# UNIT – V

**Linear Block Codes and Convolution Codes: Matrix description of Linear Block codes, Syndrome Decoding, Error detection and error correction capabilities of Linear block codes; Binary Cyclic Codes- Algebraic structure, Systematic and Non Systematic form, Encoding, Syndrome calculation; Convolution Codes- Encoding of Convolution Codes- Graphical approach- State diagram, Code tree and Trellis diagram; Decoding of Convolution Codes- Viterbi decoding algorithm.**

**Introduction**

Coding theory is concerned with the transmission of data across noisy channels and the recovery of corrupted messages.

**Principle of block coding**

For the block of k message bits, (n-k) parity bits or check bits are added. Hence the total bits at the output of channel encoder are ‘n’. Such codes are called (n,k)block codes . Figure illustrates this concept.

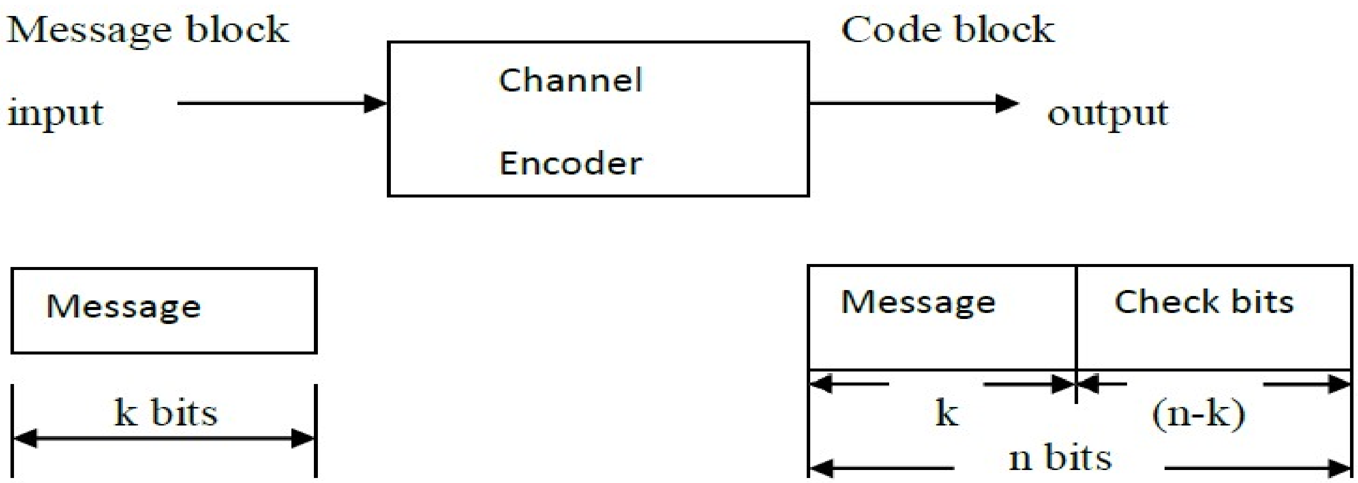



Figure: Functional block diagram of block coder

**Types of codes**

**Systematic codes:**

In the systematic block code, the message bits appear at the beginning of the codeword. The message appears first and then check bits are transmitted in a block. This type of code is called systematic code.

**Nonsystematic codes:**

In the nonsystematic block code it is not possible to identify the message bits and check bits.

- Linear Block Code
- Cyclic Code
- Convolutional Code

# Linear Block Codes

Information is divided into blocks of length ***k***

***r** parity bits or check bits are added to each block (total* length *n = k + r).*

**Code rate *R = k/n***

Decoder looks for codeword closest to received vector

(**received vector = code vector + error vector)**

Important Parameters :-

① Code word :- The encoded block of 'n' bits is called a "codeword"

Ex :- (7, 4) ⇒ (n, k) n = 7

② Block length :- The no of bits 'n' after encoding in codeword called Block length.

③. Code Rate | coding efficiency :- It is the ratio of message bits to Transmitted bits in a codeword ie

Coding efficiency | code Rate $r = \frac{K}{n}$

# Linear Block Codes

**A code is linear if the sum of any two code vectors produces another code vector.**

- A code is said to be linear if any two codewords in the code can be added in modulo-2 arithmetic to produce a third codeword in the code.
- Consider, then, an $(n,k)$ linear block code, in which $k$ message sequence bits of the $n$ *bit* code word.
- Accordingly, these $(n-k)$ bits are referred to as *parity-check bits*. Block codes in which the message bits are transmitted in unaltered form are called *systematic codes*.
- For applications requiring *both* error detection and error correction, the use of systematic block codes simplifies implementation of the decoder.

- Linear Block Code

The codeword block $C$ of the Linear Block Code is

$$C = m\,G$$

where $m$ is the information block, $G$ is the generator matrix.

$$G = [\mathbf{I}_k \mid \mathbf{P}]_{k \times n}$$

where $p_i$ = Remainder of $[x^{n-k+i-1}/g(x)]$ for $i=1, 2, .., k$, and $\mathbf{I}$ is unit matrix.

- The parity check matrix

$H = [\mathbf{P}^{\mathrm{T}} \mid \mathbf{I}_{n-k}]$, where $\mathbf{P}^{\mathrm{T}}$ is the transpose of the matrix $\mathbf{p}$.

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

Consider, then, an $(n,k)$ linear block code, in which $k$ message sequence bits of the *n bit* code word.

Accordingly, these $(n - k)$ bits are referred to as *parity-check bits*. Block codes in which the message bits are transmitted in unaltered form are called *systematic codes*.

For applications requiring *both* error detection and error correction, the use of systematic block codes simplifies implementation of the decoder.

- In $(n,k)$ linear block code, Let $m_0, m_1, \ldots, m_{k-1}$ constitute a block of $k$ arbitrary message bits.
- Thus, we have $2^k$ distinct message blocks.
- Let this sequence of message bits be applied to a linear block encoder, producing an $n$-bit codeword whose elements are denoted by $c_0, c_1, \ldots, c_{n-1}$.
- *Codeword* $c = [b_0, b_1, \ldots, b_{n-k-1}, m_0, m_1, \ldots, m_{k-1}]$
- Let $b_0, b_1, \ldots, b_{n-k-1}$ denote the $(n-k)$ *parity-check bits* in the codeword.
- Clearly, we have the option of sending the message bits of a codeword before the parity-check bits, or vice versa.

- The $(n - k)$ parity-check bits are *linear sums* of the $k$ message bits, as shown by the generalized relation
- $b_i = p_{0i}m_0 + p_{1i}m_1 + \dots + p_{k-1,i}m_{k-1}$
- $\mathbf{b} = \mathbf{m}\,\mathbf{P}$
- *parity coefficints* $p_{ij} = 1$ if $b_i$ depends on $m_j$
- $= 0$ otherwise
- $\boldsymbol{c} = [c_0, c_1, \dots, c_{n-1}]$
- *Codeword* $\boldsymbol{c} = [b_0, b_1, \dots, b_{n-k-1}, m_0, m_1, \dots, m_{k-1}]$
- *Message bits* $\boldsymbol{m} = [m_0, m_1, \dots, m_{k-1}]$
- Parity Bits $\mathbf{b} = [b_0, b_1, \dots, b_{n-k-1}]$

- The **P** is the $k$-by-$(n-k)$ *coefficient matrix* defined by

$$\mathbf{P} = \begin{bmatrix} p_{00} & p_{01} & \cdots & p_{0,\,n-k-1} \\ p_{10} & p_{11} & \cdots & p_{1,\,n-k-1} \\ \vdots & \vdots & & \vdots \\ p_{k-1,\,0} & p_{k-1,\,1} & \cdots & p_{k-1,\,n-k-1} \end{bmatrix}$$

- where the element $p_{ij}$ is 0 or 1.
- $\boldsymbol{c} = [\mathbf{b} \mid \mathbf{m}]$ $\quad$ $\mathbf{c} = [\mathbf{b} \mid \mathbf{m}]$
- $\boldsymbol{c} = \boldsymbol{m}\,[\mathrm{P} \mid \mathrm{I}_k]$
- Define the $k$-by-$n$ *generator matrix* $\quad \boldsymbol{G} = [\mathrm{P} \mid \mathrm{I}_k]$
- ***Codeword*** $\quad \boldsymbol{c} = \boldsymbol{m}\,\boldsymbol{G}$

- Let parity check bits **H** denote an $(n-k)$-by-$n$ matrix, defined as
- $\boldsymbol{H} = [I_{n-k} \mid P^T]$
- $\boldsymbol{G} H^T = [P \mid I_k] \{I$
- $P\}$
- $P + P = 0$
- $\boldsymbol{G} H^T = \boldsymbol{0}$ where **0** is a new null matrix.
- $C H^T = \boldsymbol{mG} H^T = \boldsymbol{0}$
- The generator matrix **G** is used in the encoding operation at the transmitter. On the other hand, the parity-check matrix **H** is used in the decoding operation at the receiver.

**r** denote the 1-by-*n* *received vector* that results from

$$\mathbf{r} = \mathbf{c} + \mathbf{e}$$

vector **e** is called the *error vector* or *error pattern*.

- The syndrome $S = r\,H^T$
- $S = (c+e)\,H^T = c\,H^T + e\,H^T$

  $S = e\,H^T$
- Hence, the parity-check matrix **H** of a code permits us to compute the syndrome **s**, which depends only upon the error pattern **e**.
- For a linear block code, the syndrome **s** is equal to the sum of those rows of the transposed parity-check matrix $\mathbf{H}^T$ where errors have occurred due to channel noise.
- The minimum distance of a linear block code is the smallest Hamming weight of the nonzero code vectors in the code.

* In matrix form

message bits $[M]_{1\times K} = [m_0, m_1, m_2 \,--\, m_{K-1}]_{1\times K}$. (K-bits)

Code words | codevector $[C]_{1\times n} = [c_0, c_1, c_2 \,---\, c_{K-1}]_{1\times n}$. (n-bits)

Generator Matrix $[G]_{K\times n} = \left[ I_K \;\vdots\; P_{K\times(n-K)} \right]_{K\times n}$.

Here $I_K$ = $K\times K$ Identity Matrix

$P_{K\times(n-K)}$ = Submatrix ~~of~~ ~~or~~ Parity check matrix

Ex:- The generator matrix for a (6,3) LBC is given below, find (6) all code words of this code.

The generator matrix for a (6,3) block code is given below. Find all the code vectors of this code.

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

$$\left[ 0\ 0\ 1 \ \vdots\ 1\ 1\ 0 \right]$$

$[I_3]$ $[P_{3\times3}]$ $3\times6$

* In modulo-2 arithmetic operation,
Even no 1's ⇒ 0
odd no 1's ⇒ 1

Sol:- $(n, k) = (6, 3)$

$n = 6 =$ Tx Bits
$k = 3 =$ message bits

I-Method:-

$$[D] = \begin{bmatrix} 000 \\ \vdots \\ 111 \end{bmatrix}$$

$$[C] = [D][G]$$

(i) $D_0 = [000]$

$$[C_0] = [0\,0\,0]\begin{bmatrix}1&0&0&0&1&1\\0&1&0&1&0&1\\0&0&1&1&1&0\end{bmatrix} = [0\oplus0\oplus0,\ 0\oplus0\oplus0,\ 0\oplus0\oplus0,\ 0\oplus0\oplus0,\ 0\oplus0\oplus0,\ 0\oplus0\oplus0]$$

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

$$[C_1] = [0\,0\,1,\ 1\,1\,0]$$

S|ly

$C_2 = [0\,1\,0\ \ 1\,0\,1]$ $\quad C_5 = [1\,0\,1\ \ 1\,0\,1]$

$C_3 = [0\,1\,1\ \ 0\,1\,1]$ $\quad C_6 = [1\,1\,0\ \ 1\,1\,0]$

$C_4 = [1\,0\,0\ \ 0\,1\,1]$ $\quad C_7 = [1\,1\,1\ \ 0\,0\,0]$

The generator matrix for a (6,3) block code is given below. Find all the code vectors of this code.

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

(n, k) = (6,3)
n = 6
k = 3
n – k = 6 - 3 =3 number of parity bits.

Separate the identity matrix and coefficient matrix
Generator matrix is given by:

$$[G] = [I_k \mid P]$$

$$\therefore I_k = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}_{3x3} \qquad P = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}_{3x3}$$

The generator matrix G of a (6,3) linear block code (LBC) is given as

$$G = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}_{3\times 6}$$

| d | C = dG |
|---|---|
| 000 | 000000 |
| 001 | 001101 |
| 010 | 010011 |
| 011 | 011110 |
| 100 | 100110 |
| 101 | 101011 |
| 110 | 110101 |
| 111 | 111000 |

| C | $i_0$ | $i_1$ | $i_2$ | $P_0$ | $P_1$ | $P_2$ |
|---|---|---|---|---|---|---|
| $C_1$ | | | | | | |
| $C_2$ | | | | | | |
| $C_3$ | | | | | | |

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

$$P_0 = (i_0 \oplus i_2)$$
$$P_1 = (i_1 \oplus i_2)$$
$$P_2 = (i_0 \oplus i_1)$$

# Linear Block Code

- In (7,4) linear block code code, The generator matrix G of this code can be taken as

$$\mathbf{G} = \left[\begin{array}{ccc|cccc} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{array}\right]$$
$$\underbrace{\qquad}_{\mathbf{P}} \qquad \underbrace{\qquad\qquad}_{\mathbf{I}_k}$$

- Parity-check matrix H is given by

$$\mathbf{H} = \left[\begin{array}{ccc|cccc} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{array}\right]$$
$$\underbrace{\qquad}_{\mathbf{I}_{n-k}} \qquad \underbrace{\qquad\qquad}_{\mathbf{P}^{\mathrm{T}}}$$

- Consider (7,4) Linear block code, the syndrome is given by: $S_0=r_0+r_3+r_5+r_6$

  $S_1=r_1+r_3+r_4+r_5$

  $S_2=r_2+r_4+r_5+r_6$
- Find Generator matrix (G) and Parity check matrix (H). Find all possible code vectors.
- Draw the corresponding encodin

- WKT
- Generator matrix $G=[I:P]_{k \times n}$
  - I is identity matrix, P is parity matrix
- Code word C=D.G
  - D- data, G- Generator Matrix
- $CH^T=0$
  - H is parity check matrix
- Syndrome $S=RH^T$
  - S- syndrome

- Given (n,k)=(7,4) then n=7, k=4 and n-k=3
- n= Total number of bits in the code word or code word length.
- K= message or data bits
- n-k=Extra bits/ parity check bits/Error contrc bits

- We have to find G, H, C.
- Given is syndrome equations
- $S_0=r_0+r_3+r_5+r_6$
- $S_1=r_1+r_3+r_4+r_5$
- $S_2=r_2+r_4+r_5+r_6$
- That is $H^T$

$$H^T=\begin{array}{c|ccc} & S_0 & S_1 & S_2 \\ r0 & \square & \square & \square \\ r1 & \square & \square & \square \\ r2 & \square & \square & \square \\ r3 & \square & \square & \square \\ r4 & \square & \square & \square \\ r5 & \square & \square & \square \\ r6 & \square & \square & \square \end{array}$$

- We have to find G, H, C for all.
- Given is syndrome equations
- $S_0=r_0+r_3+r_5+r_6$
- $S_1=r_1+r_3+r_4+r_5$
- $S_2=r_2+r_4+r_5+r_6$

$$\begin{array}{ccc} S_0 & S_1 & S_2 \end{array}$$
$$H^T=\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

Row -Column

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

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

- $H=[I_{n-k}:P^T]$
- $P=\begin{matrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{matrix}$

- P= $\begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$ WKT G=I:P

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

- C=D.G

- $C_0$= 0000. $\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$

- C=D.G

- $C_1$= 0001. $\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$

- = (0.1+0.0+0.0+1.0) =0 (MOD 2 Addition)
- Similarly do the remaining columns
- C1=0001 101

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

**Code words are → C= D G**

0000 000
0001 101
0010 111
0011 011
0100 011

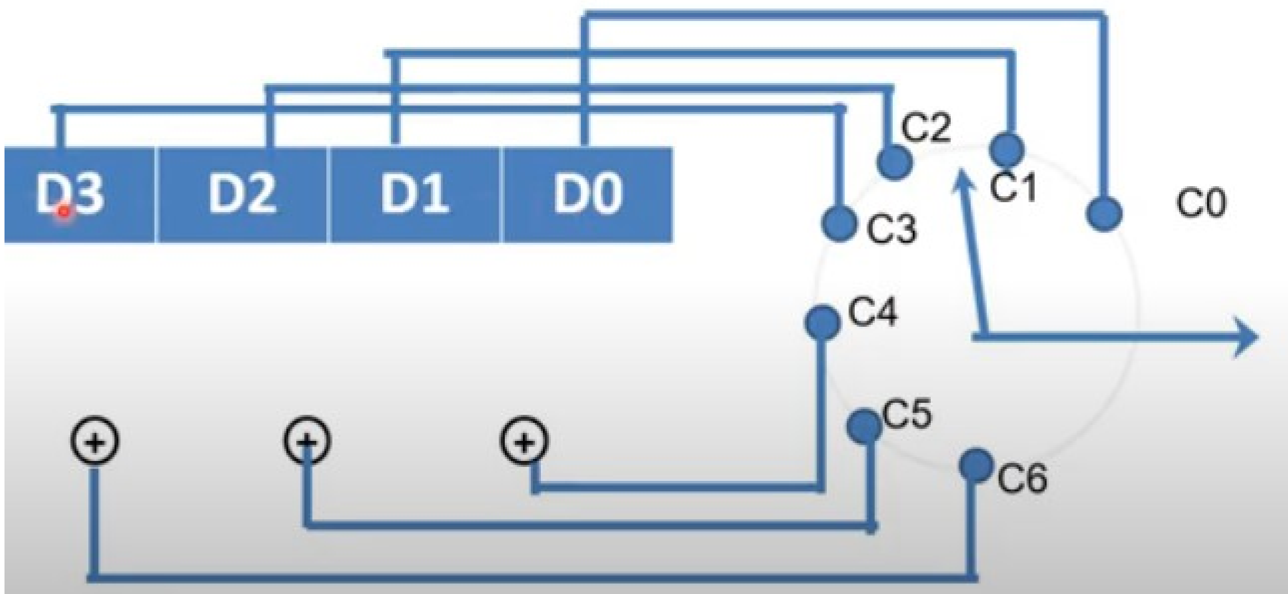
- 0111 001
- 1000 110
- 1001 011
- 1010 001
- 0101 110
- 0110 100

- 1011 100
- 1100 101
- 1101 000
- 1110 010
- 1111 111

$d_{min}$ =3=Min Hamming weight

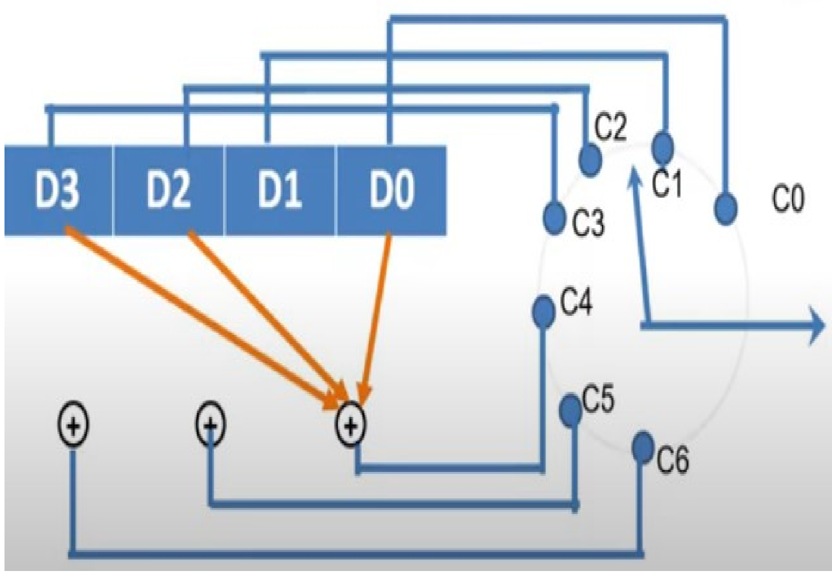
- Encoder circuit (7,4)

$$G=\begin{bmatrix}1&0&0&0&1&1&0\\0&1&0&0&0&1&1\\0&0&1&0&1&1&1\\0&0&0&1&1&0&1\end{bmatrix}\begin{matrix}D0\\D1\\D2\\D3\end{matrix}$$



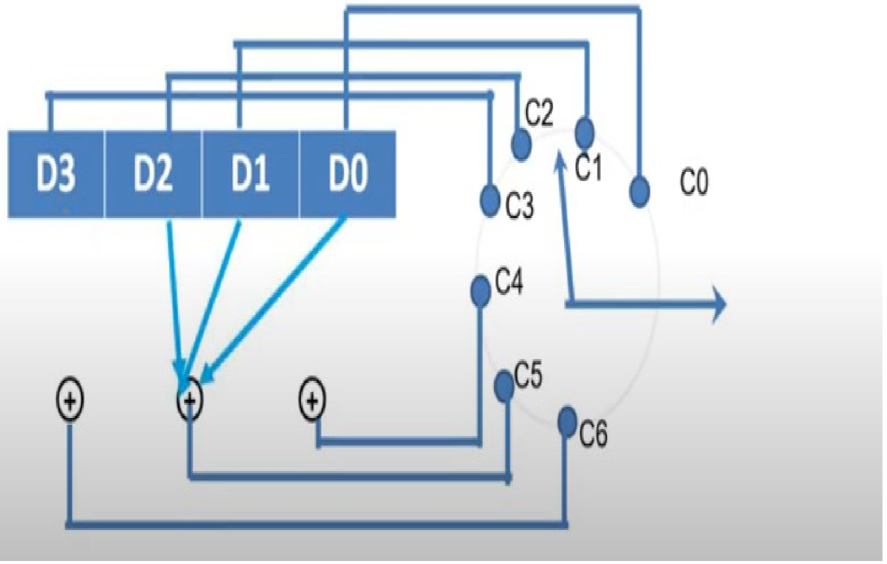
- Encoder circuit

$$G=\begin{bmatrix}1&0&0&0&1&1&0\\0&1&0&0&0&1&1\\0&0&1&0&1&1&1\\0&0&0&1&1&0&1\end{bmatrix}\begin{matrix}D0\\D1\\D2\\D3\end{matrix}$$



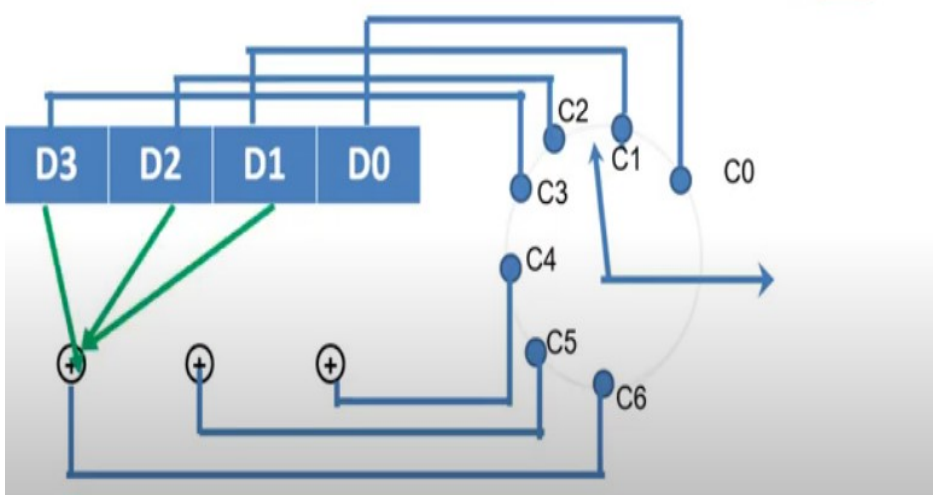

- Encoder circuit

$$G=\begin{bmatrix}1&0&0&0&1&1&0\\0&1&0&0&0&1&1\\0&0&1&0&1&1&1\\0&0&0&1&1&0&1\end{bmatrix}\begin{matrix}D0\\D1\\D2\\D3\end{matrix}$$



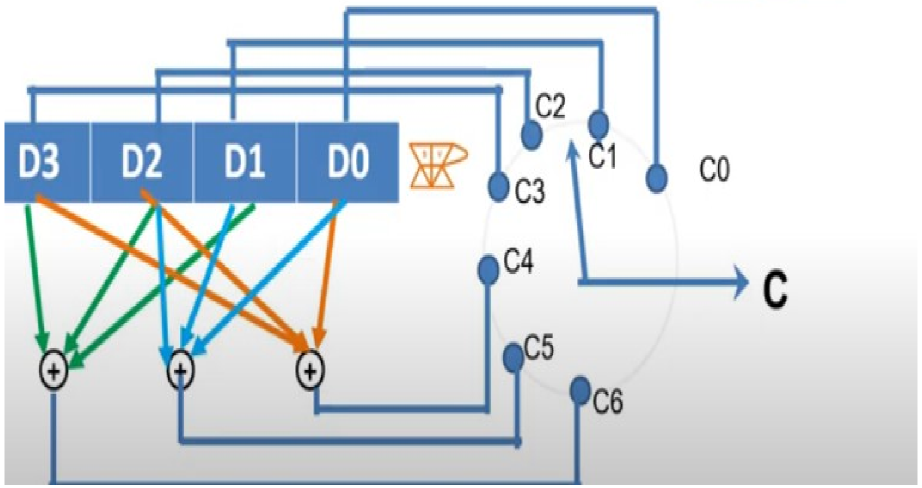
- Encoder circuit

$$G=\begin{bmatrix}1&0&0&0&1&1&0\\0&1&0&0&0&1&1\\0&0&1&0&1&1&1\\0&0&0&1&1&0&1\end{bmatrix}\begin{matrix}D0\\D1\\D2\\D3\end{matrix}$$



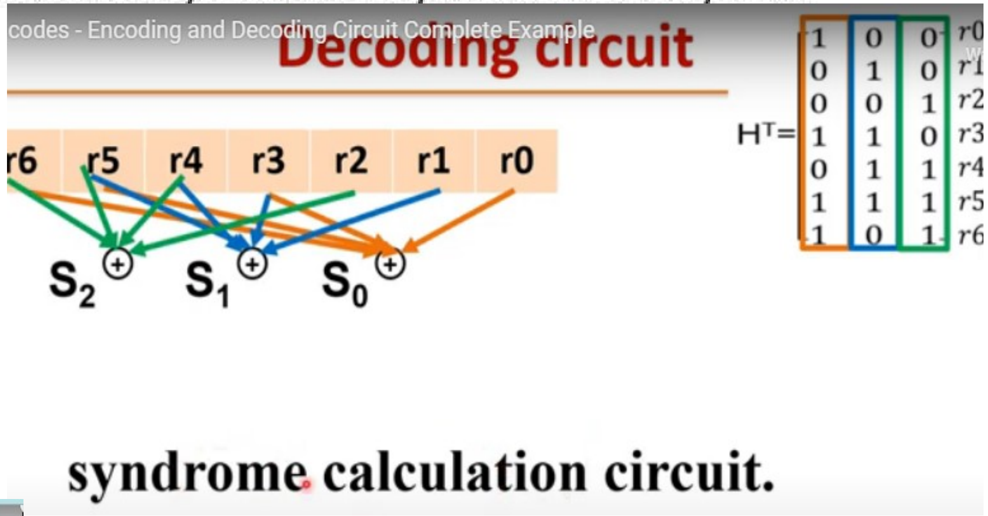

- Encoder circuit

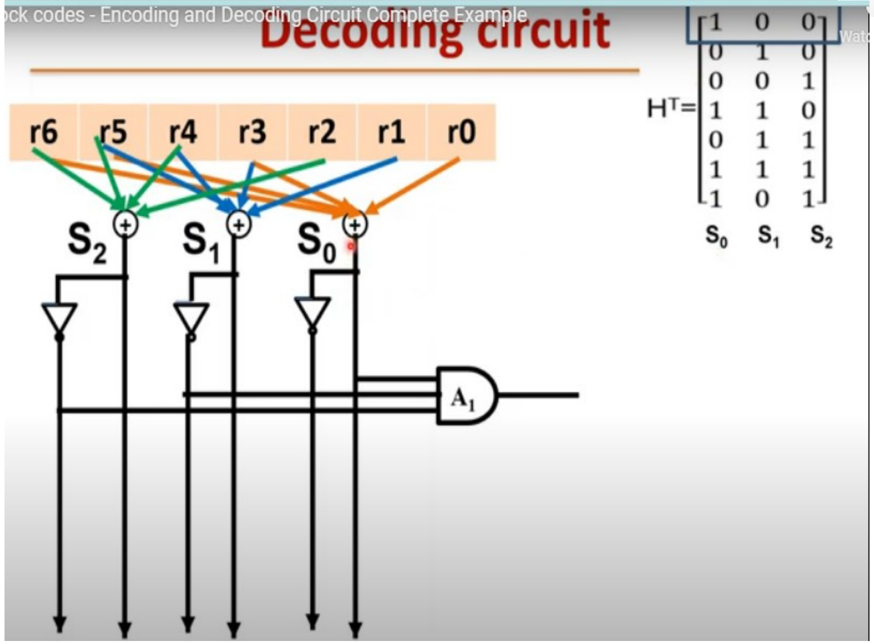
$$G=\begin{bmatrix}1&0&0&0&1&1&0\\0&1&0&0&0&1&1\\0&0&1&0&1&1&1\\0&0&0&1&1&0&1\end{bmatrix}\begin{matrix}D0\\D1\\D2\\D3\end{matrix}$$



- For Encoding circuit G was a reference or base
- Now for Decoding circuit reference/base is $H^T$

$$H^T=\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} \begin{matrix} r0 \\ r1 \\ r2 \\ r3 \\ r4 \\ r5 \\ r6 \end{matrix}$$



## Syndrome Decoding :-

* From The received code vector, [Y], calculate the syndrom vector [S], using $[S] = [Y][H^T]$ (or) $[S] = [H][Y^T]$

* If $[S] = 0$, Then we can say there are no errors in The Received code word, if $[S] \neq 0$, Then There is an error in received code vector.

* In Syndrome decoding, all the rows of $[H^T]$/ columns of $[H]$, are equal to The Syndrome vectors, If The Syndrome vector is the $i^{th}$ row of $[H^T]$/ $i^{th}$ column of $[H]$, Then There will be an error in the $i^{th}$ bit position of recieved code vector.

* The error corrected and decoded vector can be obtained by using XOR operation as $[X] = [Y] \oplus [E]$, where $[E]$ is error vector corresponding to The error bit position.

**Cyclic Code** is known to be a subclass of linear block codes where cyclic shift in the bits of the codeword results in another codeword. It is quite important as it offers easy implementation and thus finds applications in various systems.

Cyclic codes form a subclass of linear block codes

A binary code is said to be a *cyclic code* if it exhibits two fundamental properties

**Linearity Property**

*The sum of any two codewords in the code is also a codeword.*

**Cyclic Property**

*Any cyclic shift of a codeword in the code is also a codeword.*

Let the $n$-tuple denote a codeword of an linear block code.

$$\boldsymbol{c} = [c_0, c_1, \ldots, c_{n-1}]$$

The code is a cyclic code if the $n$-tuples are all codewords in the code

$$\boldsymbol{c} = [c_0, c_1, \ldots, c_{n-1}]\ , \boldsymbol{c} = [c_{n-1}, c_0, \ldots, c_{n-2}]\ , \boldsymbol{c} = [c_{n-2}, c_{n-1}, \ldots, c_{n-3}]$$

- To develop the algebraic properties of cyclic codes, we use the elements of a codeword to define the code polynomial

- $\boldsymbol{c(X)} = c_0 + c_1 \mathrm{X} + c_1 \mathrm{X}^2 + \ldots + c_{n-1}\mathrm{X}^{n-1}$

- where $X$ is an indeterminate. Naturally, for binary codes, the coefficients are 1's and 0's.
- Each power of $X$ in the polynomial represents a one-bit *shift* in time.
- Hence, multiplication of the polynomial by $X$ may be viewed as a shift to the right

- each code polynomial in the code can be expressed in the form of a polynomial product as follows
- $c(X) = a(X)\, g(X)$
- $a(X)$ is a polynomial in $X$ with degree $k-1$
- $g(X)$ is a polynomial in $X$ with degree $n-k$
- Let the k bit *message polynomial* be defined by
- $m(X) = m_0 + m_1 X + m_2 X^2 + \dots + m_{k-1} X^{k-1}$
- $b(X) = b_0 + b_1 X + b_2 X^2 + \dots + b_{n-k-1} X^{n-k-1}$

$$(\underbrace{b_0, b_1, \dots, b_{n-k-1}}_{n-k \text{ parity-check bits}}, \underbrace{m_0, m_1, \dots, m_{k-1}}_{k \text{ message bits}})$$

- **Generator Polynomial**
- The polynomial $X^n + 1$ and its factors play a major role in the generation of cyclic codes.
- Let **g**(*X*) be a polynomial of degree *(n-k)* that is a factor of $X^n + 1$;
- In general, **g**(*X*) may be expanded as follows:
- The polynomial **g**(*X*) is called the *generator polynomial* of a cyclic code.

$$\mathbf{g}(X) = 1 + \sum_{i=1}^{n-k-1} g_i X^i + X^{n-k}$$

where the coefficient $g_i$ is equal to 0 or 1 for $i = 1, 2, \ldots n-k-1$

$\boldsymbol{c(X)} = b\,(X) + m(X)\,X^{n-k}$

$\boldsymbol{a(X)g(X)} = b\,(X) + m(X)\,X^{n-k}$

the polynomial **b**($X$) is the remainder left over after dividing $m(X)\,X^{n-k}$ by **g**($X$).

Apply modulo-2 addition

$$\frac{X^{n-k}\mathbf{m}(X)}{\mathbf{g}(X)} = \mathbf{a}(X) + \frac{\mathbf{b}(X)}{\mathbf{g}(x)}$$

**Step1** Premultiply the message polynomial **m**($X$) by $\boldsymbol{X^{n-k}}$

**Step 2:** Divide $m(X)\,X^{n-k}$ by the generator polynomial **g**($X$), obtaining the remainder **b**($X$).

**Step 3:** Add **b**($X$) to $m(X)\,X^{n-k}$ obtaining the code polynomial **c**($X$).

- **Parity-Check Polynomial**
- An $(n,k)$ cyclic code is uniquely specified by its generator polynomial $\mathbf{g}(X)$ of order $(n-k)$.

$$\mathbf{g}(X) = 1 + \sum_{i=1}^{n-k-1} g_i X^i + X^{n-k}$$

- Such a code is also uniquely specified by another polynomial of order $k$, which is called the *parity-check polynomial H(X),* defined by

$$\mathbf{h}(X) = 1 + \sum_{i=1}^{k-1} h_i X^i + X^k$$

- The generator polynomial **g**($X$) is equivalent to the generator matrix **G**.
- Correspondingly, the parity-check polynomial **h**($X$) is an equivalent representation of the parity-check matrix **H**.
- We thus find that the matrix relation $\mathbf{HG}^T = \mathbf{0}$ or $\mathbf{GH}^T = \mathbf{0}$
- $\mathbf{g}(X)\mathbf{h}(X) \bmod (X^n + 1) = \mathbf{0}$
- The generator polynomial **g**($X$) and the parity-check polynomial **h**($X$) are factors of the polynomial $X^n + 1$ , as shown by
- $\mathbf{g}(X)\mathbf{h}(X) = (X^n + 1)$

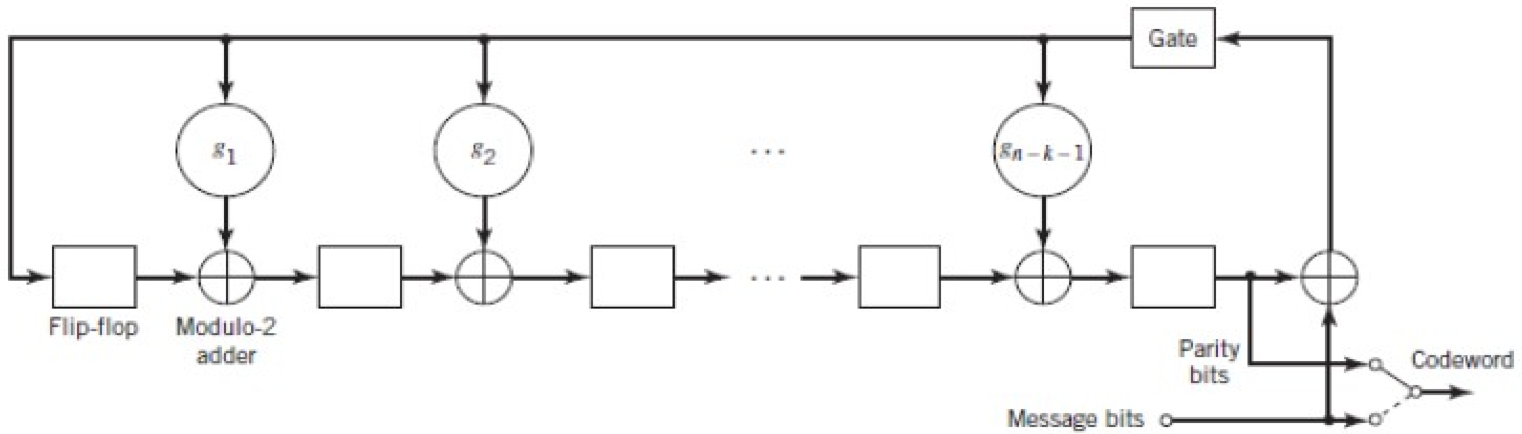
- (7,4) cyclic code message m = 1011, $g(X) = (1 + X + X^3)$
- $m(X) = (1 + X^2 + X^3)$

$$\frac{X^{n-k}\mathbf{m}(X)}{\mathbf{g}(X)} = \mathbf{a}(X) + \frac{\mathbf{b}(X)}{\mathbf{g}(x)}$$

$$\begin{array}{r|l|l}
X^3 + X + 1 & X^6 + X^5 + X^3 & X^3 + X^2 + X + 1 \\
 & X^6 + X^4 + X^3 & \\
 & \text{-----------------} & \\
 & X^5 + X^4 & \\
 & X^5 + X^3 + X^2 & \\
 & \text{-----------------} & \\
 & X^4 + X^3 + X^2 & \\
 & X^4 + X^2 + X & \\
 & \text{-----------------} & \\
 & X^3 + X & \\
 & X^3 + X + 1 & \\
 & \text{-----------------} & \\
 & 1 & \\
 & b(X) &
\end{array}$$

- $c(X) = 1 + X^3 + X^5 + X^6$
- C = [ 100 1011]

- The above three steps can be implemented by means of the encoder shown below consisting of a *linear feedback shift register* with *(n-k)* stages.
- The gate is switched on. Hence, the *k* message bits are shifted into the channel. As soon as the *k* message bits have entered the shift register, the resulting bits in the register form the parity-check bits.




Encoder for an $(n,k)$ cyclic code.

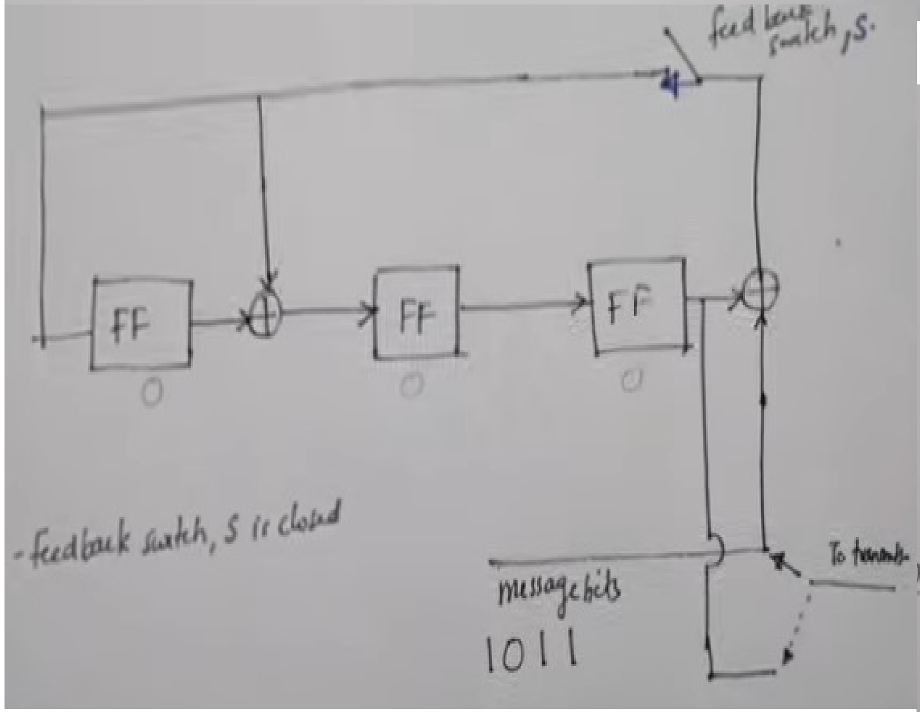
## Encoder for (7,4) cyclic

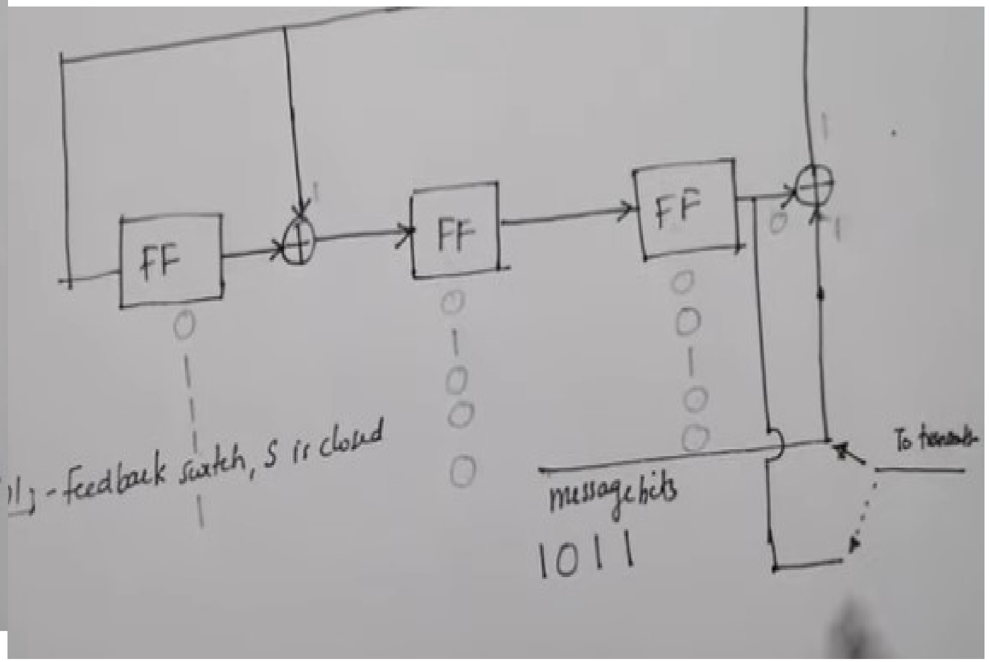
$$g(x) = 1 + \sum_{i=1}^{n-k-1} g_i x^i + x^{n-k}$$

$g(x) = x^3 + x + 1$

$$g(x) = 1 + g_1 x + g_2 x^2 + x^3$$

$$\boxed{g_1 = 1,\ g_2 = 0}$$ for this



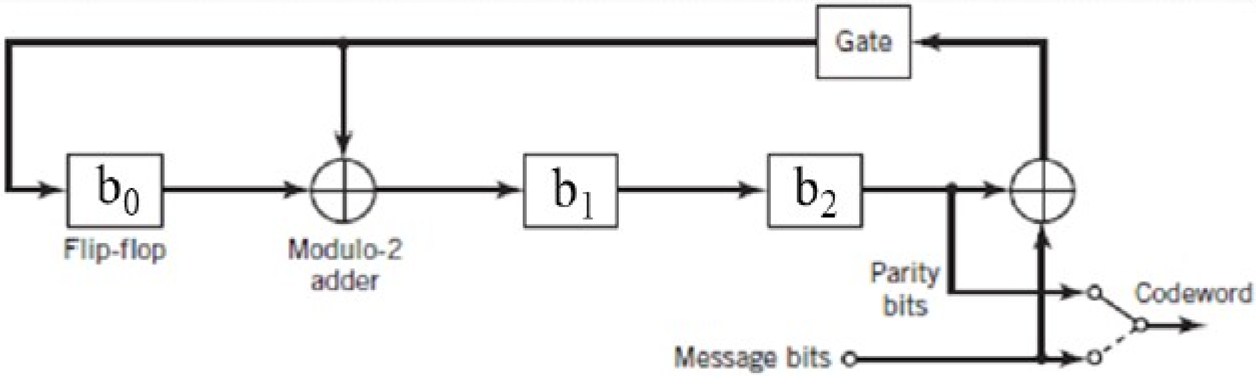



# Example Cyclic Code

| m | $b_0$ | $b_1$ | $b_2$ |
|---|---|---|---|
| | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 |

- Message m = [1011]
- $b_0 = m + b'_2$ $\quad b_1 = b'_0 + m + b'_2$ $\quad b_2 = b'_1$
- m $b_0$ $b_1$ $b_2$
- 0 0 0
- 1 1 1 0
- 1 1 0 1
- 0 1 0 0
- 1 1 0 0

- C = [ 100 1011]




Encoder for the (7,4) cyclic code generated by $g(X) = 1 + X + X^3$.

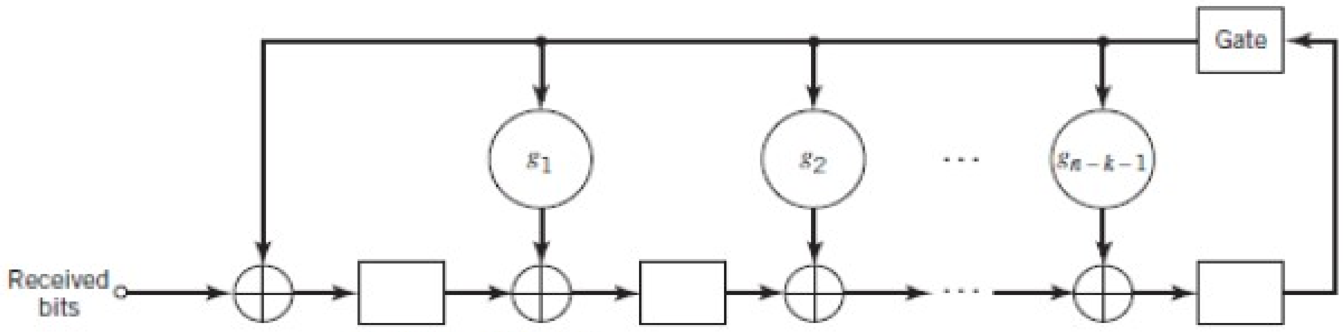
## Calculation of the Syndrome

- Suppose the codeword is transmitted over a noisy channel, resulting in the received word $\boldsymbol{r} = [r_0, r_1, \ldots, r_{n-1}]$.
- If the syndrome is zero, there are no transmission errors in the received word.
- If, on the other hand, the syndrome is nonzero, the received word contains transmission errors that require correction.
- ***Received codeword r(X)*** $= r_0 + r_1 X + r_1 X^2 + \ldots + r_{n-1}X^{n-1}$
- Let $\mathbf{q}(X)$ denote the quotient and $\mathbf{s}(X)$ denote the remainder, which are the results of dividing $\mathbf{r}(X)$ by the generator polynomial $\mathbf{g}(X)$.
- $$r(X) = q(X)g(X) + s(X)$$
- $$r(X)/g(X) = q(X) + s(X)/g(X)$$

A *syndrome calculator* that is identical to the encoder except for the fact that the received bits are fed into the (n-k) stages of the feedback shift register from the left.

As the received bits are fed into the shift register, initially set to zero.

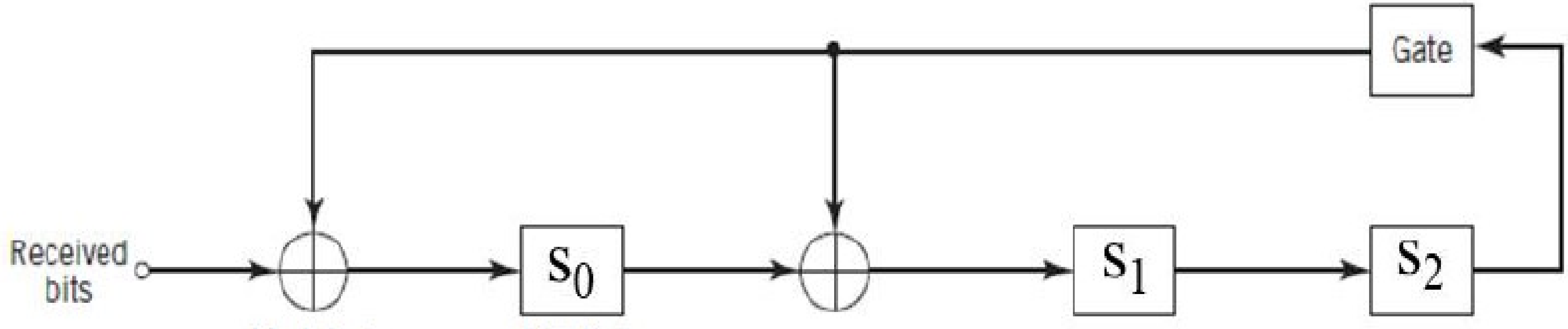
At the end of the $n^{th}$ shift, the syndrome is identified from the contents of the shift register. Since the syndrome is nonzero, the received word is in error




Syndrome computer for (n, k) cyclic code.

# Syndrome Encoder

- Syndrome Encoder for $g(X) = (1 + X + X^3)$
- $s_0\ s_1\ s_2$

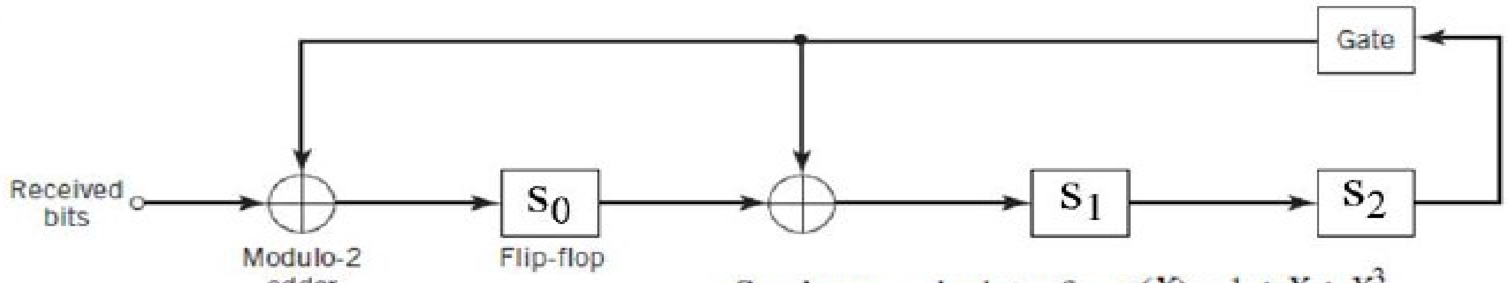



Syndrome calculator for $g(X) = 1 + X + X^3$.

# Example Cyclic code Syndrome

- Message m = [1011]
- Code word C = [ 100 1011]
- Received code r = [ 100 0011]
- $s_0 = r + s'_2$ $\quad s_1 = s'_0 + s'_2$ $\quad s_2 = s'_1$
- r $s_0\ s_1\ s_2$
- 0 0 0
- 1 1 0 0
- 1 1 1 0
- 0 0 1 1
- 0 1 1 1
- 0 1 0 1
- 0 1 0 0
- 1 1 1 0

r $s_0\ s_1\ s_2$
0 0 0
1 1 0 0
1 1 1 0
0 0 1 1
0 1 1 1
0 1 0 1
0 1 0 0
1 1 1 0




Syndrome calculator for $g(X) = 1 + X + X^3$.

The syndrome S = [110] non zero syndrome $r \neq c$

# Cyclic Code generator matrix

- To construct the 4-by-7 generator matrix **G**
- we start with four vectors represented by $\mathbf{g}(X)$ and three cyclic-shifted versions of it.
- For $g(X) = (1 + X + X^3)$
- $X\,g(X) = (X + X^2 + X^4)$
- $X^2\,g(X) = (X^2 + X^3 + X^5)$
- $X^3\,g(X) = (X^3 + X^4 + X^6)$
- If the coefficients of these polynomials are used as the elements of the rows of a 4-by-7 matrix, we get the following generator matrix which is not systematic

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

- We can put it into a systematic form by adding the first row to the third row, and adding the sum of the first two rows to the fourth row. Then G is

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

- ***As G is*** $\quad G = [P \mid I_k]$
- Parity-check matrix **H**, we get $\quad H = [I_{n-k} \mid P^T]$

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

- $C H^T = mG H^T = 0$
- The syndrome $S = r H^T$
- $S = (c+e) H^T = c H^T + e H^T$

  $S = e H^T$

- If syndrome is null then error is zero, so code word is received word
- $c = r$
- If syndrome is not null then there is an error, so code word is received word added with error pattern
- $c = r + e$

- Code word C = [ 100 1011]

Received code r = [ 100 0011]

r = [ 100 0011]

- The syndrome $S = r\,H^T$
- The syndrome S = [110]
- For Error pattern e = [0001000]
- $S = e\,H^T$ =[110]
- Then Code word c = r + e

Received code r = [ 100 0011]

Error Pattern e = [ 000 1000]

- Code word C = [ 100 1011]

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

$$H^T = \begin{vmatrix} 100 \\ 010 \\ 001 \\ 110 \\ 011 \\ 111 \\ 101 \end{vmatrix}$$

# Convolutional Codes

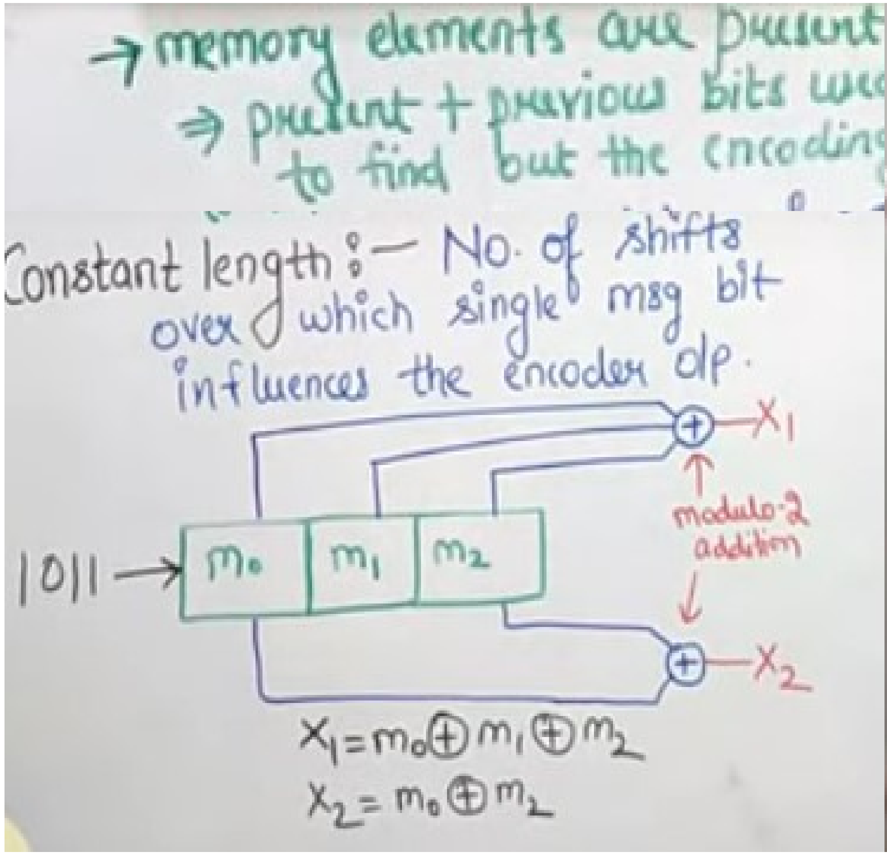
- In block coding, the encoder accepts a *k*-bit message block and generates an *n*-bit codeword, which contains $n - k$ parity-check bits. Thus, codewords are produced on a block-by-block basis. Clearly, provision must be made in the encoder to buffer an entire message block before generating the associated codeword.
- There are applications, however, where the message bits come in *serially* rather than in large blocks, in which case the use of a buffer may be undesirable. In such situations, the use of *convolutional coding* may be the preferred method.
- A convolutional coder generates redundant bits by using *modulo-2 convolutions*; hence the name *convolutional codes*.

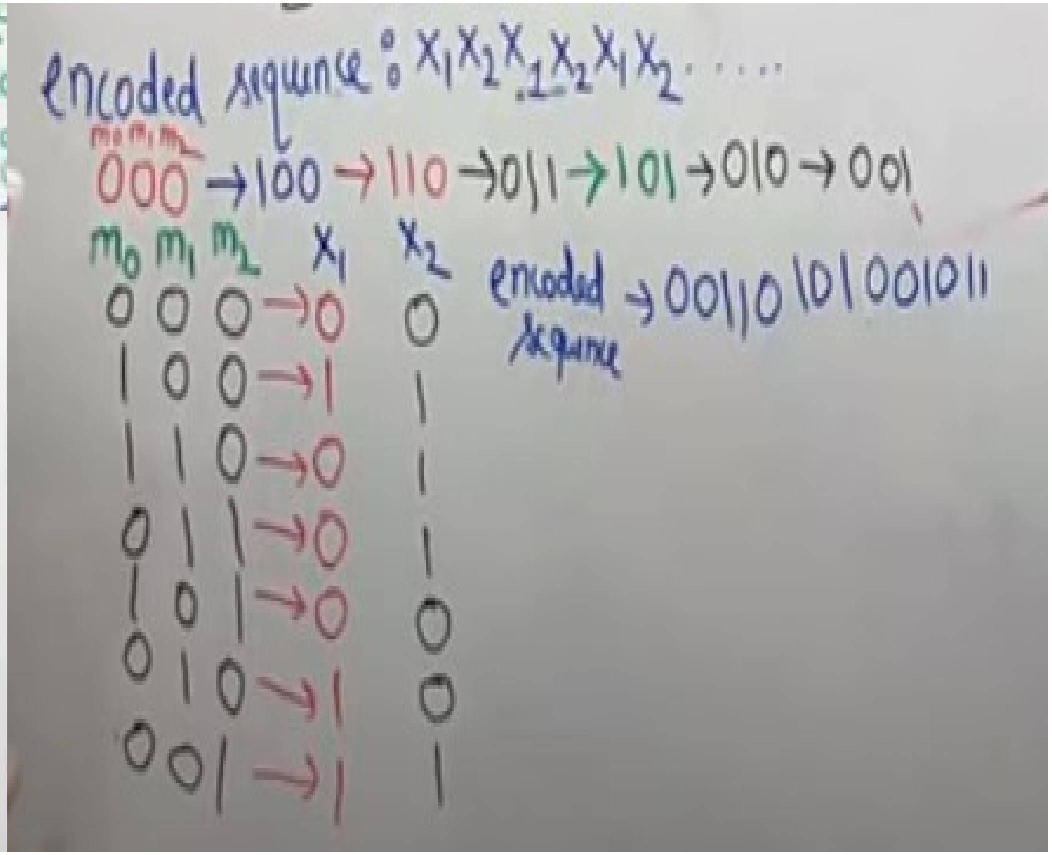
- The encoder of a binary convolutional code with rate $1/n$, measured in bits per symbol, may be viewed as a *finite-state machine* that consists of an $M$-stage shift register with prescribed connections to $n$ modulo-2 adders and a multiplexer that serializes the outputs of the adders.
- A sequence of message bits produces a coded output sequence of length $n(L + M)$ bits, where $L$ is the length of the message sequence. The *code rate* is therefore given by

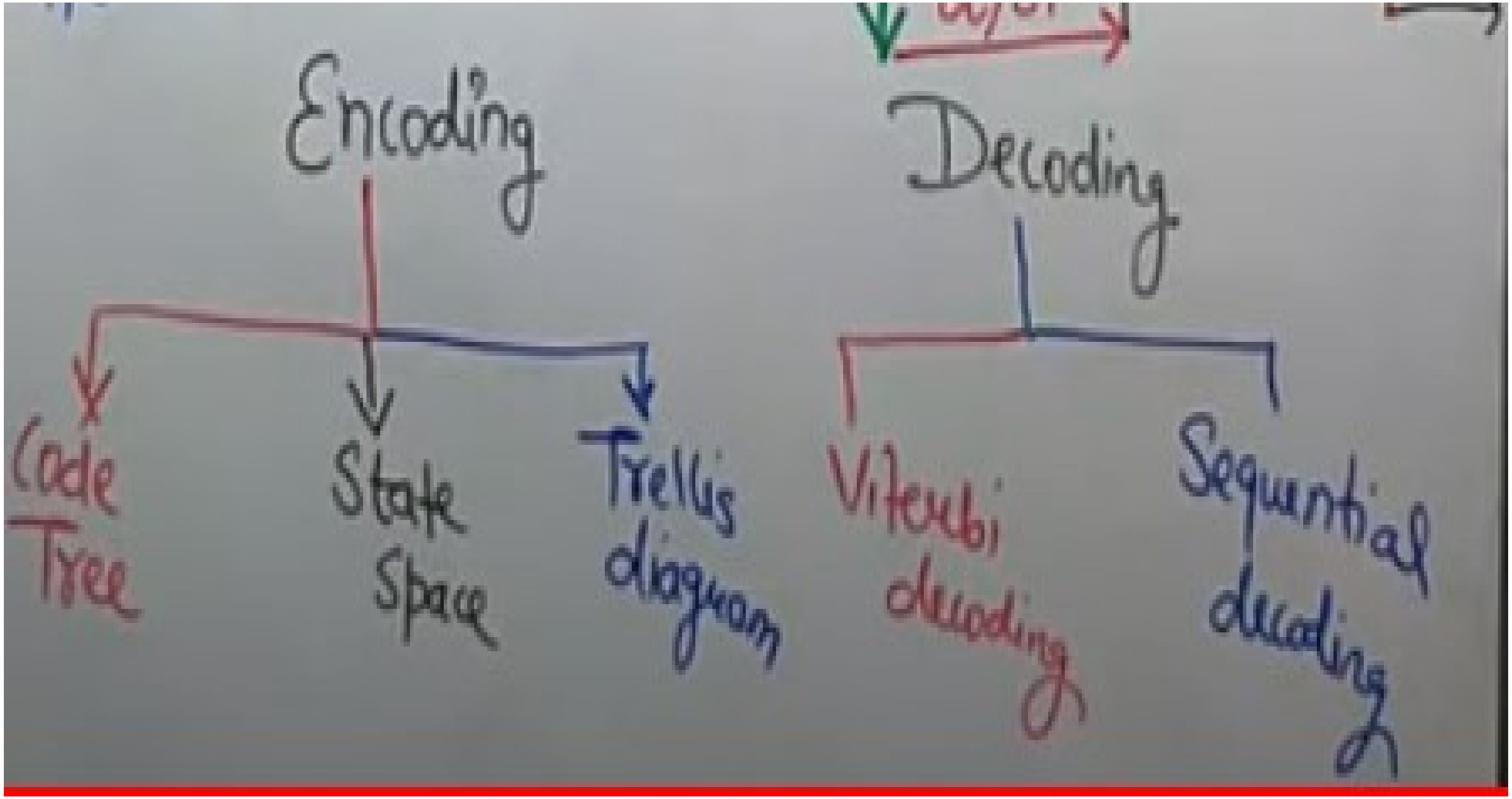
$$r = \frac{L}{n(L+M)}$$

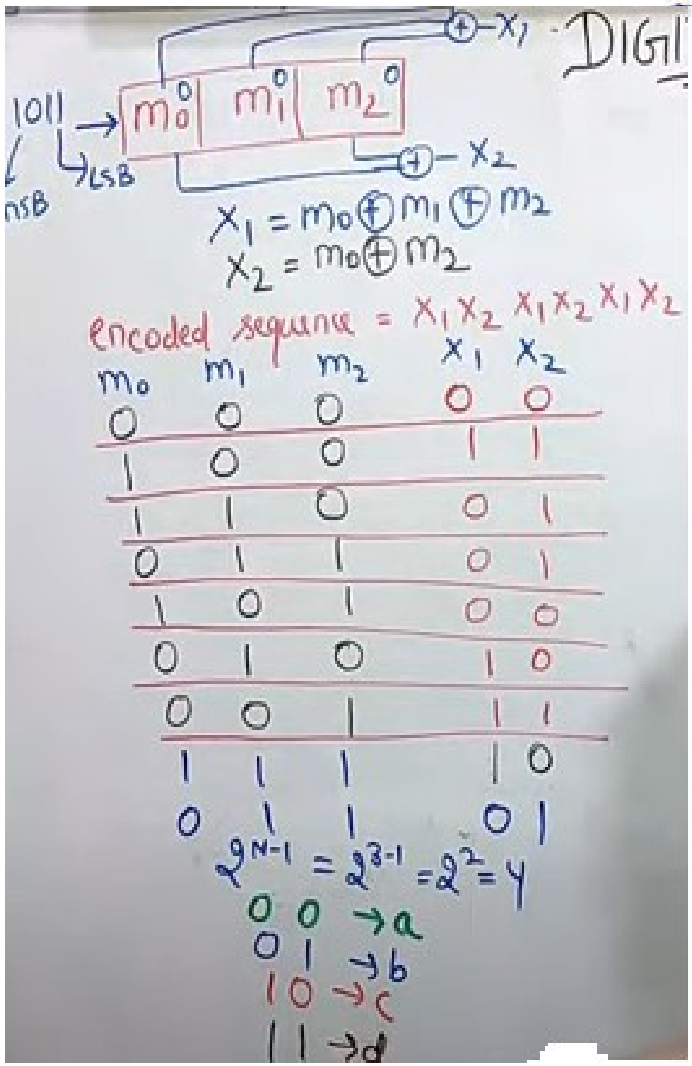
- *We have* $L \geq M$, then *code rate* is $r = 1/n$ *bits/symbol*

A convolutional code is a type of error-correcting code that generates parity symbols via the sliding application of a boolean polynomial function to a data stream. The sliding application represents the 'convolution' of the encoder over the data, which gives rise to the term 'convolutional coding'. The sliding nature of the convolutional codes facilitates



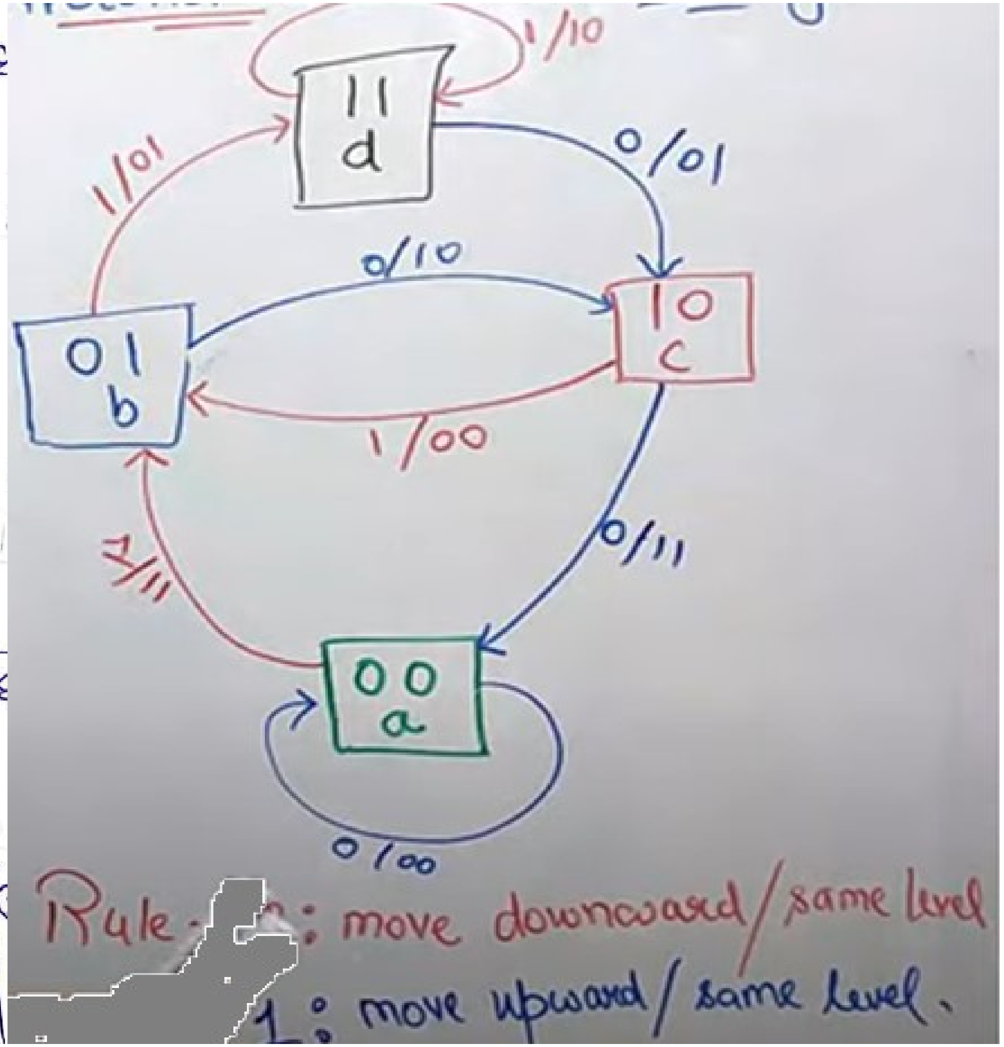

Encoding
Decoding
Code Tree
State Space
Trellis diagram
Viterbi decoding
Sequential decoding



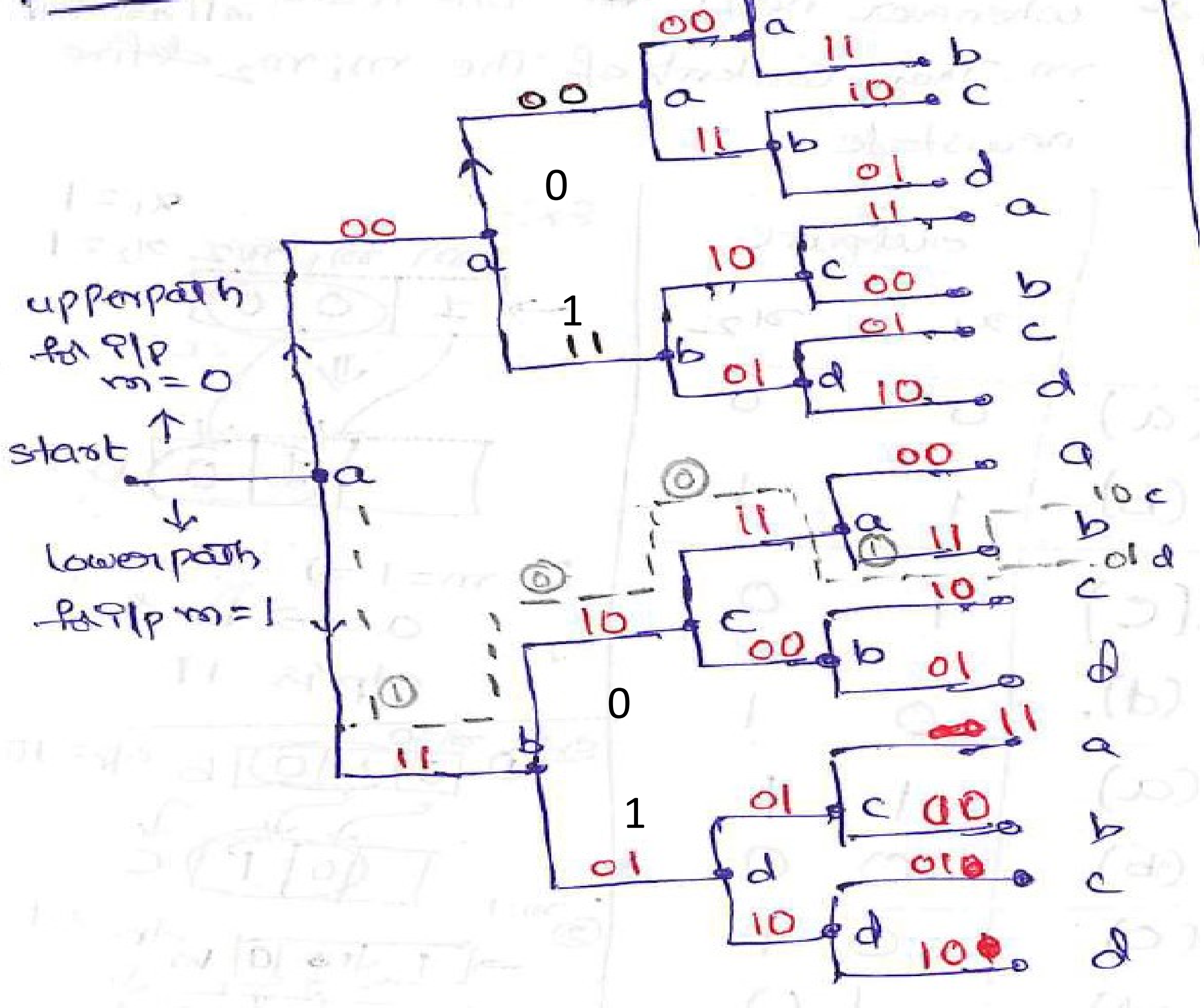

| Present state $m_1 m_2$ | Next state m (input) | $m_1$ | $m_2$ | | outputs $x_1$ | $x_2$ |
|---|---|---|---|---|---|---|
| a(00) | 0 → | 0 | 0 | (a) | 0 | 0 |
| | 1 → | 1 | 0 | (b) | 1 | 1 |
| b(010) | 0 → | 0 | 1 | (c) | 1 | 0 |
| | 1 → | 1 | 1 | (d) | 0 | 1 |
| c(01) | 0 → | 0 | 0 | (a) | 1 | 1 |
| | 1 → | 1 | 0 | (b) | 0 | 0 |
| d(11) | 0 → | 0 | 1 | (c) | 0 | 1 |
| | 1 → | 1 | 1 | (d) | 1 | 0 |

STATE DIAGRAM model

| Present State $m_1 m_2$ | Next state $m$ | $m_1$ | $m_2$ | outputs $x_1$ | $x_2$ |
|---|---|---|---|---|---|
| a(00) | 0 → | 0 | 0 (a) | 0 | 0 |
| | 1 → | 1 | 0 (b) | 1 | 1 |
| b (10) | 0 → | 0 | 1 (c) | 1 | 0 |
| | 1 → | 1 | 1 (d). | 0 | 1 |
| c (01) | 0 → | 0 | 0 (a) | 1 | 1 |
| | 1 → | 1 | 0 (b) | 0 | 0 |
| d (11) | 0 → | 0 | 1 (c) | 0 | 1 |
| | 1 → | 1 | 1 (d) | 1 | 0 |

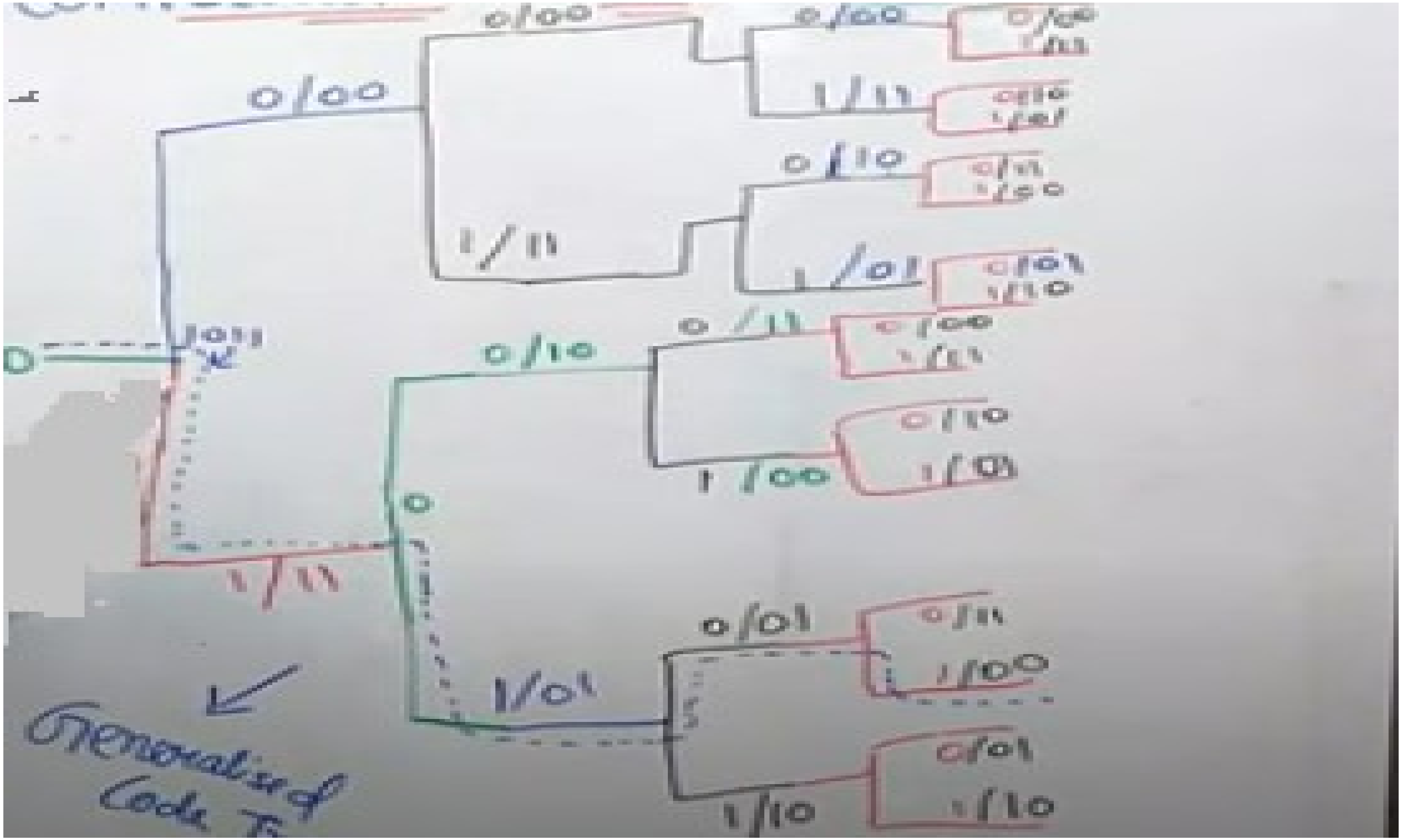


## Code Tree



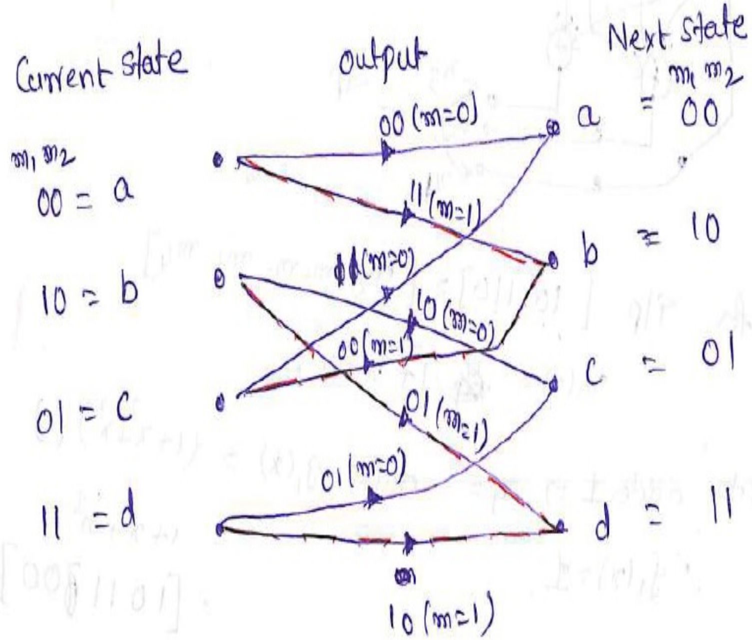

| Present state $m_1 m_2$ | Next state $m$ | $m_1$ | $m_2$ | outputs $x_1$ | $x_2$ |
|---|---|---|---|---|---|
| a(00) | 0 → | 0 | 0 (a) | 0 | 0 |
| | 1 → | 1 | 0 (b) | 1 | 1 |
| b(10) | 0 → | 0 | 1 (c) | 1 | 0 |
| | 1 → | 1 | 1 (d) | 0 | 1 |
| c(01) | 0 → | 0 | 0 (a) | 1 | 1 |
| | 1 → | 1 | 0 (b) | 0 | 0 |
| d(11) | 0 → | 0 | 1 (c) | 0 | 1 |
| | 1 → | 1 | 1 (d) | 1 | 0 |

Input 1 down
Input 0 up



0/00
1/11
0/10
1/01
1/00
0/01
1/10

**Trellis diagram :-** It is a more compact rept of the code Tree. It rept, single, an unique diagram for such Transition.




| Present State $m_1 m_2$ | Next state m → $m_1$ $m_2$ | outputs $x_1$ | $x_2$ |
|---|---|---|---|
| a(00) | 0 → 0 0 (a) | 0 | 0 |
| | 1 → 1 0 (b) | 1 | 1 |
| b (10) | 0 → 0 1 (c) | 1 | 0 |
| | 1 → 1 1 (d). | 0 | 1 |
| c (01) | 0 → 0 0 (a) | 1 | 1 |
| | 1 → 1 0 (b) | 0 | 0 |
| d (11) | 0 → 0 1 (c) | 0 | 1 |
| | 1 → 1 1 (d) | 1 | 0 |

## Decoding using Viterbi Decoding :-

Viterbi Algorithm is implemented based on trellis diagram.

Hamming distance = difference between branch code & recieved code

choose the path in the Trellis diagram with the smallest metric (d) weight and this selected selected paths are called as survival paths.

Ex:-

Input Data = m = [1 1 0 1 1]

Codeword $X$ = [11 01 01 <u>00</u> 01]

Recieved codeword $Y$ = [11 01 01 <u>10</u> 01]

# LAKIREDDY BALI REDDY COLLEGE OF ENGINEERING

(AUTONOMOUS)

Accredited by NAAC & NBA (CSE, IT, ECE, EEE & ME)

Approved by AICTE, New Delhi and Affiliated to JNTUK, Kakinada

L.B.Reddy Nagar, Mylavaram-521230, Krishna Dist, Andhra Pradesh, India