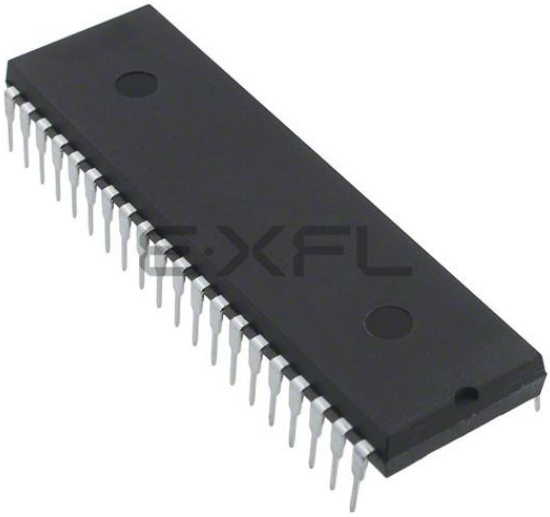


**Welcome to E-XFL.COM**

## 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 | 48MHz |
| Connectivity | I²C, SPI, UART/USART |
| Peripherals | Brown-out Detect/Reset, HLVD, POR, PWM, WDT |
| Number of I/O | 35 |
| Program Memory Size | 64KB (32K x 16) |
| Program Memory Type | FLASH |
| EEPROM Size | 1K x 8 |
| RAM Size | 3.8K x 8 |
| Voltage - Supply (Vcc/Vdd) | 2.3V ~ 5.5V |
| Data Converters | A/D 30x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 125°C (TA) |
| Mounting Type | Through Hole |
| Package / Case | 40-DIP (0.600", 15.24mm) |
| Supplier Device Package | 40-PDIP |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/pic18f46k22-e-p |

**TABLE 1-3: PIC18(L)F4XK22 PINOUT I/O DESCRIPTIONS (CONTINUED)**

| Pin Number | | | | Pin Name | Pin Type | Buffer Type | Description |
|---|---|---|---|---|---|---|---|
| PDIP | TQFP | QFN | UQFN | | | | |
| 37 | 14 | 14 | 12 | RB4/IOC0/T5G/AN11 | | | |
| | | | | RB4 | I/O | TTL | Digital I/O. |
| | | | | IOC0 | I | TTL | Interrupt-on-change pin. |
| | | | | T5G | I | ST | Timer5 external clock gate input. |
| | | | | AN11 | I | Analog | Analog input 11. |
| 38 | 15 | 15 | 13 | RB5/IOC1/P3A/CCP3/T3CKI/T1G/AN13 | | | |
| | | | | RB5 | I/O | TTL | Digital I/O. |
| | | | | IOC1 | I | TTL | Interrupt-on-change pin. |
| | | | | P3A(1) | O | CMOS | Enhanced CCP3 PWM output. |
| | | | | CCP3(1) | I/O | ST | Capture 3 input/Compare 3 output/PWM 3 output. |
| | | | | T3CKI(2) | I | ST | Timer3 clock input. |
| | | | | T1G | I | ST | Timer1 external clock gate input. |
| | | | | AN13 | I | Analog | Analog input 13. |
| 39 | 16 | 16 | 14 | RB6/IOC2/PGC | | | |
| | | | | RB6 | I/O | TTL | Digital I/O. |
| | | | | IOC2 | I | TTL | Interrupt-on-change pin. |
| | | | | PGC | I/O | ST | In-Circuit Debugger and ICSP™ programming clock pin. |
| 40 | 17 | 17 | 15 | RB7/IOC3/PGD | | | |
| | | | | RB7 | I/O | TTL | Digital I/O. |
| | | | | IOC3 | I | TTL | Interrupt-on-change pin. |
| | | | | PGD | I/O | ST | In-Circuit Debugger and ICSP™ programming data pin. |
| 15 | 32 | 34 | 30 | RC0/P2B/T3CKI/T3G/T1CKI/SOSCO | | | |
| | | | | RC0 | I/O | ST | Digital I/O. |
| | | | | P2B(2) | O | CMOS | Enhanced CCP1 PWM output. |
| | | | | T3CKI(1) | I | ST | Timer3 clock input. |
| | | | | T3G | I | ST | Timer3 external clock gate input. |
| | | | | T1CKI | I | ST | Timer1 clock input. |
| | | | | SOSCO | O | — | Secondary oscillator output. |
| 16 | 35 | 35 | 31 | RC1/P2A/CCP2/SOSCI | | | |
| | | | | RC1 | I/O | ST | Digital I/O. |
| | | | | P2A(1) | O | CMOS | Enhanced CCP2 PWM output. |
| | | | | CCP2(1) | I/O | ST | Capture 2 input/Compare 2 output/PWM 2 output. |
| | | | | SOSCI | I | Analog | Secondary oscillator input. |
| 17 | 36 | 36 | 32 | RC2/CTPLS/P1A/CCP1/T5CKI/AN14 | | | |
| | | | | RC2 | I/O | ST | Digital I/O. |
| | | | | CTPLS | O | — | CTMU pulse generator output. |
| | | | | P1A | O | CMOS | Enhanced CCP1 PWM output. |
| | | | | CCP1 | I/O | ST | Capture 1 input/Compare 1 output/PWM 1 output. |
| | | | | T5CKI | I | ST | Timer5 clock input. |
| | | | | AN14 | I | Analog | Analog input 14. |

**Legend:** TTL = TTL compatible input CMOS = CMOS compatible input or output; ST = Schmitt Trigger input with CMOS levels; I = Input; O = Output; P = Power.

**Note 1:** Default pin assignment for P2B, T3CKI, CCP3/P3A and CCP2/P2A when Configuration bits PB2MX, T3CMX, CCP3MX and CCP2MX are set.

**2:** Alternate pin assignment for P2B, T3CKI, CCP3/P3A and CCP2/P2A when Configuration bits PB2MX, T3CMX, CCP3MX and CCP2MX are clear.

## 4.0 RESET

The PIC18(L)F2X/4XK22 devices differentiate between various kinds of Reset:

a) Power-on Reset (POR)
b) $\overline{\text{MCLR}}$ Reset during normal operation
c) $\overline{\text{MCLR}}$ Reset during power-managed modes
d) Watchdog Timer (WDT) Reset (during execution)
e) Programmable Brown-out Reset (BOR)
f) RESET Instruction
g) Stack Full Reset
h) Stack Underflow Reset

This section discusses Resets generated by $\overline{\text{MCLR}}$, POR and BOR and covers the operation of the various start-up timers. Stack Reset events are covered in **Section 5.2.0.1 "Stack Full and Underflow Resets"**. WDT Resets are covered in **Section 24.3 "Watchdog Timer (WDT)"**.

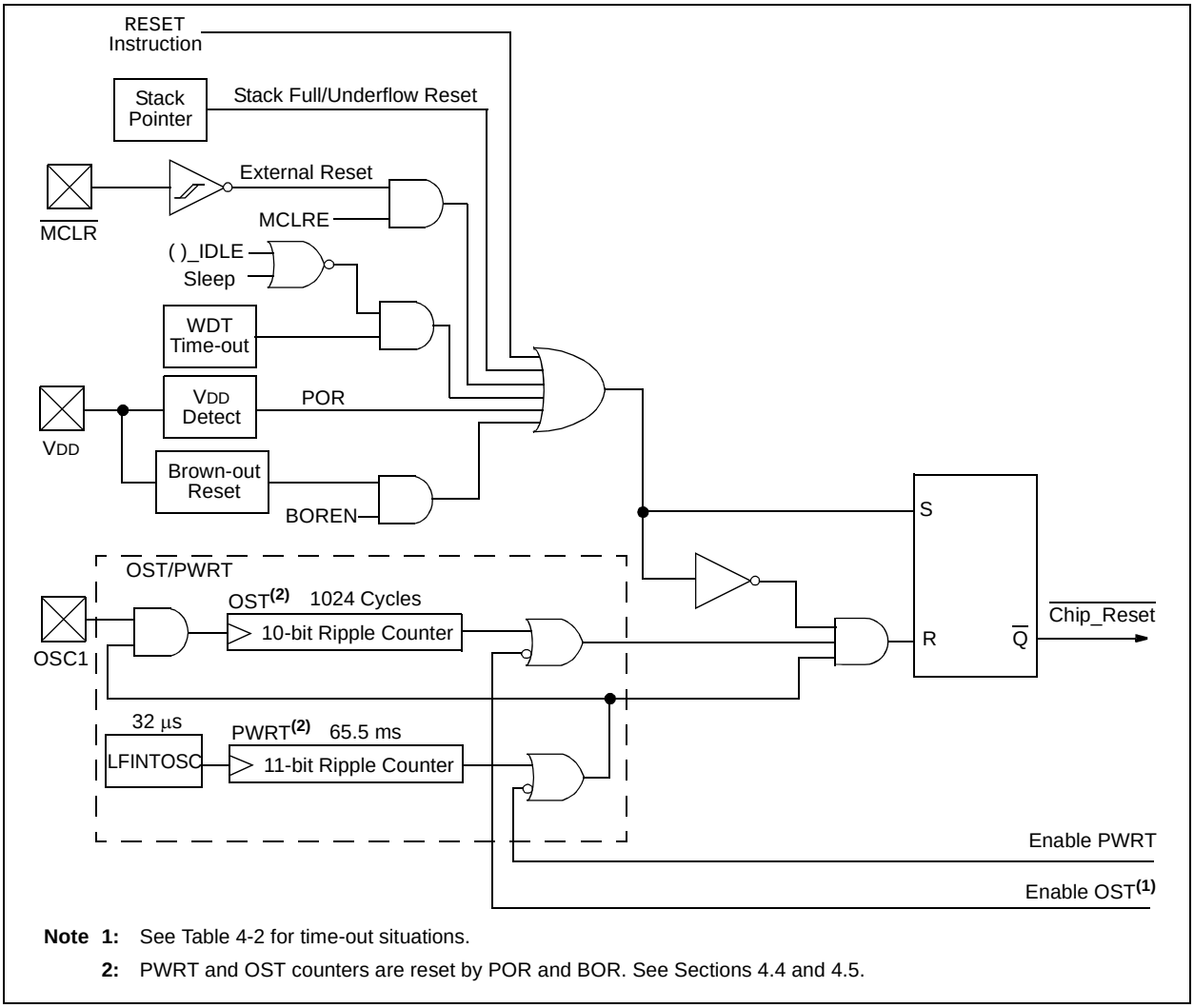
A simplified block diagram of the On-Chip Reset Circuit is shown in Figure 4-1.

### 4.1 RCON Register

Device Reset events are tracked through the RCON register (Register 4-1). The lower five bits of the register indicate that a specific Reset event has occurred. In most cases, these bits can only be cleared by the event and must be set by the application after the event. The state of these flag bits, taken together, can be read to indicate the type of Reset that just occurred. This is described in more detail in **Section 4.7 "Reset State of Registers"**.

The RCON register also has control bits for setting interrupt priority (IPEN) and software control of the BOR (SBOREN). Interrupt priority is discussed in **Section 9.0 "Interrupts"**. BOR is covered in **Section 4.5 "Brown-out Reset (BOR)"**.

**FIGURE 4-1: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT**




**Note 1:** See Table 4-2 for time-out situations.

**2:** PWRT and OST counters are reset by POR and BOR. See Sections 4.4 and 4.5.

**TABLE 5-2: REGISTER FILE SUMMARY FOR PIC18(L)F2X/4XK22 DEVICES (CONTINUED)**

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOR |
|---|---|---|---|---|---|---|---|---|---|---|
| FD1h | WDTCON | — | — | — | — | — | — | — | SWDTEN | ---- ---0 |
| FD0h | RCON | IPEN | SBOREN | — | $\overline{RI}$ | $\overline{TO}$ | $\overline{PD}$ | $\overline{POR}$ | $\overline{BOR}$ | 01-1 1100 |
| FCFh | TMR1H | Holding Register for the Most Significant Byte of the 16-bit TMR1 Register | | | | | | | | xxxx xxxx |
| FCEh | TMR1L | Least Significant Byte of the 16-bit TMR1 Register | | | | | | | | xxxx xxxx |
| FCDh | T1CON | TMR1CS<1:0> | | T1CKPS<1:0> | | T1SOSCEN | $\overline{T1SYNC}$ | T1RD16 | TMR1ON | 0000 0000 |
| FCCh | T1GCON | TMR1GE | T1GPOL | T1GTM | T1GSPM | T1GGO/$\overline{DONE}$ | T1GVAL | T1GSS<1:0> | | 0000 xx00 |
| FCBh | SSP1CON3 | ACKTIM | PCIE | SCIE | BOEN | SDAHT | SBCDE | AHEN | DHEN | 0000 0000 |
| FCAh | SSP1MSK | SSP1 MASK Register bits | | | | | | | | 1111 1111 |
| FC9h | SSP1BUF | SSP1 Receive Buffer/Transmit Register | | | | | | | | xxxx xxxx |
| FC8h | SSP1ADD | SSP1 Address Register in $I^2C$ Slave Mode. SSP1 Baud Rate Reload Register in $I^2C$ Master Mode | | | | | | | | 0000 0000 |
| FC7h | SSP1STAT | SMP | CKE | D/$\overline{A}$ | P | S | R/$\overline{W}$ | UA | BF | 0000 0000 |
| FC6h | SSP1CON1 | WCOL | SSPOV | SSPEN | CKP | SSPM<3:0> | | | | 0000 0000 |
| FC5h | SSP1CON2 | GCEN | ACKSTAT | ACKDT | ACKEN | RCEN | PEN | RSEN | SEN | 0000 0000 |
| FC4h | ADRESH | A/D Result, High Byte | | | | | | | | xxxx xxxx |
| FC3h | ADRESL | A/D Result, Low Byte | | | | | | | | xxxx xxxx |
| FC2h | ADCON0 | — | CHS<4:0> | | | | | GO/$\overline{DONE}$ | ADON | --00 0000 |
| FC1h | ADCON1 | TRIGSEL | — | — | — | PVCFG<1:0> | | NVCFG<1:0> | | 0--- 0000 |
| FC0h | ADCON2 | ADFM | — | ACQT<2:0> | | | ADCS<2:0> | | | 0-00 0000 |
| FBFh | CCPR1H | Capture/Compare/PWM Register 1, High Byte | | | | | | | | xxxx xxxx |
| FBEh | CCPR1L | Capture/Compare/PWM Register 1, Low Byte | | | | | | | | xxxx xxxx |
| FBDh | CCP1CON | P1M<1:0> | | DC1B<1:0> | | CCP1M<3:0> | | | | 0000 0000 |
| FBCh | TMR2 | Timer2 Register | | | | | | | | 0000 0000 |
| FBBh | PR2 | Timer2 Period Register | | | | | | | | 1111 1111 |
| FBAh | T2CON | — | T2OUTPS<3:0> | | | | TMR2ON | T2CKPS<1:0> | | -000 0000 |
| FB9h | PSTR1CON | — | — | — | STR1SYNC | STR1D | STR1C | STR1B | STR1A | ---0 0001 |
| FB8h | BAUDCON1 | ABDOVF | RCIDL | DTRXP | CKTXP | BRG16 | — | WUE | ABDEN | 0100 0-00 |
| FB7h | PWM1CON | P1RSEN | P1DC<6:0> | | | | | | | 0000 0000 |
| FB6h | ECCP1AS | CCP1ASE | CCP1AS<2:0> | | | PSS1AC<1:0> | | PSS1BD<1:0> | | 0000 0000 |
| FB4h | T3GCON | TMR3GE | T3GPOL | T3GTM | T3GSPM | T3GGO/$\overline{DONE}$ | T3GVAL | T3GSS<1:0> | | 0000 0x00 |
| FB3h | TMR3H | Holding Register for the Most Significant Byte of the 16-bit TMR3 Register | | | | | | | | xxxx xxxx |
| FB2h | TMR3L | Least Significant Byte of the 16-bit TMR3 Register | | | | | | | | xxxx xxxx |
| FB1h | T3CON | TMR3CS<1:0> | | T3CKPS<1:0> | | T3SOSCEN | $\overline{T3SYNC}$ | T3RD16 | TMR3ON | 0000 0000 |
| FB0h | SPBRGH1 | EUSART1 Baud Rate Generator, High Byte | | | | | | | | 0000 0000 |
| FAFh | SPBRG1 | EUSART1 Baud Rate Generator, Low Byte | | | | | | | | 0000 0000 |
| FAEh | RCREG1 | EUSART1 Receive Register | | | | | | | | 0000 0000 |
| FADh | TXREG1 | EUSART1 Transmit Register | | | | | | | | 0000 0000 |
| FACh | TXSTA1 | CSRC | TX9 | TXEN | SYNC | SENDB | BRGH | TRMT | TX9D | 0000 0010 |
| FABh | RCSTA1 | SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D | 0000 000x |
| FAAh | EEADRH(5) | — | — | — | — | — | — | EEADR<9:8> | | ---- --00 |
| FA9h | EEADR | EEADR<7:0> | | | | | | | | 0000 0000 |
| FA8h | EEDATA | EEPROM Data Register | | | | | | | | 0000 0000 |
| FA7h | EECON2 | EEPROM Control Register 2 (not a physical register) | | | | | | | | ---- --00 |
| FA6h | EECON1 | EEPGD | CFGS | — | FREE | WRERR | WREN | WR | RD | xx-0 x000 |
| FA5h | IPR3 | SSP2IP | BCL2IP | RC2IP | TX2IP | CTMUIP | TMR5GIP | TMR3GIP | TMR1GIP | 0000 0000 |
| FA4h | PIR3 | SSP2IF | BCL2IF | RC2IF | TX2IF | CTMUIF | TMR5GIF | TMR3GIF | TMR1GIF | 0000 0000 |
| FA3h | PIE3 | SSP2IE | BCL2IE | RC2IE | TX2IE | CTMUIE | TMR5GIE | TMR3GIE | TMR1GIE | 0000 0000 |

**Legend:** x = unknown, u = unchanged, — = unimplemented, q = value depends on condition

**Note 1:** PIC18(L)F4XK22 devices only.
**2:** PIC18(L)F2XK22 devices only.
**3:** PIC18(L)F23/24K22 and PIC18(L)F43/44K22 devices only.
**4:** PIC18(L)F26K22 and PIC18(L)F46K22 devices only.

## 6.5 Erasing Flash Program Memory

The minimum erase block is 32 words or 64 bytes. Only through the use of an external programmer, or through ICSP™ control, can larger blocks of program memory be bulk erased. Word erase in the Flash array is not supported.

When initiating an erase sequence from the microcontroller itself, a block of 64 bytes of program memory is erased. The Most Significant 16 bits of the TBLPTR<21:6> point to the block being erased. The TBLPTR<5:0> bits are ignored.

The EECON1 register commands the erase operation. The EEPGD bit must be set to point to the Flash program memory. The WREN bit must be set to enable write operations. The FREE bit is set to select an erase operation.

The write initiate sequence for EECON2, shown as steps 4 through 6 in **Section 6.5.1 "Flash Program Memory Erase Sequence"**, is used to guard against accidental writes. This is sometimes referred to as a long write.

A long write is necessary for erasing the internal Flash. Instruction execution is halted during the long write cycle. The long write is terminated by the internal programming timer.

### 6.5.1 FLASH PROGRAM MEMORY ERASE SEQUENCE

The sequence of events for erasing a block of internal program memory is:

1. Load Table Pointer register with address of block being erased.
2. Set the EECON1 register for the erase operation:
   - set EEPGD bit to point to program memory;
   - clear the CFGS bit to access program memory;
   - set WREN bit to enable writes;
   - set FREE bit to enable the erase.
3. Disable interrupts.
4. Write 55h to EECON2.
5. Write 0AAh to EECON2.
6. Set the WR bit. This will begin the block erase cycle.
7. The CPU will stall for duration of the erase (about 2 ms using internal timer).
8. Re-enable interrupts.

**EXAMPLE 6-2: ERASING A FLASH PROGRAM MEMORY BLOCK**

```
                    MOVLW   CODE_ADDR_UPPER         ; load TBLPTR with the base
                    MOVWF   TBLPTRU                 ; address of the memory block
                    MOVLW   CODE_ADDR_HIGH
                    MOVWF   TBLPTRH
                    MOVLW   CODE_ADDR_LOW
                    MOVWF   TBLPTRL
        ERASE_BLOCK
                    BSF     EECON1, EEPGD           ; point to Flash program memory
                    BCF     EECON1, CFGS            ; access Flash program memory
                    BSF     EECON1, WREN            ; enable write to memory
                    BSF     EECON1, FREE            ; enable block Erase operation
                    BCF     INTCON, GIE             ; disable interrupts
    Required        MOVLW   55h
    Sequence        MOVWF   EECON2                  ; write 55h
                    MOVLW   0AAh
                    MOVWF   EECON2                  ; write 0AAh
                    BSF     EECON1, WR              ; start erase (CPU stall)
                    BSF     INTCON, GIE             ; re-enable interrupts
```

**REGISTER 9-9: PIE1: PERIPHERAL INTERRUPT ENABLE (FLAG) REGISTER 1**

| U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | ADIE | RC1IE | TX1IE | SSP1IE | CCP1IE | TMR2IE | TMR1IE |
| 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 **Unimplemented:** Read as ‘0’.

bit 6 **ADIE:** A/D Converter Interrupt Enable bit
1 = Enables the A/D interrupt
0 = Disables the A/D interrupt

bit 5 **RC1IE:** EUSART1 Receive Interrupt Enable bit
1 = Enables the EUSART1 receive interrupt
0 = Disables the EUSART1 receive interrupt

bit 4 **TX1IE:** EUSART1 Transmit Interrupt Enable bit
1 = Enables the EUSART1 transmit interrupt
0 = Disables the EUSART1 transmit interrupt

bit 3 **SSP1IE:** Master Synchronous Serial Port 1 Interrupt Enable bit
1 = Enables the MSSP1 interrupt
0 = Disables the MSSP1 interrupt

bit 2 **CCP1IE:** CCP1 Interrupt Enable bit
1 = Enables the CCP1 interrupt
0 = Disables the CCP1 interrupt

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

bit 0 **TMR1IE:** TMR1 Overflow Interrupt Enable bit
1 = Enables the TMR1 overflow interrupt
0 = Disables the TMR1 overflow interrupt

**TABLE 10-4: PORT PIN FUNCTION PRIORITY (CONTINUED)**

| Port bit | Port Function Priority by Port Pin | | | | |
|---|---|---|---|---|---|
| | **PORTA** | **PORTB** | **PORTC** | **PORTD(2)** | **PORTE(2)** |
| 5 | SRNQ | CCP3(3) | SDO1 | P1B | |
| | C2OUT | P3A(3) | RC5 | RD5 | |
| | RA5 | P2B(1)(4) | | | |
| | | RB5 | | | |
| 6 | OSC2 | PGC | TX1/CK1 | TX2/CK2 | |
| | CLKO | TX2/CK2(1) | CCP3(1)(7) | P1C | |
| | RA6 | RB6 | P3A(1)(7) | RD6 | |
| | | ICDCK | RC6 | | |
| 7 | RA7 | | | | |
| | OSC1 | PGD | RX1/DT1 | RX2/DT2 | |
| | RA7 | RX2/DT2(1) | P3B(1) | P1D | |
| | | RB7 | RC7 | RD7 | |
| | | ICDDT | | | |

**Note 1:** PIC18(L)F2XK22 devices.
**2:** PIC18(L)F4XK22 devices.
**3:** Function default pin.
**4:** Function default pin (28-pin devices).
**5:** Function default pin (40/44-pin devices).
**6:** Function alternate pin.
**7:** Function alternate pin (28-pin devices).
**8:** Function alternate pin (40/44-pin devices)

**TABLE 10-5: PORTB I/O SUMMARY (CONTINUED)**

| Pin | Function | TRIS Setting | ANSEL Setting | Pin Type | Buffer Type | Description |
|---|---|---|---|---|---|---|
| RB2/INT2/CTED1/ P1B/SDI2/SDA2/ AN8 | RB2 | 0 | 0 | O | DIG | LATB<2> data output; not affected by analog input. |
| | | 1 | 0 | I | TTL | PORTB<2> data input; disabled when analog input enabled. |
| | INT2 | 1 | 0 | I | ST | External interrupt 2. |
| | CTED1 | 1 | 0 | I | ST | CTMU Edge 1 input. |
| | P1B(3) | 0 | 0 | O | DIG | Enhanced CCP1 PWM output 2. |
| | SDI2(3) | 1 | 0 | I | ST | MSSP2 SPI data input. |
| | SDA2(3) | 0 | 0 | O | DIG | MSSP2 $I^2C$ data output. |
| | | 1 | 0 | I | $I^2C$ | MSSP2 $I^2C$ data input. |
| | AN8 | 1 | 1 | I | AN | Analog input 8. |
| RB3/CTED2/P2A/ CCP2/SDO2/ C12IN2-/AN9 | RB3 | 0 | 0 | O | DIG | LATB<3> data output; not affected by analog input. |
| | | 1 | 0 | I | TTL | PORTB<3> data input; disabled when analog input enabled. |
| | CTED2 | 1 | 0 | I | ST | CTMU Edge 2 input. |
| | P2A | 0 | 0 | O | DIG | Enhanced CCP1 PWM output 1. |
| | CCP2(2) | 0 | 0 | O | DIG | Compare 2 output/PWM 2 output. |
| | | 1 | 0 | I | ST | Capture 2 input. |
| | SDO2(2) | 0 | 0 | O | DIG | MSSP2 SPI data output. |
| | C12IN2- | 1 | 1 | I | AN | Comparators C1 and C2 inverting input. |
| | AN9 | 1 | 1 | I | AN | Analog input 9. |
| RB4/IOC0/P1D/ T5G/AN11 | RB4 | 0 | 0 | O | DIG | LATB<4> data output; not affected by analog input. |
| | | 1 | 0 | I | TTL | PORTB<4> data input; disabled when analog input enabled. |
| | IOC0 | 1 | 0 | I | TTL | Interrupt-on-change pin. |
| | P1D | 0 | 0 | O | DIG | Enhanced CCP1 PWM output 4. |
| | T5G | 1 | 0 | I | ST | Timer5 external clock gate input. |
| | AN11 | 1 | 1 | I | AN | Analog input 11. |
| RB5/IOC1/P2B/ P3A/CCP3/T3CKI/ T1G/AN13 | RB5 | 0 | 0 | O | DIG | LATB<5> data output; not affected by analog input. |
| | | 1 | 0 | I | TTL | PORTB<5> data input; disabled when analog input enabled. |
| | IOC1 | 1 | 0 | I | TTL | Interrupt-on-change pin 1. |
| | P2B(1)(3) | 0 | 0 | O | DIG | Enhanced CCP2 PWM output 2. |
| | P3A(1) | 0 | 0 | O | DIG | Enhanced CCP3 PWM output 1. |
| | CCP3(1) | 0 | 0 | O | DIG | Compare 3 output/PWM 3 output. |
| | | 1 | 0 | I | ST | Capture 3 input. |
| | T3CKI(2) | 1 | 0 | I | ST | Timer3 clock input. |
| | T1G | 1 | 0 | I | ST | Timer1 external clock gate input. |
| | AN13 | 1 | 1 | I | AN | Analog input 13. |

**Legend:** AN = Analog input or output; TTL = TTL compatible input; HV = High Voltage; OD = Open Drain; XTAL = Crystal; CMOS = CMOS compatible input or output; ST = Schmitt Trigger input with CMOS levels; $I^2C$ = Schmitt Trigger input with $I^2C$.

**Note 1:** Default pin assignment for P2B, T3CKI, CCP3 and CCP2 when Configuration bits PB2MX, T3CMX, CCP3MX and CCP2MX are set.

**2:** Alternate pin assignment for P2B, T3CKI, CCP3 and CCP2 when Configuration bits PB2MX, T3CMX, CCP3MX and CCP2MX are clear.

**3:** Function on PORTD and PORTE for PIC18(L)F4XK22 devices.

## 13.0 TIMER2/4/6 MODULE

There are three identical 8-bit Timer2-type modules available. To maintain pre-existing naming conventions, the Timers are called Timer2, Timer4 and Timer6 (also Timer2/4/6).

| **Note:** | The 'x' variable used in this section is used to designate Timer2, Timer4, or Timer6. For example, TxCON references T2CON, T4CON, or T6CON. PRx references PR2, PR4, or PR6. |
|---|---|

The Timer2/4/6 module incorporates the following features:

- 8-bit Timer and Period registers (TMRx and PRx, respectively)
- Readable and writable (both registers)
- Software programmable prescaler (1:1, 1:4, 1:16)
- Software programmable postscaler (1:1 to 1:16)
- Interrupt on TMRx match with PRx, respectively
- Optional use as the shift clock for the MSSPx modules (Timer2 only)

See Figure 13-1 for a block diagram of Timer2/4/6.

**FIGURE 13-1: TIMER2/4/6 BLOCK DIAGRAM**

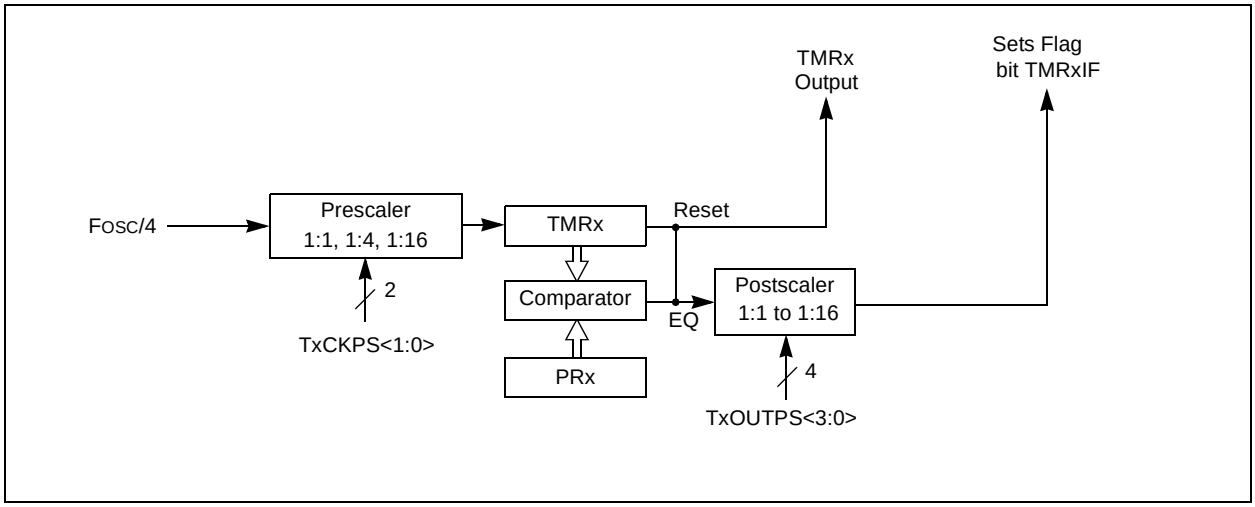


**TABLE 13-1: SUMMARY OF REGISTERS ASSOCIATED WITH TIMER2/4/6**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| CCPTMRS0 | C3TSEL<1:0> | | — | C2TSEL<1:0> | | — | C1TSEL<1:0> | | 201 |
| CCPTMRS1 | — | — | — | — | C5TSEL<1:0> | | C4TSEL<1:0> | | 201 |
| INTCON | GIE/GIEH | PEIE/GIEL | TMR0IE | INT0IE | RBIE | TMR0IF | INT0IF | RBIF | 109 |
| IPR1 | — | ADIP | RC1IP | TX1IP | SSP1IP | CCP1IP | TMR2IP | TMR1IP | 121 |
| IPR5 | — | — | — | — | — | TMR6IP | TMR5IP | TMR4IP | 124 |
| PIE1 | — | ADIE | RC1IE | TX1IE | SSP1IE | CCP1IE | TMR2IE | TMR1IE | 117 |
| PIE5 | — | — | — | — | — | TMR6IE | TMR5IE | TMR4IE | 120 |
| PIR1 | — | ADIF | RC1IF | TX1IF | SSP1IF | CCP1IF | TMR2IF | TMR1IF | 112 |
| PIR5 | — | — | — | — | — | TMR6IF | TMR5IF | TMR4IF | 116 |
| PMD0 | UART2MD | UART1MD | TMR6MD | TMR5MD | TMR4MD | TMR3MD | TMR2MD | TMR1MD | 52 |
| PR2 | Timer2 Period Register | | | | | | | | — |
| PR4 | Timer4 Period Register | | | | | | | | — |
| PR6 | Timer6 Period Register | | | | | | | | — |
| T2CON | — | T2OUTPS<3:0> | | | | TMR2ON | T2CKPS<1:0> | | 166 |
| T4CON | — | T4OUTPS<3:0> | | | | TMR4ON | T4CKPS<1:0> | | 166 |
| T6CON | — | T6OUTPS<3:0> | | | | TMR6ON | T6CKPS<1:0> | | 166 |
| TMR2 | Timer2 Register | | | | | | | | — |
| TMR4 | Timer4 Register | | | | | | | | — |
| TMR6 | Timer6 Register | | | | | | | | — |

**Legend:** — = unimplemented locations, read as '0'. Shaded bits are not used by Timer2/4/6.

### 14.4.8 SETUP FOR ECCP PWM OPERATION USING ECCP1 AND TIMER2

The following steps should be taken when configuring the ECCP1 module for PWM operation using Timer2:

1. Configure the PWM pins to be used (P1A, P1B, P1C, and P1D):
   - Configure PWM outputs to be used as inputs by setting the corresponding TRIS bits. This prevents spurious outputs during setup.
   - Set the PSTR1CON bits for each PWM output to be used.
2. Select Timer2 as the period timer by configuring CCPTMR0 register bits C1TSEL<1:0> = '00'.
3. Set the PWM period by loading the PR2 register.
4. Configure auto-shutdown as OFF or select the source with the CCP1AS<2:0> bits of the ECCP1AS register.
5. Configure the auto-shutdown sources as needed:
   - Configure each comparator used.
   - Configure the comparator inputs as analog.
   - Configure the FLT0 input pin and clear ANSB0.
6. Force a shutdown condition (OFF included):
   - Configure safe starting output levels by setting the default shutdown drive states with the PSS1AC<1:0> and PSS1BD<1:0> bits of the ECCP1AS register.
   - Clear the P1RSEN bit of the PWM1CON register.
   - Set the CCP1AS bit of the ECCP1AS register.
7. Configure the ECCP1 module for the desired PWM mode and configuration by loading the CCP1CON register with the appropriate values:
   - Select one of the available output configurations and direction with the P1M<1:0> bits.
   - Select the polarities of the PWM output signals with the CCP1M<3:0> bits.
8. Set the 10-bit PWM duty cycle:
   - Load the eight MSbs into the CCPR1L register.
   - Load the two LSbs into the DC<1:0> bits of the CCP1CON register.
9. For Half-Bridge Output mode, set the dead-band delay by loading P1DC<6:0> bits of the PWM1CON register with the appropriate value.
10. Configure and start TMR2:
    - Set the TMR2 prescale value by loading the T2CKPS bits of the T2CON register.
    - Start Timer2 by setting the TMR2ON bit.
11. Enable the ECCP1/P1A, P1B, P1C and/or P1D pin outputs by clearing the respective TRIS bits.
12. Start the PWM:
    - If shutdown auto-restart is used, then set the P1RSEN bit of the PWM1CON register.
    - If shutdown auto-restart is not used, then clear the CCP1ASE bit of the ECCP1AS register.

### 15.4.5 START CONDITION

The $I^2C$ specification defines a Start condition as a transition of SDAx from a high-to -low state while SCLx line is high. A Start condition is always generated by the master and signifies the transition of the bus from an Idle to an active state. Figure 15-12 shows wave forms for Start and Stop conditions.

A bus collision can occur on a Start condition if the module samples the SDAx line low before asserting it low. This does not conform to the $I^2C$ specification that states no bus collision can occur on a Start.

### 15.4.6 STOP CONDITION

A Stop condition is a transition of the SDAx line from a low-to-high state while the SCLx line is high.

| **Note:** | At least one SCLx low time must appear before a Stop is valid, therefore, if the SDAx line goes low then high again while the SCLx line stays high, only the Start condition is detected. |
|---|---|

### 15.4.7 RESTART CONDITION

A Restart is valid any time that a Stop would be valid. A master can issue a Restart if it wishes to hold the bus after terminating the current transfer. A Restart has the same effect on the slave that a Start would, resetting all slave logic and preparing it to clock in an address. The master may want to address the same or another slave. Figure 15-13 shows the wave form for a Restart condition.

In 10-bit Addressing Slave mode a Restart is required for the master to clock data out of the addressed slave. Once a slave has been fully addressed, matching both high and low address bytes, the master can issue a Restart and the high address byte with the R/$\overline{W}$ bit set. The slave logic will then hold the clock and prepare to clock out data.

After a full match with R/$\overline{W}$ clear in 10-bit mode, a prior match flag is set and maintained. Until a Stop condition, a high address with R/$\overline{W}$ clear, or high address match fails.

### 15.4.8 START/STOP CONDITION INTERRUPT MASKING

The SCIE and PCIE bits of the SSPxCON3 register can enable the generation of an interrupt in Slave modes that do not typically support this function. Slave modes where interrupt on Start and Stop detect are already enabled, these bits will have no effect.

**FIGURE 15-12: $I^2C$ START AND STOP CONDITIONS**

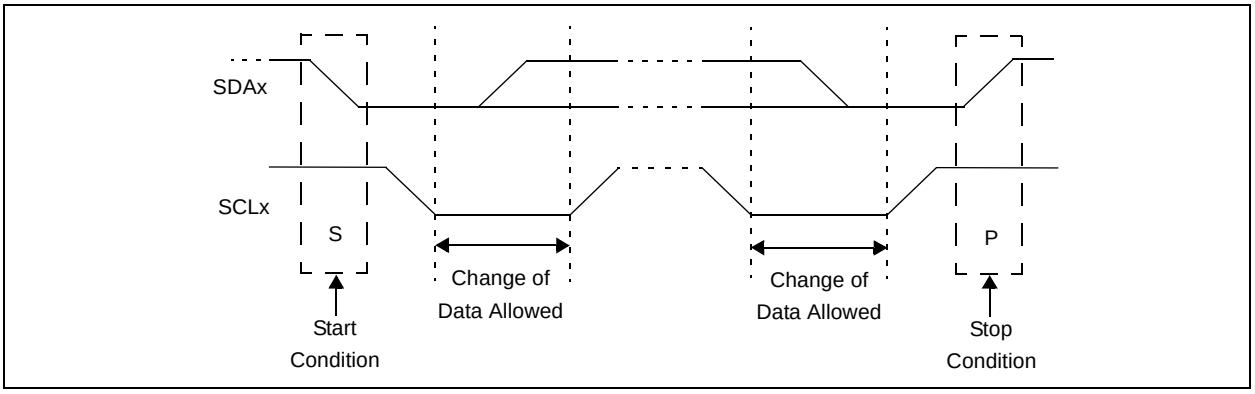



**FIGURE 15-13: $I^2C$ RESTART CONDITION**

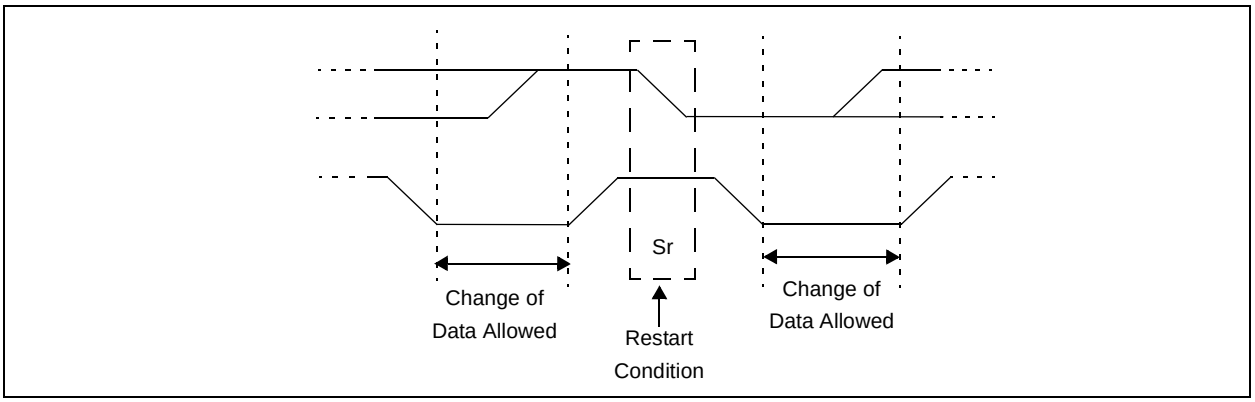
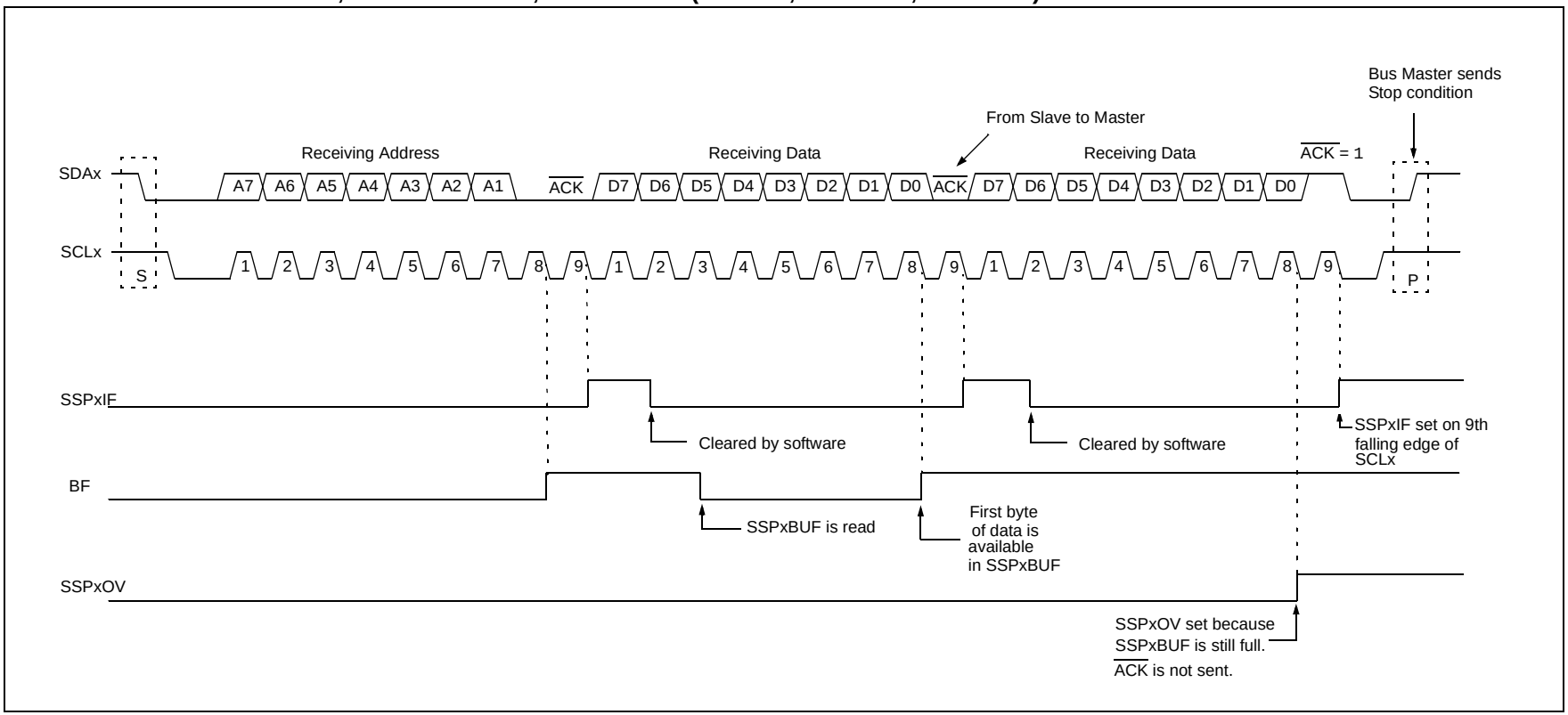


**FIGURE 15-14: I²C SLAVE, 7-BIT ADDRESS, RECEPTION (SEN = 0, AHEN = 0, DHEN = 0)**



## 16.5 EUSART Synchronous Mode

Synchronous serial communications are typically used in systems with a single master and one or more slaves. The master device contains the necessary circuitry for baud rate generation and supplies the clock for all devices in the system. Slave devices can take advantage of the master clock by eliminating the internal clock generation circuitry.

There are two signal lines in Synchronous mode: a bidirectional data line and a clock line. Slaves use the external clock supplied by the master to shift the serial data into and out of their respective receive and transmit shift registers. Since the data line is bidirectional, synchronous operation is half-duplex only. Half-duplex refers to the fact that master and slave devices can receive and transmit data but not both simultaneously. The EUSART can operate as either a master or slave device.

Start and Stop bits are not used in synchronous transmissions.

### 16.5.1 SYNCHRONOUS MASTER MODE

The following bits are used to configure the EUSART for Synchronous Master operation:

- SYNC = 1
- CSRC = 1
- SREN = 0 (for transmit); SREN = 1 (for receive)
- CREN = 0 (for transmit); CREN = 1 (for receive)
- SPEN = 1

Setting the SYNC bit of the TXSTAx register configures the device for synchronous operation. Setting the CSRC bit of the TXSTAx register configures the device as a master. Clearing the SREN and CREN bits of the RCSTAx register ensures that the device is in the Transmit mode, otherwise the device will be configured to receive. Setting the SPEN bit of the RCSTAx register enables the EUSART. If the RXx/DTx or TXx/CKx pins are shared with an analog peripheral the analog I/O functions must be disabled by clearing the corresponding ANSEL bits.

The TRIS bits corresponding to the RXx/DTx and TXx/CKx pins should be set.

#### 16.5.1.1 Master Clock

Synchronous data transfers use a separate clock line, which is synchronous with the data. A device configured as a master transmits the clock on the TXx/CKx line. The TXx/CKx pin output driver is automatically enabled when the EUSART is configured for synchronous 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 clock cycle is generated for each data bit. Only as many clock cycles are generated as there are data bits.

#### 16.5.1.2 Clock Polarity

A clock polarity option is provided for Microwire compatibility. Clock polarity is selected with the CKTXP bit of the BAUDCONx register. Setting the CKTXP bit sets the clock Idle state as high. When the CKTXP bit is set, the data changes on the falling edge of each clock and is sampled on the rising edge of each clock. Clearing the CKTXP bit sets the Idle state as low. When the CKTXP bit is cleared, the data changes on the rising edge of each clock and is sampled on the falling edge of each clock.

#### 16.5.1.3 Synchronous Master Transmission

Data is transferred out of the device on the RXx/DTx pin. The RXx/DTx and TXx/CKx pin output drivers are automatically enabled when the EUSART is configured for synchronous master transmit operation.

A transmission is initiated by writing a character to the TXREGx register. If the TSR still contains all or part of a previous character the new character data is held in the TXREGx until the last bit of the previous character has been transmitted. If this is the first character, or the previous character has been completely flushed from the TSR, the data in the TXREGx is immediately transferred to the TSR. The transmission of the character commences immediately following the transfer of the data to the TSR from the TXREGx.

Each data bit changes on the leading edge of the master clock and remains valid until the subsequent leading clock edge.

| **Note:** | The TSR register is not mapped in data memory, so it is not available to the user. |
|---|---|

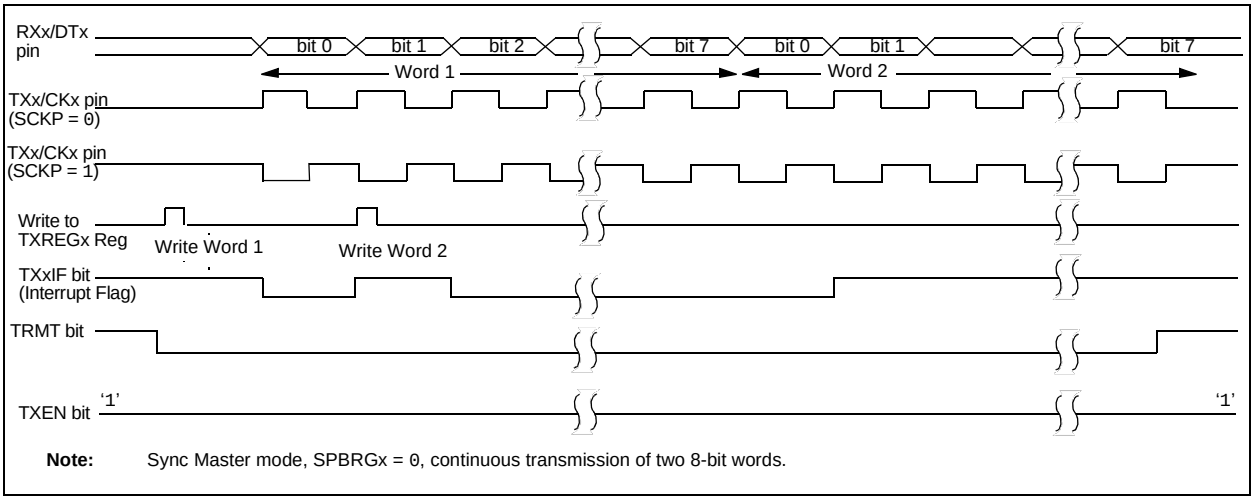
#### 16.5.1.4 Data Polarity

The polarity of the transmit and receive data can be controlled with the DTRXP bit of the BAUDCONx register. The default state of this bit is '0' which selects high true transmit and receive data. Setting the DTRXP bit to '1' will invert the data resulting in low true transmit and receive data.

16.5.1.5 Synchronous Master Transmission Setup:

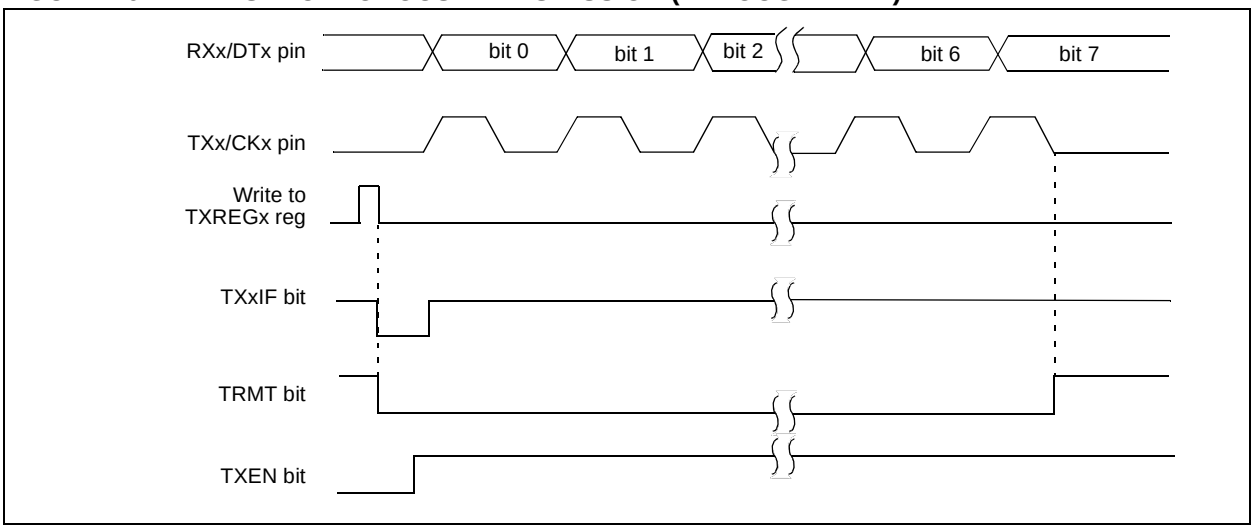
1. Initialize the SPBRGHx, SPBRGx register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see **Section 16.4 "EUSART Baud Rate Generator (BRG)"**).
2. Set the RXx/DTx and TXx/CKx TRIS controls to '1'.
3. Enable the synchronous master serial port by setting bits SYNC, SPEN and CSRC. Set the TRIS bits corresponding to the RXx/DTx and TXx/CKx I/O pins.
4. Disable Receive mode by clearing bits SREN and CREN.
5. Enable Transmit mode by setting the TXEN bit.
6. If 9-bit transmission is desired, set the TX9 bit.
7. If interrupts are desired, set the TXxIE, GIE/GIEH and PEIE/GIEL interrupt enable bits.
8. If 9-bit transmission is selected, the ninth bit should be loaded in the TX9D bit.
9. Start transmission by loading data to the TXREGx register.

**FIGURE 16-10: SYNCHRONOUS TRANSMISSION**




**Note:** Sync Master mode, SPBRGx = 0, continuous transmission of two 8-bit words.

**FIGURE 16-11: SYNCHRONOUS TRANSMISSION (THROUGH TXEN)**



**REGISTER 18-2: CM2CON1: COMPARATOR 1 AND 2 CONTROL REGISTER**

| R-0 | R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| MC1OUT | MC2OUT | C1RSEL | C2RSEL | C1HYS | C2HYS | C1SYNC | C2SYNC |
| 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 **MC1OUT:** Mirror Copy of C1OUT bit

bit 6 **MC2OUT:** Mirror Copy of C2OUT bit

bit 5 **C1RSEL:** Comparator C1 Reference Select bit
1 = FVR BUF1 routed to C1VREF input
0 = DAC routed to C1VREF input

bit 4 **C2RSEL:** Comparator C2 Reference Select bit
1 = FVR BUF1 routed to C2VREF input
0 = DAC routed to C2VREF input

bit 3 **C1HYS:** Comparator C1 Hysteresis Enable bit
1 = Comparator C1 hysteresis enabled
0 = Comparator C1 hysteresis disabled

bit 2 **C2HYS:** Comparator C2 Hysteresis Enable bit
1 = Comparator C2 hysteresis enabled
0 = Comparator C2 hysteresis disabled

bit 1 **C1SYNC:** C1 Output Synchronous Mode bit
1 = C1 output is synchronized to rising edge of TMR1 clock (T1CLK)
0 = C1 output is asynchronous

bit 0 **C2SYNC:** C2 Output Synchronous Mode bit
1 = C2 output is synchronized to rising edge of TMR1 clock (T1CLK)
0 = C2 output is asynchronous

**MOVFF** **Move f to f**

Syntax: MOVFF $f_s$,$f_d$

Operands: $0 \le f_s \le 4095$
$0 \le f_d \le 4095$

Operation: $(f_s) \rightarrow f_d$

Status Affected: None

Encoding:

| | | | |
|---|---|---|---|
| 1st word (source) | 1100 | ffff | ffff$_s$ |
| 2nd word (destin.) | 1111 | ffff | ffff$_d$ |

Description: The contents of source register '$f_s$' are moved to destination register '$f_d$'. Location of source '$f_s$' can be anywhere in the 4096-byte data space (000h to FFFh) and location of destination '$f_d$' can also be anywhere from 000h to FFFh.
Either source or destination can be W (a useful special situation).
MOVFF is particularly useful for transferring a data memory location to a peripheral register (such as the transmit buffer or an I/O port).
The MOVFF instruction cannot use the PCL, TOSU, TOSH or TOSL as the destination register.

Words: 2

Cycles: 2 (3)

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|
| Decode | Read register 'f' (src) | Process Data | No operation |
| Decode | No operation<br>No dummy read | No operation | Write register 'f' (dest) |

Example: MOVFF REG1, REG2

Before Instruction
REG1 = 33h
REG2 = 11h
After Instruction
REG1 = 33h
REG2 = 33h

**MOVLB** **Move literal to low nibble in BSR**

Syntax: MOVLW k

Operands: $0 \le k \le 255$

Operation: $k \rightarrow$ BSR

Status Affected: None

Encoding:

| | | | |
|---|---|---|---|
| 0000 | 0001 | kkkk | kkkk |

Description: The 8-bit literal 'k' is loaded into the Bank Select Register (BSR). The value of BSR<7:4> always remains '0', regardless of the value of $k_7$:$k_4$.

Words: 1

Cycles: 1

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|
| Decode | Read literal 'k' | Process Data | Write literal 'k' to BSR |

Example: MOVLB 5

Before Instruction
BSR Register = 02h
After Instruction
BSR Register = 05h

## 26.0 DEVELOPMENT SUPPORT

The PIC® microcontrollers (MCU) and dsPIC® digital signal controllers (DSC) are supported with a full range of software and hardware development tools:

- Integrated Development Environment
  - MPLAB® X IDE Software
- Compilers/Assemblers/Linkers
  - MPLAB XC Compiler
  - MPASM™ Assembler
  - MPLINK™ Object Linker/ MPLIB™ Object Librarian
  - MPLAB Assembler/Linker/Librarian for Various Device Families
- Simulators
  - MPLAB X SIM Software Simulator
- Emulators
  - MPLAB REAL ICE™ In-Circuit Emulator
- In-Circuit Debuggers/Programmers
  - MPLAB ICD 3
  - PICkit™ 3
- Device Programmers
  - MPLAB PM3 Device Programmer
- Low-Cost Demonstration/Development Boards, Evaluation Kits and Starter Kits
- Third-party development tools

### 26.1 MPLAB X Integrated Development Environment Software

The MPLAB X IDE is a single, unified graphical user interface for Microchip and third-party software, and hardware development tool that runs on Windows®, Linux and Mac OS® X. Based on the NetBeans IDE, MPLAB X IDE is an entirely new IDE with a host of free software components and plug-ins for high-performance application development and debugging. Moving between tools and upgrading from software simulators to hardware debugging and programming tools is simple with the seamless user interface.

With complete project management, visual call graphs, a configurable watch window and a feature-rich editor that includes code completion and context menus, MPLAB X IDE is flexible and friendly enough for new users. With the ability to support multiple tools on multiple projects with simultaneous debugging, MPLAB X IDE is also suitable for the needs of experienced users.

Feature-Rich Editor:

- Color syntax highlighting
- Smart code completion makes suggestions and provides hints as you type
- Automatic code formatting based on user-defined rules
- Live parsing

User-Friendly, Customizable Interface:

- Fully customizable interface: toolbars, toolbar buttons, windows, window placement, etc.
- Call graph window

Project-Based Workspaces:

- Multiple projects
- Multiple tools
- Multiple configurations
- Simultaneous debugging sessions

File History and Bug Tracking:

- Local file history feature
- Built-in support for Bugzilla issue tracker

**FIGURE 28-58: PIC18F2X/4XK22 TYPICAL IDD: PRI_RUN EC with PLL**

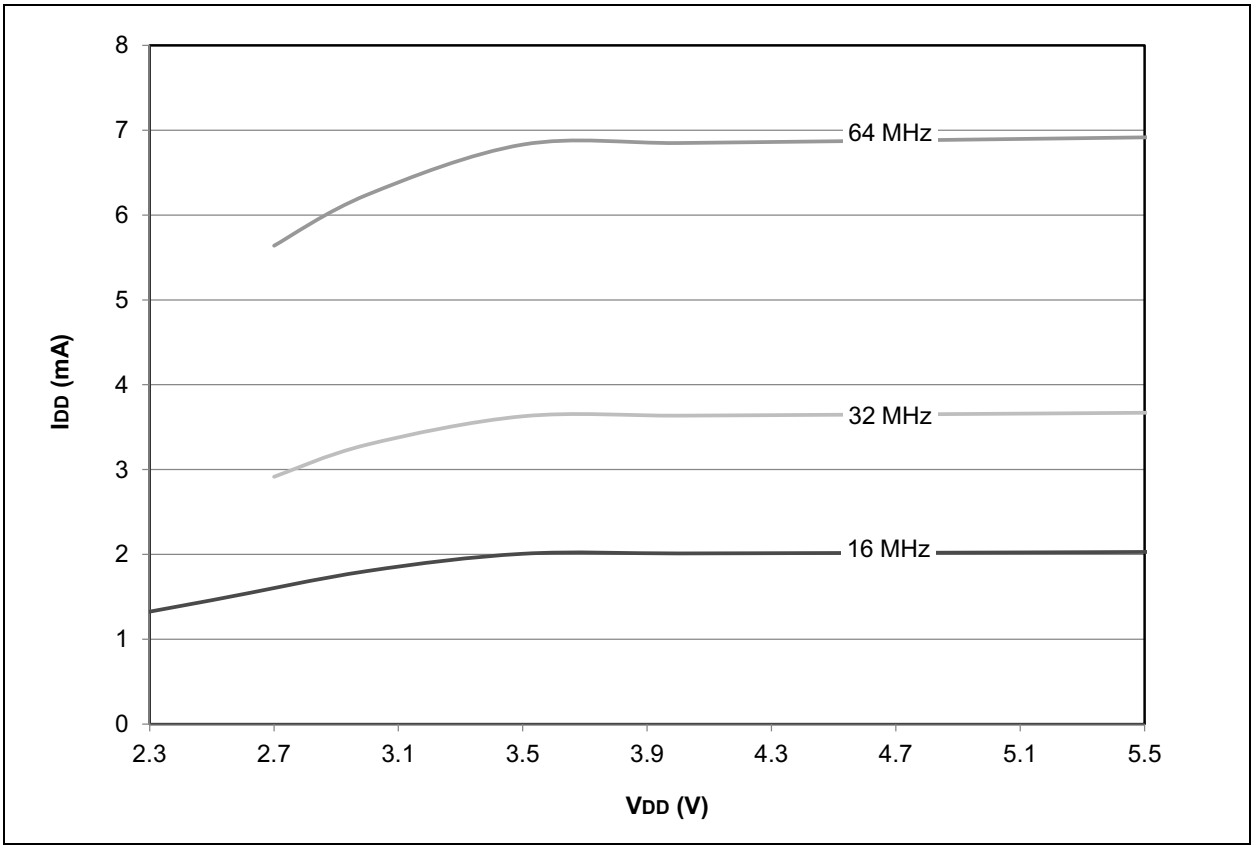



**FIGURE 28-59: PIC18F2X/4XK22 MAXIMUM IDD: PRI_RUN EC with PLL**

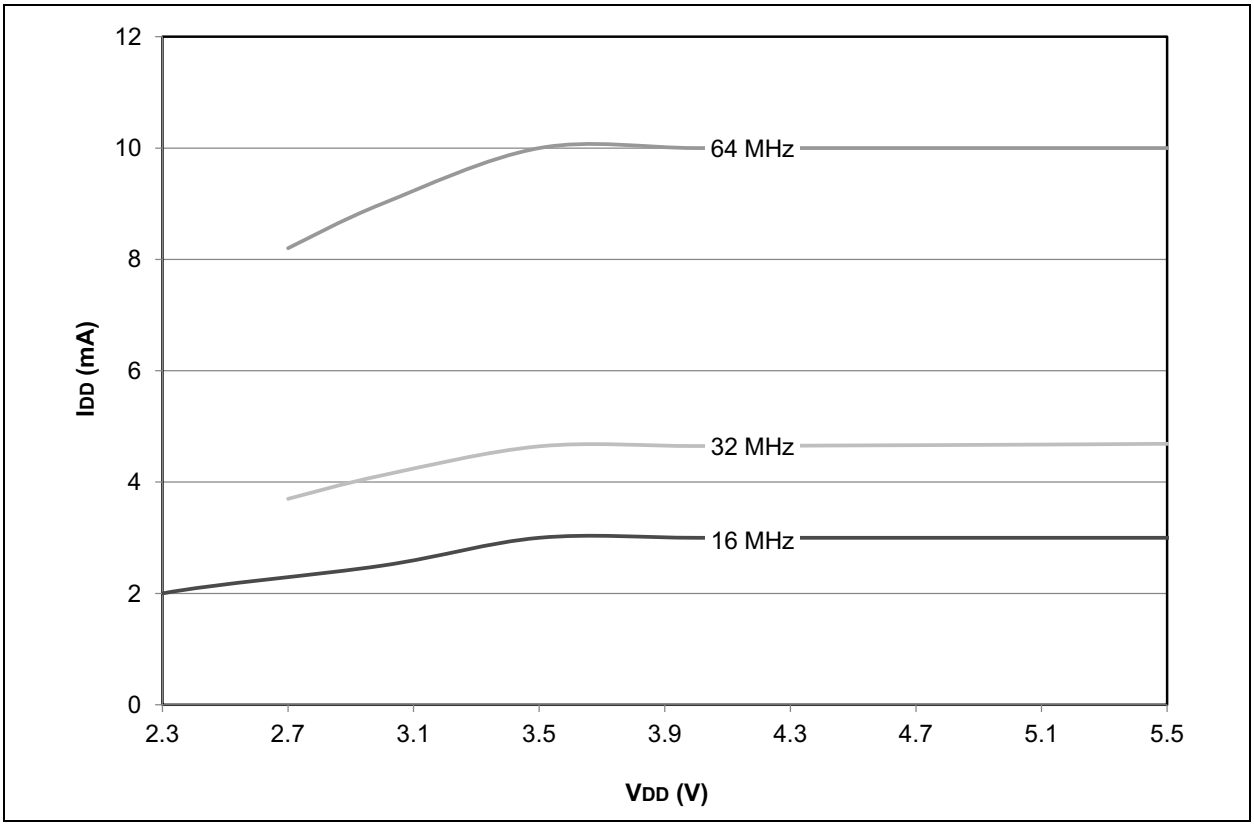
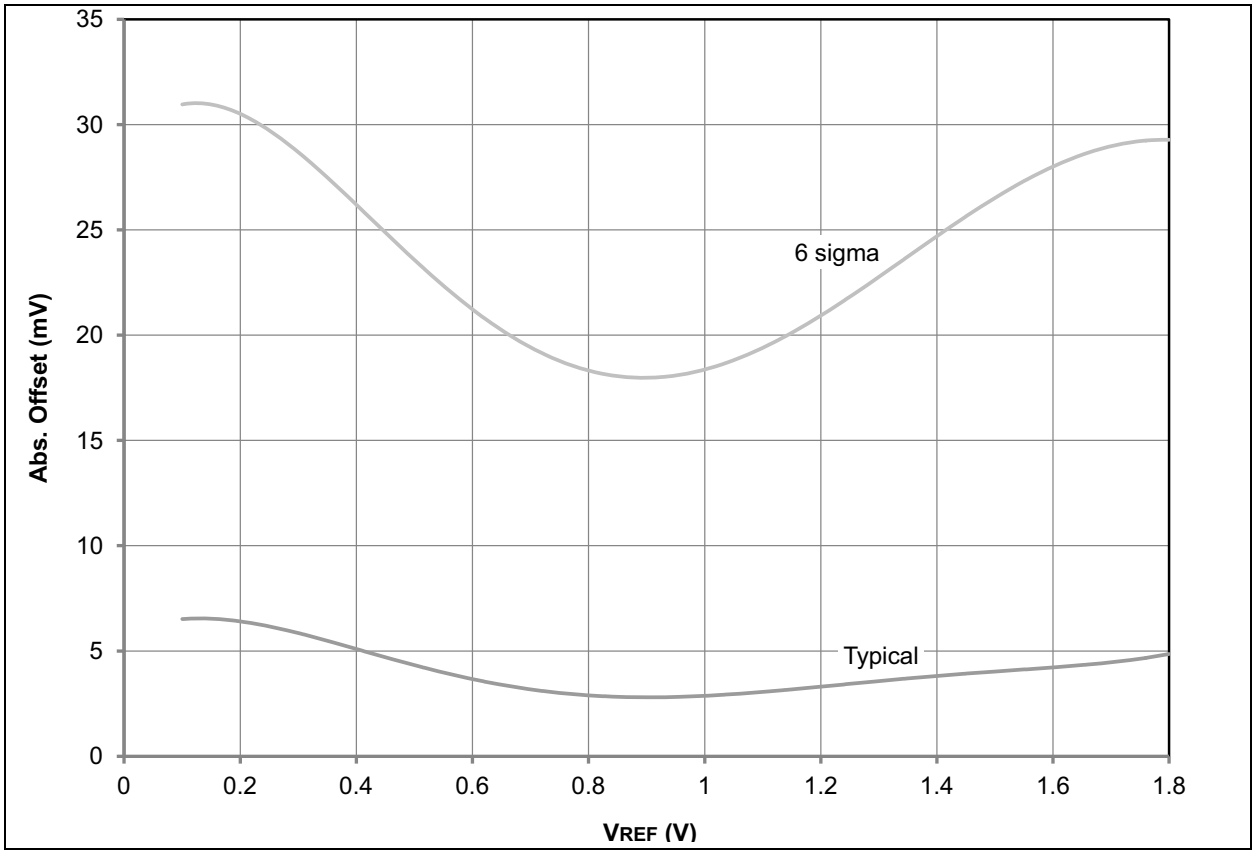


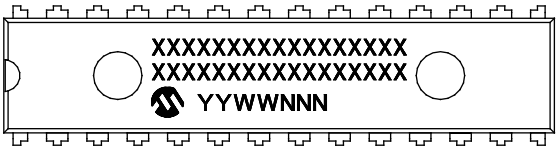
**FIGURE 28-89: PIC18LF2X/4XK22 COMPARATOR OFFSET VOLTAGE, NORMAL-POWER MODE; VDD=1.8V**



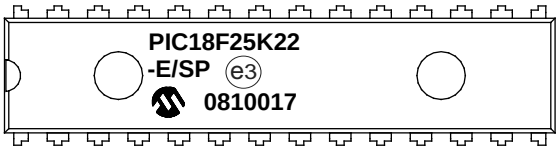
## 29.0 PACKAGING INFORMATION

### 29.1 Package Marking Information

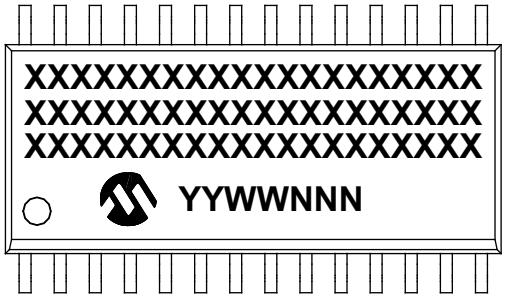
28-Lead SPDIP (.300”)




Example

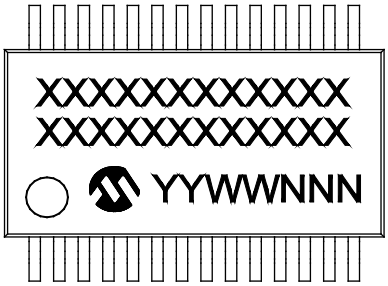



28-Lead SOIC (7.50 mm)

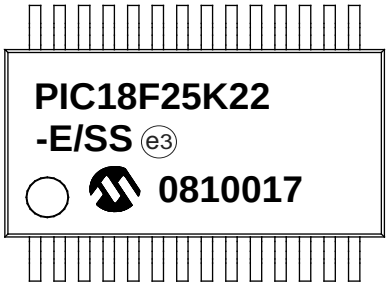



Example


28-Lead SSOP (5.30 mm)




Example




| | | |
|---|---|---|
| **Legend:** | XX...X | Customer-specific information or Microchip part number |
| | 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.