Data Communication

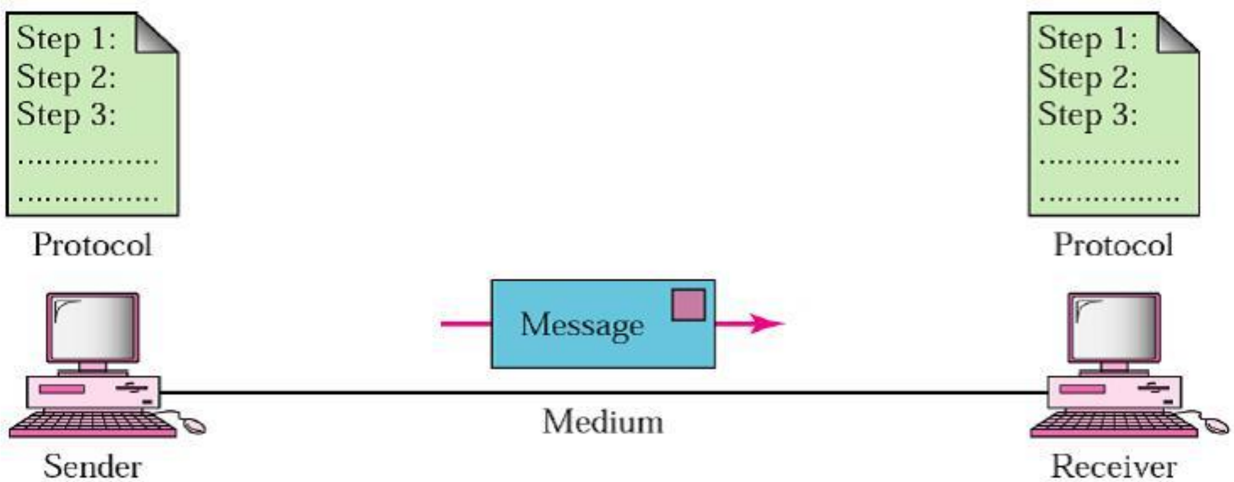
UNIT-1

The term “Data Communication” comprises two words: Data and Communication. Data can be any text, image, audio, video, and multimedia files. Communication is an act of sending or receiving data. Thus, data communication refers to the exchange of data between two or more networked or connected devices.

Components of Data Communication

A communication system is made up of the following components:

1. **Message:** A message is a piece of information that is to be transmitted from one person to another. It could be a text file, an audio file, a video file, etc.
2. **Sender:** It is simply a device that sends data messages. It can be a computer, mobile, telephone, laptop, video camera, or workstation, etc.
3. **Receiver:** It is a device that receives messages. It can be a computer, telephone mobile, workstation, etc.
4. **Transmission Medium / Communication Channels:** Communication channels are the medium that connect two or more workstations. Workstations can be connected by either wired media or wireless media.
5. **Set of rules (Protocol):** When someone sends the data (The sender), it should be understandable to the receiver also otherwise it is meaningless. For example, Sonali sends a message to Chetan. If Sonali writes in Hindi and Chetan cannot understand Hindi, it is a meaningless conversation.



Data representation

Data representation is the way information is represented and presented to users. In data communication, data can be represented as:

- Text, which is represented as a sequence of bits (0's or 1's)
- Numbers, which are also represented by bit patterns
- Images, which are also represented by bit patterns
- Audio, which refers to the recording or broadcasting of sound or music

Data Flow/ Transmission Mode

A data flow model is a diagram that shows how data moves through a system or program. It describes the processes involved in transferring data from input to file storage and reports generation. Data flows can also be called stream processing or reactive programming.

In Data Communication, networks are designed to allow communication to occur between individual devices that are interconnected. The flow of information, or data, between nodes, can take a variety of forms:

With simplex communication, all data flow is unidirectional: from the designated transmitter to the designated receiver. BogusBus is an example of simplex communication, where the transmitter sent information to the remote monitoring location, but no information is ever sent back to the water tank.

If all we want to do is send information one-way, then simplex is just fine. Most applications, however, demand more:

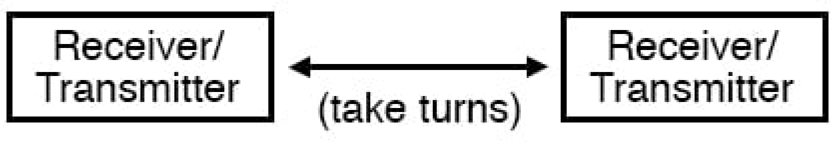
Duplex Communication

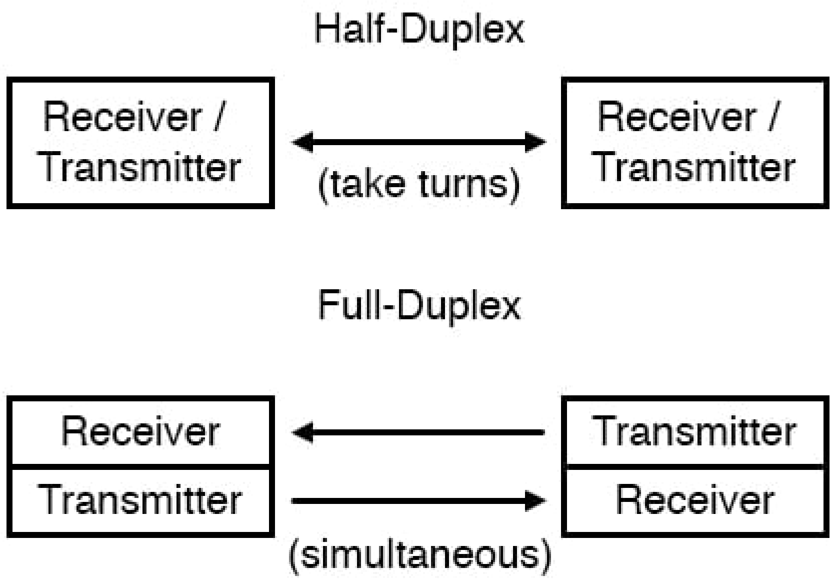
With duplex communication, the flow of information is bi-directional for each device. Duplex can be further divided into two sub-categories:

Half-Duplex

A type of communication may be likened to two tin cans on the ends of a single taut string: Either can may be used to transmit or receive, but not at the same time.

Full-duplex communication is more like a true telephone, where two people can talk at the same time and hear one another simultaneously, the mouthpiece of one phone transmitting the earpiece of the other, and vice versa.

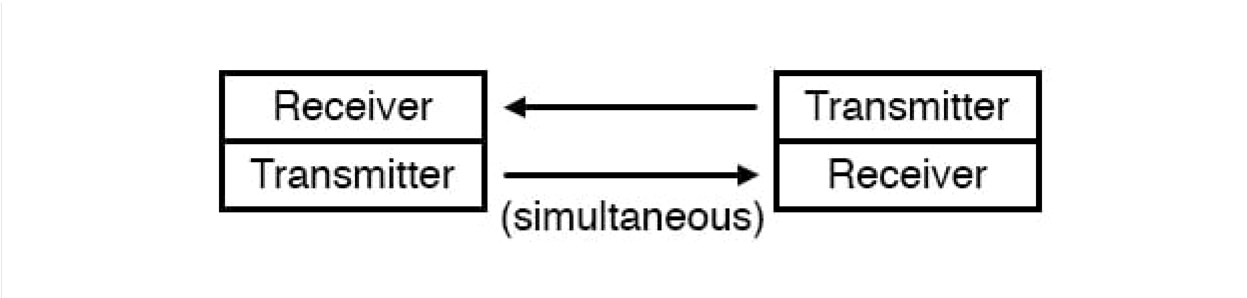


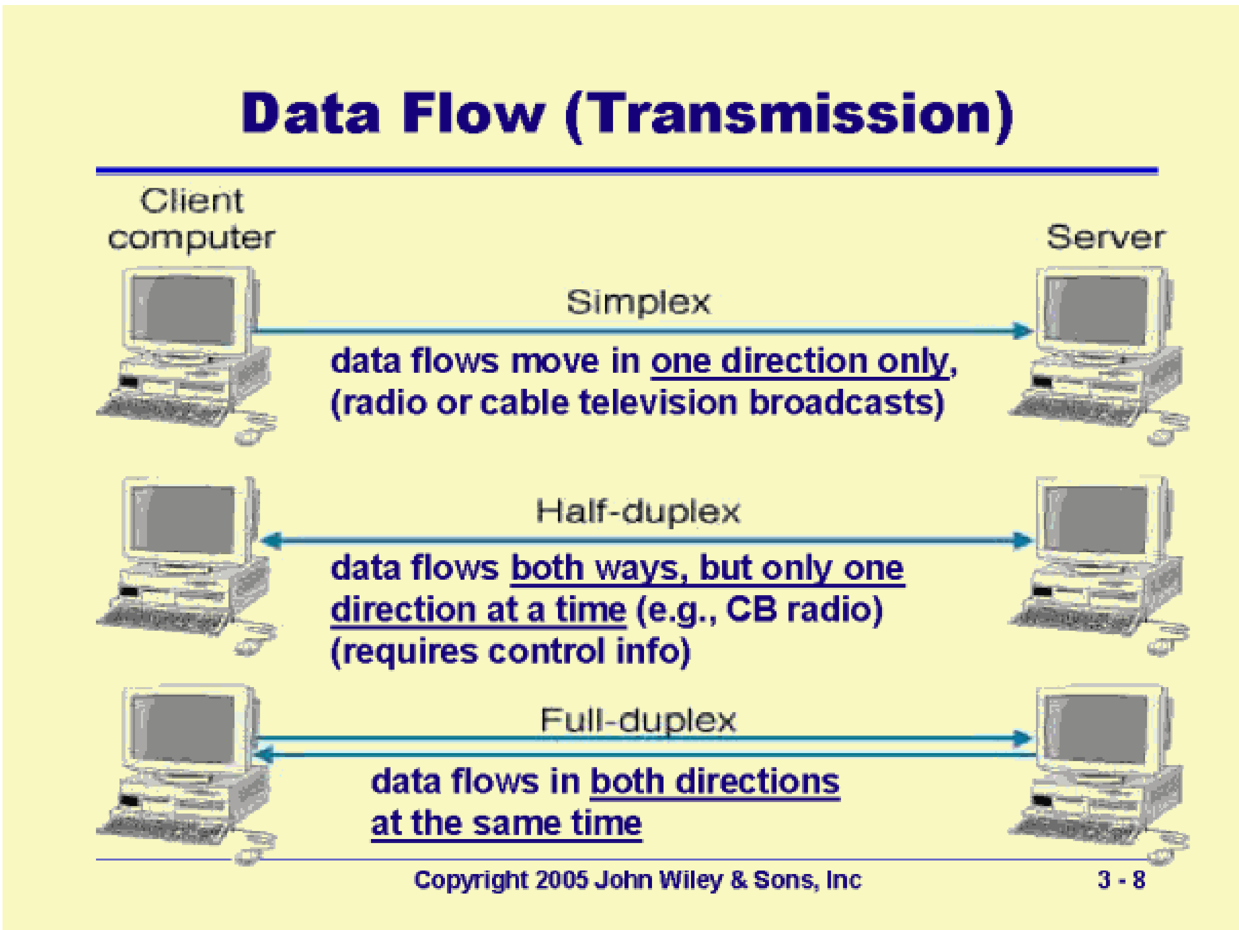


Full-Duplex

Full-duplex is often facilitated through the use of two separate channels or networks, with an individual set of wires for each direction of communication.

It is sometimes accomplished by means of multiple-frequency carrier waves, especially in radio links, where one frequency is reserved for each direction of communication.





The process of sending data between two or more digital devices is known as *data transmission*. Data is transmitted between digital devices using one of the two methods − *serial transmission* or *parallel transmission*.

In serial transmission, data bits are sent one after the other across a single channel. Parallel data transmission distributes numerous data bits through various channels at the same time.

What is Serial Transmission?

A serial transmission transfers data one bit at a time, consecutively, via a communication channel or computer bus in telecommunication and data transmission. On the other hand, parallel communication delivers multiple bits as a single unit through a network with many similar channels.

- 8-bits are conveyed at a time in serial transmission, with a start bit and a stop bit.
- All long-distance communication and most computer networks employ serial communication.
- Serial computer buses are becoming more common, even across shorter distances, since newer serial technologies' greater signal integrity and transmission speeds have begun to outperform the parallel bus's simplicity advantage.
- The majority of communication systems use serial mode. Serial networks may be extended over vast distances for far less money since fewer physical wires are required.

What is Parallel Transmission?

Parallel communication is a means of transmitting multiple binary digits (bits) simultaneously in data transmission. It differs from serial communication, which sends only one bit at a time; this distinction is one method to classify a communication channel.

- A parallel interface comprises parallel wires that individually contain data and other cables that allow the transmitter and receiver to communicate. Therefore, the wires for a similar transmission system are put in a single physical thread to simplify installation and troubleshooting.
- A large amount of data must be delivered across connection lines at high speeds that match the underlying hardware.
- The data stream must be transmitted through "n" communication lines, which necessitates using many wires. This is an expensive mode of transportation; hence it is usually limited to shorter distances.

Difference between Serial and Parallel Transmission

The following table highlights the major differences between Serial and Parallel Transmission −

Key	Serial Transmission	Parallel Transmission
Definition	Serial Transmission is the type of transmission in which a single communication link is used to transfer the data from one end to another.	Parallel Transmission is the mode of transmission in which multiple parallel links are used that transmit each bit of data simultaneously.
Bit transmission	In case of Serial Transmission, only one bit is transferred at one clock pulse.	In case of Parallel Transmission, 8-bits transferred at one clock pulse.

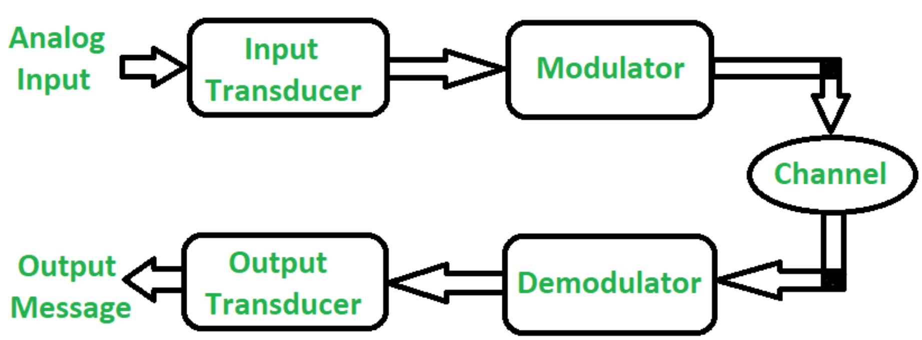
Key	Serial Transmission	Parallel Transmission
Cost Efficiency	As single link is used in Serial Transmission, it can be implemented easily without having to spend a huge amount. It is cost efficient.	Multiple links need to be implemented in case of Parallel Transmission, hence it is not cost efficient.
Performance	As single bit gets transmitted per clock in case of Serial Transmission, its performance is comparatively lower as compared to Parallel Transmission.	8-bits get transferred per clock in case of Parallel transmission, hence it is more efficient in performance.
Preference	Serial Transmission is preferred for long distance transmission.	Parallel Transmission is preferred only for short distance.
Complexity	Serial Transmission is less complex as compared to that of Parallel Transmission.	Parallel Transmission is more complex as compared to that of Serial Transmission.

Analog Communication and Digital Communication

1. Analog Communication:

In analog communication the data is transferred with the help of analog signal in between transmitter and receiver. Any type of data is transferred in analog signal. Any data is converted into electric form first and after that it is passed through communication channel. Analog communication uses a continuous signal which varies in amplitude, phase, or some other property with time in proportion to that of a variable.

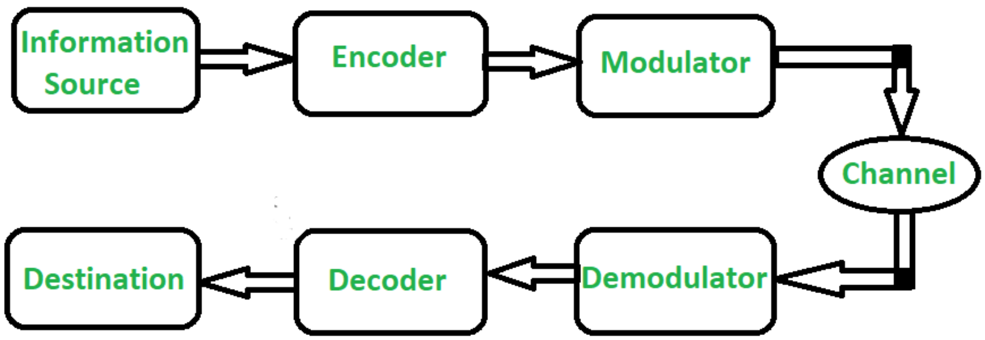
The below figure illustrates the **Analog Communication System:**



Analog Communication System

2. Digital Communication:
In digital communication digital signal is used rather than analog signal for communication in between the source and destination. The digital signal consists of discrete values rather than continuous values. In digital communication physical transfer of data occurs in the form of digital bit stream i.e 0 or 1 over a point-to-point or point-to-multipoint transmission medium. In digital communication the digital transmission data can be broken into packets as discrete messages which is not allowed in analog communication.

The below figure illustrates the **Digital Communication System**:



Digital Communication System

Difference between Analog Communication and Digital Communication:

S.No.	ANALOG COMMUNICATION	DIGITAL COMMUNICATION
01.	In analog communication analog signal is used for information transmission.	In digital communication digital signal is used for information transmission.
02.	Analog communication uses analog signal whose amplitude varies continuously with time from 0 to 100.	Digital communication uses digital signal whose amplitude is of two levels either Low i.e., 0 or either High i.e., 1.
03.	It gets affected by noise highly during transmission through communication channel.	It gets affected by noise less during transmission through communication channel.

S.No.	ANALOG COMMUNICATION	DIGITAL COMMUNICATION
04.	In analog communication only limited number of channels can be broadcasted simultaneously.	It can broadcast large number of channels simultaneously.
05.	In analog communication error Probability is high.	In digital communication error Probability is low.
06.	In analog communication noise immunity is poor.	In digital communication noise immunity is good.
07.	In analog communication coding is not possible.	In digital communication coding is possible. Different coding techniques can be used to detect and correct errors.
08.	Separating out noise and signal in analog communication is not possible.	Separating out noise and signal in digital communication is possible.
09.	Analog communication system is having complex hardware and less flexible.	Digital communication system is having less complex hardware and more flexible.
10.	In analog communication for multiplexing Frequency Division Multiplexing (FDM) is used.	In Digital communication for multiplexing Time Division Multiplexing (TDM) is used.
11.	Analog communication system is low cost.	Digital communication system is high cost.
12.	It requires low bandwidth.	It requires high bandwidth.
13.	Power consumption is high.	Power consumption is low.

S.No.	ANALOG COMMUNICATION	DIGITAL COMMUNICATION
14.	It is less portable.	Portability is high.
15.	No privacy or privacy is low, so it is not highly secured.	Privacy is high, so it is highly secured.
16.	Not assures an accurate data transmission.	It assures a more accurate data transmission.
17.	Synchronization problem is hard.	Synchronization problem is easier.

Encoding -

Digital To Digital Conversion.

It is the process of converting binary data, sequence of bits to a digital signal.

Classification of Line Codes:

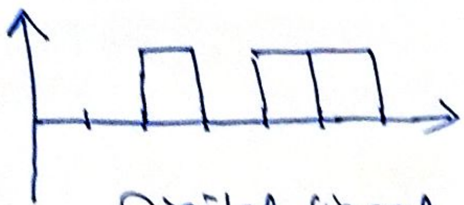
- Unipolar → Only one voltage level other than zero
  - RZ (Return to zero)
  - NRZ (Not return to zero)

  (+A) or 0
  ↓ ↓
  1 0

- Polar → Two voltage levels other than zero
  +A/2 and -A/2

- Bipolar → Three voltage levels
  1) Positive 2) Negative 3) Zero

| 0 | 0 | 1 | 0 | 1 | 1 | → | Line Coding | →

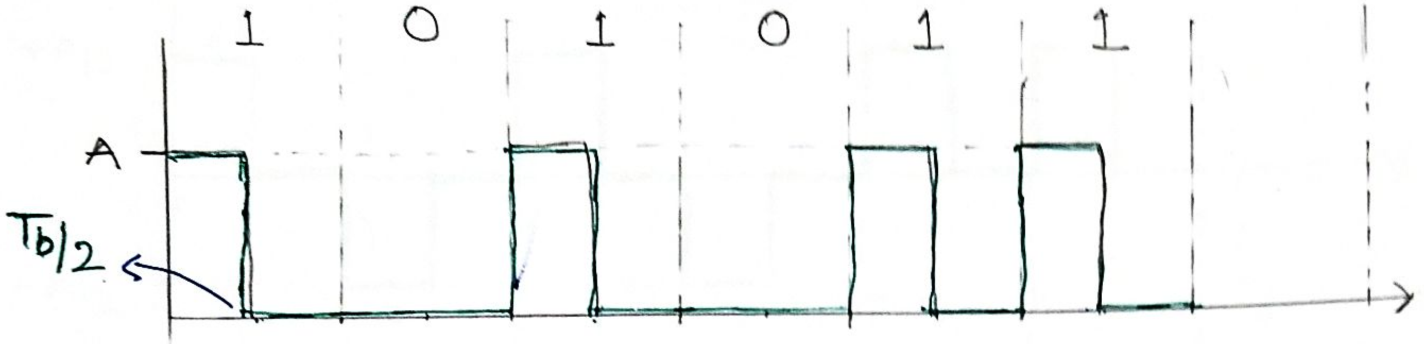


Sequence of bits

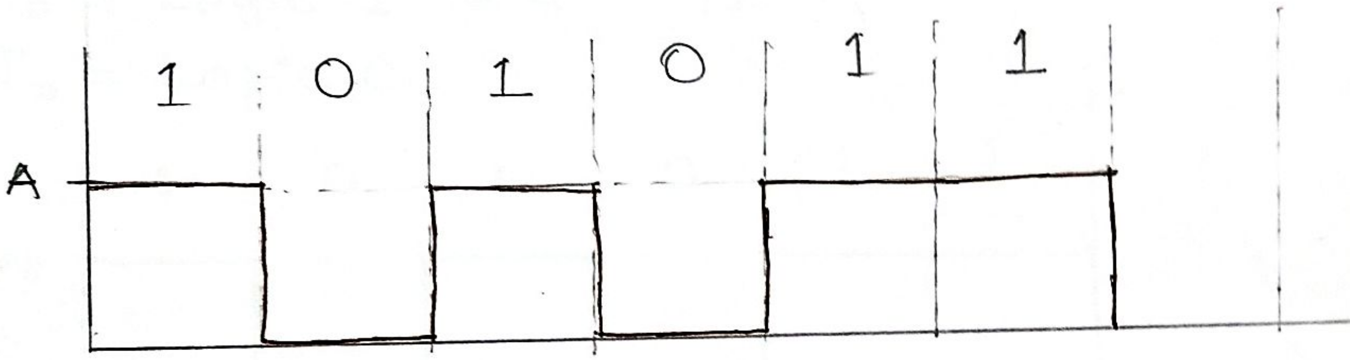
Digital signal

Types of UniPolar

(i) Unipolar RZ Format: Each 0 = Off Pulse $[a_k = 0]$ Each 1 = ON Pulse $[a_k = A]$ and duration of $T_b/2$ followed by return to zero level.

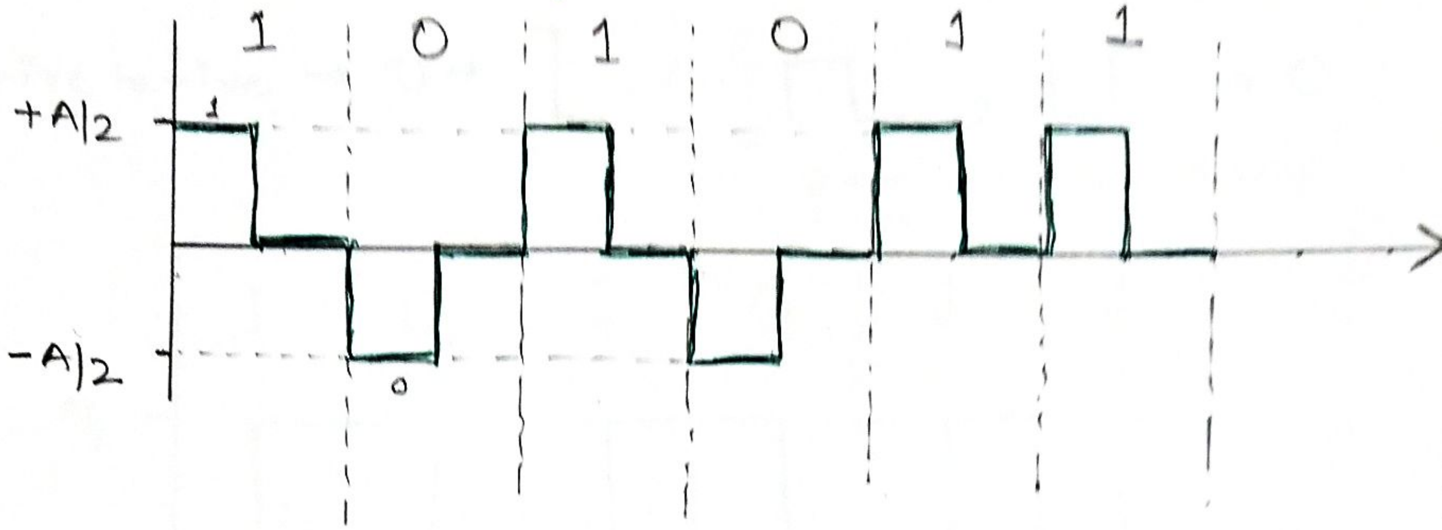


(ii) Unipolar NRZ Format: Each 1 = Pulse of full bit duration T_b and amplitude $= +A$.
Each 0 = Off Pulse.

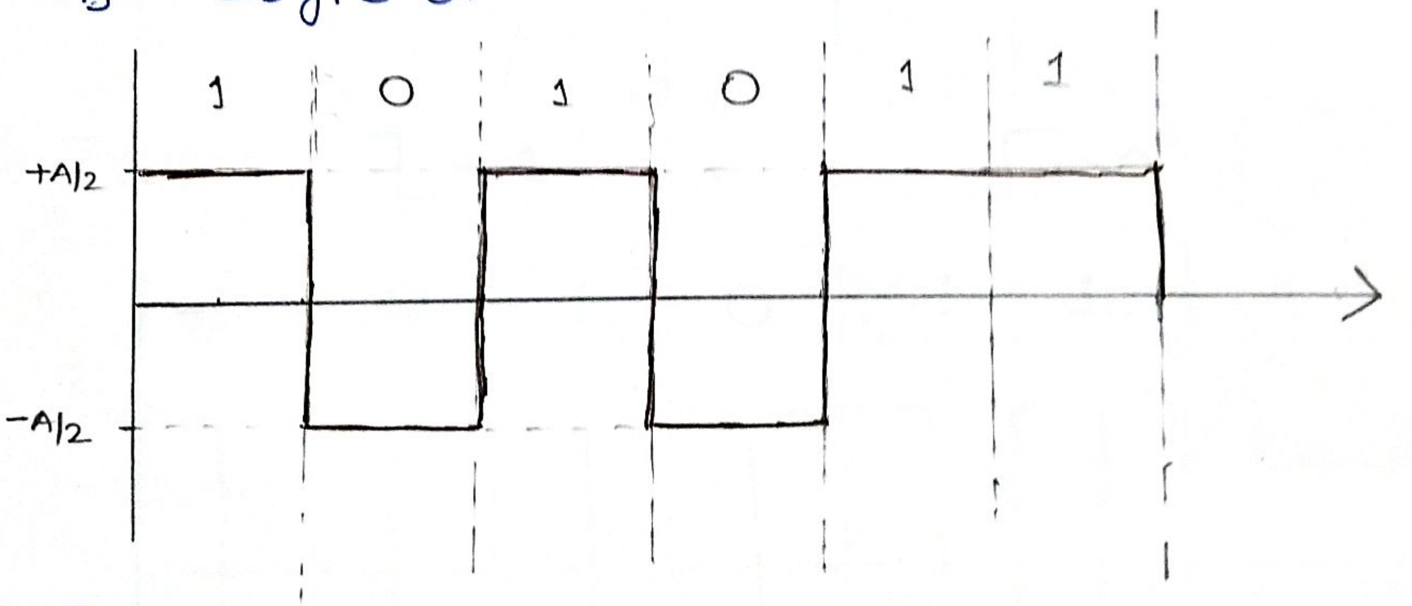


Types of Polar

(i) Polar RZ format : $+A/2$ = Logic 1 and $-A/2$ = Logic 0. Bit duration $= T_b/2$.



(ii) Polar NRZ format : $+A/2$ of duration T_b = Logic 1 and $-A/2$ of duration T_b = Logic 0.



(iii) Biphase : The signal changes at the middle of the bit interval but does not return to zero. Instead, it continues to the opposite pole.
1) Manchester.
2) Differential Manchester.

(a) Manchester :

IEEE 802.3

-ive to +ive → 1 →

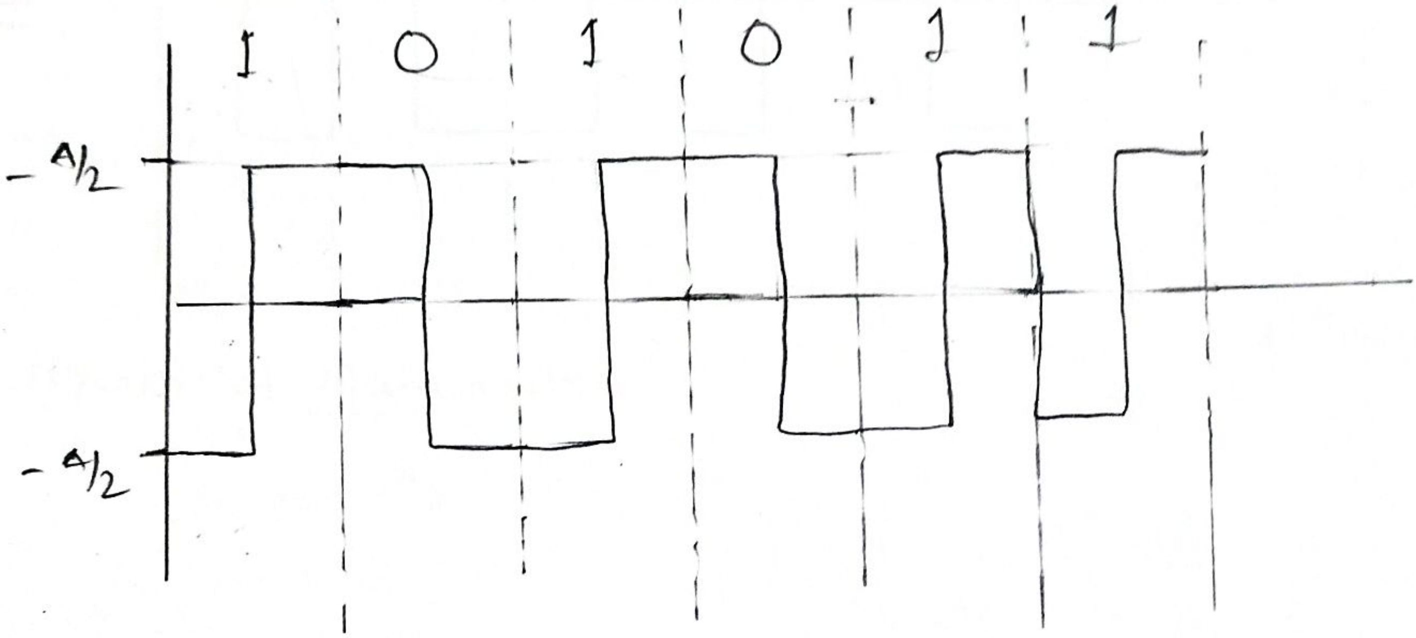
+ive to -ive → 0 →

(b) Differential Manchester

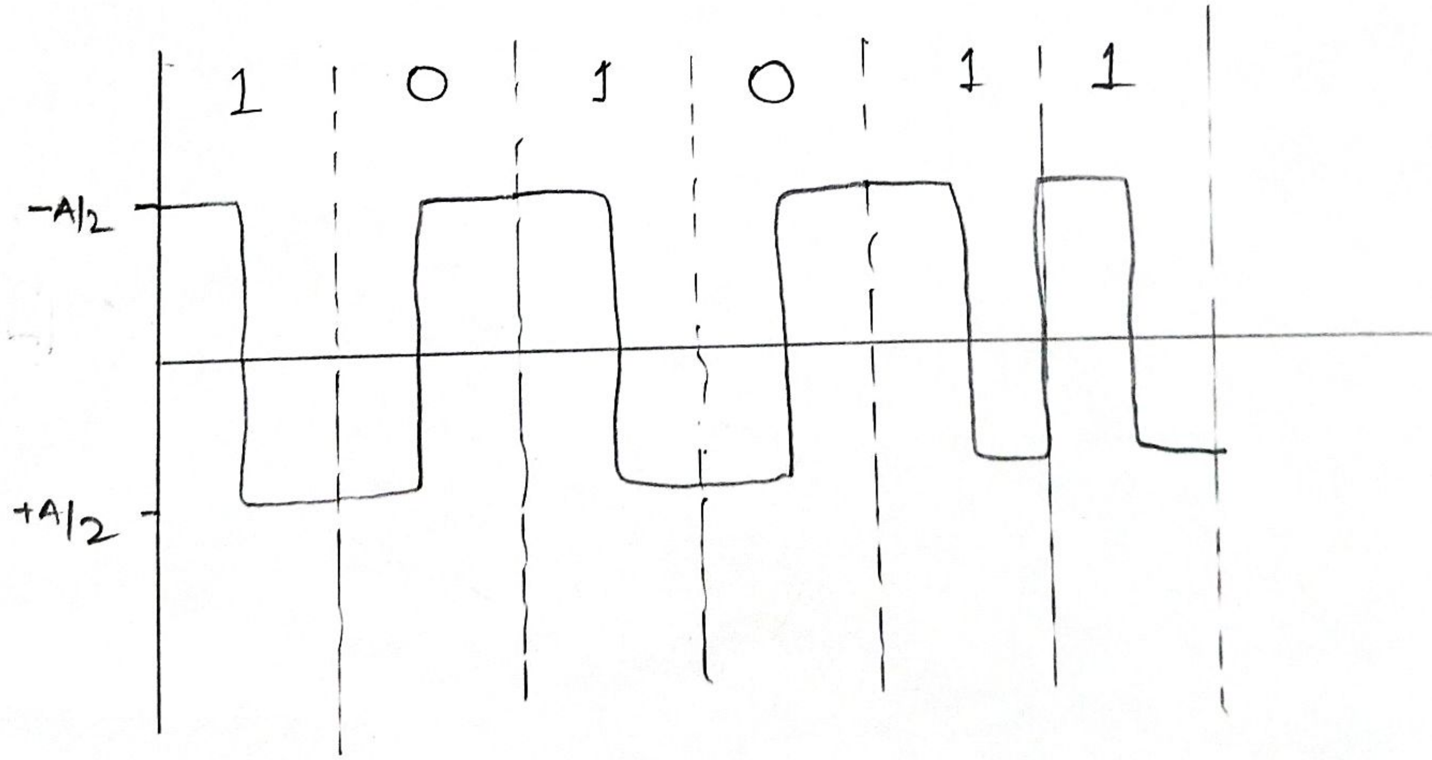
, → 1

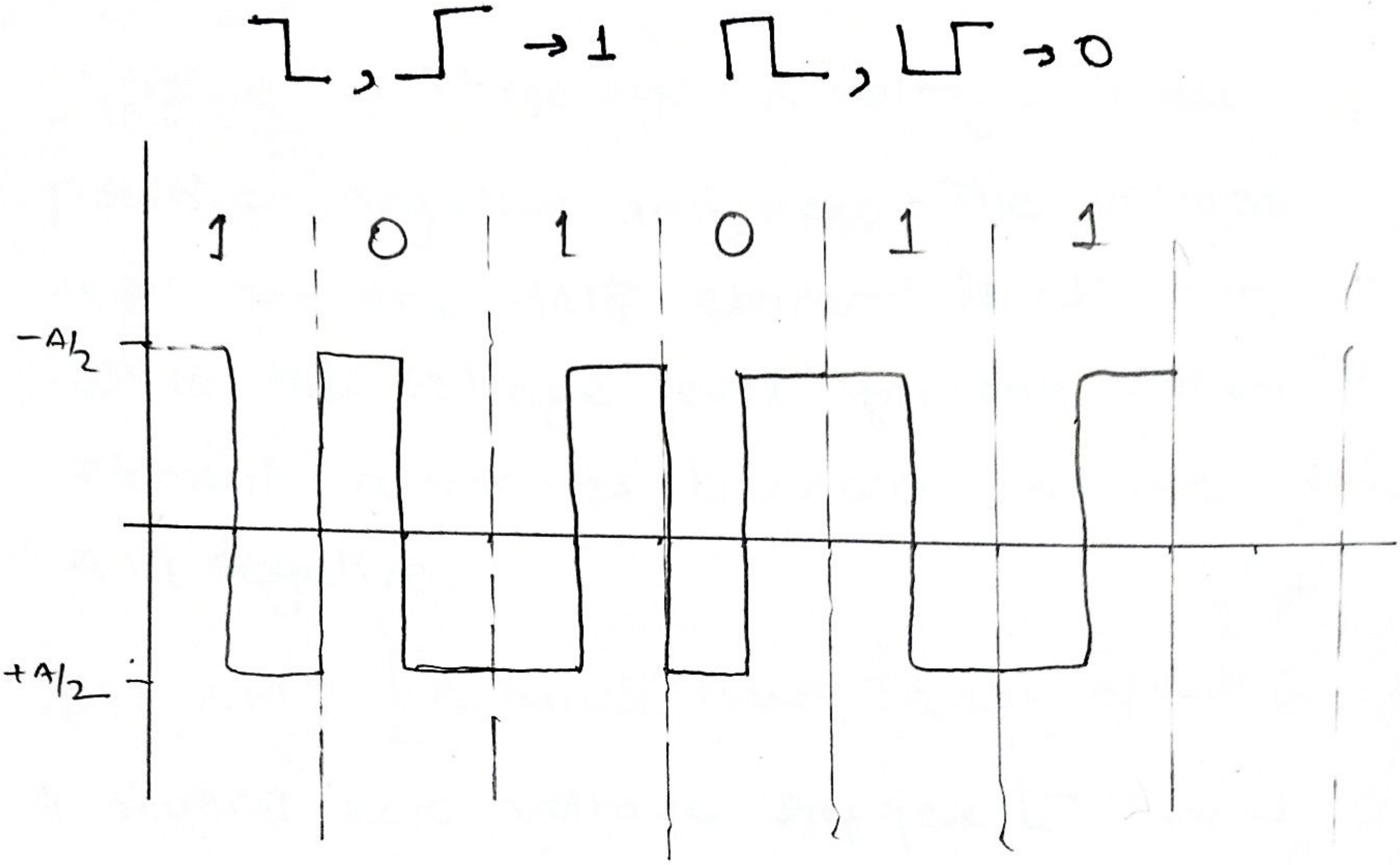
, → 0

0 should contain an edge
1 not



Dr. THOMAS → 1 → 0





Differential Manchester.

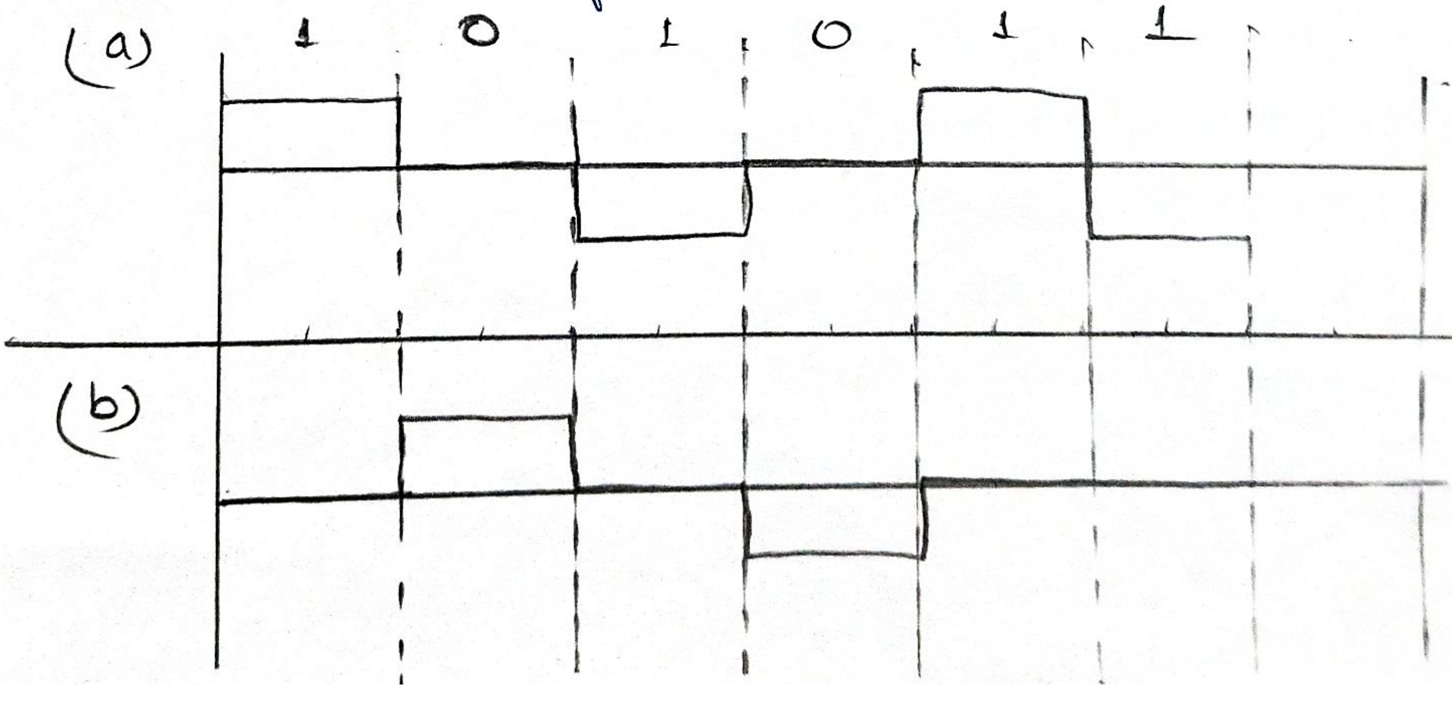
Bipolar — There are 3 voltage levels positive, negative and zero. The voltage level for one data element is at zero, while the voltage level for the other element alternates between positive and negative.

(a) AMI [Alternate Mark Inversion] —

A neutral zero voltage represents binary 0. Binary 1's are represented by alternating positive and negative voltages.

(b) Pseudo-ternary — Bit 1 is encoded as a zero voltage and the bit 0 is encoded as alternating positive and negative voltages i.e, opposite of AMI scheme.

(a)



(b)

Data Compression -

Compression is the function of presentation Layer in OSI refrence model.
It is often used to maximize the use of bandwidth across a network or to optimize disk space when saving data.

In compression, we reduce the volume of data to be exchanged.

2 broad categories:

1) Lossless Compression → Run length, Relative encoding, Huffman [Frequency dependent codes]
2) Lossy Compression

(1) Lossless Compression :

In lossless compression, the integrity of the integrity of the data is preserved because the compression and decompression algorithms are exact inverses of each other: no part of the data is lost in the process. It is used when we cannot afford to lose any data.

(a) Run length encoding :

This method replaces the consecutive occurrences of a given symbol with only one copy of the symbol along with a count of how many times that symbol occurs. Hence the name 'run length'.

e.g. AAABBCDDDDD $\xrightarrow[\text{encoded as}]{}$ 3A2B1C4D.

(b) Relative Encoding :

is a transmission technique that attempts to improve efficiency by transmitting the difference between each value and its predecessor, in place of the value itself.

e.g. 1 5 1 0 6 4 3 3 0 0 3

↓

1+4-4-1+6-2 -1+0-3+0+3

(c) Huffman Coding : [Frequency dependent Codes]

It is a popular compression technique that assigns variable length codes [VLC] to symbols, so that the most frequently occurring symbols have the shortest codes.

On decompression the symbols are reassigned their original fixed length codes. When used to compress text, for example, variable length codes ~~use~~ are used in place of ASCII codes, and the most common characters, usually spaces, e and t are assigned the shortest codes. In this way the total number of bits required to transmit the data can be considerably less than the number required if the fixed length representation is used.

Huffman encoding is particularly effective where the data are dominated by a small number of symbols.

e.g.

Symbol	Code
A	00
B	01
C	10
D	11

ABACCDA

00010010101100

14 bits

A	0
B	110
C	10
D	111

ABACCDA

0 1100 1010 111 0

13 bits

LZW Compression –

It is a method to reduce the size of Tag Image file format (TIFF) or (GIF) Graphics Interchange format.

It is a table-based lookup algorithm to remove duplicate data and compress an original file into a smaller file.

It is also suitable for compressing text and PDF files.

The algorithm is loosely based on the LZ78 algorithm that was developed by Abraham Lempel and Jacob Ziv in 1978.

LZW is invented by Abraham Lempel, Jacob Ziv and Terry Welch in 1984.

It is a type of lossless compression.

It is part of Unix OS's file compression utility.

It is often used for general purpose data compression in many PC utilities.

Basic idea LZW

→ Create a dictionary (a table) of strings used during communication.

→ If both sender and receiver have a copy of the dictionary, then previously-encountered strings can be substituted by their index in the dictionary.

→ Have 2 phases: 1) Building an index dictionary

2) Compressing a string of symbols.

Algorithm —

1) Extract the smallest substring that cannot be found in the remaining uncompressed string.

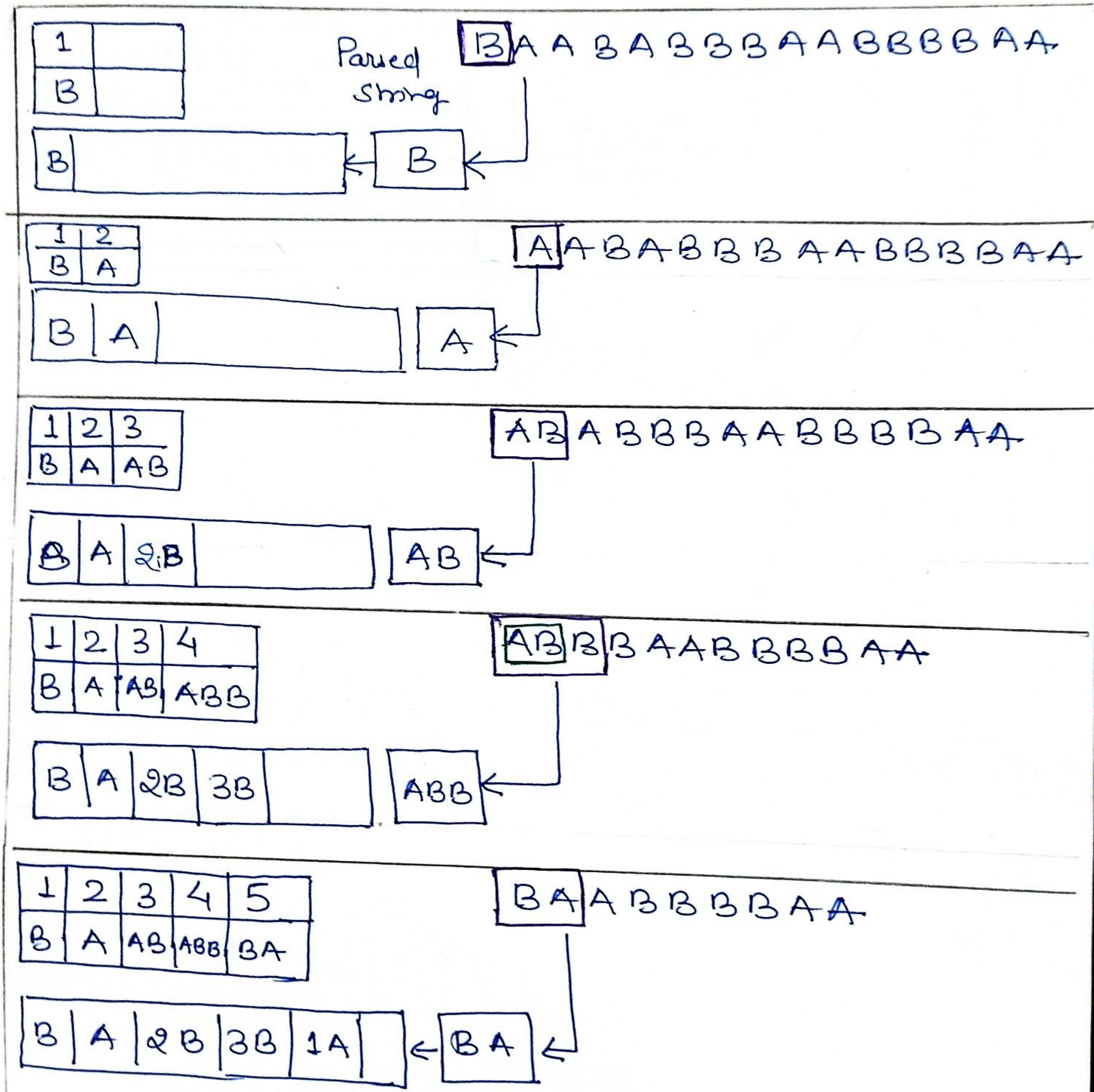
(2) Store that substring in the dictionary as a new entry and assign it an index value.

(3) Substring is replaced with the index found in the dictionary.

(4) Insert the index and the last character of the substring into the compressed string.

Example of LZW Compression technique —

Uncompressed : BAABABBBAABBBBAA



$\boxed{ABBB}$ BAA

1	2	3	4	5	6
B	A	AB	ABB	BA	ABBB

B	A	2B	3B	1A	4B	

← $\boxed{ABBB}$

$\boxed{BAA}$

1	2	3	4	5	6	7
B	A	AB	ABB	BA	ABBB	BAA

B	A	2B	3B	1A	4B	5A

← $\boxed{BAA}$

Lossy Compression :

Lossy compression reduces bits by removing unnecessary or less important information.
So we need data compression mainly because :

→ Uncompressed data can take up ~~to~~ a lot of space, which is not good for limited hard drive space and internet download speeds.

→ While hardware gets better and cheeper, algorithms to reduce data size also help Technology evolves.

⇒

NYQUIST THEOREM

Two theoretical formulas were developed to calculate the data rate: one by Nyquist for noiseless channel, another by Shannon for a noisy channel.

(A) Noiseless Channel : Nyquist Bit Rate

$$\text{Bit rate} = 2 \times \text{bandwidth} \times \log_2 L$$

L is no. of signal levels.

→ Increasing the levels of the signal may reduce the reliability of a system.

(B) Noisy Channel : Shannon Capacity

$$\text{Capacity} = \text{bandwidth} \times \log_2 (1 + SNR)$$

where SNR = signal-to-noise ratio

$$= \frac{\text{average signal power}}{\text{average noise power}}$$