UNIT - V

1] Viterbi algorithm for decoding of convolutional codes

Let the received signal is represented by Y.

- Convolutional ~~en~~ encoding operates continuously on i/p data. Hence, there are no code vectors and blocks.
- Let's assume the transmission error probability of symbols 1's and 0's is same.

Metric:

- It is the discrepancy (difference) b/w the received signal Y and the decoded signal at particular node.
- This metric can be added over few nodes for a particular path.

Surviving path:

- Path of decoded signal with minm metric.
- In viterbi decoding, a metric is assigned to each surviving path.
- Y is decoded ~~with the~~ as the surviving path with smallest metric.

let Y = 11 01 11

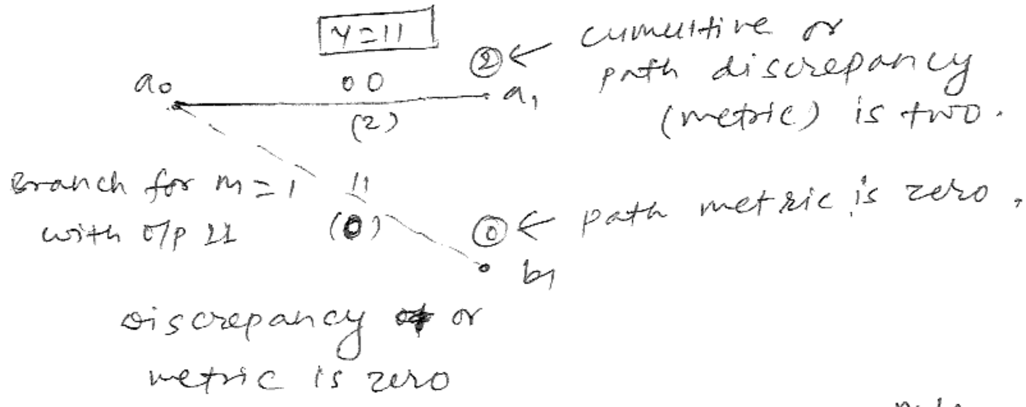


fig. viterbi decoder results for first ~~msg~~ msg bit.

- If the current state is 'a', then next state ~~to~~ will be 'a' or 'b'.
- Two branches are shown ~~for~~ from a_0.
- One branch is at next node a_1 representing decoded signal as 00 & the other branch is at b_1 representing decoded signal as 11.
- The branch from $a_0 b_1$ represents decoded o/p at 11 which is same as received signal at that node i.e. 11. Thus, there ~~is~~ is no discrepancy b/w ~~sent~~ received signal and decoded signal. Hence, metric = 0
- metric from a_0 to a_1 is ~~2~~ two.

Decoding of second message bit for y = 01.

$\boxed{y = 0 0 1}$

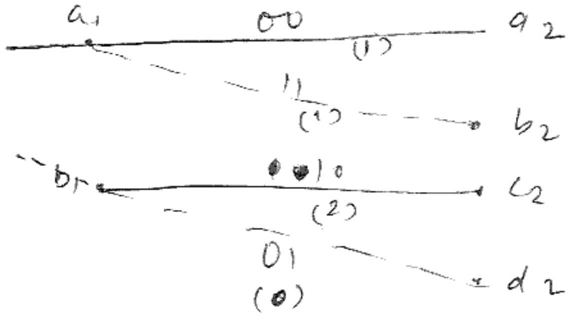


fig. Viterbi decoder results for second msg. bit.

From nodes a_1 & b_1, four possible next states $a_2, b_2, c_2,$ & d_2 are possible.

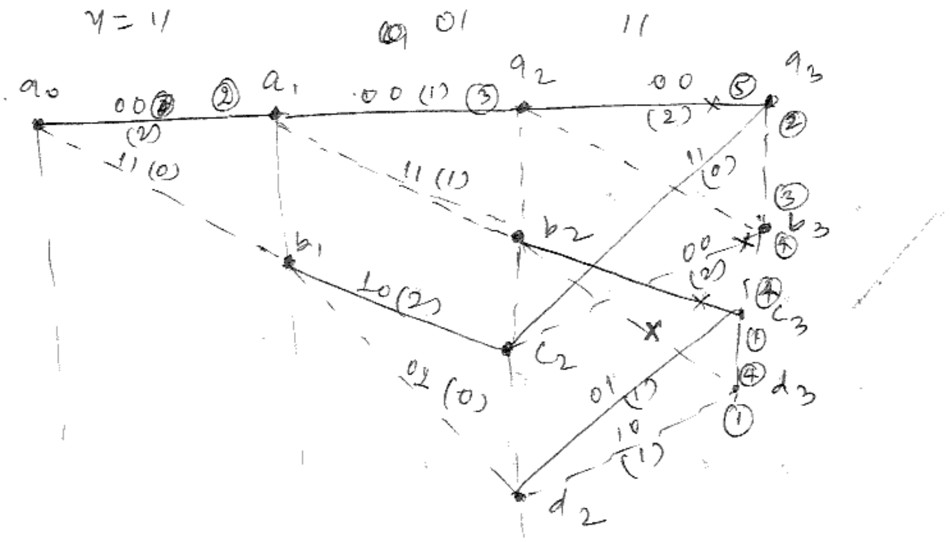


Fig. paths and their metrics for viterbi decoding.

→ Two paths are common to node 'a'.

→ one path is $a_0 a_1 a_2 a_3$ with metric 5.
other " " $a_0 b_1 c_2 a_3$ " " 2

→ A/c to viterbi decoding, only one path with lower metric should be retained at particular node

→ As shown in fig. the path marked 'x' (cross) are cancelled because they have higher metrics than other path coming to that particular node.

- These four paths with lower metrics are stored in the decoder and the decoding continues to next received bits.

2]. Sequential decoding for convolutional codes.

- Sequential decoding uses metric divergence effect.

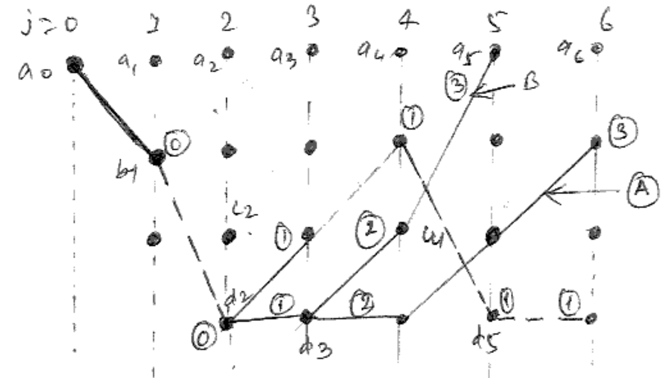


fig (a)

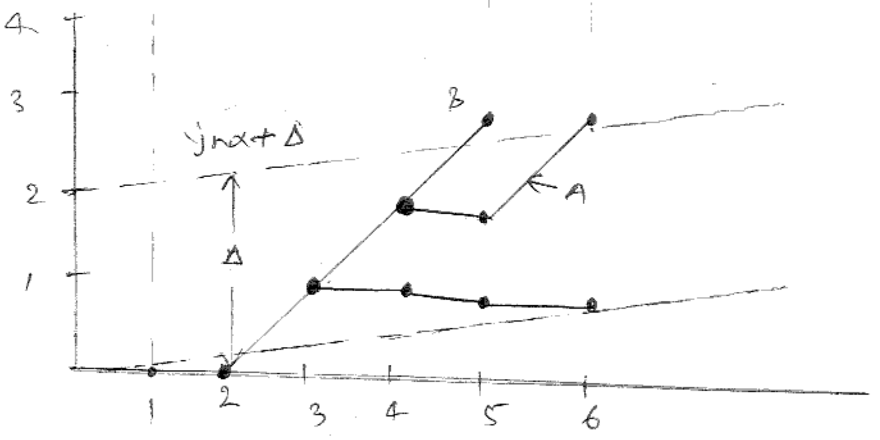


fig (b)

Sequential decoding

- fig (a) shows the code trellis for the convolutional encoder.

- i) The decoding starts at a_0, follows the single path by taking the branch with smallest metric.
  For eg. in fig (a), the path for first three nodes is $a_0 b_1 d_2$ since its metric is the lowest.

- ii) If there are two or more branches from the same node with same metric, then decoder selects anyone branch and continues decoding.

- iii) Beyond the above ~~co~~ (ii) condition, if the selected path is found to be unlikely with rapidly increasing merit, then decoder cancels that path and goes back to that node. It then selects another path emerging from that node.

As eg. two branches with same metric emerge from node d_2. One path is $d_2 d_3 c_4 a_5$ with metric 3 at a_5. Therefore, decoder drops this path and follows other path.

iv) The decision about dropping a path is based on the expected value of running metric at a given node

Running metric $= j n \alpha$.

where, j is the node at which metric is to be calculated.

$n \rightarrow$ no. of encoded o/p bits for one msg. bit

$\alpha \rightarrow$ transmission error probability / bit.

- v) The sequential decoder abandons a path whenever it's running metric exceeds $(jn\alpha + \Delta)$

$> jn\alpha$ at jth node.

v) The two dotted lines shows the range of threshold Δ above $jn\alpha$ at a particular node.

vi) Observe that, since the metric of ~~path~~ path 'B' exceeds the threshold at 5th node, it is abandoned & decoder starts from node 2. Similarly path 'A' is also ~~abandoned~~ abandoned.

(5) If N) If the running metric of every path goes out of threshold limits, then the value of threshold 'Δ' is increased & decoder tries back again.

- The computations involved in sequential decoding are less than viterbi decoding.
- The o/p error probability is more in case of sequential decoding.

3) probability of errors for soft and hard decision decoding.

Probability of error with soft Decision Decoding.

Let us consider that the all zero sequence is transmitted. Also, the coded binary digits for the j^{th} branch of the convolutional code be represented as

C_{jm} , $m = 1, 2, \ldots n$.

The i/p to the viterbi decoder be the sequence, r_{jm}, $m = 1, 2, \ldots n$ & $j = 1, 2, \ldots$

Given,

$$r_{jm} = \sqrt{E_c}\,(2C_{jm} - 1) + n_{jm} \quad -(1)$$

where:

$r_{jm} \rightarrow$ transmitted bits.

$j, m \rightarrow$ j^{th} ~~branc~~ branch, m^{th} bit in the same branch.

$E_c \rightarrow$ transmitted signal energy for each code bit &

$n_{jm} \rightarrow$ additive noise.

The relation for Viterbi soft decision decoder is given by,

$$\mu_1^i = \sum_{m=1}^{n} r_{jm} \left(2 c_{jm}^i - 1\right) \quad \text{— ②}$$

i → ith path, ~~ith~~

$c_{jm}^{(i)}$ → metric of jth branch in ith path

$$CM^{(i)} = \sum_{j=1}^{B} \mu_j^{(i)}$$

$$= \sum_{j=1}^{B} \sum_{m=1}^{n} r_{jm} \left(2 c_{jm}^{(i)} - 1\right) \quad \text{— ③}$$

Here, i → ith path & B → branches in that path.

Convolutional code does not have fixed length

we have

$$\left.\begin{aligned} P_2(d) &= P\left[CM^{(1)} \geq CM^{(0)}\right] \\ &= P\left[CM^{(1)} - CM^{(0)}\right] \end{aligned}\right\} \quad ④$$

from ③ & ④, we get,

$$P_2(d) = P\left\{2 \sum_{j=1}^{B} \sum_{m=1}^{n} r_{jm} \left[c_{jm}^{(1)} - c_{jm}^{(0)}\right]\right\} \quad \text{— ⑤}$$

∵ The diffn $\left(c_{jm}^{(1)} - c_{jm}^{(0)}\right) = 1$ at the bit positions which are in error because of incorrect path w.r.t. to all zero path.

$$\therefore P_2(d) = P\left\{\sum_{l=1}^{d} r_l \geq 0\right\} \quad \text{— ⑥}$$

Here, $\{r_l\}$ are the set of bits at 'd' positions.

$\{r_l\}$ have gaussian distribution with $-\sqrt{E_c}$ mean and $\frac{N_0}{2}$ variance. This is because these bits are basically error bits.

Hence, the above probability eqn can be written as,

$$P_2(d) = Q\left\{\sqrt{\frac{2E_c}{N_0} d}\right\} \quad \text{— ⑦}$$

The eqn can be written as,

$$P_2(d) = Q\left(\sqrt{2\gamma_b R_c d}\right) \quad — (8)$$

where,
$$\gamma_b = \frac{E_b}{N_0} \quad \& \quad R_c = \frac{E_c}{E_b}.$$

Hence, ~~...~~ probability of error with soft decision decoding is,

$$\boxed{P_e \leq \sum_{d=d_{jm}}^{\infty} a_d P_2(d) \leq \sum_{d=d_{jm}}^{\infty} a_d \left(Q\sqrt{2\gamma_b R_c d}\right)}$$ //

where, a_d ~~...~~ is the no. of paths of distance 'd' from the all zero path which merge with all zero path for the first time.

Probability of error with Hard decision decoding

The ~~BCH~~ binary symmetric channel uses hard decision decoding. Let us consider the performance of viterbi algorithm for hard decision taking.

Consider that the all zero path is transmitted, and the path which is selected has distance 'd' from the all zero path. With hard decision decoding the probability that incorrect path is selected is given as,

$$P_2(d) = \sum_{k=\frac{(d+1)}{2}}^{d} \binom{d}{k} p^k (1-p)^{d-k} \quad — (1)$$

where, p is the probability of error in the binary symmetric channel.

The probability error is,

$$\boxed{P_e < \sum_{d=d_{fm}}^{\infty} a_d P_2(d)}$$

Here, $\{a_d\}$ represents the no. of paths corresponding to the set of distances d.

4] Refer pg. no. 3-134 (Chitode).

5]. Soln:

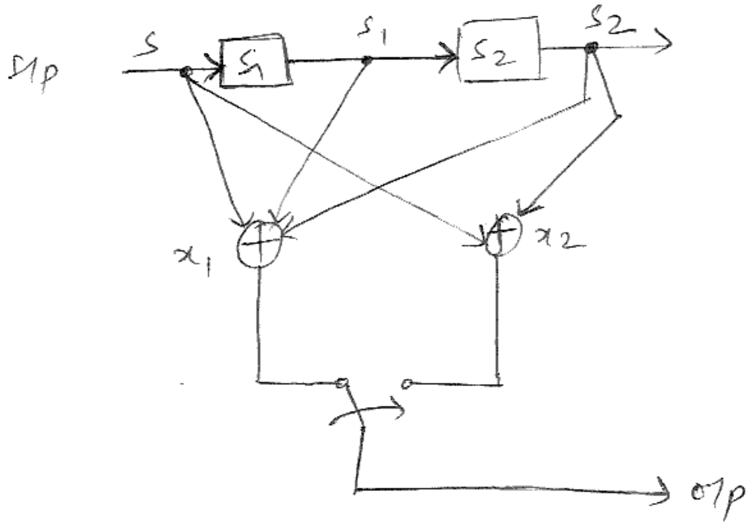


fig. convolutional encoder.

i) To ~~pe~~ prepare code Trellis

Let the states of the encoder is defined as,

$s_2 s_1 = 00$, state 'a'
$s_2 s_1 = 01$, state 'b'
$s_2 s_1 = 10$, state 'c'
$s_2 s_1 = 11$, state 'd'

A table is prepared that shows the state transitions, message i/p and o/p.

Sr. No.	current state $s_2 s_1$	I/p. s	outputs $x_1 = s \oplus s_1 \oplus s_2$	$x_2 = s \oplus s_2$	Next state $s_1 s$
1.	a = 00	0	0	0	00 ie. a
		1	1	1	01 ie. b.
2.	b = 01	0	1	0	10 ie. c
		1	0	1	11 ie. d
3	c = 10	0	1	1	00 ie. a
		1	0.	0	01 i.e b.
4.	d = 11	0	0	1	10 ie. c
		1	1	0	11. i.e. d

fig. State transition table for encoder.

Now, the code Trellis diagram can be drawn as.

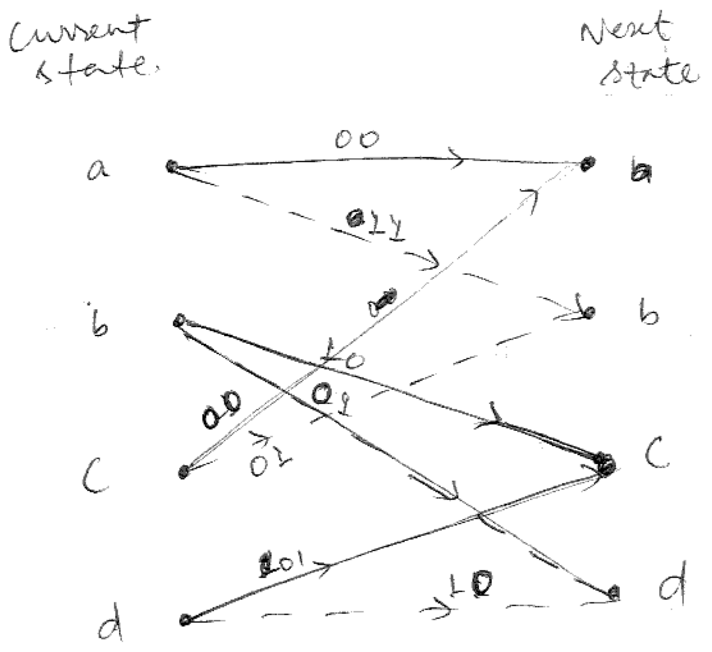


fig Code Trellis for encoder.

To prove double error detection :

Based upon code trellis of above fig, the trellis diagram is shown for multiple ~~path~~ stages. The diagram begins from node a_1. The O/Ps ~~are~~ and metric are marked along each branch. The commulative matric are also marked ~~as~~ near every node.

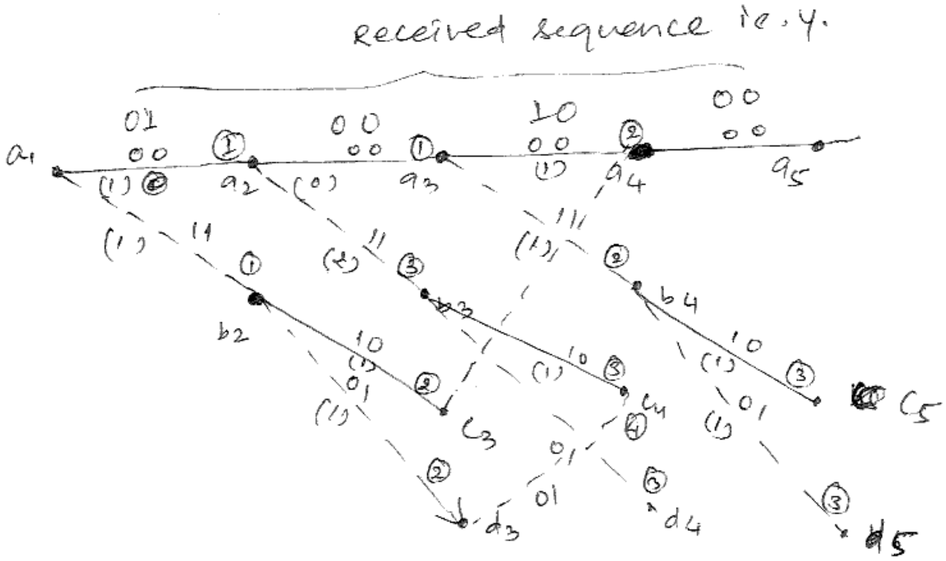


fig. viterbi algorithm for detection of all zero sequence.

c] Soln :

Constraint length = 3
Rate = 1/2

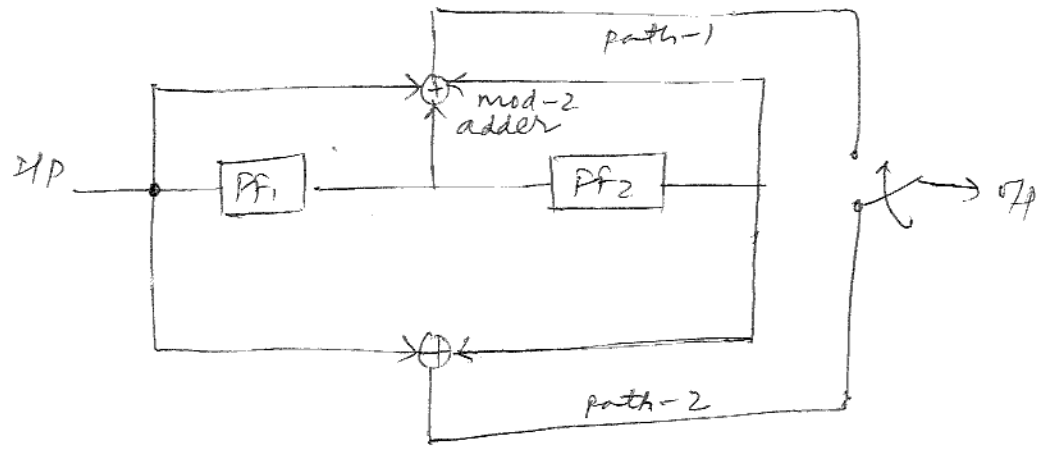


fig. convolutional encoder.

i) To determine the dimensions of the code.

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

$\therefore k = 1$
$n = 2$

For every message sequence, there are two bits encoded at the O/p.

Constraint length (K) = 3.

Hence, the O/p is influenced by three shifts in the receiver.

ii) ~~to~~ obtain state diagram & Trellis diagram

~~State of the encoder is~~

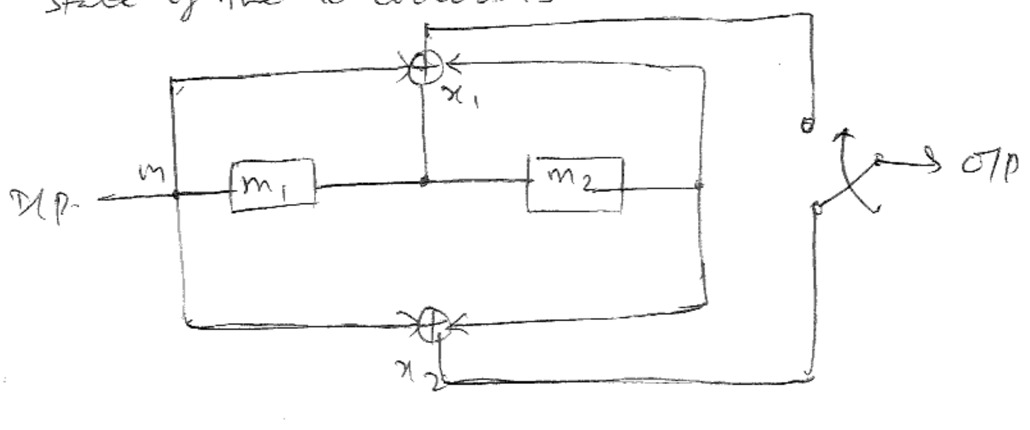


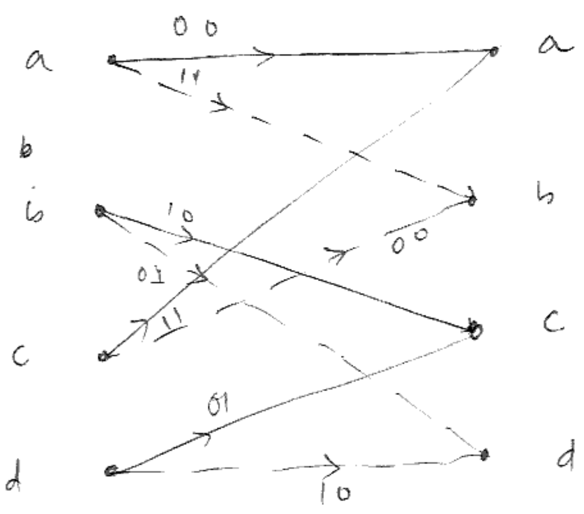
fig. convolutional encoder.

state of the encoder is represented by $m_1 m_2$, whereas i/p is m.

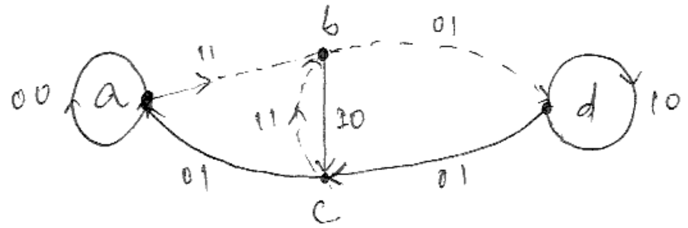
- It contains three stages in the shift register which contain m, m_1 and m_2.

Sr. No	Current state $m_2 m_1$	I/p m	O/ps $x_1 = m \oplus m_1 \oplus m_2$	$x_2 = m \oplus m_2$	Next state $m_1 m$
1.	a = 00	0	0	0	00 ie. a
		1	1	1	01 ie. b
2.	b = 01	0	1	0	10 ie. c
		1	0	1	11 ie. d
3.	c = 10	0	1	1	00 ie. a
		1	0	0	01 ie. b
4	d = 11	0	0	1	10 ie. c
		1	1	0	11 ie. d

Fig. state transition table



Trellis diagram.



State diagram

ii) For the o/p code to be systematic, the msg bit and the check bits must be identified. But this is not possible in the o/p sequence of given encoder. Hence, the generated code is not systematic.

iii) To decode 0100010000.

we have to draw viterbi decoding.