8

8051 SERIAL COMMUNICATION

Learning Objectives

After you have completed this chapter, you should be able to

- Define parallel and serial data transfer
- Define different types of communication links
- Define synchronous and asynchronous serial communication
- Explain the operation of serial port of the 8051 microcontroller
- Program the serial port for data transfer
- Explain the RS232 bus and function of the MAX232 IC

8.1 DATA COMMUNICATION

The 8051 microcontroller is a parallel device that transfers eight bits of data simultaneously over eight data lines to parallel I/O devices. The parallel I/O devices are printer, D/A converter and stepper motor. Interfacing of parallel I/O devices has been discussed in Chapter 6. However, in many situations, parallel data transfer is impractical. For example, parallel data transfer over a long distance is very expensive. Hence, serial data communication is widely used in long distance data communication, and in this mode, one bit of information, at a time is transferred over a single line.

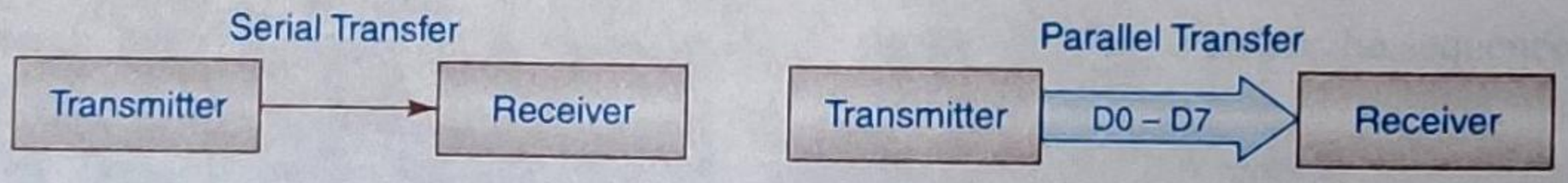

Figure 8.1 *Serial and parallel data transfer*

8.2 BASICS OF SERIAL DATA COMMUNICATION

In serial data communication, 8 bit data is converted to serial bits using a parallel in serial out shift register, and then it is transmitted over a single data line. The data byte is always transmitted with least significant bit first.

8.2.1 COMMUNICATION LINKS

Serial communication is classified into three types of communication links as shown in Fig. 8.2.

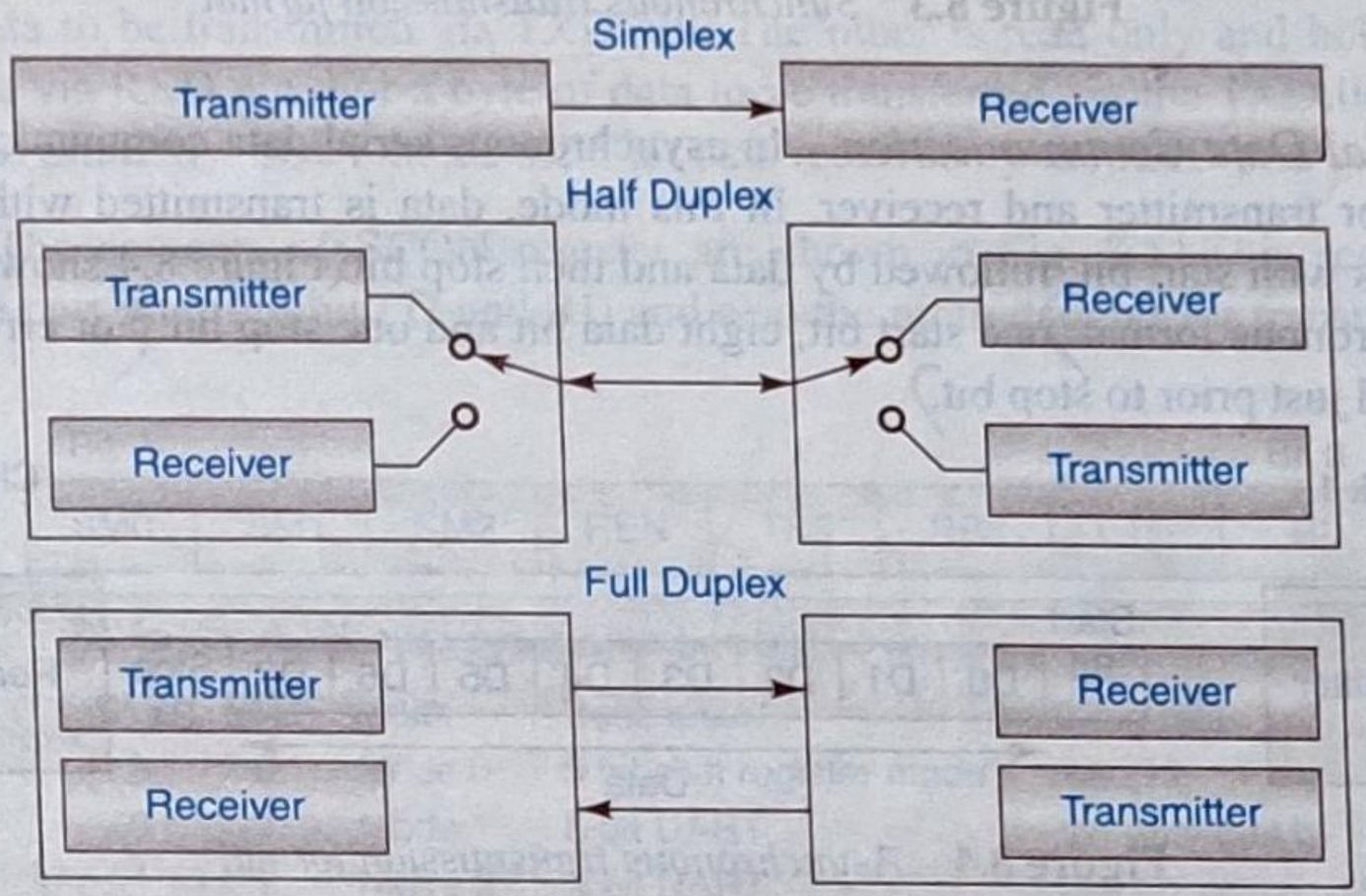

Figure 8.2 *Simplex, Half duplex and Full duplex data transfer*

Simplex In simplex transmission, the line is dedicated for transmission. The transmitter sends and the receiver receives the data..

Half duplex In half duplex, the communication link can be used for either transmission or reception. Data is transmitted in only one direction at a time.

Full duplex If the data is transmitted in both ways at the same time, it is a full duplex, i.e. transmission and reception can proceed simultaneously. This communication link requires two wires for data, one for transmission and one for reception.

8.2.2 TYPES OF SERIAL DATA COMMUNICATION

Serial data communication uses two types of communication

- Synchronous serial data communication
- Asynchronous serial data communication

Synchronous Serial Data Communication In synchronous serial data communication, transmitter and receiver are synchronised. It uses a common clock signal to synchronise the receiver and the transmitter, as

shown in Fig. 8.3. The figure shows the transmission of data; first the sync character and then, the data is transmitted. This format is generally used for high-speed transmission.

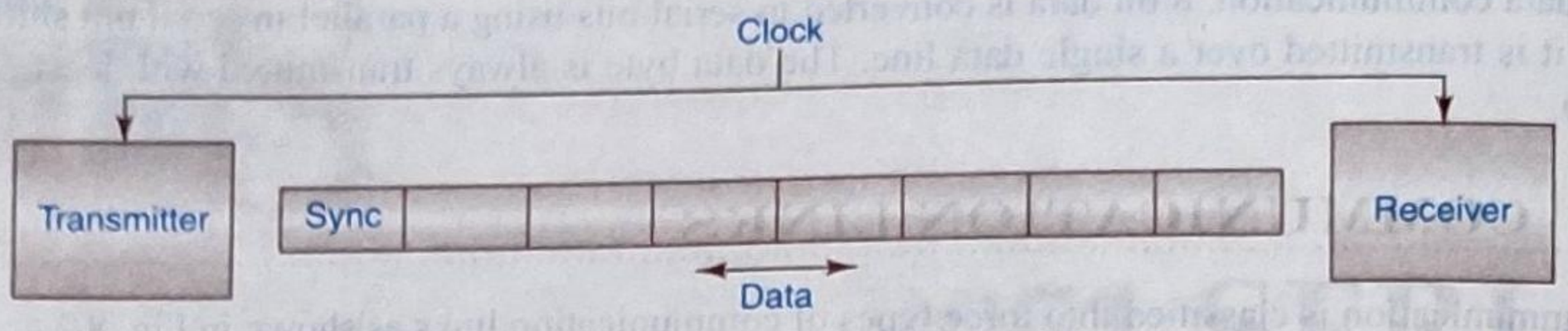

Figure 8.3 *Synchronous transmission format*

Asynchronous Serial Data Communication In asynchronous serial data communication, different clock sources are used for transmitter and receiver. In this mode, data is transmitted with start and stop bits. Transmission begins with start bit, followed by data and then stop bit. Figure 8.4 shows the transmission of 10 bit in the asynchronous format: one start bit, eight data bit and one stop bit. For error checking purpose, parity bit is included just prior to stop bit.

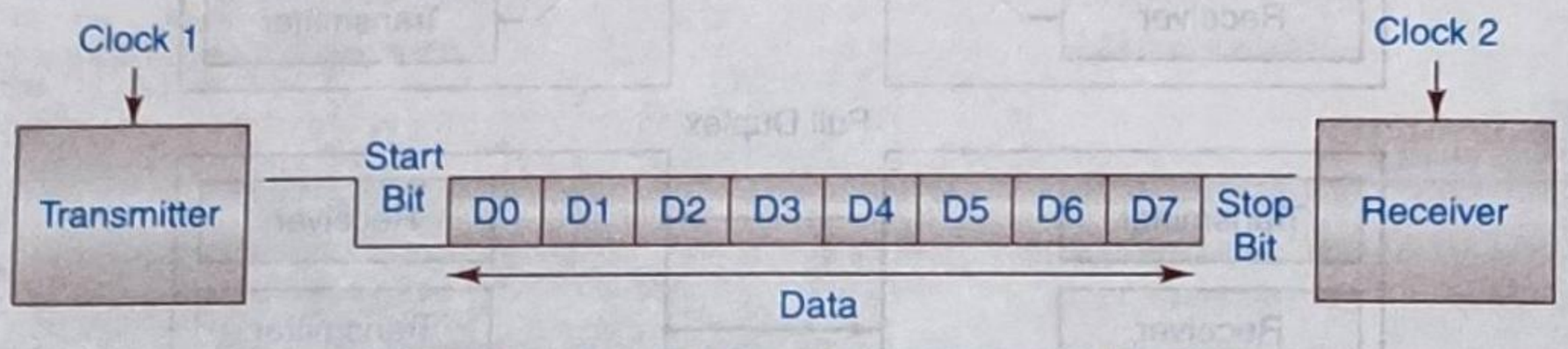

Figure 8.4 *Asynchronous transmission format*

8.2.3 BAUD RATE

The rate at which the bits are transmitted (bits/second) is called *baud or transfer rate*. The baud rate is the reciprocal of the time to send 1 bit. In asynchronous transmission, baud rate is not equal to number of bits per second. This is because, each byte is preceded by a start bit and followed by parity and stop bit. For example, in synchronous transmission, if data is transmitted with 9600 baud, it means that 9600 bits are transmitted in one second. For one bit, transmission time = 1 second/9600 = 0.104 ms.

SECTION REVIEW

1. The 8051 supports parallel and serial data transfer. True/False?
2. In serial data transfer, _____ bit is transmitted first.
3. In half duplex, data is transmitted in only one direction at a time. True/False?
4. If transmission and reception are done simultaneously, then the serial communication is called _____.
5. Mention the advantages and disadvantages of parallel data communication.
6. List the types of serial data communication.
7. In _____ serial communication, first sync character and then the data is transmitted.
8. In asynchronous communication, _____ bit is included for error checking purpose.
9. The rate at which the bits are transmitted is called _____.
10. In asynchronous transmission, the baud rate is not equal to number of bits/second. True/False?

8.3 8051 SERIAL COMMUNICATION

The 8051 supports a full duplex serial port. Full duplex means that it can transmit and receive a byte simultaneously. The 8051 has TXD (pin 11 or P3.1) and RXD (pin 10 or P3.0) pins for transmission and reception of serial data respectively. These pins are TTL compatible. The 8051 transfers and receives data serially with different baud rates. Three special function registers support serial communication, namely, SBUF Register, SCON Register and PCON Register.

SBUF Register SBUF is an 8 bit register. It has separate SBUF registers for data transmission and for data reception. These two registers are accessed by the same name, SBUF. One of the registers is write only and used to hold data to be transmitted via TXD pin. The other is read only and holds the received data from external source via RXD pin. For a byte of data to be transferred via the TXD line, it must be placed in the SBUF register. Similarly, SBUF holds the 8 bit data received by the RXD pin.

SCON Register The contents of SCON register are shown in Fig. 8.5. This register contains mode selection bits, serial port interrupt bit (TI and RI) and also the ninth data bit for transmission and reception (TB8 and RB8).

Bit 7 Bit 0

SM0	SM1	SM2	REN	TB8	RB8	TI	RI

SM0 SM1	Serial communication mode selection bits				
	SM0	SM1	Mode	Description	Baud rate
	0	0	Mode 0	8 bit shift register mode	fosc/12
	0	1	Mode 1	8 bit UART	Variable (set by timer 1)
	1	0	Mode 2	9 bit UART	fosc/32 or fosc/164
	1	1	Mode 3	9 bit UART	variable (set by timer 1)
SM2	In modes 2 and 3, if set, this will enable multiprocessor communication				
REN	Enables serial reception				
TB8	This is the 9th data bit that is transmitted in modes 2 and 3				
RB8	9th data bit that is received in modes 2 and 3, it is not used in mode 0 In mode 1, if SM2 = 0, then RB8 is the stop bit that is received				
TI	Transmit interrupt flag, set by hardware must be cleared by software				
RI	Receive interrupt flag, set by hardware must be cleared by software				

Figure 8.5 *Serial Control Register (SCON)*

PCON Register The smod bit (bit 7) of PCON register controls the baud rate in asynchronous mode transmission.

8.4 SERIAL COMMUNICATION MODES

The serial port can operate in four modes. SM0 and SM1 are D7 and D6 bits of SCON register, and these two bits determine the 4 serial modes. Serial port runs in both synchronous and asynchronous mode.

Mode 0 In mode 0, the serial port runs in synchronous mode. In this mode, data is transmitted and received by RXD pin, and TXD pin is used for clock output. In this mode, processor clock is used and the baud rate is 1/12 of the oscillator frequency. Eight bits are transmitted/received with least significant bit first. Figure 8.6 shows the timing for mode 0 data transmission. This mode is also called shift register mode.

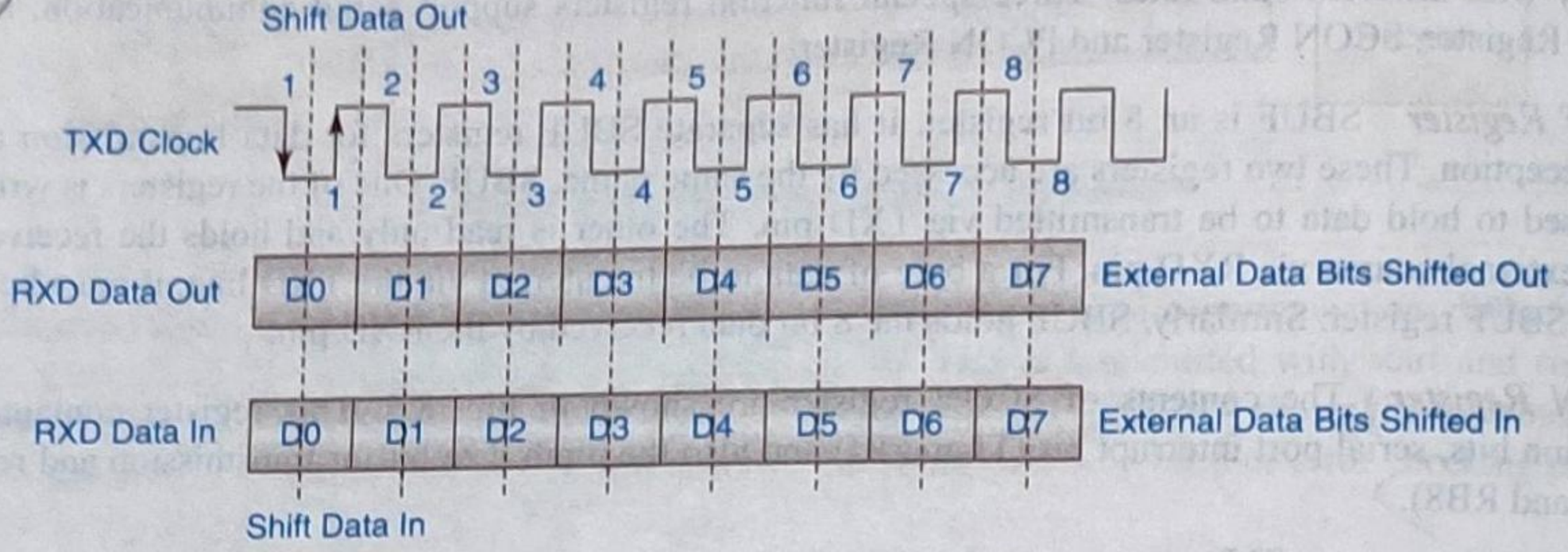

Figure 8.6 *Mode 0 Timing diagram*

In the remaining three modes, the data transmission is asynchronous and provides two different ways of clocking as well as 8 and 9 bit data transfer.

Mode 1 In this mode, SBUF becomes a 10 bit full duplex receiver/transmitter that may receive and transmit data at the same time. In this mode, ten bit are transmitted or received—1 start bit, 8 data bits and 1 stop bit. The interrupt flag TI is set, once all the ten bits have been sent. On reception, the stop bit goes into bit RB8 of the SCON register. The baud rate is variable and is determined by Timer 1 overflow rate. The value of SMOD of PCON register is as follows

$$\text{Baud rate} = [\,2^{\text{smod}}/32] \times [\text{Timer 1 overflow rate}]$$

Timer 1 interrupt should be disabled and is configured in auto-reload mode. In this case, if upper nibble of TMOD register is loaded with value 2H, then the baud rate is given by the formula.

$$\text{Baud rate} = [2^{\text{smod}} / 32] \times [\text{oscillator frequency}] / [12 \times [256 - [\text{TH1}]]]$$

For example, if TH1 contents are 230d, SMOD bit in PCON is 0 and if 12 MHz is the oscillator frequency, then the baud rate is 1201. To get exactly 1200 baud, the oscillator frequency must be 11.059 MHz. Table 8.1 lists the baud rates and how they can be obtained from Timer 1.

Mode 2 Mode 2 is similar to mode 1 except that eleven bits are transmitted or received—1 start bit, 8 data bits, a programmable ninth data bit, and 1 stop bit. During transmission, the ninth data bit is copied from bit TB8 of SCON and on reception, the ninth data bit goes into bit RB8 of the SCON register, while stop bit is ignored. The baud rate is programmable to either 1/32 or 1/64 of the oscillator frequency and depends on the SMOD bit of PCON register.

$$\text{Baud rate} = [2^{\text{smod}} / 64] \times [\text{oscillator frequency}]$$

Page 193

TABLE 8.1 *Timer 1 generated commonly used baud rates*

Baud Rate	fosc	SMOD	C/T	Timer 1 Mode	Timer 1 Reload Value
Mode 0, Max: 1000K	12 MHz	X	X	X	X
Mode 2, Max: 375K	12 MHz	1	X	X	X
Modes 1,3:62.5K	12 MHz	1	0	2	FFH
19.2K	11.059 MHz	1	0	2	FDH
9.6K	11.059 MHz	0	0	2	FDH
4.8K	11.059 MHz	0	0	2	FAH
2.4K	11.059 MHz	0	0	2	F4H
1.2K	11.059 MHz	0	0	2	E8H
137.5	11.986 MHz	0	0	2	1DH
110	6 MHz	0	0	2	72H
110	12 MHz	0	0	1	FEEBH

Mode 3 Mode 3 is identical to mode 2 except that the baud rate is determined as in mode 1. In mode 3, eleven bits are transmitted or received, a start bit, 8 data bits, a programmable ninth data bit and a stop bit.

SECTION REVIEW

1. Name the SFRs that support serial communication.
2. _____ pin is used for transmission and _____ pin is used for reception in 8051 serial communication.
3. The 8051 has two SBUF registers. True/False?
4. _____ register controls the baud rate in asynchronous mode of transmission.
5. 9 bit data is transmitted in mode 2 and mode 3 using TB8 bit of SCON register. True/False?
6. _____ bit of SCON enables serial reception.
7. _____ interrupt flag is used during serial transmission.
8. Serial interrupt has lowest priority in 8051 interrupts. True/False?
9. In mode 0 of serial communication, baud rate is Fosc/12. True/False?
10. If SMOD bit of PCON is zero, then baud rate in mode 2 of serial communication is oscillator frequency/64. True/False?

8.5 SERIAL COMMUNICATION PROGRAMMING

EXAMPLE 8.1

Write a subroutine to initialise 8051 serial port to operate with the following parameters:

- Disable interrupt for transmission and receiving
- Baud rate 9600
- One start bit, eight data bits and one stop bit; and to enable receiving and transmission

```
Step 1: Disable transmit and receive interrupt, clear bit 4 of IE register
Step 2: To set baud rate = 9600, the following parameters are selected
        a. Use 11.059 MHz crystal oscillator
        b. Choose mode 2 operation for timer 1
        c. Load 0010xxxx b to TMOD register
        d. Load reload value FD H to TH1 and TL1
        e. Clear SMOD bit of the PCON register
Baud rate = [2^smod / 32] x [oscillator frequency] / [12 x (256 - (TH1))]
Step 3: Value 01010000 b is written in SCON to operate serial port
in mode 1 and transmission and receiver is enabled.
CLR IE.4          ; Disable serial port interrupt
MOV TMOD,#20H     ; Choose mode 2 operation for timer 1
MOV TL1,#0FDH     ; Set the initial value
MOV TH1,#0FDH     ; Set the reset value
MOV PCON,#7FH     ; Clear MSB of PCON
SETB TR1          ; Start timer 1
MOV SCON,#50H     ; Operate serial port in mode 1
RET
```

EXAMPLE 8.2

Write a subroutine to initialise 8051 serial port to operate in mode 0 for transmission.

```
CLR IE.4              ; Disable serial port interrupt
MOV SCON, #00H        ; Operate serial port in mode 0
MOV SBUF, #44H        ; Load SBUF
CLR TI                ; Clear TI bit
LOOP: JNB TI, LOOP    ; Wait for the last bit to transfer
RET
```

EXAMPLE 8.3

Write a subroutine to initialise 8051 serial port to operate in mode 1 for transmission and timer 1 in auto-reload mode.

```
CLR IE.4              ; Disable serial port interrupt
MOV TMOD, #20H        ; Choose mode 2 operation for timer 1
MOV SCON, #40H        ; Operate serial port in mode 1
MOV TL1, #0FDH        ; Set the initial value
MOV TH1, #0FDH        ; Set the reset value
SETB TR1              ; Start timer 1
MOV SBUF, #56H        ; Load SBUF
CLR TI                ; Clear TI bit
LOOP: JNB TI LOOP     ; Wait for last bit to transfer
RET
```

EXAMPLE 8.4

Write a program to initialise 8051 serial port to operate in mode 1 for receiving a serial byte through RXD pin, send the received data to port 2 and operate timer 1 in auto reload mode.

```
CLR IE.4                  ; Disable serial port interrupt
MOV TMOD, #20H            ; Choose mode 2 operation for timer 1
MOV SCON, #50H            ; Operate serial port in mode 1
MOV TL1, #0FDH            ; Set the initial value
MOV TH1, #0FDH            ; Set the reset value
SETB TR1                  ; Start timer 1
AGAIN: CLR RI             ; Clear RI bit
LOOP1: JNB RI, LOOP1      ; Wait for last bit to receive
MOV A, SBUF               ; Save incoming data in A
MOV P2, A                 ; Send to port 2
SJMP AGAIN                ; Keep getting data
```

EXAMPLE 8.5

An 8051 micrcontroller has an oscillator frequency of 11.0592 MHz. Using timer 1 and configuring the UART in mode 1, write a C program that transmits ASCII character D at a baud rate of 9600.

```
#include <Intel\8051.h>
# main ( ) {                          //Start the program
          SCON = 0x42;                //Serial mode 1
          TMOD = 0x20;                // Timer 1 in mode 2
          TH1 = 0xFA;                 // Baud rate = 9600
          TL1 = 0xFA;
          TR1 = 1;                    //Start Timer1
          While (1) {
                SBUF = 'D';     // Load 'D' into serial buffer
                While ( !TI );// Wait for completion of transmission
                TI = 0;               // Clear transmission flag
          }                           // While (1)
}                                     // End of the program
```

8.6 RS232

RS232 is the most widely used serial I/O interfacing standard. The RS232 standard was published by the Electronic Industry Association (EIA) in 1960. The COM1 and COM2 ports in IBMPC are RS232 compatible ports. In RS232, 1 is represented by –3 to –25 V and 0 is represented by +3 to +25 V. In a microcontroller, serial TXD and RXD lines are TTL compatible i.e. 1 and 0 are represented by +5 V and 0 V. For this reason, in order to connect a microcontroller to RS232 bus, voltage converters are used. MAX 232 IC is commonly used to convert the TTL logic levels to the RS232 voltage levels. The significance of the number 232 is that 2 is transmission line, 3 is receiving line, and 7 (2+3+2) is signal ground line. In RS232, ground line is common to the transmitter and receiver, and they are usable up to one meter without any shield.

8.6.1 RS232 PLUG CONNECTORS

Basic data communication link is shown in Fig. 8.7. The communication link consists of Data Terminal Equipment (DTE) and an associated modem (DCE) at each end. The function of modem is to process digital information received from the computer into a form suitable for analog transmission. Also, it receives analog signal and processes it into digital information.

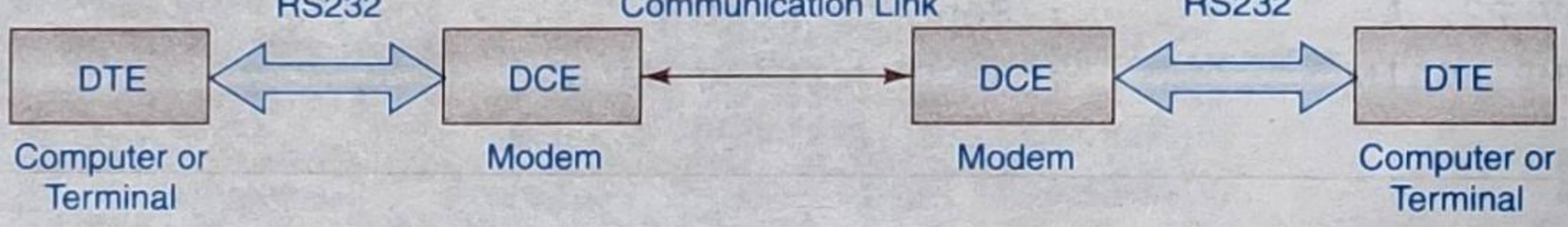

Figure 8.7 *Data communication system*

RS232 uses a 25 pin plug connector for all interface circuits and is commonly referred to as the DB-25 pin connector. DB-25P refers to the plug connector (male) and DB-25S refers to the socket connector (female). Since all the 25 pins are not used in PC, IBM introduced DB-9 connector. These plug connectors are shown in Fig. 8.8

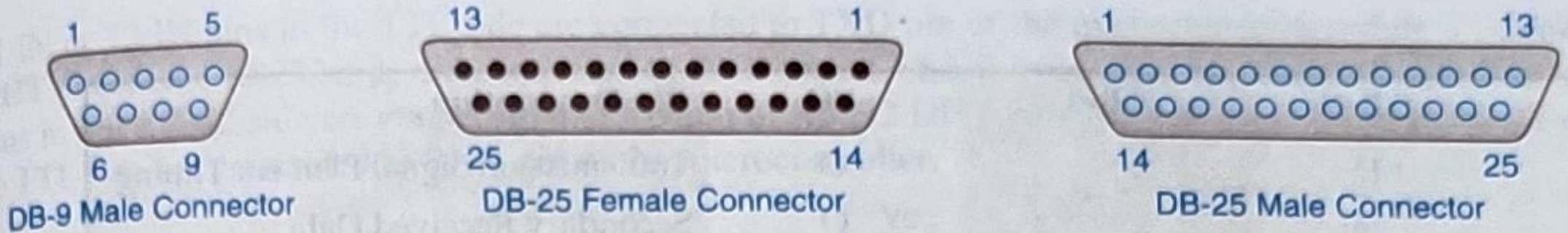

Figure 8.8 *DB-25 and DB-9 connectors*

A microcontroller with minimum three lines—TXD, RXD and ground—can be connected to another microcontroller as shown in Fig. 8.9. The remaining pins in the connector are used for handshaking signals. In this section, the functions of important signals in RS232 bus will be discussed.

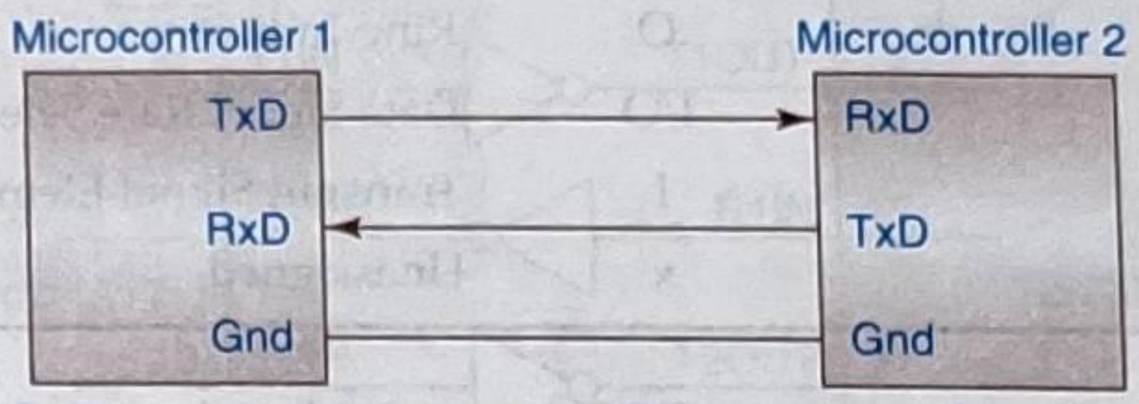

Figure 8.9 *Interface of two microcontrollers with minimum signals*

TABLE 8.2 *RS232 pins (DB-25 and DB-9)*

DB-25	DB-9	I/O	Pin Description
1		x	Protective Ground
2	3	I	Transmitted Data
3	2	O	Received Data
4	7	I	Request To Send
5	8	O	Clear To Send
6	6	O	Data Set Ready
7	5	x	Signal Ground
8	1	O	Received Line Signal Detector
9		x	Reserved For Data Set Testing
10		x	Reserved For Data Set Testing
11		x	Unassigned
12		O	Secondary Rcvd Line Signal Detector
13		O	Secondary Clear To Send
14		I	Secondary Transmitted Data

(Contd)

(*Contd*)

DB-25	DB-9	I/O	Pin Description
15		O	Transmission Signal Element Timing
16		O	Secondary Received Data
17		O	Receiver Signal Element Timing
18		x	Unassigned
19		I	Secondary Request To Send
20	4	I	Data Terminal Ready
21		O	Signal Quality Detector
22	9	O	Ring Indicator
23		I/O	Data Signal Rate Selector
24		I	Transmit Signal Element Timing
25		x	Unassigned

Transmitted Data Data Terminal Equipment (DTE) transmits data through this pin.

Received Data Data Terminal Equipment (DTE) receives data through this pin.

Signal Ground This circuit establishes a common ground reference potential for all interface circuits.

Data Terminal Ready (DTR) When Data Terminal Equipment (DTE) is turned on, it sends active low signal DTR and indicates that DTE is ready for communication. If DTE is not ready for communication, then this signal is not activated.

Data Set Ready (DSR) When modem is turned on, it sends an active low signal DSR and indicates that it is ready for communication.

Request To Send (RTS) DTE asserts this signal to its associated DCE (modem) when it has data to transmit. RTS is an active low output from DTE and an input to the modem.

Clear To Send (CTS) In response to RTS, the modem sends out signal CTS to DTE and indicates that it is ready to receive the data.

Data Carrier Detect (DCD) DCD is an output from the modem (DCE) and an input to DTE. The modem asserts signal DCD to inform DTE that a valid carrier has been detected.

Ring Indicator (RI) RI is an output from the modem (DCE) and an input to DTE. This signal indicates that the telephone is ringing and is used when DTE is in charge of answering the phone.

8.6.2 MAX232

The 8051 microcontroller has two pins TXD and RXD, specifically used for transmitting and receiving data serially. In microprocessors and microcontrollers, the pins are TTL compatible. Therefore, it requires a line driver such as MAX232 IC to convert TTL levels to RS232 voltage levels, and vice versa.

MAX232 MAX232 is 16 pin IC and requires +5 V power supply. It converts RS232 voltage levels to TTL voltage levels, and vice versa. As shown in Fig. 8.10, it has two sets of line drivers for transferring and receiving the data.

T1 IN or T2 IN pins in the TTL side are connected to TXD pin of the microcontroller, while T1 OUT or T2 OUT pins in the RS232 side are connected to RXD pin of RS232 DB connector. Similarly, R1 IN or R2 IN pins in the RS232 side are connected to TXD pin of RS232 DB connector and R1 OUT or R2 OUT pins in the TTL side are connected to RXD pin of the microcontroller.

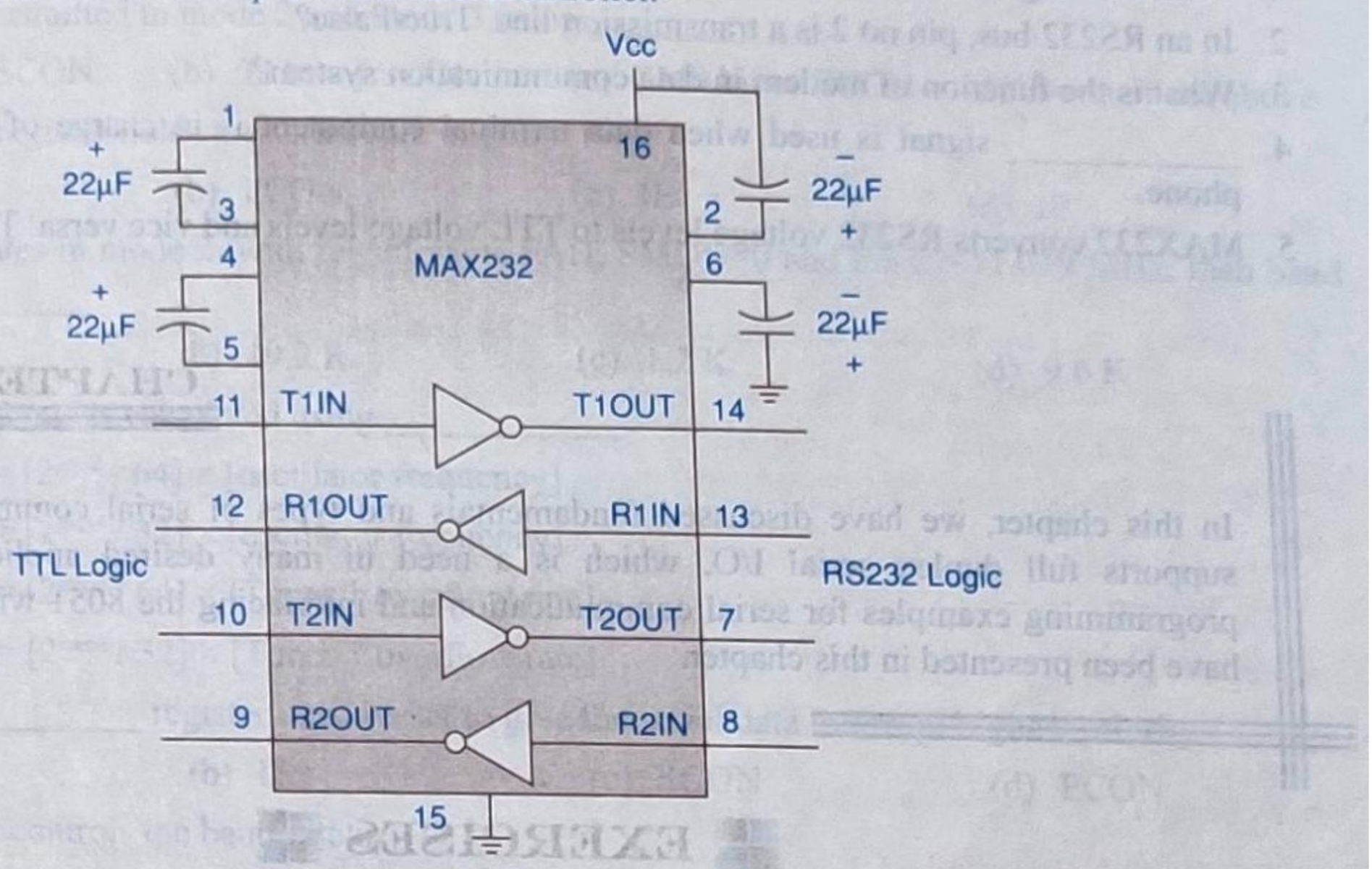

Figure 8.10 *MAX232 pin diagram*

MAX233 IC is a 20 pin IC and requires +5 V. MAX233 IC also converts RS232 voltage levels to TTL voltage levels and vice versa. As shown in Fig. 8.11, external capacitors are not required in MAX233, but it is expensive compared to MAX232.

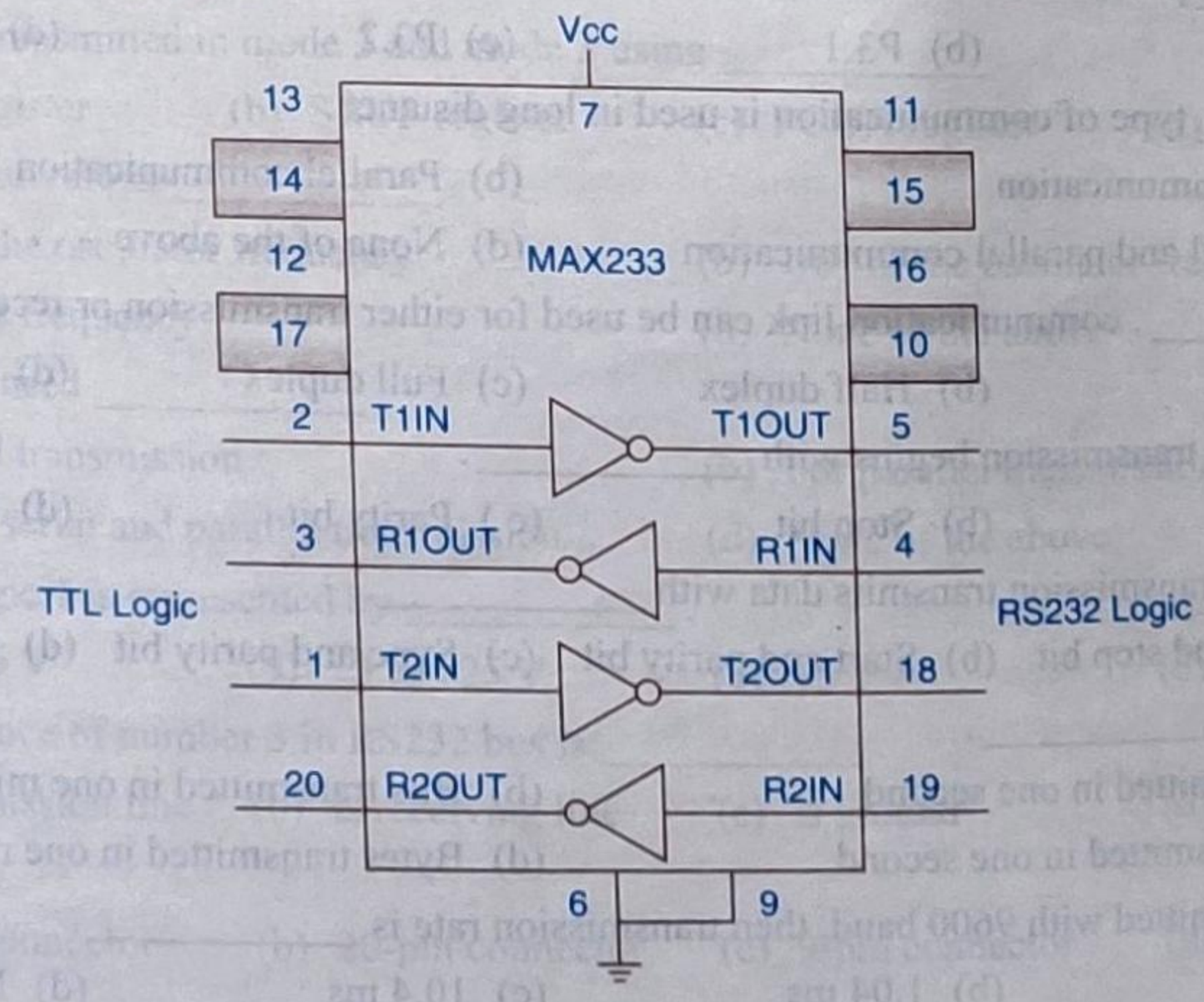

Figure 8.11 *MAX233 pin diagram*