER 3

R/W-1	R/W-1	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
INT2IP	INT1IP	INT3IE	INT2IE	INT1IE	INT3IF	INT2IF	INT1IF
bit 7							bit 0

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

bit 7 **INT2IP:** INT2 External Interrupt Priority bit
1 = High priority
0 = Low priority

bit 6 **INT1IP:** INT1 External Interrupt Priority bit
1 = High priority
0 = Low priority

bit 5 **INT3IE:** INT3 External Interrupt Enable bit
1 = Enables the INT3 external interrupt
0 = Disables the INT3 external interrupt

bit 4 **INT2IE:** INT2 External Interrupt Enable bit
1 = Enables the INT2 external interrupt
0 = Disables the INT2 external interrupt

bit 3 **INT1IE:** INT1 External Interrupt Enable bit
1 = Enables the INT1 external interrupt
0 = Disables the INT1 external interrupt

bit 2 **INT3IF:** INT3 External Interrupt Flag bit
1 = The INT3 external interrupt occurred (must be cleared in software)
0 = The INT3 external interrupt did not occur

bit 1 **INT2IF:** INT2 External Interrupt Flag bit
1 = The INT2 external interrupt occurred (must be cleared in software)
0 = The INT2 external interrupt did not occur

bit 0 **INT1IF:** INT1 External Interrupt Flag bit
1 = The INT1 external interrupt occurred (must be cleared in software)
0 = The INT1 external interrupt did not occur

Note: Interrupt flag bits are set when an interrupt condition occurs regardless of the state of its corresponding enable bit or the Global Interrupt Enable bit. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt. This feature allows for software polling.

15.0 TIMER3 MODULE

The Timer3 timer/counter module incorporates these features:

- Software selectable operation as a 16-bit timer or counter
- Readable and writable 8-bit registers (TMR3H and TMR3L)
- Selectable clock source (internal or external) with device clock or Timer1 oscillator internal options
- Interrupt on overflow
- Module Reset on CCPx/ECCPx Special Event Trigger

A simplified block diagram of the Timer3 module is shown in Figure 15-1. A block diagram of the module's operation in Read/Write mode is shown in Figure 15-2.

The Timer3 module is controlled through the T3CON register (Register 15-1). It also selects the clock source options for the CCPx and ECCPx modules; see **Section 17.1.1 "CCPx/ECCPx Modules and Timer Resources"** for more information.

REGISTER 15-1: T3CON: TIMER3 CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
RD16	T3CCP2	T3CKPS1	T3CKPS0	T3CCP1	$\overline{\text{T3SYNC}}$	TMR3CS	TMR3ON
bit 7							bit 0

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

bit 7 **RD16:** 16-Bit Read/Write Mode Enable bit
1 = Enables register read/write of Timer3 in one 16-bit operation
0 = Enables register read/write of Timer3 in two 8-bit operations

bit 6,3 **T3CCP<2:1>:** Timer3 and Timer1 to CCPx/ECCPx Enable bits
11 = Timer3 and Timer4 are the clock sources for all CCPx/ECCPx modules
10 = Timer3 and Timer4 are the clock sources for ECCP3, CCP4 and CCP5;
Timer1 and Timer2 are the clock sources for ECCP1 and ECCP2
01 = Timer3 and Timer4 are the clock sources for ECCP2, ECCP3, CCP4 and CCP5;
Timer1 and Timer2 are the clock sources for ECCP1
00 = Timer1 and Timer2 are the clock sources for all CCPx/ECCPx modules

bit 5-4 **T3CKPS<1:0>:** Timer3 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 2 **$\overline{\text{T3SYNC}}$:** Timer3 External Clock Input Synchronization Select bit
(not usable if the device clock comes from Timer1/Timer3)
When TMR3CS = 1:
1 = Do not synchronize external clock input
0 = Synchronize external clock input
When TMR3CS = 0:
This bit is ignored. Timer3 uses the internal clock when TMR3CS = 0.

bit 1 **TMR3CS:** Timer3 Clock Source Select bit
1 = External clock input from Timer1 oscillator or T13CKI (on the rising edge after the first falling edge)
0 = Internal clock (FOSC/4)

bit 0 **TMR3ON:** Timer3 On bit
1 = Enables Timer3
0 = Stops Timer3

17.1 CCPx Module Configuration

Each Capture/Compare/PWM module is associated with a control register (generically, CCPxCON) and a data register (CCPRx). The data register, in turn, is comprised of two 8-bit registers: CCPRxL (low byte) and CCPRxH (high byte). All registers are both readable and writable.

17.1.1 CCPx/ECCPx MODULES AND TIMER RESOURCES

The CCPx/ECCPx modules utilize Timers 1, 2, 3 or 4, depending on the mode selected. Timer1 and Timer3 are available to modules in Capture or Compare modes, while Timer2 and Timer4 are available for modules in PWM mode.

TABLE 17-1: CCPx/ECCPx MODE – TIMER RESOURCE

CCPx/ECCPx Mode	Timer Resource
Capture	Timer1 or Timer3
Compare	Timer1 or Timer3
PWM	Timer2 or Timer4

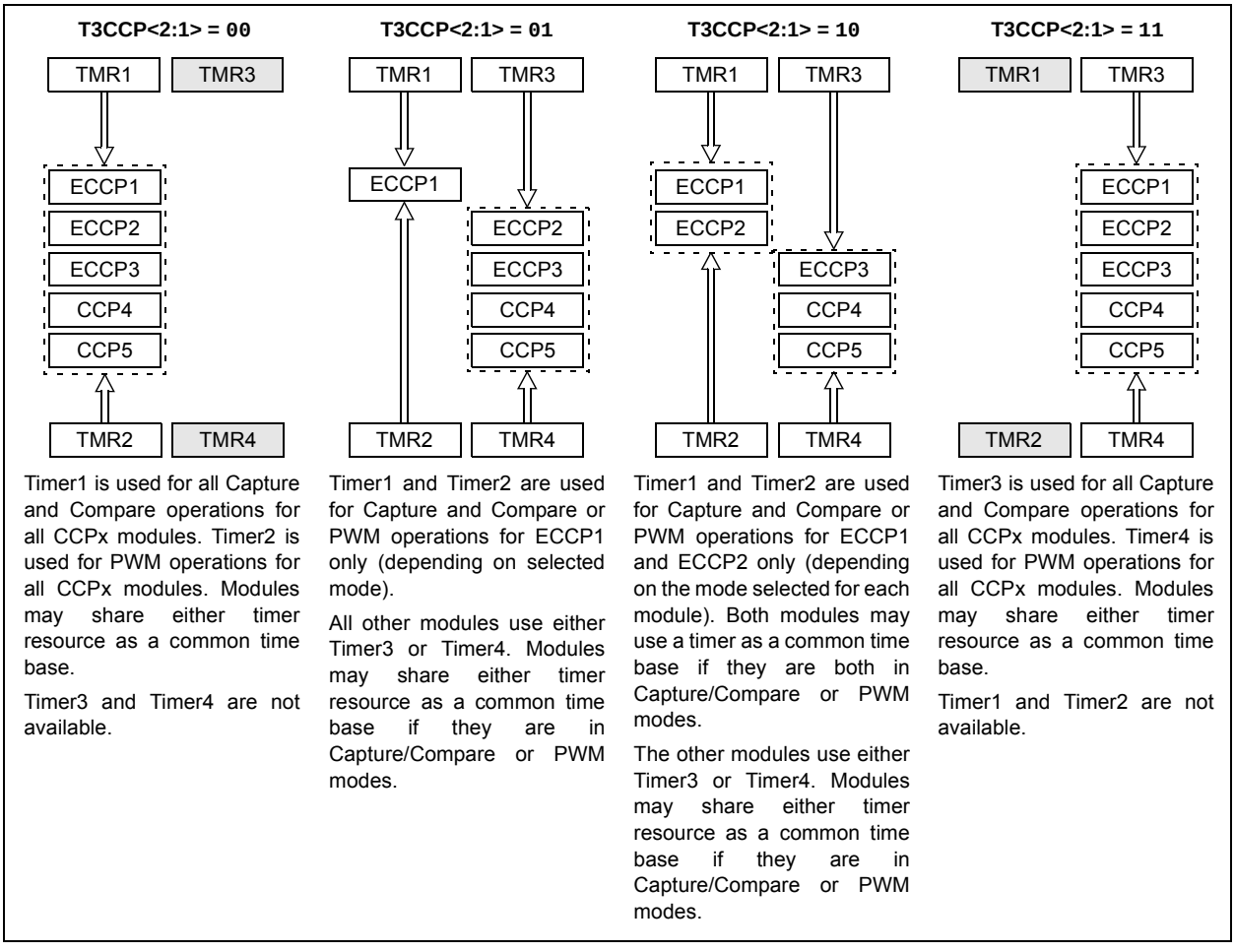
The assignment of a particular timer to a module is determined by the timer to CCPx enable bits in the T3CON register (Register 15-1, page 183). Depending on the configuration selected, up to four timers may be active at once, with modules in the same configuration (Capture/Compare or PWM) sharing timer resources. The possible configurations are shown in Figure 17-1.

17.1.2 ECCP2 PIN ASSIGNMENT

The pin assignment for ECCP2 (Capture input, Compare and PWM output) can change based on device configuration. The CCP2MX Configuration bit determines which pin ECCP2 is multiplexed to. By default, it is assigned to RC1 (CCP2MX = 1). If the Configuration bit is cleared, ECCP2 is multiplexed with RE7 on 80-pin and 100-pin devices in Microcontroller mode and RB3 on 100-pin devices in Extended Microcontroller mode.

Changing the pin assignment of ECCP2 does not automatically change any requirements for configuring the port pin. Users must always verify that the appropriate TRIS register is configured correctly for ECCP2 operation, regardless of where it is located.

FIGURE 17-1: CCPx/ECCPx AND TIMER INTERCONNECT CONFIGURATIONS



Timer1 is used for all Capture and Compare operations for all CCPx modules. Timer2 is used for PWM operations for all CCPx modules. Modules may share either timer resource as a common time base.

Timer3 and Timer4 are not available.

Timer1 and Timer2 are used for Capture and Compare or PWM operations for ECCP1 only (depending on selected mode).

All other modules use either Timer3 or Timer4. Modules may share either timer resource as a common time base if they are in Capture/Compare or PWM modes.

Timer1 and Timer2 are used for Capture and Compare or PWM operations for ECCP1 and ECCP2 only (depending on the mode selected for each module). Both modules may use a timer as a common time base if they are both in Capture/Compare or PWM modes.

The other modules use either Timer3 or Timer4. Modules may share either timer resource as a common time base if they are in Capture/Compare or PWM modes.

Timer3 is used for all Capture and Compare operations for all CCPx modules. Timer4 is used for PWM operations for all CCPx modules. Modules may share either timer resource as a common time base.

Timer1 and Timer2 are not available.

TABLE 19-3: PIC18F97J60 FAMILY PHY REGISTER SUMMARY

Addr	Name	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values
00h	PHCON1	r	r	—	—	r	r	—	PDPXMD	r	—	—	—	—	—	—	—	00-- 00-0 0--- ----
01h	PHSTAT1	—	—	—	r	r	—	—	—	—	—	—	—	—	LLSTAT	r	—	---1 1--- ---- -00-
10h	PHCON2	—	FRCLNK	r	r	r	r	r	HDLDIS	r	r	r	RXAPDIS	r	r	r	r	-000 0000 0000 0000
11h	PHSTAT2	—	—	TXSTAT	RXSTAT	COLSTAT	LSTAT	r	—	—	—	r	—	—	—	—	—	--00 00x- --0- ----
12h	PHIE	r	r	r	r	r	r	r	r	r	r	r	PLNKIE	r	r	PGEIE	r	xxxx xxxx xx00 xx00
13h	PHIR	r	r	r	r	r	r	r	r	r	r	r	PLNKIF	r	PGIF	r	r	xxxx xxxx xx00 00x0
14h	PHLCON	r	r	r	r	LACFG3	LACFG2	LACFG1	LACFG0	LBCFG3	LBCFG2	LBCFG1	LBCFG0	LFRQ1	LFRQ0	STRCH	r	0011 0100 0010 001x

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0', r = reserved, do not modify. Shaded cells are unimplemented, read as '0'.

19.5 Transmitting and Receiving Data

The Ethernet protocol (IEEE Standard 802.3) provides an extremely detailed description of the 10 Mbps, frame-based serial communications system. Before discussing the actual use of the Ethernet module, a brief review of the structure of a typical Ethernet data frame may be appropriate. It is assumed that users already have some familiarity with IEEE 802.3. Those requiring more information should refer to the official standard, or other Ethernet reference texts, for a more comprehensive explanation.

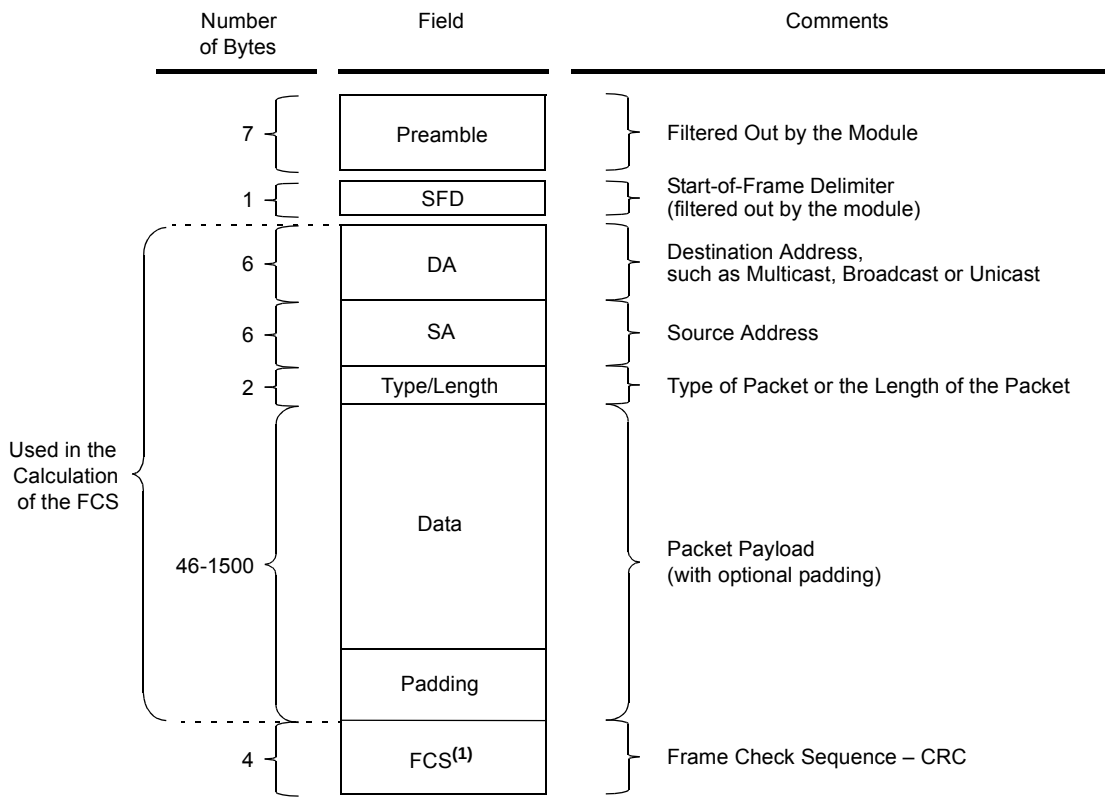
19.5.1 PACKET FORMAT

Normal IEEE 802.3 compliant Ethernet frames are between 64 and 1518 bytes long. They are made up of five or six different fields: a destination MAC address, a source MAC address, a type/length field, data payload, an optional padding field and a Cyclic Redundancy Check (CRC). Additionally, when transmitted on the Ethernet medium, a 7-byte preamble field and Start-of-Frame (SOF) delimiter byte are appended to the beginning of the Ethernet packet. Thus, traffic seen on the twisted-pair cabling will appear as shown in Figure 19-8.

19.5.1.1 Preamble/Start-of-Frame Delimiter

When transmitting and receiving data with the Ethernet module, the preamble and Start-of-Frame delimiter bytes are automatically generated, or stripped from the packets, when they are transmitted or received. It can also automatically generate CRC fields and padding as needed on transmission, and verify CRC data on reception. The user application does not need to create or process these fields, or manually verify CRC data. However, the padding and CRC fields are written into the receive buffer when packets arrive, so they may be evaluated by the user application as needed.

FIGURE 19-8: ETHERNET PACKET FORMAT



Note 1: The FCS is transmitted starting with bit 31 and ending with bit 0.

19.10 Module Resets

The Ethernet module provides selective module Resets:

- Transmit Only Reset
- Receive Only Reset

19.10.1 MICROCONTROLLER RESETS

Following any standard Reset event, the Ethernet module returns to a known state. The contents of the Ethernet buffer memory are unknown. All SFR and PHY registers are loaded with their specified Reset values, depending on the type of Reset event. However, the PHY registers must not be accessed until the PHY start-up timer has expired and the PHYRDY bit (ESTAT<0>) becomes set, or at least 1 ms has passed since the ETHEN bit was set. For more details, see **Section 19.1.3.1 "Start-up Timer"**.

19.10.2 TRANSMIT ONLY RESET

The Transmit Only Reset is performed by writing a '1' to the TXRST bit (ECON1<7>). This resets the transmit logic only. Other register and control blocks, such as buffer management and host interface, are not affected by a Transmit Only Reset event. To return to normal operation, the TXRST bit must be cleared in software. After clearing TXRST, firmware must not write to any Ethernet module SFRs for at least 1.6 μs. After the delay, normal operation can resume.

19.10.3 RECEIVE ONLY RESET

The Receive Only Reset is performed by writing a '1' to the RXRST bit (ECON1<6>). This action resets receive logic only. Other register and control blocks, such as the buffer management and host interface blocks, are not affected by a Receive Only Reset event. To return to normal operation, the RXRST bit is cleared in software. After clearing RXRST, firmware must not write to any Ethernet module SFRs for at least 1.6 μs. After the delay, normal operation can resume.

EQUATION 21-1: 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 = FOSC/(64 ([SPBRGHx:SPBRGx] + 1))
Solving for SPBRGHx:SPBRGx:

X = ((FOSC/Desired Baud Rate)/64) – 1
= ((16000000/9600)/64) – 1
= [25.042] = 25

Calculated Baud Rate=16000000/(64 (25 + 1))
= 9615
Error = (Calculated Baud Rate – Desired Baud Rate)/Desired Baud Rate
= (9615 – 9600)/9600 = 0.16%

TABLE 21-2: REGISTERS ASSOCIATED WITH BAUD RATE GENERATOR

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
TXSTAx	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	71
RCSTAx	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	71
BAUDCONx	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	—	WUE	ABDEN	72
SPBRGHx	EUSARTx Baud Rate Generator Register High Byte								72
SPBRGx	EUSARTx Baud Rate Generator Register Low Byte								72

Legend: — = unimplemented, read as '0'. Shaded cells are not used by the BRG.

NOTES:

RETURN **Return from Subroutine**

Syntax: RETURN {s}

Operands: s ∈ [0,1]

Operation: (TOS) → PC;
if s = 1,
(WS) → W,
(STATUSS) → STATUS,
(BSRS) → BSR,
PCLATU, PCLATH are unchanged

Status Affected: None

Encoding:

0000	0000	0001	001s

Description: Return from subroutine. The stack is popped and the top of the stack (TOS) is loaded into the program counter. If 's'= 1, the contents of the shadow registers WS, STATUSS and BSRS are loaded into their corresponding registers W, STATUS and BSR. If 's' = 0, no update of these registers occurs (default).

Words: 1

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	No operation	Process Data	POP PC from stack
No operation	No operation	No operation	No operation

Example: RETURN

After Instruction:
PC = TOS

RLCF **Rotate Left f through Carry**

Syntax: RLCF f {,d {,a}}

Operands: 0 ≤ f ≤ 255
d ∈ [0,1]
a ∈ [0,1]

Operation: (f<n>) → dest<n + 1>,
(f<7>) → C,
(C) → dest<0>

Status Affected: C, N, Z

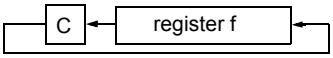
Encoding:

0011	01da	ffff	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 W. If 'd' is '1', the result is stored back in register 'f' (default).

If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).

If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever f ≤ 95 (5Fh). See **Section 26.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.



Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write to destination

Example: RLCF REG, 0, 0

Before Instruction
REG = 1110 0110
C = 0
After Instruction
REG = 1110 0110
W = 1100 1100
C = 1

28.2 DC Characteristics: Power-Down and Supply Current PIC18F97J60 Family (Industrial) (Continued)

PIC18F97J60 Family (Industrial)		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial					
Param No.	**Device**	**Typ**	**Max**	**Units**	**Conditions**		
	Supply Current (I_{DD})(2)						
	All devices	22.0	45.0	µA	-10°C	VDD = 2.0V, VDDCORE = 2.0V(4)	FOSC = 32 kHz(3) (**SEC_RUN** mode, Timer1 as clock)
		22.0	45.0	µA	+25°C		
		78.0	114.0	µA	+70°C		
	All devices	27.0	52.0	µA	-10°C	VDD = 2.5V, VDDCORE = 2.5V(4)	
		27.0	52.0	µA	+25°C		
		92.0	135.0	µA	+70°C		
	All devices	106.0	168.0	µA	-10°C	VDD = 3.3V(5)	
		106.0	168.0	µA	+25°C		
		188.0	246.0	µA	+70°C		
	All devices	18.0	37.0	µA	-10°C	VDD = 2.0V, VDDCORE = 2.0V(4)	FOSC = 32 kHz(3) (**SEC_IDLE** mode, Timer1 as clock)
		18.0	37.0	µA	+25°C		
		75.0	105.0	µA	+70°C		
	All devices	21.0	40.0	µA	-10°C	VDD = 2.5V, VDDCORE = 2.5V(4)	
		21.0	40.0	µA	+25°C		
		84.0	98.0	µA	+70°C		
	All devices	94.0	152.0	µA	-10°C	VDD = 3.3V(5)	
		94.0	152.0	µA	+25°C		
		182.0	225.0	µA	+70°C		

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, 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;

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

3: Standard, low-cost 32 kHz crystals have an operating temperature range of -10°C to +70°C. Extended temperature crystals are available at a much higher cost.

4: Voltage regulator disabled (ENVREG = 0, tied to VSS).

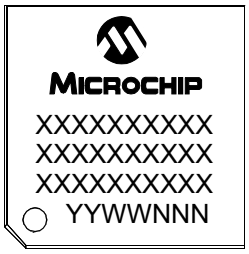
5: Voltage regulator enabled (ENVREG = 1, tied to VDD).

6: For ΔIETH, the specified current includes current sunk through TPOUT+ and TPOUT-. LEDA and LEDB are disabled for all testing.

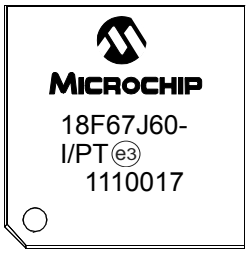
29.0 PACKAGING INFORMATION

29.1 Package Marking Information

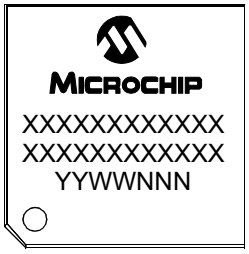
64-Lead TQFP



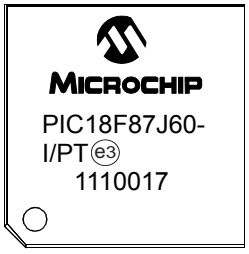
Example



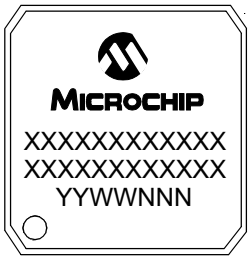
80-Lead TQFP



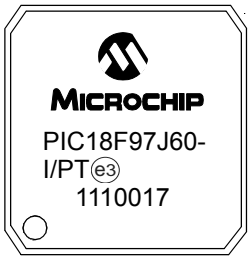
Example



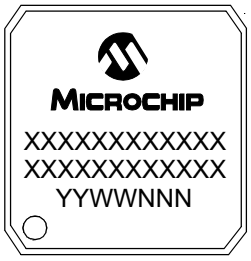
100-Lead TQFP (12x12x1 mm)



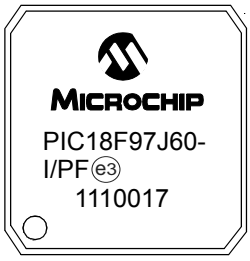
Example



100-Lead TQFP (14x14x1 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.

NOTES:

Master Mode 298
Baud Rate Generator 300
Operation 299
Reception 304
Repeated Start Condition Timing 303
Start Condition Timing 302
Transmission 304
Multi-Master Communication, Bus Collision and Arbitration 308
Multi-Master Mode 308
Operation 284
Read/Write Bit Information (R/W Bit) 284, 286
Registers 279
Serial Clock (SCKx/SCLx) 286
Slave Mode 284
Address Masking 285
Addressing 284
Reception 286
Transmission 286
Sleep Operation 308
Stop Condition Timing 307
INCF 396
INCFSZ 397
In-Circuit Debugger 373
In-Circuit Serial Programming (ICSP) 359, 373
Indexed Literal Offset Addressing and Standard PIC18 Instructions 422
Indexed Literal Offset Mode 422
Indirect Addressing 99
INFSNZ 397
Initialization Conditions for All Registers 69–75
Instruction Cycle 84
Clocking Scheme 84
Flow/Pipelining 84
Instruction Set 375
ADDLW 381
ADDWF 381
ADDWF (Indexed Literal Offset Mode) 423
ADDWFC 382
ANDLW 382
ANDWF 383
BC 383
BCF 384
BN 384
BNC 385
BNN 385
BNOV 386
BNZ 386
BOV 389
BRA 387
BSF 387
BSF (Indexed Literal Offset Mode) 423
BTFSC 388
BTFSS 388
BTG 389
BZ 390
CALL 390
CLRF 391
CLRWDT 391
COMF 392
CPFSEQ 392
CPFSGT 393
CPFSLT 393
DAW 394
DCFSNZ 395
DECF 394
DECFSZ 395
Extended Instructions 417
Considerations when Enabling 422
Syntax 417
Use with MPLAB IDE Tools 424
General Format 377
GOTO 396
INCF 396
INCFSZ 397
INFSNZ 397
IORLW 398
IORWF 398
LFSR 399
MOVF 399
MOVFF 400
MOVLB 400
MOVLW 401
MOVWF 401
MULLW 402
MULWF 402
NEGF 403
NOP 403
POP 404
PUSH 404
RCALL 405
RESET 405
RETFIE 406
RETLW 406
RETURN 407
RLCF 407
RLNCF 408
RRCF 408
RRNCF 409
SETF 409
SETF (Indexed Literal Offset Mode) 423
SLEEP 410
Standard Instructions 375
SUBFWB 410
SUBLW 411
SUBWF 411
SUBWFB 412
SWAPF 412
TBLRD 413
TBLWT 414
TSTFSZ 415
XORLW 415
XORWF 416
INTCON Register
RBIF Bit 148
INTCON Registers 131
Inter-Integrated Circuit. *See* I^2C Mode.
Internal Oscillator Block 51
Internal RC Oscillator
Use with WDT 367
Internal Voltage Regulator Specifications 442
Internet Address 488

THE MICROCHIP WEB SITE

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

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

CUSTOMER CHANGE NOTIFICATION SERVICE

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

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

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support
- Development Systems Information Line

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

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