**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 | 33MHz |
| Connectivity | I²C, SPI, UART/USART |
| Peripherals | Brown-out Detect/Reset, POR, PWM, WDT |
| Number of I/O | 50 |
| Program Memory Size | 32KB (16K x 16) |
| Program Memory Type | OTP |
| EEPROM Size | - |
| RAM Size | 902 x 8 |
| Voltage - Supply (Vcc/Vdd) | 4.5V ~ 5.5V |
| Data Converters | A/D 12x10b |
| Oscillator Type | External |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 68-LCC (J-Lead) |
| Supplier Device Package | 68-PLCC (24.23x24.23) |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/pic17c756a-33i-l |

**NOTES:**

#### 7.2.2.2 CPU Status Register (CPUSTA)

The CPUSTA register contains the status and control bits for the CPU. This register has a bit that is used to globally enable/disable interrupts. If only a specific interrupt is desired to be enabled/disabled, please refer to the Interrupt Status (INTSTA) register and the Peripheral Interrupt Enable (PIE) registers. The CPUSTA register also indicates if the stack is available and contains the Power-down ($\overline{PD}$) and Time-out ($\overline{TO}$) bits. The $\overline{TO}$, $\overline{PD}$, and STKAV bits are not writable. These bits are set and cleared according to device logic. Therefore, the result of an instruction with the CPUSTA register as destination may be different than intended.

The $\overline{POR}$ bit allows the differentiation between a Power-on Reset, external $\overline{MCLR}$ Reset, or a WDT Reset. The $\overline{BOR}$ bit indicates if a Brown-out Reset occurred.

| | |
|---|---|
| **Note 1:** | The $\overline{BOR}$ status bit is a don't care and is not necessarily predictable if the Brown-out circuit is disabled (when the BODEN bit in the Configuration word is programmed). |

**REGISTER 7-2: CPUSTA REGISTER (ADDRESS: 06h, UNBANKED)**

| U-0 | U-0 | R-1 | R/W-1 | R-1 | R-1 | R/W-0 | R/W-1 |
|---|---|---|---|---|---|---|---|
| — | — | STKAV | GLINTD | $\overline{TO}$ | $\overline{PD}$ | $\overline{POR}$ | $\overline{BOR}$ |
| bit 7 | | | | | | | bit 0 |

bit 7-6 **Unimplemented**: Read as '0'

bit 5 **STKAV**: Stack Available bit
This bit indicates that the 4-bit stack pointer value is Fh, or has rolled over from Fh → 0h (stack overflow).
1 = Stack is available
0 = Stack is full, or a stack overflow may have occurred (once this bit has been cleared by a stack overflow, only a device RESET will set this bit)

bit 4 **GLINTD**: Global Interrupt Disable bit
This bit disables all interrupts. When enabling interrupts, only the sources with their enable bits set can cause an interrupt.
1 = Disable all interrupts
0 = Enables all unmasked interrupts

bit 3 **$\overline{TO}$**: WDT Time-out Status bit
1 = After power-up, by a CLRWDT instruction, or by a SLEEP instruction
0 = A Watchdog Timer time-out occurred

bit 2 **$\overline{PD}$**: Power-down Status bit
1 = After power-up or by the CLRWDT instruction
0 = By execution of the SLEEP instruction

bit 1 **$\overline{POR}$**: Power-on Reset Status bit
1 = No Power-on Reset occurred
0 = A Power-on Reset occurred (must be set by software)

bit 0 **$\overline{BOR}$**: Brown-out Reset Status bit

When BODEN Configuration bit is set (enabled):
1 = No Brown-out Reset occurred
0 = A Brown-out Reset occurred (must be set by software)

When BODEN Configuration bit is clear (disabled):
Don't care

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

## 8.2 Table Writes to External Memory

Table writes to external memory are always two-cycle instructions. The second cycle writes the data to the external memory location. The sequence of events for an external memory write are the same for an internal write.

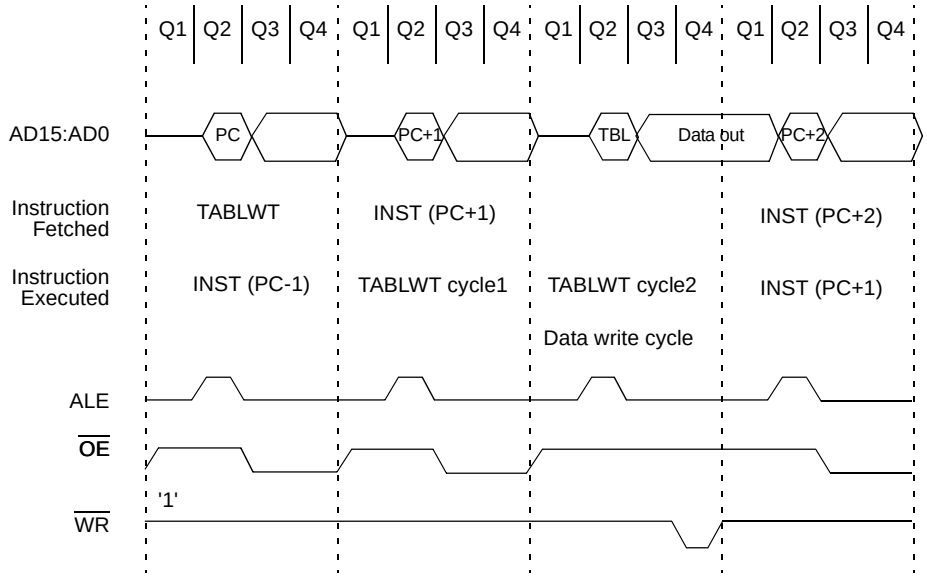
### 8.2.1 TABLE WRITE CODE

The "i" operand of the TABLWT instruction can specify that the value in the 16-bit TBLPTR register is automatically incremented (for the next write). In Example 8-1, the TBLPTR register is not automatically incremented.

**EXAMPLE 8-1: TABLE WRITE**

```
CLRWDT                  ; Clear WDT
MOVLW   HIGH (TBL_ADDR) ; Load the Table
MOVWF   TBLPTRH         ;   address
MOVLW   LOW (TBL_ADDR)  ;
MOVWF   TBLPTRL         ;
MOVLW   HIGH (DATA)     ; Load HI byte
TLWT    1, WREG         ;   in TABLATH
MOVLW   LOW (DATA)      ; Load LO byte
TABLWT  0,0,WREG        ;   in TABLATL
                        ;   and write to
                        ;   program memory
                        ;   (Ext. SRAM)
```

**FIGURE 8-5: TABLWT WRITE TIMING (EXTERNAL MEMORY)**




**Note:** If external write and GLINTD = '1' and Enable bit = '1', then when '1' → Flag bit, do table write. The highest pending interrupt is cleared.

## 10.4 PORTD and DDRD Registers

PORTD is an 8-bit bi-directional port. The corresponding data direction register is DDRD. A '1' in DDRD configures the corresponding port pin as an input. A '0' in the DDRD register configures the corresponding port pin as an output. Reading PORTD reads the status of the pins, whereas writing to PORTD will write to the port latch. PORTD is multiplexed with the system bus. When operating as the system bus, PORTD is the high order byte of the address/data bus (AD15:AD8). The timing for the system bus is shown in the Electrical Specifications section.

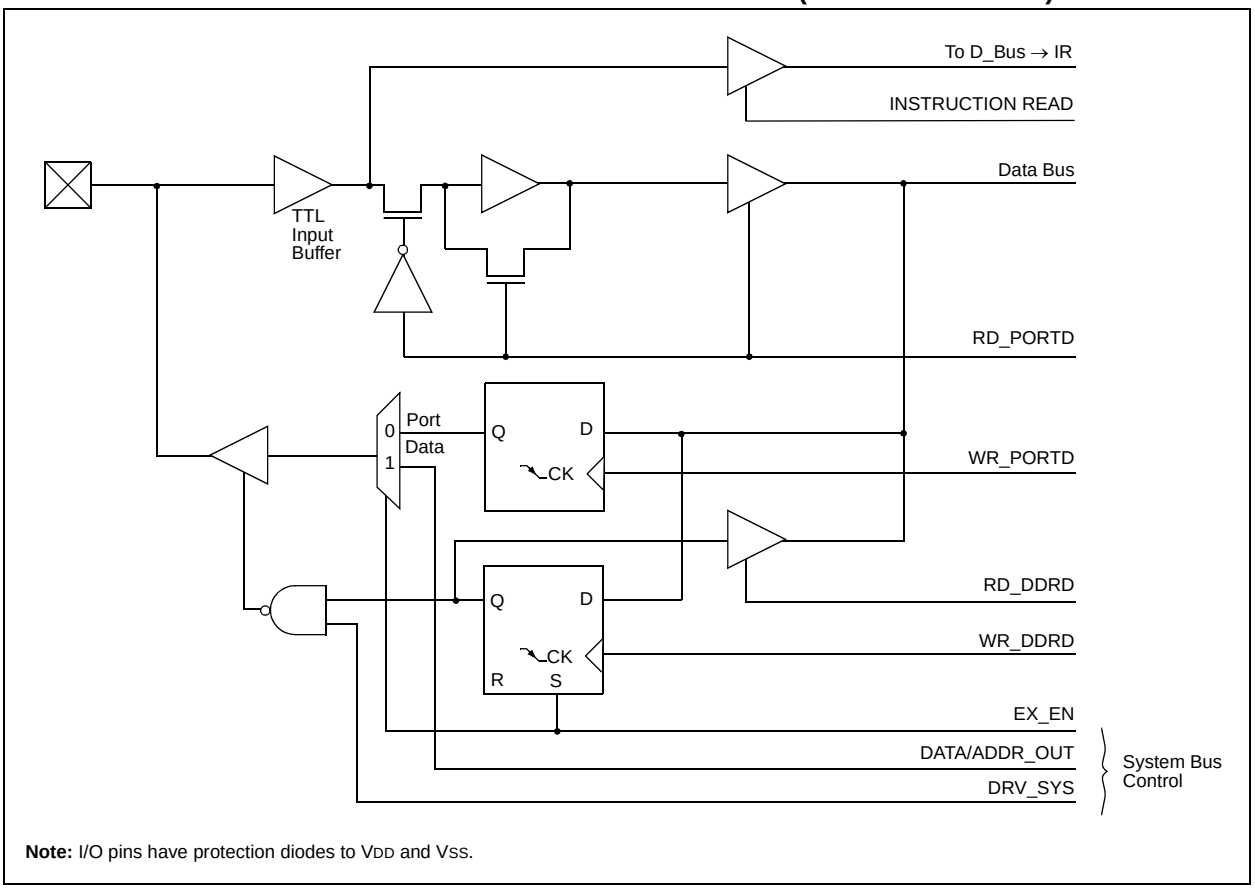
| **Note:** | This port is configured as the system bus when the device's configuration bits are selected to Microprocessor or Extended Microcontroller modes. In the two other microcontroller modes, this port is a general purpose I/O. |
|---|---|

Example 10-4 shows an instruction sequence to initialize PORTD. The Bank Select Register (BSR) must be selected to Bank 1 for the port to be initialized. The following example uses the `MOVLB` instruction to load the BSR register for bank selection.

**EXAMPLE 10-4: INITIALIZING PORTD**

```
MOVLB   1          ; Select Bank 1
CLRF    PORTD, F   ; Initialize PORTD data
                   ; latches before setting
                   ; the data direction reg
MOVLW    0xCF      ; Value used to initialize
                   ; data direction
MOVWF    DDRD      ; Set RD<3:0> as inputs
                   ; RD<5:4> as outputs
                   ; RD<7:6> as inputs
```

**FIGURE 10-10: BLOCK DIAGRAM OF RD7:RD0 PORT PINS (IN I/O PORT MODE)**




**Note:** I/O pins have protection diodes to VDD and VSS.

## 10.9 PORTJ and DDRJ Registers (PIC17C76X only)

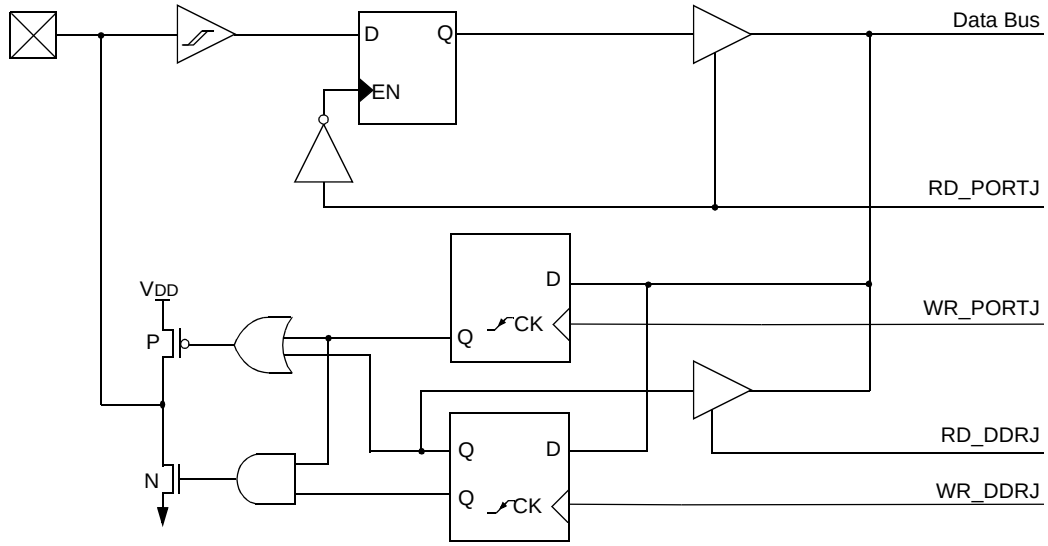
PORTJ is an 8-bit wide, bi-directional port. The corresponding data direction register is DDRJ. A '1' in DDRJ configures the corresponding port pin as an input. A '0' in the DDRJ register configures the corresponding port pin as an output. Reading PORTJ reads the status of the pins, whereas writing to PORTJ will write to the respective port latch.

PORTJ is a general purpose I/O port.

**EXAMPLE 10-9: INITIALIZING PORTJ**

```
MOVLB   8           ; Select Bank 8
CLRF    PORTJ, F    ; Initialize PORTJ data
                    ; latches before setting
                    ; the data direction
                    ; register
MOVLW    0xCF       ; Value used to initialize
                    ; data direction
MOVWF    DDRJ       ; Set RJ<3:0> as inputs
                    ; RJ<5:4> as outputs
                    ; RJ<7:6> as inputs
```

**FIGURE 10-19: PORTJ BLOCK DIAGRAM**




**Note:** I/O pins have protection diodes to VDD and VSS.

#### 13.1.3.1 PWM Periods

The period of the PWM1 output is determined by Timer1 and its period register (PR1). The period of the PWM2 and PWM3 outputs can be individually software configured to use either Timer1 or Timer2 as the time-base. For PWM2, when TM2PW2 bit (PW2DCL<5>) is clear, the time base is determined by TMR1 and PR1 and when TM2PW2 is set, the time base is determined by Timer2 and PR2. For PWM3, when TM2PW3 bit (PW3DCL<5>) is clear, the time base is determined by TMR1 and PR1, and when TM2PW3 is set, the time base is determined by Timer2 and PR2.

Running two different PWM outputs on two different timers allows different PWM periods. Running all PWMs from Timer1 allows the best use of resources by freeing Timer2 to operate as an 8-bit timer. Timer1 and Timer2 cannot be used as a 16-bit timer if any PWM is being used.

The PWM periods can be calculated as follows:

$$\text{period of PWM1} = [(PR1) + 1] \times 4T_{OSC}$$

$$\text{period of PWM2} = [(PR1) + 1] \times 4T_{OSC} \text{ or } [(PR2) + 1] \times 4T_{OSC}$$

$$\text{period of PWM3} = [(PR1) + 1] \times 4T_{OSC} \text{ or } [(PR2) + 1] \times 4T_{OSC}$$

The duty cycle of PWMx is determined by the 10-bit value DCx<9:0>. The upper 8-bits are from register PWxDCH and the lower 2-bits are from PWxDCL<7:6> (PWxDCH:PWxDCL<7:6>). Table 13-4 shows the maximum PWM frequency ($F_{PWM}$), given the value in the period register.

The number of bits of resolution that the PWM can achieve depends on the operation frequency of the device as well as the PWM frequency ($F_{PWM}$).

Maximum PWM resolution (bits) for a given PWM frequency:

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

where: $F_{PWM}$ = 1 / period of PWM

The PWMx duty cycle is as follows:

$$\text{PWMx Duty Cycle} = (DCx) \times T_{OSC}$$

where DCx represents the 10-bit value from PWxDCH:PWxDCL.

If DCx = 0, then the duty cycle is zero. If PRx = PWxDCH, then the PWM output will be low for one to four Q-clocks (depending on the state of the PWxDCL<7:6> bits). For a duty cycle to be 100%, the PWxDCH value must be greater then the PRx value.

The duty cycle registers for both PWM outputs are double buffered. When the user writes to these registers, they are stored in master latches. When TMR1 (or TMR2) overflows and a new PWM period begins, the master latch values are transferred to the slave latches and the PWMx pin is forced high.

| **Note:** | For PW1DCH, PW1DCL, PW2DCH, PW2DCL, PW3DCH and PW3DCL registers, a write operation writes to the "master latches", while a read operation reads the "slave latches". As a result, the user may not read back what was just written to the duty cycle registers (until transferred to slave latch). |
|---|---|

The user should also avoid any "read-modify-write" operations on the duty cycle registers, such as: `ADDWF PW1DCH`. This may cause duty cycle outputs that are unpredictable.

**TABLE 13-4: PWM FREQUENCY vs. RESOLUTION AT 33 MHz**

| PWM Frequency | Frequency (kHz) | | | | |
|---|---|---|---|---|---|
| | **32.2** | **64.5** | **90.66** | **128.9** | **515.6** |
| PRx Value | 0xFF | 0x7F | 0x5A | 0x3F | 0x0F |
| High Resolution | 10-bit | 9-bit | 8.5-bit | 8-bit | 6-bit |
| Standard Resolution | 8-bit | 7-bit | 6.5-bit | 6-bit | 4-bit |

#### 13.1.3.2 PWM INTERRUPTS

The PWM modules make use of the TMR1 and/or TMR2 interrupts. A timer interrupt is generated when TMR1 or TMR2 equals its period register and on the following increment is cleared to zero. This interrupt also marks the beginning of a PWM cycle. The user can write new duty cycle values before the timer rollover. The TMR1 interrupt is latched into the TMR1IF bit and the TMR2 interrupt is latched into the TMR2IF bit. These flags must be cleared in software.

**FIGURE 14-1: USART TRANSMIT**

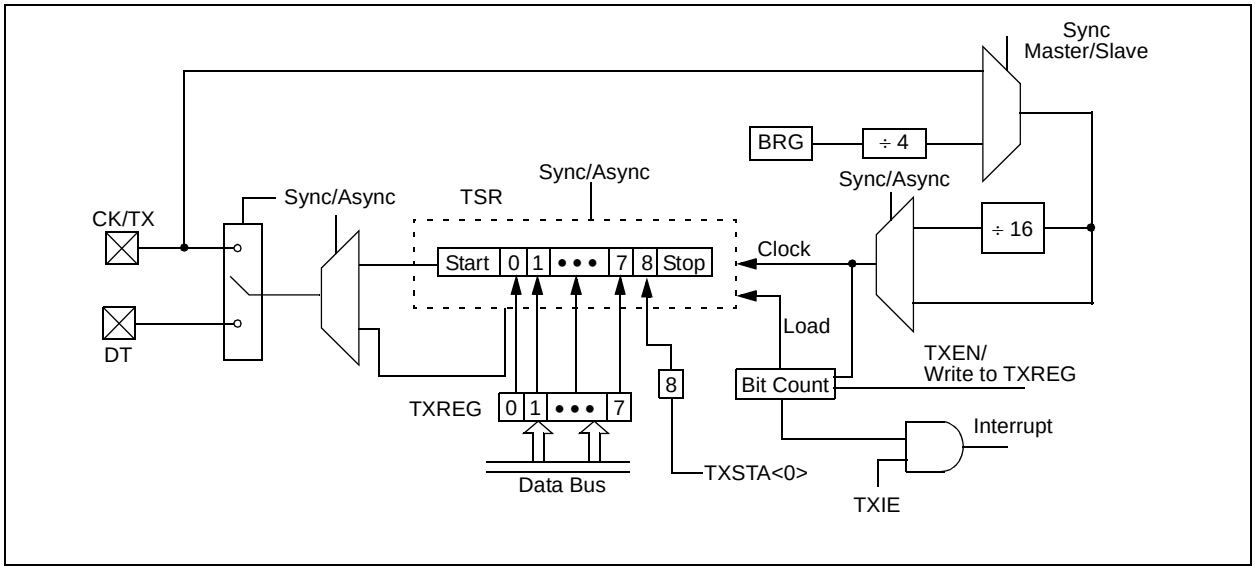



**FIGURE 14-2: USART RECEIVE**

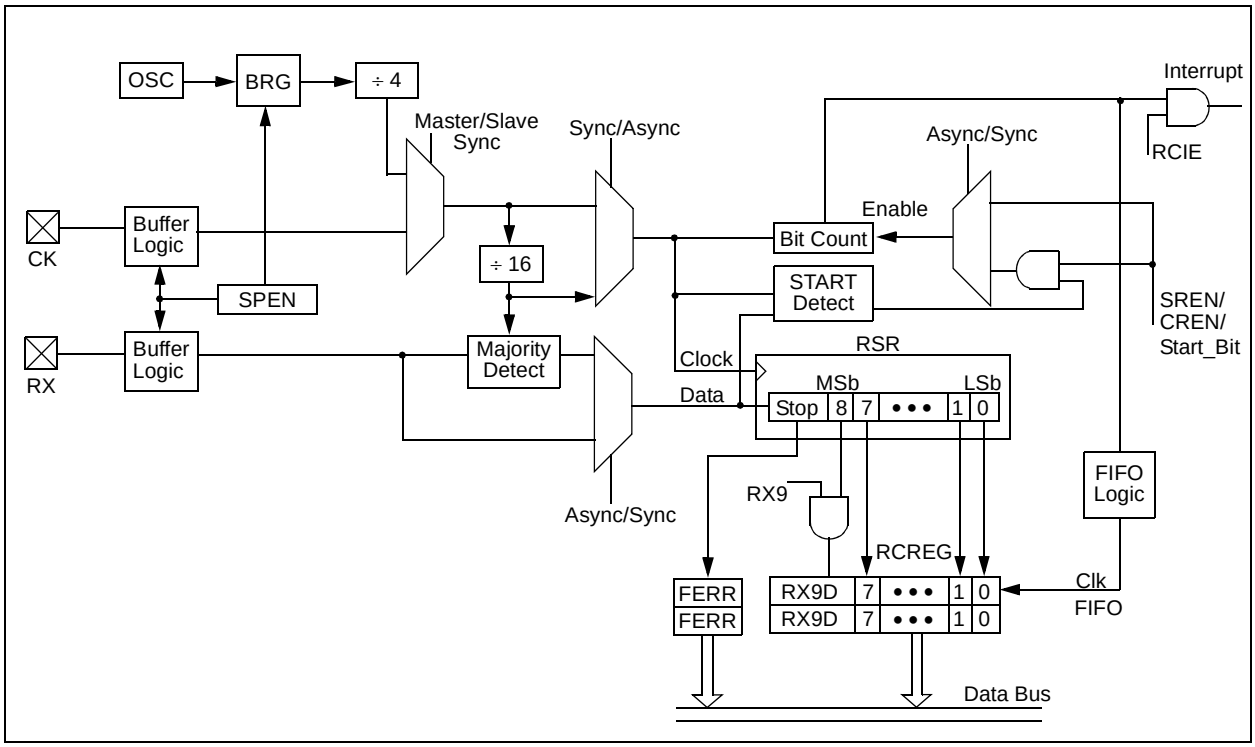


**FIGURE 15-8: SPI MODE WAVEFORM (SLAVE MODE WITH CKE = 0)**

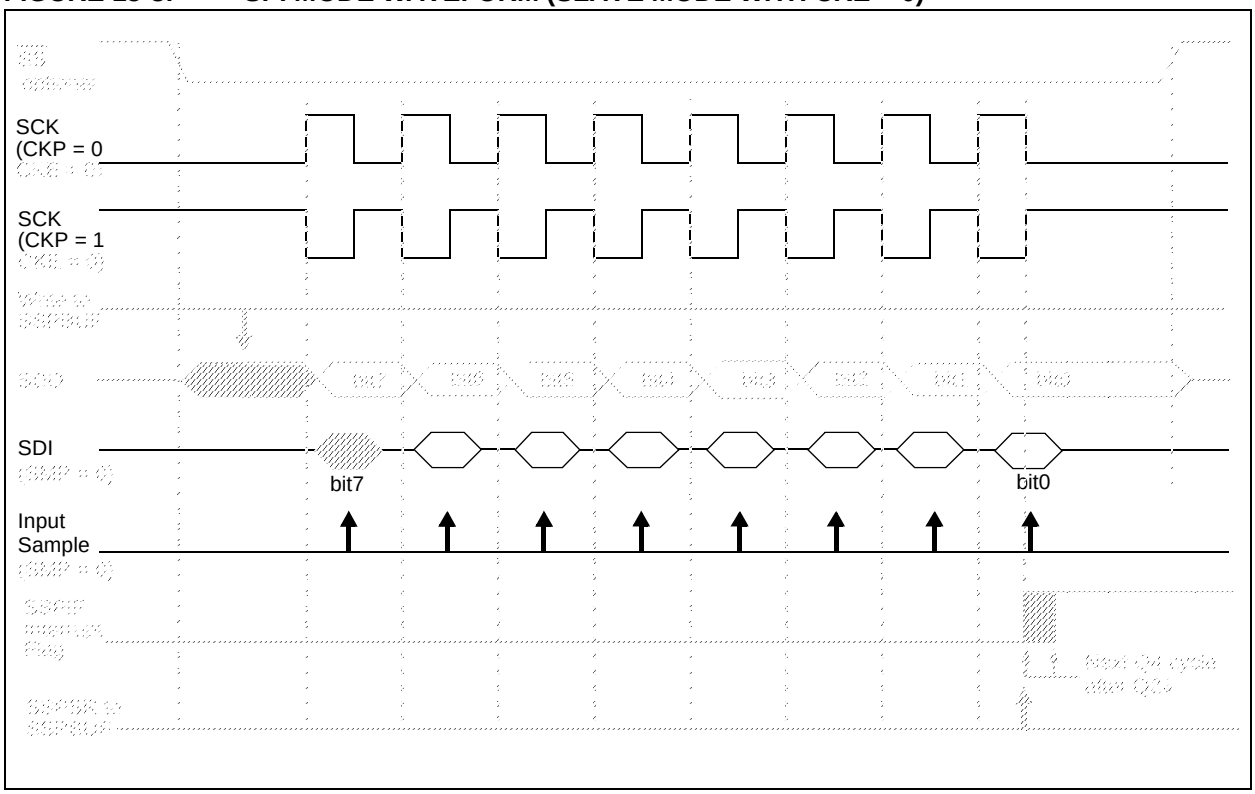
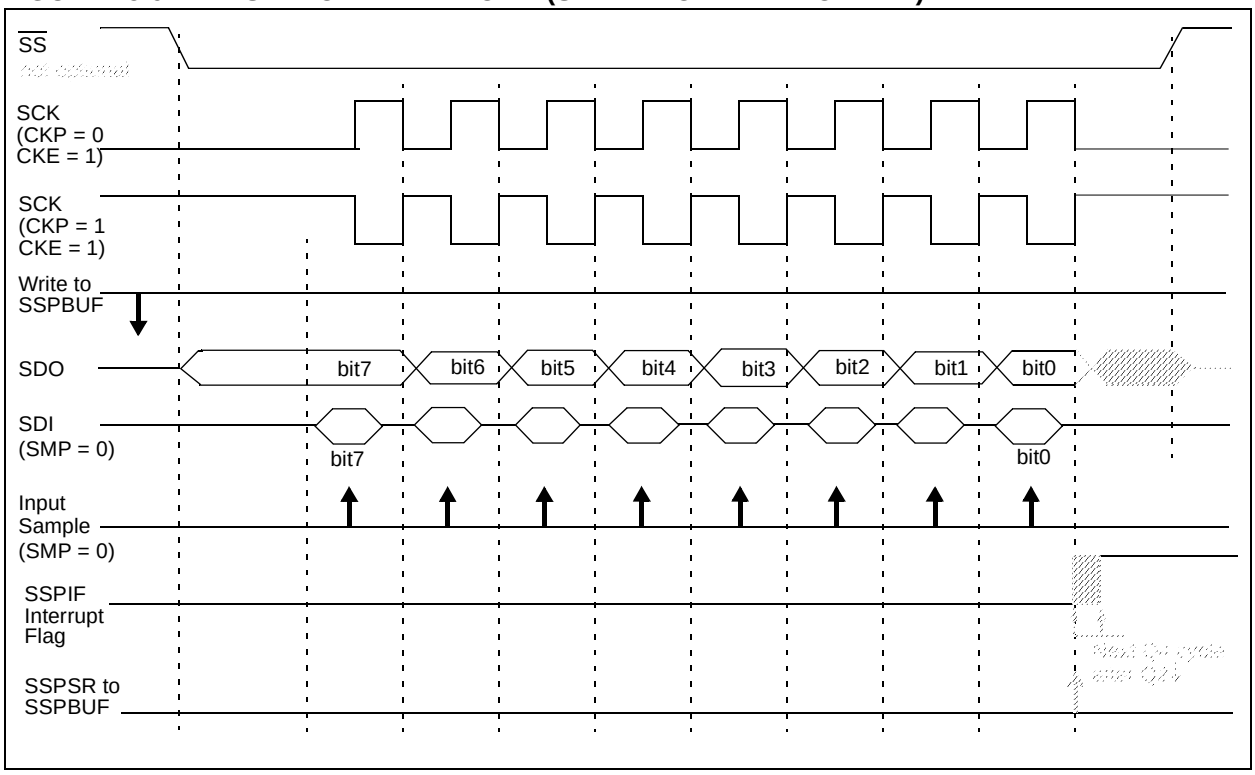



**FIGURE 15-9: SPI MODE WAVEFORM (SLAVE MODE WITH CKE = 1)**



### 15.2.1.3 Slave Transmission

When the R/$\overline{W}$ bit of the incoming address byte is set and an address match occurs, the R/$\overline{W}$ bit of the SSPSTAT register is set. The received address is loaded into the SSPBUF register. The $\overline{ACK}$ pulse will be sent on the ninth bit, and the SCL pin is held low. The transmit data must be loaded into the SSPBUF register, which also loads the SSPSR register. Then SCL pin should be enabled by setting bit CKP (SSPCON1<4>). The master must monitor the SCL pin prior to asserting another clock pulse. The slave devices may be holding off the master by stretching the clock. The eight data bits are shifted out on the falling edge of the SCL input. This ensures that the SDA signal is valid during the SCL high time (Figure 15-13).

An SSP interrupt is generated for each data transfer byte. The SSPIF flag bit must be cleared in software, and the SSPSTAT register is used to determine the status of the byte transfer. The SSPIF flag bit is set on the falling edge of the ninth clock pulse.

As a slave-transmitter, the $\overline{ACK}$ pulse from the master-receiver is latched on the rising edge of the ninth SCL input pulse. If the SDA line was high (not $\overline{ACK}$), then the data transfer is complete. When the not $\overline{ACK}$ is latched by the slave, the slave logic is reset and the slave then monitors for another occurrence of the START bit. If the SDA line was low ($\overline{ACK}$), the transmit data must be loaded into the SSPBUF register, which also loads the SSPSR register. Then, the SCL pin should be enabled by setting the CKP bit.

**FIGURE 15-12: I²C WAVEFORMS FOR RECEPTION (7-BIT ADDRESS)**

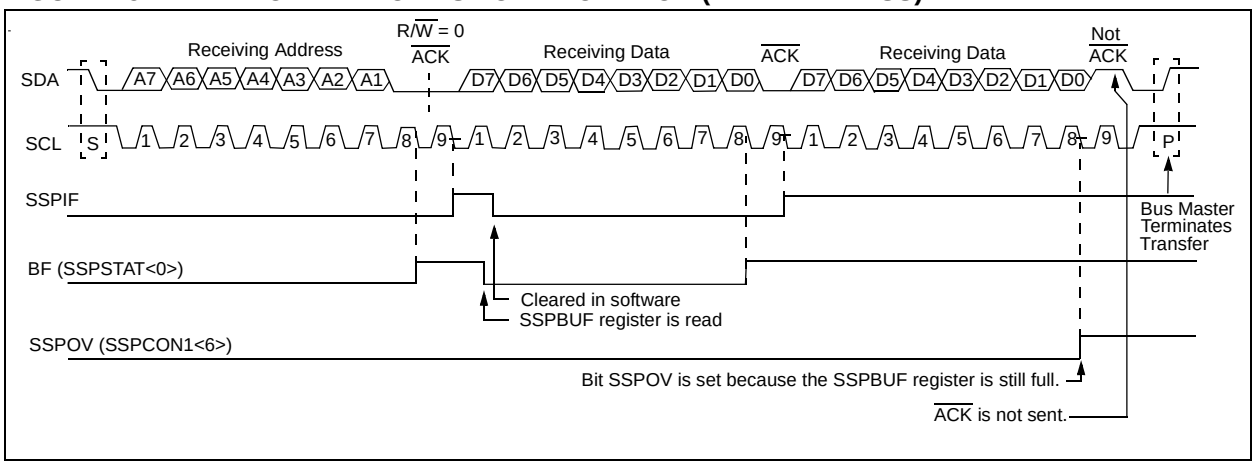
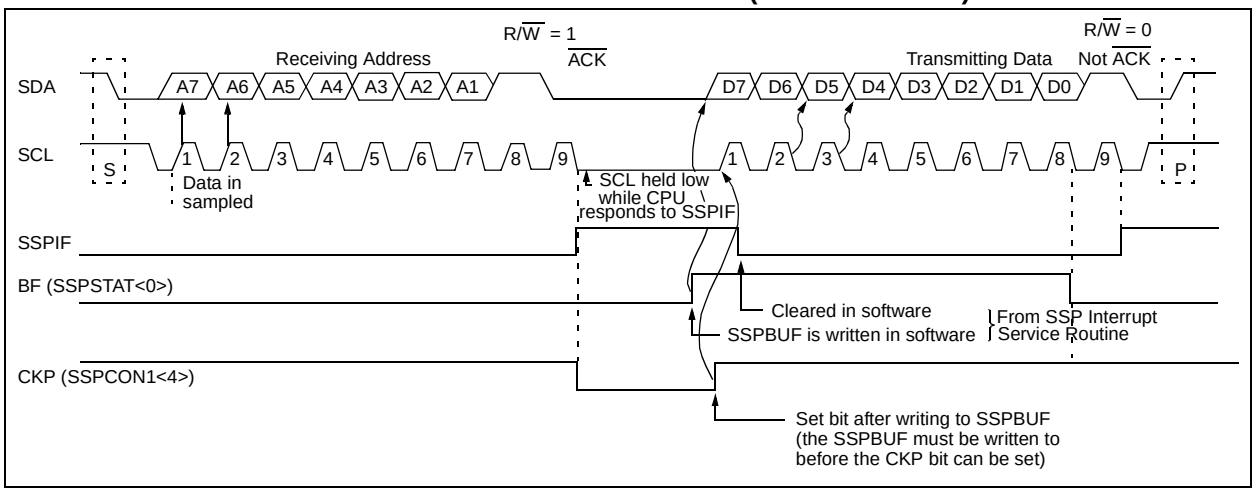



**FIGURE 15-13: I²C WAVEFORMS FOR TRANSMISSION (7-BIT ADDRESS)**



### 15.2.9 I²C MASTER MODE START CONDITION TIMING

To initiate a START condition, the user sets the START condition enable bit, SEN (SSPCON2<0>). If the SDA and SCL pins are sampled high, the baud rate generator is reloaded with the contents of SSPADD<6:0> and starts its count. If SCL and SDA are both sampled high when the baud rate generator times out (TBRG), the SDA pin is driven low. The action of the SDA being driven low while SCL is high is the START condition and causes the S bit (SSPSTAT<3>) to be set. Following this, the baud rate generator is reloaded with the contents of SSPADD<6:0> and resumes its count. When the baud rate generator times out (TBRG), the SEN bit (SSPCON2<0>) will be automatically cleared by hardware, the baud rate generator is suspended, leaving the SDA line held low and the START condition is complete.

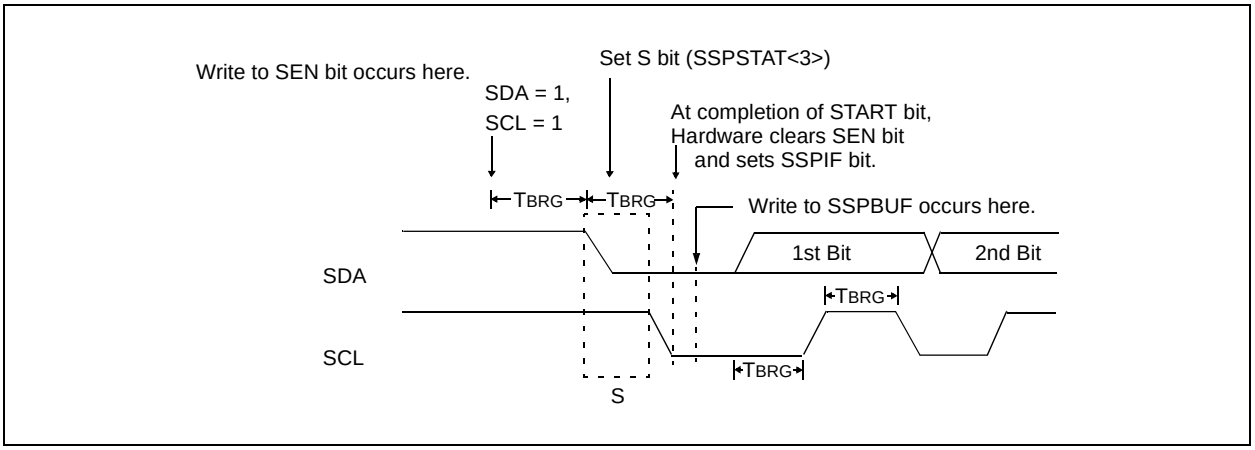
| **Note:** | If at the beginning of START condition, the SDA and SCL pins are already sampled low, or if during the START condition, the SCL line is sampled low before the SDA line is driven low, a bus collision occurs. The Bus Collision Interrupt Flag (BCLIF) is set, the START condition is aborted and the I²C module is reset into its IDLE state. |
|---|---|

#### 15.2.9.1 WCOL Status Flag

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

| **Note:** | Because queueing of events is not allowed, writing to the lower 5 bits of SSPCON2 is disabled until the START condition is complete. |
|---|---|

**FIGURE 15-20: FIRST START BIT TIMING**



### 15.2.12 $I^2C$ MASTER MODE RECEPTION

Master mode reception is enabled by programming the receive enable bit, RCEN (SSPCON2<3>).

| | |
|---|---|
| **Note:** | The SSP Module must be in an IDLE STATE before the RCEN bit is set, or the RCEN bit will be disregarded. |

The baud rate generator begins counting and on each rollover, the state of the SCL pin changes (high to low/ low to high) and data is shifted into the SSPSR. After the falling edge of the eighth clock, the receive enable flag is automatically cleared, the contents of the SSPSR are loaded into the SSPBUF, the BF flag is set, the SSPIF is set and the baud rate generator is suspended from counting, holding SCL low. The SSP is now in IDLE state, awaiting the next command. When the buffer is read by the CPU, the BF flag is automatically cleared. The user can then send an acknowledge bit at the end of reception, by setting the acknowledge sequence enable bit, ACKEN (SSPCON2<4>).

#### 15.2.12.1 BF Status Flag

In receive operation, BF is set when an address or data byte is loaded into SSPBUF from SSPSR. It is cleared when SSPBUF is read.

#### 15.2.12.2 SSPOV Status Flag

In receive operation, SSPOV is set when 8 bits are received into the SSPSR, and the BF flag is already set from a previous reception.

#### 15.2.12.3 WCOL Status Flag

If the user writes the SSPBUF when a receive is already in progress (i.e., SSPSR is still shifting in a data byte), then WCOL is set and the contents of the buffer are unchanged (the write doesn't occur).

**EXAMPLE 15-2: INTERFACING TO A 24LC01B SERIAL EEPROM (USING MPLAB C17)**

```
void ACKPoll(void)
{
        StartI2C();                     // Send start bit
        IdleI2C();                      // Wait for idle condition
        WriteI2C(CONTROL);              // Send control byte
        IdleI2C();                      // Wait for idle condition
        // Poll the ACK bit coming from the 24LC01B
        // Loop as long as the 24LC01B NACKs
        while (SSPCON2bits.ACKSTAT)
        {
                RestartI2C();           // Send a restart bit
                IdleI2C();              // Wait for idle condition
                WriteI2C(CONTROL);      // Send control byte
                IdleI2C();              // Wait for idle condition
        }
        IdleI2C();                      // Wait for idle condition
        StopI2C();                      // Send stop bit
        IdleI2C();                      // Wait for idle condition
        return;
}
```

## 17.4 Power-down Mode (SLEEP)

The Power-down mode is entered by executing a SLEEP instruction. This clears the Watchdog Timer and postscaler (if enabled). The $\overline{PD}$ bit is cleared and the $\overline{TO}$ bit is set (in the CPUSTA register). In SLEEP mode, the oscillator driver is turned off. The I/O ports maintain their status (driving high,low, or hi-impedance input).

The $\overline{MCLR}/V_{PP}$ pin must be at a logic high level ($V_{IHMC}$). A WDT time-out RESET does not drive the $\overline{MCLR}/V_{PP}$ pin low.

### 17.4.1 WAKE-UP FROM SLEEP

The device can wake-up from SLEEP through one of the following events:

- Power-on Reset
- Brown-out Reset
- External RESET input on $\overline{MCLR}/V_{PP}$ pin
- WDT Reset (if WDT was enabled)
- Interrupt from RA0/INT pin, RB port change, T0CKI interrupt, or some peripheral interrupts

The following peripheral interrupts can wake the device from SLEEP:

- Capture interrupts
- USART synchronous slave transmit interrupts
- USART synchronous slave receive interrupts
- A/D conversion complete
- SPI slave transmit/receive complete
- $I^2C$ slave receive

Other peripherals cannot generate interrupts since during SLEEP, no on-chip Q clocks are present.

Any RESET event will cause a device RESET. Any interrupt event is considered a continuation of program execution. The $\overline{TO}$ and $\overline{PD}$ bits in the CPUSTA register can be used to determine the cause of a device RESET. The $\overline{PD}$ bit, which is set on power-up, is cleared when SLEEP is invoked. The $\overline{TO}$ bit is cleared if WDT time-out occurred (and caused a RESET).

When the SLEEP instruction is being executed, the next instruction (PC + 1) is pre-fetched. For the device to wake-up through an interrupt event, the corresponding interrupt enable bit must be set (enabled). Wake-up is regardless of the state of the GLINTD bit. If the GLINTD bit is set (disabled), the device continues execution at the instruction after the SLEEP instruction. If the GLINTD bit is clear (enabled), the device executes the instruction after the SLEEP instruction and then branches to the interrupt vector address. In cases where the execution of the instruction following SLEEP is not desirable, the user should have a NOP after the SLEEP instruction.

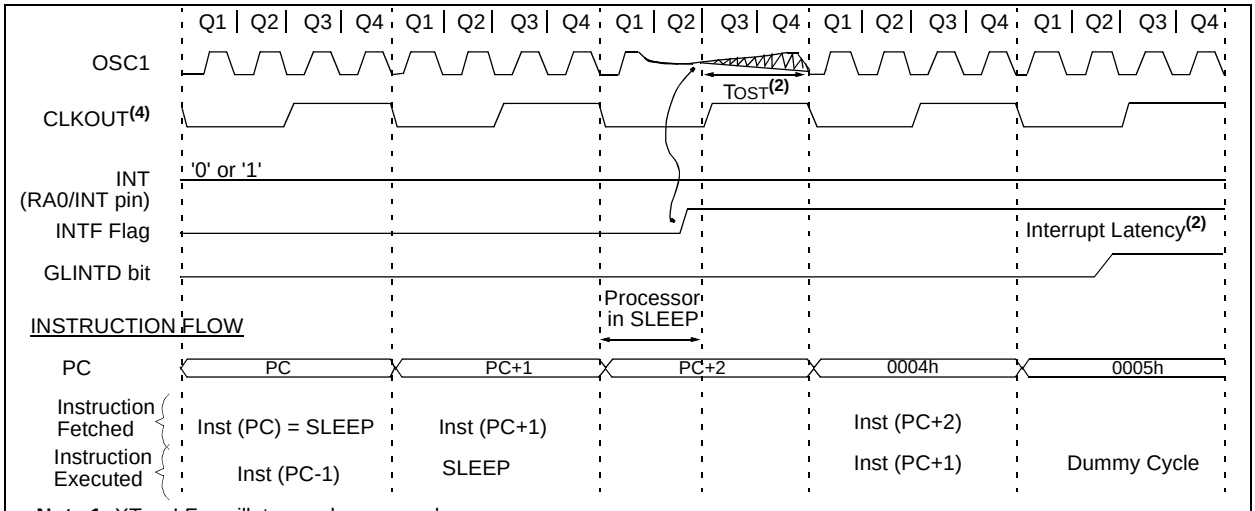
| **Note:** | If the global interrupt is disabled (GLINTD is set), but any interrupt source has both its interrupt enable bit and the corresponding interrupt flag bit set, the device will immediately wake-up from SLEEP. The $\overline{TO}$ bit is set and the $\overline{PD}$ bit is cleared. |
|---|---|

The WDT is cleared when the device wakes from SLEEP, regardless of the source of wake-up.

#### 17.4.1.1 Wake-up Delay

When the oscillator type is configured in XT or LF mode, the Oscillator Start-up Timer (OST) is activated on wake-up. The OST will keep the device in RESET for 1024TOSC. This needs to be taken into account when considering the interrupt response time when coming out of SLEEP.

**FIGURE 17-2: WAKE-UP FROM SLEEP THROUGH INTERRUPT**




**Note 1:** XT or LF oscillator mode assumed.
**2:** TOST = 1024TOSC (drawing not to scale). This delay will not be there for RC osc mode.
**3:** When GLINTD = 0, processor jumps to interrupt routine after wake-up. If GLINTD = 1, execution will continue in line.
**4:** CLKOUT is not available in these osc modes, but shown here for timing reference.

**RETURN** **Return from Subroutine**

Syntax: [ *label* ] RETURN

Operands: None

Operation: TOS → PC;

Status Affected: None

Encoding:

| | | | |
|---|---|---|---|
| 0000 | 0000 | 0000 | 0010 |

Description: Return from subroutine. The stack is popped and the top of the stack (TOS) is loaded into the program counter.

Words: 1

Cycles: 2

Q Cycle Activity:

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

Example: `RETURN`

After Interrupt
PC = TOS

**RLCF** **Rotate Left f through Carry**

Syntax: [ *label* ] RLCF f,d

Operands: $0 \le f \le 255$
$d \in [0,1]$

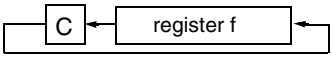
Operation: f<n> → d<n+1>;
f<7> → C;
C → d<0>

Status Affected: C

Encoding:

| | | | |
|---|---|---|---|
| 0001 | 101d | ffff | ffff |

Description: The contents of register 'f' are rotated one bit to the left through the Carry Flag. If 'd' is 0, the result is placed in WREG. If 'd' is 1, the result is stored back in register 'f'.




Words: 1

Cycles: 1

Q Cycle Activity:

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

Example: `RLCF REG,0`

Before Instruction
REG = 1110 0110
C = 0

After Instruction
REG = 1110 0110
WREG = 1100 1100
C = 1

**TLWT** **Table Latch Write**

Syntax: [ *label* ] TLWT t,f

Operands: $0 \le f \le 255$
$t \in [0,1]$

Operation: If t = 0,
f → TBLATL;
If t = 1,
f → TBLATH

Status Affected: None

Encoding:

| | | | |
|---|---|---|---|
| 1010 | 01tx | ffff | ffff |

Description: Data from file register 'f' is written into the 16-bit table latch (TBLAT).
If t = 1; high byte is written
If t = 0; low byte is written
This instruction is used in conjunction with TABLWT to transfer data from data memory to program memory.

Words: 1

Cycles: 1

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|
| Decode | Read register 'f' | Process Data | Write register TBLATH or TBLATL |

Example: TLWT t, RAM

Before Instruction
t = 0
RAM = 0xB7
TBLAT = 0x0000 (TBLATH = 0x00) (TBLATL = 0x00)

After Instruction
RAM = 0xB7
TBLAT = 0x00B7 (TBLATH = 0x00) (TBLATL = 0xB7)

Before Instruction
t = 1
RAM = 0xB7
TBLAT = 0x0000 (TBLATH = 0x00) (TBLATL = 0x00)

After Instruction
RAM = 0xB7
TBLAT = 0xB700 (TBLATH = 0xB7) (TBLATL = 0x00)

**TSTFSZ** **Test f, skip if 0**

Syntax: [ *label* ] TSTFSZ f

Operands: $0 \le f \le 255$

Operation: skip if f = 0

Status Affected: None

Encoding:

| | | | |
|---|---|---|---|
| 0011 | 0011 | ffff | ffff |

Description: If 'f' = 0, the next instruction, fetched during the current instruction execution, is discarded and a NOP is executed, making this a two-cycle instruction.

Words: 1

Cycles: 1 (2)

Q Cycle Activity:

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

If skip:

| Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|
| No operation | No operation | No operation | No operation |

Example:
```
HERE    TSTFSZ  CNT
NZERO     :
ZERO      :
```

Before Instruction
PC = Address (HERE)

After Instruction
If CNT = 0x00,
PC = Address (ZERO)
If CNT ¼ 0x00,
PC = Address (NZERO)

## 20.0 PIC17C7XX ELECTRICAL CHARACTERISTICS

**Absolute Maximum Ratings †**

Ambient temperature under bias ..... -55°C to +125°C
Storage temperature ..... -65°C to +150°C
Voltage on $V_{DD}$ with respect to $V_{SS}$ ..... 0 V to +7.5 V
Voltage on $\overline{MCLR}$ with respect to $V_{SS}$ **(Note 2)** ..... -0.3 V to +14 V
Voltage on RA2 and RA3 with respect to $V_{SS}$ ..... -0.3 V to +8.5 V
Voltage on all other pins with respect to $V_{SS}$ ..... -0.3 V to $V_{DD}$ + 0.3 V
Total power dissipation **(Note 1)** ..... 1.0 W
Maximum current out of $V_{SS}$ pin(s) - total (@ 70°C) ..... 500 mA
Maximum current into $V_{DD}$ pin(s) - total (@ 70°C) ..... 500 mA
Input clamp current, $I_{IK}$ ($V_I$ < 0 or $V_I$ > $V_{DD}$) ..... ±20 mA
Output clamp current, $I_{OK}$ ($V_O$ < 0 or $V_O$ > $V_{DD}$) ..... ±20 mA
Maximum output current sunk by any I/O pin (except RA2 and RA3) ..... 35 mA
Maximum output current sunk by RA2 or RA3 pins ..... 60 mA
Maximum output current sourced by any I/O pin ..... 20 mA
Maximum current sunk by PORTA and PORTB (combined) ..... 150 mA
Maximum current sourced by PORTA and PORTB (combined) ..... 100 mA
Maximum current sunk by PORTC, PORTD and PORTE (combined) ..... 150 mA
Maximum current sourced by PORTC, PORTD and PORTE (combined) ..... 100 mA
Maximum current sunk by PORTF and PORTG (combined) ..... 150 mA
Maximum current sourced by PORTF and PORTG (combined) ..... 100 mA
Maximum current sunk by PORTH and PORTJ (combined) ..... 150 mA
Maximum current sourced by PORTH and PORTJ (combined) ..... 100 mA

**Note 1:** Power dissipation is calculated as follows: $P_{dis} = V_{DD} \times \{I_{DD} - \sum I_{OH}\} + \sum \{(V_{DD}-V_{OH}) \times I_{OH}\} + \sum(V_{OL} \times I_{OL})$

**2:** Voltage spikes below $V_{SS}$ at the $\overline{MCLR}$ pin, inducing currents greater than 80 mA, may cause latch-up. Thus, a series resistor of 50-100 Ω should be used when applying a "low" level to the $\overline{MCLR}$ pin, rather than pulling this pin directly to $V_{SS}$.

† NOTICE: 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 to maximum rating conditions for extended periods may affect device reliability.

**FIGURE 20-13: SPI MASTER MODE TIMING (CKE = 0)**

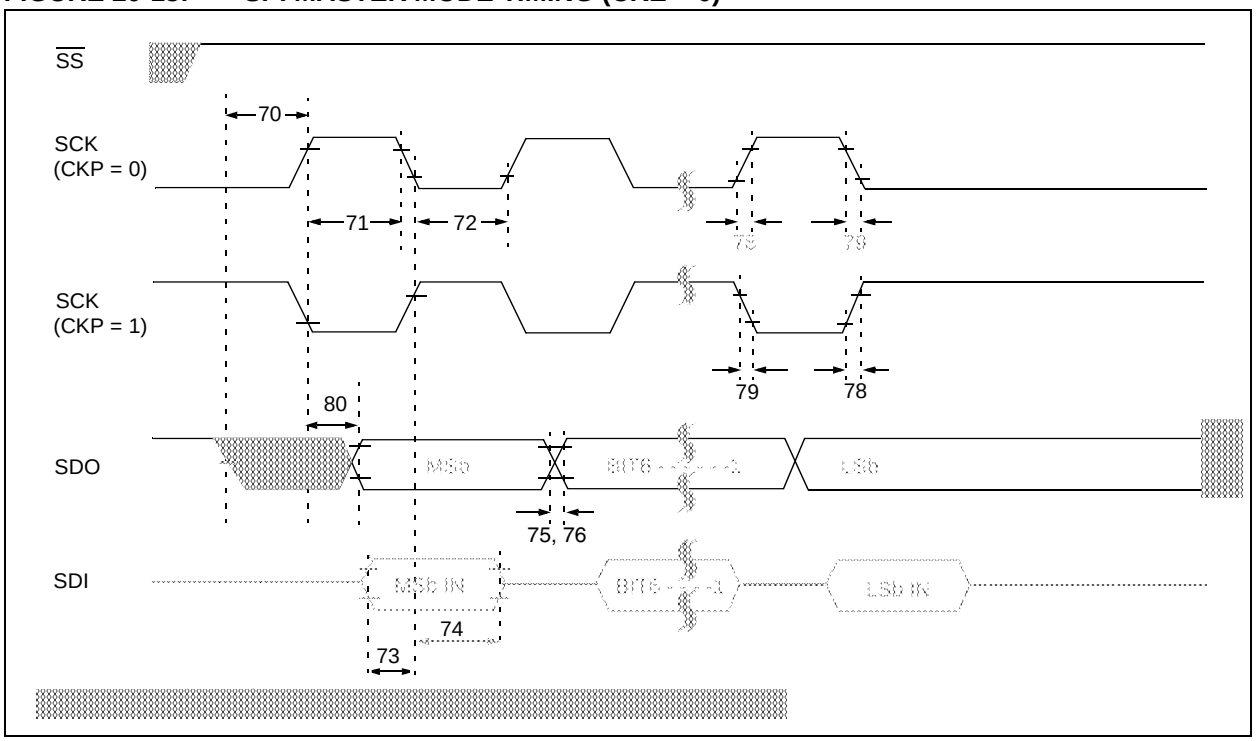


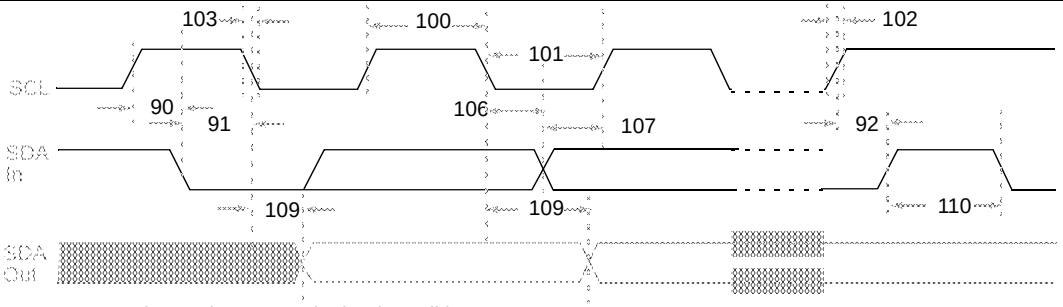

**TABLE 20-8: SPI MODE REQUIREMENTS (MASTER MODE, CKE = 0)**

| Param. No. | Symbol | Characteristic | | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|---|
| 70 | TssL2scH, TssL2scL | $\overline{SS}\downarrow$ to SCK$\downarrow$ or SCK$\uparrow$ input | | Tcy | — | — | ns | |
| 71 | TscH | SCK input high time (Slave mode) | Continuous | 1.25TCY + 30 | — | — | ns | |
| 71A | | | Single Byte | 40 | — | — | ns | **(Note 1)** |
| 72 | TscL | SCK input low time (Slave mode) | Continuous | 1.25TCY + 30 | — | — | ns | |
| 72A | | | Single Byte | 40 | — | — | ns | **(Note 1)** |
| 73 | TdiV2scH, TdiV2scL | Setup time of SDI data input to SCK edge | | 100 | — | — | ns | |
| 73A | TB2B | Last clock edge of Byte1 to the 1st clock edge of Byte2 | | 1.5TCY + 40 | — | — | ns | **(Note 1)** |
| 74 | TscH2diL, TscL2diL | Hold time of SDI data input to SCK edge | | 100 | — | — | ns | |
| 75 | TdoR | SDO data output rise time | | — | 10 | 25 | ns | |
| 76 | TdoF | SDO data output fall time | | — | 10 | 25 | ns | |
| 78 | TscR | SCK output rise time (Master mode) | | — | 10 | 25 | ns | |
| 79 | TscF | SCK output fall time (Master mode) | | — | 10 | 25 | ns | |
| 80 | TscH2doV, TscL2doV | SDO data output valid after SCK edge | | — | — | 50 | ns | |

† Data in "Typ" column is at 5V, 25°C unless otherwise stated.

**Note 1:** Specification 73A is only required if specifications 71A and 72A are used.

**FIGURE 20-18: I²C BUS DATA TIMING**




**Note:** Refer to Figure 20-5 for load conditions.

**TABLE 20-13: I²C BUS DATA REQUIREMENTS**

| Param No. | Sym | Characteristic | | Min | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| 100 | Thigh | Clock high time | 100 kHz mode | 2(Tosc)(BRG + 1) | — | ms | |
| | | | 400 kHz mode | 2(Tosc)(BRG + 1) | — | ms | |
| | | | 1 MHz mode(1) | 2(Tosc)(BRG + 1) | — | ms | |
| 101 | Tlow | Clock low time | 100 kHz mode | 2(Tosc)(BRG + 1) | — | ms | |
| | | | 400 kHz mode | 2(Tosc)(BRG + 1) | — | ms | |
| | | | 1 MHz mode(1) | 2(Tosc)(BRG + 1) | — | ms | |
| 102 | Tr | SDA and SCL rise time | 100 kHz mode | — | 1000 | ns | Cb is specified to be from 10 to 400 pF |
| | | | 400 kHz mode | 20 + 0.1Cb | 300 | ns | |
| | | | 1 MHz mode(1) | — | 300 | ns | |
| 103 | Tf | SDA and SCL fall time | 100 kHz mode | — | 300 | ns | Cb is specified to be from 10 to 400 pF |
| | | | 400 kHz mode | 20 + 0.1Cb | 300 | ns | |
| | | | 1 MHz mode(1) | — | 10 | ns | |
| 90 | Tsu:sta | START condition setup time | 100 kHz mode | 2(Tosc)(BRG + 1) | — | ms | Only relevant for Repeated Start condition |
| | | | 400 kHz mode | 2(Tosc)(BRG + 1) | — | ms | |
| | | | 1 MHz mode(1) | 2(Tosc)(BRG + 1) | — | ms | |
| 91 | Thd:sta | START condition hold time | 100 kHz mode | 2(Tosc)(BRG + 1) | — | ms | After this period, the first clock pulse is generated |
| | | | 400 kHz mode | 2(Tosc)(BRG + 1) | — | ms | |
| | | | 1 MHz mode(1) | 2(Tosc)(BRG + 1) | — | ms | |
| 106 | Thd:dat | Data input hold time | 100 kHz mode | 0 | — | ns | |
| | | | 400 kHz mode | 0 | 0.9 | ms | |
| | | | 1 MHz mode(1) | 0 | — | ns | |
| 107 | Tsu:dat | Data input setup time | 100 kHz mode | 250 | — | ns | **(Note 2)** |
| | | | 400 kHz mode | 100 | — | ns | |
| | | | 1 MHz mode(1) | 100 | — | ns | |
| 92 | Tsu:sto | STOP condition setup time | 100 kHz mode | 2(Tosc)(BRG + 1) | — | ms | |
| | | | 400 kHz mode | 2(Tosc)(BRG + 1) | — | ms | |
| | | | 1 MHz mode(1) | 2(Tosc)(BRG + 1) | — | ms | |
| 109 | Taa | Output valid from clock | 100 kHz mode | — | 3500 | ns | |
| | | | 400 kHz mode | — | 1000 | ns | |
| | | | 1 MHz mode(1) | — | 400 | ns | |

**Note 1:** Maximum pin capacitance = 10 pF for all I²C pins.

**2:** A fast mode (400 KHz) I²C bus device can be used in a standard mode I²C bus system, but the parameter # 107 ≥ 250 ns must then be met. This will automatically be the case if the device does not stretch the LOW period of the SCL signal. If such a device does stretch the LOW period of the SCL signal, it must output the next data bit to the SDA line. Parameter #102 + #107 = 1000 + 250 = 1250 ns (for 100 kHz mode) before the SCL line is released.

**3:** $C_b$ is specified to be from 10-400pF. The minimum specifications are characterized with $C_b$=10pF. The rise time spec ($t_r$) is characterized with $R_p$=$R_p$ min. The minimum fall time specification ($t_f$) is characterized with $C_b$=10pF,and $R_p$=$R_p$ max. These are only valid for fast mode operation ($V_{DD}$=4.5-5.5V) and where the SPM bit (SSPSTAT<7>) =1.)

**4:** Max specifications for these parameters are valid for falling edge only. Specs are characterized with $R_p$=$R_p$ min and $C_b$=400pF for standard mode, 200pF for fast mode, and 10pF for 1MHz mode.

**FIGURE 20-21: USART ASYNCHRONOUS MODE START BIT DETECT**

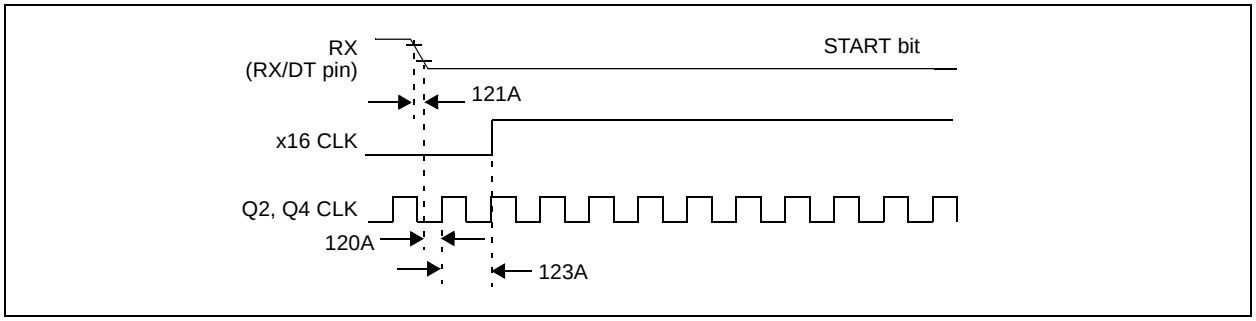



**TABLE 20-16: USART ASYNCHRONOUS MODE START BIT DETECT REQUIREMENTS**

| Param No. | Sym | Characteristic | | Min | Typ | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|---|
| 120A | TdtL2ckH | Time to ensure that the RX pin is sampled low | | — | — | TCY | ns | |
| 121A | TdtRF | Data rise time and fall time | Receive | — | — | **(Note 1)** | ns | |
| | | | Transmit | — | — | 40 | ns | |
| 123A | TckH2bckL | Time from RX pin sampled low to first rising edge of x16 clock | | — | — | TCY | ns | |

**Note 1:** Schmitt trigger will determine logic level.

**FIGURE 20-22: USART ASYNCHRONOUS RECEIVE SAMPLING WAVEFORM**

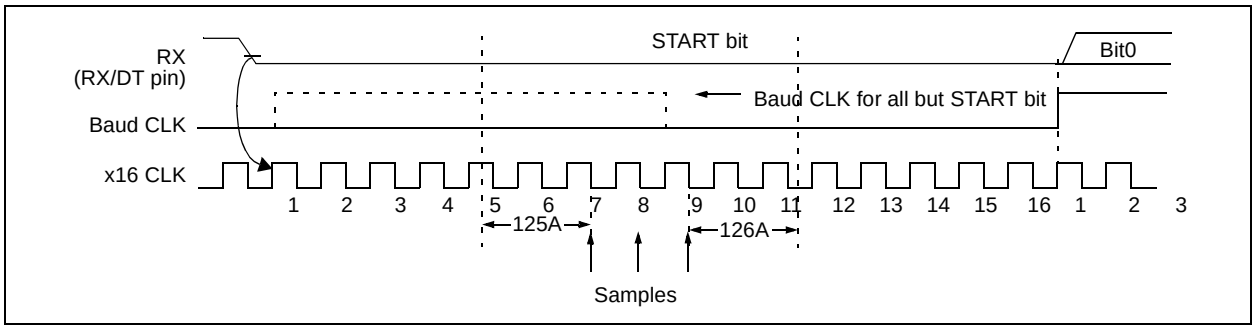



**TABLE 20-17: USART ASYNCHRONOUS RECEIVE SAMPLING REQUIREMENTS**

| Param No. | Sym | Characteristic | Min | Typ | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| 125A | TdtL2ckH | Setup time of RX pin to first data sampled | TCY | — | — | ns | |
| 126A | TdtL2ckH | Hold time of RX pin from last data sampled | TCY | — | — | ns | |