Unit-4

Channel Encoder & Decoder

Syllabus : Introduction, types of error control coding : Automatic Repeat Request, Forward Error Control - Linear block codes, error detection & Correction capabilities of linear block codes, Binary Cyclic codes, error detection & Correction capabilities in BCC, convolutional codes : Time domain & frequy domain approach, state diagram, trellis diagram, convolutional decoder : Viterbi & sequential decoder.

Introduction :

✓ The purpose of source coding (Encoding & decoding) is to convert the discrete message in to bits ('0' and '1').

✓ The purpose of channel coding is to detect and correct the errors.

* Channel coding is a combination of channel encoding and channel decoding.

✓ The signal passes through some noisy channel, because of noise errors are generated in the received data. This errors can be detected and corrected using coding techniques.

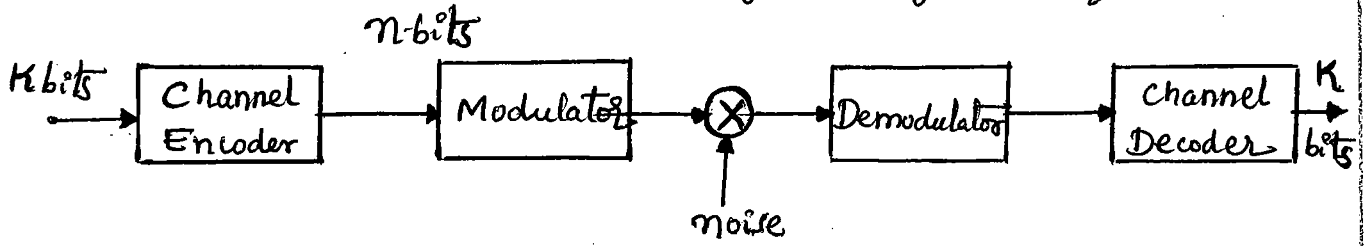


Fig: Channel Coding.

* **Channel Encoder:** The channel encoder add extra bits to the message sequence, The adding of extra bits is called parity bits (or) check bits (or) Redundant bits.

(PTO)

* Channel decoder :

The channel decoder identifies this redundant bits and use that to detect and correct the errors in the message bit.

✓ The channel encoder converts K bits into n bits by adding extra bits, such a code is called (n, K) block codes. Redundant bits is equal to (n-K).

Parameters :

① **Code Word :** The encoded block of n bits is called a code word. It contaınts message bits 'K' and parity bits (n-K).

② **Block length :** The no. of bits 'n' after coding is called block length of the code.

③ **Code rate :** The rate of the message bits (K) and the encoded bits (n) is called code rate.

$$\boxed{r_c = \frac{K}{n}}$$

④ **Channel data rate :** If the bit rate at the input of the encoder is R_s then the channel data rate at the output of the encoder will be

$$\boxed{R_c = \frac{n}{K} \cdot R_s}$$

$$\Rightarrow \boxed{R_c = \frac{R_s}{r_c}}$$

where

R_s → Source data bit rate.

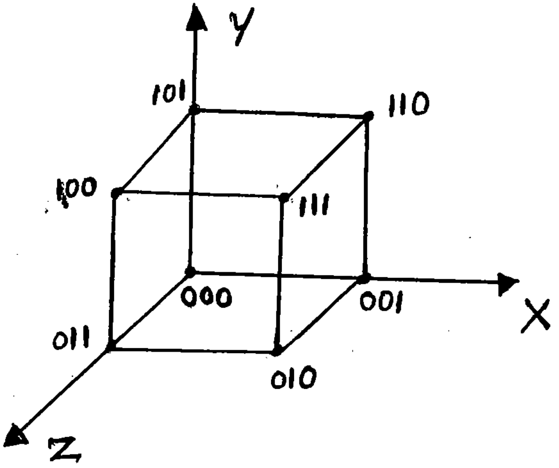
r_c → Code rate.

R_c → Channel data rate.

⑤ Code vector (or) Code book :

As n-bit code word can be visualize in an n-dimentional space as a vector whose elements & co-ordinates at the bits in ~~the~~ codeword.

ie Collection of all code words called as code book.



$K=3 \Rightarrow 2^3=8$ codewords

$K=4 \Rightarrow 2^4=16$ codewords.

⑥ Hamming distance :

The hamming distance between two code words is equal to the number of elements (or) bits in which they differ. ie The difference between two code words.

The hamming distance is denoted as 'd'

Ex:

$$X = 1\ 0\ 0$$
$$Y = 1\ 1\ 1$$

$\therefore d = 2. \quad \therefore \underline{d(x,y)=2}.$

⑦ Weight of the Code :

The number of non zero's (1's) elements in the transmitted code word is called weight of the code (or) vector rate.

It is denoted as $W(X)$, where X is Code word.

Ex: $X = 01110101$.

Weight of the Codeword is $\underline{W(X)=5}$

⑧ Minimum distance (d_{min}):

It is the smallest hamming distance between the Code Words except zero length. the following table list shows some of the requirements of error control capability of the code.

S.No	Error detection & Correction	Distance Requirement
1	Detect upto S-errors per Code word.	$d_{min} \geq s+1$.
2	Correct upto 't' error per code word.	$d_{min} \geq 2t+1$
③ *	Correct upto 't' errors and detect upto 's' errors for code	$d_{min} \geq t+s+1$ $(s>t)$

For (n,K) block code ~~for~~ the minimum distance is given by

$$\boxed{d_{min} \leq n-K+1}$$

The min-hamming distance should be two (2) for detection of errors and ~~it~~ should be more than two for correction of errors.

1 → Error can not be detected
2 → Single error detection
3 → Single error Correction

Ex: if $d_{min} = 3$ then

No-of detections are '2', No-of Corrections are '1'

ie $d_{min} \geq s+1$

$3 \geq s+1$

$2 \geq s$

$\therefore \boxed{s \leq 2}$

4 → Single error Correction + double error detection
5 → Double error detection
6 → Double error Correction + triple errors detection.

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

$3 \geq 2t+1$

$2 \geq 2t$

$1 \geq t$

$\therefore \boxed{t \leq 1}$

* $\frac{n-1}{2}$ error canbe corrected if n-odd.

$\frac{n-2}{2}$ errors canbe corrected & $n/2$ errors can be detected if n even.

Error Control Coding :

* The probability of error for a particular signalling scheme is a function of signal to noise ratio at the receiver input and the information rate.

* In practical systems, the maximum signal power and the bandwidth of the channel are restricted to some fixed values. Also, the noise power spectral density $\frac{\eta}{2}$ is fixed for a particular operating environment.

✓ With all these constraints, it is often not possible to arrive at a signalling scheme which will yield an acceptable probability of error for a given application.

★ The only practical alternative for reducing the probability of error is the use of 'error-control coding'.

Types of Error Control Coding:

There are two methods of error control coding.

① . ARQ [Automatic Retransmission Query (or) Automatic Repeat Request) (or) (Error detection with retransmission)

② FEC. [Forward Error Correction]

① ARQ [Automatic Repeat Request] :

✓ In this method the receiver checks the input sequence if there is any error occurs then the receiver discard that part of the sequence and request the transmitter for retransmission. So, it is called automatic repeat request.

ie. When an error is detected, the receiver send a request to the transmitter for repeat transmission, after which the transmitter retransmits.

✓ Consider a source which emits 'm' messages bits and receiver receives those m-messages and finds one bit is corrupted Hence the receiver send a request to source for retransmission.

✓ The source again sends the m-messages in this method, the time taken for retransmission is more.

✓ The ARQ method needs duplex arrangement as apart from the conventional transmitter to receiver signal, the request signal is to travel from receiver to transmitter. It cannot be used in real-time systems.

Advantage:

① ARQ can be used to transmit messages more occurately.

Disadvantage:

① The probability of error is low its process is also slow.

② Even though P_e is low the retransmission of entire message takes place which leads a delay / time waste in transmission.

③ Only detection possible and cann't be correct the errors.

② FEC [Forward Error Correction] :

✓ In this method the errors are detected and corrected by proper coding techniques.

✓ In FEC method, an automatic mechanism of error correction in the form of error correction code is employed & hence retransmission of data is not necessary.

✓ The FEC method needs simplex arrangement (Simple hardward as compared to duplex arrarangement) as the signal has to travel only from the transmitter to the receiver.,

✓ The FEC method is more popular & used in real time systems than ARQ method.

✓ In FEC, the channel encoder systematically adds digits/bits to the transmitted message bits. Although these additional bits convey no new information, they make it possible for the channel decoder to detect & correct the errors in the information bearing digits/bits.

✓ The overall probability of error is reduced due to error detection and error correction.

Error Control Coding Techniques (or) Structured Sequences:

Structured Sequences are divided into 3 sub catagerious.

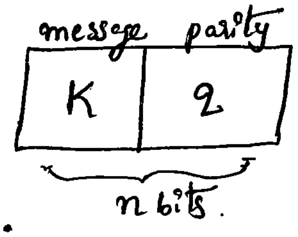
(a) Block Codes
(b) Cyclic Codes
(c) Convolutional Codes
(d). Turbo codes

Block Codes:

Block code is also known as arithmetic code (or) group code. There are two types of block codes.

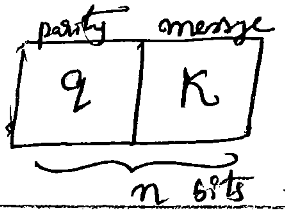
(i) Systematic block code.
(ii) Non-Systematic block code.

Systematic block code: In resultant 'n' bits, the first bits are message bits and remaining 'n-k' bits are check bits called as systematic block code.



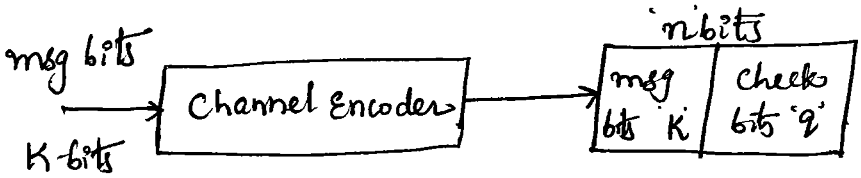
$q = n - K$

Non Systematic block code: In resultant 'n' bits the first bits are check bits/parity bits and remaining bits are message bits, called as non systematic block code.



$q = n - K$

- Each block of 'K' messages or message bits is encoded into a block of n-bits. (n > K).
- The check bits are drived from the message bits and that check bits are added to the message bits.



$n = K + q$

Parity Check Codes:

The simplest possible block code is parity check codes. In this method a no. of check bits added to the message bit is one.

There are two types of parity check codes.

(a) Even parity check code.

(b) Odd parity check code.

Even parity checkcode:

When the check bit may be '1' or '0' such that the total no. of 1's in the encoded of 'n' bits code word is even then it is called Even parity check code.

Ex:

	msg (K)	Checkbit(q)		Resultant bit (n)
X =	010011	1	→	0100111
Y =	101110	0	→	1011100

Odd parity check code:

When the check bit may be '1' & '0' such that the total no. of 1's in the encoded of 'n' bits code word is Odd ~~then~~ it is called odd parity check code.

Ex:

	msg(K)		checkbit (q)		Resultant bit (n)
X =	010011	→	0	→	0100110
Y =	101110	→	1	→	1011101

Linear Block Codes :

A Code is said to be linear block code if the sum (XOR) of any two code words in the codebook produced another codeword which is in the code book → set of code words.

- ✓ The basic notation for describing linear block codes in a matrix form.
- ✓ The error control capabilities of a linear block codes are determined by its minimum distance.

* The encoding operation in a linear block encoding scheme consists of two basic steps.

(i) The information sequence is segmented / divided into message blocks, each block consists of 'K' bits.

(ii) The channel encoder transform each block of 'K' bits into large block of n-bits.

Consider the message block as a row vector $[M_1, M_2, \dots M_K]$ each message bit can be '1' or '0'.

- ✓ There are 2^K different message blocks (or) 2^K different code words.
- ✓ Each message block is transformed to a code word 'C' of length n bits. $C = [C_1, C_2, C_3 \dots C_n]$
- ✓ The set of 2^K code words is called Codebook & Code Vector.
- ✓ The check bits are generated from the message bits.
- ✓ The output of the channel encoder is 'n' bits which can be generated by using matrix form.

$$\underbrace{\boxed{C = M\overset{\text{message bits}}{} G}}_{}$$

Code words ← $\boxed{C = MG}$ → message bits

→ Generator matrix

$$[C]_{1\times n} = [M]_{1\times K} \cdot [G]_{K\times n}.$$

$$\begin{bmatrix} C_1 & C_2 & C_3 & \dots & C_n \end{bmatrix} = \begin{bmatrix} M_1 & M_2 & M_3 & \dots & M_K \end{bmatrix} \left[\begin{array}{cccc|cccc} 1 & 0 & 0 & \dots & P_{11} & P_{12} & \dots & P_{1q} \\ 0 & 1 & 0 & \dots & P_{21} & P_{22} & \dots & P_{2q} \\ 0 & 0 & 1 & \dots & P_{31} & P_{32} & \dots & P_{3q} \\ \vdots & \vdots & & \vdots & & & & \\ 0 & 0 & 0 & \dots 1 & P_{K1} & P_{K2} & \dots & P_{Kq} \end{array}\right]$$

where $[G]_{K\times n} = [I_{K\times K} \vdots P_{K\times q}]_{K\times n}.$

where $[G]_{K\times n}$ is called generator matrix

$[I]_{K\times K}$ is called Identity matrix.

$[P]_{K\times q}$ is called parity matrix. ← Important role.

An importance steps in the design of (n,K) block codes is the selection of a 'p' matrix so that the code generated by 'G' has some properties such as

* Easy of Implementation.
* Ability of correct errors.
* High rate efficiency.

① The generator matrix for a (6,3) block code is given below Find all the code words.

$$G = \begin{bmatrix} \underbrace{\begin{matrix}1&0&0\\0&1&0\\0&0&1\end{matrix}}_{I_{3\times3}} & \vdots & \underbrace{\begin{matrix}0&1&1\\1&0&1\\1&1&0\end{matrix}}_{P_{3\times3}} \end{bmatrix}$$

Sol

The given block code $(n,K) = (6,3)$

$n = 6$
$K = 3$

∴ Length of the message block $K = 3$.

Length of the encoded codewords $n = 6$.

Check bits $q = n-k = 6-3 = 3$

Systematic system block code.

∴ There are $2^K = 2^3 = 8$ different message block, each message block consists of 3 bits.

$$\left.\begin{array}{ccc} M_1 & M_2 & M_3 \\ 0&0&0\\0&0&1\\0&1&0\\0&1&1\\1&0&0\\1&0&1\\1&1&0\\1&1&1 \end{array}\right\} \text{8 Each 3 bits}$$

$$C = M.G$$

Consider $M_1 = [000]$

$$\therefore \underset{1\times6}{C} = \underset{1\times3}{[000]} \begin{bmatrix} 100 & 011 \\ 010 & 101 \\ 001 & 110 \end{bmatrix}_{3\times6}.$$

$$\boxed{C_1 = [0\ 0\ 0\ \ 0\ 0\ 0]}$$

ie Each bit in the row matrix 'M' should be multiplied by each bits in the column of 'G' matrix.

Then all the values are added by 'XOR' operation

XOR operation :

00	0
01	1
10	1
11	0

$M_2 = [0 0 1]$, $C_2 = [0 0 1] \begin{bmatrix} 1 0 0 & 0 1 1 \\ 0 1 0 & 1 0 1 \\ 0 0 1 & 1 1 0 \end{bmatrix}$

$= [0+0+0 \quad 0+0+0 \quad 0+0+1 \quad 0+0+1 \quad 0+0+1 \quad 0+0+0]$

$\boxed{C_2 = [0 \; 0 \; 1 \; 1 \; 1 \; 0]}$

$M_3 = [0 1 0]$, $C_3 = [0 1 0] \begin{bmatrix} 1 0 0 & 0 1 1 \\ 0 1 0 & 1 0 1 \\ 0 0 1 & 1 1 0 \end{bmatrix}$

$= [0+0+0 \quad 0+1+0 \quad 0+0+0 \quad 0+1+0 \quad 0+0+0 \quad 0+1+0]$

$\boxed{C_3 = [0 \; 1 \; 0 \; 1 \; 0 \; 1]}$

$M_4 = [0 1 1]$, $C_4 = [0 1 1] \begin{bmatrix} 1 0 0 & 0 1 1 \\ 0 1 0 & 1 0 1 \\ 0 0 1 & 1 1 0 \end{bmatrix}$

$= [0+0+0 \quad 0+1+0 \quad 0+0+1 \quad 0+1+1 \quad 0+0+1 \quad 0+1+0]$

$\boxed{C_4 = [0 \; 1 \; 1 \; 0 \; 1 \; 1]}$

$M_5 = [1 0 0]$ $C_5 = [1 0 0] \begin{bmatrix} 1 0 0 & 0 1 1 \\ 0 1 0 & 1 0 1 \\ 0 0 1 & 1 1 0 \end{bmatrix}$

$= [1+0+0 \quad 0+0+0 \quad 0+0+0 \quad 0+0+0 \quad 1+0+0 \quad 1+0+0]$

$\boxed{C_5 = [1 \; 0 \; 0 \; 0 \; 1 \; 1]}$

$M_6 = [1 0 1]$, $C_6 = [1 0 1] \begin{bmatrix} 1 0 0 & 0 1 1 \\ 0 1 0 & 1 0 1 \\ 0 0 1 & 1 1 0 \end{bmatrix}$

$= [1+0+0 \quad 0+0+0 \quad 0+0+1 \quad 0+0+1 \quad 1+0+1 \quad 1+0+0]$

$\boxed{C_6 = [1 \; 0 \; 1 \; 1 \; 0 \; 1]}$

$M_7 = [1 1 0]$, $C_7 = [1 1 0] \begin{bmatrix} 1 0 0 & 0 1 1 \\ 0 1 0 & 1 0 1 \\ 0 0 1 & 1 1 0 \end{bmatrix}$

$= [1+0+0 \quad 0+1+0 \quad 0+0+0 \quad 0+1+0 \quad 1+0+0 \quad 1+1+0]$

$\boxed{C_7 = [1 \; 1 \; 0 \; 1 \; 1 \; 0]}$

$M_8 = [1 1 1]$ $C_8 = [1 1 1] \begin{bmatrix} 1 0 0 & 0 1 1 \\ 0 1 0 & 1 0 1 \\ 0 0 1 & 1 1 0 \end{bmatrix}$

$= [1+0+0 \quad 0+1+0 \quad 0+0+1 \quad 0+1+1 \quad 1+0+1 \quad 1+1+0]$

$\boxed{C_8 = [1 \; 1 \; 1 \; 0 \; 0 \; 0]}$

SL NO	Message	Code word
1	0 0 0	0 0 0 0 0 0
2	0 0 1	0 0 1 1 1 0
3	0 1 0	0 1 0 1 0 1
4	0 1 1	0 1 1 0 1 1
5	1 0 0	1 0 0 0 1 1
6	1 0 1	1 0 1 1 0 1
7	1 1 0	1 1 0 1 1 0
8	1 1 1	1 1 1 0 0 0

Alternative Method :

Consider $C = MP$.

$$C = [M_1\ M_2\ M_3] \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

$$C = [M_2 \oplus M_3 \quad M_1 \oplus M_3 \quad M_1 \oplus M_2]$$

$\therefore$ $c_1\ c_2\ c_3 \rightarrow M_1\ M_2 M_3$

$c_4 = M_2 \oplus M_3 \longrightarrow$ ①

$c_5 = M_1 \oplus M_3 \longrightarrow$ ②

$c_6 = M_1 \oplus M_2 \longrightarrow$ ③

$\therefore$ The codeword can be written as $\begin{array}{c}[c_1\ .c_2\ c_3 \quad c_4\ c_5\ c_6] \\ M_1\ M_2\ M_3 \quad ?\ \ ?\ \ ?\end{array}$

S No	Messages $M_1\ M_2\ M_3$	Check bits $c_4\ c_5\ c_6$ ① ② ③	Complete Code Word $M_1\ M_2\ M_3\ c_4\ c_5\ c_6$
1	0 0 0	0 0 0	0 0 0 0 0 0
2	0 0 1	1 1 0	0 0 1 1 1 0
3	0 1 0	1 0 1	0 1 0 1 0 1
4	0 1 1	0 1 1	0 1 1 0 1 1
5	1 0 0	0 1 1	1 0 0 0 1 1
6	1 0 1	1 0 1	1 0 1 1 0 1
7	1 1 0	1 1 0	1 1 0 1 1 0
8	1 1 1	0 0 0	1 1 1 0 0 0

② The generator Matrix for a (6,3) block code is given below. Find all the code words.

$$G = \left[\begin{array}{ccc:ccc} 1&0&0&1&1&0 \\ 0&1&0&0&1&1 \\ 0&0&1&1&1&1 \end{array}\right]$$

Sol

The given block code $(n,k) = (6,3)$.

$n = 6$
$k = 3$

$$G = \left[\begin{array}{ccc|ccc} 1&0&0&1&1&0 \\ 0&1&0&0&1&1 \\ 0&0&1&1&1&1 \end{array}\right]$$
$$\quad I_{3\times3} \qquad P_{3\times3}.$$

Length of the message block $k = 3$

length of the encoded code words $n = 6$.

Check bits $q = n - k = 6 - 3 = 3 = 3$.

∵ There are $2^k = 2^3 = 8$ diffent messge block, each message block consists of 3 bits.

$$\therefore \quad C = M \cdot [G]$$

and

$$C = [M][P]$$
$$= [M_1 \; M_2 \; M_3] \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$C = [M_1 \oplus M_3 \quad M_1 \oplus M_2 \oplus M_3 \quad M_2 \oplus M_3]$$

$\therefore \quad c_1 \; c_2 \; c_3 = M_1 \; M_2 \; M_3$
$c_4 = M_1 \oplus M_3$
$c_5 = M_1 \oplus M_2 + M_3$
$c_6 = M_2 \oplus M_3$.

SL NO	Messages M_1 M_2 M_3	Check bits C_4 C_5 C_6	Comple code word M_1 M_2 M_3 C_4 C_5 C_6
1	0 0 0	0 0 0	0 0 0 0 0 0
2	0 0 1	1 1 1	0 0 1 1 1 1
3	0 1 0	0 1 1	0 1 0 0 1 1
4	0 1 1	1 0 0	0 1 1 1 0 0
5	1 0 0	1 1 0	1 0 0 1 1 0
6	1 0 1	0 0 1	1 0 1 0 0 1
7	1 1 0	1 0 1	1 1 0 1 0 1
8	1 1 1	0 1 0	1 1 1 0 1 0

Parity Check Matrix (H):

✓ The parity check matrix can be used to verify wether the code word is generated by the matrix 'G' or not.

✓ The code word in the (n,K) block code generated by 'G' if and only if $CH^T = 0$

$$\Rightarrow MGH^T = 0$$

where M - messge matrix

G - Generator matrix, $G = \left[I_{K\times K} \cdot P_{K\times q} \right]_{K\times n}$

H - parity check matrix $\boxed{H = \left[P^T_{q\times K} \vdots I_{q\times q} \right]_{q\times n}}$

$$H^T = \begin{bmatrix} (P^T)^T \\ \hline I \end{bmatrix} = \begin{bmatrix} P_{K\times q} \\ \hline I_{q\times q} \end{bmatrix}_{n\times q}.$$

$$P = \begin{bmatrix} P_{11} & P_{12} & \cdots & P_{1q} \\ P_{21} & P_{22} & \cdots & P_{2q} \\ \vdots & & & \\ P_{K1} & P_{K2} & \cdots & P_{Kq} \end{bmatrix}, \quad P^T = \begin{bmatrix} P_{11} & P_{21} & \cdots & P_{K1} \\ P_{12} & P_{22} & \cdots & P_{K2} \\ \vdots & \vdots & & \vdots \\ P_{1q} & P_{2q} & \cdots & P_{Kq} \end{bmatrix}$$

Parity Check matrix

$$H = \left[\begin{array}{cccc:cccc} P_{11} & P_{21} & \cdots & P_{K1} & 1 & 0 & 0 \cdots & 0 \\ P & P_{22} & \cdots & P_{K2} & 0 & 1 & 0 \cdots & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & \vdots \\ P_{1q} & P_{2q} & \cdots & P_{Kq} & 0 & 0 & 0 \cdots & 1 \end{array}\right]_{q\times n}$$

Error detection & Correction Capabilities of Linear block code:

Hamming codes are (n,K) linear block codes.

These codes satisfies the following conditions.

① The number of check bits $q \geq 3$

② Block length $n = 2^q - 1$

③ The number of message bits $K = n - q$

④ The minimum distance $d_{min} \geq 2$.

If $d_{min} = 2$ then single error can be detected.

If $d_{min} = 3$ then two errors can be detected and one error can be corrected.

⑤ The code rate is given as

$$r_c = \frac{K}{n} = \frac{n-q}{n} = 1 - q/n$$

$$\boxed{r_c = 1 - \frac{q}{2^q-1}}$$

⑥ Error detection & Correction capabilities of hamming codes are.

* $d_{min} \geq s+1$ → Detect upto 's' error per codeword
* $d_{min} \geq 2t+1$ → Correct upto 't' errors per codeword.
* $d_{min} \geq s+t+1$ → Detect upto 's' errors and Correct upto 't' errors for code word. $(s>t)$.

③ The parity check matrix of a particular (7,4) linear block code is given by

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

(a) Find the generator matrix G .

(b) List all the code words.

(c) Minimum distance between the Codewords.

(d) How many errors can be detected and how many error corrected.

(e) Draw the encoded diagram for given hamming code .

Sol

Linear block code $(n,K) = (7,4)$ ie $n=7$, $K=4$.

The length of the message block $k=4$

The length of the encoded code word $n=7$.

Check bits $q = n-K = 7-4 = 3$

$\therefore$ There are $2^K = 2^4 = 16$ different message block, each message block consists of 4 bits.

Given parity check matrix $H = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}_{3\times7}$

$$H = \left[P^T_{q\times K} \vdots I_{q\times q} \right]_{q\times n}$$

$$P = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}_{K\times q \text{ ie } 4\times3} \qquad \left[\because P = (P^T)^T\right]$$

(a) Generator matrix: $G = \left[I_{K\times K} \ P_{K\times q} \right]_{K\times n}$

$$\therefore G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}_{4\times7}.$$

(b) All Code words:

$$C = MG$$

$$G = [I \vdots P]$$

$$C = [M][P]$$

$$\therefore \underset{1\times3}{C} = \underset{1\times4}{[M_1 \ M_2 \ M_3 \ M_4]} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}_{4\times3}$$

$$C = [M_1 \oplus M_2 \oplus M_3 \quad M_1 \oplus M_2 \oplus M_4 \quad M_1 \oplus M_3 \oplus M_4]$$

$$\therefore [c_1, c_2, c_3, c_4] = [M_1 \ M_2 \ M_3 \ M_4]$$

$$c_5 = M_1 \oplus M_2 \oplus M_3 \longrightarrow ①$$

$$c_6 = M_1 \oplus M_2 \oplus M_4 \longrightarrow ②$$

$$c_7 = M_1 \oplus M_3 \oplus M_4 \longrightarrow ③$$

$C_5 = M_1 \oplus M_2 \oplus M_3$, $C_6 = M_1 \oplus M_2 \oplus M_4$, $C_6 = M_1 \oplus M_3 \oplus M_4$

SL No	Message M_1 M_2 M_3 M_4	Check bits c_5 c_6 c_7	Complete Codeword M_1 M_2 M_3 M_4 c_5 c_6 c_7	Vector wt. Weight
1	0 0 0 0	0 0 0	0 0 0 0 0 0 0	0
2	0 0 0 1	0 1 1	0 0 0 1 0 1 1	3
3	0 0 1 0	1 0 1	0 0 1 0 1 0 1	3
4	0 0 1 1	1 1 0	0 0 1 1 1 1 0	4
5	0 1 0 0	1 1 0	0 1 0 0 1 1 0	3
6	0 1 0 1	1 0 1	0 1 0 1 1 0 1	4
7	0 1 1 0	0 1 1	0 1 1 0 0 1 1	4
8	0 1 1 1	0 0 0	0 1 1 1 0 0 0	3
9	1 0 0 0	1 1 1	1 0 0 0 1 1 1	4
10	1 0 0 1	1 0 0	1 0 0 1 1 0 0	3
11	1 0 1 0	0 1 0	1 0 1 0 0 1 0	3
12	1 0 1 1	0 0 1	1 0 1 1 0 0 1	4
13	1 1 0 0	0 0 1	1 1 0 0 0 0 1	3
14	1 1 0 1	0 1 0	1 1 0 1 0 1 0	4
15	1 1 1 0	1 0 0	1 1 1 0 1 0 0	4
16	1 1 1 1	1 1 1	1 1 1 1 1 1 1	7

dmin = 3 Except 0.

(c)

The minimum distance of the (n,k) block code is $\leq$ 'n-K+1'

∴ The minimum hamming distance dmin = 3

$$\therefore \quad d_{min} \leq n-k+1$$

$$3 \leq (7-4+1)$$

$$3 \leq 4 \quad (\text{True}) \checkmark$$

$$3 < 4$$

$\therefore \boxed{d_{min} = 3}$

(d) No. of detections of error 's' ⇒ $d_{min} \geq s+1$

$$3 \geq s+1$$

$$2 \geq s \qquad \therefore \; s \leq 2$$

$\therefore$ $\boxed{\text{No. of error detections } s = 2}$

No. of error Corrections 't' ⇒ $d_{min} \geq 2t+1$

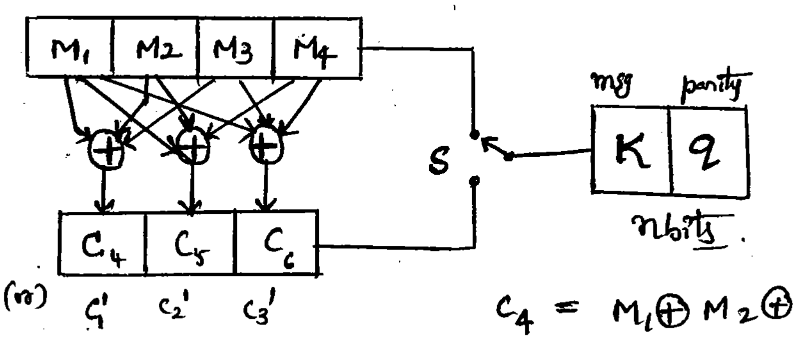
$$3 \geq 2t+1$$

$$1 \geq t \;\therefore\quad t \leq 1$$

$\therefore$ $\boxed{\text{No. of error corrections } t = 1}$

(e) Encoder diagram :

The encoder of (7,4) hamming code is implemented for the given generator matrix.



S- switch.

$C_4 = M_1 \oplus M_2 \oplus M_3$

$C_5 = M_1 \oplus M_2 \oplus M_4$

$C_6 = M_1 \oplus M_3 \oplus M_4$.

✓ The check bits c_1', c_2', c_3' are obtained from the message bits by modulo-2 & XOR operation.

✓ The switch 'S' is connected to the message register first then transmit all the message bits.

✓ Then switch 'S' is then connected to the check bit register then all check bits are transmitted.

Thus it form a 7 bit block Codes.

Syndrome Decoding :

* The generator matrix (G) is used in the encoding operation & the parity check matrix (H) is used in the decoding operation.

✓ The encoder has to store the G-matrix & perform binary arithmetic operations to generate the check bits.

✓ The complexity of the encoder increases as the no-of check bits are increases.

* Let transmitted code vector be 'X', the received code vector be 'Y'

If X = Y then there are no transmission errors.

If X ≠ Y then there are errors created during transmission.

- ✓ The decoder detects & Corrects the errors in 'y' using the stored information about the codeword.
- ✓ A direct way of performing error detection would be comparing 'y' with every codeword in the code book.

Let 'c' be a codeword that was transmitted over a noisy channel & it is received as 'y'.

The syndrome is represented by 'S' & It can be written as

Detection. $\boxed{S = YH^T}$ where $[H]_{q \times n}$ - parity check matrix.

$$[S]_{1 \times q} = [Y]_{1 \times n} [H^T]_{n \times q}$$

Let us Consider error vector 'E' & this vector represent the position of transmission errors in 'y'.

So $Y = X \oplus E$ $\Rightarrow$ $\boxed{X = Y \oplus E}$

Substitute 'y' in 'S' correction.

ie

$$S = [X \oplus E] \cdot H^T$$

$$= XH^T \oplus EH^T \qquad (\because XH^T = 0).$$

$$S = EH^T$$

$$\therefore \boxed{S = EH^T} \qquad \text{For Single bit error.}$$

This relation shows ~~that~~ ~~the~~ syndrome depends upon error pattern only. It does not depends upon a particular message.

Syndrome vector 'S' of Size '1xq'.

q bits of syndrome can only represents '$2^q - 1$' syndrome vectors.

Each syndrome vector corresponds to a particular error pattern.

Hence in syndrome decoding we can detect and correct only one error.

$\therefore$ The no. of errors can be corrected $\Leftrightarrow$ $1 - r_c \geq \left[\frac{1}{n} \log_2 \sum_{i=1}^{t} {}^{n}c_i\right]$

This eqn also called Hamming bound.

④ The parity check matrix of a particular (6,3) block code is given as

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

(a) Determine the generator matrix G

(b) Find all the code words.

(c). Find the min distance between the codeword.

(d) How many errors can be detected & Corrected.

(e) Draw the encoder diagram.

(f) Calculate the syndrome vector for single bit error.

(g) Suppose that the received codeword is '1 1 0 1 1 0', decode this code word.

(h) Draw the syndrome diagram.

Sol: Given block code $(n,K) = (6,3)$ $n = 6$, $K = 3$.

Length of the message blocks $K = 3$

Length of the encoded code word $n = 6$.

Check bits $q = n-K = 6-3 = 3$.

∴ $K=3$. There are $2^K = 2^3 = 8$ different message block, each message block consists of 3 bits.

$$H = \left[\begin{array}{ccc|ccc} 1&0&1&1&0&0 \\ 1&1&0&0&1&0 \\ 0&1&1&0&0&1 \end{array}\right]_{3\times6}$$

$$H = [\underset{3\times3}{P^T} \vdots \underset{3\times3}{I}]_{3\times6}$$

Where $P^T = \begin{bmatrix} 1&0&1 \\ 1&1&0 \\ 0&1&1 \end{bmatrix} \Rightarrow P = \begin{bmatrix} 1&1&0 \\ 0&1&1 \\ 1&0&1 \end{bmatrix}_{3\times3}, \; I = \begin{bmatrix} 1&0&0 \\ 0&1&0 \\ 0&0&1 \end{bmatrix}_{3\times3}$

(a) Generator matrix $G = [I_{K\times K} \vdots P_{K\times q}]_{K\times n}$

$$G = \begin{bmatrix} 1&0&0&1&1&0 \\ 0&1&0&0&1&1 \\ 0&0&1&1&0&1 \end{bmatrix}_{3\times6}$$

(b) Code words $C = MG$

(or) $C = [M][P] = [M_1 \; M_2 \; M_3] \begin{bmatrix} 1&1&0 \\ 0&1&1 \\ 1&0&1 \end{bmatrix}$

$\therefore\ c_1\ c_2\ c_3 \to M_1\ M_2\ M_3$

$C_4 = M_1 \oplus M_3$

$C_5 = M_1 \oplus M_2$

$C_6 = M_2 \oplus M_3$.

SL NO	Messages $M_1\ M_2\ M_3$	Check bits $C_4\ C_5\ C_6$	Complete Code word $M_1\ M_2\ M_3\ C_4\ C_5\ C_6$	Vector rate $W(X)$
1	0 0 0	0 0 0	0 0 0 0 0 0	0
2	0 0 1	1 0 1	0 0 1 1 0 1	3
3	0 1 0	0 1 1	0 1 0 0 1 1	3
4	0 1 1	1 1 0	0 1 1 1 1 0	4
5	1 0 0	1 1 0	1 0 0 1 1 0	3
6	1 0 1.	0 1 1	1 0 1 0 1 1	4
7	1 1 0	1 0 1	1 1 0 1 0 1	4
8	1 1 1	0 0 0	1 1 1 0 0 0	3

(c) The minimum distance $d_{min} = 3$ except 0. $\therefore \boxed{d_{min} = 3}$

(d) No of detection of errors.

$d_{min} \geq s+1$

$3 \geq s+1$

$s \leq 2$

$\therefore$ No of error detections $\boxed{s = 2}$

No of error corrections.

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

$3 \geq 2t+1$

$t \leq 1$

$\therefore \boxed{t = 1}$. No of error corrections.

For (n,K) block code the minimum distance is given by

$d_{min} \leq n-K+1$

$3 \leq 6-3+1$ $\quad n=6$, $K=3$

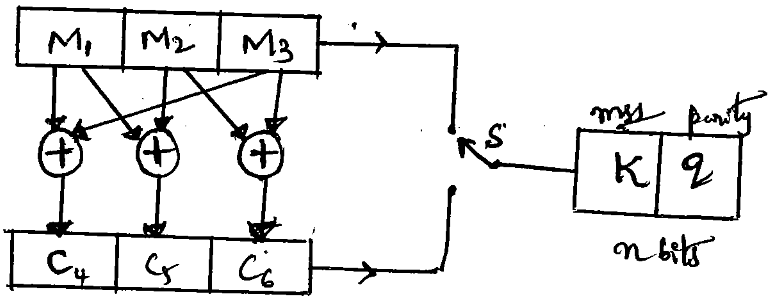
$3 \leq 4$ (True).

(e) The encoder diagram of (6,3) block code is implemented for the given generator matrix.

$C_4 = M_1 \oplus M_3$

$C_5 = M_1 \oplus M_2$

$C_6 = M_2 \oplus M_3$.



(f) • Syndrome vector $S = EH^T$. for single bit error

$$H^T = \begin{bmatrix} (P^T)^T \\ I \end{bmatrix} = \begin{bmatrix} P \\ I \end{bmatrix}_{n \times q}$$

$$\therefore \quad H^T = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}_{6\times 3} \qquad \text{where} \quad P = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

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

Error Vector E as

$$E = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}_{6\times 6}.$$

Syndrome vector $S = EH^T$

for $E = 100000$, $S = [1\,00\,000] \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = [1 \; 1 \; 0]$

Ily.

Sl No	Bit in error	Bit in Error vector	Syndrome Vector S
1	1^{st}	1 0 0 0 0 0	1 1 0
2	2^{nd}	0 1 0 0 0 0	0 1 1
3	3^{rd}	0 0 1 0 0 0	1 0 1
4	4^{th}	0 0 0 1 0 0	1 0 0
5	5^{th}	0 0 0 0 1 0	0 1 0
6	6^{th}	0 0 0 0 0 1	0 0 1

(g) For the detection of the error by using syndrome decoding is given as $S = YH^T$

Given received bits (codeword) $Y = [1 \; \underset{\uparrow}{1} \; 0 \; 1 1 0]$

$$\underset{1\times 3}{S} = [1 1 0 \; 1 1 0]_{1\times 6} \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}_{6\times 3}$$

$$\Rightarrow S = [1\oplus 0 \oplus 0 \oplus 1 \oplus 0 \oplus 0 \quad 1 \oplus 1 \oplus 0 \oplus 0 \oplus 1 \oplus 0 \quad 0 \oplus 1 \oplus 0 \oplus 0 \oplus 0 \oplus 0]$$

$$\underline{S = [0 \; 1 \; 1]} \qquad \therefore \; 2^{nd} \text{ bit Error}$$

Error correction is possible by using

$$X = Y \oplus E$$

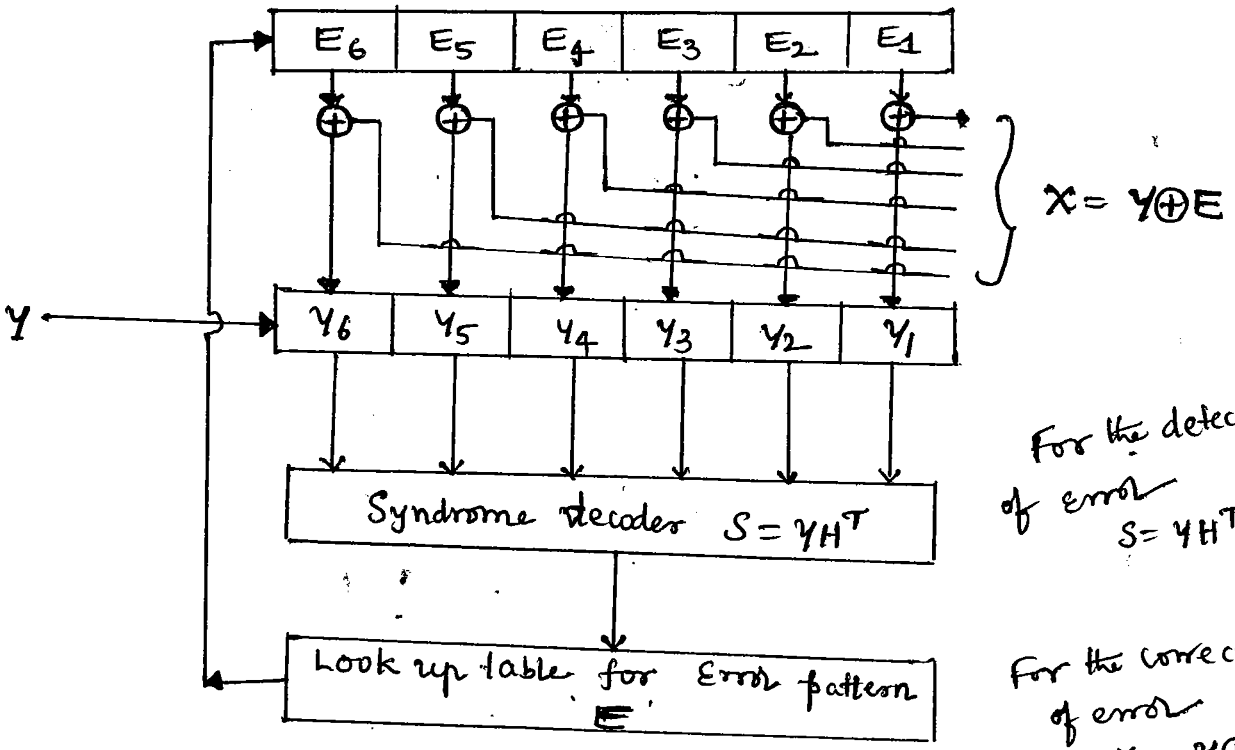
$$= [1\,1\,0\,1\,1\,0] \oplus [0\,1\,0\,0\,0\,0]$$

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

$$\therefore X = [\underbrace{1\,0\,0}_{\uparrow\ \text{message bits}}\ \underbrace{1\,1\,0}_{\uparrow\ \text{parity bits}}]$$

(h) Syndrome diagram :

Error bits → n bits
Rxed bits → n bits.



For the detection of error $S = YH^T$

For the correction of error $X = Y \oplus E$.

⑤ The parity check matrix of a particular (7,4) linear block codes is given by :

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

(a) Calculate syndrome vector for single bit error

(b) Suppose that the Rxed code word is 0110110. decode this codeword.

(c) Draw the syndrome diagram.

Sol Given linear block code $(n,k) = (7,4)$

Length of the message block $k = 4$

length of the Encoded code word $n = 7$.

check bits $q = n - k = 7 - 4 = 3$

$K = 4., \quad n = 7, \quad q = 3.$

$$H = \begin{bmatrix} 1110 & 100 \\ 1101 & 010 \\ 1011 & 001 \end{bmatrix} \qquad H = [P^T \vdots I]$$

$$P^T = \begin{bmatrix} 1110 \\ 1101 \\ 1011 \end{bmatrix}_{3\times4} \Rightarrow P = \begin{bmatrix} 111 \\ 110 \\ 101 \\ 011 \end{bmatrix}_{4\times3}$$

$$H^T = \begin{bmatrix} 111 \\ 110 \\ 101 \\ 011 \\ \hline 100 \\ 010 \\ 001 \end{bmatrix}_{7\times3} \Rightarrow H^T = \begin{bmatrix} (P^T)^T \\ \hdashline I \end{bmatrix} = \begin{bmatrix} P \\ \hdashline I \end{bmatrix}$$

$$I = \begin{bmatrix} 100 \\ 010 \\ 001 \end{bmatrix}$$

Error matrix

(a)

$$E = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}_{7\times7}$$

Syndrome Vector $S = E \cdot H^T$

For $E = [0\,1\,0\,0\,0\,0\,0] \begin{bmatrix} 111 \\ 110 \\ 101 \\ 011 \\ 100 \\ 010 \\ 001 \end{bmatrix} = \begin{matrix} 0 \oplus 1 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \\ 0 \oplus 1 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \\ 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \end{matrix}$

$= [1\;1\;0]$

Ily

S.No	Bit in error	Bits in Error Vector	Syndrome vector S
1	1st	1 0 0 0 0 0 0	1 1 1
2	2nd	0 1 0 0 0 0 0	1 1 0
3	3rd	0 0 1 0 0 0 0	1 0 1
4	4th	0 0 0 1 0 0 0	0 1 1
5	5th	0 0 0 0 1 0 0	1 0 0
6	6th	0 0 0 0 0 1 0	0 1 0
7	7th	0 0 0 0 0 0 1	0 0 1

$\cong H^T$

(b) For the detection of the error by using syndrome decoding

$$S = YH^T$$

Given $Y = \underset{MSB}{[0}\ 1\ 1\ 0\ 1\ 1\ \underset{LSB}{0]}$

$$\therefore \underset{1\times3}{S} = \begin{bmatrix}0 & 1 & 1 & 0 & 1 & 1 & 0\end{bmatrix}_{1\times7} \begin{bmatrix}1 & 1 & 1\\1 & 1 & 0\\1 & 0 & 1\\0 & 1 & 1\\1 & 0 & 0\\0 & 1 & 0\\0 & 0 & 1\end{bmatrix}_{7\times3}$$

$$S = \begin{bmatrix}1 & 0 & 1\end{bmatrix}_{1\times3}$$

$\therefore \boxed{S = [1\ 0\ 1] \text{ ie } 3^{rd} \text{ bit Error}}$

Error Correction can be possible by using

$$X = Y \oplus E$$

$$= [0110110] \oplus [0010000]$$

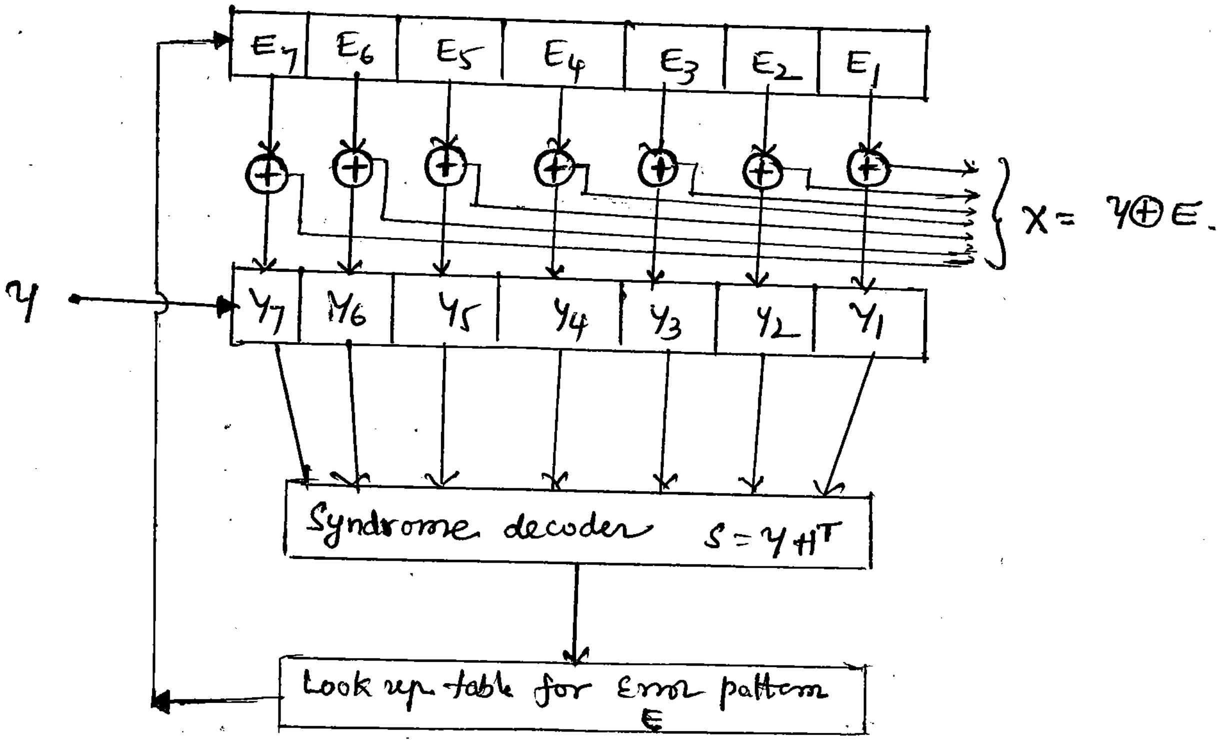
$$X = 0\ 1\ 0\ 0\ 1\ 1\ 0$$

$$\therefore \boxed{X = [\underbrace{0\ 1\ 0\ 0}_{\text{message bits}}\ \underbrace{1\ 1\ 0}_{\text{parity bits}}]}$$

(c) Syndrome diagram:

Error bits → n bits

Received bits → n bits.



2011

* ⑥ Consider (7,4) code whose generator matrix is

$$G = \begin{bmatrix} 1111000 \\ 1010100 \\ 0110010 \\ 1100001 \end{bmatrix}$$

(a) Find all the codewords of the code.

(b) Find parity check matrix H.

(c) Calculate the syndrome for the received vector '1101101'.

Sol. Given Linear block code $(n, k) = (7,4)$

$n = 7$
$k = 4$
$q = n - k = 3$.

Given generator Matrix

$$G = \begin{bmatrix} 111 & \vdots 1000 \\ 101 & \vdots 0100 \\ 011 & \vdots 0010 \\ 110 & \vdots 0001 \end{bmatrix}_{4\times 7}$$

Code Word $C = MG$

where $G = [I \vdots P]_{k\times n}$ (or) $G = [P \vdots I]_{k\times n}$ with P of order $k\times q$ and I of order $k\times k$

$k = 4$

$2^k = 16$ different message blocks.

(a) $C = M[P]$

$$I = \begin{bmatrix} 1000 \\ 0100 \\ 0010 \\ 0001 \end{bmatrix}_{4\times 4}, \quad P = \begin{bmatrix} 111 \\ 101 \\ 011 \\ 110 \end{bmatrix}_{4\times 3}$$

$$= [M_1\ M_2\ M_3\ M_4] \begin{bmatrix} 111 \\ 101 \\ 011 \\ 110 \end{bmatrix}$$

$\Rightarrow C_1\ C_2\ C_3\ C_4 \rightarrow M_1 M_2 M_3 M_4 \Rightarrow$

$C_4 = M_1 + M_2 + M_4$

$C_5 = M_1 + M_3 + M_4$

$C_6 = M_1 + M_2 + M_3$.

Sl No	Message bits M_1 M_2 M_3 M_4	Check bits C_4 C_5 C_6	Complete code word M_1 M_2 M_3 M_4 C_5 C_6 C_7	Weight
1	0 0 0 0	0 0 0	0 0 0 0 0 0 0	0
2	0 0 0 1	1 1 0	0 0 0 1 1 1 0	3
3	0 0 1 0	0 1 1	0 0 1 0 0 1 1	3
4	0 0 1 1	1 0 1	0 0 1 1 1 0 1	4
5	0 1 0 0	1 0 1	0 1 0 0 1 0 1	3
6	0 1 0 1	0 1 1	0 1 0 1 0 1 1	4
7	0 1 1 0	1 1 0	0 1 1 0 1 1 0	4
8	0 1 1 1	0 0 0	0 1 1 1 0 0 0	3
9	1 0 0 0	1 1 1	1 0 0 0 1 1 1	4
10	1 0 0 1	0 0 1	1 0 0 1 0 0 1	3
11	1 0 1 0	1 0 0	1 0 1 0 1 0 0	3
12	1 0 1 1	0 1 0	1 0 1 1 0 1 0	4
13	1 1 0 0	0 1 0	1 1 0 0 0 1 0	3
14	1 1 0 1	1 0 0	1 1 0 1 1 0 0	4
15	1 1 1 0	0 0 1	1 1 1 0 0 0 1	4
16	1 1 1 1	1 1 1	1 1 1 1 1 1 1	7

The minimum distance $\boxed{d_{min} = 3}$ Except '0'.

$d_{min} \geq s+1$

$3 \geq s+1$

$s \leq 2$

∴ No. of detection of errors $\boxed{s=2}$

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

$3 \geq 2t+1$

$t \leq 2$

∴ No of correction of errors $\boxed{t=1}$

(b) Parity check matrix 'H'.

$$H = [\underset{q\times k}{P^T} \vdots \underset{q\times q}{I}]_{q\times n}$$

$$P = \begin{bmatrix} 111 \\ 101 \\ 011 \\ 110 \end{bmatrix} \Rightarrow P^T = \begin{bmatrix} 1101 \\ 1011 \\ 1110 \end{bmatrix}_{3\times 4} \quad I = \begin{bmatrix} 100 \\ 010 \\ 001 \end{bmatrix}_{3\times 3}$$

∴

$$H = \begin{bmatrix} 1101 & 100 \\ 1011 & 010 \\ 1110 & 001 \end{bmatrix} \Rightarrow H^T = \begin{bmatrix} P \\ I \end{bmatrix} = \begin{bmatrix} 111 \\ 101 \\ 011 \\ 110 \\ 100 \\ 010 \\ 001 \end{bmatrix}_{7\times 3}$$

Syndrome vector

$$S = EH^T$$

where

Error Vector

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

$S = [E]_{1\times 7}[H^T]_{7\times 3}$

$S = 1\times 3$.

SNo	Bit in Error	Bit in Error Vector E	Syndrome Vector S = EH^T
1	1st	1 0 0 0 0 0 0	1 1 1
2	2nd	0 1 0 0 0 0 0	1 0 1
3	3rd	0 0 1 0 0 0 0	0 1 1
4	4th	0 0 0 1 0 0 0	1 1 0
5	5th	0 0 0 0 1 0 0	1 0 0
6	6th	0 0 0 0 0 1 0	0 1 0
7	7th	0 0 0 0 0 0 1	0 0 1

For the detection of error by using syndrome decoding is

$S = yH^T$.

Given . Received vector $y = [1101101]$

$$S = [1\ 1\ 0\ 1\ 1\ 0\ 1] \begin{bmatrix} 111 \\ 101 \\ 011 \\ 110 \\ 100 \\ 010 \\ 001 \end{bmatrix} \Rightarrow S = [0\ 0\ 1]$$

∴ $\boxed{S = [0\ 0\ 1]}$ ie 7th bit error

Error Correction is possible by using. $X = Y \oplus E$

$$X = [1101101] \oplus [0000001]$$

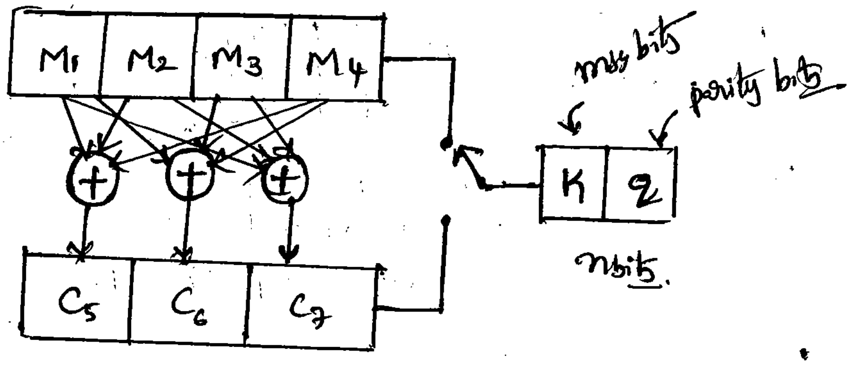
$$X = [\underbrace{1\,1\,0\,1}_{\text{msg bits}}\ \underbrace{1\,0\,0}_{\text{parity bits}}]$$

Encoder diagram:

$C_5 = M_1 \oplus M_2 \oplus M_4$

$C_6 = M_1 \oplus M_3 \oplus M_4$

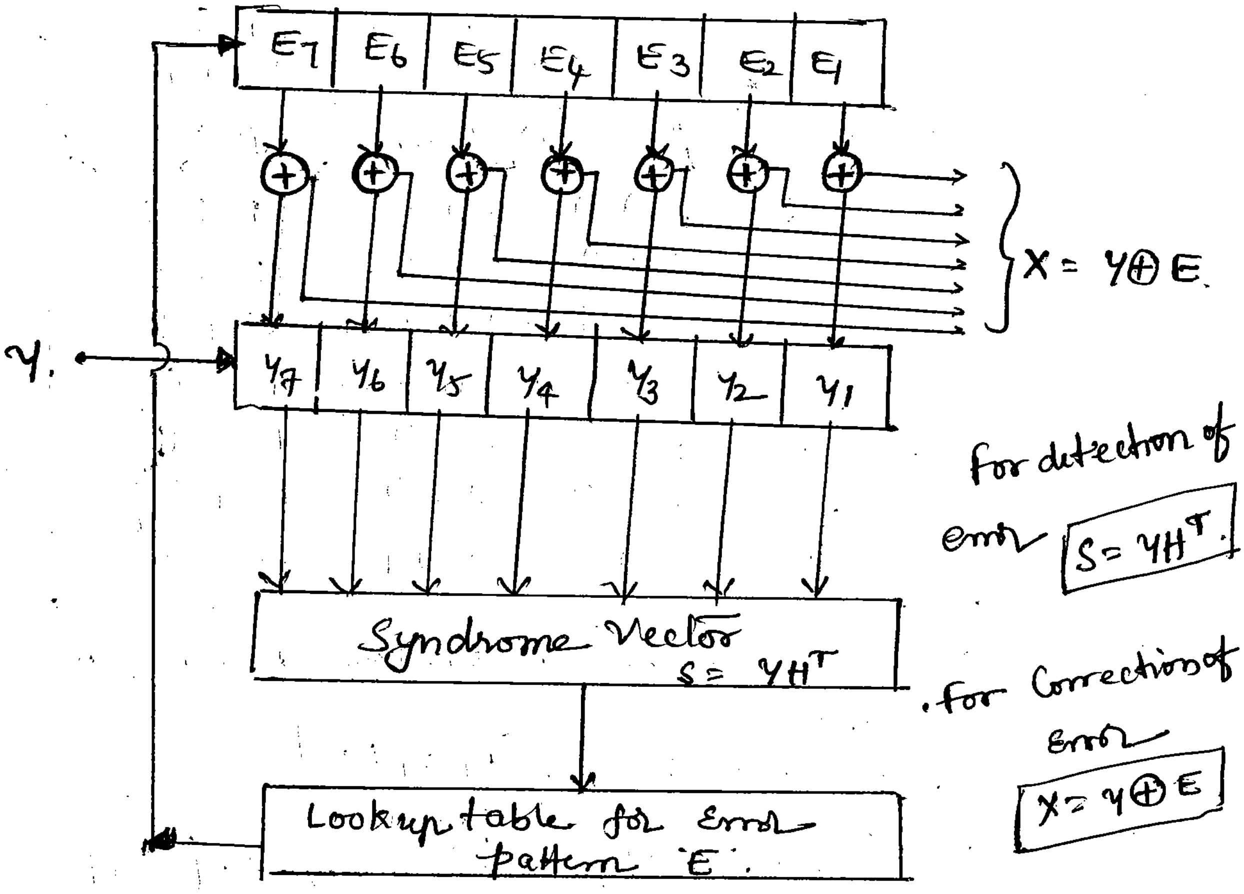
$C_7 = M_1 \oplus M_2 \oplus M_3$



Syndrome diagram:

Error bits - n bits.

Received bits - n bits.



For detection of error $\boxed{S = YH^T}$

For Correction of error $\boxed{X = Y \oplus E}$

=*=

Cyclic Codes:

(Polynomial form)

In linear block codes the design of hardware requirements are complex, so the alternative method to generate code vector is "cyclic codes".

* A linear code is called cyclic code, if every cyclic shift of the code vector is produced another code vector.

Ex: Consider an n-bit code vector

$$[X] = [\underset{MSB}{x_{n-1}}\ x_{n-2} \cdots\cdots x_1\ \underset{LSB}{x_0}].$$

Shift the above code vector cyclicly produced another Code vector.

$$[X'] = [\overset{MSB}{x_{n-2}}\ x_{n-3} \cdots\cdots x_1\ x_0\ \overset{LSB}{x_{n-1}}]$$

Actually x_{n-1} is MSB position but after cyclic shift it becomes LSB position.

A binary code is said to be a Cyclic Codes if exhibits two fundamental properties.

① Linear property : The sum of two code words is also a code word.

② Cyclic property : Any cyclic shift of a codeword is also a codeword.

Ex:

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

1 0 1 1

obtained by a cyclic shift of n-tuple (tuple) n=4

Cyclic codes are important for two reasons.

① Encoding & syndrome ~~de~~ calculations can be easily implemented by using simple shift registers with feedback connection.

② The matematical structure of these codes is such that it is possible to design codes having useful error-correcting properties.

Analysis : The code vector is represented in mathematical form of polynomials.

✓ $$X(p) = \left[\overset{MSB}{x_{n-1}} p^{n-1} + x_{n-2} p^{n-2} + \cdots\cdots + x_1 p^1 + \overset{LSB}{x_0} \right] \rightarrow ①$$

where p- arbitrary value

power of p represents the position of the codeword bit.

✓ Multiply eqn ① with 'p' we get

$$P.X(p) = x_{n-1} P^n + x_{n-2} p^{n-1} + \cdots\cdots + x_1 p^2 + x_0 . P^1 \longrightarrow ②$$

✓ Cyclic shift of eqn ② produced another code vector

$$X^1(p) = x_{n-2} p^{n-1} + x_{n-3} p^{n-2} + \cdots + x_1 p^2 + x_0 p^1 + x_{n-1} P^0 \hookrightarrow 1 \longrightarrow ③$$

✓ Add eqn ② & ③ by using modulo 2 operation ie XOR operation.

$$PX(p) \oplus X^1(p) = x_{n-1} p^n \oplus x_{n-1} + x_{n-2} p^{n-1} \oplus x_{n-2} p^{n-1} + \cdots\cdots + x_1 P^2 \oplus x_1 p^2 + x_0 P^1 \oplus x_0 P^1$$

$$= x_{n-1} P^n \oplus x_{n-1} + [x_{n-2} \oplus x_{n-2}] P^{n-1} + [x_{n-3} \oplus x_{n-3}] p^{n-2} + \cdots + p^2 (x_1 \oplus x_1) + p^1 (x_0 \oplus x_0)$$

$$= x_{n-1} [P^n \oplus 1] + 0 + 0 + \cdots 0 + 0$$

$$PX(P) \oplus X^1(P) = x_{n-1} [p^n \oplus 1]$$

$$\Rightarrow X^1(P) = P.[X(P)] \oplus x_{n-1} [P^n \oplus 1]$$

$$\boxed{X^1(P) = P\,X(P) \oplus x_{n-1} [P^n \oplus 1]}$$

The polynomial '$p^n \oplus 1$' and its factors plays major role in cyclic codes.

Cyclic Codes → non systematic Cyclic codes ;

Cyclic Codes → Systematic Cyclic codes

(a) Non Systematic :

Cyclic Code $\boxed{X(p) = M(p) \cdot G(p)}$ $\longrightarrow$ ①

where M(P) - Message signal polynomial of degree 'K'.

$$\boxed{M(P) = M_{K-1}P^{K-1} + M_{K-2}P^{K-2} + \cdots M_2P^2 + M_1P + M_0}$$

G(p) - Generating polynomial of degree 'q'.

$$\boxed{G(P) = P^q + g_{q-1}P^{q-1} + \cdots g_2P^2 + g_1P^1 + 1}$$

C(P) - Cyclic code polynomial

$$\boxed{C(P) = C_{q-1}P^{q-1} + C_{q-2}P^{q-2} + \cdots C_2P^2 + C_1P^1 + C_0}$$

(b) Systematic Cyclic Code :

cyclic code $\boxed{X(P) = P^q \cdot M(P) + C(P)}$ $\longrightarrow$ ②

Equating eqn ① & ②

$$M(P) \cdot G(P) = P^q M(P) + C(P)$$

Divide above eqn by G(P).

$$\therefore M(P) = \frac{P^q M(P)}{G(P)} + \frac{C(P)}{G(P)}$$

∴ Systamatic

$$\boxed{\frac{P^q M(P)}{G(P)} = M(P) \underset{\substack{\uparrow \\ \text{XOR}}}{\oplus} \frac{C(P)}{G(P)}}$$

C(P)

This equation has the form of $\frac{N}{D} = Q + \frac{R}{D}$

where N- numarator
D- Denominator
Q- Quiscent
R- Reminder

Thus the check bit polynomial is obtained as reminder 'R' is dividing $\frac{P^q . M(P)}{G(P)}$. ie $\boxed{\frac{N}{D} = Q + \frac{R}{D}}$

Encoder (or) Encoder diagram for Cyclic Codes :

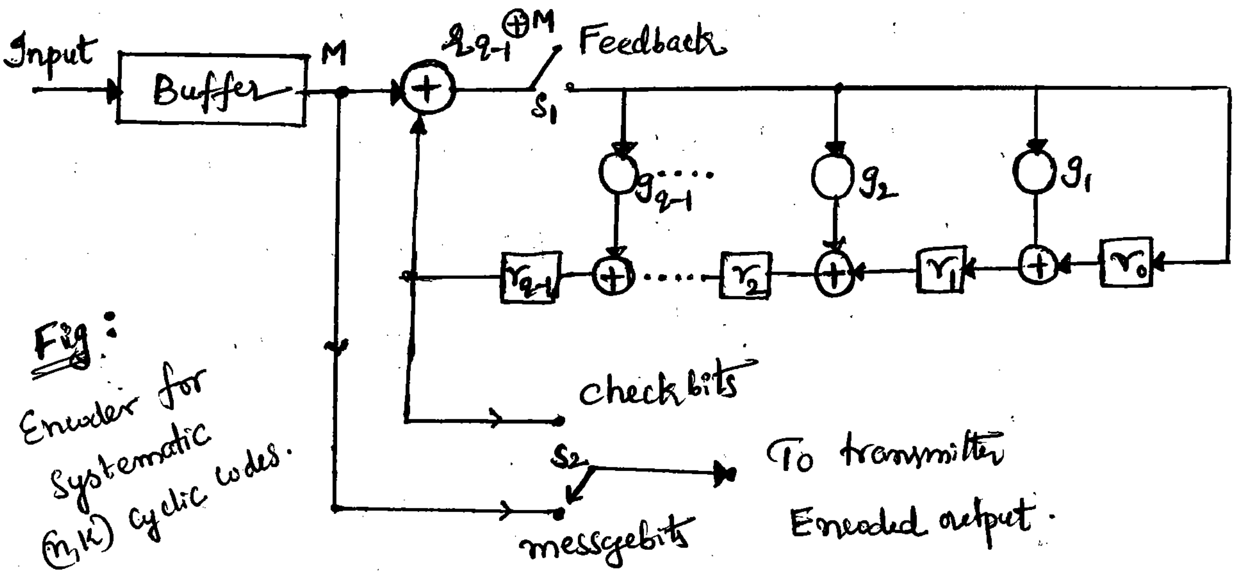


Fig: Encoder for Systematic (n,k) cyclic codes.

where

$r_0, r_1, r_2, \cdots r_{q-1}$ represents flipflops (or) registers they are connected in sequential order to make a shift register.

$g_1, g_2, \cdots g_{q-1}$ represents path establishment.

- ✓ Encoding starts with the feedback switch 'S_1' closed, the output switch 'S_2' is connected to the message bits.
- ✓ All the shift registers are initiated to all zero state.
- ✓ K message bits are shifted to the transmitter as well as shifted into the registers.
- ✓ After the shift of K-message bits, the register contains 'q' check bits.
- ✓ The feedback switch S_1 is now opened and output switch is connected to check bit position.

Thus we can obtain all codewords as cyclic code.

Syndrome decoding for Cyclic Codes:

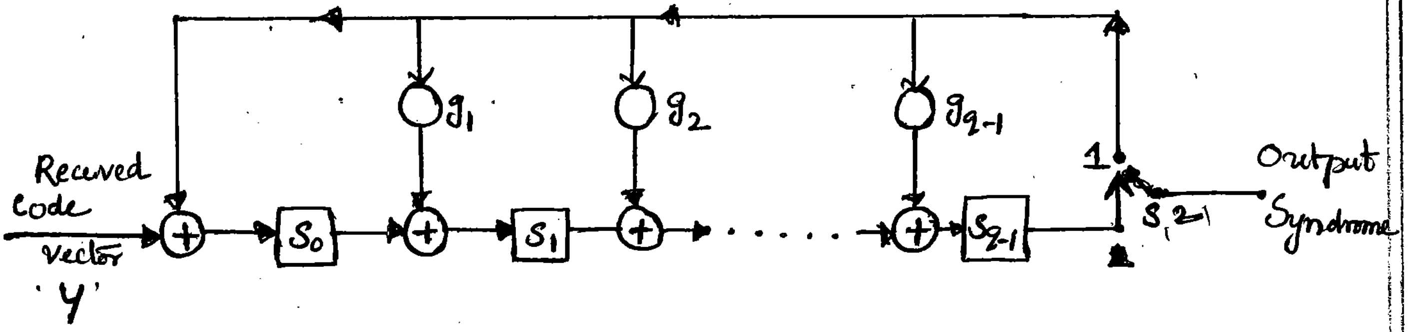


Fig: Syndrome decoding for cyclic codes.

- ✓ Initially all the shift registers are zeros and switch is connected to position ①.
- ✓ The received vector 'y' is shifted bit by bit into the shift registers.
- ✓ After all the bits of 'y' are shifted, 'q' flip flops of shift registers contains 'q' bit syndrome vector.
- ✓ The switch is then closed to position ② & clock (For reset) are applied to the shift registers.
- ✓ The output ~~of~~ is a syndrome vector.

$$S = (S_{q-1}, S_{q-2}, \cdots S_1 \, S_0)$$

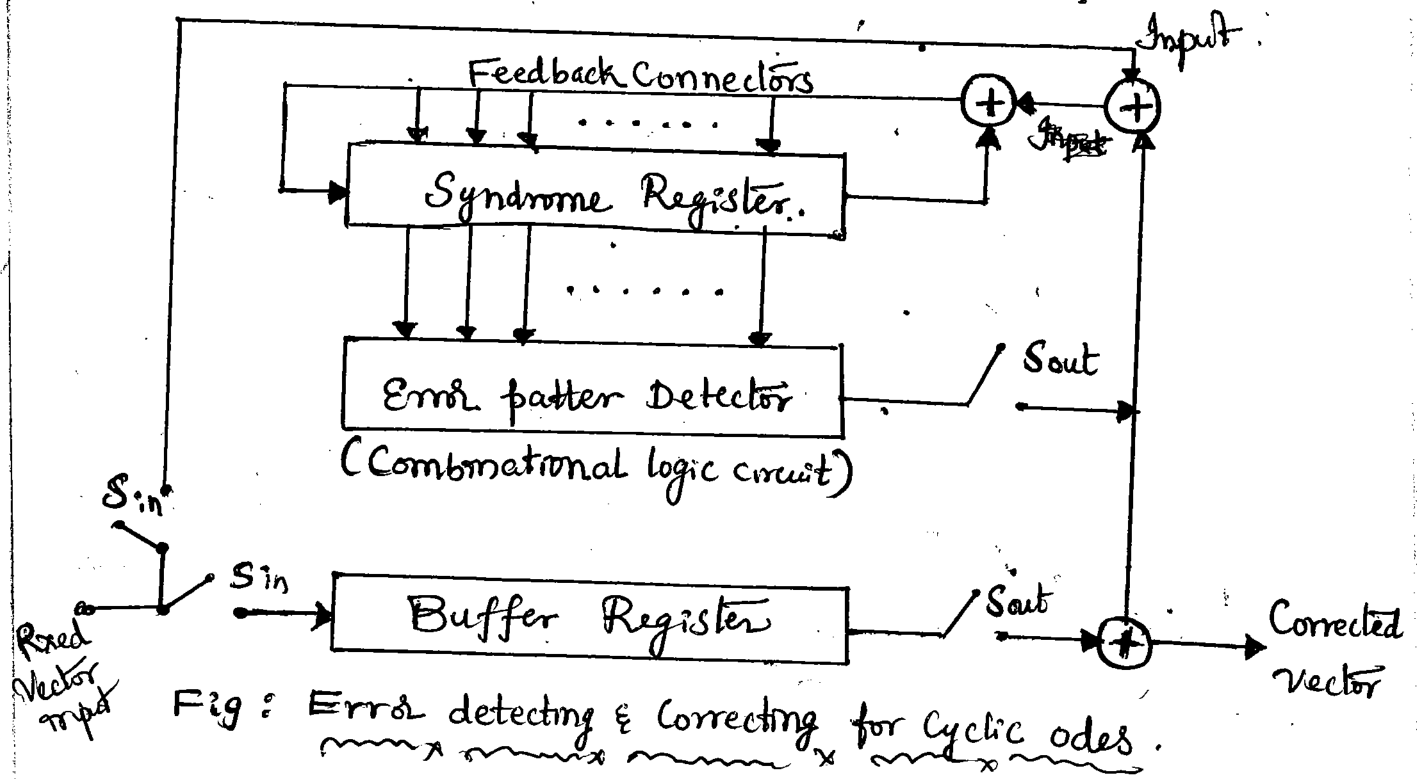


Fig: Error detecting & Correcting for Cyclic odes.

✓ Once the syndrome is calculated, then the error pattern is detected for that particular syndrome vector.

✓ When this error vector is added to the received vector 'Y' then it gives corrected code vector at the output. $\boxed{X = Y \oplus E}$

✓ The switches 'Sin' is closed then the bits of the received vector 'Y' are shifted into the buffer register as well as they are shifted into the syndrome register.

✓ The output of the syndrome register is given to the error pattern detector.

✓ A particular syndrome detects a specific error pattern.

✓ The switches 'Sin' is opened, Sout is closed.
The error pattern is then added bit by bit with the received vector.

* Syndrome decoding can be used to correct the errors.

Let 'Y' represents the received code vector.

The error can be detected as $\boxed{Y = X \oplus E.}$

The error can be corrected as $\boxed{X = Y \oplus E.}$

In the polynomial form $Y(P) = X(P) + E(P)$

We know $X(P) = M(P) \cdot G(P)$

$$\therefore \boxed{Y(P) = M(P) \cdot G(P) + E(P)}$$

Divide with G(P) then

$$\frac{Y(P)}{G(P)} = M(P) + \frac{E(P)}{G(P)}$$

$$\frac{\overset{\text{numerator} \leftarrow}{N}}{\underset{\text{Denominator} \leftarrow}{D}} = \underset{\text{Quisent}}{Q} + \frac{R^{\curvearrowright \text{Remainder}}}{D_{\hookrightarrow \text{Denominator}}}$$

The syndrome vector is obtained by dividing the received vector Y(P) with G(P).

∴ Syndrome polynomial $\boxed{S(p) = \text{Rem}\left(\frac{Y(P)}{G(P)}\right)}$

Advantages of Cyclic Codes:

✓ The error correcting and detecting methods of cyclic codes are simple and easy to implement.

✓ It is very simple compared to linear block codes.

Disadvantage of cyclic code:

✓ Error detection in cyclic codes is simple but the error correction is complecated in cyclic codes.

Problem:

① A (7,4) cyclic code generated by $G(P) = P^3+P+1$.

(a) Find all the code vectors for the code in non systematic code.

(b) Find all the code vectors for the code in systematic code.

(c) Design the encoder for given cyclic code and verify its operation for any message vector.

(d) Design a syndrome calculator for a given cyclic code and calculate the syndrome for $Y = 1001101$

(e) Find out the generator matrix by using of cyclic codes.

(f) Find the generator polynomial for the given cyclic code.

Solution:

Sol. Given $G(P) = P^3+P+1$

(7,4) Cyclic Code $\therefore$ $n=7$
$K=4$

Parity bits $q = n-k = 7-4 = 3$.

Since $K=4$, there will be $2^K = 2^4 = 16$ different message code vectors.

Each message vector consists of 4 bits.

For $\underset{M_3\;M_2\;M_1\;M_0}{0\,0\,0\,1} \rightarrow M(P) = M_3P^3 + M_2P^2 + M_1P^1 + M_0 \cdot P^0$

$= 0 \cdot P^3 + 0 \cdot P^2 + 0 \cdot P^1 + 1 \cdot 1$

$= 0+1$

$= 1$ $\therefore \boxed{M(P) = 1}$

Ily $0010 \rightarrow P$, $1001 \rightarrow P^3+1$ ie.

SL No	Message bits M3 M2 M1 M0	Message Polynomial M(P).
1	0 0 0 0	0
2	0 0 0 1	1
3	0 0 1 0	P
4	0 0 1 1	$P+1$
5	0 1 0 0	P^2
6	0 1 0 1	P^2+1
7	0 1 1 0	P^2+P
8	0 1 1 1	P^2+P+1
9	1 0 0 0	P^3
10	1 0 0 1	P^3+1
11	1 0 1 0	P^3+P
12	1 0 1 1	P^3+P+1
13	1 1 0 0	P^3+P^2
14	1 1 0 1	P^3+P^2+1
15	1 1 1 0	P^3+P^2+P
16	1 1 1 1	P^3+P^2+P+1

(a)

NonSystematic Code: $\boxed{X(P) = M(P)\cdot G(P)}$ $G(p) = P^3+P+1$

For ① (0000) → $M(P) = 0$ ∴ $X_1(P) = 0\cdot(P^3+P+1) = (0000000)$

For ② (0001) → $M(P) = 1$. ∴ $X(P) = 1\cdot(P^3+P+1) = P^3+P+1$

∴ $X_2(P) = (0001011)$

For ③ (0010) → $M(P) = P$ $X(P) = P(P^3+P+1) = P^4+P^2+P$.

$= (0010110)$

For ④ (0011) $M(P) = P+1$ $X(P) = (P+1)(P^3+P+1)$

$= P^4+P^2+P+P^3+P+1$

$= P^4+P^3+P^2+\underbrace{P\oplus P}_{0}+1$

$= (0011101)$

For ⑤ (0100) $M(P) = P^2$. $X(P) = P^2(P^3+P+1)$

$= P^5+P^3+P^2 = (0101100)$

For ⑥ (0101) $M(P) = P^2+1$ $X(P) = (P^2+1)(P^3+P+1)$

$= P^5+P^3+P^2+P^3+P+1$

$= P^5+P^3\oplus P^3+P^2+P+1$

$= P^5+P^2+P+1$

$= (0100111)$

For ⑦ (0110) → $M(p) = p^2+p$ ∴ $X(p) = (p^2+p)(p^3+p+1)$

$= p^5+p^3+\cancel{p^2}+p^4+\cancel{p^2}+p$

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

$= \underline{(0\ 1\ 1\ 1\ 0\ 1\ 0)}$

→ $\underline{\underline{0111}}$ at last.

for ⑨ (1000) → $M(p) = p^3$

∴ $X(p) = p^3\,(p^3+p+1)$

$= p^6+p^4+p^3 = \underline{(1\,0\,1\,1\,0\,0\,0)}$

for ⑩ (1001) → $M(p) = p^3+1$

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

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

$= p^6+p^4+p+1 = \underline{(1\,0\,1\,0\,0\,1\,1)}$

for ⑪ (1010) → $M(p) = p^3+p$

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

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

$= p^6+p^3+p^2+p = \underline{(1\,0\,0\,1\,1\,1\,0)}$

for ⑫ (1011) → $M(p) = p^3+p+1$

∴ $X(p) = (p^3+p+1)(p^3+p+1)$

$= p^6+\cancel{p^4}+\cancel{p^3}+\cancel{p^4}+p^2+\cancel{p}+\cancel{p^3}+\cancel{p}+1$

$= p^6+p^2+1 = \underline{(1\,0\,0\,0\,1\,0\,1)}$

for ⑬ (1100) → $M(p) = p^3+p^2$ ∴ $X(p) = (p^3+p^2)(p^3+p+1)$

$= p^6+p^4+\cancel{p^3}+p^5+\cancel{p^3}+p^2$

$= (p^6+p^5+p^4+p^2) = \underline{(1\,1\,1\,0\,1\,0\,0)}$

for ⑭ (1101) $M(p) = p^3+p^2+1$ ∴ $X(p) = (p^3+p^2+1)(p^3+p+1)$

$= p^6+p^4+p^3+p^5+\cancel{p^3}+p^2+p^3+p+1$

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

$= \underline{(1\ 1\ 1\ 1\ 1\ 1\ 1)}$

for ⑮ (1110) $M(p) = p^3+p^2+p$ ∴ $X(p) = (p^3+p^2+p)(p^3+p+1)$

$= p^6+\cancel{p^4}+\cancel{p^3}+p^5+\cancel{p^3}+\cancel{p^2}+\cancel{p^4}+\cancel{p^2}+p$

$= p^6+p^5+p = \underline{(1\,1\,0\,0\,0\,1\,0)}$

for ⑯ (1111) $M(p) = p^3+p^2+p+1$ ∴ $X(p) = (p^3+p^2+p+1)(p^3+p+1)$

$= p^6+\cancel{p^4}+\cancel{p^3}+p^5+\cancel{p^3}+\cancel{p^2}+\cancel{p^4}+\cancel{p^2}+\cancel{p}+p^3+\cancel{p}+1$

$= p^6+p^5+p^3+1 = \underline{(1\,1\,0\,1\,0\,0\,1)}$

→ ⑧ (0111) $M(p) = p^2+p+1$ $X(p) = (p^2+p+1)(p^3+p+1)$

$= p^5+\cancel{p^3}+\cancel{p^2}+p^4+\cancel{p^2}+\cancel{p}+\cancel{p^3}+\cancel{p}+1$

$= (p^5+p^4+1) = \underline{(0\,1\,1\,0\,0\,0\,1)}$

SL NO	Message bits $M_3 M_2 M_1 M_0$	Non Systematic Cyclic code Vectors	Systematic Cyclic Code Vectors.
1	0000	0000000	0000000
2	0001	0001011	0001011
3	0010	0010110	0010110
4	0011	0011101	0011101
5	0100	0101100	0100111
6	0101	0100111	0101100 ✓
7	0110	0111010	0110001
8	0111	0110001	0111010
9	1000	1011000	1000101
10	1001	1010011	1001110
11	1010	1001110	1010011
12	1011	1000101	1011000
13	1100	1110100	1100010
14	1101	1111111	1101001
15	1110	1100010	1110100
16	1111	1101001	1111111

(b)

Systematic Code : message $\boxed{K \mid q}$ parity.

The check bits can be generated by using

$$C(P) = \text{Rem}\left[P^q \frac{M(P)}{G(P)} \right] \qquad q = n-k = 3.$$

1×3

For ① $(0000) \rightarrow M(P) = 0 \Rightarrow P^3(0) = 0$, $P^q \frac{(M(P))}{G(P)} \Rightarrow C(p) = (0\ 0\ 0)$

$c_2\ c_1\ c_0$

$\therefore X = (M_3 M_2 M_1 M_0\ c_2\ c_1\ c_0)$

$X = (0000\ 000)$

For ② $(0001) \rightarrow M(p) = 1$, $P^3(M(P)) = P^3$

$$C(p) = \frac{P^3 M(P)}{G(P)} = \frac{P^3}{P^3+P+1} = 011$$

$$P^3+P+1 \overline{\smash{\big)}\ P^3\ }$$ quotient 1

$\cancel{P^3} + P + 1$

$P+1 \Rightarrow 011$

$\therefore X(P) = (0001\ 011)$

For ③ $(0010) \rightarrow M(P) = P \rightarrow P^3(M(P)) = P^4$.

$$\frac{P^3 M(P)}{G(P)} = \frac{P^4}{P^3+P+1} = 110$$

$$P^3+P+1 \overline{\smash{\big)}\ P^4\ }$$ quotient P

$\cancel{P^4} + P^2 + P$

$P^2 + P \rightarrow (1\ 1\ 0)$

$\therefore X(P) = (0010\ 110)$.

for ④ (0011), $M(p) = p+1$, $p^3(M(p)) = p^3(p+1) = p^4+p^3$.

$$\frac{p^3 M(p)}{G(p)} = \frac{p^4+p^3}{p^3+p+1} = (101)$$

$$p^3+p+1 \overline{) \cancel{p^4}+p^3} \quad p+1$$
$$\cancel{p^4}+p^2+p$$
$$\cancel{p^3}+p^2+\cancel{p}$$
$$\cancel{p^3}+\cancel{p}+1$$
$$(101) \leftarrow \underline{p^2+1}.$$

$\therefore X(p) = (0011\ 101)$

for ⑤ (0100), $M(p) = p^2$, $p^3 M(p) = p^3 \cdot p^2 = p^5$

$$\frac{p^3 M(p)}{G(p)} = \frac{p^5}{p^3+p+1} = (111)$$

$$p^3+p+1 \overline{) \cancel{p^5}} \quad p^2+1$$
$$\cancel{p^5}+p^3+p^2$$
$$\cancel{p^3}+p^2$$
$$\cancel{p^3}+p+1$$
$$(111) \leftarrow \underline{p^2+p+1}$$

$X(p) = (0100\ 111)$

for ⑥ (0101), $M(p) = p^2+1$, $p^3(M(p)) = p^3(p^2+1)$
$= p^5+p^3$.

$$\frac{p^3 M(p)}{G(p)} = \frac{p^5+p^3}{p^3+p+1} = (100)$$

$$p^3+p+1 \overline{) \cancel{p^5}+\cancel{p^3}} \quad p^2$$
$$\cancel{p^5}+\cancel{p^3}+p^2$$
$$(100) \leftarrow \underline{p^2}$$

$\therefore X(p) = (0101\ 100)$

for ⑦ (0110), $M(p) = p^2+p$, $p^3 M(p) = p^3(p^2+p)$
$= p^5+p^4$

$$\frac{p^3 M(p)}{G(p)} = \frac{p^5+p^4}{p^3+p+1} = (001)$$

$$p^3+p+1 \overline{) \cancel{p^5}+p^4} \quad p^2+p+1$$
$$\cancel{p^5}+p^3+p^2$$
$$\cancel{p^4}+p^3+\cancel{p^2}$$
$$\cancel{p^4}+\cancel{p^2}+p$$
$$\cancel{p^3}+\cancel{p}$$
$$p^3+\cancel{p}+1$$
$$(001) \leftarrow 1$$

$\therefore X(p) = (0110\ 001)$

for ⑧ (0111), $M(p) = p^2+p+1$, $p^3 M(p) = p^3(p^2+p+1)$
$= p^5+p^4+p^3$

$$\frac{p^3 M(p)}{G(p)} = \frac{p^5+p^4+p^3}{p^3+p+1} = (010)$$

$$p^3+p+1 \overline{) \cancel{p^5}+p^4+\cancel{p^3}} \quad p^2+p$$
$$\cancel{p^5}+\ +\cancel{p^3}+p^2$$
$$\cancel{p^4}+\cancel{p^2}$$
$$\cancel{p^4}+\cancel{p^2}+p$$
$$(010) \leftarrow \underline{p}$$

$\therefore X(p) = (0111010)$

for ⑨ (1000), $M(p) = p^3$, $p^3 M(p) = p^3 \cdot p^3 = p^6$

$$\frac{p^3 M(p)}{G(p)} = \frac{p^6}{p^3+p+1} = (101)$$

$$p^3+p+1 \overline{) \cancel{p^6}} \quad p^3+p+1$$
$$\cancel{p^6}+p^4+p^3$$
$$\cancel{p^4}+p^3$$
$$\cancel{p^4}+\ p^2+p$$
$$\cancel{p^3}+p^2+\cancel{p}$$
$$\cancel{p^3}+\cancel{p}+1$$
$$(101) \leftarrow \underline{p^2+1}$$

$\therefore X(p) = (1000\ 101)$

for ⑩ (1001), $M(p) = p^3+1$, $p^3 M(p) = p^3(p^3+1)$
$= p^6+p^3$

$$\frac{p^3 M(p)}{G(p)} = \frac{p^6+p^3}{p^3+p+1} = (110)$$

$$p^3+p+1 \overline{) \cancel{p^6}+\cancel{p^3}} \quad p^3+p$$
$$\cancel{p^6}+p^4+\cancel{p^3}$$
$$\cancel{p^4}$$
$$\cancel{p^4}+p^2+p$$
$$(110) \leftarrow \underline{p^2+p}$$

$\therefore X(p) = (1001\ 110)$

⑪ For (1010) $M(p) = p^3+p$ ∴ $p^3M(p) = p^3(p^3+p) = p^6+p^4$

$$\frac{p^3M(p)}{G(p)} = \frac{p^6+p^4}{p^3+p+1} = (011)$$

$$\begin{array}{r} p^3+1 \\ p^3+p+1 \overline{) \cancel{p^6}+\cancel{p^4}} \\ p^6+p^4+p^3 \\ \hline p^3 \\ \cancel{p^3}+p+1 \\ \hline (011) \leftarrow p+1 \end{array}$$

∴ $X(p) = (1010011)$

⑫ For (1011), $M(p) = p^3+p+1$, $p^3M(p) = p^3(p^3+p+1)$
$= p^6+p^4+p^3$.

$$\therefore \frac{p^3M(p)}{G(p)} = \frac{p^6+p^4+p^3}{p^3+p+1} = (000)$$

$$\begin{array}{r} p^3 \\ p^3+p+1 \overline{) \cancel{p^6+p^4+p^3}} \\ p^6+p^4+p^3 \\ \hline (000) \leftarrow 0 \end{array}$$

∴ $X(p) = (1011000)$

⑬ For (1100), $M(p) = p^3+p^2$, $p^3M(p) = p^3(p^3+p^2)$
$= p^6+p^5$.

$$\therefore \frac{p^3M(p)}{G(p)} = \frac{p^6+p^5}{p^3+p+1} = (010)$$

$$\begin{array}{r} p^3+p^2+p \\ p^3+p+1 \overline{) \cancel{p^6}+p^5} \\ p^6+p^4+p^3 \\ \hline \cancel{p^5}+p^4+\cancel{p^3} \\ \cancel{p^5}+p^3+p^2 \\ \hline \cancel{p^4}+\cancel{p^2} \\ p^4+p^2+p \\ \hline (010) \leftarrow p \end{array}$$

∴ $X(p) = (1100010)$.

⑭ For (1101), $M(p) = p^3+p^2+1$, $p^3M(p) = p^3(p^3+p^2+1)$
$= p^6+p^5+p^3$

$$\frac{p^3M(p)}{G(p)} = \frac{p^6+p^5+p^3}{p^3+p+1} = (001)$$

$$\begin{array}{r} p^3+p^2+p+1 \\ p^3+p+1 \overline{) \cancel{p^6}+p^5+\cancel{p^3}} \\ p^6+p^4+p^3 \\ \hline \cancel{p^5}+p^4 \\ p^5+p^3+p^2 \\ \hline \cancel{p^4}+p^3+\cancel{p^2} \\ p^4+p^2+p \\ \hline \cancel{p^3}+\cancel{p} \\ p^3+p+1 \\ \hline (001) \leftarrow 1 \end{array}$$

∴ $X(p) = (1101001)$

⑮ For (1110), $M(p) = p^3+p^2+p$
$p^3M(p) = p^3(p^3+p^2+p)$
$= p^6+p^5+p^4$

$$\frac{p^3M(p)}{G(p)} = \frac{p^6+p^5+p^4}{p^3+p+1} = (100)$$

$$\therefore \begin{array}{r} p^3+p^2 \\ p^3+p+1 \overline{) \cancel{p^6}+p^5+\cancel{p^4}} \\ p^6+p^4+p^3 \\ \hline \cancel{p^5}+\cancel{p^3} \\ p^5+p^3+p^2 \\ \hline (100) \leftarrow p^2 \end{array}$$

∴ $X(p) = (1110100)$

⑯ For (1111) $M(p) = p^3+p^2+p+1$
$p^3M(p) = p^3(p^3+p^2+p+1)$
$= p^6+p^5+p^4+p^3$

$$\frac{p^3M(p)}{G(p)} = \frac{p^6+p^5+p^4+p^3}{p^3+p+1} = (111)$$

$$\therefore \begin{array}{r} p^3+p^2+1 \\ p^3+p+1 \overline{) \cancel{p^6}+p^5+\cancel{p^4}+\cancel{p^3}} \\ p^6+p^4+p^3 \\ \hline \cancel{p^5} \\ p^5+p^3+p^2 \\ \hline \cancel{p^3}+p^2 \\ p^3+p+1 \\ \hline (111) \leftarrow p^2+p+1 \end{array}$$

∴ $X(p) = (1111111)$.

(C) The given generator polynomial is $G(p) = p^3 + p + 1$

$$\therefore \; G(p) = p^q + g_{q-1}p^{q-1} + g_{q-2}p^{q-2} + \cdots\cdots + g_2p^2 + g_1p^1 + 1.$$

$\underline{q=3}$ $\quad \therefore G(p) = p^3 + g_2p^2 + g_1p^1 + 1.$

Compare with $G(p) = p^3 + p + 1$ (given)

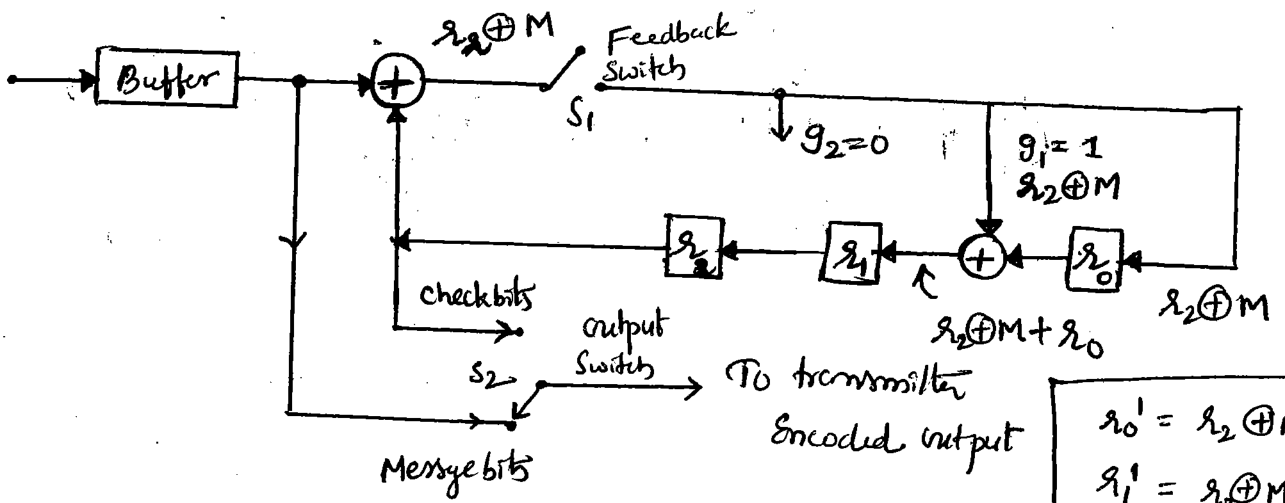
$\therefore$ $g_2 = 0$ (Not existing path).
$g_1 = 1$ (Existing path).

Since $q=3$. there are 3 flip flops in shift registers.

$q = n-K$
$= 7-4$
$= 3$

Since $g_2 = 0$ its link is not connected and
$g_1 = 1$ its link is connected.

Encoder diagram :



$r_0' = r_2 \oplus M$
$r_1' = r_2 \oplus M + r_0$
$r_2' = r_1.$

- After cyclic shift of 4 times $(K=4)$ the register hold the check bits.

For Message (0101)

(a) Input Message bits	(b) Register bit input before shift			(c) Register bit output after shift.		
M	r_2	r_1	r_0	$r_2' = r_1$	$r_1' = r_0 \oplus r_2 \oplus M$	$r_0' = r_2 \oplus M$
–	0	0	0	0	0	0
0	0	0	0	0	0	0
1	0	0	0	0	1	1
0	0	1	1	1	1	0
1	1	1	0	1	0	0

$$\therefore \; X = [M_3\, M_2\, M_1\, M_0\, C_2\, C_1\, C_0] \Rightarrow X = [\underbrace{0\,1\,0\,1}_{\text{msg bits}}\;\underbrace{1\,0\,0}_{\text{Check bits}}]$$

~~Hence~~ Verified

(d) **Syndrome decoder :** For the given code $n = 7$, $K = 4$, $q = n - k = 3$.

Method ①: The given polynomial of the generator $G(p) = p^3 + p + 1$

$$G(p) = p^q + g_{q-1}p^{q-1} + g_{q-2}p^{q-2} + \dots + g_1 p + 1 .$$

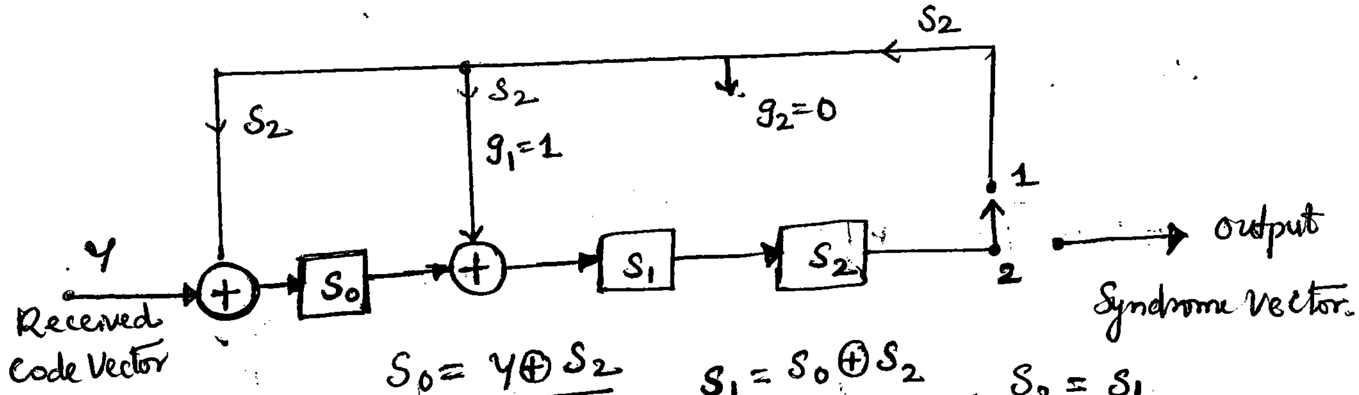
$q = 3$ $\quad G(p) = p^3 + g_2 p^2 + g_1 p^1 + 1 .$

Compare with $G(p) = p^3 + p + 1$

$\therefore g_2 = 0$ (No path)

$g_1 = 1$ (path established).

$Y = [1001101]$



$S_0 = Y \oplus S_2$ $\quad S_1 = S_0 \oplus S_2$ $\quad S_2 = S_1$

Shift	Received Vector ie bits in 'Y'	Content of flipflop in shift registers		
		$S_0 = Y + S_2$	$S_1 = S_0 \oplus S_2$	$S_2 = S_1$
-	-	0	0	0
1	1	1	0	0
2	0	0	1	0
3	0	0	0	1
4	1	0	1	0
5	1	1	0	1
6	0	1	0	0
7	1	1	1	0
		S_0	S_1	S_2

(previous bits operation)

The end of the last shift registers contents are. $S_0 S_1 S_2 = 110$

∴ Hence the Calculated Syndrome $\boxed{S = (S_2 S_1 S_0) = 011}$

Method ② : Syndrome vector is obtained by dividing received vector $y(p)$ by $G(p)$.

$$\therefore \quad S(p) = \text{Rem}\left[\frac{y(p)}{G(p)}\right]$$

Given Rxed Vector $y = [1 \;\; 0 \;\; 0 \;\; 1 \;\; 1 \;\; 0 \;\; 1]$

$$\therefore \; y(p) = 1\cdot p^6 + 0\cdot p^5 + 0\cdot p^4 + 1\cdot p^3 + 1\cdot p^2 + 0\cdot p^1 + 1$$

$$\therefore \; y(p) = p^6 + p^3 + p^2 + 1$$

Generator polynomial $G(p) = p^3 + p + 1$.

$$\frac{y(p)}{G(p)} \Rightarrow \frac{p^6 + p^3 + p^2 + 1}{p^3 + p + 1} \quad \therefore$$

$$\begin{array}{r|l} & p^3 + p \\ \hline p^3 + p + 1 & \cancel{p^6} + \cancel{p^3} + p^2 + 1 \\ & \cancel{p^6} + p^4 + \cancel{p^3} \\ \hline & \cancel{p^4} + \cancel{p^2} + 1 \\ & p^4 + p^2 + p \\ \hline & p + 1 \end{array}$$

$\text{Rem}: 011 \leftarrow p+1$

$\therefore$ Syndrome vector

$$\boxed{S = 0\,1\,1}$$

$S_2 \; S_1 \; S_0$.

(e) Generator Matrix by using Cyclic Codes:

Given $n = 7$, $K = 4$, $q = n - k = 7 - 4 \Rightarrow q = 3$.

$$G(p) = p^3 + p + 1 \Rightarrow 1\cdot p^3 + 0\cdot p^2 + 1\cdot p + 1.$$

multiply with 'p^i' on both sides ie

$$p^i \cdot G(p) = p^i \cdot p^3 + p^i \cdot p + p^i \cdot 1$$

$$= p^{3+i} + p^{1+i} + p^i$$

Since $K=4$, the no. of rows of generated matrix is given as $K-1 = 3$, $i = 3, 2, 1, 0$.

For row 1, $i=3$, $p^3 \cdot G(p) = p^6 + p^4 + p^3 \rightarrow 1\,0\,1\,1\,0\,0\,0$

For row 2, $i=2$, $p^2 \cdot G(p) = p^5 + p^3 + p^2 \Rightarrow 0\,1\,0\,1\,1\,0\,0$

For row 3, $i=1$, $p^1 \cdot G(p) = p^4 + p^2 + p \Rightarrow 0\,0\,1\,0\,1\,1\,0$

For row 4, $i=0$, $p^0 \cdot G(p) = p^3 + p + 1 \Rightarrow 0\,0\,0\,1\,0\,1\,1$

∴ Generator Matrix

$$G_{K\times n} = \begin{array}{c} \begin{array}{ccccccc} p^6 & p^5 & p^4 & p^3 & p^2 & p^1 & p^0 \end{array} \\ \left[\begin{array}{ccccccc} 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{array}\right]_{4\times 7} \end{array}$$

Note : Since the generator matrix is non systematic form Hence parity check matrix can not be obtained using direct method ($H = [P^T ; I]$).

(f) For (7,4) Cyclic Codes.

$n = 7, \; K = 4, \; q = n-k = 7-4 = 3 \quad \therefore q = 3.$

The generator polynomial is a factor of $[p^n + 1]$

$\therefore p^7 + 1 = [p+1][p^3+p^2+1][p^3+p+1]$

$$\therefore \boxed{p^7+1 = (p+1)(p^3+p^2+1)(p^3+p+1)}.$$

The valid generating polynomial is given by $G(P)$ as

$$G(p) = p^q + g_{q-1}p^{q-1} + \cdots\cdots + g_2 p^2 + g_1 p^1 + 1.$$

$q = 3$ $\quad \therefore G(P) = p^3 + g_2 p^2 + g_1 p^1 + 1.$

The degree of generating polynomial should be 'q'

For this problem $q = 3$

∴ The valid generator polynomials for 'p^7+1' will be $(p^3 + p^2+1)$ and (p^3+p+1)

'$p+1$' will not be a generated polynomial since its degree is not q $(q=3)$.

∴ The generator polynomials for given (7,4) cyclic codes are

$$\boxed{\begin{aligned} G_1(p) &= p^3+p^2+1 \\ G_2(p) &= p^3+p+1. \end{aligned}}$$

The characteristics of five Important classes of Cyclic Codes :

① Cyclic Redundancy Check codes (CRC Codes) :

✓ Cyclic Codes are extremely well suited for error detection.

ie "A cyclic code used for error detection is referred to as Cyclic redundancy check (CRC) code".

✓ A CRC error burst of length B in an n-bit received word as a contiguous sequence of 'B' bits in which the first & last bits of any no. of intermediate bits are received in error.

* Binary (n, K) CRC codes are capable of detecting the following error patterns.

(i) All CRC error bursts of length 'n-K' or less.

(ii) A fraction of CRC error bursts of length equal to 'n-K+1', the fraction equals $1 - 2^{-(n-k-1)}$.

(iii) A fraction of CRC error bursts of length greater than 'n-K+1', the fraction equals $1 - 2^{-(n-K)}$.

(iv) All combinations of $d_{min} - 1$ (or fewer) errors.

(v) All error patterns with an odd no. of errors if the generator polynomial $g(x)$ for the code has an even no. of non zero coefficients.

Ex: CRC-12 Code : $1+p+p^2+p^3+p^{11}+p^{12}$, $n-k=12$

CRC-16 code : $1+p^{2}+p^{15}+p^{16}$, $n-k=16$.

② Maximum length codes :

For positive integer $k \geq 3$, there exists a maximum length code with the parameters as

Block length $n = 2^K - 1$

Min. distance $d_{min} = 2^{K-1}$

∴ The max. length codes are generated by polynomials of the form

$$g(p) = \frac{1+p^n}{h(p)}$$

where $h(P)$ = any primitive polynomial of degree 'k'.

ie The maximum length codes are the dual of Hamming Codes.

✓ The maximum length codes are also referred to as Pseudo-Noise (PN) codes.

ie The name "Pseudo Noise" is derived from the fact that these codes have correlation & spectral characteristics that resemble those of a white noise sequence.

③ Golay Codes: $(n, K) \rightarrow [23, 12]$

✓ A Golay code is a very special binary code that is capable of correcting any combinations of three or fewer random error in a block of 23 bits.

✓ The code has minimum distance of '7'.

ie Indeed the (23,12) Goly code is the only known three correcting binary perfect cyclic code.

Ex: The (23,12) Golay code is generated either by the polynomial

$$g_1(p) = 1 + p^2 + p^4 + p^6 + p^{10} + p^{11}.$$

(or)

$$g_2(p) = 1 + p + p^5 + p^6 + p^7 + p^9 + p^{11}.$$

∴ Both $g_1(p)$ & $g_2(p)$ are factor of '$1+D^{23}$'.

ie

$$1 + D^{23} = (1+p)[G_1(p) \cdot G_2(p)]$$

∴ The golay codes does not generalize to other combination of parameters n & K.

④ BCH (Bose-Chaudhuri-Hocquenghem) Codes:

✓ One of the most important & powerful classes of linear block codes are "BCH" codes.

✓ Specifically, for any positive integers K (equal to or >3) & t (less than $(2^K-1)/2$) there exists a binary BCH code with parameters

Block length $n = 2^K - 1$

no of msg bits $K \geq n -$

Minimum distance $d_{min} \geq 2t + 1$.

✓ Each BCH code is a t-error correcting code

ie It can detect & correct upto 't' random errors per codeword.

The hamming single error correcting codes can be described BCH codes.

✓ BCH codes offer flexibility in the choice of code parameters

ie Code rate and block length.

✓ Generator polynomials for binary block BCH codes of length upto $2^5 - 1$.

⑤ Reed-Solomon (RS) Codes:

✓ The Reed solomon codes are an important sub class of non binary BCH codes.

✓ The encoder for an RS code differs from a binary encoder in that it operates on multiple bits rather than individual bits

* The encoder for an RS (n,K) code on m-bit symbols groups the incoming binary data stream into blocks, each 'km' bits long. Each block is treated as K-symbols, with each symbol having m-bits.

✓ The m-bit symbols are called bytes, m is integer power of two'

∴ 8 bit **RS** codes are extremely powerful.

A t-error correcting RS code has

- ✓ Block length : $n = 2^K - 1$ symbols
- ✓ Message size : K-symbols
- ✓ Parity check size : $n - K = 2t$ symbols
- ✓ Minimum distance : $d_{min} = 2t + 1$ symbols.

✓ Every RS code is a maximum distance seperable code.

* RS-Codes → Highly efficient use of redundancy, & block length & symbol sizes can be adjusted readily & effcient decoding.

Convolutional Codes :

✓ Convolutional codes were first introduced by Elias (1995) as an alternative to block codes.

* In block coding, the encoder accepts a K-bit message blocks and generates an n-bit code word. Thus code words are produced on a block-by block basic.

However, where the messages bits come in serially rather than in large blocks, in which case the use of a buffer may be undesirable, In such situations, use 'convolutional coding'.

* The main <u>differences</u> between <u>block Codes</u> and <u>Convolutional Codes</u> as follows

① In a block codes, a block of n-bits generated by the encoder in a perticular time unit depends only on the block of K-input message bits within that time during.

② In a Convolutional code, the block of n-code bits generated by the encoder in the time during depends on not only the block of k-message bits but also on the previous (N-1) block of messages (N>1).

③ Like block codes, Convolutional codes can be designed both for detecting and Correcting of the errors.

④ Block Codes are better suitable for error detection and Convolutional Codes are mainly used for error correction.

⑤ A Convolutional encoder operates on the incoming message sequence continuously in a serial manner.

⑥ Convolutional Codes are well suited for mobile communications and satellite communications etc.

⑦ The use of non systematic codes is ordinarily preferred over systematic codes in Convolutional Codes.

Encoder for Convolutional Codes :

✓ A convolutional code can be generated by using of shift register and X-OR operation.

✓ The general representation of convolutional code is

$$(n, K, M)$$

where
$n \rightarrow$ No. of X-OR adders.
$K \rightarrow$ At a time no. of bits given to the register
$M \rightarrow$ No. of Shift Registers.

let

$L \rightarrow$ No. of bits present in the messge sequence.
$L+M \rightarrow$ Total no. of shifts required to reset the register.
$n(L+M) \rightarrow$ Total no. of bits in the Encoder output Sequence.

✓ The code rate for the convolutional code is given by

$$r_c = \frac{L}{n(L+M)} \qquad \left(\because\ r_c = \frac{K}{n} \text{ for block codes}\right)$$

If $L >> M$ then

$$r_c = \frac{L}{n \cdot L} = \frac{1}{n}$$

$\therefore$ Code rate $\boxed{r_c = \frac{1}{n} \text{ bits/symbol}}$

✓ The Constraint length (N) of convolutional codes is defined as the no. of shift over which a single message bit can influence the encoder output.

In an encoder with an M-stage shift registers the memory of the encoder equals, M messge bits and $N=M+1$ shifts are required for a messge bit to enter the shift register and finally come out.

$\therefore$ Hence the constraint length of the encoder is $\boxed{N = M+1}$

Example: (2,1,3) Convolutional Codes. Find the output of the encoder for an input data stream of 1011.

Sol: Given (n, K, M) convolutional Codes as $(2,1,3)$

$n = 2$ ∴ → No. of X-OR adders.

$K = 1$ → At a time no. of bits given to the register

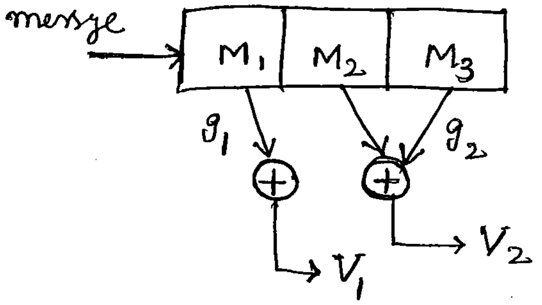
$M = 3$ → No. of shift registers.

Given messge: 1011.

∴ $L = 4$ → No. of bits present in the messge sequence.

$L+M = 3+4 = 7$ → shifts are required to reset the registors.

$n(L+M) = 2(7) = 14$ → The total no of bits in the encoder output Sequence.



(∵ let $g_1 = 100$

$g_2 = 011$).

$v_1 = M_1$

$v_2 = M_2 \oplus M_3$.

∴ Assume initially all shift registers are reset position

Now $M = \underset{MSB}{1}011$ is entered in the shift register from MSB bit

First bit of the input data stream is enter into M_1, the bit is initially present in M_1 is shifted to M_2 in next state ... this process continuous until the last bit of the messge has been entered and comes out.

Shift	Input	Registers			Outputs	
		M_1	M_2	M_3	$V_1 = M_1$	$V_2 = M_2 \oplus M_3$
-	-	0	0	0	0	0
1	1	1	0	0	1	0
2	0	0	1	0	0	1
3	1	1	0	1	1	1
4	1	1	1	0	1	1
5	0	0	1	1	0	0
6	0	0	0	1	0	1
7	0	0	0	0	0	0

$L+M = 7$ so 7 shifts.

where v_1 - First Code Symbol
v_2 - Second code Symbol
in Encoded output.

∴ The Encoder output Sequence $X = (\overset{v_1}{1}\overset{v_2}{0}\,\overset{v_1}{0}\overset{v_2}{1}\,11\,11\,00\,01\,\overset{v_1}{0}\overset{v_2}{0})$.

Encoding methods of Convolutional Codes:

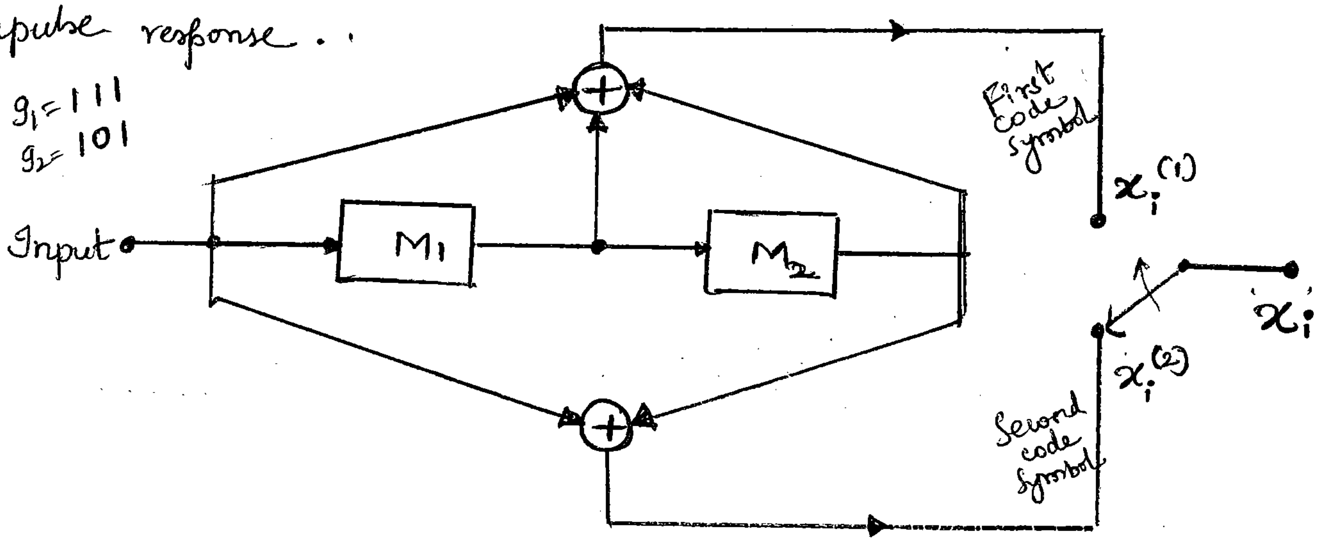
There are five methods for encoding of convolutional codes.

1. Time domain approach (or) Impulse approach (or) Connection diagram
2. Frequny domain approach (or) Transfer domain approach (or) Connection polynomial (or) Connection representation.
3. Tree diagram
4. State diagram
5. Trellis diagram.

① Time domain approach:

The time domain behaviour of convolutional codes with a code rate of '$\frac{1}{n}$' may be defined interms of a set of 'n' impulse response..

let $g_1 = 111$
$g_2 = 101$



- ✓ This consists of two modulo-2 adders and the code rate is $\frac{1}{2}$
- ✓ This Encoder are non systematic codes.
- ✓ For a given encoder two impulse responses are needed to characterised its behaviour in time domain

Ex: $g_1 = (1\ 1\ 1)$ $g_2 = (1\ 0\ 1)$

up

down ↳ connection doesnot exists.

* Let the sequence $[g_0^{(1)}, g_1^{(1)}, g_2^{(1)} \dots g_M^{(1)}]$ denotes the impulse response of the input top-adder output path of the encoder.

$[g_0^{(2)}, g_1^{(2)}, g_2^{(2)} \dots g_M^{(2)}]$ denotes the impulse response of the input bottom-adder output path of the encoder.

✓ The impulse response is so defined as the generated sequence of the code.

* The top output sequence is defined by the convolutional sum as

$$\boxed{x_i^{(1)} = \sum_{l=0}^{M} g_l^{(1)} \cdot M_{i-l}}$$ where $i = 0, 1, 2, \dots$

* The bottom output sequence is defined as

$$\boxed{x_i^{(2)} = \sum_{l=0}^{M} g_l^{(2)} \cdot M_{i-l}}$$, where $i = 0, 1, 2, \dots$

The two output sequences $x_i^{(1)}$ and $x_i^{(2)}$ are combined by the multiplexer to produce the encoder output sequence (x_i)

$$\therefore \boxed{\{x_i\} = (x_0^{(1)} x_0^{(2)} \quad x_1^{(1)} x_1^{(2)} \quad x_2^{(1)} x_2^{(2)} \dots\dots\dots)}$$

Problem ①

A convolutional encoder has a single shift register with two stages (ie constraint length 3), two X-OR adders and an output ~~sequ~~ multiplexer. The generated sequences of the encoder as

$$g_1 = (\underset{g_0}{1}\ \underset{g_1}{1}\ \underset{g_2}{1}) \quad \& \quad g_2 = (\underset{g_0}{1}\underset{g_1}{0}\underset{g_2}{1}).$$

Draw the block diagram of encoder and find the output of the encoder for an input data stream of '$\underset{M_0}{1}\ \underset{M_1}{0}\ \underset{M_2}{0}\ \underset{M_3}{1}\ \underset{M_4}{1}$' by using time domain approach.

(PTO) →

Sol Given generated Sequences

$$g_1 = (\underset{g_0}{1}\ \underset{g_1}{1}\ \underset{g_2}{1}) \quad \& \quad g_2 = (\underset{g_0}{1}\underset{g_1}{0}\underset{g_2}{1})$$

Constraint length $N = M+1 = 3$

$$\boxed{M=2}$$

∴ So. no-of shift registers are '2' $\boxed{M=2}$ M_0, M_1.

Given two x-OR adders. $\boxed{n=2}$

At a time no-of bits given to the register ie Single shift register | $K=1 \Rightarrow \boxed{K=1}$.

$(n,K,M) = (2,1,2)$ Convolutional Codes.

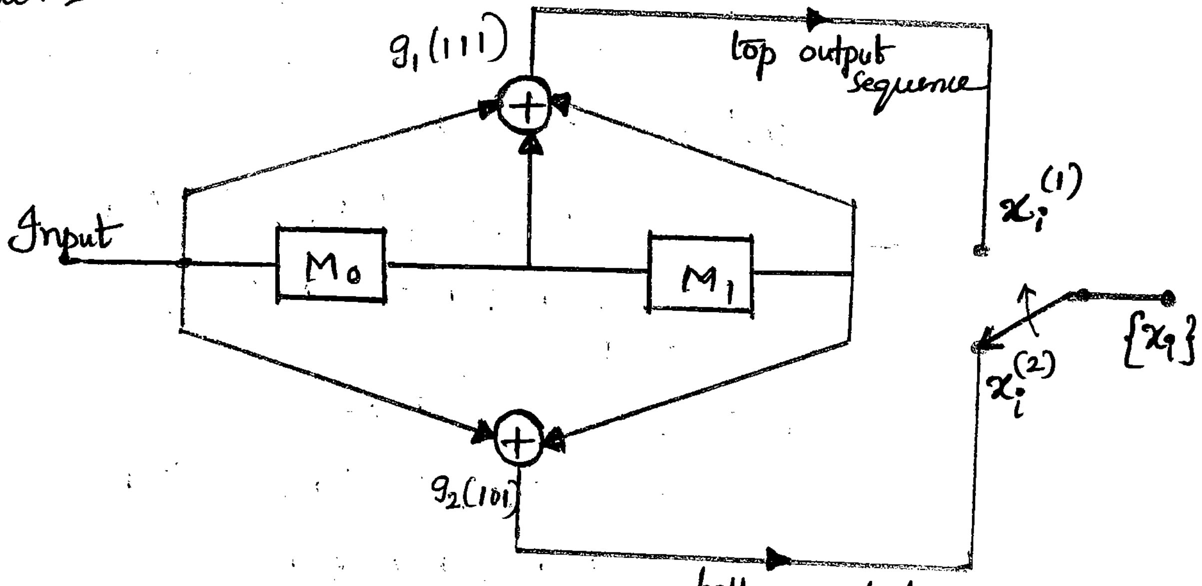
Given Input data stream 1 0 0 1 1

∴ $\boxed{L=5}$ ie no-of bits in message.

$L+M = 5+2 = 7 \Rightarrow \boxed{L+M=7}$

Total no-of bits in Encoded o/p sequence. $n(L+M) = 2(7) = 14 \Rightarrow \boxed{n(L+M)=14}$.

Thus.



* The top output Sequence can be generated by using.

$$x_i^{(1)} = \sum_{l=0}^{M} g_l^{(1)} M_{i-l} \,.$$ when $i = 0,1,2,3,4,5,6$. $\overset{L+M-1}{\swarrow}$

* The bottom output sequence can be generated by using

$$x_i^{(2)} = \sum_{l=0}^{M} g_l^{(2)} M_{i-l} \,, \quad i = 0,1,2,3,4,5,6 .$$

✓ The top output sequence

$$x_i^{(1)} = \sum_{l=0}^{2} g_l^{(1)} M_{i-l} \quad , i = 0,1,2,3,4,5,6.$$

For $i=0$, $x_0^{(1)} = \sum_{l=0}^{2} g_l^{(1)} \cdot M_{-l}$

$$= g_0^{(1)} M_0 + g_1^{(1)} \cdot M_{-1} + g_2^{(1)} \cdot M_{-2}$$

M

$$\underset{M_0}{\overset{1}{\downarrow}}\ \underset{M_1}{\overset{0}{\downarrow}}\ \underset{M_2}{\overset{0}{\downarrow}}\ \underset{M_3}{\overset{1}{\downarrow}}\ \underset{M_4}{\overset{1}{\downarrow}}$$

$$= 1(1) + 1(0) + (1)(0)$$

$$= 1 \Rightarrow \boxed{x_0^{(1)} = 1}$$

$$g_1 = \underset{g_0}{\overset{1}{\downarrow}}\ \underset{g_1}{\overset{1}{\downarrow}}\ \underset{g_2}{\overset{1}{\downarrow}}$$

For $i=1$, $x_1^{(1)} = \sum_{l=0}^{2} g_l^{(1)} \cdot M_{1-l}$

$$= g_0^{(1)} \cdot M_1 + g_1^{(1)} \cdot M_0 + g_2^{(1)} \cdot M_{-1}$$

$$= 1(0) + 1(1) + 1(0)$$

$$= 1 \quad \therefore \boxed{x_1^{(1)} = 1}$$

For $i=2$, $x_2^{(1)} = \sum_{l=0}^{2} g_l^{(1)} \cdot M_{2-l}$.

$$= g_0^{(1)} M_2 + g_1^{(1)} \cdot M_1 + g_2^{(1)} \cdot M_0.$$

$$= 1(0) + 1(0) + 1(1)$$

$$= 1 \quad \therefore \boxed{x_2^{(1)} = 1}$$

For $i=3$, $x_3^{(1)} = \sum_{l=0}^{2} g_l^{(1)} \cdot M_{3-l}$

$$= g_0^{(1)} M_3 + g_1^{(1)} M_2 + g_2^{(1)} M_1$$

$$= 1(1) + 1(0) + 1(0) \quad \therefore \boxed{x_3^{(1)} = 1}$$

$$= 1$$

For $i=4$, $x_4^{(1)} = \sum_{l=0}^{2} g_l^{(1)} \cdot M_{4-l}$

$$= g_0^{(1} M_4 + g_1^{(1)} \cdot M_3 + g_2^{(1)} \cdot M_2$$

$$= 1(1) + 1(1) + 1(0)$$

$$= 1 \oplus 1 \quad \therefore \boxed{x_4^{(1)} = 0}$$

$$= 0$$

For $i=5$, $x_5^{(1)} = \sum_{l=0}^{2} g_l^{(1)} M_{5-l}$

$$= g_0^{(1)} M_5 + g_1^{(1)} M_4 + g_2^{(1)} M_3$$

$$= 1(0) + 1(1) + 1(1)$$

$$= 1 \oplus 1 \quad \therefore \boxed{x_5^{(1)} = 0}$$

$$= 0$$

For $i=6$, $x_6^{(1)} = \sum_{l=0}^{2} g_l^{(1)} M_{6-l}$

$$= g_0^{(1)} M_6 + g_1^{(1)} M_5 + g_2^{(1)} M_4$$

$$= 1(0) + 1(0) + 1(1)$$

$$= 1 \qquad \therefore \boxed{x_6^{(1)} = 1}$$

$\therefore$ The top output sequence

$$\boxed{x_i^{(1)} = (1\ 1\ 1\ 1\ 0\ 0\ 1)}$$

✓ The bottom output sequence

$$x_i^{(2)} = \sum_{l=0}^{2} g_l^{(2)} \cdot M_{i-l}, \quad i = 0,1,2,3,4,5,6.$$

For $i=0$, $x_0^{(2)} = \sum_{l=0}^{2} g_l^{(2)} \cdot M_{-l}$

$$= g_0^{(2)} \cdot M_0 + g_1^{(2)} \cdot M_{-1} + g_2^{(2)} \cdot M_{-2}.$$

$\underset{M_0}{1}\ \underset{M_1}{0}\ \underset{M_2}{0}\ \underset{M_3}{1}\ \underset{M_4}{1}$

$$= (1)(1) + 0(0) + 1(0)$$

$$= 1 \qquad \therefore \boxed{x_0^{(2)} = 1}.$$

$g_2 = \underset{g_0}{1}\ \underset{g_1}{0}\ \underset{g_2}{1}$

For $i=1$, $x_1^{(2)} = \sum_{l=0}^{2} g_l^{(2)} M_{1-l}$

$$= g_0^{(2)} M_1 + g_1^{(2)} \cdot M_0 + g_2^{(2)} M_{-1}$$

$$= 1(0) + 0(1) + 1(0)$$

$$= 0 \qquad \therefore \boxed{x_1^{(2)} = 0}$$

For $i=2$, $x_2^{(2)} = \sum_{l=0}^{2} g_l^{(2)} \cdot M_{2-l}.$

$$= g_0^{(2)} \cdot M_2 + g_1^{(2)} M_1 + g_2^{(2)} M_0$$

$$= 1(0) + 0(0) + 1(1)$$

$$= 1. \qquad \therefore \boxed{x_2^{(2)} = 1}.$$

For $i=3$, $x_3^{(2)} = \sum_{i=0}^{2} g_l^{(2)} M_{3-l}$

$$= g_0^{(2)} M_3 + g_1^{(2)} M_2 + g_2^{(2)} M_1$$

$$= 1(1) + 0(0) + 1(0)$$

$$= 1 \qquad \therefore \boxed{x_3^{(2)} = 1}.$$

For $i=4$, $x_4^{(2)} = \sum_{l=0}^{2} g_l^{(2)} M_{4-l}$

$$= g_0^{(2)} M_4 + g_1^{(2)} M_3 + g_2^{(2)} M_2$$

$$= 1(1) + 0(1) + 1(0)$$

$$= 1 \qquad \therefore \boxed{x_4^{(2)} = 1}$$

For $i=5$, $x_5^{(2)} = \sum_{l=0}^{2} g_l^{(2)} M_{5-l}$

$$= g_0^{(2)} M_5 + g_1^{(2)} M_4 + g_2^{(2)} M_3$$

$$= 1(0) + 0(1) + 1(1)$$

$$= 1 \qquad \therefore \boxed{x_5^{(2)} = 1}$$

For $i=6$, $x_6^{(2)} = \sum_{l=0}^{2} g_l^{(2)} M_{6-l}$

$$= g_0^{(2)} M_6 + g_1^{(2)} M_5 + g_2^{(2)} M_4$$

$$= 1(0) + 0(0) + 1(1)$$

$$= 1 \qquad \therefore \boxed{x_6^{(2)} = 1}$$

∴ The bottom output sequence

$$\boxed{x_i^{(2)} = (1\ 0\ 1\ 1\ 1\ 1\ 1)}$$

∴ The Encoded output sequence is given by.

$$\{x_i\} = (x_0^{(1)}\ x_0^{(2)}\ \ x_1^{(1)} x_1^{(2)}\ \ \ldots\ldots\ \ x_6^{(1)}\ x_6^{(2)})$$

$$x_i^{(1)} = (1\ 1\ 1\ 1\ 0\ 0\ 1)$$

$$\downarrow\ \ \downarrow\qquad\qquad\quad \downarrow$$

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

$$x_i = (11\ 10\ 11\ 11\ 01\ 01\ 11)$$

∴ The Encoded output Sequence

$$\boxed{x_i = (1\ 1\ 1\ 0\ 1\ 1\ 1\ 1\ 0\ 1\ 0\ 1\ 1\ 1)}$$

② Frequency domain approach :

- Transfer domain response is represented in the form of polynomials.

* The input top adder - output path of the encoder

$$g^{(1)}(D) = g_0^{(1)} + g_1^{(1)}D + g_2^{(1)}D^2 + \cdots\cdots + g_M^{(1)}D^M.$$

* The input - bottom adder - output path of the encoder

$$g^{(2)}(D) = g_0^{(2)} + g_1^{(2)}D + g_2^{(2)}D^2 + \cdots\cdots + g_M^{(2)}D^M.$$

where $g_0^{(1)}, g_1^{(1)}, \cdots g_M^{(1)}$ and $g_0^{(2)}, g_1^{(2)} \cdots g_M^{(2)}$ are the elements of the transfr domain response of the path.

The message polynomial is given as

$$M(D) = M_0 + M_1D + M_2D^2 + \cdots\cdots + M_{L-1}D^{L-1}.$$

where L - length of the messege sequence.

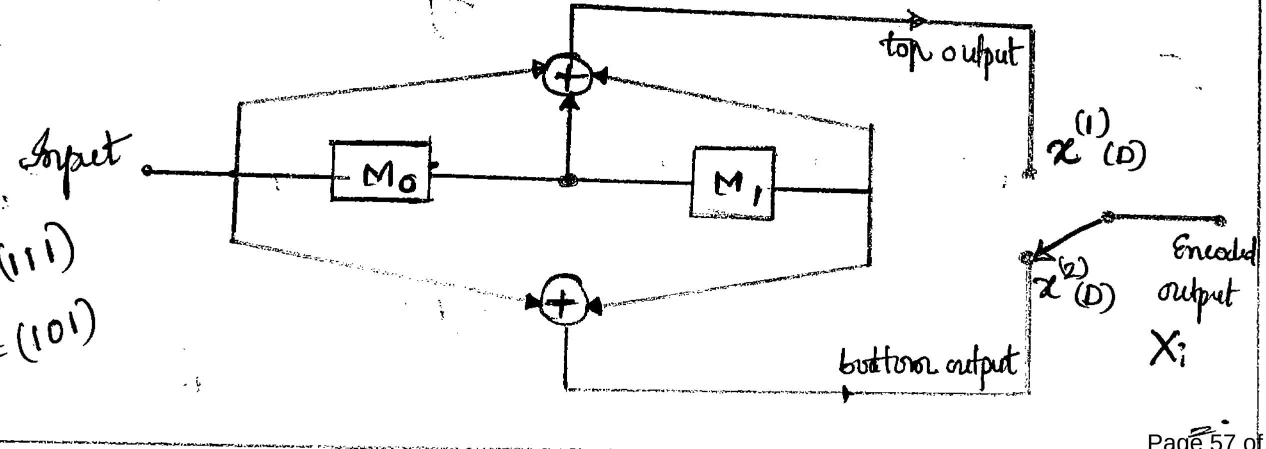
∴ The top-output polynomial is given as

$$\boxed{X^{(1)}(D) = g^{(1)}(D) \cdot M(D)}$$

The bottom output polynomial is given as

$$\boxed{X^{(2)}(D) = g^{(2)}(D) \cdot M(D).}$$

∴ The Encoded output Sequence $X_i = [X^1(D)\ X^{(2)}(D) \cdots\cdots]$



let
$g_1 = (111)$
$g_2 = (101)$

Problem ②

A convolutional encoder has a single shift register with two stages (ie Constraint length 3), two X-OR adders and an output multiplexer. The generated polynomials of the encoder as follows $g^{(1)}(D) = 1+D+D^2$ and $g^{(2)}(D) = 1+D^2$.

Draw the block diagram of encoder and find the output of the encoder for an input data stream of (10011) by using frequency / transfer domain approach.

Sol Given data $g^{(1)}(D) = 1+D+D^2$

$g^{(2)}(D) = 1+D^2$

Input data stream ie message as $\underset{M_0}{1}\ \underset{M_1}{0}\ \underset{M_2}{0}\ \underset{M_3}{1}\ \underset{M_4}{1}$.

Given single shift register $\boxed{K=1}$

Constraint length $N = M+1 = 3 \Rightarrow \boxed{M=2}$

$\therefore$ No-of shift register 2

No-of X-OR adders 2 ; $\boxed{n=2}$

$\boxed{L=5}$, $L+M = 5+2=7$ $\therefore \boxed{L+M=7}$

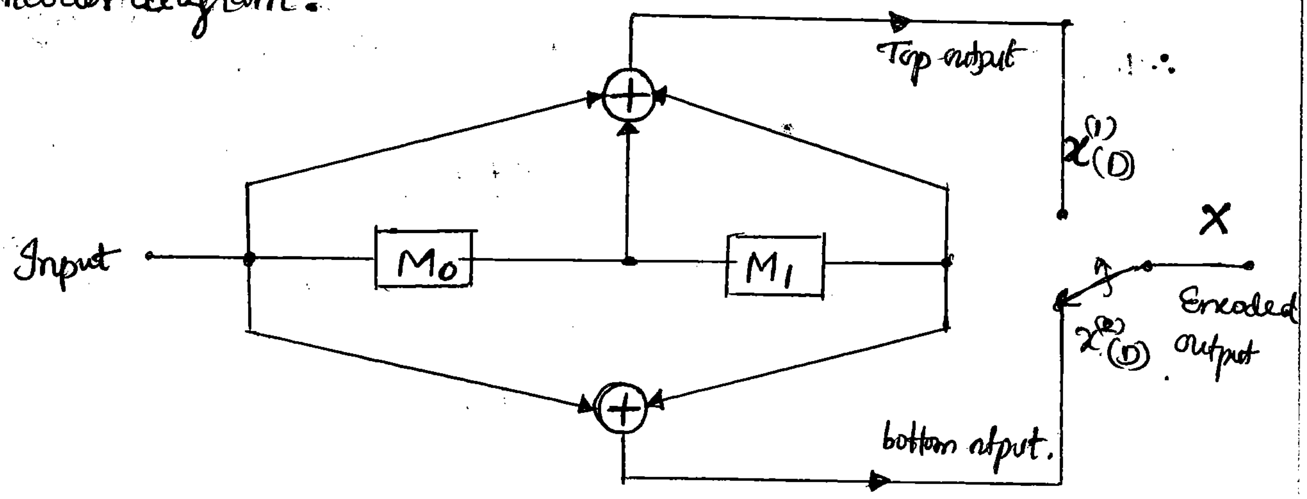
$n(L+M) = 2(7) = 14$ $\therefore \boxed{n(L+M)=14}$

Message polynomial as

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

$\therefore \boxed{M(D) = 1+D^3+D^4}$.

Encoder diagram:



✓ The top output polynomial is given as

$$X^{(1)}(D) = g^{(1)}(D) \cdot M(D)$$

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

$$= 1+D^3+\cancel{D^4}+D+D^4+\cancel{D^5}+D^2+\cancel{D^5}+D^6$$

$$= 1+D+D^2+D^3+D^6.$$

(XOR operat
1+1=0
0+0=0
0+1=1
1+0=1)

$$\therefore \boxed{X^1(D) = (1\ 1\ 1\ 1\ 0\ 0\ 1)}$$

✓ The bottom output polynomial is given as

$$X^2(D) = g^{(2)}(D) \cdot M(D)$$

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

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

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

$$\therefore \boxed{X^{(2)}(D) = (1\ 0\ 1\ 1\ 1\ 1\ 1)}$$

∴ The Encoded output polynomial in bits.

$$\begin{array}{l} X^1(D) = (1\ 1\ 1\ 1\ 0\ 0\ 1) \\ \qquad\qquad\ \ \downarrow\ \downarrow\ \downarrow \qquad\quad \downarrow \\ X^2(D) = (1\ 0\ 1\ 1\ 1\ 1\ 1) \end{array}$$

$$\therefore \boxed{\mathbf{X}_i = (1\ 1\ 1\ 0\ 1\ 1\ 1\ 1\ 0\ 1\ 0\ 1\ 1\ 1)}.$$

HW ③ A convolutional encoder has a single shift register with two stages, 3 modulo-2 adders & an output multiplexer. The generated sequence of the encoder as follows.

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

Draw the block diagram of encoder & the output of the encoder using time and frequency domain approach with message data (1 0 0 1 1).

Sol Given data Input message (10011).

$L=5$ → no of bits in the input data

$K=1$ → single shift register.

$M=2$ → No. of shift registers.

$L+M = 5+2=7$ → total no of shift required to reset the register.

$n=3$ → no of modulo-2 adders.

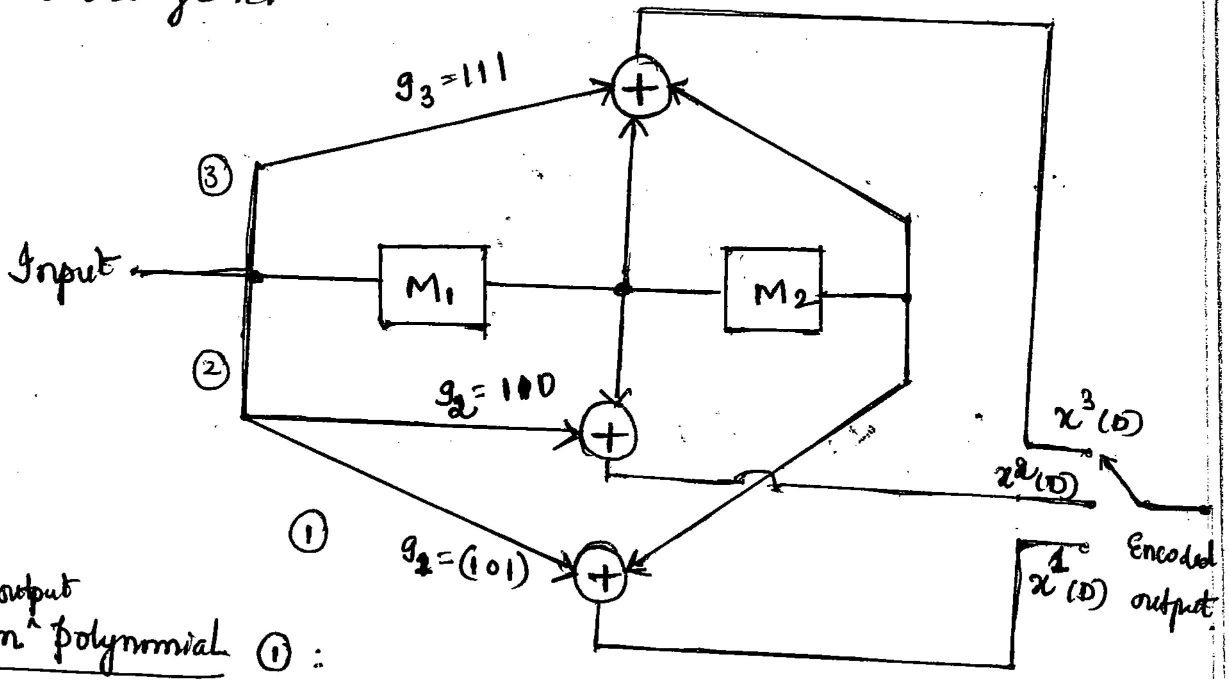
$n(L+M) = 3(7) = 21$ → Total no of bits in the encoder output sequence.

Transfer / Frequency domain approach:

- ✓ Message polynomial $M(D) = 1+D^3+D^4$.
- ✓ Generated polynomial

$$g^{(1)}(D) = (1\,0\,1) = 1+D^2$$
$$g^{(2)}(D) = (1\,1\,0) = 1+D.$$
$$g^{(3)}(D) = (1\,1\,1) = 1+D+D^2.$$

∴ Encoder block diagram:



The bottom output polynomial ①:

$$X^1(D) = g^{(1)}(D)\cdot M(D)$$
$$= (1+D^2)(1+D^3+D^4)$$
$$= 1+D^3+D^4+D^2+D^5+D^6$$
$$= 1+D^2+D^3+D^4+D^5+D^6$$
$$\boxed{X^1(D) = (1\,0\,1\,1\,1\,1\,1)}$$

The bottom output polynomial ② :

$$X^{(2)}(D) = g^{(2)}(D)\cdot M(D)$$
$$= (1+D)(1+D^3+D^4)$$
$$= 1+D^3+\cancel{D^4}+D+\cancel{D^4}+D^5$$
$$= 1+D+D^3+D^5$$

$$\boxed{X^{(2)}(D) = 1\ 1\ 0\ 1\ 0\ 1\ 0}$$

The top output polynomial ③ :

$$X^{(3)}(D) = g^{(3)}(D)\cdot M(D)$$
$$= (1+D+D^2)(1+D^3+D^4)$$
$$= 1+D^3+\cancel{D^4}+D+\cancel{D^4}+\cancel{D^5}+D^2+\cancel{D^5}+D^6$$
$$= 1+D+D^2+D^3+D^6.$$

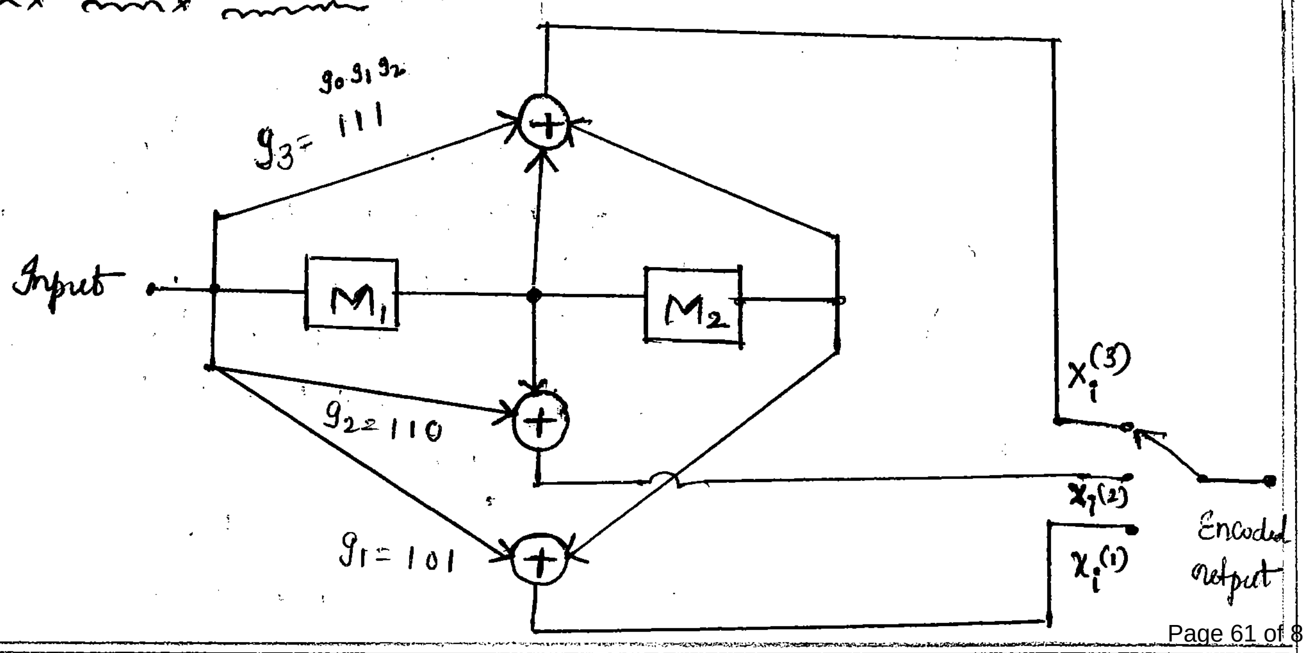
$$\boxed{X^{(3)}(D) = (1\ 1\ 1\ 1\ 0\ 0\ 1)}$$

∴ The Encoder output polynomial is given by $\{X\}$.

$X^1 = (1\,0\,1\,1\,1\,1\,1)$
$X^2 = (1\,1\,0\,1\,0\,1\,0)$
$X^3 = (1\,1\,1\,1\,0\,0\,1)$

$$\therefore \boxed{\{X\} = (1\,1\,1\,0\,1\,1\,1\,0\,1\,1\,1\,1\,1\,0\,0\,1\,1\,0\,1\,0\,1)}$$

Time domain approach :



✓ Message bits $(\underset{M_0}{1}\ \underset{M_1}{0}\ \underset{M_2}{0}\ \underset{M_3}{1}\ \underset{M_4}{1})$

~ Given generated Sequences

$$g_1 = (\underset{g_0}{1}\ \underset{g_1}{0}\ \underset{g_2}{1}) \quad g_2 = (\underset{g_0}{1}\ \underset{g_1}{1}\ \underset{g_2}{0}) \ , \ g_3 = (\underset{g_0}{1}\ \underset{g_1}{1}\ \underset{g_2}{1})$$

✓ The <u>bottom output sequence ①</u> can be generated by using.

$$x_i^{(1)} = \sum_{l=0}^{M} g_l^{(1)} M_{i-l} \quad ; \quad i = 0, 1, 2, 3, 4, 5, 6.$$

For $i = 0$, $x_0^{(1)} = \sum_{l=0}^{2} g_l^{(1)} M_{-l}$

$= g_0^{(1)} M_0 + g_1^{(1)} M_{-1} + g_2^{(1)} M_{-2}$

$= 1(1) + 0(0) + 1(0)$

$= 1$ $\therefore$ $\boxed{x_0^{(1)} = 1}$

$M = (\underset{M_0}{1}\ \underset{M_1}{0}\ \underset{M_2}{0}\ \underset{M_3}{1}\ \underset{M_4}{1})$

$g_1 = (\underset{g_0}{1}\ \underset{g_1}{0}\ \underset{g_2}{1})$

for $i = 1$, $x_1^{(1)} = \sum_{l=0}^{2} g_l^{(1)} M_{1-l}$

$= g_0^{(1)} M_1 + g_1^{(1)} M_0 + g_2^{(1)} M_{-1}$

$= 1(0) + 0(1) + 1(0) = 0$ $\therefore$ $\boxed{x_1^{(1)} = 0}$

for $i = 2$, $x_2^{(1)} = \sum_{l=0}^{2} g_l^{(1)} M_{2-l}$

$= g_0^{(1)} M_2 + g_1^{(1)} M_1 + g_2^{(1)} M_0$

$= 1(0) + 0(0) + 1(1) = 1$ $\therefore$ $\boxed{x_2^{(1)} = 1}$

for $i = 3$ $x_3^{(1)} = \sum_{l=0}^{2} g_l^{(1)} M_{3-l}$

$= g_0^{(1)} M_3 + g_1^{(1)} M_2 + g_2^{(1)} M_1$

$= 1(1) + 0(0) + 1(0) = 1$ $\therefore$ $\boxed{x_3^{(1)} = 1}$

For $i = 4$, $x_4^{(1)} = \sum_{l=0}^{2} g_l^{(1)} M_{4-l}$

$= g_0^{(1)} M_4 + g_1^{(1)} M_3 + g_2^{(1)} M_2$

$= 1(1) + 0(1) + 1(0) = 1$ $\therefore$ $\boxed{x_4^{(1)} = 1}$

for $i = 5$, $x_5^{(1)} = \sum_{l=0}^{2} g_l^{(1)} M_{5-l}$

$= g_0^{(1)} M_5 + g_1^{(1)} M_4 + g_2^{(1)} M_3$

$= 1(0) + 0(1) + 1(1) = 1$ $\therefore$ $\boxed{x_5^{(1)} = 1}$

for $i = 6$, $x_6^{(1)} = \sum_{l=0}^{2} g_l^{(1)} M_{6-l}$

$= g_0^{(1)} M_6 + g_1^{(1)} M_5 + g_2^{(1)} M_4$

$= 1(0) + 0(0) + 1(1) = 1$ $\therefore$ $\boxed{x_6^{(1)} = 1}$

The bottom output Sequence ② can be generated by using

$$x_i^{(2)} = \sum_{l=0}^{2} g_l^{(2)} M_{i-l} \ . \quad ; \ i = 0, 1, 2, 3, 4, 5, 6 \ .$$

$M_0 \ M_1 \ M_2 \ M_3 \ M_4$
$M = 1 \ 0 \ 0 \ 1 \ 1$

$g_0 \ g_1 \ g_2$
$g_2 = 1 \ 1 \ 0$

For $i = 0$

$$x_0^{(2)} = \sum_{l=0}^{2} g_l^{(2)} M_{-l}$$
$$= g_0^{(2)} M_0 + g_1^{(2)} M_{-1} + g_2^{(2)} M_{-2}$$
$$= 1(1) + 1(0) + 0(0) = 1 \ . \quad \therefore \boxed{x_0^{(2)} = 1}$$

For $i = 1$

$$x_1^{(2)} = \sum_{l=0}^{2} g_l^{(2)} M_{1-l}$$
$$= g_0^{(2)} M_1 + g_1^{(2)} M_0 + g_2^{(2)} M_{-1}$$
$$= 1(0) + 1(1) + 0(0) = 1 \quad \therefore \boxed{x_1^{(2)} = \_}$$

For $i = 2$,

$$x_2^{(2)} = \sum_{l=0}^{2} g_l^{(2)} M_{2-l}$$
$$= g_0^{(2)} M_2 + g_1^{(2)} M_1 + g_2^{(2)} M_0$$
$$= 1(0) + 1(0) + 0(1) = 0 \quad \therefore \boxed{x_2^{(2)} = 0}$$

For $i = 3$,

$$x_3^{(2)} = \sum_{l=0}^{2} g_l^{(2)} M_{3-l} \ .$$
$$= g_0^{(2)} M_3 + g_1^{(2)} M_2 + g_2^{(2)} M_1$$
$$= 1(1) + 1(0) + 0(0) = 1 \quad \therefore \boxed{x_3^{(2)} = 1}$$

For $i = 4$,

$$x_4^{(2)} = \sum_{l=0}^{2} g_l^{(2)} M_{4-l}$$
$$= g_0^{(2)} M_4 + g_1^{(2)} M_3 + g_2^{(2)} M_2$$
$$= 1(1) + 1(1) + 0(0) = 0 \quad \therefore \boxed{x_4^{(2)} = 0}$$

For $i = 5$,

$$x_5^{(2)} = \sum_{l=0}^{2} g_l^{(2)} M_{5-l}$$
$$= g_0^{(2)} M_5 + g_1^{(2)} M_4 + g_2^{(2)} M_3$$
$$= 1(0) + 1(1) + 0(1) = 1 \quad \therefore \boxed{x_5^{(2)} = 1}$$

For $i = 6$,

$$x_6^{(2)} = \sum_{l=0}^{2} g_l^{(2)} M_{6-l}$$
$$= g_0^{(2)} M_6 + g_1^{(2)} M_5 + g_2^{(2)} M_4$$
$$= 1(0) + 1(0) + 0(1) = 0 \quad \therefore \boxed{x_6^{(2)} = 0}$$

∴ The bottom output sequences ① & ②

$$\boxed{x_i^{(1)} = (1 \ 0 \ 1 \ 1 \ 1 \ 1 \ 1)}$$
$$\boxed{x_i^{(2)} = (1 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0)}$$

∴ The top output sequence ③ can be generated by using

$$x_i^{(3)} = \sum_{l=0}^{M} g_l^{(3)} M_{i-l} \ ., \quad i = 0, 1, 2, 3, 4, 5, 6 \ .$$

$M_0\ M_1\ M_2\ M_3\ M_4$
$M = (1\ 0\ 0\ 1\ 1)$

$g_3 = (1\ 1\ 1)$
$\quad\ \ g_0\ g_1\ g_2$

For $i=0$, $x_0^{(3)} = \sum_{l=0}^{2} g_l^{(3)} M_{-l}$

$= g_0^{(3)} M_0 + g_1^{(3)} M_{-1} + g_2^{(3)} M_{-2}$

$= 1(1)+1(0)+1(0) = 1$ $\quad\therefore \boxed{x_0^{(3)} = 1}$

For $i=1$, $x_1^{(3)} = \sum_{l=0}^{2} g_l^{(3)} M_{1-l}$

$= g_0^{(3)} M_1 + g_1^{(3)} M_0 + g_2^{(3)} M_{-1}$

$= 1(0) + 1(1) + 1(0) = 1$ $\quad\therefore \boxed{x_1^{(3)} = 1}$

For $i=2$, $x_2^{(3)} = \sum_{l=0}^{2} g_l^{(3)} M_{2-l}$

$= g_0^{(3)} M_2 + g_1^{(3)} M_1 + g_2^{(3)} M_0$

$= 1(0) + 1(0) + 1(1) = 1$ $\quad\therefore \boxed{x_2^{(3)} = 1}$

For $i=3$, $x_3^{(3)} = \sum_{l=0}^{2} g_l^{(3)} M_{3-l}$

$= g_0^{(3)} M_3 + g_1^{(3)} M_2 + g_2^{(3)} M_1$

$= 1(1) + 1(0) + 1(0) = 1$ $\quad\therefore \boxed{x_3^{(3)} = 1}$

For $i=4$, $x_4^{(3)} = \sum_{l=0}^{2} g_l^{(3)} M_{4-l}$

$= g_0^{(3)} M_4 + g_1^{(3)} M_3 + g_2^{(3)} M_2$

$= 1(1) + 1(1) + 1(0) = 0$ $\quad\therefore \boxed{x_4^{(3)} = 0}$

For $i=5$, $x_5^{(3)} = \sum_{l=0}^{2} g_l^{(3)} M_{5-l}$

$= g_0^{(3)} M_5 + g_1^{(3)} M_4 + g_2^{(3)} M_3$

$= 1(0) + 1(1) + 1(1) = 0$ $\quad\therefore \boxed{x_5^{(3)} = 0}$

For $i=6$, $x_6^{(3)} = \sum_{l=0}^{2} g_l^{(3)} M_{6-l}$

$= g_0^{(3)} M_6 + g_1^{(3)} M_5 + g_2^{(3)} M_4$

$= 1(0) + 1(0) + 1(1) = 1$ $\quad\therefore \boxed{x_6^{(3)} = 1}$

∴ The top output sequence ③ as

$$\therefore \boxed{x_i^{(3)} = (1\ 1\ 1\ 1\ 0\ 0\ 1)}$$

$\therefore\ x_i^{(1)} = (1\ 0\ 1\ 1\ 1\ 1\ 1)$

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

$x_i^{(3)} = (1\ 1\ 1\ 1\ 0\ 0\ 1)$

∴ The encoder output sequence can be generated by using

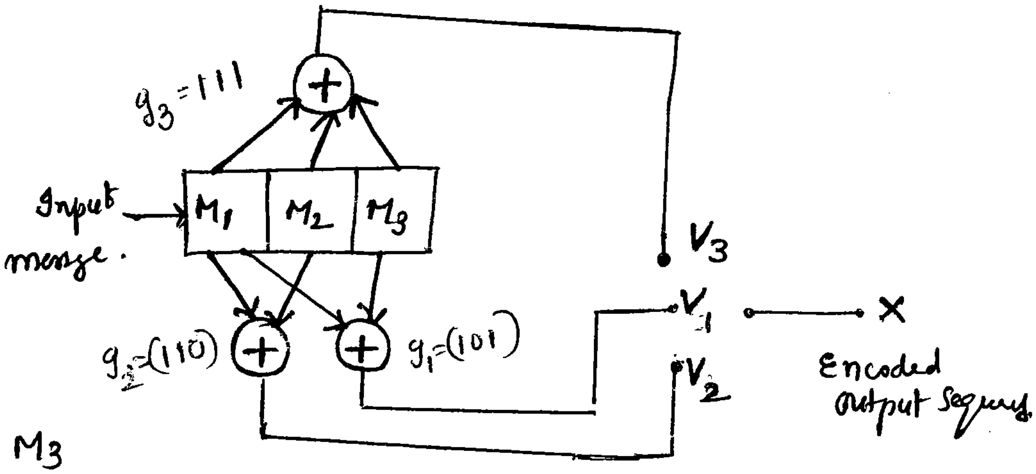
$$X = \{ x_0^{(1)} x_0^{(2)} x_0^{(3)}\ \ x_1^{(1)} x_1^{(2)} x_1^{(3)}\ \ldots\ldots\ x_6^{(1)} x_6^{(2)} x_6^{(3)} \}.$$

$$\therefore \boxed{X = (1\ 1\ 1\ 0\ 1\ 1\ 1\ 0\ 1\ 1\ 1\ 1\ 1\ 0\ 0\ 1\ 1\ 0\ 1\ 0\ 1)}$$

II. Method :

$g_1 = (101)$, $g_2 = (110)$, $g_3 = (111)$.

Message Sequence (10011).



$\therefore$ $V_1 = M_1 \oplus M_3$

$V_2 = M_1 \oplus M_2$

$V_3 = M_1 \oplus M_2 \oplus M_3$.

shift NO	Input M	Shift registers M_1 M_2 M_3	Output sequences $V_1 = M_1 \oplus M_3$	 $V_2 = M_1 \oplus M_2$	 $V_3 = M_1 \oplus M_2 \oplus M_3$
-	-	0 0 0	0	0	0
1	1	1 0 0	1	1	1
2	0	0 1 0	0	1	1
3	0	0 0 1	1	0	1
4	1	1 0 0	1	1	1
5	1	1 1 0	1	0	0
6	0	0 1 1	1	1	0
7	0	0 0 1	1	0	1

$\therefore$ $v_1 = (1\,0\,1\,1\,1\,1\,1)$

$V_2 = (1\,1\,0\,1\,0\,1\,0)$

$V_3 = (1\,1\,1\,1\,0\,0\,1)$.

$\therefore$ The Encoded output $\boxed{X = (111\,011\,101\,111\,100\,110\,101)}$

HW :

④ Consider the code rate is $1/2$, Constraint length '2'. Find encoded output sequence for the given input data (10111) & $g_1 = (11)$, $g_2 = (00)$, using time & transfer domain approach methods. $\therefore$ $x_1 = (111001)$

$x_2 = (000000)$

$\therefore$ $X = (101010000010)$.

***

⑤ A (2,1,3) Convolutional Code is described by

$$g_1 = (1\,1\,1) \quad \& \quad g_2 = (1\,0\,1)$$

(a) Draw the Encoder diagram for this code.

(b) Find the encoded output sequence for an input message sequence 1 1 0 1 1.

(c) Draw the Code tree diagram for this code and find the encoded output sequence for the given message sequence.

(d) Draw the State diagram & find encoded output sequence for the given message sequence.

(e) Draw the trellis diagram & find encoder output sequence for the given message sequence.

Sol. Given (n, K, M) convolutional code → (2, 1, 3).

∴ $n = 2$ → no. of X-OR adders.

$K = 1$ → At a time no. of bits given to the register.

$M = 3$ → no. of shift registers.

Given message sequence (1 1 0 1 1).

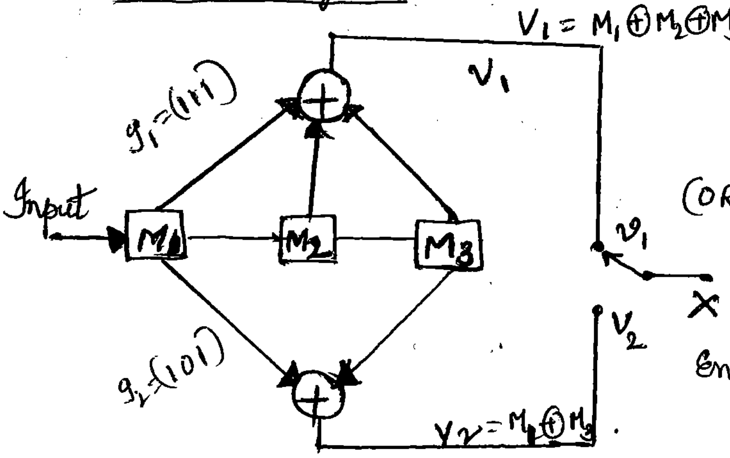
∴ $L = 5$ → no. of bits present in message sequence.

$L+M = 5+3 = 8$ → no. of shift required to reset the registers.

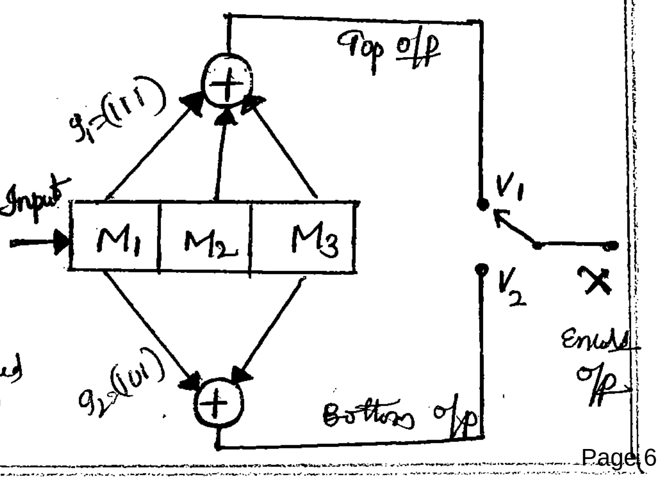
$n(L+M) = 2(8) = 16$ → Total no. of bits in the encoded output sequence.

Given $g_1 = (1\,1\,1)$ & $g_2 = (1\,0\,1)$.

(a) Encoder diagram:



(OR)



(b) Encoded Output Sequence :

Given message '11011'.

Shift	Input	Registers			top & bottom output sequences	
		M_1	M_2	M_3	$V_1 = M_1 \oplus M_2 + M_3$	$V_2 = M_1 \oplus M_3$
–	–	0	0	0	0	0
1	1	1	0	0	1	1
2	1	1	1	0	0	1
3	0	0	1	1	0	1
4	1	1	0	1	0	0
5	1	1	1	0	0	1
6	0	0	1	1	0	1
7	0	0	0	1	1	1
8	0	0	0	0	0	0

∴ The top output Sequence $v_1 = (1\ 0\ 0\ 0\ 0\ 0\ 1\ 0)$

The bottom output sequence $v_2 = (1\ 1\ 1\ 0\ 1\ 1\ 1\ 0)$

∴ The Encoded output Sequence is given by.

$$\boxed{X = (\ 1101\ \ 0100\ \ 0101\ \ 1100)}$$

(OR) Frequency domain approach:

Message 11011 ∴ $M(D) = 1 + D + D^3 + D^4$

$g_1 = (1\ 1\ 1)$ $g^{(1)}(D) = 1 + D + D^2$

$g_2 = (1\ 0\ 1)$ $g^{(2)}(D) = 1 + D^2$

∴ Top o/p sequence is

$$\underline{X^{(1)}(D) = g^{(1)}(D) \cdot M(D)}$$

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

$$= 1 + \cancel{D} + \cancel{D^3} + \cancel{D^4} + \cancel{D} + \cancel{D^2} + \cancel{D^4} + \cancel{D^5} + \cancel{D^2} + \cancel{D^3} + \cancel{D^5} + D^6$$

$$= 1 + D^6$$

$$X^1(D) = (1\ 0\ 0\ 0\ 0\ 0\ 1\ 0)$$

Bottom o/p Sequence

$$\underline{X^{(2)}(D) = g^{(2)}(D) \cdot M(D)}$$

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

$$= 1 + D + \cancel{D^3} + D^4 + D^2 + \cancel{D^3} + D^5 + D^6$$

$$= 1 + D + D^2 + D^4 + D^5 + D^6$$

$$X^{(2)}(D) = (1\ 1\ 1\ 0\ 1\ 1\ 1\ 0)$$

∴ O/p Encoded sequence $\boxed{X = (1101010001011100)}$

(c) Code tree diagram :

The number of states is given by '$2^{K(M-1)}$'

where K - At a time no.of bits given to the register

M - no.of shift registers.

$(n,K,M) = (2,1,3)$

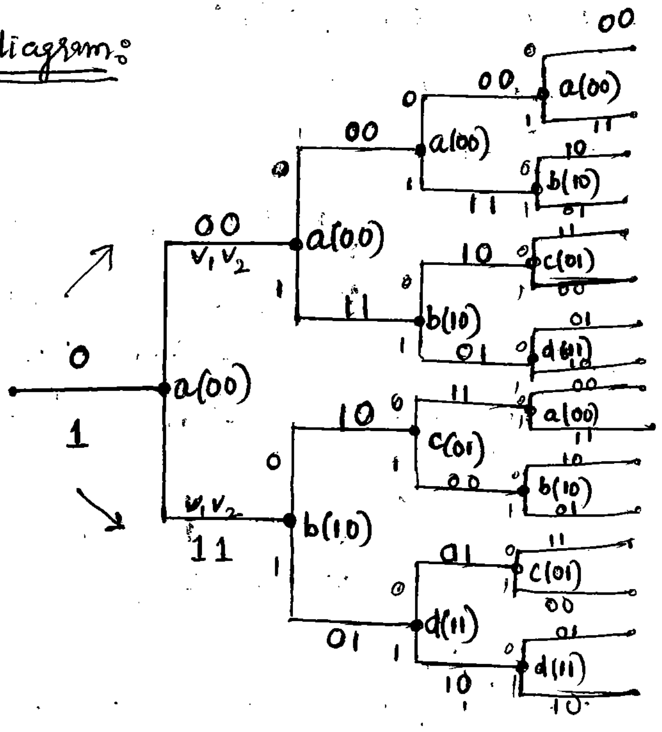
∴ no.of states $2^{1(3-1)} = 2^2 = 4$ ie
00
01
10
11

ie two states S_1 and S_2.

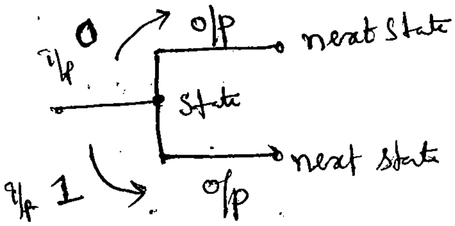
Let a=00, b=10, c=01, d=11.
↑ ↑

Input M_1	Present States S_1 M_2	S_2 M_3		outputs $V_1 = M_1 \oplus M_2 \oplus M_3$	$V_2 = M_1 \oplus M_3$	Next states S_1 M_1	S_2 M_2	
0	0	0	a	0	0	0	0	a
1	0	0	a	1	1	1	0	b
0	0	1	c	1	1	0	0	a
1	0	1	c	0	0	1	0	b
0	1	0	b	1	0	0	1	c
1	1	0	b	0	1	1	1	d
0	1	1	d	0	1	0	1	c
1	1	1	d	1	0	1	1	d

Tree diagram:

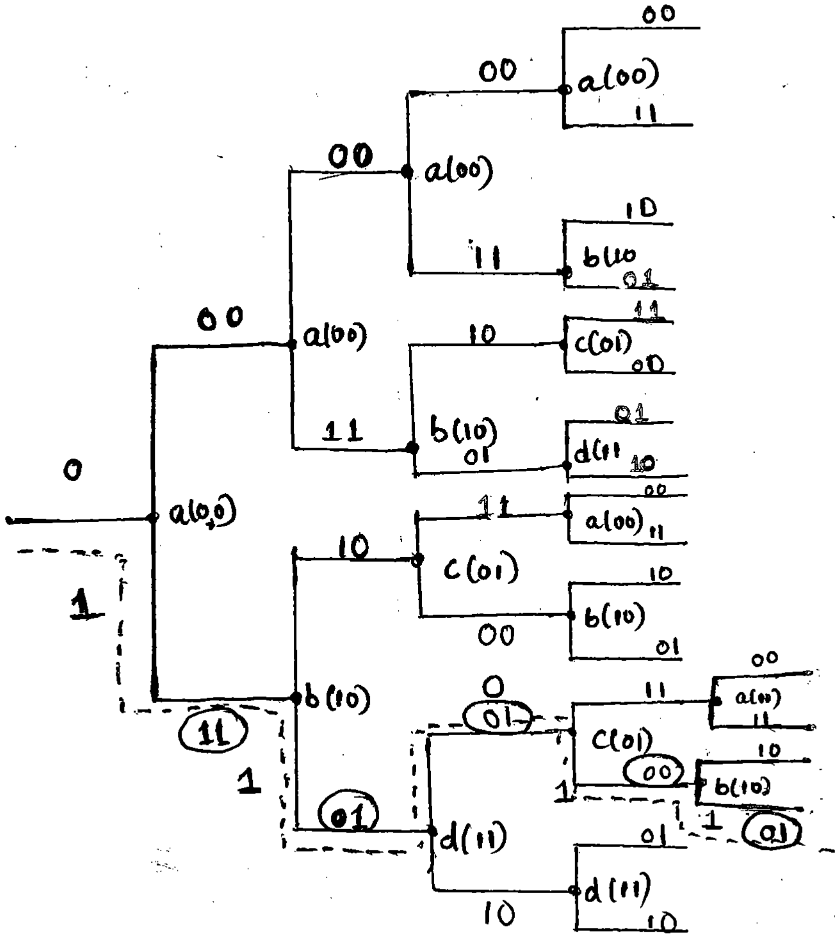


• • • • →



The given message sequence 1 1 0 1 1.

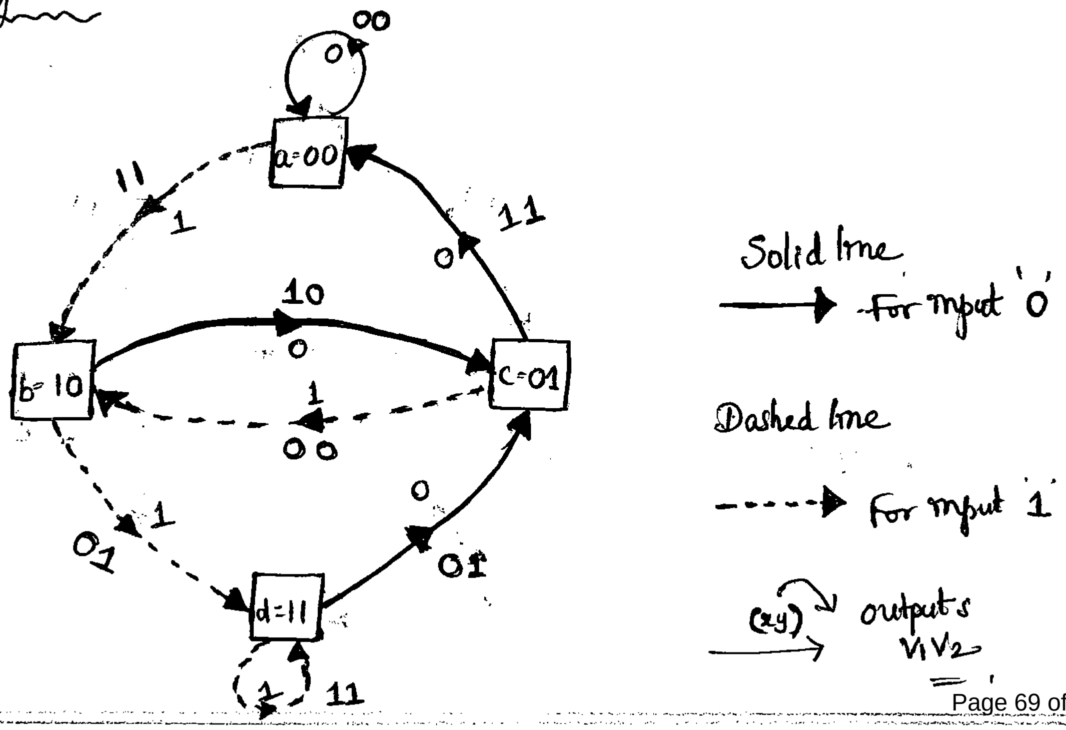
The tree diagram



The Encoded output for the given message 1 1 0 1 1.

(1 1 0 1 0 1 0 0 0 1)

(d) State diagram:



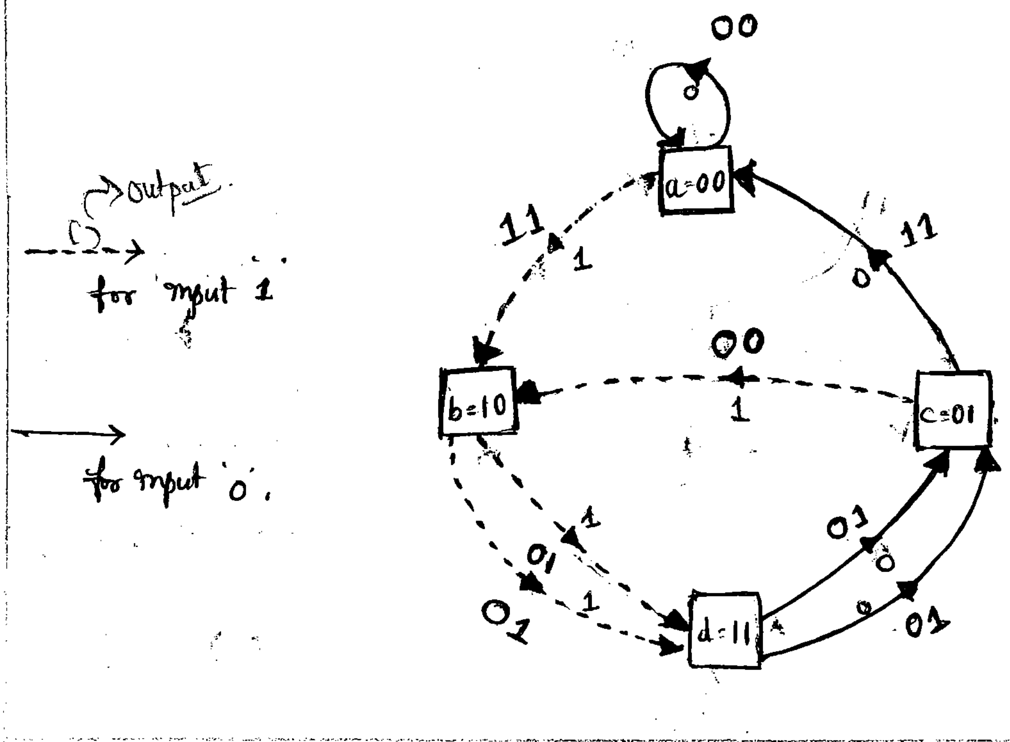
For the given message sequence 1 1 0 1 1, the state table is given by.

	Input	Registers			Present States		Future/next state		outputs	
		M_1	M_2	M_3	M_2	M_3	M_1	M_2	$V_1 = M_1 \oplus M_2 \oplus M_3$	$V_2 = M_1 + M_3$
	-	0	0	0	0	0	0	0	0	0
1.	1	1	0	0	0	0 a	1	0 b	1	1
2.	1	1	1	0	1	0 b	1	1 d	0	1
3.	0	0	1	1	1	1 d	0	1 c	0	1
4.	1	1	0	1	0	1 c	1	0 b	0	0
5.	1	1	1	0	1	0 b	1	1 d	0	1
6.	0	0	1	1	1	1 d	0	1 c	0	1
7.	0	0	0	1	0	1 c	0	0 a	1	1
8.	0	0	0	0	0	0 a	0	0 a	0	0

The Encoded output for the given message sequence is given by

(1 1 0 1 0 1 0 0 0 1 0 1 1 1 0 0).

State diagram for given message sequence. from above state stable.



(e) Trellis diagram:

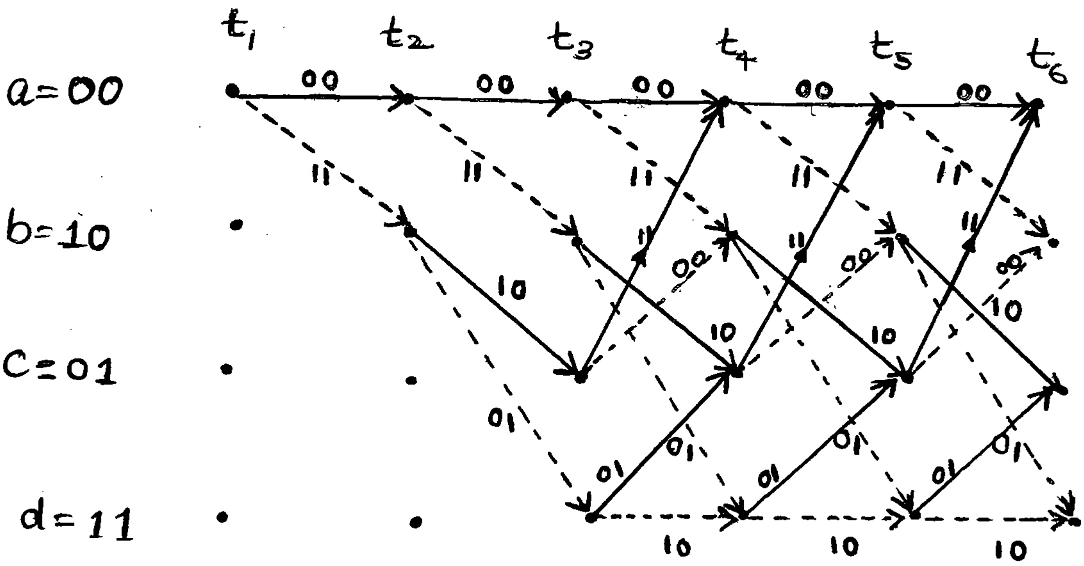
The number of time axis length is given as

$t_1, t_2, t_3, \ldots\ldots t_{L+1}.$

where L - no-of bits present in the messge sequence.

Trellis diagram can be obtained from state diagram.

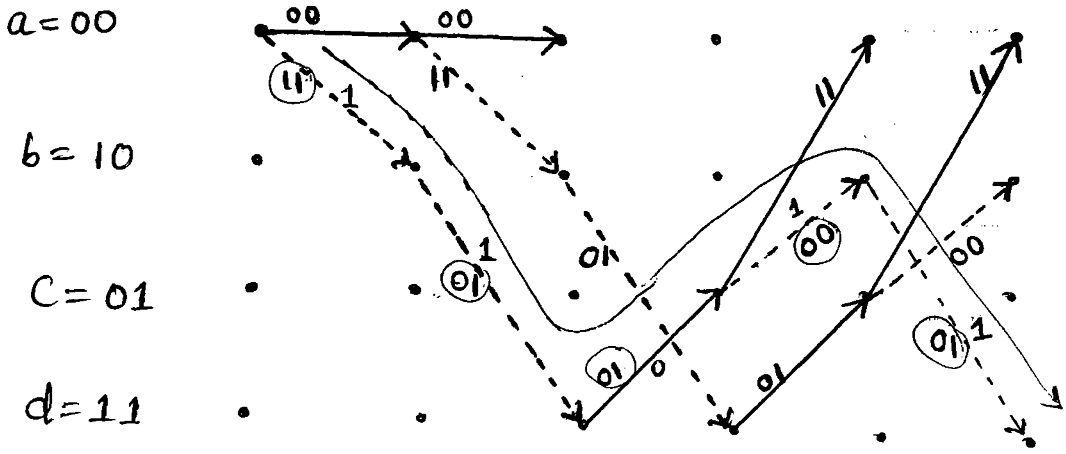
$2^{K(M-1)} = 2^{1(3-1)} = 2^2 = 4$ stages.



Solide line : ⟶ The output generated by an input bit '0'

Dashed line : - - - -➤ The output generated by an input bit '1'.

For the given messge sequene 11011 the trellis diagram using state diagram for the messge sequence.



∴ The Encoded output sequence : (11 01 01 00 01)

Decoding Methods of Convolutional Codes :

There are two methods for decoding of Convolutional Codes

1. Viterbi decoding.
2. Sequential decoding.

① Viterbi Decoding :

✓ The viterbi algorithm are currently used in about one million cellphones, which is probably the largest numbers in any application.

✓ The largest current consumer of viterbi algorithm processor cycle is probably digital video broadcasting.

✓ Now being decoded by the viterbi algorithm in digital TV sets around the world every second of every day.

✓ Now a days they are also used in bluetooth implementations.

✓ Viterbi algorithm essentially performs max-likelihood decoding.

* The algorithm involves calculating a measure of similarity or distance between the received signal at time 't' and all the trellis paths entering each state at time 't'.

✓ When two paths enter the same state, the one having the best metric is choosen. this path is called the "Serviving Path".

* The hamming distance between two code vectors is called "Metric".

ie The smallest path from transmitter to receiver is called "metric path" (or) "Serviving path".

① A (2,1,3) Convolutional code is described by $g_1 = (111)$ & $g_2 = (101)$. The output of the detector is '1101011001' using the viterbi algorithm. Find the transmitted data?

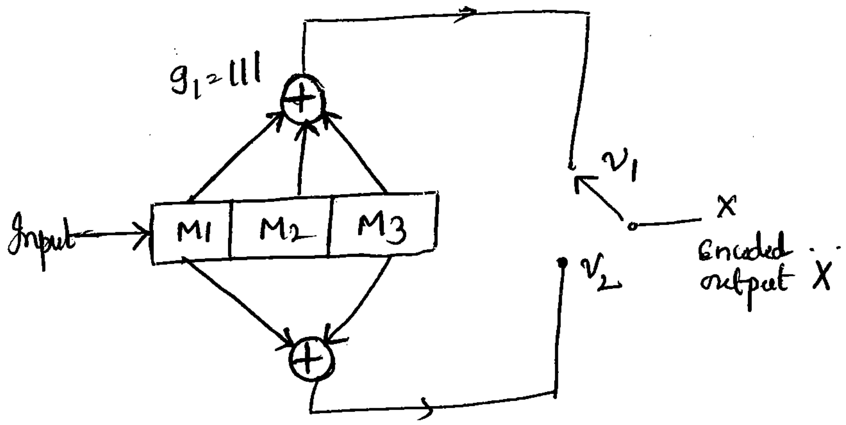
Sol

Given (2,1,3) = (n,K,M) Convolutional Code.

$n = 2$, $K = 1$, $M = 3$.

Given $g_1 = (111)$, $g_2 = (101)$

The encoder diagram as



$v_1 = M_1 \oplus M_2 \oplus M_3$

$v_2 = M_1 \oplus M_3$.

The state stable is given by.

The no. of states $2^{K(M-1)} = 2^2 = 4$ states

let $a = 00$, $b = 10$, $c = 01$, $d = 11$.

In general.

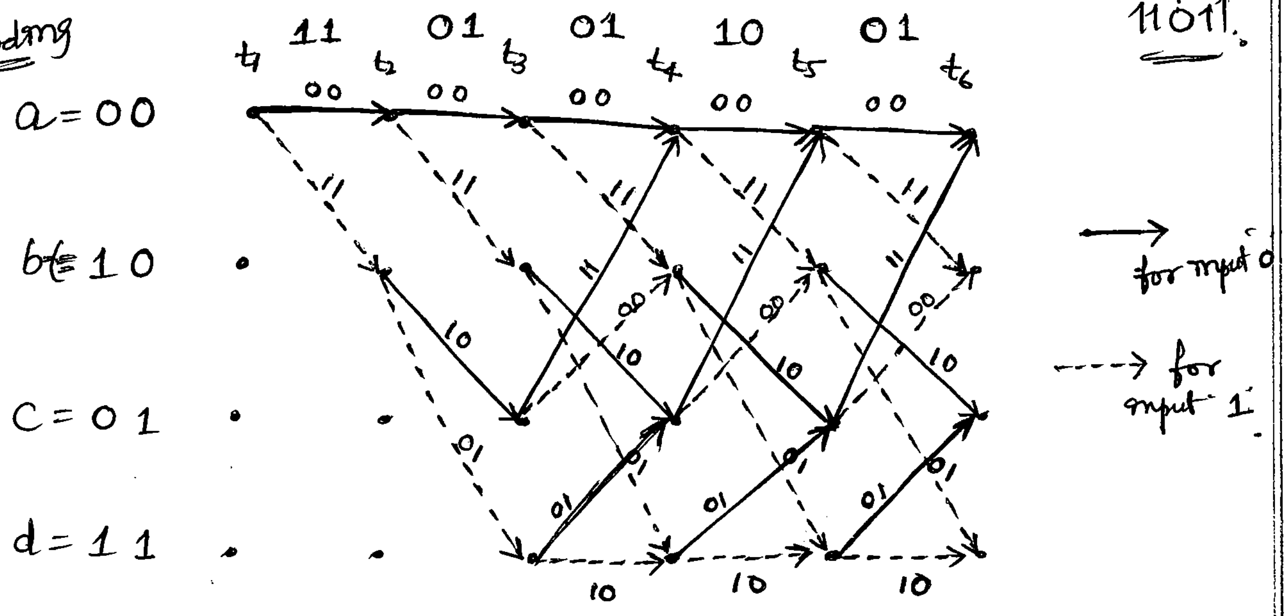
Input M_1	Present states M_2 M_3	Next states M_1 M_2	Outputs $v_1 = M_1 \oplus M_2 \oplus M_3$	$V_2 = M_1 \oplus M_3$.
0	0 0 a	0 0 a	0	0
1	0 0 a	1 0 b	1	1
0	0 1 c	0 0 a	1	1
1	0 1 c	1 0 b	0	0
0	1 0 b	0 1 c	1	0
1	1 0 b	1 1 d	0	1
0	1 1 d	0 1 c	0	1
1	1 1 d	1 1 d	1	0

(PTO)

Viterbi decoding can be obtained by using trellis diagram.

Trellis diagram: Let the no of time axis $t_1, t_2, \dots t_{L+1}$. i/p 11011.

Encoding

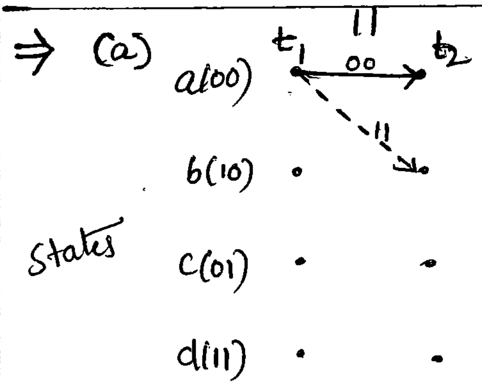


Viterbi decoding:

The output of the decoder is 1 1 0 1 0 1 1 0 0 1

The smallest path from transmitter to receiver is metric path.

⇒ (a)



$\gamma_a = 2$

$\gamma_b = 0$

State

(∵ $\gamma_a \rightarrow$ hamming distance

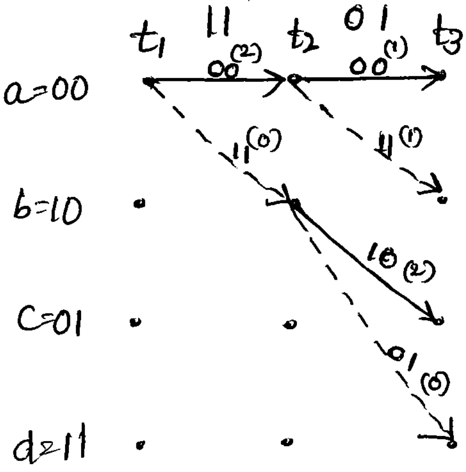
$\overset{11}{\downarrow\downarrow}_{00} \rightarrow \gamma_a = 2$

$\gamma_b \rightarrow \frac{11}{11} \rightarrow 0 \Rightarrow \gamma_b = 0$.

metric path.

Fig(a) Survivors at t_2

⇒ (b)



$\gamma_a = 2+1 = 3$

$\gamma_b = 2+1 = 3$

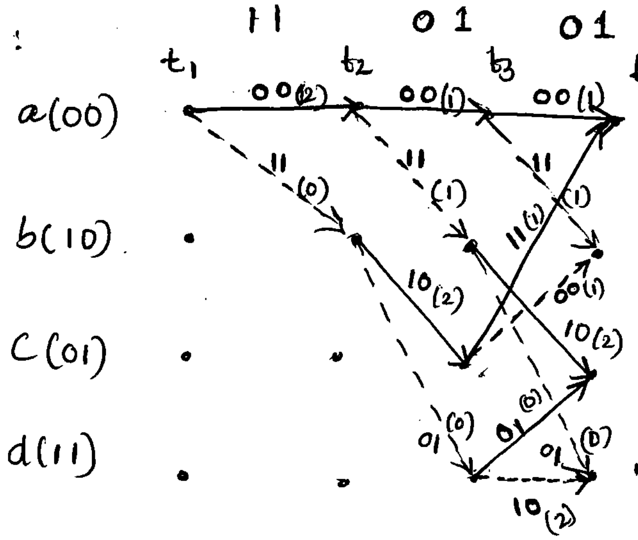
$\gamma_c = 0+2 = 2$

$\gamma_d = 0+0 = 0$

Fig: Survivors at t_3.

(c) :

⇒



$\gamma_a = 2+1+1 = 4$ (or)
$0+2+1 = 3$ ✓

$\gamma_b = 2+1+1 = 4$ (or)
$0+2+1 = 3$ ✓

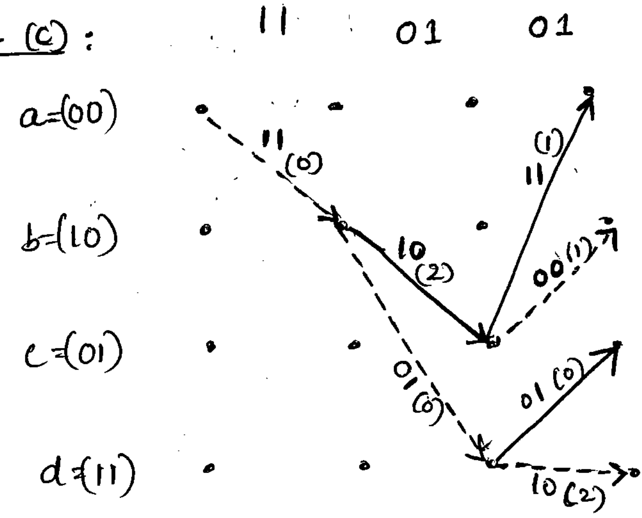
$\gamma_c = 2+1+2 = 5$ (or)
$= 0+0+0 = 0$ ✓

$\gamma_d = 2+1+0 = 3$
$= 0+0+2 = 2$ ✓

Fig: Metric components/comparison at t_4.

Modified (c) :

⇒



$\gamma_a = 0 + 2+1 = 3$

$\gamma_b = 0+2+1 = 3$

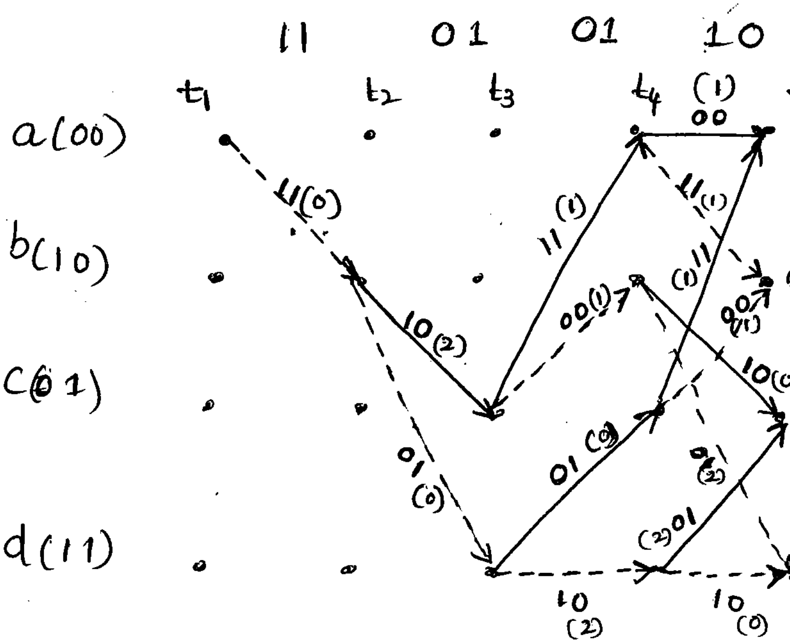
$\gamma_c = 0+0+0 = 0$

$\gamma_d = 0+0+2 = 2$

Fig: Survivors at t_4.

(d) :

⇒



$\gamma_a = 0+2+1+1 = 5$
(or) $0+0+0+1 = 1$ ✓

$\gamma_b = 0+2+1+1 = 4$
(or) $0+0+0+1 = 1$ ✓

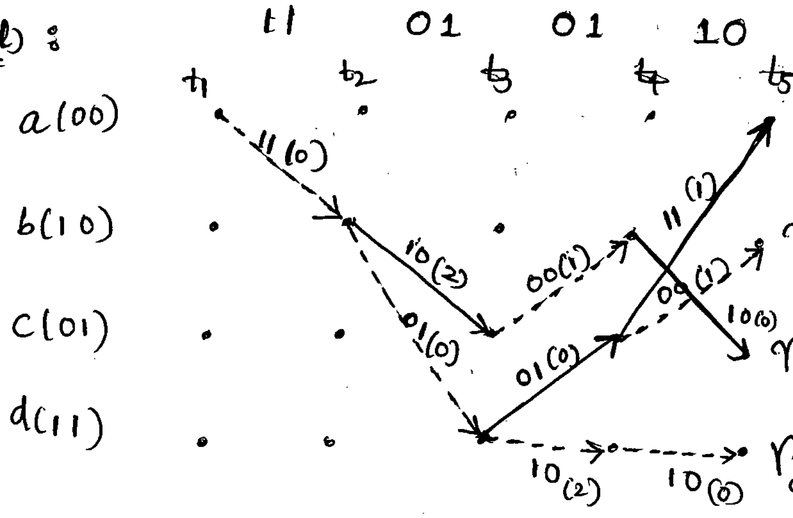
$\gamma_c = 0+2+1+0 = 3$ ✓
(or) $0+0+2+2 = 4$

$\gamma_d = 0+2+1+2 = 5$
(or) $0+0+2+0 = 2$ ✓

fig: Metric Component/Comparison at 't_5'

Modified (d) :

⇒



$\gamma_a = 0+0+0+1 = 1$

$\gamma_b = 0+0+0+1 = 1$

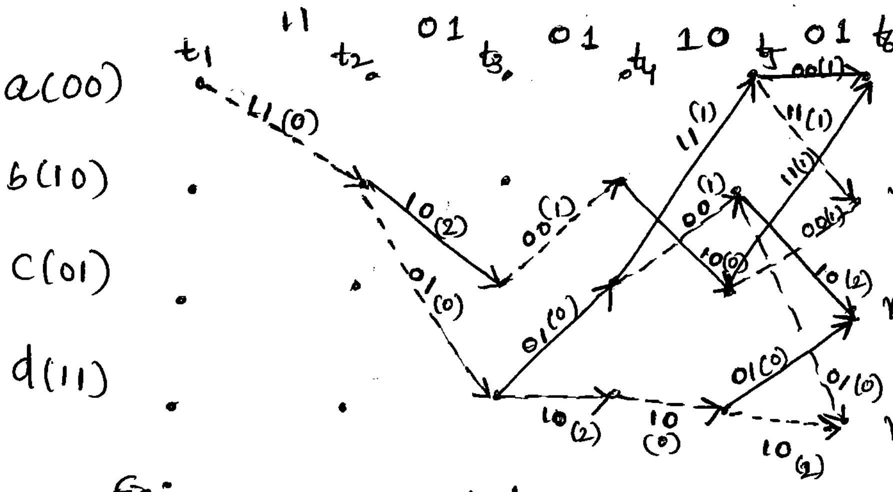
$\gamma_c = 0+2+1+0 = 3$

$\gamma_d = 0+0+2+0 = 2$

Fig : Survivors at t_5.

(e).

⇒



$\gamma_a = 0+0+0+1+1 = 2$ ✓
(or) $0+2+1+0+1 = 4$

$\gamma_b = 0+2+1+0+1 = 4$
(or) $0+0+0+1+1 = 2$ ✓

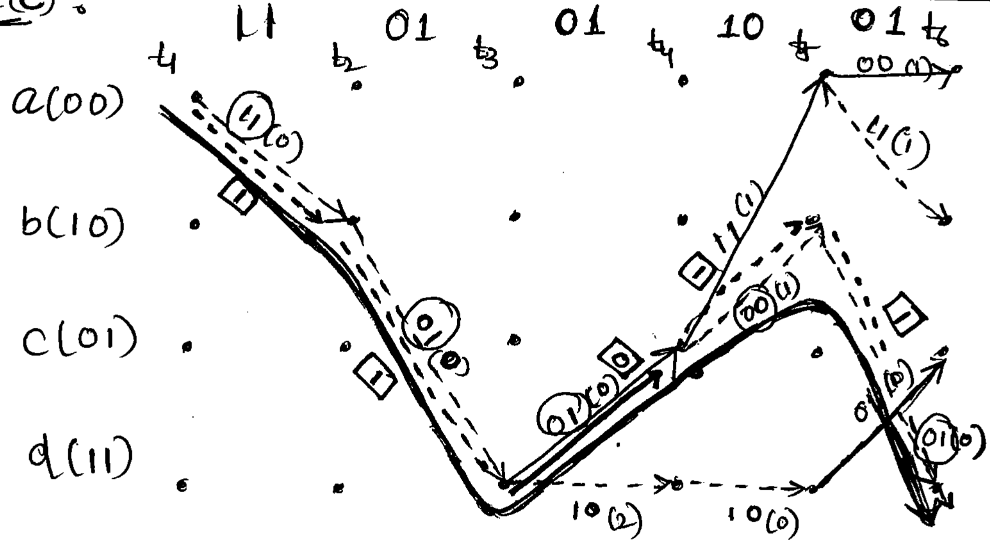
$\gamma_c = 0+0+0+1+2 = 3$
(or) $0+0+2+0+0 = 2$ ✓

$\gamma_d = 0+0+0+1+0 = 1$ ✓
(or) $0+0+2+0+2 = 4$

Fig: Metric Components / Comparison at t_6

Modified (e) :

⇒



$\gamma_a = 2$

$\gamma_b = 2$

$\gamma_c = 2$

$\gamma_d = ①$

Shortest path

Fig: Survivors at t_6.

✓ Among four state metric path values $\gamma_d = 1$ as a minimum length So, The path from t_1 to t_6 through γ_d is selected for decoding the message.

Messages <u>1 1 0 1 1</u>

∴ Transmitted Data : 1 1 0 1 0 1 0 0 0 1
Received Data : 1 1 0 1 0 1 1 0 0 1

Message 11011

② Sequential Decoding :

✓ Viterbi decoding algorithm for a convolutional code is maximum likelihood, the error performance of the algorithm over a discrete memoryless channel is optimum.

✓ The computational complexity of the algorithm is large.

✓ The constraint length 'N=M+1' is limited, so error correcting capability of the code is restricted.

Hence a decoding algorithm that avoids computing the likelihood & metric, of every path in the trellis, thereby reducing computational complexity and allowing the constraint length 'N' to take on very large values. there is a suboptimum class of such algorithm known as "Sequential decoding algorithm".

✓ The complexity of a sequential decoder is essentially ~~em~~ independent of the constraint length 'N' so that very large values of 'N' can be employed.

✓ Although sequential decoding algorithms are not quite as good as maximum likelihood decoding / viterbi decoding algorithms.

* Sequential decoding is an intuitive trial-and-error technique for searching out the correct path in a code tree.

✓ During the search, the decoder moves forward & backward in the code tree, one node at a time.

✓ The decision whether to move forward & backward is determined by the manner in which the metric of the algorithm varies along the path followed by the decoder.

(a) Fano Matric

(b) Fano Algorithm.

(a) Fano Metric :

Let us consider constraint length 'N'

Code rate $r = K/n$

Convolutional code over a discrete memory less channel.

let C_{ij} → i^{th} bit of binary label on the j^{th} branch of the code tree.

let y_{ij} → The corresponding bit of received sequence at the channel output.

∴ The bit metric is defined by

$$y_{ij} = \log_2\left(\frac{p(y_{ij}/c_{ij})}{p(y_{ij})}\right) - \mathcal{R} \longrightarrow ①$$

where $p(y_{ij}/c_{ij})$ → a transition probability of the channel.

$p(y_{ij})$ → The nomial (nominal) probability of the channel output.

If the binary label on each branch of the code tree has 'n' bits.

Hence bit metric becomes branch metric as

$$\therefore y_j = \sum_{i=1}^{n} y_{ij}$$

$$y_j = \sum_{i=1}^{n}\left[\log_2\left(\frac{p(y_{ij}/c_{ij})}{p(y_{ij})}\right) - \mathcal{R}\right] \longrightarrow ②$$

The Sequential decoder has followed a path with 'l' branches, starting from the origin of the code tree.

∴ The path metric (or) 'Fano metric' as

$$\gamma(l) = \sum_{j=1}^{l} y_j$$

$$\boxed{\gamma(l) = \sum_{i=1}^{n}\sum_{j=1}^{l}\left[\log_2\left[\frac{p(y_{ij}/c_{ij})}{p(y_{ij})}\right] - r\right]} \longrightarrow ③$$

where $l = 1, 2, \ldots\ldots$

(b) Fano Algorithm :

✓ In this the decoder moves forward (or) backward through the code tree always one node at a time.

✓ The decision whether or not the decoder favours a node is made by comparing the path metric at a node with a →

running threshold maintained by the decoder.

* The running threshold 'T' is defined as an integer multiple of the threshold spacing 'Δ', which is a design parameter.

- ✓ When 'Δ' is too small, the decoder frequently back tracks even when it is on the correct path.
- ✓ When 'Δ' is too large, the decoder incorrect turns are not identified quickly.

Thus a compromise choice of 'Δ' is required to minimize computation.

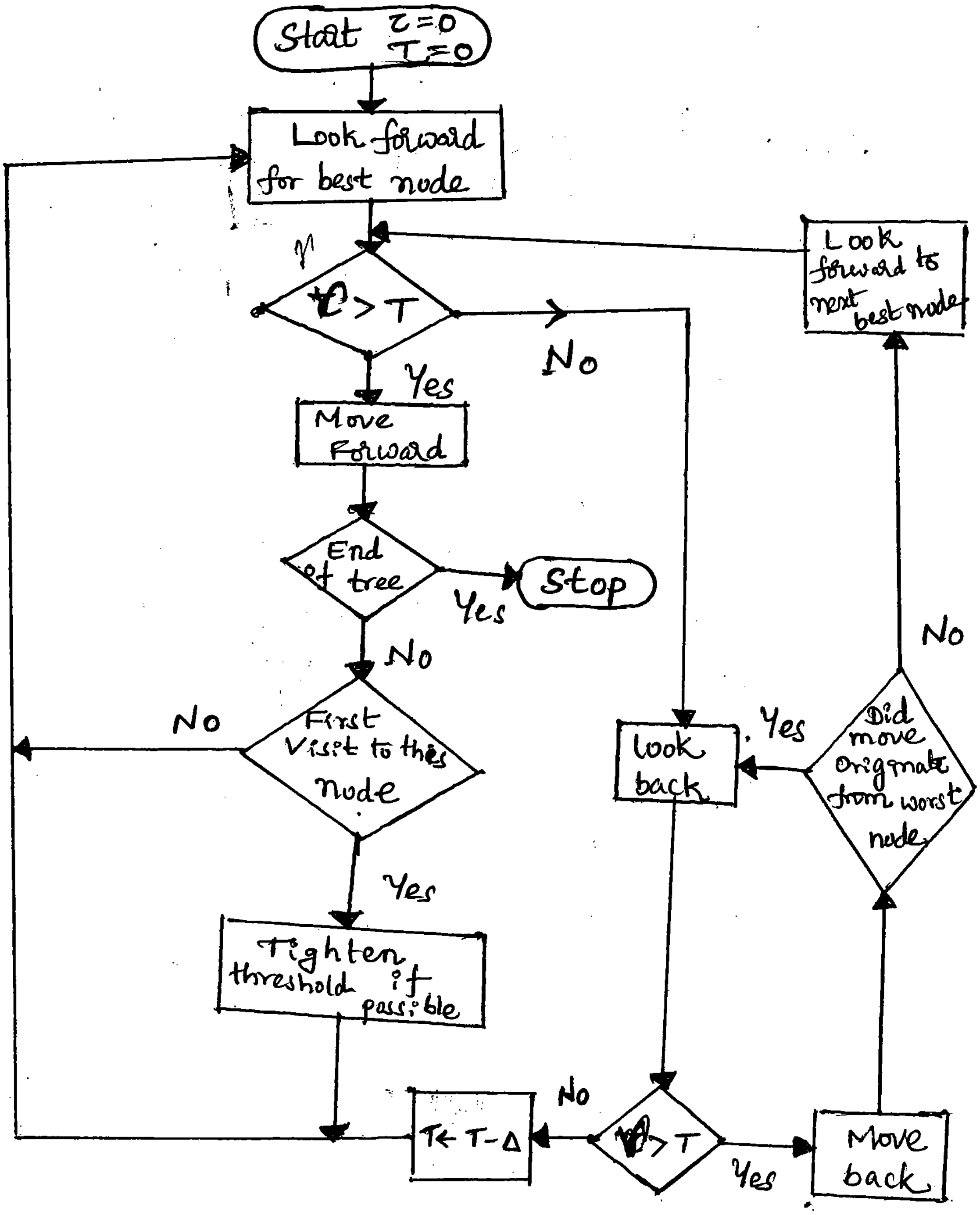


Fig: Flow chart for the Fano Algorithm.

Example :

Let us consider code tree diagram for Code rate = 1/2 & Constraint length '7' of Convolutional code.

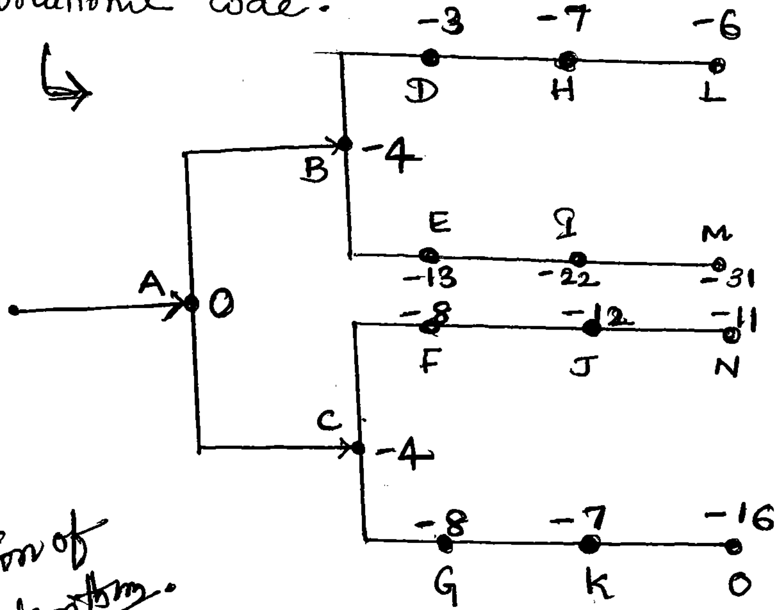


Illustration of Fano algorithm.

Step	Node	Fano Metric 'Γ(ℓ)'	Threshold 'T'
1	A	0	0
2	A	0	-4
3	C	-4	-4
4	A	0	-4
5	B	-4	-4
6	D	-3	-4
7	B	-4	-4
8	A	0	-4
9	A	0	-8
10	C	-4	-8
11	G	-8	-8
12	K	-7	-8
13	G	-8	-8
14	C	-4	-8
15	A	0	-8
16	B	-4	-8
17	D	-3	-8
18	H	-7	-8
19	L	-6	-8
20	← STOP →		

* The End