CHAPTER 9
SERIAL I/O

9.1 UART INTRODUCTION

Select Z8® MCU microcontrollers contain an on-board full-duplex Universal Asynchronous Receiver/Transmitter (UART) for data communications. The UART consists of a Serial I/O Register (SIO) located at address F0H, and its associated control logic (Figure 9-1). The SIO is actually two registers, the receiver buffer and the transmitter buffer, which are used in conjunction with Counter/Timer T0 and Port 3 I/O lines P30 (input) and P37 (output). Counter/Timer T0 provides the clock input for control of the data rates.

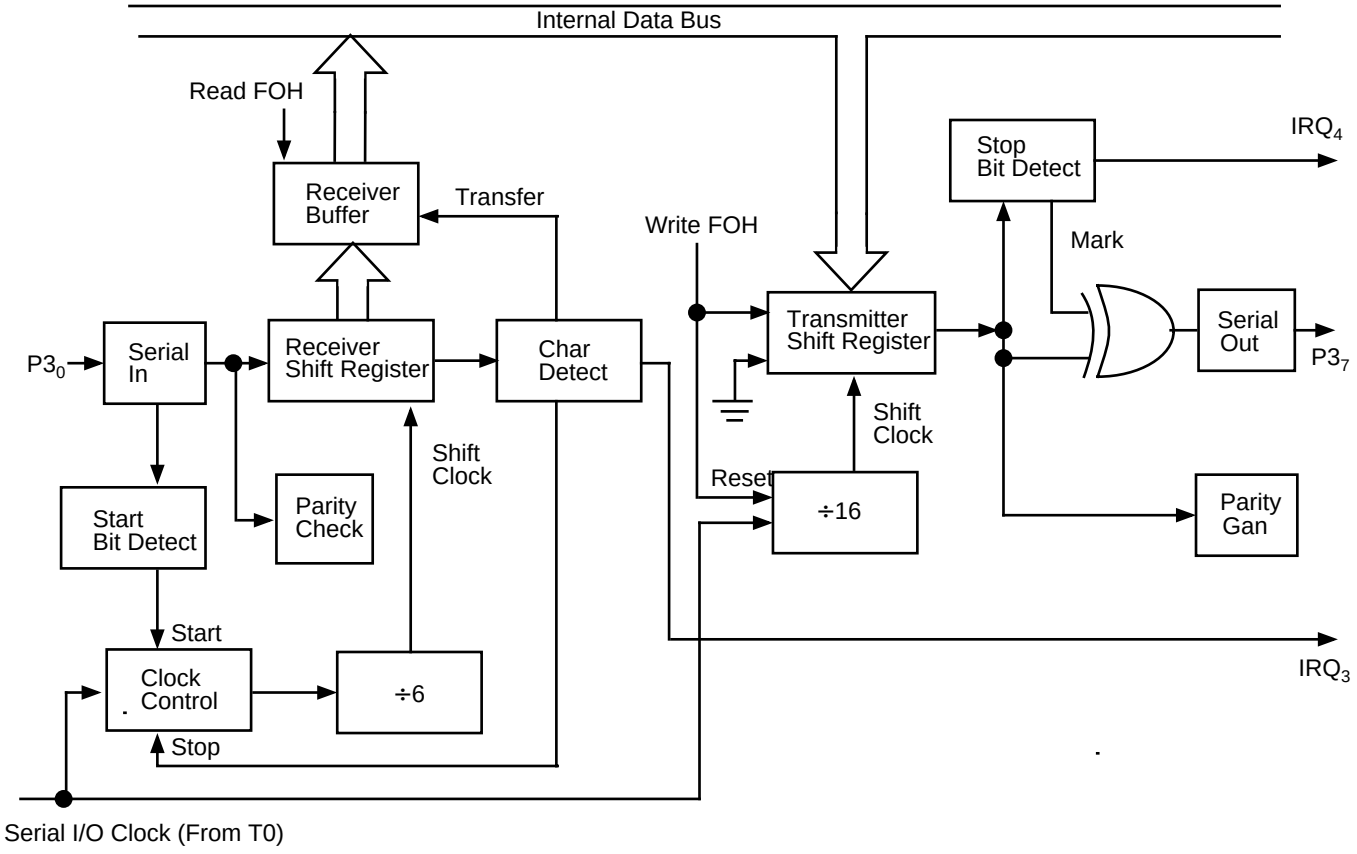

Figure 9-1. UART Block Diagram

Configuration of the UART is controlled by the Port 3 Mode Register (P3M) located at address F7H. The Z8® always transmits eight bits between the start and stop bits (eight Data Bits or seven Data Bits and one Parity Bit). Odd parity generation and detection is supported.

The SIO Register and its associated Mode Control Registers are mapped into the Standard Z8 Register File as shown in Table 9-1. The organization allows the software to access the UART as general-purpose registers, eliminating the need for special instructions.

Table 9-1. UART Register Map

Register Name	Identifier	Hex Address
Port 3 Mode	P3M	F7
T0 Prescaler	PRE0	F5
Timer/Counter0	T0	F4
Timer Mode	TMR	F1
UART	SIO	F0

9.2 UART BIT-RATE GENERATION

When Port 3 Mode Register bit 6 is set to 1, the UART is enabled and T0 automatically becomes the bit rate generator (Figure 9-2). The end-of-count signal of T0 no longer generates Interrupt Request IRQ4. Instead, the signal is used as the input to the divide-by-16 counters (one each for the receiver and the transmitter) that clock the data stream.

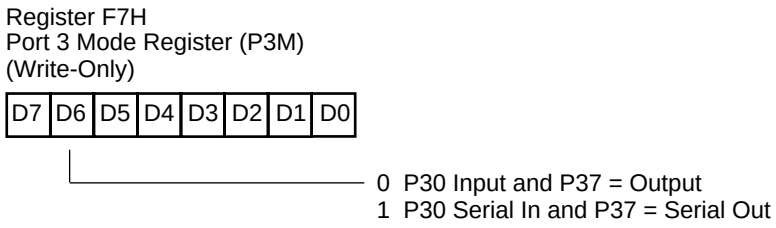

Figure 9-2. Port 3 Mode Register (P3M) and Bit-Rate Generation

The divide chain that generates the bit rate is shown in Figure 9-3. The bit rate is given by the following equation:

Bit Rate = XTAL Frequency/(2 x 4 x p x t x 16)

where p and t are the initial values in Prescaler0 and Counter/Timer0, respectively. The final divide-by-16 is required since T0 runs at 16 times the bit rate in order to synchronize on the incoming data.

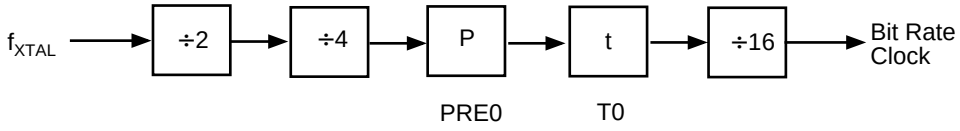

Figure 9-3. Bit Rate Divide Chain

To configure the Z8 for a specific bit rate, appropriate values as determined by the above equation must be loaded into registers PRE0 (F5H) and T0 (F4H). PRE0 also controls the counting mode for T0 and should therefore be set to the Continuous Mode (D0 = 1).

For example, given an input clock frequency (XTAL) of 11.9808 MHz and a selected bit rate of 1200 bits per second, the equation is satisfied by p = 39 and t = 2. Counter/Timer T0 should be set to 02H. With T0 in Continuous Mode, the value of PRE0 becomes 9DH (Figure 9-4).

Table 9-2 lists several commonly used bit rates and the values of XTAL, p, and t required to derive them. This list is presented for convenience and is not intended to be exhaustive.

Table 9-2. Bit Rates

Bit Rate	7,3728		7,9872		9,8304		11,0592		11,6736		11,9808		12,2880	
	p	t	p	t	p	t	p	t	p	t	p	t	p	t
19200	3	1	–	–	4	1	–	–	–	–	–	–	5	1
9600	3	2	–	–	4	2	9	1	–	–	–	–	5	2
4800	3	4	13	1	4	4	9	2	19	1	–	–	5	4
2400	3	8	13	2	4	8	9	4	19	2	39	1	5	8
1200	3	16	13	4	4	16	9	8	19	4	39	2	5	16
600	3	32	13	8	4	32	9	16	19	8	39	4	5	32
300	3	64	13	16	4	64	9	32	19	16	39	8	5	64
150	3	128	13	32	4	128	9	64	19	32	39	16	5	128
110	3	175	3	189	4	175	5	157	4	207	17	50	8	109

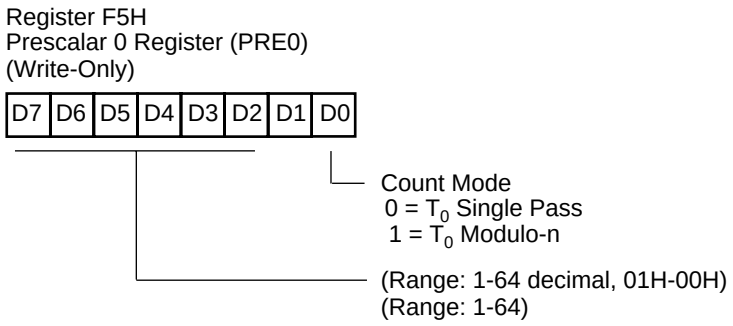

Figure 9-4. Prescaler 0 Register (PRE0) Bit-Rate Generation

The bit rate generator is started by setting the Timer Mode Register (TMR) (F1H) bit 1 and bit 0 both to 1 (Figure 9-5). This transfers the contents of the Prescaler 0 Register and Counter/Timer0 Register to their corresponding down counters. In addition, counting is enabled so that UART operations begin.

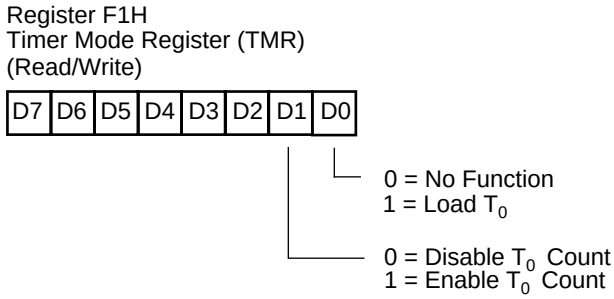

Figure 9-5. Timer Mode Register (TMR) Bit Rate Generation

9.3 UART RECEIVER OPERATION

The receiver consists of a receiver buffer (SIO Register [F0H]), a serial-in, parallel-out shift register, parity checking, and data synchronizing logic. The receiver block diagram is shown as part of Figure 9-1.

9.3.1 Receiver Shift Register

After a hardware reset or after a character has been received, the Receiver Shift Register is initialized to all 1s and the shift clock is stopped. Serial data, input through Port 3 bit 0, is synchronized to the internal clock by two D-type flip-flops before being input to the Shift Register and the start bit detection circuitry.

The start bit detection circuitry monitors the incoming data stream, looking for a start bit (a High-to-Low input transition). When a start bit is detected, the shift clock logic is enabled. The T0 input is divided-by-16 and, when the count equals eight, the divider outputs a shift clock. This clock shifts the start bit into the Receiver Shift Register at the center of the bit time. Before the shift actually occurs, the input is rechecked to ensure that the start bit is valid. If the detected start bit is false, the receiver is reset and the process of looking for a start bit is repeated. If the start bit is valid, the data is shifted into the Shift Register every sixteen counts until a full character is assembled (Figure 9-6).

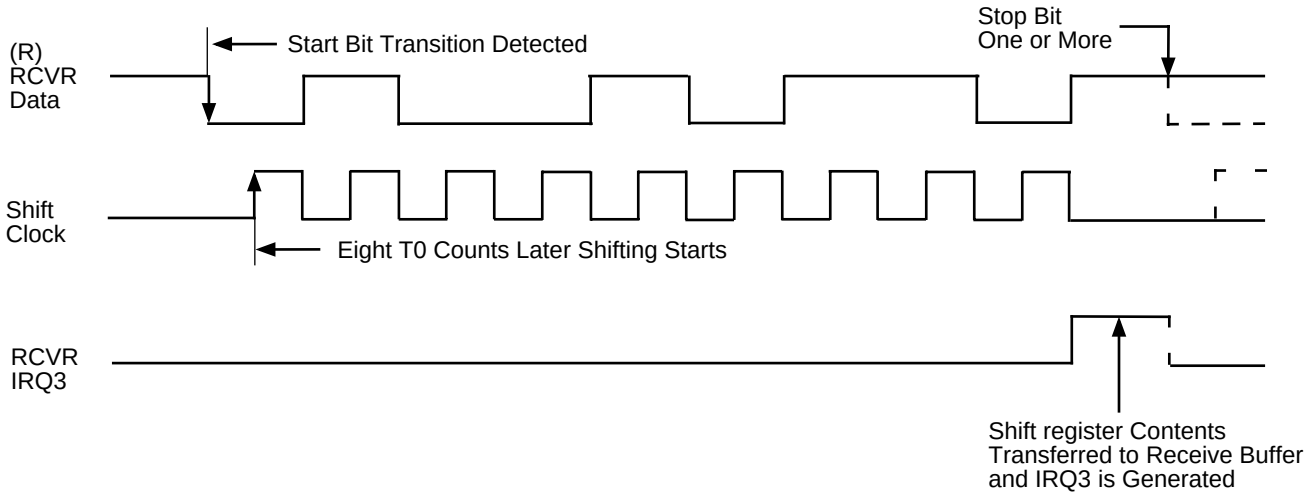

Figure 9-6. Receiver Timing

After a full character has been assembled in the receiver's buffer, SIO Register (F0H), Interrupt Request IRQ3 is generated. The shift clock is stopped and the Shift Register reset to all 1s. The start bit detection circuitry begins monitoring the data input for the next start bit. This cycle allows the receiver to synchronize on the center of the bit time for each incoming character.

9.3.2 Overwrites

Although the receiver is single buffered, it is not protected from being overwritten, so the software must read the SIO Register (F0H) within one character time after the interrupt request (IRQ3). The Z8 does not have a flag to indicate this overrun condition. If polling is used, the IRQ3 bit in the Interrupt Request Register must be reset by software.

9.3.3 Framing Errors

Framing error detection is not supported by the receiver hardware, but by responding to the interrupt request within one character bit time, the software can test for a stop bit on P30. Port 3 bits are always readable, which facilitates break detection. For example, if a null character is received, testing P30 results in a 0 being read.

9.3.4 Parity

The data format supported by the receiver must have a start bit, eight data bits, and at least one stop bit. If parity is on, bit 7 of the data received will be replaced by a Parity Error Flag. A parity error sets bit 7 to 1, otherwise, bit D7 is set to 0. Figure 9-7 shows these data formats.

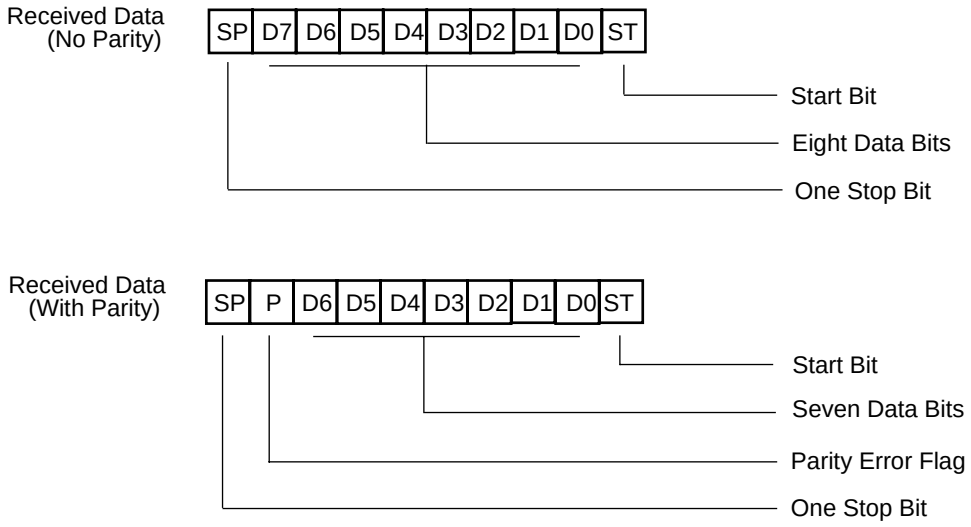

Figure 9-7. Receiver Data Formats

The Z8 hardware supports odd parity only, that is enabled by setting the Port 3 Mode Register bit 7 to 1 (Figure 9-8). If even parity is required, the Parity Mode should be disabled (P3M bit 7 set to 0), and software must calculate the received data's parity.

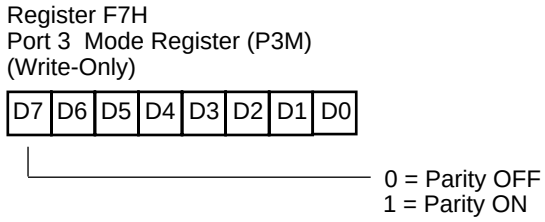

Figure 9-8. Port 3 Mode Register (P3M) Parity

9.4 TRANSMITTER OPERATION

The transmitter consists of a transmitter buffer (SIO Register [F0H]), a parity generator, and associated control logic. The transmitter block diagram is shown as part of Figure 9-1.

After a hardware reset or after a character has been transmitted, the transmitter is forced to a marking state (output always High) until a character is loaded into the transmitter buffer, SIO Register (F0H). The transmitter is loaded by specifying the SIO Register as the destination register of any instruction.

T0's output drives a divide-by-16 counter that in turn generates a shift clock every 16 counts. This counter is reset when the transmitter buffer is written by an instruction. This reset synchronizes the shift clock to the software. The transmitter then outputs one bit per shift clock, through Port 3 bit 7, until a start bit, the character written to the buffer, and two stop bits have been transmitted. After the second stop bit has been transmitted, the output is again forced to a marking state. Interrupt request IRQ4 is generated at this time to notify the processor that the transmitter is ready to accept another character.

9.4.1 Overwrites

The user is not protected from overwriting the transmitter, so it is up to the software to respond to IRQ4 appropriately. If polling is used, the IRQ4 bit in the Interrupt Request Register must be reset.

9.4.2 Parity

The data format supported by the transmitter has a start bit, eight data bits, and at least two stop bits. If parity is on, bit 7 of the data transmitted will be replaced by an odd parity bit. Figure 9-9 shows the transmitter data formats.

Parity is enabled by setting Port 3 Mode Register bit 7 to 1. If even parity is required, the parity mode should be disabled (P3M bit 7 reset to 0), and software must modify the data to include even parity.

Since the transmitter can be overwritten, the user is able to generate a break signal. This is done by writing null characters to the transmitter buffer (SIO Register [F0H]) at a rate that does not allow the stop bits to be output. Each time the SIO Register is loaded, the divide-by-16 counter is resynchronized and a new start bit is output followed by data.

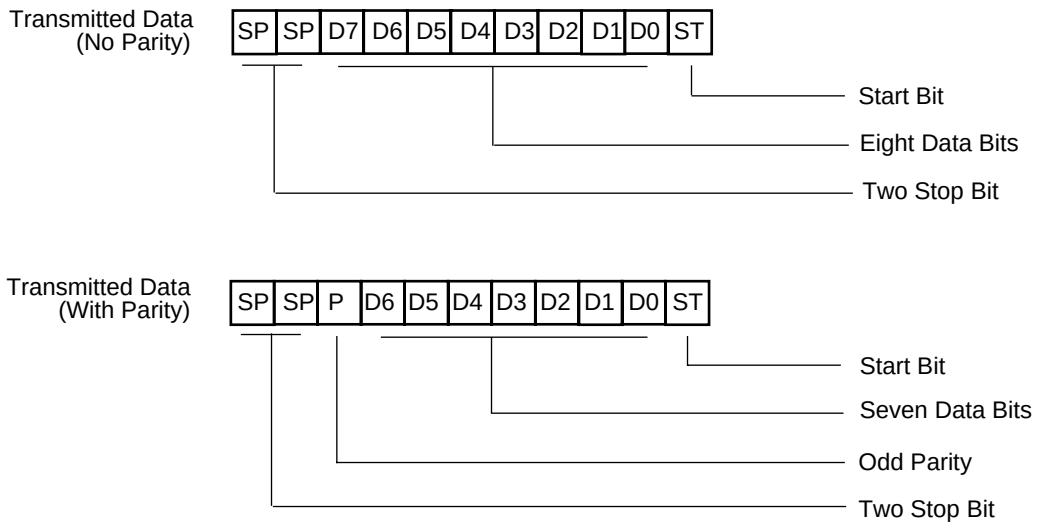

Figure 9-9. Transmitter Data Formats

9.5 UART RESET CONDITIONS

After a hardware reset, the SIO Register contents are undefined, and Serial Mode and parity are disabled. Figures 9-10 and 9-11 show the binary reset values of the SIO Register and its associated mode register P3M.

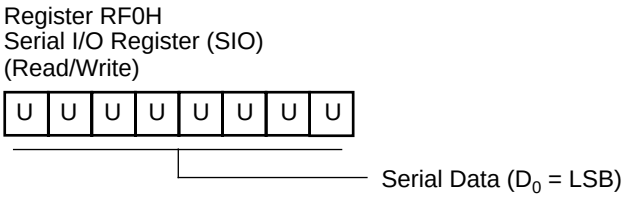

Figure 9-10. SIO Register Reset

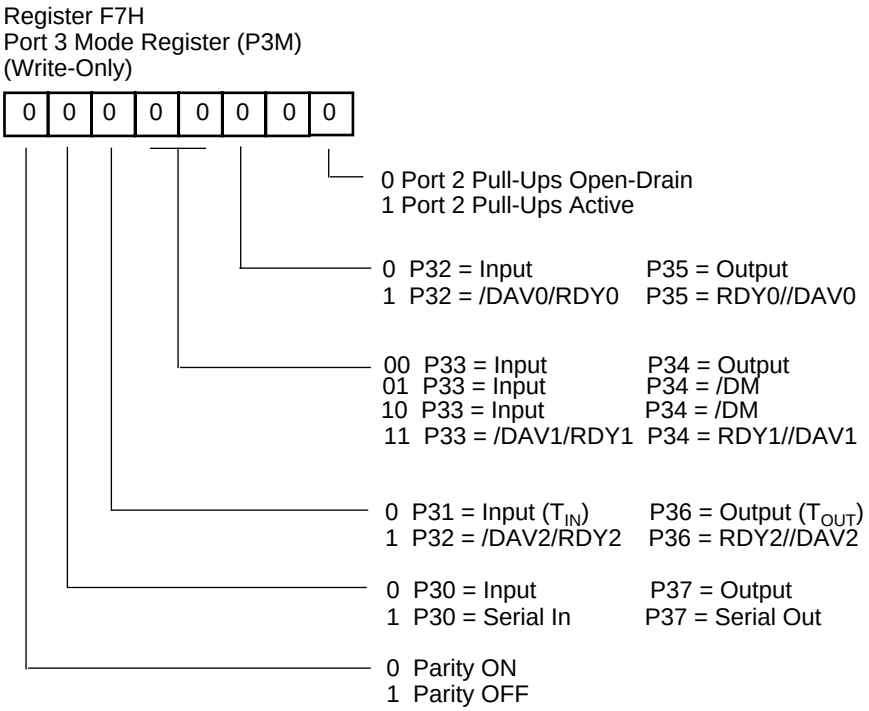

Figure 9-11. P3M Register Reset

9.6 SERIAL PERIPHERAL INTERFACE (SPI)

Select Z8® microcontrollers incorporate a serial peripheral interface (SPI) for communication with other microcontrollers and peripherals. The SPI includes features such as Stop-Mode Recovery, Master/Slave selection, and Compare mode. Table 9-3 contains the pin configuration for the SPI feature when it is enabled. The SPI consists of four registers: SPI Control Register (SCON), SPI Compare Register (SCOMP), SPI Receive/Buffer Register (RxBUF), and SPI Shift Register. SCON is located in bank (C) of the Expanded Register File at address 02.

Table 9-3. SPI Pin Configuration

Name	Function	Pin Location
DI	Data-In	P20
DO	Data-Out	P27
SS	Slave Select	P35
SK	SPI Clock	P34

The SPI Control Register (SCON) (Figure 9-12), is a read/write register that controls Master/Slave selection, interrupts, clock source and phase selection, and error flag. Bit 0 enables/disables the SPI with the default being SPI disabled. A 1 in this location will enable the SPI, and a 0 will disable the SPI. Bits 1 and 2 of the SCON register in Master Mode select the clock rate. The user may choose whether internal clock is divide-by-2, 4, 8, or 16. In Slave Mode, Bit 1 of this register flags the user if an overrun of the RxBUF Register has occurred. The RxCharOverrun flag is only reset by writing a 0 to this bit. In slave mode, bit 2 of the Control Register disables the data-out I/O function. If a 1 is written to this bit, the data-out pin is released to its original port configuration. If a 0 is written to this bit, the SPI shifts out one bit for each bit received. Bit 3 of the SCON Register enables the compare feature of the SPI, with the default being disabled. When the compare feature is enabled, a comparison of the value in the SCOMP Register is made with the value in the RxBUF Register. Bit 4 signals that a receive character is available in the RxBUF Register.

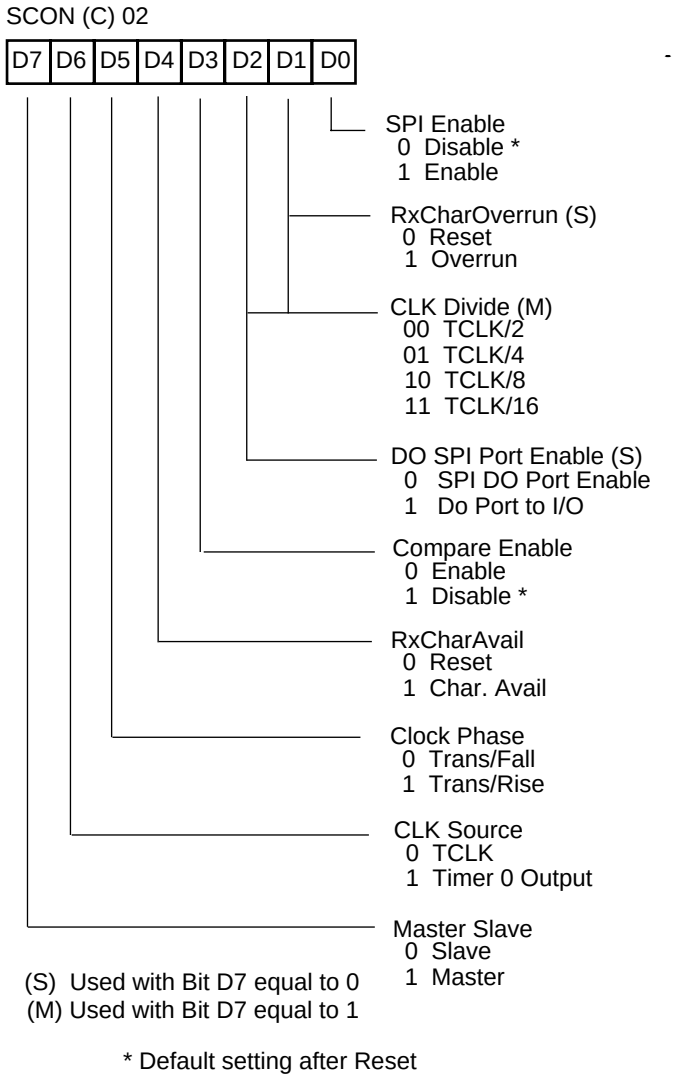

Figure 9-12. SPI Control Register (SCON)

If the associated IRQ3 is enabled, an interrupt is generated. Bit 5 controls the clock phase of the SPI. A 1 in bit 5 allows for receiving data on the clock's falling edge and transmitting data on the clock's rising edge. A 0 allows receiving data on the clock's rising edge and transmitting on the clock's falling edge. The SPI clock source is defined in bit 6. A 1 uses Timer0 output for the SPI clock, and a 0 uses TCLK for clocking the SPI. Finally, bit 7 determines whether the SPI is used as a Master or a Slave. A 1 puts the SPI into Master mode and a 0 puts the SPI into Slave mode.

9.7 SPI OPERATION

The SPI is used in one of two modes: either as system slave, or as system master. Several of the possible system configurations are shown in Figure 9-13. In the slave mode, data transfer starts when the slave select (SS) pin goes active. Data is transferred into the slave's SPI Shift Register through the DI pin, which has the same address as the RxBUF Register. After a byte of data has been received by the SPI Shift Register, a Receive Character Available (RCA/IRQ3) flag and interrupt is generated. The next byte of data will be received at this time. The RxBUF Register must be cleared, or a Receive Character Overrun (RxCharOverrun) flag will be set in the SCON Register, and the data in the RxBUF Register will be overwritten. When the communication between the master and slave is complete, the SS goes inactive.When the SPI is activated as a slave, it operates in all system modes: STOP, HALT, and RUN.

Unless disconnected, for every bit that is transferred into the slave through the DI pin, a bit is transferred out through the DO pin on the opposite clock edge. During slave operation, the SPI clock pin (SK) is an input. In master mode, the CPU must first activate a SS through one of its I/O ports. Next, data is transferred through the master's DO pin one bit per master clock cycle. Loading data into the shift register initiates the transfer. In master mode, the master's clock will drive the slave's clock. At the conclusion of a transfer, a Receive Character Available (RCA/IRQ3) flag and interrupt is generated. Before data is transferred via the DO pin, the SPI Enable bit in the SCON Register must be enabled.

9.8 SPI COMPARE

When the SPI Compare Enable bit, D3 of the SCON Register is set to 1, the SPI Compare feature is enabled. The compare feature is only valid for slave mode. A compare transaction begins when the (SS) line goes active. Data is received as if it were a normal transaction, but there is no data transmitted to avoid bus contention with other slave devices. When the compare byte is received, IRQ3 is not generated. Instead, the data is compared with the contents of the SCOMP Register. If the data does not match, DO remains inactive and the slave ignores all data until the (SS) signal is reset. If the data received matches the data in the SCOMP register, then a SMR signal is generated. DO is activated if it is not tri-stated by D2 in the SCON Register, and data is received the same as any other SPI slave transaction.

Slaves' not comparing remain in their current mode, whereas slaves' comparing wake from a STOP mode by means of an SMR

9.9 SPI CLOCK

The SPI clock maybe driven by three sources: Timer0, a division of the internal system clock, or the external master when in slave mode. Bit D6 of the SCON Register controls what source drives the SPI clock. A 0 in bit D6 of the SCON Register determines the division of the internal system clock if this is used as the SPI clock source. Divide by 2, 4, 8, or 16 is chosen as the scaler.

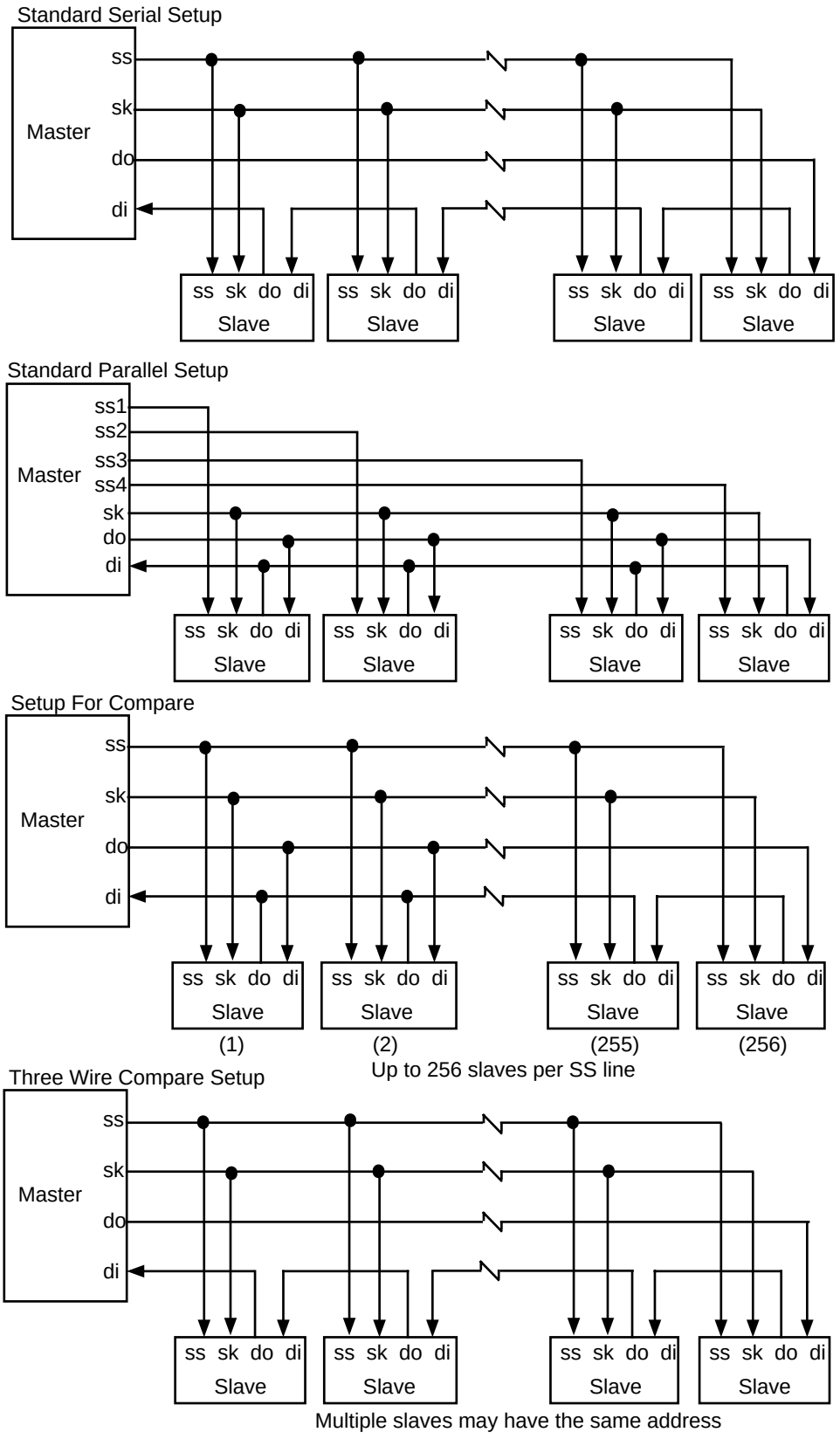

Figure 9-13. SPI System Configuration

9.10 RECEIVE CHARACTER AVAILABLE AND OVERRUN

When a complete data stream is received, an interrupt is generated and the RxCharAvail bit in the SCON Register is set. Bit 4 in the SCON Register is for enabling or disabling the RxCharAvail interrupt. The RxCharAvail bit is available for interrupt polling purposes and is reset when the RxBUF Register is read. RxCharAvail is generated in both master and slave modes. While in slave mode, if the RxBUF is not read before the next data stream is received and loaded into the RxBUF Register, Receive Character Overrun (RxCharOverrun) occurs. Since there is no need for clock control in slave mode, bit D1 in the SPI Control Register is used to log any RxCharOverrun (Figure 9-14 and Figure 9-15).

No	Parameter	Min	Units
1	DI to SK Setup	10	ns
2	SK to D0 Valid	15	ns
3	SS to SK Setup	.5 Tsk	ns
4	SS to D0 Valid	15	ns
5	SK to DI Hold Time	10	ns

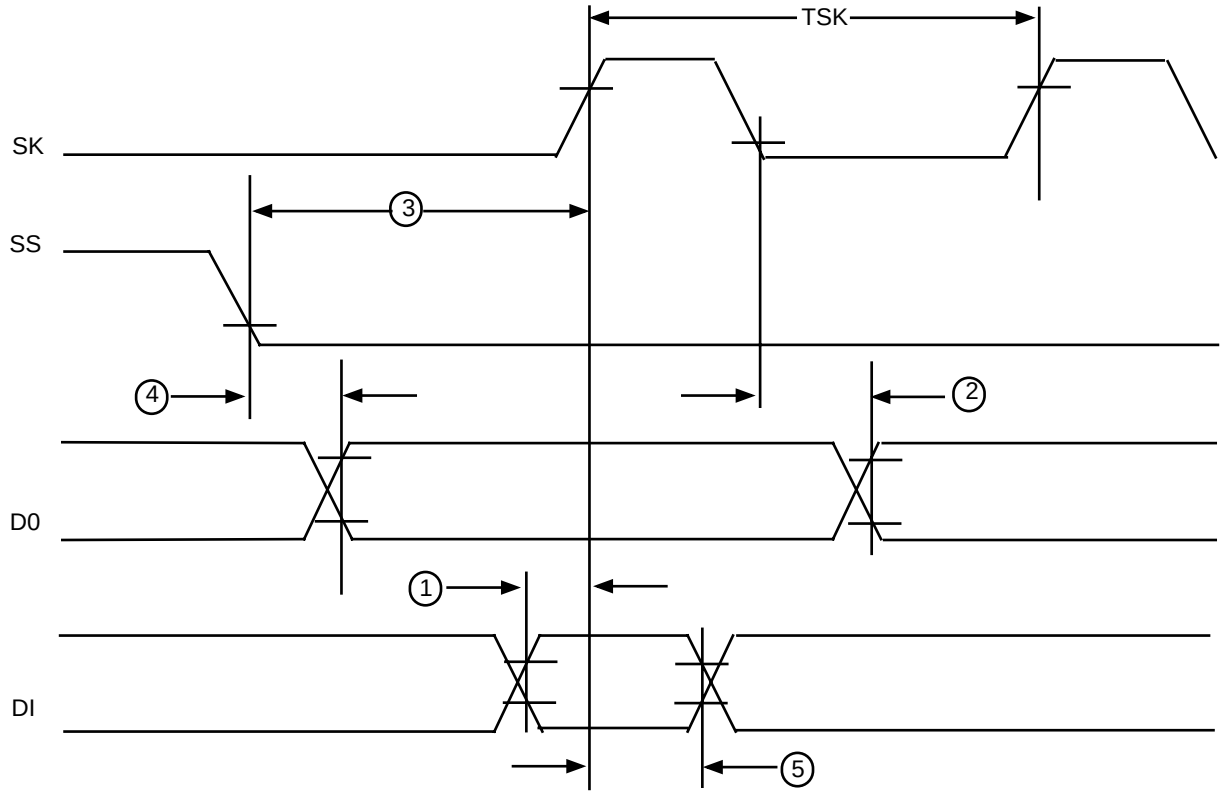

Figure 9-14. SPI Timing

9.10 RECEIVE CHARACTER AVAILABLE AND OVERRUN (Continued)

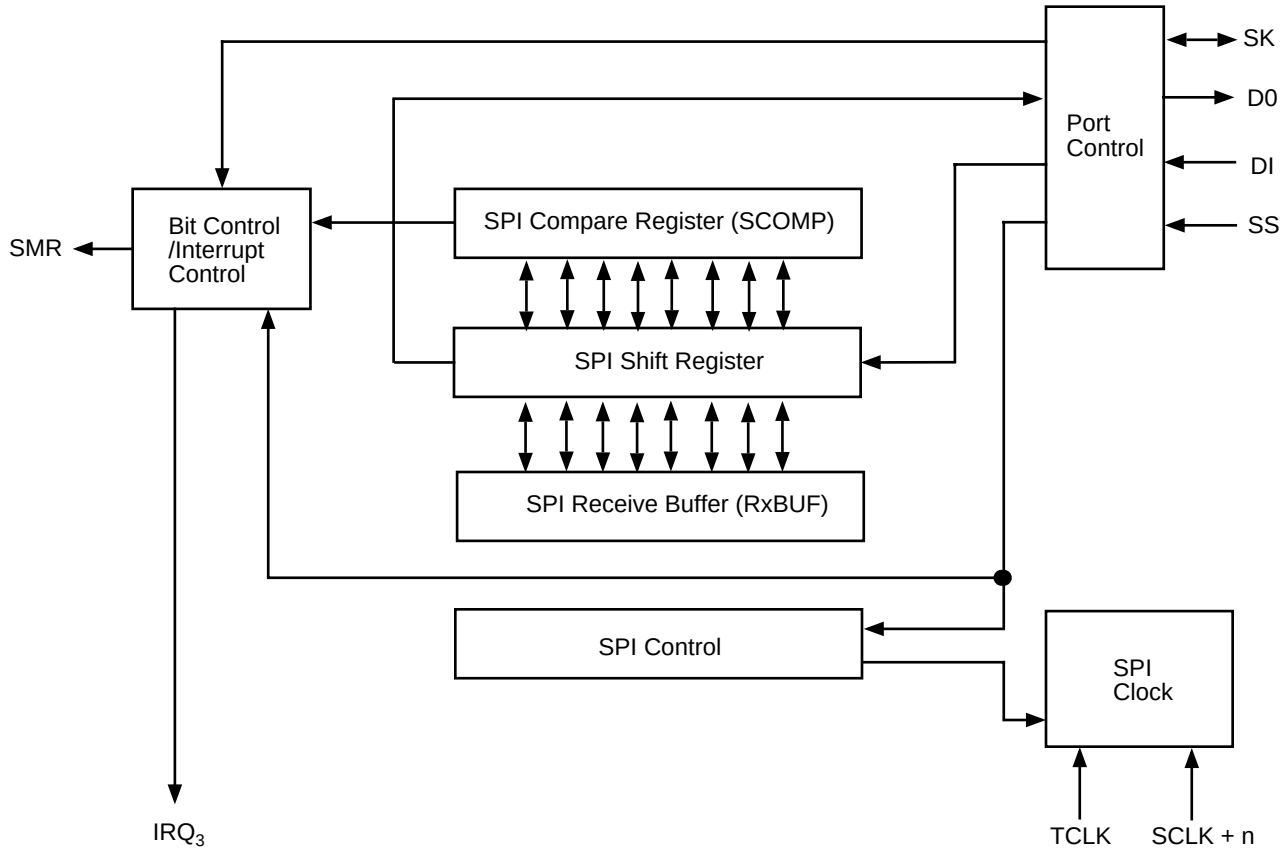

Figure 9-15. SPI Logic

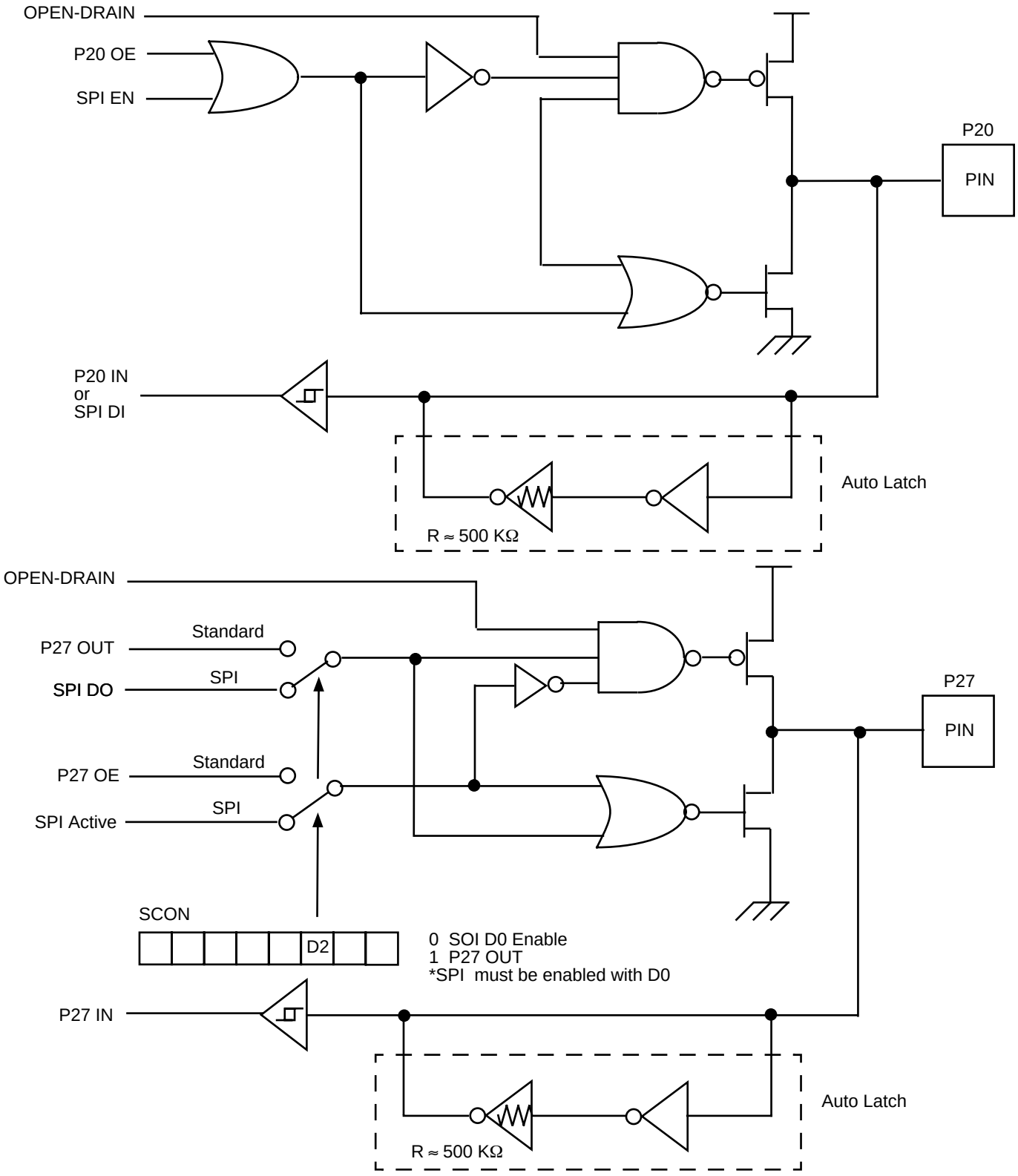

Figure 9-16. SPI Data In/Out Configuration

9.10 RECEIVE CHARACTER AVAILABLE AND OVERRUN (Continued)

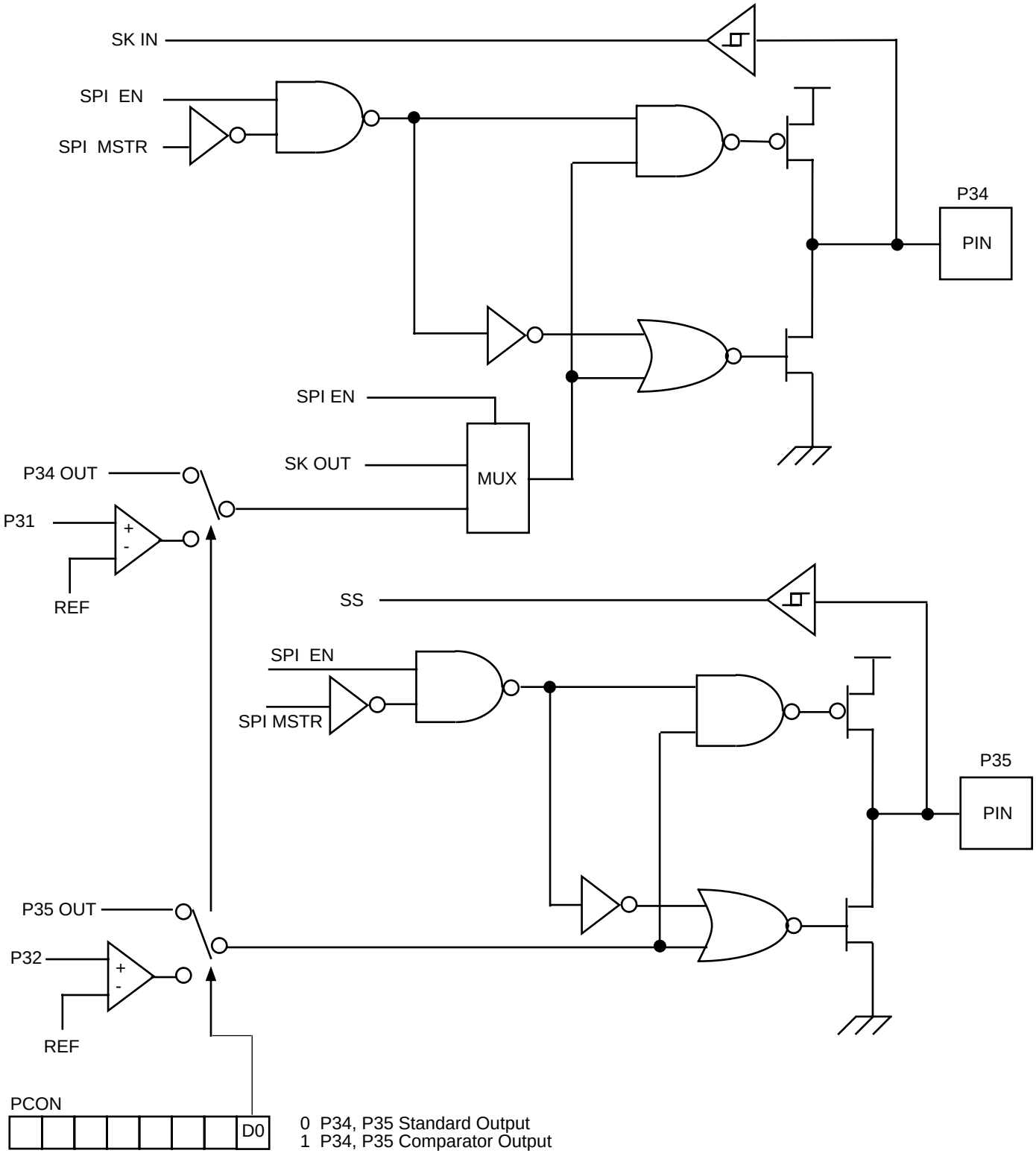

Figure 9-17. SPI Clock / SPI Slave Select Output Configuration