8/12/23

UNIT-5

→ Error Control Coding:-

In digital transmission, there may arise some errors. They are classified as:-

~~Parity~~ ~~Error~~

i, Fixed Bit Error Rate: (Fixed BER)

→ $\frac{E_b}{N_0} = \frac{\text{Energy per bit}}{\text{Noise Density}}$

→ In fixed BER, the transmission power decreases.

→ Errors are classified as:
- Content errors ('0's transmitted ... '1's received)
- Flow integrity errors (Missing blocks of data.)

→ Classification of codes based on errors:

Error detecting codes
- i) Parity Error detecting codes
- ii) Checksum " "
- iii) CRC " "

Error correcting codes
- i) Block codes
- ii) Convolutional codes
  - * Generate codewords
  - * Codewords contain data bits & check bits

→ Classifying codes
- Linear codes
- Non-linear codes

→ Error control coding Techniques:-
- Forward Error Correction Codes (FEC)
  - Concealment Error correcting codes
  - Linear Error correcting codes
    - Block codes
      - Cyclic codes
      - BCH codes
        - Red-solomen codes (RS) ⇓ Non-Binary codes
        - Hamming codes ⇓ Binary codes
    - Convolutional codes
      - self-orthogonal code
      - Trial & error codes
      - Recursive systematic codes (RSC)
- Automatic Repeat Request code (ARQ)
- Hybrid ARQ

8/12/23

UNIT-5

→ Error Control Coding:-

In digital transmission, there may arise some errors. They are classified as:-

~~Parity errors~~

(i) Fixed Bit Error Rate: (Fixed BER)

→ $\frac{E_b}{N_0} = \frac{\text{Energy per bit}}{\text{Noise Density}}$

→ In fixed BER, the transmission power decreases.

→ Errors are classified as:
- Content errors (0's transmitted --- 1's received)
- Flow integrity errors (missing blocks of data.)

→ Classification of codes based on errors:

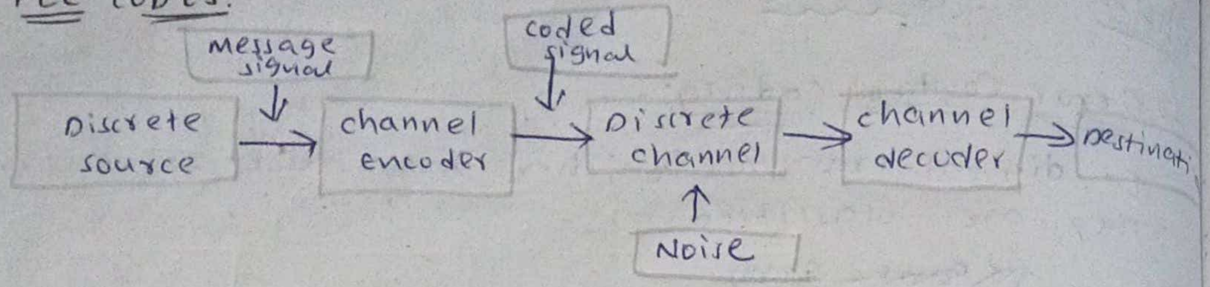
- Error detecting codes
  - (i) Parity Error detecting codes
  - (ii) Checksum " "
  - (iii) CRC " "
- Error correcting codes
  - i, Block codes
  - ii, Convolutional codes
    - * Generate codewords
    - * Codewords contain data bits & check bits

→ Classifying codes
- Linear codes
- Non-linear codes

→ Error control coding Techniques:-

- Forward Error Correction Codes (FEC)
  - Concealment Error correcting codes
  - Linear Error correcting codes
    - Block codes
      - Cyclic codes
        - Fire codes
      - BCH codes (Bose-Chaudhuri-Hocquenghem codes)
        - Reed-solomon codes (RS) ⇓ Non-Binary Codes
        - Hamming codes ⇓ Binary codes
    - Convolutional codes
      - self-orthogonal code
      - Trial & error codes
      - Recursive systematic codes (RSC)
- Automatic Repeat Request code (ARQ)
- Hybrid ARQ

M-2 **FEC CODES:**

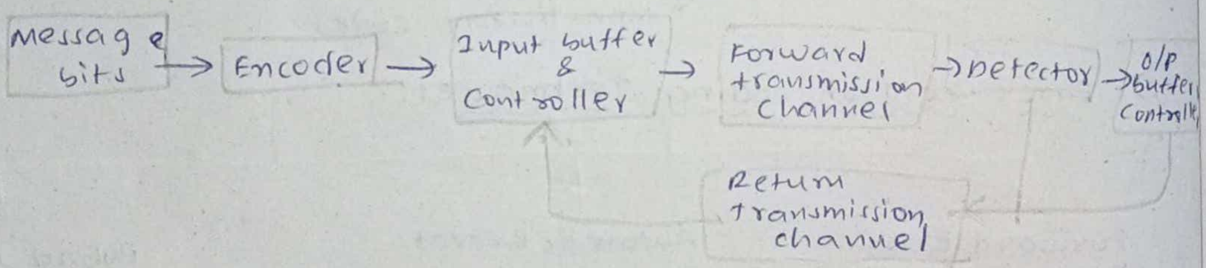


In FEC technique, the discrete source generates information in the form of binary symbols. The channel encoder accepts this message bits and will add redundancy according to the prescribed rule. Thus, the encoded data is produced at higher bit rate. The channel decoder uses the redundancy to decide which msg bit was actually transmistted at the tranimitter end.

→ In FEC technique, there is no feedback path and there is no request is made for Retransmission.

M-2 **ARQ code:-**

In the Automatic repeat request technique, the reciever can request for retransmission of the complete a part of the message if it finds some error in the received message, This requires an additional channel called Feedback (FB) channel to send the reciever's request for retransmission.



ACK/NAK (commands like Acknowledgment /Neg ACK)

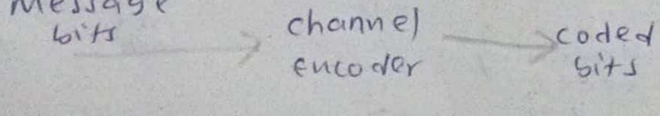
TYPES:

1. stop and wait ARQ
2. Go Back N ARQ
3. Selective Repeat ARQ

M-2 **BLOCK CODES:** (n, k code): These codes do not require memory

→ Structure of Block codes is shown below:

→ It is one of the linear correcting codes.



codeword length, n bits	
k-msg bits	(n-k) parity bits (or) check bits

→ Redundant bits (They contain no related information)

* $x = [M ; c]$ (code vector, msg vector (k), parity vectors (n-k))

Convolutional codes:-

→ These codes require memory.
→ This code is also a type of linear correcting error codes.

$m(t) \rightarrow$ [Encoder $h(t)$] $\rightarrow m(t) \otimes h(t)$

→ In convolutional codes, the code words are generated by discrete time convolution of the input sequence with the impulse response of the encoder. Convolutional codes need memory for their generation.

→ the encoder of a convolutional code operates on the incoming message sequence using a sliding window equal to the duration of its own memory. Hence in convolutional code unlike a block code, the channel encoder accepts msg bits as a continuous sequence and generates a continuous sequence of encoded bits at its output.

→ The major difference b/w the block codes and the convolutional codes is that the " convolutional codes require memory where as the block codes do not require any memory and both of them belong to the category of linear codes.

TRANSMISSION ERRORS:-

→ Random Errors
→ Burst Errors

→ The errors that occurs in a purely random manner are known as "Random errors".

→ The errors which occur in the form of bunches and hence which are not independent are known as "Burst errors".

→ BCH codes are useful in dealing with Random Errors. Because of impulse noise ~~more~~ more than one corresponding data bits change their state instantaneously.

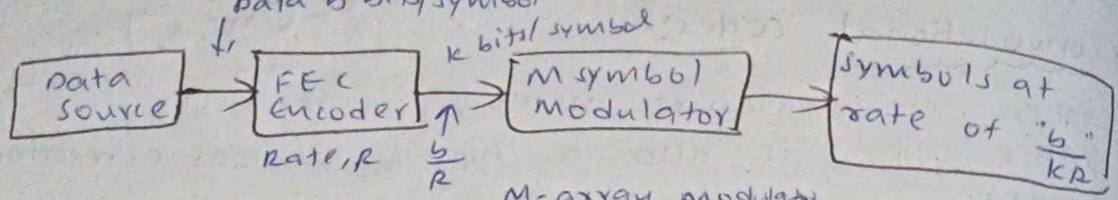
→ The error introduced in the received word is called " Burst error".

→ The convolutional error correcting codes are not effective in correcting the Burst errors.

→ Special codes are developed for detecting & correcting Burst

→ The best known codes of this category are "Fire codes" which belong to sub-class of cyclic codes.

→ Classical Error Control system :-



M-array modulator
M-array PSK
M-array FSK
QAM (Quadrature Amp modulator)

The figure shows a classical error control system where conventional FEC system is followed by a multi-level modulator.

→ Let the input data rate be 'b' bits/sec. The FEC encoder converts this into a coded data which has a rate b/R, where 'R' is the information rate. The 'M' symbol modulator converts the encoder o/p into 'M' possible symbol considerations. The symbol rate at o/p of the modulator is

$$\boxed{r_s = \frac{b}{kR} \text{ baud}}.$$

Therefore, minimum system bandwidth that is required to transmit the signal sucessfully is given by:

$$\boxed{B = r_s = \frac{b}{kR} \text{ Hz}}$$

Hence, the corresponding BW efficiency is given by:

$$\eta = \frac{b}{r_s} = kR = R \cdot \log_2^M \text{ bits/Hz}$$

where,
R = rate of encoder
M = M-array system

* TYPES OF CHANNELS :-

There are two types of channels.
① Band-limited channel
② power-limited channel.

① **Band-limited channel**:- These channels have a fixed finite BW. Therefore, signal which requires larger bandwidth cannot be transmitted over such channels without distorting them.

→ Ex: Telephone lines, Parallel lines etc.

- When we add FEC to the Bandlimited channel, due to the additional parity bits, the data rate is reduced.

② **Power-limited channel**:- The power-limited channels have a limited power associated with them. But they have a large BW.

Ex: satelite channel.

- It is possible to accomodate the FEC inspite of increased parity bits.

→ **Definitions**:-

* **CODEWORD**:- A codeword is the 'n' bit encoded block of bits. It contains message and Redundant bits.

* **CODE RATE**: The coderate is defined as the "ratio of the number of message bits (k) to the total no. of bits (n) in a codeword.

$$\boxed{r = \frac{k}{n}}$$

* **CODE VECTORS**:- We can visualize an 'n' bit codeword to be present in an N-dimensional space. The coordinates are elements of this code vector are the bits present in the codeword.

* **HAMMING DISTANCE**:- Consider two code vectors/ code words having the same number of elements. The hamming distance is simply the distance between two codewords is defined as the no. of locations in which their respective elements will differ.

* **HAMMING WEIGHT $W(x)$**:- The hamming weight of a code word x is defined as "the non-zero elements in a codeword".

→ Hamming weight of a codevector is "the distance between that codeword and an all-zero code vector".

* **CODE EFFICIENCY**:- The code efficiency is defined as "the ratio of message bits to no. of transmitted bits per block".

code efficiency = code rate = $\frac{k}{n}$.

* **MINIMUM DISTANCE (dmin):-** The minimum distance dmin of a linear blockcode is defined as" the smallest Hamming distance b/w any pair of code vectors in c

- code word-1: 1101 0100
- code word-2: 0101 1110

$\Rightarrow$ Hamming distance = 3

· **Error detection & correction based on Hamming distance dmin:**

1. Detect upto 's' errors per word: $d_{min} \geq (s+1)$
2. correct upto 't' errors per word: $d_{min} \geq (2t+1)$
3. correct upto 't' errors & detect upto 's' errors per word $(s>t)$: $d_{min} \geq (t+s+1)$

Q For a Hamming distance of '5'. How many errors can be detected, How many errors can be corrected?

sol: Given, $d_{min} = 5$,

$\rightarrow$ we can detect: $d_{min} \geq s+1$;

$5 \geq s+1$

$4 \geq s$

$\Rightarrow \boxed{\leq 4 \text{ errors}}$;

$\rightarrow$ we can correct: $d_{min} \geq 2t+1$

$5 \geq 2t+1$

$4 \geq 2t$

$t \leq 2$

$\Rightarrow \boxed{\leq 2 \text{ erros}}$

Q For a (6,3) code, $G = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$ i) Find code rate ii) Find parity check matrix

- $n = 6$
- $k = 3$
- $G = [I / P]$

$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$; $P = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}_{3\times3}$

· code rate $= \frac{k}{n} = \frac{3}{6} = \frac{1}{2}$

·

, code rate $= \frac{k}{n} = \frac{3}{6} = \frac{1}{2}$

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

→ General representation of Block codes:-

$$* \quad \boxed{X = [M : C]} \quad -①$$

where,

X = code vector of size $1 \times n$

M = Message vector of size $1 \times k$

C = parity vectors of size $(n-k)$

$$\boxed{X = M G} \quad -②$$

→ Matrix Representation of Block Codes:-

* G: Generator matrix, of $k \times n$ size is,

$$\boxed{[G] = [I_k \,|\, P]} \quad -③$$

where,

$I_k = k \times k$ matrix $= \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & \cdots & 0 \\ 0 & 0 & \cdots & \cdots & 1 \end{bmatrix}_{k \times k}$

P = coefficient matrix of $k \times (n-k)$ $= \begin{bmatrix} P_{00} & P_{10} & \cdots & P_{(n-k)0} \\ P_{01} & P_{11} & \cdots & P_{(n-k)1} \\ \vdots & & & \\ P_{0(k-1)} & P_{1(k-1)} & \cdots & P_{(n-k)(k-1)} \end{bmatrix}$

$$* \quad \boxed{C = MP} \quad -④$$

$$[c_0, c_1, c_2 \cdots c_{n-k}]_{1 \times (n-k)} = [m_0, m_1, \cdots m_k]_{1 \times k} \times P$$

$\Rightarrow c_0 = m_0 P_{00} \oplus m_1 P_{01} \oplus m_2 P_{02} \cdots \oplus m_k P_{0(k-1)}$

$c_1 = m_0 P_{10} \oplus m_1 P_{11} \oplus \cdots \oplus m_k P_{1(k-1)}$

$c_2 = m_0 P_{20} \oplus m_1 P_{21} \oplus \cdots \oplus m_k P_{2(k-1)}$

$\vdots$

$c_{n-k} = m_0 P_{(n-k)0} + m_1 P_{(n-k)1} \oplus \cdots \oplus m_k P_{(n-k)(k-1)}$ //

Q: Generator matrix for a (6,3) block code is given as $G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$, Find all code vectors of this code.

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

w.k.t, $[G] = [I_k \,|\, P]$

$\Rightarrow I_k = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}_{3 \times 3}$; $P = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}_{3 \times 3} = \begin{bmatrix} P_{00} & P_{10} & P_{20} \\ P_{01} & P_{11} & P_{21} \\ P_{02} & P_{12} & P_{22} \end{bmatrix}$

→ $k = 3$

⇒

$m_0\ m_1\ m_2$	c_0	c_1	c_2
0 0 0	0	0	0
0 0 1	1	1	0
0 1 0	1	0	1
0 1 1	0	1	1
1 0 0	0	1	1
1 0 1	1	0	1
1 1 0	1	1	0
1 1 1	0	0	0

* $c_0 = m_1 \oplus m_2$

* $c_1 = m_0 \oplus m_2$

* $c_2 = m_0 \oplus m_1$

XOR

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

→ $c_0 = [0 \quad 0 \quad 0] \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} = 0 \oplus 0 \oplus 0 = 0 \oplus 0 = 0$

→ $c_1 = [0 \quad 0 \quad 0] \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = 0 \oplus 0 \oplus 0 = 0$

→ $c_2 = [0 \quad 0 \quad 0] \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = 0 \oplus 0 \oplus 0 = 0$

→ $c_0 = [0 \quad 0 \quad 1] \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} = 0 \oplus 0 \oplus 1 = 0 \oplus 1$ $\boxed{c_0 = 1}$; $c_1 = [0 \quad 0 \quad 1] \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$ $\boxed{c_1 = 1}$; $c_2 = [0 \quad 0 \quad 1]$ $\boxed{c_2 = 0}$

→ $c_0 = [0 \quad 1 \quad 0] \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$; $c_1 = [0 \quad 1 \quad 0] \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$; c

→ Alternate way of expressing parity check matrix;

$$\boxed{H = [P^T \mid I_{(n-k)}]}$$, size of $P = (n-k) \times k$

Q:- The parity check matrix of a particular (7,4) linear block code is given by $[H] = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$

(i) Find Generator matrix [G]
(ii) Find all codevectors
(iii) What is the minimum distance b/w codevectors
(iv) How many errors can be detected & corrected.

Sol: Given, $n = 7$; $k = 4$

$\Rightarrow P^T = (n-k) \times k = 3 \times 4 = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}_{3 \times 4} \Rightarrow P = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$

(i) Generator matrix, $[G] = [I_{k \times k} \mid P]$

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

(ii) $c_0 = m_0 \times 1 \oplus m_1 \times 1 \oplus m_2 \times 1 \oplus m_3 \times 0 = m_0 \oplus m_1 \oplus m_2$

$c_1 = m_0 \oplus m_1 \oplus m_3$

$c_2 = m_0 \oplus m_2 \oplus m_3$

m_0	m_1	m_2	m_3	c_0	c_1	c_2	weights
0	0	0	0	0	0	0	0
0	0	0	1	0	1	1	3
0	0	1	0	1	0	1	3
0	0	1	1	1	1	0	4
0	1	0	0	1	1	0	3
0	1	0	1	1	0	1	4
0	1	1	0	0	1	1	4
0	1	1	1	0	0	0	3
1	0	0	0	1	1	1	4
1	0	0	1	1	0	0	3
1	0	1	0	0	1	0	3
1	0	1	1	0	0	1	4
1	1	0	0	0	0	1	3
1	1	0	1	0	1	0	4
1	1	1	0	1	0	0	4
1	1	1	1	1	1	1	7

* $c_0 = m_0 \oplus m_1 \oplus m_2$
* $c_1 = m_0 \oplus m_1 \oplus m_3$
* $c_2 = m_0 \oplus m_2 \oplus m_3$

(iii) The minimum distance, d_{min} = The minimum weight of any non-zero code vector.

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

(iv) → Errors can be detected: $\boxed{d_{min} \geq s+1}$

$3 \geq s+1 \Rightarrow \boxed{s \leq 2}$

⇒ we can detect at most $\underline{\underline{2}}$ errors.

→ errors can be corrected: $\boxed{d_{min} \geq 2t+1}$

$3 \geq 2t+1 \Rightarrow \boxed{t \leq 1}$

⇒ we can correct at most $\underline{\underline{1}}$ error.

→ Error can be detected & corrected: $d_{min} \geq t+s+1$

* The transpose of parity check matrix $[H^T]$ exhibits a very important property which is $\boxed{XH^T = [0, 0, --- 0]}$

This means that the product of any code vector and transponse of parity check matrix will always be zero. We shall use this property for the detection of errors in the recieved codewords.

→ This is called "Syndrome decoding for block codes"

Q: Find a code vector $x = [0\ 1\ 1\ 1\ 0\ 0\ 0]$ and the parity check matrix $H = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}_{3\times 7}$. Find xH^T.

Sol: $xH^T = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} [0\ 1\ 1\ 1\ 0\ 0\ 0] = [2\ \ 2\ \ 2]$

$\Rightarrow$ No error. $\because$ [XOR]

Properties of Hamming Code:- (n, k); $q \geq 3$

1. Length of codeword, $n = 2^q - 1$
2. No. of msg bits, $k = 2^q - q - 1$
3. No. of parity bits, $q = (n-k)$; $q \geq 3$
4. Minimum distance, $d_{min} = 3$
5. Code rate / code efficiency, $\frac{k}{n} = \frac{2^q - q - 1}{2^q - 1}$

CONVOLUTIONAL CODES:

- In Block coding, the encoder accepts a k-bit msg block and generates an n-bit code word. Code words are produced on a block-by-block basis. Therefore provision must be made in the encoder to buffer an entire msg block before generating associated codeword.

There are applications, where the msg bits comes serially rather than in large blocks. In which, the use of buffer may be undesirable. In such situations the convolutional coding may be the preferred method.

→ A convolutional encoder operates on the incoming msg sequence continuosly in a serial manner.

→ The encoder of a binary convolutional code with rate "1/n" measured in bits/symbol may be viewed as "Finite state Machine (FSM)", that consists of M-stage shift register with prescribed connections to n modulo-2 adders and a multiplexer that serializes the o/ps of adders.

→ To this encoder, if L-bit msg is given, this produces a coded o/p sequence of length "$n(L+M)$" bits.

⇒ code rate = $\frac{L}{n(L+M)}$ bits/symbol, where $\boxed{L >> M}$

No. of shift

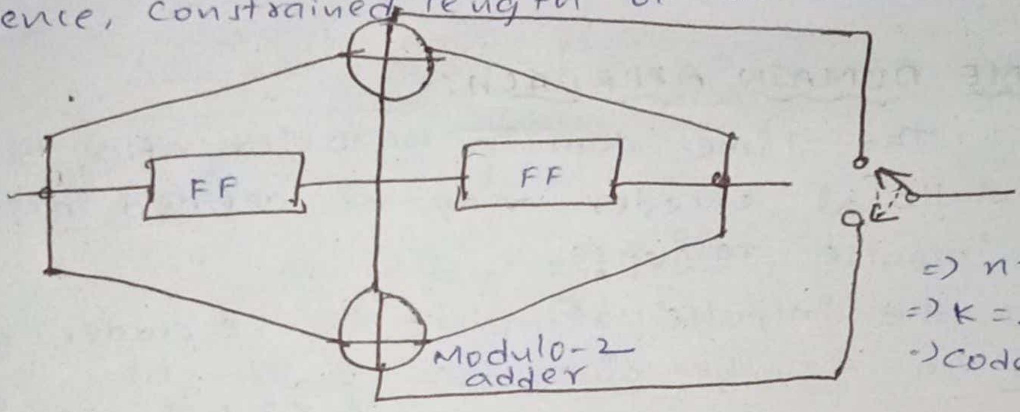
$= \frac{L}{nL}$

coder rate $\simeq \frac{1}{n}$ bits/symbol.

* **Constraint length:-** Constraint length of a convolution code is expressed in terms of msg bits. It is defined as "No. of shifts over which a single msg bit can influence the encoder o/p".

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

Hence, constrained length of the encoder is 'K'.



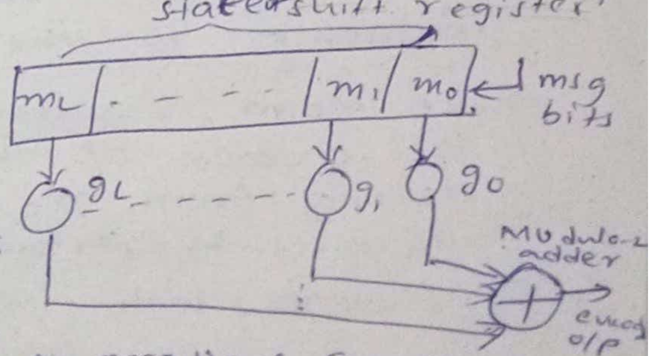
⇒ $n = 2$

⇒ $K = 2+1 = 3$

⇒ Coderate $= \frac{1}{2}$

** Generalised structure of encoder *

→ This encoder operates on, the incoming msg sequence, one-bit at a time.



· The o/p is given as:

→ $x = m_L g_L \oplus \cdots \oplus m_1 g_1 + m_0 g_0$ tap gains

→ $x = \sum_{i=0}^{L} m_i g_i$: mod-2 addition & binary convolution

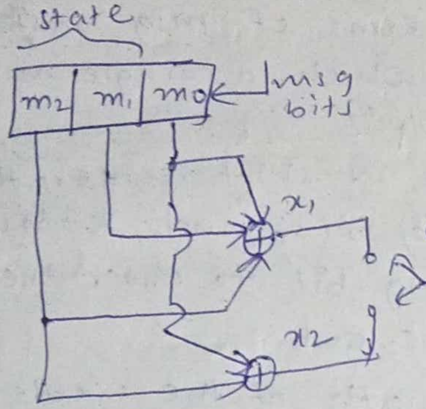
* pratical Encoder *

· The name convolutional encoding comes from the fact that, the above eqn. has a form of binary convolution, which is analogous to the convolutional integral.

The msg bit m_0 represents the current i/p, whereas bits $m_1 \to m_L$ represent the past i/p (or) state of shift register. This eqn. clearly shows that a msg bit 'x' depends on current msg bit 'mo' and state of shift register defined by the previous 'L' msg bits. It is important to note that a particular msg bit influences a span of "L+1" sucessive encoded bits as it shift through the register.

→ The code is represented as (n, k, L).

Ex:



- $x_1 = m_0 \oplus m_1 \oplus m_2$
- $x_2 = m_0 \oplus m_2$

commutator switch
- $x = x_1 x_2 x_1' x_2' x_1'' x_2''$

Encoded bits
- $n = 2$
- $L = 2$ (state)

① TIME DOMAIN APPROACH:

The Time domain behavior of a binary convolutional encoder may be defined interms of 'n' impulse response.

→ Let the impulse response of the adder generating x_i in fig. be given by $\{g_0^{(1)}, g_1^{(1)}, \ldots g_L^{(1)}\}$.

→ similarly, for the adder generating the sequence x_2 the impulse response is represented by $\{g_0^{(2)}, g_1^{(2)}, \ldots g_L^{(2)}\}$

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

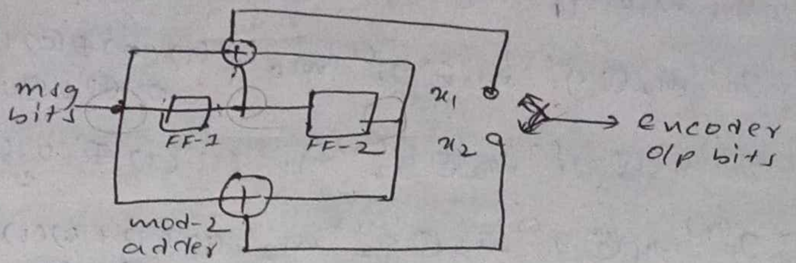
→ Let $m_0, m_1, \ldots$ denote the msg sequence entering the encoder of the figure one-bit at a time, starting from "mo". The encoder generates 2 o/p sequences by performing convolutions on the msg sequence with impulse response, given by:

* $x_1 = x_i^{(1)} = \sum_{i=0}^{L} g_i^{(1)} m_{i-1}$

* $x_2 = x_i^{(2)} = \sum_{i=0}^{L} g_i^{(2)} m_{i-1}$

$\Rightarrow x = \{x_0^{(1)} x_0^{(2)} x_1^{(1)} x_1^{(2)} \ldots\}$

Q: The convolutional encoder which is shown in figure



The figure shown has generator sequences each of length 3, $\{g_0^{(1)}\ g_1^{(1)}\ g_1^{(2)} \ldots\} = (1,1,1)$ and $\{g_0^{(2)} g_1^{(2)} g_2^{(2)}\} = (1,0,1)$. Determine the Encoded sequence for the following i/p msg,
$(m_0\ m_1\ m_2\ m_3\ m_4) = (1\ 0\ 0\ 1\ 1)$.

Sol:- Given, $(m_0\ m_1\ m_2\ m_3\ m_4) = (1\ 0\ 0\ 1\ 1)$

$\{g_0^{(1)}\ g_1^{(1)}\ g_1^{(2)} \ldots\} = (1,1,1)$

$\{g_0^{(2)}, g_1^{(2)}, g_2^{(2)} \ldots\} = (1,0,1)$

$\Rightarrow \boxed{x_1 = \sum_{i=0}^{L} g_i^{(1)} \cdot m_{i-1}}$

$= g_0^{(1)} m_{-1} \oplus g_1^{(1)} m_0 \oplus g_1^{(2)} m_1$

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

$= 1 \oplus 1 \oplus 0 = 0 \oplus 0 = \underline{0}$

$\Rightarrow \boxed{x_2 = \sum_{i=0}^{L} g_i^{(2)} m_{i-1}}$

$= g_0^{(2)} m_{-1} \oplus g_1^{(2)} m_0 \oplus g_2^{(2)} m_1$

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

$= 1 \oplus 0 \oplus 0$

$= 1 \oplus 0$

$\boxed{x_2 = 1}$

$x_0^{(1)} = g_0^{(1)} m_0 = (1)(1) = 1$ $\qquad \Rightarrow x_1 = (1\ 0\ 0\ 1\ 1\ 1\ 1)$

$x_1^{(1)} = g_0^{(1)} m_1 + g_1^{(1)} m_0 = (1)(0) \oplus (1)(1) = 0+1 = 1$

$x_2^{(1)} = g_0^{(1)} m_2 + g_1^{(1)} m_1 + g_2^{(1)} m_0 = (1)(0) \oplus (1)(0) \oplus (1)(1) = 1$

$x_3^{(1)} = g_0^{(1)} m_3 + g_1^{(1)} m_2 + g_2^{(1)} m_1 = (1)(1) \oplus (1)(0) \oplus (1)(0) = 1$

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

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

$x_6^{(1)} = +(1)(1) = 1$

* $x_0^{(2)} = g_0^{(2)} m_0 = (1)(1) = 1$

* $x_1^{(2)} = g_0^{(2)} m_1 \oplus g_1^{(2)} m_0 = (1)(0) \oplus (0)(1) = 0$

* $x_2^{(2)} = g_0^{(2)} m_2 \oplus g_1^{(2)} m_1 \oplus g_2^{(2)} m_0 = (1)(0) \oplus (0)(0) \oplus (1)(1) = 0 \oplus 0 \oplus 1 = 1$

* $x_3^{(2)} = g_0^{(2)} m_3 \oplus g_1^{(2)} m_2 \oplus g_2^{(2)} m_1 = (1)(1) \oplus (0)(0) \oplus (1)(0) = 1 \oplus 0 \oplus 0 = 1$

* $x_4^{(2)} = g_0^{(2)} m_4 \oplus g_1^{(2)} m_3 \oplus g_2^{(2)} m_2 = (1)(1) + (0)(1) + (1)(0) = 1 \oplus 0 \oplus 0 = 1$

* $x_5^{(2)} = \oplus (0)(1) \oplus (1)(1) = 0 \oplus 1 = 1$

* $x_6^{(2)} = \oplus \quad \oplus (1)(1) = 1$

→ code word = (1 1 1 1 1 1 0 1)

∴ $X = \{ x_0^{(1)} x_0^{(2)} \; x_1^{(1)} x_1^{(2)} \; x_2^{(1)} x_2^{(2)} \; x_3^{(1)} x_3^{(2)} \; x_4^{(1)} x_4^{(2)} \; x_5^{(1)} x_5^{(2)} \; x_6^{(1)} x_6^{(2)} \}$

$X = \{ 11 \; 10 \; 11 \; 11 \; 01 \; 01 \; 11 \}$

* Convolution Decoding:

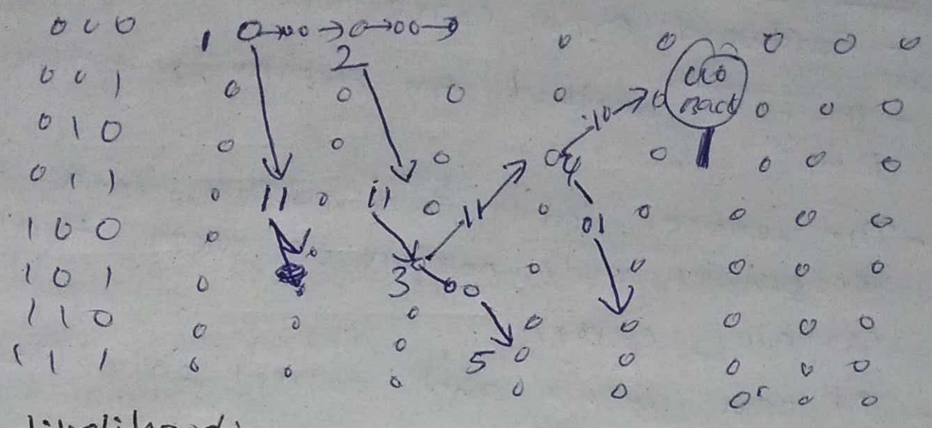
- There are 2 different approaches for decoding of convolutional codes:
  * Sequential decoding - Fano algorithm
  * Maximum likelihood Viterbi Algorithm

- Both of 2 methods represent two different approaches

① Sequential Decoding:

- It was one of 1st methods for decoding of convolution codes.
- It follows fano algorithm.
- At each coding stage, fano algorithm retains the information regarding three paths:-
  * Current path
  * Immediate predecessor path
  * One of successor paths.
- It concentrates only on a certain number of code words.
- purpose of sequential decoding is to search through nodes of code tree in an efficient way to find the maximum likelihood path

Ex: incoming bits 01 11 01 11 01



② Maximum likelihood:

- This decoding uses viterbi algorithm.
- This examines entire recieved seauence of a given length.
- It works on maximum likelihood rule which tried to reduce error b/w the detected seauence and original transmitted seauence.
- Trelli diagram is constructed based on ~~system~~ recieved seauence for a system
- The path is straight.
- If a condition raises in such a way that there is no path for corresponding seauence then the viterbi decoding helps to detect best path based on subsequent seauence.

- FEC codes:
  - Fordward error correction
  - It is a method for obtaining error control in data.
  - The source sends redundant data and destination recognizes only portion of data that does not contains errors.
  - It can detect and correct only limited number of errors.
  - If there are too many errors, sender must retransmit packets that contain errors.
  - It doesn't require handshaking which means no connection b/w transmitter & receiver before data is transmitted.
  - This makes it possible to broadcast data to multiple destinations simultaneously from single source.

- Applications:
  - Video streaming
  - unreliable communication ⇒ more noise
  - routers
  - cable communication
  - optic communication
  - wireless communication

- working:
  - Let w = 01 01 01 11 is sent 3 times.

transmitted	recieved
01 01 01 11	01 01 01 11
01 01 01 11	01 01 00 11
01 01 01 11	00 01 01 11

- Takes more occured bit as correct bit.
- High level systems uses encoder that adds parity and decoder that extracts original signal.
- codes used:
  - Hamming
  - BCH
  - Low density parity check (LDPC)
  - RS
  - Turbo

- **Hamming:-**
  - It is based on linearity property.
  - i.e., sum of two codewords is a code word.
  - There are type of linear error correcting codes
    - Can detect upto 2 bit errors
    - Can correct only one bit error without detection.
    - extra parity parity bits are used to identify single bit error.
    - To get from one-bit pattern to other, few bits are to be changed in data.
    - such numbers of bits are termed as "Hamming distance"
- **Properties:-** 2^{q-1}, $2^q - q - 1$, $n - \cancel{k}$, $\frac{2^{q-1}}{2^q - q - 1}$ dmin
- **Advantages:-**
- **Disadvantages:-**
- **BCH codes:-**
  - BCH resembles Bose, chaudari & Hocquenghem codes.
  - During BCH code design, there is a control on no. of symbols to be corrected and hence multiple bit correction is possible.
  - It is a powerful technique in error correcting codes.
  - **properties:-**
    1. Block length, $n = 2^m - 1$
    2. parity digits in $n - k \leq mt$
    3. minimum distance, $d_{min} \leq 2t+1$
  - also called as t-error correcting BCH code.
- **cyclic codes:**
  - These follow cyclic shift property.
  - i.e., cyclic shift of a code word is also a codeword.
  - These are mainly used to correct double errors & burst errors.
  - shift register & modulo-2 adder are 2 crucial elements considered as building blocks of cyclic encoding.

· Using shift register, encoding can be performed.

→ properties:

① property of linearity:

$c_i + c_j = c_p$ → also code word

② property of cyclic shift:

· $c = [c_1, c_2 \ldots c_{n-1}]$

after shifts,

$c^0 = [c_{n-1}, c_1, c_2 \ldots c_{n-2}]$

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

c^0, c^1 also coder wordes.

→ There are 2 types:

* systematic cyclic encodeng

* Non-systematic cyclic encoding → $C(X) = m(X)g$

$\boxed{C(X) = X^{n-k} M(X) + P(X)}$

$$= \frac{x^{n-k} M(X)}{g(x)}$$