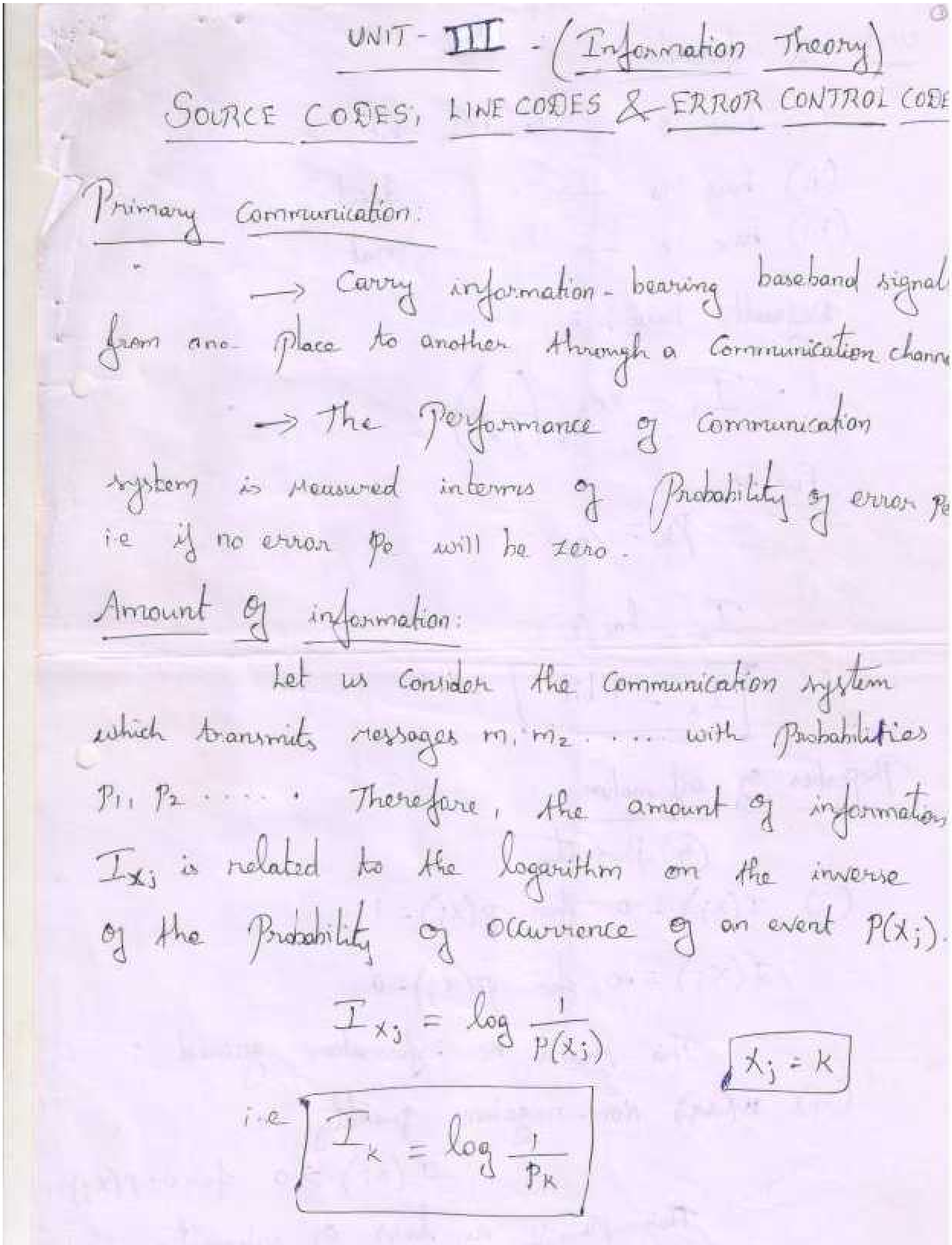


UNIT - III - (Information Theory)

SOURCE CODES, LINE CODES & ERROR CONTROL CODE

Primary communication.

→ Carry information - bearing baseband signal from one place to another through a Communication chann

→ The Performance of Communication system is measured interms of Probability of error pe i.e if no error pe will be zero.

Amount of information:

Let us consider the communication system which transmits messages $m_1, m_2, \dots$ with probabilities $p_1, p_2, \dots$. Therefore, the amount of information I_{x_j} is related to the logarithm on the inverse of the Probability of occurrence of an event $P(x_j)$.

$$I_{x_j} = \log \frac{1}{P(x_j)}$$

$$\boxed{x_j = k}$$

i.e $$\boxed{I_k = \log \frac{1}{p_k}}$$

Units of information

(i) base '2' → Unit is bit,

(ii) base '10' → " " decit

(iii) base 'e' → " " nat

Default base is '2',

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

for eg:

$$p_k = 1/2$$

$$I_k = \log_2(2)$$

$$\boxed{I_k = 1 \text{ bit}}$$

Properties of information:

④ Property

(i) $I(x_j) = 0$ for $p(x_j) = 1$

$I(x_j) = \infty$ for $p(x_j) = 0$

This Means no information gained.

(ii) ~~$I(x_j)$~~ Non-negative quantity i.e

$I(x_j) \geq 0$ for $0 \leq p(x_j) \leq 1$

This Means no loss of informatio

(iii) $I(x_j) > I(y_k)$ for $p(x_j) < p(y_k)$

This ~~[illegible]~~ Means more information gain.

(iv) $I(x_j, y_k) = I(x_j) + I(y_k)$

if x_j & y_k are statistically independent

Entropy (or) Average information

It is defined as the source which "Produces average information per Message in a particular interval". It is also called as Comentropy.

Let $m_1, m_2 \dots m_k$ be k different Messages with corresponding Probabilities $p_1, p_2 \dots p_k$.

Let us assume that 'L' sequence of Messages have been generated for a long time interval with $L >> k$

Then, the number of Messages

$$m_1 = p_1 L \longrightarrow ①$$

The amount of information in Message M_1 is

$$I_1 = \log_2\left(\frac{1}{p_1}\right) \longrightarrow ②$$

The total amount of information due to Message m_1 is,

$I_L = I \times m$

IIIly, the total amount of information due to message M_2 is

$$I_{t2} = p_2 L \log_2\left(\frac{1}{p_2}\right) \longrightarrow \textcircled{A}.$$

The total amount of information due to sequence of L messages is given as

$$I_t = I_{t1} + I_{t2} + \dots + I_{tK}$$

$$= p_1 L \log_2\left(\frac{1}{p_1}\right) + p_2 L \log_2\left(\frac{1}{p_2}\right) + \dots$$

$$p_K L \log_2\left(\frac{1}{p_K}\right).$$

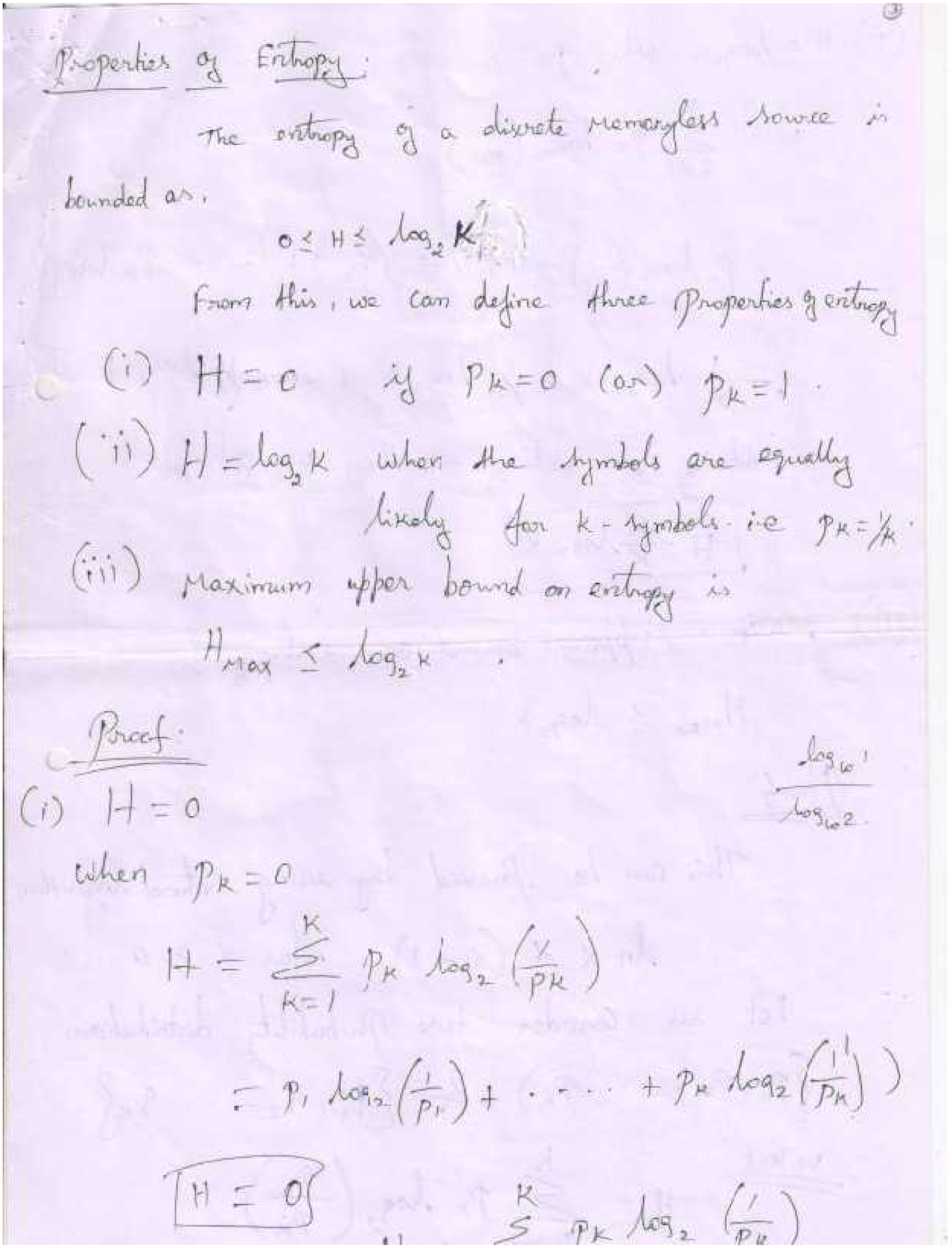
The average information per message is called entropy & it is represented by H (or) H(S).

$$H(S) = \frac{I_t}{L} = p_1 \log_2\left(\frac{1}{p_1}\right) + p_2 \log_2\left(\frac{1}{p_2}\right) + \dots$$

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

$$\boxed{H(S) = \sum_{k=1}^{K} p_k \log_2\left(\frac{1}{p_k}\right)}$$

Entropy is called as discrete memoryless source because each & every symbol emitted at any time are independent of the previous one



Properties of Entropy:

The entropy of a discrete memoryless source is bounded as,

$$0 \leq H \leq \log_2 K$$

From this, we can define three properties of entropy

(i) $H = 0$ if $p_k = 0$ (or) $p_k = 1$.

(ii) $H = \log_2 K$ when the symbols are equally likely for k-symbols i.e $p_k = 1/k$.

(iii) Maximum upper bound on entropy is

$$H_{max} \leq \log_2 K .$$

Proof:

$$\frac{\log_{10} 1}{\log_{10} 2}$$

(i) $H = 0$

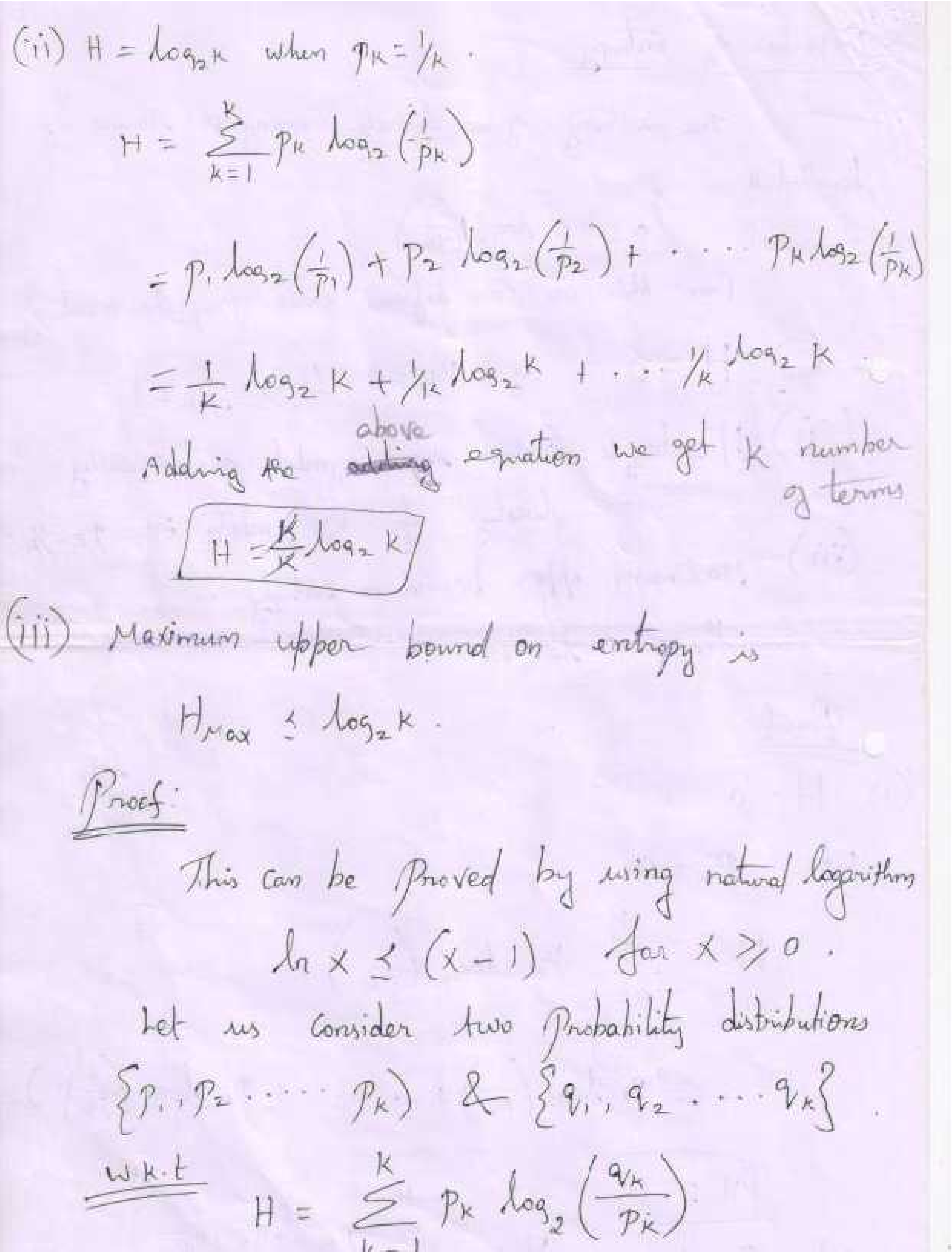
when $p_k = 0$

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

$$= p_1 \log_2 \left(\frac{1}{p_1}\right) + \cdots + p_k \log_2 \left(\frac{1}{p_k}\right))$$

$$\boxed{H = 0}$$

$$H = \sum^{K} p_k \log_2 \left(\frac{1}{p_k}\right)$$



(ii) $H = \log_2 K$ when $p_K = 1/K$.

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

$$= p_1 \log_2\left(\frac{1}{p_1}\right) + p_2 \log_2\left(\frac{1}{p_2}\right) + \cdots p_K \log_2\left(\frac{1}{p_K}\right)$$

$$= \frac{1}{K} \log_2 K + 1/K \log_2 K + \cdots 1/K \log_2 K$$

Adding the above ~~adding~~ equation we get K number of terms

$$\boxed{H = \frac{K}{K} \log_2 K}$$

(iii) Maximum upper bound on entropy is

$$H_{max} \leq \log_2 K.$$

Proof:

This can be proved by using natural logarithm

$$\ln x \leq (x-1) \quad \text{for } x \geq 0.$$

Let us consider two probability distributions

$\{p_1, p_2 \cdots p_K)$ & $\{q_1, q_2 \cdots q_K\}$.

w.k.t $H = \sum_{k=1}^{K} p_K \log_2\left(\frac{q_K}{p_K}\right)$

It can be written as

$$\sum_{k=1}^{K} p_k \log_2\left(\frac{q_k}{p_k}\right) = \sum_{k=1}^{K} p_k \frac{\log_{10}\left(\frac{q_k}{p_k}\right)}{\log_{10} 2}$$

$\log_a b = \frac{\log_{10} b}{\log_{10} a}$

Multiply and divide by $\log_{10} e$ in R.H.S

$$\sum_{k=1}^{K} p_k \log_2\left(\frac{q_k}{p_k}\right) = \sum_{k=1}^{K} p_k \frac{\log_{10} e}{\log_{10} e} \frac{\log_{10}\left(\frac{q_k}{p_k}\right)}{\log_{10} 2}$$

$$= \sum_{k=1}^{K} p_k \frac{\log_{10} e}{\log_{10} 2} \cdot \frac{\log_{10}\left(\frac{q_k}{p_k}\right)}{\log_{10} e}$$

$$= \sum_{k=1}^{K} p_k \log_2 e \cdot \log_e\left(\frac{q_k}{p_k}\right)$$

From natural logarithm

$$\log_e\left(\frac{q_k}{p_k}\right) = \ln\left(\frac{q_k}{p_k}\right)$$

$$\sum_{k=1}^{K} p_k \log_2\left(\frac{q_k}{p_k}\right) = \sum_{k=1}^{K} p_k \log_2 e \cdot \ln\left(\frac{q_k}{p_k}\right)$$

$$= \log_2 e \sum^{K} p_k \ln\left(\frac{q_k}{\ }\right)$$

$$\ln\left(\frac{q_K}{p_K}\right) \leq \left(\frac{q_K}{p_K} - 1\right).$$

$$\sum_{K=1}^{K} p_K \log_2\left(\frac{q_K}{p_K}\right) \leq \log_2 e \sum_{K=1}^{K} p_K\left(\frac{q_K}{p_K} - 1\right)$$

$$\leq \log_2 e \sum_{K=1}^{K} \cancel{p_K}\left(\frac{q_K - p_K}{\cancel{p_K}}\right)$$

$$\leq \log_2 e \sum_{K=1}^{K} (q_K - p_K)$$

$$\leq \log_2 e \left(\sum_{K=1}^{K} q_K - \sum_{K=1}^{K} p_K\right)$$

$$\leq \log_2 e\,(1 - 1)$$

$$\sum_{K=1}^{K} p_K \log_2\left(\frac{q_K}{p_K}\right) \leq 0$$

$$\sum_{K=1}^{K} p_K\left[\log_2 q_K + \log_2\left(\frac{1}{p_K}\right)\right] \leq 0$$

$$\sum_{K=1}^{K} p_K \log_2(q_K) + \sum_{K=1}^{K} p_K \log_2\left(\frac{1}{p_K}\right) \leq 0$$

$$\sum_{K=1}^{K} p_K \log_2\left(\frac{1}{p_K}\right) \leq -\sum_{K=1}^{K} p_K \log_2 q_K$$

$$\sum_{k=1}^{K} p_k \log_2\left(\frac{1}{p_k}\right) \le \sum_{k=1}^{K} p_k \log_2\left(\frac{1}{q_k}\right).$$

Let us assume all symbols in the alphabet are equally likely suppose we next put

Then $q_k = 1/K$

$$\sum_{k=1}^{K} p_k \log_2\left(\frac{1}{p_k}\right) \le \sum_{k=1}^{K} p_k \log_2 K$$

$$\le \log_2 K \sum_{k=1}^{K} p_k$$

$$\sum_{k=1}^{K} p_k \log_2\left(\frac{1}{p_k}\right) \le \log_2 K$$

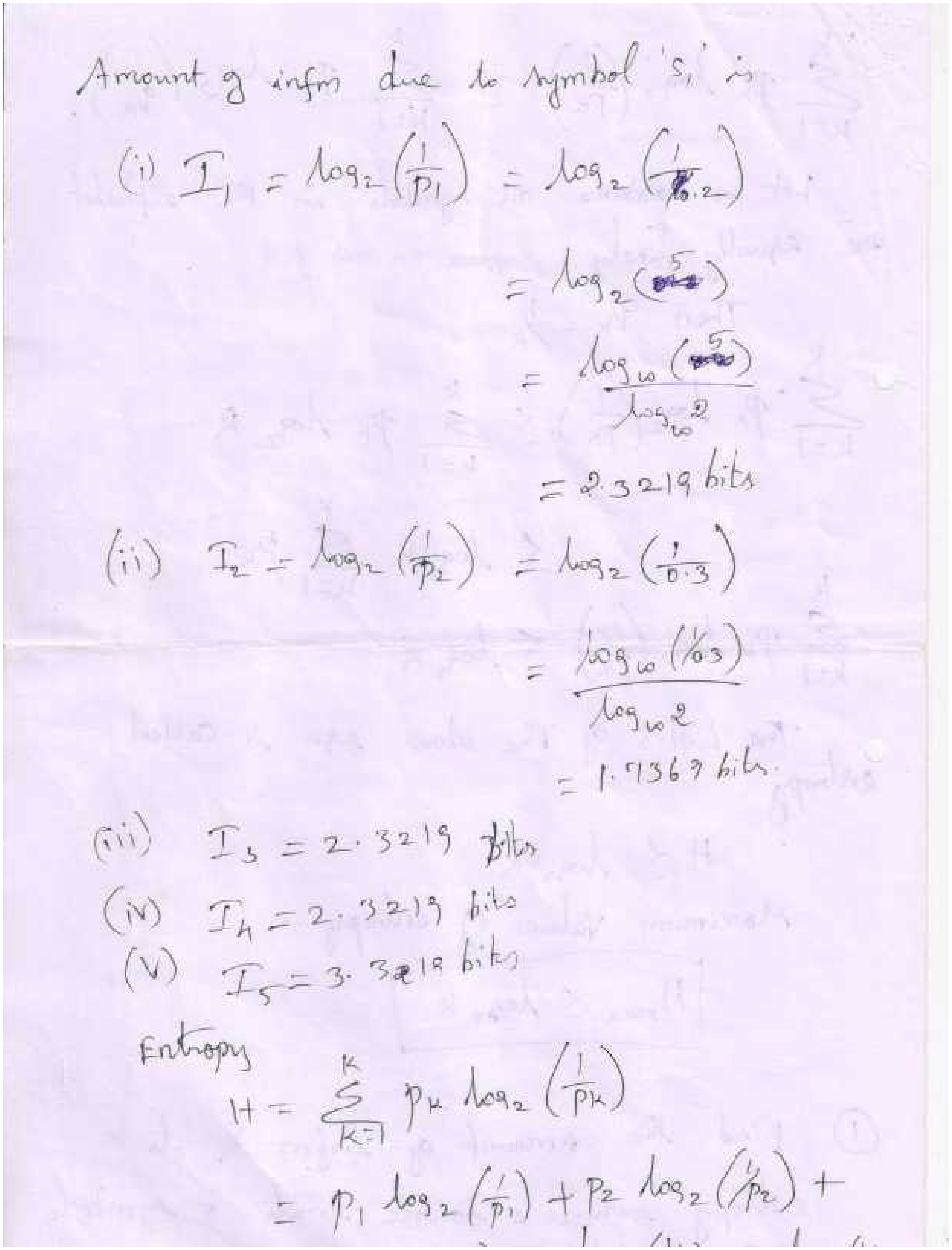
The L.H.S of the above eqn is called entropy

$$H \le \log_2 K$$

Maximum value of entropy is

$$\boxed{H_{max} \le \log_2 K}$$

① Find the amount of infm & its entropy when a source emits 5 symbols



Amount of infm due to symbol 's_1' is

(i) $I_1 = \log_2\left(\frac{1}{p_1}\right) = \log_2\left(\frac{1}{0.2}\right)$

$= \log_2(5)$

$= \frac{\log_{10}(5)}{\log_{10} 2}$

$= 2.3219$ bits

(ii) $I_2 = \log_2\left(\frac{1}{p_2}\right) = \log_2\left(\frac{1}{0.3}\right)$

$= \frac{\log_{10}(1/0.3)}{\log_{10} 2}$

$= 1.7367$ bits.

(iii) $I_3 = 2.3219$ bits

(iv) $I_4 = 2.3219$ bits

(V) $I_5 = 3.3219$ bits

Entropy

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

$= p_1 \log_2\left(\frac{1}{p_1}\right) + p_2 \log_2\left(1/p_2\right) +$

$$= 0.2 \times 2.3219 + 0.3 \times 1.7369 + 0.2 \times 2.3219 + 0.2 \times 2.3219 + 0.1 \times 3.3219$$

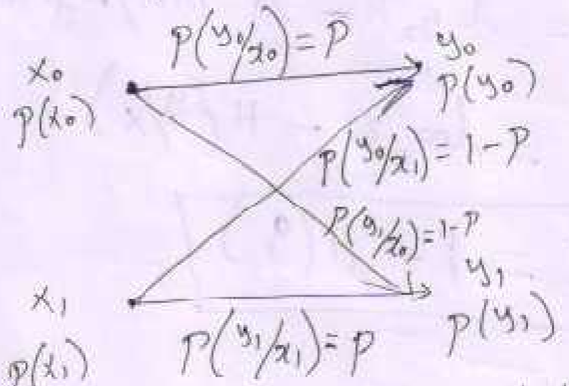
$$= 0.4644 + 0.5211 + 0.4644 + 0.4644 + 0.3322$$

$$\boxed{H = 2.246 \text{ bits/symbol.}}$$

BSC (Binary Symmetric channel)

Binary Communication channel is said to be symmetric if

$$P\left(\frac{y_0}{x_0}\right) = P\left(\frac{y_1}{x_1}\right) = P$$



It Can be represented using the Matrix

$$P(y) = P(x)\, P(y/x)$$

$$\begin{bmatrix} P(y_0) \end{bmatrix} = \begin{bmatrix} P(x_0) & P(x_1) \end{bmatrix} \begin{bmatrix} P(y_0/x_0) & P(y_1/x_0) \\ P(y_0/x_1) & P(y_1/x_1) \end{bmatrix}$$

$$\begin{bmatrix} p(y_0) \\ p(y_1) \end{bmatrix} = \begin{bmatrix} p(x_0) & p(x_1) \end{bmatrix} \begin{bmatrix} p & 1-p \\ 1-p & p \end{bmatrix}$$

In Binary symmetric channel, the value of $m = n = 2$ (number of rows & columns),

$\therefore k = 2$ &

The channel Matrix $Q = P(y/x)$

$$= \begin{bmatrix} p & 1-p \\ 1-p & p \end{bmatrix}$$

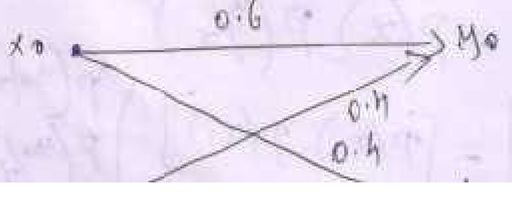
The channel capacity for a binary symmetric channel is,

$$C = \log_2 k - A \text{ bits/symbol}$$

$$= \log_2 2 - H(y/x)$$

$$\boxed{C = 1 - H(q)}$$

① A BSC is shown as



(i) find the channel matrix

(ii) find $p(y_0)$ & $p(y_1)$ if $p(x_0)=0.3$, $p(x_1)=0.7$.

Gn

$p(y_0/x_0) = 0.6$

$p(y_0/x_1) = 0.4$

$p(y_1/x_0) = 0.4$

$p(y_1/x_1) = 0.6$

(i) channel matrix $p(y/x) = \begin{bmatrix} p(y_0/x_0) & p(y_1/x_0) \\ p(y_0/x_1) & p(y_1/x_1) \end{bmatrix}$

$= \begin{bmatrix} 0.6 & 0.4 \\ 0.4 & 0.6 \end{bmatrix}$

~~$= 0.56 - 0.16$~~

~~$p(y/x) = 0.2$~~

(ii) $p(y) = p(x)\, p(y/x)$

$\begin{bmatrix} p(y_0) \\ p(y_1) \end{bmatrix} = \begin{bmatrix} p(x_0) & p(x_1) \end{bmatrix} \; p(y/x)$

$= \begin{bmatrix} 0.3 & 0.7 \end{bmatrix} \begin{bmatrix} 0.6 & 0.4 \\ 0.4 & 0.6 \end{bmatrix}$

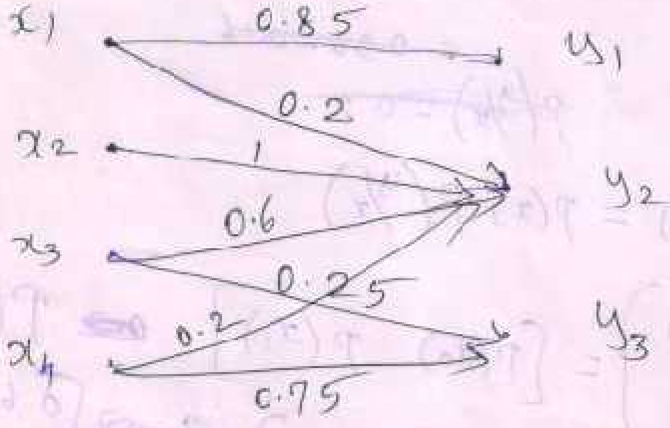
$= \begin{bmatrix} 0.46 & 0.54 \\ 0.46 & 0.54 \end{bmatrix}$

$p(y) = \begin{bmatrix} p(y_0) & p(y_1) \end{bmatrix}$ $\quad p(y\ ) = 0.46$

② Draw the ~~Binary Asymmetric~~ channel diagram for

$$P(y/x) = \begin{array}{c|cccc} & y_1 & y_2 & y_3 & y_4 \\ \hline x_1 & 0.85 & 0.2 & 0 & 0 \\ x_2 & 0 & 1 & 0 & 0 \\ x_3 & 0 & 0.6 & 0.25 & 0 \\ x_4 & 0 & 0.2 & 0.75 & 0 \end{array}$$

$$P(y/x) = \begin{bmatrix} P(y_1/x_1) & P(y_2/x_1) & P(y_3/x_1) & P(y_4/x_1) \\ P(y_1/x_2) & P(y_2/x_2) & P(y_3/x_2) & P(y_4/x_2) \\ P(y_1/x_3) & P(y_2/x_3) & P(y_3/x_3) & P(y_4/x_3) \\ P(y_1/x_4) & P(y_2/x_4) & P(y_3/x_4) & P(y_4/x_4) \end{bmatrix}$$



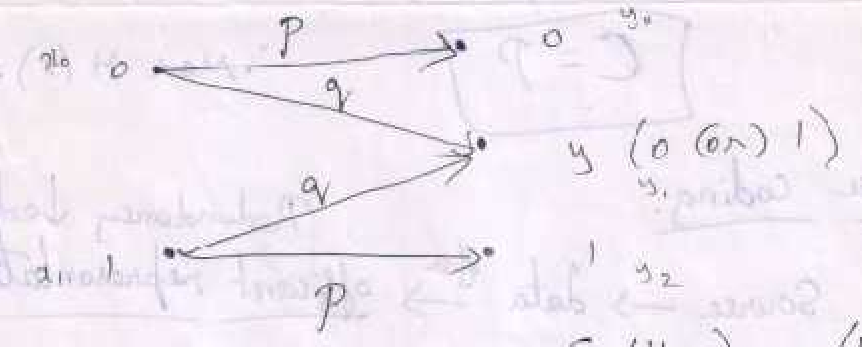
(8)

Binary Erasure channel

It has two inputs 0, 1 & three outputs 0, y, 1 i.e we transmitted 0, 1 & we received 0, y, 1.

y indicates due to noise, that can't be Predicted, whether it is 0 & 1. Alternatively y indicates that the o/p is erased.

channel Diagram



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

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

channel capacity $C = \text{Max } I(X;Y)$

$$I(X;Y) = H(X) - H(X/Y)$$

$$= H(X) - H(X)(1-p)$$

$$= H(X)\left[\cancel{1} - \cancel{1} + p\right]$$

$$= p\,H(X)$$

$H(y)=H(x)\,H(y/x)$

$$C = \text{Max } p\,H(X)$$

$$= p \text{ Max } H(X)$$

$$\boxed{C = p} \qquad \therefore \text{Max } H(X) = 1$$

__Source__ __coding:__

Redundancy ↓ed.

Source → data $\xrightarrow{\text{for}}$ efficient representation →

Source coding is used.

(i) Shannon fano coding → used to encode the messages depending upon their Probabilities

(ii) Huffman Coding.

Shannon Fano Coding:

Procedure:

Step 1: List the source symbols in order of decreasing probabilities

Step 2: Partition the set into two sets that are as close to equiprobable as possible and assign 0 to the upper set and assign 1 to the lower set

Step 3: Continue this process each time partitioning the sets with as nearly probabilities as possible until further partitioning is not possible

① A discrete memoryless source has 5 symbols s_1, s_2, s_3, s_4 & s_5 with probabilities 0.4, 0.2, 0.1, 0.2, 0.1 respectively. Construct a shannon Fano coding and calculate its efficiency.

Given

symbols	Probabilities
s_1	0.4
s_2	0.2
s_3	0.1
s_4	0.2
s_5	0.1

symbols	Probabilities		stage 1		stage 2					CW
s_1	0.4	0.4	0							0
s_2	0.2	0.6	1	0.2	0					10
s_4	0.2		1	0.4	1	0.2	0	0.2		110
s_3	0.1		1		1	0.2	1	0.1	0	1110
s_5	0.1		1		1		1	0.1	1	1111

CW	Length
0	1
10	2
110	3
1110	4
1111	4

$$\eta = H/\bar{L}$$

$$\bar{L} = \sum_{k=1}^{k} p_k L_k = \sum_{k=1}^{5} p_k L_k$$

$$= p_1 L_1 + p_2 L_2 + p_3 L_3 + p_4 L_4 + p_5 L_5$$

$$= (0.4)(1) + (0.2)(2) + (0.2)(3) + (0.1)(4) + (0.1)(4)$$

$$= 0.4 + 0.4 + 0.6 + 0.4 + 0.4$$

$$= 2.2 \text{ bits/symbol}$$

$$H = \sum_{k=1}^{K} p_k \log_2 (1/p_k)$$

$$= p_1 \log_2(1/p_1) + p_2 \log_2(1/p_2) + p_3 \log_2(1/p_3) + p_4 \log_2(1/p_4) + p_5 \log_2(1/p_5)$$

$$= (0.4) \log_2(1/0.4) + (0.2) \log_2(1/0.2) + (0.2) \log_2(1/0.2) + (0.1) \log_2(1/0.1) + (0.1) \log_2(1/0.1)$$

$$= (0.4) \frac{\log_{10}(1/0.4)}{\log_{10} 2} + (0.2) \frac{\log_{10}(1/0.2)}{\log_{10} 2} + (0.2) \frac{\log_{10}(1/0.2)}{\log_{10}(2)} +$$

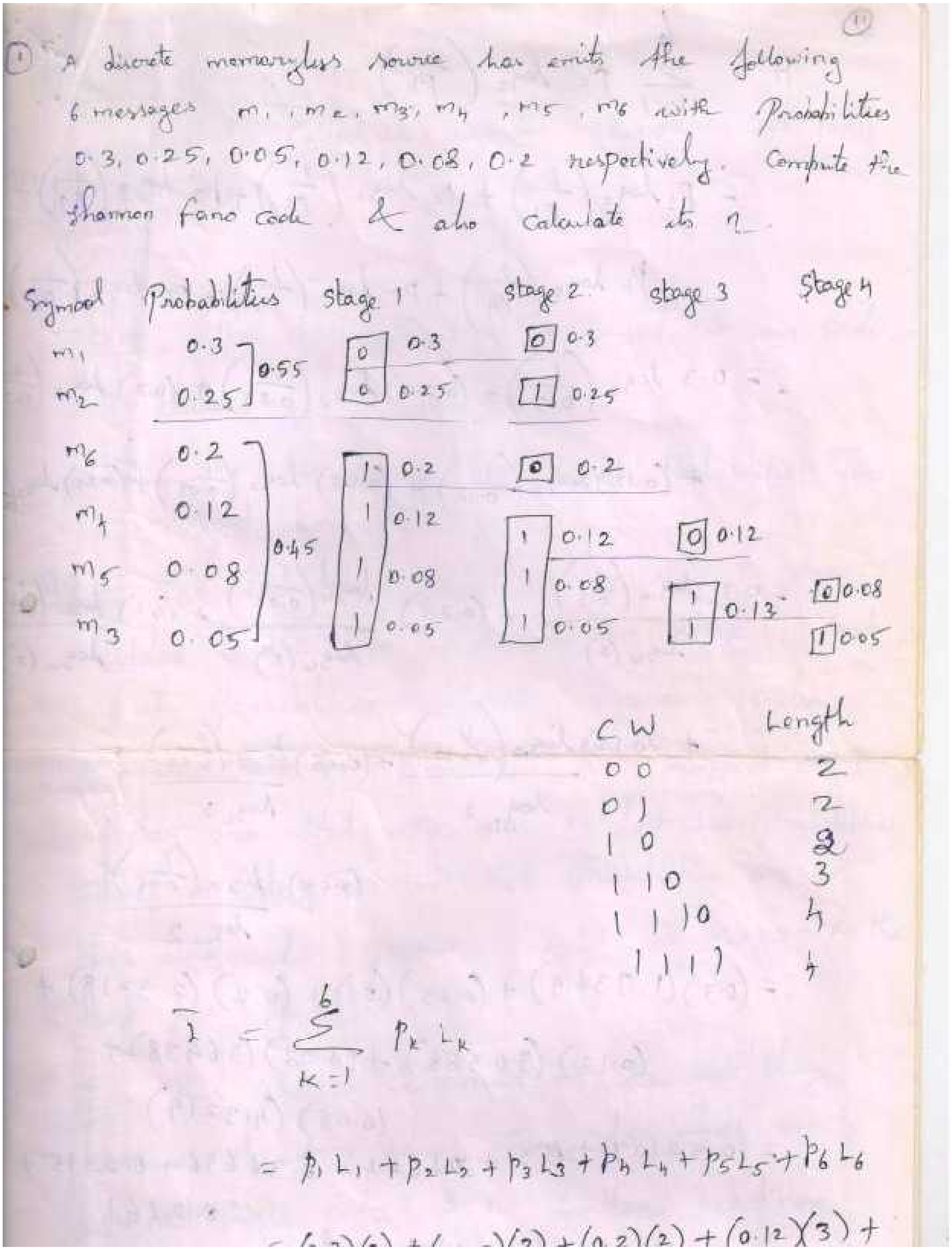
$$(0.1) \frac{\log_{10}(1/0.1)}{\log_{10} 2} + (0.1) \frac{\log_{10}(1/0.1)}{\log_{10} 2}$$

$$= 0.5287 + 0.4644 + 0.4644 + 0.3322 + 0.3322$$

$$H = 2.122 \text{ bits/symbols}$$

$$\eta = H/L = \frac{2.122}{2.2} = 0.9645$$

$$\boxed{\eta = 96.45\%}$$



⑪

① A discrete memoryless source has emits the following 6 messages $m_1, m_2, m_3, m_4, m_5, m_6$ with Probabilities 0.3, 0.25, 0.05, 0.12, 0.08, 0.2 respectively. Compute the Shannon fano code & also calculate its η

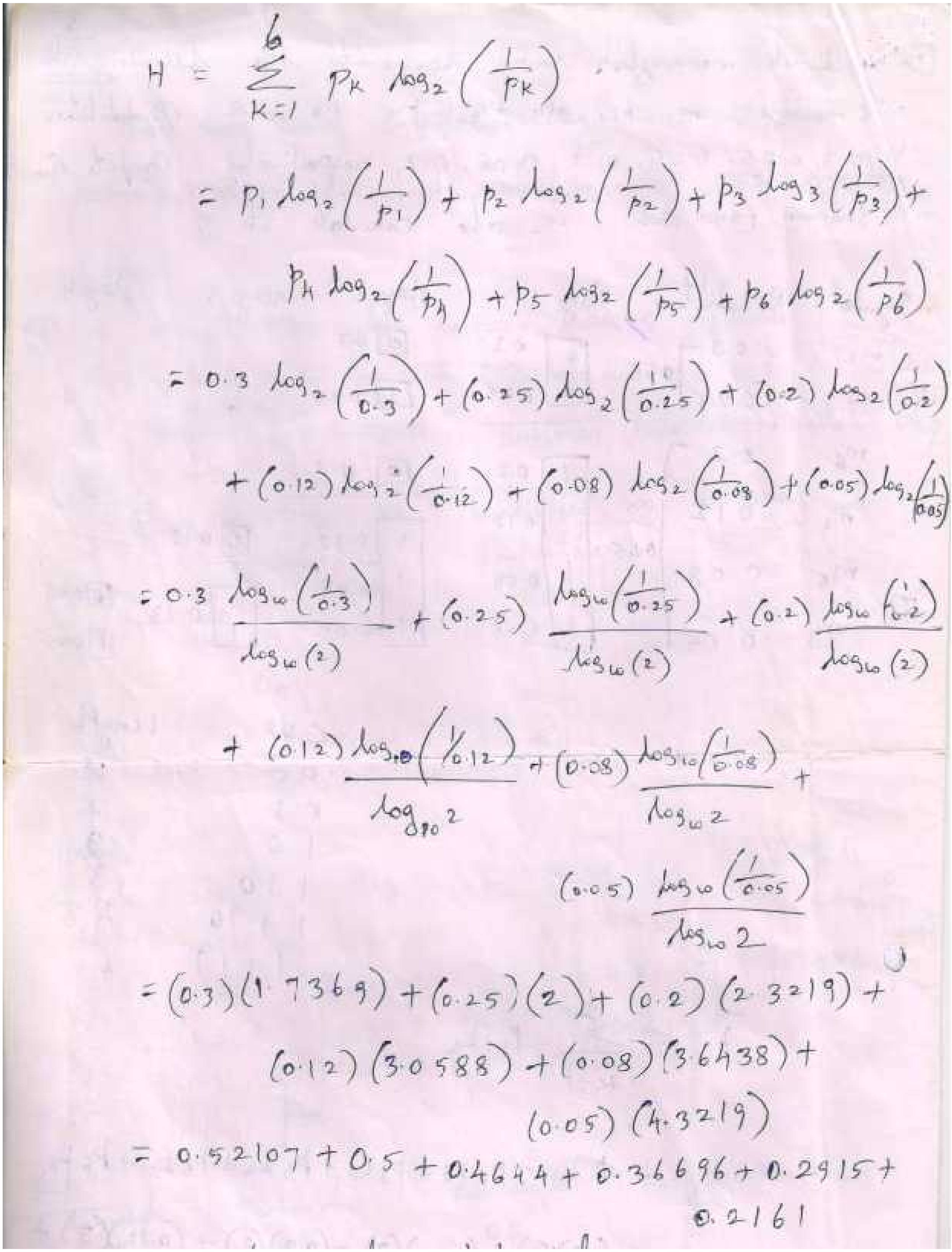
Symbol	Probabilities	Stage 1	Stage 2	Stage 3	Stage 4
m_1	0.3] 0.55	0 0.3	0 0.3		
m_2	0.25	0 0.25	1 0.25		
m_6	0.2] 0.45	1 0.2	0 0.2		
m_4	0.12	1 0.12	1 0.12	0 0.12	
m_5	0.08	1 0.08	1 0.08	1 0.13	0 0.08
m_3	0.05	1 0.05	1 0.05	1	1 0.05

CW	Length
00	2
01	2
10	2
110	3
1110	4
1111	4

$$\bar{L} = \sum_{k=1}^{6} p_k L_k$$

$$= p_1 L_1 + p_2 L_2 + p_3 L_3 + p_4 L_4 + p_5 L_5 + p_6 L_6$$

$$(0.2)(2) + (0.12)(3) +$$



$$H = \sum_{K=1}^{6} p_K \log_2\left(\frac{1}{p_K}\right)$$

$$= p_1 \log_2\left(\frac{1}{p_1}\right) + p_2 \log_2\left(\frac{1}{p_2}\right) + p_3 \log_3\left(\frac{1}{p_3}\right) +$$

$$p_4 \log_2\left(\frac{1}{p_4}\right) + p_5 \log_2\left(\frac{1}{p_5}\right) + p_6 \log_2\left(\frac{1}{p_6}\right)$$

$$= 0.3 \log_2\left(\frac{1}{0.3}\right) + (0.25) \log_2\left(\frac{1}{0.25}\right) + (0.2) \log_2\left(\frac{1}{0.2}\right)$$

$$+ (0.12) \log_2\left(\frac{1}{0.12}\right) + (0.08) \log_2\left(\frac{1}{0.08}\right) + (0.05) \log_2\left(\frac{1}{0.05}\right)$$

$$= 0.3 \frac{\log_{10}\left(\frac{1}{0.3}\right)}{\log_{10}(2)} + (0.25) \frac{\log_{10}\left(\frac{1}{0.25}\right)}{\log_{10}(2)} + (0.2) \frac{\log_{10}\left(\frac{1}{0.2}\right)}{\log_{10}(2)}$$

$$+ (0.12) \frac{\log_{10}\left(1/0.12\right)}{\log_{10} 2} + (0.08) \frac{\log_{10}\left(\frac{1}{0.08}\right)}{\log_{10} 2} +$$

$$(0.05) \frac{\log_{10}\left(\frac{1}{0.05}\right)}{\log_{10} 2}$$

$$= (0.3)(1.7369) + (0.25)(2) + (0.2)(2.3219) +$$

$$(0.12)(3.0588) + (0.08)(3.6438) +$$

$$(0.05)(4.3219)$$

$$= 0.52107 + 0.5 + 0.4644 + 0.36696 + 0.2915 +$$

$$0.2161$$

$$\overline{\log_{10} 2} \quad \overline{\log_{10} 2} +$$

$$(0.05)\frac{\log_{10}\left(\frac{1}{0.05}\right)}{\log_{10} 2}$$

$$= (0.3)(1.7369) + (0.25)(2) + (0.2)(2.3219) + (0.12)(3.0588) + (0.08)(3.6438) + (0.05)(4.3219)$$

$$= 0.52107 + 0.5 + 0.4644 + 0.36696 + 0.2915 + 0.2161$$

$H = 2.36003$ inform bits/symbol.

$$\eta = H/L = \frac{2.36003}{2.38} = 0.9916$$

$$\boxed{\eta = 99\%}$$

Huffman Coding:

→ also Called as Minimum redundancy code (or) Optimum Code.

Procedure:

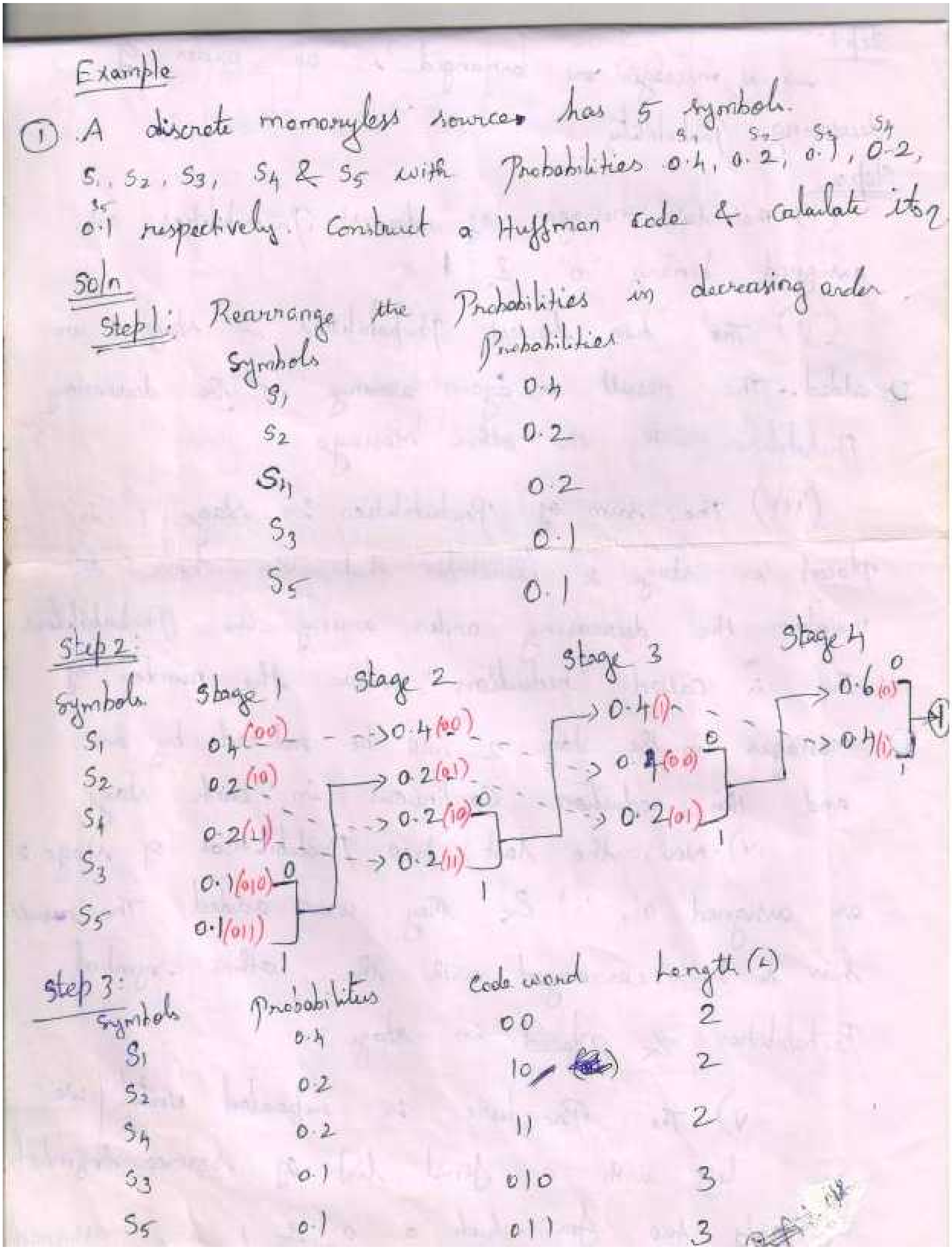
Step1: The Messages are arranged in order of decreasing Probabilities

Step 2: The two Messages of lowest Probabilities are assigned binary '0' & '1'.

Step3: The two lowest Probabilities in stage I are added & the sum is placed in stage II, such that Probabilities are in descending order

4: Now last two Probabilities are assigned 0 & 1 & they are added. The sum of last two Probabilities placed in stage III such that Probabilities are in descending order. Again '0' & '1' is assigned to the last two Probabilities.

5: This Process Continued till the last stage Contains only two values These two values are assigned digits 0 & 1 & no further repetition



Example

① A discrete memoryless source has 5 symbols S_1, S_2, S_3, S_4 & S_5 with Probabilities 0.4, 0.2, 0.1, 0.2, 0.1 respectively. Construct a Huffman code & calculate its η.

Soln

Step 1: Rearrange the Probabilities in decreasing order

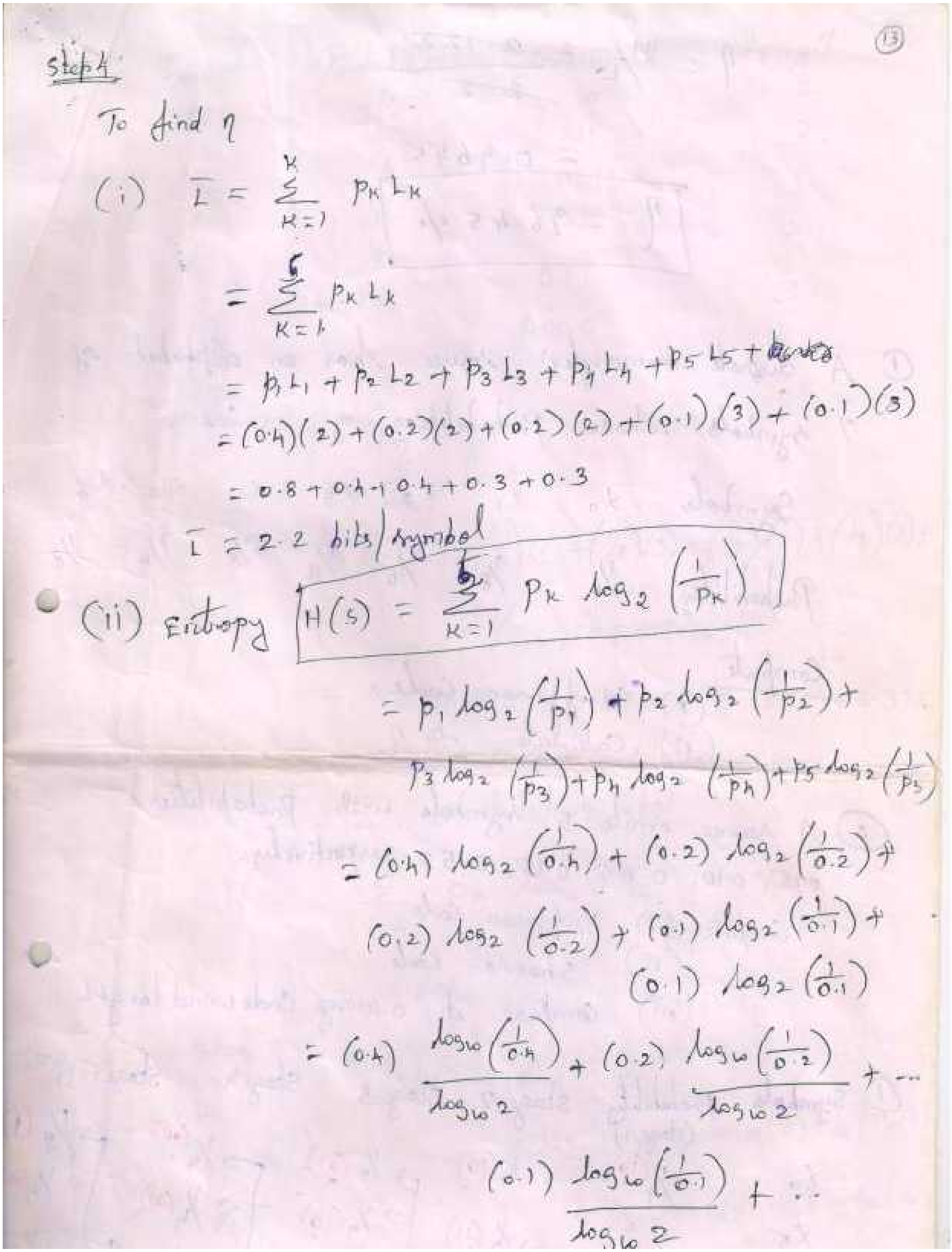
Symbols	Probabilities
S_1	0.4
S_2	0.2
S_4	0.2
S_3	0.1
S_5	0.1

Step 2:

Symbols	Stage 1	Stage 2	Stage 3	Stage 4
S_1	0.4 (00)	0.4 (00)	0.4 (1)	0.6 (0) 0
S_2	0.2 (10)	0.2 (01)	0.4 (00) 0	0.4 (1) 1
S_4	0.2 (11)	0.2 (10) 0	0.2 (01) 1	
S_3	0.1 (010) 0	0.2 (11) 1		
S_5	0.1 (011) 1			

Step 3:

Symbols	Probabilities	Code word	Length (L)
S_1	0.4	00	2
S_2	0.2	10	2
S_4	0.2	11	2
S_3	0.1	010	3
S_5	0.1	011	3



⑬

Step 4

To find η

(i) $\bar{L} = \sum_{K=1}^{K} p_K L_K$

$= \sum_{K=1}^{5} p_K L_K$

$= p_1 L_1 + p_2 L_2 + p_3 L_3 + p_4 L_4 + p_5 L_5 +$

$= (0.4)(2) + (0.2)(2) + (0.2)(2) + (0.1)(3) + (0.1)(3)$

$= 0.8 + 0.4 + 0.4 + 0.3 + 0.3$

$\bar{L} = 2.2$ bits/symbol

(ii) Entropy $\boxed{H(S) = \sum_{K=1}^{5} p_K \log_2\left(\frac{1}{p_K}\right)}$

$= p_1 \log_2\left(\frac{1}{p_1}\right) + p_2 \log_2\left(\frac{1}{p_2}\right) +$

$p_3 \log_2\left(\frac{1}{p_3}\right) + p_4 \log_2\left(\frac{1}{p_4}\right) + p_5 \log_2\left(\frac{1}{p_5}\right)$

$= (0.4) \log_2\left(\frac{1}{0.4}\right) + (0.2) \log_2\left(\frac{1}{0.2}\right) +$

$(0.2) \log_2\left(\frac{1}{0.2}\right) + (0.1) \log_2\left(\frac{1}{0.1}\right) +$

$(0.1) \log_2\left(\frac{1}{0.1}\right)$

$= (0.4) \frac{\log_{10}\left(\frac{1}{0.4}\right)}{\log_{10} 2} + (0.2) \frac{\log_{10}\left(\frac{1}{0.2}\right)}{\log_{10} 2} + \cdots$

$(0.1) \frac{\log_{10}\left(\frac{1}{0.1}\right)}{\log_{10} 2} + \cdots$

$$\eta = H/L = \frac{2.122}{2.2}$$

$$= 0.9645$$

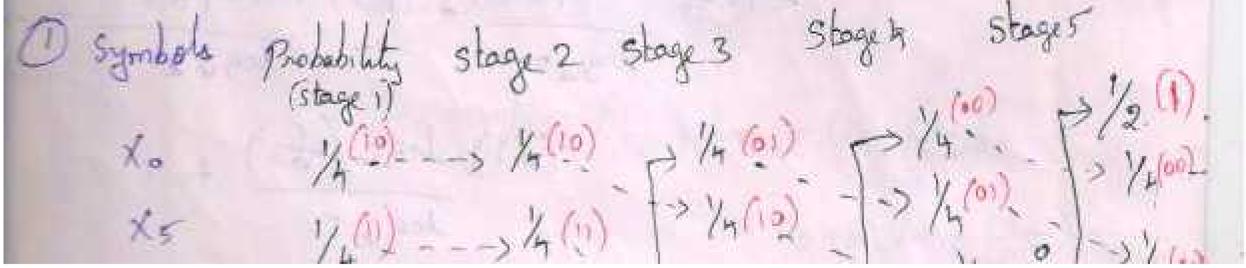
$$\boxed{\eta = 96.45\%}$$

① A discrete memoryless source has an alphabet of 7 symbols whose probabilities of occurrence

Symbols	x_0	x_1	x_2	x_3	x_4	x_5	x_6
Probability	$1/4$	$1/8$	$1/16$	$1/16$	$1/8$	$1/4$	$1/8$

Compute
(i) Huffman code
(ii) Calculate its η.

② A source emits 5 symbols with probabilities
0.15, 0.10, 0.05, 0.15, 0.55 respectively.
Compute (i) Huffman code
(ii) Shannon code
(iii) Compare its average code word length.



(14)

Symbol	Probability	Code word	Length
x_0	1/4	10	2
x_5	1/4	11	2
x_1	1/8	001	3
x_4	1/8	010	3
x_6	1/8	011	3
x_2	1/16	0000	4
x_3	1/16	0001	4

$$\bar{L} = \sum_{k=1}^{7} p_k l_k$$

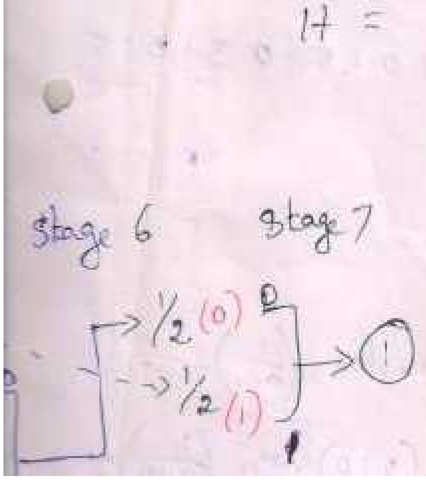
$$= \left(\tfrac{1}{4}\right)(2) + \left(\tfrac{1}{4}\right)(2) + \left(\tfrac{1}{8}\right)(3) + \left(\tfrac{1}{8}\right)(3) + \left(\tfrac{1}{8}\right)(3) + \left(\tfrac{1}{16}\right)(4) + \left(\tfrac{1}{16}\right)(4)$$

$$= 0.5 + 0.5 + 0.375 + 0.375 + 0.375 + 0.25 + 0.25$$

$\bar{L} = 2.625$ bits/symbol.

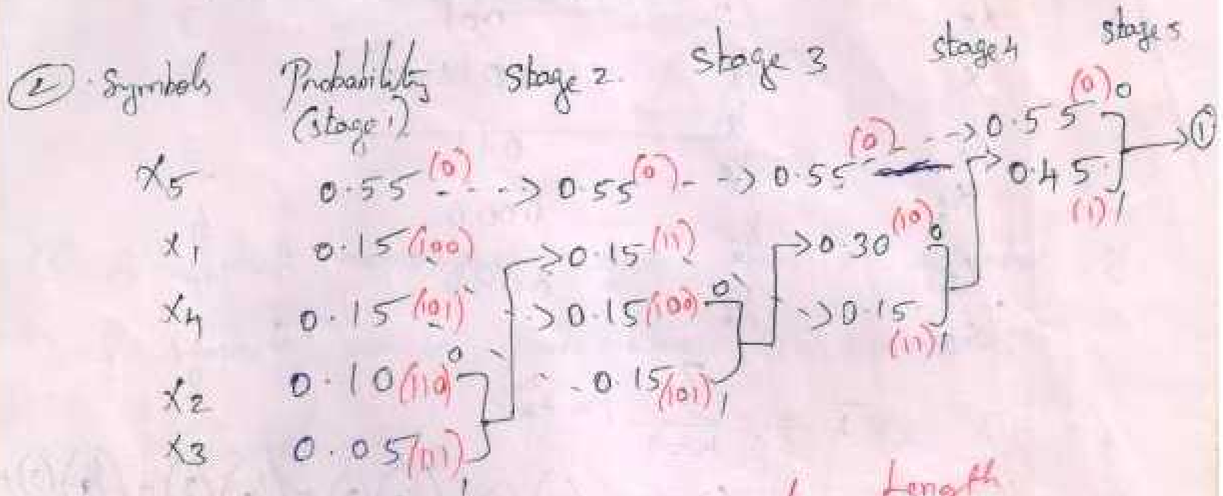
$$H = \sum_{k=1}^{K} p_k \log_2\left(\tfrac{1}{p_k}\right)$$

$$= p_1 \log_2\left(\tfrac{1}{p_1}\right) + \cdots$$



$$= \left(\tfrac{1}{4}\right)\log_2(4) + \tfrac{1}{4}\log_2(4) + \tfrac{1}{8}\log_2(8) + \tfrac{1}{8}\log_2(8) + \tfrac{1}{8}\log_2(8) + \tfrac{1}{16}\log_2(16) + \tfrac{1}{16}\log_2(16)$$

$\boxed{\eta = 100\%}$ $\eta = 1$



	Code word	Length
0.55	0	1
0.15	100	3
0.15	101	3
0.10	110	3
0.05	111	3

$$\bar{L} = \sum_{k=1}^{K} p_k L_k = (0.55)(1) + (0.15)(3) + (0.15)(3) + (0.10)(3) + (0.05)(3)$$

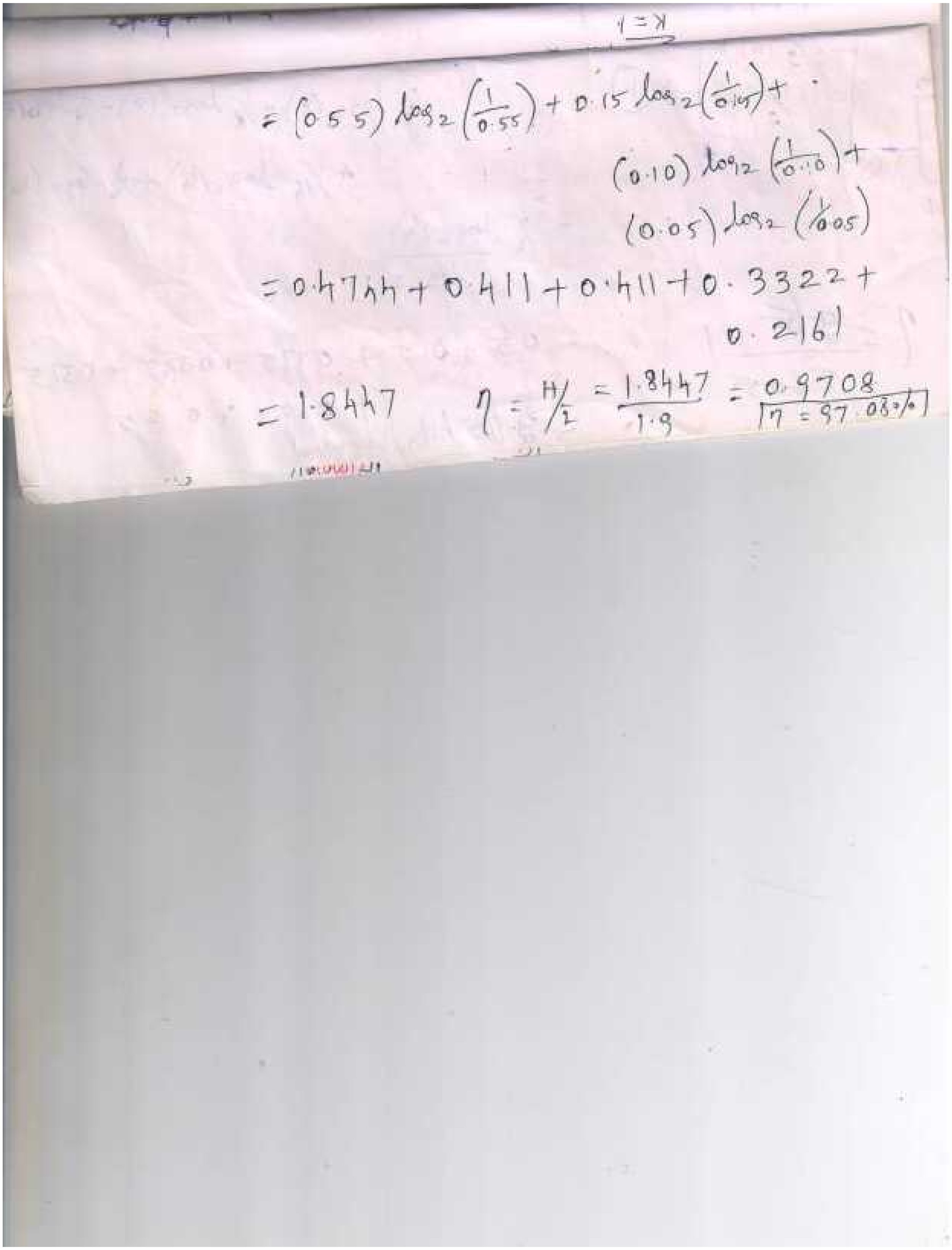
$$= 0.55 + 0.45 + 0.45 + 0.3 + 0.15$$

$$= 1.9$$

$$H = \sum_{k=1}^{K} p_k \log_2 \left(1/p_k\right)$$

$$= (0.55)\log_2\left(\frac{1}{0.55}\right) + 0.15\log_2\left(\frac{1}{0.15}\right) +$$

$$(0.10)\log_2\left(\frac{1}{0.10}\right) +$$



$$= (0.55)\log_2\left(\frac{1}{0.55}\right) + 0.15\log_2\left(\frac{1}{0.15}\right) + (0.10)\log_2\left(\frac{1}{0.10}\right) + (0.05)\log_2\left(1/0.05\right)$$

$$= 0.4744 + 0.411 + 0.411 + 0.3322 + 0.2161$$

$$= 1.8447 \qquad \eta = H/\bar{L} = \frac{1.8447}{1.9} = 0.9708$$

$$\boxed{\eta = 97.08\%}$$

Shannon's Theorem (Noiseless Coding theorem)

(i) Shannon's first theorem (or) Source coding theorem

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

(iii) Shannon's third theorem (or) Information capacity theorem (or) Shannon's Hartley ~~theo~~ theorem

(i) Shannon's first theorem:

Source coding theorem such as Shannon fano coding, Huffman coding, Prefix coding is used to remove redundancy & to improve efficiency.

$$\boxed{\eta = H/\bar{L}}$$

$$\bar{L} \geq H$$

(ii) Shannon's second theorem:

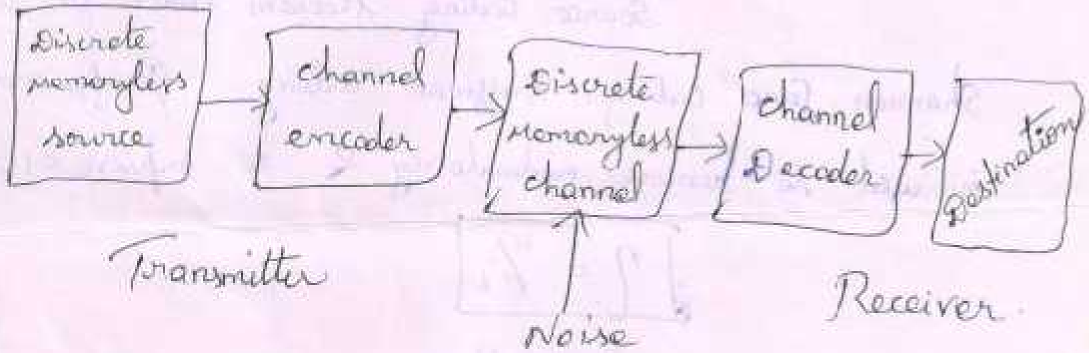
The Presence of noise in the channel can be minimized using channel coding.

For example: In wireless communication channel, the error Probability may be 10^{-1} ($9/10$ received correctly). But in many applications, this level of reliability is unacceptable. A Probability is often a necessary requirement.

To achieve this requirement, we are using channel coding.

The goal is to increase the resistance of digital communication system to channel noise.

Process:

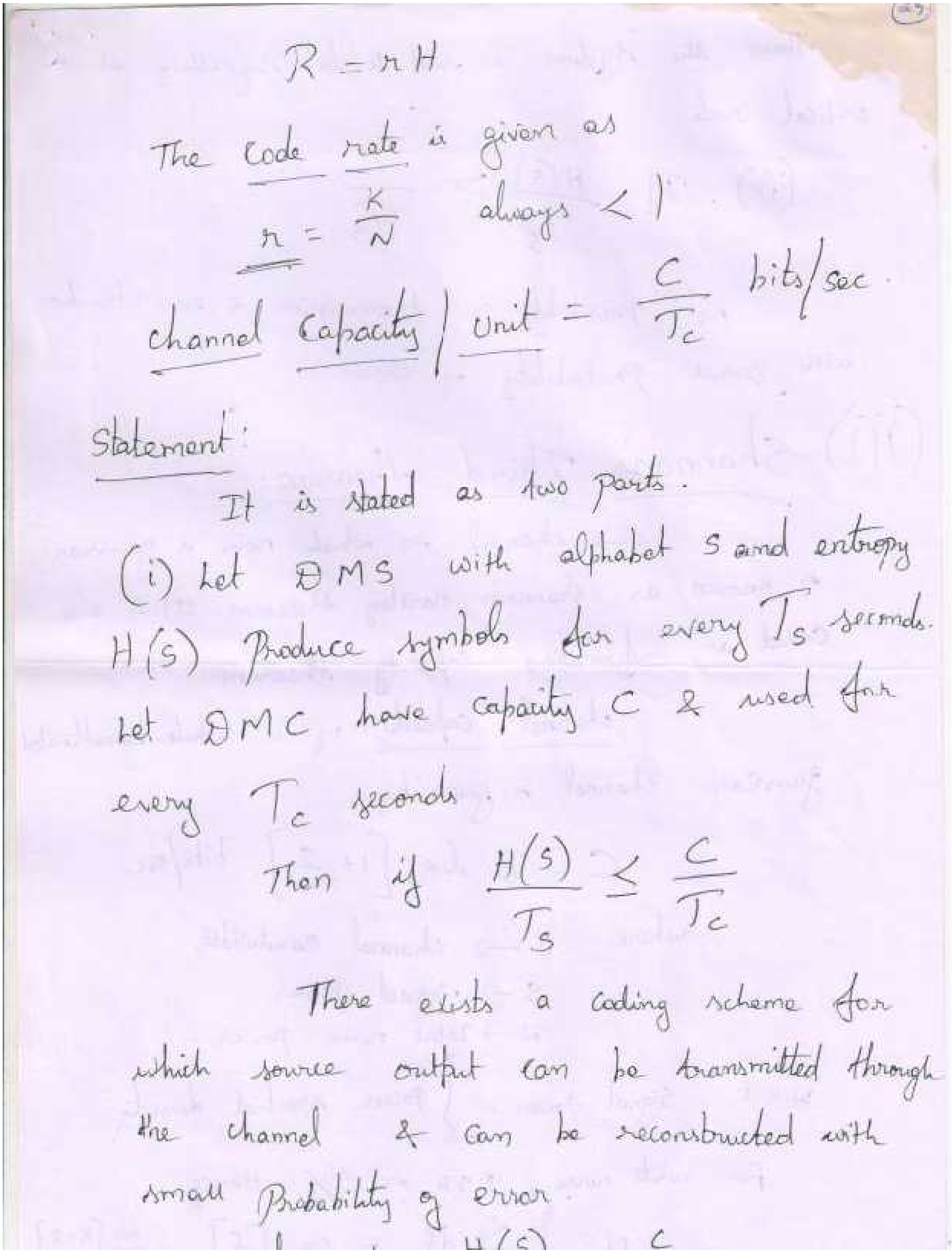


Approach:

To introduce redundancy in channel encoder so as to reconstruct the original source sequence as accurately as possible

This theorem says that if

$R \leq C$, it is possible to transmit information without any error even



$$R = rH$$

The code rate is given as

$$r = \frac{K}{N} \quad \text{always} < 1$$

$$\text{channel capacity / unit} = \frac{C}{T_c} \text{ bits/sec}$$

Statement:

It is stated as two parts.

(i) Let DMS with alphabet S and entropy H(S) produce symbols for every T_s seconds. Let DMC have capacity C & used for every T_c seconds.

Then if $$\frac{H(S)}{T_s} \leq \frac{C}{T_c}$$

There exists a coding scheme for which source output can be transmitted through the channel & can be reconstructed with small probability of error.

$H(S)$ C

Then the system is said to be signalling at a critical rate.

(ii) If $\frac{H(S)}{T_s} > \frac{C}{T_s}$

No possibility of transmission & reconstruction with small probability of error.

(iii) Shannon's Third theorem:

→ channel in which noise is gaussian is known as shannon's Hartley theorem. It is also called as information capacity theorem.

channel capacity of a white bandlimited gaussian channel is given by,

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

where B → channel Bandwidth
S → signal power
N → Total noise power

w.k.t Signal power $= \int_{-B}^{B}$ Power spectral density.

For white noise, PSD is $N_0/2$. Hence

$$N = \int^{B} \frac{N_0}{} df = \frac{N_0}{} \left[f \right]^{B} = \frac{N_0}{2}[B+B]$$

Bandwidth - S/N Trade off :

→ channel capacity of the Gaussian channel is given by,

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

from the above equation, it is clear that the channel capacity depends on two factors.

(i) Band width

(ii) signal to noise ratio.

Noiseless channel has infinite capacity :

If there is no noise in the channel, then $N = 0$. Hence $S/N = \infty$. Such a channel is called Noiseless channel. Then capacity of such a channel will be

$$C = B \log_2 \left[1 + \infty\right] = \infty .$$

Thus the noiseless channel has infinite capacity.

Infinite bandwidth channel has limited capacity :

→ Now if the bandwidth 'B' is infinite,

the channel capacity is limited. This is because, as bandwidth increases, noise power (N) also increases.

Noise Power is given as,

$$N = N_0 B.$$

★ Due to this increase in noise power, S/N ratio decreases. Hence even if B approaches infinity, capacity does not approach infinity.

As $B \to \infty$, Capacity approaches an upper limit. It is given as,

$$\lim_{B \to \infty} (c) = S/N \log_2 e$$

$$\boxed{\lim_{B \to \infty} (c) = 1.44\ S/N}$$

1/1/2002

Data Communications and Networking Fourth Edition

Forouzan

Chapter 4

Digital Transmission

4-1 DIGITAL-TO-DIGITAL CONVERSION

In this section, we see how we can represent digital data by using digital signals. The conversion involves three techniques: line coding, block coding, and scrambling. Line coding is always needed; block coding and scrambling may or may not be needed.

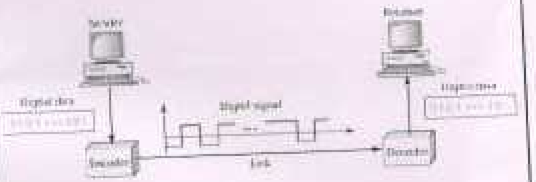
Topics discussed in this section:

- Line Coding
- Line Coding Schemes
- Block Coding
- Scrambling

Line Coding

- Converting a string of 1's and 0's (digital data) into a sequence of signals that denote the 1's and 0's.
- For example a high voltage level (+V) could represent a "1" and a low voltage level (0 or -V) could represent a "0".

Figure 4.1 *Line coding and decoding*



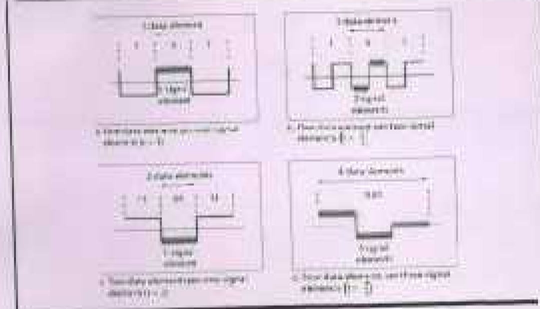
Mapping Data symbols onto Signal levels

- A data symbol (or element) can consist of a number of data bits:
  - 1 , 0 or
  - 11, 10, 01, ...
- A data symbol can be coded into a single signal element or multiple signal elements
  - 1 -> +V, 0 -> -V
  - 1 -> +V and -V, 0 -> -V and +V
- The ratio 'r' is the number of data elements carried by a signal element.

Relationship between data rate and signal rate

- The data rate defines the number of bits sent per sec - bps. It is often referred to the bit rate.
- The signal rate is the number of signal elements sent in a second and is measured in bauds. It is also referred to as the modulation rate.
- Goal is to increase the data rate whilst reducing the baud rate.

Figure 4.2 *Signal element versus data element*



Data rate and Baud rate

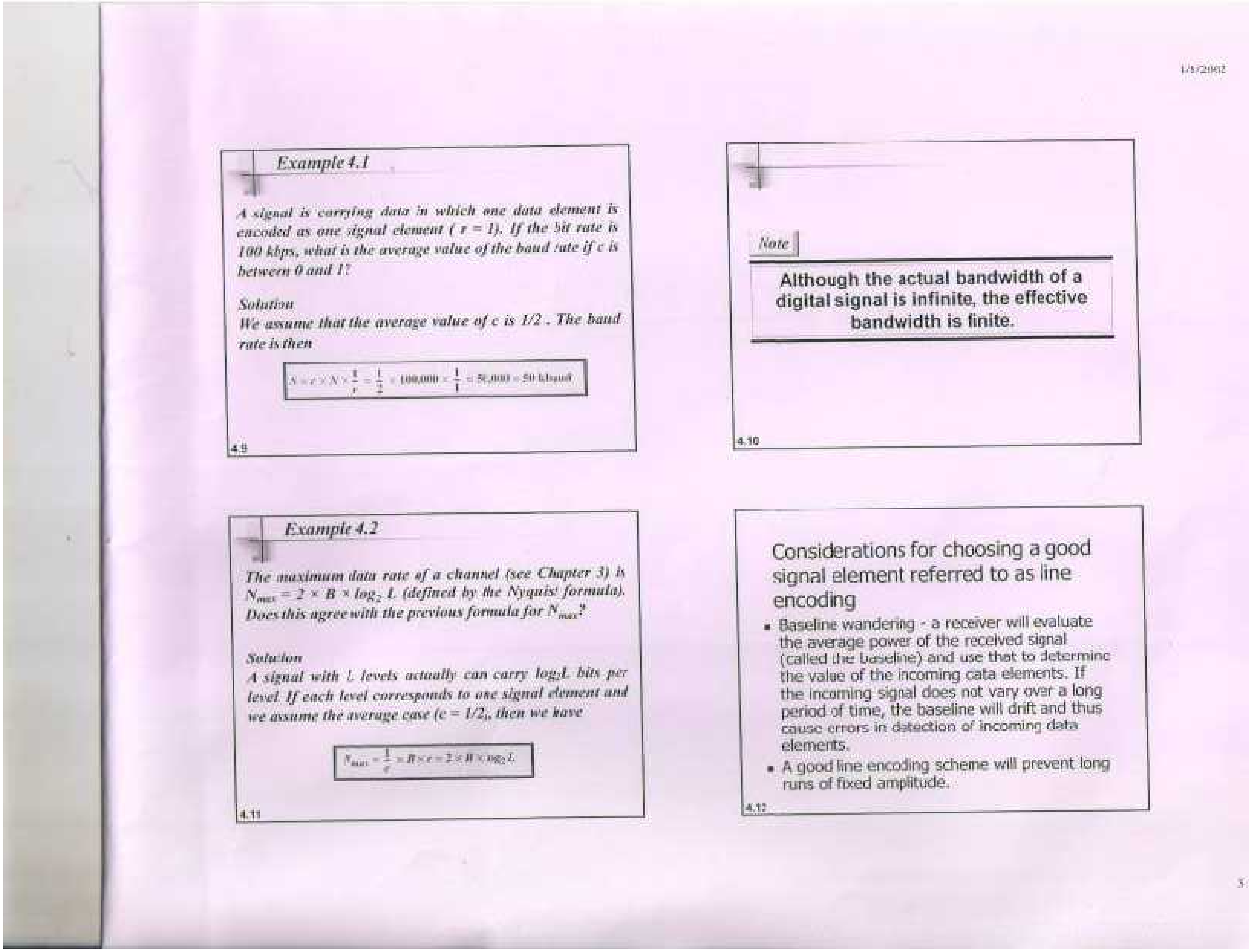
- The baud or signal rate can be expressed as:

$$S = c \times N \times 1/r \text{ bauds}$$

where N is data rate

c is the case factor (worst, best & avg.)

r is the ratio between data element & signal element



1/1/2002

Example 4.1

A signal is carrying data in which one data element is encoded as one signal element (r = 1). If the bit rate is 100 kbps, what is the average value of the baud rate if c is between 0 and 1?

Solution

We assume that the average value of c is 1/2 . The baud rate is then

$$S = c \times N \times \frac{1}{r} = \frac{1}{2} \times 100{,}000 \times \frac{1}{1} = 50{,}000 = 50 \text{ kbaud}$$

4.9

Note

Although the actual bandwidth of a digital signal is infinite, the effective bandwidth is finite.

4.10

Example 4.2

The maximum data rate of a channel (see Chapter 3) is $N_{max} = 2 \times B \times \log_2 L$ (defined by the Nyquist formula). Does this agree with the previous formula for N_{max}?

Solution

A signal with L levels actually can carry $\log_2 L$ bits per level. If each level corresponds to one signal element and we assume the average case (c = 1/2), then we have

$$N_{max} = \frac{1}{c} \times B \times r = 2 \times B \times \log_2 L$$

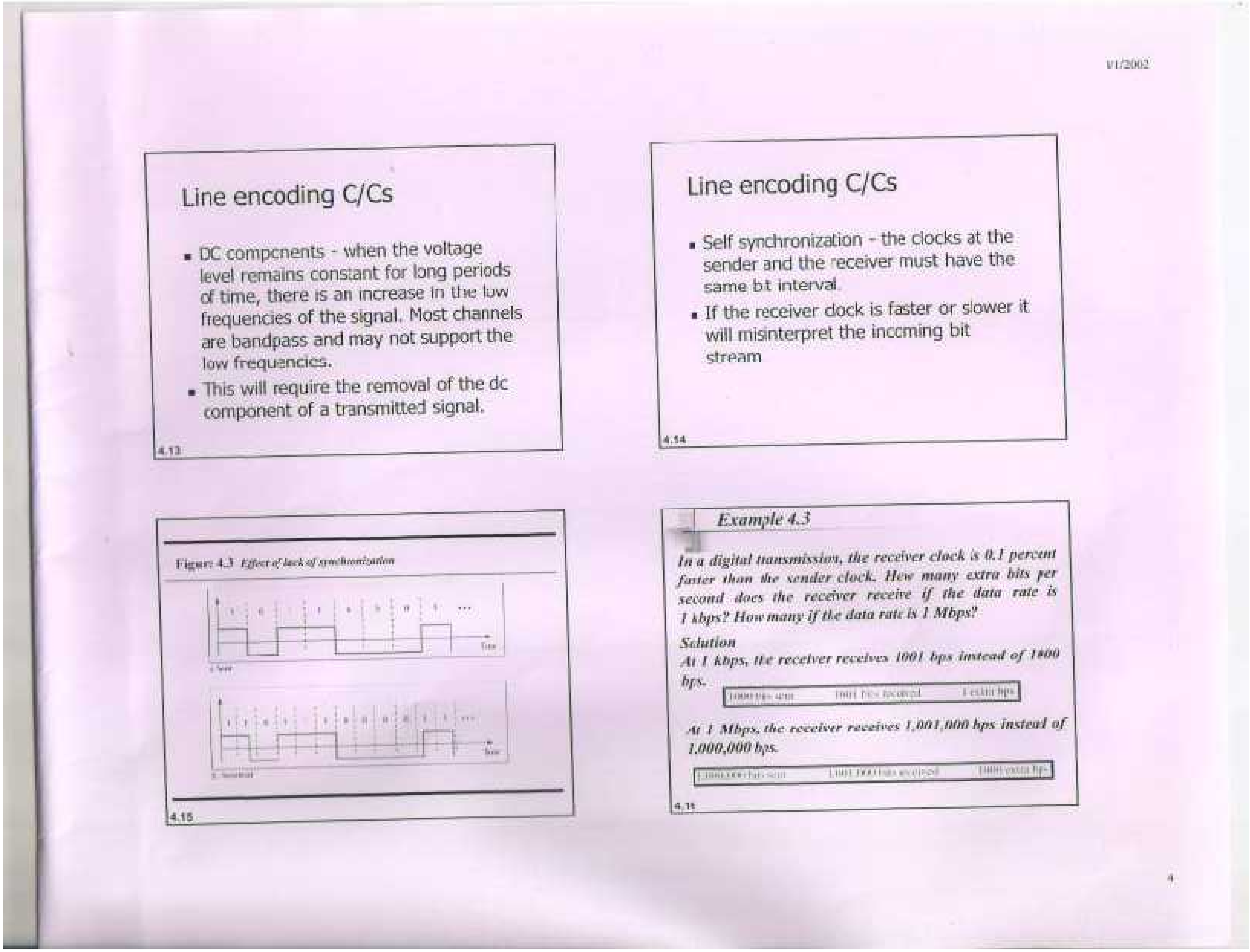
4.11

Considerations for choosing a good signal element referred to as line encoding

- Baseline wandering - a receiver will evaluate the average power of the received signal (called the baseline) and use that to determine the value of the incoming data elements. If the incoming signal does not vary over a long period of time, the baseline will drift and thus cause errors in detection of incoming data elements.
- A good line encoding scheme will prevent long runs of fixed amplitude.

4.12

3



Line encoding C/Cs

- DC components - when the voltage level remains constant for long periods of time, there is an increase in the low frequencies of the signal. Most channels are bandpass and may not support the low frequencies.
- This will require the removal of the dc component of a transmitted signal.

Line encoding C/Cs

- Self synchronization - the clocks at the sender and the receiver must have the same bit interval.
- If the receiver dock is faster or slower it will misinterpret the incoming bit stream

Figure 4.3 Effect of lack of synchronization

Example 4.3

In a digital transmission, the receiver clock is 0.1 percent faster than the sender clock. How many extra bits per second does the receiver receive if the data rate is 1 kbps? How many if the data rate is 1 Mbps?

Solution

At 1 kbps, the receiver receives 1001 bps instead of 1000 bps.

1000 bits sent	1001 bits received	1 extra bps

At 1 Mbps, the receiver receives 1,001,000 bps instead of 1,000,000 bps.

1,000,000 bits sent	1,001,000 bits received	1000 extra bps

Line encoding C/Cs

- Error detection - errors occur during transmission due to line impairments.
- Some codes are constructed such that when an error occurs it can be detected. For example: a particular signal transition is not part of the code. When it occurs, the receiver will know that a symbol error has occurred.

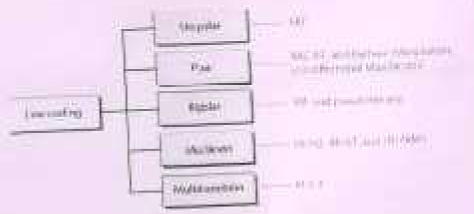
Line encoding C/Cs

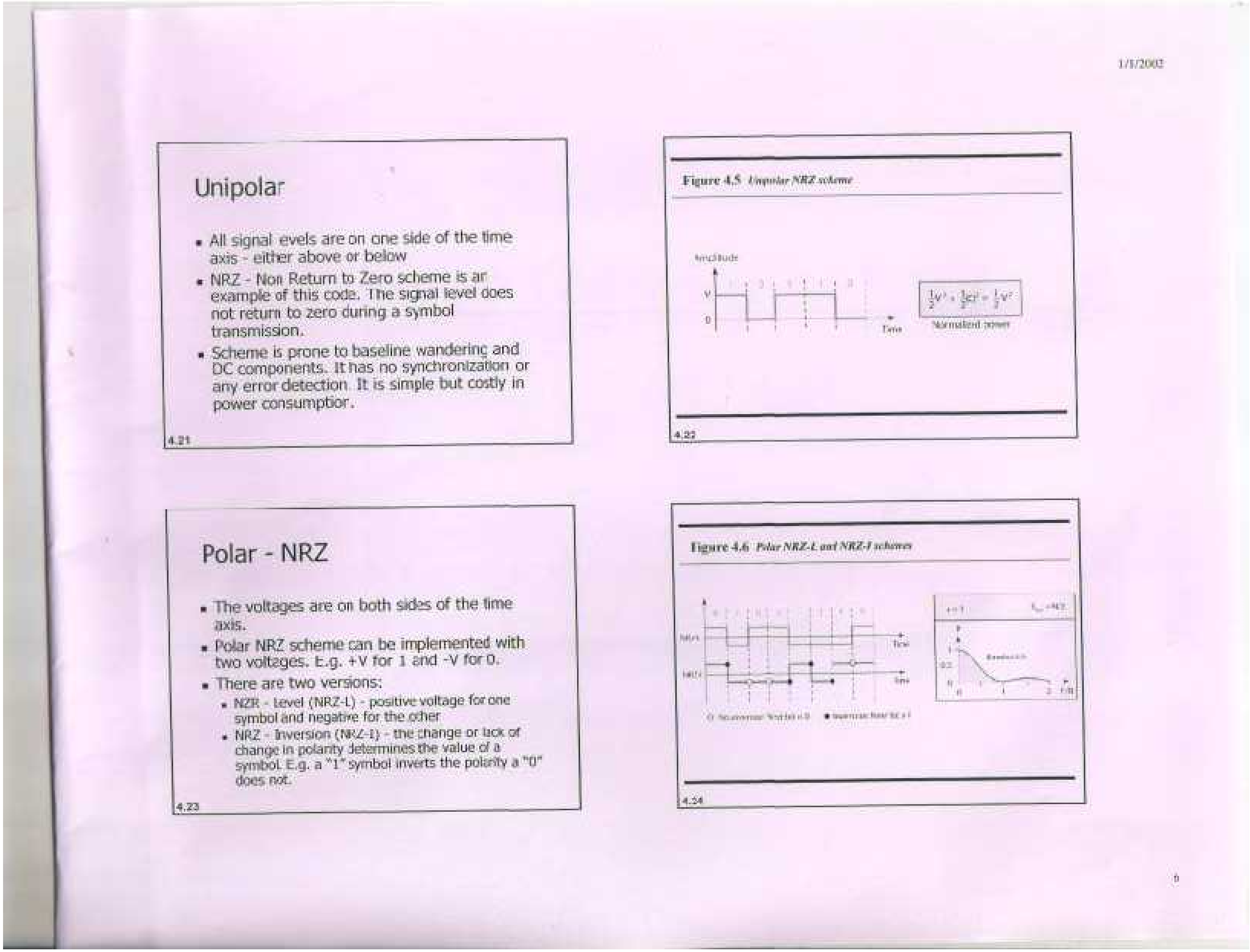
- Nose and interference - there are line encoding techniques that make the transmitted signal "immune" to noise and interference.
- This means that the signal cannot be corrupted, it is stronger than error detection.

Line encoding C/Cs

- Complexity - the more robust and resilient the code, the more complex it is to implement and the price is often paid in baud rate or required bandwidth.

Figure 4.4 *Line coding schemes*





1/1/2002

Unipolar

- All signal levels are on one side of the time axis - either above or below
- NRZ - Non Return to Zero scheme is an example of this code. The signal level does not return to zero during a symbol transmission.
- Scheme is prone to baseline wandering and DC components. It has no synchronization or any error detection. It is simple but costly in power consumption.

4.21

Figure 4.5 Unipolar NRZ scheme

4.22

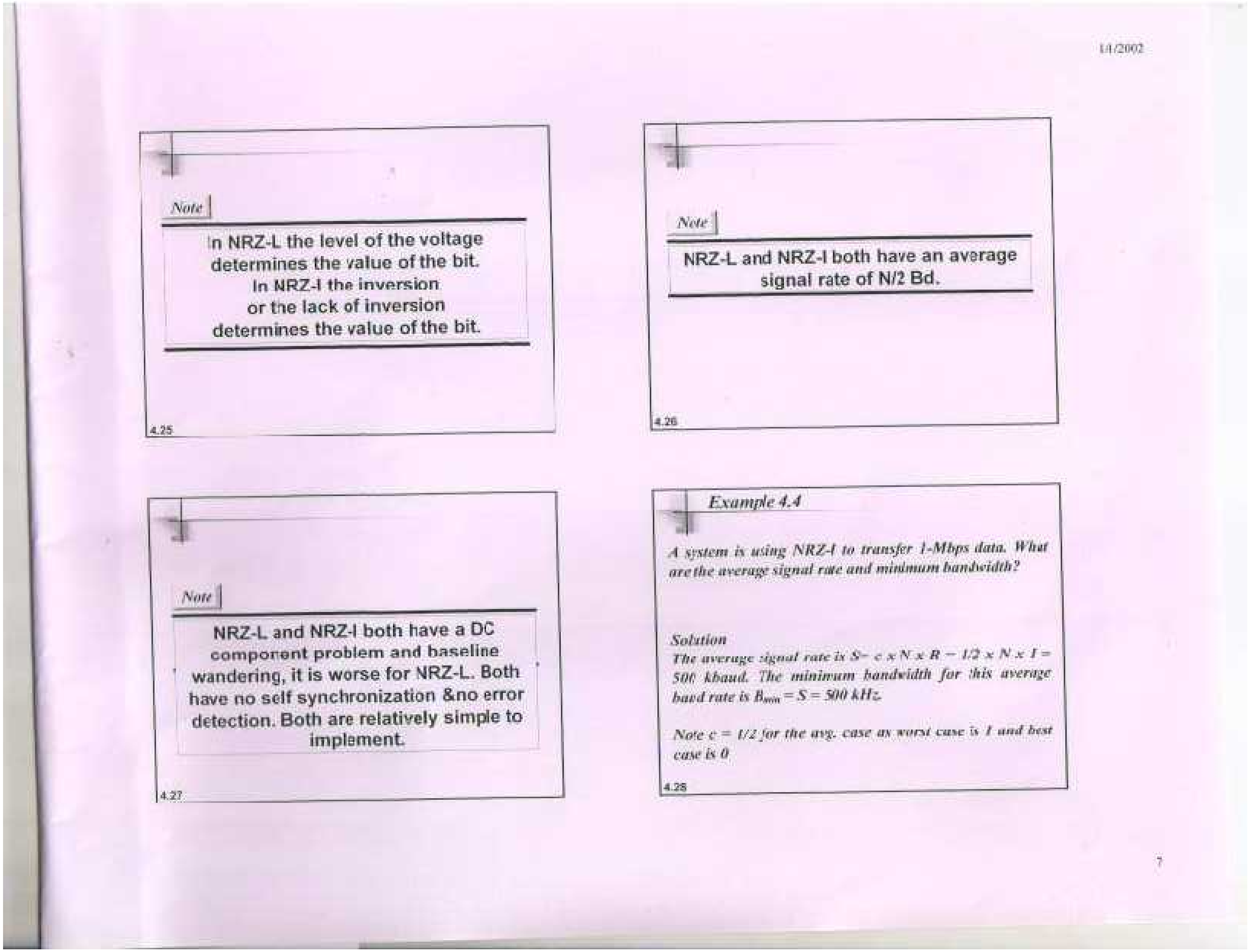
Polar - NRZ

- The voltages are on both sides of the time axis.
- Polar NRZ scheme can be implemented with two voltages. E.g. +V for 1 and -V for 0.
- There are two versions:
  - NZR - Level (NRZ-L) - positive voltage for one symbol and negative for the other
  - NRZ - Inversion (NRZ-I) - the change or lack of change in polarity determines the value of a symbol. E.g. a "1" symbol inverts the polarity a "0" does not.

4.23

Figure 4.6 Polar NRZ-L and NRZ-I schemes

4.24



1/1/2002

Note

In NRZ-L the level of the voltage determines the value of the bit.
In NRZ-I the inversion
or the lack of inversion
determines the value of the bit.

4.25

Note

NRZ-L and NRZ-I both have an average signal rate of N/2 Bd.

4.26

Note

NRZ-L and NRZ-I both have a DC component problem and baseline wandering, it is worse for NRZ-L. Both have no self synchronization &no error detection. Both are relatively simple to implement.

4.27

Example 4.4

A system is using NRZ-I to transfer 1-Mbps data. What are the average signal rate and minimum bandwidth?

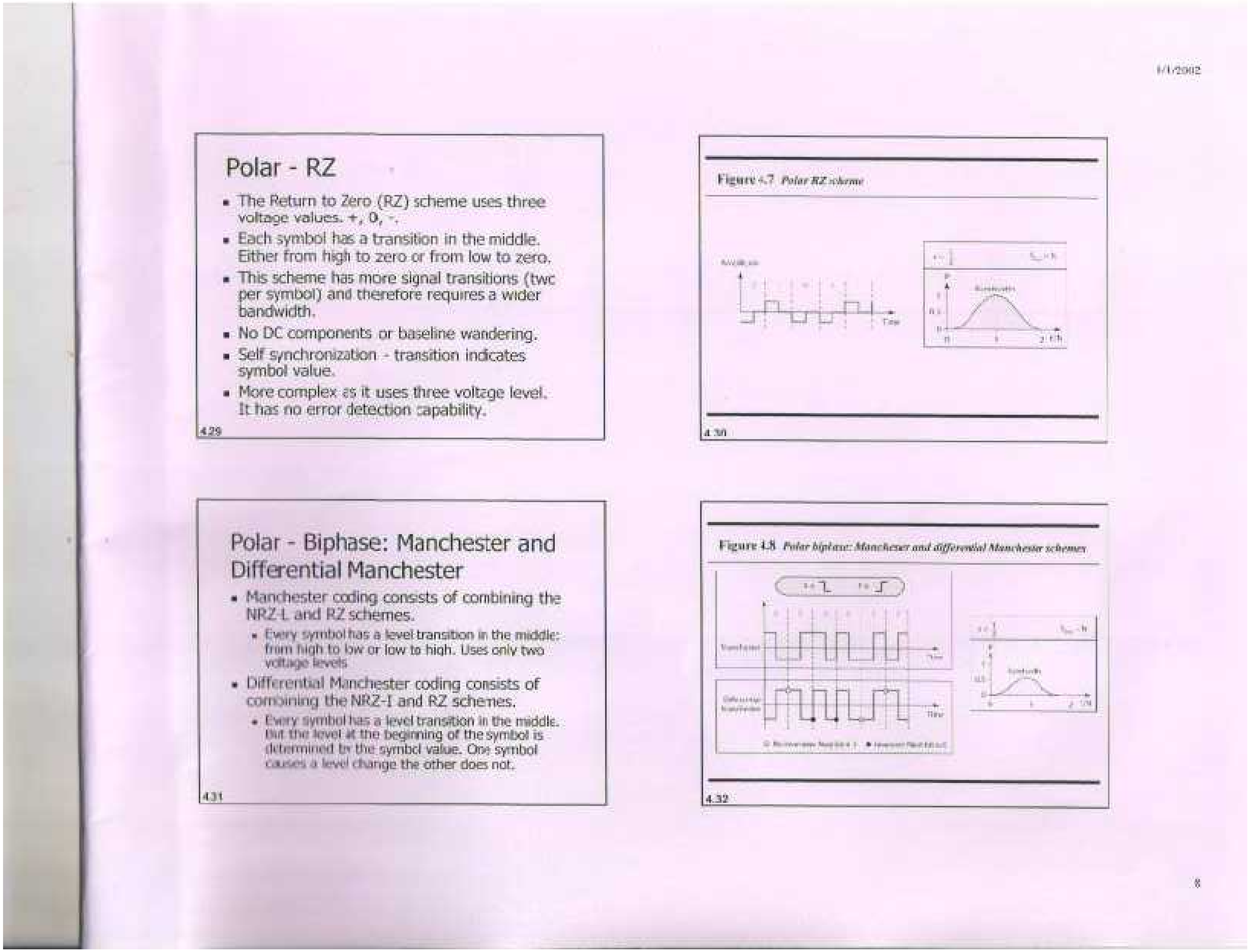
Solution

The average signal rate is $S = c \times N \times R = 1/2 \times N \times 1 =$ 500 kbaud. The minimum bandwidth for this average baud rate is $B_{min} = S = 500$ kHz.

Note $c = 1/2$ for the avg. case as worst case is 1 and best case is 0

4.28

7



1/1/2002

Polar - RZ

- The Return to Zero (RZ) scheme uses three voltage values. +, 0, -.
- Each symbol has a transition in the middle. Either from high to zero or from low to zero.
- This scheme has more signal transitions (twc per symbol) and therefore requires a wider bandwidth.
- No DC components or baseline wandering.
- Self synchronization - transition indicates symbol value.
- More complex as it uses three voltage level. It has no error detection capability.

4.29

Figure 4.7 *Polar RZ scheme*

4.30

Polar - Biphase: Manchester and Differential Manchester

- Manchester coding consists of combining the NRZ-L and RZ schemes.
  - Every symbol has a level transition in the middle: from high to low or low to high. Uses only two voltage levels
- Differential Manchester coding consists of combining the NRZ-I and RZ schemes.
  - Every symbol has a level transition in the middle. But the level at the beginning of the symbol is determined by the symbol value. One symbol causes a level change the other does not.

4.31

Figure 4.8 *Polar biphase: Manchester and differential Manchester schemes*

4.32

8

Note

In Manchester and differential Manchester encoding, the transition at the middle of the bit is used for synchronization.

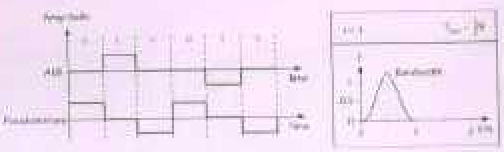
Note

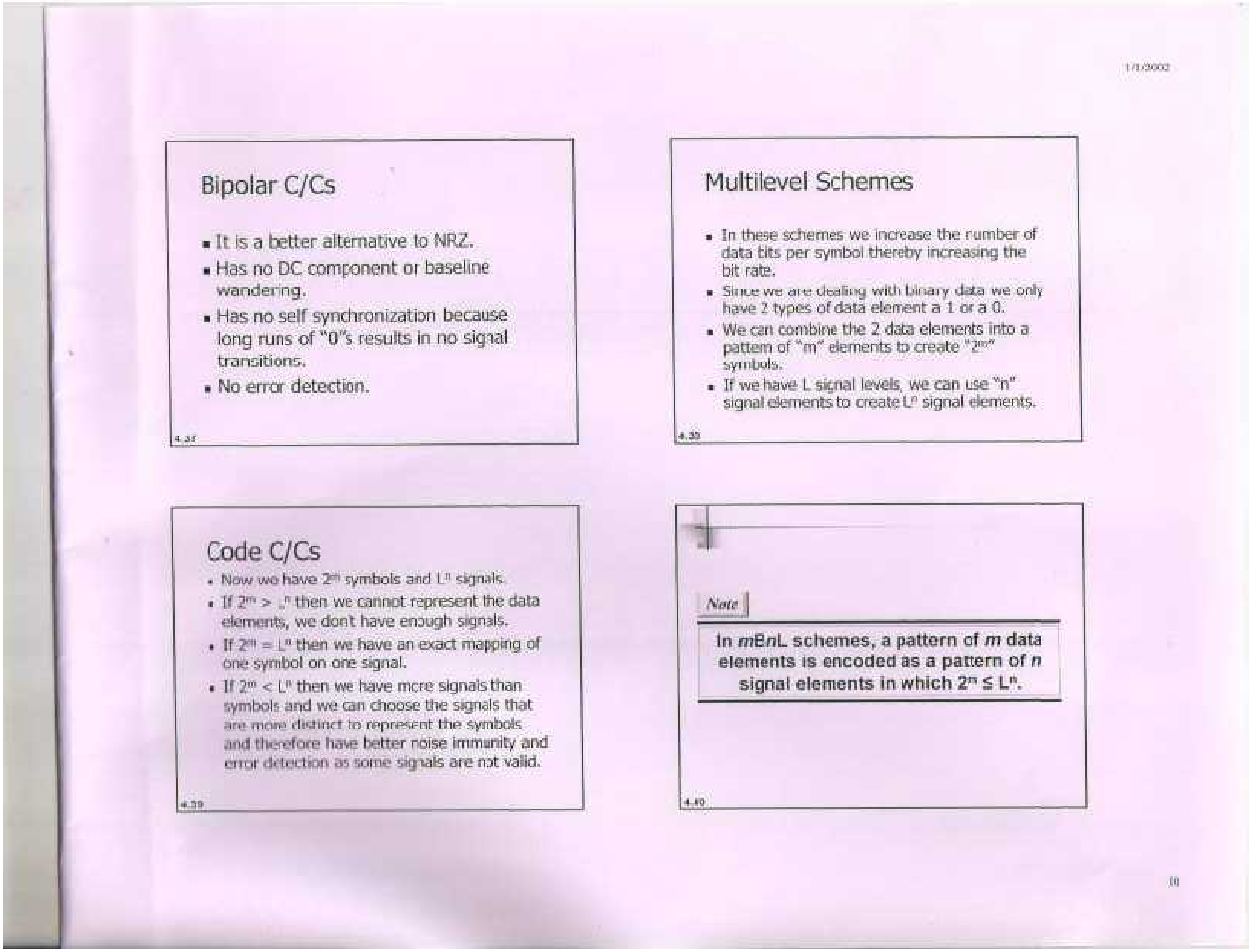
The minimum bandwidth of Manchester and differential Manchester is 2 times that of NRZ. The is no DC component and no baseline wandering. None of these codes has error detection.

Bipolar - AMI and Pseudoternary

- Code uses 3 voltage levels: - +, 0, -, to represent the symbols (note not transitions to zero as in RZ).
- Voltage level for one symbol is at "0" and the other alternates between + & -.
- Bipolar Alternate Mark Inversion (AMI) - the "0" symbol is represented by zero voltage and the "1" symbol alternates between +V and -V.
- Pseudoternary is the reverse of AMI.

Figure 4.9 *Bipolar schemes: AMI and pseudoternary*





1/1/2002

Bipolar C/Cs

- It is a better alternative to NRZ.
- Has no DC component or baseline wandering.
- Has no self synchronization because long runs of "0"s results in no signal transitions.
- No error detection.

4.37

Multilevel Schemes

- In these schemes we increase the number of data bits per symbol thereby increasing the bit rate.
- Since we are dealing with binary data we only have 2 types of data element a 1 or a 0.
- We can combine the 2 data elements into a pattern of "m" elements to create "2^m" symbols.
- If we have L signal levels, we can use "n" signal elements to create L^n signal elements.

4.38

Code C/Cs

- Now we have 2^m symbols and L^n signals.
- If $2^m > L^n$ then we cannot represent the data elements, we don't have enough signals.
- If $2^m = L^n$ then we have an exact mapping of one symbol on one signal.
- If $2^m < L^n$ then we have more signals than symbols and we can choose the signals that are more distinct to represent the symbols and therefore have better noise immunity and error detection as some signals are not valid.

4.39

Note

In *m*B*n*L schemes, a pattern of *m* data elements is encoded as a pattern of *n* signal elements in which $2^m \leq L^n$.

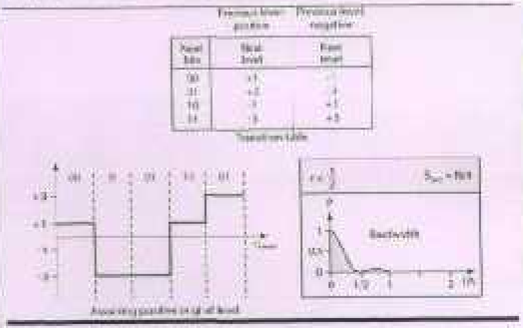
4.40

10

Representing Multilevel Codes

- We use the notation mBnL, where m is the length of the binary pattern, B represents binary data, n represents the length of the signal pattern and L the number of levels.
- L = B binary, L = T for 3 ternary, L = Q for 4 quaternary.

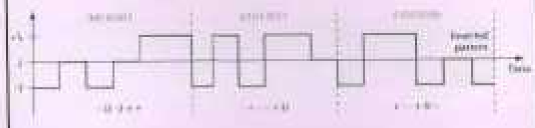
Figure 4.10 *Multilevel: 2B1Q scheme*



Redundancy

- In the 2B1Q scheme we have no redundancy and we see that a DC component is present.
- If we use a code with redundancy we can decide to use only "0" or "+" weighted codes (more +'s than -'s in the signal element) and invert any code that would create a DC component. E.g. '+00++-' -> '-00--+'
- Receiver will know when it receives a "-" weighted code that it shoud invert it as it doesn't represent any valid symbol.

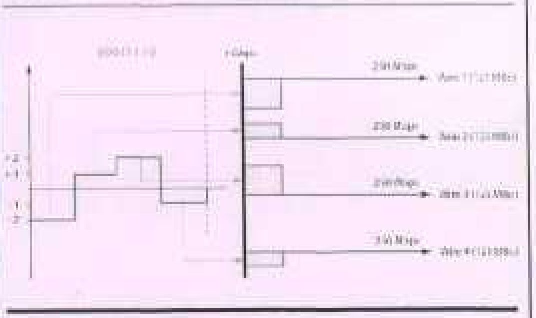
Figure 4.11 *Multilevel: 8B6T scheme*



Multilevel using multiple channels

- In some cases, we split the signal transmission up and distribute it over several links.
- The separate segments are transmitted simultaneously. This reduces the signalling rate per link -> lower bandwidth.
- This requires all bits for a code to be stored.
- xD: means that we use 'x' links
- YYYz: We use 'z' levels of modulation where YYY represents the type of modulation (e.g. pulse ampl. mod. PAM).
- Codes are represented as: xD-YYYz

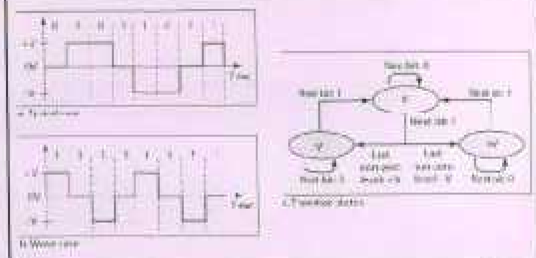
Figure 4.12 *Multilevel: 4D-PAM5 scheme*

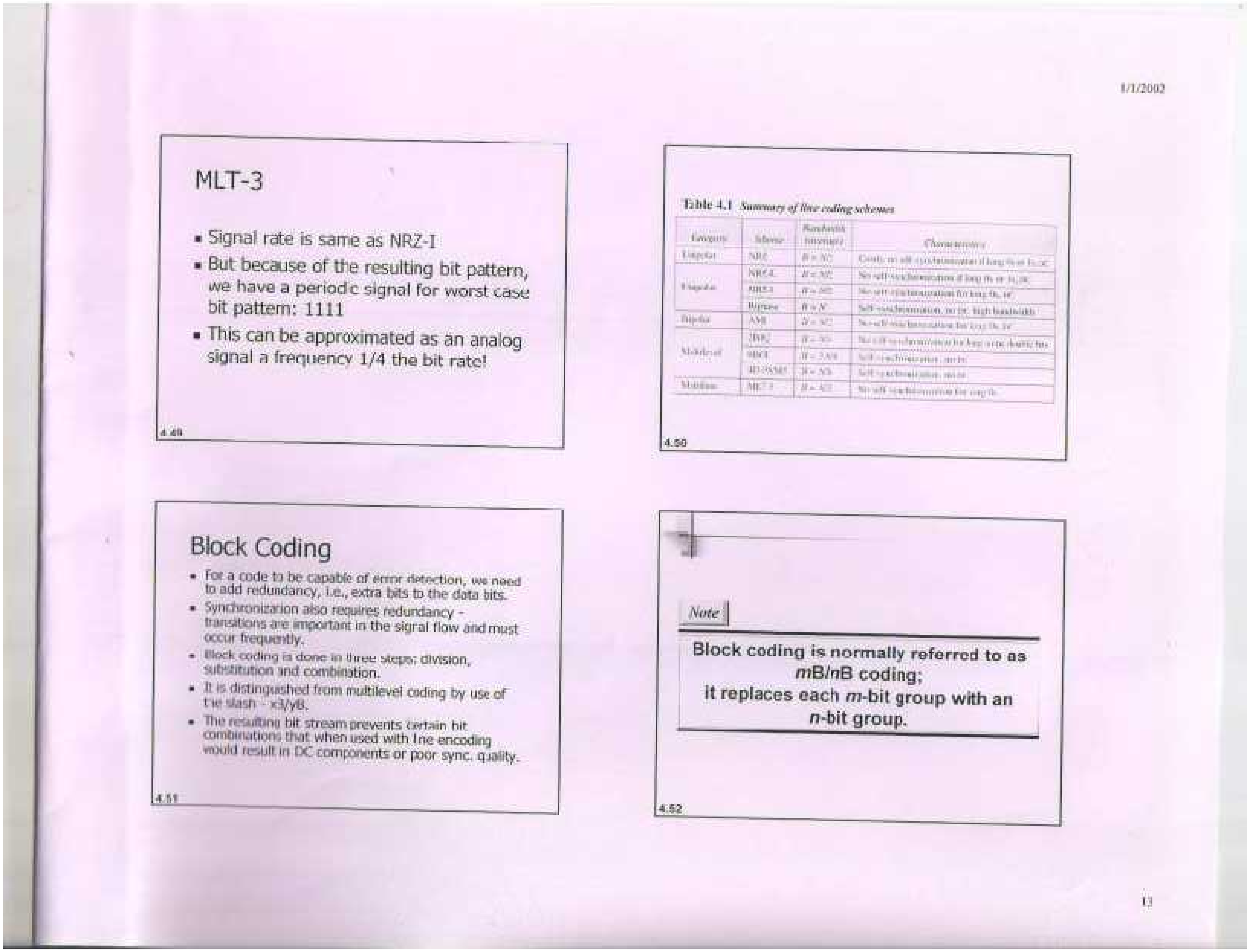


Multitransition Coding

- Because of synchronization requirements we force transitions. This can result in very high bandwidth requirements -> more transitions than are bits (e.g. mid bit transition with inversion).
- Codes can be created that are differential at the bit level forcing transitions at bit boundaries. This results in a bandwidth requirement that is equivalent to the bit rate.
- In some instances, the bandwidth requirement may even be lower, due to repetitive patterns resulting in a periodic signal.

Figure 4.13 *Multitransition: MLT-3 scheme*





1/1/2002

MLT-3

- Signal rate is same as NRZ-I
- But because of the resulting bit pattern, we have a periodic signal for worst case bit pattern: 1111
- This can be approximated as an analog signal a frequency 1/4 the bit rate!

4.49

Table 4.1 *Summary of line coding schemes*

Category	Scheme	Bandwidth (average)	Characteristics
Unipolar	NRZ	B = N/2	Costly, no self-synchronization if long 0s or 1s, DC
Unipolar	NRZ-L	B = N/2	No self-synchronization if long 0s or 1s, DC
	NRZ-I	B = N/2	No self-synchronization for long 0s, DC
	Biphase	B = N	Self-synchronization, no DC, high bandwidth
Bipolar	AMI	B = N/2	No self-synchronization for long 0s, DC
Multilevel	2B1Q	B = N/4	No self-synchronization for long same double bits
	8B6T	B = 3N/4	Self-synchronization, no DC
	4D-PAM5	B = N/8	Self-synchronization, no DC
Multiline	MLT-3	B = N/3	No self-synchronization for long 0s

4.50

Block Coding

- For a code to be capable of error detection, we need to add redundancy, i.e., extra bits to the data bits.
- Synchronization also requires redundancy - transitions are important in the signal flow and must occur frequently.
- Block coding is done in three steps: division, substitution and combination.
- It is distinguished from multilevel coding by use of the slash - xB/yB.
- The resulting bit stream prevents certain bit combinations that when used with line encoding would result in DC components or poor sync. quality.

4.51

Note

Block coding is normally referred to as *m*B/*n*B coding; it replaces each *m*-bit group with an *n*-bit group.

4.52

13

Figure 4.14 *Block coding concept*

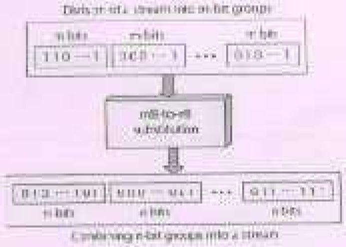


Figure 4.15 *Using block coding 4B/5B with NRZ-I line coding scheme*

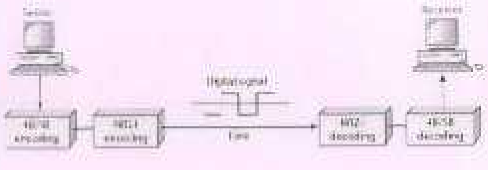
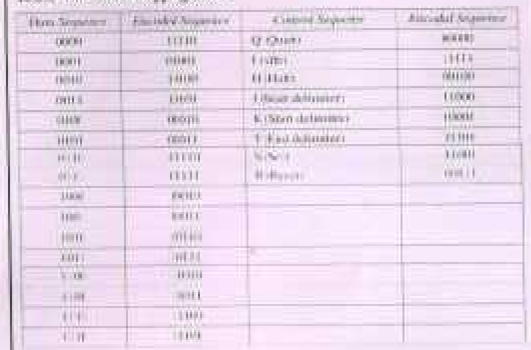
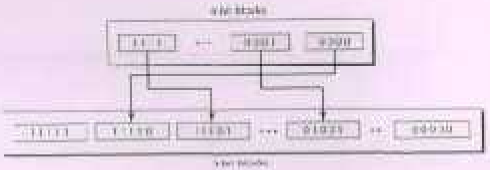


Table 4.2 *4B/5B mapping codes*



Data Sequence	Encoded Sequence	Control Sequence	Encoded Sequence
0000	11110	Q (Quiet)	00000
0001	01001	I (Idle)	11111
0010	10100	H (Halt)	00100
0011	10101	J (Start delimiter)	11000
0100	01010	K (Start delimiter)	10001
0101	01011	T (End delimiter)	01101
0110	01110	S (Set)	11001
0111	01111	R (Reset)	00111
1000	10010		
1001	10011		
1010	10110		
1011	10111		
1100	11010		
1101	11011		
1110	11100		
1111	11101		

Figure 4.16 *Substitution in 4B/5B block coding*



Redundancy

- A 4 bit data word can have 24 combinations.
- A 5 bit word can have 25=32 combinations.
- We therefore have 32 - 26 = 16 extra words.
- Some of the extra words are used for control/signalling purposes.

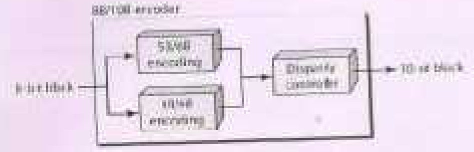
Example 4.5

We need to send data at a 1-Mbps rate. What is the minimum required bandwidth, using a combination of 4B/5B and NRZ-I or Manchester coding?

Solution

First 4B/5B block coding increases the bit rate to 1.25 Mbps. The minimum bandwidth using NRZ-I is N/2 or 625 kHz. The Manchester scheme needs a minimum bandwidth of 1.25 MHz. The first choice needs a lower bandwidth, but has a DC component problem; the second choice needs a higher bandwidth, but does not have a DC component problem.

Figure 4.17 *8B/10B block encoding*



More bits - better error detection

- The 8B10B block code adds more redundant bits and can thereby choose code words that would prevent a long run of a voltage level that would cause DC components.

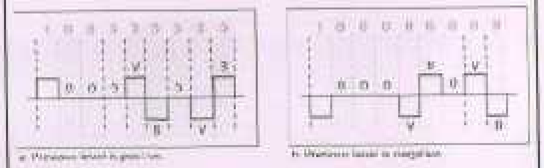
Scrambling

- The best code is one that does not increase the bandwidth for synchronization and has no DC components.
- Scrambling is a technique used to create a sequence of bits that has the required c/c's for transmission - self clocking, no low frequencies, no wide bandwidth.
- It is implemented at the same time as encoding, the bit stream is created on the fly.
- It replaces 'unfriendly' runs of bits with a violation code that is easy to recognize and removes the unfriendly c/c.

Figure 4.18 *AMI used with scrambling*

For example: B8ZS substitutes eight consecutive zeros with 000VB0VB.
The V stands for violation, it violates the line encoding rule
B stands for bipolar, it implements the bipolar line encoding rule

Figure 4.19 *Two cases of B8ZS scrambling technique*

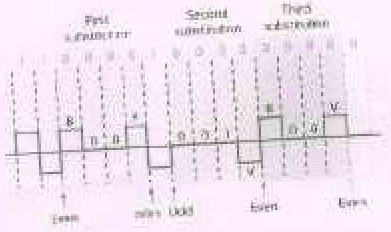


HDB3 substitutes four consecutive zeros with 000V or B00V depending on the number of nonzero pulses after the last substitution.

If # of non zero pulses is even the substitution is B00V to make total # of non zero pulse even.

If # of non zero pulses is odd the substitution is 000V to make total # of non zero pulses even.

Figure 4.20 *Different situations in HDB3 scrambling technique*



Error Control Codes

→ Mechanism involves both error detection & error Correction.

→ to improve data quality.

Two error control codes

- Block codes
- Convolutional Codes.

Terms

(i) ~~Block~~ (or) Code word : → It Consists of 'n' number of bits. This CW contain Message bits & redundant bits

(ii) Block length: The number of bits 'n' after Coding

(iii) Code rate : $r = \frac{K \text{ (Message bits)}}{n \text{ (encoder o/p bits)}}$ $0 < r < 1$

(iv) channel data rate : → It is the bit rate at the o/p of the encoder. If the bit rate at the i/p of the encoder is R_s, then

$$R_o = \left[\frac{n}{K}\right] R_s .$$

(v) Code Vectors :

0	0	0
0	0	1
0	1	0
0	1	1
1	0	0

(vi) Hamming distance

Eg: $A = (1\,1\,1)$, $B = (1\,1\,0)$. The two code vectors differ in third bit. Therefore hamming distance b/w A & B is 'one'.

$$\boxed{d = 1.}$$

In 3 bit code vectors, Max. hamming distance is 3.

Eg: $A = (1\,1\,1)$, $B = (0\,1\,0)$.

$$\boxed{d = 2}.$$

(vii) Minimum distance (d_{min}):

$$\boxed{d_{min} = n - k + 1}$$

(viii) weight of the code

The number of non-zero elements in the transmitted code vector is a code weight.

for eg: $A = 1101000 1$,

$$\boxed{W(A) = 4}$$

Linear block Codes:

Defn ⟹ In channel encoder, a block of 'k' message bits are encoded into a block of 'n' bits by adding (n-k) number of parity bits with condition n>k. Such a code formed is called (n,k) block code.

A block code ~~[illegible]~~ is said to be (n,k) linear block code, if it satisfies the

(i) Linear Property → Let C_1 & C_2 be any two code words of 'n' bits belonging to a set of (n, k) block code. If $C_1 \oplus C_2$ is also an n-bit code word belonging to the same set of (n, k) block code, then such a code is called (n, k) linear block code

(ii) systematic Property

(i) Linear Property.

Let us consider two code ~~vectors~~ words of 'n' bits

C_1 & C_2 ⟹ $C_1 = 1111$, $C_2 = 0000$.

→ belonging to a set of block codes

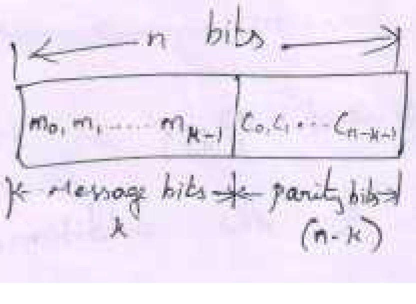
$C_1 \oplus C_2$ → belonging to the same set of " ..

(ii) systematic Property.

A (n,k) linear block code is said to be systematic, if k message bits appear either at the beginning or at the end of the code word.

Matrix description of linear block Codes:

Structure of code word



* In a systematic linear block code, the first k bits of the code word are the message bits i.e

$$C_i = m_i \longrightarrow ①$$

where $i = 1, 2 \cdots k$

* The last (n-k) bits in the code word are parity bits generated from the k message bits according to some determined rule

$$C_{k+1} = p_{11} m_1 + p_{21} m_2 + \cdots p_{k1} m_k$$

n-k=1
n=k+1
n=k+2

$$C_{k+2} = p_{12} m_1 + p_{22} m_2 + \cdots p_{k2} m_k$$

$$\vdots \qquad \longrightarrow ②$$

$$C_n = p_{1,n-k} m_1 + p_{2,n-k} m_2 + \cdots p_{k,n-k} m$$

where $C_{k+1}, C_{k+2} \cdots C_n \rightarrow$ Corresponding c w bits.

$m_1, m_2 \cdots m_k \rightarrow$ " Msg bits.

The coefficient p_{ij} in the above equation are 0's & 1's & the additional operations

$1 \oplus 1 = 0$

$1 \oplus 0 = 1$

$0 \oplus 1 = 1$

$0 \oplus 0 = 0$

- Combining equation ① & ② and write them in matrix form.

$$[C_1, C_2 \cdots\cdots C_n] = [m_1, m_2 \cdots\cdots m_k] \begin{bmatrix} 1000\cdots0 & p_{11} & p_{12} \cdots & p_{1,n} \\ 0100\cdots0 & p_{21} & p_{22} \cdots & p_{2,n-k} \\ 0010\cdots0 & & & \\ \vdots & & & \\ 0000\cdots1 & p_{k1} & p_{k2} \cdots & p_{k,n-k} \end{bmatrix}$$

It can be written as,

$$\boxed{C = MG}$$

$G \rightarrow$ Generator Matrix of order $(k \times n)$

- The Generator Matrix is of the form

$$G = [I_k : P_k]_{k \times n}$$

$I_k \rightarrow$ Identity Matrix

If $k = 3$, then $I_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Parity check Matrix (H)

→ For every block code, there is a $H (q \times n)$.

It is defined as,

$$H = \left[P^T : I_q \right]_{q \times n} \longrightarrow ①$$

P^T → transpose of P sub-matrix.

The P sub matrix is defined as,

$$P = \begin{bmatrix} P_{11} & P_{12} & P_{13} & \cdots & P_{1q} \\ P_{21} & P_{22} & P_{23} & \cdots & P_{2q} \\ \vdots & \vdots & \vdots & & \\ P_{K1} & P_{K2} & P_{K3} & \cdots & P_{Kq} \end{bmatrix}_{k \times q}$$

The transpose of this sub matrix is

$$P^T = \begin{bmatrix} P_{11} & P_{21} & \cdots & P_{K1} \\ P_{12} & P_{22} & \cdots & P_{K2} \\ \vdots & \vdots & & \vdots \\ P_{1q} & P_{2q} & \cdots & P_{Kq} \end{bmatrix}_{\substack{q \times k \\ \cancel{k \times q}}}$$

$$H_{q \times n} = \begin{bmatrix} P_{11} & P_{21} & \cdots & P_{K1} & : & 1 & 0 & \cdots & 0 \\ P_{12} & P_{22} & \cdots & P_{K2} & : & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \vdots & & & & \ddots & \end{bmatrix}$$

i) Syndrome Decoding:

→ It is one of the methods used to correcting errors in linear block coding.

→ Let the transmitted code vector be 'X' and Corresponding received code vector be 'Y'. Then, it can be expressed as,

$X = Y$ if there are no transmission errors.

$X \neq Y$ if there are errors created during transmission.

w.r.t with every (n, k) linear block codes, there exists a Parity check matrix (H) of size $q \times n$ is given as.

$$H = \left[P^T : I_q \right]_{q \times n}$$

→ The transpose of the above matrix can be obtained by interchanging the rows & the columns, i.e

$$H^T = \begin{bmatrix} P \\ \cdots \\ I_q \end{bmatrix}_{q \times n}$$

Important Property used in syndrome decoding

→ The transpose of parity check matrix (H^T) has very important Property as follows,

Definition of syndrome (s):

When some errors are present in received vector Y, then it will not be from valid code vectors and it will not satisfy the property of syndrome ~~[illegible]~~ as in above equation.

$XH^T = (0\,0\,0 \ldots 0)$. This shows that whenever YH^T is non-zero, some errors are present in Y. The non zero output of the product YH^T is called syndrome and it is used to detect the errors in Y.

Syndrome represented by 's' and can be written as,

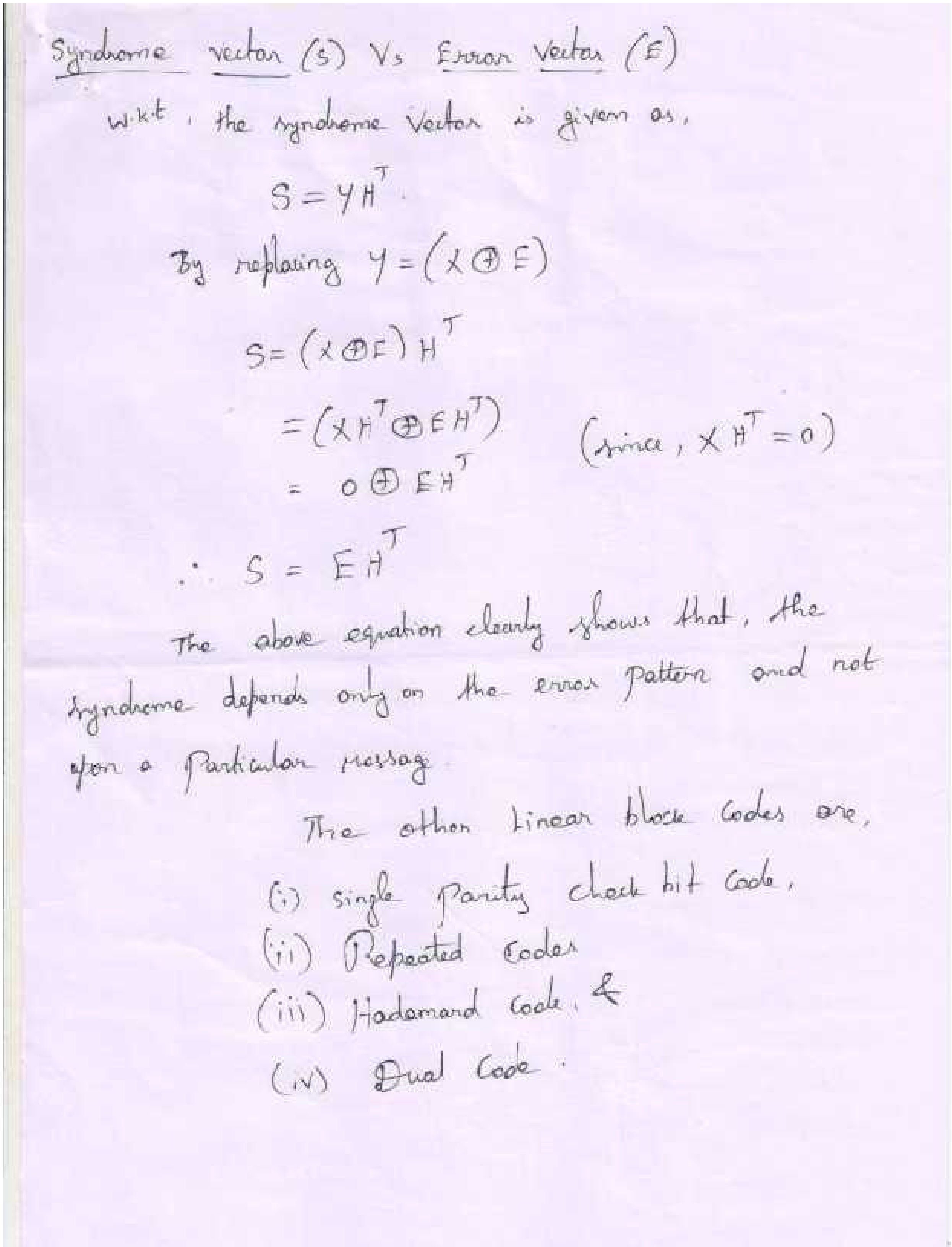
$$\boxed{S = YH^T}$$

Detecting error with the help of syndrome and error vector (E)

→ The non-zero elements of 's' represent error in the output. When all elements of 's' are zero, the two cases are possible.

(i) No error in the o/p and $Y = X$.

(ii) Y is some other valid code vector other than X. This means the transmission errors are undetectable.



Syndrome vector (S) Vs Error Vector (E)

w.k.t, the syndrome Vector is given as,

$$S = YH^T.$$

By replacing $Y = (X \oplus E)$

$$S = (X \oplus E) H^T$$

$$= (XH^T \oplus EH^T) \quad (\text{since}, XH^T = 0)$$

$$= 0 \oplus EH^T$$

$$\therefore S = EH^T$$

The above equation clearly shows that, the syndrome depends only on the error pattern and not upon a particular messag.

The other Linear block codes are,

(i) single parity check bit code,

(ii) Repeated codes

(iii) Hadamard code, &

(iv) Dual code.

Eg: The Parity Matrix P for a systematic (6,3) linear block code is given as, $[P] = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix}$. Find all the possible code vectors.

Gn:

(i) $n = 6$, $k = 3$ for (6, 3) linear block code.

since $k = 3$, the possible Message vectors are $2^k = 2^3 = 8$, which is given as,

S.NO	Bits of Message Vector		
	m_1	m_2	m_3
1	0	0	0
2	0	0	1
3	0	1	0
4	0	1	1
5	1	0	0
6	1	0	1
7	1	1	0
8	1	1	1

(ii) The parity matrix as,

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

To find all the possible code vectors.

The equation for code vectors are:

$$C = MG$$

$1 \oplus 0 \oplus 0$

$1 \oplus$

$1 \oplus 0$

w.k.t

$$G = [I_k : P]$$

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

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

$$\therefore \begin{bmatrix} c_1 & c_2 & c_3 \end{bmatrix} = \begin{bmatrix} m_1 & m_2 & m_3 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix}$$

which means,

$$C_1 = m_1 \oplus m_3$$

$$C_2 = m_1 \oplus M_2 \oplus M_3$$

$$C_3 = 0.$$

Message Vector			Code vector .		
m_1	m_2	m_3	C_1	C_2	C_3
0	0	0	0	0	0
0	0	1	1	1	0
0	1	0	0	1	0
0	1	1	1	0	0
1	0	0	1	1	0
1	0	1	0	0	0
1	1	0	1	0	0
1	1	1	0	1	0

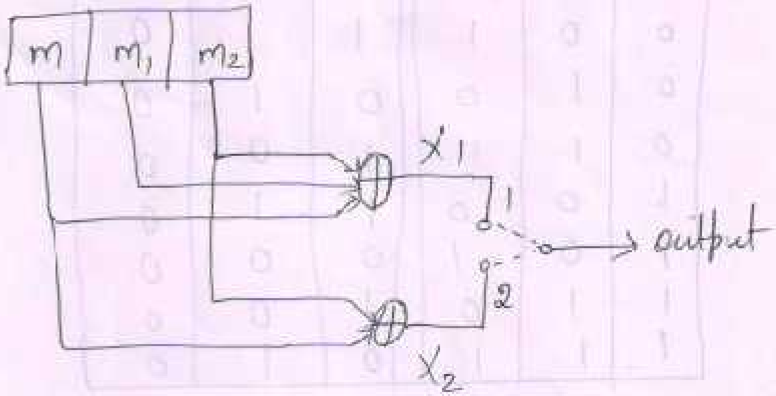
② The parity matrix P for a systematic (6,3) linear block Code is given as, $[P] = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$. Find all the possible Code vectors.

Convolutional Codes:

A Convolutional Coding is done by Combining the fixed number of input bits. The input bits are stored in the fixed length shift register and they are Combined with the help of Modulo-2 adders. This operation is equivalent to the binary Convolution and hence it is called Convolutional Coding

Encoder:

Consider the following encoder diagram



$$X_1 = m \oplus m_1 \oplus m_2$$

$$X_2 = m \oplus m_2$$

→ The output switch first samples X_1 and then X_2. The shift register then shifts contents of m_1 to m_2 and contents of m to m_1.

→ Next i/p bit is then taken & stored in m.

→ Again X_1 and X_2 are generated according to this new combination of m, m_1 & m_2.

→ The o/p switch then samples X_1 & then X_2.

Thus the o/p bit stream for successive input bits will be, For every input message bit, 2 output bits X_1 and X_2 are transmitted.

∴ Message bit $k=1$ which results in the 2 output bits i.e $n=2$.

Code Rate: The code rate of this encoder is

$$r = \left[\frac{k}{n}\right] = \left[\frac{1}{2}\right].$$

From the above figure.

First shift → message bit is entered in position 'm'
" " " 'm₁'
second shift → " " " 'm₂'.
Third shift → "
fourth shift → the message bit is discarded.

Constraint Length: (K)

It is defined as, "the number of shifts over which a single message bit can influence the encoder output".

From the above ~~figure~~ encoder diagram encoder, a single message bit influences encoder for three successive shifts.

At the fourth shift, the message bit is lost and it has no effect on the output.

The constraint length (K) can also be expressed as,

$$K = (n \times m)$$

where $n \rightarrow$ number of o/p bits for every i/p bit
$m \rightarrow$ " " storage elements in the shift regist.

$\therefore n = 2, \; m = 3$

$\therefore K = 2 \times 3 = 6$ bits.

Dimension of the code:

The dimension of the code is given by n and k.

where k = number of message bits taken at a time.
n = encoded output bits for 1 message bit.

Hence, dimension of code is $(n, k) = (2, 1)$.

Convolutional encoder has two approaches.

Convolutional Encoder has two approaches

(i) Time Domain Approach to analysis of Convolutional Encoder:

- Let us consider the impulse response of the adder with sequences $\{g_0^{(1)}, g_1^{(1)}, g_2^{(1)} \ldots g_m^{(1)}\}$ which generates X_1 as shown in above figure.

- ||ly, the impulse response of the adder with sequences $\{g_0^{(2)}, g_1^{(2)}, g_2^{(2)} \ldots g_m^{(2)}\}$ which generates X_2 as shown in above figure.

→ These impulse responses are also called as generator sequences of the code.

Let the incoming Message sequence be $\{m_0, m_1, m_2 \ldots\}$.

→ The encoder generates the 2 output sequences X_1 and X_2. These are obtained by Convolving the message sequence with generator sequence

The sequence X_1 and X_2 is given as,

$$X_1 = X_i^{(1)} = \left[\sum_{l=0}^{m} g_l^{(1)} m_{i-l}\right] \text{ for}$$

The sequence x_2 is given as . .

$$X_2 = x_i^{(2)} = \left[\sum_{l=0}^{m} g_l^{(2)} m_{i-l} \right] \quad i = 0,1,2 \ldots\ldots l > i, \quad i-l=0 \longrightarrow ②$$

∴ output sequence is given by

$$\{x_i\} = \{ x_0^{(1)}\ x_0^{(2)}\ x_1^{(1)}\ x_1^{(2)}\ x_2^{(1)}\ x_2^{(2)}\ x_3^{(1)}\ x_3^{(2)} \ldots\ldots \}$$

Here

$$x_i^{(1)} = \{ x_0^{(1)}, x_1^{(1)}, x_2^{(1)} \ldots\ldots \}$$

$$x_i^{(2)} = \{ x_0^{(2)}, x_1^{(2)}, x_2^{(2)} \ldots\ldots \}$$

(ii) Transform Domain Approach to analysis of Convolutional Encoder

The convolution of generating and message sequences takes place in the time domain approach. It can be simplified by adding transformations to the sequences.

Let the impulse response be represented by,

$$g^{(1)}(p) = g_0^{(1)} p^0 + g_1^{(1)} p^1 + g_2^{(1)} p^2 + \ldots + g_m^{(1)} p^m$$

$$g^{(2)}(p) = g_0^{(2)} p^0 + g_1^{(2)} p^1 + g_2^{(2)} p^2 + \ldots + g_m^{(2)} p^m$$

→ with respect to the above generator polynomial the message polynomial is given as,

$$m(p) = m_0 + m_1 p + m_2 p^2 + \dots m_{L-1} p^{L-1}$$

where, L = length of message sequence

The output of adders is given as,

$$\boxed{\begin{aligned} x^{(1)}(p) &= g^{(1)}(p)\, m(p) \\ x^{(2)}(p) &= g^{(2)}(p)\, m(p) \end{aligned}}$$

Problem ①

For the convolutional encoder, determine the following

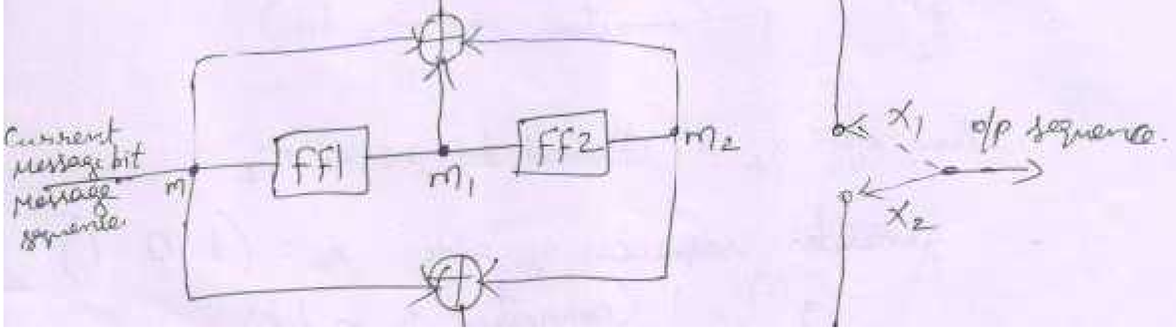
(i) dimension of the code

(ii) code rate

(iii) Constraint length

(iv) Generating sequences (impulse responses)

(v) output sequences for the message sequence m = 10011



Given:

$n = 2$ (no. of outputs)
$K = 1$ (no. of i/p bit).

Soln:

(i) Dimension of the code

$$(n, k) = (2, 1)$$

(ii) code rate

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

(iii) Constraint length

$K = 3$ bits (or) $n \times M = 2 \times 3 = 6$ bits.

(iv) Generator sequences:

The output x_1 is obtained by all m, m_1, m_2.

$\therefore$ generator sequences of adder $x_1 = (1\ 1\ 1)$

$g_0^{(1)} = 1$ (Connection of m bit)

$g_1^{(1)} = 1$ (Connection of m_1 bit)

$g_2^{(1)} = 1$ (connection of m_2 bit)

The o/p x_2 is obtained by m & m_2

$\therefore$ generator sequences of adder $x_2 = (1\ 0\ 1)$

$g^{(2)} = 1$ (Connection of m bit)

$g_2^{(2)} = 1$ (Connection of m_2 bit).

(v) Output sequences for Message sequence $m = \overset{m_0}{1}\overset{m_1}{0}\overset{m_2}{0}\overset{m_3}{1}\overset{m_4}{1}$

<u>o/p due to adder 1</u>

$$X_1 = X_i^{(1)} = \left[\sum_{l=0}^{m} g_l^{(1)} m_{i-l}\right] \quad \text{for } i = 0, 1, 2 \ldots, l > i, \; i-l=0$$

when $i = 0$, $X_0^{(1)} = \sum_{l=0}^{m} g_l^{(1)} m_{0-l}$

$$= g_0^{(1)} m_0 = 1 \times 1 = 1.$$

$i = 1$, $X_1^{(1)} = \sum_{l=0}^{m} g_l^{(1)} m_{1-l} = g_0^{(1)} m_1 \oplus g_1^{(1)} m_0$

$$= (1)(0) \oplus (1)(1)$$

$$= 0 \oplus 1 = 1$$

$i = 2$, $X_2^{(1)} = \sum_{l=0}^{m} g_l^{(1)} m_{2-l} = g_0^{(1)} m_2 \oplus g_1^{(1)} m_1 \oplus g_2^{(1)} m_0$

$$= (1)(1) \oplus (1)(0) \oplus (1)(1)$$

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

$$= 0$$

$i = 3$, $X_3^{(1)} = \sum_{l=0}^{m} g_l^{(1)} m_{3-l} = g_0^{(1)} m_3 \oplus g_1^{(1)} m_2 \oplus g_2^{(1)} m_1 \oplus$

i = 4, $x_4^{(1)} = 1$

i = 5, $x_5^{(1)} = 0$

i = 6, $x_6^{(1)} = 1$

$\boxed{1111001}$

The o/p of adder 1 is 1100101

<u>o/p due to adder 2</u>

$$x_2 = x_i^{(2)} = \sum_{l=0}^{m} g_l^{(2)} m_{i-l} \quad \text{for } i = 0, 1, 2 \ldots l > i, \; i-l=0$$

when i = 0, $x_0^{(2)} = \sum_{l=0}^{m} g_l^{(2)} m_{0-l}$

$$= g_0^{(2)} m_0 = 1$$

i = 1, $x_1^{(2)} = 0$

i = 2, $x_2^{(2)} = 0$

i = 3, $x_3^{(2)} = 1$

i = 4, $x_4^{(2)} = 0$

i = 5, $x_5^{(2)} = 1$

i = 6, $x_6^{(2)} = 1$

$\boxed{1011111}$

The o/p of adder 2 is 1001011

∴ The overall output is obtained by combining output of adder 1 with adder 2 i.e 2 first bits, 2 second bits, & so on. (11, 10, 11, 11, 01, 01, 11)

② Solve the above eg. problem of Convolutional encoder using Transform domain Approach.

Generator sequences:

The o/p x_1 is obtained by m, m_1, m_2.

$\therefore$ generator sequences of adder $x_1 = (1\ 1\ 1) = g_i^{(1)}$

The polynomial can be obtained as,

$$g^{(1)}(p) = g_0^{(1)} p^0 + g_1^{(1)} p^1 + g_2^{(1)} p^2$$

$$= 1 + p + p^2.$$

The output x_2 is obtained by m & m_2.

$\therefore$ generator sequences of adder $x_2 = (1\ 0\ 1) = g_i^{(2)}$

The Polynomial can be obtained as,

$$g^{(2)}(p) = g_0^{(2)} p^0 + g_1^{(2)} p^1 + g_2^{(2)} p^2$$

$$= 1 + p^2$$

Output sequences for Message sequence $m = \overset{m^0}{1}\overset{m^1}{0}\overset{m^2}{0}\overset{m^3}{1}\overset{m^4}{1}$

with message polynomial $= 1 + p^3 + p^4$.

Output due to adder 1:

$$= (1+p+p^2)(1+p^3+p^4)$$

$$= \overset{\checkmark}{1}+p^3+\overset{\times}{p^4}+\overset{\checkmark}{p}+\overset{\times}{p^4}+\overset{\times}{p^5}+p^2+\overset{\times}{p^5}+p^6$$

$$= 1+p+p^2+p^3+p^8+p^{10}+p^6$$

$$= 1+p+p^2+p^3+p^6+\boxed{p^8}+\boxed{p^{10}} \quad \boxed{\times}$$

∵ if repeated neglect it p^4 & p^4 two times so neglect

$$= (1\ 1\ 1\ 1\ 0\ 0\ 1)$$

<u>output due to adder 2</u>:

$$x^{(2)}(p) = g^{(2)}(p)\, m(p)$$

$$= (1+p^2)(1+p^3+p^4)$$

$$= 1+p^3+p^4+p^2+p^5+p^6$$

$$= 1+p^2+p^3+p^4+p^5+p^6$$

$$= (1\ 0\ 1\ 1\ 1\ 1\ 1)$$

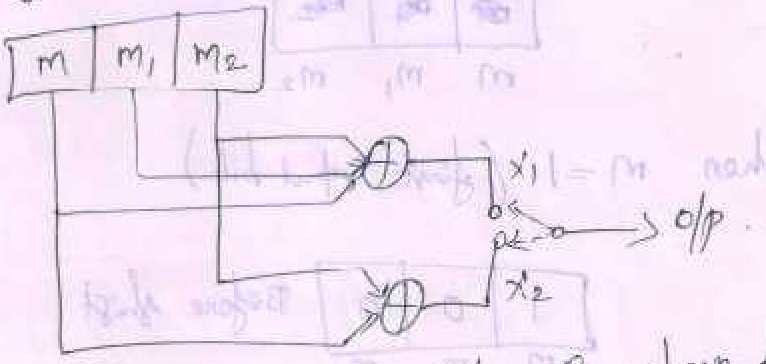
The overall output is obtained by combining output of adder 1 with adder 2.

$$(1111001)\ (1011111) = (11,10,11,11,01,01,11)$$

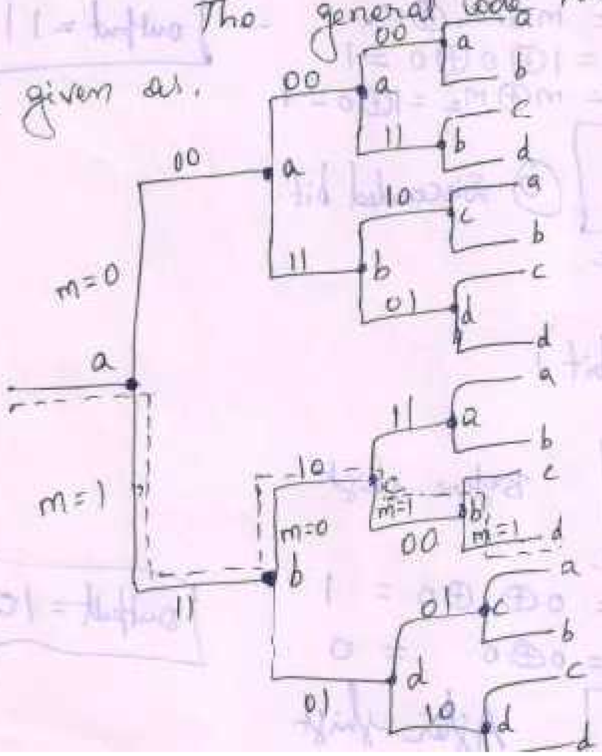
Decoding Convolutional codes:

To decode a convolutional code, we consider the code tree.

Here we explain it with the convolutional encoder diagram.



The general code tree for the above figure is given as.

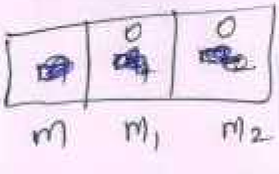


output = 11 10 00 01

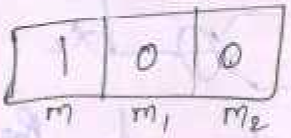
The dotted lines represent the trace path of message sequence 1011

Let us consider the development of code tree for the input sequence $\boxed{m = 1011}$

Initially assume $m_1\ m_2 = 00$



When $m = 1$ (first input bit)

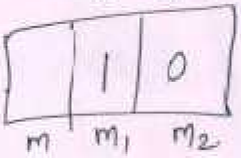


Before shift

$m = 1$ (first bit) $\Big\Downarrow$ $x_1 = m \oplus m_1 \oplus m_2 = 1 \oplus 0 \oplus 0 = 1$

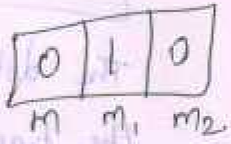
$x_2 = m \oplus m_2 = 1 \oplus 0 = 1$

$\boxed{\text{output} = 11}$



ⓞ Discarded bit.

When $m = 0$ (second bit)



Before shift

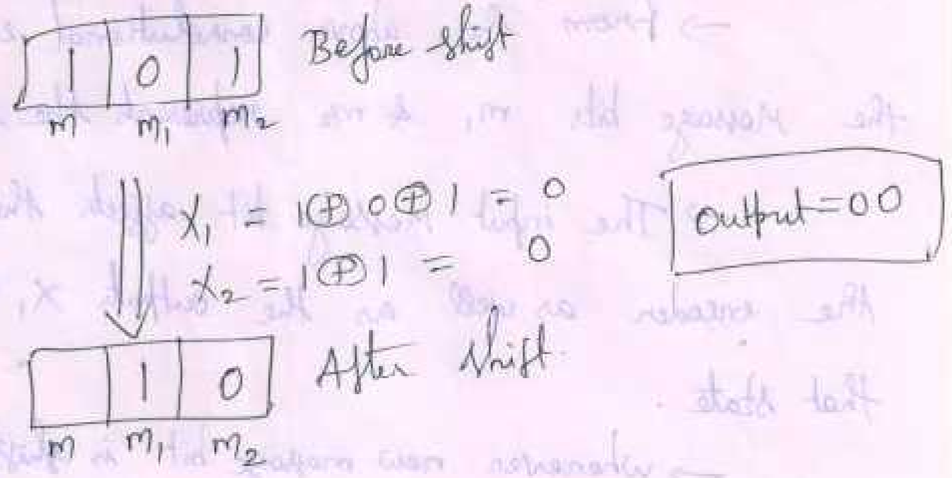
$m = 0$ (second bit) $\Big\Downarrow$ $x_1 = 0 \oplus 1 \oplus 0 = 1$

$x_2 = 0 \oplus 0 = 0$

$\boxed{\text{output} = 10}$

After shift

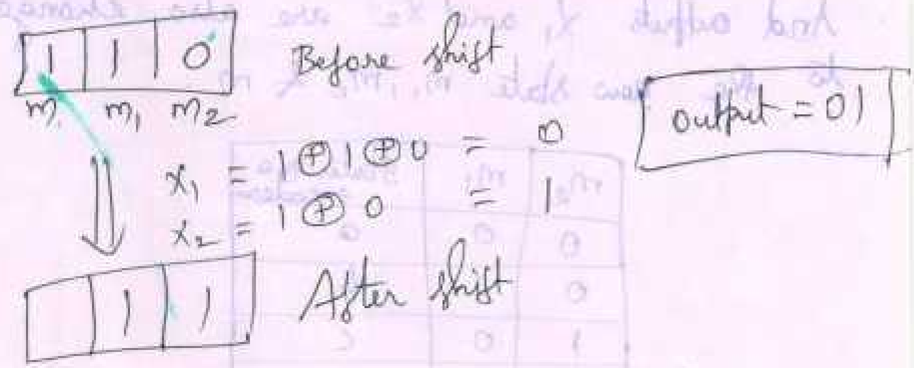
when m=1 (third bit)



$x_1 = 1 \oplus 0 \oplus 1 = 0$

$x_2 = 1 \oplus 1 = 0$

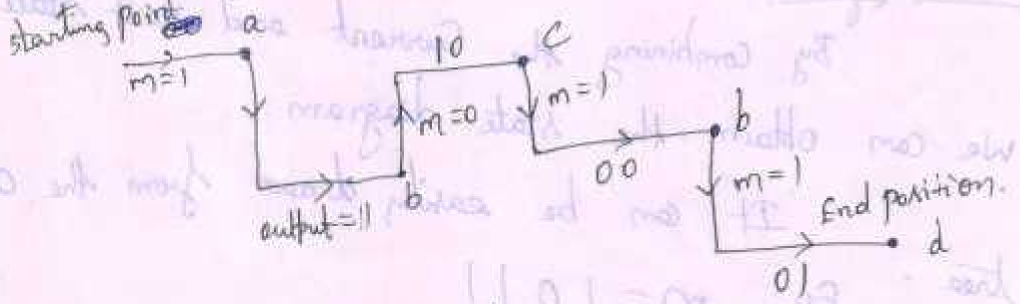
when m=1 (fourth bit)



$x_1 = 1 \oplus 1 \oplus 0 = 0$

$x_2 = 1 \oplus 0 = 1$

Finally the code tree was constructed as,



State Diagram and its States:

→ from the above convolutional encoder diagram, the Message bits m_1 & m_2 represent the state.

→ The input Message bit affects the state of the encoder as well as the outputs X_1 & X_2 during that state.

→ whenever new message bit is shifted to 'm', the contents of m_1 and m_2 define new state. And outputs X_1 and X_2 are also changed according to the new state m_1, m_2 & m.

m_2	m_1	States of a encoder
0	0	a
0	1	b
1	0	c
1	1	d

State diagram:

By combining the Current and next states. We can obtain the state diagram. It can be easily drawn from the code tree.

Eg: $m = 1011$

a = 1 $x_1 x_2 = 11$

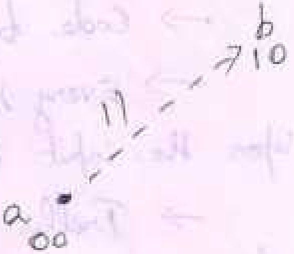
m=1 $x_1 x_2 = 00$

m=1 $x_1 x_2 = 01$

m=1 (first bit)

m_1 m_2 $\Longrightarrow$ m_1 m_2

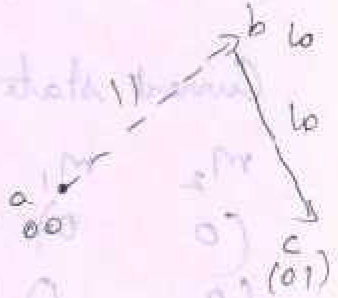
0 0 o/p = 11 1 0



m = 0 (second bit)

m_1 m_2 $\Longrightarrow$ m_1 m_2

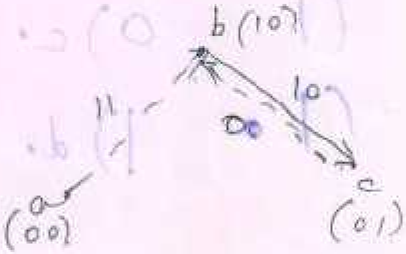
1 0 o/p = 10 0 1



m = 1 (third bit)

m_1 m_2 $\Longrightarrow$ m_1 m_2

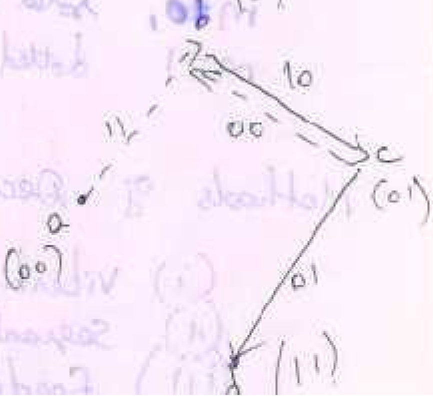
0 1 o/p = 00 1 0



m = 1 (fourth bit)

m_1 m_2 $\Longrightarrow$ m_1 m_2

1 0 o/p = 01 1



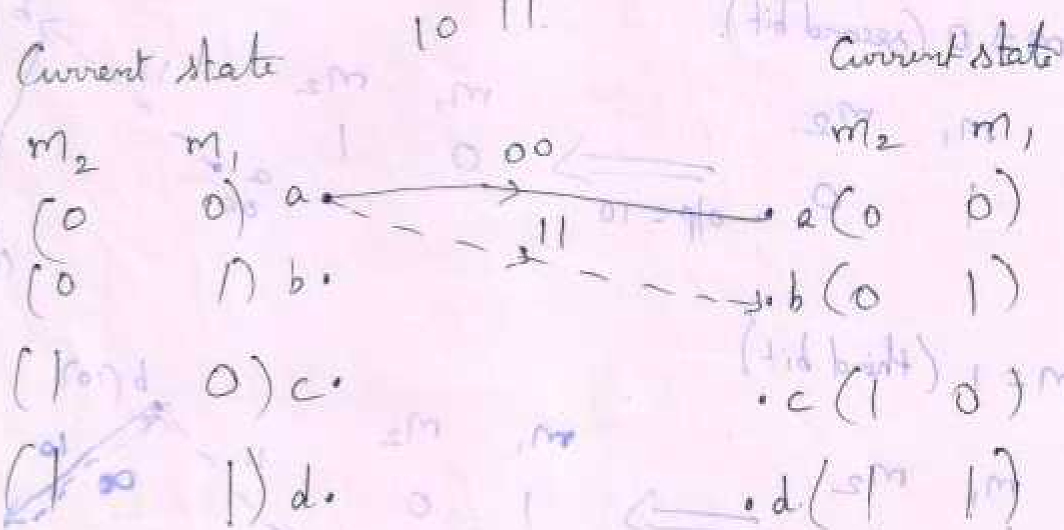
Code Trellis:

→ It is the more compact representation of code tree.

→ Code tree has 4 states on nodes.

→ Every state goes to some other state depending upon the input code.

→ Trellis represents the single, an unique diagram for such transitions. It is shown as,



m=0, solid transition line
m=1 dotted " "

m	m1	m2
	0	0
	1	0

Methods of Decoding the convolutional code.

(i) Viterbi Algorithm
(ii) Sequential decoding
(iii) Feedback Coding