Instrumentation II

Chapter 3
Serial Interfacing with Microprocessor Based System

BCT III/I

Dr. Raksha Dangol,
Assistant Professor,
Advanced College of Engineering and Management

Chapter - 3 Serial Interfacing with Microprocessor Based System

3.1 Advantages of Serial Data Transfer Over Parallel

3.2 Synchronous and Asynchronous Data Transfer

3.3 Errors in Serial Data Transfer

3.4 Simplex, Half Duplex and Full Duplex Data Communication

3.5 Parity and Baud Rates

3.6 Introduction Serial Standards RS232, RS423, RS422

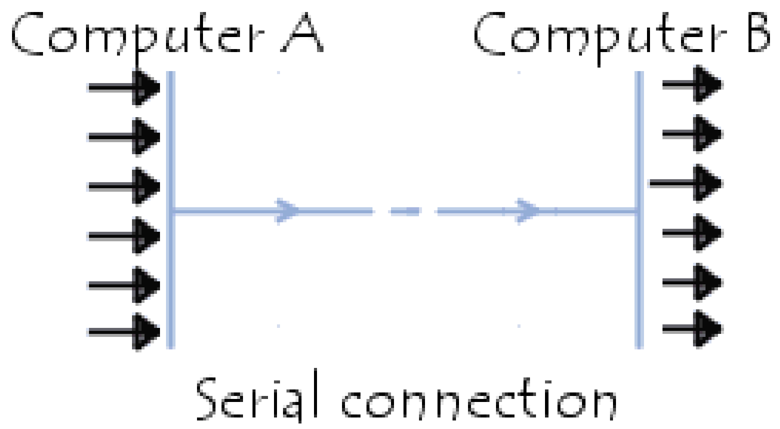
3.7 Universal Serial Bus

- 3.7.1 The Standards:- USB 1.1 and USB 2.0
- 3.7.2 Signals, Throughput & Protocol
- 3.7.3 Devices, Hosts and On-The-Go
- 3.7.4 Interface Chips: USB Device and USB Host

Introduction: Serial Data Transmission

Serial transmission data is

- sent in a serial form i.e. bit by bit on a single line.
- cost of communication hardware is considerable reduced since only a single wire or channel is require for the serial bit transmission.
- slow as compared to parallel transmission. Serial data can be sent synchronously or asynchronously.



Most processors process data in parallel, so the transmitter needs to transform incoming parallel data into serial data and the receiver needs to do the opposite.

Introduction

Within a microcomputer data is transferred in parallel, because that is the fastest way to do it. For transferring data over long distances, however, parallel data transmission requires too many wires. Therefore, data to be sent long distances is usually converted from parallel form to serial form so that it can be sent on a single wire or pair of wires. Serial data received from a distant source is converted to parallel form so that it can easily be transferred on the microcomputer buses.

Advantages of Serial Data Transfer Over Parallel

- Longer data transmission in serial mode
  - Serial; $1 \rightarrow$ -3V to -25V
    $0 \rightarrow$ +3V to +25V
  - Parallel; $1 \rightarrow$ +5V
    $0 \rightarrow$ 0V
  - Voltage loss is not much a problem in serial communication.
- Serial transmission requires less number of wires than parallel and so cheaper to transmit data.
- Crosstalk is less of an issue because there are fewer conductors' compared to that of parallel cables.
- Many IC and peripherals have serial interface
- Clock skew between different cables is not an issue
- Serials can be clocked at higher data rate
- Serial cable can be longer than parallel

But in serial mode of transfer, only one bit of a word is transferred at a time so that data transfer rate is very slow; it is the one of the demerit over parallel data transfer.

Serial Synchronous Data Transmission

In serial synchronous data transmission,

- Data is transmitted or received based on a clock signal.
- Transmitting device sends a data bit at each clock pulse at a specific rate of data transmission. It must know the number of data units to be transferred
- Receiving device must know the start and end of each data unit in order to interpret the data correctly. It must be synchronized with the data boundaries.
- There must be synchronization between the transmitter and receiver. Usually one or more SYNC characters or a unique bit pattern called a flag are used to indicate the start of each synchronous data stream or frame of data.
- After receiving, serial data needs to be convertd to parallel form so they can be read in by a computer.

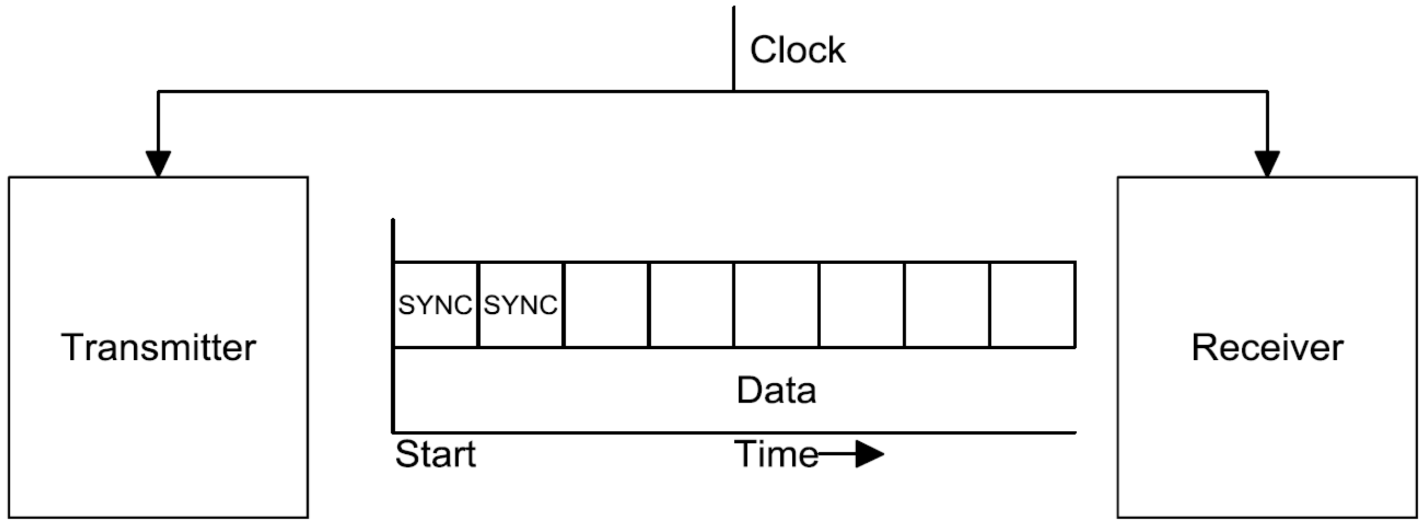


Fig: Synchronous Serial Transmission Format

Advantages and Disadvantages of Synchronous Communication

Main advantage of Synchronous data communication is the high speed. The synchronous communications require high-speed peripherals/devices and a good-quality, high bandwidth communication channel. Other advantages are:

Timing information is accurately aligned to the received data, allowing operation at much higher data rates. Data transmission is synchronized.

The disadvantage includes the possible in accuracy. Because when a receiver goes out of synchronization, loosing tracks of where individual characters begin and end. Correction of errors takes additional time.

Receiver tracks any clock drift which may arise (for instance due to temperature variation). The penalty is however a more complex interfaces design, and potentially a more difficult interface to configure (since there are many more interface options).

Serial Asynchronous Data Transmission

The receiving device does not need to be synchronized with the transmitting device. The transmitting device can send one or more data units when it is ready to send data. Each data unit must be formatted i.e. must contain start and stop bits for indicating beginning and the end of data unit. It also includes one parity bit to identify odd or even parity of data. To send ASCII character, the framing of data should contain:

- 1 start bit: Beginning of data
- 8 bit character: Actual data transferred
- 1 or 2 stop bits: End of data

When no data is being sent, the signal line is in a constant high or marking state. The beginning of the data character is indicated by the **line going low for 1 bit time** and this bit is called a **start bit**. The data bits are then sent out on the line one after the other where the least significant bit is sent out first. Parity bit should contain to check for errors in received data. After the data bit and a parity bit, the signal line is returned high for at least 1 bit time to identify the end of the character, this **always high bit** is referred to as a **stop bit**. Some older systems use 2 stop bits.

Serial Asynchronous Data Transmission

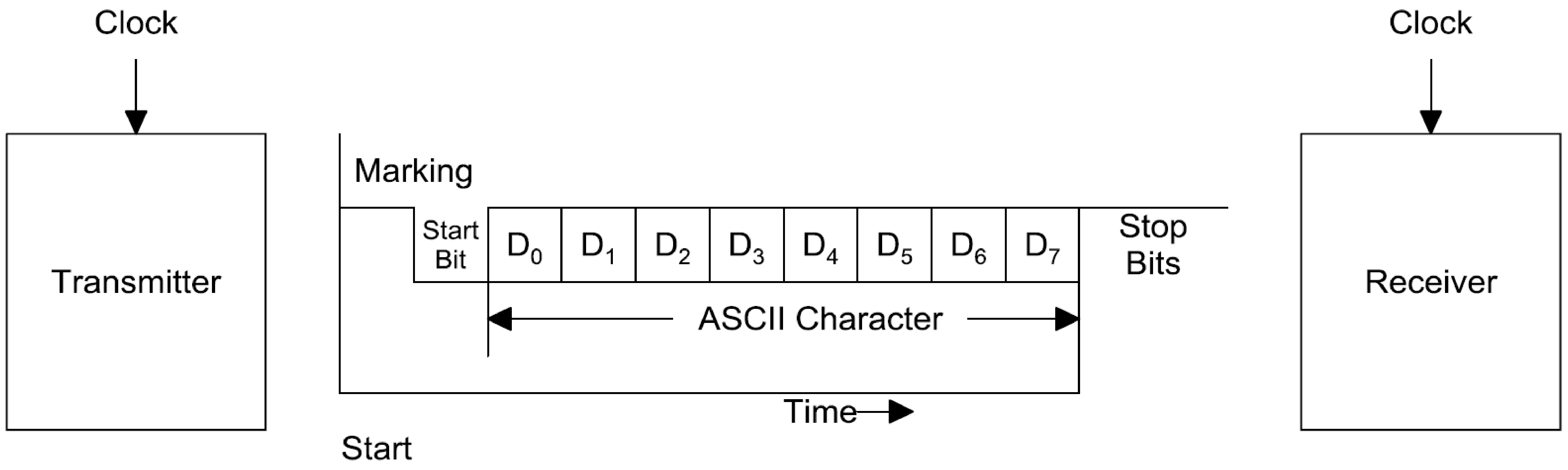


Fig: Asynchronous Serial Transmission Format

Serial Asynchronous Data Transmission

In asynchronous transmission each character is transmitted separately, that is one character at a time. The character (8-bits) is preceded by a start bit (1-bit), which tells the receiving end where the character coding begins, and is followed by a stop bit (1 or 2-bits), which tells the receiver where the character coding ends. There will be intervals of ideal time on the channel shown as gaps. Thus there can be gaps between two adjacent characters in the asynchronous communication scheme. In this scheme, the bits within the character frame (including start, parity and stop bits) are sent at the baud rate.

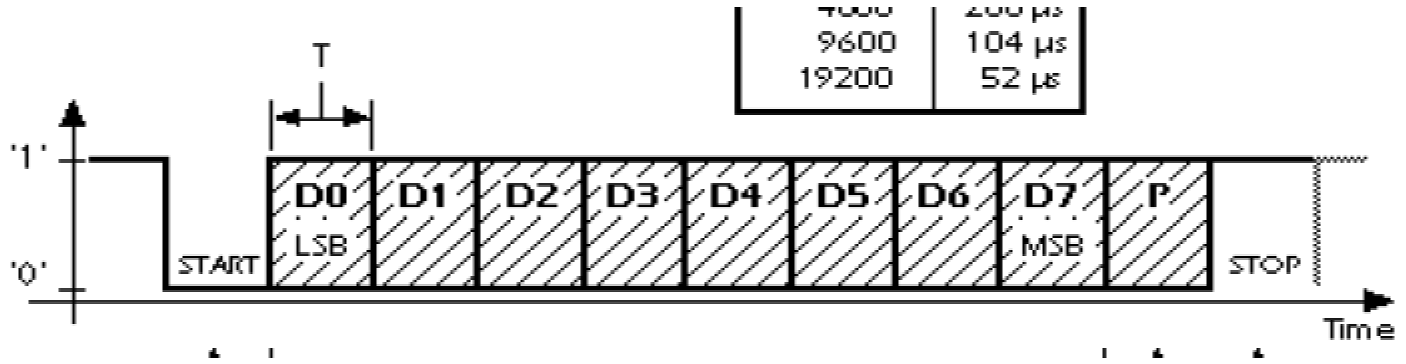
The START bit and STOP bit including gaps allow the receiving and sending computers to synchronize the data transmission. Asynchronous communication is used when slow speed peripherals communicate with the computer. The main disadvantage of asynchronous communication is slow speed transmission. Asynchronous communication however, does not require the complex and costly hardware equipment as is required for synchronous transmission.

Synchronous versus Asynchronous Serial Data Transmission

S.N.	Parameter	Asynchronous	Synchronous
1.	Fundamental	Transmission is not based on clock signal	Transmission based on clock signal
2.	Data Format	One character at a time	Group of characters i.e. a block of characters
3.	Speed	Low (< 20 kbps)	High (> 20 kbps)
4.	Framing Information	Start and stop bits are sent with each character.	SYNC characters are sent with each character.
5.	Implementation	Hardware / Software	Hardware

Serial Data Unit (SDU) & Serialization

- SDU is a unit with 1 start bit, 8 data bits, 1 parity bit and 1 or 2 stop bits.
- Start bit always has a value of 0 & stop bits always have a value of 1.
- Following figure shows a SDU format; for asynchronous data transmission, sender and receiver must be set up to the same format.



9600	104 µs
19200	52 µs

Fig: SDU or frame format

Transmitting SDU

- The interface chip has a transmitter hold register for transmitting data which first fetches the data bytes from CPU.
- According to the selected data format, the SDU logic puts the start bit in front of data bits; it then calculates the parity bit and appends it together with the stop bits to the data bits.
- Thus formed SDU is transferred into the transmitter shift register, which is operated by a clock source determined by baud rate and thus provides the individual bits at the serial output (LSB first). If no data, then the chip possesses a logical

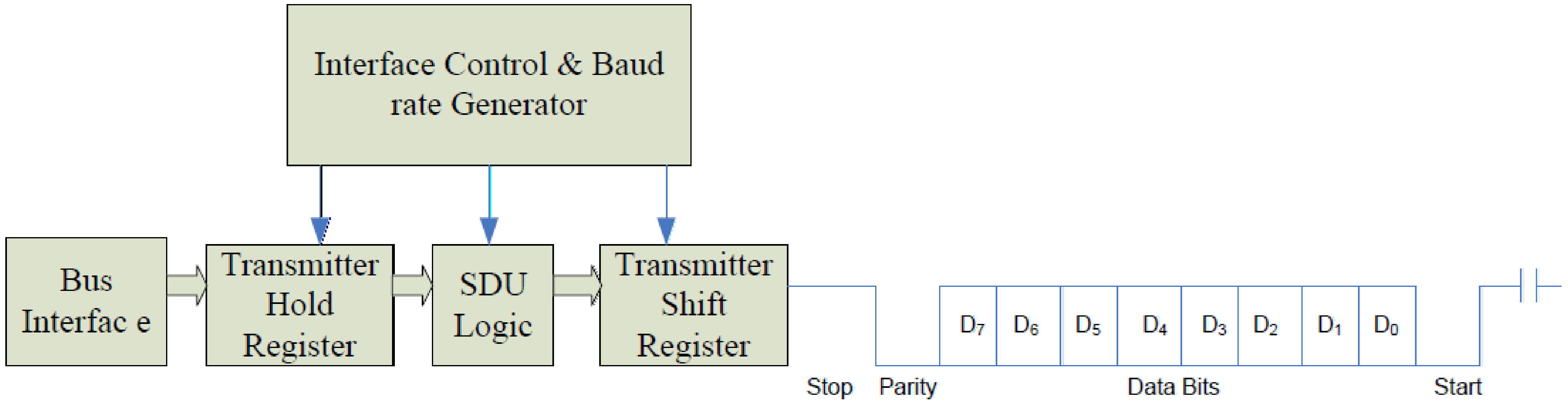

Interface Control & Baud rate Generator
Bus Interfac e
Transmitter Hold Register
SDU Logic
Transmitter Shift Register
D7
D6
D5
D4
D3
D2
D1
D0
Stop
Parity
Data Bits
Start

Fig: SDU at transmitter side

Receiving SDU

- Inverse reception process
- Start bit acts as trigger pulse & starts the receiver in the serial input chip.
- The SDU bits are loaded into the receiver shift register according to the phase of the setup baud rate.
- The receiver SDU logic then separates the start, parity, stop bits from the received SDU bits, calculates the parity of the data bits & compares it with the setup parity.
- Afterwards, the extracted data bits are transferred into the receiver buffer register from which they may be read out as the received data byte by the CPU.

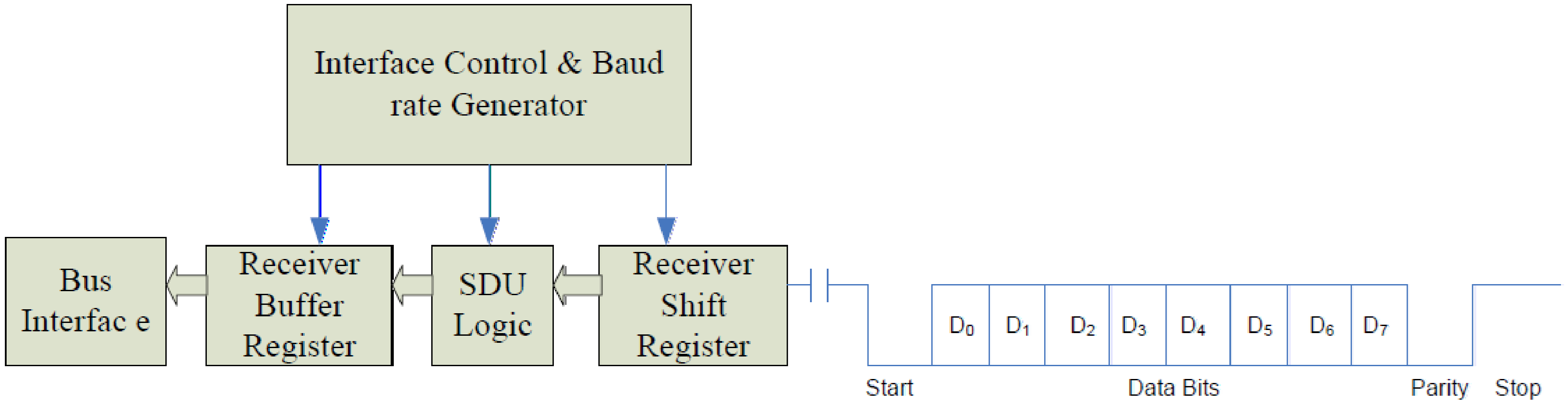


Fig: SDU at receiver Side

Errors in Serial Data Transfer

- In the transmission and reception process, transmitter and receiver must be set to the same baud rate. Additionally, the set data formats (i.e. number of data bits, parity, start and stop bits) must also coincide; otherwise the receiver may resemble possibly a different byte from that which the transmitter was passed for transmitting. Upon reception of an SDU, various errors may occur.

1. Framing Error : Data does not fit in frame that data format and baud rate defined i.e. non-synchronized start / stop bit. Eg:- Change in no. of bits in receiving and transmitting end.

2. Break Error : If the reception line is at logic low level for longer time than the SDU usually lasts, then the receiver assumes that the connection to the transmitter has broken. Unless the transmitter drives the line to a logical high level, no data is transferred.

3. Overrun Error : If data arriving at the receiver is much faster than it can be read from the receiver buffer; the latter received byte overwrites the older data in the buffer.

4. Parity Error : The calculated parity does not coincide with the set one. It may be due to the noise or a different set for parity at transmitter and receiver sides.

- No parity
- Even parity
- Odd parity
- Mark parity
- Space parity

Error Checks in Data Communication

- During transmission, various types of errors can occur such as data bits may change because of noise or can be misunderstood by the receiver due to different clocks between transmitter and receiver. These errors need to be checked; therefore, additional information for error checking is sent during the transmission. The receiver can check the received data against the error check information, and if an error is detected, the receiver can request the retransmission of that data segment or it can correct by proper coding techniques. Three methods are generally in common practice; they are parity check, checksum and cyclic redundancy check.

1. Checksum

The checksum technique is used when blocks of data are transmitted. It involves adding all the bytes in a block without carriers. Then, the 2's complement of the sum (negative of the sum) is transmitted as the last byte. The receiver adds all the bytes, including the 2's complement of the sum; thus, the result should be zero if there is no error in the block.

Error Checks in Data Communication

2. Parity Check

This is the simplest method of error checking which checks the characters by counting the number of 1s. In this method, D7 of each ASCII code is used to transmit parity check information. Parity may be the even parity (having even number of 1s in a character) or the odd parity (having odd number of 1s in a character).

In an even parity system, when a character has an odd number of 1s, the bit D7 is set to 1 and an even number of 1s is transmitted. On the other hand, in an odd parity system, when a character has an even number of 1s, the bit D7 is set to 1 and an odd number of 1s is transmitted.

For an example, character to be sent is ‘A’ whose ASCII code is 41H (0100 0001) with two 1s. If the character is transmitted in an odd parity system, the bit D7 is set to 1 and if it is transmitted in an even parity system, the bit D7 is set to 0. Most of microprocessors are designed to detect parity using the parity flag. However, the parity check cannot detect multiple errors in any given character.

Error Checks in Data Communication

3. Cyclic Redundancy Check (CRC)

This technique is based on mathematical relationships of polynomials. A stream of data can be represented as a polynomial that is divided by a constant polynomial, and the remainder, unique to that set of bits, is generated. The remainder is sent out as a check for errors. The receiver checks the remainder to detect an error in the transmission. This is a somewhat complex technique for error checking.

Bit and Baud Rate

- Bit rate is how many data bits are transmitted per second.
- Bit rates measure the number of data bits (that is 0′s and 1′s) transmitted in one second in a communication channel. A figure of 2400 bits per second means 2400 zeros or ones can be transmitted in one second, hence the abbreviation "bps." Individual characters (for example letters or numbers) that are also referred to as bytes are composed of several bits.
- A baud rate is the number of times per second a signal in a communications channel changes or varies state.
- For example, a 2400 baud rate means that the channel can change states up to 2400 times per second. The term "change state" means that it can change from 0 to 1 or from 1 to 0 up to X (in this case, 2400) times per second. It also refers to the actual state of the connection, such as voltage, frequency, or phase level).
- Bit rate (bps) and baud rate (baud per second) have this connection:

  If signal is changing every 10/3 ns then,

  Baud rate = 1/(10/3 ns) = $3/10*10^9 = 3*10^8$ = 300 mbd

Note: If 1 frame of data is coded with 1 bit then band rate and bit rate are same. Sometimes frame of data are coded with two or more bits then baud rate and bit rate are not same.

Simplex, Half Duplex and Full Duplex Data Communication

- **Simplex Mode**

Simplex transmission allows data to travel only in a **single, pre-specified direction**. Example –

- ➢Doorbell the signal can go only from the button to the chime
- ➢Television and Radio broadcasting.

The simplex standard is relatively uncommon for most types of computer-based telecommunications applications; even devices that are designed primarily to receive information, such as printers must be able to communicate acknowledgement signals back to the sender devices.

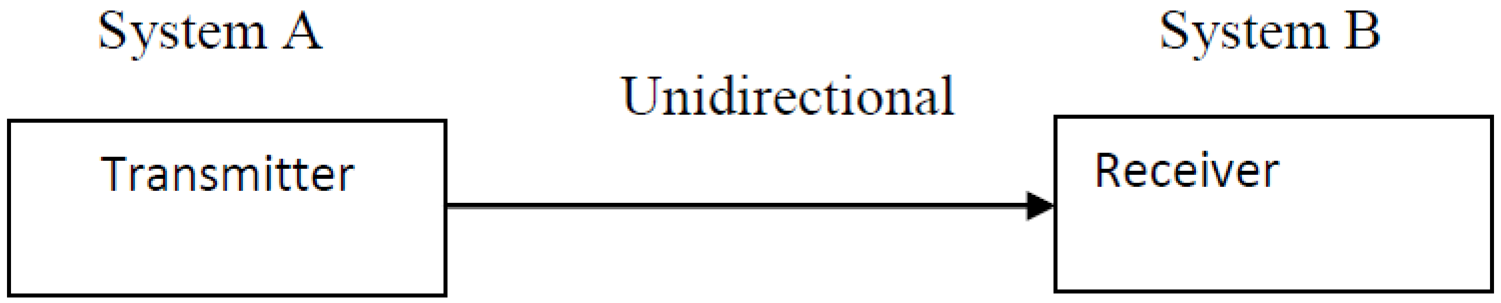


Fig: Simplex mode

Simplex, Half Duplex and Full Duplex Data Communication

- **Half Duplex Mode**

It is a **two way communication between two ports** provided that only one party can communicate at a time. Example - the press to talk radio phones used in police cars employs the half-duplex standard; only one person can talk at a time. Often the line between a desktop workstation and a remote CPU conforms to the half duplex patterns as well. If another computer is transmitting to a workstation, the operator cannot send new messages until the other computer finishes its message to acknowledge an interruption.

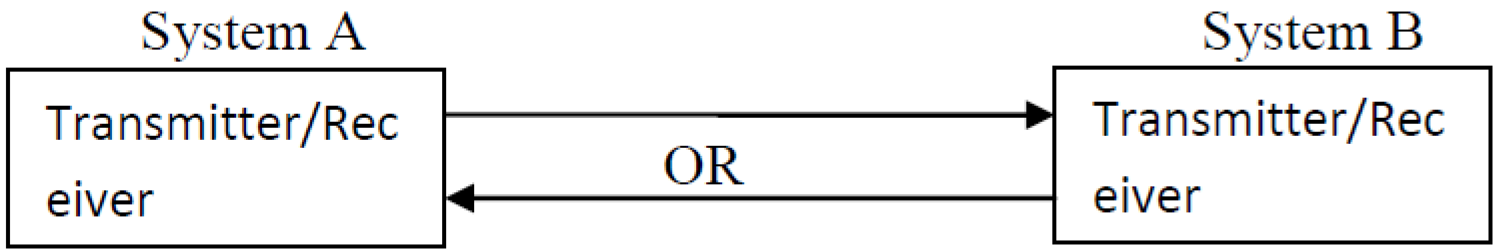


Fig:Half Duplex mode

Simplex, Half Duplex and Full Duplex Data Communication

- **Full Duplex Mode**

It provides **simultaneous two way transmission without the intervening** stop-and-wait aspect of half duplex. Full duplex is widely used in applications requiring continuous channels usage. Full duplex transmission works like traffic on a busy two way street the flow moves in two directions at the same time. Full-duplexing is ideal for hardware units that need to pass large amounts of data between each other as in mainframe-to-mainframe communications.

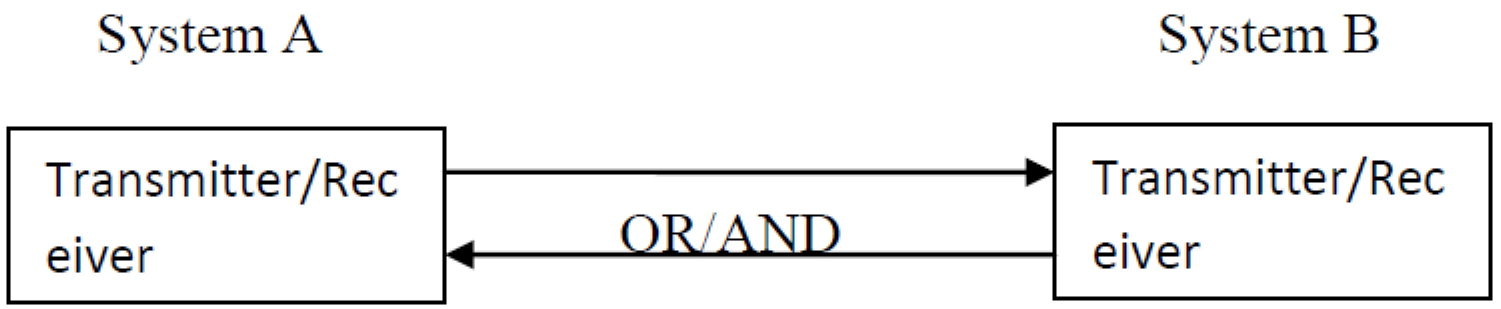


Fig: Full Duplex mode

Standards in Serial I/O

- A common understanding must exist, among various manufacturing and user groups that can ensure compatibility among different equipment.
- Standard is defined as the understanding which is accepted in industry and by users.
- A standard is normally defined by a professional organizations such as IEEE (Institute of Electrical and Electronics Engineers), EIA (Electronic Industries Association) as a de jure standard. However, a widespread practice can become a de facto standard.
- In serial I/O, data can be transmitted as either current or voltage.
- When data are transmitted with current signal such for teletype equipment, 20 mA (or 60 mA) current loops are used. When a teletype is marking or at logic 1, current flows; when it is at logic 0 (space), the current flow is interrupted. The advantage of the current loop method is that signals are relatively noise-free and are suitable for transmission over a distance.
- When data are transmitted with voltage signal, there are various standards which are explained next.

RS-232C

- Serial transmission of data is used as an efficient means for transmitting digital information across long distances, the existing communication lines usually the telephone lines can be used to transfer information which saves a lot of hardware.
- RS-232C is an interface developed to standardize the interface between data terminal equipment (DTE) and data communication equipment (DCE) employing serial binary data exchange.
  - Modem and other devices used to send serial data are called data communication equipment (DCE).
  - The computers or terminals that are sending or receiving the data are called data terminal equipment (DTE).
- RS- 232C is the interface standard developed by electronic industries Association (EIA) in response to the need for the signal and handshake standards between the DTE and DCE.

RS-232C (Contd…)

RS-232C has following standardize features:

- Uses 25 pins (DB – 25P) or 9 Pins (DE – 9P) standard where 9 pins standard does not use all signals i.e. data, control, timing and ground.
- Describes the voltage levels, impendence levels, rise and fall times, maximum bit rate and maximum capacitance for all signal lines.
- Specifies that DTE connector should be male and DCE connector should be female.
- Can send 20kBd for a distance of 50 ft.
- The voltage level for RS-232 are:
  - A logic high or 1 or mark, -3V to -15V
  - A logic low or 0 or space, +3v to +15v
  - Normally ±12V voltage levels are used
- Mc1488 line driver converts
  - Logic 1 to -9V
  - Logic 0 to +9v
- Mc1489 line receiver converts RS – 232 to TTL
- Signal levels of RS-232 are not compatible with that of the DTE and DCE which are TTL signals for that line driver such as M 1488 and line receiver MC1489 are used.

RS-232C (Contd…)

- RS- 232 signals used in handshaking:

Signal Flow	DE-9P	DB-25P	Signal	Description
-	1	-	Protective Ground	-
DTE to DCE	3	2	TxD	Transmitted Data
DCE to DTE	2	3	RxD	Received Data
DTE to DCE	7	4	$\overline{RTS}$	Request To Send
DCE to DTE	8	5	$\overline{CTS}$	Clear To Send
DCE to DTE	6	6	$\overline{DSR}$	Data Set Ready
Common Ref	5	7	GND	Signal Ground
DCE to DTE	1	8	$\overline{DCD}$	Data Carrier Detect
DTE to DCE	4	20	$\overline{DTR}$	Data Terminal Ready
DCE to DTE	9	22	RI	Ring Indicator
DCE to DTE	-	23	DSRD	Data Signal Rate Detector

RS-232C (Contd…)

- **Data Terminal Ready (DTR)**: After the terminal power is turned on and terminal runs any self checks, it asserts data terminal ready (DTR’) signal to tell the modem that it is ready.
- **Data Set Ready (DSR):** When the MODEM is powered up and ready to transmit or receive data, it will assert data set ready (DSR’) to the terminal. Under manual control or terminal control, modem then dials up the computer. If the computer is available, it will send back a specified tone.
- **Request to send (RTS)**: When a terminal has a character ready to send, it will assert a request-to-send (RTS’) signal to the modem.
- **Data Carrier Detect (DCD):** The modem will then assert its data-carrier-detect (DCD’) signal to the terminal to indicate that it has established connection with the computer.
- **Clear to send (CTS):** When the modem is fully ready to receive data, it asserts the clear-to-send (CTS’) signal back to the terminal.
- **Ring indicator (RI)**: It indicates that a ring has occurred at modem. Deactivating DTR or DSR breaks the connection but RI works independently of DTR i.e. a modem may activate RI signal even if DTR is not active.
- **Transmitted Data (TxD)**: The terminal then sends serial data characters to the modem.
- **Received Data (RxD)**: Modem will receive data from terminal through this line.
- **Data Signal Rate Detect (DSRD)**: It is used for switching different baud rate.

RS-232C (Contd…)

Digital Data Transmission Using Modem and standard Phone Lines

Standard telephone system can be used for sending serial data over long distances. However, telephone lines are designed to handle voice, bandwidth of telephone lines ranges from 300 Hz to 3400 Hz. Digital signal requires a bandwidth of several megahertz. Therefore, data bits should be converted into audio tones, this is accomplished through modems.

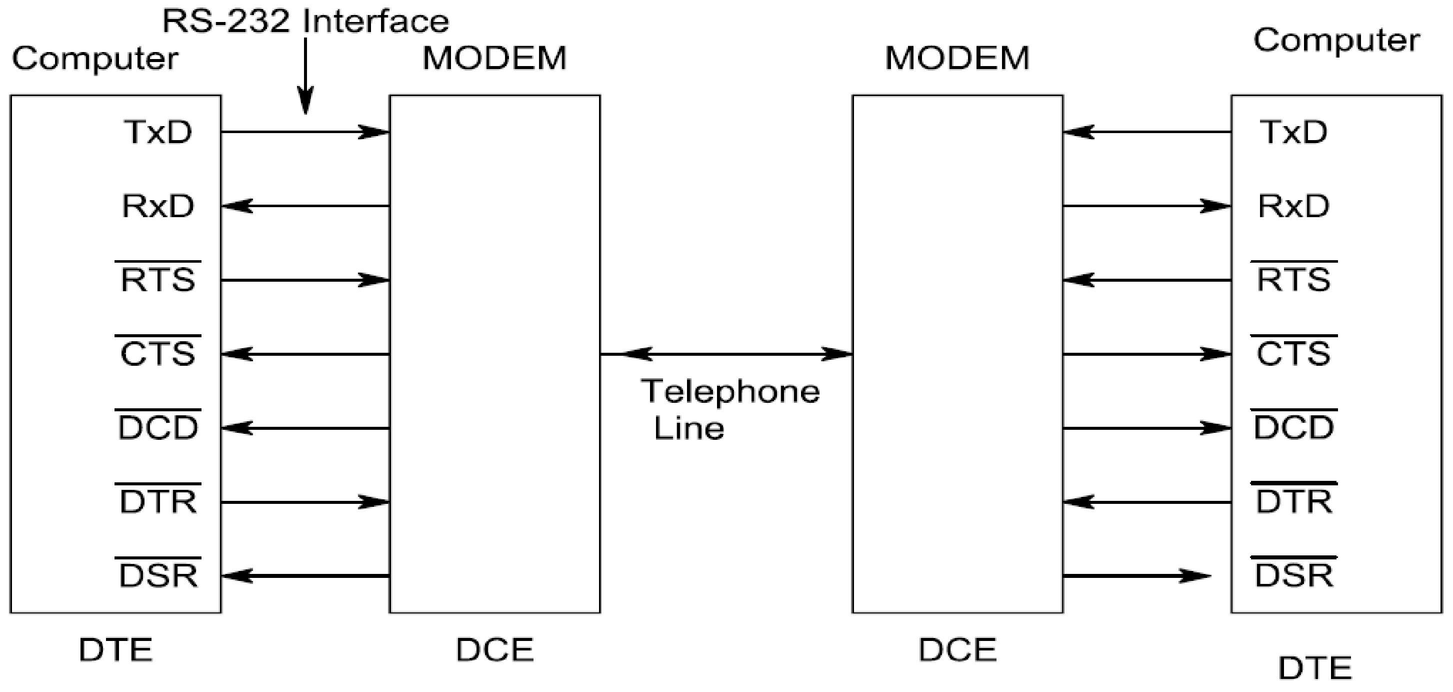


Fig: Digital Data transmission using MODEM and Telephone Line

RS-232C (Contd…)

Operation Mechanism

- DTE asserts $\overline{DTR}$ to tell the modem it is ready.
- Then DCE asserts $\overline{DSR}$ signal to the terminal and dials up.
- DTE asserts $\overline{RTS}$ signal to the modem.
- Modem then asserts $\overline{DCD}$ signal to indicate that it has established connection with the computer.
- DCE asserts $\overline{CTS}$ signals, then DTE sends serial data.
- When sending completed, DTE asserts $\overline{RTS}$ high, this causes modem to un assert its $\overline{CTS}$ signal and stop transmitting similar handshake taken between DCE and DTE other side.
- To communicate from serial port of a computer to serial port of another computer without modem, null-modem is used.

RS-232C (Contd…)

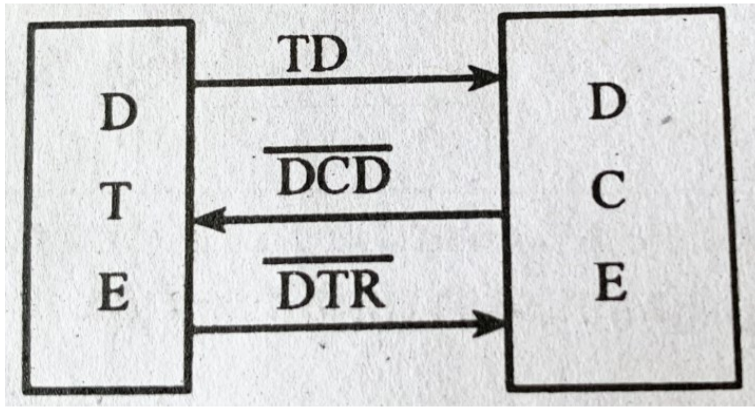
Simplex, Half Duplex and Full Duplex Operation Using RS-232 port

- **Simplex Connection for RS-232C**

There are two possibilities; data transfer from DTE to DCE or vice versa.

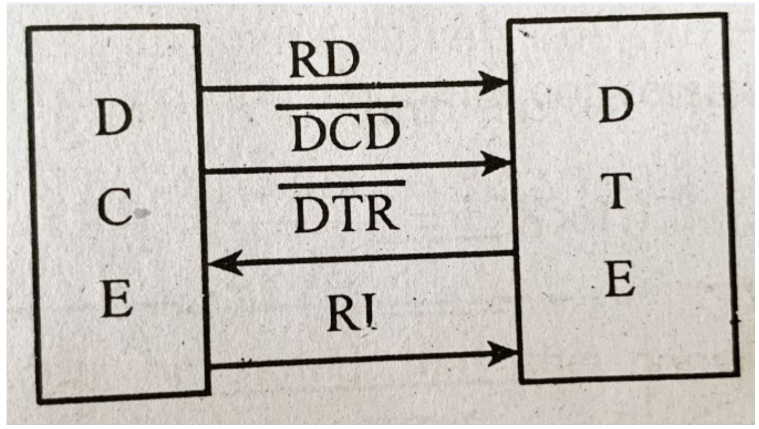
From DTE to DCE:

The DTE transfers data to the DCE via the **TxD line**. The RxD line is not connected. The DCE does not use RTS or DTE holds RTS signal active all the time. The DCE always outputs an **inactive DCD** signal as it can receive data from DTE and transfer it to destination. By means of **DTR** signal, DTE can indicate DCE that it is ready for operation as usual and may activate or disable DCE. RI signal has no meaning because normally transmitter calls receiver.



From DCE to DTE:

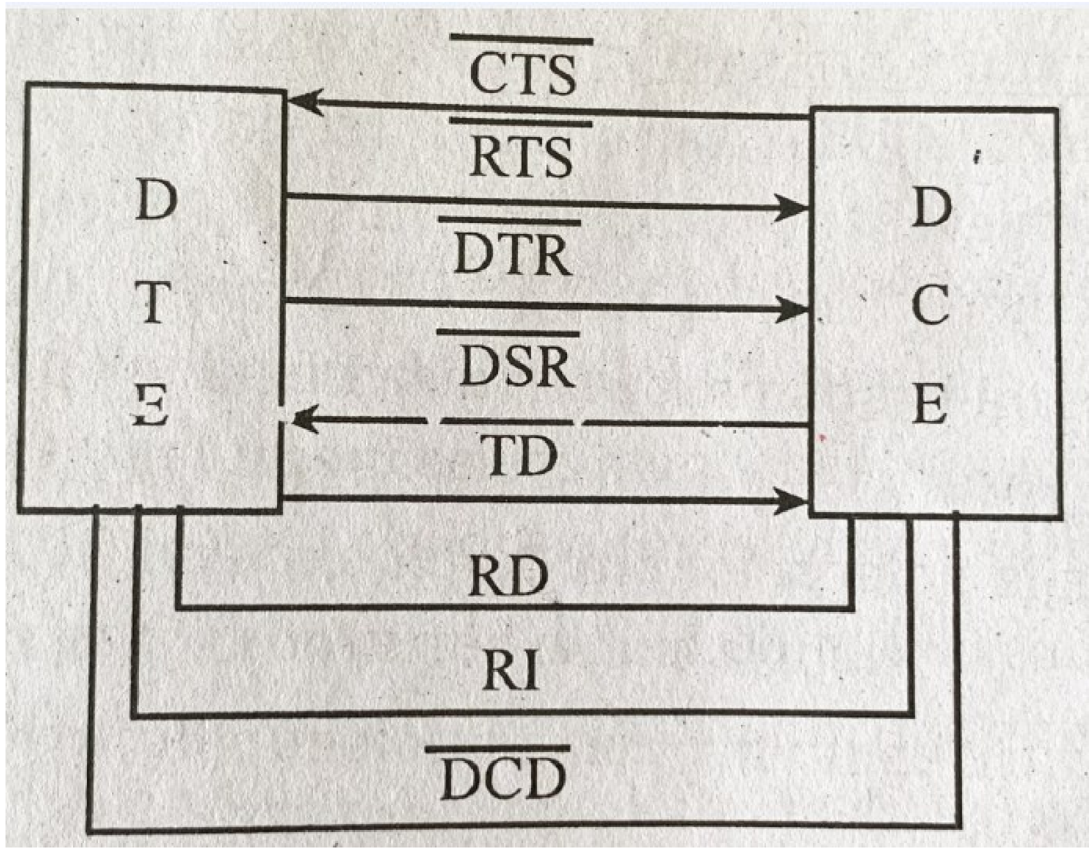
In this case, only the DCE transfers data to the DTE via **RxD line**. The TxD line is not connected. The DCE does not either use RTS or CTS signal or holds them constantly at an active level. The DCE may output an **active DCD** signal as it can detect a carrier signal from an external device and transfer data to DTE. By means of **DTR**, the DTE can indicate that it is ready for operation and it can activate or disable the DCE as usual. The RI signal has a meaning as external device may call DTE via DCE.



RS-232C (Contd…)

- **Half Duplex Connection for RS-232C**

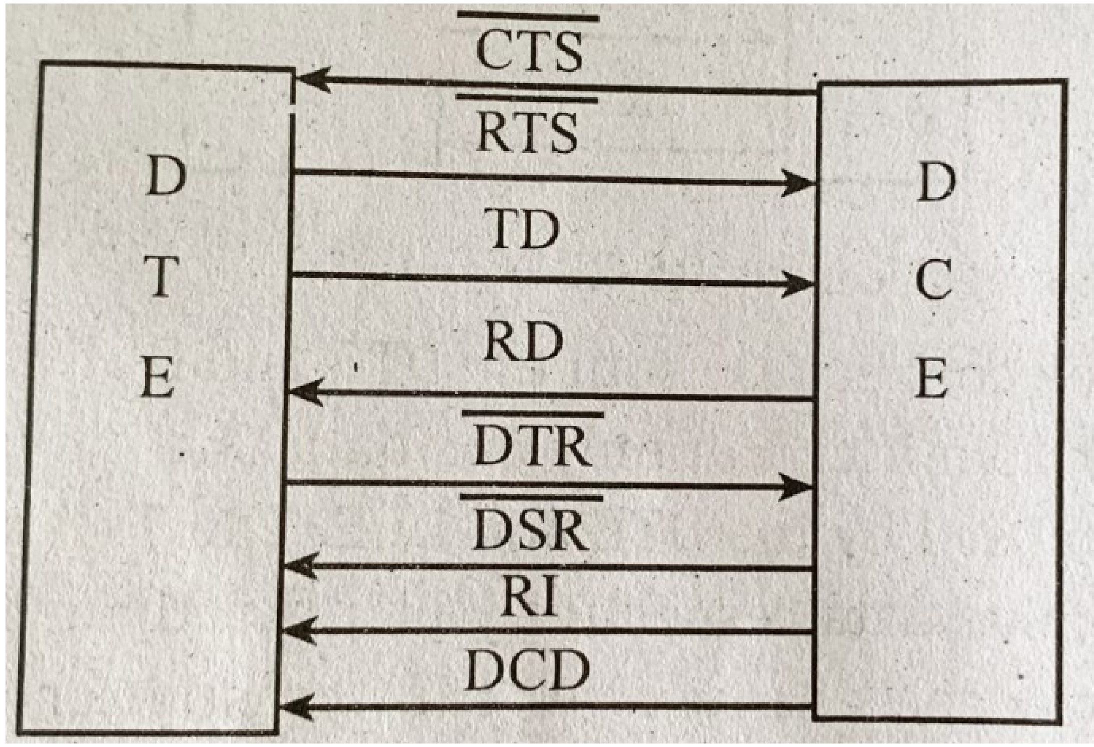
On a half duplex connection, both the DTE and DCE can operate as receiver and transmitter, but only one data line is available which is alternatively used by the DTE and DCE. The TxD and RxD lines output and receive data respectively in a strictly ordered manner for assigning the roles as receiver and transmitter between DTE and DCE; the handshake control signals **RTS and CTS** are used. If a DTE device wants to act as a transmitter, then it activates the RTS signal and waits for an acknowledgement of other DCE device by means of CTS signal. Now, data can be exchanged while DTE acting as transmitter and DCE as receiver otherwise DCE may operate as transmitter and DTE as receiver.



RS-232C (Contd…)

- **Full Duplex Connection for RS-232C**

Most microcomputer modems are full duplex, and transfer data in both directions simultaneously; thus DTE and DCE act simultaneously as receiver and transmitter. The RTS and CTS signals are meaningless and are usually not used or are always active. Further, the DSR signal is also enabled all the time on most modems but on some DCEs, DSR may be active only if preparations for calling destination device are completed. The signal is normally activated by DCE only if it has detected a carrier signal from the destination device. Also, in this connection, DTR signal acts as a main switch and RI indicates that an external device wants to establish a connection with DTE via DCE. A full duplex connection is very comfortable, as we need not pay attention to the roles of receiver and transmitter i.e. we may keep RTS signal active all time ignoring CTS and DSR signals.



RS-232C (Contd…)

- **Null (Zero) Modem Connection**

A zero modem serves for data exchan between DTEs. Since both the comput are configured as DTEs, directly connecti them by means of the conventional ser interface cable is impossible; not even t plug fits into the jack of the seco terminal. Also the TxD meets TxD a RxD meets RxD, DTR meets DTR a DSR meets DSR etc. This means tl outputs are connected to outputs and inpu are connected to inputs. With tl convention, no data transfer is possible.

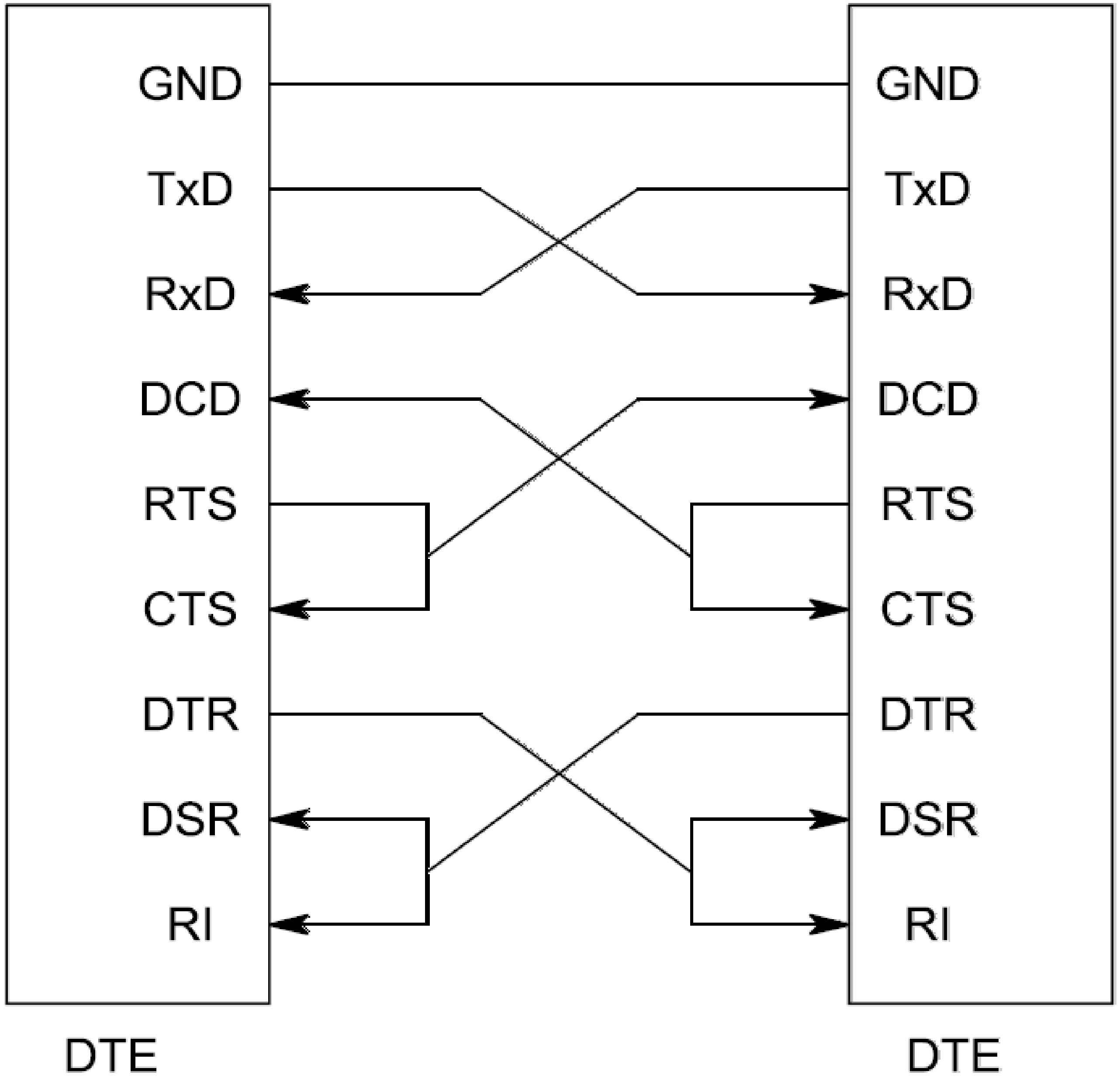


Fig: Null MODEM Connection for RS-232 Terminals

RS-232C (Contd…)

- **Null (Zero) Modem Connecti (Contd…)**

For the transmission of data, it is requir to twist the TxD and RxD lines. In this wa the transmitted data of one terminal (P becomes received data of other and vi versa. As shown in figure, activation **RTS** to begin a data transfer gives rise to activation of **CTS** on same DTE and to activation of **DCD** on other DTE. Furth an activation of **DTR** leads to rise of **DS** and RI on other DTE. Hence for eve DTE, it is simulated that a DCE is on t end of line, although a connection betwe two DTEs is actually present. Zero mode can be operated with standard BIOS a DOS functions.

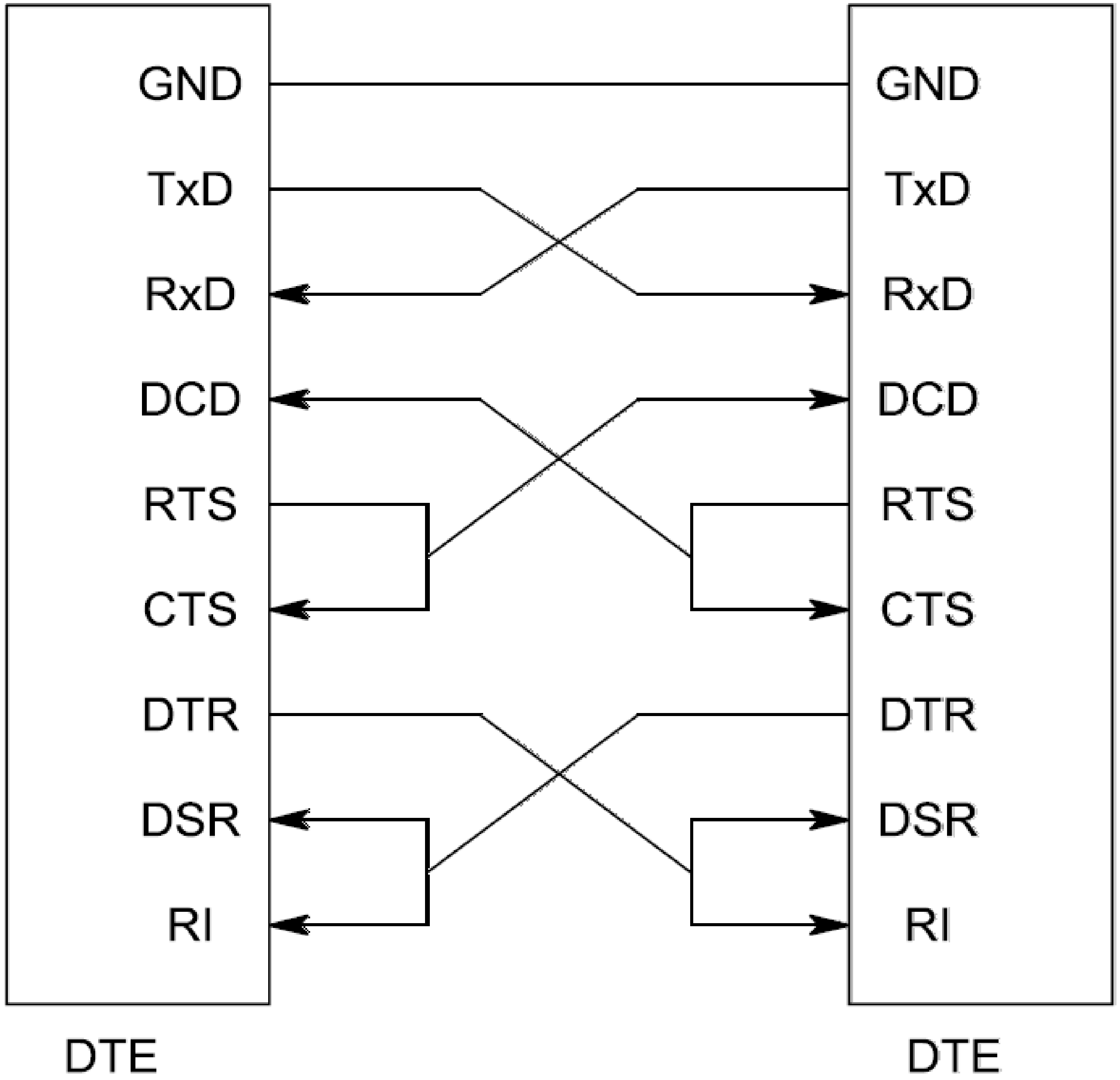


Fig: Null MODEM Connection for RS-232 Terminals

RS-232C (Contd…)

- **Connection to Printers**

As a printer is not DCE, various control and status lines have to be connected or interchanged to emulate behavior of a DCE. TxD data of PC becomes received data of printer. DCD and RI signals on PC are meaningless. On PC, RTS and CTS are connected to each other so that a transmission request from PC immediately enables the transmission. Since, printer as DTE refers to print anything as long as no active signal is present at inputs CTS, DSR and DCD. This problem is solved by connecting RTS with CTS and DTR with DCD and DSR. Thus, activating RTS gives rise to an activation of CTS and that of DTR to an activation of DCD and DSD.

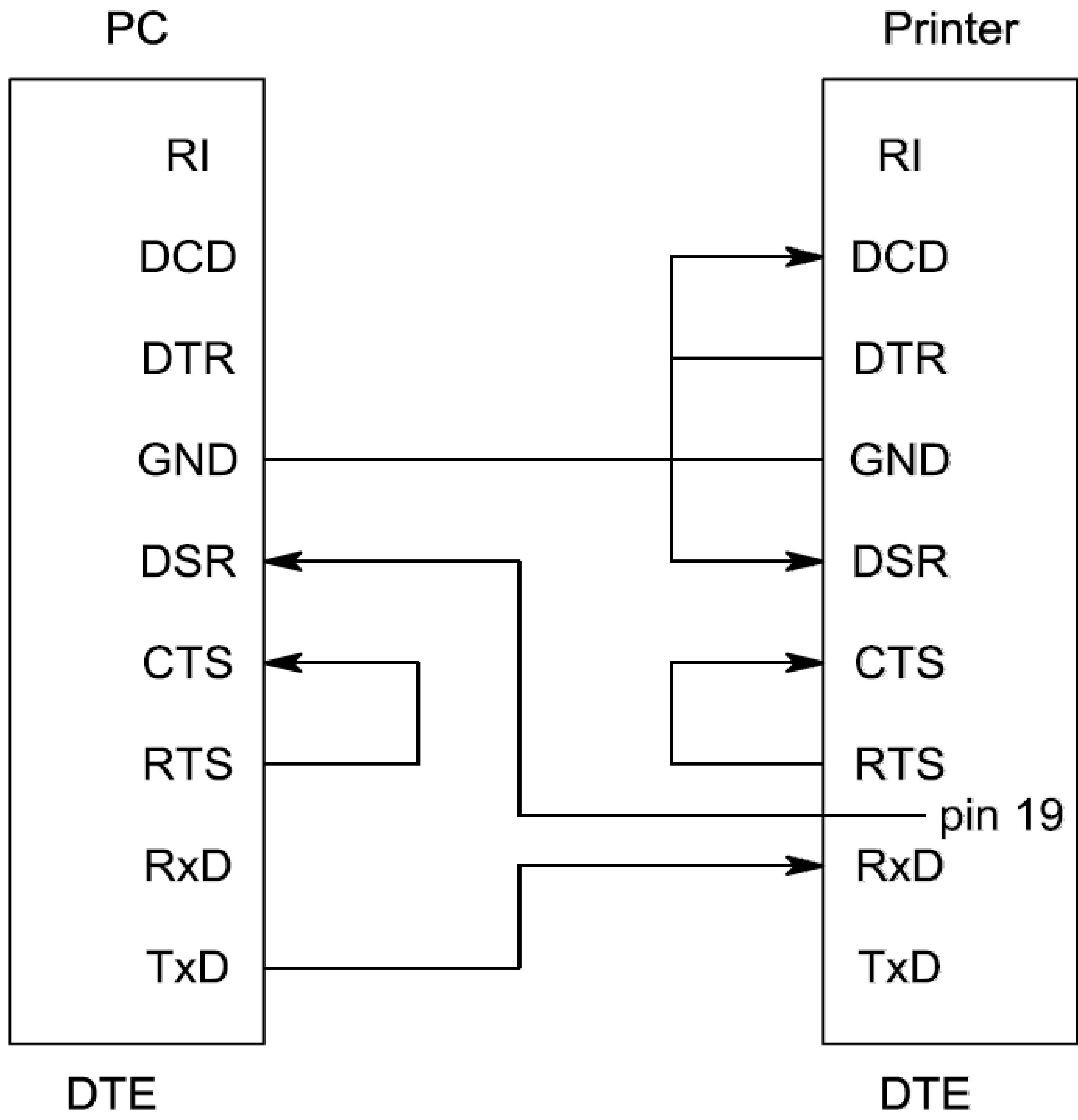


Fig: Connection to Printer

RS-232C (Contd…)

- **Connection to Printers (Contd…)**

Overrun error arises in serial interface as PC can transmit data much faster than printer can print it so internal printer buffer gets full. On parallel interface, this problem is solved as printer activates BUSY signal informing PC that it cannot accept data temporarily. In serial interface, pin 19 of printer is used to output a <<Buffer Full Signal>>. On DTE, DSR provides an input for this signal. If printer buffer is full, printer simply disables handshake signal at pin 19 and DTE knows that temporarily no additional data can be transferred. If enough room is available in buffer again, printer enables signal once more; PC may transfer data to printer. Not all printers with serial interface provide such a buffer full signal at pin 19.

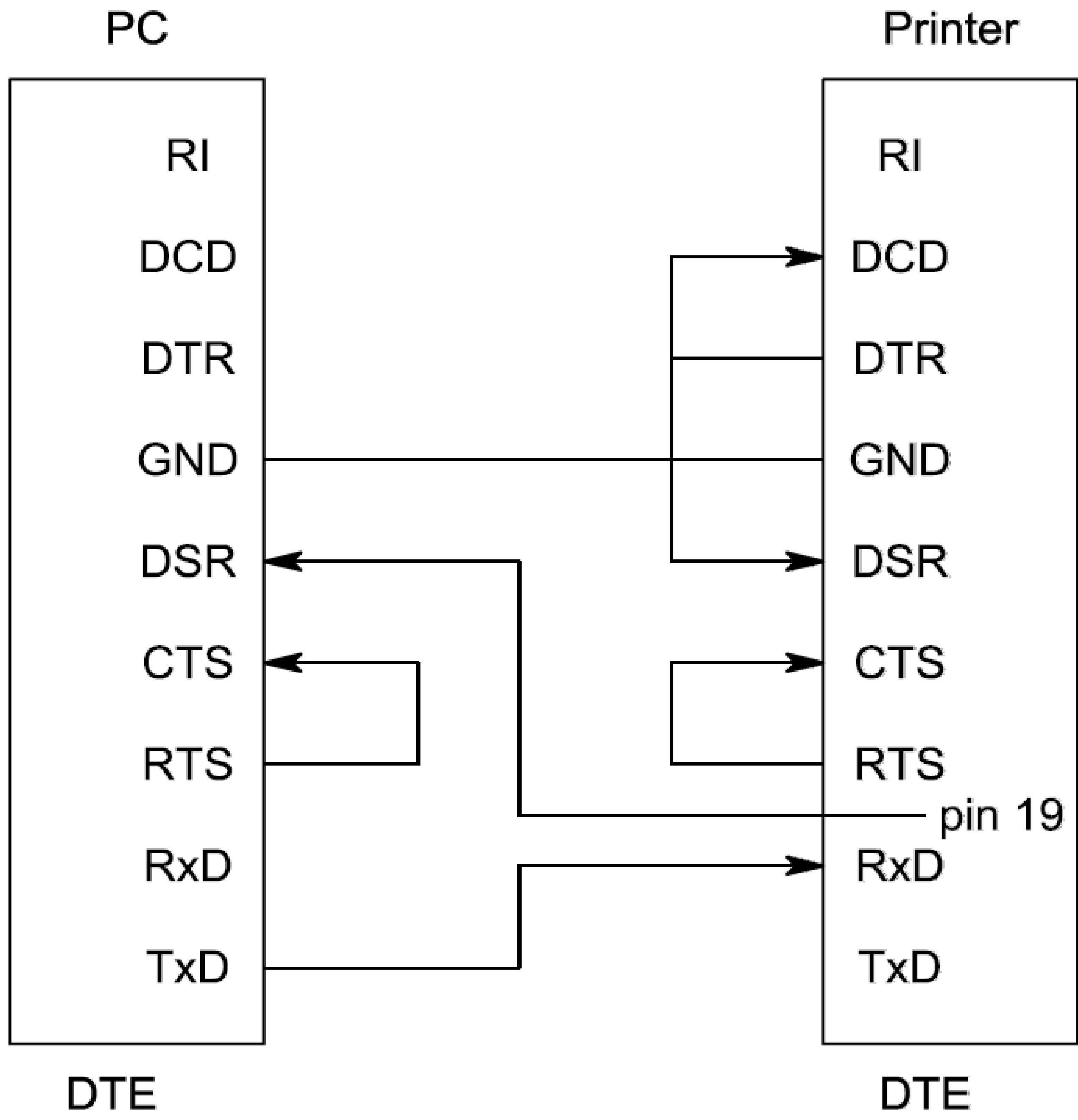


Fig: Connection to Printer

RS-423A

- Major problem with RS-232C - it can only transmit data reliably for about 50 ft at its maximum rate of 20Kbd
- If longer lines are used the transmission rate has to be drastically reduced due to open signal lines with a common signal ground.
- RS-423A - improvement over RS-232C
- Features of RS-423 are:
  - a low impendence single ended signal which can be sent over 50 coaxial cable and partially terminated at the receiving end to prevent reflection
  - Voltage levels
    - Logic High 4V - 6V negative
    - Logic Low 4V - 6V positive
  - allows a maximum data rate of 100 Kbd over 40 ft line or a maximum baud rate of 1 Kbd over 4000 ft line

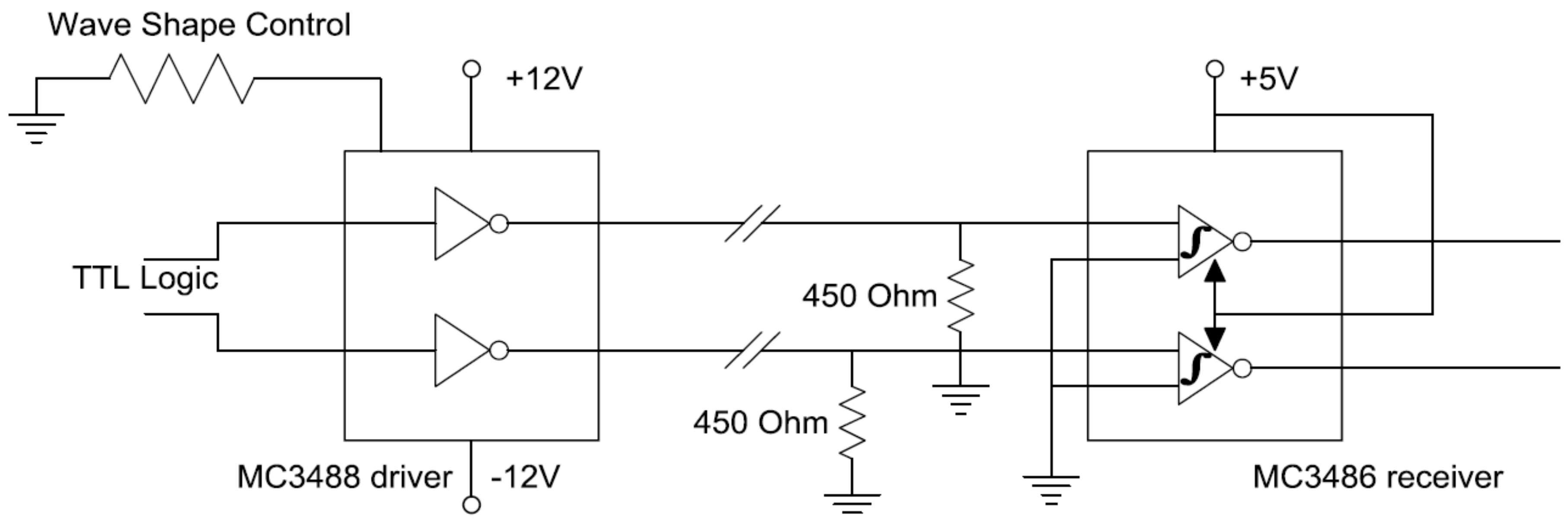


Fig: MC3488 driver and MC3489 receiver used for RS-423A Interface

RS-422A

- newer standard for serial data transfer
- each signal will be sent differentially over two adjacent wires in a ribbon cable or a twisted pair of wires uses differential amplifier to reject noise
- A differential line receiver MC3486 responds only to the voltage difference between its two inputs so any noise voltage that is induced equally on two inputs will not have any effect on the output of the differential receiver

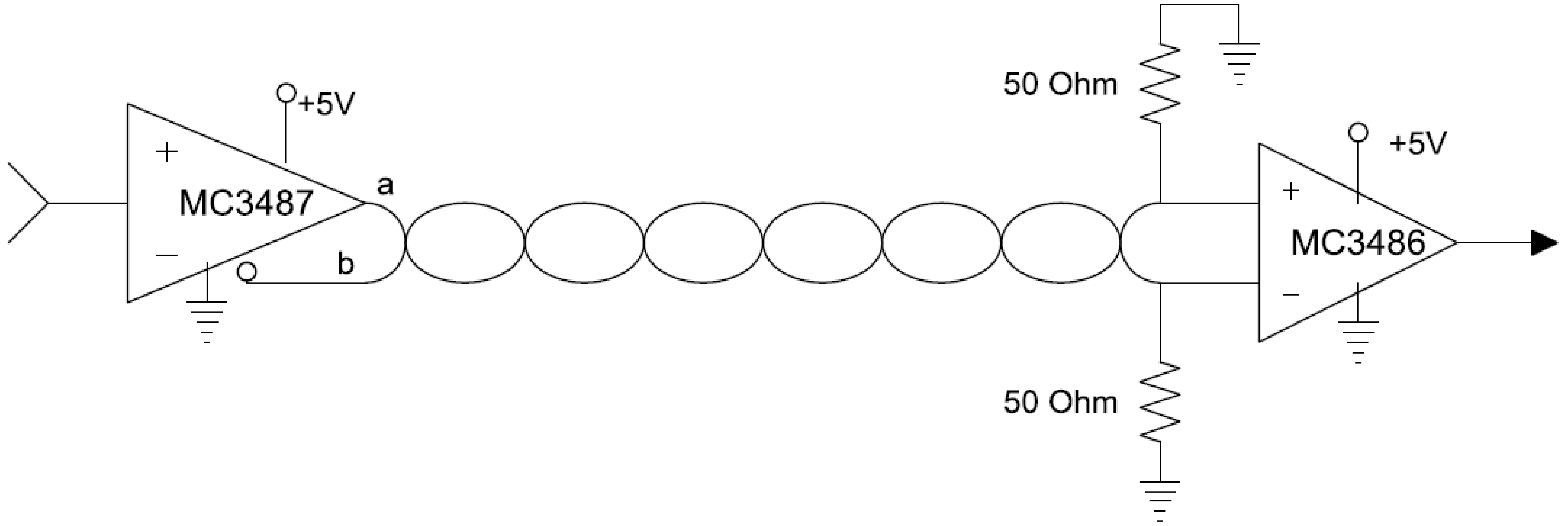


Fig: MC3487 driver and MC3486 receiver used for RS-422A Interface

RS-422A (Contd…)

- RS-422A has following standardized features:
- - Logic high is transmitted by making ‘b’ line more positive than ‘a’ line.
- - Logic low is transmitted by making ‘a’ line more positive than ‘b’ line.
- - The voltage difference between the two lines must be greater than 0.4V but less than
- 12V.
- - The MC3487 driver provides a differential voltage of about 2V.
- - The center or common mode voltage on the lines must be between -7v and +7v
- - Transmission rate is 10 MBd for 40 ft and 100 KBd for 4000 ft.
- - The high data transfer is because of differential line functions as a fully terminated
- transmission line.
- - MC3486 receiver only responds to the differential voltage eliminating noise.

Comparison of Serial I/O Standards

S.N.	Specifications	RS-232C	RS-423A	RS-422A
1.	Speed	20 Kbaud	100 Kbaud at 40 ft 1 kbaud at 4000 ft	10 Mbaud at 40 ft 100 kbaud at 4000 ft
2.	Distance	50 ft	4000 ft	4000 ft
3.	Logic 0	+3 V to +25 V	+4 V to +6 V	B line > A line
4.	Logic 1	-3 V to -25 V	-4 V to –6 V	A line > B line
5.	Receiver Input Voltage	±15V	±12V	±7V
6.	Mode of Operation	Single ended input and output	Differential input and single ended output	Differential input and output
7.	Noise Immunity	2.0 V	3.4 V	1.8 V
8.	Input Impedance	3-7 KOhm and 2500 pf	>4 KOhm	>4 KOhm
9.	Short circuit	500 mA	150 mA	150 mA