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	39
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 11x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-TQFP
Supplier Device Package	64-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f67j60t-i-pt

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 uluu'. 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 Borrow and 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/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/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 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 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.

6.6.1 INDEXED ADDRESSING WITH LITERAL OFFSET

Enabling the PIC18 extended instruction set changes the behavior of Indirect Addressing using the FSR2 register pair and its associated file operands. Under the proper conditions, instructions that use the Access Bank – that is, most bit-oriented and byte-oriented instructions – can invoke a form of Indexed Addressing using an offset specified in the instruction. This special addressing mode is known as Indexed Addressing with Literal Offset or Indexed Literal Offset mode.

When using the extended instruction set, this addressing mode requires the following:

• The use of the Access Bank is forced ('a' = 0); and

• The file address argument is less than or equal to 5Fh.

Under these conditions, the file address of the instruction is not interpreted as the lower byte of an address (used with the BSR in Direct Addressing) or as an 8-bit address in the Access Bank. Instead, the value is interpreted as an offset value to an Address Pointer specified by FSR2. The offset and the contents of FSR2 are added to obtain the target address of the operation.

6.6.2 INSTRUCTIONS AFFECTED BY INDEXED LITERAL OFFSET MODE

Any of the core PIC18 instructions that can use Direct Addressing are potentially affected by the Indexed Literal Offset Addressing mode. This includes all byte-oriented and bit-oriented instructions, or almost half of the standard PIC18 instruction set. Instructions that only use Inherent or Literal Addressing modes are unaffected.

Additionally, byte-oriented and bit-oriented instructions are not affected if they use the Access Bank (Access RAM bit is '1') or include a file address of 60h or above. Instructions meeting these criteria will continue to execute as before. A comparison of the different possible addressing modes when the extended instruction set is enabled is shown in Figure 6-10.

Those who desire to use byte-oriented or bit-oriented instructions in the Indexed Literal Offset mode should note the changes to assembler syntax for this mode. This is described in more detail in **Section 26.2.1 "Extended Instruction Syntax"**.

The CCPRxH register and a 2-bit internal latch are used to double-buffer the PWM duty cycle. This double-buffering is essential for glitchless PWM operation.

When the CCPRxH and 2-bit latch match TMR2 (TMR4), concatenated with an internal 2-bit Q clock or 2 bits of the TMR2 (TMR4) prescaler, the CCPx pin is cleared.

The maximum PWM resolution (bits) for a given PWM frequency is given by Equation 17-3:

EQUATION 17-3:

$$\text{PWM Resolution (max)} = \frac{\log\left(\frac{F_{OSC}}{F_{PWM}}\right)}{\log(2)} \text{ bits}$$

Note:	If the PWM duty cycle value is longer than the PWM period, the CCPx pin will not be cleared.

17.4.3   SETUP FOR PWM OPERATION

The following steps should be taken when configuring the CCPx module for PWM operation:

1. Set the PWM period by writing to the PR2 (PR4) register.
2. Set the PWM duty cycle by writing to the CCPRxL register and CCPxCON<5:4> bits.
3. Make the CCPx pin an output by clearing the appropriate TRIS bit.
4. Set the TMR2 (TMR4) prescale value, then enable Timer2 (Timer4) by writing to T2CON (T4CON).
5. Configure the CCPx module for PWM operation.

TABLE 17-3:   EXAMPLE PWM FREQUENCIES AND RESOLUTIONS AT 40 MHz

PWM Frequency	2.44 kHz	9.77 kHz	39.06 kHz	156.25 kHz	312.50 kHz	416.67 kHz
Timer Prescaler (1, 4, 16)	16	4	1	1	1	1
PR2 Value	FFh	FFh	FFh	3Fh	1Fh	17h
Maximum Resolution (bits)	10	10	10	8	7	6.58

REGISTER 19-12: PHSTAT2: PHYSICAL LAYER STATUS REGISTER 2

U-0	U-0	R-0	R-0	R-0	R-0	R-x	U-0
—	—	TXSTAT	RXSTAT	COLSTAT	LSTAT	r	—
bit 15							bit 8

U-0	U-0	R-0	U-0	U-0	U-0	U-0	U-0
—	—	r	—	—	—	—	—
bit 7							bit 0

Legend:		r = Reserved bit	
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 15-14 **Unimplemented:** Read as '0'

bit 13 **TXSTAT:** PHY Transmit Status bit
1 = PHY is transmitting data
0 = PHY is not transmitting data

bit 12 **RXSTAT:** PHY Receive Status bit
1 = PHY is receiving data
0 = PHY is not receiving data

bit 11 **COLSTAT:** PHY Collision Status bit
1 = A collision is occuring (PHY is both transmitting and receiving while in Half-Duplex mode)
0 = A collision is not occuring

bit 10 **LSTAT:** PHY Collision Status bit
1 = Link is up
0 = Link is down

bit 9 **Reserved:** Ignore on read

bit 8-6 **Unimplemented:** Read as '0'

bit 5 **Reserved:** Ignore on read

bit 4-0 **Unimplemented:** Read as '0'

REGISTER 19-15: EIR: ETHERNET INTERRUPT REQUEST (FLAG) REGISTER

| U-0	R-0	R/C-0	R-0	R/C-0	U-0	R/C-0	R/C-0
—	PKTIF	DMAIF	LINKIF	TXIF	—	TXERIF	RXERIF
bit 7							bit 0

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

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

bit 6 **PKTIF:** Receive Packet Pending Interrupt Flag bit

1 = Receive buffer contains one or more unprocessed packets; cleared only when EPKTCNT is decremented to 0 by setting PKTDEC (ECON2<6>)
0 = Receive buffer is empty

bit 5 **DMAIF:** DMA Interrupt Flag bit

1 = DMA copy or checksum calculation has completed
0 = No DMA interrupt is pending

bit 4 **LINKIF:** Link Change Interrupt Flag bit

1 = PHY reports that the link status has changed; read PHIR register to clear
0 = Link status has not changed

bit 3 **TXIF:** Transmit Interrupt Flag bit

1 = Transmit request has ended
0 = No transmit interrupt is pending

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

bit 1 **TXERIF:** Transmit Error Interrupt Flag bit

1 = A transmit error has occurred
0 = No transmit error has occurred

bit 0 **RXERIF:** Receive Error Interrupt Flag bit

1 = A packet was aborted because there is insufficient buffer space, or a buffer overrun has occurred
0 = No receive error interrupt is pending

FIGURE 20-11: I^2C™ SLAVE MODE TIMING WITH SEN = 0 (RECEPTION, 10-BIT ADDRESS)

PIC18F97J60 FAMILY

FIGURE 20-16: I²C™ SLAVE MODE TIMING WITH SEN = 1 (RECEPTION, 10-BIT ADDRESS)

© 2011 Microchip Technology Inc.
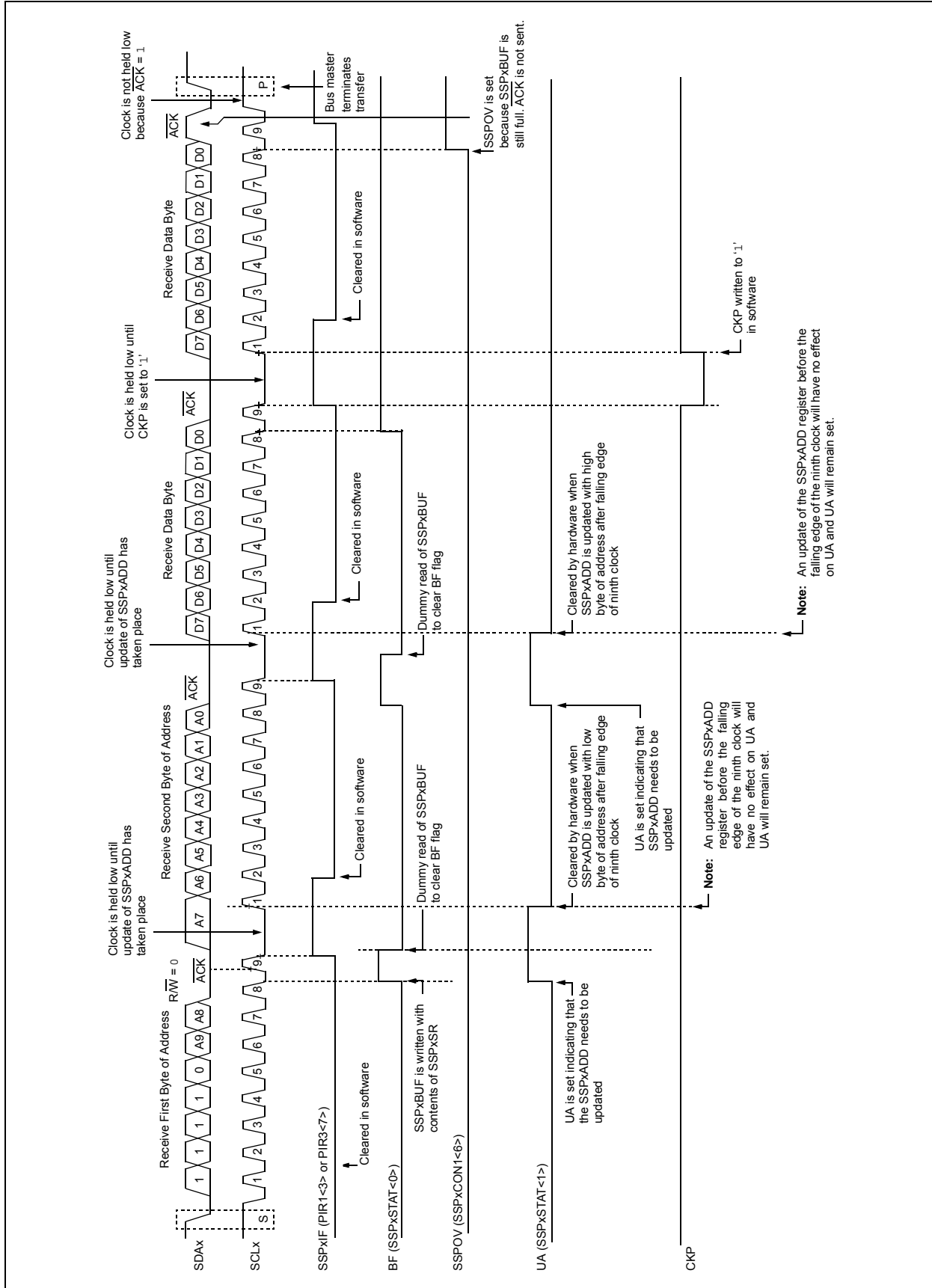
20.4.12 ACKNOWLEDGE SEQUENCE TIMING

An Acknowledge sequence is enabled by setting the Acknowledge Sequence Enable bit, ACKEN (SSPxCON2<4>). When this bit is set, the SCLx pin is pulled low and the contents of the Acknowledge data bit are presented on the SDAx pin. If the user wishes to generate an Acknowledge, then the ACKDT bit should be cleared. If not, the user should set the ACKDT bit before starting an Acknowledge sequence. The Baud Rate Generator then counts for one rollover period (TBRG) and the SCLx pin is deasserted (pulled high). When the SCLx pin is sampled high (clock arbitration), the Baud Rate Generator counts for TBRG. The SCLx pin is then pulled low. Following this, the ACKEN bit is automatically cleared, the Baud Rate Generator is turned off and the MSSP module then goes into Idle mode (Figure 20-25).

20.4.12.1 WCOL Status Flag

If the user writes the SSPxBUF when an Acknowledge sequence is in progress, then WCOL is set and the contents of the buffer are unchanged (the write doesn't occur).

20.4.13 STOP CONDITION TIMING

A Stop bit is asserted on the SDAx pin at the end of a receive/transmit by setting the Stop Sequence Enable bit, PEN (SSPxCON2<2>). At the end of a receive/transmit, the SCLx line is held low after the falling edge of the ninth clock. When the PEN bit is set, the master will assert the SDAx line low. When the SDAx line is sampled low, the Baud Rate Generator is reloaded and counts down to '0'. When the Baud Rate Generator times out, the SCLx pin will be brought high and one TBRG (Baud Rate Generator rollover count) later, the SDAx pin will be deasserted. When the SDAx pin is sampled high while SCLx is high, the P bit (SSPxSTAT<4>) is set. A TBRG later, the PEN bit is cleared and the SSPxIF bit is set (Figure 20-26).

20.4.13.1 WCOL Status Flag

If the user writes the SSPxBUF when a Stop sequence is in progress, then the WCOL bit is set and the contents of the buffer are unchanged (the write doesn't occur).

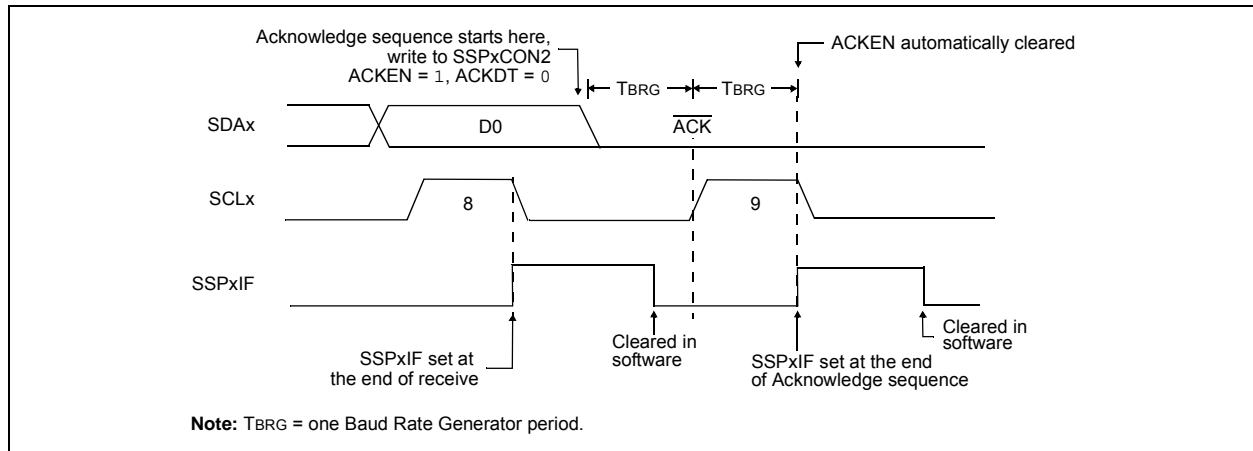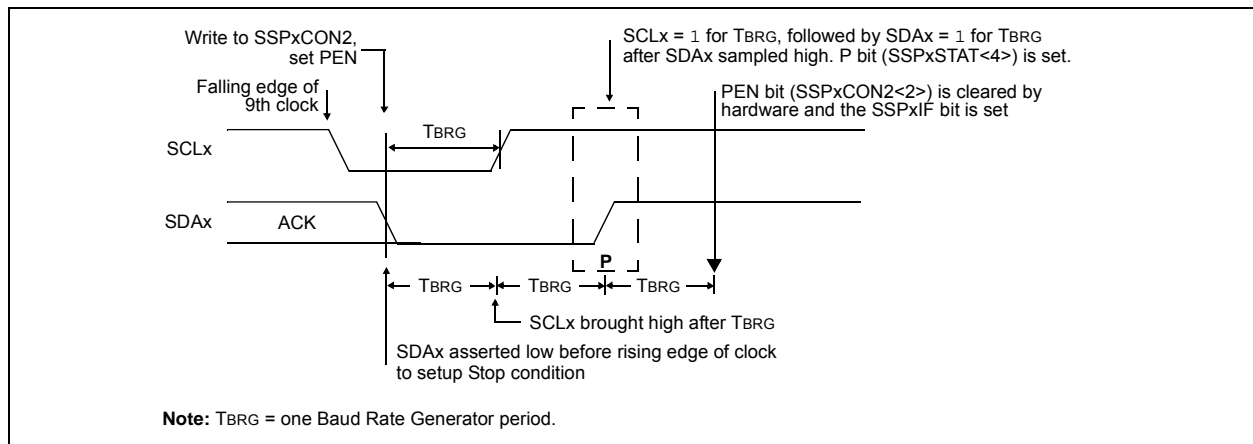**FIGURE 20-25: ACKNOWLEDGE SEQUENCE WAVEFORM**

FIGURE 20-26: STOP CONDITION RECEIVE OR TRANSMIT MODE

21.0 ENHANCED UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER (EUSART)

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

The Enhanced USART module implements additional features, including automatic baud rate detection and calibration, automatic wake-up on Sync Break reception and 12-bit Break character transmit. These features make it ideally suited for use in Local Interconnect Network bus (LIN/J2602 bus) systems.

The 64-pin devices of the PIC18F97J60 family are equipped with one EUSART module, referred to as EUSART1. The 80-pin and 100-pin devices each have two independent EUSART modules, referred to as EUSART1 and EUSART2. They can be configured in the following modes:

- Asynchronous (full-duplex) with:
  - Auto-Wake-up on Character Reception
  - Auto-Baud Calibration
  - 12-Bit Break Character Transmission
- Synchronous – Master (half-duplex) with Selectable Clock Polarity
- Synchronous – Slave (half-duplex) with Selectable Clock Polarity

The pins of EUSART1 and EUSART2 are multiplexed with the functions of PORTC (RC6/TX1/CK1 and RC7/RX1/DT1) and PORTG (RG1/TX2/CK2 and RG2/RX2/DT2), respectively. In order to configure these pins as an EUSART:

- For EUSART1:
  - SPEN bit (RCSTA1<7>) must be set (= 1)
  - TRISC<7> bit must be set (= 1)
  - TRISC<6> bit must be cleared (= 0) for Asynchronous and Synchronous Master modes
  - TRISC<6> bit must be set (= 1) for Synchronous Slave mode
- For EUSART2:
  - SPEN bit (RCSTA2<7>) must be set (= 1)
  - TRISG<2> bit must be set (= 1)
  - TRISG<1> bit must be cleared (= 0) for Asynchronous and Synchronous Master modes
  - TRISG<1> bit must be set (= 1) for Synchronous Slave mode

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

The operation of each Enhanced USART module is controlled through three registers:

- Transmit Status and Control (TXSTAx)
- Receive Status and Control (RCSTAx)
- Baud Rate Control (BAUDCONx)

These are detailed on the following pages in Register 21-1, Register 21-2 and Register 21-3, respectively.

> **Note:** Throughout this section, references to register and bit names that may be associated with a specific EUSART module are referred to generically by the use of 'x' in place of the specific module number. Thus, "RCSTAx" might refer to the Receive Status register for either EUSART1 or EUSART2.

TABLE 21-3: BAUD RATES FOR ASYNCHRONOUS MODES

BAUD RATE (K)	SYNC = 0, BRG16 = 0, BRGH = 0											
	FOSC = 41.667 MHz			FOSC = 31.25 MHz			FOSC = 25.000 MHz			FOSC = 20.833 MHz		
	Actual Rate (K)	% Error	SPBRG Value (decimal)	Actual Rate (K)	% Error	SPBRG Value (decimal)	Actual Rate (K)	% Error	SPBRG Value (decimal)	Actual Rate (K)	% Error	SPBRG Value (decimal)
0.3	—	—	—	—	—	—	—	—	—	—	—	—
1.2	—	—	—	—	—	—	—	—	—	1.271	5.96	255
2.4	2.543	5.96	255	2.405	0.22	202	2.396	-0.15	162	2.393	-0.27	135
9.6	9.574	-0.27	67	9.574	-0.27	50	9.527	-0.76	40	9.574	-0.27	33
19.2	19.148	-0.27	33	19.531	1.73	24	19.531	1.73	19	19.147	-0.27	16
57.6	59.186	2.75	10	61.035	5.96	7	55.804	-3.12	6	54.253	-5.81	5
115.2	108.508	-5.81	5	122.070	5.96	3	130.208	13.03	2	108.505	-5.81	2

BAUD RATE (K)	SYNC = 0, BRG16 = 0, BRGH = 0								
	FOSC = 13.889 MHz			FOSC = 6.250 MHz			FOSC = 4.167 MHz		
	Actual Rate (K)	% Error	SPBRG Value (decimal)	Actual Rate (K)	% Error	SPBRG Value (decimal)	Actual Rate (K)	% Error	SPBRG Value (decimal)
0.3	—	—	—	—	—	—	0.300	0.01	216
1.2	1.198	-0.08	180	1.206	0.47	80	1.206	0.48	53
2.4	2.411	0.47	89	2.382	-0.76	40	2.411	0.48	26
9.6	9.435	-1.71	22	9.766	1.73	9	9.301	-3.11	6
19.2	19.279	2.75	10	19.531	1.73	4	21.703	13.04	2
57.6	54.254	-5.81	3	48.828	-15.23	1	65.109	13.04	0
115.2	108.508	-5.81	1	97.656	-15.23	0	65.109	-43.48	0

BAUD RATE (K)	SYNC = 0, BRG16 = 0, BRGH = 1											
	FOSC = 41.667 MHz			FOSC = 31.25 MHz			FOSC = 25.000 MHz			FOSC = 20.833 MHz		
	Actual Rate (K)	% Error	SPBRG Value (decimal)	Actual Rate (K)	% Error	SPBRG Value (decimal)	Actual Rate (K)	% Error	SPBRG Value (decimal)	Actual Rate (K)	% Error	SPBRG Value (decimal)
0.3	—	—	—	—	—	—	—	—	—	—	—	—
1.2	—	—	—	—	—	—	—	—	—	—	—	—
2.4	—	—	—	—	—	—	—	—	—	—	—	—
9.6	10.172	5.96	255	9.621	0.22	202	9.586	-0.15	162	9.573	-0.27	135
19.2	19.148	-0.27	135	19.148	-0.27	101	19.290	0.47	80	19.147	-0.27	67
57.6	57.871	0.47	44	57.445	-0.27	33	57.870	0.47	26	56.611	-1.72	22
115.2	113.226	-1.71	22	114.890	-0.27	16	111.607	-3.12	13	118.369	2.75	10

BAUD RATE (K)	SYNC = 0, BRG16 = 0, BRGH = 1								
	FOSC = 13.889 MHz			FOSC = 6.250 MHz			FOSC = 4.167 MHz		
	Actual Rate (K)	% Error	SPBRG Value (decimal)	Actual Rate (K)	% Error	SPBRG Value (decimal)	Actual Rate (K)	% Error	SPBRG Value (decimal)
0.3	—	—	—	—	—	—	—	—	—
1.2	—	—	—	—	—	—	1.200	0.01	216
2.4	—	—	—	2.396	-0.15	162	2.389	-0.44	108
9.6	9.645	0.47	89	9.527	-0.76	40	9.645	0.48	26
19.2	19.290	0.47	44	19.531	1.73	19	18.603	-3.11	13
57.6	57.871	0.47	14	55.804	-3.12	6	52.088	-9.57	4
115.2	108.508	-5.81	7	130.208.	13.03	2	130.219	13.04	1

PIC18F97J60 FAMILY

21.2.5   BREAK CHARACTER SEQUENCE

The EUSARTx module has the capability of sending the special Break character sequences that are required by the LIN/J2602 bus standard. The Break character transmit consists of a Start bit, followed by twelve '0' bits and a Stop bit. The Frame Break character is sent whenever the SENDB and TXEN bits (TXSTAx<3> and TXSTAx<5>) are set while the Transmit Shift Register (TSR) is loaded with data. Note that the value of data written to TXREGx will be ignored and all '0's will be transmitted.

The SENDB bit is automatically reset by hardware after the corresponding Stop bit is sent. This allows the user to preload the transmit FIFO with the next transmit byte following the Break character (typically, the Sync character in the LIN/J2602 support specification).

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

The TRMT bit indicates when the transmit operation is active or Idle, just as it does during normal transmission. See Figure 21-10 for the timing of the Break character sequence.

21.2.5.1   Break and Sync Transmit Sequence

The following sequence will send a message frame header made up of a Break, followed by an Auto-Baud Sync byte. This sequence is typical of a LIN/J2602 bus master.

1. Configure the EUSARTx for the desired mode.
2. Set the TXEN and SENDB bits to set up the Break character.
3. Load the TXREGx with a dummy character to initiate transmission (the value is ignored).
4. Write '55h' to TXREGx to load the Sync character into the transmit FIFO buffer.
5. After the Break has been sent, the SENDB bit is reset by hardware. The Sync character now transmits in the preconfigured mode.

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

21.2.6   RECEIVING A BREAK CHARACTER

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

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

The second method uses the auto-wake-up feature described in **Section 21.2.4 "Auto-Wake-up on Sync Break Character"**. By enabling this feature, the EUSARTx will sample the next two transitions on RXx/DTx, cause an RCxIF interrupt and receive the next data byte followed by another interrupt.

Note that following a Break character, the user will typically want to enable the Auto-Baud Rate Detect feature. For both methods, the user can set the ABDEN bit once the TXxIF interrupt is observed.

FIGURE 21-10:   SEND BREAK CHARACTER SEQUENCE
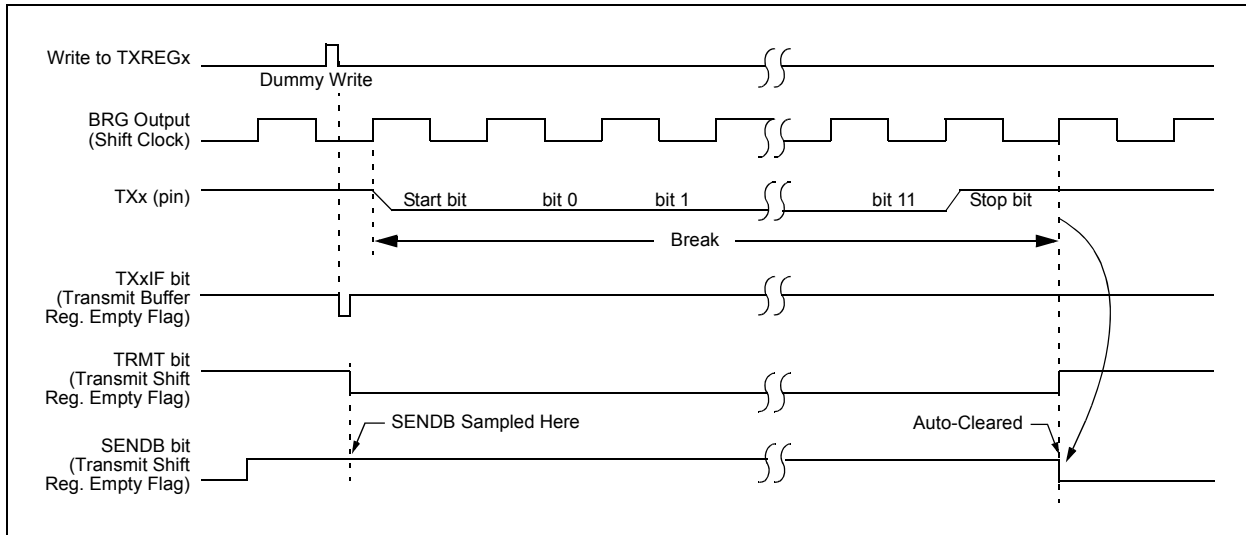
REGISTER 22-3: ADCON2: A/D CONTROL REGISTER 2

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	—	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0
bit 7							bit 0

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

bit 7   **ADFM:** A/D Result Format Select bit

1 = Right justified
0 = Left justified

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

bit 5-3   **ACQT<2:0>:** A/D Acquisition Time Select bits

111 = 20 T_{AD}
110 = 16 T_{AD}
101 = 12 T_{AD}
100 = 8 T_{AD}
011 = 6 T_{AD}
010 = 4 T_{AD}
001 = 2 T_{AD}
000 = 0 T_{AD} [1]

bit 2-0   **ADCS<2:0>:** A/D Conversion Clock Select bits

111 = F_{RC} (clock derived from A/D RC oscillator) [1]
110 = $F_{OSC}/64$
101 = $F_{OSC}/16$
100 = $F_{OSC}/4$
011 = F_{RC} (clock derived from A/D RC oscillator) [1]
010 = $F_{OSC}/32$
001 = $F_{OSC}/8$
000 = $F_{OSC}/2$

Note 1: If the A/D F_{RC} clock source is selected, a delay of one T_{CY} (instruction cycle) is added before the A/D clock starts. This allows the SLEEP instruction to be executed before starting a conversion.

PIC18F97J60 FAMILY

TABLE 26-1:   OPCODE FIELD DESCRIPTIONS

Field	Description
a	RAM access bit: a = 0: RAM location in Access RAM (BSR register is ignored) a = 1: RAM bank is specified by BSR register
bbb	Bit address within an 8-bit file register (0 to 7).
BSR	Bank Select Register. Used to select the current RAM bank.
C, DC, Z, OV, N	ALU Status bits: **C**arry, **D**igit Carry, **Z**ero, **Ov**erflow, **N**egative.
d	Destination select bit: d = 0: store result in WREG d = 1: store result in file register f
dest	Destination: either the WREG register or the specified register file location.
f	8-bit Register file address (00h to FFh), or 2-bit FSR designator (0h to 3h).
f_s	12-bit Register file address (000h to FFFh). This is the source address.
f_d	12-bit Register file address (000h to FFFh). This is the destination address.
GIE	Global Interrupt Enable bit.
k	Literal field, constant data or label (may be either an 8-bit, 12-bit or a 20-bit value).
label	Label name.
mm	The mode of the TBLPTR register for the table read and table write instructions. Only used with table read and table write instructions:
*	No Change to register (such as TBLPTR with table reads and writes)
*+	Post-Increment register (such as TBLPTR with table reads and writes)
*-	Post-Decrement register (such as TBLPTR with table reads and writes)
+*	Pre-Increment register (such as TBLPTR with table reads and writes)
n	The relative address (2's complement number) for relative branch instructions or the direct address for Call/Branch and Return instructions.
PC	Program Counter.
PCL	Program Counter Low Byte.
PCH	Program Counter High Byte.
PCLATH	Program Counter High Byte Latch.
PCLATU	Program Counter Upper Byte Latch.
$\overline{PD}$	Power-Down bit.
PRODH	Product of Multiply High Byte.
PRODL	Product of Multiply Low Byte.
s	Fast Call/Return mode select bit: s = 0: do not update into/from shadow registers s = 1: certain registers loaded into/from shadow registers (Fast mode)
TBLPTR	21-bit Table Pointer (points to a program memory location).
TABLAT	8-bit Table Latch.
$\overline{TO}$	Time-out bit.
TOS	Top-of-Stack.
u	Unused or Unchanged.
WDT	Watchdog Timer.
WREG	Working register (accumulator).
x	Don't care ('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.
z_s	7-bit offset value for indirect addressing of register files (source).
z_d	7-bit offset value for indirect addressing of register files (destination).
{ }	Optional argument.
[text]	Indicates an indexed address.
(text)	The contents of text.
[expr]<n>	Specifies bit n of the register indicated by the pointer expr.
$\rightarrow$	Assigned to.
< >	Register bit field.
$\in$	In the set of.
italics	User-defined term (font is Courier New).

BNC	Branch if Not Carry
Syntax:	BNC n
Operands:	-128 ≤ n ≤ 127
Operation:	if Carry bit is '0', (PC) + 2 + 2n → PC
Status Affected:	None

Encoding:

1110	0011	nnnn	nnnn

Description: If the Carry bit is '0', then the program will branch.

The 2's complement number '2n' is added to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC + 2 + 2n. This instruction is then a two-cycle instruction.

Words: 1

Cycles: 1(2)

Q Cycle Activity:
If Jump:

Q1	Q2	Q3	Q4
Decode	Read literal 'n'	Process Data	Write to PC
No operation	No operation	No operation	No operation

If No Jump:

Q1	Q2	Q3	Q4
Decode	Read literal 'n'	Process Data	No operation

Example: HERE   BNC   Jump

Before Instruction
        PC   =   address (HERE)
After Instruction
        If Carry   =   0;
            PC   =   address (Jump)
        If Carry   =   1;
            PC   =   address (HERE + 2)

BNN	Branch if Not Negative
Syntax:	BNN n
Operands:	-128 ≤ n ≤ 127
Operation:	if Negative bit is '0', (PC) + 2 + 2n → PC
Status Affected:	None

Encoding:

1110	0111	nnnn	nnnn

Description: If the Negative bit is '0', then the program will branch.

The 2's complement number '2n' is added to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC + 2 + 2n. This instruction is then a two-cycle instruction.

Words: 1

Cycles: 1(2)

Q Cycle Activity:
If Jump:

Q1	Q2	Q3	Q4
Decode	Read literal 'n'	Process Data	Write to PC
No operation	No operation	No operation	No operation

If No Jump:

Q1	Q2	Q3	Q4
Decode	Read literal 'n'	Process Data	No operation

Example: HERE   BNN   Jump

Before Instruction
        PC   =   address (HERE)
After Instruction
        If Negative =   0;
            PC   =   address (Jump)
        If Negative =   1;
            PC   =   address (HERE + 2)

MOVLW **Move Literal to W**

Syntax:	MOVLW k
Operands:	$0 \leq k \leq 255$
Operation:	$k \rightarrow W$
Status Affected:	None

Encoding:

0000	1110	kkkk	kkkk

Description: The eight-bit literal 'k' is loaded into W.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'	Process Data	Write to W

Example: MOVLW 5Ah

After Instruction
W   =   5Ah

MOVWF **Move W to f**

Syntax:	MOVWF f {,a}
Operands:	$0 \leq f \leq 255$ $a \in [0,1]$
Operation:	$(W) \rightarrow f$
Status Affected:	None

Encoding:

0110	111a	ffff	ffff

Description: Move data from W to register 'f'. Location 'f' can be anywhere in the 256-byte bank.

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 $f \leq 95$ (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 register 'f'

Example: MOVWF REG, 0

Before Instruction
W   =   4Fh
REG   =   FFh
After Instruction
W   =   4Fh
REG   =   4Fh

TBLRD	Table Read

Syntax:	TBLRD (*; *+; *-; +*)
Operands:	None
Operation:	if TBLRD*, (Prog Mem (TBLPTR)) → TABLAT; TBLPTR – No Change if TBLRD*+, (Prog Mem (TBLPTR)) → TABLAT; (TBLPTR) + 1 → TBLPTR if TBLRD*-, (Prog Mem (TBLPTR)) → TABLAT; (TBLPTR) – 1 → TBLPTR if TBLRD +*, (TBLPTR) + 1 → TBLPTR; (Prog Mem (TBLPTR)) → TABLAT
Status Affected:	None

Encoding:

0000	0000	0000	10nn nn=0 * =1 *+ =2 *– =3 +*

Description:

This instruction is used to read the contents of Program Memory (P.M.). To address the program memory, a pointer called Table Pointer (TBLPTR) is used.

The TBLPTR (a 21-bit pointer) points to each byte in the program memory. TBLPTR has a 2-Mbyte address range.

TBLPTR[0] = 0: Least Significant Byte of Program Memory Word

TBLPTR[0] = 1: Most Significant Byte of Program Memory Word

The TBLRD instruction can modify the value of TBLPTR as follows:

- no change
- post-increment
- post-decrement
- pre-increment

Words:	1
Cycles:	2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	No operation	No operation	No operation
No operation	No operation (Read Program Memory)	No operation	No operation (Write TABLAT)

TBLRD	Table Read (Continued)

Example 1:   TBLRD *+ ;

Before Instruction

TABLAT	=	55h
TBLPTR	=	00A356h
MEMORY(00A356h)	=	34h

After Instruction

TABLAT	=	34h
TBLPTR	=	00A357h

Example 2:   TBLRD +* ;

Before Instruction

TABLAT	=	AAh
TBLPTR	=	01A357h
MEMORY(01A357h)	=	12h
MEMORY(01A358h)	=	34h

After Instruction

TABLAT	=	34h
TBLPTR	=	01A358h

TSTFSZ	Test f, Skip if 0

Syntax:	TSTFSZ f {,a}
Operands:	$0 \le f \le 255$ $a \in [0,1]$
Operation:	skip if f = 0
Status Affected:	None
Encoding:	0110	011a	ffff	ffff

Description:	If 'f' = 0, the next instruction fetched during the current instruction execution is discarded and a NOP is executed, making this a two-cycle instruction. 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 $\le$ 95 (5Fh). See **Section 26.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.
Words:	1
Cycles:	1(2) **Note:** 3 cycles if skip and followed by a 2-word instruction.

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	No operation

If skip:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation

If skip and followed by 2-word instruction:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation
No operation	No operation	No operation	No operation

Example:

```
HERE    TSTFSZ  CNT, 1
NZERO   :
ZERO    :
```

Before Instruction
PC   =   Address (HERE)
After Instruction
If CNT   =   00h,
PC   =   Address (ZERO)
If CNT   ≠   00h,
PC   =   Address (NZERO)

XORLW	Exclusive OR Literal with W

Syntax:	XORLW k
Operands:	$0 \le k \le 255$
Operation:	(W) .XOR. k $\rightarrow$ W
Status Affected:	N, Z
Encoding:	0000	1010	kkkk	kkkk
Description:	The contents of W are XORed with the 8-bit literal 'k'. The result is placed in W.
Words:	1
Cycles:	1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'	Process Data	Write to W

Example: XORLW   0AFh

Before Instruction
W   =   B5h
After Instruction
W   =   1Ah

PIC18F97J60 FAMILY

26.2.2   EXTENDED INSTRUCTION SET

ADDFSR	**Add Literal to FSR**
Syntax:	ADDFSR f, k
Operands:	$0 \leq k \leq 63$ $f \in [\, 0, 1, 2\,]$
Operation:	FSR(f) + k → FSR(f)
Status Affected:	None
Encoding:	`1110` `1000` `ffkk` `kkkk`
Description:	The 6-bit literal 'k' is added to the contents of the FSR specified by 'f'.
Words:	1
Cycles:	1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'	Process Data	Write to FSR

Example: ADDFSR 2, 23h

Before Instruction
    FSR2   =   03FFh

After Instruction
    FSR2   =   0422h

ADDULNK	**Add Literal to FSR2 and Return**
Syntax:	ADDULNK k
Operands:	$0 \leq k \leq 63$
Operation:	FSR2 + k → FSR2, (TOS) → PC
Status Affected:	None
Encoding:	`1110` `1000` `11kk` `kkkk`
Description:	The 6-bit literal 'k' is added to the contents of FSR2. A RETURN is then executed by loading the PC with the TOS. The instruction takes two cycles to execute; a NOP is performed during the second cycle. This may be thought of as a special case of the ADDFSR instruction, where f = 3 (binary '11'); it operates only on FSR2.
Words:	1
Cycles:	2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'	Process Data	Write to FSR
No Operation	No Operation	No Operation	No Operation

Example: ADDULNK 23h

Before Instruction
    FSR2   =   03FFh
    PC   =   0100h

After Instruction
    FSR2   =   0422h
    PC   =   (TOS)

Note: All PIC18 instructions may take an optional label argument preceding the instruction mnemonic for use in symbolic addressing. If a label is used, the instruction format then becomes: {label} instruction argument(s).

PIC18F97J60 FAMILY

TABLE 28-2: COMPARATOR SPECIFICATIONS

Operating Conditions: $3.0V \leq V_{DD} \leq 3.6V$, $-40°C \leq T_A \leq +85°C$ (unless otherwise stated)							
Param No.	Sym	Characteristics	Min	Typ	Max	Units	Comments
D300	V_IOFF	Input Offset Voltage*	—	±5.0	±25	mV	
D301	V_ICM	Input Common-Mode Voltage*	0	—	AV_DD – 1.5	V	
D302	CMRR	Common-Mode Rejection Ratio*	55	—	—	dB	
300	T_RESP	Response Time[1]*	—	150	400	ns	
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 (AV_DD – 1.5)/2, while the other input transitions from V_SS to AV_DD.

TABLE 28-3: VOLTAGE REFERENCE SPECIFICATIONS

Operating Conditions: $3.0V \leq V_{DD} \leq 3.6V$, $-40°C \leq T_A \leq +85°C$ (unless otherwise stated)							
Param 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	LSb	
D312	V_RUR	Unit Resistor Value (R)	—	2k	—	Ω	
310	T_SET	Settling Time[1]	—	—	10	μs	

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

TABLE 28-4: INTERNAL VOLTAGE REGULATOR SPECIFICATIONS

Operating Conditions: $-40°C \leq T_A \leq +85°C$ (unless otherwise stated)							
Param No.	Sym	Characteristics	Min	Typ	Max	Units	Comments
	V_RGOUT	Regulator Output Voltage	—	2.5	—	V	
	C_F	External Filter Capacitor Value	1	10	—	μF	Capacitor must be low series resistance

NOTES: