Serial I/O Port 7

CHAPTER 7
SERIAL I/O PORT

The serial I/O port on the 8XC196KC/KD is an asynchronous/synchronous port that includes a Universal Asynchronous Receiver and Transmitter (UART). The UART has one synchronous mode (Mode 0) and three asynchronous modes (Modes 1, 2, and 3) for both transmission and reception. This chapter provides an overview of each mode and describes how to program the serial port.

To aid in understanding the discussions of each mode, the following paragraphs briefly describe the registers associated with the serial I/O port.

The SP_CON register selects the communications mode and enables or disables the receiver, even parity checking, and nine-bit data transmission. (See "Programming the Serial Port" on page 7-6.) The SP_STAT register contains the Receive Interrupt (RI) and Transmit Interrupt (TI) flags and three other status bits. (See "Serial Port Status" on page 7-10.) Note that reading the SP_STAT register clears all flags except TXE. For this reason, we recommend that you copy the contents of the SP_STAT register into a shadow register and then execute bit-test instructions such as JBC and JBS on the shadow register. Otherwise, if more than one bit-test instruction is executed, false values may be returned.

The serial port receives data into the SBUF (RX) register; it transmits data from the port through the SBUF (TX) register. The transmit and receive registers are accessed via separate *horizontal windows* (see Chapter 4, "Memory Partitions"), permitting simultaneous reads and writes to both. The transmitter and receiver are buffered to support continuous transmissions and to allow reception of a second byte before the first byte has been read.

The serial port interrupts are enabled and disabled through the Interrupt Mask register (INT_MASK or INT_MASK1); pending interrupts are indicated by the Interrupt Pending register (INT_PEND or INT_PEND1). (See "Enabling the Serial Port Interrupts" on page 7-8.)

An independent baud-rate generator controls the baud rate of the serial port. Either XTAL1 or T2CLK can provide the clock signal. The BAUD_RATE register selects the baud rate and the clock source. (See "Programming the Baud Rate and Clock Source" on page 7-8.)

Refer to Appendix C for details about the registers and to Appendix B for additional information about the signals discussed in this chapter.

7.1. MODE 0

The most common use of Mode 0, the synchronous mode, is to expand the I/O capability of the 8XC196KC/KD with shift registers (see Figure 7-1). In this mode, the TXD pin outputs a set of eight clock pulses, while the RXD pin either transmits or receives data. Data is transferred eight bits at a time with the least-significant bit first. Figure 7-2 shows a diagram of the relative timing of these signals. Note that only Mode 0 uses RXD as an open-drain output.

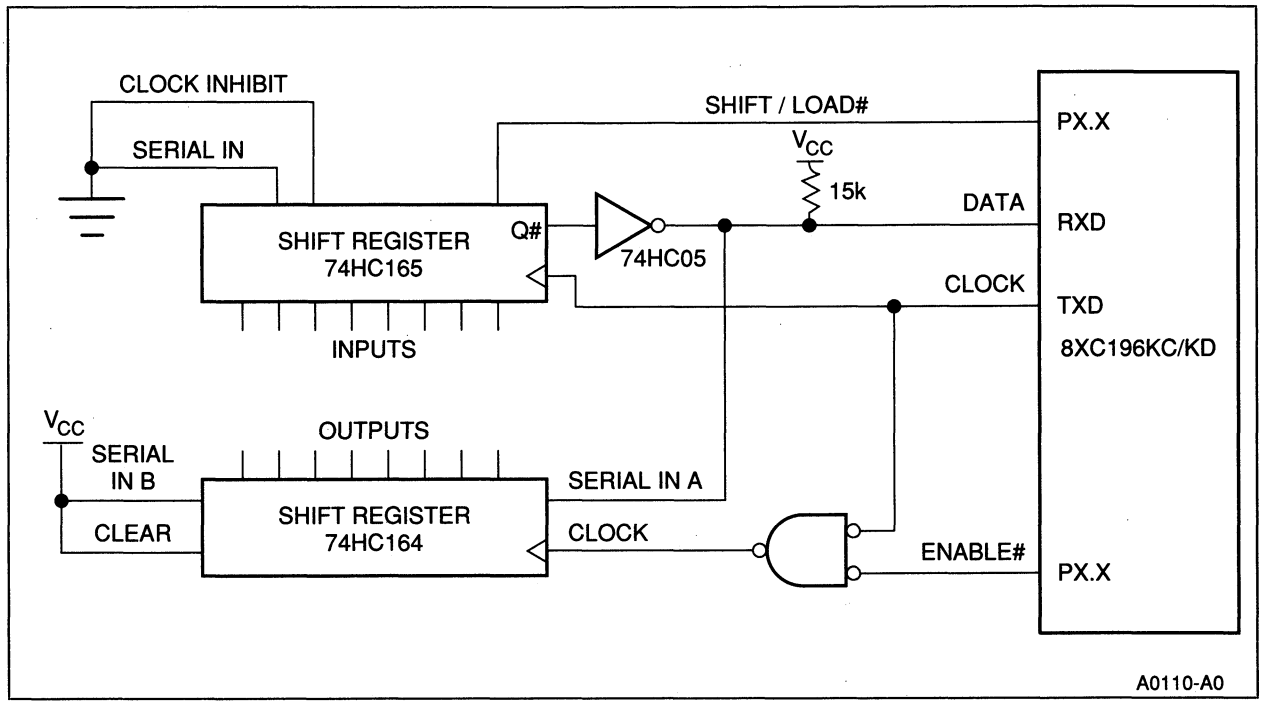

Figure 7-1. Typical Shift Register Circuit for Mode 0

In Mode 0, RXD must be disabled for transmissions to begin and enabled for receptions to begin. (See "Configuring TXD and RXD" on page 7-7.) When RXD is disabled, writing to SBUF (TX) starts a transmission. When RXD is enabled, either a rising edge on the RXD input or clearing the Receive Interrupt (RI) flag starts a reception.

Disabling RXD stops a reception in progress and inhibits further receptions. To avoid a partial or undesired complete reception, disable RXD before clearing the RI flag. This can be handled in an interrupt environment by using software flags or in straight-line code by using the Interrupt Pending register to signal the completion of a reception.

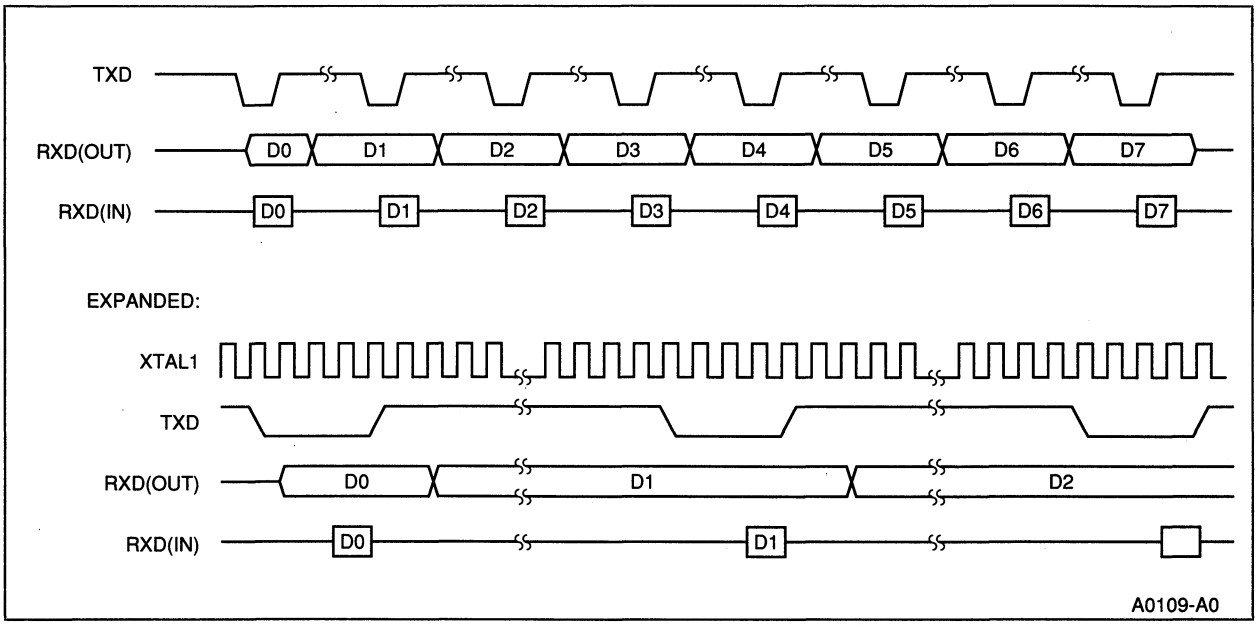

Figure 7-2. Mode 0 Timing

During a reception, the RI flag is set one bit time after the last data bit has been received. The receive interrupt signal is generated immediately before the RI flag is set. During a transmission, the TI flag is set immediately after the end of the last (eighth) data bit has been transmitted. The transmit interrupt signal is generated when the TI flag is set.

7.2. ASYNCHRONOUS MODES

Modes 1, 2, and 3 are full-duplex serial transmit/receive modes, meaning that they can transmit and receive data simultaneously. Mode 1 is the standard 8-bit, asynchronous mode used for normal serial communications. Modes 2 and 3 are 9-bit asynchronous modes typically used for interprocessor communications (see "Multiprocessor Communications" on page 7-5). In Mode 2, the serial port generates an interrupt only if the ninth data bit is set. In Mode 3, the serial port always generates an interrupt upon completion of a data transmission or reception.

When the serial port is configured for Mode 1, 2, or 3, writing to SBUF (TX) causes the serial port to start transmitting data. New data placed in SBUF (TX) is not transmitted until the stop bit of the previous data has been sent. A falling edge on the RXD input causes the serial port to begin receiving data if RXD is enabled. Disabling RXD stops a reception in progress and inhibits further receptions. (See "Configuring TXD and RXD" on page 7-7.)

7.2.1. Mode 1

Mode 1 is the standard asynchronous communications mode. The data frame used in this mode (see Figure 7-3) consists of ten bits: a start bit (0), eight data bits (LSB first), and a stop bit (1). If parity is enabled, an even parity bit is sent instead of the eighth data bit, and parity is checked on reception.

The transmit and receive functions are controlled by separate shift clocks. The transmit shift clock starts when the baud rate generator is initialized. The receive shift clock is reset when a 1-to-0 transition (start bit) is received. Therefore, the transmit clock may not be in sync with the receive clock, although both will be at the same frequency.

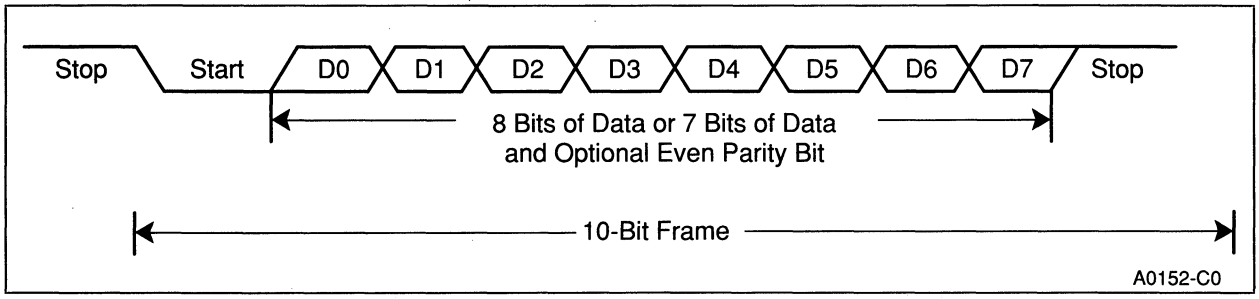

Figure 7-3. Serial Port Frames for Mode 1

The Transmit Interrupt (TI) and Receive Interrupt (RI) flags are set to indicate when operations are complete. During a reception, the RI flag is set just before the end of the stop bit. The receive interrupt signal is generated when the RI flag is set. During a transmission, the TI flag is set at the beginning of the stop bit. The transmit interrupt signal is generated when the TI flag is set. The next byte cannot be sent until the stop bit is sent.

Use caution when connecting more than two devices with the serial port in half-duplex (i.e., with one wire for transmit and receive). The receiving processor must wait for one bit time after the RI flag is set before starting to transmit. Otherwise, the transmission could corrupt the stop bit, causing a problem for other devices listening on the link.

7.2.2. Mode 2

Mode 2 is the asynchronous, ninth-bit recognition mode. This mode is commonly used with Mode 3 for multiprocessor communications. Figure 7-4 shows the data frame used in this mode. It consists of a start bit (0), nine data bits (LSB first), and a stop bit (1). During transmissions, setting the TB8 bit in the SP_CON register before writing to SBUF (TX) sets the ninth transmission bit. The hardware clears the TB8 bit after every transmission, so it must be set (if desired) prior to each write to SBUF (TX). During reception, the RI flag is set and an interrupt is generated **only** if the TB8 bit is set. This provides an easy way to have selective reception on a data link. (See "Multiprocessor Communications" on page 7-5.) Parity cannot be enabled in this mode.

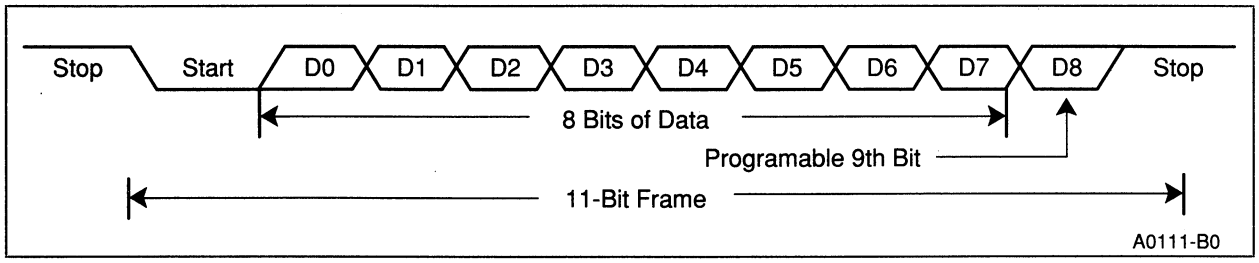

Figure 7-4. Serial Port Frames in Mode 2 and 3

7.2.3. Mode 3

Mode 3 is the asynchronous, ninth-bit mode. The data frame for this mode is identical to that of Mode 2. Mode 3 differs from Mode 2 during transmissions because parity can be enabled, in which case the ninth bit becomes the even parity bit. When parity is disabled, data bits 0–7 are written to the serial port transmit buffer, and the ninth data bit is written to bit 4 (TB8) bit in the SP_CON register. In Mode 3, a reception always causes an interrupt, regardless of the state of the ninth bit. If parity is disabled, the ninth data bit can be read in bit 7 (RB8) of the SP_STAT register. If parity is enabled, then RB8 becomes the Received Parity Error (RPE) flag.

7.2.4. Mode 2 and 3 Timings

Operation in Modes 2 and 3 is similar to Mode 1 operation. The only difference is that the data consists of 9 bits, so 11-bit packages are transmitted and received. During a reception, the RI flag is set just after the end of the stop bit. The receive interrupt signal is generated when the RI flag is set. During a transmission, the TI flag is set at the beginning of the stop bit. The transmit interrupt signal is generated when the TI flag is set. The ninth bit can be used for parity or multiprocessor communications.

7.2.5. Multiprocessor Communications

Modes 2 and 3 are provided for multiprocessor communications. In Mode 2, the serial port generates the Receive Interrupt (RI) only when the ninth data bit is set. In Mode 3, the serial port generates the Receive Interrupt (RI) regardless of the value of the ninth bit. The ninth bit is always set in address frames and always cleared in data frames. One way to use these modes for multiprocessor communication is to set the master processor to Mode 3 and the slave processors to Mode 2. When the master processor wants to transmit a block of data to one of several slaves, it sends out an address frame that identifies the target slave. An address frame will interrupt all slaves because the ninth bit is set. Each slave can examine the address byte and see if it is being addressed. The addressed slave switches to Mode 3 to receive the data frames, while the slaves that were not addressed remain in Mode 2 and are not interrupted.

7.3. PROGRAMMING THE SERIAL PORT

Table 7-1 lists the registers that affect the performance and function of the serial port.

Table 7-1. Serial Port Control and Status Registers

Register Mnemonic	Register Name	Description
BAUD_RATE	Baud Rate	This register selects the serial port baud rate and clock source. It must be written with two bytes, the least-significant byte first. The most-significant bit selects the clock source. The lower 15 bits represent the BAUD_VALUE, an unsigned integer that determines the baud rate.
IOC1	I/O Control Register 1	Setting bit 5 of this register enables the TXD function of P2.0.
SBUF (RX)	Serial Port Receive Buffer	This register contains data received from the serial port.
SBUF (TX)	Serial Port Transmit Buffer	This register contains data that is ready for transmission. In Modes 1, 2, and 3, writing to SBUF (TX) starts a transmission. In Mode 0, writing to SBUF (TX) starts a transmission only if the receiver is disabled (SP_CON.3=0)
SP_CON	Serial Port Control Register	This register selects the communications mode and enables or disables the receiver, even parity checking, and ninth-bit data transmissions. The TB8 bit is cleared after each transmission.
SP_STAT	Serial Port Status Register	This register contains the serial port status bits. It has a status bit for receive Overrun Errors (OE), Transmit buffer empty (TXE), Framing Errors (FE), Transmit Interrupts (TI), Receive Interrupts (RI), and Received Parity Error (RPE) or Received Bit 8 (RB8). Reading SP_STAT clears the OE, FE, TI, RI, and RPE/RB8 bits. Writing a byte to SBUF (TX) clears the TXE bit.

NOTE

Always write zeros to the reserved bits in these registers.

7.3.1. Selecting the Communications Mode and Enabling Parity

Bits 0 and 1 of the SP_CON register select the serial communications mode (see Figure 7-5 and Table 7-2). Selecting a new mode resets the serial I/O port and aborts any transmission or reception in progress on the channel. Setting bit 2 enables parity. Setting bit 3 enables the RXD function of pin P2.1. Bit 4 is the ninth data bit that will be transmitted in Mode 2 or 3 or, if parity is enabled, it is the even parity value. Note that bit 4 (TB8) is cleared after each transmission.

Table 7-2. Mode Selections

Bit 1	Bit 0	Selected Mode
0	0	Mode 0
0	1	Mode 1
1	0	Mode 2
1	1	Mode 3

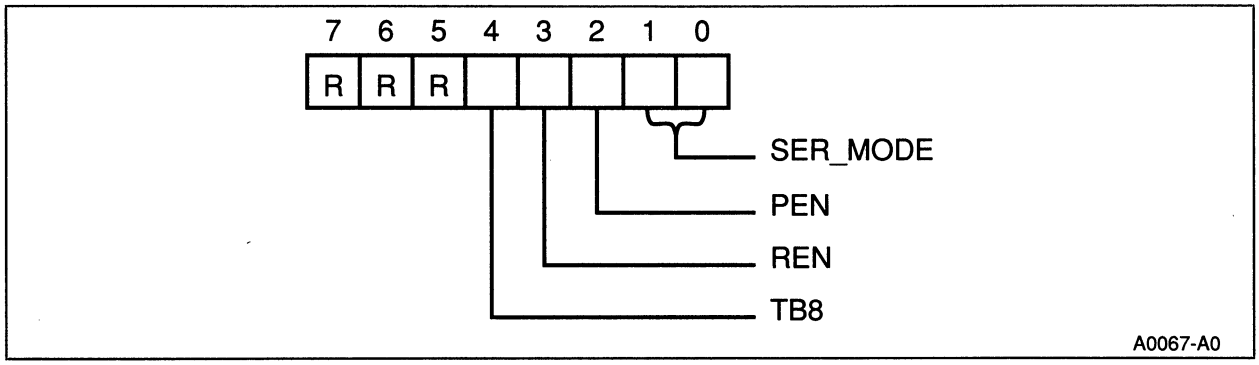

Figure 7-5. SP_CON Register

7.3.2. Configuring TXD and RXD

Table 7-3 lists the pins associated with the serial port. To enable TXD, set bit 5 of register IOC1; to disable TXD, clear the bit. To enable RXD, set bit 3 of register SP_CON; to disable RXD, clear the bit. (The "Programming the Baud Rate and Clock Source" section on page 7-9 discusses enabling T2CLK).

NOTE

The TXD pin controls entry into the ONCE mode when the 8XC196KC/KD is reset. In order to prevent inadvertently entering ONCE mode, external circuitry must not drive this pin low. See Chapter 12, "Special Operating Modes," for additional information on the ONCE mode.

Table 7-3. Serial Port Signals

Function Name	Additional Functions	Selected by	Type	Description
RXD	P2.1/PALE#	SP_CON.3=1	I/O	Receive Serial Data. In Modes 1, 2, and 3, RXD receives serial port input data. In Mode 0, it functions as an input or an open-drain output for data.
T2CLK	P2.3	BAUD_RATE MSB=0	I	Timer 2 clock input and one of two possible clock sources for the baud-rate generator input.
TXD	P2.0/PVER#	IOC1.5=1	O	Transmit Serial Data. In Modes 1, 2, and 3, TXD transmits serial port output data. In Mode 0, it is the serial clock output.

7.3.3. Enabling the Serial Port Interrupts

The serial port can be configured to generate either a single Serial Port interrupt or both a Transmit Interrupt (TI) and a Receive Interrupt (RI). If the Serial Port interrupt is masked and the Receive and Transmit interrupts are enabled, the RI flag and TI flag generate separate Receive and Transmit interrupts. If 8096BH compatibility is not an issue, this configuration is preferred.

If 8096H compatibility is desired, disable the Receive and Transmit interrupts and enable the Serial interrupt. Both RI flag and TI flag generate the Serial Port interrupt. When the TI flag generates the interrupt, bit 5 in the SP_STAT register is set. When the RI flag generates the interrupt, bit 6 in the SP_STAT register is set.

To enable the individual interrupts, set the TI_MASK and RI_MASK bits in the INT_MASK1 register. To enable the 8096BH-compatible interrupt, set the SER_MASK bit in the INT_MASK register. See Chapter 5, "Interrupts," for more information about interrupts.

Table 7-4. Serial Port Interrupt Registers

Register Mnemonic	Register Name	Hex Address	Description
INT_MASK	Interrupt Mask	08H	Setting bit 6 of this register enables the Serial Port interrupt (INT06, 200CH), which is the 8096BH-compatible configuration. Clearing bit 6 disables (masks) the interrupt.
INT_MASK1	Interrupt Mask 1	13H	Setting bit 0 of this register enables the Transmit interrupt (INT08, 2030H). Clearing bit 0 disables (masks) the interrupt. Setting bit 1 of this register enables the Receive interrupt (INT09, 2032H). Clearing bit 1 disables (masks) the interrupt.
INT_PEND	Interrupt Pending	09H	When set, bit 6 of this register indicates a pending Serial Port interrupt (INT06). It is cleared when the interrupt vectors to 200CH.
INT_PEND1	Interrupt Pending 1	12H	When set, bit 0 indicates a pending Transmit interrupt (INT08). It is cleared when the interrupt vectors to 2030H. When set, bit 1 indicates a pending Receive interrupt (INT09). It is cleared when the interrupt vectors to 2032H.

7.3.4. Programming the Baud Rate and Clock Source

The BAUD_RATE register selects the clock input into the baud-rate generator and defines the baud rate for all serial I/O modes. This word register must be loaded sequentially with two bytes. Always load the least-significant byte first.

The most-significant bit selects one of two possible clock sources for the baud-rate generator. To select the XTAL1 input signal (F_{OSC}) as the baud-rate clock source, set bit 15; to select an external signal on the T2CLK pin, clear bit 15. The maximum T2CLK input frequency is $F_{OSC}/4$.

The lower 15 bits represent BAUD_VALUE, an unsigned integer that determines the baud rate. The maximum value of BAUD_VALUE is 7FFFH (32,767 decimal). The minimum value in Modes 1, 2, and 3 is 0000H if XTAL1 is the baud-rate generator clock; otherwise, it is 0001H. The minimum value in Mode 0 is always 0001H.

Use the following equations to determine the BAUD_VALUE for a given baud rate.

Synchronous Mode 0:

$$\text{BAUD\_VALUE} = \frac{F_{OSC}}{\text{Baud Rate} \times 2} - 1 \quad \text{or} \quad \frac{\text{T2CLK}}{\text{Baud Rate}}$$

Asynchronous Modes 1, 2, and 3:

$$\text{BAUD\_VALUE} = \frac{F_{OSC}}{\text{Baud Rate} \times 16} - 1 \quad \text{or} \quad \frac{\text{T2CLK}}{\text{Baud Rate} \times 8}$$

Table 7-5 shows the BAUD_RATE values for common baud rates when using a 16 MHz XTAL1 clock input. Because of rounding, the BAUD_VALUE formula is not exact and the resulting baud rate is slightly different than desired. Table 7-5 shows the percentage of error when using the sample BAUD_RATE values. In most cases, a serial link will work with up to 5.0% difference in the receiving and transmitting baud rates.

Table 7-5. BAUD_RATE Values when Using XTAL1 at 16 MHz

Baud Rate	BAUD_RATE (Note 1)		% Error	
	Mode 0	Mode 1, 2, 3	Mode 0	Mode 1, 2, 3
9600	8340H	8067H	0.04	0.16
4800	8682H	80CFH	0.02	0.16
2400	8D04H	81A0H	0.01	0.08
1200	8A9AH	8340H	0	0.04
300	E82BH	8D04H	0	0.01

NOTES:

1. Bit 15 is always set when XTAL1 is selected as the clock source for the baud-rate generator.

7.4. SERIAL PORT STATUS

The status of the serial port is reflected in the SP_STAT register (see Figure 7-6). Note that reading the SP_STAT register clears all bits except TXE. For this reason, we recommend that you copy the contents of the SP_STAT register into a shadow register and then execute bit-test instructions such as JBC and JBS on the shadow register. Otherwise, the flags will be cleared when the bit-test instruction is executed.

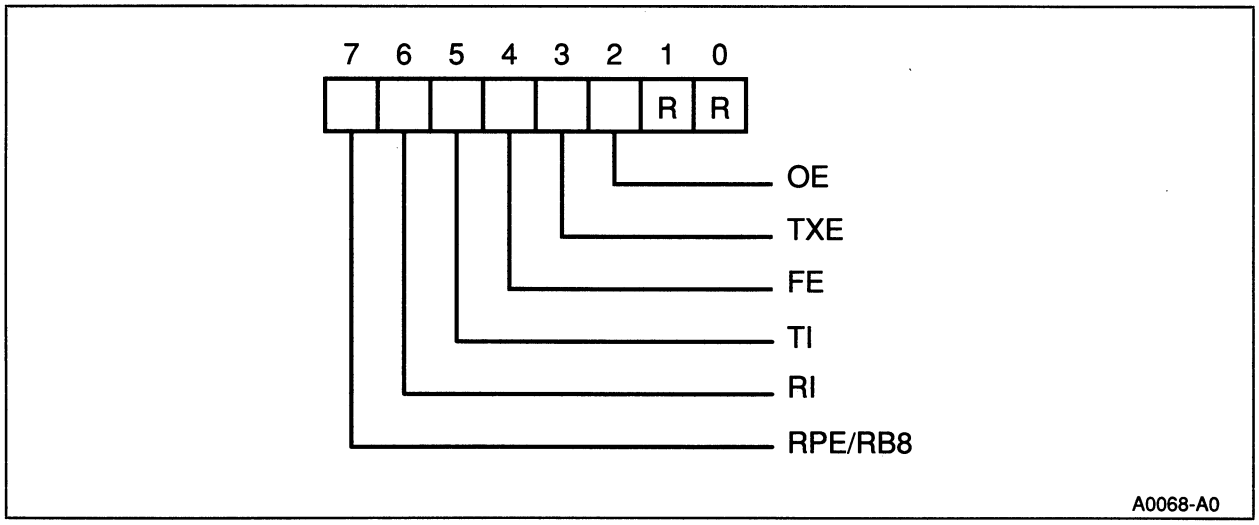

Figure 7-6. SP_STAT Register

The receiver on the 8XC196KC/KD checks for a valid stop bit. If a stop bit is not found within the appropriate time, the Framing Error (FE) bit in the SP_STAT register is set. When the stop bit is detected, the data in the receive shift register is loaded into SBUF (RX) and the Receive Interrupt (RI) flag is set. If this happens before the previous byte in SBUF (RX) is read, the Overrun Error (OE) bit is set. SBUF (RX) always contains the latest byte received; it is never a combination of the two bytes.

The Receive Interrupt (RI) flag indicates whether an incoming data byte has been received. The Transmit Interrupt (TI) flag indicates whether a data byte has finished transmitting. These flags also trigger interrupts and set the corresponding bits in the Interrupt Pending register. (See "Enabling the Serial Port Interrupts" on page 7-8.) Note that while a receive/transmit event sets the RI/TI bit in SP_STAT and the corresponding Interrupt Pending bit, a software write to RI/TI in SP_STAT does not affect the Interrupt Pending bits and does not cause an interrupt. Similarly, reading SP_STAT clears the RI and TI bits, but does not clear the corresponding bits in the Interrupt Pending register.

The Transmitter Empty (TXE) bit is set if SBUF (TX) and its buffer are empty and ready to take up to two bytes. TXE is cleared as soon as a byte is written to SBUF (TX). One byte may be written if TI alone is set. By definition, if TXE has just been set, a transmission has completed and TI will be set.

The Received Parity Error (RPE) flag or the Received Bit 8 (RB8) flag applies for parity enabled or disabled, respectively. If parity is enabled, RPE is set if a parity error is detected. If parity is not enabled, RB8 is the ninth data bit received in Modes 2 and 3.