Chapter 3

Source Codes, Line Codes & Error Control

3.1 Primary Communication

- Information theory deals with mathematical representation and analysis of a communication system rather than physical sources and channels.
- It is based on the application of probability theory i.e.,) calculation of probability of error.

Discrete message

The information source is said to be discrete if it emits only one symbol from finite number of symbols or message.

- Let source generating 'N' number of set of alphabets.

$$X = \{x_1, x_2, x_3, ..., x_M\}$$

- The information source generates any one alphabet from the set. The probability of various symbols in X can be written as

$$P(X = x_k) = P_k \qquad k = 1, 2, ...M$$

$$\sum_{k=1}^{M} P_k = 1 \tag{1}$$

Discrete memoryless source (DMS)

It is defined as when the current output symbol depends only on the current input symbol and not an any of the previous symbols.

- Letter or symbol or character

  - Any individual member of the alphabet set.
- Message or word:
  - A finite sequence of letters of the alphabet set
- Length of a word
  - Number of letters in a word
- Encoding
  - Process of converting a word of finite set into another format of encoded word.
- Decoding
  - Inverse process of converting a given encoded word to a original format.

3.2 Block Diagram of Digital Communication System

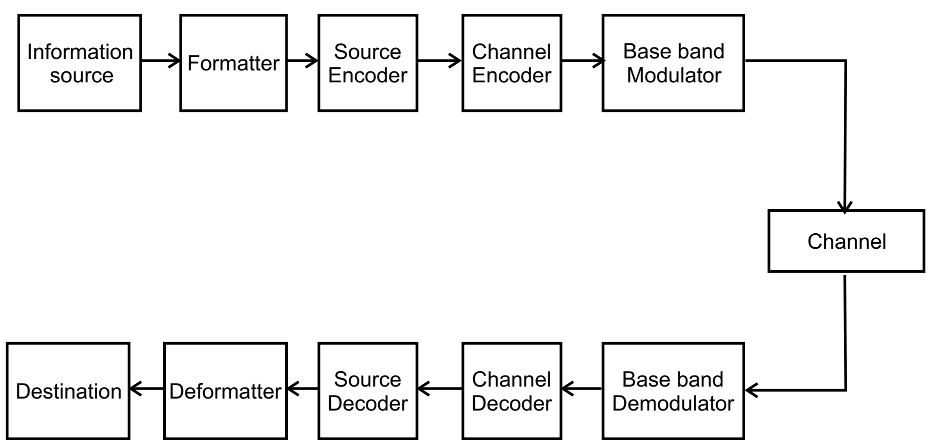


Fig .3.1 Typical communication system (or) Digital communication system

Information source

It may analog or digital.
Example: voice, video

Formetter

It converts analog signal into a digital signal.

Source encoder

- Is used efficient representation of data generated by source.
- It represents digital signal into a few digits as possible depending on the information content of message. (i.e.,) minimizes the requirements of digits.

Channel encoder

- Some redundancy is introduced in message to combine noise in channel.

Baseband modulator

- Encoded signal is modulated here by precise modulating techniques.

Channel

- Transmitted signal gets corrupted by random noise, thermal noise, shot noise, atmospheric noise.

Channel decoder

- It removes the redundancy bits by channel decoding algorithm.

Deformetter

It converts digital data into a discrete form or analog form.

- The above communication system is used to carry information bearing baseband signal from one place to another over a communication channel.
- Performance of communication symbol measured by probability of error (P_e)
- Condition to get error free communication is Entropy of the source $<$ capacity of a channel
- Capacity of a channel:

  The ability of a channel to convey information.
- Entropy:

  Average information per symbol.

3.3 Amount of Information

- Amount of information defined interms of probability i.e.,

$$I = f\left(\frac{1}{P_i}\right)$$

- Probability of occurrence of event is more, very less amount of information, otherwise probability of occurrence of an event is less then there will be more amount of information.
- Example:

  If a dog bites a man the probability of occurrence is more so loess information. Otherwise if a man bites a dog the probability of occurrence is less hence more information.

$$I(xj) = f\left[\frac{1}{P(xj)}\right] \tag{1}$$

X_j	$\rightarrow$	Event
$P(xj)$	$\rightarrow$	Probability of an event
$I(xj)$	$\rightarrow$	Amount of information

Equation (1) can be rewrite as

$$\boxed{I(xj) = \log\frac{1}{P(xj)}\text{bits or } \; I_k = \log\frac{1}{P_k} \text{ bits}} \tag{2}$$

Definition

The amount of information I_{xj}, is related to the logarithm on the inverse of the probability of occurrence of an event $P(xj)$.

3.4 Average Information or Entropy

Definition

The entropy of a source is defined as the source which produces average information per message or symbol in a particular interval.

Let m_1, m_2, m_3, ... m_k, 'k' different messages with p_1, p_2,, p_3, ... p_k, be corresponding probabilities of occurrences.

Example:
Message generated by source is

$$'ABCACBCABCAABC'$$
$$A, B, C \rightarrow m_1, m_2, m_3.... \qquad \therefore k = 3$$

Then the number of m_1 message is

$$m_1 = P_1 L$$

$L \rightarrow$ Total no. of messages generated by the source.

$$m_1 \rightarrow A, L = 15$$
$$m_1 = P_{115}$$

Similarly for

$$m_2 \rightarrow B, L = 15$$
$$m_2 = P_{215}$$

The amount of information in messages 'm_1' is given as,

$$I_1 = \log_2 \left(\frac{1}{P_1} \right)$$

Total amount of information due to m_1 message is

$$I_{t1} = P_1 L \log_2 \left(\frac{1}{P_1} \right)$$

similarly total amount of information due to 'm_2' message is,

$$I_{t2} = P_2 L \log_2 \left(\frac{1}{P_2} \right)$$

Thus the total amount of information due to 'L' messages, is given as

$$I_t = I_{t1} + I_{t2} + ... + I_{tk}$$
$$I_t = P_1 \log_2 \left(\frac{1}{P_1} \right) + P_2 \log_2 \left(\frac{1}{P_2} \right) + ...P_k \log_2 \left(\frac{1}{P_k} \right)$$

$$\therefore \quad \text{Average information} = \frac{\text{Total information}}{\text{Number of messages}}$$

$$\text{Average information} = \frac{I_t}{L}$$

Average information per message is nothing but entropy $H(x)$ or H

$$H = \frac{I_t}{L}$$

$$\therefore\ H(s) = \frac{P_1 L \log_2\left(\frac{1}{P_1}\right) + P_2 L \log_2\left(\frac{1}{P_2}\right) + ... + P_k L \log_2\left(\frac{1}{P_k}\right)}{L}$$

$$= \frac{L\left[P_1 L \log_2\left(\frac{1}{P_1}\right) + P_2 L \log_2\left(\frac{1}{P_2}\right) + ... + P_k L \log_2\left(\frac{1}{P_k}\right)\right]}{L}$$

$$H = \sum_{k=1}^{M} P_k \log_2\left(\frac{1}{P_k}\right) \text{ bits/symbol} \tag{3}$$

The entropy 'H' of discrete memoryless source is bounded as

$$0 \leq H \leq \log_2 M$$

3.4.1 Properties of entropy

Property 1: $H = 0$***, if*** $P_k = 0$ ***or 1***

Entropy is zero, when its probability of event is possible or not

When $P_k = 0$

$$H = \sum_{k=1}^{M} P_k \log_2\left(\frac{1}{P_k}\right)$$

$$= 0 \log_2\left(\frac{1}{0}\right)$$

$$H = 0$$

When $P_k = 1$

$$H = \sum_{k=1}^{M} P_k \log_2\left(\frac{1}{P_k}\right)$$

$$= 1 \log_2\left(\frac{1}{1}\right)$$

$$H = 1$$

Property 2: All the symbols are equi-probable

$$H = P_k \log_2 \frac{1}{P_k}$$

$$H = P_1 \log_2\left(\frac{1}{P_1}\right) + P_2 \log_2\left(\frac{1}{P_2}\right) + ... + P_M \log_2\left(\frac{1}{P_M}\right)$$

For a minimum number of equally likely messages probability is

$$P_1 = P_2 = P_3 \,....P_M = \frac{1}{M}$$
$$H = \frac{1}{M}\log_2(k) + \frac{1}{M}\log_2(M) + ... + \frac{1}{M}\log_2(M)$$
$$H = \frac{M}{M}\log_2(M)$$
$$H = \log_2(M)$$

Property 3: Upper bound on entropy $0 \leq H_{\max} \leq \log_2 k$

Consider any two probability distribution $(P_1, P_2....P_n)$ and $(q_1, q_2....q_n)$ are the alphabet $X = \{x_1, x_2, ...x_m\}$ of a DMS source.

Then

$$\sum_{k=1}^{M} P_k \log_2\left(\frac{q_k}{P_k}\right) = \sum_{k=1}^{M} \frac{P_k \log_2\left(\frac{q_k}{P_k}\right)}{\log_{10} 2} \qquad \left[\because \log_2 x = \frac{\log_{10} x}{\log_{10} 2}\right]$$

By a property of natural log,

$$\log x \leqslant x - 1; \quad x > 0$$
$$\sum_{k=1}^{M} P_k \frac{\log_2\left(\frac{q_k}{P_k}\right)}{\log_{10} 2} \leq \sum_{k=1}^{M} \frac{1}{\log_{10} 2} P_k \left(\frac{q_k}{P_k} - 1\right)$$
$$\leq \frac{1}{\log_{10} 2} \sum_{k=1}^{M} (q_k - P_k)$$
$$\leq \log_{10} 2 \left(\sum_{k=1}^{M} q_k - \sum_{k=1}^{M} p_k\right)$$

W.K.T

$$\sum_{k=1}^{M} P_k = \sum_{k=1}^{M} q_k = 1$$
$$\sum_{k=1}^{M} P_k \log_2\left(\frac{q_k}{P_k}\right) \leqslant 0$$

$$\sum_{k=1}^{M} P_k \log_2 q_k + \sum_{k=1}^{M} P_k \log_2 \frac{1}{P_k} \leqslant 0$$

$$\sum_{k=1}^{M} P_k \log_2 \frac{1}{P_k} \leqslant -\sum_{k=1}^{M} P_k \log_2 q_k$$

Sub $q_k = \dfrac{1}{m}$

$$\begin{aligned}
\sum_{k=1}^{M} P_k \log_2 \frac{1}{P_k} &\leq \sum_{k=1}^{M} P_k \log_2 \left(\frac{1}{q_k}\right) \\
&\leq \sum_{k=1}^{M} P_k \log_2 M \\
&\leq \log_2 M \sum_{k=1}^{M} P_k \\
\sum_{k=1}^{M} P_k \log_2 \frac{1}{P_k} &\leq \log_2 M \\
H &\leq \log_2 M
\end{aligned}$$

The entropy H holds all the equiprobable symbols.

3.4.2 Entropy of a binary memoryless source (BMS)

- Assume that the source is memoryless so that successive symbols emitted by the source are statistically independent.
- Consider symbol ‘0’ occurs with probability P_0 and symbol ‘1’ with probability $P_1 = 1 - P_0$.

  Entropy os BMS

$$\begin{aligned}
H &= \sum_{k=1}^{2} P_k \log_2 \frac{1}{P_k} \\
&= P_0 \log_2 \frac{1}{P_0} + (1 - P_0) \log_2 \left(\frac{1}{1 - P_0}\right) \\
H &= -P_0 \log_2 P_0 - (1 - P_0) \log (1 - P_0)
\end{aligned}$$

1. When $P_0 = 0$, $H = 0$

2. When $P_0 = 1$, $H = 0$
3. When $P_0 = P_1 = \dfrac{1}{2}$, (i.e.,) symbol 0 and 1 are equally probable then $H = 1$.

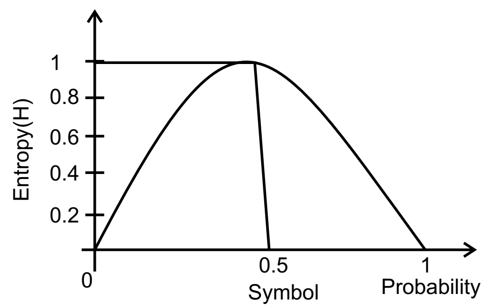


Fig .3.2 Plot of Entropy Vs Probability

3.4.3 Extension of a discrete memoryless source

- Consider a blocks with n-successive symbols rather than individual symbols. Each block is produced by an extended source alphabet (X^n) that has k^n distinct blocks, where $k =$ number of distinct symbols in the source alphabet (X) of original source.

  Extended entropy $\boxed{H\left(X^n\right) = nH\left(X\right)}$

3.4.4 Differential entropy

Consider continuous random variable 'X' having probability density function of $f_X(x)$.

$$\boxed{H = \int_{-\infty}^{\infty} f_X\left(x\right) \log_2 \frac{1}{f_X\left(x\right)} dx}$$

3.4.5 Information rate (R)

Rate of information (R) is defined as the average number of bits of information transmitted per second.

$$\boxed{R = rH \quad \text{bit/sec}}$$

The channel types are classified as,

1. Discrete Memoryless Channel (DMC)

2. Binary Communication Channel (BCC)
3. Binary Symmetric Channel (BSC)
4. Binary Erasable Channel (BEC)
5. Lossless Channel
6. Deterministic Channel

3.4.6 Discrete Memoryless Channel (BMC)

Discrete: Channel is said to be discrete when alphabets X have finite.

Memoryless: channel is said to be memoryless, when the current output symbols depends only on the current input symbol and not an any of the previous symbols.

- It is a statistical model with an input X and output Y that is a noisy version of X. Both X and Y are random variables.
- Let consider input alphabet $X = \{x_1, x_2, ..., x_j\}$ and output alphabet as $Y = \{y_1, y_2, ..., y_k\}$ and set of transition probability $P\left(y_k/x_j\right)$.
- The transmission matric or channel matrix with row as input and column as output.

$$X \begin{bmatrix} x_1 & & y_1 \\ x_2 & & y_2 \\ x_3 & X \leftarrow P\left(y_k/x_j\right) \rightarrow Y & y_3 \\ \vdots & & \vdots \\ x_j & & y_k \end{bmatrix} Y$$

- The transition matrix of order $j \times k$ can be written with the transition probability as,

$$P\left(y_k/x_j\right) = \begin{bmatrix} P\left(y_1/x_1\right) & P\left(y_2/x_1\right) & .... & P\left(y_k/x_1\right) \\ P\left(y_1/x_2\right) & P\left(y_2/x_2\right) & .... & P\left(y_k/x_2\right) \\ \vdots & \vdots & & \\ P\left(y_1/x_j\right) & P\left(y_2/x_j\right) & .... & P\left(y_k/x_j\right) \end{bmatrix}$$

- The sum of the elements along any row of the matric is equal to one

$$P\left(y_1/x_1\right) + P\left(y_2/x_1\right) + P\left(y_3/x_1\right) + ... + P\left(y_k/x_1\right) = 1$$

In general

$$\boxed{\sum_{k=1}^{M} P\left(y_k/x_j\right) = 1 \ \text{ for all } j}$$

W.K.T

$$P\left(x_j, y_k\right) = P\left(y_k/x_j\right) P\left(x_j\right) \quad \left[\because\ P\left(XY\right) = P\left(y/x\right) P\left(x\right)\right]$$

- Addition of all the joint probabilities for fixed y_k then,

$$P\left(yk\right) = \sum_{j=1}^{M} P\left(y_k/x_j\right) P\left(x_j\right)$$

3.4.7 Binary Communication Channel (BCC)

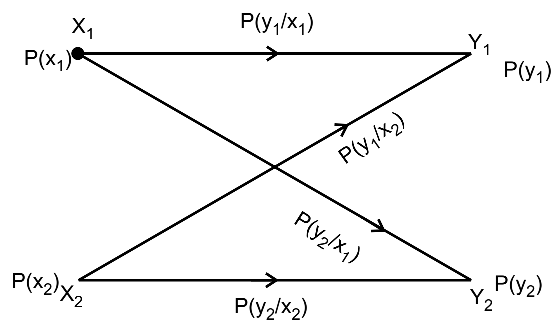


Fig .3.3 Binary communication channel

when only two symbols are transmitted through a discrete channel it is called as binary communication channel.

- Output probability $P(y_1)$ and $P(y_2)$ as

$$P\left(y_1\right) = P\left(y_1/x_1\right) P\left(x_1\right) + P\left(y_1/x_2\right) P\left(x_2\right)$$
$$P\left(y_2\right) = P\left(y_2/x_2\right) P\left(x_2\right) + P\left(y_2/x_1\right) P\left(x_1\right)$$

- Output probability in terms of matrix

$$\left[P\left(y_1\right) \cdot P\left(y_2\right)\right] = \left[P\left(x_1\right) P\left(x_2\right)\right] \begin{bmatrix} P\left(y_1/x_1\right) & P\left(y_2/x_1\right) \\ P\left(y_1/x_2\right) & P\left(y_1/x_2\right) \end{bmatrix}$$

3.4.8 Binary Symmetric Channel (BSC)

- BSC is discrete memoryless channel (DMC)
- It has two inputs (X_0, X_1)
- Two outputs (Y_0, Y_1) (i.e.,) noisy version of inputs
- Transition probability is conditional probability (i.e.,) $P\left(y_k/x_j\right)$

- correct reception is obtained when '$k = j$'
- If $k \neq j$ error will occur.
- channel matrix or probability transition matrix is

$$\text{channel matrix} = \begin{bmatrix} P\left(y_0/x_0\right) & P\left(y_1/x_0\right) \\ P\left(y_0/x_1\right) & P\left(y_2/x_1\right) \end{bmatrix}$$

where $P\left(y_1/x_0\right)$ & $P\left(y_0/x_1\right)$ is said to be conditional probability of error.

(i.e.,)

$$P_{10} = P\left(\frac{Y=1}{X=0}\right) = p$$
$$P_{01} = P\left(\frac{Y=0}{X=1}\right) = p$$

$P_{10} \rightarrow$ Probability of receiving '1' when '0' is sent.

- Similarly probability of receiving '0' when '1' is sent.
- Transition probability is explained by a diagram is said to be transition probability diagram.

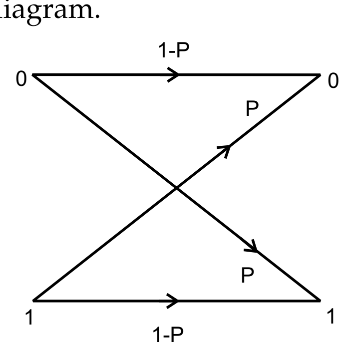


Fig .3.4 Binary symmetric channel

Channel - capacity of BSC

Channel capacity of a DM channel is defined as maximum mutual information $I(X,Y)$ in any single use of the channel.

$$C = \max I\left(X : Y\right) \text{ bits/channel}$$

3.4.9 Mutual information $[I\left(X;Y\right)]$

The difference between $H\left(X\right) - H\left(X/Y\right)$ is said to be mutual information

$$I\left(X,Y\right) = H\left(X\right) - H\left(X/Y\right)$$

or

$$I\left(X,Y\right)=H\left(X\right)-H\left(Y\right)$$

$H(X)$	$\rightarrow$	Entropy of channel input
$H(Y)$	$\rightarrow$	Entropy of channel output
$H(X/Y)$	$\rightarrow$	Conditional entropy of the channel input

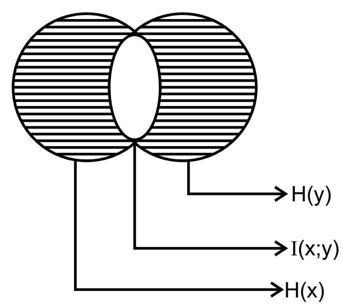


Fig .3.5 Mutual information

3.4.10 Binary Erasable Channel (BEC)

- It has two inputs (0, 1)
- It has three outputs (0, Y, 1)
- The symbol 'Y' indicates that due to noise, no definite decisions can be made as to whether the received symbol is a '0' or '1' (i.e.,) it indicates that the output is erased.
- Whenever the received symbol is y, the receiver requests the transmitter to retransmitter till the direction is taken in favour of either "0" or "1".

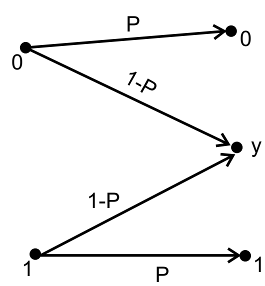


Fig .3.6 Transition probability diagram of BEC

Channel matrix

$$CM=\begin{bmatrix} P\left(Y_1/X_1\right) & P\left(Y_2/X_1\right) & P\left(Y_3/X_1\right) \\ P\left(Y_1/X_2\right) & P\left(Y_2/X_2\right) & P\left(Y_3/X_2\right) \end{bmatrix}$$

$$X_1 \to 0 \quad Y_1 = 0$$
$$X_2 \to 1 \quad Y_2 = 1$$

CM is simplified as

$$= \begin{bmatrix} p & q & 0 \\ 0 & q & p \end{bmatrix}$$

Mutual information of BEC is

$$\begin{aligned} I(X;Y) &= H(X) - H(X/Y) \\ &= H(X) - (1-P)\,H(X) \\ &= H(X) - H(X) + PH(X) \end{aligned}$$

$H(X/Y) \to$ conditional entropy of input.

3.4.11 Channel capacity

Channel capacity of DMC is defined as maximum mutual information $I(X:Y)$ in any single use of the channel.

$$\begin{aligned} C &= \max I(X;Y) \\ &= \max\,[PH(X)] \\ &= P \max H(X) \end{aligned}$$

$\because$ using property of entropy max

$$\max H(X) = \log_2 k$$

where $k = 2$

$$\therefore \quad \max H(X) = \log_2 2 = 1$$

3.4.11.1 Channel capacity of BSC

$$\begin{aligned} C &= \log_2 k - H(Y_k/X_j) \\ &= \log_2 2 - \sum_{j=1}^{2} P(y_k/x_j)\log_2\left(\frac{1}{P(Y_k/X_j)}\right) \qquad \because k = 2 \\ &= 1 - \sum_{j=1}^{2} -P(y_k/x_j)\log_2(P(Y_k/X_j)) \quad \because \log_2 2 = \frac{\log_{10} 2}{\log_{10} 2} = 1 \\ &= 1 - [P(Y_2/X_1)\log_2 P(Y_2/X_1) + P(Y_2/X_2)\log_2 P(Y_2/X_2)] \\ &= 1 - [-P\log_2 P - (1-P)\log_2(1-P)] \\ &= 1 - [-P\log_2 P - q\log_2 q] \\ C &= 1 - [H(q)] \end{aligned}$$

Channel capacity Vs transition probability p

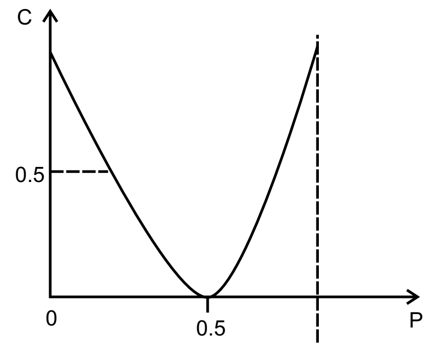


Fig .3.7 Channel capacity Vs transition probability

- C obtains maximum value when conditional probability is $P = 0$ or 1
- C attains maximum value i.e., 0 when conditional probability is $P = 0.5$.

3.5 Source Coding

Definition

- The efficient representation of data generated by a discrete source is called as **source encoding**. The device that perform this representation of data is called as **source encoder**.
- For efficient coding the generation of source code by assigning short code words to frequent source and long code word to rare source. This type of code is called as **variable length code.**
- Efficient source encoder satisfies the two functional requirements.
  - code word produced by the encoder are in binary form.
  - code word should be uniquely decoded so that the original source sequence can be reconstructed perfectly.
- Consider a discrete memoryless source that emits a symbol S_k is converted by a source encoder into block of 0′s and 1′s denoted by b_k.

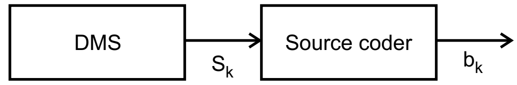


Fig .3.8 Source Encoder

- We assume that source has an alphabet with k-different symbols and the probability is represents as P_k. where $k = 1, 2, 3, ...$M.
- Let binary code assigned to each symbol S_k by the encoder have a length l_k measured in bits.

Code length **($\overline{L}$)**

- Average code word length τ of the source encoder is

$$\overline{L} = \sum_{k=1}^{M} P_k l_k \quad \text{bits/symbol} \tag{1}$$

Efficiency **(η)**

- Efficiency of source encoder is

$$\eta = \frac{L_{\min}}{\overline{L}}$$

Where $L_{\min}$ = minimum possible value of $\overline{L}$.

Source coder is said to be efficient when $\eta \leq 1$ with $\tau \geq L_{\min}$.

Entropy 'H' represents a fundamental limit on the average number of bits per symbol necessary to represent a discrete memoryless source, so that it can be made as small as possible.

$$\therefore \quad L_{\min} = H$$

- Efficiency of source encoder can be written as,

$$\eta = \frac{H}{\overline{L}} \tag{2}$$

Code redundancy **(r)**

$$r = 1 - \eta \tag{3}$$

Redundancy should be as low as possible.

Code variance **(σ^2)**

Variance of the code

$$\sigma^2 = \sum_{k=1}^{M} P_k \left(l_k - \overline{L}\right) \tag{4}$$

Variance should be as low as possible.

where

σ^2 → variance of the code
M → number of symbols
P_k → probability of k^{th} symbol
l_k → number of bits in the source encoder
$\overline{L}$ → average code word length

3.6 Data Compaction (or) Compression

In order to transmit a signal efficiently the redundant information should be removed from the signal prior to transmission. This operation with no loss of information referred as data compaction (or) compression.

3.7 Types of Source Coding (or) Entropy Coding

1. Shannon-Fano coding
2. Huffman coding
3. Lempel-Ziv coding
4. Prefix coding

3.7.1 Shannon-Fano coding

Coding the message with different number of bits, according to their probability.

Definition

Shannon Fano algorithm is one where less number of bits are used for highly probable messages and more number of bits for rare occurrence.

Algorithm (or) procedure

Step 1: List the source symbols or message in the descending (decreasing) order of their probability.

Step 2: Partition the set into two sets that are equi-probable or close as equiprobable.

Assume 0 → to upper partition set

1 → lower partition set

Step 3: Those partitions are further sub-divided into new partitioning sets are worth nearly equi-probabilities and further partitioning is not possible.

- Partitioning process is stopped when single message in a set was found.

Step 4: Calculate code length ($\overline{L}$), Entropy (H), Efficiency (η), code redundancy (r) and code variance (σ^2).

Note: Addition (sum) of all the symbol probabilities is equal to 1

$$\sum(\text{symbol probability}) = 1$$

3.7.2 Example Problems

Type 1

Solved Problem 3.1 *The discrete memoryless source has 6-symbols 0.25, 0.2, 0.12, 0.08, 0.3, 0.05. Construct Shannon Fano code.*

Given data

$x_1 = 0.25$, $x_2 = 0.2$, $x_3 = 0.12$, $x_4 = 0.08$, $x_5 = 0.3$,, $x_6 = 0.05$,

Solution:

Arrange the symbol probabilities in decreasing order

Symbol	Probability
x_1	0.3
x_2	0.25
x_3	0.2
x_4	0.12
x_5	0.08
x_6	0.05

Partition the set into two sets of equi-probable allot 0 for upper and 1 for lower partition.

Partitioning process stops, when single message in a set found.

Symbol (k)	Probability(P_k)	Stage 1	Stage 2	Stage 3	Stage 4	Code word	No. of bits (or) Length (l_k)
x_1	**0.3**] 0.55	**0**] 0.3	**0**	-	-	**00**	**2**
x_2	**0.25**	**0**] 0.25	**1**	-	-	**01**	**2**
x_3	**0.2**] 0.45	**1**] 0.2	**0**	-	-	**10**	**2**
x_4	**0.12**	**1**] 0.25	**1**] 0.12	**0**	-	**110**	**3**
x_5	**0.08**	**1**	**1**] 0.13	**1**] 0.08	**0**	**1110**	**4**
x_6	**0.05**	**1**	**1**	**1**] 0.05	**1**	**1111**	**4**

Calculation:

1. To find entropy (H)

$$H = \sum_{k=1}^{M} P_k \log_2 \frac{1}{P_k} \text{ bits/symbol}$$

or

$$H = -\sum_{k=1}^{M} P_k \log_2 P_k \text{ bits/symbol}$$

Given message symbol $(M) = 6$

$$H = -\sum_{k=1}^{6} P_k \log_2 P_k \qquad \left(\because \log_2 x = \frac{\log_{10} x}{\log_{10} 2}\right)$$

$$H = -\left[\frac{0.3 \log 0.3 + 0.25 \log 0.25 + 0.2 \log 0.2 + 0.12 \log 0.12 + 0.08 \log 0.08 + 0.05 \log 0.05}{\log 2}\right]$$

$$H = -\left[\frac{-0.156 - 0.15 - 0.139 - 0.11 - 0.087 - 0.065}{0.301}\right]$$

$H = 2.34$ bits/symbol

2. To find average codeword length ($\overline{L}$)

$$\overline{L} = \sum_{k=1}^{M} P_k l_k \qquad (\therefore M = 6)$$

$$\overline{L} = \sum_{k=1}^{6} P_k l_k$$

$$= 0.3\,(2) + 0.25\,(2) + 0.2\,(2) + 0.12\,(3) + 0.08\,(4) + 0.05\,(4)$$

$$= 0.6 + 0.5 + 0.4 + 0.36 + 0.32 + 0.2$$

$$\overline{L} = 2.38 \text{ bits/symbol}$$

3. To find efficiency (η)

$$\eta = \frac{H}{\overline{L}}$$

$$= \frac{2.34}{2.38} = 0.99$$

$$\%\eta = 99\%$$

4. To find redundancy (r)

$$r = 1 - \eta$$
$$r = 1 - 0.99$$
$$r = 0.01$$

5. To find variance (σ^2)

$$\sigma^2 = \sum_{k=1}^{M} P_k \left(l_k - \overline{L}\right)$$
$$= 0.3\,(2 - 2.38) + 0.25\,(2 - 2.38) + 0.2\,(2 - 2.38) + 0.12\,(3 - 2.38) + 0.08\,(4 - 2.38) + 0.05\,(4 - 2.38)$$
$$= -0.114 - 0.095 - 0.076 + 0.0744 + 0.1296 + 0.081$$
$$\sigma^2 = 0$$

Type 2: If two possible partitions are having same value then the problems can be analyze in two methods

Solved Problem 3.2 *The discrete memoryless source with* $X_1 = 0.4$, $X_2 = 0.12$, $X_3 = 0.08$, $X_4 = 0.04$, $X_5 = 0.2$, $X_6 = 0.08$, $X_6 = 0.08$. *Construct Shannon - Fano coding.*

Method 1: Upper partition high vale & lower partition low value.

Solution:

Arrange the symbol probabilities in decreasing order.

Symbol (k)	Probability (P_k)	Stage 1	Stage 2	Stage 3	Stage 4	Code word	No. of bits (or) Length (l_k)
x_1	0.4 (0.6)	0 (0.4)	0	-	-	00	2
x_2	0.2	0 (0.2)	1	-	-	01	2
x_3	0.12 (0.4)	1 (0.2)	0 (0.12)	0	-	100	3
x_4	0.08	1	0 (0.08)	1	-	101	3
x_5	0.08	1 (0.2)	1 (0.08)	0 (0.08)	-	110	3
x_6	0.08	1	1 (0.12)	1 (0.08)	1	1110	4
x_7	0.04	1	1	1 (0.04)	1	1111	4

Calculation:

1. To find average codeword length ($\overline{L}$)

$$\tau = \sum_{k=1}^{M} P_k l_k \qquad (\therefore M = 7)$$

$$\tau = \sum_{k=1}^{7} P_k l_k$$
$$= 0.4\,(2) + 0.2\,(2) + 0.12\,(3) + 0.08\,(3) + 0.08\,(3) + 0.08\,(4) + 0.04\,(4)$$
$$= 0.8 + 0.4 + 0.36 + 0.24 + 0.24 + 0.32 + 0.16$$
$$\tau = 2.52 \text{ bits/symbol}$$

2. To find entropy (H)

$$H = -\sum_{k=1}^{7} P_k \log_2 P_k$$
$$H = -\left[\frac{0.4\log 0.4 + 0.2\log 0.2 + 0.12\log 0.12 + 3\times 0.08\log 0.08 + 0.04\log 0.04}{\log 2}\right]$$
$$H = -\left[\frac{0.159 + 0.139 + 0.110 + 0.2632 + 0.055}{0.301}\right]$$
$$H = 2.42 \text{ bits/symbol}$$

3. To find efficiency (η)

$$\eta = \frac{H}{\overline{L}}$$
$$= \frac{2.42}{2.52} = 0.9585$$
$$\%\eta = 95.85\%$$

4. To find redundancy (r)

$$r = 1 - \eta$$
$$r = 1 - 0.958$$
$$r = 0.04$$

5. To find variance (σ^2)

$$\sigma^2 = \sum_{k=1}^{M} P_k \left(l_k - \overline{L}\right)$$
$$= 0.4\,(2-0.95) + 0.2\,(2-0.95) + 0.12\,(3-0.95) + 0.08\,(3-0.95)$$
$$+ 0.08\,(3-0.95) + 0.08\,(4-0.95) + 0.04\,(4-0.95)$$
$$= 0.42 + 0.21 + 0.246 + 0.164 + 0.164 + 0.244 + 0.122$$
$$\sigma^2 = 1.57$$

Method 2: Upper partition as low value & lower partition as high value.

Solution:

Symbol (k)	Probability (P_k)		Stage 1		Stage 2	Stage 3	Stage 4	Code word	No. of bits (or) Length (l_k)
x_1	0.4	0.4	0		-	-	-	0	1
x_2	0.2	0.6	1	0.32	0] 0.12	0	-	100	3
x_3	0.12		1		0] 0.08	1	-	101	3
x_4	0.08		1	0.28	1] 0.16	0] 0.08	0	1100	4
x_5	0.08		1		1	0] 0.08	1	1101	4
x_6	0.08		1		1] 0.12	1] 0.08	0	1110	4
x_7	0.04		1		1	1] 0.04	1	1111	4

Calculation:

1. Average codeword length ($\overline{L}$)

$$\tau = \sum_{k=1}^{7} P_k l_k$$
$$= 0.4\,(1) + 0.2\,(3) + 0.12\,(3) + 3 \times (0.08 \times 4) + 0.04\,(4)$$
$$= 0.4 + 0.6 + 0.36 + 0.96 + 0.16$$
$$\tau = 2.48 \text{ bits/symbol}$$

2. Entropy (H)

$$H = -\sum_{k=1}^{7} P_k \log_2 P_k$$
$$H = -\left[\frac{0.4\log 0.4 + 0.2\log 0.2 + 0.12\log 0.12 + (0.08\log 0.08)\times 3 + 0.04\log 0.04}{\log 2}\right]$$
$$H = 2.42 \text{ bits/symbol}$$

3. To find efficiency (η)

$$\eta = \frac{H}{\overline{L}}$$
$$= \frac{2.42}{2.48} = 0.975$$
$$\%\eta = 97.58\%$$

4. To find redundancy (r)

$$r = 1 - \eta$$
$$r = 1 - 0.975$$
$$r = 0.02$$

Conclusion: From the analysis of two possible ways of partition. The method-2 (upper partition low value and lower partition high value) has high efficiency than the method-1. So second method is better.

Solved Problem 3.3 *Source which generates symbols x_1, x_2, x_3, x_4 with probabilities 1/8, 1/2, 1/4 and 1/8 respectively and determine the coding efficiency.*

Solution:

$$x_1 = \frac{1}{8} = 0.125$$
$$x_2 = \frac{1}{2} = 0.5$$
$$x_3 = \frac{1}{4} = 0.25$$
$$x_4 = \frac{1}{8} = 0.125$$

Symbol (x)	Probability (P_k)	Stage 1	Stage 2	Stage 3	Code word	Length (l_k)
x_1	0.5	0	-	-	0	1
x_2	0.25	1	0	-	10	2
x_3	0.125	1	1	0	110	3
x_4	0.125	1	1	1	111	3

1. Average codeword length ($\overline{L}$)

$$\overline{L} = \sum_{k=1}^{4} P_k l_k$$
$$= 0.5\,(1) + 0.25\,(2) + 2 \times (0.125 \times 3)$$
$$= 0.5 + 0.5 + 0.75$$
$$\overline{L} = 1.75 \text{ bits/symbol}$$

2. Entropy (H)

$$H = -\sum_{k=1}^{4} P_k \log_2 P_k$$
$$H = -\left[\frac{0.5\log 0.5 + 0.25\log 0.25 + 2\,(0.125\log 0.125)}{\log 2}\right]$$

$$= \frac{0.150 + 0.150 + 0.225}{0.301}$$
$$H = 1.744 \text{ bits/symbol}$$

3. To code efficiency (η)

$$\eta = \frac{H}{\overline{L}}$$
$$= \frac{1.744}{1.75} = 0.99$$
$$\%\eta = 99\%$$

The code efficiency for a given source symbol is 99%.

Solved Problem 3.4 *For DMS 'X' with two symbols X_1 and X_2 as $P(X_1) = 0.9$ and $P(X_2) = 0.1$. find second order extension. Find efficiency & redundancy of extended code.*

Solution:

Extended Entropy $\quad H\left(X^n\right) = nH\left(X\right)$

where n = order of extension

- Entropy

$$H\left(x\right) = -\sum_{i=1}^{M} P\left(X\right) \log_2 P\left(X\right)$$
$$= -\sum_{i=1}^{2} P\left(X\right) \log_2 P\left(X\right)$$
$$= -\left[\frac{0.9 \log_2 0.9 + 0.1 \log_2 0.1}{\log_2}\right]$$
$$H\left(X\right) = 0.469 \text{ bits/symbol}$$

- Here order of extension ($n = 2$)

$$H\left(X^2\right) = 2 \times H\left(x\right)$$
$$= 2 \times 0.469$$
$$H\left(X^2\right) = 0.9378 \text{ bits/symbol}$$

Symbol	P_k	**Code word**	l_k
x_1	0.9	0	1
x_2	0.1	1	1

$$\overline{L} = \sum_{k=1}^{2} P_k l_k$$
$$= 0.9 \times 1 + 0.1 \times 1$$
$$\overline{L} = 1.0 \text{ bits/symbol}$$

(i) Efficiency (η)

$$\eta = \frac{H\left(x^2\right)}{\overline{L}}$$
$$= \frac{0.9378}{1}$$
$$\%\eta = 93.78\%$$

(ii) Redundancy (r)

$$r = 1 - \eta$$
$$r = 1 - 0.9378$$
$$r = 0.0622$$

3.7.3 Huffman coding

Definition

It is a coding process used to maximize the efficiency (η) by assigning different number of binary digits to the messages according to their probability of occurrence.

It is also called as efficient coding or minimum redundancy code.

Procedure (or) Algorithm

Step 1: Arrange the source symbols or message in descending (decreasing) order of their probabilities in first stage.

Step 2: Add lost two symbols into one symbol by assigning binary '0' and '1'. Rearrange the symbols in descending order of probability again as in second stage.

Step 3: Again add two symbols into one symbol by assigning binary '0' and '1'. The sum probability in the second stage is placed in the third stage after arranging the symbols in descending order.

Step 4: This process is repeated till no further repetition required.

Step 5: (binary digits) can be obtained by tracing the path for each single symbols.

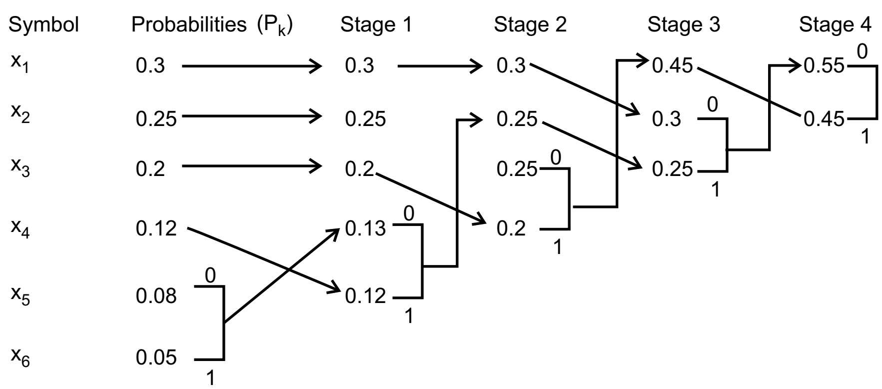
Step 6: Code word is obtained by reversing the trace path binary digits as LSB to MSB.

Step 7: Calculate $\overline{L}$, H, η, r, and σ^2.

- Two methods of constructing Huffmann code:
    1. **As high as possible:** When added symbol and existing symbol are same probability then added symbol exist in higher position.
    2. **As low as possible:** When added symbol and existing symbol are same probability then added symbol exist in lower position.

Type 1: (As high as possible)

Solved Problem 3.5 *A source transmits a messages having probabilities 0.05, 0.08, 0.25, 0.2, 0.12 and 0.3. Construct Huffman code.*



Symbol (x)	**Probability** (P_k)	**Trace path**	**code word**	**length** (l_k)
x_1	0.3	00	00	2
x_2	0.25	01	10	2
x_3	0.2	11	11	2
x_4	0.12	110	011	3
x_5	0.08	0010	0100	4
x_6	0.05	1010	0101	4

1. Average codeword length ($\overline{L}$)

$$\overline{L} = \sum_{k=1}^{6} P_k l_k$$
$$= 0.3\,(2) + 0.25\,(2) + 0.2\,(2) + 0.12\,(3) + 0.08\,(4) + 0.05\,(4)$$
$$= 0.6 + 0.5 + 0.4 + 0.36 + 0.32 + 0.2$$
$$\overline{L} = 1.75 \text{ bits/symbol}$$

2. Entropy (H)

$$H = -\sum_{k=1}^{6} P_k \log_2 P_k$$
$$H = -\left[\frac{0.3\log 0.3 + 0.25\log 0.25 + 0.2\log 0.2 + 0.12\log 0.12 + 0.08\log 0.08 + 0.05\log 0.05}{\log 2}\right]$$
$$= \frac{0.156 + 0.150 + 0.139 + 0.110 + 0.087 + 0.065}{0.301}$$
$$H = 2.34 \text{ bits/symbol}$$

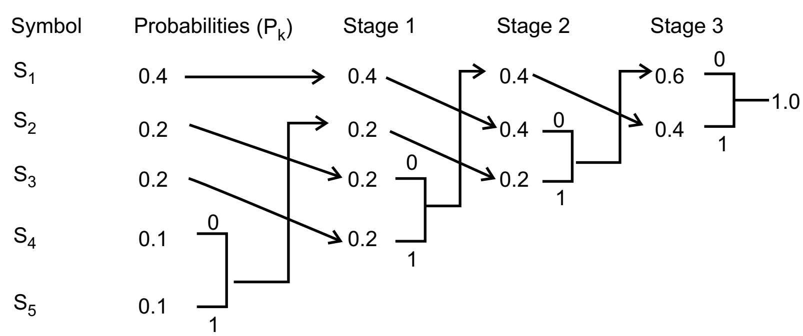
3. To code efficiency (η)

$$\eta = \frac{H}{\overline{L}} = 0.983$$
$$\%\eta = 98.31\%$$

4. Redundancy (r):

$$r = 1 - \eta = 0.017$$

Solved Problem 3.6 *A discrete memoryless source has 5 symbols S_1, S_2, S_3, S_4, S_5 with probabilities 0.1, 0.2, 0.4, 0.2, 0.1 respectively. Construct a Huffman code and calculate its efficiency (η).*



Symbol S	**Probability (**P_k**)**	**Trace path**	**code word**	**length (**l_k**)**
S_1	0.4	00	00	2
S_2	0.2	01	10	2
S_3	0.2	11	11	2
S_4	0.1	010	010	3
S_5	0.1	110	011	3

1. Average codeword length ($\overline{L}$)

$$\begin{aligned}\overline{L} &= \sum_{k=1}^{5} P_k l_k \\ &= 0.4\,(2) + 0.2\,(2) + 0.2\,(2) + 0.1\,(3) + 0.1\,(3) \\ &= 0.8 + 0.4 + 0.4 + 0.3 + 0.3 \\ \overline{L} &= 2.2 \text{ bits/symbol}\end{aligned}$$

2. Entropy (H)

$$\begin{aligned}H &= -\sum_{k=1}^{5} P_k \log_2 P_k \\ H &= -\left[\frac{0.4 \log 0.4 + 0.2 \log 0.2 + (0.1 \log 0.1) \times 2}{\log 2}\right] \\ &= \frac{0.159 + 0.279 + 0.2}{0.301} \\ H &= 2.11 \text{ bits/symbol}\end{aligned}$$

3. To code efficiency (η)

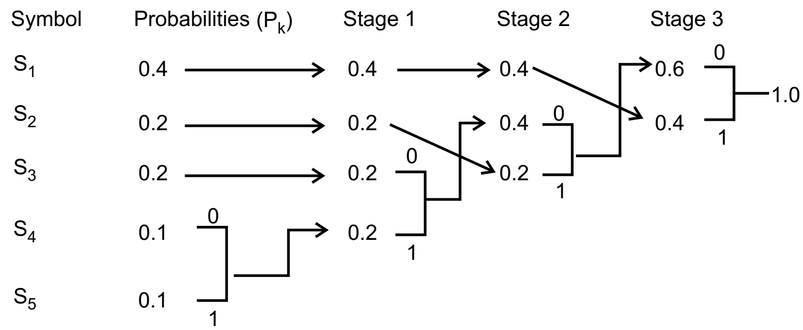
$$\begin{aligned}\eta &= \frac{H}{\overline{L}} \\ &= \frac{2.11}{2.2} = 0.9590 \\ \%\eta &= 95.9\%\end{aligned}$$

4. Redundancy (r):

$$\begin{aligned}r &= 1 - \eta \\ r &= 1 - 0.9590 \\ r &= 0.040\end{aligned}$$

Method 2: [As low as possible]

Refer problem (2) [same problem can be solved by method - 2].



Symbol S	**Probability (**P_k**)**	**Trace path**	**code word**	**length (**l_k**)**
S_1	0.4	1	1	1
S_2	0.2	10	01	2
S_3	0.2	000	000	3
S_4	0.1	0100	0010	4
S_5	0.1	1100	0011	4

1. Average codeword length ($\overline{L}$)

$$\begin{aligned}\overline{L} &= \sum_{k=1}^{5} P_k l_k \\ &= 0.4\,(1) + 0.2\,(2) + 0.2\,(3) + 0.1\,(4) + 0.1\,(4) \\ &= 0.4 + 0.4 + 0.6 + 0.4 + 0.4 \\ \overline{L} &= 2.2 \text{ bits/symbol}\end{aligned}$$

2. Entropy (H)

$$\begin{aligned}H &= -\sum_{k=1}^{5} P_k \log_2 P_k \\ H &= -\left[\frac{0.4\log 0.4 + 0.2\log 0.2 + (0.1\log 0.1)\times 2}{\log 2}\right] \\ &= \frac{0.159 + 0.279 + 0.2}{0.301} \\ H &= 2.11 \text{ bits/symbol}\end{aligned}$$

3. To code efficiency (η)

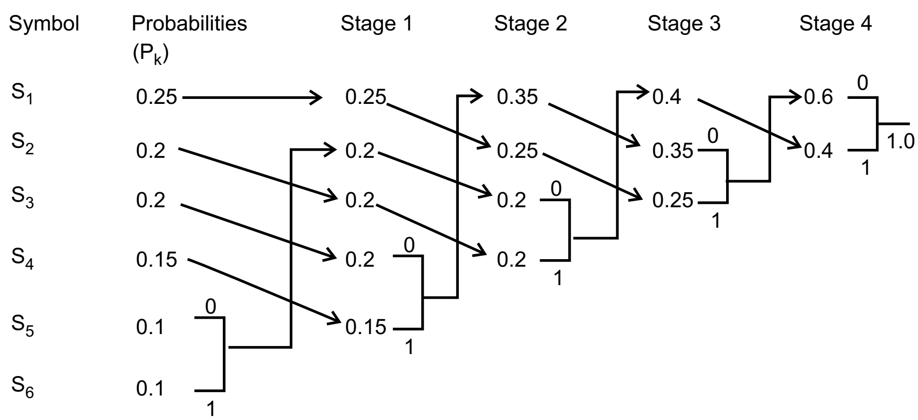
$$\eta = \frac{H}{\overline{L}}$$
$$= \frac{2.11}{2.2} = 0.9590$$
$$\%\eta = 95.9\%$$

4. Redundancy (r):

$$r = 1 - \eta$$
$$r = 0.040$$

Solved Problem 3.7 *Six symbols of the alphabet of discrete memory less source and their probabilities are given below.* $S = \{S_0, S_1, S_2, S_3, S_4, S_5\}$, $P(S) = \{0.1, 0.1, 0.2, 0.2, 0.25, 0.15\}$. *Code the symbols using Huffman code and Shannon Fano coding and compare the efficiency.*

Solution by Huffman coding



Symbol S	Probability (P_k)	Trace path	code word	length (l_k)
S_0	0.25	10	01	2
S_1	0.2	11	11	2
S_2	0.2	000	000	3
S_3	0.15	100	001	3
S_4	0.1	001	100	3
S_5	0.1	101	101	3

1. Average codeword length ($\overline{L}$)

$$\begin{aligned}
\overline{L} &= \sum_{k=1}^{6} P_k l_k \\
&= 0.25\,(2) + 0.2\,(2) + 0.2\,(3) + 0.15\,(3) + [0.1\,(3)]\,2 \\
&= 0.5 + 0.4 + 0.6 + 0.45 + 0.6 \\
\overline{L} &= 2.55 \text{ bits/symbol}
\end{aligned}$$

2. Entropy (H)

$$\begin{aligned}
H &= -\sum_{k=1}^{6} P_k \log_2 P_k \\
H &= -\left[\frac{0.25\log 0.25 + 0.2\log 0.2 + 0.15\log 0.15 + [0.1\log 0.1]\,2}{\log 2}\right] \\
&= \frac{0.1505 + 0.279 + 0.123 + 0.2}{0.301} \\
H &= 2.5 \text{ bits/symbol}
\end{aligned}$$

3. To code efficiency (η)

$$\begin{aligned}
\eta &= \frac{H}{\overline{L}} = 0.908 \\
\%\eta &= 98.03\%
\end{aligned}$$

Solution by Shannon-Fano coding

Symbol (S)	Probabilities (P_k)	Stage 1	Stage 2	Stage 3	Stage 4	Code word	length (l_k)
S_1	0.25 (0.45)	0 (0.25)	0	–		00	2
S_2	0.2	0 (0.2)	1	–		01	2
S_3	0.2	1 (0.2)	0	–		10	2
S_4	0.15	1	1 (0.15)	0		110	3
S_5	0.1 (0.55)	1 (0.35)	1	1	0	1110	4
S_6	0.1	1	1 (0.2)	1	1	1111	4

1. Average codeword length ($\overline{L}$)

$$\begin{aligned}\overline{L} &= \sum_{k=1}^{5} P_k l_k \\ &= 0.25\,(2) + 0.2\,(2) + 0.2\,(2) + 0.15\,(3) + 0.1\,(4) + 0.1\,(4) \\ &= 0.5 + 0.4 + 0.4 + 0.45 + 0.4 + 0.4 \\ \overline{L} &= 2.55 \text{ bits/symbol}\end{aligned}$$

2. Entropy (H)

$$H = 2.5 \text{ bits/symbol}$$

3. To code efficiency (η)

$$\eta = \frac{H}{\overline{L}}$$
$$\%\eta = 98.03\%$$

Conclusion: Efficiency of Huffman code & Shannon Fano code is same (i.e.,) $\eta = 98.03\%$.

Solved Problem 3.8 *Construct Shannon-Fano code & Huffman code and compare the efficiency and redundancy for the following symbols and probabilities* $x_1 = 0.15$, $x_2 = 0.15$, $x_3 = 0.4$, $x_4 = 0.15$, $x_5 = 0.15$.

Solution by Shannon-Fano coding

Symbol (x)	Probabilities (P_k)	Stage 1	Stage 2	Stage 3	Code word	length (l_k)
x_1	0.4 (0.55)	0 (0.4)	0	–	00	2
x_2	0.15	0 (0.15)	1	–	01	2
x_3	0.15 (0.45)	1 (0.3)	0	0	100	3
x_4	0.15	1	1	1	101	3
x_5	0.15	1 (0.15)	1	_	111	2

1. Average codeword length ($\overline{L}$)

$$\overline{L} = \sum_{k=1}^{5} P_k l_k$$
$$= 0.4\,(2) + 0.15\,(2) + 0.15\,(3) + 0.15\,(3) + 0.15\,(2)$$
$$\overline{L} = 2.3 \text{ bits/symbol}$$

2. Entropy (H)

$$H = -\sum_{k=1}^{5} P_k \log_2 P_k$$
$$H = 2.16 \text{ bits/symbol}$$

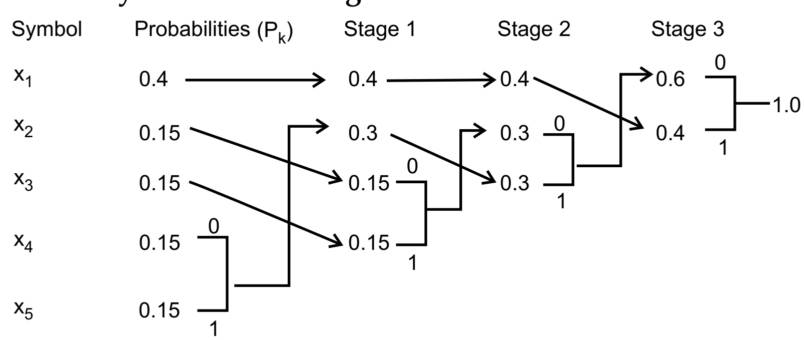
3. To code efficiency (η)

$$\eta = \frac{H}{\overline{L}} = 0.939$$
$$\eta = 93.9\%$$

4. Redundancy (r):

$$r = 1 - \eta$$
$$r = 0.06$$

Solution by Huffman coding



Symbol x	**Probability (P_k)**	**Trace path**	**code word**	**length (l_k)**
x_1	0.4	1	1	1
x_2	0.15	000	000	3
x_3	0.15	100	001	3
x_4	0.15	010	010	3
x_5	0.15	110	110	3

1. Average codeword length ($\overline{L}$)

$$\begin{aligned}\overline{L} &= \sum_{k=1}^{5} P_k l_k \\ &= 0.4\,(1) + [(0.15 \times 3)]\,4 \\ \overline{L} &= 2.2 \text{ bits/symbol}\end{aligned}$$

2. Entropy (H)

$$\begin{aligned}H &= -\sum_{k=1}^{5} P_k \log_2 P_k \\ H &= -\left[\frac{0.4\log 0.4 + (0.15\log 0.15)\,4}{\log 2}\right] \\ H &= 2.16 \text{ bits/symbol}\end{aligned}$$

3. To code efficiency (η)

$$\begin{aligned}\eta &= \frac{H}{\overline{L}} \\ &= \frac{2.16}{2.2} \\ \%\eta &= 98.18\%\end{aligned}$$

4. Redundancy (r):

$$\begin{aligned}r &= 1 - \eta \\ r &= 0.018\end{aligned}$$

	Characteristics	**Shannon-Fano Coding**	**Huffman Coding**
1)	Entropy (H)	2.16 bits/symbol	2.16 bits/symbol
2)	Average code length (L)	2.3 bits/symbol	2.2 bits/symbol
3)	Efficiency (η)	93.9%	98.18%
4)	Redundancy (r)	0.06	0.018

Conclusion: By comparing Huffman coding and Shannon-Fano coding the efficiency of Huffman is more and redundancy is less.

Drawbacks of Huffman coding and Shannon coding:

1. It requires the knowledge of probability of source model. In practice source statistics are not always know prior.
2. Reduced efficiency.

To overcome there practical difficulties, Lempel Ziv encoding is used.

3.7.4 Lempel-Ziv coding

Algorithm

It is a process of passing the source data stream into segments that are the shortest subsequences not encountered previously.

- It does not requires a prior probabilities of the data sequence.
- It assigns fixed length odes for variable number of source symbols.
- For English text, LZ coding achieves approximately 55% compression against 43% of Huffman coding.
- This algorithm is adaptive and gives higher coding efficiency for long data sequence.

Application

Lempel-ziv coding is used for file compression.

3.8 Noiseless Coding Theorem

- In order to achieve a high performance of coding techniques in communication systems, the noise level in the information should be reduce.
- The presence of noise in the channel causes some errors between input and output of a communication system. The probability of error may have a value as high as 10^{-2}.
- Shannon's Information theory for discrete memoryless sources, channels and information capacity are explained below,

  1. Shannon's first theorem (or) source coding theorem
  2. Shannon's second theorem (or) channel coding theorem (or) channel capacity (or) Shannon's theorem.
  3. Shannon's third theorem (or) information capacity theorem (or) Shannon's Hartley theorem.

3.8.1 Shannon's first theorem (or) source coding theorem

Definition

- The efficient representation of data generated by a discrete source is called as **source encoding**. The device that perform this representation of data is called as **source encoder**.

- For efficient coding the generation of source code by assigning short code words to frequent source and long code word to rare source. This type of code is called as **variable length code.**
- Efficient source encoder satisfies the two functional requirements.
  - code word produced by the encoder are in binary form.
  - code word should be uniquely decoded so that the original source sequence can be reconstructed perfectly.
- Consider a discrete memoryless source that emits a symbol S_k is converted by a source encoder into block of 0's and 1's denoted by b_k.

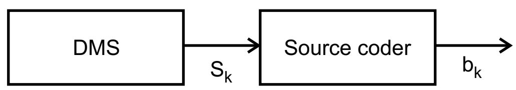


Fig .3.9 Source Encoder

- We assume that source has an alphabet with k-different symbols and the probability is represents as P_k. where $k = 1, 2, 3, ...$M.
- Let binary code assigned to each symbol S_k by the encoder have a length l_k measured in bits.

Code length ($\overline{L}$)

- Average code word length τ of the source encoder is

$$\overline{L} = \sum_{k=1}^{M} P_k l_k \quad \text{bits/symbol} \tag{1}$$

Efficiency (η)

- Efficiency of source encoder is

$$\eta = \frac{L_{\min}}{\overline{L}}$$

Where $L_{\min}$ = minimum possible value of $\overline{L}$.

Source coder is said to be efficient when $\eta \leq 1$ with $\tau \geq L_{\min}$.

Entropy 'H' represents a fundamental limit on the average number of bits per symbol necessary to represent a discrete memoryless source, so that it can be made as small as possible.

$$\therefore \quad L_{\min} = H$$

- Efficiency of source encoder can be written as,

$$\eta = \frac{H}{\overline{L}} \tag{2}$$

Code redundancy r

$$r = 1 - \eta \tag{3}$$

Redundancy should be as low as possible.

Code variance (σ^2)

Variance of the code

$$\sigma^2 = \sum_{k=1}^{M} P_k \left(l_k - \overline{L}\right) \tag{4}$$

Variance should be as low as possible.

where

σ^2 → variance of the code
M → number of symbols
P_k → probability of k^{th} symbol
l_k → number of bits in the source encoder
τ → average code word length

3.8.2 Shannon's second theorem (or) channel coding theorem

Channel coding is a process of mapping the incoming data sequence into channel input sequence and in reverse channel output sequence into output data sequence.

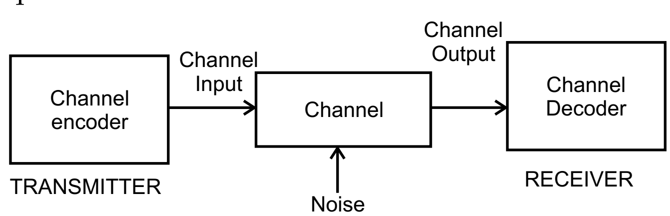


Fig .3.10 channel input and output sequence

- The channel encoder and decoder are designed in such a way to achieve a high efficiency of the communication systems.
- channel encoding process involves two steps

  1) source encoding
  2) removal of redundant data

- This channel coding is obtained by block code and convolution codes.

- Consider a block codes, in which message sequence is divided into blocks of K-bits long and each K-bits block is mapped into n-bit block.

  where $(n \geq k)$. The number of redundant bits added by the encoder to each transmitted block is $(n - m)$ bits. Then,

$$\boxed{\text{code rate} \quad r = \frac{k}{n} \qquad r < 1}$$

- Channel coding theorem is stated in two parts,

  (i) Let DMS with alphabet X and entropy $H(x)$ produce symbols at every 'T_s' seconds. Let channel capacity 'C' used once for every 'T_c' seconds. Then if,

$$\boxed{\frac{H(X)}{T_s} \leq \frac{C}{T_c}}$$

  where $\frac{C}{T_c}$ = critical rate.

  (ii) coding scheme for which the source output can be transmitted over the channel and can be reconstructed with small probability error. Else if,

$$\boxed{\frac{H(X)}{T_s} = \frac{C}{T_s}}$$

  Then the system is at critical rate.

  (iii) If

$$\boxed{\frac{H(X)}{T_s} > \frac{C}{T_s}}$$

  No possibilities of transmission and reception with small probability of error.

3.8.3 Shannon's third theorem (or) Channel capacity theorem

- When Shannon's theorem of channel capacity is applied specifically to a channel in which noise is Gaussian is known as Shannon's Hartley (or) Information capacity theorem.

- The channel capacity of Gaussian channel is

$$\boxed{C = B\log_2\left[1+\frac{S}{N}\right] \quad \text{bits/sec}}$$

where,

B = channel bandwidth

S = signal power

N = noise power

- Here signal power $\int_{-B}^{B} (P.S.D)$

Power spectral density (PSD) of white noise is $\frac{N_o}{2}$

- Noise power

$$\int_{-B}^{B} \frac{N_o}{2}\, df = N_o B$$

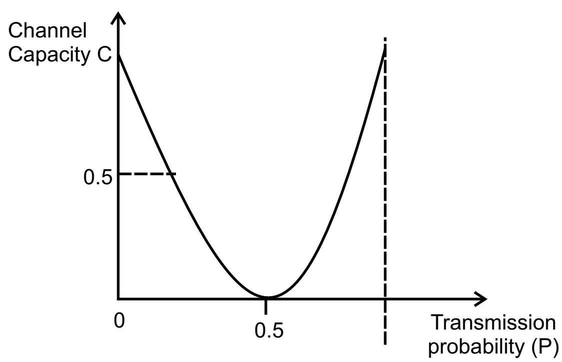


Fig .3.11 Channel capacity Vs transmission probability

Table 3.1 Status of channel capacity

Transmission probability (P)	Channel capacity (C)	Status
0	1	1 bit/channel
0.5	0.5	Maximum noise
1	0	Useless channel

3.9 Bandwidth - SNR Tradeoff Codes

- Channel capacity of the Gaussian channel is given as,

$$\boxed{C = B\log_2\left(1+\frac{S}{N}\right) \quad \text{bits/sec}}$$

Channel capacity depends on two factors (i.e.,) (i) bandwidth and (ii) signal to noise ration (SNR)

- Depends on bandwidth and SNR, channel capacity of noiseless channel is analyzed in two ways.

1. **Infinite capacity for noiseless channel**

Consider a channel with no noise ($N = 0$)

$$C = B\log_2\left(1+\frac{S}{0}\right) \quad \text{bits/sec}$$
$$C = B\log_2(1+\infty)$$
$$C = \infty$$

$\therefore$ noiseless channel has infinite capacity.

2. **Limited capacity for infinite bandwidth**

Consider a bandwidth '$B = \infty$' that is increased bandwidth, noise power (N) also increases. Then noise power is given by

$$\boxed{N = \eta B}$$

3. Maximum value by increasing bandwidth is given as

$$C = B\log_2\left[1+\frac{S}{N}\right]$$
$$C = B\log_2\left[1+\frac{S}{N_oB}\right] \qquad (\because N = N_oB)$$

Multiply and divide by $\dfrac{S}{N_o}$

$$C = \frac{S}{N_o}\times\frac{N_o}{S}B\log_2\left[1+\frac{S}{N_oB}\right]$$
$$= \frac{S}{N_o}\log_2\left[1+\frac{S}{N_oB}\right]^{\frac{N_oB}{S}}$$
$$C = \frac{S}{N_o}\log_2\left[1+\frac{S}{N_oB}\right]^{\frac{1}{S/N_oB}}$$

Upper limit calculation:

Apply $X = \dfrac{S}{N_o B}$, then $N_o B \to \infty$, $X \to 0$ in the above equation

$$C = \frac{S}{N_o} \lim_{x \to 0} \log_2 (1+x)^{1/x}$$

but

$$\lim_{x \to 0} (1+x)^{1/x} = e$$

$$C_\infty = \frac{S}{N_o} \log_2 e$$

$$C_\infty = \frac{S}{N_o} \frac{\log_{10} e}{\log_{10} 2}$$

$$C_\infty = 1.44 \frac{S}{N_o}$$

Conclusion

The tradeoff between bandwidth and SNR is

1. To decrease the bandwidth, the signal power has to be increased.
2. Similarly to decrease the signal power, the bandwidth must be increased.

3.10 Line Coding (or) Line Encoding (or) BW-SNR Tradeoff Codes

Definition

Electrical representation of binary data streams is called as line codes.

(i.e.,) Digital digits are mapped to a pulse waveform this waveform is line codes.

Properties of line codes

1. DC component
2. Self synchronization
3. Error detection
4. Bandwidth compression
5. Differential encoding
6. Noise immunity
7. Spectral compatibility with channel
8. Transparency

3.11 Types of Line Coding

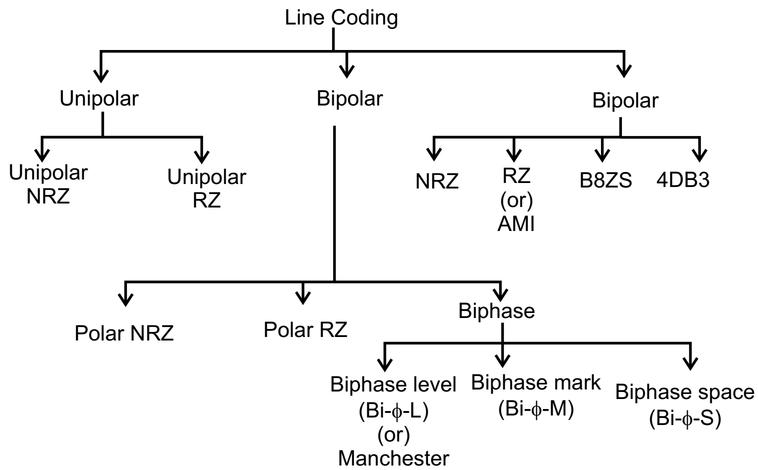


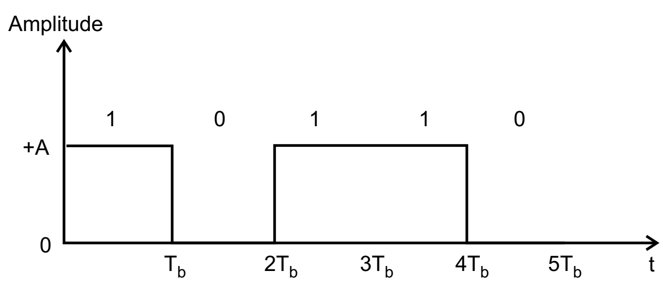
Fig .3.12 Classification of line codes

3.11.1 Unipolar

- Unipolar encoding is single non-zero voltage level and zero voltage level are used.
- It is primary and very simple coding.

Unipolar Non Return Zero (NRZ)

Non-Return zero is a code in which '1' is represented by a voltage level and '0' is represented by another voltage level. There is no neutral or rest condition in both 0's and 1's.

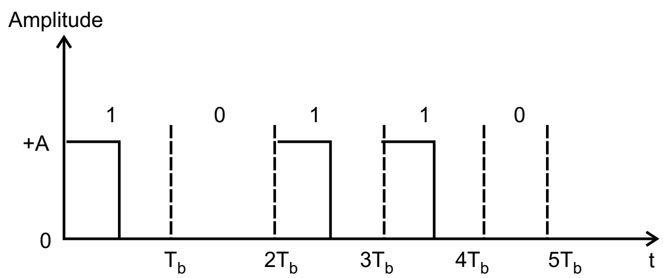


1's $\rightarrow$ +A voltage

0's $\rightarrow$ 0 voltage

Unipolar Return Zero (RZ)

Waveform comes back to zero level after half of bit intervals '$\frac{T_b}{2}$'.



1′s → +A voltage for 0 to $\frac{T_b}{2}$ intervals

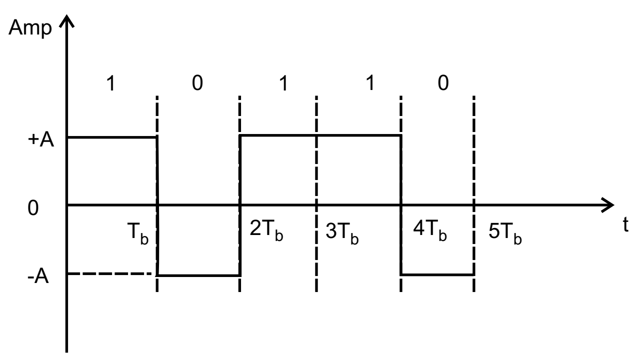
0′s → 0 voltage

3.11.2 Polar

- Polar encoding uses two levels (positive and negative voltage).
- It has single positive voltage and single negative voltage and no non-zero voltage level.

Non Return Zero Polar NRZ

0′s are represented by -A voltage for full duration and 1′s are represented by +A voltage for half duration.

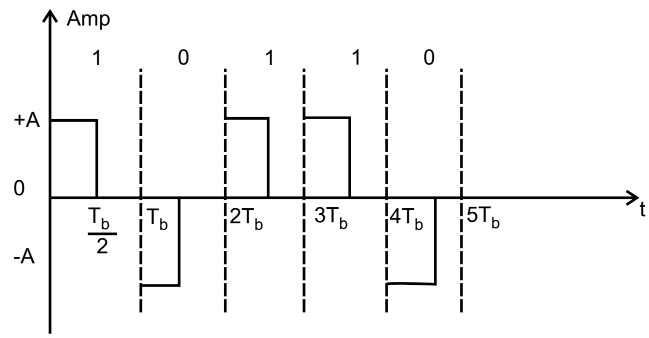


1′s → +A voltage

0′s → -A voltage

Polar RZ

0′s are represented by -A voltage for half duration and 0 voltage for another half duration.



$$1'\text{s} \rightarrow \frac{T_{b1}}{2}(+A)$$
$$\frac{T_{b2}}{2}(0)$$
$$0'\text{s} \rightarrow \frac{T_{b1}}{2}(-A)$$
$$\frac{T_{b2}}{2}(0)$$

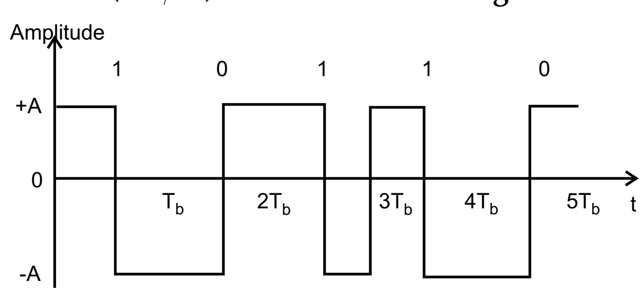
Biphase

The signal changes at the middle of the bit intervals but does not return to zero.

Biphase encoding is divided into 3 types.

(i) Biphase level (Bi-ϕ-L) or Manchester coding
(ii) Biphase mark (Bi-ϕ-M)
(iii) Biphase space (Bi-ϕ-S)

(i) Biphase level (Bi-ϕ-L) or Manchester coding:



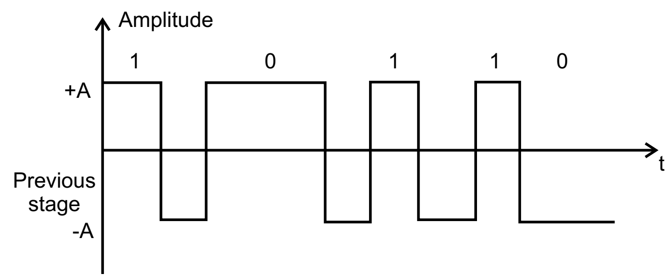
- 0's represented as transition from negative voltage (-A) positive voltage (+A) for every $\frac{T_b}{2}$ duration.

- 1′s represented as transition from positive voltage (+A) negative voltage (-A) for every $\frac{T_b}{2}$ duration.

  0 $\rightarrow$ -A to +A

  1 $\rightarrow$ +A to -A

(ii) Biphase mark (Bi-ϕ-M):

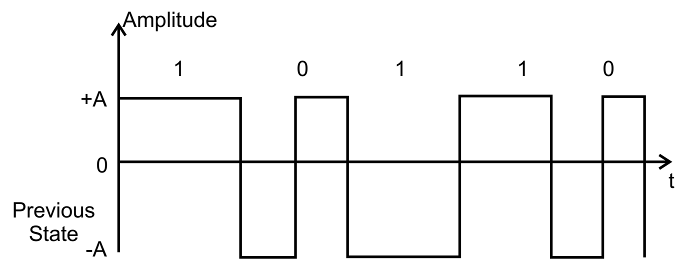


- 0′s represented by a single transition and 1′s represented by a double transition for every $\frac{T_b}{2}$ duration.

  0 $\rightarrow$ single transition and

  1 $\rightarrow$ double transition
- For the transition consider the previous state as negative voltage (-A).

(iii) Biphase space (Bi-ϕ-S):



- It is inverse of Biphase-mark
- 0′s represented by a double transition and 1′s represented by a single transition for ever $\frac{T_b}{2}$ duration.

  0 $\rightarrow$ single transition

  1 $\rightarrow$ double transition
- For a transition, consider the previous state as negative voltage (-A).

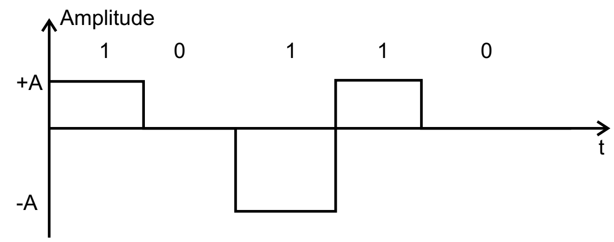
3.11.3 Bipolar

Bipolar coding uses three voltage levels, positive, negative and zero.

0′s → 0 voltage

1′s → represents alternating positive and negative voltage.

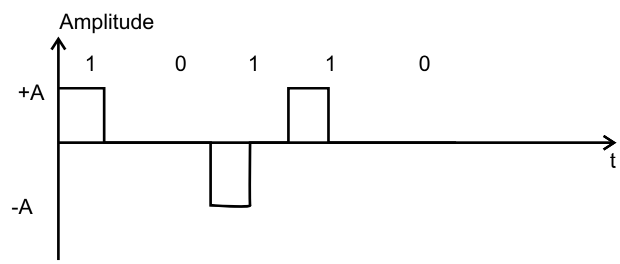
Bipolar NRZ



0′s → 0 voltage

1′s → alternate +A and -A voltage

Bipolar RZ (or) alternate mark inversion (AMI)



0′s → always 0 voltage

1′s → for first duration $\frac{T_{b1}}{2}$ (+A) and second half duration $\frac{T_{b2}}{2}$ (0V) it is represented as alternate manner like $\frac{T_{b1}}{2}$ (-A) and $\frac{T_{b2}}{2}$ (0V).

Bipolar 8-zero substitution (B8ZS)

- To provide synchronization of long strings of 0′s B8ZS is used. This encoding is used in North America.
- Whenever eight or more consecutive 0′s are encountered in the data stream. the pattern of data stream is modified in two ways based on the polarity of the previous one.

1st way:

- If previous bit is positive polarity then, eight 0's data stream will be encoded as 0, 0, 0, +ve, -ve, 0, -ve and +ve.

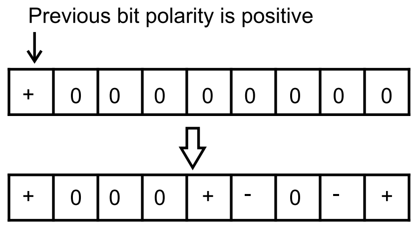


Fig .3.13 B8ZS encoding if previous bit is positive

2nd way:

- If previous bit is negative polarity then, eight 0's data stream will be encoded as 0, 0, 0, -ve, +ve, 0, +ve and -ve.

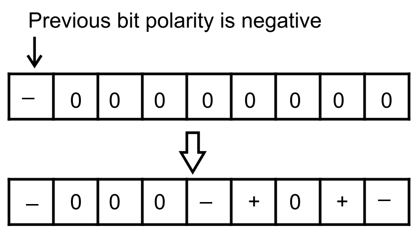


Fig .3.14 B8ZS encoding if previous bit is negative

High density bipolar 3 (HDB3) or HDBP

- To provide synchronization of long strings of 0's HDBP is used in Europe & Japan.
- Whenever 8 0's are consecutively encountered in the data stream, it is divided into two sets of four 0's of consecutive data stream. The pattern of this data stream is modified in two ways based on the polarity of previous one and number of 1's since the last substitution.

1st way:

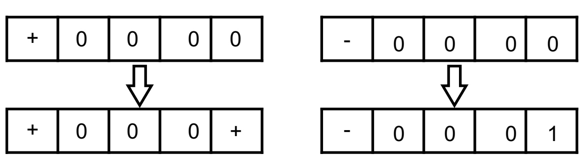


Fig .3.15 If the number of 1's since last substitution is odd

2nd way:

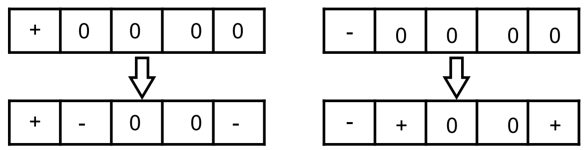


Fig .3.16 If the number of 1's since last substitution is even

3.12 MBnB Codes

- Extended type of line codes is called as MBnB code (or) extended codes.
- MBnB codes can be written in different ways,

  (i) 4B5B codes (when $M = 4$ & $N = 5$)

  (ii) 6B8B codes (when $M = 6$ & $N = 8$)

  (iii) 8B10B codes (when $M = 8$ & $N = 10$)

  (iv) 64B66B codes (when $M = 64$ & $N = 66$)

(i) 4B5B codes

- It is a form of data communication in telecommunication
- Process of mapping a groups of 4-bits into groups of 5-bits, with the output of minimum 1-bit.
- For example in NRZ if consecutive 0's of 4-bits (0000) is encountered there will be no transitions and that causes clocking problems for the receiver.
- This problem of 4B5B is solved by assigning each block of 4 consecutive bits into block of 5 consecutive bits. These 5-bits are predetermined in a dictionary and they are chosen to ensure that there will be at least two transition per block of bits.

(ii) 6B8B codes

- It is a process of mapping 6-bit codes into 8-bit codes with minimum output of 1-bit used for maintaining DC-balance in a communication system.
- Each 8-bit output symbol contains 4-zero bits and 4 one bits, so the code can like a parity bit, detect all single -bit errors.

(iii) 8B10B codes

- 8B/10B line code is a process of mapping 8 bit codes into 10 bit codes to achieve DC-balance and bounded disparity and yet provide enough state changes to allow reasonable clock recovery in a communication systems.
- The difference between the count of 1's and 0's in a string of at least 20 bits is no more than 2, and that there are not more than five 1's or 0's in a row, which reduce the demand for the lower bandwidth limit of the channel necessary to transfer the signal.

(iv) 64B66B codes

- 64B/66B is a process of mapping 64-bit data into 66-bit data to achieve DC balance and bounded disparity and yet provide enough state changes to allow reasonable clock recovery in the field of data networking and transmission
- There are just as many 1's and 0's in a string of two symbols and that there are not too many 1's or 0's in a row. This is an important attribute in a signal that needs to be send at high rates because it helps reduce intersymbol interference.

3.13 Efficiency of Transmission

In order to achieve a efficiency transmission of signal from source to destination, the following process are consider.

- Multiplexing
- Compression
  - Data compression
  - Video compression

3.13.1 Multiplexing

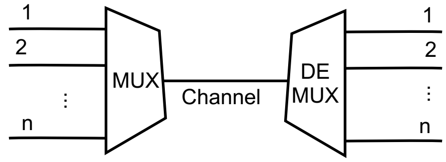


Fig .3.17 Multiplexing & Demultiplexing

- Several number of signals from several sources are transmitted and received simultaneously through a common communication channel is called as multiplexing.

3.13.2 Types of Multiplexing

- Time Division Multiplexing (TDM)
- Frequency Division Multiplexing (FDM)

Time Division Multiplexing (TDM)

TDM is a process of transmitting and receiving two or more independent signal over a common channel with respect to fixed-length time slots.

Applications:

- GSM telephone system

Frequency Division Multiplexing (FDM)

FDM is a process of transmitting and receiving two or more independent signal over a common channel with respect to different frequency bands.

Applications

- Acoustic telegraphy and harmonic telegraphy
- FM and AM radio broadcasting
- Television broadcasting

3.13.3 Data compression

- Data compression reduces the size of data files to transmit more information with fewer bits.
- Used for storage
- Used to increase the transmission efficiency
- Principle: redundant information should be removed from the signal prior to transmission
- For lossless data compression, reconstituted data is identical to original. Example: ZIP, GIF.
- For lossy data compression, reconstituted data is only precepterally equivalent. Example: JPEG, MPEG.

3.13.4 Video compression

- Video compression uses modern coding techniques used to reduce redundancy in video data.
- Application: Video conferencing to video on demand to video phones.
- Standards: Video compression standards (MPEG - 1, 2, 4, 7) and teleconferencing standards (H.2XX)
- Algorithms: 1) Motion estimation 2) motion compensation and image subtraction 3) discrete cosine transform 4) quantization 5) run length encoding 6) entropy coding - huffman coding.

3.14 Error Control Codes

3.14.1 Introduction

Coding is a procedure for mapping a given set of message (M_1, M_2, ... M_n) into a new set of encoded messages (C_1, C_2, ... C_n) such a way that the transformation is one to one for each message.

Advantages of coding

- Coding is used to reduce the probability of error and correcting errors.
- improves efficiency of transmission

Need for coding

- To change the data quality to the acceptable level.
- To reduce SNR for a fixed bit error rate.
- Reduction in SNR reduces the transmitted power.

Disadvantages of coding

- Coding increases receiver complexity
- Addition of redundancy (i.e.,) extra bits increase the transmission bandwidth.

Error control codes

The objectives of a channel codes are,

- To be able to encode the symbol in a fast and efficient way
- To have the capacity to detect and correct errors.
- To be able to decode the message in fast and efficient way.

Types

1. Error detection & retransmission
2. Error detection and correction

Error detection & retransmission

- The receiver detects an error and requests the transmitter for transmission so the receiver does not endevour to rectify the detected error.
- Delay due to retransmission is not tolerable in real time applications.
- Example is ARQ commonly used in computer communication.

Error detection and correction

- The receiver is not only able to detect the error also correct the error.
- Correction of errors needs more redundancy bits than detecting the error so it needs more overhead than ARQ method.
- Also referred as forward acting error correction.

Types

1. Block codes (Linear)
2. Convolutional codes (Non-linear)

- Convolutional codes (Non-linear)

  The convolutional code is a type of non-linear code.

  Non-linear code is defined as "addition of two code word does not produce another codeword."
- Block codes (Linear)

  The 'n' bit codeword can be obtained from k-bit message and $(n-k)$ bit parity such a code is called as (n, k) block code.
- A block code is a set of fixed length of codewords. The fixed length of these codewords is called the block length and its denoted by 'n'.

3.15 Convolutional codes

- Convolutional encoder is a finite state machine that consists of M-stage shift register with prescribed connections to n modulo-2 adders and multiplexer that serially send the output to the adder.

Definition

"The encoder accepts a 'k'-bits message come in serially rather than in large blocks."

3.15.1 Structure of convolutional encoder

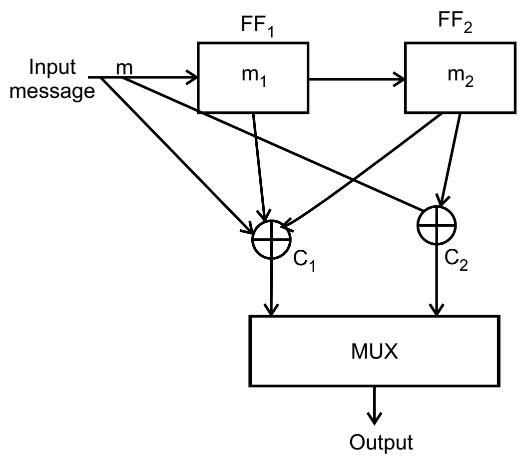


Fig .3.18 Structure of 1/2 rate convolutional encoder

Operation

- Let $m = (m_0, m_1, ...)$ information sequence enters one bit at a time.
- The values of C_1 & C_2 are obtained by

$$C_1 = m \oplus m_1 \oplus m_2$$
$$C_2 = m \oplus m_2$$

- Now the shift register shifts the value of m_0 to m_1 & m_1 to m_2.

  $\therefore$ Hence for every 1 bit, 2 output = bitscore coded.

  Number of message bits; $k = 1$

  Number of encoded output bits for one message bits $n = 2$.

Code rate (r)

It is defined as ratio between 'k' & 'n'

$$\boxed{r = \frac{k}{n}}$$

$k \rightarrow$ number of input taken at a time

$n \rightarrow$ number of output

Constraint length K

Defined as the number of shifts over which a single message bit can influence the encoder output. It is expressed in terms of bits. $(K = M + 1)$.

where $M \rightarrow$ Number of shift register.

$n \rightarrow$ Number of output

$K \rightarrow$ Number of input taken at a time.

Dimension of the code (n, k)

Given by (n, k), for above example dimension is (2, 1).

Two types of approach to analysis the convolutional codes:

1. Time domain approach
2. Transform domain approach

3.15.2 Time Domain Approach

- Consider a k/n convolutional encoder for the sequence

  $\{g_1', g_2', ...g_m'\}$ denote the impulse response of the adder which generates 'C_1'.

  Similarly $\left\{g_1^2, g_2^2, ...g_m^2\right\}$ denote the impulse response of the adder which generates 'C_2'.

  These responses are called as the generator sequences of codes

$$C_1' = \sum_{l=0}^{M} g_e^{(1)} m_{i-l} \qquad i = 0, 1, 2....$$

  Similarly for

$$C_2' = \sum_{l=0}^{M} g_e^2 m_{i-l}$$

$$\text{Final output sequence} = \left\{C_1^1 C_1^2, C_2^1 C_2^2, ...\right\}$$

3.15.3 Transform domain approach

- Message polynomial can be written as $m(D)$
- The generator polynomial of each path is

$$g^1 (D), g^2 (D), ...g^n (D)$$

- compute code polynomials or output of encoder of each path

$$C^1(D) = m(D)\, g^1(D)$$
$$C^2(D) = m(D)\, g^2(D)$$
$$\vdots$$
$$C^n(D) = m(D)\, g^n(D)$$

- The codeword of each bit by combining all the code polynomials.

3.15.4 Structural properties of convolutional encoder

Convolutional code takes one message bit at a time and generates two (or) more encoded bits.

There are three structural properties are,

1. State diagram
2. Code tree
3. Trellis

State diagram

- In which the nodes represent the four possible states ($n = 2$) of the encoder.
- Each node has two incoming and two outgoing branches.
- For binary input zero '0' → solid branch is used.
- For binary input one '1' → dotted branch is used.

State of encoder

It defines the state of message

$$2^2 = 4$$

m_1	m_3	state
0	0	a
0	1	b
1	0	c
1	1	d

Case 1: If number of flipflop $M = 2$

then the possible states are four (a, b, c, d).

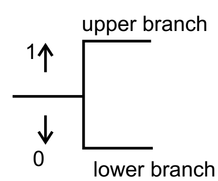
Case 2: If number of flipflop is 3, then 8 possible states. (i.e.,)

$$2^M = 2^3 = 8$$

m_1	m_2	m_2	state
0	0	0	a
0	0	1	b
0	1	0	c
0	1	1	d
1	0	0	e
1	0	1	f
1	1	0	g
1	1	1	h

Code tree

- Each branch of the tree represents an input symbol with a corresponding pair of output binary symbols indicated on the branch.
- Input bit '0' specifies the upper branch of a tree and input bit '1' specifies the lower branch.
- The tree becomes repetitive after the first $(M + 1)$ branches.



Trellis

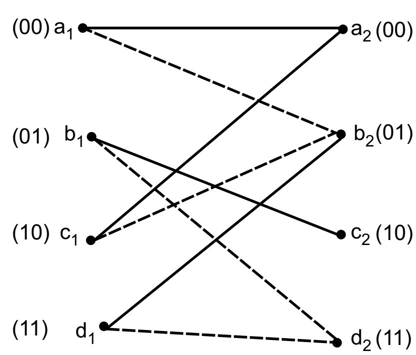


Fig .3.19 Trellis structure

- It is so called since it is a tree like structure with re-emerging branches.
- Code branch produced by an input '0' is drawn as a solid line and a branch produced by input '1' is drawn as dotted line.

- Trellis contains $(k+K)$ levels.

  $k \rightarrow$ Number of input (always 1 for convolutional code)
  $K \rightarrow$ Constraint length
- Node on left side denotes four present state & right side denotes four next state.

Comparison between code tree & code trellis

Code tree	**Code Trellis**
• It is lengthy way of representing coding process.	• Comnpact way of representing coding process.
• Complex to implement	• Simpler to implement
• Indicates flow of coded signal	• Indicates transitions from current to next states.
• Decoding is simple	• Decoding is complex

Solved Problem 3.9 *Design a 1/3 convolutional encoder with constraint length 3 and encode the message bits 1001, for the generated sequences are* $g_1 = (1,0,1)$, $g_2 = (1,1,0)$ *&* $g_3 = (1,1,1)$. *Also draw the state table, state diagram, code tree and trellis.*

Solution:

Definition: Convolutional encoder takes one bit at a time and generates two or more encoded bits.

Given:

- Code rate

$$(r) = \frac{1}{3} = \frac{k}{n}$$

$$\therefore \; k = 1; \quad \text{Number of input bit}$$

$$n = 3; \quad \text{Number of output bit (at a time)}$$

- Constraint length

$$K = 3$$

$$K = M + 1$$

$M \rightarrow$ Number of flip flops

$$3 = M + 1$$

$$M = 2$$

- Message bits 1 0 0 1
- Generator sequences

$$g_1 = (101)\,; g_2 = (110)\,; g_3 = (111)$$

Structure of encoder

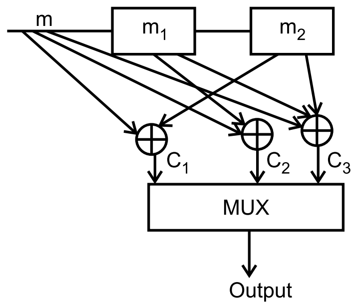


Fig .3.20 Structure of encoder

$n = 3$; Modulo 2 adder

$M = 2$; Flip flops

$$\begin{aligned} g_1 &= (\ 1\ \ 0\ \ 1\ ) \\ g_2 &= (\ 1\ \ 1\ \ 0\ ) \\ g_3 &= (\ 1\ \ 1\ \ 1\ ) \\ & \quad\ \ m\ \ m_1 m_2 \end{aligned}$$

Output of

$$C_1 = m \oplus m_2$$
$$C_2 = m \oplus m_1$$
$$C_3 = m \oplus m_1 \oplus m_2$$

State table

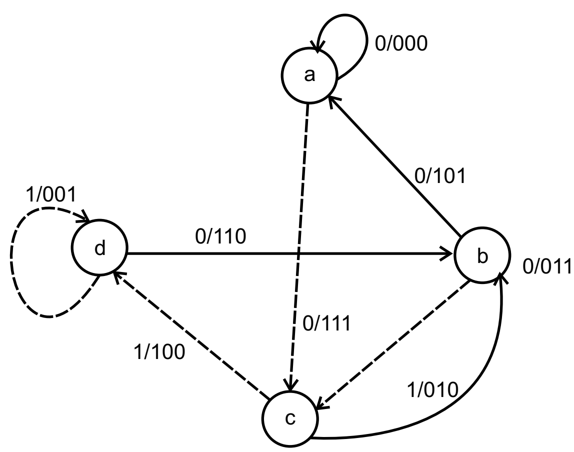
Msgbits m	presentstate m_1 m_2	Nextstate	C_1 $m \oplus m_2$	C_2 $m \oplus m_1$	C_3 $m \oplus m_1 \oplus m_2$	
0	0 0 a	0 0 a	0	0	0	a — 0: a(000)
1	0 0 a	1 0 c	1	1	1	1: c(111)
0	0 1 b	0 0 a	1	0	1	b — 0: a(101)
1	0 1 b	1 0 c	0	1	0	1: c(010)
0	1 0 c	0 1 b	0	1	1	c — 0: b(011)
1	1 0 c	1 1 d	1	0	0	1: d(100)
0	1 1 d	0 1 b	1	1	0	d — 0: b(110)
1	1 1 d	1 1 d	0	0	1	1: d(001)

- State table, in which present state and next state of all possible inputs are calculated. The code word can be calculated from all possible input.

State diagram:

- State diagram in which nodes represent in four possible states of the encoder.
- Each node has two incoming and outgoing branches.
- For '0' $\rightarrow$ solid line
- For '1' $\rightarrow$ solid line
- Two flip flops $\rightarrow$ Four states

$00 \rightarrow a$, $01 \rightarrow b$, $10 \rightarrow c$, $11 \rightarrow d$



Code tree:

Code tree is developed with the help of state table.

Step 1: Begin with node 'a'

Step 2: Draw its states for the message input bit '0' and '1'.

Step 3: Similarly, draw the next states for each and every state.

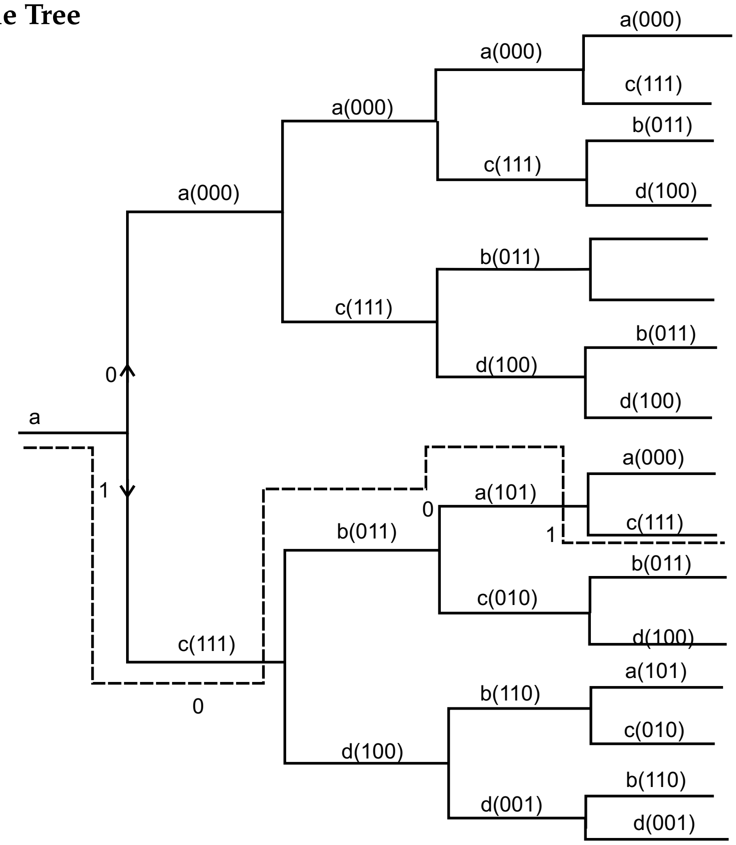
Step 4: Repeat till code tree starts repeating.

Step 5: Upward movement indicates $m = 0$ & downward movement indicates $m = 1$

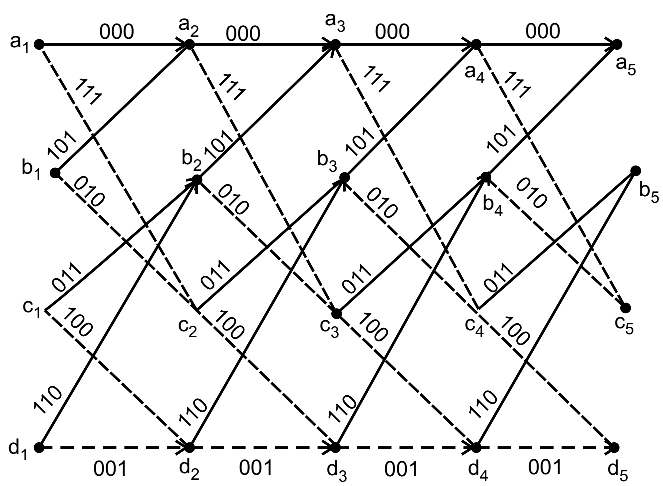
Given message bits 1 0 0 1

Message bits	1	0	0	1
Tracing path	$a-c$	$c-b$	$b-a$	$a-c$
Encoded bits	111	011	101	111

Code Tree



Code Trellis



- Nodes on left side denotes four current states of encoder.
- Nodes on right side denotes four next states of encoder.

Solved Problem 3.10 *Design 1/2 convolutional encoder with constraint length 3. Find the encoded bits for the message 1001. The generated sequences are* $g_1 = (1,1,1)$, $g_2 = (1,0,1)$. *Also draw the state table, state diagram, code tree and trellis.*

Solution:

Definition: Convolutional encoder takes one bit at a time and generates two or more encoded bits.

Given:

- Code rate

$$(r) = \frac{1}{2} = \frac{k}{n}$$

$$\therefore \quad k = 1; \quad \text{Number of input bit}$$

$$n = 2; \quad \text{Number of output bit (at a time)}$$

- Constraint length

$$K = 3$$

$$K = M + 1$$

$M \rightarrow$ Number of flip flops

$$3 = M + 1$$

$$M = 2$$

And also given $M = 1\ 0\ 0\ 1$

- Generator sequences

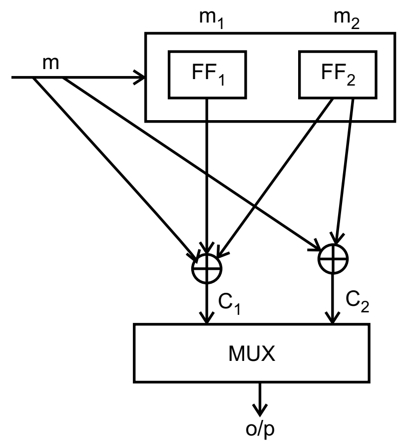
$$g_1 = (1,1,1)$$

$$g_2 = (1,0,1)$$

Structure of encoder

$$C_1 = m \oplus m_1 \oplus m_2$$

$$C_2 = m \oplus m_2$$

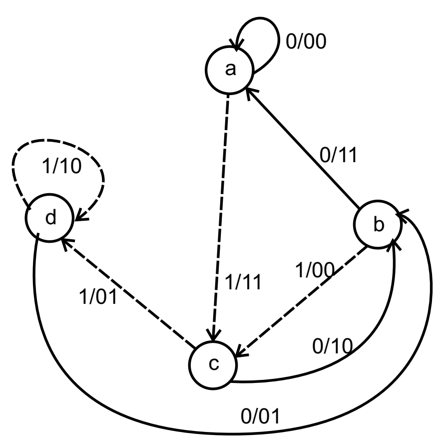


State table

State table, in which present state and next state of all possible inputs are calculated. The code word can be calculated from all possible input.

Msgbits m	presentstate m_1 m_2	Nextstate	C_1 $m \oplus m_2$	C_2 $m \oplus m_1$	State diagram
0	0 0 a	0 0 a	0	0	a 0 a(00)
1	0 0 a	1 0 c	1	1	1 c(11)
0	0 1 b	0 0 a	1	1	b 0 a(11)
1	0 1 b	1 0 c	0	0	1 c(00)
0	1 0 c	0 1 b	1	0	c 0 a(10)
1	1 0 c	1 1 d	0	1	1 c(01)
0	1 1 d	0 1 b	0	1	d 0 a(01)
1	1 1 d	1 1 d	1	0	1 c(10)

State diagram



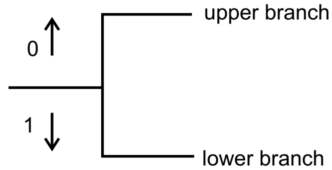
State diagram, in which nodes represent in four possible states of the encoder. Each node has two incoming and outgoing branches.

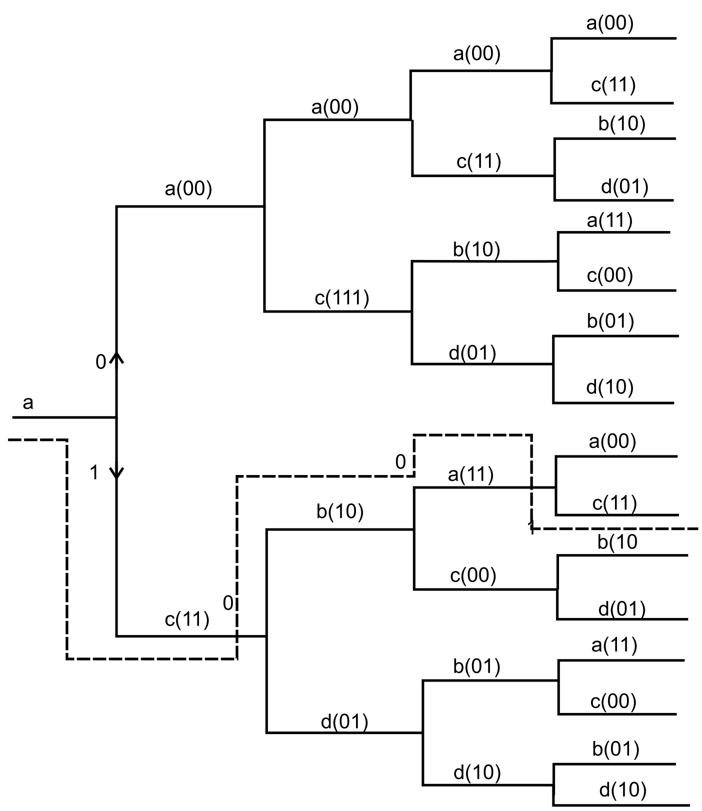
For 0 → solid branches

For 1 → dotted branches

Code tree

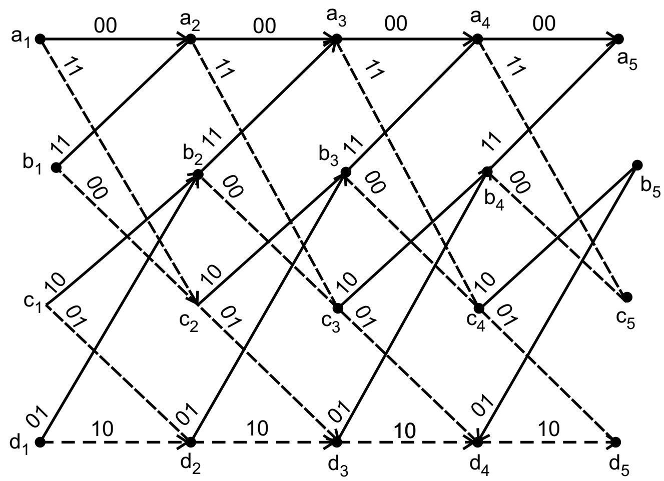
It is a graphical way to represent the structural properties of convolutional encoder.





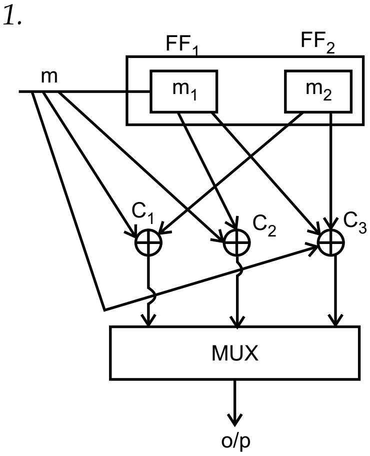
Code Trellis

- Nodes on left side denote four current state of encoder.
- Nodes on right side denote four next state of encoder.



Given msg:	1	0	0	1
Tracing path:	a-c	c-b	b-a	a-b
Encoded bits:	11	10	11	11

Solved Problem 3.11 *For the given diagram find the code rate, dimension of the code and code word using transform domain approach and time domain approach for the message bit 10111.*



By transform domain approach,

$$g_1\,(1, 0, 1) \Rightarrow g_1\,(D) = D^0 + D^2$$
$$g_2\,(1, 1, 0) \Rightarrow g_2\,(D) = D^0 + D^1$$
$$g_3\,(1, 1, 1) \Rightarrow g_3\,(D) = D^0 + D^1 + D^2$$

$$M = 1\ 0\ 1\ 1\ 1 = D^0 + D^2 + D^3 + D^4$$

$$C_1(D) = g_1(D) \cdot M(D) = \left[D^0 + D^2\right]\left[D^0 + D^2 + D^3 + D^4\right]$$

$$= D^0 + D^3 + D^5 + D^6$$

$$C_1 = 1\ 0\ 0\ 1\ 0\ 1\ 1$$

M	=	1	0	1	1	1
o/p	=	111	011	010	100	001

$$C_2(D) = g_2(D) \cdot M(D)$$

$$= \left(D^0 + D^1\right)\left(D^0 + D^2 + D^3 + D^4\right)$$

$$C_2(D) = D^0 + D^1 + D^2 + D^5$$

$$C_2 = 1\ 1\ 1\ 0\ 0\ 1\ 0$$

$$C_3(D) = g_3(D) \cdot M(D)$$

$$= \left(D^0 + D^1 + D^2\right)\left(D^0 + D^2 + D^3 + D^4\right)$$

$$C_3(D) = D^0 + D^1 + D^4 + D^6$$

$$C_3 = 1\ 1\ 0\ 0\ 1\ 0\ 1$$

$$\therefore\ M = 1\ 0\ 1\ 1\ 1$$

$$C_1 = 1\ 0\ 0\ 1\ 0\ 1\ 1$$

$$C_2 = 1\ 1\ 1\ 0\ 0\ 1\ 0$$

$$C_3 = 1\ 1\ 0\ 0\ 1\ 0\ 1$$

By time domain approach

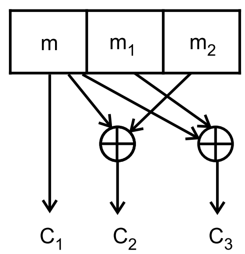
Msg bits	present state		C_1	C_2	C_3
m	m_1	m_2	$m \oplus m_2$	$m \oplus m_1$	$m \oplus m_1 \oplus m_2$
1	0	0	1	1	1
0	1	0	0	1	1
1	0	1	0	1	0
1	1	0	1	0	0
1	1	1	0	0	1

M	=	1	0	1	1	1
o/p	=	111	011	010	100	001

From the given diagram,

$$g_1 = m \oplus m_2$$
$$g_2 = m \oplus m_1$$
$$g_3 = m \oplus m_1 \oplus m_2$$

Solved Problem 3.12 *For the given diagram draw the state table, code tree, trellis and state diagram also determine encode the given message 1 1 0 1.*



Given:

- Message bits 1 1 0 1 0
- From figure

(i) To determine dimension of the code

for every message bit $(k = 1)$ and three output bits $(n = 3)$ are generated. This is $\frac{1}{3}$ code rate.

Where two stages in shift register is available hence constraint length $K = M + 1 = 2 + 1$.

$K = 3$; $M \rightarrow$ Number of shift register.

$\therefore \quad n = 3$, $k = 1$ & $K = 3$.

Dimension: $(n, k) = (3, 1)$

Constraint length $K = 3$

Code rate $r = \dfrac{K}{n} = \dfrac{1}{3}$

$$C_1 = m$$
$$C_2 = m \oplus m_2$$
$$C_3 = m \oplus m_1$$

(ii) To obtain the state table

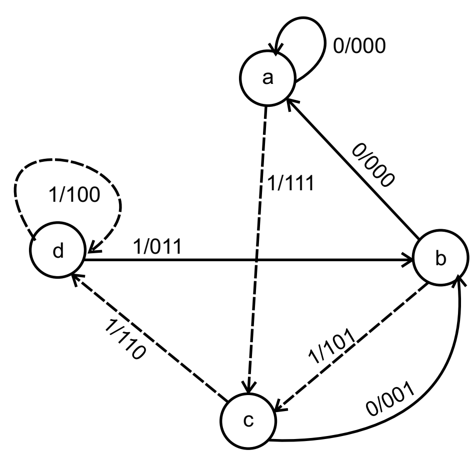
$$m_1m_2 = 00, \quad \text{State } 'a'$$
$$m_1m_2 = 01, \quad \text{State } 'b'$$
$$m_1m_2 = 10, \quad \text{State } 'c'$$
$$m_1m_2 = 11, \quad \text{State } 'd'$$

State table

Msgbits m	presentstate m_1 m_2	Nextstate	C_1 m	C_2 $m \oplus m_2$	C_3 $m \oplus m_1$	
0	0 0 a	0 0 a	0	0	0	a: 0 → a(000)
1	0 0 a	1 0 c	1	1	1	a: 1 → c(111)
0	0 1 b	0 0 a	0	1	0	b: 0 → a(010)
1	0 1 b	1 0 c	1	0	1	b: 1 → c(101)
0	1 0 c	0 1 b	0	0	1	c: 0 → b(001)
1	1 0 c	1 1 d	1	1	0	c: 1 → d(110)
0	1 1 d	0 1 b	0	1	1	d: 0 → b(011)
1	1 1 d	1 1 d	1	0	0	d: 1 → d(100)

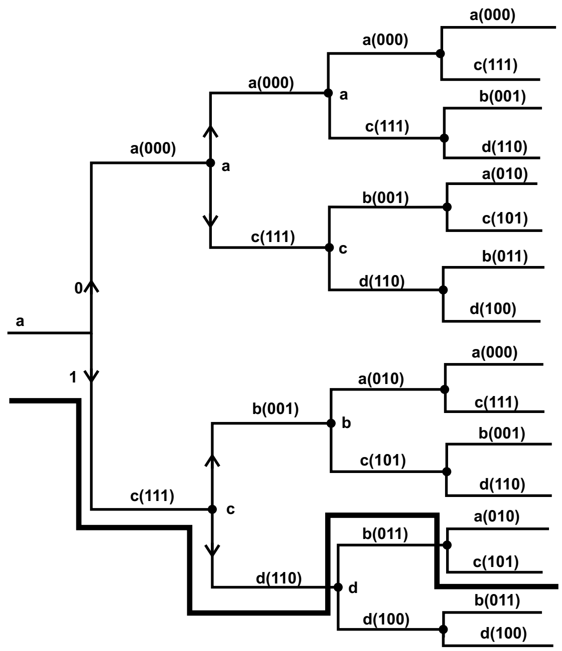
State diagram:

State diagram can be easily drawn based of state table.



(iii) **Code tree:**

It can be drawn easily based on state table.

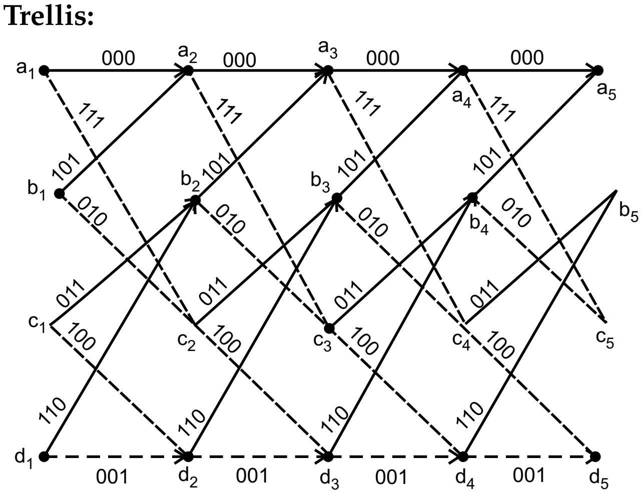


Given message bits 1 1 0 1

Encoded bits 1 1 1

Message bits	1	1	0	1
Tracing path	$a-c$	$c-d$	$d-b$	$b-c$
Encoded bits	111	110	011	101

Code Trellis:



Solved Problem 3.13 *A convolutional encoder has a single shift register with two stages, constraint length = 3, three modulo - 2 adders and an output multiplexer. The generator sequences of the encoder are as follows.* $g_1 = (1,0,1)$, $g_2 = (1,1,0)$ *&* $g_3 = (1,1,1)$. *Realize a convolutional encoder.*

Solution:

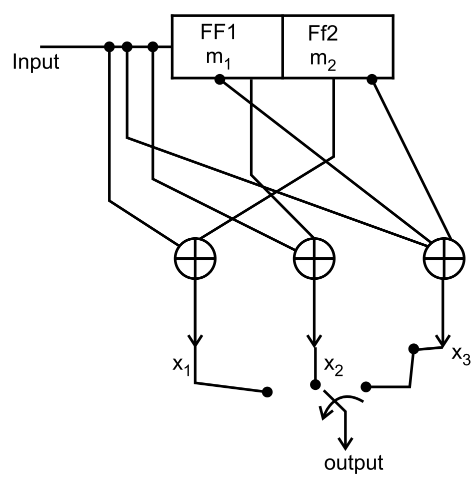
$$\begin{aligned}\text{No. of flip flops } (M) &= \text{Constraint length} - 1 \\ &= 3 - 1 \\ &= 2\end{aligned}$$

No. of adders $n = 3$ (or) No. of outputs.

Generator sequences

$$g^1 = (101)$$
$$g^2 = (110)$$
$$g^3 = (111)$$

Convolutional encoder



$$C_1 \text{ or } x_1 = \text{I/P} + m_2$$
$$C_2 \text{ or } x_2 = \text{I/P} + m_1$$
$$C_3 \text{ or } x_3 = \text{I/P} + m_1 + m_2$$

Solved Problem 3.14 *Find the encoder output for the message sequence 10111... specifications are (i) code rate =* $\frac{1}{2}$, *constraint length* $k = 2$. *The generator sequences are* $g^1 = (1,1)$, $g^2 = (1,0)$.

Solution:

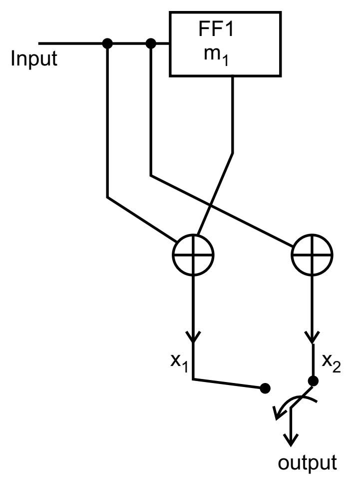
Given:

Code rate $= \frac{k}{n} = \frac{1}{2}$

$\therefore \quad n = 2$, (i.e.,) no. of adders = 2

Message bits entered at a time =1

Constraint length $K = 2$



(i.e.,) No. of flip flops are $= K - 1 = 2 - 1 = 1$

$$x_1 = \text{I/P} + m_1$$
$$x_2 = \text{I/P}$$

At initial stage FF is '0'.

Output of encoder

Message	FF	x_1	x_2
1	0	1	1
0	1	1	0
1	0	1	1
1	1	0	1
1	1	0	1
-	1		

$$m = 1$$
$$\therefore \quad x_1 = \text{I/P} + \text{FF}$$
$$= 1 \oplus 0 = 1$$

$m_1 \rightarrow$ previous state

$$x_2 = 1$$
$$\because m = 1$$

$\oplus \rightarrow$ Modulo addition

$$1 \oplus 0 = 1$$
$$0 \oplus 1 = 1$$
$$0 \oplus 0 = 0$$
$$1 \oplus 1 = 0$$

$m = 0$

$$x_1 = \text{I/P} + \text{FF}$$
$$= 0 + 1 = 1$$
$$x_1 = 1$$
$$x_2 = 0$$

$m = 1$

$$x_1 = \text{I/P} + \text{FF}$$
$$= 1 + 0 = 1$$
$$x_1 = 1$$
$$x_2 = 1$$

$m = 2$

$$x_1 = \text{I/P} + \text{FF}$$
$$= 1 + 1 = 0$$
$$x_1 = 0$$
$$x_2 = 1$$

$m = 1$

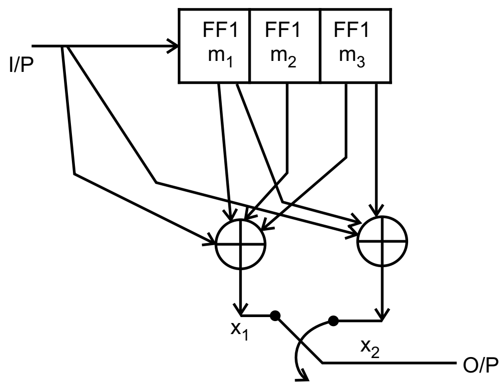
$$x_1 = \text{I/P} + \text{FF}$$
$$= 1 + 1 = 0$$
$$x_1 = 0$$
$$x_2 = 1$$

Solved Problem 3.15 *Draw the encoder for a rate* $\frac{1}{2}$ *constraint length* $K = 4$ *&* $g^{(1)} = (1, 1, 1, 1)$ *&* $g^{(2)} = (1, 1, 0, 1)$ *respectively. Find the coded output produced by the message sequence 10111... Using transform domain approach.*

Solution:

No. of flip flops $(M) = k - 1 = 4 - 1 = 3$

Number of adders = 2



Encoder

$$x_1 = \text{I/P} + m_1 + m_2 + m_3$$
$$x_2 = \text{I/P} + m_1 + m_3$$

Transform domain approach:

$$g^{(1)}(x) = 1 + x + x^2 + x^3$$
$$g^{(2)}(x) = 1 + x + x^3$$
$$\because \quad g^{(1)} = (1, 1, 1, 1)$$

Generator polynomials are $g^{(1)}$ & $g^{(2)}$.

Message polynomial is $m(x) = 1 + x^2 + x^3 + x^4 + ...$

Hence,

$$\begin{aligned} C^{(1)}(x) &= g^{(1)}(x)\, m(x) \\ &= \left(1 + x + x^2 + x^3\right)\left(1 + x^2 + x^3 + x^4\right) \\ &= 1 + x^2 + x^3 + x^4 + x + x^3 + x^4 + x^5 + x^2 + x^4 + x^5 + x^6 \\ &\qquad + x^3 + x^5 + x^6 + x^7 \end{aligned}$$

$$C^{(1)}(x) = 1 + x^4 + x^3 + x^5 + x^7 + x$$
$$C^{(2)}(x) = g^{(2)}(x)\, m(x)$$
$$= \left(1 + x + x^3\right)\left(1 + x^2 + x^3 + x^4 + ...\right)$$
$$= 1 + x^2 + x^3 + x^4 + x + x^3 + x^4 + x^5 + x^3 + x^5 + x^6 + x^7 + ...$$
$$C^{(2)}(x) = 1 + x + x^2 + x^3 + x^6 + x^7 + ...$$

By multiplexing the two output sequences of $C^{(1)}$ & $C^{(2)}$.

$$C^{(1)} = 1\ 1\ 0\ 1\ 1\ 1\ 0\ 1$$
$$C^{(2)} = 1\ 1\ 1\ 1\ 0\ 0\ 1\ 1$$

$\therefore$ The encoder output is 11, 11, 01, 11, 10, 10, 01, 11

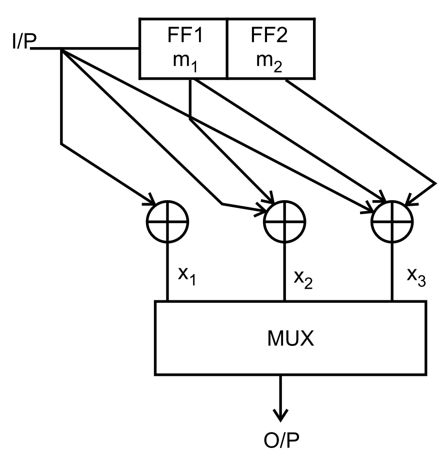
1 0 1 1 1

Solved Problem 3.16 *Construct a convolutional encoder whose constraint length (k) is 3 & has 3 modulo-2 adders and an multiplexer. Generator sequence of the encoder are*

$$g^{(1)} = (1, 0, 1)$$
$$g^{(2)} = (1, 1, 0)$$
$$g^{(3)} = (1, 1, 1)$$

Find the encoder output produced by the message sequence 10111.... Verify the codeword using algorithm.

Solution:



Number of flip flop's $(m) = K - 1 = 3 - 1 = 2$
Number of adders = 3

$$\because \; g^{(1)} = (1, 0, 1)$$
$$g^{(2)} = (1, 1, 0)$$
$$g^{(3)} = (1, 1, 1)$$

$$x_1 = \text{I/P} + m_2$$
$$x_2 = \text{I/P} + m_1$$
$$x_3 = \text{I/P} + m_1 + m_2$$

Using Algorithm:

1. Write the message polynomial

$$m(D) = \left(1 + D^2 + D^3 + D^4\right)$$

(or)

$$m(x) = 1 + x^2 + x^3 + x^4 \qquad \because m(10111)$$

2. Write the generator polynomial of each sequence

$$g^1(x) = \left(1 + x^2\right)$$
$$g^2(x) = 1 + x$$
$$g^3(x) = 1 + x + x^2$$

3. Compute the code polynomial

$$C^1(x) = m(x)\, g^{(1)}(x)$$
$$= \left(1 + x^2 + x^3 + x^4\right)\left(1 + x^2\right)$$
$$= 1 + x^2 + x^2 + x^4 + x^3 + x^5 + x^4 + x^6 \qquad \left(\because x^2 + x^2 = 0\right)$$
$$C^1(x) = 1 + x^3 + x^5 + x^6$$
$$C^1(x) = 1001011$$

$$C^2(x) = m(x)\, g^{(2)}(x)$$
$$= \left(1 + x^2 + x^3 + x^4\right)(1 + x)$$
$$= 1 + x + x^2 + x^3 + x^3 + x^4 + x^4 + x^5$$
$$C^2(x) = 1 + x + x^2 + x^5$$
$$C^2(x) = 111001$$

$$C^3(x) = m(x)\,g^{(3)}(x)$$
$$= \left(1 + x^2 + x^3 + x^4\right)\left(1 + x + x^2\right)$$
$$= 1 + x + x^2 + x^2 + x^3 + x^4 + x^3 + x^4 + x^5 + x^4 + x^5 + x^6$$
$$C^3(x) = 1 + x + x^4 + x^6$$
$$C^3(x) = 1100101$$

4. Multiplex the code polynomial to get code sequence

$$\begin{array}{lcccccccc} C^1(x) & = & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ C^2(x) & = & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ C^3(x) & = & 1 & 1 & 0 & 0 & 1 & 0 & 1 \end{array}$$

$C^2(x)$ length is not equal to $C^1(x)$ & $C^3(x)$ so append zero to make it equal length.

$C^1(x)$	=	1	0	0	1	0	1	1
$C^2(x)$	=	1	1	1	0	0	1	0
$C^3(x)$	=	1	1	0	0	1	0	1

$$\text{code sequences are} = \left(111, 011, 010, 100, 001, \underbrace{110, 101}_{\text{tail bits}}\right)$$

$$\text{message} = 1\ 0\ 1\ 1\ 1$$

Solved Problem 3.17 *Construct a convolutional encoder with the following specifications*

(i) Constraint length is 3

(ii) Code rate is $\frac{1}{2}$

(iii) Generator sequence are $g^{(1)}(1,1,1)$ *&* $g^{(2)}(1,0,1)$.

(iv) The input sequence is 10011

(v) Draw the following

(a) Code tree

(b) Trellis

(c) State diagram

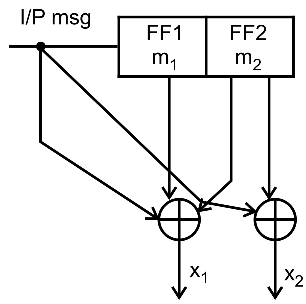
Solution:

Constraint length $K = 3$, code word $\frac{K}{n} = \frac{1}{2}$.

no. of flip flops $= K - 1 = 3 - 1 = 2$

$n = 2$; No. of adders $= 2$

Encoder Diagram

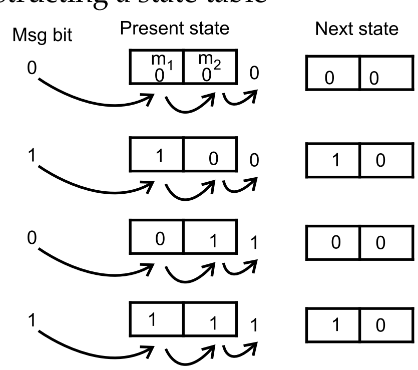


$$g^{(1)} = 1 + x + x^2$$
$$g^{(2)} = 1 + x^2$$
$$x_1 = \text{Input} \oplus m_1 \oplus m_2$$
$$x_2 = \text{Input} \oplus m_2$$

Time domain approach: Find the output of encoder:

Msgbits	State		x_1	x_2	$x_1\ x_2$
	FF_1	FF_2	I/P $\oplus m_1 \oplus m_2$	I/P $\oplus m_2$	
Initial	0	0			
1	0	0	1⊕0⊕0=1	1⊕0=1	1 1
0	1	0	0⊕1⊕0=1	0⊕0=0	1 0
0	0	1	0⊕0⊕1=1	0⊕1=1	1 1
1	0	0	1⊕0⊕0=1	1⊕0=1	1 1
1	1	0	1⊕1⊕0=0	1⊕0=1	0 1

State Table: constructing a state table



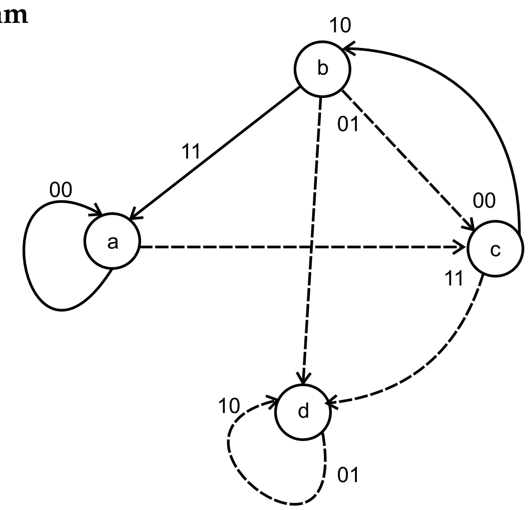
msg bit	Present state	Next state	o/p x_1 I/P ⊕ m_1 ⊕ m_2	x_2 I/P ⊕ m_2	
0	00-a	00-a	0	0	a(00) a 0
1	00-a	10-c	1	1	c(11)
0	01-b	00-a	1	1	a(11) b 0
1	01-b	10-c	0	0	c(00)
0	10-c	01-b	1	0	a(10) c 0
1	10-c	11-d	0	1	c(01)
0	11-d	01-b	0	1	a(01) d 0
1	11-d	11-d	1	0	c(10)

Output is calculated by present state not next state.

Ex:

Msg	P.S	x_1	x_2
0	00	0⊕0⊕0=0	0⊕0=0
1	00	1⊕0⊕0=1	1⊕0=1
0	01	0⊕0⊕1=1	0⊕1=1

State diagram



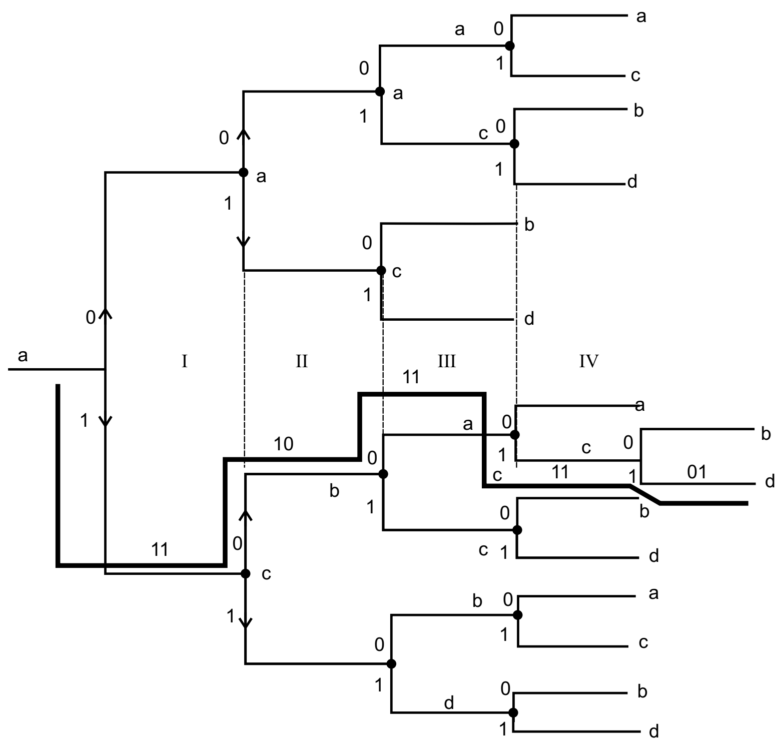
If message '1' → dotted line.

If message '0' → solid line.

Code Tree:

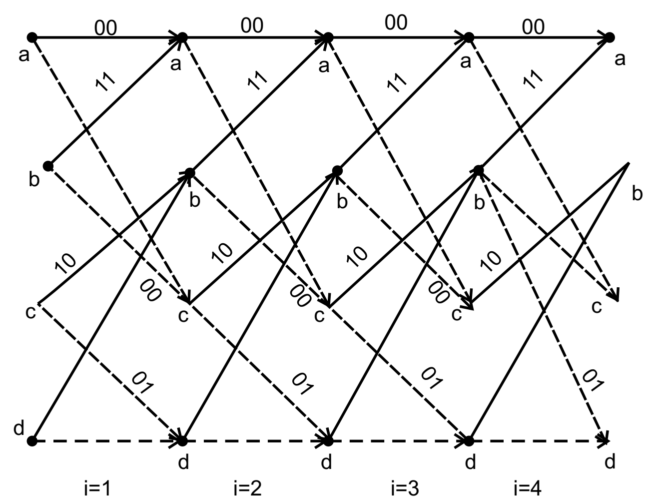
$$\text{Trellis} = 2^{(K-1)} = 2^{3-1} = 2^2 = 4 = 4 \text{ states}$$

Code tree $= M$; no. of branches.



Msg bits:	1	0	0	1	1
Tracing path:	a-c	c-b	b-a	a-c	c-d
Encoded bits:	11	10	11	11	01

Trellis diagram



3.15.5 Viterbi decoding

- Viterbi algorithm used to decode the convolution codes.
- It is also said to be maximum likelihood decoding.

Metric

It is the discrepancy between the received signal 'Y' and the decoded signal at particular node.

Surviving path

- This is the path of the decoded signal with minimum metric.
- A metric is assigned to each surviving path
- Metric of a particular path is obtained by adding individual metric on the nodes along that path.
- Y is decoded as the surviving path with smallest metric.

$$\text{Surviving paths} = 2^{(K-1)k}$$

$K \rightarrow$ Constraint length
$k \rightarrow$ No. of message bits

Decoding algorithm

Step 1: Draw code trellis diagram.

Step 2: Find metric of each branch and add all the metric.

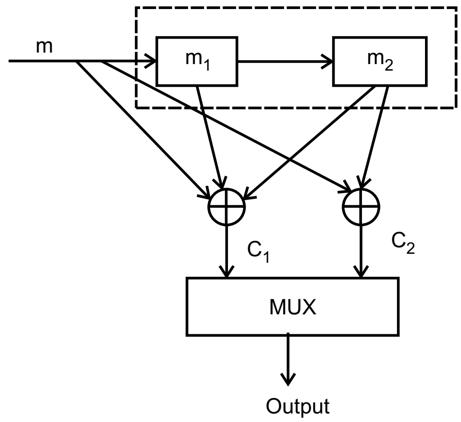
Step 3: Choose a node with lower metric.

Step 4: If two path exist with lower metric any one can be taken.

Step 5: Maximum likelihood decoding path is determined and message are calculated.

Solved Problem 3.18 *Example for viterbi decoding:-*

Decode the message 110111 for the convolutional encoder shown below.



Solution:

Given; Number of shift register $M = 2$

Number of output bits $= 2 = n$

- Constraint length $\Rightarrow K = M + 1 = 2 + 1 = 3$
- Code rate $r = \frac{k}{n} = \frac{1}{2}$
- Generator sequences are,

$$C_1 = m \oplus m_1 \oplus m_2$$
$$C_2 = m \oplus m_2$$

From C_1 & C_2

$$g_1 = (1, 1, 1)$$
$$g_2 = (1, 0, 1)$$

State Table:

Msgbits m	presentstate m_1 m_2	Nextstate	C_1 $m \oplus m_1 \oplus m_2$	C_2 $m \oplus m_2$	State diagram
0	0 0 a	0 0 a	0	0	a: 0 → a(00)
1	0 0 a	1 0 c	1	1	1 → c(11)
0	0 1 b	0 0 a	1	1	b: 0 → a(11)
1	0 1 b	1 0 c	0	0	1 → c(00)
0	1 0 c	0 1 b	1	0	c: 0 → b(10)
1	1 0 c	1 1 d	0	1	1 → d(01)
0	1 1 d	0 1 b	0	1	d: 0 → b(01)
1	1 1 d	1 1 d	1	0	1 → d(10)

Decoding

Step 1: Draw code trellis diagram

- Decode the message $Y = 1\ \ 1\ \ 0\ \ 1\ \ 1\ \ 1$ divide the sequence Y into 'n' number of sequence.

$$n = 2; \qquad Y = 1\,1 \quad | \quad 0\,1 \quad | \quad 1\,1 \qquad \Rightarrow \quad Y = 11$$

- The above received bits represent the outputs for three successive message bits, since for a single input bits, the encoder transmits two bits (C_1, C_2).
- In the code trellis diagram shown below if the current state is 'a', then next state is 'a_1' or 'b_1'.

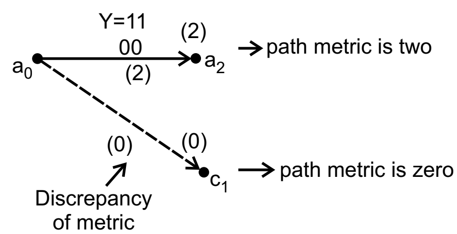


Fig .3.21 Viterbi decoder for first message bit

- The branch a_0-a_1 represents the decoded signal 00, discrepancy between the received signal and decoded signal is two. Hence metric of that branch is two.
- The branch a_0-C_1 represents the decoded signal 11. Since the received signal and the decoded signal along this branch are same, there is no discrepancy here and the metric of this branch is zero.
- The number which is encircled near a node indicates the path metric reaching to the node.

Step 2:

- From nodes a_1 & c_1, four next states a_2, b_2, c_2 & d_2 are possible for the next set of received bits $Y = 01$.
- The below figure shows the decoded outputs and branch metric for those branches.
- The encircled members at a_2, b_2, c_2 & d_2 indicate the corresponding path metrics emerging from a_0.

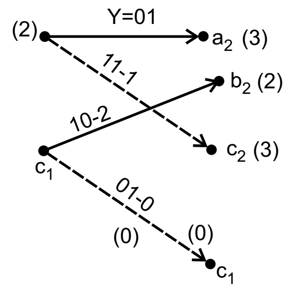


Fig .3.22 Viterbi decoder for second message bit

Step 3: $Y = 11$

- The below figure shows the nodes a_3, b_3, c_2 & d_3 with their corresponding path metrics.

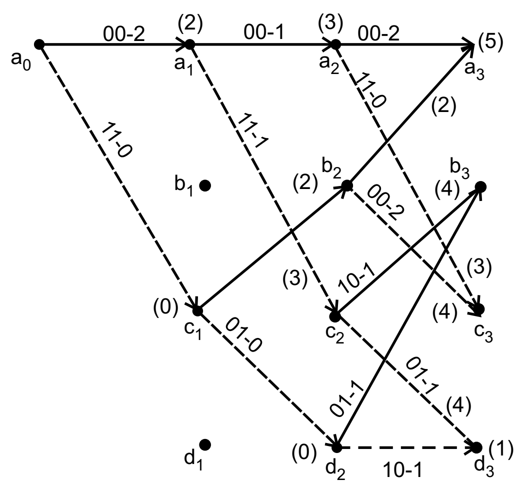


Fig .3.23 Viterbi decoder for third message bit

- Two paths exist at nodes a_3. One path is $a_0c_1b_2a_3$ with metric '2' and other path is $a_0a_1a_2a_3$ with metric '5'. Similarly two paths exist at other nodes also.
- But only one path with lower metric should be retained and the path with higher metric which is eliminated.
- This procedure is continued for other nodes also.

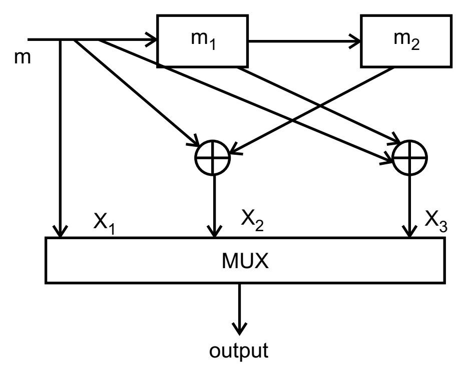
  Shortest path

  1. a_0 - c_1 - d_2 - b_3 with metric '1'
  2. a_0 - c_1 - d_2 - d_3 with metric '1'

– Two shortest path with same metric is obtained. We can take any one path to decode the received message.

Shortest path:	a_0-c_1-d_2-b_3		
Received sequence Y:	11	01	11
Decode output:	11	01	01
Detected message:	1	1	0
Error status:	No	No	Yes
Corrected message:	1	1	1

Solved Problem 3.19 *Decode the message 110101001 for the convolutional encoder shown below.*



Solution:

Given; Number of shift register $M = 2$

Number of output bits $= 3 = n$

- Constraint length

$$K = M + 1$$
$$K = 2 + 1$$
$$K = 3$$

- Code rate

$$r = \frac{K}{n} = \frac{1}{2}$$

- Generator sequences are,

$$X = m$$
$$X_2 = m \oplus m_2$$
$$X_3 = m \oplus m_1$$

From X_1, X_2 & X_3

$$g_1 = (1, 0, 0)$$
$$g_2 = (1, 0, 1)$$
$$g_3 = (1, 1, 0)$$

State Table:

Msgbits	presentstate			Nextstate			$X_1 =$	$X_2 =$	$X_3 =$	
m	m_1	m_2					m	$m \oplus m_2$	$m \oplus m_1$	
0	0	0	a	0	0	a	0	0	0	a: 0 → a(000); 1 → c(111)
1	0	0	a	1	0	c	1	1	1	
0	0	1	b	0	0	a	0	1	0	b: 0 → a(010); 1 → c(101)
1	0	1	b	1	0	c	1	0	1	
0	1	0	c	0	1	b	0	0	1	c: 0 → b(001); 1 → d(110)
1	1	0	c	1	1	d	1	1	0	
0	1	1	d	0	1	b	0	1	1	d: 0 → b(011); 1 → d(100)
1	1	1	d	1	1	d	1	0	0	

Decoding

Step 1: Draw code trellis diagram

– Decode the message $Y = 1\ 1\ 0\ 1\ 0\ 1\ 0\ 0\ 1$

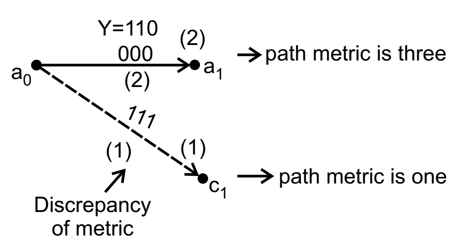


Fig .3.24 Viterbi decoder for first message bit

Divide the sequence Y into 'n' number of sequence.

$$n = 3; \qquad Y = 1\,1\,0 \;\;|\;\; 1\,0\,1 \;\;|\;\; 0\,0\,1 \qquad \Rightarrow \qquad Y = 110$$

Step 2:

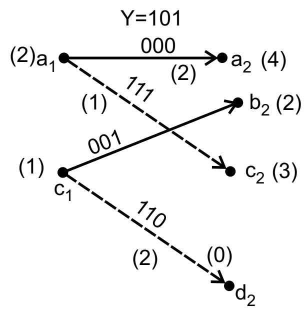


Fig .3.25 Viterbi decoder for second message bit

Step 3: $Y = 001$

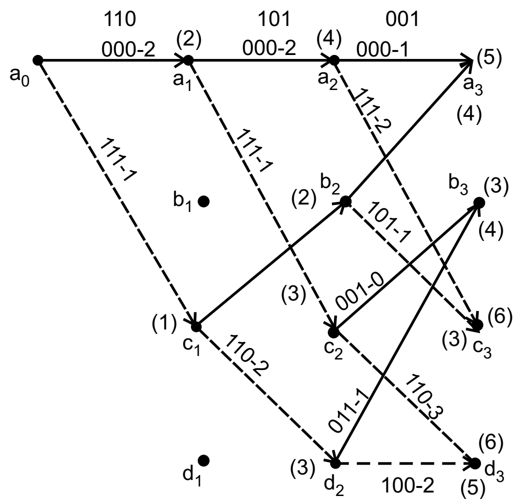


Fig .3.26 Viterbi decoder for third message bit

Shortest path

1. a_0 - a_1 - c_2 - b_3 with metric '3'

2. a_0 - c_1 - b_2 - c_3 with metric '3'

Two shortest path with same metric '3' is obtained. Any one of the path can be taken to decode the message bit.

Shortest path:	a_0-a_1-c_2-b_3		
Received sequence Y:	110	101	001
Decode output:	000	111	001
Detected message:	0	1	0
Error status:	Yes	Yes	No
Corrected message:	1	0	0

3.16 Linear Block Codes

In block codes, each block of 'K' message bits is encoded into a block of 'n' bits ($n > k$). This n bit block is called a codeword. The $(n-k)$ check bits or parity bits are derived from the message bits and added to them.

- When the 'k' message bits appear at the beginning of a codeword, the code is called a systematic code.
- **Non-systematic code:** In non systematic code, it is not possible to identify message bits and check bits they are mixed in the block.

3.16.1 Encoding of linear block codes

Definition

A code is said to be linear if any two code words in the code can be added in modulo-2 arithmetic to produce a third code word in the code.

Dimension of LBC is (n, k), $(n-k)$ number of parity bits and k number of message bits available.

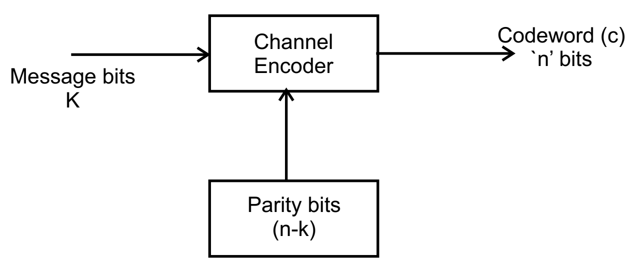


Fig .3.27 Structure of LBC

Word

Word is a sequence of symbols. If the word is a code, it is called a code word.

Hamming weight

Hamming weight of a code word is equal to the number of non-zero components in it.

Code word (c)	Hamming weight $w(c)$
0110100	3
1000001	2

Hamming distance

Hamming distance between two codewords is the number of places the codeword differ.

Example:

$$c_1 = 010101$$
$$c_2 = 101001$$

Hamming distance between c_1 & c_2 $d(c_1, c_2) = 4$

Code rate (r)

Ratio between n & k

$$r = \frac{k}{n}$$

where

$k \rightarrow$ Number of message bits
$n \rightarrow$ Number of output bits

Minimum distance (d*)

The minimum distance of a code is smallest hamming distance between any pair of codewords in the code.

$$d* = \min\left[d\left(c_j, c_k\right)\right], \qquad i \neq \iota$$

Minimum weight

The minimum weight of a code is the smallest weight of any non zero code word and it is denoted by ω^*.

Important formulas

1. Code word

$$c = M : b$$

$M \rightarrow$ message bits
$b \rightarrow$ parity bits

2. Another formula for code word

$$C = MG$$

$G \rightarrow$ generator matrix

3. $\boxed{b = MP}$

$P \rightarrow$ Probability sub matrix (or) sub matrix

4. Generator matrix

$$G = I_K : P_{KXM}$$

K	$\rightarrow$	Number of message bits
m	$\rightarrow$	Number of parity bits
I_K	$\rightarrow$	Identity matrix

5. Parity check matrix

$$H = P^T_{mxk} : I_m$$

3.16.2 Properties of linear block codes (LBC)

Property 1

$$GH^T = 0 \quad (1)$$

W.K.T

$$G = I_k : P_{kxM} \quad (2)$$

&

$$H = P^T_{mxK} : I_m$$

$$H^T = \begin{bmatrix} P_{Kxm} \\ I_k \end{bmatrix} \quad (3)$$

substitute (2) & (3) in (1)

$$(1) \Rightarrow GH^T = [I_K : P_{Kxm}] \begin{bmatrix} P_{Kxm} \\ I_K \end{bmatrix}$$

$$\therefore GH^T = I_K P_{Kxm} \oplus P_{Kxm} I_K$$

$$GH^T = P_{Kxm} \oplus P_{Kxm} = 0$$

Hence $\boxed{GH^T = 0}$ proved.

Property 2

$$G^T H = 0 \quad (4)$$

From (2)

$$G^T = \begin{bmatrix} I_m \\ P_{mxK} \end{bmatrix} \quad (5)$$

$$H = P^T_{mxK} : I_m \quad (6)$$

Substitute (5) & (6) in (4)

$$G^T H = \begin{bmatrix} I_m \\ P_{mxK} \end{bmatrix} [P^T_{mxK} : I_m]$$

$$G^T H = \begin{bmatrix} I_m \\ P_{mxK} \end{bmatrix} \left[P^T_{mxK} : I_m \right]$$

$\boxed{HG^T}$ Hence proved.

property 3

$$CH^T = 0 \tag{7}$$

$$C = MG \tag{8}$$

Multiply Eqn. (8) by H^T on both sides

$$CH^T = MGH^T$$
$$CH^T = M\,(0)$$
$$CH^T = 0$$

$\because GH^T = 0$ since property (1).

Property 4

The sum of two codewords belonging to the code is also a code word belonging to the code.

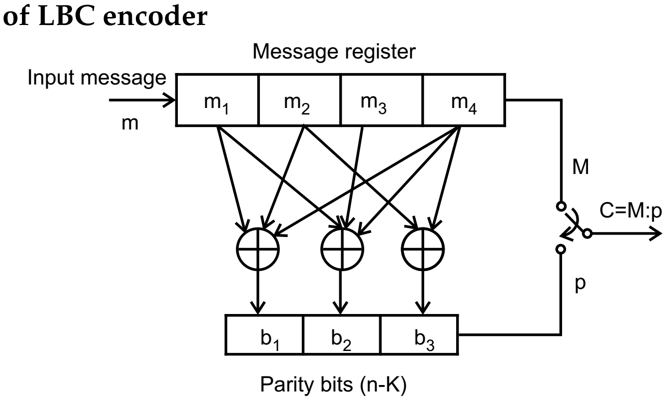
Property 5

The all zero word is always a code word.

Property 6

The minimum distance between two code words of a linear code is equal to the minimum weight of the code.

Structure of LBC encoder



Error detection and error detection

- Number of error can be detected by LBC is 'S';

$$d_{\min} \geq S+1$$
$$S \leq d_{\min}-1$$

- Number of error can be corrected by LBC is 't';

$$d_{\min} \geq 2t+1$$
$$t \leq \frac{d_{\min}-1}{2}$$

Classification of LBC

- Repetition codes
- Hamming codes
- cyclic codes

Advantages of LBC

- Easy to encode and decode
- simple

Disadvantages

- Detect only 2 errors
- correct only 1 error

Hamming codes

For a family of (n, k) linear block code it should satisfies following conditions

1. Block length, $n = 2^m - 1$
2. Number of parity bits $m = n - K$
3. Number of message bits $K = 2^m - m - 1$

   (or)

   $K = n - m$

   where

   n → Number of output bits
   m → Number of parity bits
   K → Number of message bits

When the above three conditions are satisfied, then the code is said to be hamming codes.

Solved Problem 3.20 *A generator matrix for (6, 3) block code is given find all code vectors.*

$$G = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

Solution:

Given that

$$G = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

W.K.T

$$G = I_K : P_{Kxm}$$

$$G = \begin{bmatrix} \underbrace{\begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}}_{I_K} & \underbrace{\begin{matrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{matrix}}_{P_{Kxm}} \end{bmatrix}$$

(i) $(n, K) = (6, 3)$

$$(n - k) = m$$
$$(6 - 3) = 3$$

$m \rightarrow$ Number of parity bits
$n \rightarrow$ Number of output bits
$k \rightarrow$ Number of message bits

$$G = \begin{bmatrix} \underbrace{\begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}}_{I_3} & \underbrace{\begin{matrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{matrix}}_{P_{3\times 3}} \end{bmatrix}$$

$$\text{Probability matrix} = P_{3\times 3} = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

Code word $\boxed{c = m : b}$; $b = m \cdot p$

parity bits 'b' can be find by using the formula

$$\begin{bmatrix} b_0 & b_1 & b_2 \end{bmatrix} = \begin{bmatrix} m_0 & m_1 & m_2 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

$$b_0 = m_0 \oplus m_1$$
$$b_1 = m_1 \oplus m_2$$
$$b_2 = m_0 \oplus m_2$$

code vectors can be written as

m_0	m_1	m_2	$b_0 = m_0 \oplus m_1$	$b_1 = m_1 \oplus m_2$	$b_2 = m_0 \oplus m_2$	$c = m : p$
0	0	0	0	0	0	0 0 0 0 0 0
0	0	1	0	1	1	0 0 1 0 1 1
0	1	0	1	1	0	0 1 0 1 1 0
0	1	1	1	0	1	0 1 1 1 0 1
1	0	0	1	0	1	1 0 0 1 0 1
1	0	1	1	1	0	1 0 1 1 1 0
1	1	0	0	1	1	1 1 0 0 1 1
1	1	1	0	0	0	1 1 1 0 0 0

Solved Problem 3.21 *A generator matrix of a particular (7, 4) linear block code is given by*

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

1. *Find the parity check matrix (H)*
2. *List all the code vectors*
3. *What is the minimum distance between code vectors.*
4. *How many errors can be detected? How many errors can be corrected?*

Solution:

$$(n, k) = (7, 4)$$
$$(n - k) = m$$
$$(7 - 4) = 3$$

$m \quad \rightarrow$ Number of parity bits

$n = 7 \quad \rightarrow$ Number of output bits

$k = 4 \quad \rightarrow$ Number of message bits

W.K,.T

$C = M : b$ (code word)
$b = MP$ ($b \rightarrow$ parity bits (i.e.,) $b_0 b_1 b_2$)

$$G = \begin{bmatrix} \underbrace{\begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}}_{\substack{I_K \\ I_4}} & \underbrace{\begin{matrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{matrix}}_{\substack{P_{K \times m} \\ P_{4\times 3}}} \end{bmatrix}$$

$$P = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$b = M \times P$$

$$\begin{bmatrix} b_0 & b_1 & b_2 \end{bmatrix} = \begin{bmatrix} m_0 & m_1 & m_2 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$b_0 = M_0 \oplus M_1 \oplus M_3$$
$$b_1 = M_1 \oplus M_2 \oplus M_3$$
$$b_2 = M_2 \oplus M_2 \oplus M_3$$

Error detected (S)

$$d_{\min} \geq S + 1$$
$$3 \geq S + 1$$
$$S \leq 2$$

Error corrected (t)

$$d_{\min} \geq 2t + 1$$
$$t \leq 1$$

M_0	M_1	M_2	M_3	$b_0 = M_0 \oplus M_2 \oplus M_3$	$b_1 = M_0 \oplus M_1 \oplus M_3$	$b_2 = M_1 \oplus M_2 \oplus M_3$	C= M :b	Hamming code H(ω)	Hamming distance d
0	0	0	0	0	0	0	0000000	0	4
0	0	0	1	1	1	1	0001111	4	3
0	0	1	0	1	0	1	0010101	3	4
0	0	1	1	0	1	0	0011010	3	4
0	1	0	0	0	1	1	0100011	3	4
0	1	0	1	1	0	0	0101100	3	3
0	1	1	0	1	1	0	0110110	4	4
0	1	1	1	0	0	1	0111001	4	7
1	0	0	0	1	1	0	1000110	3	4
1	0	0	1	0	0	1	1001001	3	3
1	0	1	0	0	1	1	1010011	4	4
1	0	1	1	1	0	0	1011100	4	4
1	1	0	0	1	0	1	1100101	4	4
1	1	0	1	0	1	0	1101010	4	3
1	1	1	0	0	0	0	1110000	3	4
1	1	1	1	1	1	1	1111111	7	

Solved Problem 3.22 *The parity check matric of a particular (7, 4) linear block code is given by*

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

1. *Find the generator matrix* $[G]$
2. *List all the code vectors*
3. *What is the minimum distance between code vectors?*
4. *How many errors can be detected? How many errors can be corrected?*

Solution:

Dimension (n, k); $(7, 4)$

$$n = 7 \quad \& K = 4$$

$$n - k = m$$

$$7 - 4 = 3$$

$m = 3$; Number of parity bits $(b_0 \;\; b_1 \;\; b_2)$

$k \rightarrow$ Number of message bits $(M_0 \;\; M_1 \;\; M_2 \;\; M_3)$

W.K.T

$C = M : b$ ($C \rightarrow$ code word)

$b = MP$ ($b \rightarrow$ parity bits)

$$G = \begin{bmatrix} \underbrace{\begin{matrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{matrix}}_{\substack{P^T_{m \times K} \\ P^T_{3 \times 4}}} & \underbrace{\begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}}_{\substack{P_{I_m} \\ I_3}} \end{bmatrix}$$

$$P^T_{3\times 4} = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}$$

$$P_{4\times 3} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

$$\therefore \quad b = M \times P$$

$$\begin{bmatrix} b_0 & b_1 & b_2 \end{bmatrix} = \begin{bmatrix} M_0 & M_1 & M_2 & M_3 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

M_0	M_1	M_2	M_3	$b_0 = M_0 \oplus M_2 \oplus M_3$	$b_1 = M_0 \oplus M_1 \oplus M_3$	$b_2 = M_1 \oplus M_2 \oplus M_3$	C= M :b	Hamming code H(ω)	Hamming distance d
0	0	0	0	0	0	0	0000000	0	3
0	0	0	1	0	1	1	0001011	3	4
0	0	1	0	1	0	1	0010101	3	3
0	0	1	1	1	1	0	0011110	4	3
0	1	0	0	1	1	0	0100110	3	3
0	1	0	1	1	0	1	0101101	4	3
0	1	1	0	0	1	1	0110011	4	3
0	1	1	1	0	0	0	0111000	3	7
1	0	0	0	1	1	1	1000111	4	3
1	0	0	1	1	0	0	1001100	3	4
1	0	1	0	0	1	0	1010010	3	3
1	0	1	1	0	0	1	1011001	4	3
1	1	0	0	0	0	1	1100001	3	3
1	1	0	1	0	1	0	1101010	4	4
1	1	1	0	1	0	0	1110100	4	3
1	1	1	1	1	1	1	1111111	7	

$$b_0 = M_0 \oplus M_1 \oplus M_2$$
$$b_1 = M_0 \oplus M_1 \oplus M_3$$
$$b_2 = M_0 \oplus M_2 \oplus M_3$$

$d_{\min} = 3$

Error detected (S)

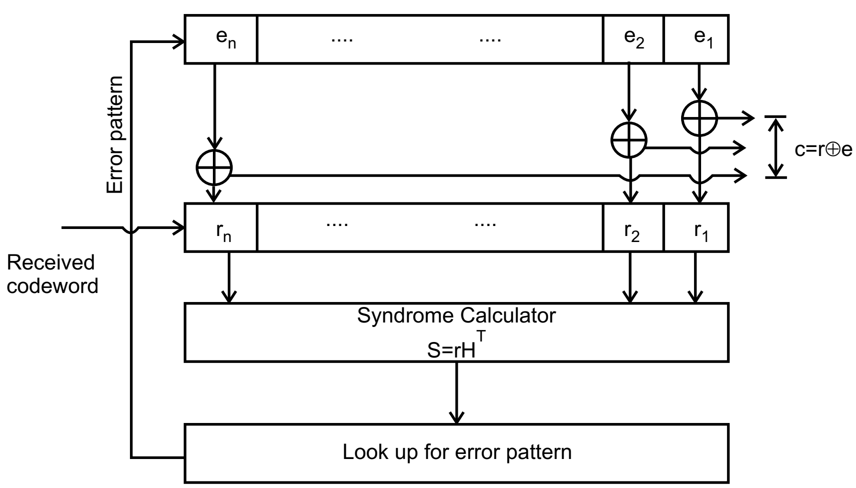
$$d_{\min} \geq S + 1$$
$$3 \geq S + 1$$
$$S \leq 2$$

Error corrected (t)

$$d_{\min} \geq 2t + 1$$
$$t \leq 1$$

3.16.3 Decoding of Linear block codes

Block diagram of syndrome decoder



Definition

The non zero output of the product rH^T is called syndrome (S).

$$\boxed{S = rH^T}$$

- Syndrome is used to know the information about error pattern.
- $r \rightarrow$ received data

  $H^T \rightarrow$ transpose of parity check matrix
- Dimension of syndrome is $1 \times (n-k)$

Properties of syndrome

Property 1: Syndrome depends only on error pattern not on code word.

$$\begin{aligned} S &= rH^T \\ r &= c \oplus e \\ S &= cr \oplus eH^T \\ S &= cH^T \oplus eH^T \\ S &= eH^T \end{aligned}$$

Property 2: All error pattern that differ by code word have same syndrome.

Decoding algorithm

1. Compute parity check matrix

$$H = P^T_{m \times k} : I_m$$

2. Get the received vector 'r'
3. Compute the syndrome $S = rH^T$
4. Construct the decoding table
5. Locate the error pattern 'e' corresponding to the syndrome by referring the decoding table.
6. Add the error with received vector to correct the error.

Solved Problem 3.23 *The generator matrix of a (6, 3) systematic block code is given by*

$$G = \begin{bmatrix} 1 & 0 & 0 & & 1 & 1 & 1 \\ 0 & 1 & 0 & & 1 & 1 & 0 \\ 0 & 0 & 1 & & 0 & 1 & 1 \end{bmatrix}$$

1. *Construct parity check matrix (H)*
2. *Prepare a decoding table*

3. *Decode the received sequence 101101 and find message bit.*

Solution:

Step-1: Compute parity check matrix (H)

$$H = P^T_{m\times k} : I_m \qquad ; \qquad G = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

Given; $(n, k) = (6, 3)$

W.K.T $(n - k) = m$

$(6 - 3) = 3$

$$G = \begin{bmatrix} \underbrace{\begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}}_{I_3} & \underbrace{\begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{matrix}}_{P_{3\times 3}} \end{bmatrix} \qquad \text{since } G = I_k : P_{k\times m}$$

$$\therefore \quad P = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}; \qquad P^T = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

'H' matrix can be obtained by

$$H = P^T_{m\times k} : I_m$$

$$H = P^T_{3\times 3} : I_3$$

$$H = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}; \; H^T = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Step-2: Get the received vector 'r'

$$r = 1\ 0\ 1\ 1\ 0\ 1$$

Step-3: Compute syndrome

$$S = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \left[\; 1 \oplus 0 \oplus 0 \oplus 1 \oplus 0 \oplus 0 \quad 1 \oplus 0 \oplus 1 \oplus 0 \oplus 0 \oplus 0 \quad 1 \oplus 0 \oplus 1 \oplus 0 \oplus 0 \oplus 1 \; \right]$$

$$S = \left[\; 0 \quad 0 \quad 1 \; \right]$$

Step-4: Construct decoding table

where, $n = 6$

Decoding Table

e_1	e_2	e_3	e_4	e_5	e_6	S_0	S_1	S_2
1	0	0	0	0	0	1	1	1
0	1	0	0	0	0	1	1	0
0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	1	0	0
0	0	0	0	1	0	0	1	0
0	0	0	0	0	1	0	0	1

Step-5: Error pattern for respective syndrome (e) =0 0 0 0 0 1

Step-6: Add error with received vector to correct the error.

$$C = e + r$$

$$\begin{array}{ccccc} C & = & 000001 & \rightarrow & e \\ & & 101101 & \rightarrow & r \\ \hline C & = & 101100 & & \end{array}$$

Decoded code word $C = \underbrace{1 \quad 0 \quad 1} \quad \underbrace{1 \quad 0 \quad 0}$

Message bits = 101

Parity bits = 100

Solved Problem 3.24 *For systematic linear block code, 3 parity bits b_0, b_1, b_2 are*

$$b_0 = M_0 \oplus M_1 \oplus M_2$$
$$b_1 = M_0 \oplus M_1$$
$$b_2 = M_0 \oplus M_2$$

1. *Construct & generator matrix*

2. *Construct code generated by this matrix*
3. *Determine error correcting capabilities*
4. *prepare a decoding table*
5. *decode 000110*

Solution:

Step-1: To construct generator matrix

$$C = M \cdot P$$
$$C_{1\times m} = M_{1\times k} \cdot P_{k\times m}$$

Message bits M_0, M_1 & M_2

& parity bits b_0, b_1 & b_2

So number of parity bits $= 3\,(m = 3)$

number of message bits $= 3\,(k = 3)$

$$C_{1\times 3} = M_{1\times 3} \cdot P_{3\times 3}$$

$$\begin{bmatrix} b_0 & b_1 & b_2 \end{bmatrix}_{1\times 3} = \begin{bmatrix} M_0 & M_1 & M_2 \end{bmatrix}_{1\times 3} \begin{bmatrix} P_{11} & P_{12} & P_{13} \\ P_{21} & P_{22} & P_{23} \\ P_{31} & P_{32} & P_{33} \end{bmatrix}_{1\times 3}$$

$$b_0 = M_0P_{11} \oplus M_1P_{21} \oplus M_2P_{31}$$
$$b_1 = M_0P_{12} \oplus M_1P_{22} \oplus M_2P_{32}$$
$$b_2 = M_0P_{13} \oplus M_1P_{23} \oplus M_2P_{33}$$

Comparing with the given equation

$$b_0 = M_0 \oplus M_1 \oplus M_2 = 1\ 1\ 1$$
$$b_1 = M_0 \oplus M_1 = 1\ 1\ 0$$
$$b_2 = M_0 \oplus M_2 = 1\ 0\ 1$$

$$P = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

Compute generator matrix

$$G = \begin{bmatrix} I_k & : & P_{k\times m} \end{bmatrix}$$
$$G = I_3 : P_{3\times 3}$$
$$G = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

Step-2: To obtain code vectors

M_0	M_1	M_2	$b_0 = M_0 \oplus M_1 \oplus M_2$	$b_1 = M_0 \oplus M_1$	$b_2 = M_1 \oplus M_2$	Codevectors (C)	Hamming code H(ω)	Hamming distance d
0	0	0	0	0	0	000000	0	
0	0	1	1	0	1	001101	3	3
0	1	0	1	1	0	010110	3	4
0	1	1	0	1	1	011011	4	3
1	0	0	1	1	1	100111	4	4
1	0	1	0	1	0	101010	3	3
1	1	0	0	0	1	110001	3	4
1	1	1	1	0	0	111100	4	3

$d_{\min} = 3$

Step-3: To obtain error detecting & correcting capabilities.

Error detected

$$d_{\min} \geq S + 1$$
$$3 \geq S + 1$$
$$S \leq 2$$

Error corrected

$$d_{\min} \geq 2t + 1$$
$$3 \geq 2t + 1$$
$$t \leq 1$$

Step-4 To prepare a decoding table

Parity check matrix

$$H = \left[\ P^T_{m \times K} \ :\ I_m\ \right]$$

$$\therefore\ H = \begin{bmatrix} \underbrace{\begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{matrix}}_{P^T_{3\times 3}} & \underbrace{\begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}}_{I_3} \end{bmatrix}; \qquad P = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

$$P^T = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

$$H^T = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Syndrome vector $S = rH^T$ can be calculated from H^T & r

  Given:

$$r = 0\,0\,0\,1\,1\,0$$

$$\therefore\ S = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$S = \begin{bmatrix} 1 & 1 & 0 \end{bmatrix}$$

Decoding table ($n = 6$)

e_1	e_2	e_3	e_4	e_5	e_6	S_1	S_2	S_3
1	0	0	0	0	0	1	1	1
0	1	0	0	0	0	1	1	0
0	0	1	0	0	0	1	0	1
0	0	0	1	0	0	1	0	0
0	0	0	0	1	0	0	1	0
0	0	0	0	0	1	0	0	1

Error pattern for respective syndrome

$$S = \begin{bmatrix} 1 & 1 & 0 \end{bmatrix};\ e = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Step-5: Add received vector (r) with error pattern to find correct code word.

$$\begin{array}{rcccc} C & = & 000110 & \rightarrow & r \\ & & \underline{010000} & \rightarrow & e \\ C & = & \underline{010110} & & \end{array}$$

$$\text{Corrected code word } C = \underbrace{0 \quad 1 \quad 0} \quad \underbrace{1 \quad 1 \quad 0}$$

3.17 Hamming codes

For a family of (n, k) linear block code is said to be hamming code if it is satisfies the below three condition.

1. Block length $n = 2^m - 1$
2. Number of parity bits $m = n - k$
3. Number of message bits $k = 2^m - m - 1$ (or) $k = n - m$

Solved Problem 3.25 *The parity check matrix (7, 4) of linear block code is given by*

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

1. *Prove that this linear block codes is hamming code.*
2. *Find generator (G) and list out all the code vectors.*
3. *What is minimum distance, hamming distance and hamming weight.*
4. *How many errors can be detected & corrected.*

Solution:

Given

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

$$(n, k) = (7, 4)$$

$$n = 7$$

$$k = 4$$

$$m = n - k = 7 - 4 = 3$$

$$m = 3$$

1. To prove given LBC as Hamming code,

   (1) Block length

$$\begin{aligned} n &= 2^m - 1 \\ 7 &= 2^3 - 1 \\ 7 &= 7 \end{aligned}$$

   (2) Number of parity bits

$$\begin{aligned} m &= n - k \\ m &= 7 - 4 = 3 \\ 3 &= 3 \end{aligned}$$

   (3) Number of message bits

$$\begin{aligned} k &= 2^m - m - 1 \\ 4 &= 2^3 - 3 - 1 \\ 4 &= 4 \end{aligned}$$

   Above conditions are satisfied hence given LBC proved as Hamming code.

2. To compute generator matrix

$$G = \begin{bmatrix} I_k & : & P_{k\times m} \end{bmatrix}$$

$$H = \begin{bmatrix} \underbrace{\begin{matrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{matrix}}_{P^T_{m\times k}} & \underbrace{\begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}}_{I_m} \end{bmatrix}$$

$$P^T = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}; P = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

$$\therefore\ G = \begin{bmatrix} I_4 & : & P_{4\times 3} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

$$\therefore\ G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

To find code vectors

$$C = M : b$$
$$b = MP$$
$$\begin{bmatrix} b_0 & b_1 & b_2 \end{bmatrix} = \begin{bmatrix} M_0 & M_1 & M_2 & M_3 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$
$$b_0 = M_0 \oplus M_1 \oplus M_2$$
$$b_1 = M_0 \oplus M_1 \oplus M_3$$
$$b_2 = M_0 \oplus M_2 \oplus M_3$$

Code vectors

Message bits				Parity bits			Code vectors	Hamming code H(ω)	Hamming distance d
M_0	M_1	M_2	M_3	$b_0 = M_0 \oplus M_2 \oplus M_3$	$b_1 = M_0 \oplus M_1 \oplus M_3$	$b_2 = M_1 \oplus M_2 \oplus M_3$			
0	0	0	0	0	0	0	0000000	0	3
0	0	0	1	0	1	1	0001011	3	4
0	0	1	0	1	0	1	0010101	3	3
0	0	1	1	1	1	0	0011110	4	3
0	1	0	0	1	1	0	0100110	3	3
0	1	0	1	1	0	1	0101101	4	4
0	1	1	0	0	1	1	0110011	4	3
0	1	1	1	0	0	0	0111000	3	7
1	0	0	0	1	1	1	1000111	4	3
1	0	0	1	1	0	0	1001100	3	5
1	0	1	0	0	1	0	1010010	3	3
1	0	1	1	0	0	1	1011001	4	3
1	1	0	0	0	0	1	1100001	3	3
1	1	0	1	0	1	0	1101010	4	4
1	1	1	0	1	0	0	1110100	4	3
1	1	1	1	1	1	1	1111111	7	

- In LBC the minimum distance = minimum hamming weight

$$3 = d_{\min}$$

- To detect error:

$$\begin{aligned} d_{\min} &\geq S+1 \\ S &\leq 3-1 \\ S &\leq 2 \end{aligned}$$

- To correct error:

$$\begin{aligned} d_{\min} &\geq 2t+1 \\ t &\leq 1 \end{aligned}$$

Solved Problem 3.26 *The parity check matrix of (7, 4) hamming code is given by*

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Calculate syndrome vector for single bit error.

Solution:

Given

$$\begin{aligned} (n,k) &= (7,4) \\ n &= 7;\ k = 4 \\ n-k &= 3 \\ 7-4 &= 3 \end{aligned}$$

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix};\ H^T = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- To find syndrome $S = rH^T$
- Assume a code vector $C = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}$ and also assume received vector as $r = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$

(i.e.,) Error in third bit

$$S = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$S = \begin{bmatrix} 1 & 1 & 0 \end{bmatrix}$$

- Compute decoding table to find error pattern

Error pattern (e) $[n = 7]$							Syndrome place (H^T)		
e_1	e_2	e_3	e_4	e_5	e_6	e_7	S_1	S_2	S_3
1	0	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	1	1	1
0	0	1	0	0	0	0	1	1	0
0	0	0	1	0	0	0	0	1	1
0	0	0	0	1	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0
0	0	0	0	0	0	1	0	0	1

- To obtain correct code word, $C = r \oplus e$

$$\begin{array}{rcll} C & = & 1100100 & \rightarrow r \\ & & 0010000 & \rightarrow e \\ \hline C & = & 1110100 & \end{array}$$

Correct code word $C = \underbrace{1110}_{\text{message bits}} \; \underbrace{100}_{\text{parity bits}}$

Solved Problem 3.27 *The telephone network has bandwidth 3.4 KHz. Calculate the capacity of telephone channel of 30 dB SNR.*

Given data:

B.W = 3.4 Khz

SNR = 30 dB

Solution:

$$\begin{aligned} C &= B \log_2\left(1+\frac{S}{N}\right) \text{ bits/sec} \\ &= 3.4\times 10^3 \log_2\ (1+30) \\ C &= 16.84\times 10^3 \text{ bits/sec} \end{aligned}$$

Solved Problem 3.28 *Calculate the amount of information if the probability of occurrence of the message is 0.25.*

Given Data:

$$\begin{aligned} P &= 0.25 \\ \text{Amount of information} &= \log_2\left(\frac{1}{P_k}\right) \\ &= \log_2\left(\frac{1}{0.25}\right) = \frac{\log_{10} 4}{\log_{10} 2} \\ \text{Amount of information} &= 2\,\text{bits} \end{aligned}$$

Solved Problem 3.29 *An analog signal is band limited to 'B' Hz and sampled at Nyquist rate. The samples are quantised into 4-levels and each level represents one message. Thus there are 4-messages. The probability of these messages are* $P_1 = P_4 = \frac{1}{8}$, $P_3 = P_2 = \frac{3}{8}$. *Find the information rate of the source.*

Given data:

$$\text{Nyquist rate} = 2B \text{ Hz} = r$$

$$P_1 = P_4 = \frac{1}{8} \quad \text{and} \quad P_2 = P_3 = \frac{3}{8}$$

Solution:

Entropy

$$\begin{aligned} H &= -\sum_{K=1}^{4} P_k \log_2 P_k \\ &= -\left[\frac{\left(2\times\frac{1}{8}\log\frac{1}{8}\right)+\left(2\times\frac{3}{8}\log\frac{3}{8}\right)}{\log 2}\right] \\ H &= 1.811 \text{ bits/symbols} \end{aligned}$$

Information rate

$$\begin{aligned} R &= r \cdot H \text{ bits/sec} \\ &= 2B \times H \\ &= 2B \times 1.811 \\ R &= 3.62B \text{ bits/sec} \end{aligned}$$

Solved Problem 3.30 *Consider a AWGN channel with 4 KHz B.W and noise power spectral density is 10^{-12} watts/Hz. The signal power required at receiver is 0.1 mω. Calculate channel capacity of this channel.*

Given data:

Signal power

$$\begin{aligned} P &= 0.1\, m\omega \\ \frac{N_0}{2} &= 10^{-12} \text{ watts/Hz} \\ N_0 &= 2 \times 10^{-12} \text{ watts/Hz} \\ \text{B.W} &= 4\,\text{KHz} \end{aligned}$$

Solution:

$$\begin{aligned} C &= B \log_2 \left(1 + \frac{P}{N_o B}\right) \\ &= 4000 \log \left(1 + \frac{0.1 \times 10^{-3}}{2 \times 10^{-12} \times 4 \times 10^3}\right) \\ C &= 54.430 \times 10^3 \text{ bits/sec} \end{aligned}$$

Part-A: Two marks Question and Answer

(1) Define lossless channel.

The channel described by a channel matrix with only one nonzero element in each column is called a lossless channel. In the lossless channel no sources information is lost in transmission.

(2) Define Deterministic channel

A channel described by a channel matrix with only one nonzero element in each row is called a deterministic channel and this element must be unity.

(3) Define noiseless channel.

A channel is called noiseless if it is both lossless and deterministic. The channel matrix has only one element in each row and in each column and this element is unity. The input and output alphabets are of the same size.

(4) What are the types of Correlation?

The types of Correlation are Cross Correlation and Auto Correlation

(5) What is the difference between Correlation and Convolution?

1. In Correlation physical time 't' is dummy variable and it disappears after solution of an integral. But in convolution 't' is a dummy variable.

(a) Convolution is a function of delay parameter 't' but convolution is a function of 't'.

1. Convolution is commutative but correlation is noncom mutative.

(6) Define Signal.

A signal is defined as any physical quantity carrying information that varies with time. The value of signal may be real or complex. The types of signal are continuous signal and discrete time signal.

(7) State Shannon's capacity theorem for a power and band limited channel.

The information capacity of a continuous channel of BW B Hz perturbed by a AWGN of PSD $\frac{N_o}{2}$ and limited to BW B is given by $C = \log_2 \left[1 + \left(\frac{P}{N_o B}\right)\right]$. where P is the average transmitted power

(8) Define entropy. **(May/June 2015)**

Entropy is the measure of the average information content per second. It is given by the expression

$$H(X) = \sum_{I} P(xi) \log_2 P(xi) \text{ bits/sample.}$$

(9) Define mutual information.

Mutual information $I(X,Y)$ of a channel is defined by

$$I(X,Y) = H(X) - H(X/Y) \text{ bits/symbol}$$

$H(X)$ - entropy of the source
$H(X/Y)$ - conditional entropy of Y.

(10) State the properties of mutual information.

1. $I(X,Y) = I(Y,X)$
2. $I(X,Y) >= 0$
3. $I(X,Y) = H(Y) - H(Y/X)$
4. $I(X,Y) = H(X) + H(Y) - H(X,Y)$

(11) Give the relation between the different entropies.

$$\begin{aligned} H(X,Y) &= H(X) + H(Y/X) \\ &= H(Y) + H(X/Y) \end{aligned}$$

$H(X)$ - entropy of the source (Y/X),

$H(X/Y)$ - conditional entropy

$H(Y)$ - entropy of destination

$H(X,Y)$ - Joint entropy of the source and destination

(12) Define information rate.

If the time rate at which source X emits symbols is r symbols per second. The information rate R of the source is given by

$$R = rH(X) \text{ bits/second}$$

$H(X)$ - entropy of the source

(13) What is data compaction?

For efficient signal transmission the redundant information must be removed from the signal prior to transmission .This information with no loss of information is ordinarily performed on a signal in digital form and is referred to as data compaction or lossless data compression.

(14) State the property of entropy. **(May/June 2015)**

1. $\log M \geq H(x) \geq 0$

(a) $H(X) = 0$ if all probabilities are zero

(b) $H(X) = \log_2 M$ if all probabilities are equal

(15) What is differential entropy?

The average amount of information per sample value of $x(t)$ is measured by

$$H(X) = -\int_{\infty} f_x(x) \log f_x(x) dx \text{ bit/sample}$$

$H(X)$ - differential entropy of X.

(16) What is the channel capacity of a discrete signal?

The channel capacity of a discrete signal $C = \max I(X, Y)P(xi)$

$I(X, Y)$ - mutual information.

(17) What is source coding and entropy coding?

A conversion of the output of a DMS into a sequence of binary symbols is called source coding. The design of a variable length code such that its average cod word length approaches the entropy of the DMS is often referred to as entropy coding.

(18) State Shannon Hartley theorem.

The capacity 'C' of a additive Gaussian noise channel is

$$C = B \log_2 \left(1 + \frac{S}{N}\right)$$

B = channel bandwidth, S/N = signal to noise ratio.

(19) What is the entropy of a binary memory-less source?

The entropy of a binary memory-less source $H(X) = -p_0 \log_2 p_0 - (1 - p_0) \log_2 (1 - p_0)\, p_0$ probability of symbol '0', $p_1 = (1 - p_0)$ = probability of transmitting symbol '1'.

(20) What happens when the number of coding alphabet increases?

When the number of coding alphabet increases the efficiency of the coding technique decreases.

(21) What is information theory?

Information theory deals with the mathematical modeling and analysis of a communication system rather than with physical sources and physical channels

(22) What is the channel capacity of a BSC and BEC?

For BSC the channel capacity $C = 1 + p\log_2 p + (1-p)\log_2(1-p)$.

For BEC the channel capacity $C = (1-p)$

(23) List the properties of Hamming distance. (NOV/DEC 2014)

The Hamming distance is a metric on the set of the words of length n (also known as a Hamming space), as it fulfills the conditions of non-negativity, identity of indiscernibles and symmetry, and it can be shown by complete induction that it satisfies the triangle inequality as well.

(24) What are the popular coding sequences of CDMA system. (NOV/DEC 2014)

Popular code sequences used in spread-spectrum transmission are

- Maximum Length sequences
- Walsh Hadamard sequences
- Gold codes, and
- Kasami codes.

(25) Explain Shannon-Fano coding. (May/June 2015)

An efficient code can be obtained by the following simple procedure, known as Shannon- Fano algorthim.

1. List the source symbols in order of decreasing probability.
2. Partition the set into two sets that are as close to equiprobable as possible, and sign 0 to the upper set and 1 to the lower set.
3. Continue this process, each time partitioning the sets with as nearly equal probabilities as possible until further partitioning is not possible.

(26) A source generates 3 messages with probability 0.5, 0.25 and 0.25. Calculate the entropy.

$$H(X) = \sum P_k \log \frac{1}{n}$$

(27) What are the types of characters used in data communication codes

(April/May 2015)

Prove that $I(x_i x_j) = I(x_i) + I(x_j)$ if x_i and x_j are independent.

If x_i and x_j are independent.

$$\begin{aligned} P(x_i x_j) &= P(x_i)P(x_j) \\ I(x_i x_j) &= \log 1/P(x_i x_j) \\ &= \log 1/P(x_i)P(x_j) \\ &= I(x_i) + I(x_j) \end{aligned}$$

(28) Differentiate between lossless and lossy coding.

Lossless Coding	**Lossy coding**
Coding that reduces the number of bits required to represent the symbol without affecting the equality of information by removing the redundant information	Lossy coding involves the loss of information due to compression in controlled manner
Process is reversible	Process is not reversible
Eg: Data compaction	Eg: Lempel Ziv algorithm

(29) How is the efficiency of the coding technique measured?

Efficiency of the code $= H(X)/L$

$L = \sum p(xi)li$ average code word length & li = length of the code word.

(30) Calculate the Entropy of the source with symbol probabilities 0.6, 0.3, 0.1.

$$H\,(X) = \sum P_k \log \frac{1}{n} = 1.299\,\text{bits/msg}$$

Part-B: Review Questions

1. Explain the coding and decoding process of block codes.
2. Explain the convolution codes. Constraint length 6 and rate is 1/2.
3. What do you mean by binary symmetric channel? Derive channel capacity formula for symmetric channel
4. Derive the channel capacity theorem and discuss the implications of the information capacity theorem.

5. Discuss the Viterbi algorithm by showing the possible paths through the trellis of a coder. Assume the state diagram of any coder
6. Briefly discuss on various error control codes and explain in detail with one example of convolution codes
7. Draw in detail about the procedure for Shannon-Fano Coding Scheme
8. Explain coding and decoding procedure for block codes
9. Discuss the BW-SNR trade off of a communication systems
10. For a given 8 bit stream 11010100 plot NRZ, RZ ,AMI, and differential Manchester code
11. Apply the following coding and dram the waveform for bit stream 10011100 on NRZ,RZ,AMI,HDBP,ABQ and MBnB.
12. An analog signal is bandlimited to B Hz and sampled at Nyquist rate. The samples are quantized into 4 levels. Each level represents one message. Thus there are 4 messages. The probabilities of occurrence of these 4 levels (Messages) are $p_1 = 0.1$, $p_4 = 0.2$, $p_2 = 0.45$ & $p_3 = 0.25$. Find out information rate of the source
13. A black and white TV picture consists of about 2X106 picture elements with 1 different brightness levels, with equal probabilities. If pictures are repeated at the rate of 32 per second, calculate average rate of information conveyed by this TV picture Source. If SNR is 30 db, what is the maximum bandwidth required to support the transmission of the resultant video signal?
14. A discrete memory less source has five symbols x_1, x_2, x_3, x_4 and x_5 with probabilities 0.4, 0.19, 0.16, 0.15 and 0.15 respectively attached to every symbol Construct a Shannon-Fano and Huffman code for the source and calculate code efficiency. compare the two techniques of source coding
15. The generator polynomial of a (7, 4) cyclic code is $G(p) = p_3 + p + 1$ Find all the Code vectors for the code in non systematic form.
16. Draw the polar, unipolar, bipolar, Manchester NRZ for the data(1 0 1 1 0 0)
17. Apply the Shannon-Fano Coding Scheme for the source x_1, x_2, x_3, x_4 with the probabilities 1/8, 1/2, 1/4, 1/8 respectively and determine the code efficiency