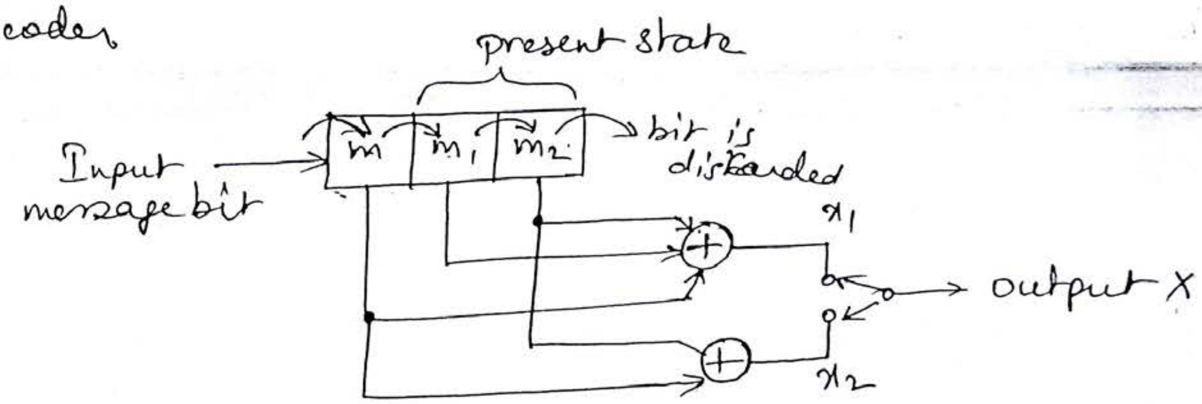
Unit - VII

Convolutional Codes

Convolutional coding is done by combining fixed number of input bits. The input bits are stored in the fixed length shift register and they are combined with the help of mod-2 adders (XOR). This operation is equivalent to binary convolution, hence it is called convolutional coding. Fig shows a basic convolutional encoder



operation

whenever the message bit is shifted to position 'm' the new values of x_1 and x_2 are generated depending upon m, m_1 and m_2.

m_1 and m_2 store the previous two message bits. The current bit is present in m.

$$\therefore \quad x_1 = m \oplus m_1 \oplus m_2$$

$$x_2 = m \oplus m_2$$

The o/p switch first samples x_1 and then x_2. The shift register then shifts the contents of m_1 to m_2 and contents of m to m_1. The next input bit is taken and stored in m. Again x_1 and x_2 are generated according to this new combination of m, m_1 and m_2.

Hence o/p switch samples x_1 and then x_2. Therefore the o/p bit stream for successive input bits will be

$$X = x_1^0 x_2^0 x_1^1 x_2^1 x_1^2 x_2^2 \ldots \text{ and so on.}$$

For every input message bit, two encoded output bits x_1 and x_2 are transmitted, i.e for a single message bit, the encoded code word is two bits.

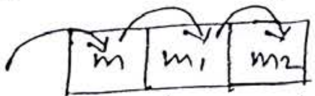
Terms used in convolutional encoder

1) <u>k</u> :- No. of message bits taken at a time (small letter k)
   For the encoder in fig $k = 1$.

2) <u>n</u> :- No. of encoded output bits for message bits, here $n = 2$ (x_1 and x_2)

3) <u>Code rate r</u> :- $r = \frac{k}{n} = \frac{1}{2}$

4) <u>Constraint length</u> :- K (capital K)
   The constraint length of a convolutional code is defined as the number of shifts over which a single message bit stays in the shift register.
   In the previous encoder $K = 3$, because the message bit stays for three successive bits.



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

5) Dimensions of the code :-

$$D = (n, k)$$

6) <u>Outputs</u> :-
   x_1 and x_2

$$x_1 = m \oplus m_1 \oplus m_2 \quad , \; x_2 = m \oplus m_2$$

Hence o/p switch samples x_1 and then x_2. Therefore the o/p bit stream for successive input bits will be

$$X = x_1^0 x_2^0 x_1^1 x_2^1 x_1^2 x_2^2 \ldots \text{ and so on.}$$

For every input message bit, two encoded output bits x_1 and x_2 are transmitted. i.e for a single message bit, the encoded code word is two bits.

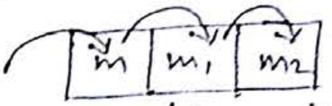
Terms used in convolutional encoder

1) k :- No. of message bits taken at a time (small letter k)
   For the encoder in fig k = 1.

2) n :- No. of encoded output bits for message bits, here n = 2 (x_1 and x_2)

3) Code rate r :- $r = \frac{k}{n} = \frac{1}{2}$

4) Constraint length :- K (capital K)
   The constraint length of a convolutional code is defined as the number of shifts over which a single message bit stays in the shift register.
   In the previous encoder K = 3, because the message bit stays for three successive bits.



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

5) Dimensions of the code :-

$$D = (n, k)$$

6) Outputs :-
   x_1 and x_2
   $x_1 = m \oplus m_1 \oplus m_2$, $x_2 = m \oplus m_2$

7) <u>Impulse inputs</u> :- The generator sequences are the impulse inputs. The o/p x_1 is obtained by $(m \oplus m_1 \oplus m_2)$ and x_2 is $(m \oplus m_2)$

Hence generator sequence $g_1 = (1\ 1\ 1)$ (all three I/ps are taken for x_1)

$g_2 = (1\ 0\ 1)$ (only m and m_2 are taken for x_2)

8) <u>Generator matrix</u> :-

$$G = [g_1\ g_2]$$

The outputs are obtained from generator matrices as follows.

$$x_1 = \text{I/p message} \times g_1 = Mg_1$$

$$x_2 = \text{I/p message} \times g_2 = Mg_2$$

$$X = [x_1\ x_2] = [Mg_1\ Mg_2]$$

9) <u>Polynomials</u>

generator polynomial $g_{1p} = 1 + p + p^2$

$g_{2p} = 1 + p^2$

<u>Message polynomial</u> .

If $M = 101$

$M_p = 1 + 0p + p^2 = 1 + p^2$

problem

$M = [1011] \quad g_1 = [1\,1\,1] \quad g_2 = [1\,0\,1]$

Find the output sequence.

sol:- $M = [1\,0\,1\,1]$

$M_p = 1 + 0p + p^2 + p^3 = (1 + p^2 + p^3)$

$g_1 = [1\,1\,1]$

$g_{1p} = (1 + p + p^2)$

$g_2 = [1\,0\,1]$

$g_{2p} = 1 + 0p + p^2 = (1 + p^2)$

o/p $x_1 = Mg_1 = M_p\, g_{1p}$

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

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

$= 1 + p + (p^2 \oplus p^2) + (p^3 \oplus p^3) + (p^4 \oplus p^4) + p^5$

$= 1 + p + p^5$

$= 1 + p + 0p^2 + 0p^3 + 0p^4 + p^5$

$= [1\,1\,0\,0\,0\,1]$

o/p $x_2 = M_p\, g_{2p}$

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

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

$= 1 + (p^2 \oplus p^2) + p^3 + p^4 + p^5$

$= 1 + 0p + 0p^2 + p^3 + p^4 + p^5$

$= [1\,0\,0\,1\,1\,1]$

The o/p of the encoder

$X = [x_1^0 x_2^0, x_1^1 x_2^1, x_1^2 x_2^2, x_1^3 x_2^3, x_1^4 x_2^4, x_1^5 x_2^5]$

$[11, 10, 00, 01, 01, 11]$

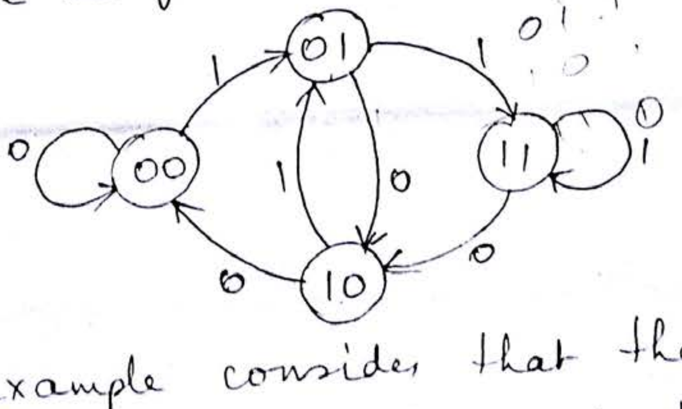
Transition Table or State Table

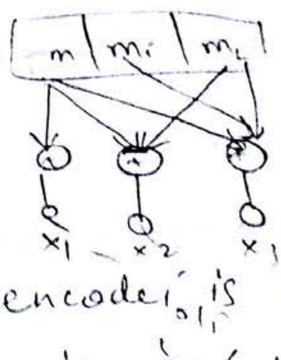
Present State		Input	Next State		Outputs $x_2 = m \oplus m_2$	$x_1 = m \oplus m_1 \oplus m_2$
m_2	m_1	m	m_1	m	x_2	x_1
0	0	0	0	0	0	0
0	0	1	0	1	1	1
0	1	0	1	0	0	1
0	1	1	1	1	1	0
1	0	0	0	0	1	1
1	0	1	0	1	0	0
1	1	0	1	0	1	0
1	1	1	1	1	0	1

State diagram

If we combine the current and next states, we obtain the state diagram.

$g_1 = \{100\}$ $g_2 = \{101\}$ $g_3 = \{111\}$





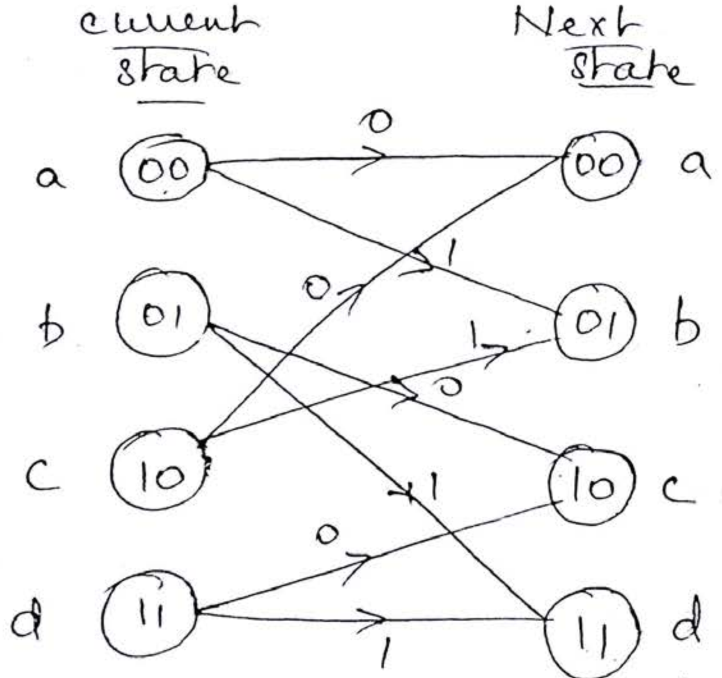
For example consider that the encoder is in state '00'. If input m = 0, the next state is '00', which is shown by self loop at '00'. If input m = 1, the next state is '01'.

states of the encoder

m_2	m_1	State
0	0	a
0	1	b
1	0	c
1	1	d

Trellis diagram

This is a more compact representation. Trellis represents the signal, as unique diagram for such transitions of the signal.



The nodes or circles on the left denote four possible current states and those on the right represent the next state.

For example if encoder is in current state 'a', and input m=0, the next state is 'a'. If input m=1, the next state is 'b'.

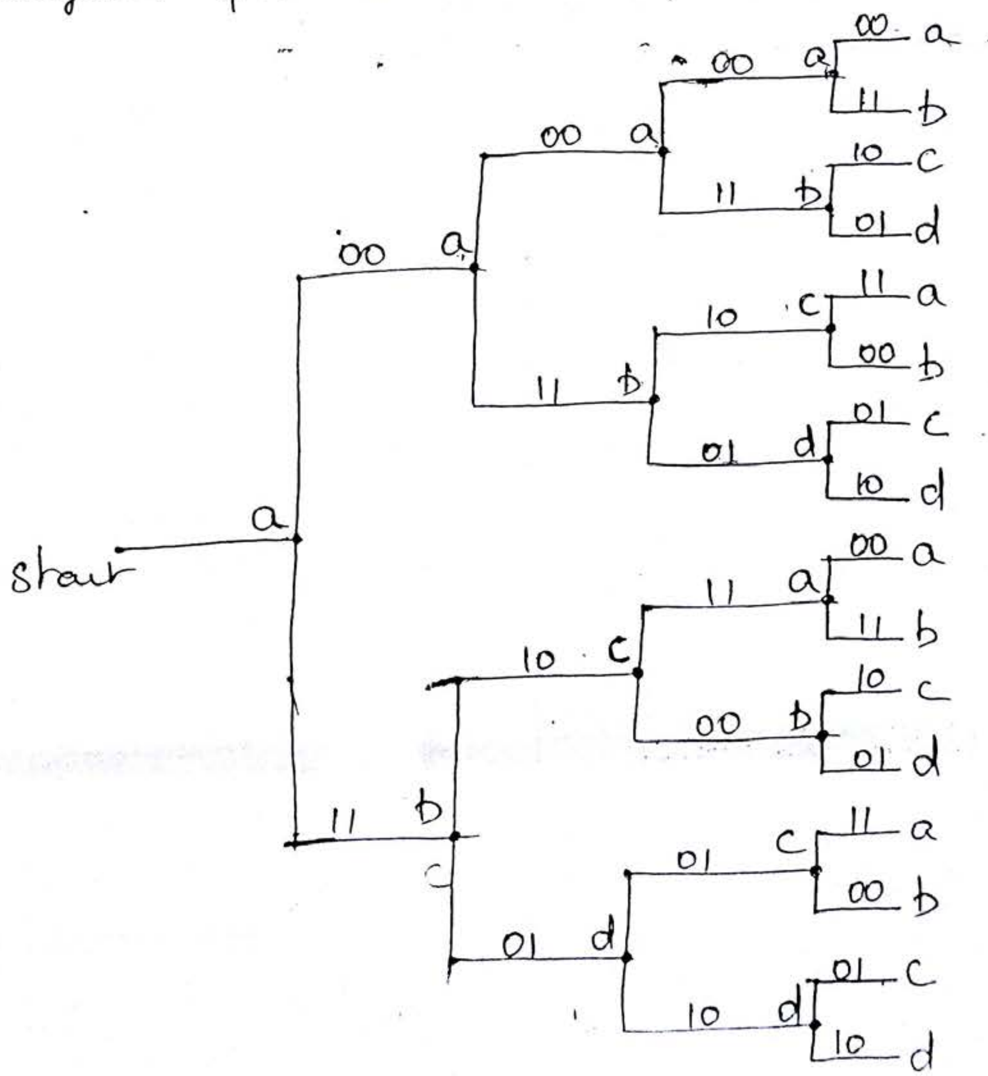
Tree diagram

The code tree diagram starts at node or state a.

a = 00
b = 01
c = 10
d = 11

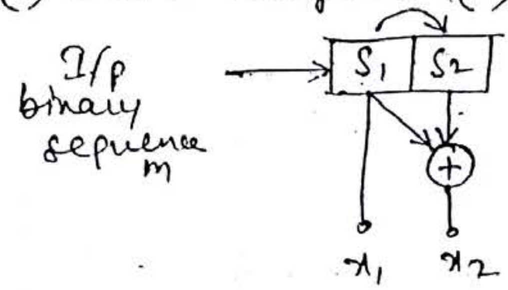
↑ upward arrow for m=0
start
↓ downward arrow for m=1

If the input m=0, we go upward from node 'a'. If m=1, we go downward from node 'a'. The code tree diagram for the convolutional encoder is shown below.



problem :- For the convolutional encoder shown determine the following

i) state diagram ii) Trellis diagram iii) Tree diagram

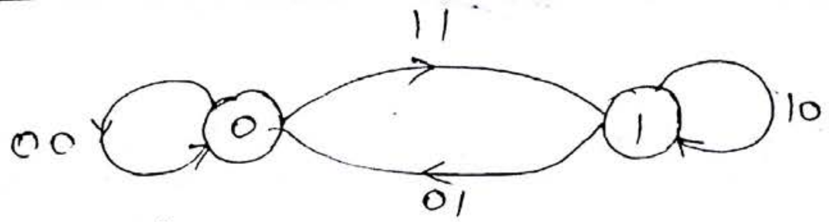


sol:- Here, input $= m$, present state $= S_2$

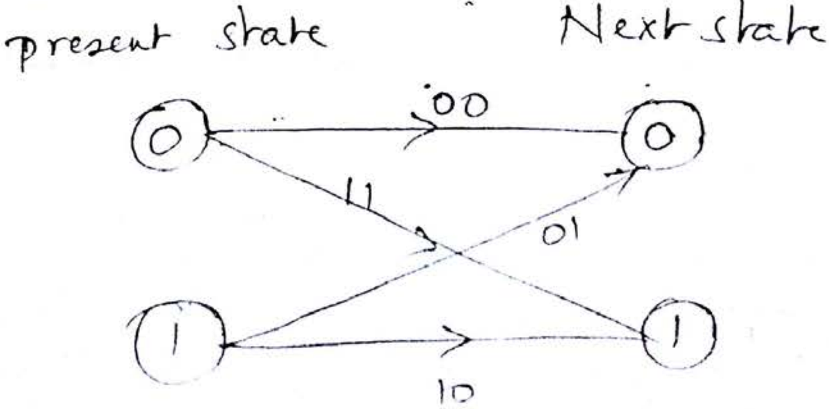
Next state $= S_1$ $(\because S_2 = S_1$ shifted$)$

Input (m) $(m = S_1)$	Present state (S_2)	Next state (S_1)	output x_1	x_2
0	0	0	0	0
1	0	1	1	1
0	1	0	0	1
1	1	1	1	0

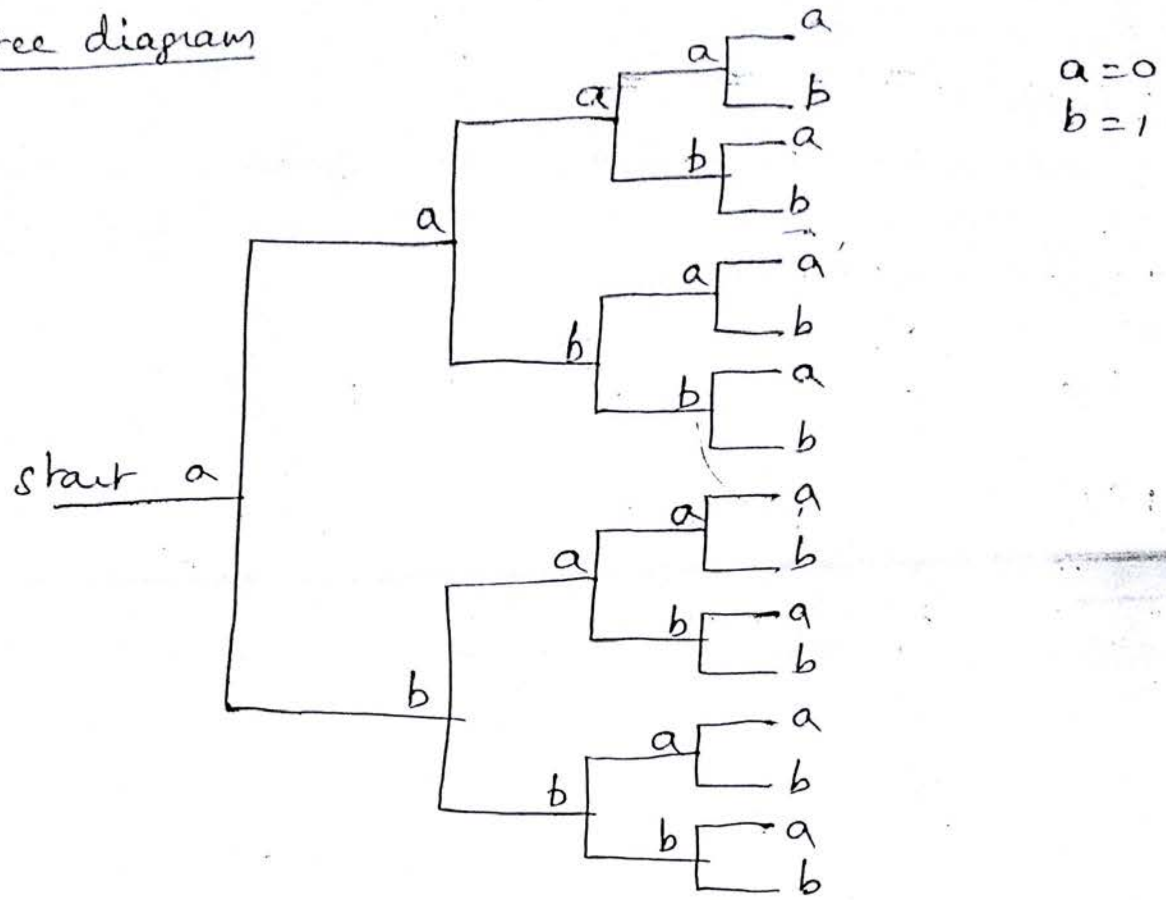
state diagram



Trellis diagram



- Tree diagram



Viterbi Algorithm for decoding of convolutional codes

Terms used

(i) k :- No. of message bits taken at a time (small k)

(ii) n :- No. of encoded output bits

(iii) code rate r :- $r = \frac{k}{n}$

(iv) constraint length K (capital K)

It is defined as the number of shifts over which a single message bit influences the encoder output.

(v) Impulse inputs :- the generator sequences are the impulse inputs. Ex:- o/p $x_1 = m \oplus m_1 \oplus m_2$

$x_2 = m \oplus m_2$

$g_1 = [1\ 1\ 1]$ (all 3 inputs are taken)

$g_2 = [1\ 0\ 1]$ (only m and m_2 are taken)

vi) Dimensions of the code = (n, k)

vii) Received signal = Y

viii) branch metric :- It is the discrepancy between the received signal Y and the decoded signal i.e Hamming distance.

Ex:- (a, b) & (c, d)

Hamming distance or branch metric $= |a-c| + |b-d|$

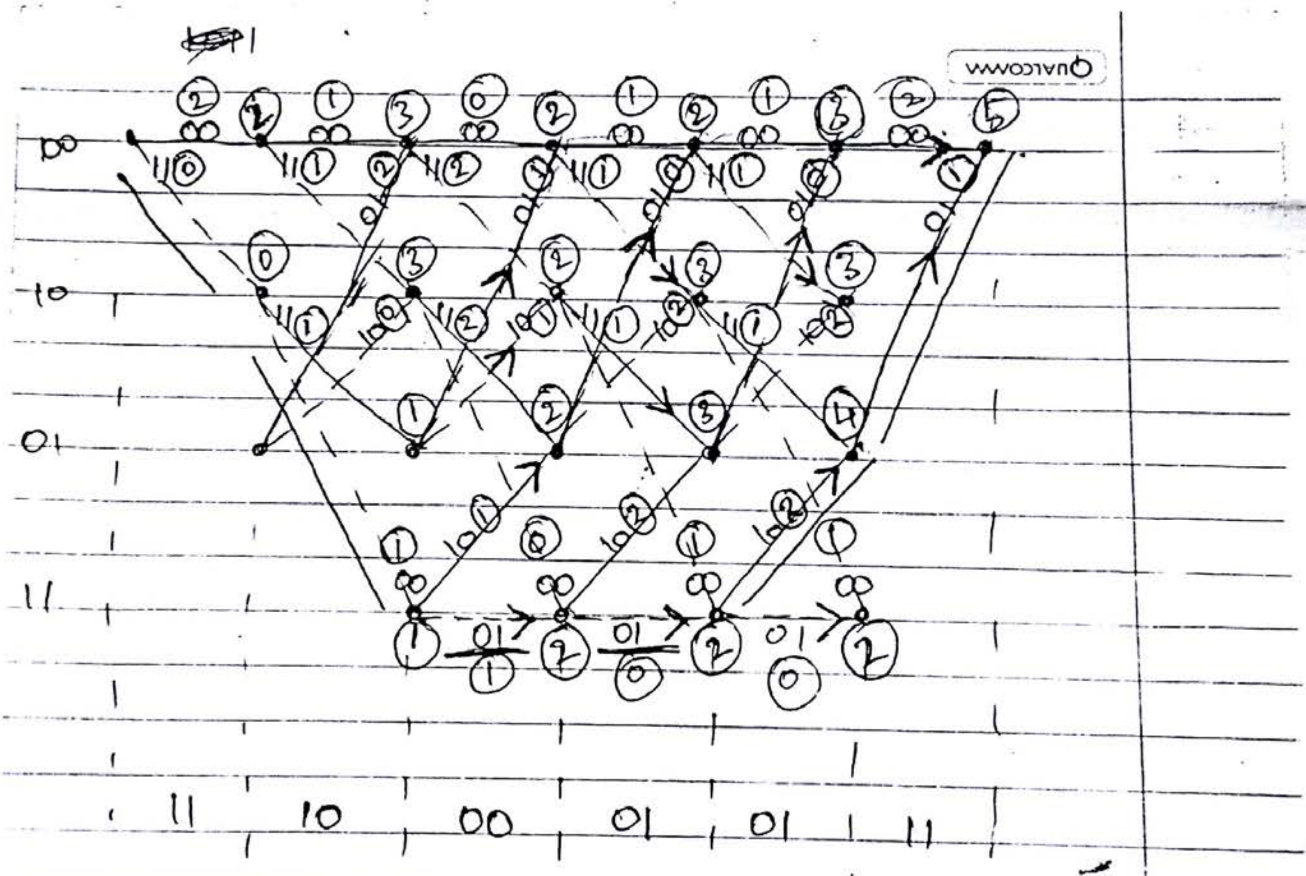
ix) Path metric :- It is the sum of the branch metrics

x) surviving path :- It is the path of the decoded signal with minimum metric

Algorithm for viterbi decoding

1) For n o/ps draw the complete trellis diagram by taking $(n+1)$ nodes
2) Calculate the individual branch metrics using the Hamming distance.
3) Calculate the path metric by adding the branch metrics taking into account only the surviving paths
4) Trace and decode the surviving path from back to front by taking a '0' for a solid line and a '1' for a dotted line.
5) The decoded message is the input message.

Problem :- Using Viterbi algorithm, decode the sequence 11 10 00 01 01 11



message = 1 1 1 1 0 0