Source coding theorem

Source Coding Theorem : Suppose a DMS outputs a symbol every t seconds. Each symbol is selected from a finite set of symbols x_i, $i = 1, 2, \dots, L$, occuring with probabilities $P(x_i)$, $i = 1, 2, \dots, L$. The entropy of this DMS in bits per source symbol is

$$H(X) = \sum_{i=1}^{L} P(x_i) \cdot \log \frac{1}{P(x_i)} \leq \log_2 L$$

The equality holds when the symbols are equally likely.

Source Coding Theorem : Suppose a DMS outputs a symbol every t seconds. Each symbol is selected from a finite set of symbols x_i, $i = 1, 2, \dots, L$, occuring with probabilities $P(x_i)$, $i = 1, 2, \dots, L$. The entropy of this DMS in bits per source symbol is

$$H(X) = \sum_{i=1}^{L} P(x_i) \cdot \log \frac{1}{P(x_i)} \leq \log_2 L$$

The equality holds when the symbols are equally likely.

$$H(X) = -\sum_{i=1}^{L} P(x_i) \log P(x_i) \leq \log_2 L$$

x_1 $\quad P(x_1) = 0.5$

x_2 $\quad P(x_2) = 0.5$

$$H(x) = -0.5 \log_2 0.5 - 0.5 \log_2 0.5$$
$$= -0.5 \log_2 2^{-1} - 0.5 \log_2 2^{-1}$$
$$= -0.5(-1) \log_2 2 - (0.5)(-1) \log_2 2$$
$$= 0.5 + 0.5 = 1$$

...Coding Theorem : Suppose a DMS outputs a symbol every t seconds. Each symbol is selected from a finite set of symbols x_i, $i = 1, 2, \dots, L$, occuring with probabilities $P(x_i)$, $i = 1, 2, \dots, L$. The entropy of this DMS in bits per source symbol is

$$H(X) = \sum_{i=1}^{L} P(x_i) \cdot \log \frac{1}{P(x_i)} \leq \log_2 L$$

The equality holds when the symbols are equally likely.

$$H(X) = -\sum_{i=1}^{L} P(x_i) \log P(x_i) \leq \log_2 L$$

x_1 $P(x_1) = 0.5$

x_2 $P(x_2) = 0.5$

$$H(x) = -0.5 \log_2 0.5 - 0.5 \log_2 0.5$$
$$= -0.5 \log_2 2^{-1} - 0.5 \log_2 2^{-1}$$
$$= -0.5(-1) \log_2 2 - (0.5)(-1) \log_2 2$$
$$= 0.5 + 0.5 = 1$$

$$3010 \overline{)\,4771\,} \; 1.58$$
$$3010$$

x_1 $P(x_1) = 0.5$

x_2 $P(x_2) = 0.25$

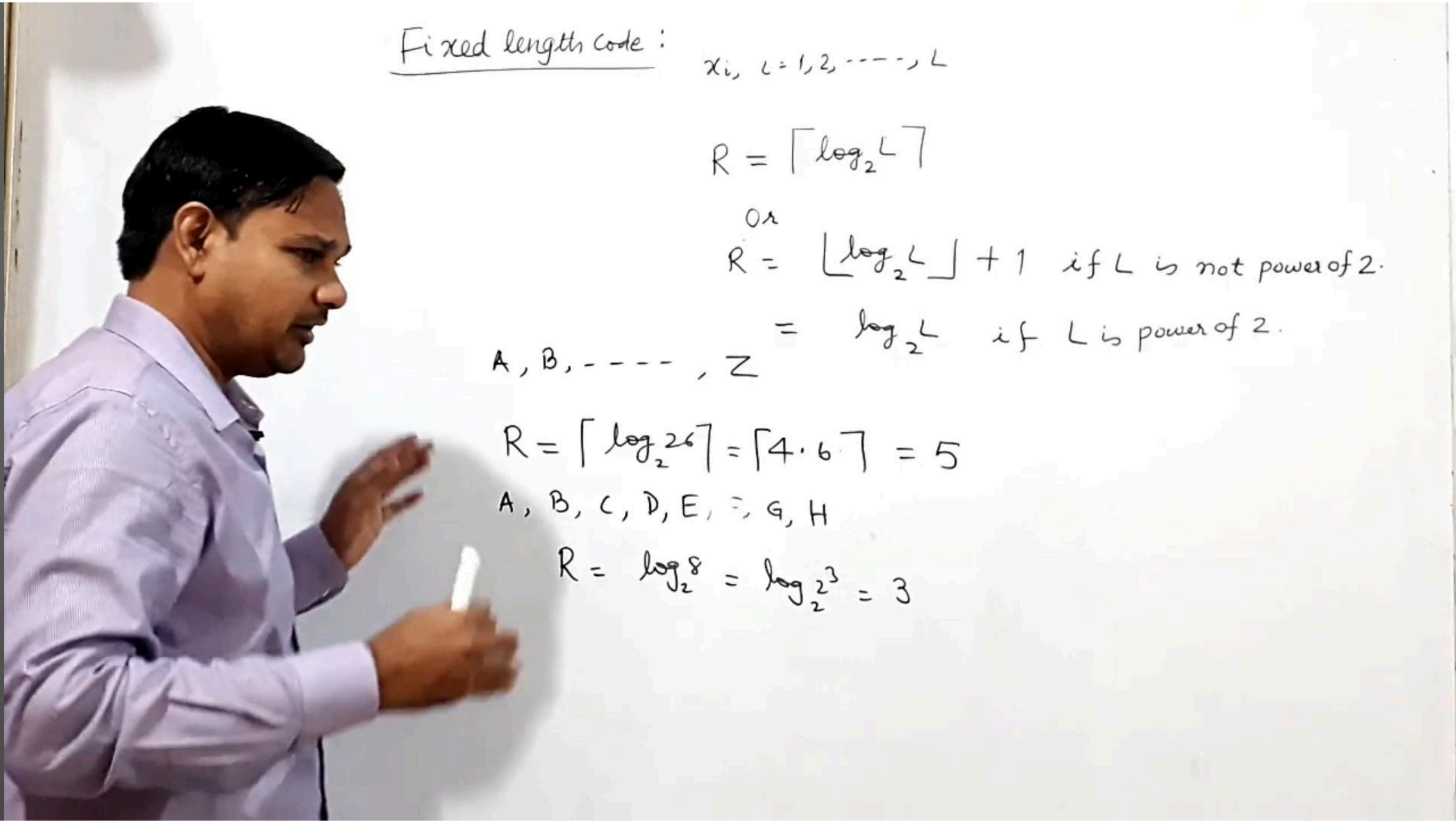
x_3 $P(x_3) = 0.25$

LHS

$$H(x) = -0.5 \log_2 0.5 - 0.25 \log_2 0.25 - 0.25 \log_2 0.25$$
$$= -0.5 \log_2 2^{-1} - 0.25 \log_2 2^{-2} - 0.25 \log_2 2^{-2}$$
$$= +0.5 + 0.5 + 0.5 = 1.5$$

RHS

$$\log_2 3 = \frac{\log_{10} 3}{\log_{10} 2} = \frac{0.4771}{0.3010}$$
$$= 1.58$$



Fixed length code :
$x_i,\ i = 1, 2, \ldots, L$
$R = \lceil \log_2 L \rceil$
Or
$R = \lfloor \log_2 L \rfloor + 1$ if L is not power of 2.
$= \log_2 L$ if L is power of 2.
A, B, ----, Z
$R = \lceil \log_2 26 \rceil = \lceil 4.6 \rceil = 5$
A, B, C, D, E, G, H
$R = \log_2 8 = \log_2 2^3 = 3$

Vector v/s Scalar Quantization

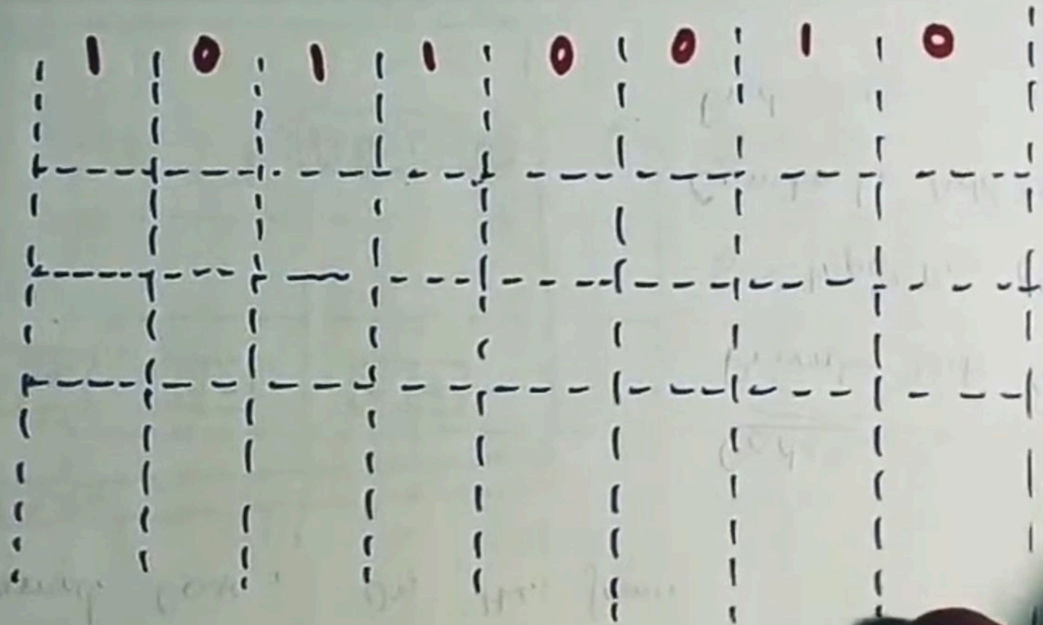
Vector Quantization Vs Scalar Quantization

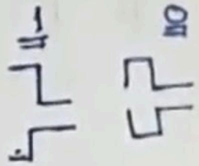
Vector Quantization	Scalar Quantization
1. The input symbols are grouped together called vector and process then to give output.	1. Each input is treated separately and produce the output.
2. Increase the optimality of the quantizer.	2. This is not the case for SQ.
3. More efficient	3. Less efficient
4. The granular error is affected by size and shape of quantization interval.	4. The granular error is determined by the size of quantization interval.
5. $Q(X) = Y_j$ if $d(X, Y_j) \leq d(X, Y_i)$ for all $i \neq j$	5. $Q(x) = y_i$ if $b_{i-1} < x \leq b_i$.
6. VQ is used for low-bit rate applications where low resolution is sufficient. It is widely used for image compression.	6. In image compression, SQ creates annoying effects in decompressed image.
7. For a given rate, VQ results in lower distortion than SQ.	7. For a given rate, SQ results in greater distortion than VQ.

Differential Manchester Encoding

Differential Manchester Encoding

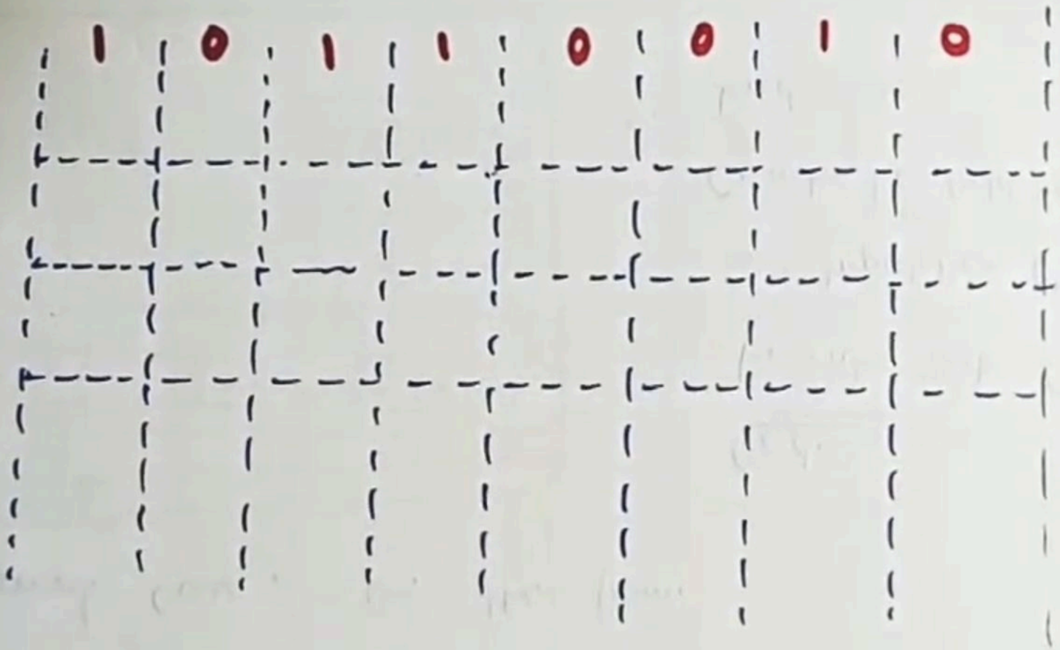
bit Stream = 10110010

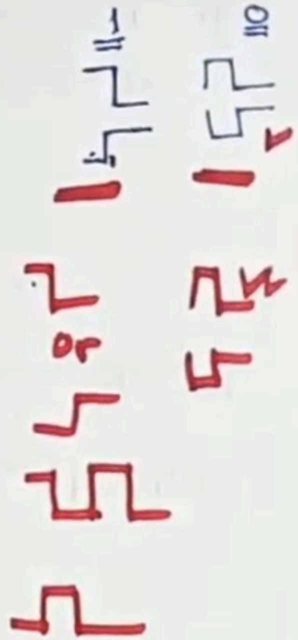




Differential Manchester Encoding

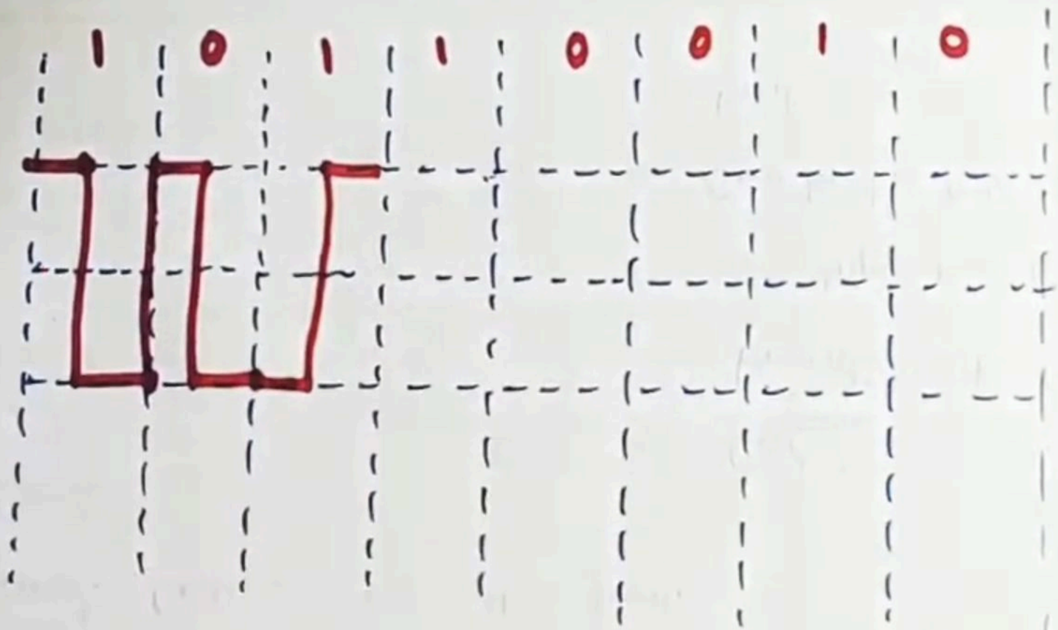
bit Stream = 10110010

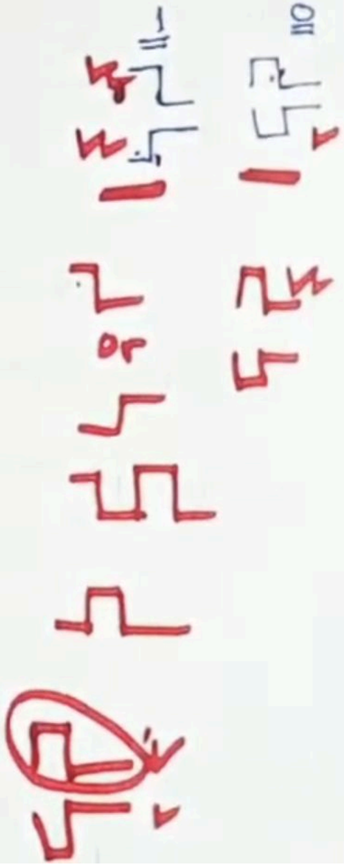




Differential Manchester Encoding

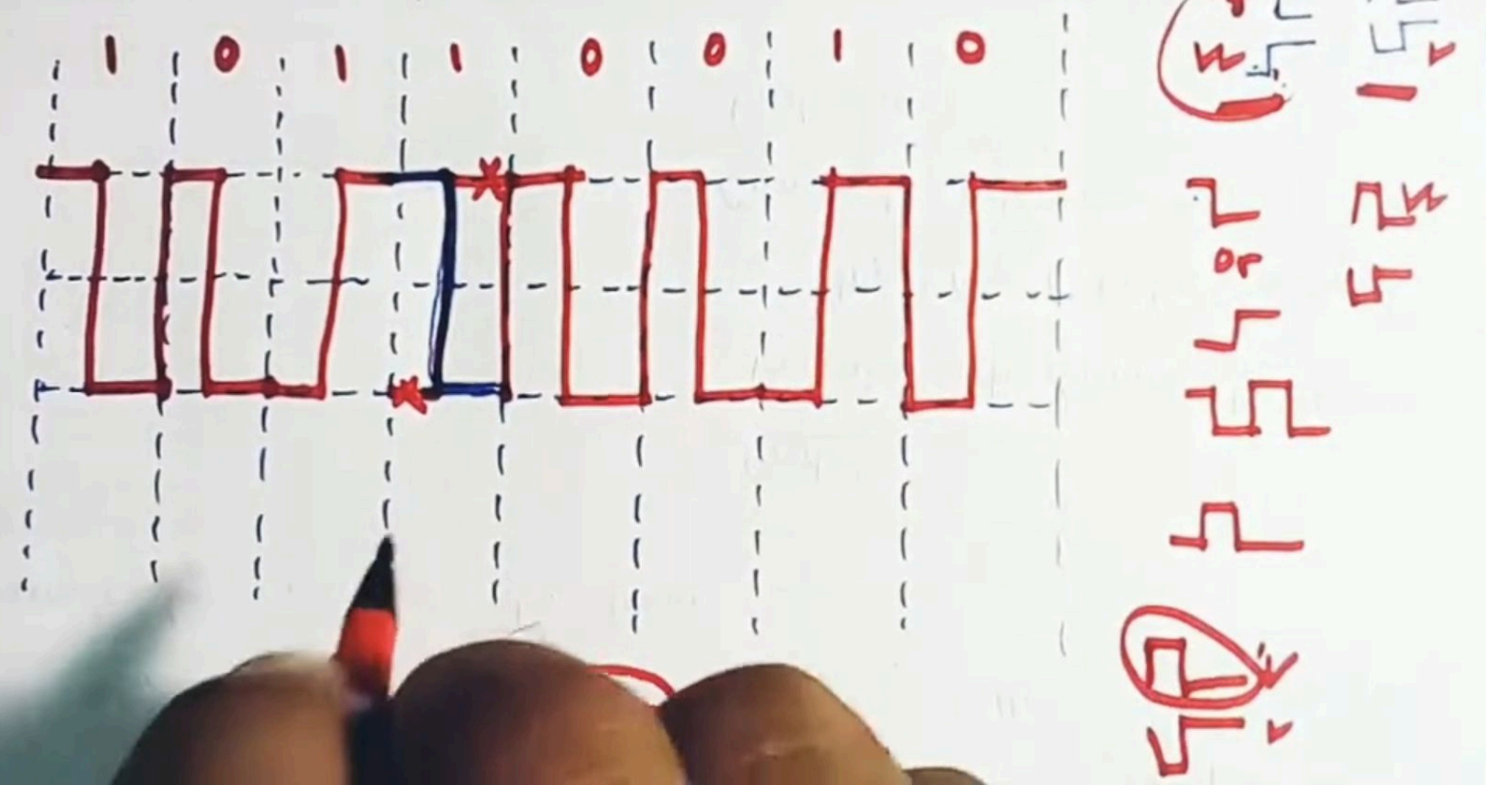
- bit Stream = 1 0 1 1 0 0 1 0





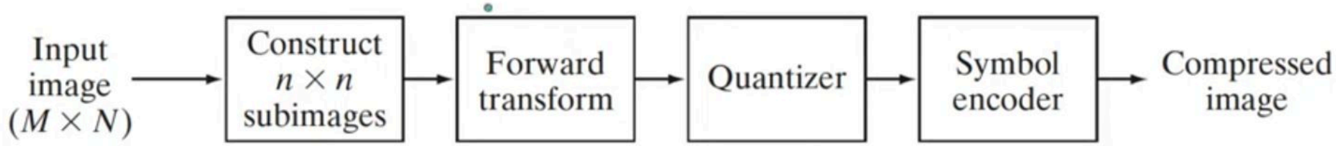
Differential Manchester Encoding

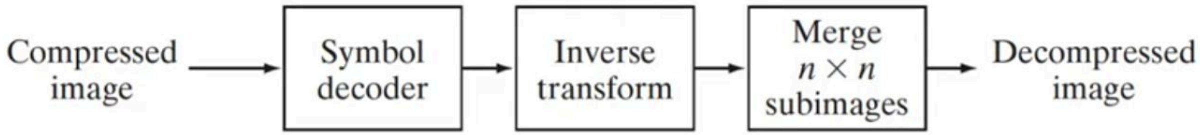
- bit Stream = 10110010



Transform coding

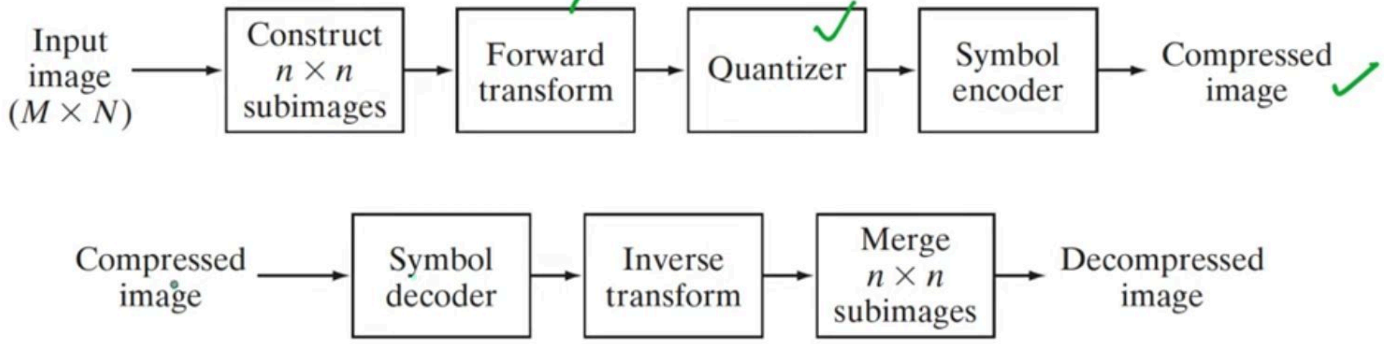
Transform Coding(Block transform coding)

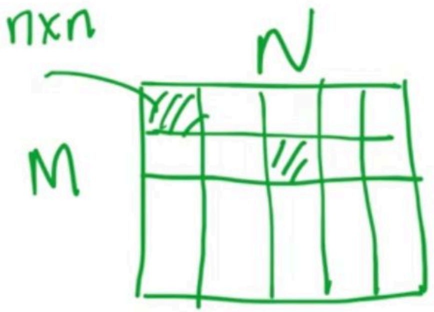




Transform Coding(Block transform coding)

DCT , Hadamard
DFT

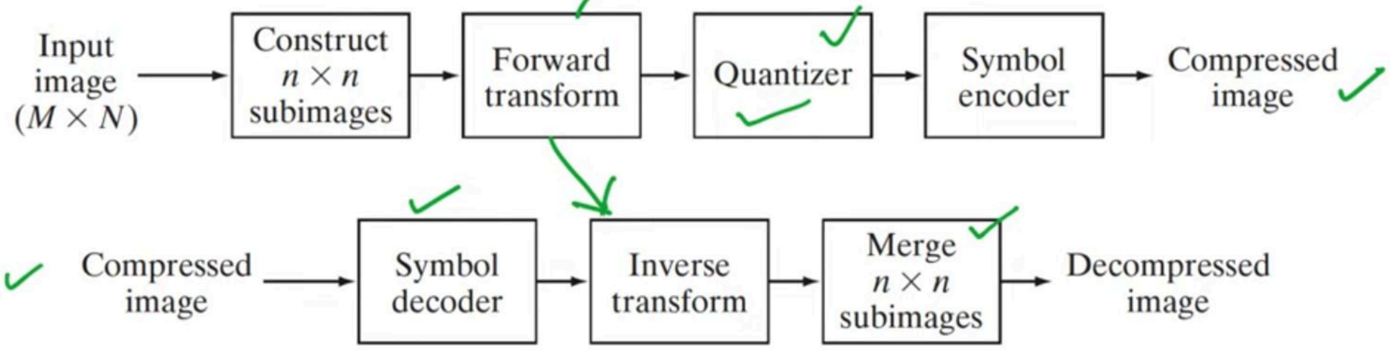


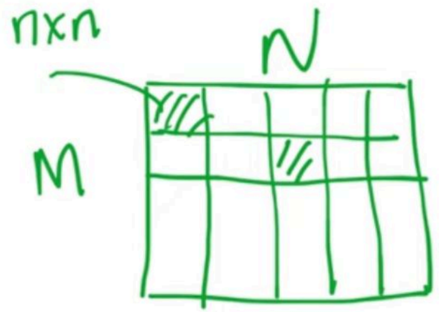


$\lfloor 3.5 \rfloor = 3$

Transform Coding(Block transform coding)

DCT , Hadamard
DFT





$3 \cdot \boxed{5} = 3$

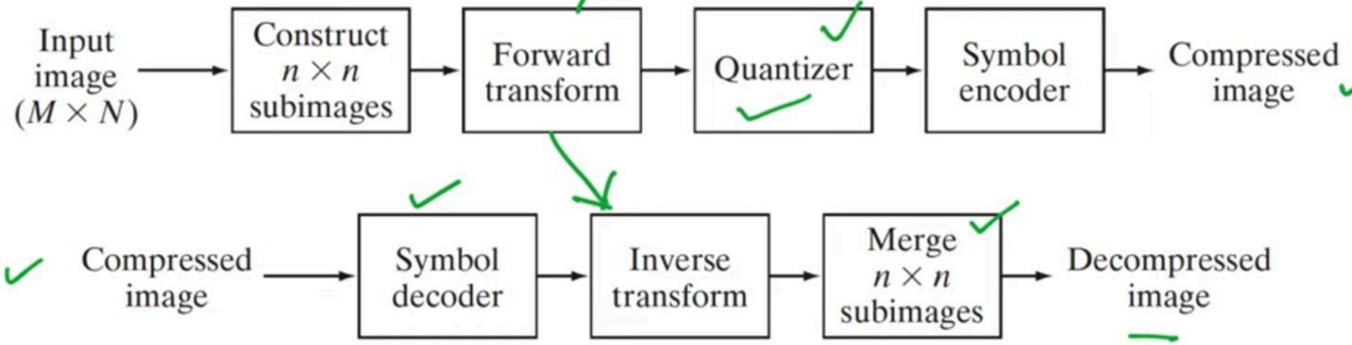
$3 \cdot 4 = 3$

$3 \cdot 9 = 3$

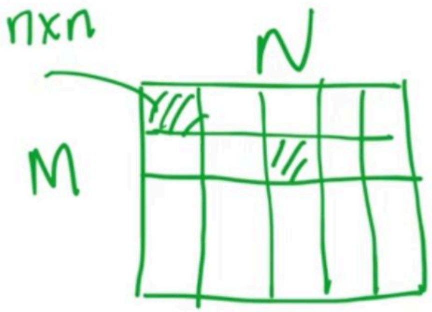
Transform Coding(Block transform coding)

$\boxed{DCT}$, Hadamard

DFT



$$\frac{MN}{n^2} = \text{\#sub images}$$



$3 \cdot 5 = 3$

$3 \cdot 4 = 3$

$3 \cdot 9 = 3$

jpeg

Sub band coding (audio coding)

Introduction to Audio Coding

Audio coding is the process of compressing audio signals to reduce the file size while maintaining acceptable sound quality. This is crucial for applications like music streaming, telecommunication, and multimedia storage.

Key Concepts in Audio Coding:

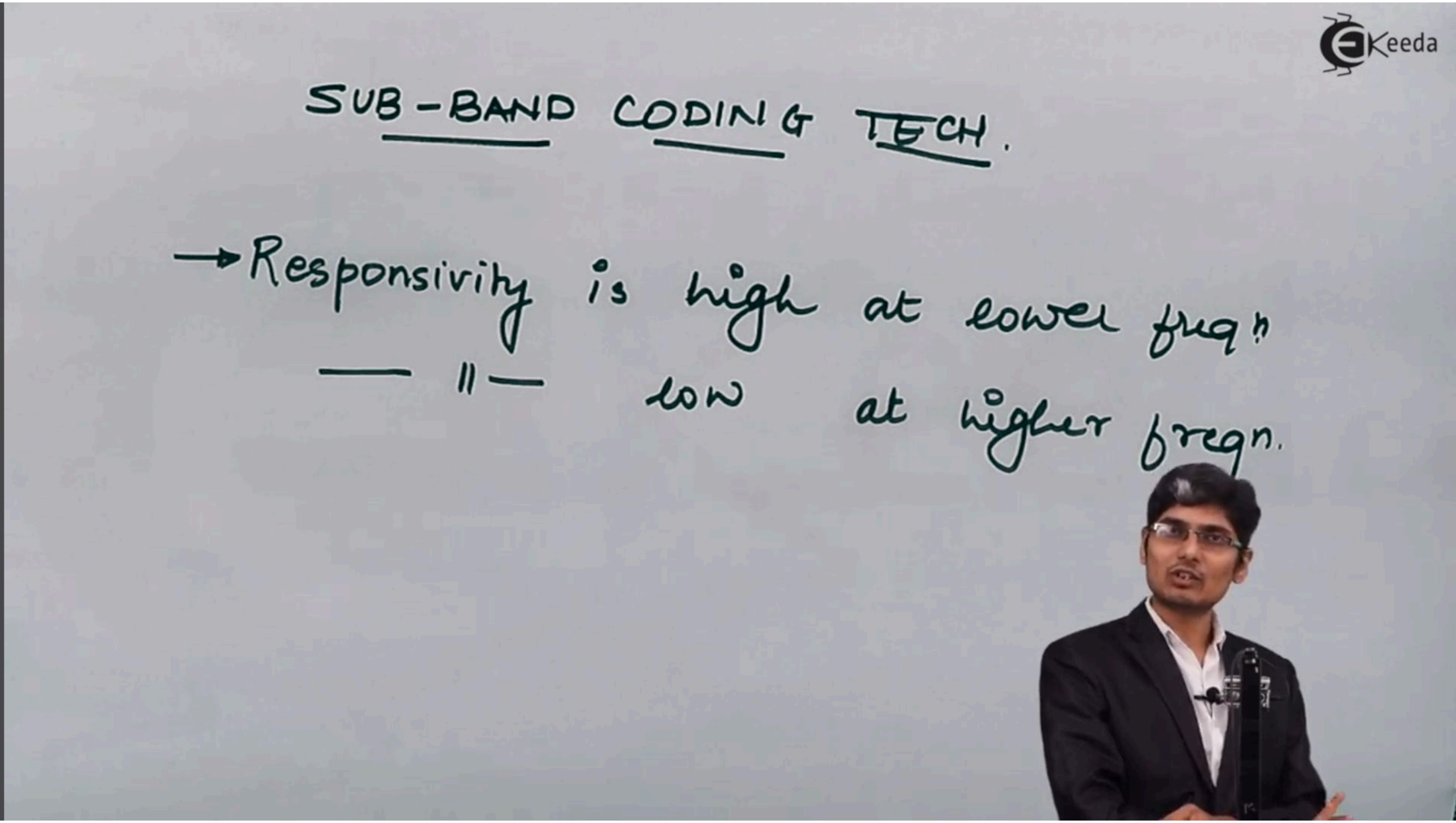
1. **Perceptual Coding:**
   - Human ears do not perceive all frequencies equally.
   - Uses **psychoacoustic models** to remove sounds that are inaudible.
   - Example: **MP3 (MPEG-1 Audio Layer 3)** compresses by discarding frequencies that are masked by sounds.
2. **Lossy vs Lossless Compression:**

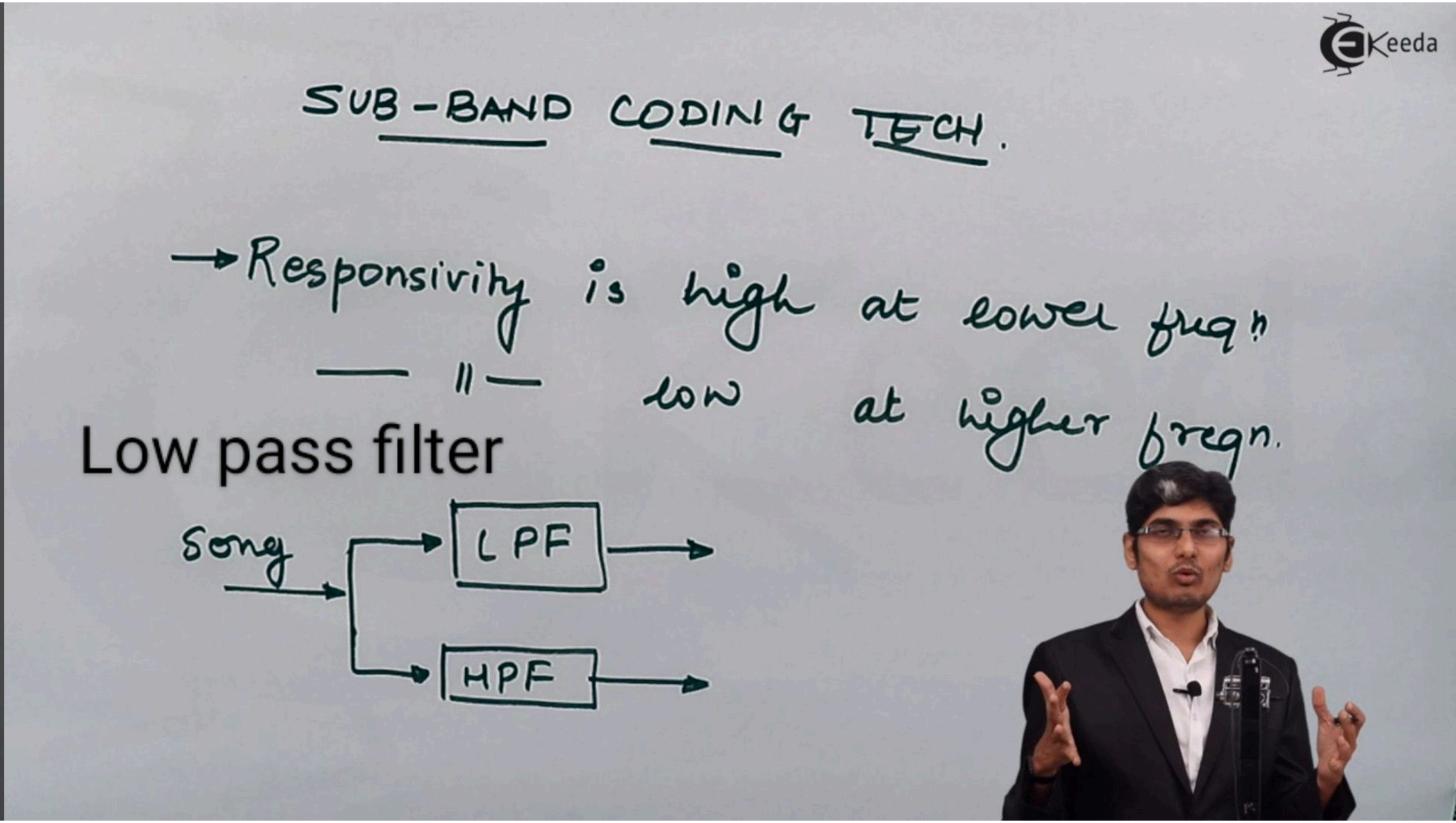
2. **Lossy vs Lossless Compression:**
   - **Lossy Compression**: Some data is discarded permanently. Example: MP3, AAC.
   - **Lossless Compression**: Retains all audio data but reduces redundancy. Example: FLAC, ALAC.
3. **Common Audio Coding Formats:**
   - **MP3**: Uses perceptual coding to reduce file size significantly.
   - **AAC (Advanced Audio Coding)**: More efficient than MP3, widely used in Apple devices and YouTube.
   - **FLAC (Free Lossless Audio Codec)**: No quality loss, but larger file size.
4. **Bitrate and Q[illegible] Trade-offs:**
   - **Higher bitra[illegible] better quality,

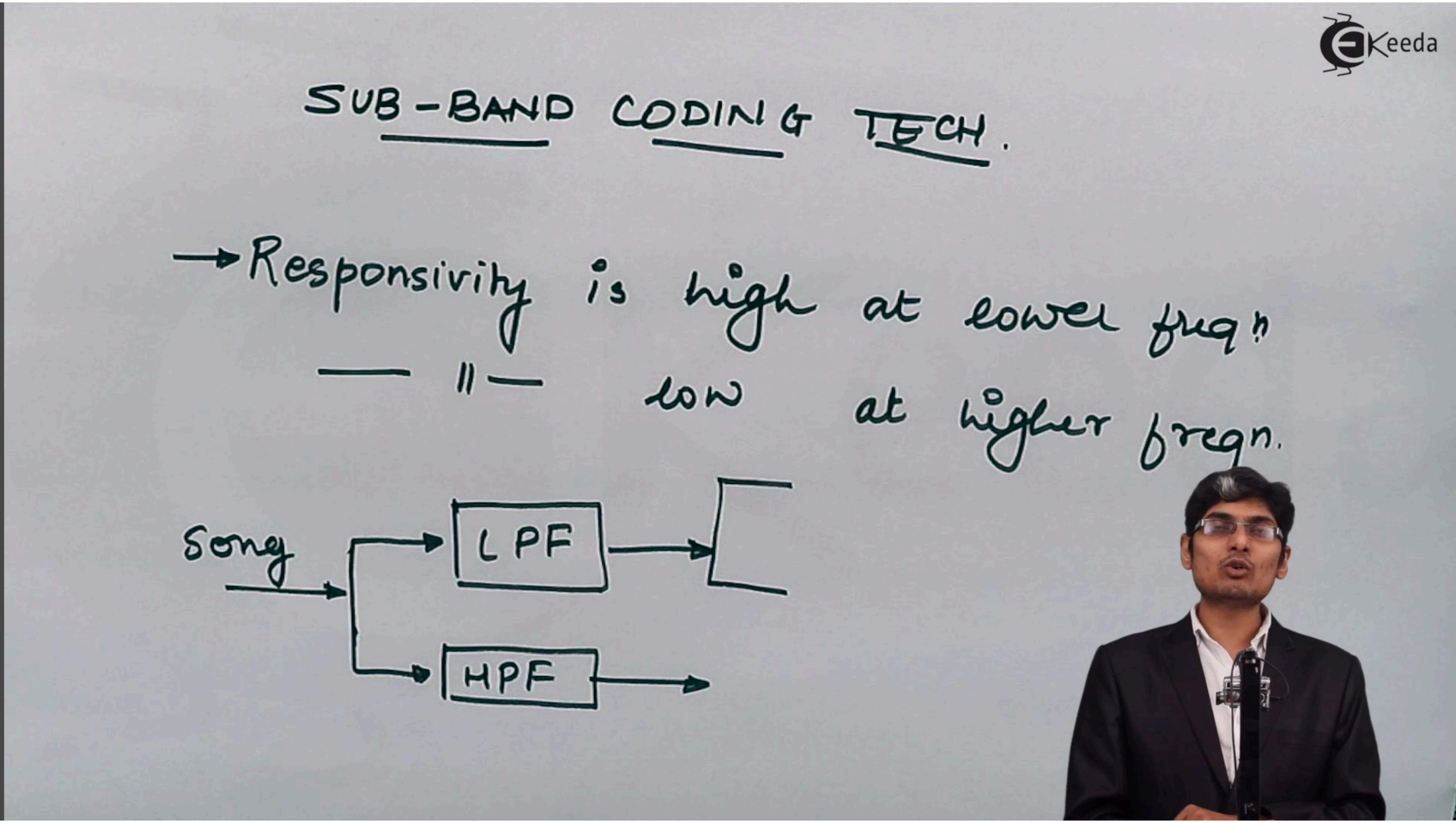
reduce file size significantly.

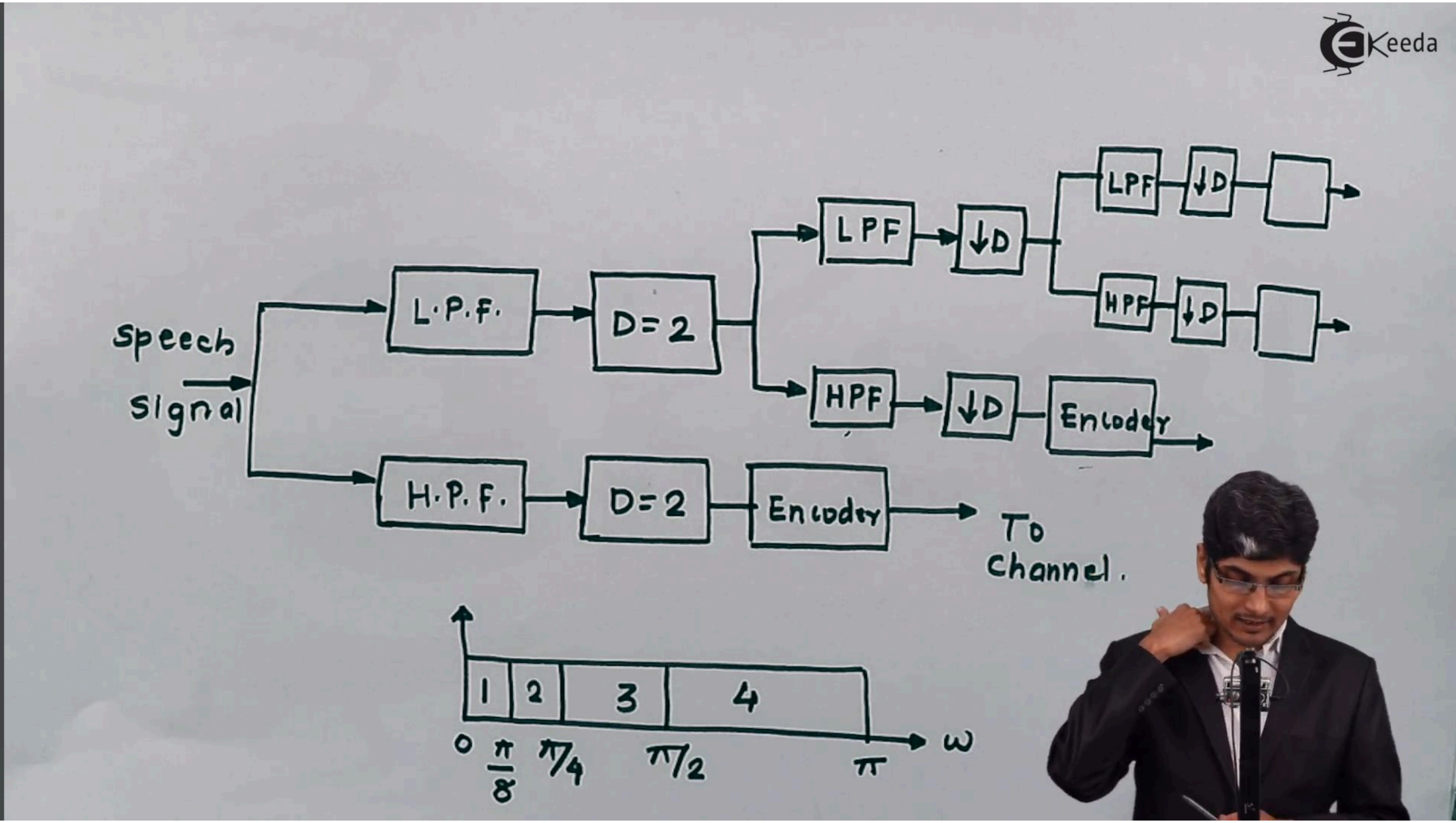
- **AAC (Advanced Audio Coding)**: More efficient than MP3, widely used in Apple devices and YouTube.
- **FLAC (Free Lossless Audio Codec)**: No quality loss, but larger file size.

4. **Bitrate and Quality Trade-offs:**
   - **Higher bitrate** = better quality, larger file size (e.g., 320 kbps MP3).
   - **Lower bitrate** = reduced quality, smaller file size (e.g., 128 kbps MP3).
   - **Variable Bitrate (VBR)** dynamically adjusts compression based on audio complexity.

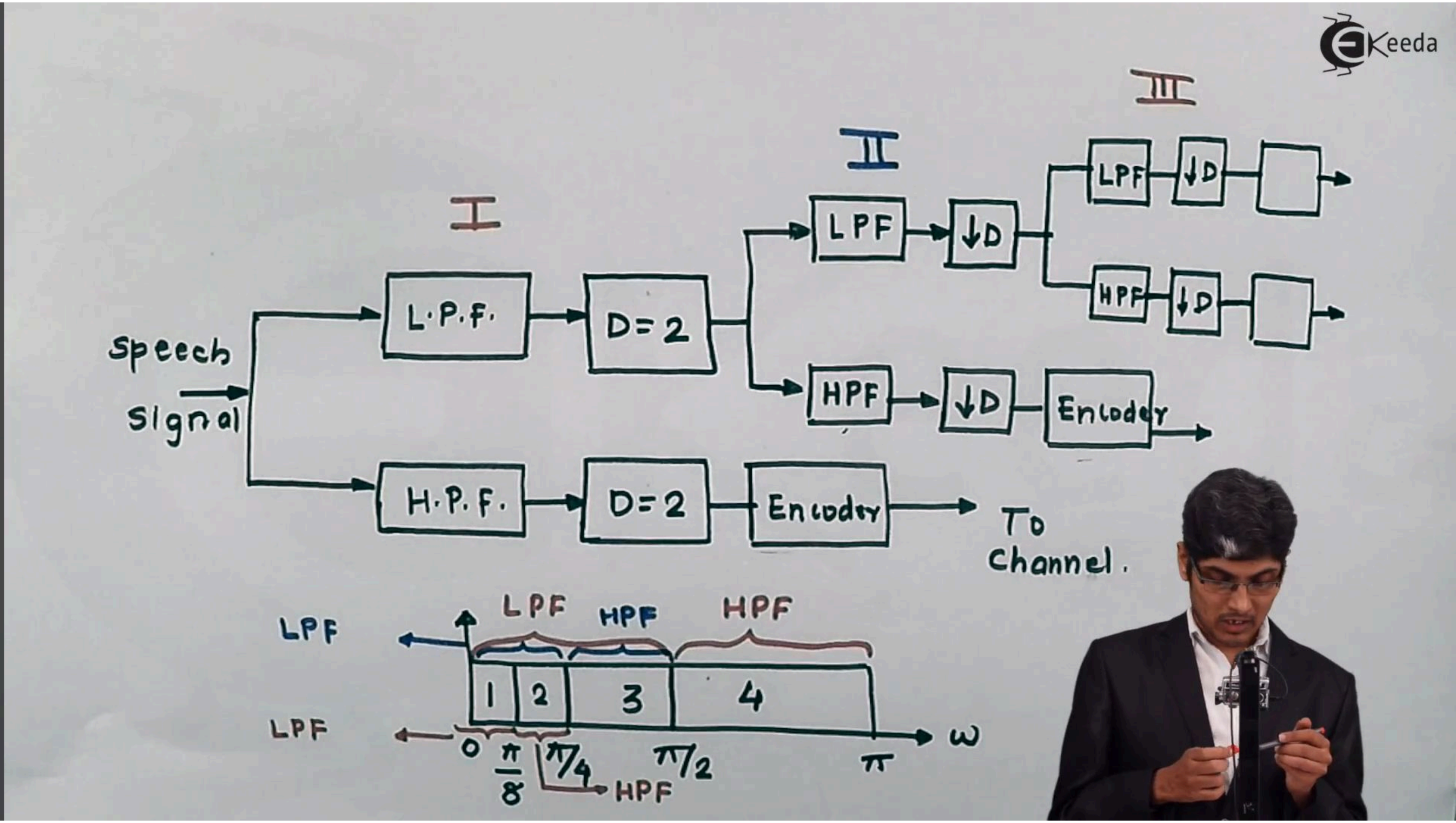

Keeda
SUB-BAND CODING TECH.
→ Responsivity is high at lower freqn
— 〃 — low at higher freqn.


SUB-BAND CODING TECH.
→ Responsivity is high at lower freqn
— " — low at higher freqn.
Low pass filter
Song
LPF
HPF

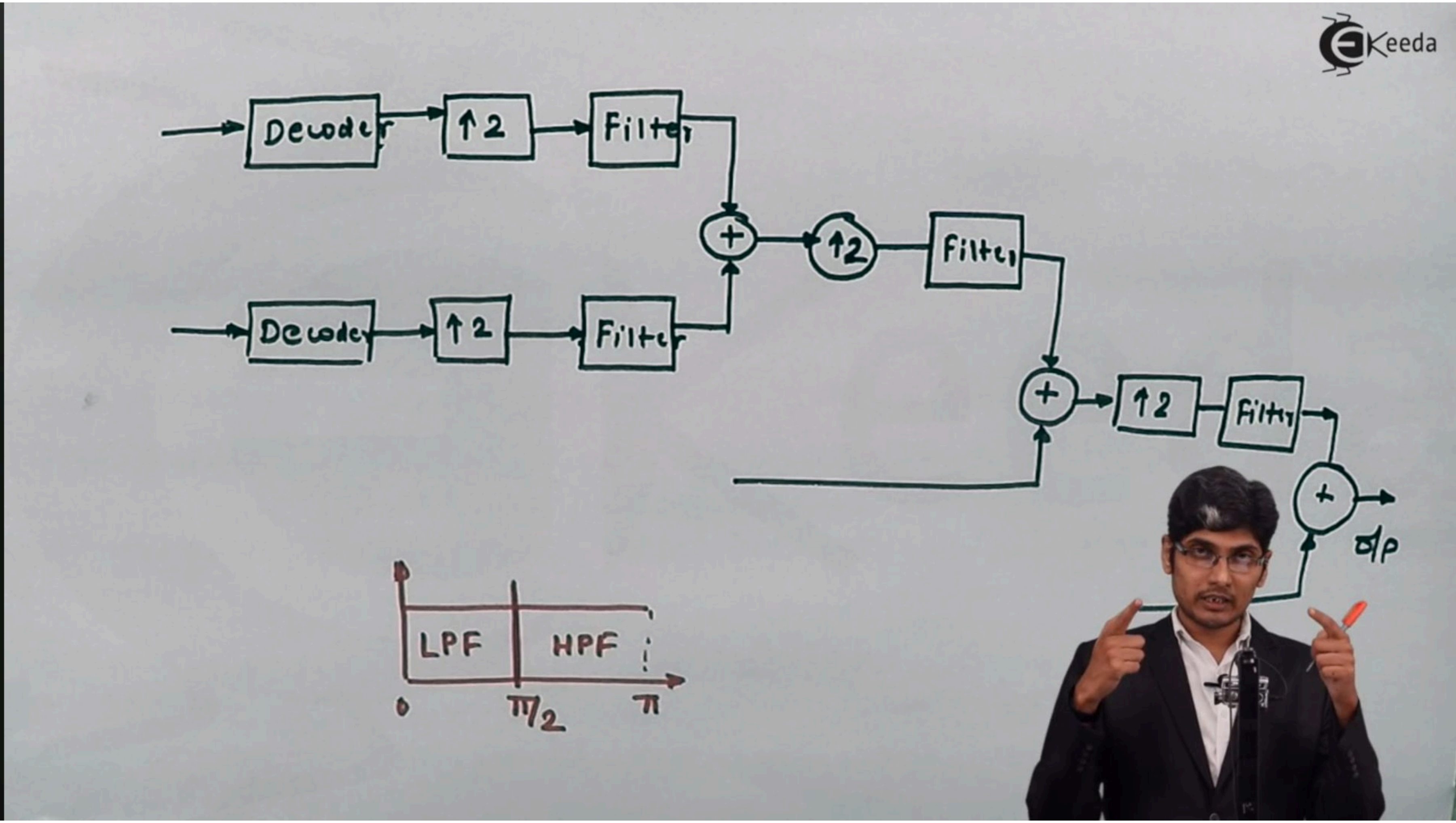

SUB-BAND CODING TECH.
→ Responsivity is high at lower freqn
— " — low at higher freqn.
Song
LPF
HPF
Keeda

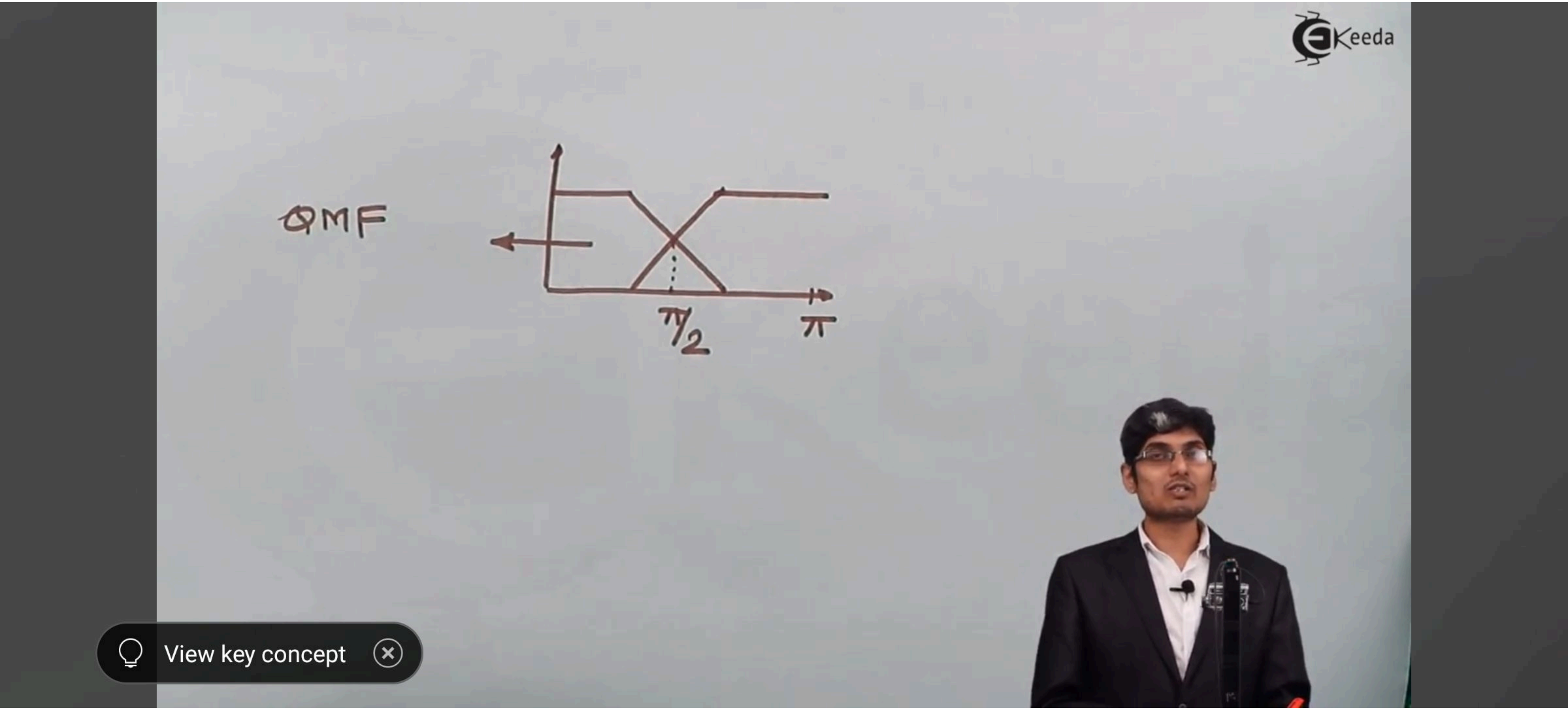


Speech signal
L.P.F.
D=2
LPF
↓D
LPF
↓D
HPF
↓D
HPF
↓D
Encoder
H.P.F.
D=2
Encoder
To channel.
1
2
3
4
0
π/8
π/4
π/2
π
ω



Ekeeda
I
II
III
Speech signal
L.P.F.
D=2
H.P.F.
D=2
Encoder
To channel.
LPF
↓D
HPF
↓D
Encoder
LPF
↓D
HPF
↓D
LPF
HPF
HPF
LPF
LPF
1
2
3
4
0
π/8
π/4
π/2
π
ω
HPF


Decoder
↑2
Filter
Decoder
↑2
Filter
+
↑2
Filter
+
↑2
Filter
+
o/p
LPF
HPF
0
π/2
π


QMF
π/2
π
View key concept

Video Compression

Overview of Video Compression

Video compression reduces the amount of data required to represent video content while maintaining visual quality. This is essential for streaming services, video storage, and broadcasting.

Key Concepts in Video Compression:

1. **Spatial and Temporal Redundancy:**
    - **Spatial Redundancy**: Nearby pixels often have similar colors (intra-frame compression).
    - **Temporal Redundancy**: Consecutive frames in a video are similar (inter-frame compression).
2. **Lossy vs Lossless Compression:**
    - **Lossy Compression**: Removes redundant data (e.g., H.264,

2. **Lossy vs Lossless Compression:**
   - **Lossy Compression**: Removes redundant data (e.g., H.264, H.265).
   - **Lossless Compression**: Keeps all details but results in large file sizes (e.g., FFV1, HuffYUV).
3. **Popular Video Compression Standards:**
   - **H.264 (AVC)**: Used in Blu-ray, YouTube, and video calls.
   - **H.265 (HEVC)**: More efficient than H.264, supports 4K/8K video.
   - **VP9**: Open-source codec developed by Google for YouTube.
   - **AV1**: Next-gen open-source codec for high efficiency.
4. **Key Techniques Used in Video Compression**
   - **Motion Compensation**: Predicts

for high efficiency.

4. **Key Techniques Used in Video Compression:**
   - **Motion Compensation**: Predicts frame changes to reduce redundancy.
   - **Transform Coding (DCT)**: Converts image blocks into frequency components.
   - **Entropy Coding**: Uses Huffman coding or arithmetic coding for efficient bit representation.
5. **Bitrate and Compression Efficiency:**
   - **CBR (Constant Bitrate)**: Maintains a fixed bitrate, useful for streaming.
   - **VBR (Variable Bitrate)**: Adjusts bitrate based on complexity, better for quality.

By understanding these principles, you