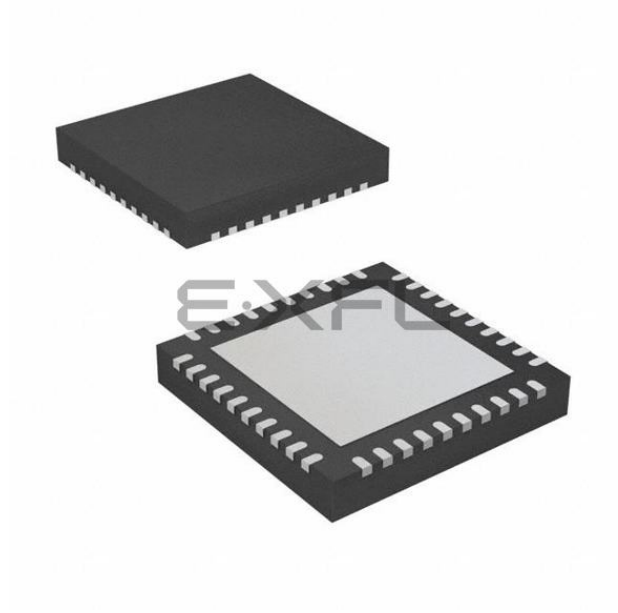


**Welcome to E-XFL.COM**

## 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 | 32MHz |
| Connectivity | I²C, LINbus, SPI, UART/USART |
| Peripherals | Brown-out Detect/Reset, POR, PWM, WDT |
| Number of I/O | 36 |
| Program Memory Size | 14KB (8K x 14) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 1K x 8 |
| Voltage - Supply (Vcc/Vdd) | 2.3V ~ 5.5V |
| Data Converters | A/D 35x10b; D/A 1x5b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 40-UFQFN Exposed Pad |
| Supplier Device Package | 40-UQFN (5x5) |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/pic16f15375-i-mv |

**TABLE 4: 40/44-PIN ALLOCATION TABLE (PIC16(L)F15375, PIC16(L)F15376)**

| I/O(2) | 40-Pin PDIP | 40-Pin UQFN | 44-Pin QFN | 44-Pin TQFP | ADC | Reference | Comparator | NCO | DAC | Timers | CCP | PWM | CWG | MSSP | ZCD | EUSART | CLC | CLKR | Interrupt | Pull-up | Basic |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RA0 | 2 | 17 | 19 | 19 | ANA0 | — | C1IN0-<br>C2IN0- | — | — | — | — | — | — | — | — | — | CLCIN0(1) | — | IOCA0 | Y | — |
| RA1 | 3 | 18 | 20 | 20 | ANA1 | — | C1IN1-<br>C2IN1- | — | — | — | — | — | — | — | — | — | CLCIN1(1) | — | IOCA1 | Y | — |
| RA2 | 4 | 19 | 21 | 21 | ANA2 | — | C1IN0+<br>C2IN0+ | — | DAC1OUT1 | — | — | — | — | — | — | — | — | — | IOCA2 | Y | — |
| RA3 | 5 | 20 | 22 | 22 | ANA3 | VREF+ | C1IN1+ | — | DACREF+ | — | — | — | — | — | — | — | — | — | IOCA3 | Y | — |
| RA4 | 6 | 21 | 23 | 23 | ANA4 | — | — | — | — | T0CKI(1) | — | — | — | — | — | — | — | — | IOCA4 | Y | — |
| RA5 | 7 | 22 | 24 | 24 | ANA5 | — | — | — | — | T1G(1) | — | — | — | $\overline{SS1}$(1) | — | — | — | — | IOCA5 | Y | — |
| RA6 | 14 | 29 | 33 | 31 | ANA6 | — | — | — | — | — | — | — | — | — | — | — | — | — | IOCA6 | Y | CLKOUT/<br>OSC1 |
| RA7 | 13 | 28 | 32 | 30 | ANA7 | — | — | — | — | — | — | — | — | — | — | — | — | — | IOCA7 | Y | CLKIN/<br>OSC2 |
| RB0 | 33 | 8 | 9 | 8 | ANB0 | — | C2IN1+ | — | — | — | — | — | CWG1(1) | $\overline{SS2}$(1) | ZCD1 | — | — | — | INT(1)<br>IOCB0 | Y | — |
| RB1 | 34 | 9 | 10 | 9 | ANB1 | — | C1IN3-<br>C2IN3- | — | — | — | — | — | — | SCL1<br>SCK1(1,4) | — | — | — | — | IOCB1 | Y | — |
| RB2 | 34 | 10 | 11 | 10 | ANB2 | — | — | — | — | — | — | — | — | SDA1<br>SDI1(1,4) | — | — | — | — | IOCB2 | Y | — |
| RB3 | 36 | 11 | 12 | 11 | ANB3 | — | C1IN2-<br>C2IN2- | — | — | — | — | — | — | — | — | — | — | — | IOCB3 | Y | — |
| RB4 | 37 | 12 | 14 | 14 | ANB4<br>ADACT(1) | — | — | — | — | — | — | — | — | — | — | — | — | — | IOCB4 | Y | — |
| RB5 | 38 | 13 | 15 | 15 | ANB5 | — | — | — | — | — | — | — | — | — | — | — | — | — | IOCB5 | Y | — |
| RB6 | 39 | 14 | 16 | 16 | ANB6 | — | — | — | — | — | — | — | — | — | — | TX2<br>CK2(1) | CLCIN2(1) | — | IOCB6 | Y | ICSPCLK |
| RB7 | 40 | 15 | 17 | 17 | ANB7 | — | — | — | DAC1OUT2 | — | — | — | — | — | — | RX2<br>DT2(1) | CLCIN3(1) | — | IOCB7 | Y | ICSPDAT |
| RC0 | 15 | 30 | 34 | 32 | ANC0 | — | — | — | — | SOSCO<br>T1CKI(1) | — | — | — | — | — | — | — | — | IOCC0 | Y | — |
| RC1 | 16 | 31 | 35 | 35 | ANC1 | — | — | — | — | SOSCI | CCP2(1) | — | — | — | — | — | — | — | IOCC1 | Y | — |
| RC2 | 17 | 32 | 36 | 36 | ANC2 | — | — | — | — | — | CCP1(1) | — | — | — | — | — | — | — | IOCC2 | Y | — |

**Note 1:** This is a PPS re-mappable input signal. The input function may be moved from the default location shown to one of several other PORTx pins.

**2:** All digital output signals shown in this row are PPS remappable. These signals may be mapped to output onto one of several PORTx pin options.

**3:** This is a bidirectional signal. For normal module operation, the firmware should map this signal to the same pin in both the PPS input and PPS output registers.

**4:** These pins are configured for $I^2C$ logic levels. PPS assignments to the other pins will operate, but input logic levels will be standard TTL/ST as selected by the INLVL register, instead of the $I^2C$ specific or SMBUS input buffer thresholds.

## 2.5 External Oscillator Pins

Many microcontrollers have options for at least two oscillators: a high-frequency primary oscillator and a low-frequency secondary oscillator (refer to **Section 9.0 "Oscillator Module (with Fail-Safe Clock Monitor)"** for details).

The oscillator circuit should be placed on the same side of the board as the device. Place the oscillator circuit close to the respective oscillator pins with no more than 0.5 inch (12 mm) between the circuit components and the pins. The load capacitors should be placed next to the oscillator itself, on the same side of the board.

Use a grounded copper pour around the oscillator circuit to isolate it from surrounding circuits. The grounded copper pour should be routed directly to the MCU ground. Do not run any signal traces or power traces inside the ground pour. Also, if using a two-sided board, avoid any traces on the other side of the board where the crystal is placed.

Layout suggestions are shown in Figure 2-3. In-line packages may be handled with a single-sided layout that completely encompasses the oscillator pins. With fine-pitch packages, it is not always possible to completely surround the pins and components. A suitable solution is to tie the broken guard sections to a mirrored ground layer. In all cases, the guard trace(s) must be returned to ground.

In planning the application's routing and I/O assignments, ensure that adjacent port pins, and other signals in close proximity to the oscillator, are benign (i.e., free of high frequencies, short rise and fall times, and other similar noise).

For additional information and design guidance on oscillator circuits, refer to these Microchip Application Notes, available at the corporate website (www.microchip.com):

- *AN826,* "*Crystal Oscillator Basics and Crystal Selection for rfPIC™ and PICmicro® Devices"*
- *AN849, "Basic PICmicro® Oscillator Design"*
- *AN943, "Practical PICmicro® Oscillator Analysis and Design"*
- *AN949, "Making Your Oscillator Work"*

## 2.6 Unused I/Os

Unused I/O pins should be configured as outputs and driven to a logic low state. Alternatively, connect a 1 kΩ to 10 kΩ resistor to VSS on unused pins and drive the output to logic low.

**FIGURE 2-3: SUGGESTED PLACEMENT OF THE OSCILLATOR CIRCUIT**

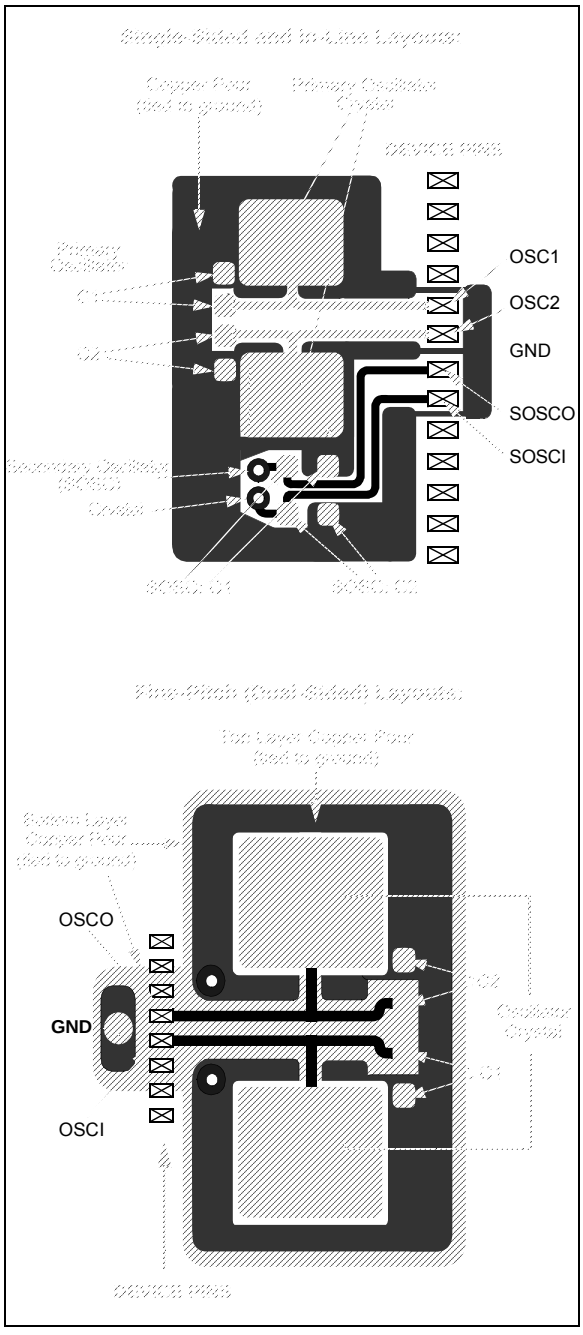


**TABLE 4-11: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-63 (CONTINUED)**

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on: POR, BOR | Value on: MCLR |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Bank 5** | | | | | | | | | | | |
| CPU CORE REGISTERS; see Table 4-3 for specifics | | | | | | | | | | | |
| 28Ch | T2TMR | Holding Register for the 8-bit TMR2 Register | | | | | | | | 0000 0000 | 0000 0000 |
| 28Dh | T2PR | TMR2 Period Register | | | | | | | | 1111 1111 | 1111 1111 |
| 28Eh | T2CON | ON | CKPS<2:0> | | | OUTPS<3:0> | | | | 0000 0000 | 0000 0000 |
| 28Fh | T2HLT | PSYNC | CKPOL | CKSYNC | MODE<4:0> | | | | | 0000 0000 | 0000 0000 |
| 290h | T2CLKCON | — | — | — | — | CS<3:0> | | | | ---- 0000 | ---- 0000 |
| 291h | T2RST | — | — | — | — | RSEL<3:0> | | | | ---- 0000 | ---- 0000 |
| 292h — 29Fh | — | Unimplemented | | | | | | | | — | — |

**Legend:** x = unknown, u = unchanged, q = depends on condition, - = unimplemented, read as '0', r = reserved. Shaded locations unimplemented, read as '0'.

**TABLE 4-11: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-63 (CONTINUED)**

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on: POR, BOR | Value on: MCLR |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Bank 60 (Continued)** | | | | | | | | | | | |
| 1E2Bh | CLC3GLS1 | LC3G2D4T | LC3G4D3N | LC3G2D3T | LC3G2D3N | LC3G2D2T | LC3G2D2N | LC3G2D1T | LC3G2D1N | xxxx xxxx | uuuu uuuu |
| 1E2Ch | CLC3GLS2 | LC3G3D4T | LC3G4D3N | LC3G3D3T | LC3G3D3N | LC3G3D2T | LC3G3D2N | LC3G3D1T | LC3G3D1N | xxxx xxxx | uuuu uuuu |
| 1E2Dh | CLC3GLS3 | LC3G4D4T | LC3G4D3N | LC3G4D3T | LC3G4D3N | LC3G4D2T | LC3G4D2N | LC3G4D1T | LC3G4D1N | xxxx xxxx | uuuu uuuu |
| 1E2Eh | CLC4CON | LC4EN | — | LC4OUT | LC4INTP | LC4INTN | LC4MODE<2:0> | | | 0-00 0000 | 0-00 0000 |
| 1E2Fh | CLC4POL | LC4POL | — | — | — | LC4G4POL | LC4G3POL | LC4G2POL | LC4G1POL | 0--- xxxx | 0--- uuuu |
| 1E30h | CLC4SEL0 | — | — | LC4D1S<5:0> | | | | | | --xx xxxx | --uu uuuu |
| 1E31h | CLC4SEL1 | — | — | LC4D2S<5:0> | | | | | | --xx xxxx | --uu uuuu |
| 1E32h | CLC4SEL2 | — | — | LC4D3S<5:0> | | | | | | --xx xxxx | --uu uuuu |
| 1E33h | CLC4SEL3 | — | — | LC4D4S<5:0> | | | | | | --xx xxxx | --uu uuuu |
| 1E34h | CLC4GLS0 | LC4G1D4T | LC4G4D3N | LC4G1D3T | LC4G1D3N | LC4G1D2T | LC4G1D2N | LC4G1D1T | LC4G1D1N | xxxx xxxx | uuuu uuuu |
| 1E35h | CLC4GLS1 | LC4G2D4T | LC4G4D3N | LC4G2D3T | LC4G2D3N | LC4G2D2T | LC4G2D2N | LC4G2D1T | LC4G2D1N | xxxx xxxx | uuuu uuuu |
| 1E36h | CLC4GLS2 | LC4G3D4T | LC4G4D3N | LC4G3D3T | LC4G3D3N | LC4G3D2T | LC4G3D2N | LC4G3D1T | LC4G3D1N | xxxx xxxx | uuuu uuuu |
| 1E37h | CLC4GLS3 | LC4G4D4T | LC4G4D3N | LC4G4D3T | LC4G4D3N | LC4G4D2T | LC4G4D2N | LC4G4D1T | LC4G4D1N | xxxx xxxx | uuuu uuuu |
| 1E38h | RF0PPS(1) | — | — | — | RF0PPS<4:0> | | | | | ---0 0000 | ---u uuuu |
| 1E39h | RF1PPS(1) | — | — | — | RF1PPS<4:0> | | | | | ---0 0000 | ---u uuuu |
| 1E3Ah | RF2PPS(1) | — | — | — | RF2PPS<4:0> | | | | | ---0 0000 | ---u uuuu |
| 1E3Bh | RF3PPS(1) | — | — | — | RF3PPS<4:0> | | | | | ---0 0000 | ---u uuuu |
| 1E3Ch | RF4PPS(1) | — | — | — | RF4PPS<4:0> | | | | | ---0 0000 | ---u uuuu |
| 1E3Dh | RF5PPS(1) | — | — | — | RF5PPS<4:0> | | | | | ---0 0000 | ---u uuuu |
| 1E3Eh | RF6PPS(1) | — | — | — | RF6PPS<4:0> | | | | | ---0 0000 | ---u uuuu |
| 1E3Fh | RF7PPS(1) | — | — | — | RF7PPS<4:0> | | | | | ---0 0000 | ---u uuuu |
| 1E40h — 1E4Fh | — | Unimplemented | | | | | | | | — | — |
| 1E50h | ANSELF(1) | ANSF7 | ANSF6 | ANSF5 | ANSF4 | ANSF3 | ANSF2 | ANSF1 | ANSF0 | 1111 1111 | 1111 1111 |
| 1E51h | WPUF(1) | WPUF7 | WPUF6 | WPUF5 | WPUF4 | WPUF3 | WPUF2 | WPUF1 | WPUF0 | 0000 0000 | 0000 0000 |
| 1E52h | ODCONF(1) | ODCF7 | ODCF6 | ODCF5 | ODCF4 | ODCF3 | ODCF2 | ODCF1 | ODCFCF0 | 0000 0000 | 0000 0000 |
| 1E53h | SLRCONF(1) | SLRF7 | SLRF6 | SLRF5 | SLRF4 | SLRF3 | SLRF2 | SLRF1 | SLRF0 | 1111 1111 | 1111 1111 |
| 1E54h | INLVLF(1) | INLVLF7 | INLVLF6 | INLVLF5 | INLVLF4 | INLVLF3 | INLVLF2 | INLVLF1 | INLVLF0 | 1111 1111 | 1111 1111 |
| 1E55h — 1E6Fh | — | Unimplemented | | | | | | | | — | — |

**Legend:** x = unknown, u = unchanged, q = depends on condition, - = unimplemented, read as '0', r = reserved. Shaded locations unimplemented, read as '0'.

**Note 1:** Present only on PIC16(L)F15385/86.

**REGISTER 5-4: CONFIGURATION WORD 4: MEMORY**

| R/W-1 | U-1 | R/W-1 | U-1 | R/W-1 | R/W-1 |
|---|---|---|---|---|---|
| LVP | — | WRTSAF(1) | — | WRTC(1) | WRTB(1) |
| bit 13 | 12 | 11 | 10 | 9 | bit 8 |

| R/W-1 | U-1 | U-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 |
|---|---|---|---|---|---|---|---|
| WRTAPP(1) | — | — | SAFEN(1) | BBEN(1) | BBSIZE2 | BBSIZE1 | BBSIZE0 |
| bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | bit 0 |

| **Legend:** | | | |
|---|---|---|---|
| R = Readable bit | P = Programmable bit | x = Bit is unknown | U = Unimplemented bit, read as '1' |
| '0' = Bit is cleared | '1' = Bit is set | W = Writable bit | n = Value when blank or after Bulk Erase |

bit 13 **LVP:** Low Voltage Programming Enable bit
1 = Low voltage programming enabled. MCLR/VPP pin function is MCLR. MCLRE Configuration bit is ignored.
0 = HV on MCLR/VPP must be used for programming.
The LVP bit cannot be written (to zero) while operating from the LVP programming interface. The purpose of this rule is to prevent the user from dropping out of LVP mode while programming from LVP mode, or accidentally eliminating LVP mode from the configuration state.
The preconditioned (erased) state for this bit is critical.

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

bit 11 **WRTSAF:** Storage Area Flash Write Protection bit
1 = SAF NOT write-protected
0 = SAF write-protected
Unimplemented, if SAF is not supported in the device family and only applicable if SAFEN = 0.

bit 10 **Unimplemented:** Read as '1'

bit 9 **WRTC:** Configuration Register Write Protection bit
1 = Configuration Register NOT write-protected
0 = Configuration Register write-protected

bit 8 **WRTB:** Boot Block Write Protection bit
1 = Boot Block NOT write-protected
0 = Boot Block write-protected
Only applicable if BBEN = 0.

bit 7 **WRTAPP:** Application Block Write Protection bit
1 = Application Block NOT write-protected
0 = Application Block write-protected

bit 6-5 **Unimplemented:** Read as '1'

bit 4 **SAFEN:** SAF Enable bit
1 = SAF disabled
0 = SAF enabled

bit 3 **BBEN:** Boot Block Enable bit
1 = Boot Block disabled
0 = Boot Block enabled

bit 2-0 **BBSIZE[2:0]:** Boot Block Size Selection bits
BBSIZE is used only when BBEN = 0
BBSIZ bits can only be written while BBEN = 1; after BBEN = 0, BBSIZ is write-protected.

**Note 1:** Bits are implemented as sticky bits. Once protection is enabled, it can only be reset through a Bulk Erase.

**REGISTER 9-5: OSCEN: OSCILLATOR MANUAL ENABLE REGISTER**

| R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | U-0 | U-0 |
|---|---|---|---|---|---|---|---|
| EXTOEN | HFOEN | MFOEN | LFOEN | SOSCEN | ADOEN | — | — |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | |

bit 7 **EXTOEN:** External Oscillator Manual Request Enable bit(1)
1 = EXTOSC is explicitly enabled, operating as specified by FEXTOSC
0 = EXTOSC could be enabled by some modules

bit 6 **HFOEN:** HFINTOSC Oscillator Manual Request Enable bit
1 = HFINTOSC is explicitly enabled, operating as specified by OSCFRQ
0 = HFINTOSC could be enabled by another module

bit 5 **MFOEN:** MFINTOSC Oscillator Manual Request Enable bit
1 = MFINTOSC is explicitly enabled
0 = MFINTOSC could be enabled by another module

bit 4 **LFOEN:** LFINTOSC (31 kHz) Oscillator Manual Request Enable bit
1 = LFINTOSC is explicitly enabled
0 = LFINTOSC could be enabled by another module

bit 3 **SOSCEN:** Secondary (Timer1) Oscillator Manual Request bit
1 = Secondary oscillator is explicitly enabled, operating as specified by SOSCPWR
0 = Secondary oscillator could be enabled by another module

bit 2 **ADOEN:** FRC Oscillator Manual Request Enable bit
1 = FRC is explicitly enabled
0 = FRC could be enabled by another module

bit 1-0 **Unimplemented:** Read as '0'

**REGISTER 10-6: PIE4: PERIPHERAL INTERRUPT ENABLE REGISTER 4**

| U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| — | — | — | — | — | — | TMR2IE | TMR1IE |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | HS = Hardware set |

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

bit 1 **TMR2IE:** TMR2 to PR2 Match Interrupt Enable bit
- 1 = Enables the Timer2 to PR2 match interrupt
- 0 = Disables the Timer2 to PR2 match interrupt

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

**Note:** Bit PEIE of the INTCON register must be set to enable any peripheral interrupt controlled by registers PIE1-PIE7.

## 14.0 I/O PORTS

**TABLE 14-1: PORT AVAILABILITY PER DEVICE**

| Device | PORTA | PORTB | PORTC | PORTD | PORTE | PORTF |
|---|---|---|---|---|---|---|
| PIC16(L)F15356 | ● | ● | ● | | ● | |
| PIC16(L)F15375/76 | ● | ● | ● | ● | ● | |
| PIC16(L)F15385/86 | ● | ● | ● | ● | ● | ● |

Each port has ten standard registers for its operation. These registers are:

- PORTx registers (reads the levels on the pins of the device)
- LATx registers (output latch)
- TRISx registers (data direction)
- ANSELx registers (analog select)
- WPUx registers (weak pull-up)
- INLVLx (input level control)
- SLRCONx registers (slew rate)
- ODCONx registers (open-drain)

Most port pins share functions with device peripherals, both analog and digital. In general, when a peripheral is enabled on a port pin, that pin cannot be used as a general purpose output; however, the pin can still be read.

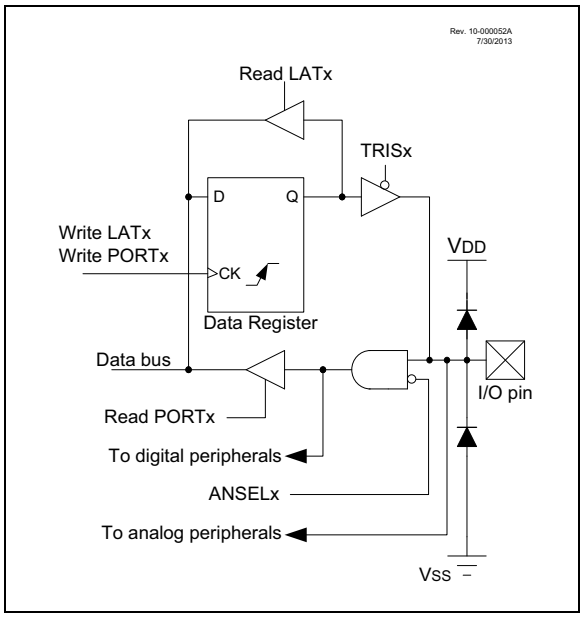
The Data Latch (LATx registers) is useful for read-modify-write operations on the value that the I/O pins are driving.

A write operation to the LATx register has the same effect as a write to the corresponding PORTx register. A read of the LATx register reads of the values held in the I/O PORT latches, while a read of the PORTx register reads the actual I/O pin value.

Ports that support analog inputs have an associated ANSELx register. When an ANSEL bit is set, the digital input buffer associated with that bit is disabled.

Disabling the input buffer prevents analog signal levels on the pin between a logic high and low from causing excessive current in the logic input circuitry. A simplified model of a generic I/O port, without the interfaces to other peripherals, is shown in Figure 14-1.

**FIGURE 14-1: GENERIC I/O PORT OPERATION**




### 14.1 I/O Priorities

Each pin defaults to the PORT data latch after Reset. Other functions are selected with the peripheral pin select logic. See **Section 15.0 "Peripheral Pin Select (PPS) Module"** for more information.

Analog input functions, such as ADC and comparator inputs, are not shown in the peripheral pin select lists. These inputs are active when the I/O pin is set for Analog mode using the ANSELx register. Digital output functions may continue to control the pin when it is in Analog mode.

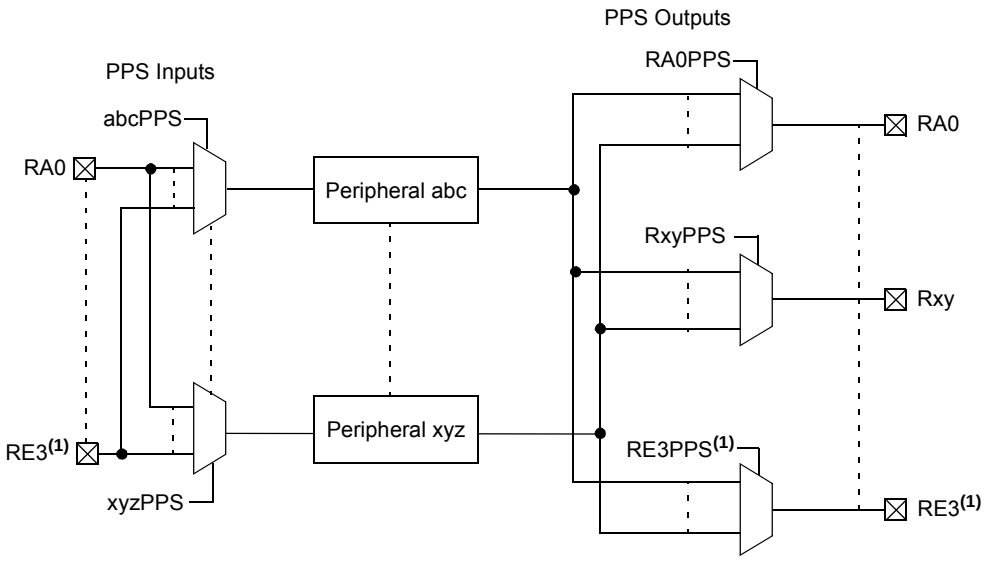
Analog outputs, when enabled, take priority over the digital outputs and force the digital output driver to the high-impedance state.

## 15.0 PERIPHERAL PIN SELECT (PPS) MODULE

The Peripheral Pin Select (PPS) module connects peripheral inputs and outputs to the device I/O pins. Only digital signals are included in the selections.

All analog inputs and outputs remain fixed to their assigned pins. Input and output selections are independent as shown in the simplified block diagram Figure 15-1.

**FIGURE 15-1: SIMPLIFIED PPS BLOCK DIAGRAM**




**Note 1:** RE3 is PPS input capable only (when $\overline{\text{MLCR}}$ is disabled).

### 15.1 PPS Inputs

Each peripheral has a PPS register with which the inputs to the peripheral are selected. Inputs include the device pins.

Although every peripheral has its own PPS input selection register, the selections are identical for every peripheral as shown in Register 15-1.

| **Note:** | The notation “xxx” in the register name is a place holder for the peripheral identifier. For example, CLC1PPS. |
|---|---|

### 15.2 PPS Outputs

Each I/O pin has a PPS register with which the pin output source is selected. With few exceptions, the port TRIS control associated with that pin retains control over the pin output driver. Peripherals that control the pin output driver as part of the peripheral operation will override the TRIS control as needed. These peripherals are (See **Section 15.3 “Bidirectional Pins”**):

- EUSART (synchronous operation)
- MSSP ($I^2C$)

Although every pin has its own PPS peripheral selection register, the selections are identical for every pin as shown in Register 15-2.

| **Note:** | The notation “Rxy” is a place holder for the pin port and bit identifiers. For example, x and y for PORTA bit 0 would be A and 0, respectively, resulting in the pin PPS output selection register RA0PPS. |
|---|---|

**REGISTER 22-2: NCO1CLK: NCO1 INPUT CLOCK CONTROL REGISTER**

| R/W-0/0 | R/W-0/0 | R/W-0/0 | U-0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| N1PWS<2:0>(1,2) | | | — | N1CKS<3:0> | | | |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | |

bit 7-5 **N1PWS<2:0>:** NCO1 Output Pulse Width Select bits(1)
- 111 = NCO1 output is active for 128 input clock periods
- 110 = NCO1 output is active for 64 input clock periods
- 101 = NCO1 output is active for 32 input clock periods
- 100 = NCO1 output is active for 16 input clock periods
- 011 = NCO1 output is active for 8 input clock periods
- 010 = NCO1 output is active for 4 input clock periods
- 001 = NCO1 output is active for 2 input clock periods
- 000 = NCO1 output is active for 1 input clock period

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

bit 3-0 **N1CKS<3:0>:** NCO1 Clock Source Select bits
- 1011-1111 = Reserved
- 1010 = LC4_out
- 1001 = LC3_out
- 1000 = LC2_out
- 0111 = LC1_out
- 0110 = CLKR
- 0101 = SOSC
- 0100 = MFINTOSC (32 kHz)
- 0011 = MFINTOSC (500 kHz)
- 0010 = LFINTOSC
- 0001 = HFINTOSC
- 0000 = FOSC

**Note 1:** N1PWS applies only when operating in Pulse Frequency mode.

## 23.12 Register Definitions: Comparator Control

**REGISTER 23-1: CMxCON0: COMPARATOR Cx CONTROL REGISTER 0**

| R/W-0/0 | R-0/0 | U-0 | R/W-0/0 | U-0 | U-0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| ON | OUT | — | POL | — | — | HYS | SYNC |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | |

bit 7 **ON:** Comparator Enable bit
1 = Comparator is enabled
0 = Comparator is disabled and consumes no active power

bit 6 **OUT:** Comparator Output bit
If CxPOL = 1 (inverted polarity):
1 = CxVP < CxVN
0 = CxVP > CxVN
If CxPOL = 0 (noninverted polarity):
1 = CxVP > CxVN
0 = CxVP < CxVN

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

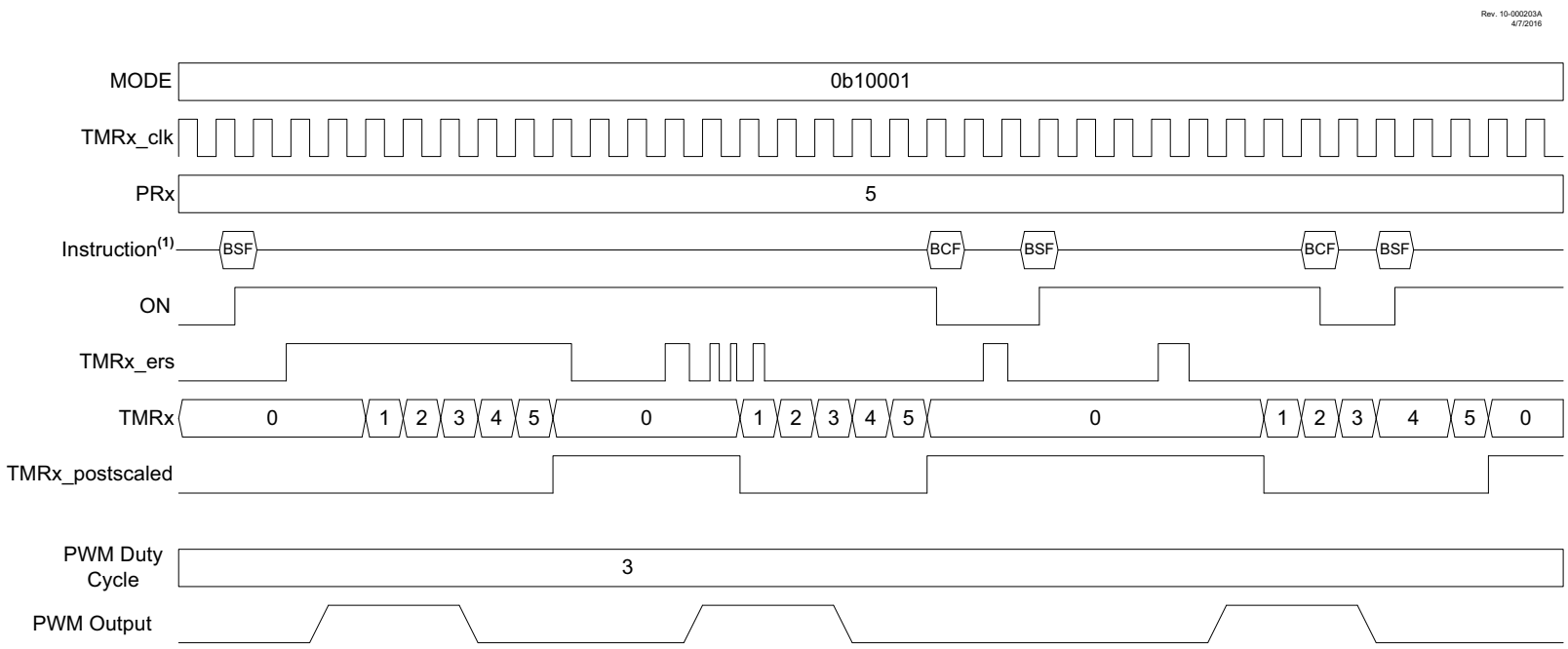
bit 4 **POL:** Comparator Output Polarity Select bit
1 = Comparator output is inverted
0 = Comparator output is not inverted

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

bit 1 **HYS:** Comparator Hysteresis Enable bit
1 = Comparator hysteresis enabled
0 = Comparator hysteresis disabled

bit 0 **SYNC:** Comparator Output Synchronous Mode bit
1 = Comparator output to Timer1 and I/O pin is synchronous to changes on Timer1 clock source. Output updated on the falling edge of Timer1 clock source.
0 = Comparator output to Timer1 and I/O pin is asynchronous

**FIGURE 27-12: RISING EDGE-TRIGGERED MONOSTABLE MODE TIMING DIAGRAM (MODE = 10001)**




Note 1: BSF and BCF represent Bit-Set File and Bit-Clear File instructions executed by the CPU to set or clear the ON bit of TxCON. CPU execution is asynchronous to the timer clock input.

**REGISTER 28-2: CCPxCAP: CAPTURE INPUT SELECTION REGISTER**

| U-0 | U-0 | U-0 | U-0 | U-0 | R/W-0/x | R/W-0/x | R/W-0/x |
|---|---|---|---|---|---|---|---|
| — | — | — | — | — | CTS<2:0> | | |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Reset |
| ‘1’ = Bit is set | ‘0’ = Bit is cleared | |

bit 7-3 **Unimplemented:** Read as ‘0’

bit 2-0 **CTS<2:0>:** Capture Trigger Input Selection bits

| CTS | CCP1.capture | CCP2.capture |
|---|---|---|
| 111 | LC4_out | |
| 110 | LC3_out | |
| 101 | LC2_out | |
| 100 | LC1_out | |
| 011 | IOC_interrupt | |
| 010 | C2OUT | |
| 001 | C1OUT | |
| 000 | CCP1PPS | CCP2PPS |

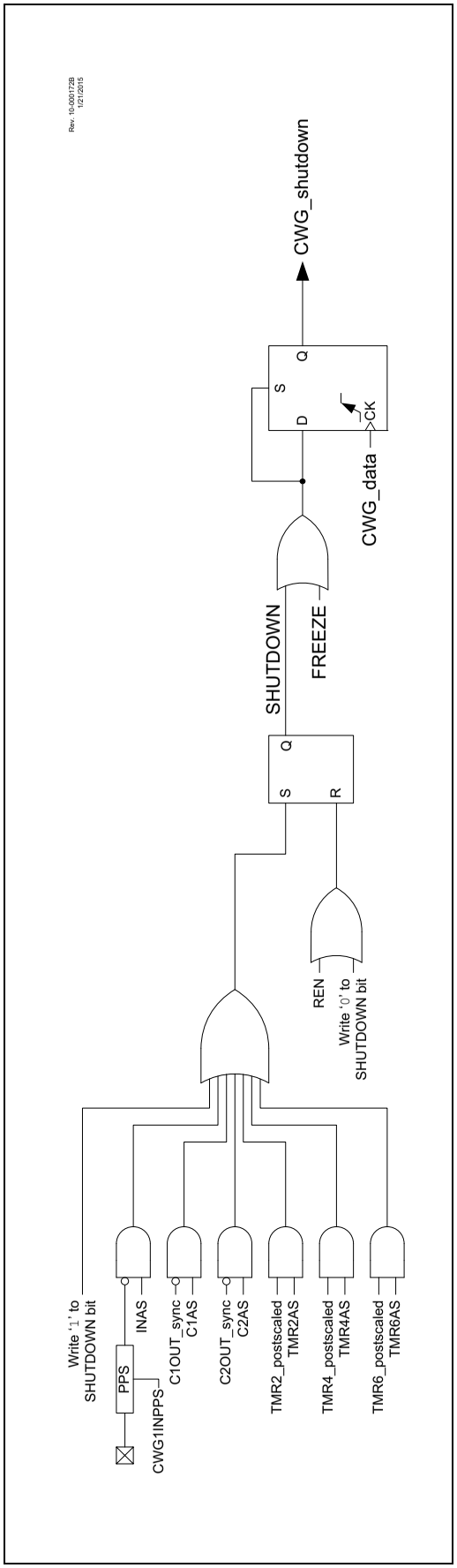
**REGISTER 28-3: CCPRxL REGISTER: CCPx REGISTER LOW BYTE**

| R/W-x/x | R/W-x/x | R/W-x/x | R/W-x/x | R/W-x/x | R/W-x/x | R/W-x/x | R/W-x/x |
|---|---|---|---|---|---|---|---|
| CCPRx<7:0> | | | | | | | |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Reset |
| ‘1’ = Bit is set | ‘0’ = Bit is cleared | |

bit 7-0 CCPxMODE = Capture mode
**CCPRxL<7:0>:** Capture value of TMR1L
CCPxMODE = Compare mode
**CCPRxL<7:0>:** LS Byte compared to TMR1L
CCPxMODE = PWM modes when CCPxFMT = 0:
**CCPRxL<7:0>:** Pulse-width Least Significant eight bits
CCPxMODE = PWM modes when CCPxFMT = 1:
**CCPRxL<7:6>:** Pulse-width Least Significant two bits
**CCPRxL<5:0>:** Not used.

**FIGURE 30-12: CWG SHUTDOWN BLOCK DIAGRAM**



**REGISTER 31-11: CLCDATA: CLC DATA OUTPUT**

| U-0 | U-0 | U-0 | U-0 | R-0 | R-0 | R-0 | R-0 |
|---|---|---|---|---|---|---|---|
| — | — | — | — | MLC4OUT | MLC3OUT | MLC2OUT | MLC1OUT |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| ‘1’ = Bit is set | ‘0’ = Bit is cleared | |

bit 7-4 **Unimplemented:** Read as ‘0’

bit 3 **MLC4OUT:** Mirror copy of LC4OUT bit

bit 2 **MLC3OUT:** Mirror copy of LC3OUT bit

bit 1 **MLC2OUT:** Mirror copy of LC2OUT bit

bit 0 **MLC1OUT:** Mirror copy of LC1OUT bit

#### 33.1.2.3 Receive Interrupts

The RXxIF interrupt flag bit of the PIR3 register is set whenever the EUSART receiver is enabled and there is an unread character in the receive FIFO. The RXxIF interrupt flag bit is read-only, it cannot be set or cleared by software.

RXxIF interrupts are enabled by setting all of the following bits:

- RXxIE, Interrupt Enable bit of the PIE3 register
- PEIE, Peripheral Interrupt Enable bit of the INTCON register
- GIE, Global Interrupt Enable bit of the INTCON register

The RXxIF interrupt flag bit will be set when there is an unread character in the FIFO, regardless of the state of interrupt enable bits.

#### 33.1.2.4 Receive Framing Error

Each character in the receive FIFO buffer has a corresponding framing error Status bit. A framing error indicates that a Stop bit was not seen at the expected time. The framing error status is accessed via the FERR bit of the RCxSTA register. The FERR bit represents the status of the top unread character in the receive FIFO. Therefore, the FERR bit must be read before reading the RCxREG.

The FERR bit is read-only and only applies to the top unread character in the receive FIFO. A framing error (FERR = 1) does not preclude reception of additional characters. It is not necessary to clear the FERR bit. Reading the next character from the FIFO buffer will advance the FIFO to the next character and the next corresponding framing error.

The FERR bit can be forced clear by clearing the SPEN bit of the RCxSTA register which resets the EUSART. Clearing the CREN bit of the RCxSTA register does not affect the FERR bit. A framing error by itself does not generate an interrupt.

| | |
|---|---|
| **Note:** | If all receive characters in the receive FIFO have framing errors, repeated reads of the RCxREG will not clear the FERR bit. |

#### 33.1.2.5 Receive Overrun Error

The receive FIFO buffer can hold two characters. An overrun error will be generated if a third character, in its entirety, is received before the FIFO is accessed. When this happens the OERR bit of the RCxSTA register is set. The characters already in the FIFO buffer can be read but no additional characters will be received until the error is cleared. The error must be cleared by either clearing the CREN bit of the RCxSTA register or by resetting the EUSART by clearing the SPEN bit of the RCxSTA register.

#### 33.1.2.6 Receiving 9-Bit Characters

The EUSART supports 9-bit character reception. When the RX9 bit of the RCxSTA register is set the EUSART will shift nine bits into the RSR for each character received. The RX9D bit of the RCxSTA register is the ninth and Most Significant data bit of the top unread character in the receive FIFO. When reading 9-bit data from the receive FIFO buffer, the RX9D data bit must be read before reading the eight Least Significant bits from the RCxREG.

#### 33.1.2.7 Address Detection

A special Address Detection mode is available for use when multiple receivers share the same transmission line, such as in RS-485 systems. Address detection is enabled by setting the ADDEN bit of the RCxSTA register.

Address detection requires 9-bit character reception. When address detection is enabled, only characters with the ninth data bit set will be transferred to the receive FIFO buffer, thereby setting the RXxIF interrupt bit. All other characters will be ignored.

Upon receiving an address character, user software determines if the address matches its own. Upon address match, user software must disable address detection by clearing the ADDEN bit before the next Stop bit occurs. When user software detects the end of the message, determined by the message protocol used, software places the receiver back into the Address Detection mode by setting the ADDEN bit.

**FIGURE 33-7: AUTO-WAKE-UP BIT (WUE) TIMING DURING NORMAL OPERATION**

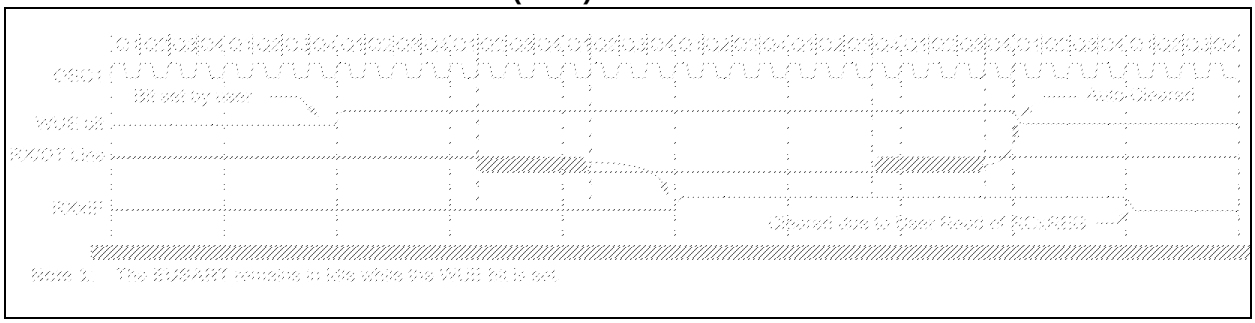
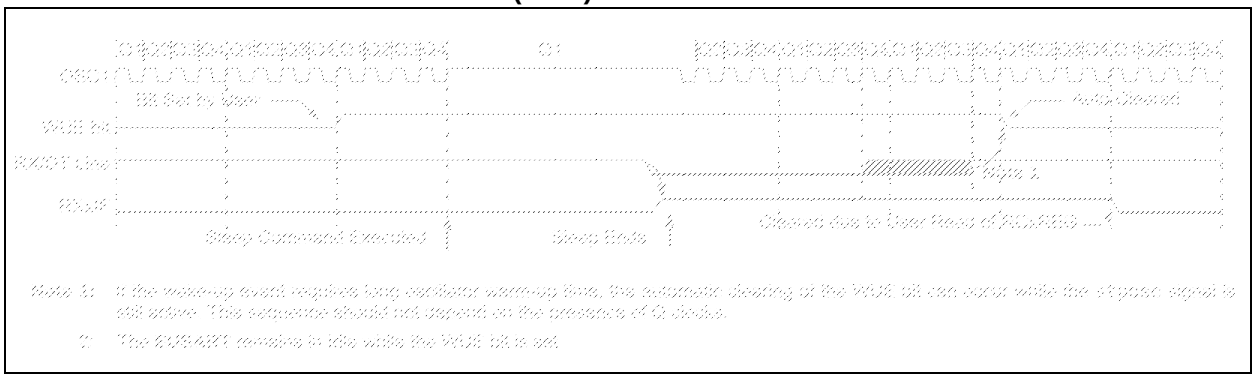


**FIGURE 33-8: AUTO-WAKE-UP BIT (WUE) TIMINGS DURING SLEEP**



### 33.3.4 BREAK CHARACTER SEQUENCE

The EUSART module has the capability of sending the special Break character sequences that are required by the LIN bus standard. A Break character consists of a Start bit, followed by 12 ‘0’ bits and a Stop bit.

To send a Break character, set the SENDB and TXEN bits of the TXxSTA register. The Break character transmission is then initiated by a write to the TXxREG. The value of data written to TXxREG will be ignored and all ‘0’s will be transmitted.

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

The TRMT bit of the TXxSTA register indicates when the transmit operation is active or idle, just as it does during normal transmission. See Figure 33-9 for the timing of the Break character sequence.

#### 33.3.4.1 Break and Sync Transmit Sequence

The following sequence will start a message frame header made up of a Break, followed by an auto-baud Sync byte. This sequence is typical of a LIN bus master.

1. Configure the EUSART for the desired mode.
2. Set the TXEN and SENDB bits to enable the Break sequence.
3. Load the TXxREG with a dummy character to initiate transmission (the value is ignored).
4. Write ‘55h’ to TXxREG to load the Sync character into the transmit FIFO buffer.
5. After the Break has been sent, the SENDB bit is reset by hardware and the Sync character is then transmitted.

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

#### 33.4.1.6 Slave Clock

Synchronous data transfers use a separate clock line, which is synchronous with the data. A device configured as a slave receives the clock on the TX/CK line. The TX/CK pin output driver is automatically disabled when the device is configured for synchronous slave transmit or receive operation. Serial data bits change on the leading edge to ensure they are valid at the trailing edge of each clock. One data bit is transferred for each clock cycle. Only as many clock cycles should be received as there are data bits.

| | |
|---|---|
| **Note:** | If the device is configured as a slave and the TX/CK function is on an analog pin, the corresponding ANSEL bit must be cleared. |

#### 33.4.1.7 Receive Overrun Error

The receive FIFO buffer can hold two characters. An overrun error will be generated if a third character, in its entirety, is received before RCxREG is read to access the FIFO. When this happens the OERR bit of the RCxSTA register is set. Previous data in the FIFO will not be overwritten. The two characters in the FIFO buffer can be read, however, no additional characters will be received until the error is cleared. The OERR bit can only be cleared by clearing the overrun condition. If the overrun error occurred when the SREN bit is set and CREN is clear then the error is cleared by reading RCxREG. If the overrun occurred when the CREN bit is set then the error condition is cleared by either clearing the CREN bit of the RCxSTA register or by clearing the SPEN bit which resets the EUSART.

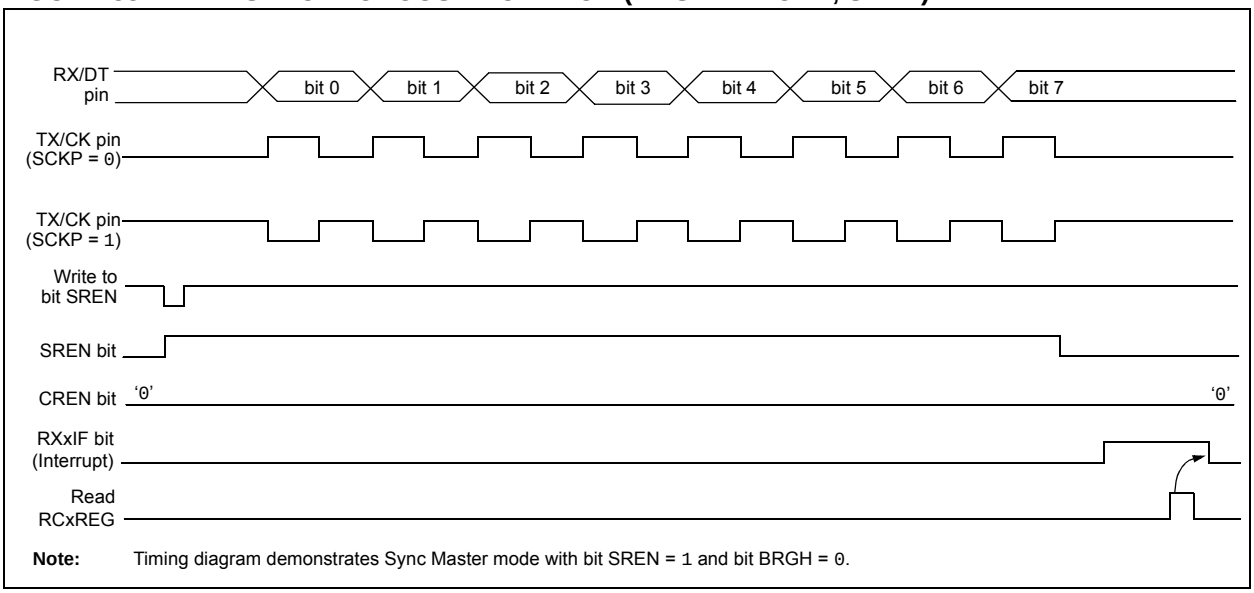
#### 33.4.1.8 Receiving 9-bit Characters

The EUSART supports 9-bit character reception. When the RX9 bit of the RCxSTA register is set the EUSART will shift nine bits into the RSR for each character received. The RX9D bit of the RCxSTA register is the ninth, and Most Significant, data bit of the top unread character in the receive FIFO. When reading 9-bit data from the receive FIFO buffer, the RX9D data bit must be read before reading the eight Least Significant bits from the RCxREG.

#### 33.4.1.9 Synchronous Master Reception Set-up:

1. Initialize the SPxBRGH, SPxBRGL register pair for the appropriate baud rate. Set or clear the BRGH and BRG16 bits, as required, to achieve the desired baud rate.
2. Clear the ANSEL bit for the RX pin (if applicable).
3. Enable the synchronous master serial port by setting bits SYNC, SPEN and CSRC.
4. Ensure bits CREN and SREN are clear.
5. If interrupts are desired, set the RXxIE bit of the PIE3 register and the GIE and PEIE bits of the INTCON register.
6. If 9-bit reception is desired, set bit RX9.
7. Start reception by setting the SREN bit or for continuous reception, set the CREN bit.
8. Interrupt flag bit RXxIF will be set when reception of a character is complete. An interrupt will be generated if the enable bit RXxIE was set.
9. Read the RCxSTA register to get the ninth bit (if enabled) and determine if any error occurred during reception.
10. Read the 8-bit received data by reading the RCxREG register.
11. If an overrun error occurs, clear the error by either clearing the CREN bit of the RCxSTA register or by clearing the SPEN bit which resets the EUSART.

**FIGURE 33-12: SYNCHRONOUS RECEPTION (MASTER MODE, SREN)**




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

## 36.0 INSTRUCTION SET SUMMARY

Each instruction is a 14-bit word containing the operation code (opcode) and all required operands. The opcodes are broken into three broad categories.

- Byte Oriented
- Bit Oriented
- Literal and Control

The literal and control category contains the most varied instruction word format.

Table 36-3 lists the instructions recognized by the MPASM™ assembler.

All instructions are executed within a single instruction cycle, with the following exceptions, which may take two or three cycles:

- Subroutine entry takes two cycles (CALL, CALLW)
- Returns from interrupts or subroutines take two cycles (RETURN, RETLW, RETFIE)
- Program branching takes two cycles (GOTO, BRA, BRW, BTFSS, BTFSC, DECFSZ, INCSFZ)
- One additional instruction cycle will be used when any instruction references an indirect file register and the file select register is pointing to program memory.

One instruction cycle consists of 4 oscillator cycles; for an oscillator frequency of 4 MHz, this gives a nominal instruction execution rate of 1 MHz.

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

### 36.1 Read-Modify-Write Operations

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

**TABLE 36-1: OPCODE FIELD DESCRIPTIONS**

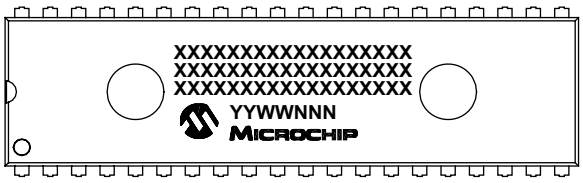
| Field | Description |
|---|---|
| f | Register file address (0x00 to 0x7F) |
| W | Working register (accumulator) |
| b | Bit address within an 8-bit file register |
| k | Literal field, constant data or label |
| x | Don't care location (= 0 or 1).<br>The assembler will generate code with x = 0.<br>It is the recommended form of use for compatibility with all Microchip software tools. |
| d | Destination select; d = 0: store result in W,<br>d = 1: store result in file register f.<br>Default is d = 1. |
| n | FSR or INDF number. (0-1) |
| mm | Prepost increment-decrement mode selection |

**TABLE 36-2: ABBREVIATION DESCRIPTIONS**

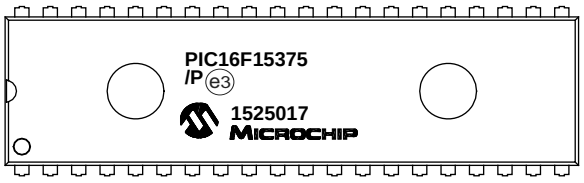
| Field | Description |
|---|---|
| PC | Program Counter |
| $\overline{TO}$ | Time-Out bit |
| C | Carry bit |
| DC | Digit Carry bit |
| Z | Zero bit |
| $\overline{PD}$ | Power-Down bit |

## 40.1 Package Marking Information (Continued)

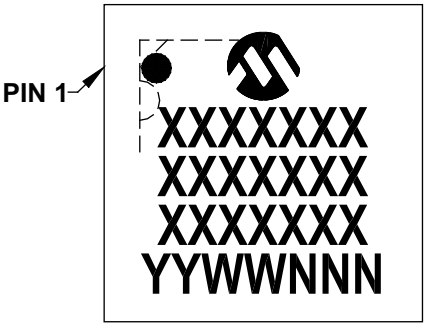
40-Lead PDIP (600 mil)



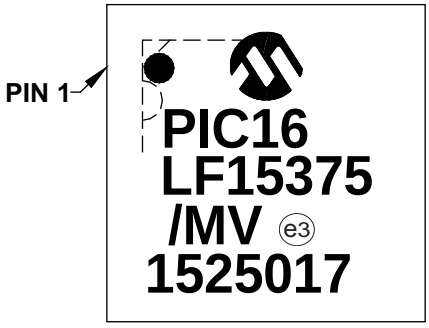

Example




40-Lead UQFN (5x5x0.5 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 |
| | | 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.