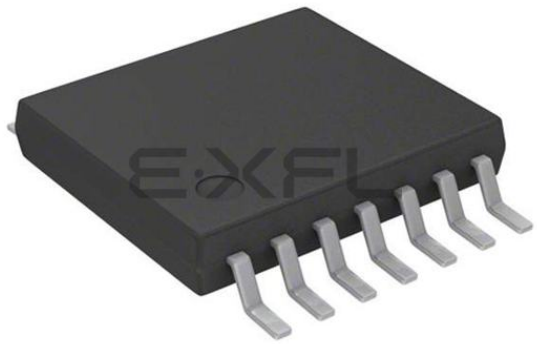


**Welcome to E-XFL.COM**

## 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 | 11 |
| Program Memory Size | 14KB (8K x 14) |
| Program Memory Type | FLASH |
| EEPROM Size | 256 x 8 |
| RAM Size | 1K x 8 |
| Voltage - Supply (Vcc/Vdd) | 1.8V ~ 5.5V |
| Data Converters | A/D 8x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 14-TSSOP (0.173", 4.40mm Width) |
| Supplier Device Package | 14-TSSOP |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/pic16f1825t-i-st |

**TABLE 3-6: PIC16(L)F1825/9 MEMORY MAP, BANKS 24-31**

| BANK 24 | | BANK 25 | | BANK 26 | | BANK 27 | | BANK 28 | | BANK 29 | | BANK 30 | | BANK 31 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| C00h | INDF0 | C80h | INDF0 | D00h | INDF0 | D80h | INDF0 | E00h | INDF0 | E80h | INDF0 | F00h | INDF0 | F80h | INDF0 |
| C01h | INDF1 | C81h | INDF1 | D01h | INDF1 | D81h | INDF1 | E01h | INDF1 | E81h | INDF1 | F01h | INDF1 | F81h | INDF1 |
| C02h | PCL | C82h | PCL | D02h | PCL | D82h | PCL | E02h | PCL | E82h | PCL | F02h | PCL | F82h | PCL |
| C03h | STATUS | C83h | STATUS | D03h | STATUS | D83h | STATUS | E03h | STATUS | E83h | STATUS | F03h | STATUS | F83h | STATUS |
| C04h | FSR0L | C84h | FSR0L | D04h | FSR0L | D84h | FSR0L | E04h | FSR0L | E84h | FSR0L | F04h | FSR0L | F84h | FSR0L |
| C05h | FSR0H | C85h | FSR0H | D05h | FSR0H | D85h | FSR0H | E05h | FSR0H | E85h | FSR0H | F05h | FSR0H | F85h | FSR0H |
| C06h | FSR1L | C86h | FSR1L | D06h | FSR1L | D86h | FSR1L | E06h | FSR1L | E86h | FSR1L | F06h | FSR1L | F86h | FSR1L |
| C07h | FSR1H | C87h | FSR1H | D07h | FSR1H | D87h | FSR1H | E07h | FSR1H | E87h | FSR1H | F07h | FSR1H | F87h | FSR1H |
| C08h | BSR | C88h | BSR | D08h | BSR | D88h | BSR | E08h | BSR | E88h | BSR | F08h | BSR | F88h | BSR |
| C09h | WREG | C89h | WREG | D09h | WREG | D89h | WREG | E09h | WREG | E89h | WREG | F09h | WREG | F89h | WREG |
| C0Ah | PCLATH | C8Ah | PCLATH | D0Ah | PCLATH | D8Ah | PCLATH | E0Ah | PCLATH | E8Ah | PCLATH | F0Ah | PCLATH | F8Ah | PCLATH |
| C0Bh | INTCON | C8Bh | INTCON | D0Bh | INTCON | D8Bh | INTCON | E0Bh | INTCON | E8Bh | INTCON | F0Bh | INTCON | F8Bh | INTCON |
| C0Ch | — | C8Ch | — | D0Ch | — | D8Ch | — | E0Ch | — | E8Ch | — | F0Ch | — | F8Ch | See Table 3-7 for register mapping details |
| C0Dh | — | C8Dh | — | D0Dh | — | D8Dh | — | E0Dh | — | E8Dh | — | F0Dh | — | F8Dh | |
| C0Eh | — | C8Eh | — | D0Eh | — | D8Eh | — | E0Eh | — | E8Eh | — | F0Eh | — | F8Eh | |
| C0Fh | — | C8Fh | — | D0Fh | — | D8Fh | — | E0Fh | — | E8Fh | — | F0Fh | — | F8Fh | |
| C10h | — | C90h | — | D10h | — | D90h | — | E10h | — | E90h | — | F10h | — | F90h | |
| C11h | — | C91h | — | D11h | — | D91h | — | E11h | — | E91h | — | F11h | — | F91h | |
| C12h | — | C92h | — | D12h | — | D92h | — | E12h | — | E92h | — | F12h | — | F92h | |
| C13h | — | C93h | — | D13h | — | D93h | — | E13h | — | E93h | — | F13h | — | F93h | |
| C14h | — | C94h | — | D14h | — | D94h | — | E14h | — | E94h | — | F14h | — | F94h | |
| C15h | — | C95h | — | D15h | — | D95h | — | E15h | — | E95h | — | F15h | — | F95h | |
| C16h | — | C96h | — | D16h | — | D96h | — | E16h | — | E96h | — | F16h | — | F96h | |
| C17h | — | C97h | — | D17h | — | D97h | — | E17h | — | E97h | — | F17h | — | F97h | |
| C18h | — | C98h | — | D18h | — | D98h | — | E18h | — | E98h | — | F18h | — | F98h | |
| C19h | — | C99h | — | D19h | — | D99h | — | E19h | — | E99h | — | F19h | — | F99h | |
| C1Ah | — | C9Ah | — | D1Ah | — | D9Ah | — | E1Ah | — | E9Ah | — | F1Ah | — | F9Ah | |
| C1Bh | — | C9Bh | — | D1Bh | — | D9Bh | — | E1Bh | — | E9Bh | — | F1Bh | — | F9Bh | |
| C1Ch | — | C9Ch | — | D1Ch | — | D9Ch | — | E1Ch | — | E9Ch | — | F1Ch | — | F9Ch | |
| C1Dh | — | C9Dh | — | D1Dh | — | D9Dh | — | E1Dh | — | E9Dh | — | F1Dh | — | F9Dh | |
| C1Eh | — | C9Eh | — | D1Eh | — | D9Eh | — | E1Eh | — | E9Eh | — | F1Eh | — | F9Eh | |
| C1Fh | — | C9Fh | — | D1Fh | — | D9Fh | — | E1Fh | — | E9Fh | — | F1Fh | — | F9Fh | |
| C20h | Unimplemented Read as '0' | CA0h | Unimplemented Read as '0' | D20h | Unimplemented Read as '0' | DA0h | Unimplemented Read as '0' | E20h | Unimplemented Read as '0' | EA0h | Unimplemented Read as '0' | F20h | Unimplemented Read as '0' | FA0h | |
| C6Fh | | CEFh | | D6Fh | | DEFh | | E6Fh | | EEFh | | F6Fh | | FEFh | |
| C70h | Accesses 70h – 7Fh | CF0h | Accesses 70h – 7Fh | D70h | Accesses 70h – 7Fh | DF0h | Accesses 70h – 7Fh | E70h | Accesses 70h – 7Fh | EF0h | Accesses 70h – 7Fh | F70h | Accesses 70h – 7Fh | FF0h | Accesses 70h – 7Fh |
| CFFh | | CFFh | | D7Fh | | DFFh | | E7Fh | | EFFh | | F7Fh | | FFFh | |

**Legend:** = Unimplemented data memory locations, read as '0'.

© 2010-2015 Microchip Technology Inc.

**TABLE 3-8: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)**

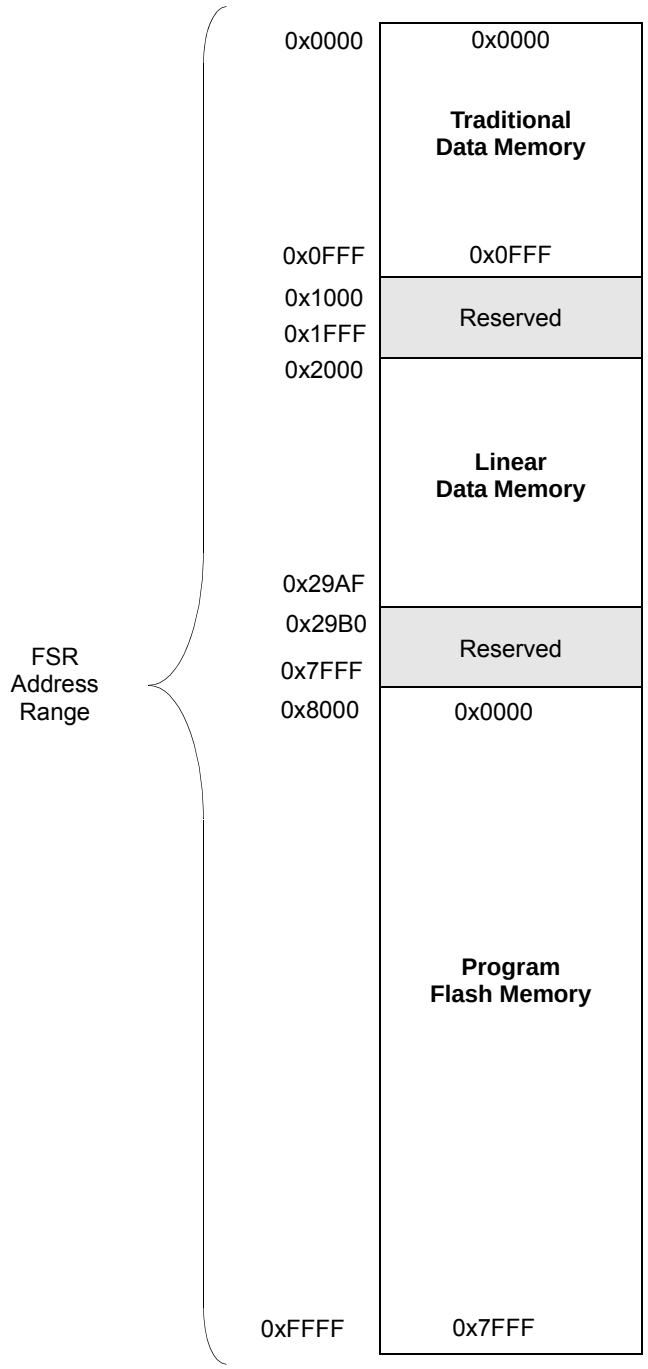
| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOR | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Bank 5** | | | | | | | | | | | |
| 280h(1) | INDF0 | Addressing this location uses contents of FSR0H/FSR0L to address data memory (not a physical register) | | | | | | | | xxxx xxxx | xxxx xxxx |
| 281h(1) | INDF1 | Addressing this location uses contents of FSR1H/FSR1L to address data memory (not a physical register) | | | | | | | | xxxx xxxx | xxxx xxxx |
| 282h(1) | PCL | Program Counter (PC) Least Significant Byte | | | | | | | | 0000 0000 | 0000 0000 |
| 283h(1) | STATUS | — | — | — | $\overline{TO}$ | $\overline{PD}$ | Z | DC | C | ---1 1000 | ---q quuu |
| 284h(1) | FSR0L | Indirect Data Memory Address 0 Low Pointer | | | | | | | | 0000 0000 | uuuu uuuu |
| 285h(1) | FSR0H | Indirect Data Memory Address 0 High Pointer | | | | | | | | 0000 0000 | 0000 0000 |
| 286h(1) | FSR1L | Indirect Data Memory Address 1 Low Pointer | | | | | | | | 0000 0000 | uuuu uuuu |
| 287h(1) | FSR1H | Indirect Data Memory Address 1 High Pointer | | | | | | | | 0000 0000 | 0000 0000 |
| 288h(1) | BSR | — | — | — | BSR<4:0> | | | | | ---0 0000 | ---0 0000 |
| 289h(1) | WREG | Working Register | | | | | | | | 0000 0000 | uuuu uuuu |
| 28Ah(1) | PCLATH | — | Write Buffer for the upper 7 bits of the Program Counter | | | | | | | -000 0000 | -000 0000 |
| 28Bh(1) | INTCON | GIE | PEIE | TMR0IE | INTE | IOCIE | TMR0IF | INTF | IOCIF | 0000 0000 | 0000 0000 |
| 28Ch | — | Unimplemented | | | | | | | | — | — |
| 28Dh | — | Unimplemented | | | | | | | | — | — |
| 28Eh | — | Unimplemented | | | | | | | | — | — |
| 28Fh | — | Unimplemented | | | | | | | | — | — |
| 290h | — | Unimplemented | | | | | | | | — | — |
| 291h | CCPR1L | Capture/Compare/PWM Register 1 (LSB) | | | | | | | | xxxx xxxx | uuuu uuuu |
| 292h | CCPR1H | Capture/Compare/PWM Register 1 (MSB) | | | | | | | | xxxx xxxx | uuuu uuuu |
| 293h | CCP1CON | P1M<1:0> | | DC1B<1:0> | | CCP1M<3:0> | | | | 0000 0000 | 0000 0000 |
| 294h | PWM1CON | P1RSEN | P1DC<6:0> | | | | | | | 0000 0000 | 0000 0000 |
| 295h | CCP1AS | CCP1ASE | CCP1AS<2:0> | | | PSS1AC<1:0> | | PSS1BD<1:0> | | 0000 0000 | 0000 0000 |
| 296h | PSTR1CON | — | — | — | STR1SYNC | STR1D | STR1C | STR1B | STR1A | ---0 0001 | ---0 0001 |
| 297h | — | Unimplemented | | | | | | | | — | — |
| 298h | CCPR2L | Capture/Compare/PWM Register 2 (LSB) | | | | | | | | xxxx xxxx | uuuu uuuu |
| 299h | CCPR2H | Capture/Compare/PWM Register 2 (MSB) | | | | | | | | xxxx xxxx | uuuu uuuu |
| 29Ah | CCP2CON | P2M1 | P2M0 | DC2B1 | DC2B0 | CCP2M3 | CCP2M2 | CCP2M1 | CCP2M0 | 0000 0000 | 0000 0000 |
| 29Bh | PWM2CON | P2RSEN | P2DC6 | P2DC5 | P2DC4 | P2DC3 | P2DC2 | P2DC1 | P2DC0 | 0000 0000 | 0000 0000 |
| 29Ch | CCP2AS | CCP2ASE | CCP2AS2 | CCP2AS1 | CCP2AS0 | PSS2AC1 | PSS2AC0 | PSS2BD1 | PSS2BD0 | 0000 0000 | 0000 0000 |
| 29Dh | PSTR2CON | — | — | — | STR2SYNC | STR2D | STR2C | STR2B | STR2A | ---0 0001 | ---0 0001 |
| 29Eh | CCPTMRS | C4TSEL1 | C4TSEL0 | C3TSEL1 | C3TSEL0 | C2TSEL1 | C2TSEL0 | C1TSEL1 | C1TSEL0 | 0000 0000 | 0000 0000 |
| 29Fh | — | Unimplemented | | | | | | | | — | — |

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

**Note 1:** These registers can be addressed from any bank.
**2:** PIC16(L)F1829 only.
**3:** PIC16(L)F1825 only.
**4:** Unimplemented, read as '1'.


© 2010-2015 Microchip Technology Inc.

**FIGURE 3-8: INDIRECT ADDRESSING**




**Note:** Not all memory regions are completely implemented. Consult device memory tables for memory limits.


© 2010-2015 Microchip Technology Inc.

#### 5.2.2.7 Internal Oscillator Clock Switch Timing

When switching between the HFINTOSC, MFINTOSC and the LFINTOSC, the new oscillator may already be shut down to save power (see Figure 5-7). If this is the case, there is a delay after the IRCF<3:0> bits of the OSCCON register are modified before the frequency selection takes place. The OSCSTAT register will reflect the current active status of the HFINTOSC, MFINTOSC and LFINTOSC oscillators. The sequence of a frequency selection is as follows:

1. IRCF<3:0> bits of the OSCCON register are modified.
2. If the new clock is shut down, a clock start-up delay is started.
3. Clock switch circuitry waits for a falling edge of the current clock.
4. The current clock is held low and the clock switch circuitry waits for a rising edge in the new clock.
5. The new clock is now active.
6. The OSCSTAT register is updated as required.
7. Clock switch is complete.

See Figure 5-7 for more details.

If the internal oscillator speed is switched between two clocks of the same source, there is no start-up delay before the new frequency is selected. Clock switching time delays are shown in Table 5-1.

Start-up delay specifications are located in the oscillator tables of **Section 30.0 "Electrical Specifications"**

**TABLE 6-1: SUMMARY OF REGISTERS ASSOCIATED WITH REFERENCE CLOCK SOURCES**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| CLKRCON | CLKREN | CLKROE | CLKRSLR | CLKRDC<1:0> | | CLKRDIV<2:0> | | | 72 |

**Legend:** — Unimplemented locations read as '0'. Shaded cells are not used by reference clock sources.

**TABLE 6-2: SUMMARY OF CONFIGURATION WORD WITH REFERENCE CLOCK SOURCES**

| Name | Bits | Bit -/7 | Bit -/6 | Bit 13/5 | Bit 12/4 | Bit 11/3 | Bit 10/2 | Bit 9/1 | Bit 8/0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|---|
| CONFIG1 | 13:8 | — | — | FCMEN | IESO | $\overline{\text{CLKOUTEN}}$ | BOREN<1:0> | | $\overline{\text{CPD}}$ | 48 |
| | 7:0 | $\overline{\text{CP}}$ | MCLRE | $\overline{\text{PWRTE}}$ | WDTE<1:0> | | FOSC<2:0> | | | |

**Legend:** — Unimplemented locations read as '0'. Shaded cells are not used by reference clock sources.

**EXAMPLE 11-5: WRITING TO FLASH PROGRAM MEMORY**

```
; This write routine assumes the following:
; 1. The 16 bytes of data are loaded, starting at the address in DATA_ADDR
; 2. Each word of data to be written is made up of two adjacent bytes in DATA_ADDR,
;    stored in little endian format
; 3. A valid starting address (the least significant bits = 000) is loaded in ADDRH:ADDRL
; 4. ADDRH and ADDRL are located in shared data memory 0x70 - 0x7F
;
        BCF         INTCON,GIE     ; Disable ints so required sequences will execute properly
        BANKSEL     EEADRH         ; Bank 3
        MOVF        ADDRH,W        ; Load initial address
        MOVWF       EEADRH         ;
        MOVF        ADDRL,W        ;
        MOVWF       EEADRL         ;
        MOVLW       LOW DATA_ADDR  ; Load initial data address
        MOVWF       FSR0L          ;
        MOVLW       HIGH DATA_ADDR ; Load initial data address
        MOVWF       FSR0H          ;
        BSF         EECON1,EEPGD   ; Point to program memory
        BCF         EECON1,CFGS    ; Not configuration space
        BSF         EECON1,WREN    ; Enable writes
        BSF         EECON1,LWLO    ; Only Load Write Latches
LOOP
        MOVIW       FSR0++         ; Load first data byte into lower
        MOVWF       EEDATL         ;
        MOVIW       FSR0++         ; Load second data byte into upper
        MOVWF       EEDATH         ;

        MOVF        EEADRL,W       ; Check if lower bits of address are '000'
        XORLW       0x07           ; Check if we're on the last of 8 addresses
        ANDLW       0x07           ;
        BTFSC       STATUS,Z       ; Exit if last of eight words,
        GOTO        START_WRITE    ;

  Required Sequence
        MOVLW       55h            ; Start of required write sequence:
        MOVWF       EECON2         ; Write 55h
        MOVLW       0AAh           ;
        MOVWF       EECON2         ; Write AAh
        BSF         EECON1,WR      ; Set WR bit to begin write
        NOP                        ; Any instructions here are ignored as processor
                                   ; halts to begin write sequence
        NOP                        ; Processor will stop here and wait for write to complete.

                                   ; After write processor continues with 3rd instruction.

        INCF        EEADRL,F       ; Still loading latches Increment address
        GOTO        LOOP           ; Write next latches

START_WRITE
        BCF         EECON1,LWLO    ; No more loading latches - Actually start Flash program
                                   ; memory write

  Required Sequence
        MOVLW       55h            ; Start of required write sequence:
        MOVWF       EECON2         ; Write 55h
        MOVLW       0AAh           ;
        MOVWF       EECON2         ; Write AAh
        BSF         EECON1,WR      ; Set WR bit to begin write
        NOP                        ; Any instructions here are ignored as processor
                                   ; halts to begin write sequence
        NOP                        ; Processor will stop here and wait for write complete.

                                   ; after write processor continues with 3rd instruction
        BCF         EECON1,WREN    ; Disable writes
        BSF         INTCON,GIE     ; Enable interrupts
```


© 2010-2015 Microchip Technology Inc.

**REGISTER 11-5: EECON1: EEPROM CONTROL 1 REGISTER**

| R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W/HC-0/0 | R/W-x/q | R/W-0/0 | R/S/HC-0/0 | R/S/HC-0/0 |
|---|---|---|---|---|---|---|---|
| EEPGD | CFGS | LWLO | FREE | WRERR | WREN | WR | RD |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| S = Bit can only be set | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | HC = Bit is cleared by hardware |

bit 7 **EEPGD:** Flash Program/Data EEPROM Memory Select bit
- 1 = Accesses program space Flash memory
- 0 = Accesses data EEPROM memory

bit 6 **CFGS:** Flash Program/Data EEPROM or Configuration Select bit
- 1 = Accesses Configuration, User ID and Device ID Registers
- 0 = Accesses Flash Program or data EEPROM Memory

bit 5 **LWLO:** Load Write Latches Only bit

If CFGS = 1 (Configuration space) **OR** CFGS = 0 and EEPGD = 1 (program Flash):
- 1 = The next WR command does not initiate a write; only the program memory latches are updated.
- 0 = The next WR command writes a value from EEDATH:EEDATL into program memory latches and initiates a write of all the data stored in the program memory latches.

If CFGS = 0 and EEPGD = 0: (Accessing data EEPROM)
LWLO is ignored. The next WR command initiates a write to the data EEPROM.

bit 4 **FREE:** Program Flash Erase Enable bit

If CFGS = 1 (Configuration space) **OR** CFGS = 0 and EEPGD = 1 (program Flash):
- 1 = Performs an erase operation on the next WR command (cleared by hardware after completion of erase).
- 0 = Performs a write operation on the next WR command.

If EEPGD = 0 and CFGS = 0: (Accessing data EEPROM)
FREE is ignored. The next WR command will initiate both a erase cycle and a write cycle.

bit 3 **WRERR:** EEPROM Error Flag bit
- 1 = Condition indicates an improper program or erase sequence attempt or termination (bit is set automatically on any set attempt (write '1') of the WR bit).
- 0 = The program or erase operation completed normally.

bit 2 **WREN:** Program/Erase Enable bit
- 1 = Allows program/erase cycles
- 0 = Inhibits programming/erasing of program Flash and data EEPROM

bit 1 **WR:** Write Control bit
- 1 = Initiates a program Flash or data EEPROM program/erase operation. The operation is self-timed and the bit is cleared by hardware once operation is complete. The WR bit can only be set (not cleared) in software.
- 0 = Program/erase operation to the Flash or data EEPROM is complete and inactive.

bit 0 **RD:** Read Control bit
- 1 = Initiates an program Flash or data EEPROM read. Read takes one cycle. RD is cleared in hardware. The RD bit can only be set (not cleared) in software.
- 0 = Does not initiate a program Flash or data EEPROM data read.


© 2010-2015 Microchip Technology Inc.

### 12.2.2 PORTA FUNCTIONS AND OUTPUT PRIORITIES

Each PORTA pin is multiplexed with other functions. The pins, their combined functions and their output priorities are briefly described here. For additional information, refer to the appropriate section in this data sheet.

When multiple outputs are enabled, the actual pin control goes to the peripheral with the lowest number in the following lists.

Analog input functions, such as ADC, comparator and Cap Sense inputs, are not shown in the priority lists. These inputs are active when the I/O pin is set for Analog mode using the ANSELx registers. Digital output functions may control the pin when it is in Analog mode with the priority shown below.

**TABLE 12-2: PORTA OUTPUT PRIORITY**

| Pin Name | Function Priority(1) |
|---|---|
| RA0 | ICSPDAT<br>ICDDAT<br>DACOUT |
| RA1 | ICSPCLK<br>ICDCLK<br>RX/DT(2) |
| RA2 | SRQ<br>C1OUT<br>CCP3 |
| RA3 | None (input only) |
| RA4 | CLKOUT<br>T1OSO<br>CLKR<br>SDO1<br>P2B(2) |
| RA5 | SDO2 (PIC16(L)F1829 only)<br>CCP2(2)/P2A(2) |

**Note 1:** Priority listed from highest to lowest.
**2:** Pin function is selectable via the APFCON0 or APFCON1 register.

## 21.1 Timer1 Operation

The Timer1 module is a 16-bit incrementing counter which is accessed through the TMR1H:TMR1L register pair. Writes to TMR1H or TMR1L directly update the counter.

When used with an internal clock source, the module is a timer and increments on every instruction cycle. When used with an external clock source, the module can be used as either a timer or counter and increments on every selected edge of the external source.

Timer1 is enabled by configuring the TMR1ON and TMR1GE bits in the T1CON and T1GCON registers, respectively. Table 21-1 displays the Timer1 enable selections.

**TABLE 21-1: TIMER1 ENABLE SELECTIONS**

| TMR1ON | TMR1GE | Timer1 Operation |
|---|---|---|
| 0 | 0 | Off |
| 0 | 1 | Off |
| 1 | 0 | Always On |
| 1 | 1 | Count Enabled |

## 21.2 Clock Source Selection

The TMR1CS<1:0> and T1OSCEN bits of the T1CON register are used to select the clock source for Timer1. Table 21-2 displays the clock source selections.

### 21.2.1 INTERNAL CLOCK SOURCE

When the internal clock source is selected the TMR1H:TMR1L register pair will increment on multiples of FOSC as determined by the Timer1 prescaler.

When the FOSC internal clock source is selected, the Timer1 register value will increment by four counts every instruction clock cycle. Due to this condition, a 2 LSB error in resolution will occur when reading the Timer1 value. To utilize the full resolution of Timer1, an asynchronous input signal must be used to gate the Timer1 clock input.

The following asynchronous sources may be used:

- Asynchronous event on the T1G pin to Timer1 gate
- C1 or C2 comparator input to Timer1 gate

### 21.2.2 EXTERNAL CLOCK SOURCE

When the external clock source is selected, the Timer1 module may work as a timer or a counter.

When enabled to count, Timer1 is incremented on the rising edge of the external clock input T1CKI or the capacitive sensing oscillator signal. Either of these external clock sources can be synchronized to the microcontroller system clock or they can run asynchronously.

When used as a timer with a clock oscillator, an external 32.768 kHz crystal can be used in conjunction with the dedicated internal oscillator circuit.

**Note:** In Counter mode, a falling edge must be registered by the counter prior to the first incrementing rising edge after any one or more of the following conditions:

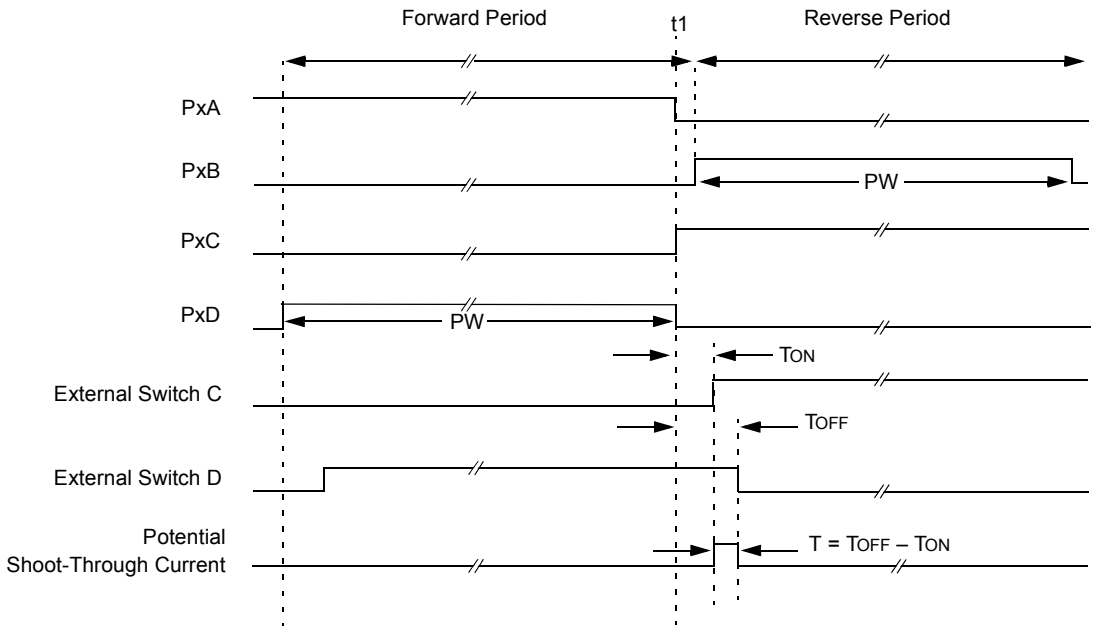
- Timer1 enabled after POR
- Write to TMR1H or TMR1L
- Timer1 is disabled
- Timer1 is disabled (TMR1ON = 0) when T1CKI is high then Timer1 is enabled (TMR1ON=1) when T1CKI is low.

**TABLE 21-2: CLOCK SOURCE SELECTIONS**

| TMR1CS1 | TMR1CS0 | T1OSCEN | Clock Source |
|---|---|---|---|
| 0 | 0 | x | Instruction Clock (FOSC/4) |
| 0 | 1 | x | System Clock (FOSC) |
| 1 | 0 | 0 | External Clocking on T1CKI Pin |
| 1 | 0 | 0 | External Clocking on T1CKI Pin |
| 1 | 1 | x | Capacitive Sensing Oscillator |


© 2010-2015 Microchip Technology Inc.

**FIGURE 24-13: EXAMPLE OF PWM DIRECTION CHANGE AT NEAR 100% DUTY CYCLE**




**Note 1:** All signals are shown as active-high.

**2:** TON is the turn-on delay of power switch QC and its driver.

**3:** TOFF is the turn-off delay of power switch QD and its driver.

## 25.2 SPI Mode Overview

The Serial Peripheral Interface (SPI) bus is a synchronous serial data communication bus that operates in Full-Duplex mode. Devices communicate in a master/slave environment where the master device initiates the communication. A slave device is controlled through a chip select known as Slave Select.

The SPI bus specifies four signal connections:

- Serial Clock (SCKx)
- Serial Data Out (SDOx)
- Serial Data In (SDIx)
- Slave Select ($\overline{SS}$x)

Figure 25-1 shows the block diagram of the MSSPx module when operating in SPI Mode.

The SPI bus operates with a single master device and one or more slave devices. When multiple slave devices are used, an independent Slave Select connection is required from the master device to each slave device.

Figure 25-4 shows a typical connection between a master device and multiple slave devices.

The master selects only one slave at a time. Most slave devices have tri-state outputs so their output signal appears disconnected from the bus when they are not selected.

Transmissions involve two shift registers, eight bits in size, one in the master and one in the slave. With either the master or the slave device, data is always shifted out one bit at a time, with the Most Significant bit (MSb) shifted out first. At the same time, a new Least Significant bit (LSb) is shifted into the same register.

Figure 25-5 shows a typical connection between two processors configured as master and slave devices.

Data is shifted out of both shift registers on the programmed clock edge and latched on the opposite edge of the clock.

The master device transmits information out on its SDOx output pin which is connected to, and received by, the slave's SDIx input pin. The slave device transmits information out on its SDOx output pin, which is connected to, and received by, the master's SDIx input pin.

To begin communication, the master device first sends out the clock signal. Both the master and the slave devices should be configured for the same clock polarity.

The master device starts a transmission by sending out the MSb from its shift register. The slave device reads this bit from that same line and saves it into the LSb position of its shift register.

During each SPI clock cycle, a full-duplex data transmission occurs. This means that while the master device is sending out the MSb from its shift register (on its SDOx pin) and the slave device is reading this bit and saving it as the LSb of its shift register, that the slave device is also sending out the MSb from its shift register (on its SDOx pin) and the master device is reading this bit and saving it as the LSb of its shift register.

After eight bits have been shifted out, the master and slave have exchanged register values.

If there is more data to exchange, the shift registers are loaded with new data and the process repeats itself.

Whether the data is meaningful or not (dummy data), depends on the application software. This leads to three scenarios for data transmission:

- Master sends useful data and slave sends dummy data.
- Master sends useful data and slave sends useful data.
- Master sends dummy data and slave sends useful data.

Transmissions may involve any number of clock cycles. When there is no more data to be transmitted, the master stops sending the clock signal and it deselects the slave.

Every slave device connected to the bus that has not been selected through its slave select line must disregard the clock and transmission signals and must not transmit out any data of its own.


© 2010-2015 Microchip Technology Inc.

### 25.2.3 SPI MASTER MODE

The master can initiate the data transfer at any time because it controls the SCKx line. The master determines when the slave (Processor 2, Figure 25-5) is to broadcast data by the software protocol.

In Master mode, the data is transmitted/received as soon as the SSPxBUF register is written to. If the SPI is only going to receive, the SDOx output could be disabled (programmed as an input). The SSPxSR register will continue to shift in the signal present on the SDIx pin at the programmed clock rate. As each byte is received, it will be loaded into the SSPxBUF register as if a normal received byte (interrupts and Status bits appropriately set).

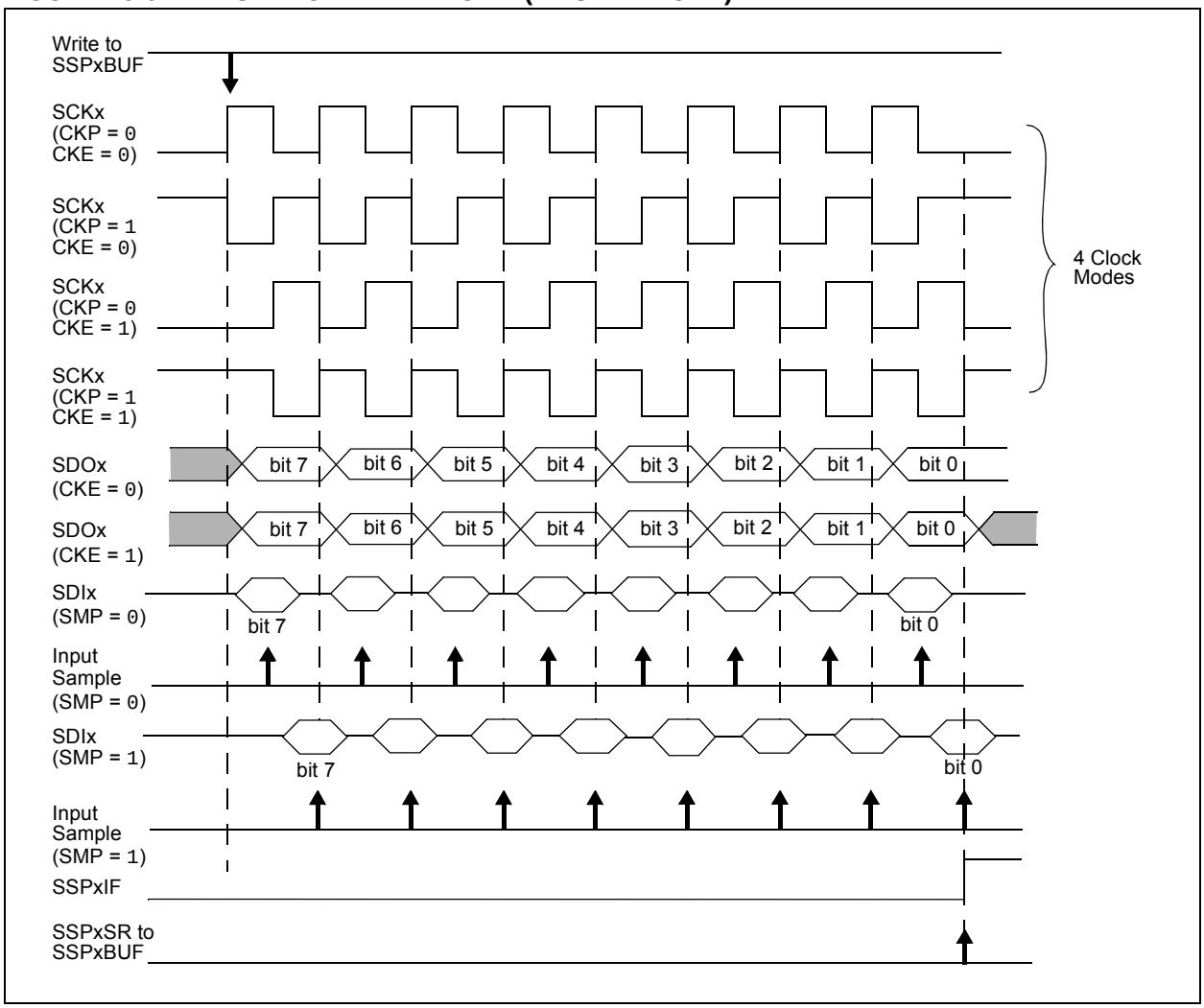
The clock polarity is selected by appropriately programming the CKP bit of the SSPxCON1 register and the CKE bit of the SSPxSTAT register. This then, would give waveforms for SPI communication as shown in Figure 25-6, Figure 25-8, Figure 25-9 and Figure 25-10, where the MSB is transmitted first. In Master mode, the SPI clock rate (bit rate) is user programmable to be one of the following:

- FOSC/4 (or TCY)
- FOSC/16 (or 4 * TCY)
- FOSC/64 (or 16 * TCY)
- Timer2 output/2
- FOSC/(4 * (SSPxADD + 1))

Figure 25-6 shows the waveforms for Master mode.

When the CKE bit is set, the SDOx data is valid before there is a clock edge on SCKx. The change of the input sample is shown based on the state of the SMP bit. The time when the SSPxBUF is loaded with the received data is shown.

**FIGURE 25-6: SPI MODE WAVEFORM (MASTER MODE)**



## 25.6 I²C MASTER MODE

Master mode is enabled by setting and clearing the appropriate SSPM bits in the SSPxCON1 register and by setting the SSPEN bit. In Master mode, the SDAx and SCKx pins must be configured as inputs. The MSSP peripheral hardware will override the output driver TRIS controls when necessary to drive the pins low.

Master mode of operation is supported by interrupt generation on the detection of the Start and Stop conditions. The Stop (P) and Start (S) bits are cleared from a Reset or when the MSSPx module is disabled. Control of the I²C bus may be taken when the P bit is set, or the bus is Idle.

In Firmware Controlled Master mode, user code conducts all I²C bus operations based on Start and Stop bit condition detection. Start and Stop condition detection is the only active circuitry in this mode. All other communication is done by the user software directly manipulating the SDAx and SCLx lines.

The following events will cause the SSPx Interrupt Flag bit, SSPxIF, to be set (SSPx interrupt, if enabled):

- Start condition detected
- Stop condition detected
- Data transfer byte transmitted/received
- Acknowledge transmitted/received
- Repeated Start generated

**Note 1:** The MSSPx module, when configured in I²C Master mode, does not allow queuing of events. For instance, the user is not allowed to initiate a Start condition and immediately write the SSPxBUF register to initiate transmission before the Start condition is complete. In this case, the SSPxBUF will not be written to and the WCOL bit will be set, indicating that a write to the SSPxBUF did not occur

**2:** When in Master mode, Start/Stop detection is masked and an interrupt is generated when the SEN/PEN bit is cleared and the generation is complete.

### 25.6.1 I²C MASTER MODE OPERATION

The master device generates all of the serial clock pulses and the Start and Stop conditions. A transfer is ended with a Stop condition or with a Repeated Start condition. Since the Repeated Start condition is also the beginning of the next serial transfer, the I²C bus will not be released.

In Master Transmitter mode, serial data is output through SDAx, while SCLx outputs the serial clock. The first byte transmitted contains the slave address of the receiving device (7 bits) and the Read/Write (R/$\overline{W}$) bit. In this case, the R/$\overline{W}$ bit will be logic '0'. Serial data is transmitted eight bits at a time. After each byte is transmitted, an Acknowledge bit is received. Start and Stop conditions are output to indicate the beginning and the end of a serial transfer.

In Master Receive mode, the first byte transmitted contains the slave address of the transmitting device (7 bits) and the R/$\overline{W}$ bit. In this case, the R/$\overline{W}$ bit will be logic '1'. Thus, the first byte transmitted is a 7-bit slave address followed by a '1' to indicate the receive bit. Serial data is received via SDAx, while SCLx outputs the serial clock. Serial data is received eight bits at a time. After each byte is received, an Acknowledge bit is transmitted. Start and Stop conditions indicate the beginning and end of transmission.

A Baud Rate Generator is used to set the clock frequency output on SCLx. See **Section 25.7 "Baud Rate Generator"** for more detail.


© 2010-2015 Microchip Technology Inc.

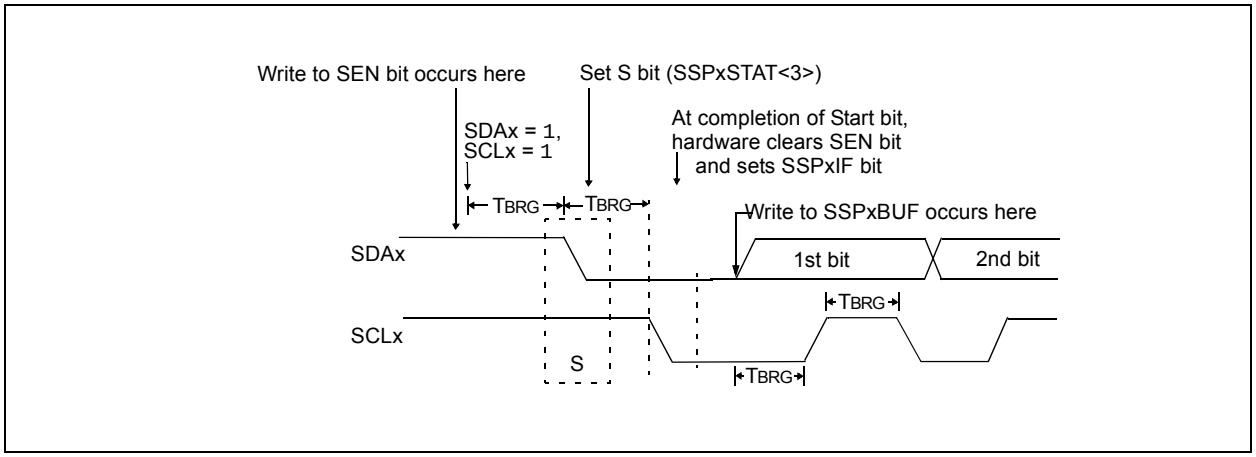
### 25.6.4 I²C MASTER MODE START CONDITION TIMING

To initiate a Start condition (Figure 25-26), the user sets the Start Enable bit, SEN bit of the SSPxCON2 register. If the SDAx and SCLx pins are sampled high, the Baud Rate Generator is reloaded with the contents of SSPxADD<7:0> and starts its count. If SCLx and SDAx are both sampled high when the Baud Rate Generator times out (TBRG), the SDAx pin is driven low. The action of the SDAx being driven low while SCLx is high is the Start condition and causes the S bit of the SSPxSTAT1 register to be set. Following this, the Baud Rate Generator is reloaded with the contents of SSPxADD<7:0> and resumes its count. When the Baud Rate Generator times out (TBRG), the SEN bit of the SSPxCON2 register will be automatically cleared by hardware; the Baud Rate Generator is suspended, leaving the SDAx line held low and the Start condition is complete.

**Note 1:** If at the beginning of the Start condition, the SDAx and SCLx pins are already sampled low, or if during the Start condition, the SCLx line is sampled low before the SDAx line is driven low, a bus collision occurs, the Bus Collision Interrupt Flag, BCLxIF, is set, the Start condition is aborted and the I²C module is reset into its Idle state.

**2:** The Philips I²C™ Specification states that a bus collision cannot occur on a Start.

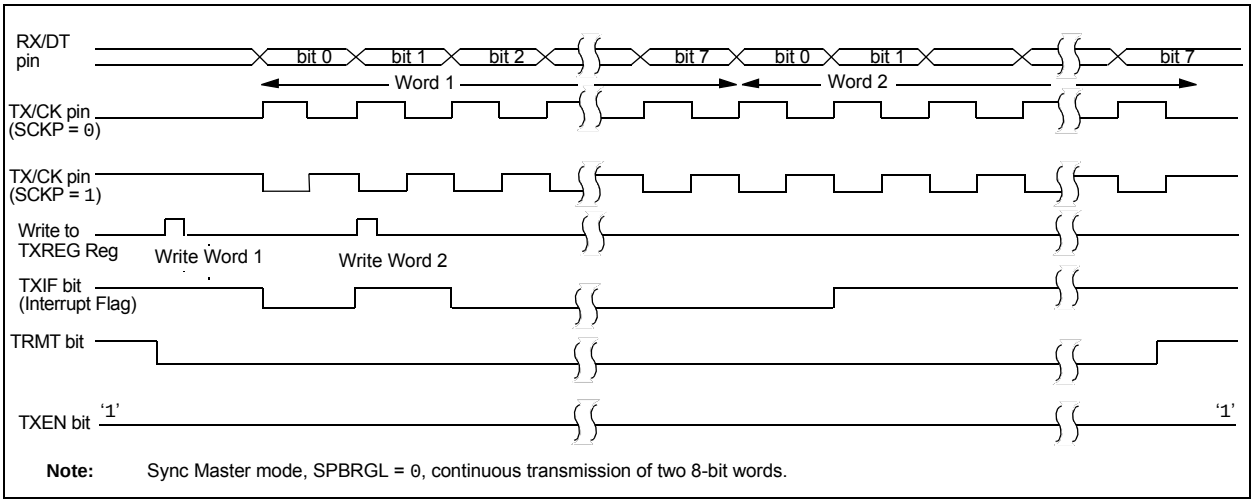
**FIGURE 25-26: FIRST START BIT TIMING**





© 2010-2015 Microchip Technology Inc.

**FIGURE 26-10: SYNCHRONOUS TRANSMISSION**




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

**FIGURE 26-11: SYNCHRONOUS TRANSMISSION (THROUGH TXEN)**

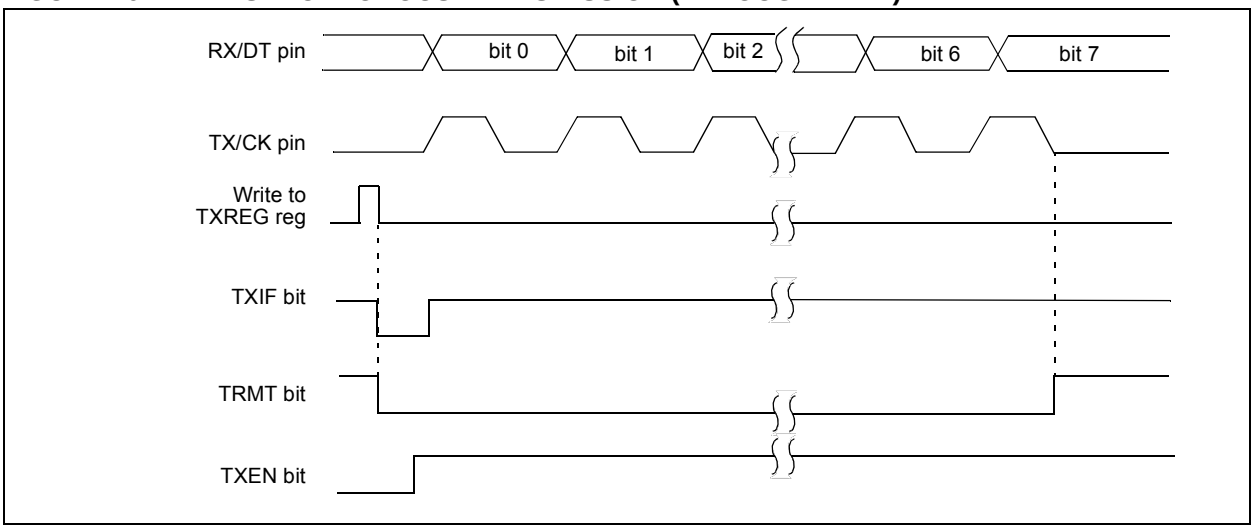



**TABLE 26-7: SUMMARY OF REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER TRANSMISSION**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| APFCON0 | RXDTSEL | SDO1SEL(1) | SS1SEL(1) | — | T1GSEL | TXCKSEL | — | — | 118 |
| BAUDCON | ABDOVF | RCIDL | — | SCKP | BRG16 | — | WUE | ABDEN | 292 |
| INTCON | GIE | PEIE | TMR0IE | INTE | IOCIE | TMR0IF | INTF | IOCIF | 87 |
| PIE1 | TMR1GIE | ADIE | RCIE | TXIE | SSP1IE | CCP1IE | TMR2IE | TMR1IE | 88 |
| PIR1 | TMR1GIF | ADIF | RCIF | TXIF | SSP1IF | CCP1IF | TMR2IF | TMR1IF | 92 |
| RCSTA | SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D | 291 |
| SPBRGL | SPBRG<7:0> | | | | | | | | 293* |
| SPBRGH | SPBRG<15:8> | | | | | | | | 293* |
| TRISC | TRISC7 | TRISC6 | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 | 133 |
| TXREG | EUSART Transmit Data Register | | | | | | | | 283* |
| TXSTA | CSRC | TX9 | TXEN | SYNC | SENDB | BRGH | TRMT | TX9D | 290 |

**Legend:** — Unimplemented location, read as '0'. Shaded cells are not used for synchronous master transmission.
* Page provides register information.

**Note 1:** PIC16(L)F1825 only.

**SWAPF** **Swap Nibbles in f**

| | |
|---|---|
| Syntax: | [ *label* ] SWAPF f,d |
| Operands: | $0 \le f \le 127$<br>$d \in [0,1]$ |
| Operation: | (f<3:0>) → (destination<7:4>),<br>(f<7:4>) → (destination<3:0>) |
| Status Affected: | None |
| Description: | The upper and lower nibbles of register 'f' are exchanged. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed in register 'f'. |

**TRIS** **Load TRIS Register with W**

| | |
|---|---|
| Syntax: | [ *label* ] TRIS f |
| Operands: | $5 \le f \le 7$ |
| Operation: | (W) → TRIS register 'f' |
| Status Affected: | None |
| Description: | Move data from W register to TRIS register.<br>When 'f' = 5, TRISA is loaded.<br>When 'f' = 6, TRISB is loaded.<br>When 'f' = 7, TRISC is loaded. |

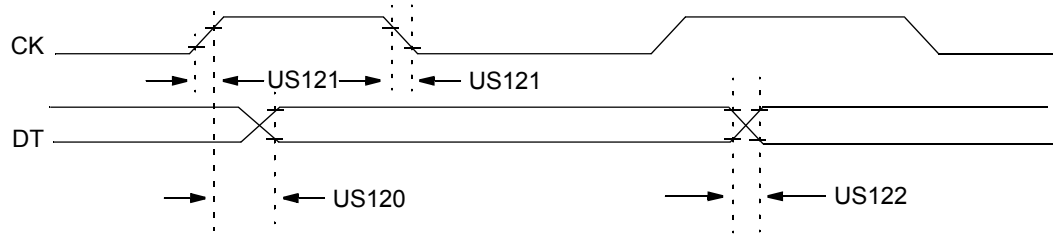
**XORLW** **Exclusive OR literal with W**

| | |
|---|---|
| Syntax: | [ *label* ] XORLW k |
| Operands: | $0 \le k \le 255$ |
| Operation: | (W) .XOR. k → (W) |
| Status Affected: | Z |
| Description: | The contents of the W register are XOR'ed with the 8-bit literal 'k'. The result is placed in the W register. |

**XORWF** **Exclusive OR W with f**

| | |
|---|---|
| Syntax: | [ *label* ] XORWF f,d |
| Operands: | $0 \le f \le 127$<br>$d \in [0,1]$ |
| Operation: | (W) .XOR. (f) → (destination) |
| Status Affected: | Z |
| Description: | Exclusive OR the contents of the W register with register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'. |

**FIGURE 30-14: USART SYNCHRONOUS TRANSMISSION (MASTER/SLAVE) TIMING**

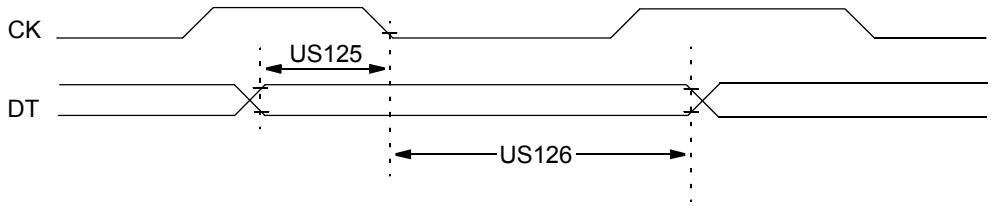



**Note:** Refer to Figure 30-5 for load conditions.

**TABLE 30-13: USART SYNCHRONOUS TRANSMISSION REQUIREMENTS**

| **Standard Operating Conditions (unless otherwise stated)**<br>Operating Temperature -40°C ≤ TA ≤ +125°C | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Param. No.** | **Symbol** | **Characteristic** | | **Min.** | **Max.** | **Units** | **Conditions** |
| US120 | TCKH2DTV | SYNC XMIT (Master and Slave)<br>Clock high to data-out valid | 3.0-5.5V | — | 80 | ns | |
| | | | 1.8-5.5V | — | 100 | ns | |
| US121 | TCKRF | Clock out rise time and fall time (Master mode) | 3.0-5.5V | — | 45 | ns | |
| | | | 1.8-5.5V | — | 50 | ns | |
| US122 | TDTRF | Data-out rise time and fall time | 3.0-5.5V | — | 45 | ns | |
| | | | 1.8-5.5V | — | 50 | ns | |

**FIGURE 30-15: USART SYNCHRONOUS RECEIVE (MASTER/SLAVE) TIMING**




**Note:** Refer to Figure 30-5 for load conditions.

**TABLE 30-14: USART SYNCHRONOUS RECEIVE REQUIREMENTS**

| **Standard Operating Conditions (unless otherwise stated)**<br>Operating Temperature -40°C ≤ TA ≤ +125°C | | | | | | |
|---|---|---|---|---|---|---|
| **Param. No.** | **Symbol** | **Characteristic** | **Min.** | **Max.** | **Units** | **Conditions** |
| US125 | TDTV2CKL | SYNC RCV (Master and Slave)<br>Data-hold before CK ↓ (DT hold time) | 10 | — | ns | |
| US126 | TCKL2DTL | Data-hold after CK ↓ (DT hold time) | 15 | — | ns | |

## 30.9 High Temperature Operation

This section outlines the specifications for the following devices operating in the high temperature range between -40°C and 150°C.(1)

- PIC16F1825(4)
- PIC16F1829(4)

When the value of any parameter is identical for both the 125°C Extended and the 150°C High Temp. temperature ranges, then that value will be found in the standard specification tables shown earlier in this chapter, under the fields listed for the 125°C Extended temperature range. If the value of any parameter is unique to the 150°C High Temp. temperature range, then it will be listed here, in this section of the data sheet.

If a Silicon Errata exists for the product and it lists a modification to the 125°C Extended temperature range value, one that is also shared at the 150°C High Temp. temperature range, then that modified value will apply to both temperature ranges.

**Note 1:** AEC-Q100 reliability testing for devices intended to operate at 150°C is 1,000 hours. Any design in which the total operating time from 125°C to 150°C will be greater than 1,000 hours is not warranted without prior written approval from Microchip Technology Inc.

**2:** Writes are **not allowed** for Flash program memory above 125°C.

**3:** The temperature range indicator in the catalog part number and device marking is “H” for -40°C to 150°C.

Example: PIC16F1825T-H/SL indicates the device is shipped in a Tape and Reel configuration, in the SOIC package, and is rated for operation from -40°C to 150°C.

**4:** The low voltage versions of these devices, PIC16LF1825 and PIC16LF1829, are not released for operation above +125°C.

**5:** Errata Sheet DS80517 lists various mask revisions. 150°C operation applies only to revisions A4 and later.

**6:** The Capacitive Sensing module (CPS) should not be used in high temperature devices. Function and its parametrics are not warranted.

**7:** Only SOIC (SL), TSSOP (ST), SSOP (SS) and QFN (ML) packages will be offered, not PDIP or UQFN.

**TABLE 30-18: ABSOLUTE MAXIMUM RATINGS**

| Parameter | Condition | Value |
|---|---|---|
| Max. Current: $V_{DD}$ | Source | 15 mA |
| Max. Current: $V_{SS}$ | Sink | 15 mA |
| Max. Current: Pin | Source | 5 mA |
| Max. Current: Pin | Sink | 5 mA |
| Max. Storage Temperature | — | -65°C to 155°C |
| Max. Junction Temperature | — | +155°C |
| Ambient Temperature under Bias | — | -40°C to +150°C |

**Note:** Stresses above those listed under *Absolute Maximum Ratings* may cause permanent damage to the device. This is a stress rating only, and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure above maximum rating conditions for extended periods may affect device reliability.

**FIGURE 31-39: IPD, COMPARATOR, NORMAL-POWER MODE (CxSP = 1), PIC16LF1825/9 ONLY**

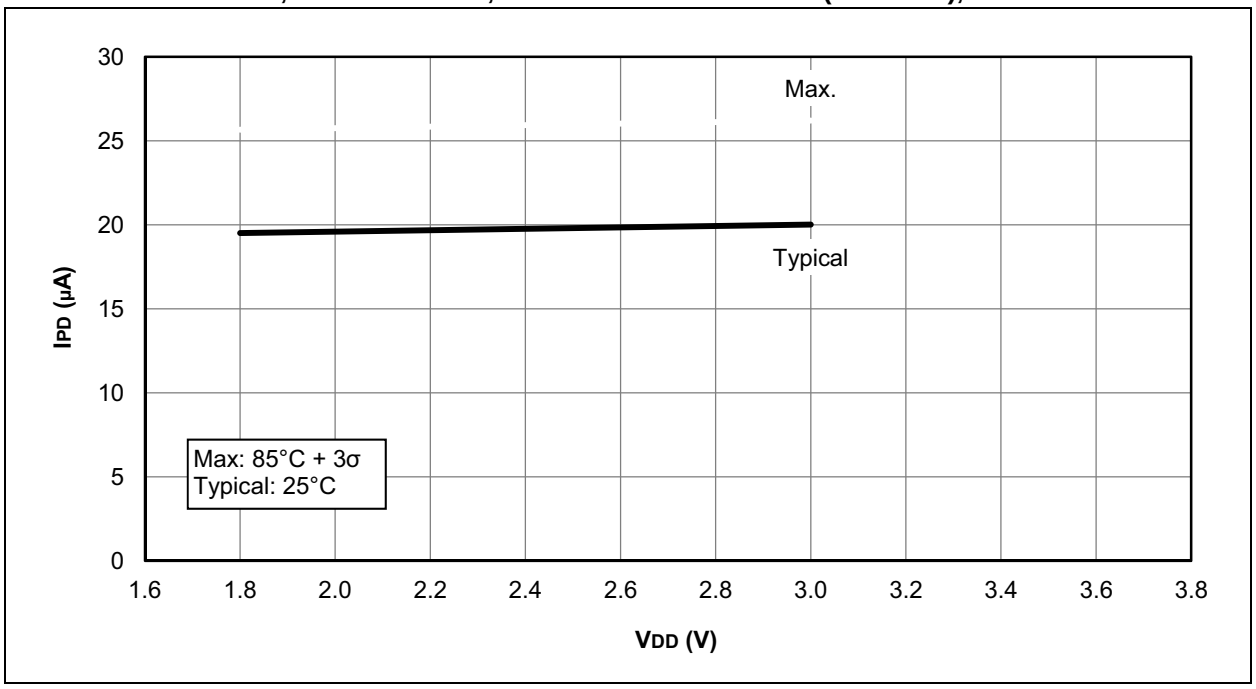
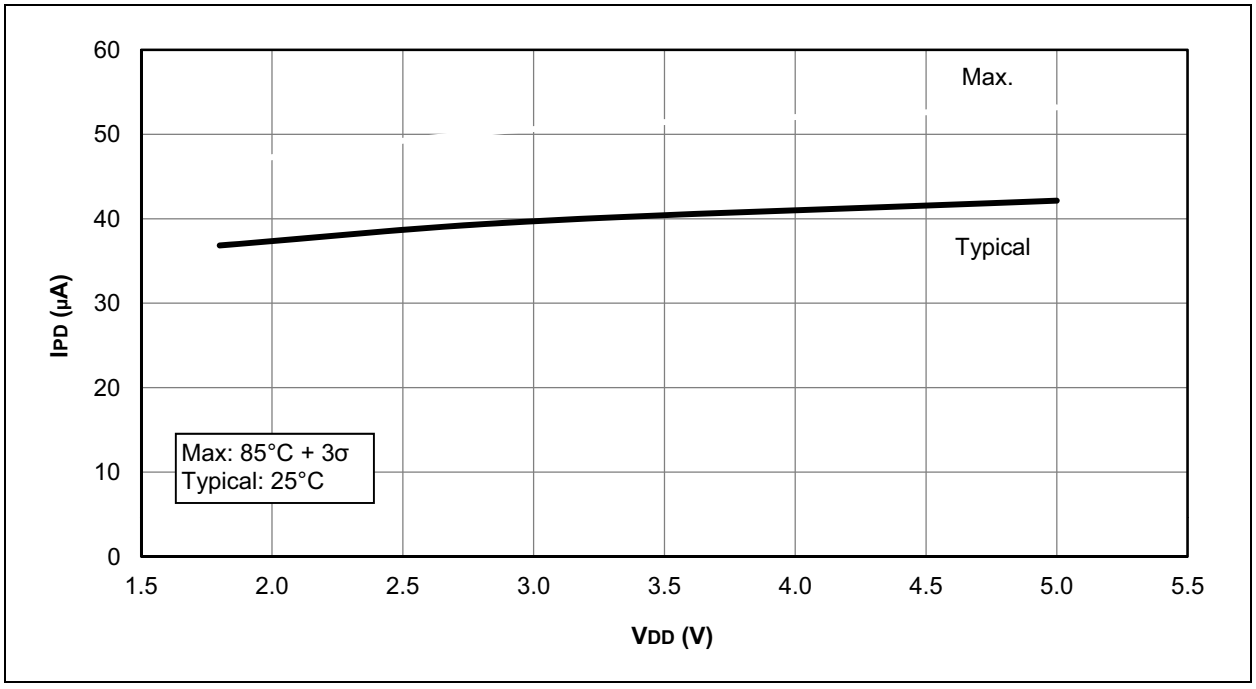



**FIGURE 31-40: IPD, COMPARATOR, NORMAL-POWER MODE (CxSP = 1), PIC16F1825/9 ONLY**





© 2010-2015 Microchip Technology Inc.

## 32.6 MPLAB X SIM Software Simulator

The MPLAB X SIM Software Simulator allows code development in a PC-hosted environment by simulating the PIC MCUs and dsPIC DSCs on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a comprehensive stimulus controller. Registers can be logged to files for further run-time analysis. The trace buffer and logic analyzer display extend the power of the simulator to record and track program execution, actions on I/O, most peripherals and internal registers.

The MPLAB X SIM Software Simulator fully supports symbolic debugging using the MPLAB XC Compilers, and the MPASM and MPLAB Assemblers. The software simulator offers the flexibility to develop and debug code outside of the hardware laboratory environment, making it an excellent, economical software development tool.

## 32.7 MPLAB REAL ICE In-Circuit Emulator System

The MPLAB REAL ICE In-Circuit Emulator System is Microchip's next generation high-speed emulator for Microchip Flash DSC and MCU devices. It debugs and programs all 8, 16 and 32-bit MCU, and DSC devices with the easy-to-use, powerful graphical user interface of the MPLAB X IDE.

The emulator is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with either a connector compatible with in-circuit debugger systems (RJ-11) or with the new high-speed, noise tolerant, Low-Voltage Differential Signal (LVDS) interconnection (CAT5).

The emulator is field upgradable through future firmware downloads in MPLAB X IDE. MPLAB REAL ICE offers significant advantages over competitive emulators including full-speed emulation, run-time variable watches, trace analysis, complex breakpoints, logic probes, a ruggedized probe interface and long (up to three meters) interconnection cables.

## 32.8 MPLAB ICD 3 In-Circuit Debugger System

The MPLAB ICD 3 In-Circuit Debugger System is Microchip's most cost-effective, high-speed hardware debugger/programmer for Microchip Flash DSC and MCU devices. It debugs and programs PIC Flash microcontrollers and dsPIC DSCs with the powerful, yet easy-to-use graphical user interface of the MPLAB IDE.

The MPLAB ICD 3 In-Circuit Debugger probe is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with a connector compatible with the MPLAB ICD 2 or MPLAB REAL ICE systems (RJ-11). MPLAB ICD 3 supports all MPLAB ICD 2 headers.

## 32.9 PICkit 3 In-Circuit Debugger/ Programmer

The MPLAB PICkit 3 allows debugging and programming of PIC and dsPIC Flash microcontrollers at a most affordable price point using the powerful graphical user interface of the MPLAB IDE. The MPLAB PICkit 3 is connected to the design engineer's PC using a full-speed USB interface and can be connected to the target via a Microchip debug (RJ-11) connector (compatible with MPLAB ICD 3 and MPLAB REAL ICE). The connector uses two device I/O pins and the Reset line to implement in-circuit debugging and In-Circuit Serial Programming™ (ICSP™).

## 32.10 MPLAB PM3 Device Programmer

The MPLAB PM3 Device Programmer is a universal, CE compliant device programmer with programmable voltage verification at $V_{DDMIN}$ and $V_{DDMAX}$ for maximum reliability. It features a large LCD display (128 x 64) for menus and error messages, and a modular, detachable socket assembly to support various package types. The ICSP cable assembly is included as a standard item. In Stand-Alone mode, the MPLAB PM3 Device Programmer can read, verify and program PIC devices without a PC connection. It can also set code protection in this mode. The MPLAB PM3 connects to the host PC via an RS-232 or USB cable. The MPLAB PM3 has high-speed communications and optimized algorithms for quick programming of large memory devices, and incorporates an MMC card for file storage and data applications.