SECX1033 - DIGITAL COMMUNICATION

UNIT IV - ERROR CONTROL CODING

Introduction :

Errors are introduced in the data when it passes through the channel. Channel noise interferes the signal, so the signal power is reduced. Transmission power and channel bandwidth are the main parameters in transmission of data over the channel. With this parameters power spectral density of channel noise is also determine signal to noise ratio. This SNR determines the probability of error. Using coding Techniques Signal to noise ratio is reduced for fixed probability of error.

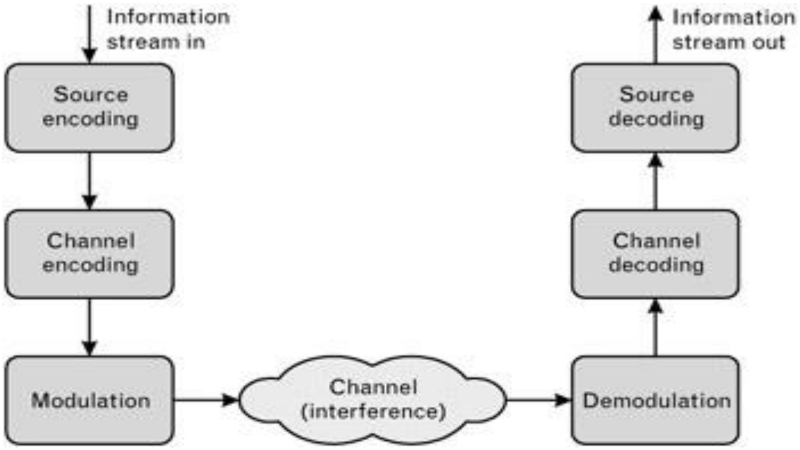


Fig. Digital Communication Systems with channel encoding

Channel encoder :

The channel encoder adds bits (redundancy) to the message bits. The encoder signal is then over the noisy channel.

Channel decoder :

It identifies the redundant bits and uses them to detect and correct the errors in the message bits if any. Thus the number of errors introduced due to channel noise are minimized by encoder and decoder. Due to the redundant bits the overall data rate increases. Hence channel has to accommodate this increased data rate. Systems becomes slightly complex because of coding techniques.

Types of codes:

(1) Block Codes :

These Codes consists of ‘n’ number of bits in one block or codeword. This codeword consists of ‘k’ message bits and (n-k) redundant bits. Such block codes are called (n, k) block codes.

(2) Convolutional Codes :

The coding operation is discrete time convolution of input sequence with the impulse response of the encoder. The Convolutional encoder accepts the message bits continuously and generates the encoded sequence continuously.

The codes can also be classified as linear or nonlinear codes

a. **Linear Code:** If the two codewords of the linear codes are added by modulo-2 arithmetic the it produces third codeword in the code.

b. **Nonlinear Code:** Addition of the nonlinear codeword does not necessarily produce third codeword.

Discrete memory less channel :

A discrete channel comprises of an input alphabet X, output alphabet Y, and a likelihood function (probability transition matrix) p(y│x). The channel is said to be memory less if the probability distribution of the output depends only on the input at that time and is conditionally independent of previous channel inputs or outputs."Information" channel capacity of a discrete memory less channel is:

$$C = \max_{p(x)} I(X;Y),$$

where the maximum is taken over all possible input distributions p(x).

Methods of error correction

Forward Error Correction (FEC)

- Coding designed so that errors can be corrected at the receiver
- Appropriate for delay sensitive and one-way transmission (e.g., broadcast TV) of data
- Two main types, namely block codes and Convolutional codes.

Error Correction with retransmission or Automatic repeat request (ARQ)

- Decoder check the input sequence
- When it detects any error, it discards that part of the sequence and request the transmitter for retransmission.
- Transmitter then again retransmits the part of the sequence in which error was detected .
- Hence the decoder does not corrects the error, it just detects the error.
- It has low probability error but it is slow.

Types of errors

Random errors: These errors are due to white Gaussian noise in the channel. These error generated in a particular interval does not affect the performance of the system in subsequent intervals. These errors are totally uncorrelated.

Burst errors: These errors due to impulsive noise in the channel. Impulsive noise due to lighting and switching transient. These error generated in a particular interval will affect the performance of the system in subsequent intervals.

Important words in error control coding techniques:

Codeword: The encoded block of ‘n’ bits is called a codeword. It contains message bits and redundant bits.

Block length: The number of bits ‘n’ after coding is called the block length of the code.

Code rate: The ration of message bits (k) and the encoder output bits (n) is called code rate. Code rate r is defined by

$$r = \frac{k}{n} \qquad 0 < r < 1$$

Channel Data Rate : It is the bit rate at the output of encoder. If the bit rate at the input of encoder is R_S, Then channel data rate will be,

$$R_o = \frac{n}{k} R_s$$

Code Vectors : An ‘n’ bit code word can be visualised in an n dimensional space as a vector whose elements are the bits in the code word. To visualise 3 bit code vectors there will be 8 different code words because of 2^k symbol.

Sl.No	Bits of Code vector		
	$b_2 = Z$	$b_1 = Y$	$b_0 = X$
1	0	0	0
2	0	0	1
3	0	1	0
4	0	1	1
5	1	0	0
6	1	0	1
7	1	1	0
8	1	1	1

Table. Code vectors in 3 dimensional space

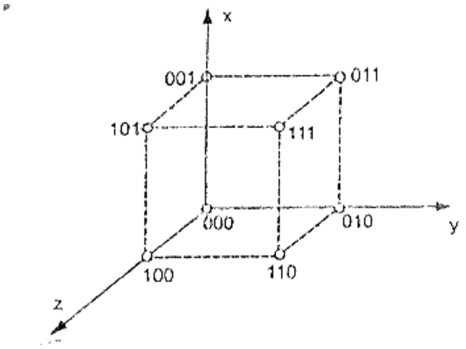


Fig. Code vectors representing 3bit codewords

Hamming Distance:

- Error control capability is determined by the Hamming distance
- The Hamming distance between two codewords is equal to the number of differences between them, e.g., 10011011, 11010010 have a Hamming distance = 3
- Alternatively, can compute by adding codewords (mod 2) =01001001 (now count up the ones)
- The maximum number of detectable errors is

$$t_{det} = d_{min} - 1$$

- That is the maximum number of correctable errors is given by,

$$t_{corr} = \frac{d_{min} - 1}{2}$$

where *d*min is the minimum Hamming distance between two codewords and t means the smallest integer.

- From the example, t_{corr} = (3-1)/2 = 1 bit error can be corrected.
- A two-bit error will cause either an undecided correction or a failed correction.
- The number of errors that can be detected is
- From the example, t_{det} = 3-1 = 2
- All two-bit errors will be detected.
- As little as a three bit error might cause a failed detection.

Code efficiency :

$$\text{Code efficiency} = \frac{\text{message bits in the block}}{\text{transmitted bits for block}} = \frac{\text{k}}{\text{n}}$$

Weight of the Code :

The number of non zero elements in the transmitted code vector is called weight of the code. It is denoted by w(X) where X is code vector.

Block Codes: (consider only binary data)

- Data is grouped into blocks of length *k* bits (dataword or message)
- Each message is coded into blocks of length *n* bits (codeword), where in general *n>k*
- This is known as (*n,k*) block code
- A vector notation is used for the message and codewords,
- Message M = (m_1 m_2….m_k)
- Codeword C = (c_1 c_2……..c_n)

Linear Block Codes:

- Sum of two codewords will produce another codeword.
- It shows that any code vector can be expressed as a linear combination of other code vector.
- Consider any code vector is having m_1, m_2, m_3 ... m_k message bits and c_1, c_2, c_3 ... c_n check bits then the code vector can be written as

$$X = (m_1, m_2, m_3 \dots m_k\ c_1, c_2, c_3 \dots c_n)$$

- Where q is the number of redundant bits added by the encoder q = n-k

- The code vector can also be written as X = (M|C)
- M = k bit message vector
- C = q bit check vector (Check bits play the role of error correction and detection)
- Code vector can be represented as X = MG
- X= Code vector of 1×n size or n bits
- M = Message vector of 1×k size or k bits
- G = Message vector of k×n size

In Matrix form $[\mathrm{X}]_{1\times \mathrm{n}} = [\mathrm{M}]_{1\times \mathrm{k}}\,[\mathrm{G}]_{\mathrm{k}\times \mathrm{n}}$ and Generate matrix G can be represented as $\mathrm{G} = \left[\mathrm{I}_{\mathrm{k}}\left[\mathrm{P}_{\mathrm{k}\times \mathrm{q}}\right]\right]_{\mathrm{k}\times \mathrm{n}}$ where I = k×k identity matrix ; P = k×q Submatrix

Check vector cab be represented $C = MP$

The expanded form is

$$\left[C_1\ C_2\ C_3 \ldots\ C_q\right]_{1\times q} = [M_1\ M_2\ M_3 \ldots\ M_k]_{1\times k} \begin{bmatrix} P_{11} & P_{12} & \ldots P_{1q} \\ P_{21} & P_{22} & \ldots P_{2q} \\ P_{k1} & P_{k2} & \ldots P_{kq} \end{bmatrix}_{k\times q}$$

By solving the above equation check vector can be obtained (additions are mod 2 addition)

$C_1 = M_1P_{11} \oplus M_2P_{21} \oplus M_3P_{31} \oplus \ldots \oplus M_kP_{k1}$

$C_2 = M_1P_{12} \oplus M_2P_{22} \oplus M_3P_{32} \oplus \ldots \oplus M_kP_{k2}$

$C_3 = M_1P_{13} \oplus M_2P_{23} \oplus M_3P_{33} \oplus \ldots \oplus M_kP_{k3}$ and So on....

Problem

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

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

Solution :

(i) <u>Determination of P Submatrix from generator matrix</u>

We know that

$$\mathrm{G} = \left[\mathrm{I}_{\mathrm{k}}\left[\mathrm{P}_{\mathrm{k}\times \mathrm{q}}\right]\right]_{\mathrm{k}\times \mathrm{n}}$$

$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ $\qquad$ $\mathrm{P}_{\mathrm{k}\times \mathrm{q}} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$

(ii) To obtain equations for check bits

Here k=3, q=3 and n=6, here the block size of message vector 3 bits. Hence there will be 8 message vector as shown in the table.

Sl.No	Message vector		
	M1	M2	M3
1	0	0	0
2	0	0	1
3	0	1	0
4	0	1	1
5	1	0	0
6	1	0	1
7	1	1	0
8	1	1	1

Then the check vector

$$[C_1\ C_2\ C_3] = [M_1\ M_2\ M_3]\begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

$C_1 = M_1 0 \oplus M_2 \oplus M_3 = M_2 \oplus M_3$

$C_2 = M_1 \oplus M_2 0 \oplus M_3 = M_1 \oplus M_3$

$C_3 = M_1 \oplus M_2 \oplus M_3 0 = M_1 \oplus M_2$

(iii) To determine check bits and code vectors for every message vector

For m1, m2, m3 = (000)

$C_1 = M_2 \oplus M_3 = 0 \oplus 0 = 0$

$C_2 = M_1 \oplus M_3 = 0 \oplus 0 = 0$

$C_3 = M_1 \oplus M_2 = 0 \oplus 0 = 0$ ie $[C_1\ C_2\ C_3] = (0\ 0\ 0)$

For (001)

$C_1 = M_2 \oplus M_3 = 0 \oplus 1 = 1$

$C_2 = M_1 \oplus M_3 = 0 \oplus 1 = 1$

$C_3 = M_1 \oplus M_2 = 0 \oplus 0 = 0$ ie $[C_1\ C_2\ C_3] = (1\ 1\ 0)$

Sl.No	Message Bits			Check bits			Complete Code Vector					
	M_1	M_2	M_3	$C_1 = M_2 \oplus M_3$	$C_2 = M_1 \oplus M_3$	$C_3 = M_1 \oplus M_2$	M_1	M_2	M_3	C_1	C_2	C_3
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

Parity Check Matrix:

For every block code the parity check matrix can be defined as

$$\mathrm{H} = \left[\mathrm{P}^{\mathrm{T}} : \mathrm{I_q}\right]_{\mathrm{q}\times\mathrm{n}}$$

Submatrix P is represented as

$$P = \begin{bmatrix} P_{11} & P_{12} & \dots\ P_{1q} \\ P_{21} & P_{22} & \dots\ P_{2q} \\ P_{k1} & P_{k2} & \dots\ P_{kq} \end{bmatrix}_{k\times q}$$

$$\mathrm{P}^{\mathrm{T}} = \begin{bmatrix} P_{11} & P_{21} & \dots\ P_{k1} \\ P_{12} & P_{22} & \dots\ P_{k2} \\ P_{1q} & P_{2q} & \dots\ P_{kq} \end{bmatrix}_{q\times k}$$

$$[\mathrm{H}]_{\mathrm{q}\times\mathrm{n}} = \begin{bmatrix} P_{11} & P_{21} & \dots\ P_{k1} & & 1 & 0 \dots & 0 \\ P_{12} & P_{22} & \dots\ P_{k2} & : & 0 & 1 \dots & 0 \\ P_{1q} & P_{2q} & \dots\ P_{kq} & & 0 & 0 \dots & 1 \end{bmatrix}_{\mathrm{q}\times\mathrm{n}}$$

Hamming Codes:

These are (n,k) linear block codes, will satisfy the following conditions.

1. Number of check bits $q \geq 3$.
2. Block length $n = 2^q - 1$
3. Number of message bits $k = n - q$
4. Minimum distance $d_{min} = 3$

We know that

Code rate $r = \frac{k}{n}$ $\quad 0 < r < 1$

$$r = \frac{n-q}{n} = 1 - \frac{q}{n} = 1 - \frac{q}{2\mathrm{q}-1}$$

Error detection and correction capabilities of hamming codes

Since d_{min} is 3 for hamming code, it can detect double errors and correct single errors.

Problem

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

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

1. Find the generator Matrix G
2. List all the code vectors
3. What is the minimum distance between code vectors
4. How many errors can be detected and how many errors can be corrected.

Solution :

Here n =7, k =4

1. Number of check bits q= n-k = 7- 4 = 3.
2. Block length n= 2^q-1 = 8-1 = 7. This shows the given code is hamming code.

(1) <u>To determine the P Submatrix</u>

The parity check matrix of q×n size is given and q = 3, n=7, k=4.

$$[\mathrm{H}]_{3\times7} = \begin{bmatrix} P_{11} & P_{21} & P_{31} & \mathrm{P}_{41} \ldots & 1 & 0 & 0 \\ P_{12} & P_{22} & P_{32} & \mathrm{P}_{42} \ldots & 0 & 1 & 0 \\ P_{13} & P_{23} & P_{33} & \mathrm{P}_{43} \ldots & 0 & 0 & 1 \end{bmatrix}_{3\times7}$$

$$\mathrm{H} = [\mathrm{P}^{\mathrm{T}} : \mathrm{I}_3]$$

$$\mathrm{P}^{\mathrm{T}} = \begin{bmatrix} P_{11} & P_{21} & P_{31} & \mathrm{P}_{41} \\ P_{12} & P_{22} & P_{32} & \mathrm{P}_{42} \\ P_{13} & P_{23} & P_{33} & \mathrm{P}_{43} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}$$

Therefore

$$P = \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \\ p_{41} & p_{42} & p_{43} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

(2) <u>To obtain generator matrix G</u>

$$\mathrm{G} = \left[\mathrm{I}_{\mathrm{k}} : [\mathrm{P}_{\mathrm{k}\times\mathrm{q}}]\right]_{\mathrm{k}\times\mathrm{n}} \qquad \mathrm{G} = \left[\mathrm{I}_4 : [\mathrm{P}_{4\times3}]\right]_{4\times7}$$

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

(3) To find all Code words

Codeword C = MP

$$\left[C_1\ C_2\ C_3 \ldots\ C_q\right]_{1\times q} = [M_1\ M_2\ M_3 \ldots\ M_k]_{1\times k} \begin{bmatrix} P_{11} & P_{12} & \ldots\ P_{1q} \\ P_{21} & P_{22} & \ldots\ P_{2q} \\ P_{k1} & P_{k2} & \ldots\ P_{kq} \end{bmatrix}_{k\times q}$$

$$[C_1\ C_2\ C_3]_{1\times3} = [M_1\ M_2\ M_3\ M_4]_{1\times4} \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \\ p_{41} & p_{42} & p_{43} \end{bmatrix}_{4\times3}$$

$$[C_1\ C_2\ C_3]_{1\times3} = \ [M_1\ M_2\ M_3\ M_4]_{1\times4}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 0\\ 1 & 0 & 1\\ 0 & 1 & 1\end{bmatrix}_{4\times3}$$

$$C_1 = \ M_1 1 \oplus \ M_2 1 \oplus \ M_3 1 \oplus M_4 0 \ = \ M_1 \oplus \ M_2 \ \oplus \ M_3$$

$$C_2 = \ M_1 1 \oplus \ M_2 1 \oplus \ M_3 0 \oplus M_4 1 \ = \ M_1 \oplus \ M_2 \ \oplus \ M_4$$

$$C_3 = \ M_1 1 \oplus \ M_2 0 \oplus \ M_3 1 \oplus M_4 1 \ = \ M_1 \oplus \ M_3 \ \oplus \ M_4$$

For example if (m1, m2, m3, m4) = (1011)

$$C_1 = \ \ M_1 \oplus \ M_2 \ \oplus \ M_3 \ = 1 \oplus \ 0 \ \oplus \ 1 \ = 0$$

$$C_2 = \ \ M_1 \oplus \ M_2 \ \oplus \ M_4 \ = 1 \oplus \ 0 \ \oplus \ 1 \ = 0$$

$$C_3 = \ \ M_1 \oplus \ M_3 \ \oplus \ M_4 \ \ = 1 \oplus \ 1 \ \oplus \ 1 \ = 1$$

Sl.No	Message Bits				Check bits			Complete Code Vector							Weight of the code vector
	M_1	M_2	M_3	M_4	C_1	C_2	C_3	M_1	M_2	M_3	M_4	C_1	C_2	C_3	
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

(4) Minimum distance between code vectors

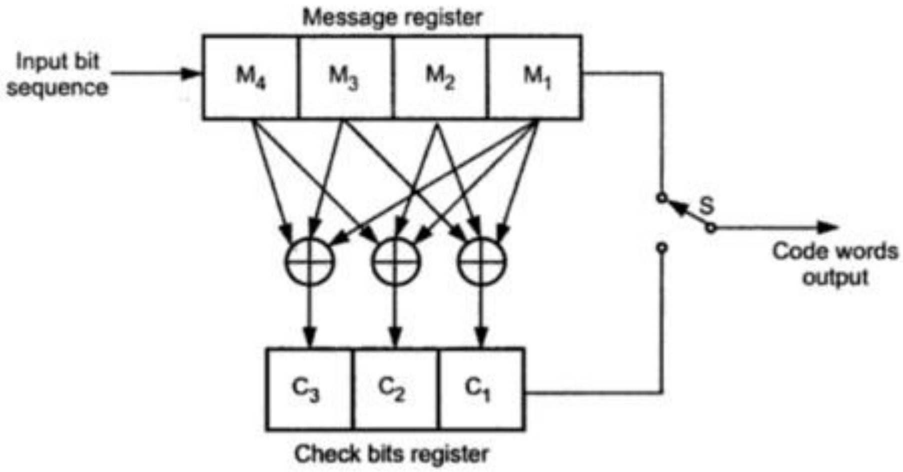
$2^k = 2^8 = 16$ code vectors along with their weights**.** The smallest weight of any non zero code vector is 3 therefore the minimum distance **d_{min} =3.**

(5) Error correction and Detection capabilities

d_{min} =3

$d_{min} \geq s+1$ ➔ $3 \geq s+1$ therefore $s \leq 2$ thus two errors will be detected. And
$d_{min} \geq 2t+1$ ➔ $3 \geq s+1$ therefore $t \leq 1$ thus one errors will be corrected.

Encoder of (7,4) Hamming code



Definition of Syndrome:

When Some errors are present in received vector Y then it will be not from the valid vector and it will not satisfy the property

$$if\ XH^T\ = (0\ 0\ 0\ ....\ .0) and\ \ YH^T\ = (0\ 0\ 0\ ....\ .0)$$

then X=Y i.e no errors or Y is valid code vector or

$if\ XH^T\ =\ \ YH^T\ = non\ zero$ then X≠Y i.e some errors.

The non zero output of the product of YH^T is called as syndrome and it is used to detect the errors in Y. Syndrome represented by S cand be written as

$[S]_{1\times q} = \ [Y]_{1\times n}[\,H^T]_{n\times q}$ and $\quad Y = X \oplus E \ \ \& \ X = Y \oplus E$

Relationship between syndrome vector (S)and Error vector (E)

$$\mathrm{S} = \mathrm{YH^T} \ = (X \oplus E)H^T = XH^T \ \oplus \ EH^T$$

$$S \ = EH^T \ since \ XH^T \ = 0$$

Detecting error with the help of syndrome:

Problem

The parity check parity matrix of (7,4) block code is given as

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

Calculate the syndrome vector.

(1) To determine error pattern for single bit erros

Syndrome is 3 bit vector, here q=3. Therefore 2^q-1= 7 non zero syndrome. This shows that 7 single bit error pattern will be represented by these 7 non zero syndrome. Error vector E is a n bit vector representing error pattern.

Sl.No	Bit in error	Bits of vector (E), Non zero bits shows error						
1	1st	1	0	0	0	0	0	0
2	2nd	0	1	0	0	0	0	0
3	3rd	0	0	1	0	0	0	0
4	4th	0	0	0	1	0	0	0
5	5th	0	0	0	0	1	0	0
6	6th	0	0	0	0	0	1	0
7	7th	0	0	0	0	0	0	1

(2) Calculation of Syndrome

$$[S]_{1\times q} = \ [Y]_{1\times n}[\,H^T]_{n\times q} \ \text{ and } \quad [S]_{1\times 3} = \ [Y]_{1\times 7}[\,H^T]_{7\times 3}$$

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

For example Syndrome of first bit error is

$$S = EH^T = (1000000)\begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$S = (\,1 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \quad 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \quad 1 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0\,)$$

$$\mathrm{S} = (1\ 0\ 1)$$

Syndrome of second bit error is

$$S = EH^T = (0100000)\begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$S = (\,0 \oplus 1 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \quad 0 \oplus 1 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \quad 0 \oplus 1 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0\,)$$

$$\mathrm{S} = (1\ 1\ 1)$$

Syndrome vector are rows of H^T

Sl.No	Bits of vector (E), Non zero bits shows error							Syndrome vector			
1	0	0	0	0	0	0	0	0	0	0	
2	1	0	0	0	0	0	0	1	0	1	1st of H^T
3	0	1	0	0	0	0	0	1	1	1	2nd of H^T
4	0	0	1	0	0	0	0	1	1	0	3rd of H^T
5	0	0	0	1	0	0	0	0	1	1	4th of H^T
6	0	0	0	0	1	0	0	1	0	0	5th of H^T
7	0	0	0	0	0	1	0	0	1	0	6th of H^T
8	0	0	0	0	0	0	1	0	0	1	7th of H^T

Error correctin using syndrome vector

Let us consider above (7,4) block code and the Code vector

$$X = (1\ 0\ 0\ 1\ 1\ 1\ 0\ )$$

Let the error created in 3rd bit so

$$Y = (1\ 0\ (\mathbf{1})\ 1\ 1\ 1\ 0\ )$$

Now error correction can be done by adopting following steps

(1) Calculate the syndrome S = YH^T
(2) Check the row of H^T which is same as of S
(3) For Pth of row of H^T , Pth bit is in error. Hence write the corresponding error vector E.
(4) Obtain the correct vector by $X\ =\ Y \oplus E$

(1) To obtain syndrome Vector

$$S = YH^T$$

$$S = (1\ 0\ 1\ 1\ 1\ 1\ 0\ )\begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = (1\ 1\ 0)$$

(2) To determine the row of H^T which is same as of S = 3rd row
(3) To determine E ; E = (0 0 1 0 0 0 0)

(4) Correct Vector X

$$X = Y \oplus E$$

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

Thus the single bit error can be corrected using syndrome.

If double error occurs:

Consider the same message vector

$$X = (1\ 0\ 0\ 1\ 1\ 1\ 0\ )$$

and

$$Y = (1\ 0\ (\mathbf{1})\ (\mathbf{0})\ 1\ 1\ 0\ )$$

$$S = YH^T$$

$$S = (1\ 0\ 1\ 0\ 1\ 1\ 0\ )\begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = (1\ 0\ 1)$$

S is equal to row of H^T which is same as of S = 1st row therefore E = 1000000. Thus the error correction and detection goes wrong. The probability occurrence of multiple errors is less compared to single errors. To correct multiple errors extended hamming codes are used. In these codes one extra bit is provided to correct double errors. We know that for (n,k) block codes there are 2^n-1distinct non zero syndromes. There are nC_1 = n single error pattern, nC_2 double error pattern, nC_3 triple error pattern and so on. Therefore to correct t error pattern

$$2^q - 1 \geq n\,C_1 + n\,C_2 + n\,C_3 + \ \ldots\ldots + n\,C_t$$

Hamming Bound

$$2^q \geq 1 + n\,C_1 + n\,C_2 + n\,C_3 + \ \ldots\ldots + n\,C_t$$

$$2^q \geq \sum_{i=0}^{t} n\, C_i$$

$$2^{n-k} \geq \sum_{i=0}^{t} n\, C_i$$

By taking logarithmic on base 2 on both sides

$$n - k \geq log_2 \left(\sum_{i=0}^{t} n\, C_i \right)$$

$$1 - \frac{k}{n} \geq \frac{1}{n} log_2 \left(\sum_{i=0}^{t} n\, C_i \right)$$

$Coding\ rate\ \frac{k}{n} = r$

$$1 - r \geq \frac{1}{n} log_2 \left(\sum_{i=0}^{t} n\, C_i \right)$$

Problem:

For a linear block code provide with an example that

1. Syndrome depends only on error pattern not on transmitted codeword.
2. All error pattern the differ by a codeword have the same syndrome.

Solution:

Syndrome depends only on error pattern not on transmitted codeword.

We know that $S = EH^T$

This equation shows that Syndrome depends only on error pattern not on transmitted codeword.

All error pattern the differ by a codeword have the same syndrome.

Syndrome for the first received code

$$\mathrm{S}_1 = \mathrm{Y}_1\mathrm{H}^\mathrm{T} = (X_1 \oplus E)H^T = X_1H^T \oplus EH^T$$

$$S_1 \ = EH^T \ since \ X_1H^T \ = 0$$

Syndrome for the Second received code

$$\mathrm{S}_2 = \mathrm{Y}_2\mathrm{H}^\mathrm{T} \ = (X_2 \oplus E)H^T = X_2H^T \ \oplus \ EH^T$$

$$S_2 = EH^T \ since \ X_2H^T \ = 0$$

For example let us consider

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

Consider two code words

$\mathrm{X}_2 = 0\ 0\ 0\ 1\ 0\ 1\ 1\ ;\ \mathrm{X}_3 = 0\ 0\ 1\ 0\ 1\ 0\ 1$

Error is introduced in MSB

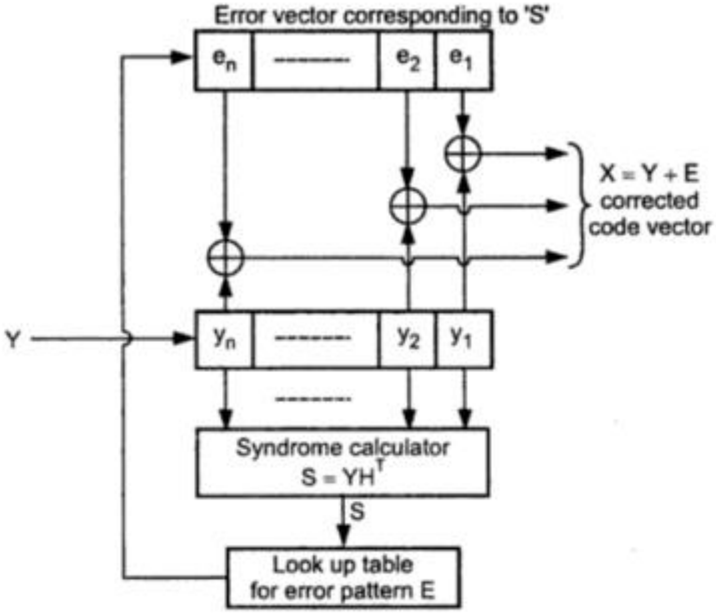
$\mathrm{Y}_2 = (\mathbf{1})0\ 0\ 1\ 0\ 1\ 1\ ;\ \mathrm{Y}_3 = (\mathbf{1})\ 0\ 1\ 0\ 1\ 0\ 1$

$$\mathrm{S}_2 = \mathrm{Y}_2\mathrm{H}^\mathrm{T} = (\ \mathbf{1}\ 0\ 0\ 1\ 0\ 1\ 1) \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = (\ 1\ 1\ 1)$$

$$\mathrm{S}_3 = \mathrm{Y}_3\mathrm{H}^\mathrm{T} = (\ \mathbf{1}\ 0\ 1\ 0\ 1\ 0\ 1) \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = (\ 1\ 1\ 1)$$

Thus the syndrome $S_2 = S_3 = (1\ 1\ 1)$ even if two codewords are different. This proves all error pattern the differ by a codeword have the same syndrome.

Syndrome decoder for (n,k) block code



Other linear codes:

Single Parity bit code:

If there are m_1, m_2, m_3 ... m_k are the bits of the k bit message word, then $m_1 \oplus m_2 \oplus m_3 \oplus \ldots\ldots \oplus m_k \oplus c_1 = 0$ In the above equation C_1 is the parity bit added with the message. If there are even number of parity check bit $C_1 = 0$ and vice versa. For this code the transmitted bits are n = k+1 and q=1. This code can correct single bit.

Repeated codes

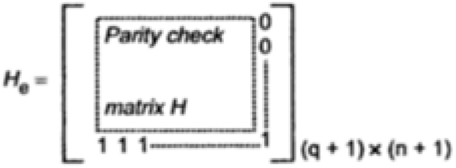
In this code , a single message bit is transmitted and q=2t bit are the parity bit and k=1, then the transmitted bits are n = 2t+1. This code can correct t errors per block. It requires large band width.

Hadamard Code

It is derived from hadamard matrix here n = 2^k and q = n-k = 2^k –k. the code rate is $r = \frac{k}{n} = \frac{k}{2^k}$ This shows the code rate will be very small.

Extended Code

(n,k) linear block code has a parity check matrix H. One column of zero elements (except last element) and one row of 1's is added to the parity check matrix as shown below



The code turned by such parity is called extended code. Thus it will be described as (n+1, k) linear block code. Now the parity check matrix size is (q+1) by (n+1). For example (7,4) matrix is represented as below with the extended parity bits and the minimum distance for extended code is

$$d_{e\,(min)} = d_{min} + 1$$

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}_{3\times 7}$$

The parity check matrix for extended code will be

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

Newly added row and column.

Dual code :

Consider an (n,k) block code. This code satisfies $HG^T = 0$, then the (n, n-k) i.e (n,q) block code called as dual code. The generator matrix and parity matrix will be,

$[G]_{q\times n} = \left[I_{q\times q} \mid p_{q\times n}\right]_{q\times n}$ and $[H]_{k\times n} = \left[p^T{}_{k\times q} \mid I_{k\times k}\right]_{k\times n}$

Cyclic Codes :

Cylic codes are the sub class of linear code. A code *C* is cyclic if
(i) *C* is a linear code;
(ii) any cyclic shift of a codeword is also a codeword.

- Systematic cyclic codes
- Non Systematic cyclic codes.

Properties :

Linearity property :- Sum of any two codewords is also a valid codeword. For example X_1 and X_2 is a codeword then $X_3 = X_1 \oplus X_2$ Here X_3 is also a codeword. This property shows cyclic code is also a linear code.

Cyclic property :- Every cyclic shift of the valid code vector produces another code vector. Let us consider n bit code vector

$$X = \{x_{n-1}, x_{n-2}, \dots\ x_1, x_0\}$$

Here $x_{n-1}, x_{n-2}, \dots\ x_1, x_0$ represents the individual bits of code vectors. Let us shift the above code vector cyclically to the left side

$$X' = \{x_{n-2}, x_{n-3}, \dots\ x_1, x_0, x_{n-1}\}$$

Represents of codewords by a polynomial:

This code word will be represents by a polynomial of a degree less than or equal to (n-1)

$$X(p) = \{x_{n-1}\,p^{n-1} + \ x_{n-2}\,p^{n-2} + \ \dots + x_1 p + x_0\}$$

Here X(p) is the polynomial of degree (n-1), P is the arbitrary variable of the polynomial
The power of ‘p’ represents positions of the codeword.

$$p^{n-1}\ represents\ MSB$$

$$p^{0}\ represents\ LSB$$

$$p^{1}\ represents\ Second\ bit\ from\ LSB\ side$$

Polynomial represents is due to following reasons

1. These are algebraic codes. The algebraic operations such as addition, Subtraction, Multiplication and division etc becomes very simple.
2. Position of the bits are represented with the help of powers of ‘p’ in a polynomial.

Generation of code vectors in nonsystematic Form

Let $M = \{m_{k-1}, m_{k-2}, \dots\, m_1, m_0\}$ be k bits of message vector. Then it can be represented by the polynomial as

$$M(p) = m_{k-1}p^{k-1} + m_{k-2}\, p^{k-2} + \dots \dots \dots + m_1 p + m_0$$

Let X(p) represent the codeword polynomial. It is given as

$$X(p) = M(p)G(p)$$

Here G(p) is the generating polynomial of degree 'q'. For an (n,k) cyclic code, q=n-k represent the number of parity bits. The generating polynomial is given as,

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

Here $g_{q-1}\,, g_{q-2}\, \dots \dots g_1$ are the parity bits.

If M1, M2, M3 … etc are the other message vectors, then the corresponding code vectors can be calculated as

$$X_1(p) = M_1(p)G(p)$$

$$X_2(p) = M_2(p)G(p)$$

$X_3(p) = M_3(p)G(p)$ and So on. All the above code vectors in non systematic form and they satisfy cyclic property. Note that G(p) is same for all code vectors.

Problem

The generator polynomial of (7,4) cyclic code is $G(p) = p^3 + p + 1$ find all code vectors in nonsystematic form.

Solution :

Here n =7, k =4 , Number of check bits q= n-k = 7- 4 = 3 and Block length n= 2^q-1 = 8-1 = 7.

For ex Consider any message vector M = (m_3, m_2, m_1, m_0) = (0 1 0 1)

Then the message polynomial will be

$$M(p) = m_{k-1}p^{k-1} + m_{k-2}\, p^{k-2} + \dots \dots \dots + m_1 p + m_0$$

$M(p) = m_3 p^3 + m_2\, p^2 + m_1 p + m_0 = p^2 + 1$

and the given generator polynomial is $G(p) = p^3 + p + 1$

To obtain non systematic code vectors

$$X(p) = M(p)G(p)$$

$$X(p) = (\,p^2 + 1\,)(p^3 + p + 1)$$

$$X(p) = (\mathrm{p}^5 + \ p^3 + p^2 + p^3 + p + 1)$$

$$X(p) = (\mathrm{p}^5 + \ p^3 + p^3 + p^2 + p + 1)$$

$$X(p) = (\mathrm{p}^5 + (1 \oplus 1)\, p^3 + p^2 + p + 1)$$

$$X(p) = (0\mathrm{p}^6 + \mathrm{p}^5 + 0\mathrm{p}^4 + 0p^3 + p^2 + p + 1)$$

Note that the degree of above polynomial is n-1 = 6, The code vector of above polynomial is

$$X = \ (x_6 x_5 x_4 x_3 x_2 x_1 x_0) = (0\ 1\ 0\ 0\ 1\ 1\ 1)$$

List of code vectors in non systematic form is

Sl.No	Message Bits				Complete Code Vector						
	M_3	M_2	M_1	M_0	X_6	X_5	X_4	X_3	X_2	X_1	X_0
1	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	1	0	0	0	1	0	1	1
3	0	0	1	0	0	0	1	0	1	1	0
4	0	0	1	1	0	0	1	1	1	0	1
5	0	1	0	0	0	1	0	1	1	0	0
6	0	1	0	1	0	1	0	0	1	1	1
7	0	1	1	0	0	1	1	1	0	1	0
8	0	1	1	1	0	1	1	0	0	0	1
9	1	0	0	0	1	0	1	1	0	0	0
10	1	0	0	1	1	0	1	0	0	1	1
11	1	0	1	0	1	0	0	1	1	1	0
12	1	0	1	1	1	0	0	0	1	0	1
13	1	1	0	0	1	1	1	0	1	0	0
14	1	1	0	1	1	1	1	1	1	1	1
15	1	1	1	0	1	1	0	0	0	1	0
16	1	1	1	1	1	1	0	1	0	0	1

Cyclic properties:

Let us consider Xg from above table is $X_g = (0\ 1\ 1\ 0\ 0\ 0\ 1\ )$

Shift this code vector cyclically by left side one

$$X_g{}' = (1\ 0\ 1\ 1\ 0\ 0\ 0\ )$$

Thus cyclic shift of X_8 produces X_9

Generation of code vectors in systematic form

The Systematic form of the block code is,

$X = (k\ message : (n-k) check\ bits) = \ m_{k-1}, m_{k-2}, \dots\ m_1, m_0 :\ c_{q-1}, c_{q-2}, \dots\ c_1, c_0$

Here the check bits forms the polynomial as

$C(p) = c_{q-1}p^{q-1} + c_{q-2}\, p^{q-2} +\ \dots\dots\dots +\ c_1 p +\ c_0$

The check bit polynomial is

$$C(p) =\ rem\ \left[\frac{p^q M(p)}{G(p)}\right]$$

Problem

The generator polynomial of (7,4) cyclic code is $G(p) = p^3 + p + 1$ find all code vectors in systematic form.

Solution :

Here n =7, k =4 , Number of check bits q= n-k = 7- 4 = 3 and Block length n= 2^q-1 = 8-1 = 7.

For ex Consider any message vector M = (m_3, m_2, m_1, m_0) = (0 1 0 1)

Then the message polynomial will be

$$M(p) = m_{k-1}p^{k-1} + m_{k-2}\, p^{k-2} +\ \dots\dots\dots +\ m_1 p +\ m_0$$

$M(p) = m_3 p^3 + m_2\, p^2 + m_1 p +\ m_0 =\ \ p^2 + 1$

and the given generator polynomial is $G(p) = p^3 + p + 1$

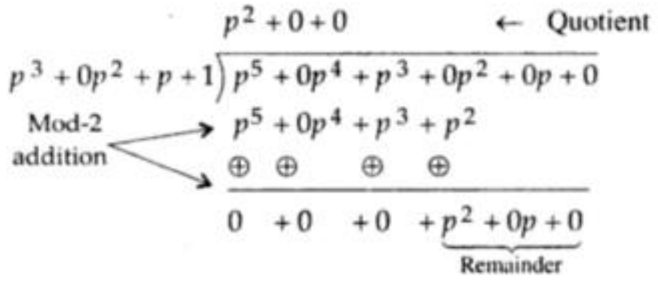
To obtain systematic code vectors

$$C(p) = rem\left[\frac{p^q M(p)}{G(p)}\right]$$

To obtain $p^q M(p)$ Since q =3

$$p^q M(p) = p^3(p^2+1) = p^5 + 0\,p^4 + p^3 + 0p^2 + 0p + 0$$

$$\left[\frac{p^q M(p)}{G(p)}\right] = \frac{p^5 + 0\,p^4 + p^3 + 0p^2 + 0p + 0}{p^3 + p + 1}$$



$$\begin{array}{r l}
 & p^2 + 0 + 0 \quad \leftarrow \text{Quotient} \\
p^3 + 0p^2 + p + 1 \,\big) & p^5 + 0p^4 + p^3 + 0p^2 + 0p + 0 \\
\text{Mod-2} \rightarrow & p^5 + 0p^4 + p^3 + p^2 \\
\text{addition} \rightarrow & \oplus \quad \oplus \quad \oplus \quad \oplus \\
\hline
 & 0 \quad +0 \quad +0 \quad + \underbrace{p^2 + 0p + 0}_{\text{Remainder}}
\end{array}$$

$$Therefore\ C(p) = p^2 + 0p + 0$$

Since q =3, the check bits are C = (c$_2$, c$_1$, c$_0$) = (1 0 0)

The code vector can be written as

$$X = (\, m_{k-1}, m_{k-2}, \ldots\ m_1, m_0 :\ c_{q-1}, c_{q-2}, \ldots\ c_1, c_0)$$

$$X = \ (m_3, m_2, m_1, m_0 :\ c_2, c_1, c_0) = (0101 : 100)$$

List of code vectors in non systematic form is

Sl.No	Message Bits				Complete Code Vector						
	M_3	M_2	M_1	M_0	M_3	M_2	M_1	M_0	C_2	C_1	C_0
1	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	1	0	0	0	1	0	1	1
3	0	0	1	0	0	0	1	0	1	1	0
4	0	0	1	1	0	0	1	1	1	0	1
5	0	1	0	0	0	1	0	0	1	1	1
6	0	1	0	1	0	1	0	1	1	0	0
7	0	1	1	0	0	1	1	0	0	0	1
8	0	1	1	1	0	1	1	1	0	1	0
9	1	0	0	0	1	0	0	0	1	0	1
10	1	0	0	1	1	0	0	1	1	1	0
11	1	0	1	0	1	0	1	0	0	1	1
12	1	0	1	1	1	0	1	1	0	0	0
13	1	1	0	0	1	1	0	0	0	1	0
14	1	1	0	1	1	1	0	1	0	0	1
15	1	1	1	0	1	1	1	0	1	0	0
16	1	1	1	1	1	1	1	1	1	1	1

Problem :

An n digit code polynomial X(p) is obtained as $X(p) = C(p)p^{n-k}M(p)$ where M(p) is message polynomial with k bit data and C(P) is the remainder polynomial obtained by dividing $p^{n-k}M(p)$ by proper generator polynomial G(p). prove that X(p) is systematic cyclic code if G(p) is the factor of p^{n+1} in modulo 2 sense

Generator and parity matrix of a cyclic code

Non Systematic form

The generator matrix has the size of k×n is given by

$$G(p) = p^q + g_{q-1}\, p^{q-1} + \ldots\ldots\ldots + g_1 p + 1$$

Multiply both sides by p^i

$G(p)p^i = p^{i+q} + g_{q-1}\, p^{i+q-1} + \ldots\ldots\ldots + g_1 p^{i+1} + 1$ and i= (k-1), (k-2), …..2,1,0

Problem : -

Obtain the generator matrix of (7,4) cyclic code if $G(p) = p^3 + p + 1$. and the code vectors.

Solution :

Here n =7, k =4 , Number of check bits q= n-k = 7- 4 = 3, $G(p)p^i$ will be

$$G(p)p^i = p^{i+3} + p^{i+2} + p^i \; gor \; given \; G(p) \; since \; k - 1 = 3, i = 3{,}2{,}1{,}0$$

There will be four polynomials corresponding to 4 values of y. These four polynomials represent rows of generator matrix

$$\left.\begin{array}{ll}
\text{For row 1 : } i = 3 \Rightarrow & p^3\,G(p) = p^6 + p^5 + p^3 \\
\text{For row 2 : } i = 2 \Rightarrow & p^2\,G(p) = p^5 + p^4 + p^2 \\
\text{For row 3 : } i = 1 \Rightarrow & p\,G(p) = p^4 + p^3 + p \\
\text{For row 4 : } i = 0 \Rightarrow & G(p) = p^3 + p^2 + 1
\end{array}\right\}$$

Polynomials for above four row is

$$\left.\begin{array}{lcrcl}
\text{Row 1} & \Rightarrow & p^3\,G(p) & = & p^6 + p^5 + 0p^4 + p^3 + 0p^2 + 0p + 0 \\
\text{Row 2} & \Rightarrow & p^2\,G(p) & = & 0p^6 + p^5 + p^4 + 0p^3 + p^2 + 0p + 0 \\
\text{Row 3} & \Rightarrow & p\,G(p) & = & 0p^6 + 0p^5 + p^4 + p^3 + 0p^2 + p + 0 \\
\text{Row 4} & \Rightarrow & G(p) & = & 0p^6 + 0p^5 + 0p^4 + p^3 + p^2 + 0p + 1
\end{array}\right\}$$

Then the generater matrix is

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

To obtain code vectors X =MG:

$$X = \;(m_3, m_2, m_1, m_0 \,:\, G) = \left(1001 \,:\, \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}\right) = (1010011)$$

(follow the same procedure to find the code vectors of others)

Systematic generator matrix

The systematic form of generator matrix is given by

$$G = \left[I_k : P_{k\times q}\right]_{k\times n}$$

The t^th row of this matrix will be represented in the polynomial form as,

$$t^{th}\, row\ of\ G = p^{n-t} + R_t(p)\ where\ t = 1,2,3,\ldots k$$

$$\frac{p^{n-t}}{G(p)} = Quotient\ Q_t(p) + \frac{Reminder R_t(p)}{G(p)}$$

$$i.e\ p^{n-t} = Q_t(p)G(p) \oplus R_t(p)\ and\ t = 1,2,\ldots.k$$

Parity check matrix

Find out the the generator matrix for a systematic (7,4) cyclic code if $G(p) = p^3 + p + 1$. Also find the parity matrix.

Solution :

To obtain the generator polynomial

$$t^{th}\, row\ of\ Generator\ polynomial\ is$$

$$p^{n-t} + R_t(p) = Q_t(p)G\,(p) \qquad where\ t = 1,2,3,\ldots k$$

Given=7, k=4 and q=n-k =3

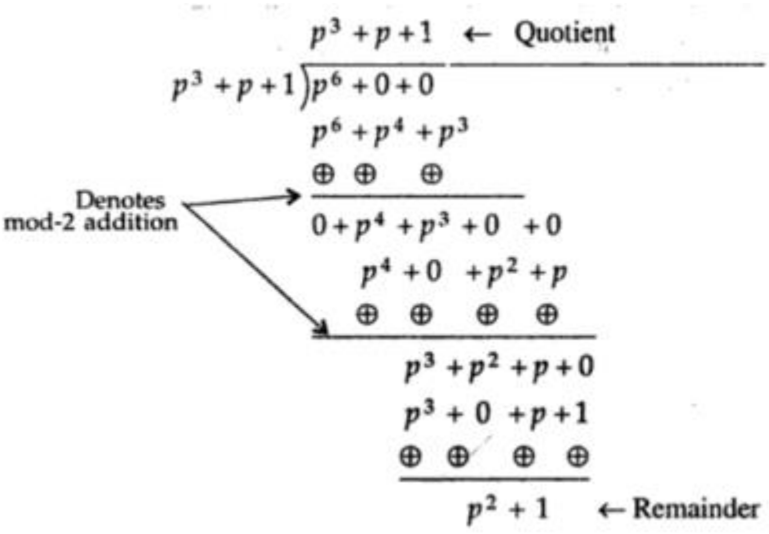
The above equation will be

$$p^{7-t} + R_t(p) = Q_t(p)(\,p^3 + p + 1) \quad where\ t = 1,2,3,4$$

With t=1, the above equation becomes

$$p^6 + R_t(p) = Q_t(p)(\,p^3 + p + 1)$$

To obtain $R_t(p), Q_t(p)$ for 1st Row



$$\begin{array}{r l}
 & p^3+p+1 \leftarrow \text{Quotient}\\
p^3+p+1 & \overline{)\,p^6+0+0}\\
 & p^6+p^4+p^3\\
 & \oplus\ \oplus\ \ \oplus\\
\hline
 & 0+p^4+p^3+0\ +0\\
 & \quad p^4+0\ +p^2+p\\
 & \quad \oplus\ \oplus\ \oplus\ \oplus\\
\hline
 & \qquad p^3+p^2+p+0\\
 & \qquad p^3+0\ +p+1\\
 & \qquad \oplus\ \oplus\ \oplus\ \oplus\\
\hline
 & \qquad\quad p^2+1 \leftarrow \text{Remainder}
\end{array}$$

Here

$$Q_t(p) = (p^3+p+1) and\ R_t(p) = (p^2+1)$$

Putting those values

$$(p^6+p^2+1) = (p^3+p+1)(p^3+p+1)$$

1st row of polynomial is $= p^6+p^2+1$

Other row polynomials

$$\left.\begin{array}{l}
t=2 \Rightarrow 2^{nd} \text{ row polynomial} = p^5+p^2+p+1\\
t=3 \Rightarrow 3^{rd} \text{ row polynomial} = p^4+p^2+p\\
t=4 \Rightarrow 4^{th} \text{ row polynomial} = p^3+p+1
\end{array}\right\}$$

Conversion of row polynomials into matrix forms generator matrix

$$G = \begin{array}{c c}
 & \begin{array}{ccccccc} p^6 & p^5 & p^4 & p^3 & p^2 & p^1 & p^0 \end{array}\\
\begin{array}{c} Row\,1\\ Row\,2\\ Row\,3\\ Row\,4 \end{array} & \left[\underbrace{\begin{array}{cccc} 1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1 \end{array}}_{I_3} \begin{array}{c} :\\ :\\ :\\ : \end{array} \underbrace{\begin{array}{ccc} 1 & 0 & 1\\ 1 & 1 & 1\\ 1 & 1 & 0\\ 0 & 1 & 1 \end{array}}_{P_{4\times3}}\right]_{4\times7}
\end{array}$$

To obtain code vectors X =MG:

$$X = \ (m_3, m_2, m_1, m_0 \ : \ G) = \left(1100 \ : \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \right) = (1100010)$$

(follow the same procedure to find the code vectors of others)

To obtain parity check matrix (H)

The systematic form of generator matrix is given by

$$G = \ [I_k \ : \ P_{k\times q}]_{k\times n}$$

The P Submatrix

$$p = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}_{4\times 3}$$

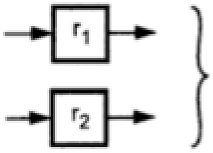
The parity check matrix is given

$$H = \ [p^T \ : \ I_q]_{q\times n}$$

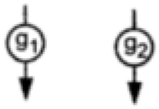
$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & : & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & : & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & : & 0 & 0 & 1 \end{bmatrix}_{3\times 7}$$

$$\underbrace{}_{P^T} \quad \underbrace{}_{I_3}$$

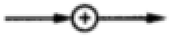
ENCODERS AND DECODERS



These are the flipflops. Used to make Shift register.



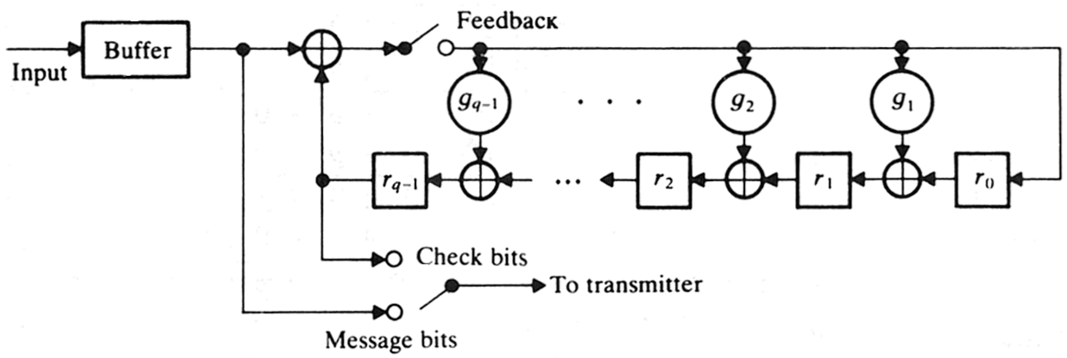
if g=1 closed path and if g=0 open path.



Mod 2 addition

Operation :

- Feedback switch is closed first.
- Output switch is connected in the message input.
- All shift registers are initialized to zero.
- K messages are shifted to transmitter as well as shifted into the registers.
- After the shift of k message bits registers contain ‘q’ check bits.
- The feedback switch is now opened and output switch is connected to check bits position.

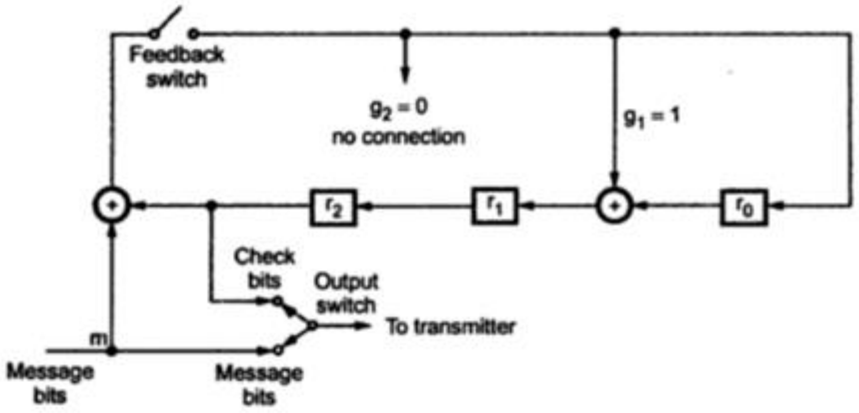


Problem : -

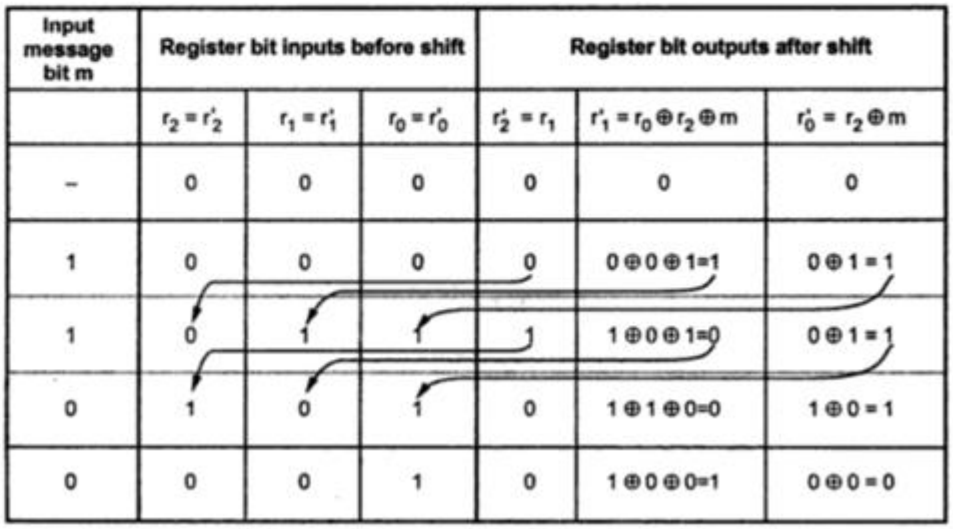
Design the encoder for (7,4) cyclic code generator polynomial is $G(p) = p^3 + p + 1$ and verify its operation for any message vector.

Solution:

The generator polynomial is $G(p) = p^3 + 0p^2 + p + 1 = p^3 + g_2p^2 + g_1p + 1$ and q=3;

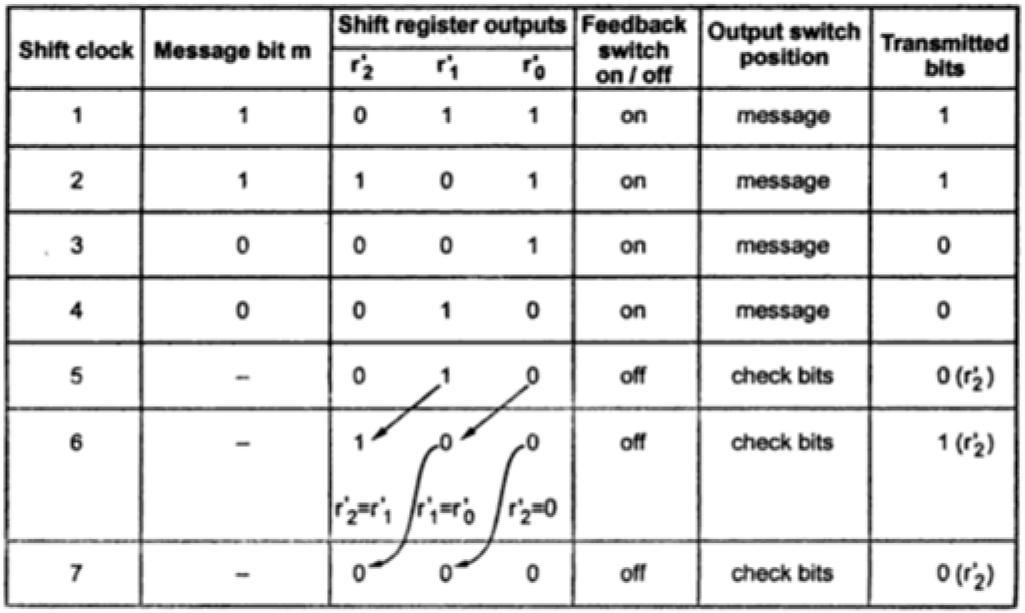


Lets verify the operation for message vector v $(m_3, m_2, m_1, m_0\ ) = (1100\ )$



Input message bit m	Register bit inputs before shift			Register bit outputs after shift		
	$r_2 = r'_2$	$r_1 = r'_1$	$r_0 = r'_0$	$r'_2 = r_1$	$r'_1 = r_0 \oplus r_2 \oplus m$	$r'_0 = r_2 \oplus m$
–	0	0	0	0	0	0
1	0	0	0	0	$0 \oplus 0 \oplus 1 = 1$	$0 \oplus 1 = 1$
1	0	1	1	1	$1 \oplus 0 \oplus 1 = 0$	$0 \oplus 1 = 1$
0	1	0	1	0	$1 \oplus 1 \oplus 0 = 0$	$1 \oplus 0 = 1$
0	0	0	1	0	$1 \oplus 0 \oplus 0 = 1$	$0 \oplus 0 = 0$

$$X = \ (m_3, m_2, m_1, m_0, c_1, c_2, c_3\ ) = (1100010\ )$$



Shift clock	Message bit m	Shift register outputs			Feedback switch on / off	Output switch position	Transmitted bits
		r'_2	r'_1	r'_0			
1	1	0	1	1	on	message	1
2	1	1	0	1	on	message	1
3	0	0	0	1	on	message	0
4	0	0	1	0	on	message	0
5	–	0	1	0	off	check bits	0 (r'_2)
6	–	1 $r'_2 = r'_1$	0 $r'_1 = r'_0$	0 $r'_2 = 0$	off	check bits	1 (r'_2)
7	–	0	0	0	off	check bits	0 (r'_2)

Syndrome Decoding, Error Detection and Error Correction

We know that $X\ = Y \oplus E$ and $Y\ = X \oplus E$

In the polynomial $Y(p) = X(p) \oplus E(p)$

$$X(p) = M(p)G(p)$$

$$Y(p) = M(p)G(p) \oplus E(p)$$

Let the received polynomial

$$\frac{Y(p)}{G(p)} = Quotient + \frac{Reminder}{G(p)}$$

If Y(p) = X(p), there is no errors then

$$\frac{X(p)}{G(p)} = Quotient + \frac{Reminder}{G(p)}$$

If Y(p) ≠ X(p), there is errors then

$$\frac{Y(p)}{G(p)} = Quotient + \frac{Reminder}{G(p)} = Q(p) + \frac{R(p)}{G(p)}$$

R(p) will be the polynomial of degree less than or equal to q-1. Multyply both sides of above equation by G(p)

$$Y(p) = Q(p)G(p) + R(p)$$

$$M(p)G(p) \oplus E(p) = E(p)G(p) \oplus R(p)$$

$$M(p)G(p) \oplus E(p) = E(p)G(p) \oplus R(p)$$

$$E(p) = M(p)G(p) \oplus Q(p)G(p) \oplus R(p)$$

$$E(p) = M(p)G(p) \oplus Q(p)G(p) \oplus R(p)$$

$$E(p) = [M(p) \oplus Q(p)]G(p) \oplus R(p)$$

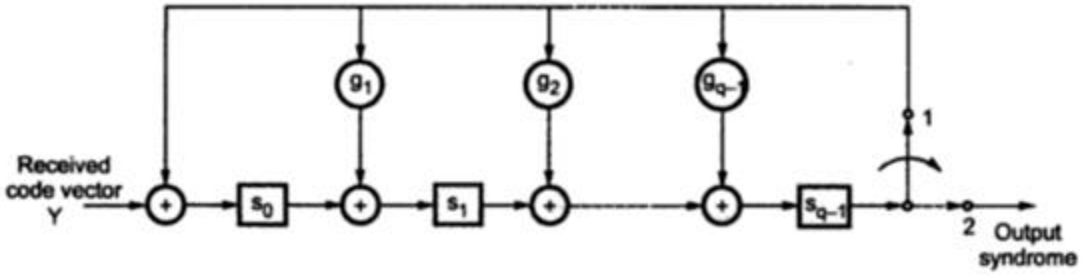
E depends upon the reminder R. For every reminder ‘R’ there will be specific error vector. So ‘R’ is a syndrome vector S or R(p) = S(p)

$$\frac{\mathrm{Y(p)}}{\mathrm{G(p)}} = \mathrm{Q(p)} + \frac{\mathrm{S(p)}}{\mathrm{G(p)}}$$

The syndrome vector is obtaind by

$$S(p) = rem\left[\frac{Y(p)}{G(p)}\right]$$

Decoder



Operation :

- Initially all the shift register contents are zero and the switch is closed in position 1.
- The received vector Y is shifted bit by bit into the shift register.
- Flipflops keep on changing the values according to input bits of Y and values of g_1, g_2 etc.
- After all the bits of Y are shifted, q flipflops of shift register contains the q bit Syndrome vector.
- The switch id then closed to position 2 and clocks are applied to the shift register.
- The output is a syndrome vector S = (S_{q-1}, S_{q-2}, ….S_1, S_0).

Some block codes that can be realized by cyclic codes

- (n,1) Repetition codes. High coding gain (minimum distance always n-1), but very low rate: 1/n
- (n,k) Hamming codes. Minimum distance always 3. Thus can detect 2 errors and correct one error. $n=2^m-1, k = n - m,$
- Maximum-length codes. For every integer there exists a maximum length code (n,k) with $n = 2^k - 1, d_{min} = 2^{k-1}$.
- BCH-codes. For every integer there exist a code with $n = 2^m-1$, and where t is the error correction capability
- (n,k) Reed-Solomon (RS) codes. Works with k symbols that consists of m bits that are encoded to yield code words of n symbols. For these codes and
- Nowadays BCH and RS are very popular due to large $d_{\min}$, large number of codes, and easy generation

Advantages :

- Simpler and easy to implement.
- Eliminate the storage needed for lookup table decoding. Therefore it is powerful and efficient.

- Encoder and decoder is simpler than non cyclic codes
- Well defined mathematical structure, hence very efficient decoder.

Disadvantages:

- Error correction is complicated since the combinational logic circuits in error detector are complex.

Convolutional Code:

Introduction:

Convolutional codes offer an approach to *error control* *coding* substantially different from that of block codes.

- *encodes the entire data stream*, into a single codeword.
- maps information to code bits sequentially by *convolving a sequence of information bits with "generator" sequences*.
- does not need to segment the data stream into blocks of fixed size (*Convolutional codes are often forced to block structure by periodic truncation*).
- Is *a machine with memory*.

  – This fundamental difference imparts a different nature to the design and evaluation of the code.
- Block codes are based on algebraic/combinatorial techniques.
- Convolutional codes are based on construction techniques.
- *Easy implementation using a linear finite-state shift register.*
- A Convolutional code is specified by three parameters (n, k, k)

  – *k* inputs and *n* outputs In practice, usually *k=1* is chosen.

  – $R_c = k / n$ is the coding rate, determining the number of data bits per coded bit.

  – *K* is ***the constraint length*** of the convolutinal code (where the encoder has *K-1* memory elements).

- The performance of a convolutional code depends on the coding rate and the constraint length

  – *Longer constraint length K*

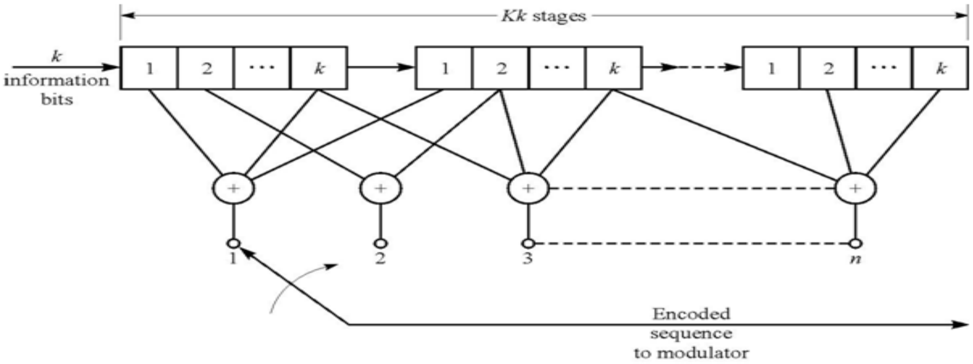
- More powerful code
- More coding gain

– Coding gain: the measure in *the difference between the signal to noise ratio (SNR) levels* between the uncoded system and coded system required to reach the same bit error rate (BER) level

- More complex decoder
- More decoding delay

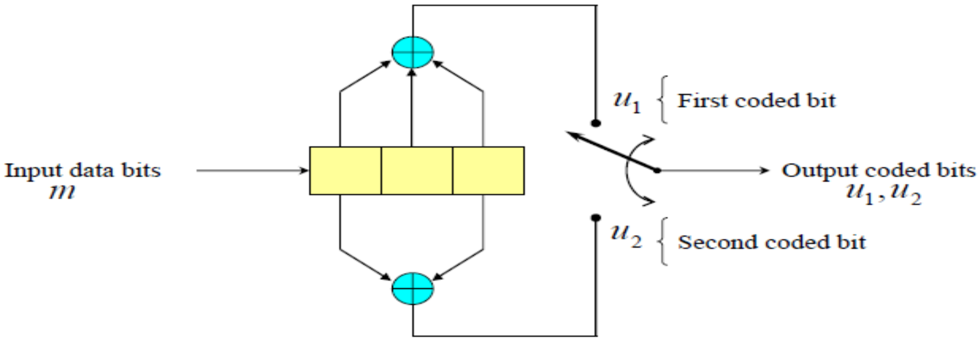
  – *Smaller coding rate R_c=k/n*

- More powerful code due to extra redundancy
- Less bandwidth efficiency

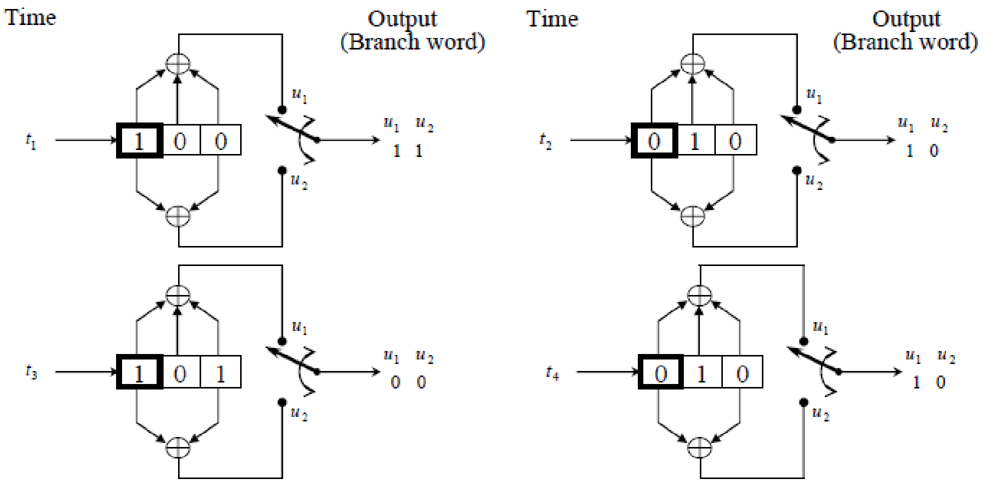


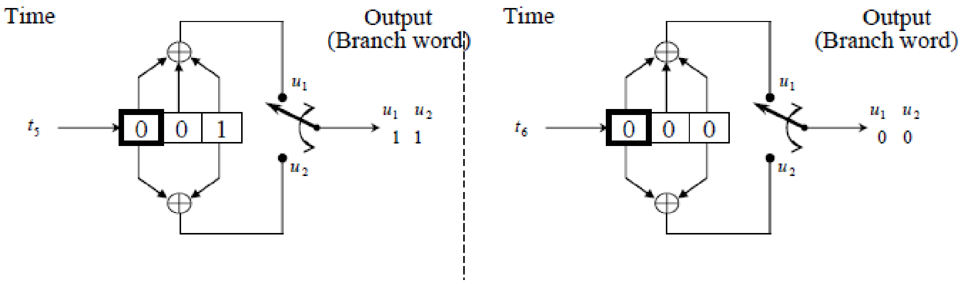
Convolutional encoder (rate ½, K=3)

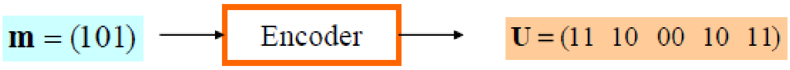
– 3 shift-registers, where the first one takes the incoming data bit and the rest form the memory of the encoder.



Message sequence: $\mathbf{m} = (101)$





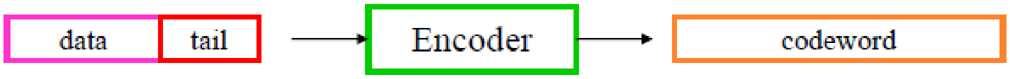

$\mathbf{m} = (101)$ → Encoder → $\mathbf{U} = (11\ \ 10\ \ 00\ \ 10\ \ 11)$

$$R_{eff} = \frac{3}{10} < R_c = \frac{1}{2}$$

Effective Code Rate

Initialize the memory before encoding the first bit (all-zero)

- Clear out the memory after encoding the last bit (all-zero)
- Hence, a tail of zero-bits is appended to data bits.



➤ *Effective code rate :*

- L is the number of data bits, L should be divisible by k

$$R_{eff} = \frac{L}{n\left[L/k + (K-1)\right]} < R_c$$

Example: **m**=[101]

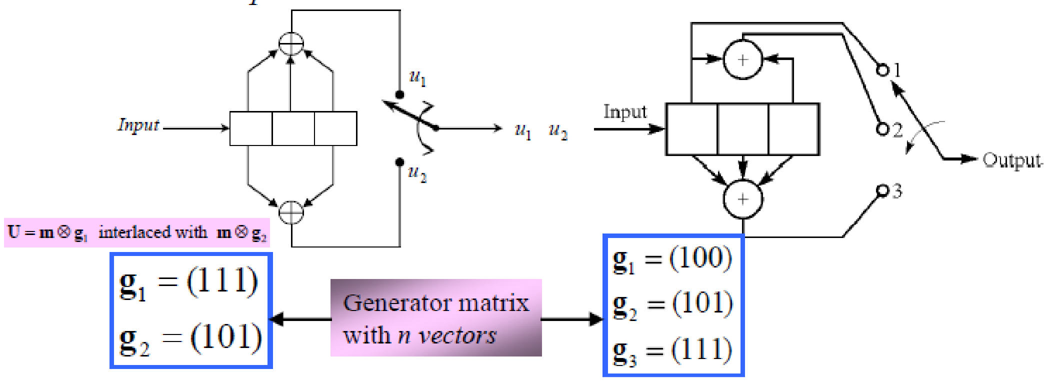
n=2, K=3, k=1, L=3

R_{eff}=3/[2(3+3-1)]=0.3

Vector representation:

– Define *n* vectors, each with *Kk* elements (one vector for each modulo-2 adder). The *i*-th element in each vector, is "1" if the *i*-th stage in the shift register is connected to the corresponding modulo-2 adder, and "0" otherwise.

– *Examples: k=1*



$\mathbf{U} = \mathbf{m} \otimes \mathbf{g}_1$ interlaced with $\mathbf{m} \otimes \mathbf{g}_2$

$\mathbf{g}_1 = (111)$

$\mathbf{g}_2 = (101)$

Generator matrix with *n vectors*

$\mathbf{g}_1 = (100)$

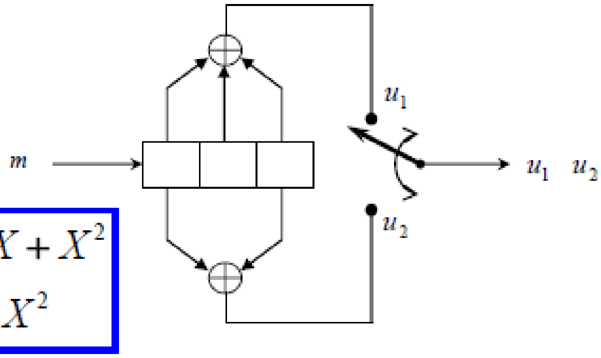
$\mathbf{g}_2 = (101)$

$\mathbf{g}_3 = (111)$

Polynomial representation :

– Define *n* generator polynomials, one for each modulo-2 adder. Each polynomial is of degree *Kk*-1 or less and describes the connection of the shift registers to the corresponding modulo- 2 adder.

– *Examples: k=1*



$$\mathbf{g}_1(X) = g_0^{(1)} + g_1^{(1)}X + g_2^{(1)}X^2 = 1 + X + X^2$$
$$\mathbf{g}_2(X) = g_0^{(2)} + g_1^{(2)}X + g_2^{(2)}X^2 = 1 + X^2$$

The output sequence is found as follows:

$$\begin{aligned}\mathbf{U}(X) &= \mathbf{m}(X)\mathbf{g}_1(X) \text{ interlaced with } \mathbf{m}(X)\mathbf{g}_2(X)\\ &= \mathbf{m}(X)\mathbf{g}_1(X) + X\mathbf{m}(X)\mathbf{g}_2(X)\end{aligned}$$

Polynomial representation : *Example: m=(1 0 1)*

$$\mathbf{m}(X)\mathbf{g}_1(X) = (1 + X^2)(1 + X + X^2) = 1 + X + X^3 + X^4$$
$$\mathbf{m}(X)\mathbf{g}_2(X) = (1 + X^2)(1 + X^2) = 1 + X^4$$

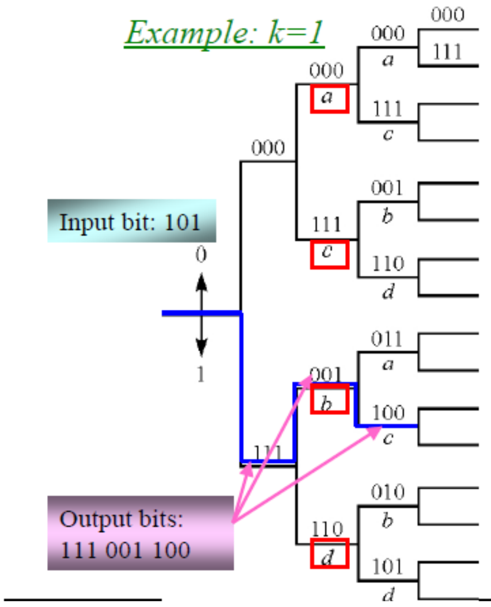
$$\mathbf{m}(X)\mathbf{g}_1(X) = 1 + X + 0.X^2 + X^3 + X^4$$
$$\mathbf{m}(X)\mathbf{g}_2(X) = 1 + 0.X + 0.X^2 + 0.X^3 + X^4$$

$$\mathbf{U}(X) = (1,1) + (1,0)X + (0,0)X^2 + (1,0)X^3 + (1,1)X^4$$
$$\mathbf{U} = 11 \quad 10 \quad 00 \quad 10 \quad 11$$

Tree Diagram (1)

- **One method to describe a convolutional code**

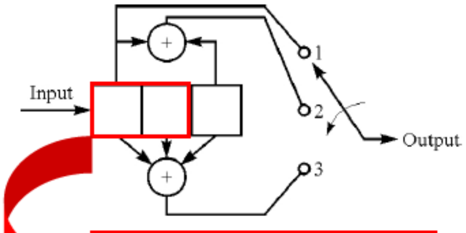
Example: k=1



Input bit: 101

Output bits:
111 001 100

K=3, k=1, n=3 convolutional encoder



The state of the first *(K-1)k* stages of the shift register:

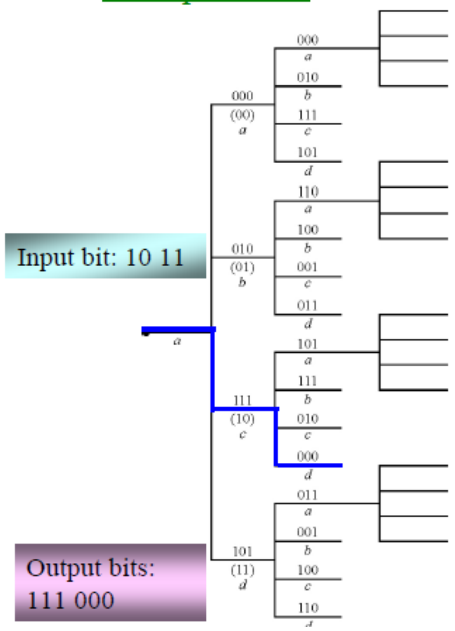
a=00; b=01;

c=10; d=11

The structure repeats itself after K stages(3 stages in this example).

Tree Diagram (2)

- *Example: k=2*



Input bit: 10 11

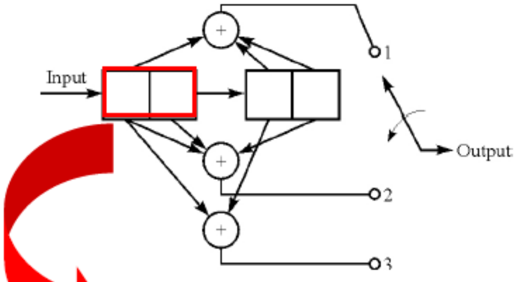
Output bits:
111 000

K=2, k=2, n=3 convolutional encoder

$$\mathbf{g}_1 = (1011)$$

$$\mathbf{g}_2 = (1101)$$

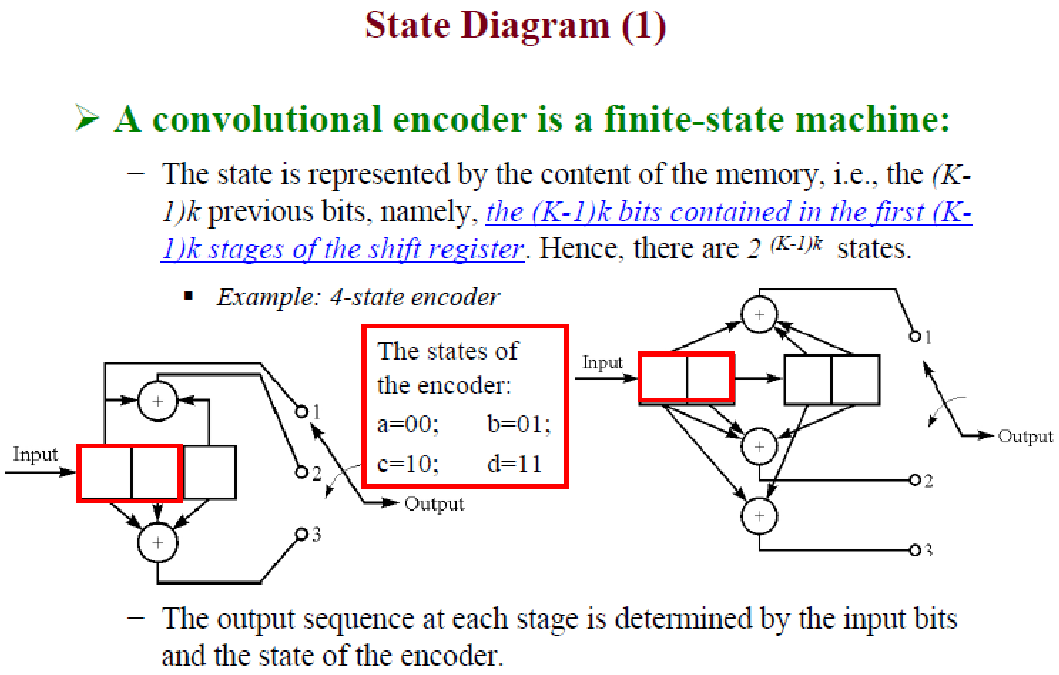
$$\mathbf{g}_3 = (1010)$$



The state of the first *(K-1)k* stages of the shift register:

a=00; b=01;

c=10; d=11



State Diagram (1)

- **A convolutional encoder is a finite-state machine:**
  - The state is represented by the content of the memory, i.e., the *(K-1)k* previous bits, namely, *the (K-1)k bits contained in the first (K-1)k stages of the shift register*. Hence, there are $2^{(K-1)k}$ states.
    - *Example: 4-state encoder*

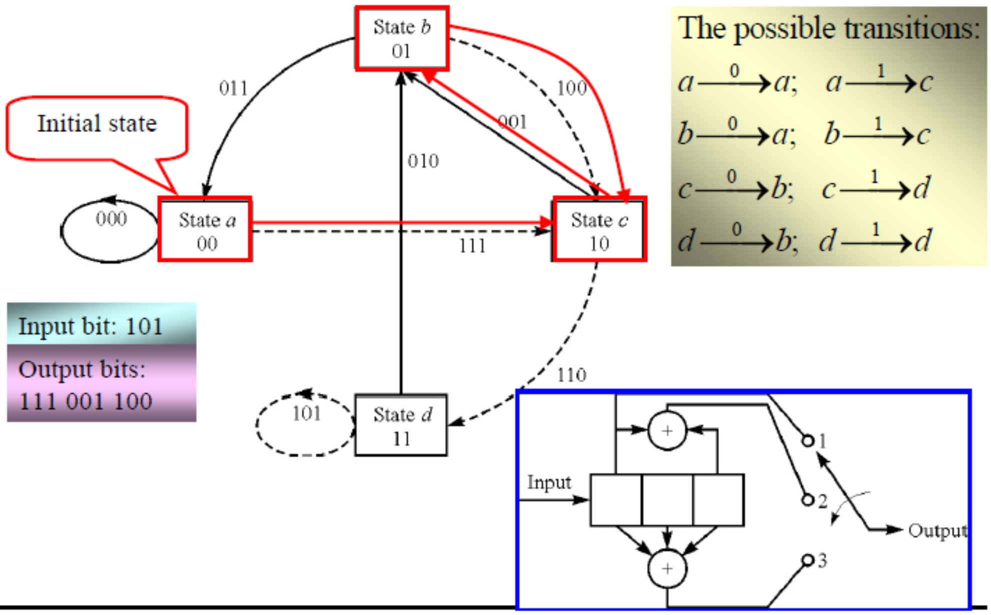
The states of the encoder:
a=00; b=01;
c=10; d=11

  - The output sequence at each stage is determined by the input bits and the state of the encoder.

State Diagram (2)

- A state diagram is simply a graph of the possible states of the encoder and the possible transitions from one state to another. It can be used to show *the relationship between the encoder state, input, and output*.
- The stage diagram has *2 (K-1)k* nodes, each node standing for one encoder state.
- Nodes are connected by branches
  - Every node has *2k* branches entering it and *2k* branches leaving it.
  - The branches are labeled with c, where c is the output.
  - When *k=1*
- The solid branch indicates that the input bit is 0.
- The dotted branch indicates that the input bit is 1.

Example of State Diagram (1)



The possible transitions:

$a \xrightarrow{0} a$; $a \xrightarrow{1} c$

$b \xrightarrow{0} a$; $b \xrightarrow{1} c$

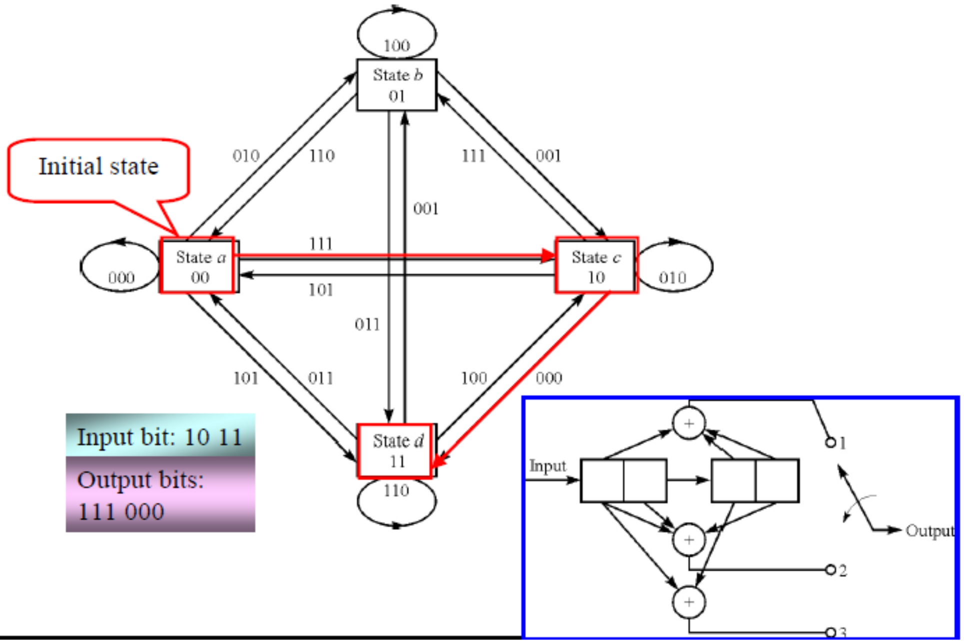
$c \xrightarrow{0} b$; $c \xrightarrow{1} d$

$d \xrightarrow{0} b$; $d \xrightarrow{1} d$

Input bit: 101

Output bits:
111 001 100

Example of State Diagram (2)



Input bit: 10 11

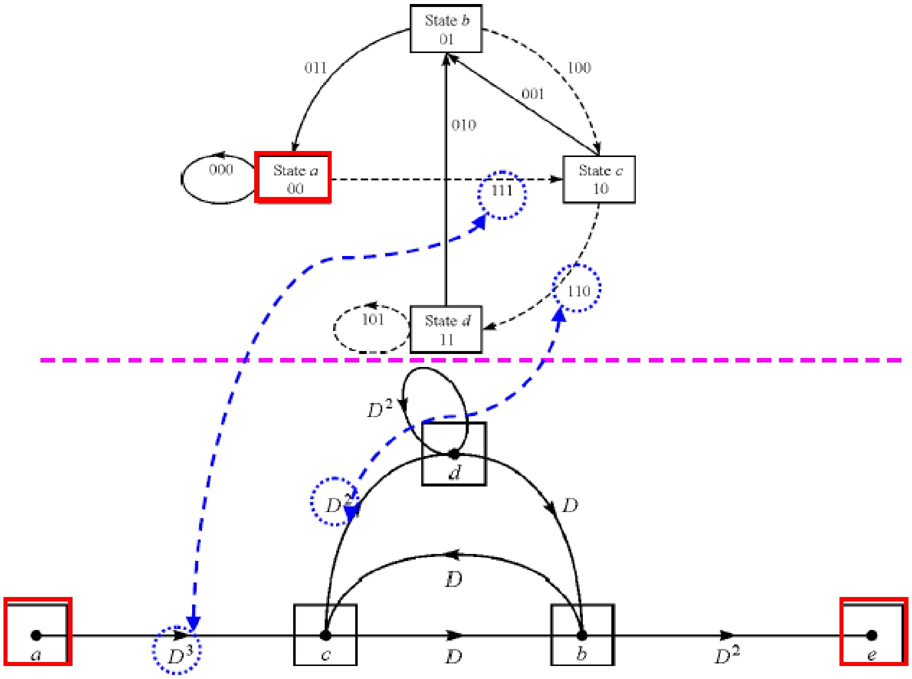
Output bits:
111 000

Distance Properties of Convolutional Codes (1)

- The state diagram can be modified to yield information on code distance properties.
- *How to modify the state diagram:*

  - *Split* the state *a* (all-zero state) into initial and final states, remove the self loop
  - *Label* each branch by the branch gain *Di*, where *i* denotes the Hamming weight of the *n* encoded bits on that branch
- Each path connecting the initial state and the final state represents a non-zero codeword that diverges from and re-emerges with state *a* (all-zero state) only once.

Example of Modifying the State Diagram



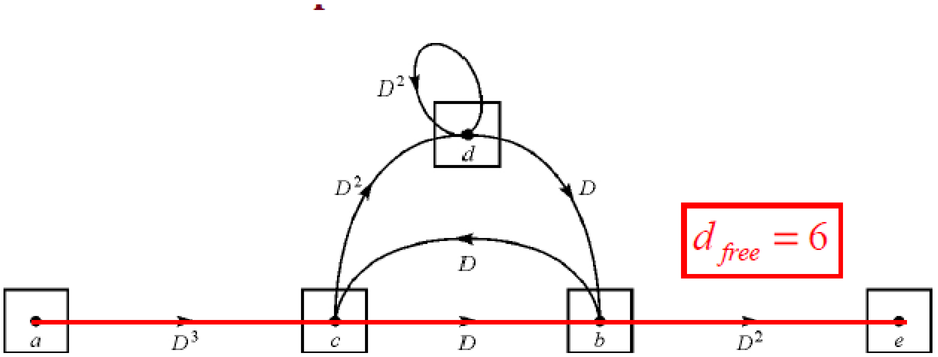
Distance Properties of Convolutional Codes (2)

- *Transfer function* (which represents the input-output equation in the modified state diagram) indicates the distance properties of the convolutional code by

$$T(X) = \sum_d a_d D^d$$

a_d represents the number of paths from the initial state to the final state having a distance *d*.

- The minimum free distance *dfree* denotes
  - The minimum weight of all the paths in the modified state diagram that diverge from and re-emerge with the all-zero state *a*.
  - The lowest power of the transfer function *T(X)*



$$
\begin{aligned}
X_c &= D^3 X_a + D X_b \\
X_b &= D X_c + D X_d \\
X_d &= D^2 X_c + D^2 X_d \\
X_e &= D^2 X_b
\end{aligned}
$$

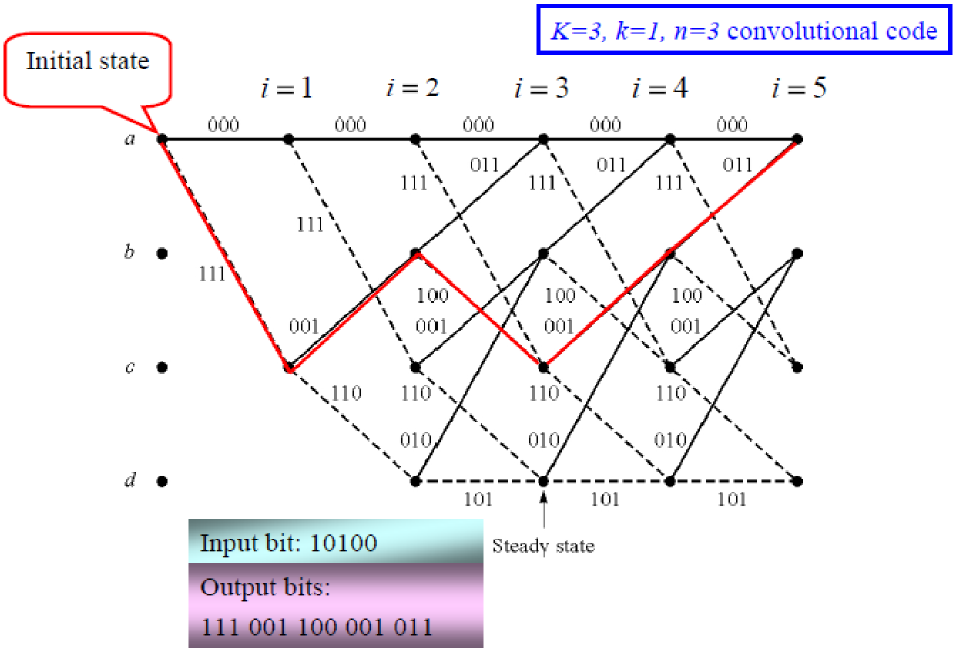
$$
\begin{aligned}
T(X) = X_e / X_a &= D^6 / (1 - 2D^2) \\
&= D^6 + 2D^8 + 4D^{10} + 8D^{12} + \cdots \\
&= \sum_{d=6}^{\infty} a_d D^d
\end{aligned}
$$

$$
a_d = \begin{cases} 2^{(d-6)/2} & (\text{even } d) \\ 0 & (\text{odd } d) \end{cases}
$$

Trellis Diagram

- Trellis diagram is an extension of state diagram which *explicitly shows the passage of time*.
  - All the possible states are shown for each instant of time.
  - Time is indicated by a movement to the right.
  - The input data bits and output code bits are represented by a
- unique path through the trellis.
  - The lines are labeled with c, where c is the output.
  - After the second stage, each node in the trellis has *2k*
- incoming paths and *2k* outgoing paths.
  - When *k=1*
- The solid line indicates that the input bit is 0.
- The dotted line indicates that the input bit is 1.

Example of Trellis Diagram (1)



Initial state

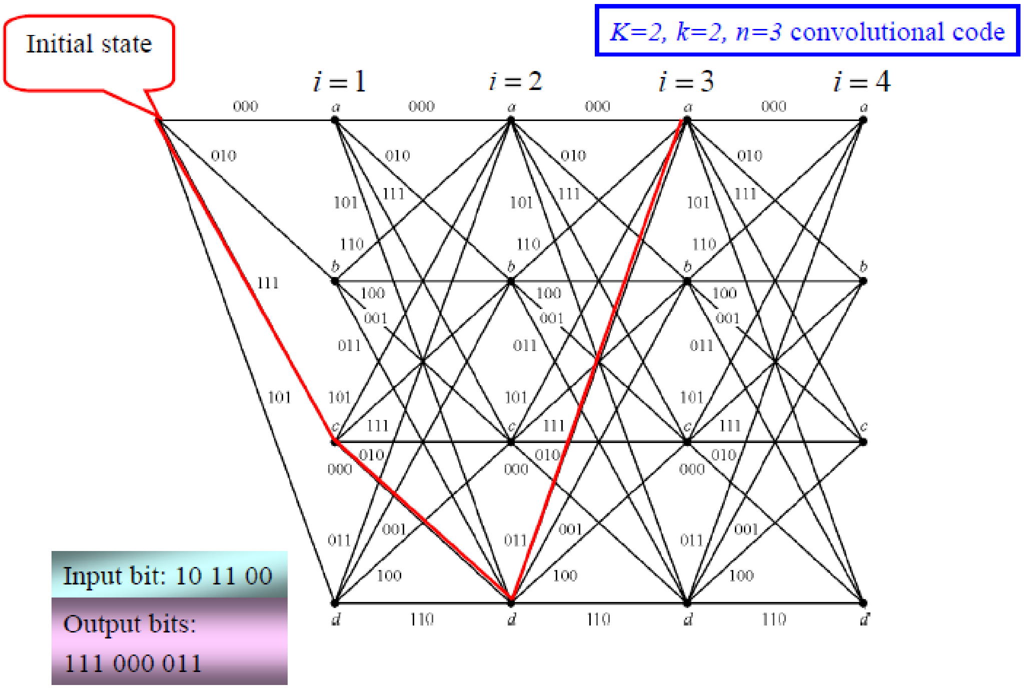
K=3, k=1, n=3 convolutional code

Input bit: 10100

Output bits:

111 001 100 001 011

Example of Trellis Diagram (2)



Initial state

K=2, k=2, n=3 convolutional code

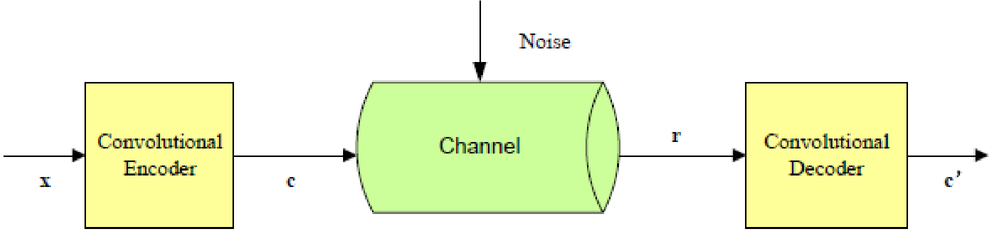
Input bit: 10 11 00

Output bits:

111 000 011

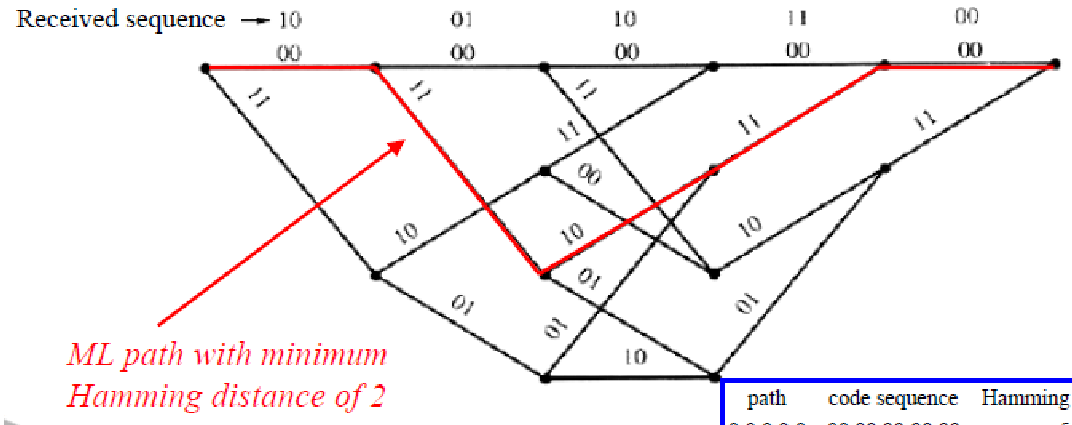
Maximum Likelihood Decoding

- Given the received code word **r**, determine the most likely path through the trellis. (maximizing *p*(**r**|**c'**))
  - Compare **r** with the code bits associated with each path
  - Pick the path whose code bits are "closest" to **r**
  - Measure distance using either *Hamming distance* for harddecision
- decoding or *Euclidean distance* for soft-decision
- decoding
  - Once the most likely path has been selected, the estimated
- data bits can be read from the trellis diagram



Example of Maximum Likelihood Decoding

hard-decision



All path metrics should be computed.

path	code sequence	Hamming distance
0 0 0 0 0	00 00 00 00 00	5
0 0 1 0 0	00 00 11 10 11	6
0 1 0 0 0	00 11 10 11 00	2
0 1 1 0 0	00 11 01 01 11	7
1 0 0 0 0	11 10 11 00 00	6
1 0 1 0 0	11 10 00 10 11	7
1 1 0 0 0	11 01 01 11 00	3
1 1 1 0 0	11 01 10 01 11	4

The Viterbi Algorithm

- A breakthrough in communications in the late 60’s
  - Guaranteed to find the ML solution
- However the complexity is only O(*2K*)
- Complexity does not depend on the number of original data bits
  - Is easily implemented in hardware
- Used in satellites, cell phones, modems, etc
- Example: Qualcomm Q1900

- Takes advantage of the structure of the trellis:
  - Goes through the trellis one stage at a time
  - At each stage, finds the most likely path leading into each
- state (*surviving path*) and discards all other paths leading into the state (*non-surviving paths*)
  - Continues until the end of trellis is reached
  - At the end of the trellis, traces the most probable path from
- right to left and reads the data bits from the trellis
  - Note that in principle whole transmitted sequence must be
- received before decision. However, *in practice storing of stages with length of 5K is quite adequate*

Implementation:

1. Initialization:

   − Let *Mt(i)* be the path metric at the *i*-th node, the *t*-th stage in trellis

   − Large metrics corresponding to likely paths; small metrics corresponding to unlikely paths

   − Initialize the trellis, set *t=0* and *M0(0)=0;*

2. At stage (*t+1)*,

   − *Branch metric calculation*

   • Compute the metric for each branch connecting the states at time *t* to states at time *(t+1)*

   • The metric is related to the likelihood probability between the received bits and the code bits corresponding to that branch:

$p(\mathbf{r}(t+1)|\mathbf{c}'(t+1))$

− *Branch metric calculation*

• In hard decision, the metric could be the number of same bits between the received bits and the code bits.

− *Path metric calculation*

• For each branch connecting the states at time *t* to states at time *(t+1)*, add the branch metric to the corresponding partial path metric *Mt(i).*

− *Trellis update*

• At each state, pick the most likely path which has the largest metric and delete the other paths

• Set *M(t+1)(i)*= the largest metric corresponding to the state *i*

3. Set *t=t+1*; go to step 2 until the end of trellis is reached
4. Trace back

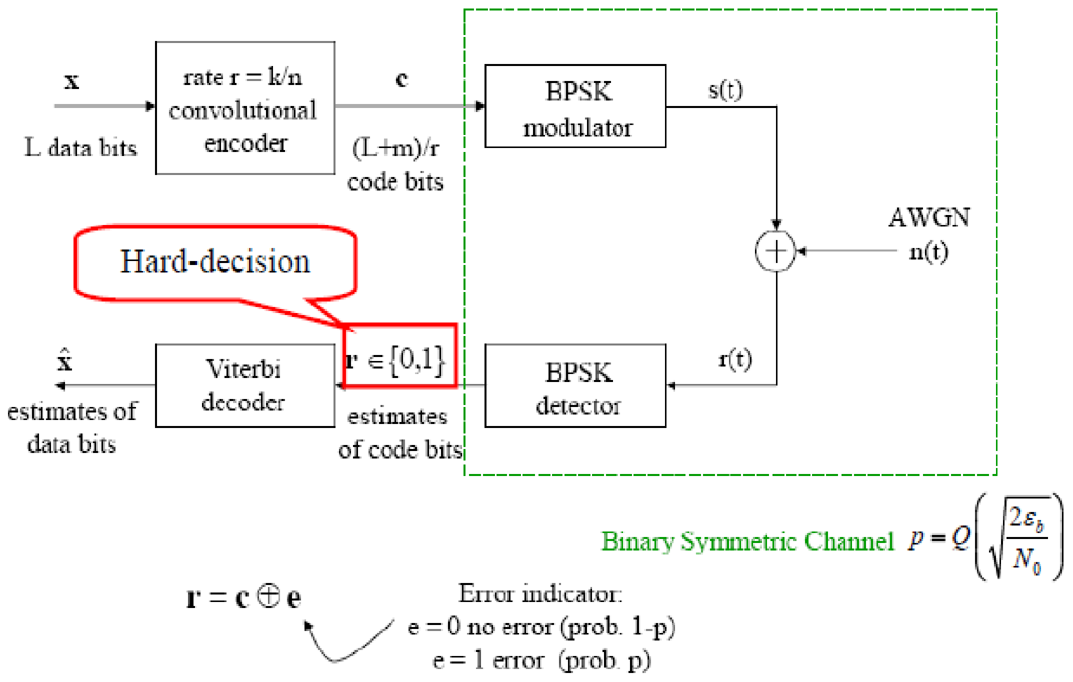
− Assume that the encoder ended in the all-zero state

− The most probable path leading into the last all-zero state in the trellis has the largest metric

• Trace the path from right to left

• Read the data bits from the trellis

Examples of Hard-Decision Viterbi Decoding (1)



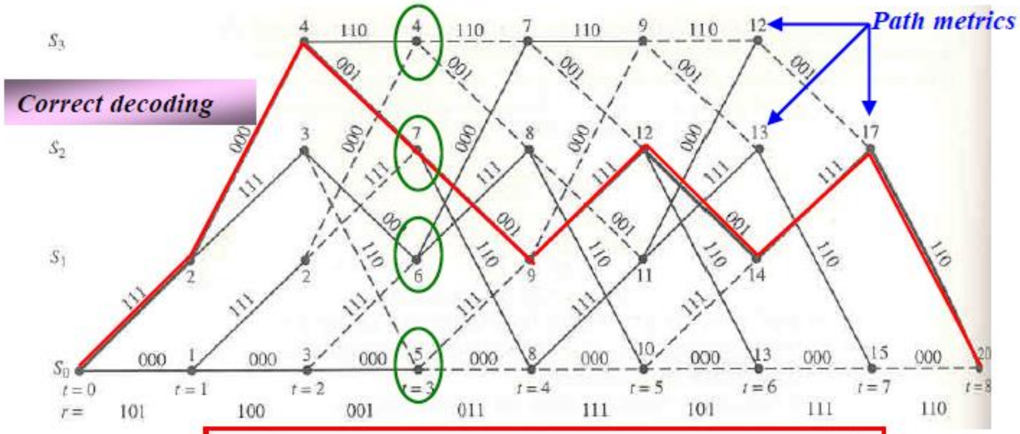
$p = Q\left(\sqrt{\frac{2\varepsilon_b}{N_0}}\right)$

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

Error indicator:
e = 0 no error (prob. 1-p)
e = 1 error (prob. p)

Examples of the Hard-Decision Viterbi Decoding (2)

Non-surviving paths are denoted by dashes lines.



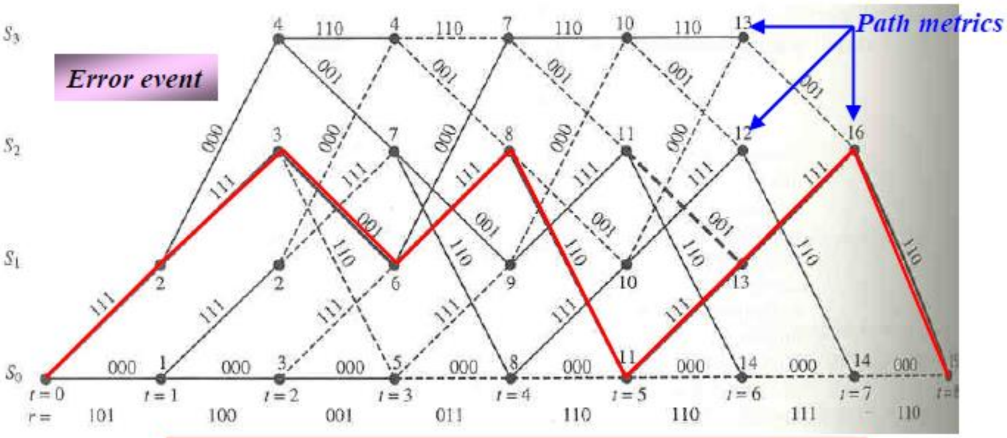
$$\mathbf{c}=(111\ \ 000\ \ 001\ \ 001\ \ 111\ \ 001\ \ 111\ \ 110)$$

$$\mathbf{r}=(1\bar{0}1\ \ \bar{1}00\ \ 001\ \ 0\bar{1}1\ \ 111\ \ \bar{1}01\ \ 111\ \ 110)$$

$$\mathbf{c}'=(111\ \ 000\ \ 001\ \ 001\ \ 111\ \ 001\ \ 111\ \ 110)$$

Examples of the Hard-Decision Viterbi Decoding (3)

Non-surviving paths are denoted by dashes lines.



$$\mathbf{c}=(111\ \ 000\ \ 001\ \ 001\ \ 111\ \ 001\ \ 111\ \ 110)$$

$$\mathbf{r}=(1\bar{0}1\ \ \bar{1}00\ \ 001\ \ 0\bar{1}1\ \ 11\bar{0}\ \ \bar{1}\bar{1}0\ \ 111\ \ 110)$$

$$\mathbf{c}'=(111\ \ \bar{1}\bar{1}\bar{1}\ \ 001\ \ \bar{1}\bar{1}1\ \ 11\bar{0}\ \ \bar{1}\bar{1}1\ \ 111\ \ 110)$$

Error Rate of Convolutional Codes (1)

- An error event happens when an erroneous path is selected at the decoder
- *Error-event probability*:

$$P_e \leq \sum_{d=d_{free}}^{\infty} a_d \, P_2(d)$$

$$a_d = The\ number\ of\ paths\ with\ the\ hamming\ distance\ d$$
$$P_2(d) = Probality\ of\ the\ path\ with\ the\ hamming\ code\ and\ it$$
$$depends\ on\ modulation\ scheme, hard\ or\ soft\ decision$$

BER is obtained by multiplying the error-event probability by the number of data bit errors associated with each error event.

BER is upper bounded by

$$P_b \leq \sum_{d=d_{free}}^{\infty} f(d) a_d \, P_2(d)$$

f(d) the number of data bit errors corresponding to the erroneous path with the Hamming distance of d

Turbo Codes:

- Turbo codes were proposed by Berrou and Glavieux in the 1993 International Conference in Communications.
- Performance within 0.5 dB of the channel capacity limit for BPSK was demonstrated.
- Features of turbo codes
- Parallel concatenated coding
- Recursive convolutional encoders
- Pseudo-random interleaving
- Iterative decoding

Pseudo-random Interleaving:

- The coding dilemma:
- Shannon showed that large block-length random codes achieve channel capacity.
- However, codes must have structure that permits decoding with reasonable complexity.
- Codes with structure don’t perform as well as random codes.
- “Almost all codes are good, except those that we can think of.”

Solution:

- Make the code appear random, while maintaining enough structure to permit decoding.
- This is the purpose of the pseudo-random interleaver.
- Turbo codes possess random-like properties.
- However, since the interleaving pattern is known, decoding is possible.

Why Interleaving and Recursive Encoding?

In a coded systems:

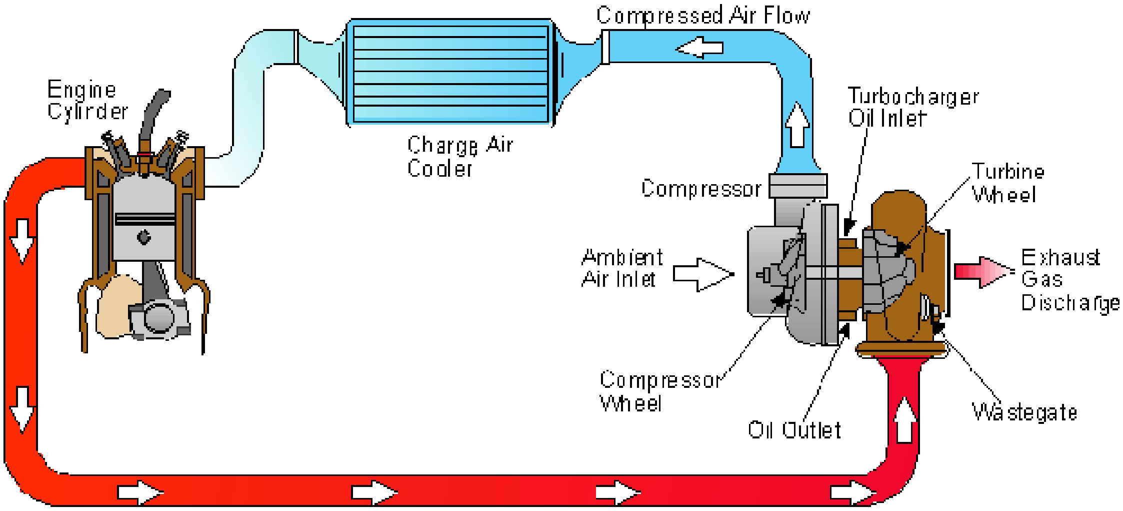
- Performance is dominated by low weight code words.
- A “good” code: will produce low weight outputs with very low probability.
- An RSC code: Produces low weight outputs with fairly low probability.
- However, some inputs still cause low weight outputs.
- Because of the interleaver: The probability that both encoders have inputs that cause low weight outputs is very low.
- Therefore the parallel concatenation of both encoders will produce a “good” code.

Iterative Decoding:

- There is one decoder for each elementary encoder.
- Each decoder estimates the *a posteriori probability* (APP) of each data bit.
- The APP’s are used as *a priori* information by the other decoder.
- Decoding continues for a set number of iterations.
- Performance generally improves from iteration to iteration, but follows a law of diminishing returns

The Turbo-Principle:

Turbo codes get their name because the decoder uses feedback, like a turbo engine



Turbo Code Summary:

- Turbo code advantages:
- Remarkable power efficiency in AWGN and flat-fading channels for moderately low BER.
- Deign tradeoffs suitable for delivery of multimedia services.

Turbo code disadvantages:

- Long latency.
- Poor performance at very low BER.
- Because turbo codes operate at very low SNR, channel estimation and tracking is a critical issue.
- The principle of iterative or “turbo” processing can be applied to other problems.
- Turbo-multiuser detection can improve performance of coded multiple-access systems.

QUESTIONS FOR PRACTICE

PART A (2 marks)

1. What is hamming distance?
2. Define code efficiency.
3. What is meant by systematic and non-systematic codes?
4. What is meant by linear code?
5. What are the error detection and correction capabilities of hamming co d es ?
6. What is meant by cyclic codes?
7. How syndrome is calculated in Hamming codes and cyclic codes?
8. What is BCH code?
9. What is RS code?
10. What is difference between block codes and convolutional codes?
11. Define constraint length in convolutional code?
12. Define free distance and coding gain.
13. What is convolution code?
14. What is meant by syndrome of linear block code?
15. What are the advantages and disadvantages of convolutional codes?
16. Define sates of encoder?
17. Compare between code tree and trellis diagram?
18. Write the features of BCH Codes?
19. Define constraint length in convolutional codes?
20. Define constraint length in convolutional codes?
21. What is Golay codes?

PAT B (12 Marks)

1. Draw the code tree of a Convolutional code of code rate r=1/2 and Constraint length of K=3 starting from the state table and state diagram for an encoder which is commonly used.
   a. Draw the state Diagram.
   b. Draw the state Table.
   c. Draw the code Tree
2. Draw the trellis diagram of a Convolutional code of code rate r=1/2 and Constraint length of K=3 starting from the state table and state diagram for an encoder which is commonly used.
   a. Draw the state Diagram.
   b. Draw the state Table.
   c. Draw the trellis diagram
3. Decode the given sequence 11 01 01 10 01 of a convolutional code with a code rate of r=1/2 and constraint length K=3, using viterbi decoding algorithm.
   a. Draw the state Diagram.
   b. Draw the state Table.
   c. Draw the code Tree
   d. Decode the given sequence using trellis diagram.
4. Explain the construction of Block Code and explain how error syndrome is calculated
   a. Representation of Block Code.
   b. Generator Matrix.
   c. Generation of Codewords.
   d. Generation of Parity Check Matrix.
   e. Calculation OF Error Syndrome.
5. Consider a (6,3) linear block code defined by the generator matrix

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

   a. Determine if the code is a hamming code. Find the parity check matrix in systematic code
   b. Find the encoding tale for the linear block code.
   c. What is the minimum hamming distance and how many errors can detect and correct.
   d. Find the decoding tale for the linear block code.
   e. Draw the hardware encoder diagram
   f. Draw the hardware syndrome generator diagram.

5. Prove that $GH^T = HG^T = 0$ for a systematic linear block code.
6. The parity check bits of a (8,4) block code is given by

$C_1 = m_1 + m_2 + m_4$, $C_2 = m_1 + m_2 + m_3$

$C_2 = m_1 + m_3 + m_4$ *and* $C_4 = m_2 + m_3 + m_4$

Here m1, m2, m3, m4 are message bits.

a. Find the generator matrix and parity check matrix for this code.
b. Find minimum weight of this code.
c. Find error detecting capabilities of this code.

7. Design the encoder for the (7,4) cyclic code generated by $G(p) = p^3 + p + 1$ and verify its operation for any vector.
8. Sketch the encoder and syndrome calculator for the generator polynomial $g(x) = 1 + x^2 + x^3$ and obtain the syndrome for the received code word 1001011.

\