Processor and I/O Communication

Synchronous Transmission

- In **synchronous transmission**, data moves in a complete paired approach in the form of chunks or frames. Synchronization between the source and target is required so that the source knows where the new byte begins since there is no space between the data.
- Synchronous transmission is effective, dependable and is utilized for transmitting a large amount of data. It offers real-time communication between linked devices.

Characteristics of Synchronous Transmission

- There are no spaces in between characters being sent.
- Timing is provided by modems or other devices at the end of the transmission.
- Special syn characters goes before the data being sent.
- The syn characters are applied between chunks of data for timing functions.

Examples of Synchronous Transmission

- Chatrooms
- Video conferencing
- Telephonic conversations
- Face-to-face interactions

Asynchronous Transmission

- In **asynchronous transmission** data moves in a half-paired approach, 1 byte or 1 character at a time. It sends the data in a constant current of bytes. The size of a character transmitted is 8 bits where a parity bit is added each at the beginning and at the end which makes it a total of 10 bits.
- It doesn’t need a clock for integration; rather it utilizes the parity bits to inform the receiver how to translate the data.
- It is straightforward, quick, cost-effective and doesn’t need a 2-way communication.

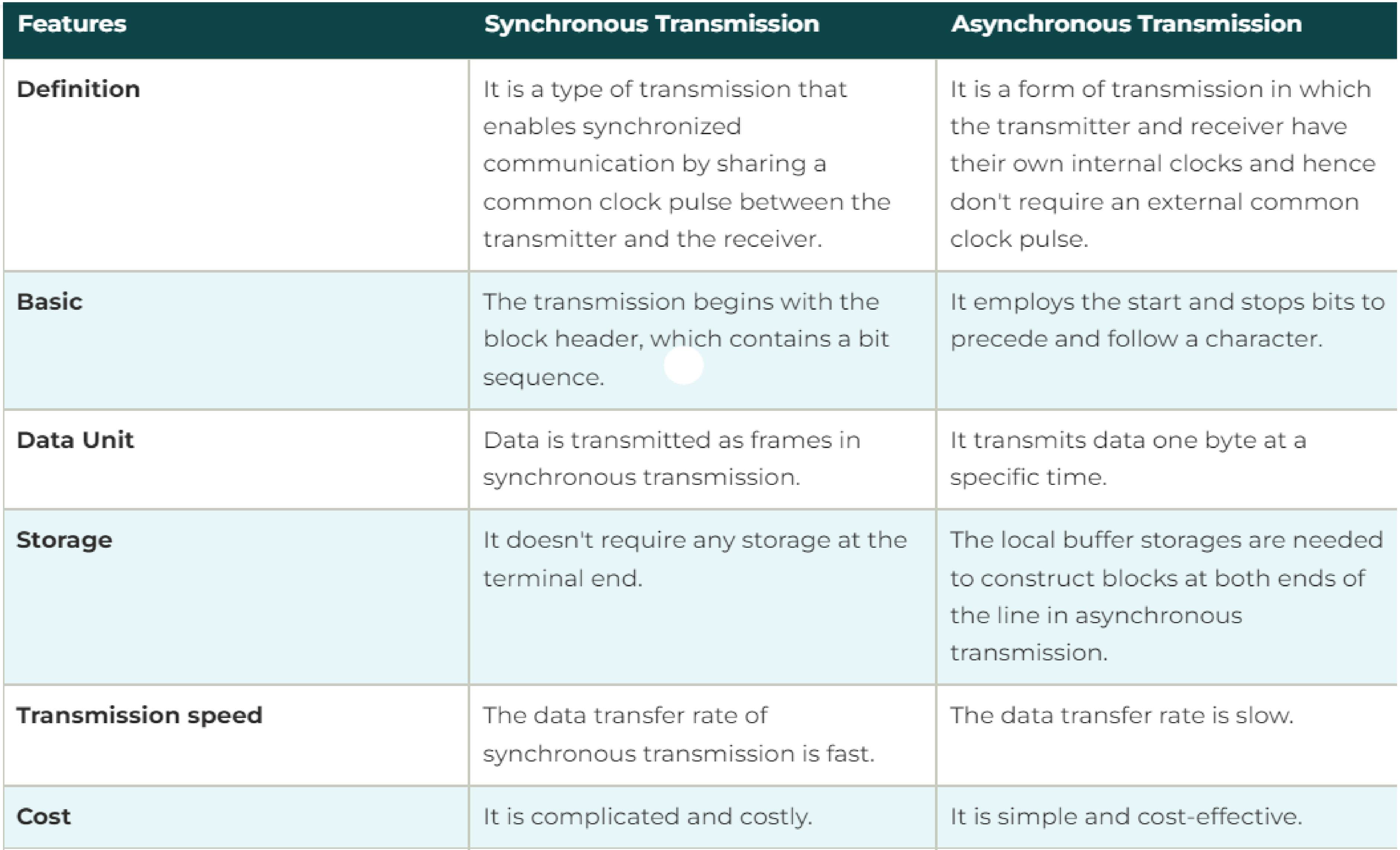
Characteristics of Asynchronous Transmission

- Each character is headed by a beginning bit and superseded by one or more end bits.
- There may be gaps or spaces in between characters.

Examples of Asynchronous Transmission

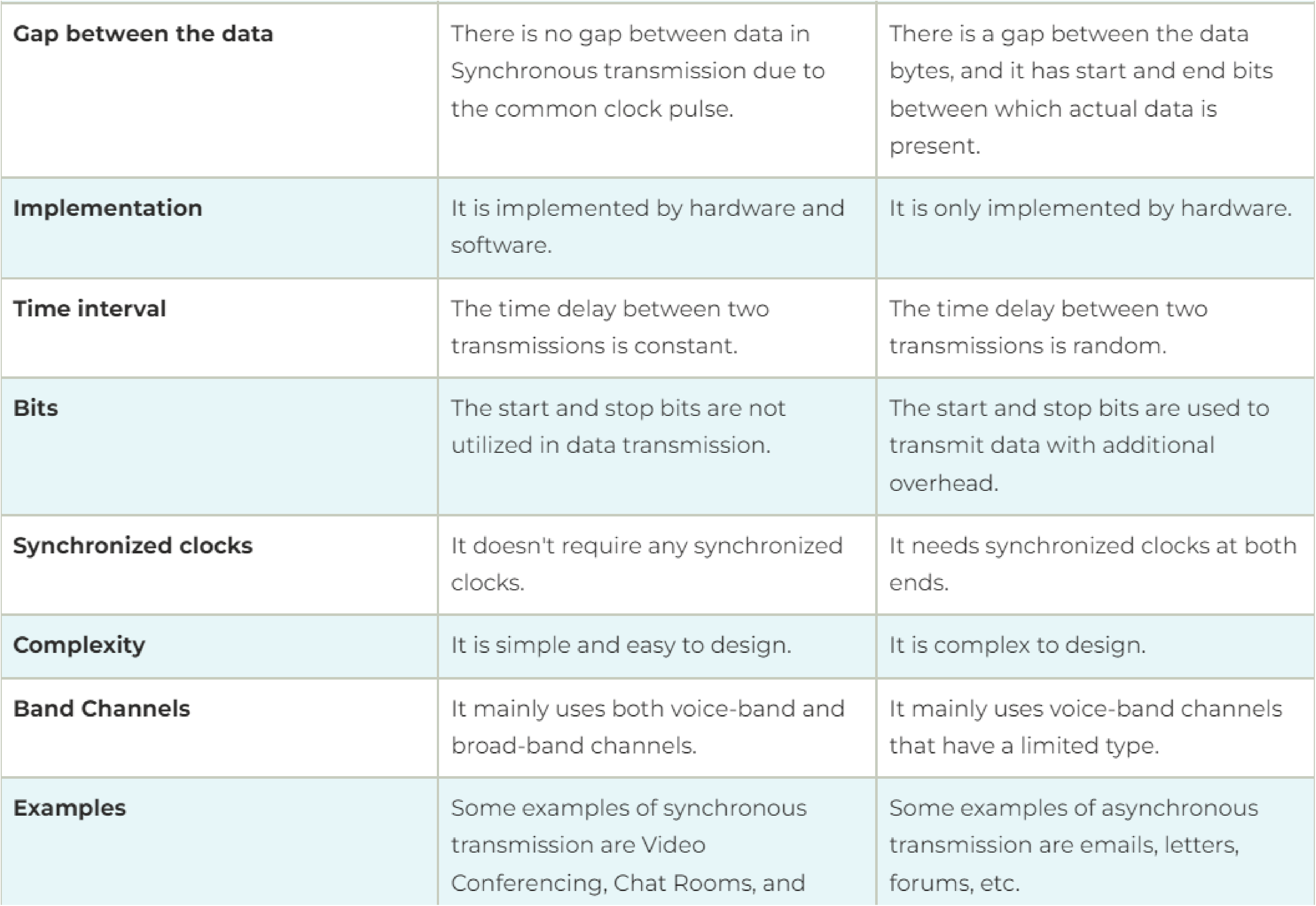
- Emails
- Forums
- Letters
- Radios
- Televisions

Synchronous and Asynchronous Transmission



Features	Synchronous Transmission	Asynchronous Transmission
Definition	It is a type of transmission that enables synchronized communication by sharing a common clock pulse between the transmitter and the receiver.	It is a form of transmission in which the transmitter and receiver have their own internal clocks and hence don't require an external common clock pulse.
Basic	The transmission begins with the block header, which contains a bit sequence.	It employs the start and stops bits to precede and follow a character.
Data Unit	Data is transmitted as frames in synchronous transmission.	It transmits data one byte at a specific time.
Storage	It doesn't require any storage at the terminal end.	The local buffer storages are needed to construct blocks at both ends of the line in asynchronous transmission.
Transmission speed	The data transfer rate of synchronous transmission is fast.	The data transfer rate is slow.
Cost	It is complicated and costly.	It is simple and cost-effective.

Synchronous and Asynchronous Transmission

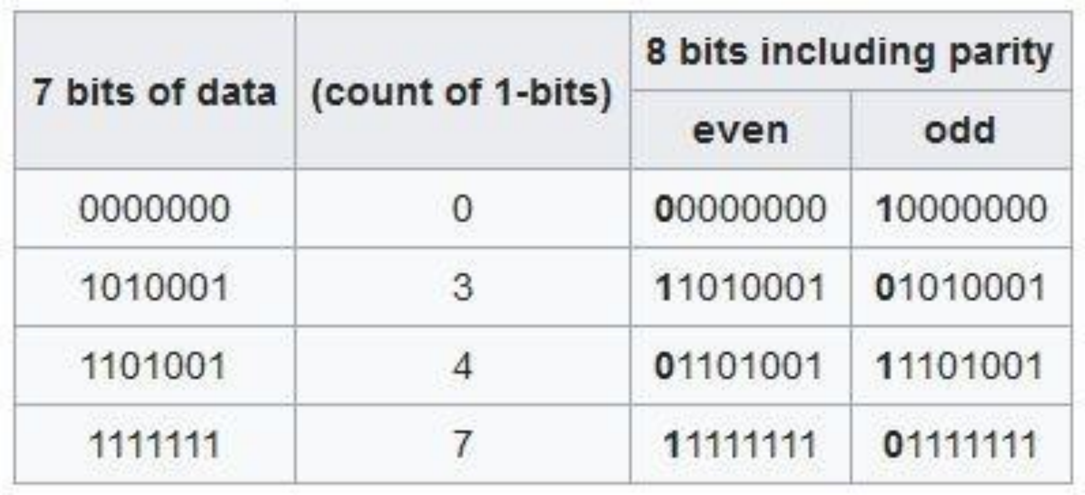


Gap between the data	There is no gap between data in Synchronous transmission due to the common clock pulse.	There is a gap between the data bytes, and it has start and end bits between which actual data is present.
Implementation	It is implemented by hardware and software.	It is only implemented by hardware.
Time interval	The time delay between two transmissions is constant.	The time delay between two transmissions is random.
Bits	The start and stop bits are not utilized in data transmission.	The start and stop bits are used to transmit data with additional overhead.
Synchronized clocks	It doesn't require any synchronized clocks.	It needs synchronized clocks at both ends.
Complexity	It is simple and easy to design.	It is complex to design.
Band Channels	It mainly uses both voice-band and broad-band channels.	It mainly uses voice-band channels that have a limited type.
Examples	Some examples of synchronous transmission are Video Conferencing, Chat Rooms, and	Some examples of asynchronous transmission are emails, letters, forums, etc.

Parity Check

- A **parity check** is the process that ensures accurate data transmission between nodes during communication.
- A **parity bit** is appended to the original data bits to create an even or odd bit number; the number of bits with value one.
- The source then transmits this data via a link, and bits are checked and verified at the destination. Data is considered accurate if the number of bits (even or odd) matches the number transmitted from the source.
- The parity check method is used for error detection.

- A **parity bit**, or **check bit**, is a bit added to a string of binary code to ensure that the total number of 1-bits in the string is even or odd. Parity bits are used as the simplest form of error detecting code.
- There are two variants of parity bits: **even parity bit** and **odd parity bit**.



7 bits of data	(count of 1-bits)	8 bits including parity	
		even	odd
0000000	0	**0**0000000	**1**0000000
1010001	3	**1**1010001	**0**1010001
1101001	4	**0**1101001	**1**1101001
1111111	7	**1**1111111	**0**1111111

- In the case of **even parity**, for a given set of bits, the occurrences of bits whose value is 1 is counted. If that count is odd, the parity bit value is set to 1, making the total count of occurrences of 1s in the whole set (including the parity bit) an even number. If the count of 1s in a given set of bits is already even, the parity bit's value is 0.
- In the case of **odd parity**, the coding is reversed. For a given set of bits, if the count of bits with a value of 1 is even, the parity bit value is set to 1 making the total count of 1s in the whole set (including the parity bit) an odd number. If the count of bits with a value of 1 is odd, the count is already odd so the parity bit's value is 0.

BAUD

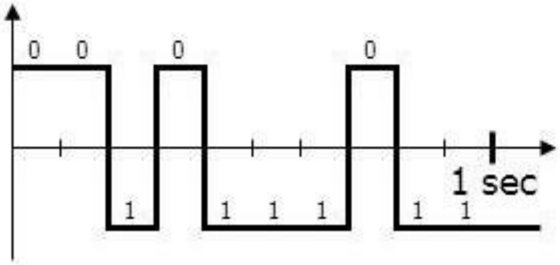
- Defined as number of signal changes per second, that is the rate at which the serial data is being transferred (bits per second).

- Common baud rates are 300, 600, 1200, 2400, 4800, 9600 & 19200.

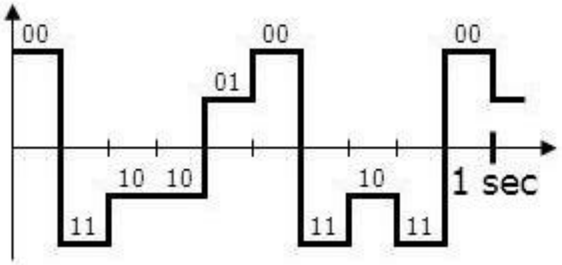
- Bit time is the delay between two successive bits, determined as 1/baud.

Baud and Bit Rate

- **Baud** → How many times a signal changes per second
- **Bit rate** → How many bits can be sent per time unit (usually per second)
- Bit rate is controlled by baud and number of signal levels



Baud = 10
Bit rate = 10 bps



Baud = 10
Bit rate = 20 bps

RS 232 Standard

- **RS-232 (Recommended Standard 232).**
- The RS-232C standard is an ***asynchronous serial***

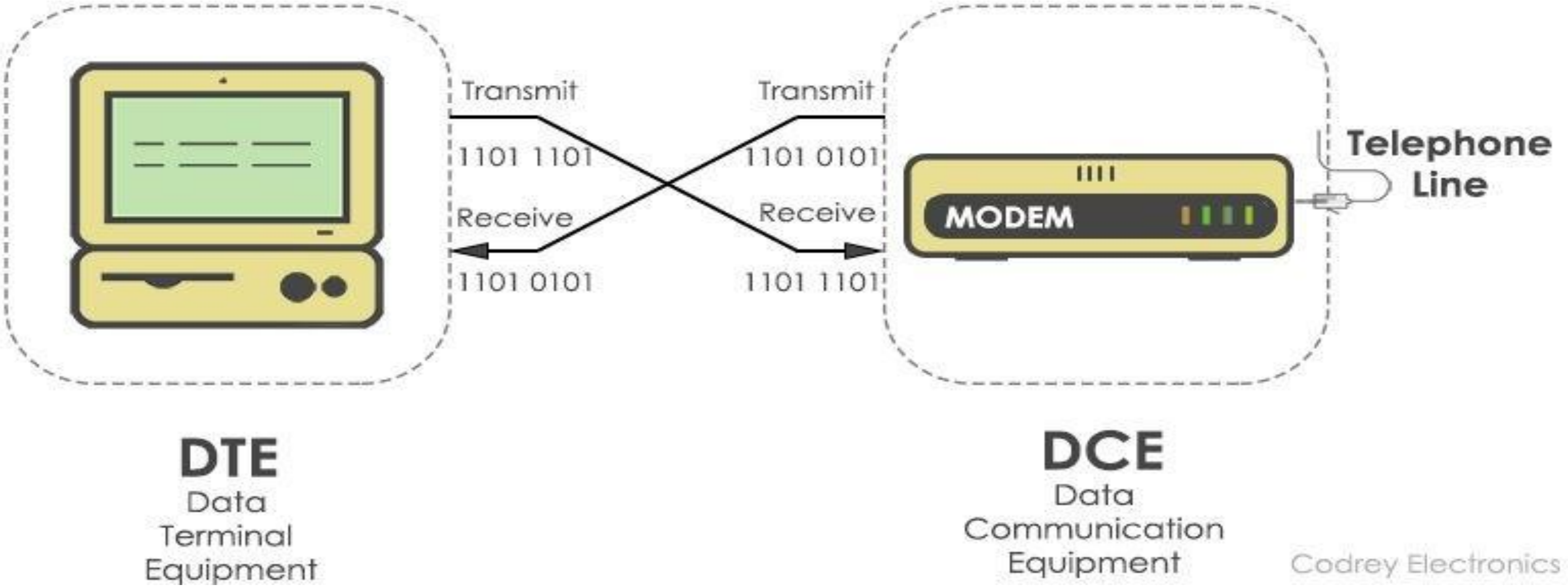
  communication method.
- Serial means that the information is sent 1-bit at a time.
- Asynchronous means that no clock signal is sent with the data. Each side uses its own clock and a start and stop bit. Synchronous communication means that a clock signal is sent in addition to a data signal (or interwoven with it)

RS 232 Standard (Cont.)

- In RS-232, user data is sent as a timeseries of bits.
- Both synchronous and asynchronous transmissions are supported by the standard.
- RS-232 devices may be classified as Data Terminal Equipment (DTE) or Data Communication Equipment (DCE), this defines that each device which wires will be sending and receiving each signal.
- DTE refers to terminals and computers that sends and receives data.
- DCE refers to communication equipment, such as modems, that are responsible for data transferring.

- The DTE (computer) transmits the information serially to the other end equipment DCE (modem). In this case, DTE sends binary data “11011101” to DCE and DCE sends binary data “11010101” to the DTE device.
- RS232 describes the common voltage levels, electrical standards, operation mode and number of bits to be transferred from DTE to DCE. This standard is used for transmission of information exchange over the telephone lines.

Communication between DTE and DCE



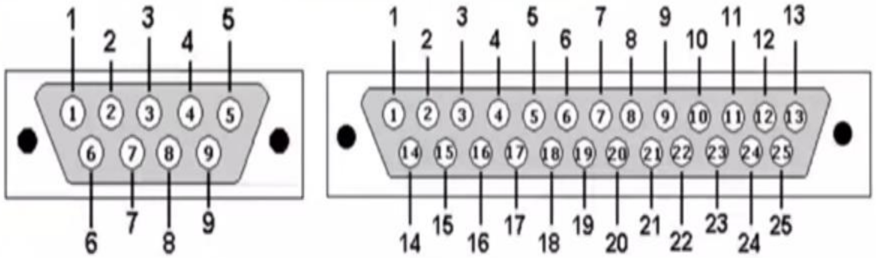
RS 232 Protocol

❖ Basics of RS 232 Protocol

- ❑ RS 232 protocol is used for serial communication.
- ❑ RS 232 protocol provides Asynchronous full duplex communication. {Without Clock}
- ❑ RS 232 protocol can handle serial communication with the distance of 50ft, depending on cable type and bit rate.
- ❑ RS 232 protocol can handle maximum data rate of 1Mbps.
- ❑ As we increase the length of cable, capacitance will also increase which will limits the data rate.
- ❑ Some special cables are also available, by which we can have communication up to 130fts.
- ❑ RS 232 protocol uses DB9 or DB25 connector for serial communication.
- ❑ In DB25 connector, only Nine lines are used as it is there with DB9.

❖ Connectors of RS 232 Protocol

❖ Pin Details of DB9 and DB25 connector

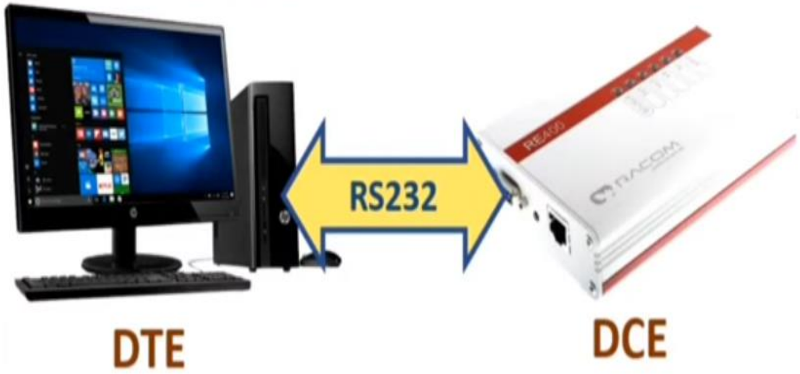


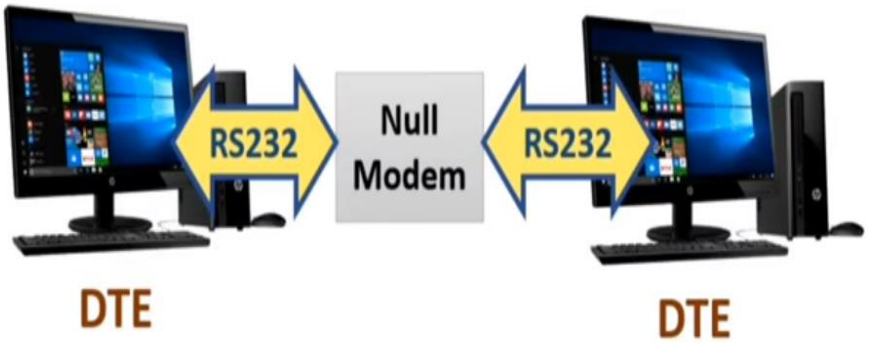
Pin-D9	Pin-D25	Signal	Source	Type	Description
1	8	CD	DCE	Control	Carrier Detect
2	3	RX	DCE	Data	Receive Data
3	2	TX	DTE	Data	Transmit Data
4	20	DTR	DTE	Control	Data Terminal Ready
5	7	SG	-	-	Signal Ground
6	6	DSR	DCE	Control	Data Set Ready
7	4	RTS	DTE	Control	Request to Send
8	5	CTS	DCE	Control	Clear to Send
9	22	RI	DCE	Control	Ring Indicator
-	Rest	None	-	-	-

RS 232 Protocol

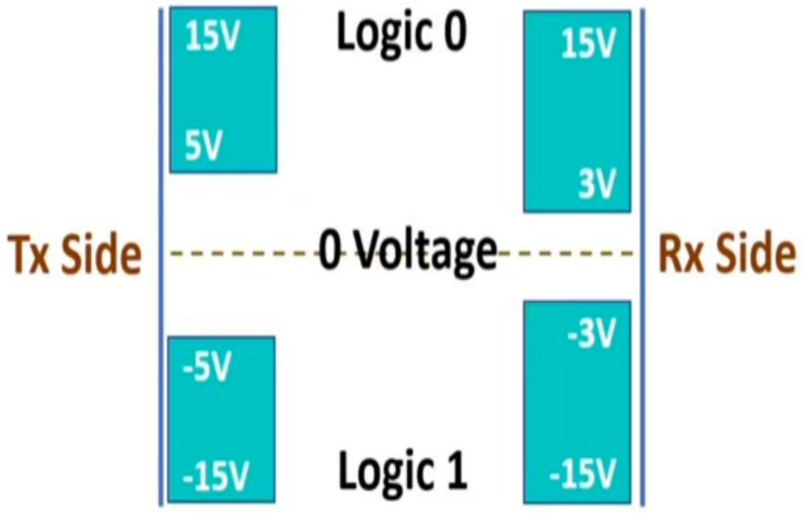
❖ DTE and DCE of RS 232 Protocol

- DTE – Data Terminal Equipment. {Computer}
- DCE – Data Communication Equipment. {Modem}
- Communication happens between DTE and DCE, we can not have communication between two DTE or two DCE.
- For two DTE communication, we use Null modem in between two DTE for communication.
- In Null Modem, we just need to configure Rx, Tx and SG lines of DB9 or DB25 connector.





❖ Voltages of RS 232 Protocol



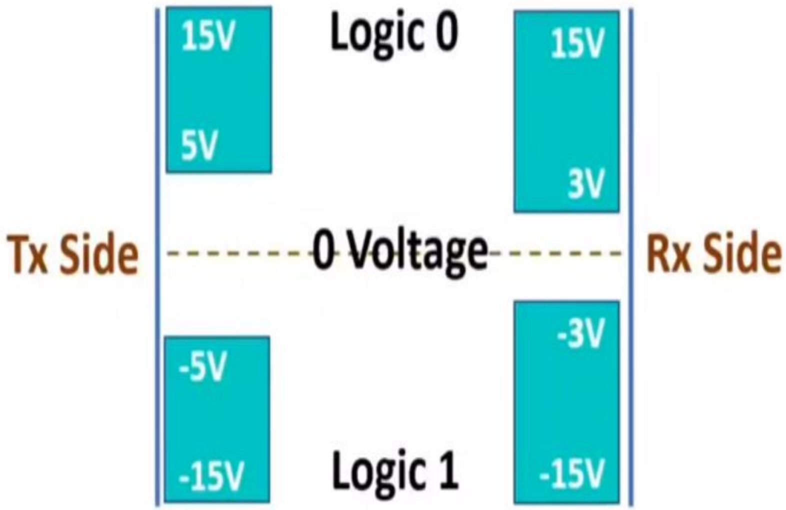
RS 232 Protocol

❖ DTE and DCE of RS 232 Protocol

- ❑ DTE – Data Terminal Equipment. {Computer}
- ❑ DCE – Data Communication Equipment. {Modem}
- ❑ Communication happens between DTE and DCE, we can not have communication between two DTE or two DCE.
- ❑ For two DTE communication, we use Null modem in between two DTE for communication.

Pin-D9	Pin-D25	Signal	Source	Type	Description
1	8	CD	DCE	Control	Carrier Detect
2	3	RX	DCE	Data	Receive Data
3	2	TX	DTE	Data	Transmit Data
4	20	DTR	DTE	Control	Data Terminal Ready
5	7	SG	-	-	Signal Ground
6	6	DSR	DCE	Control	Data Set Ready
7	4	RTS	DTE	Control	Request to Send
8	5	CTS	DCE	Control	Clear to Send
9	22	RI	DCE	Control	Ring Indicator
-	Rest	None	-	-	-

❖ Voltages of RS 232 Protocol



❖ Handshaking of RS 232 Protocol

- ❑ RTS of DTE is connected with CTS of DCE and visa versa.
- ❑ DSR of DTE is connected with DTR of DCE and visa versa.
- ❑ Tx of DTE is connected with Rx of DCE and visa versa.
- ❑ SG of DTE and DCE is commonly connected.
- ❑ 1st DTE is giving RTS {Request to Send} Signal to DCE.
- ❑ If DCE is not busy then only DCE sends respond on DTR {Data Transmit Ready}
- ❑ Then DTE can send data to DCE.

RS 232 Protocol

❖ Data Frame Format of RS 232 Protocol

❑ Data Frame Format of RS 232 Protocol is similar to UART.

Start bit = 0	Data {8 bits}	Parity Bit	Stop Bit = 1

❑ In one frame of RS 232 Protocol, along with 8 bits of data, start bit, stop bit and parity redundancy is also added.

❑ Parity bit is optional but it can be used to identify the error in data reception.

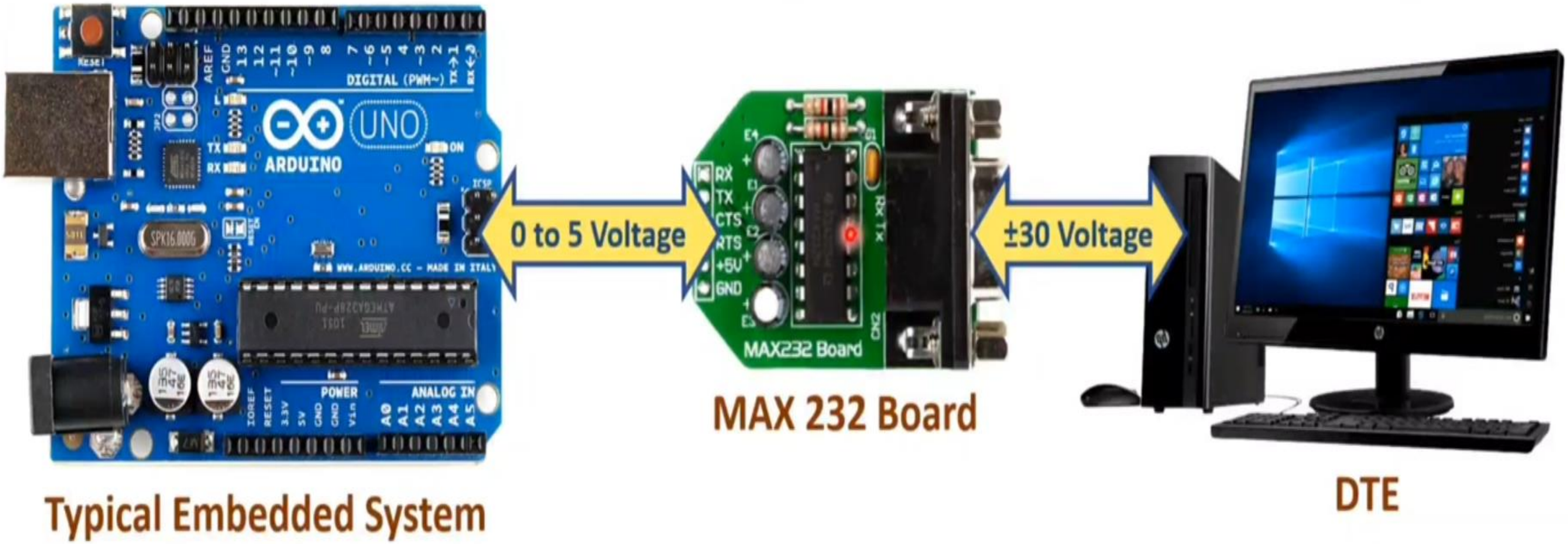
❖ RS 232 Protocol in Embedded System

❑ Majority of embedded system work with voltage range of 0 to 5V and RS 232 works with ±15V.

❑ To make data transfer compactible, we need to use MAX 232 IC with embedded system to enable RS 232 Protocol.

❑ MAX 232 converts voltage range of (0-5V) into ±25V and visa versa.

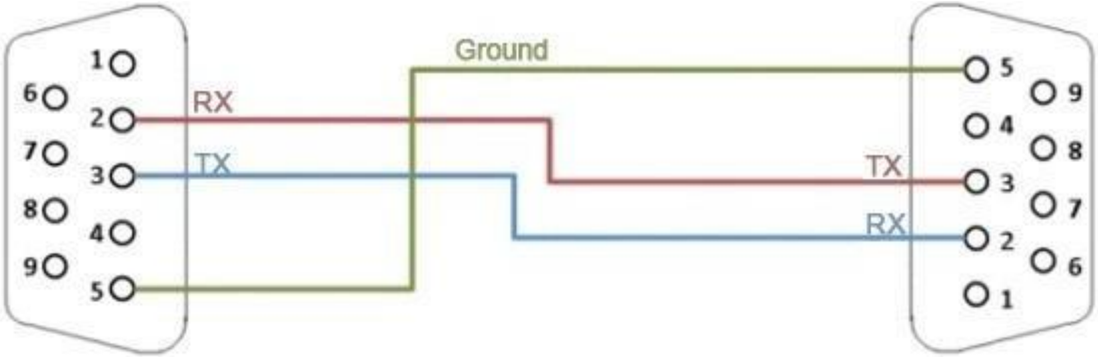
❑ NOTE: MAX 232 will also limit the data transfer speed of RS 232 Protocol.



Null Modem

- A null modem is a specially designed cable that allows a “head-to-head” connection between two nearby serial devices (computers) through their communication ports (RS-232). Having a length limitation of up to 30 feet, it is most commonly used to connect PCs within the same room for gaming and other purposes such as sending and receiving files.
- A null modem is also known as a crossover cable.

- **Null modem** is a communication method to directly connect two DTEs (computer, terminal, printer, etc.) using an RS-232 serial cable.
- Null-modem connections are made possible through the use of a null mode cable. This is the simplest solution to connecting the two computers.
- The null modem cable is comprised of three lines. One wire serves as the signal ground with the other two lines fostering the transmitting and receiving of your data. Based on the software used in this connection, some kind of authenticating handshake may be necessary.



Limitations of the Standard

- The limited transmission speed, relatively large voltage swing, and large standard connectors motivated development of the universal serial bus(USB) which has displaced RS-232.
- Multi-drop connection among more than two devices is not defined.
- Also multi-drop have limitations in speed and compatibility.

8251 Programmable Communication Interface

- 8251 is a programmable device to perform either synchronous or asynchronous serial communication , called as universal synchronous asynchronous receiver transmitter - USART.

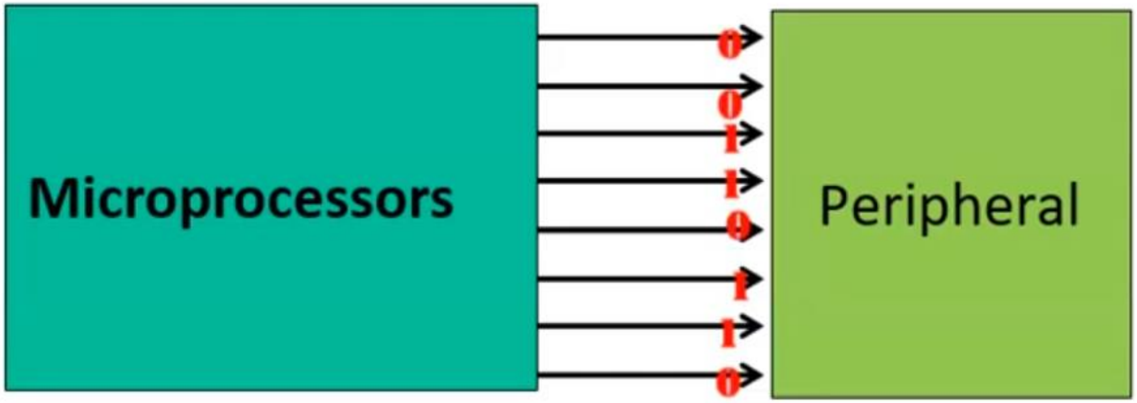
USART stands for **U**niversal **S**ynchronous and **A**synchronous **R**eceiver **T**ransmitter

8251 USART is a universal synchronous and asynchronous controller designed by Intel basically **to facilitate communication.**

For synchronous data transfer, **both the sender and receiver access the data according to the same clock**.

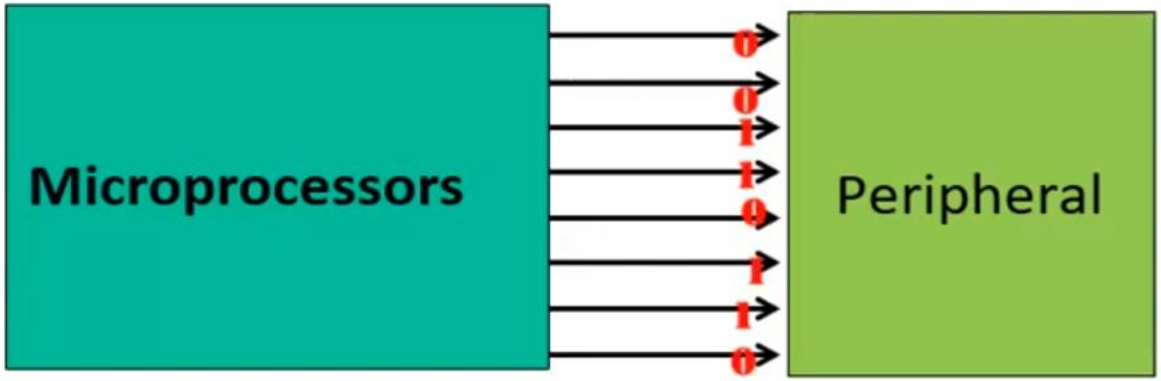
For asynchronous data transfer, there is no common clock signal between the sender and receivers.

Microprocessors allow parallel communication. And in parallel communication, the number of cables required for data transmission is equal to the number of bits to be transmitted



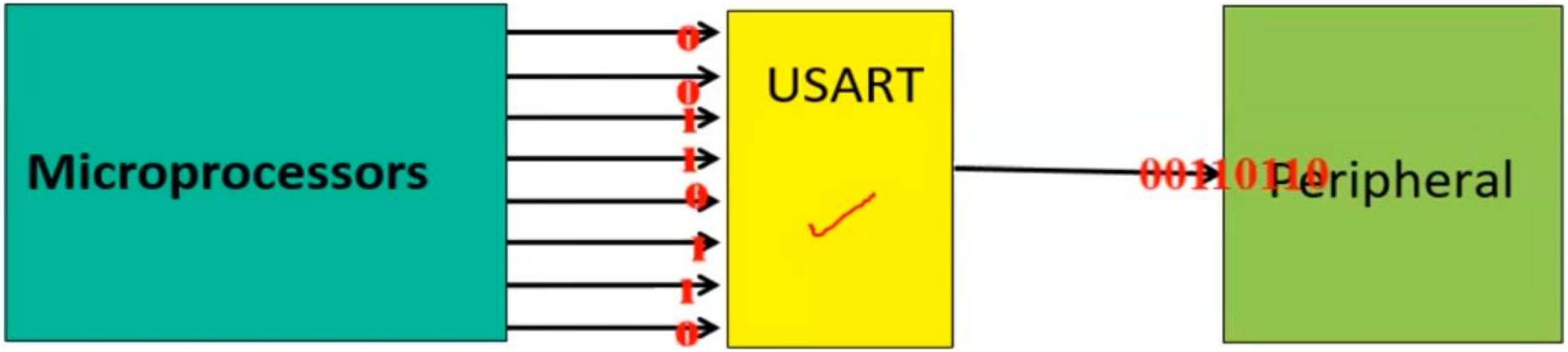
This approach of transmitting data parallelly to long distance is not cost-effective.

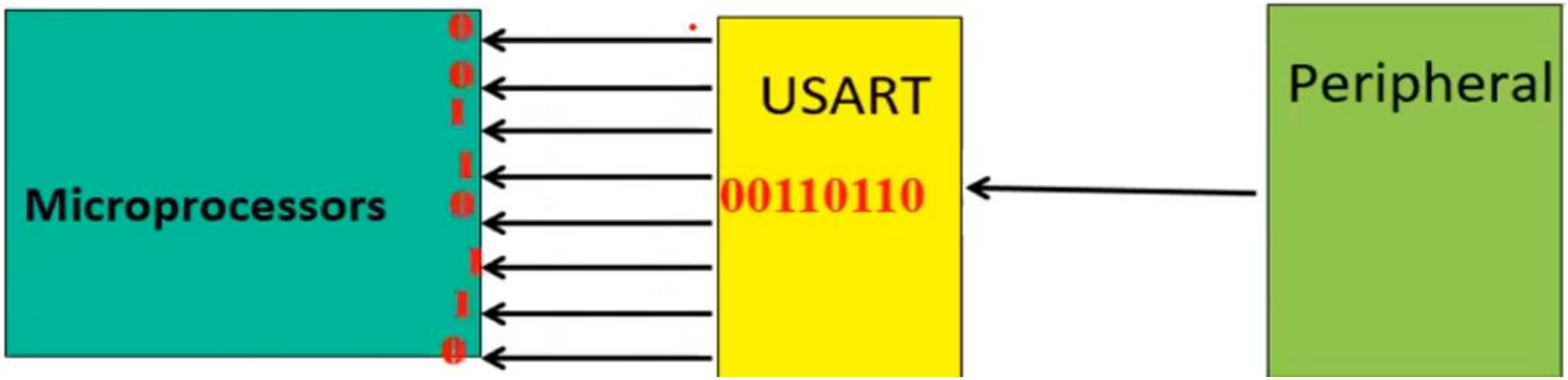
So, to reduce the overall cost of the system, the serial transfer of data is permitted.

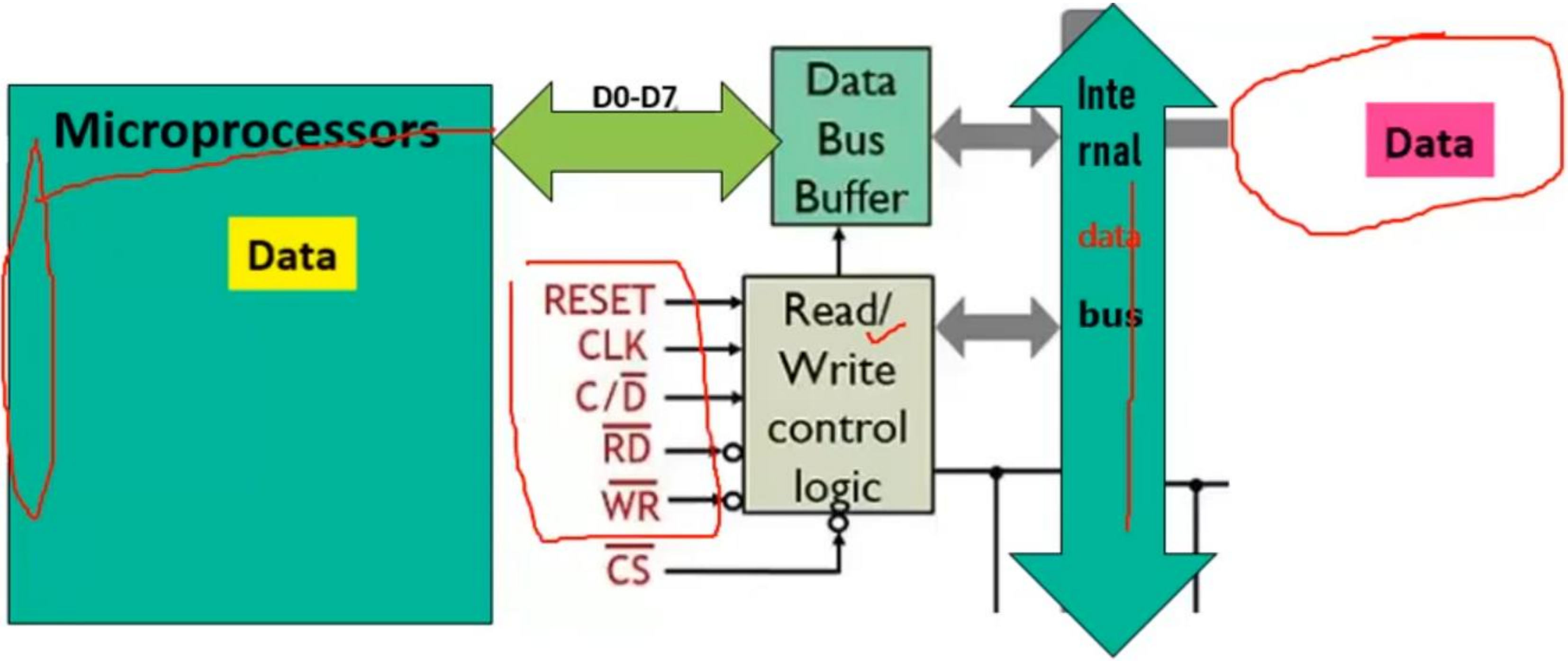


Peripherals to

- 8251 (USART) acts as a mediator between microprocessor and periph
  transmit serial data into parallel form and vice versa.





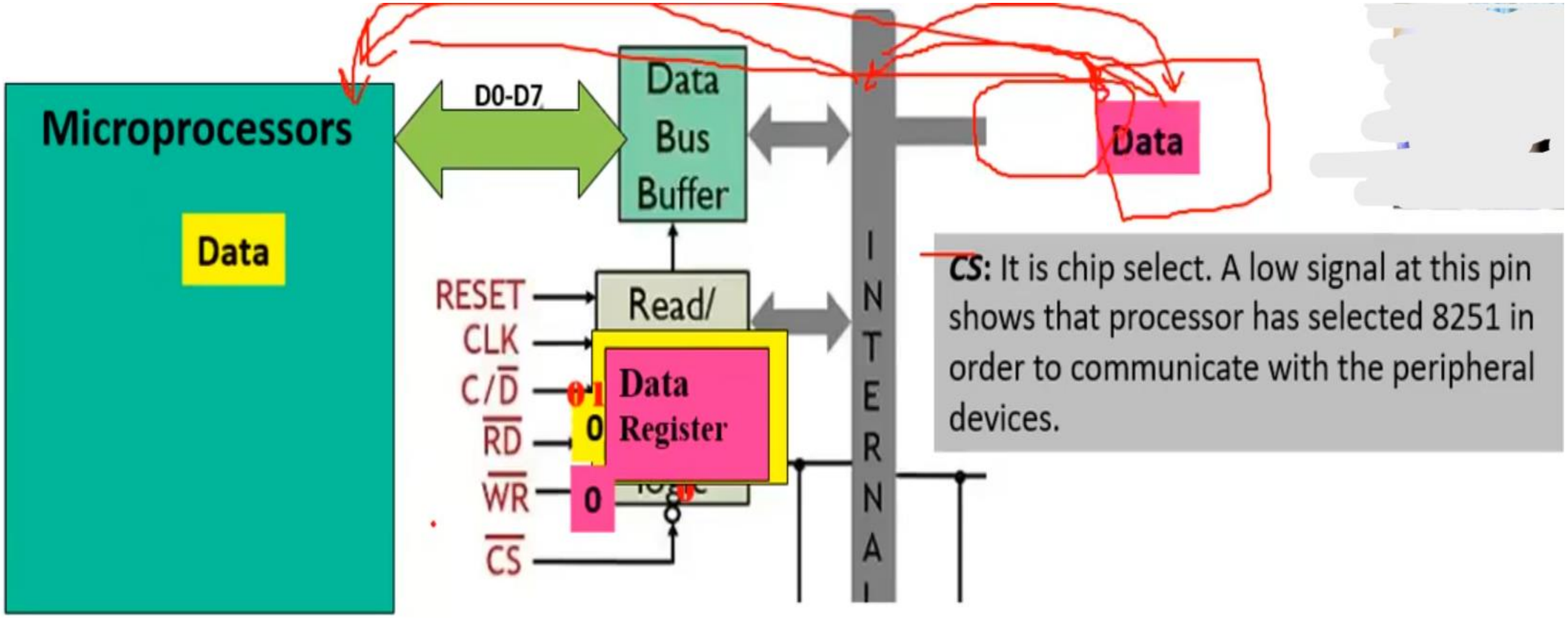


Data bus buffer –

This block helps in interfacing the internal data bus of 8251 to the system data bus. The data transmission is possible between 8251 and CPU by the data bus buffer block.

- **Read/Write control logic –**

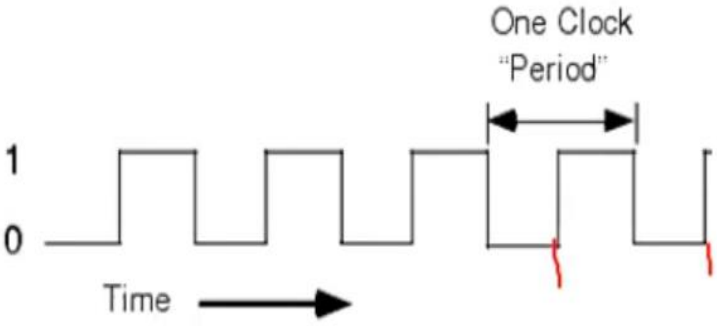
  It is a control block for overall device. It controls the overall working by selecting the operation to be done. The operation selection depends upon input signals as:

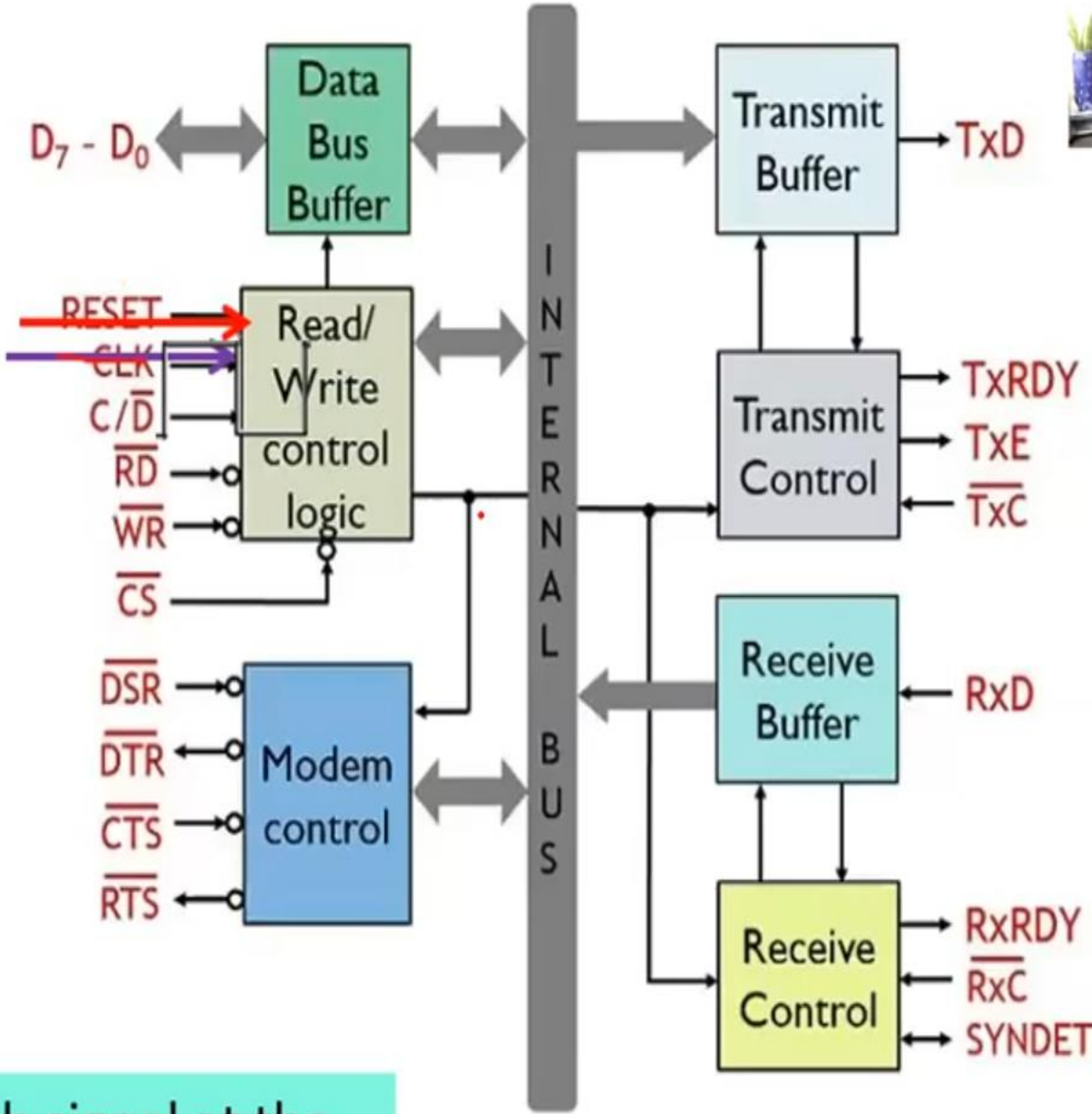


CS: It is chip select. A low signal at this pin shows that processor has selected 8251 in order to communicate with the peripheral devices.

C̄/D: system has control, status and data register. So, when a high signal is present at this pin then control or status register is addressed. While in case of low signal data register is addressed.

R̄D̄ **and** ***W̄R̄***: Both are active low signal . A low signal at RD shows that the processor is reading the control, status or data bytes from the 8251. While at WR indicates the write operation over the data bus of 8251.



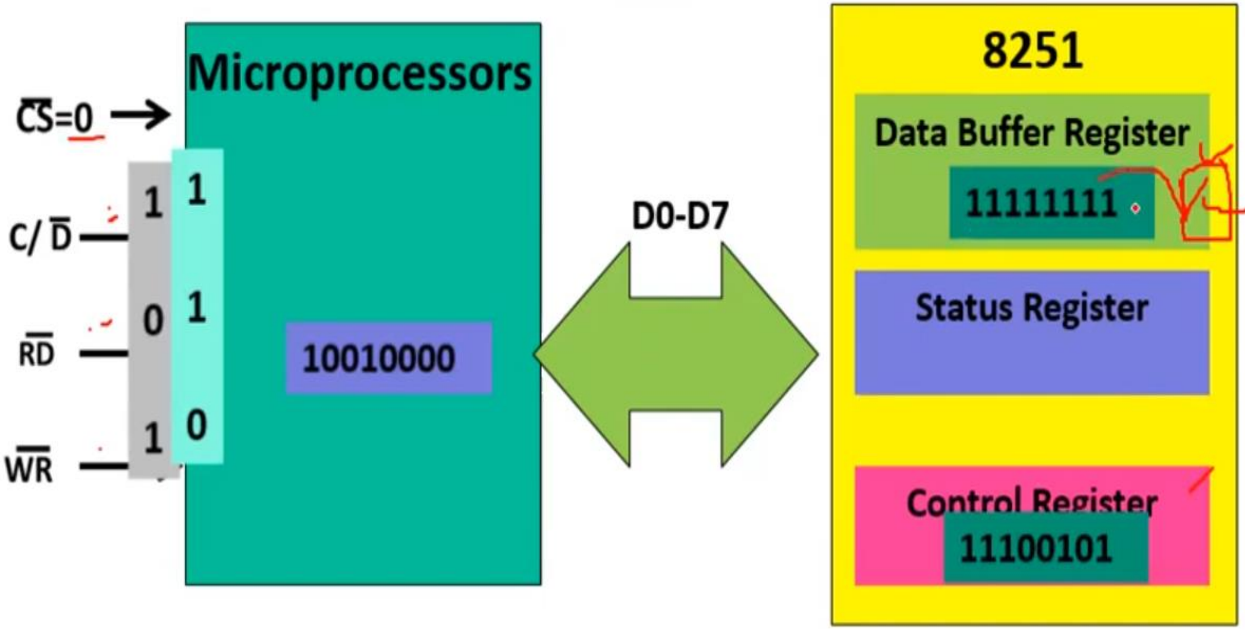


- ***CLK*** : CLK stands for clock and it produces the internal timing for the device.

RESET : While an active high signal at the RESET pin puts the 8251 in the idle mode.

$\overline{CS}$	C/$\overline{D}$	$\overline{RD}$	$\overline{WR}$	Operation
1	X	X	X	Invalid
0	0	0	1	data CPU <---- 8251
0	0	1	0	data CPU ----> 8251
0	1	0	1	Status word CPU <----------8251
0	1	1	0	Control word CPU------------ > 8251

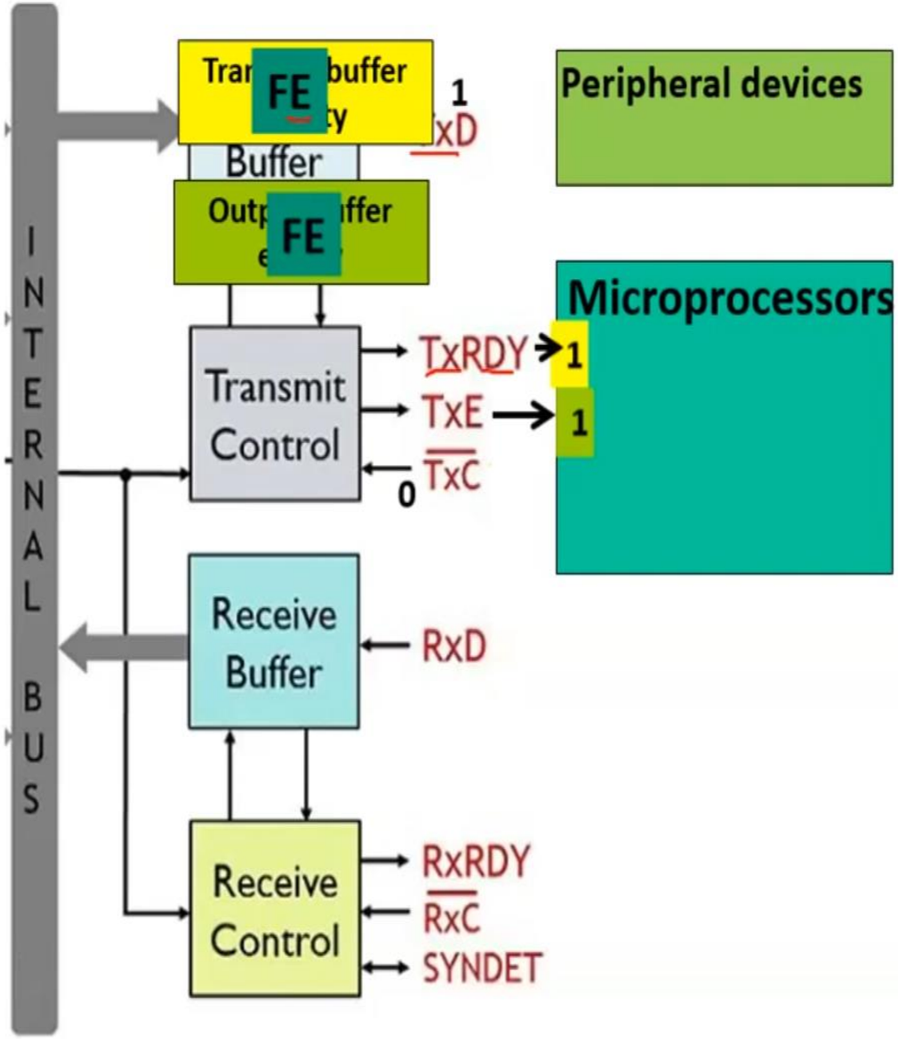
In this way, this unit selects one of the three registers- data buffer register, control register, status register.



Buffer register: Data provided by the processor is stored in the buffer register. We know that initially, the CPU provides parallel data to 8251. So, the processor loads the parallel data to the buffer register. Further, this data is fed to the output register.

Output register:

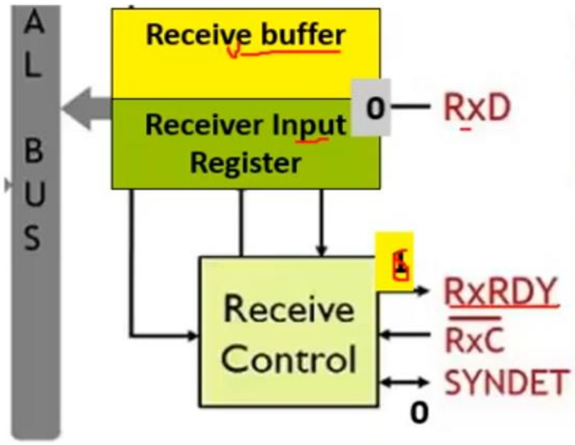
This register changes the 8-bit parallel data into a stream of serial bits



- **Transmit Control**: It controls the transmission of data. And it does so by accepting and sending signals both externally and internally.

TxRDY (Transmit ready) : This signal is used to notify the processor that the buffer register of the 8251 is empty and ready to accept the data.

TxC: It stands for transmitter clock and is an active low pin. It controls the rate of character transmission by the USART.



External Device

Microprocessor

full

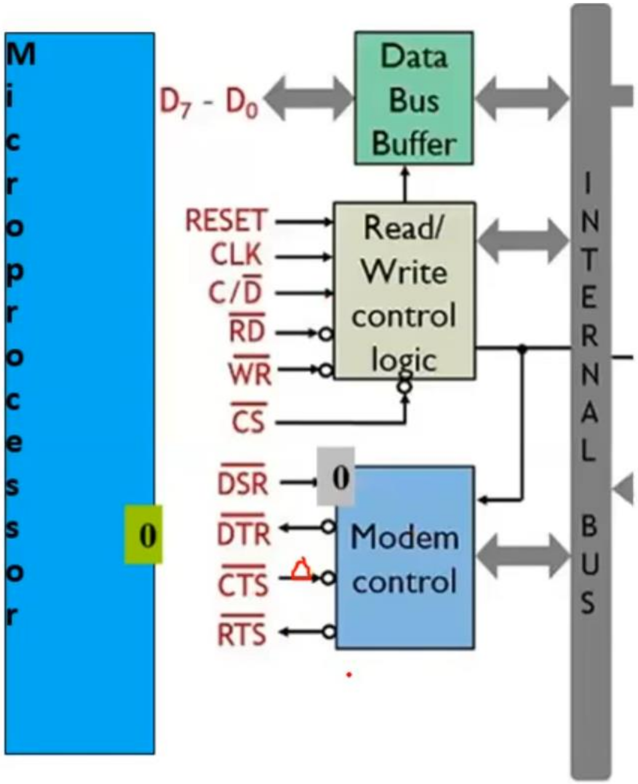
Receive Buffer: This unit takes the serial data from the external devices, changes the serial data into the parallel .It consists of 2 registers: receiver input register and buffer register.

RxRDY: It stands for receiver ready. When this signal goes high then it indicates that the receiver buffer register is holding the data and is ready to transfer it to the processor. Once the CPU reads the data sent by the 8251 then this pin is reset.

RxD: When the external device is ready to send the data to the 8251 then it sends a low signal to the RxD line of the 8251. In asynchronous mode, once 8251 receives a low signal it considers that signal as start bit of the data.

RxC: It stands for receiver clock. This clock signalling controls the rate at which the 8251 receives the data in the synchronous mode

SYNDET/BD: An input or output terminal. External synchronous mode-input terminal and asynchronous mode-output terminal.



Modem Control: This unit of 8251 holds input and output control signals that simplify the operation of the whole system.

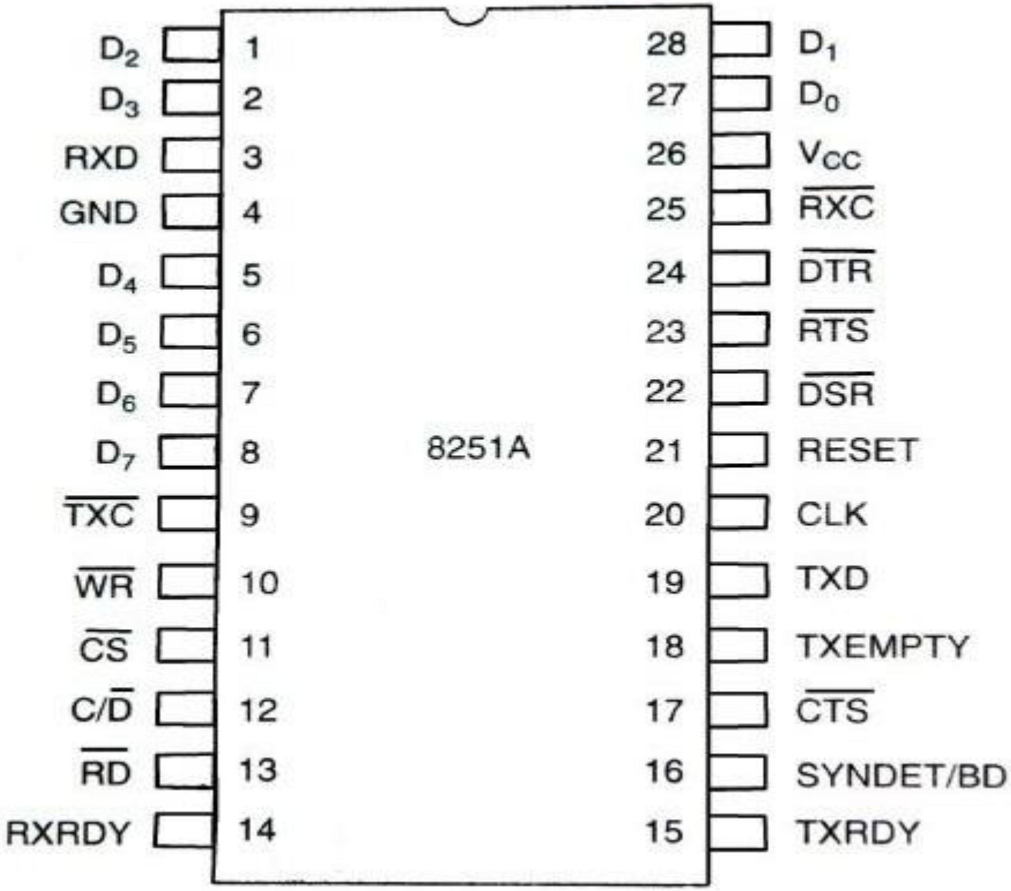
DSR (**Data Set Ready)** : This signal is used to check whether the data set is ready or not when the processor is in the urge of communication.

DTR (Data Terminal Ready) : This pin shows that the 8251 is now ready to accept the data from the processor.

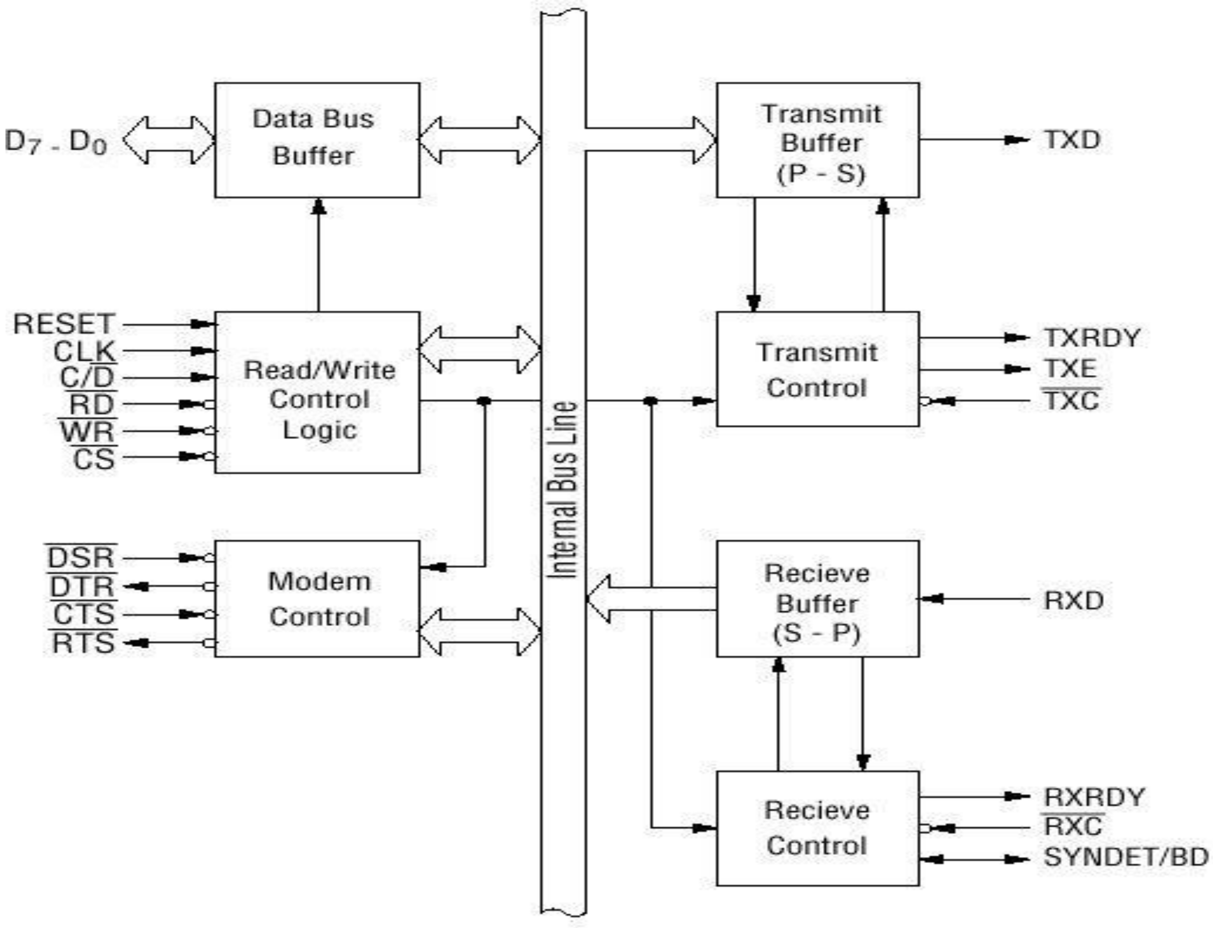
RTS **(Request To Send)** : A low signal shows an assertion for data transmission.

CTS **(Clear to send) :** When 8251 receives a low signal at this pin then it clears all the data present in the modem in order to allow further communication.

Pin diagram

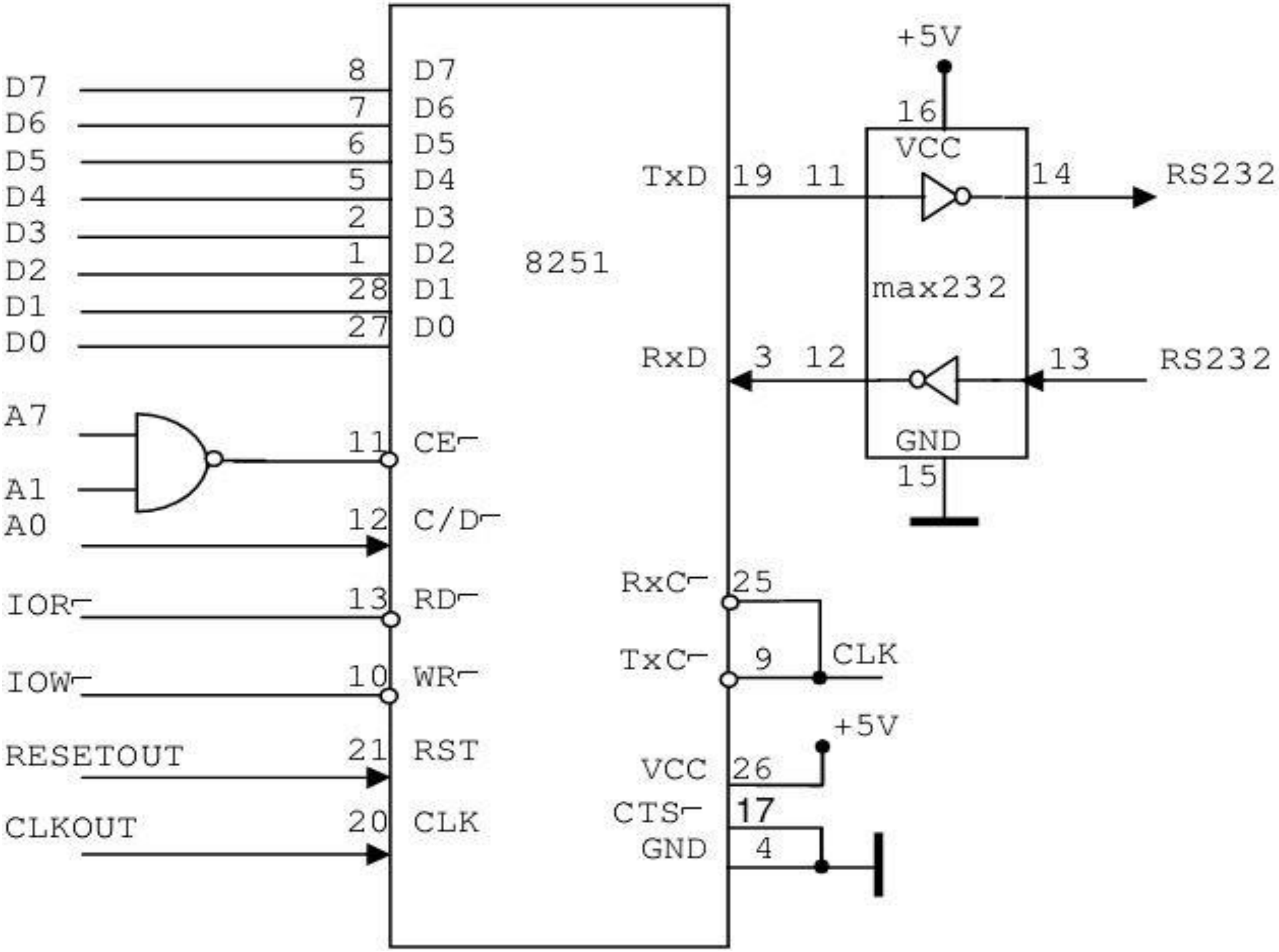

D_2 1
D_3 2
RXD 3
GND 4
D_4 5
D_5 6
D_6 7
D_7 8
$\overline{TXC}$ 9
$\overline{WR}$ 10
$\overline{CS}$ 11
$C/\overline{D}$ 12
$\overline{RD}$ 13
RXRDY 14
8251A
28 D_1
27 D_0
26 V_{CC}
25 $\overline{RXC}$
24 $\overline{DTR}$
23 $\overline{RTS}$
22 $\overline{DSR}$
21 RESET
20 CLK
19 TXD
18 TXEMPTY
17 $\overline{CTS}$
16 SYNDET/BD
15 TXRDY

8251: Block diagram



C/D–	: control or data write / read
DSR–	: data set ready
DTR–	: data terminal ready
CTS–	: clear to send data
RTS–	: request to send data
TxD	: transmitter data
TxRDY	: transmitter ready
TxE	: transmitter empty
TxC–	: transmitter clock
RxD	: receiver data
RxRDY	: receiver ready
RxC–	: receiver clock
SYNDET / BD	: sync detect / break detect

Interfacing RS 232 Terminal using the 8251A


D7
D6
D5
D4
D3
D2
D1
D0
8 D7
7 D6
6 D5
5 D4
2 D3
1 D2
28 D1
27 D0
8251
A7
A1
A0
11 CE
12 C/D
IOR
13 RD
IOW
10 WR
RESETOUT
21 RST
CLKOUT
20 CLK
+5V
16
VCC
TxD 19 11
14 RS232
max232
RxD 3 12
13 RS232
GND
15
RxC 25
TxC 9
CLK
+5V
VCC 26
CTS 17
GND 4