UNIT-2

By Govind Patidar

We must transform data into signals to send them from one Place to another.

A simple signal by itself doesn't carry information any more than a straight line conveys words. First information must be translated into, agreed-upon patterns of 0's and 1's for example using ASCII

Digital to Digital :→ To carry the stored 1 or 0 in computer data are usually converted to digital signals. This is called digital to digital conversion or encoding digital data into a digital signal.

Analog to ~~Analog~~ Digital :→ Sometimes we need to convert an analog signal (such as voice in telephone conversation) into a digital signal for several reasons, such as decrease the effect of noise. This is called analog-to-digital conversion (A/D) or digitizing an analog signal.

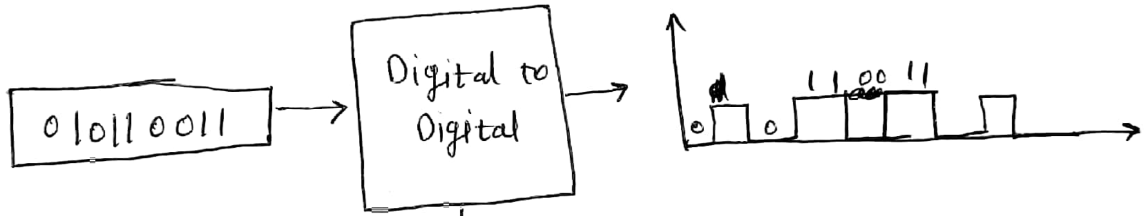
Digital to Analog :→ Whenever we want to send digital signal coming out of a computer through a medium designed for an analog signal. For ex. send data from one Place to another Place using Public telephone line, the digital signal Produce by computer should be converted to an analog signal. This is called digital-to digital conversion or modulating a digital signal.

Analog to Analog :→ Analog Signal is sent over long distance using analog media. for ex voice or music in air. However freq of voice or music is not appropriate for this kind of transmission, signal should be carried by high freq signal. This is called Analog to Analog conversion (A/A)

Conversion Methods

- Digital/Digital
- Analog/Digital
- Digital/Analog
- Analog/Analog

1) **Digital to digital Conversion** :→ Representation of digital information by a digital signal.
In this encoding, the binary 1s and 0s generated by a computer are translated into a sequence of voltage pulses that can be propagated over a wire.



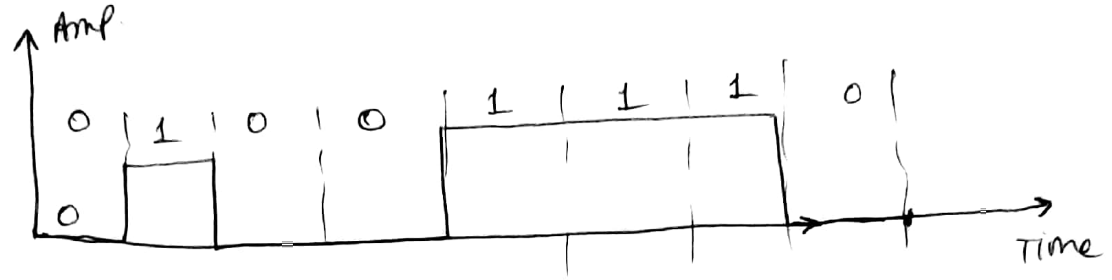
- Unipolar
- Polar
  - NRZ
    - NRZ-L
    - NRZ-I
  - RZ
  - Biphase
    - Manchester
    - Differential Manchester
- Bipolar
  - AMI
  - B8ZS
  - HDB3

Unipolar :→ very simple, obsolete today
→ It has only one polarity
→ Usually
1's → +ve voltage
0's → zero voltage

Govind Patidar.

35 '6
77 3

Example



Two problem

1) DC component :→ The average Amp. of unipolar encoded signal is non zero. This creates what is called a direct current (DC) component (a component with zero Freq). when a signal contain DC component, it can not travel through media that can not handle DC component

2) Synchronization :→ when signal is not varying. the receiver can not determine the begning and ending of each bit. Specially when long 0's or 1's are appearing. Receiver use timer for such problem but there may be mismatch in timining also. that may cause the Problem. this problem can be avoided by using Parallel line that carries a clock Pulse but it will increase the cost.

~~Polar:→~~

Polar :→ uses two voltage levels :→ +ve and -ve

By using both levels DC component problem is alleviated.

NRZ :→ The level of the signal is always either +ve or -ve.

Non Return to zero, Level

a) NRZ-L :→ Level of the signal depends on the type of bit it represents.

Usually → +ve voltage → 0
→ -ve voltage → 1

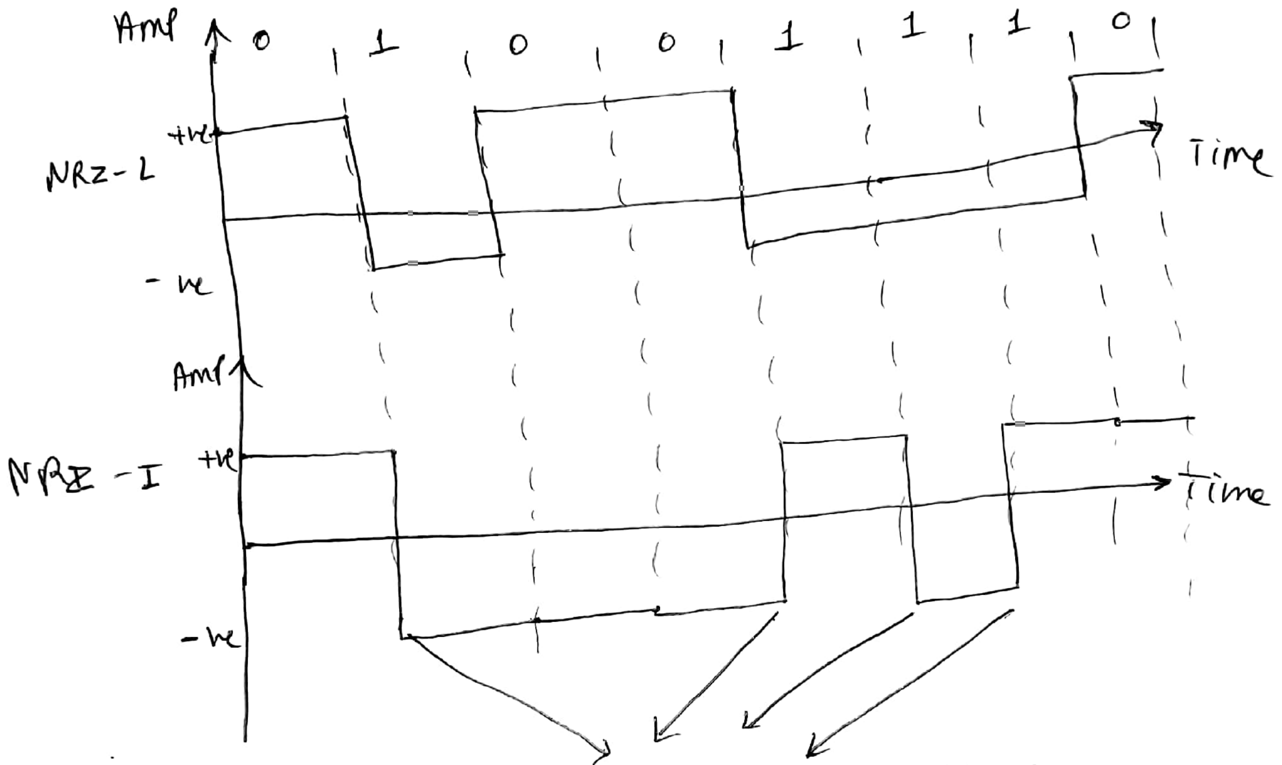
Synchronization problem if long stream of 1 or 0 is there

Non Return to zero Invert
NRZ-I → an inversion of the voltage level represents a 1 bit.

→ bit 0 is represented by no change.

→ NRZ-I is superior to NRZ-L due to the synchronization provided by the signal change each time a 1 bit is encountered.

→ Long stream of 0 may cause synchronization problem.



Transition as next bit is one.

Drawback of NRZ-I: → Consecutive seq. of 1's and 0's will create syn. problem.

Return to zero (RZ) : → To assure synchronization, there must be signal change at each bit.
One solⁿ is RZ which uses 3 levels, +ve, 0, -ve

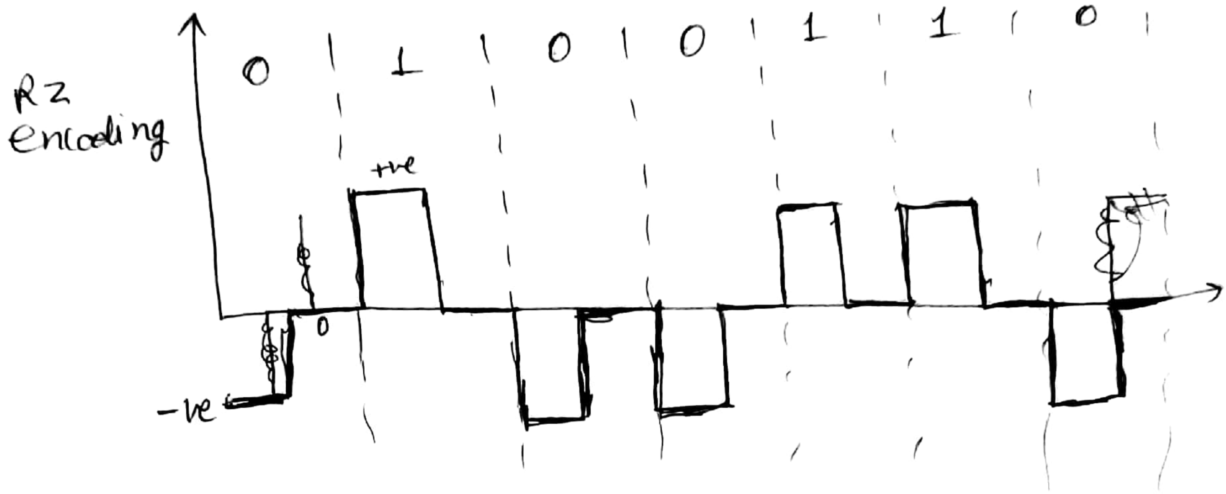
1 bit :→ +ve to zero

0 bit :→ -ve to zero

Disadvantage: → Requires two signal changes to encode one bit and therefore occupies more bandwidth

★ A good encoded digital signal must contain a provision for synch.

ex: →



transitions can be use for synch.

Biphase: → Best sol^n to the problem of synch.

→ Signal changes at the middle of the bit interval but does not return to zero. Instead it ~~continues~~ continues to opposite pole.

Two types:

1) Manchester: → Inversion ~~of~~ at the middle of each bit interval for both synch. and bit representation.

→ A -ve to +ve transition represents binary 1

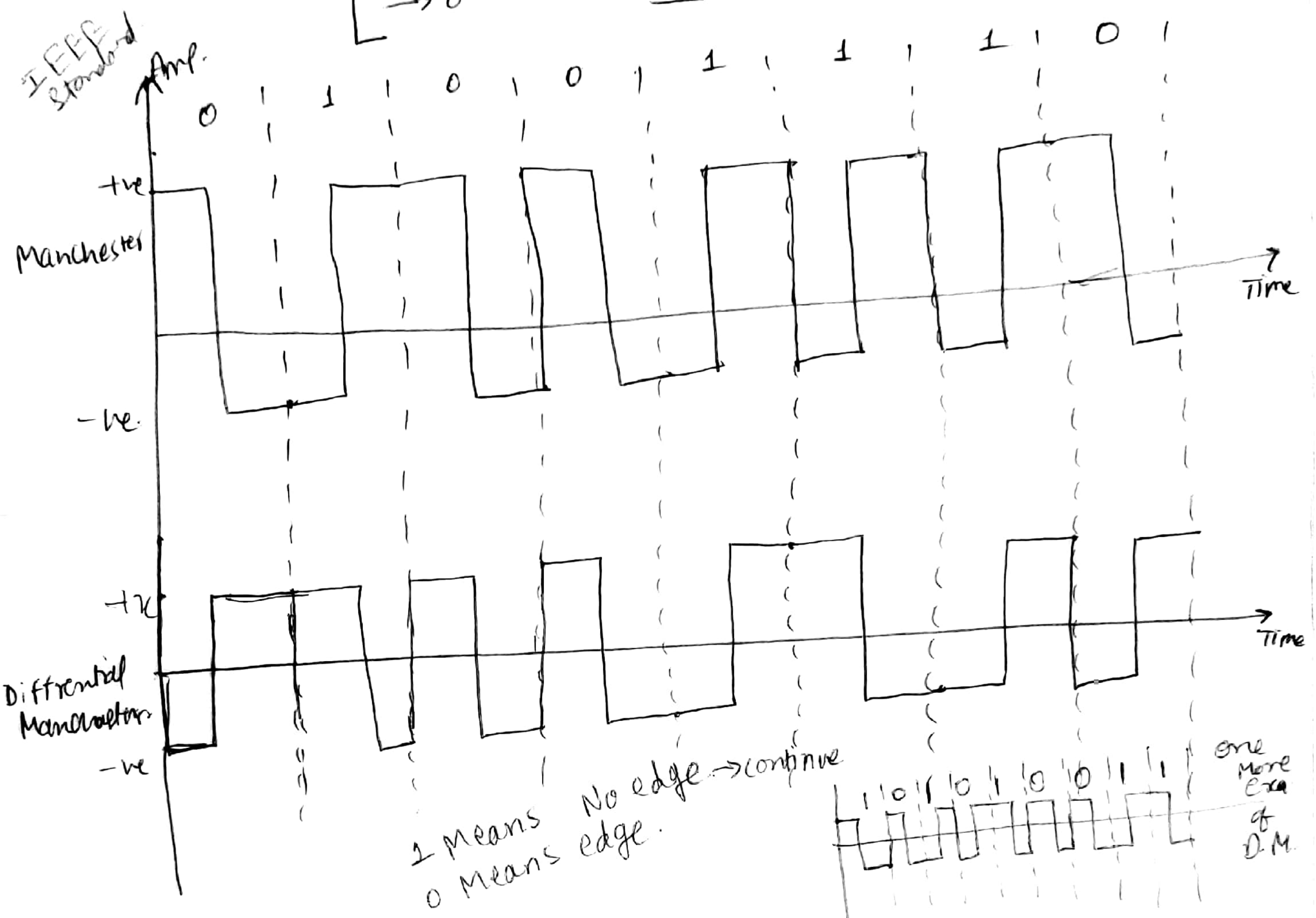
→ A +ve to -ve transition represents binary 0

2) Differential Manchester: → The inversion at the middle of the bit interval is used for synch., but the presence or absence of an additional transition at the beginning of the interval is used to identify the bit.

→ A transition means binary 0.

→ No transition means binary 1.

→ Requires two signal change for binar 0 but one for binary 1



Bipolar: → like RZ uses three voltage levels: → +ve, 0, -ve

Unlike RZ, however the zero level in bipolar ~~Endo~~ encoding is used to represent binary 0. The 1s are represented by alternating positive and -ve voltages.

→ If the first 1 bit is represented by positive ampl. the second will be represented by -ve ampl. the third by +ve ampl. and so on.

→ alternation occurs even when 1 bit are not consecutive.

1) Bipolar Alternate Mark Inversion (AMI): →

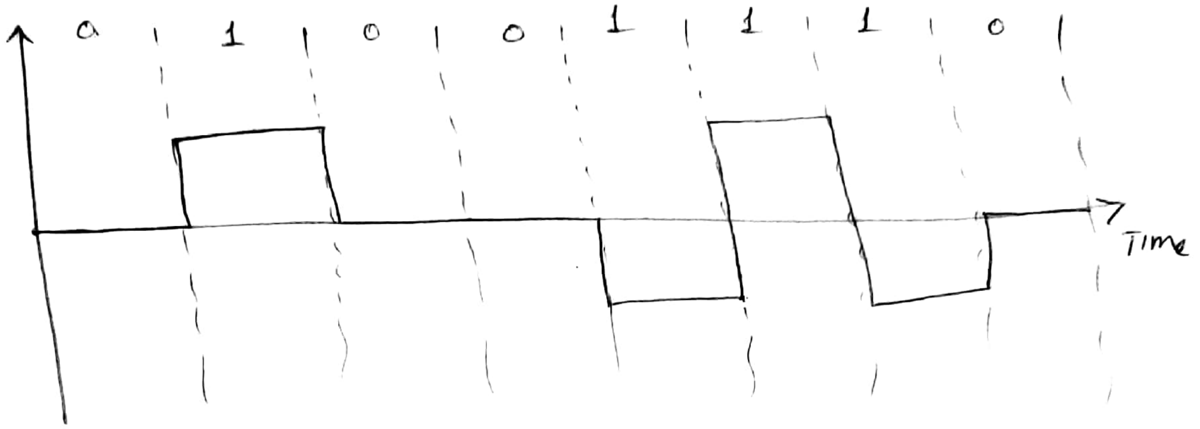
→ Simplest.

→ Word mark comes from telephony means 1.

→ so AMI means alternat 1 inversion.

→ zero voltage represents binary 0.

→ Binary 1 are represented by alternating +ve and -ve voltages.



Advantages: →

1) DC component is zero.
2) Long seq. of 1s stay synchronized.

disad: →

No mechanism to ensure the synch. of long string of 0s.

→ Variation of AMI (Bipolar & zero-substitutions) are B8ZS and HDB3.

2) B8ZS: → adopted in North America to provide synch. of long strings of 0s.

→ Work identical to AMI. as synch is often lost when 0's are there in AMI.

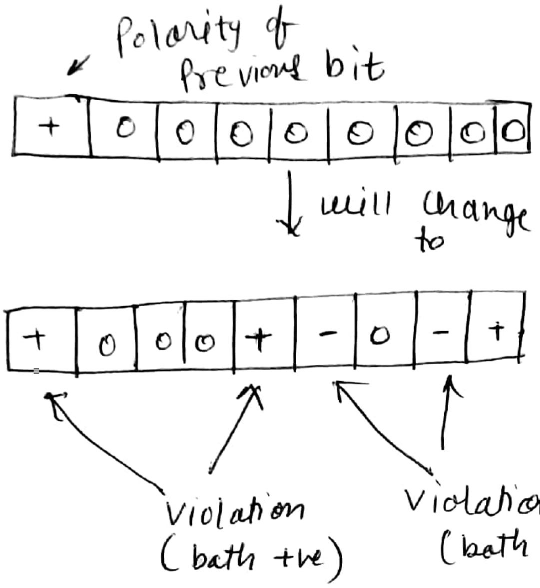
→ The difference between B8ZS and bipolar AMI occurs whenever eight or more consecutive 0s are encountered.

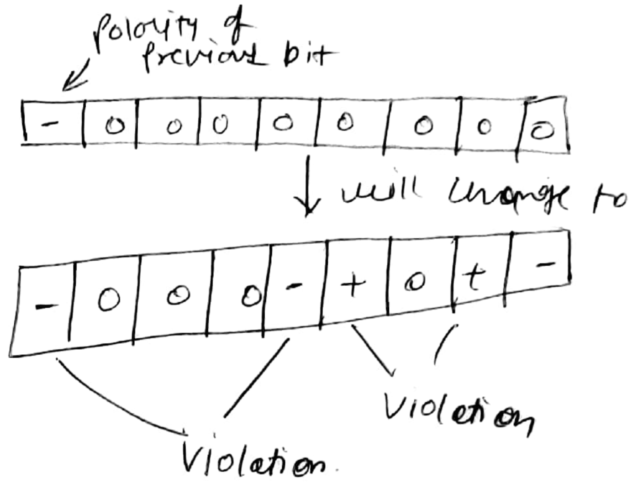
→ Solution is to force artificial signal changes, called violation, within the 0 string.

→ Anytime eight 0s occurs in succession, B8ZS introduce changes in the pattern based on the polarity of the previous 1 (1 occuring just before the 0s)

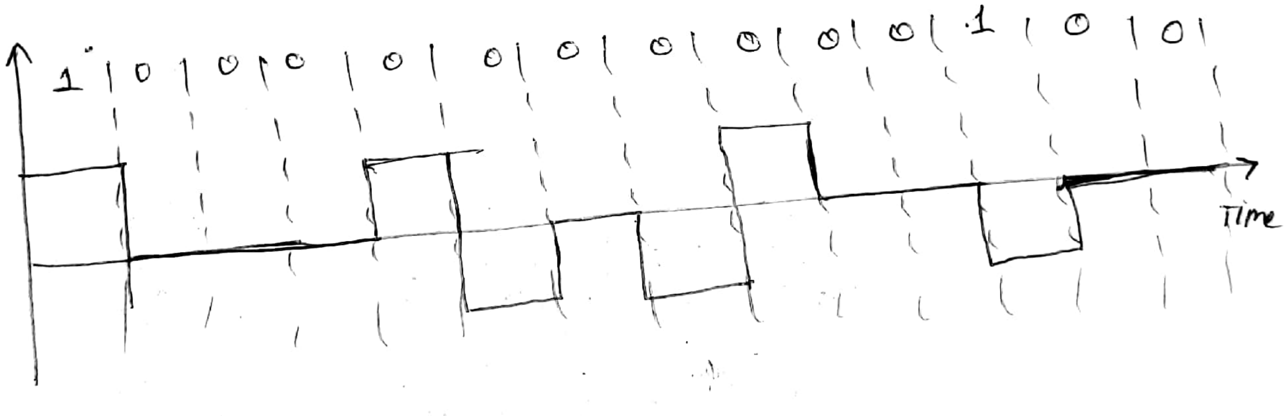
⇒ If the previous 1 bit was positive, the eight 0s will be encoded as zero, zero, zero, positive, negative, zero, negative, positive.

⇒ If the Polarity of the Previous 1 is -ve. the Pattern of violations is the same but with inverted Polarities
i.e zero, zero, zero, negative, Positive, zero, Positive, negative.



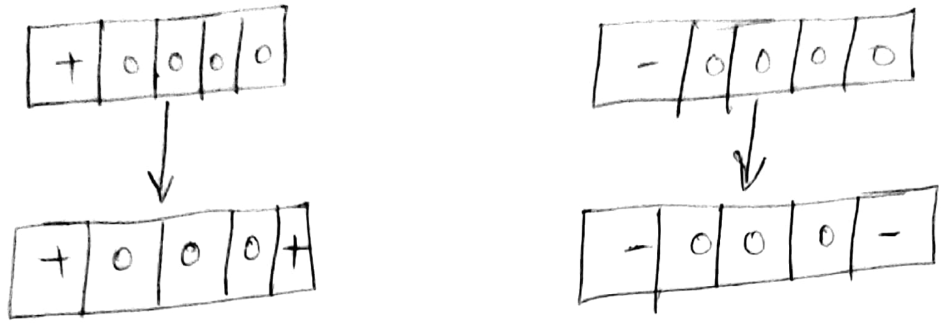


Ex: Encode the bit stream 10000000000100 using B8ZS, assume Polarity of first 1 is +ve.

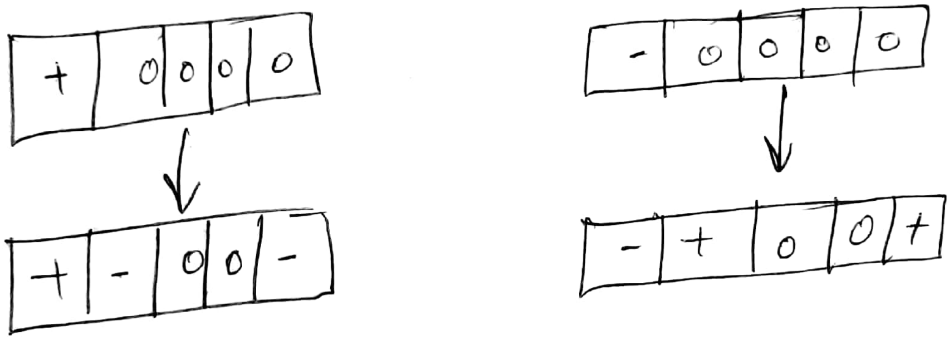


HDB3 (High density bipolar 3) :→ Problem of synch. in string of consecutive 0s is solved differently in Europe and Jafan. This is called HDB3.
Introduce changes int AMI pattern every time four consecutive 0s are encountered instead of 8.

HDB3 encoding



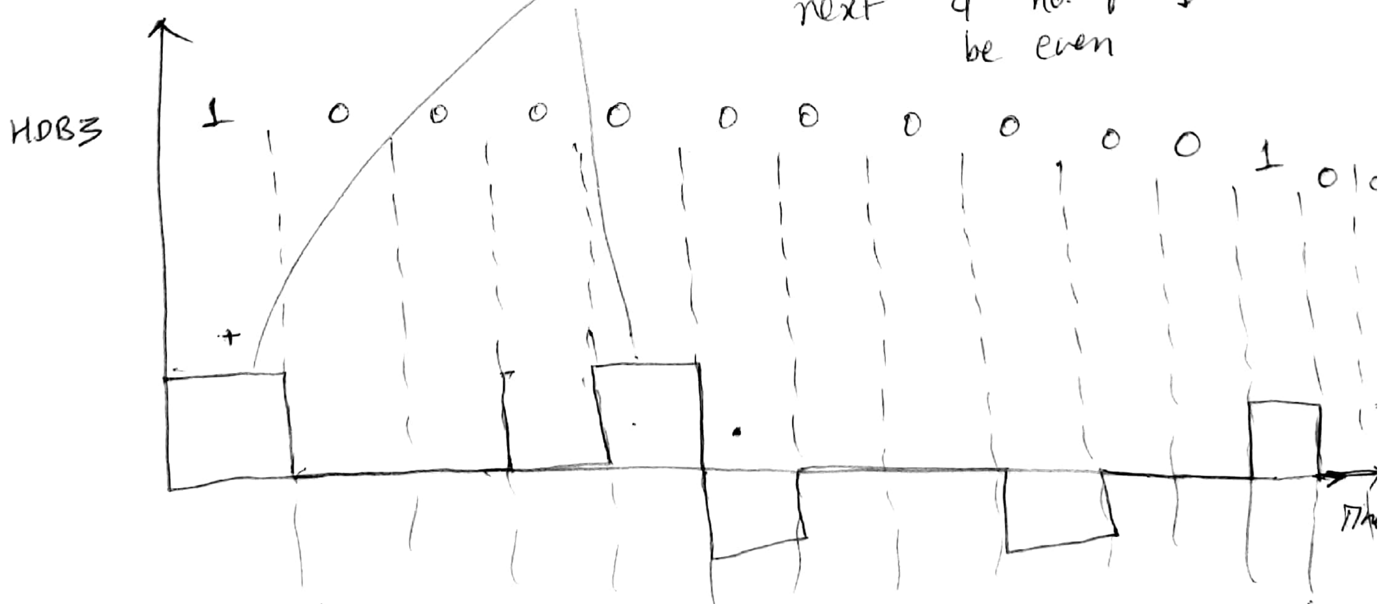
(a) If the no. of 1s since the last substitution is ~~even~~ odd.



(b) If the no. of 1s since the last substitution is even.

Ex: Using HDB3, encode the bit stream 10000000000100. Assume that the no. of 1s so far is odd and the first 1 is +ve.

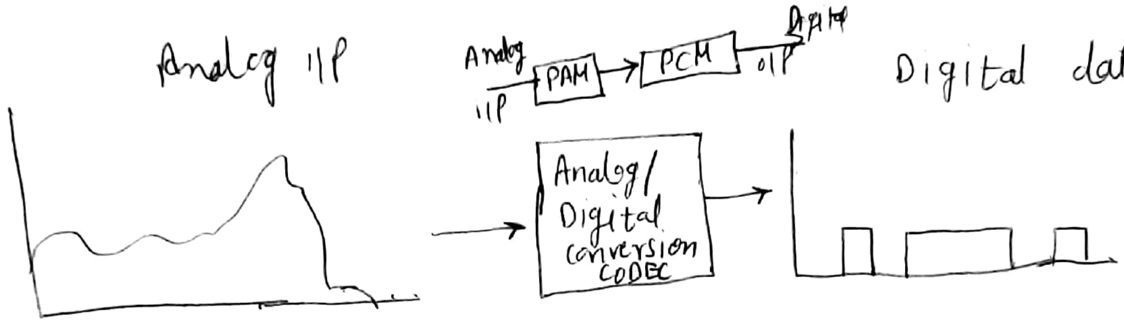
No of 1's here is 2 so far
next 4 no. of 1's will be even



42

~~44~~

Analog to Digital conversion: → To send human voice over a long distance we need to digitize it since digital signals are less prone to noise



Problem: → How to translate information from an infinite number of values to a discrete number of values without sacrificing sense or quality.

PULSE AMPLITUDE MODULATION (PAM): → First step in analog-to-digital conversion.

→ In this analog sigals are sampled and generates a series of pulses based on the results of the sampling. The term sampling means measuring the amplitude of signals at equal intervals.

→ Uses a technique called sample and hold: → At a given moment, the signal level is read, then held briefly.

→ Samples are done according to sampling Theorem / Nyquist criteria

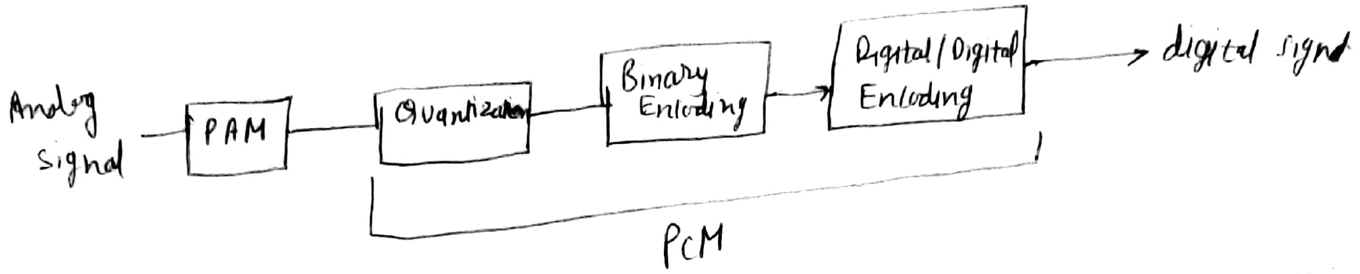
Pulse Code Modulation (PCM): → modifies the pulses created by PAM to create a completely digital signal.

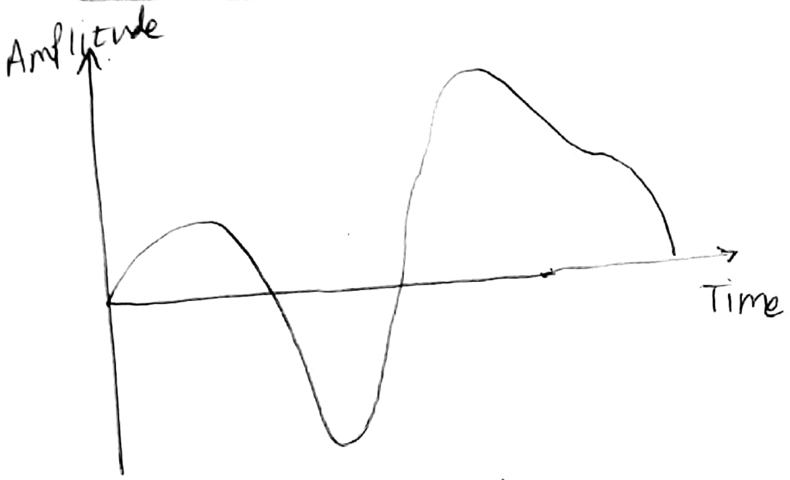
→ To do so PCM first quantizes the PAM pulses.

→ Quantization is a method of assigning integral values in a specific range to sampled instances.

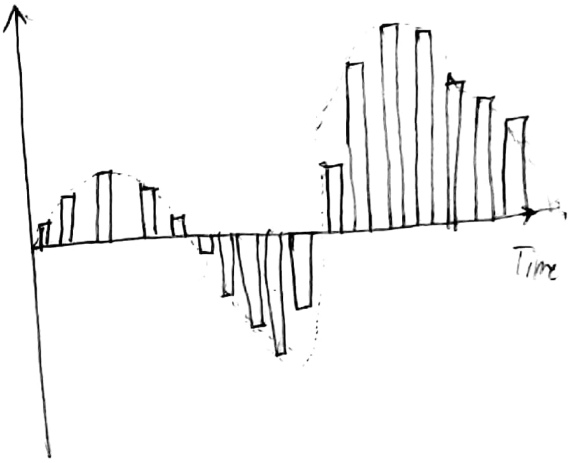
→ Each value is translated into its seven bit binary equivalent. Eight bit is sign bit.

→ Binary digits are then transformed into a digital signal using one of the digital-to-digital encoding techniques.





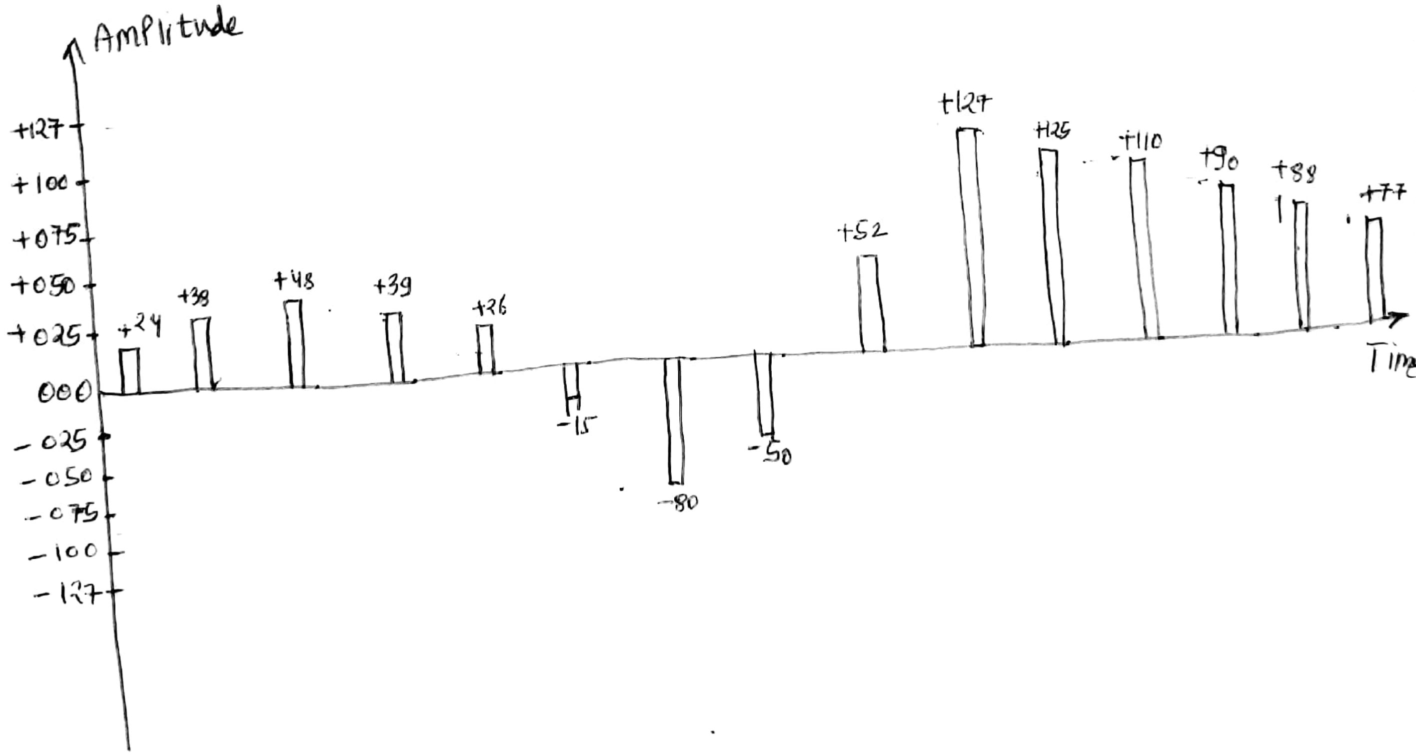
a) Analog signal



b) PAM signal (Sampled output)

Step-I : PAM

Quantized PAM signal

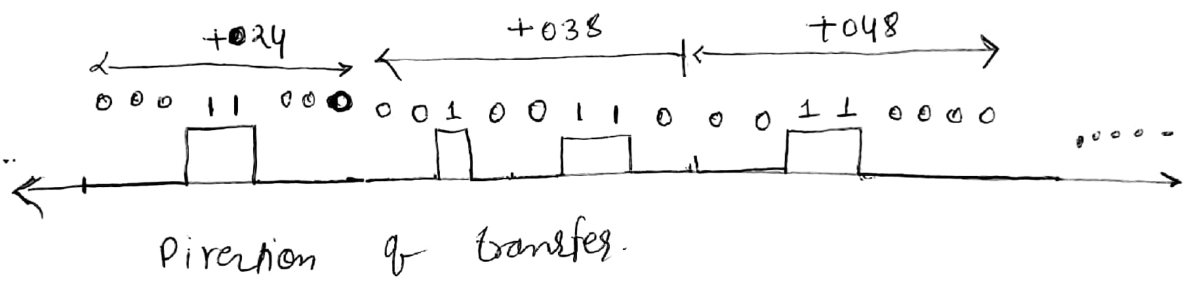


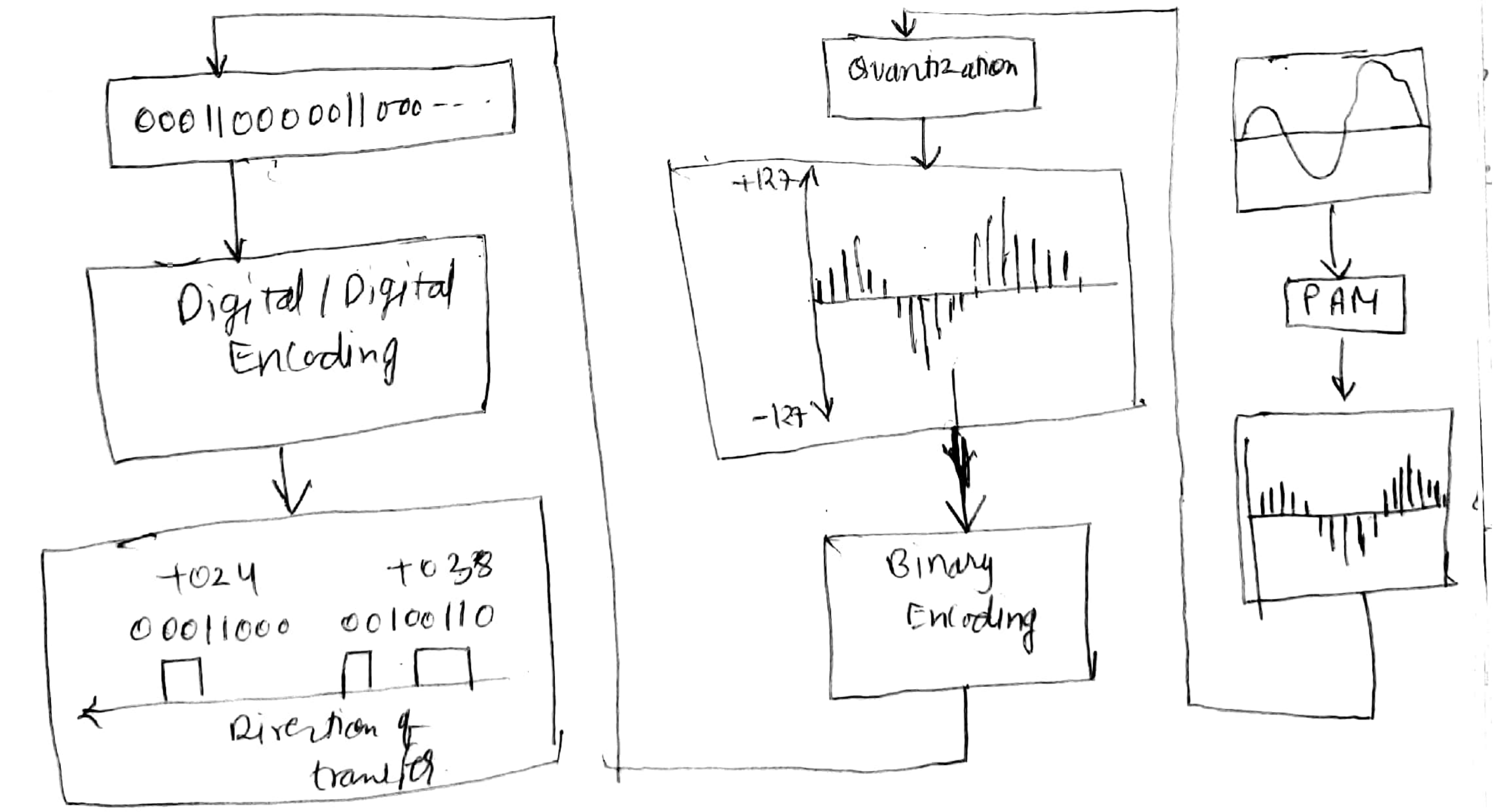
Quantizing using Sign and magnitude

+024	00011000	-015	10001111	+125	01111101
+038	00100110	-080	11010000	+110	01101110
+048	00110000	-050	10110010	+090	01011010
+039	00100111	+052	00110110	+088	01011000
+026	00011010	+127	01111111	+077	01001101

Sign bit
+ is 0 - is 1

PCM.

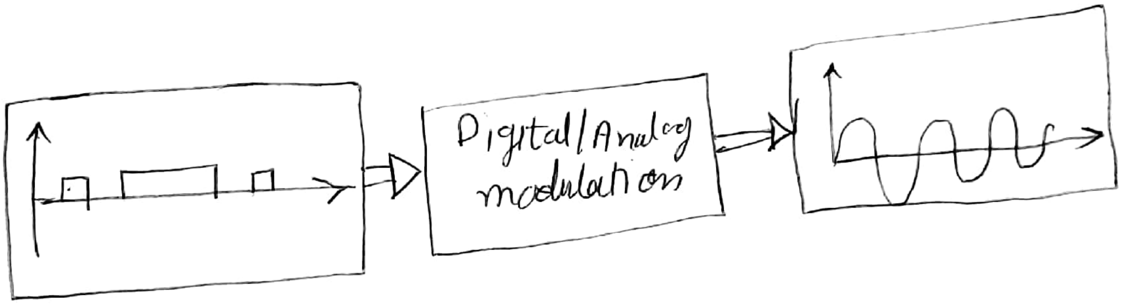




Digital - to - Analog conversion: → ASK, PSK, FSK and QAM = ASK + PSK.

Modulation: → Changing the characteristic (Amplitude, freq, Phase) of high freq signal called carrier with respect to the message signal.

Demodulation: → Reverse process of modulation in which message is extracted from carrier signal at the receiving end.



Bit rate: → No of bits per second.

Baud rate: → No. of signal units per second. Baud rate is less than or equal to the bit rate

$$\text{Baud rate} = \frac{\text{Bit rate}}{\text{No. of bits per signal Element}}$$

Ex: → An analog signal carries four bits in each signal element. If 1000 signal elements are sent per second, find baud rate and bit rate.

Sol^n: Baud rate = No of signal elements = 1000 baud per second

Bit rate = Baud rate × No. of bits per signal element

Bit rate = 1000 × 4 bps = 4000 bps

Ex. The bit rate of a signal is 3000. If each signal element carries six bits, what is the baud rate?

Solⁿ → $\frac{3000}{6} = 500$ baud rate

ASK (Amplitude shift keying) :→ The strength of the carrier signal is varied to represent binary 1 or 0. Both freq and phase remains constant while amplitude change.

Also called OOK on-off keying

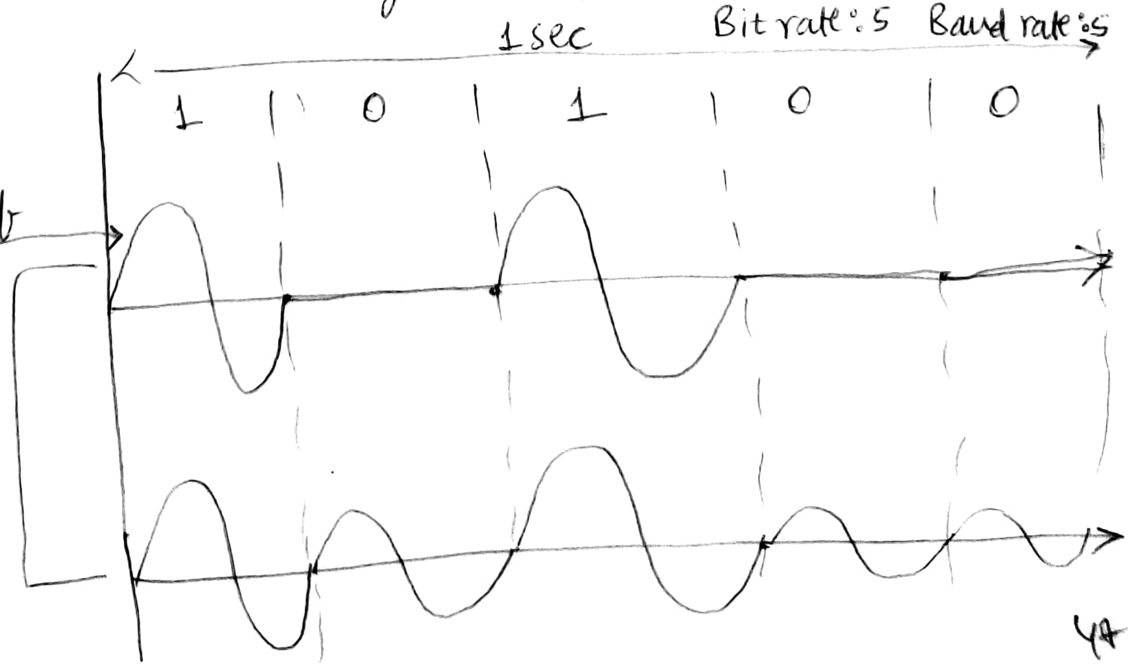
1 → sine wave (V, -V) and 0 → —— 0V

Disadvantages of ASK.

ASK transmission is highly susceptible to noise interference. The noise voltages combines with the signal to change the amplitude. 0 can be changed to 1 and 1 can be changed to 0.

→ Noise usually affects the amplitude therefore ASK is the modulationg method most affected by noise.

Waveforms of ASK

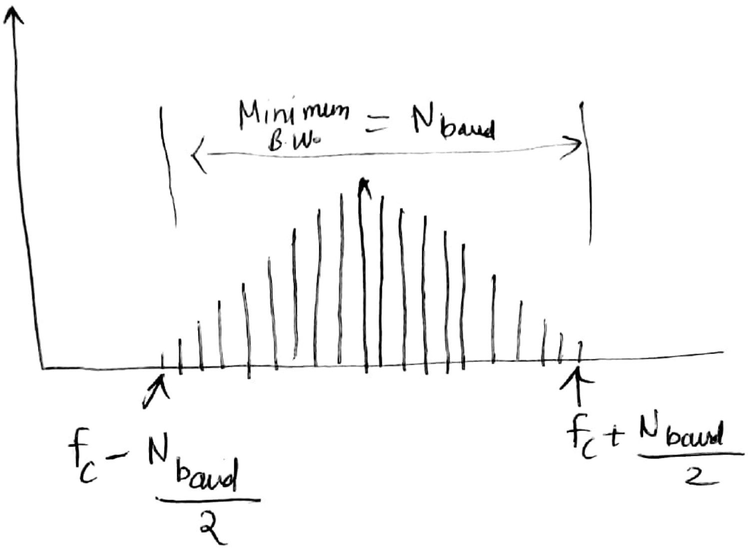


Such kind of ASK are OOK.

Both are representing ASK of 10100

47

Bandwidth of ASK :→ when we decompose an ASK modulated signal, we get a spectrum of many simple freq.



$$\boxed{\text{B.W. requirement for A.S.K.} = (1+d)\, N_{baud}}$$

Here . B.W. = Bandwidth
N_{baud} = baud rate
d = factor related to the condition of line (with min. d = 0)

$$\boxed{B.W. = N_{baud}} \rightarrow d = 0$$

ex:→ Find the min b/w for ASK signal transmitting 2000 bps. The transmission mode is half duplex.

Sol^n:→ In ASK Baud rate and bit rate are same as B.W. ∴ Baud rate = 2000, and Hence

$\boxed{B.W = 2000 \text{ HZ}}$

Ex:→ Given B.W. = 5000 Hz for an ASK. what is bit rate and baud rate.

Sol^n:→ In ASK B.W. = bit rate = Baud rate = 5000 bps

Ex. Given a B.W. of 10,000 Hz (1000 to 11,000 Hz). Draw the full duplex ASK dig of the system. Find the carriers and B.W in each direction. Assume there is no gap between the bands in two directions.

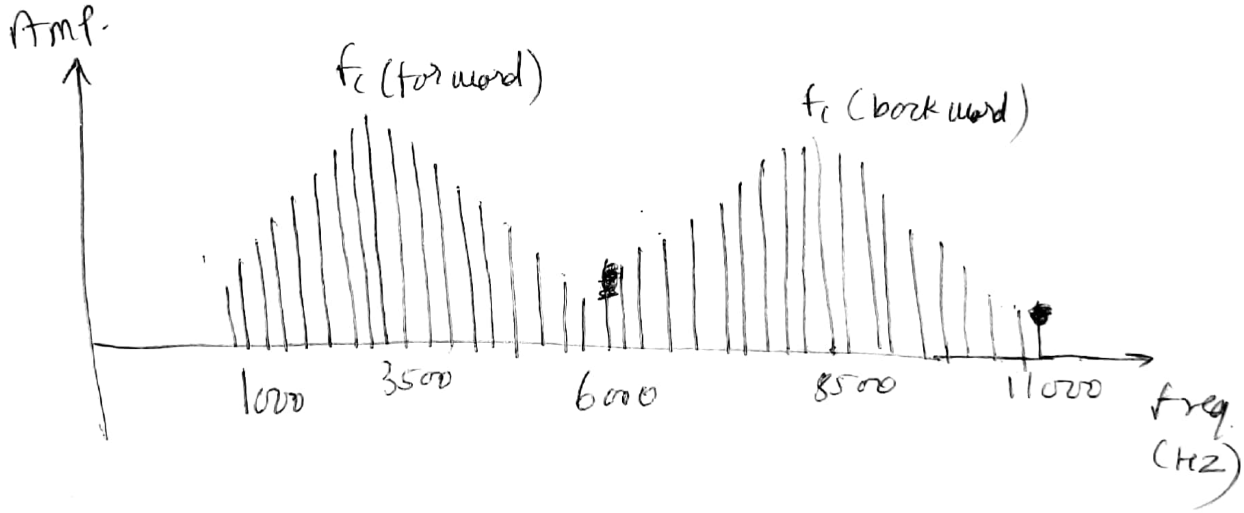
Solⁿ: → For full duplex ASK, the B.W for each direction is

$$B.W. = 10000/2 = 5000\,Hz$$

The carrier freq. can be chosen at the middle of each band.

$$f_c\,(\text{forward}) = 1000 + 5000/2 = 3500\,Hz$$

$$f_c\,(\text{backward}) = 11000 - 5000/2 = 8500\,Hz$$



~~Frequency shift keying (FSK) :→ frequency of the carrier signal is varied to represent binary 1 or 0. Both Amp and freq remain constant.~~

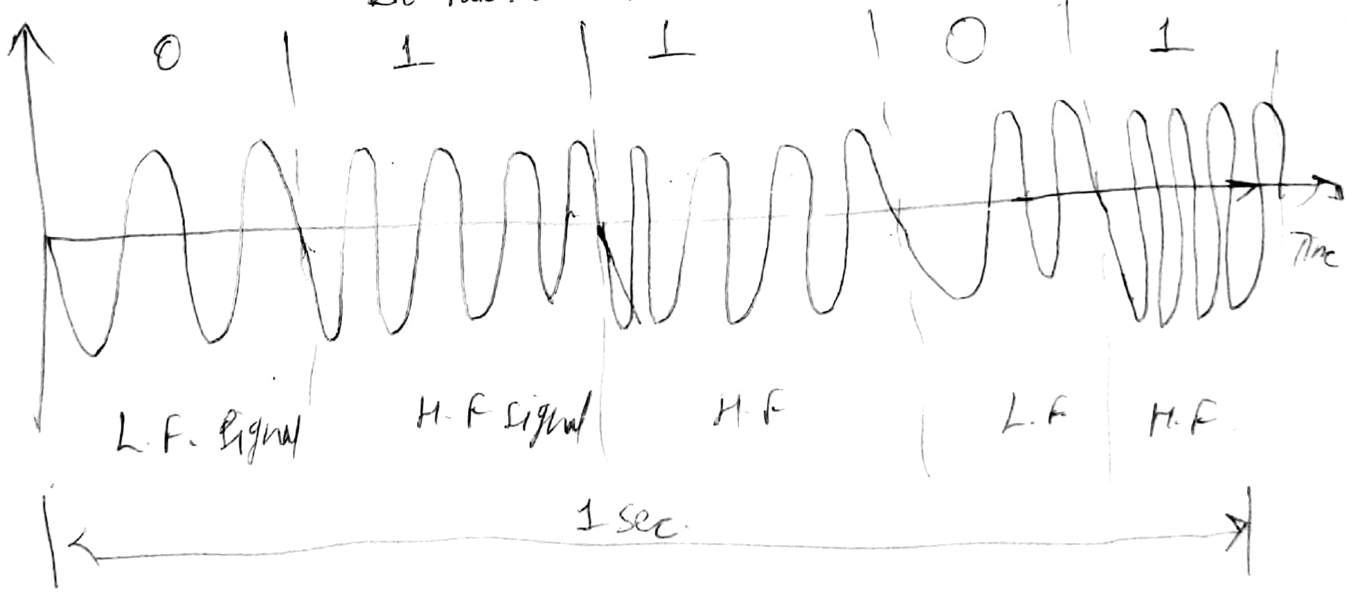
Frequency shift keying (FSK):→ the freq of the carrier signal is varied to represent binary 1 or 0. freq of the signal during each bit duration is constant and its value depends upon the bit 0 or 1.

→ Both Ampl. and Phase remain constant.

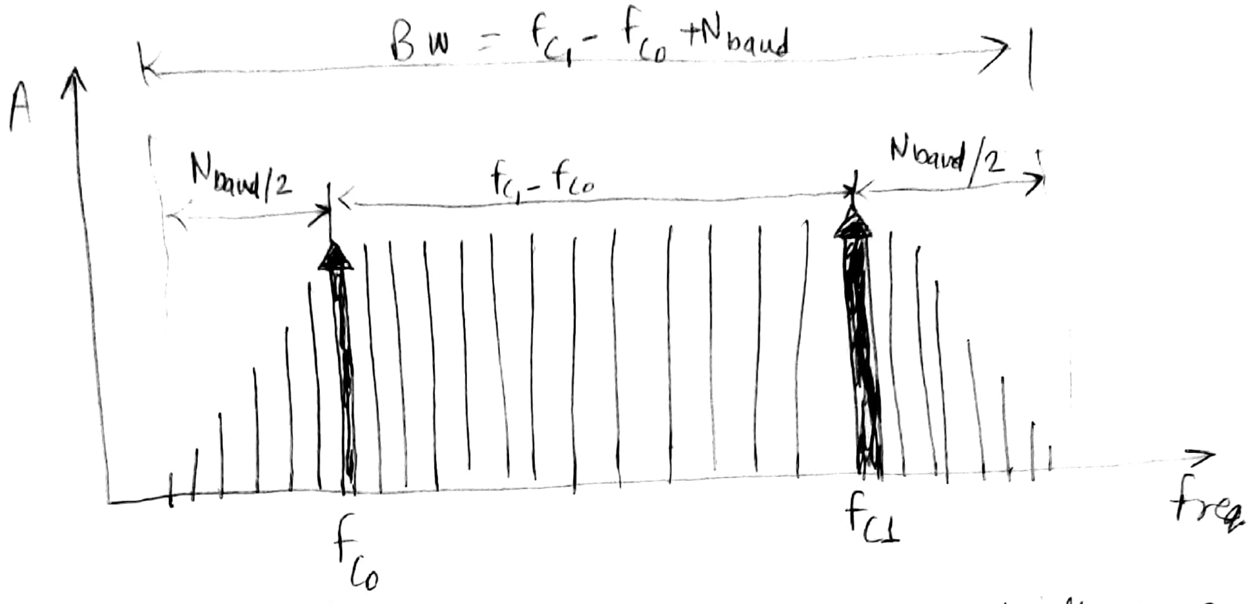
FSK avoids most of the noise problem of ASK. of receiver is looking for freq. Changes it may avoids the ampl.

→ Limiting factors of FSK are the Physical capabilities of the carrier.

Bit rate: 5 Baud rate: 5



Bandwidth of FSK:→ $B.W. = (f_{c_1} - f_{c_0}) + N_{baud}$



Relationship between baud rate and bandwidth in FSK

Ex:→ Find the min. B.W. for an FSK signal transmitting at 2000 bps. transmission is in Half duplex mode and carriers must be separated by 3000 Hz.

$$B.W. = \cancel{2000}\ \text{Baud rate} + (f_{c_1} - f_{c_0}) \quad \rightarrow 3000\,Hz$$

$$= \text{Bit rate} + (f_{c_1} - f_{c_0})$$

$$= 2000 + 3000 = 5000\,Hz.$$

Ex:→ Find the Max ~~B.W.~~ bit rate for an FSK signal if the B.W. of the medium is 12000 Hz. and difference between the two carriers must be atleast 2000 Hz. Transmission in full duplex

Sol^{n}:→ Full duplex B.W. $= \frac{12000}{2} = 6000\,Hz$

$$\text{Baud rate} = BW - (f_{c_1} - f_{c_0}) =$$

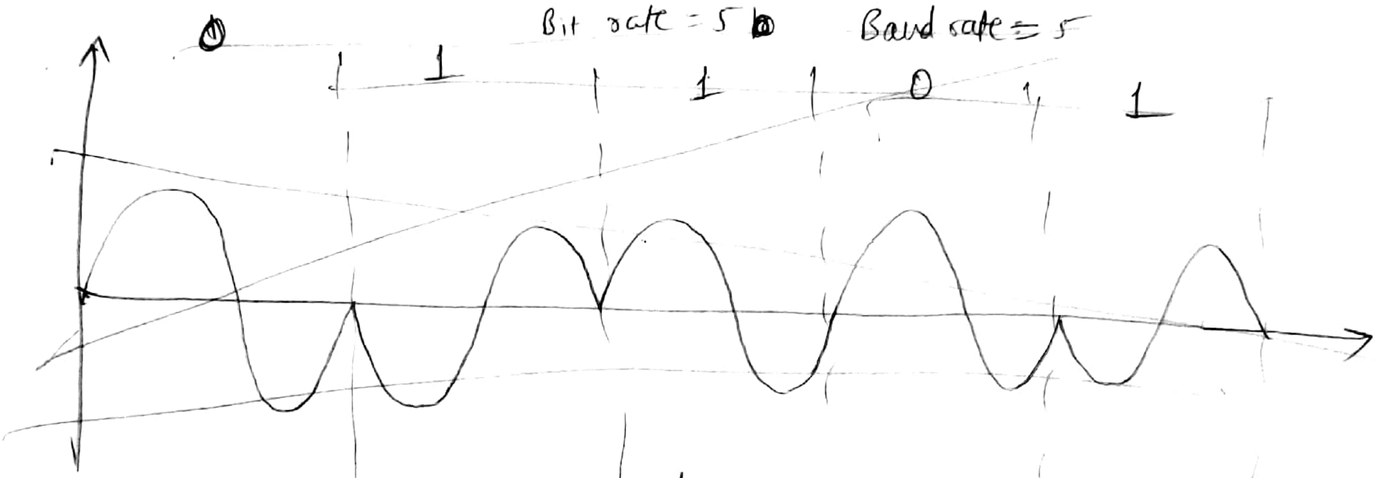
$$= 6000 - 4000$$

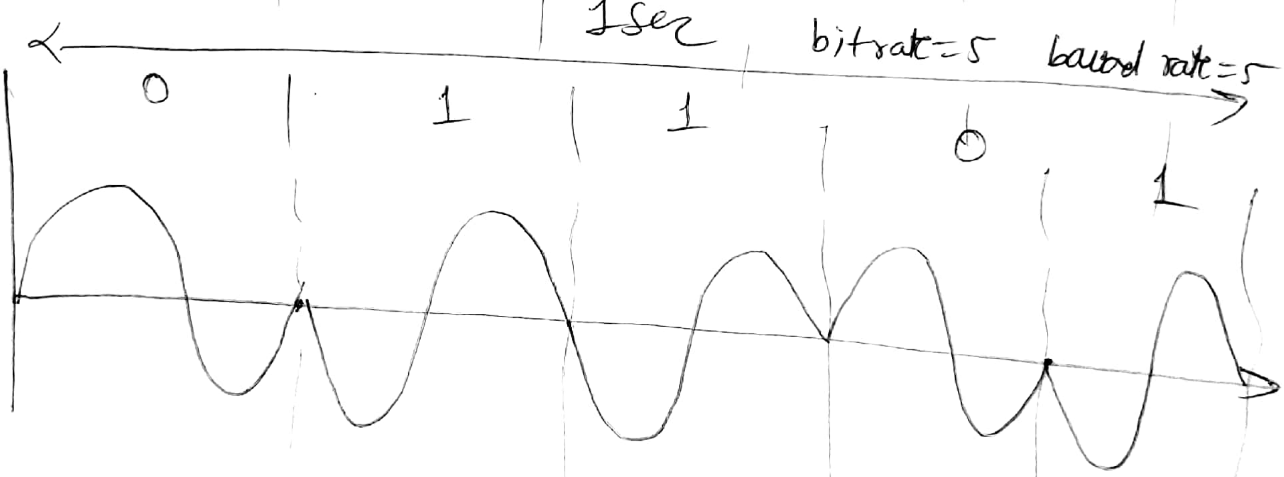
$$= 4000\ \text{bps}.$$

Scanned with CamScanner

Phase shift keying (PSK) : → the phase of the carrier varied to represent binary 1 or 0. Both peak ~~and~~ ampl and freq remain constant as the phase changes.

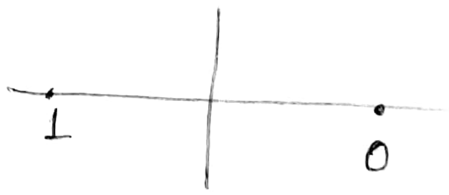
0° phase → 0 180° phase → 1





PSK wave form

Bit	Phase
0	0°
1	180°

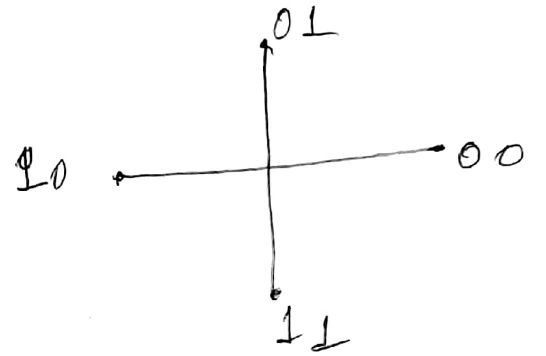


Constellation

PSK is not susceptible to the noise degradation that affects ASK, nor to the bandwidth limitations of FSK.

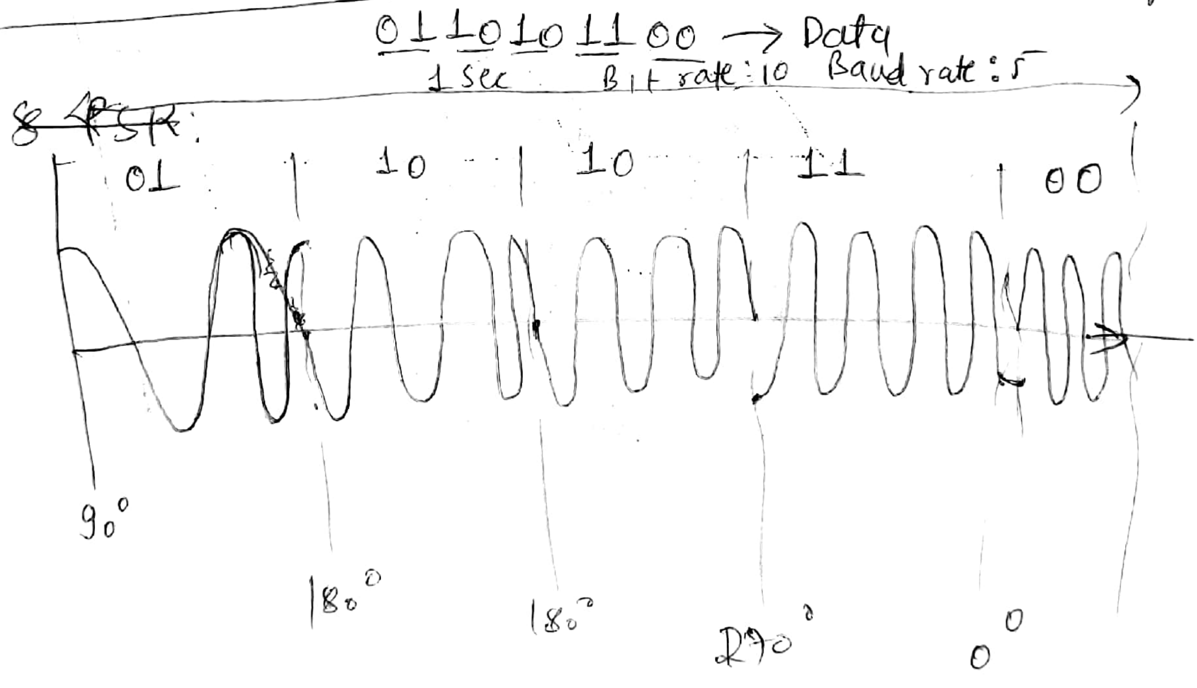
4-PSK: → If we change the phase of the carrier according to the ~~to~~ two bit combination of bit we can acheive 4-psk.

Dibit	Phase
00	0
01	90
10	180
11	270



Constilation dig.

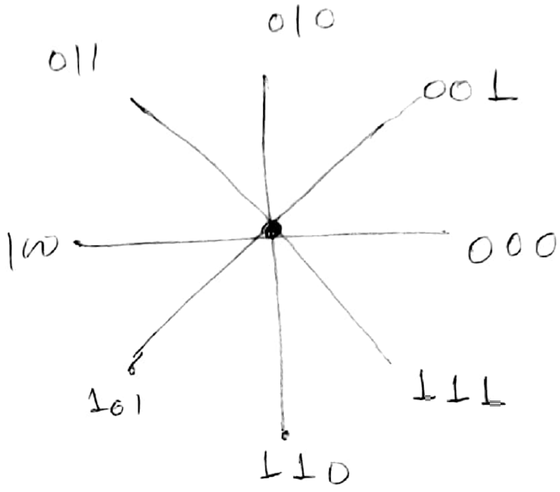
$$\boxed{Angle = \frac{360}{2^n} \quad n = \text{bit combination}}$$



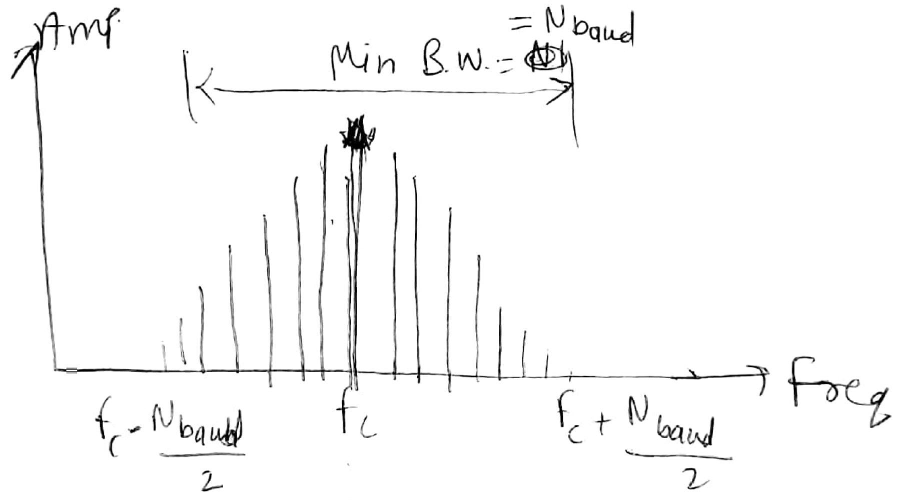
8 PSK: → $Angle = \frac{360}{2^3} = \frac{360}{8} = 45, 90, 135, ---$

& PSK. →

Tribit	Phase
000	0
001	45
010	90
011	135
100	180
101	225
110	270
111	315



Constellation diagram.



BW. of P.S.K.

Ex: Find the B.W. of 4-PSK signal transmitting at 2000bps. Transmission is in half-duplex mode.

Solⁿ: → 4 PSK baud rate is half of the bit rate.

baud rate = $\frac{2000}{2}$ = 1000 baud.

PSK signal B.W = Baud rate
= 1000Hz

Ex: Given B.W. = 5000 Hz for an 8-PSK signal, what are the baud rate and bit rate?

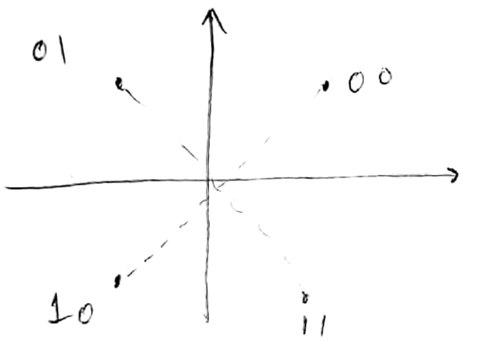
Sol^n: → For PSK baud rate = Bit rate = 5000.

But in 8-PSK bit rate = 3 × 5000 = 15000 bPS

$$\boxed{\text{Bit rate} = n \times \text{baud rate}}$$

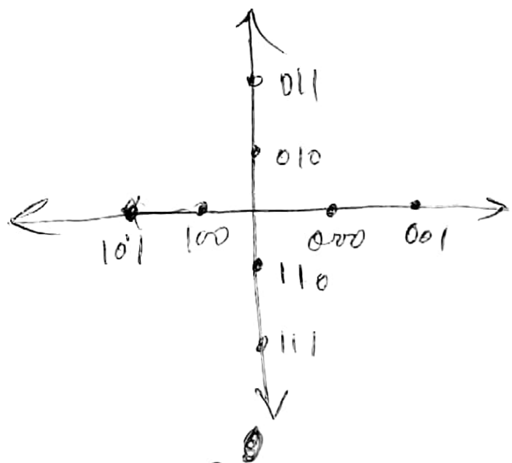
Quadrature-Amplitude Modulation (QAM) :→ Combining ASK and PSK in such a way that we have maximum contrast between bit, dibit, tribit, quadbit and so on.

4 QAM Constellation

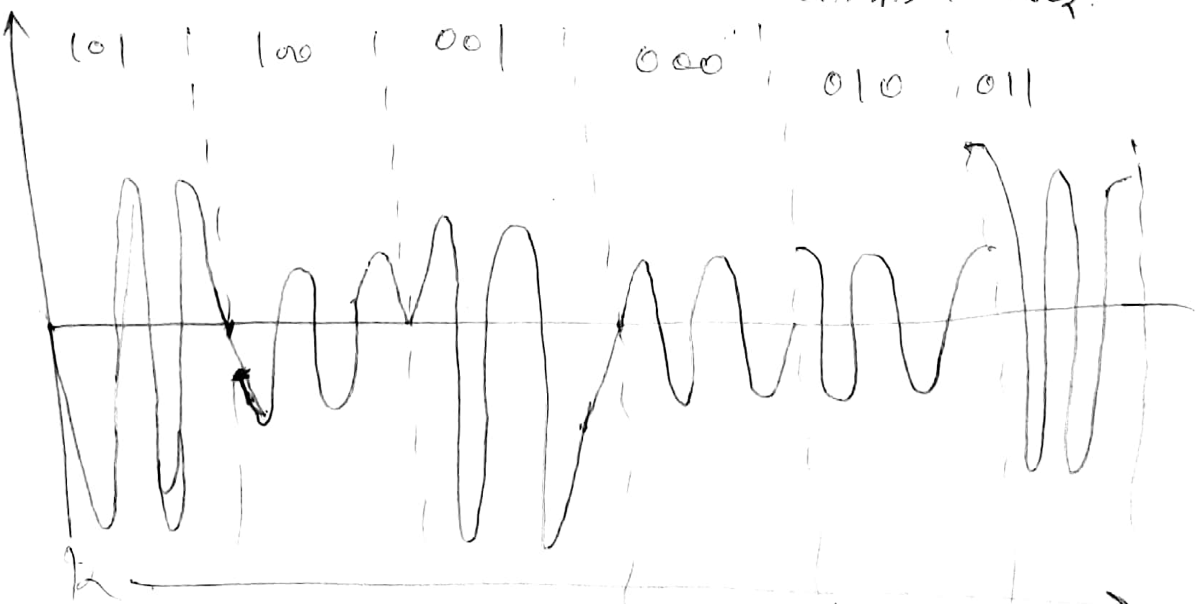


1 ampl, 4 phase

8QAM



2 Ampl, 4 Phases.



1 sec.

Baud rate. 6
bit rate. 18

Bandwidth of QAM: → Same as that required for ASK and PSK transmission.

→ QAM has the same advantages as PSK over ASK

16 QAM: → we can have

→ 3 amplitude, 12 phases

→ 4 Amplitude, 8 phases

→ 2 Amplitude, 8 phases

Ex: → A constellation dig consists of eight equally spaced points on a circle. If the bit rate is 4800 bps. what is baud rate.

Solⁿ: → 8 - PSK, $2^3 = 8$ Baud $= \frac{4800}{3} = 1600$ baud

Ex: → Compute the bit rate for 1000 baud 16 QAM

Solⁿ → 16 QAM Means ∵ 4 bit per signal $(2^4 = 16)$

$1000 \times 4 = 4000$ bps

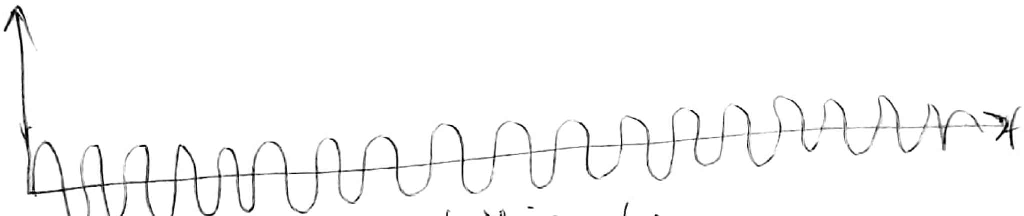
Ex: → Compute the baud rate for a 72,000 bps 64 QAM signal.

Solⁿ: → 64 QAM means there are six bit per signal element since $2^6 = 64$

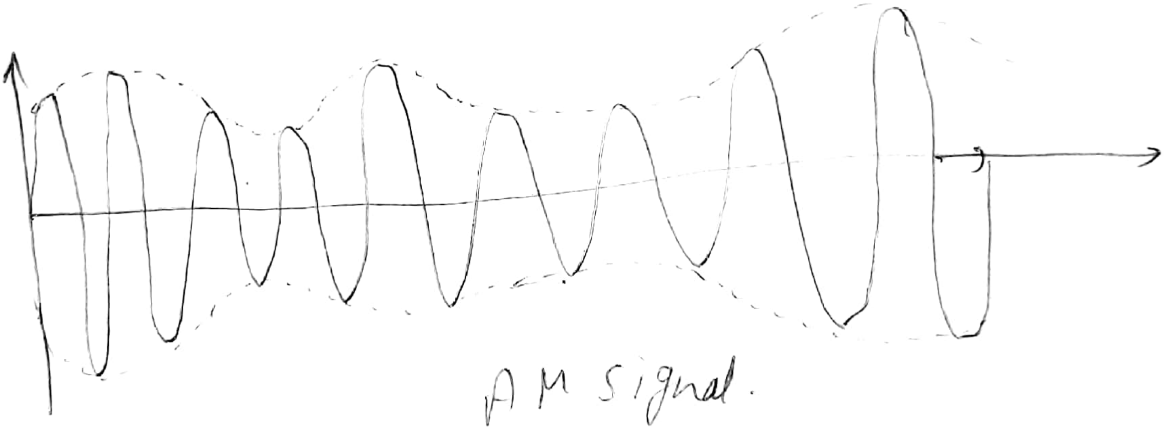
∴ $\frac{72000}{6} = 12,000$ baud.

Analog to Analog conversion → Three types AM, FM, PM

1) Amplitude modulation (A.M) :→ Amplitude of carrier signal is varied with changing amp of message signal.



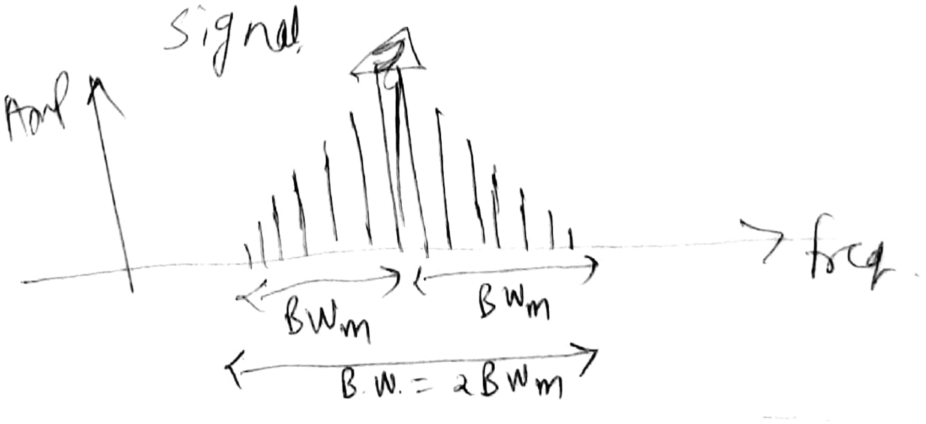
carrier freq



AM signal.

AM B.W. :→ $\boxed{B.W. = 2 \times B.W_m}$

B.W of AM is 2 twice of B.W. of message signal

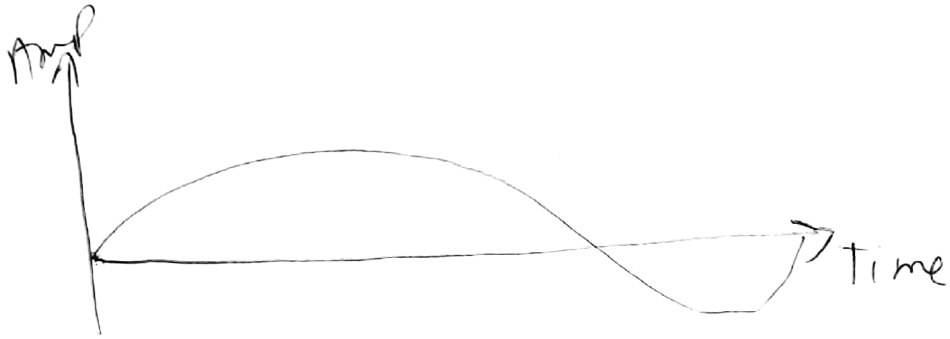


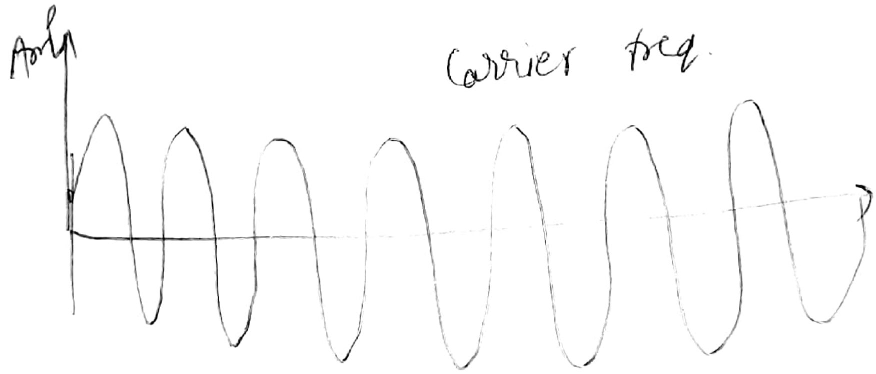
Ex: we have an audio signal with B.W. = 4KHZ, what is the B.W. needed if we modulate the signal using AM.

Soln: → B.W. = 2 × 4KHZ = 8KHZ.

FM

Frequency Modulation (FM): → The freq of carrier signal is change with respect to the changing voltage level (amplitude) of message signal. Ampl. and Phase remains constant.

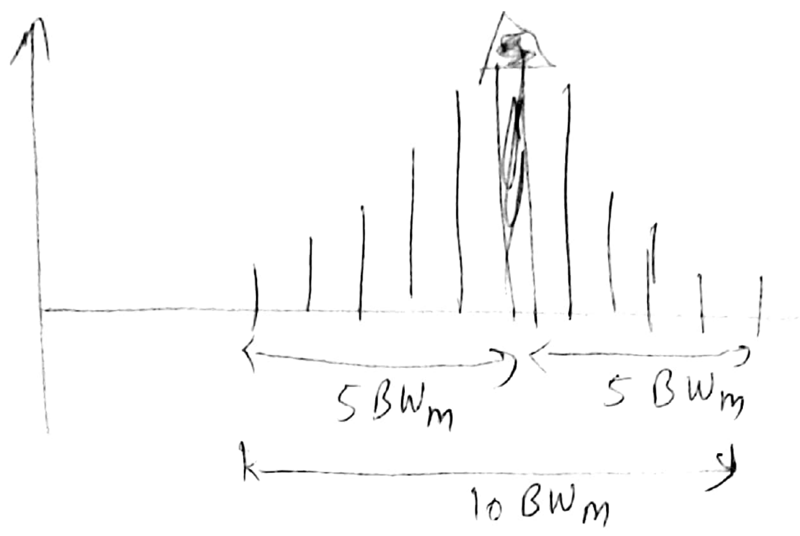




Carrier freq.

FM signal

$\boxed{\text{F.M. Bandwidth} = 10 \times BW_m}$ approximately.

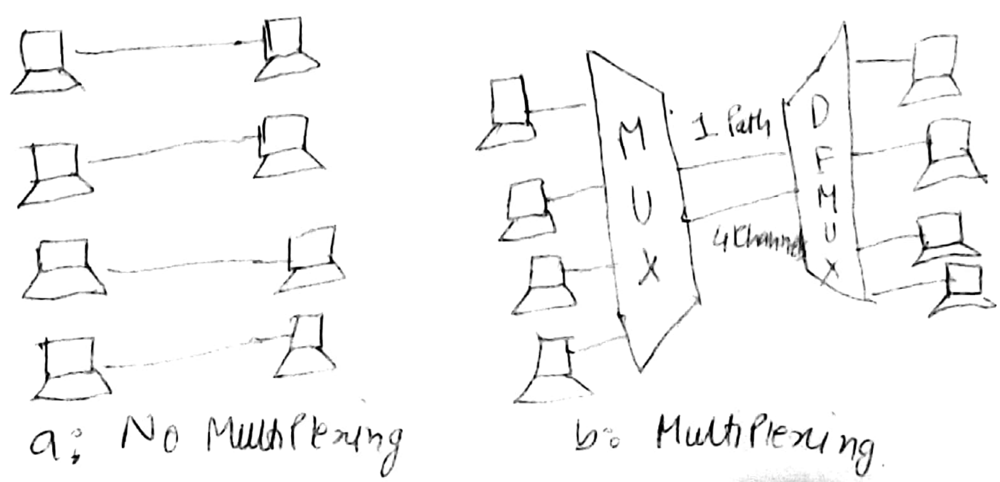


PM (Phase modulation) :→ In PM the Phase of the Carrier signal is changed to follow the Changing voltage level of the message signal. Ampl. and freq. remains Constant.

Multiplexing: → Many to one → is the set of techniques that allows the simultaneous transmission of multiple signals across a single data link.

→ As data- and telecommunications usage increases, so does traffic we can accommodate this increase by continuing to add individual lines each time a new channel is needed or we can install higher capacity links and use each to ~~each~~ carry multiple signal

~~Many to~~



a: No Multiplexing   b: Multiplexing

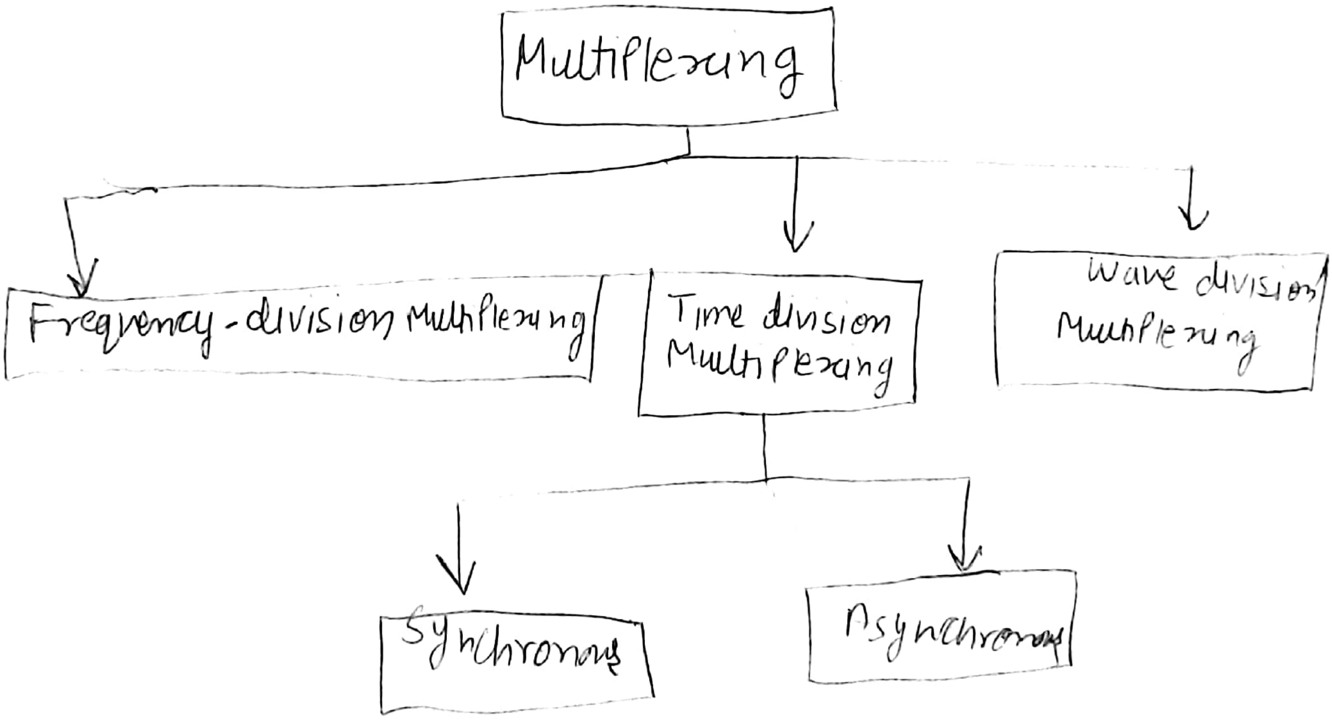
99

Many to one / One to many :→ In a multiplexed system, n devices share the capacity of one link. In fig b. the basic format of multiplexed system is shown in which four devices on left direct their transmission stream to a multiplexer (MUX), which combines them into a single stream (many to one). At the receiving end, that stream is fed into demultiplexer (DEMUX), which separates the stream back into its component transmission (one to many) and directs them to their intended receiving devices.

→ Path refers to the physical link.

→ Channel refers to a portion of a path that carries a transmission between a given pair of devices.

→ One path can have many (n) channels.



Frequency Division Multiplexing (FDM)

→ Analog technique.

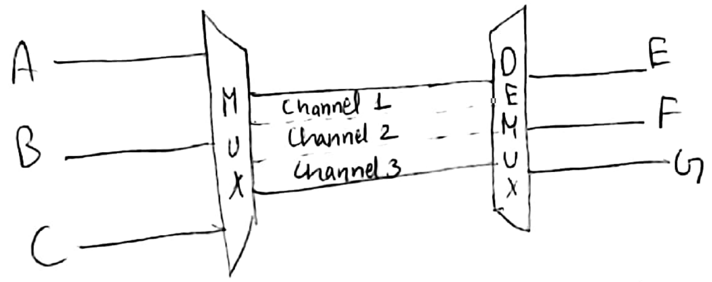
→ Can be applied when B.W. of link is greater than the combined B.W. of the signal is to be transmitted.

→ In FDM, signals generated by each sending device modulate different carrier frequencies. These modulated signals are then combined into a single composite signal that can be transported by link.

→ Carrier frequencies are separated by enough bandwidth to accommodate the modulated signal. These B.W. ranges are the channels through which the various signal travel.

→ Channel must be separated by strips of unused b.w. (Guard bands) to prevent signals from overlapping.

→ Analogy :→ three narrow streets merge to form a three-lane highway.

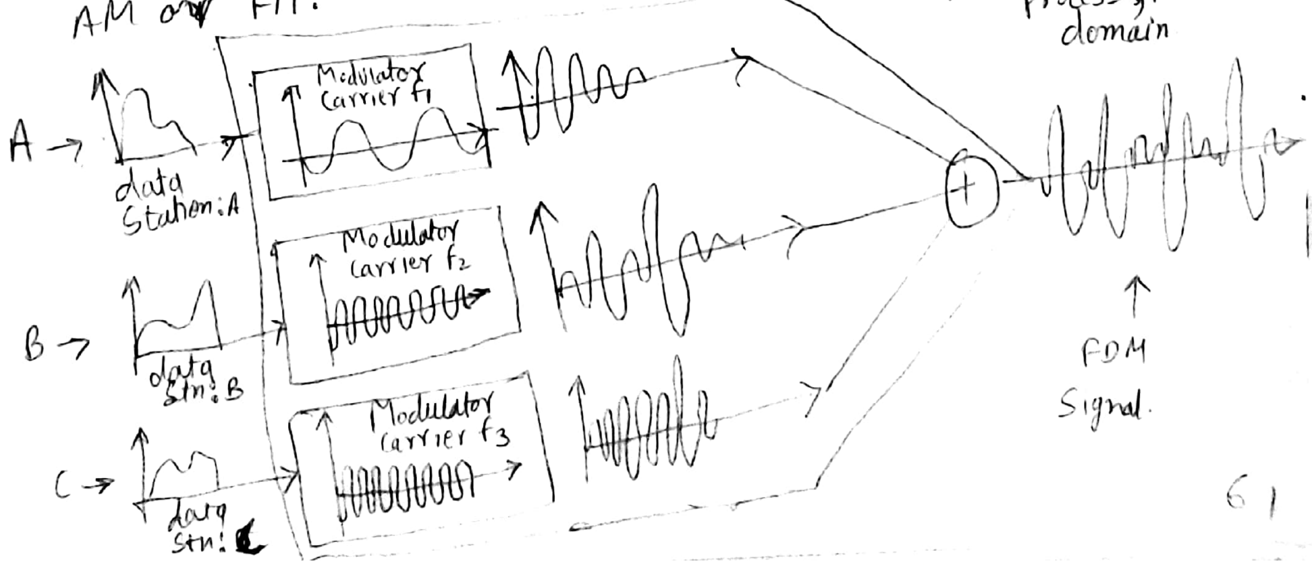


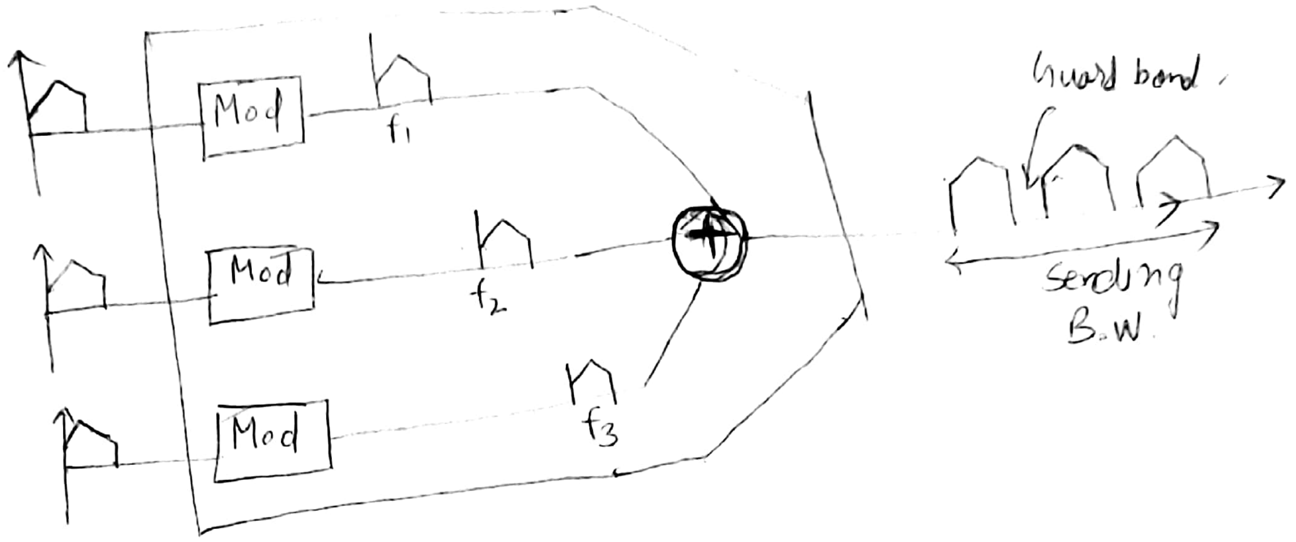
→ Channel Division is achieved by freq rather than space.

<u>The FDM Process</u> :→ In FDM, signals are modulated onto separate carrier frequencies (f_1, f_2 and f_3) using either AM or FM.

Multiplexing :→

Fig: FDM Multiplexing Process, T-DM time domain





Demultiplexing: Demultiplexer uses series of filters to decompose the multiplexed signal into its constituent component signals. Individual signals are then passed to a demodulator that separates them from their carriers and passes them to the waiting receivers.

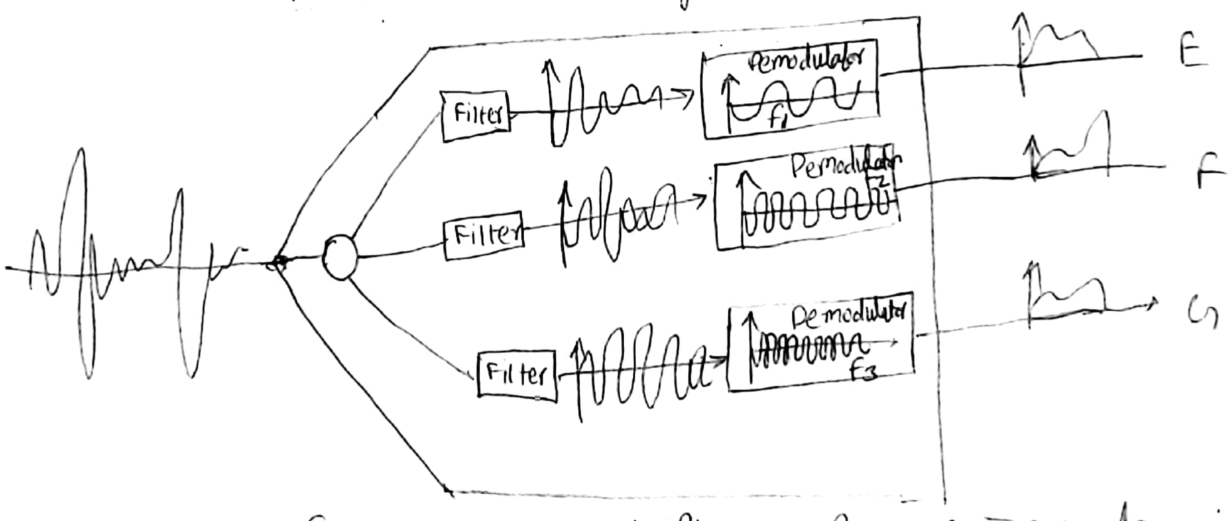
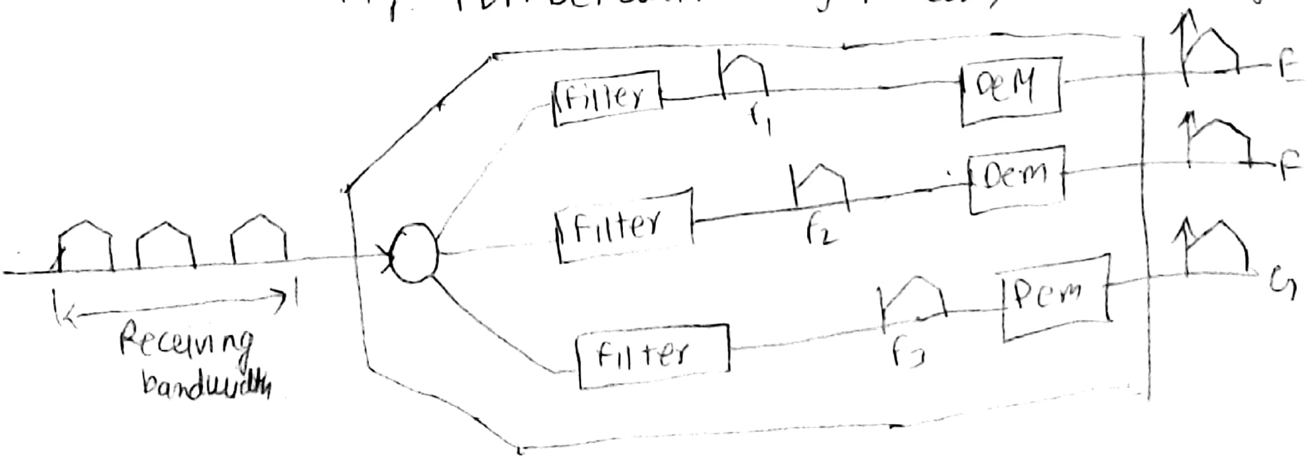


Fig. FDM Demultiplexing Process, Time domain



WAVE DIVISION MULTIPLEXING (WDM) :→ Same as FDM, except that the multiplexing and demultiplexing involve light signals transmitted through fiber optic channel.

- The idea is the same, we are combining different signals of different frequencies. However the difference is that frequencies are very high.

→ Technology is very complex, the idea is very simple.

→ Combine multiple light sources into one single light at the multiplexer and do the reverse at demultiplexer.
Combining and splitting of light sources are easily handled by a prism.
Prism bends a beam of light based on the angle of incidence and the freq.

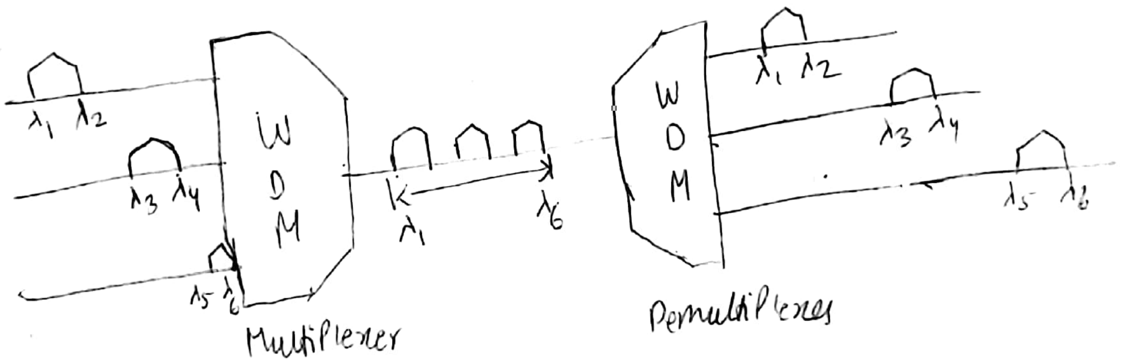


Fig. WDM

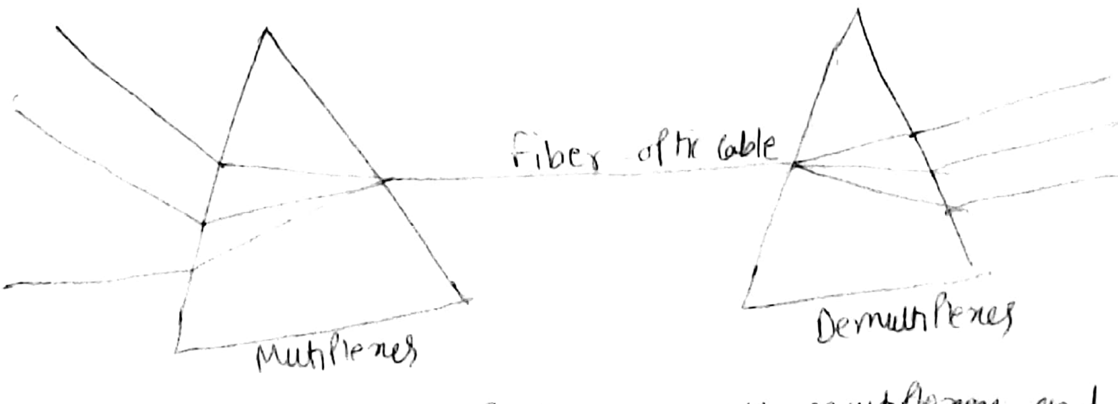


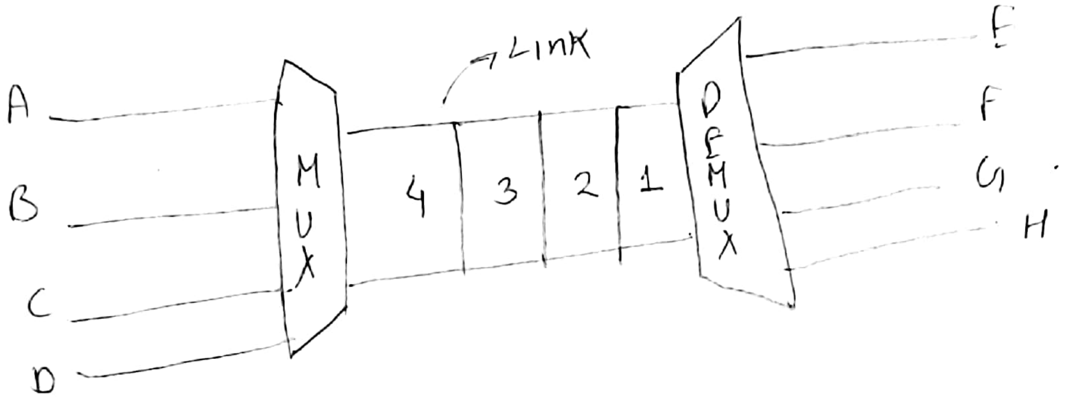
fig. Prism in WDM multiplexing and demultiplexing

TIME DIVISION MULTIPLEXING (TDM) :→

TDM is a digital Process

→ TDM is a digital Process
→ Data rate capacity of medium should be greater than data rate required by the sending and receiving devices.

→ In this link is sectioned by time rather than frequency.



Synchronous TDM :→ here synchronous means that the multiplexer allocates exactly the same time slot to each device at all times, wheather or not a device has any thing to transmit

→ for example time slot assigned for A can not be use by other devices.

→ at allotted time slot device has opportunity to send a portion of its data.

→ If a device is unable to transmit or does not have data to send, its time slot remain empty

Frames :→ Time slots are grouped into frames. A frame consists of one complete cycle of time slots, including one or more slots, dedicated to each sending device. In a system with n input lines, each frame has at least n slots

→ If all the input devices sharing a link are transmitting at the same data rate, each device has one time slot per frame. However it is possible to accommodate varying data rates.

No. of I/P : 5
No. of slots in each frame : 5

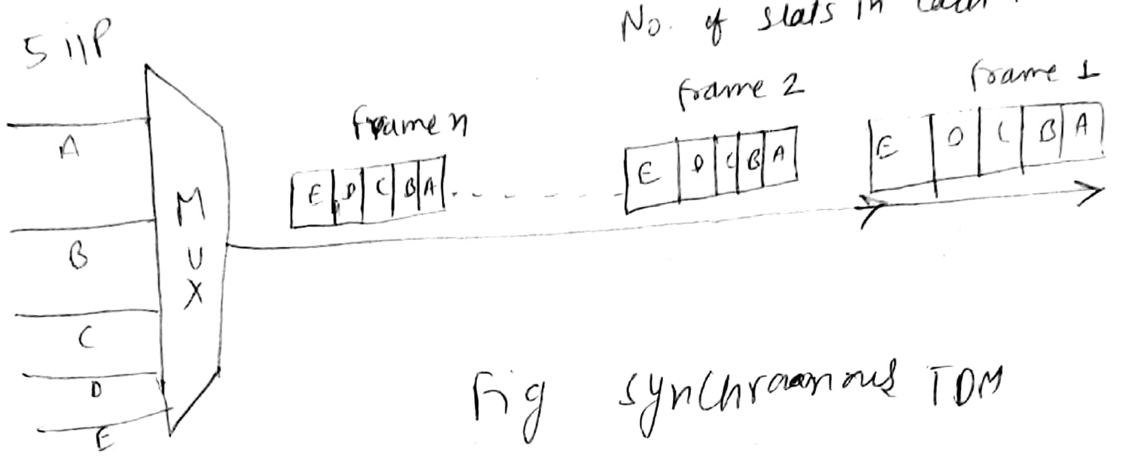


Fig synchronous TDM

Interleaving :→ Syn. TDM can be compared to a very fast rotating switch. The switch moves from device to device at a constant rate and in a fixed order to take the data. This process is called interleaving.

It can be done bit by bit, byte by byte or by any other data unit.

→ The multiplexer interleaves the different messages and form them into frames before putting them onto the link.

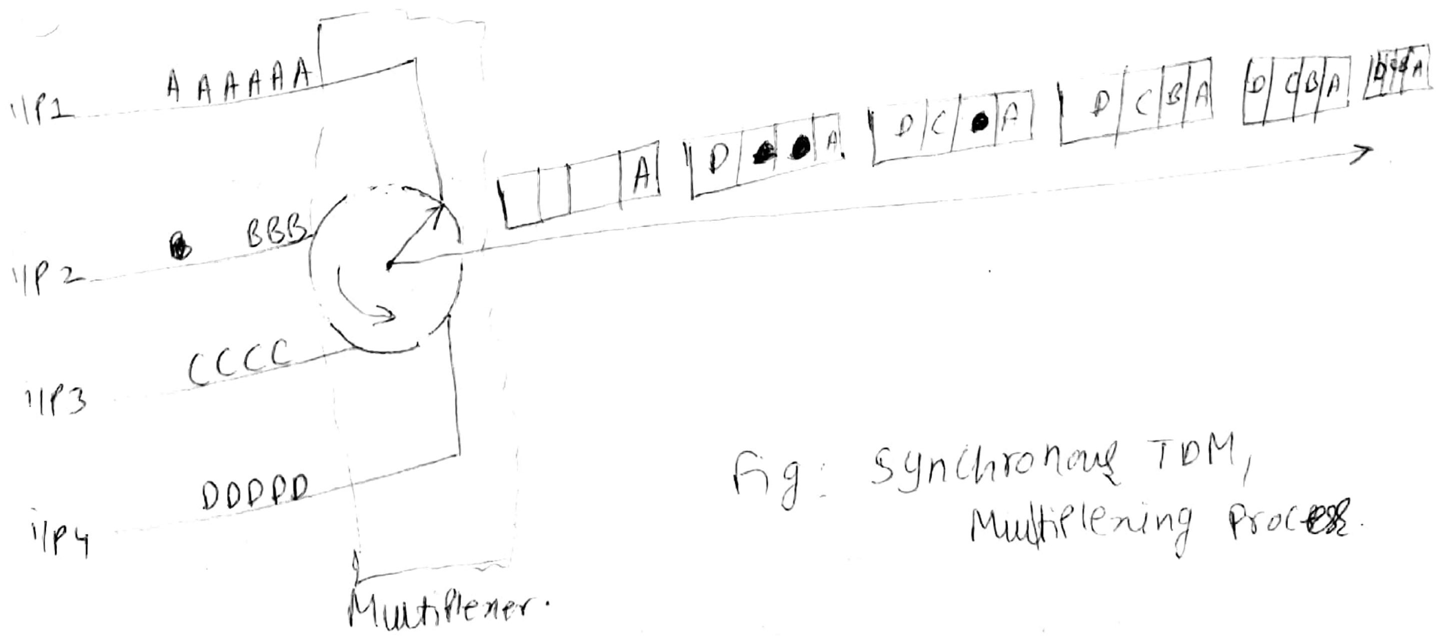
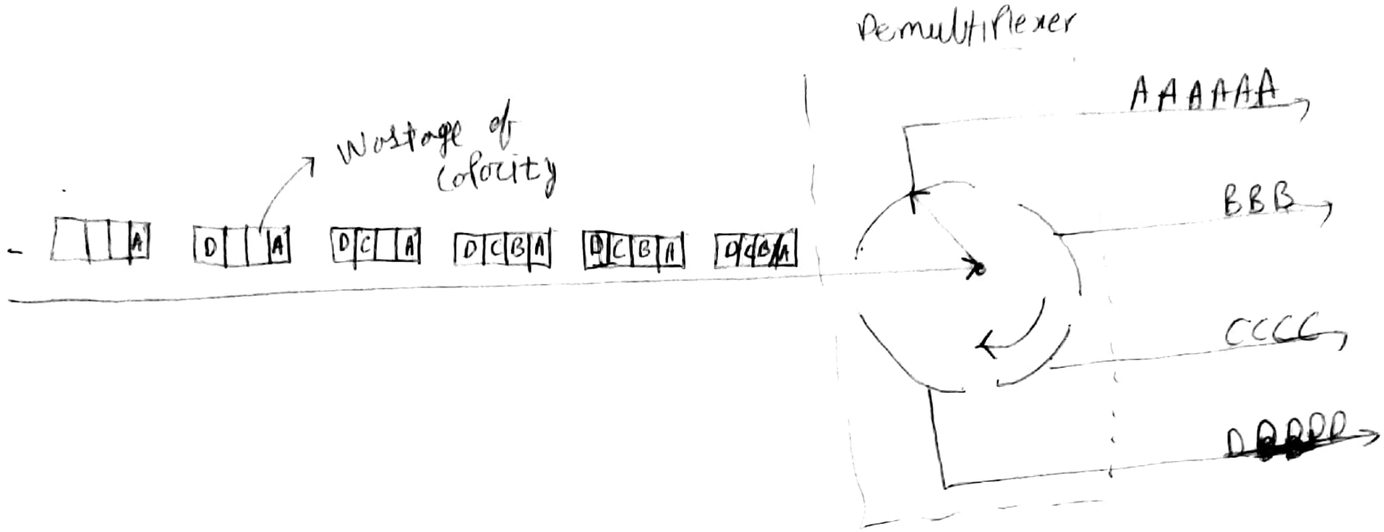


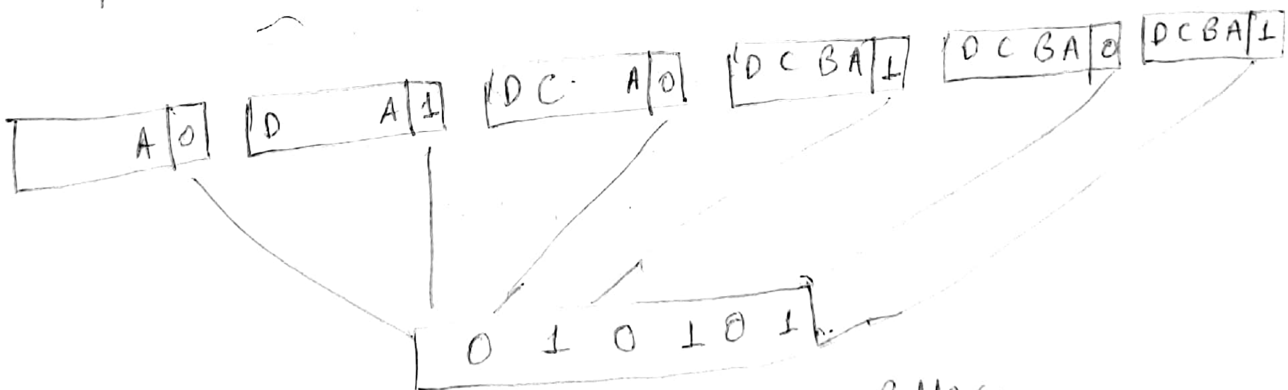
Fig: Synchronous TDM, Multiplexing process.



Framming bits: → Various fortors can cause timing Inconsistences, for this reason, one or more synchronization bits are usually added to the begning of each frame. These bits are called framing bits,

→ It follows a pattern frame to frame, that allows the demultiplexer to synchronize with the incoming stream so that it can separate the time slots accurately.

→ In most case, this synchronization information consists of one bit per frame, alternating between 0 and 1.



Synchronization pattern

Synchronous TDM example: → imagene we have four input sources on a syn. TDM line. each source is creating 250 characters per second and each frame is carring 1 char. from each source, the transmission path must able to carry 250 frames per second.

→ Assume each char 8 bits then Each frame $= 8 \times 4 + 1$ framming bit
$= 33$ bit

→ Each device is creating 2000 bps (250 × 8 bit per char)

→ And the line is carrying 8250 bps (250 frames with 33 bits per frame): 8000 bits of data and 250 bits of overload

8250 bps = 250 frames/sec + 33 bits/frame
or
8250 bps = 4 × 2000 bps + 250 synch bps
250 frames/second

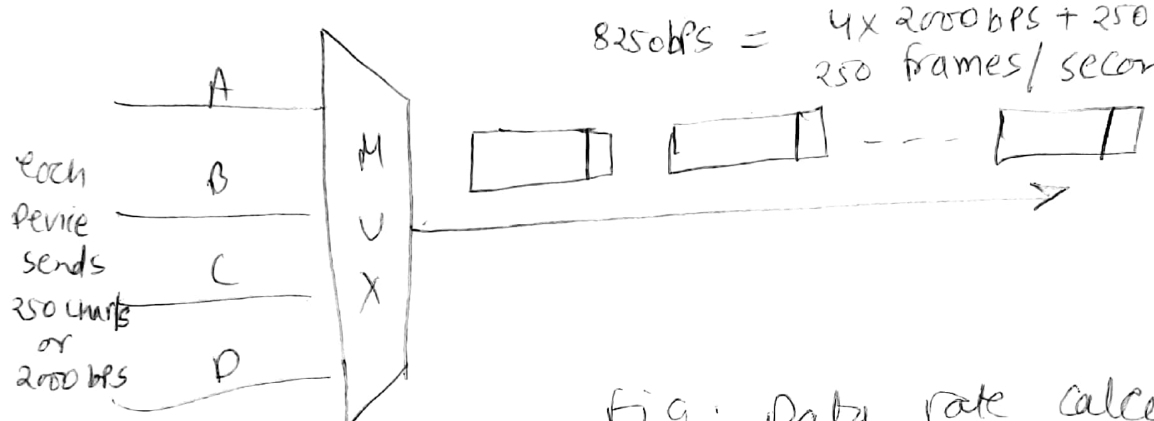


Fig: Data rate calculation for frame

Bit stuffing: → when the speed of each device is not integral multiple of other, they can be made to behave as it they were, by a technique called bit stuffing.

→ In bit stuffing multiplexer adds extra bits to devices source stream to force the speed relationship among the various devices into integral multiple of each other

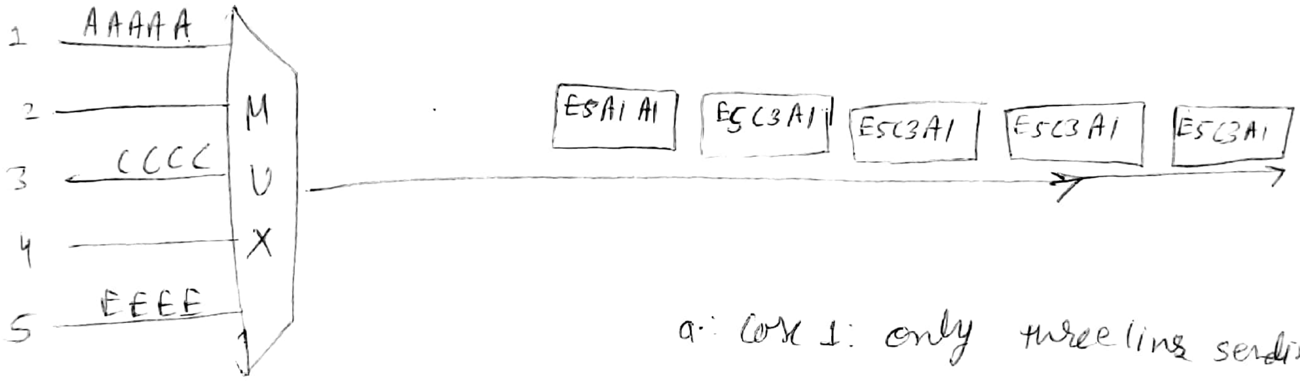
→ For ex: If one device has bit rate of 2.75 times that of the other device, we can add enough bits to raise the rate to 3 times that of the other. The extra bits are then discarded by the demultiplexer.

Asynchronous TDM :→ disadvantage of Syn. TDM
In Syn TDM full capacity of link in not used.

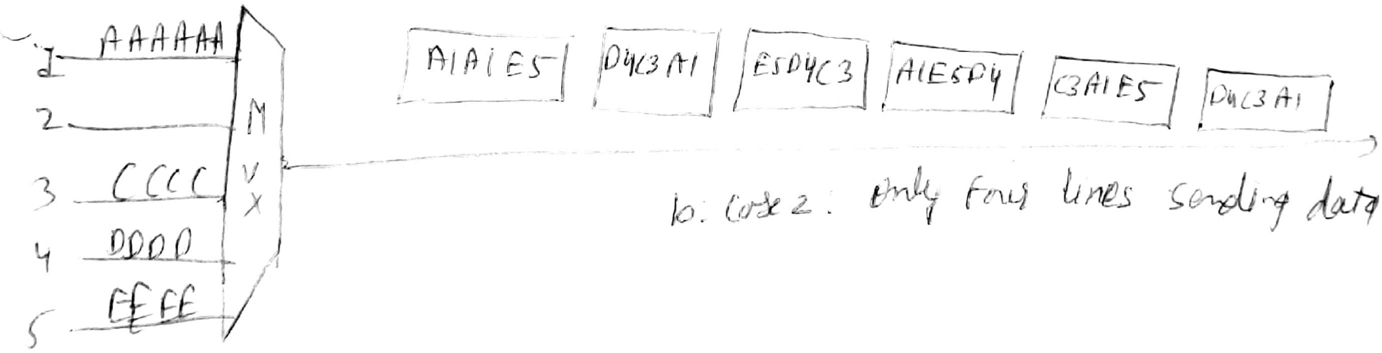
→ As the time slots are preassigned and fixed in Syn. TDM a problem arises that when connected device is not transmitting, the corresponding slot is empty and ~~that~~ that much of the path is wasted.

→ Asy. TDM or Statistical TDM is designed to avoid this type of waste.

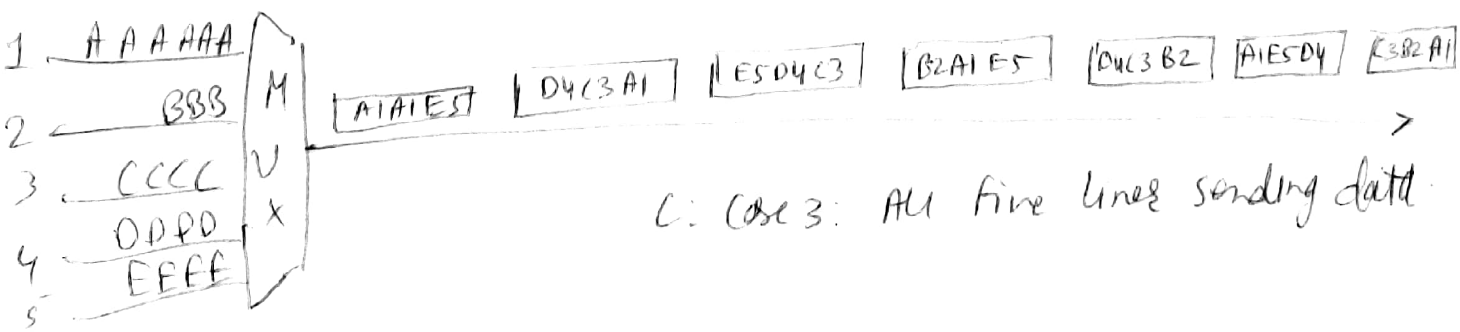
→ Here asy. means flexible or not fixed.



a: Case 1: only three lines sending data



b: Case 2: only four lines sending data



c: Case 3: All five lines sending data

→ In an asy. system. If we have n i/p lines, the frame contains no more than m slots with m less than n

here n = no. of i/p. and m = no of slots.

→ Asy. TDM can support more devices than synchronous TDM

Addressing and overhead → How does the Multiplexer know which slot belongs to which output line?

each time slot must carry an address telling the demultiplexer how to direct data. This address is attached by multiplexer and remove by demultiplexer after ~~the~~ it has been read.

disadvantage of Asy. TDM :→

Adding address bits to each time slot increases the overhead of an asynchronous system and somewhat limits its potential efficiency.

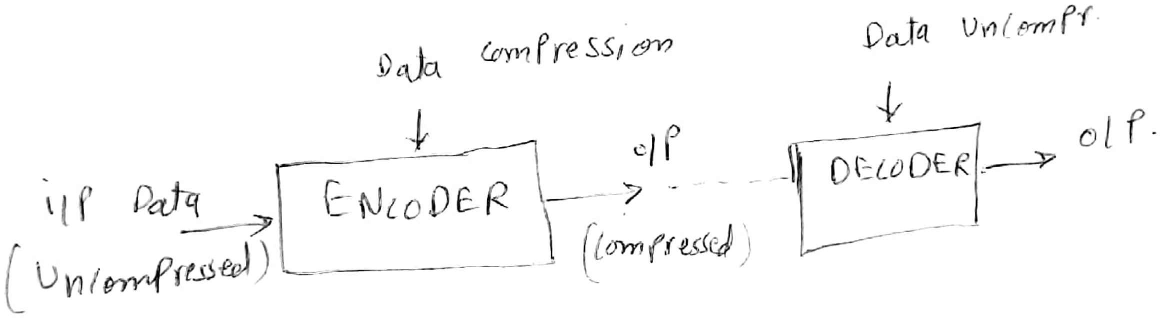
overhead :→ suppose one bit + 3 bit address is require so this 3 bits are overhead.

Variable length time slot: → Asy TDM can accommodate traffic of varying data rates by varying the length of the time slot.

Data Compression Introduction → It is the process of reducing the amount of data required to represent a given quantity of information

Video file (2hr) → 2 GB (Uncompressed)
↳ 750 MB (Compressed) 2 hr

Data comp. : ↓ amount of disk space required
↓ Transmission B.W.



TYPES of Data compression

↓ Lossless compression
No Data is Lost (Exact replica can be extracted)
ex: Text Encryption
(Run Length, Huffman code,

↓ Lossy compression
Some info. is lost, less imp. inf. from media is removed.
ex: Image, Audio, Video.

Terms : →

1) Compression Ratio : → $\frac{\text{Compressed file size}}{\text{origional file size}}$

2) Compression fortion : → 1 / Compression Ratio

3) Compression time : → Time require to compress. (ms)

4) Decompression Time : → Time require to decompress

To

Fixed Length Encoding :→

	ASCII		Let convert in 3 bit	
a →	97 →	7 bits (1100001) →		a → 000
b →	⋮			b → 001
c	⋮			c → 010
d	⋮	all 7 bit	7×7 = 49 bits	d → 011
e	⋮			e → 100
f	⋮			f → 101
g	⋮			g → 110

7×3 bits = 21 bits

Uncompressed data — Compressed Data.

Run Length Coding (RLE) :→ simplest (Lossless) data comp.

→ Data are stored as a single data value and count rather than origional.

runs :→ seq. of same symbol / data value.

Let we have long seq of B

$\underbrace{BBBBBBBB}_{\text{Long seq of B}}$:→ $\underbrace{B}_{\text{Data value}}$ $\underbrace{8}_{\text{count of data value}}$

Ex: $\underbrace{BBBBBBBBB}_{09}$, $\underbrace{AAAAA}_{05}$, $\underbrace{N}_{01}$ $\underbrace{GG}_{02}$ $\underbrace{MMM}_{03}$, → total 20 data

↓ after RLE

B09 A05 N01 G02 M03 → total 15 data

Ex 000000000000000 | 0000 | 1 | 000000000000,

14 4 0 12

1110 0100 0000 1100

Binary of 14, Bin of 4, Bin of 0, Bin of 12

can use 01 or 1

Drawback → ex xyz → 3 data

01 01 01 or

↓ RLE ↓ RLE

x1 y1 z1 x01 y01 z01

$\underbrace{x1\ y1\ z1}_{6\ data}$ $\underbrace{x01\ y01\ z01}_{9\ data}$

So when repeating values are not there then no. of bit increases

Frequency dependant code

HUFFMAN CODE Data COMPression (Variable Length code)

Based on binary tree frequency sorting method that allows to encode any message sequence into shorter encoded messages

Algorithm: → i) Create sorted nodes based on probability / freq.

ii) Start loop

iii) Find and remove two smallest Prob Node

iv) Create new node [Wt [node] = Wt[N1] + Wt[N2] (weight)

v) Insert new node, back to sorted list.

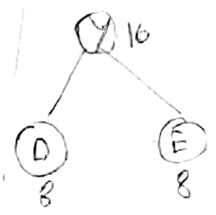
vi) Repeat the loop until only one last node is present in the list

Ex1. ~~Ento~~ compress the following data symbol with given freq as follows.

→ from final tree

Symbol	Freq	Code	Code Length
A	24	→ 0	→ 1
B	12	→ 101	→ 3
C	10	→ 100	→ 3
D	08	→ 110	→ 3
E	08	→ 111	→ 3

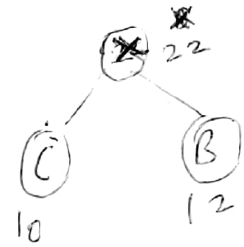
add (D 08 + E 08)



New Table 1

A - 24   B - 12] add

X - 16   C - 10]



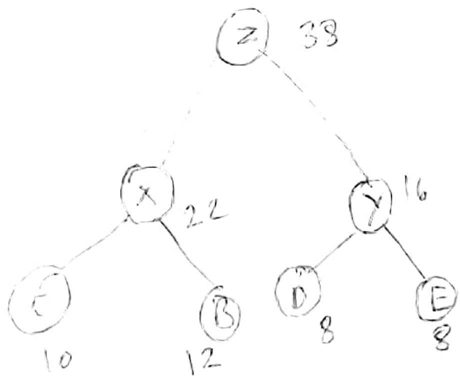
Add above two trees

New Table 2

A - 24

Y - 22] add

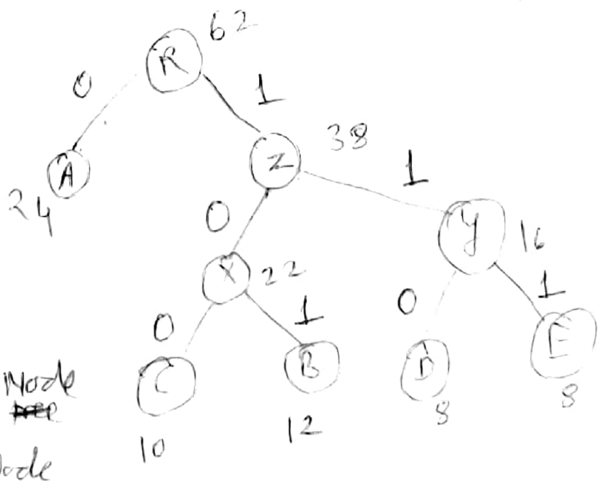
X - 16]



New Table 3

A - 24] add

Z - 38]



→ Now assign 0 to left of Node ~~tree~~ and 1 to Right of Node

Ex- 2

A 45

B 13

C 12

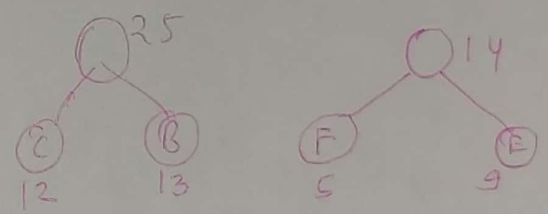
D 16 → first sort this table 14

E 9

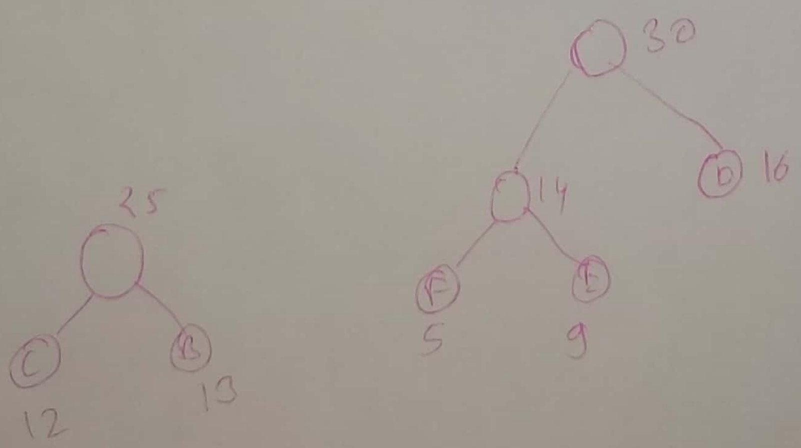
F 5

A	45
D	16
B	13
C	12
E	9
F	5

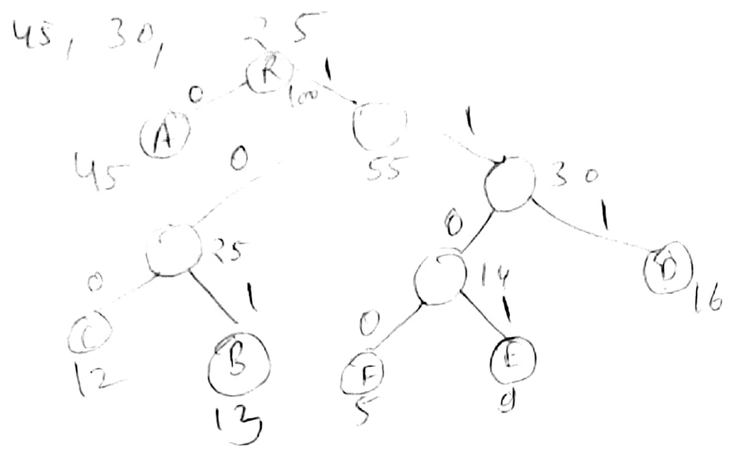
$45, 16, 14, \underbrace{13, 12}_{\text{add}}$



$45, 25, \underbrace{16, 14}_{\text{add}}$



45, 55 → 100



A	45 →	0 →	1
B	13 →	101 →	3
C	12 →	100 →	3
D	16 →	111 →	3
E	9 →	1101 →	4
F	5 →	1100 →	4

Now suppose we have

A A B C D . . . . . .

then

0 0 101 100 111 . . .
A A B C D

Lampel Zev (Assignment)

It is accomplished by parsing the ~~source~~ source data stream into segment that are the shortest subsequent not encountered previously.

Ex: 0 0 0 1 0 1 1 1 0 0 1 0 1 0 0 1 0 1

It is assumed that 0 and 1 are already stored in that order

	1	2	3	4	5	6	7	8	9
a)	0,	1,	00,	01,	011,	10,	010,	100,	101

Numerical Posn: 1 2 3 4 / 0 1

No. repn ← 11 12 42 21 41 61 62

Convert to binary; 1 is at loc 0; 2 is at loc 1

Binary Encoded Block: 0010 0011 1001 0100 1000 1100 1101

75

Ex 2: A|A B|A B B|B A B A|A B A B|B B|A B B A|B B|

Pos^n:	1	2	3	4	5	6	7	8	9
seq:	A	AB	ABB	B	ABA	ABAB	BB	ABBA	BB
Numerical Position	∅A	1B	2B	∅B	2A	5B	4B	3A	7
Binary	0	01B	101	1	10,0	101,1	100,1	110	0111

7 will be use for BB

$A \rightarrow 0$
$B \rightarrow 1$

We code the phrase by writing the location of the prefix and value of the last bit