ITC = U-2 Bhavana Yadav

Error Control Coding

→ Transmission of the data over the channel depends on a Parameters.

i) Transmitted Power

ii) Channel Bandwidth.

→ Power Spectral Density of channel noise and these two Parameters determine Signal to noise Power Ratio.

→ S/N Ratio Determines the Probability of error of the modulation Scheme.

→ For any given S/N Ratio Probability of error Can be reduced further by using coding techniques.

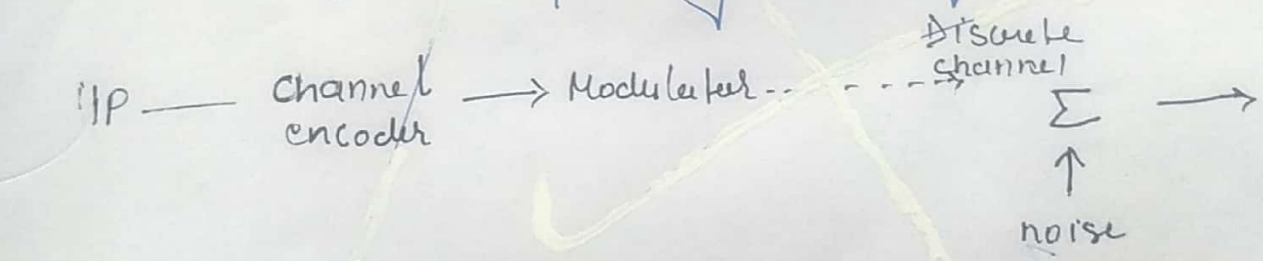


Fig:- Digital communn. system with channel encoding

→ The channel encoder adds extra bits (redundancy) to the message bits.

→ The encoded signal is then transmitted over the noisy channel.

→ The channel decoder identifies the redundancy bits & uses them to detect and correct the error in the message bit if any.

Example of error control coding:-

→ Let us consider a error control coding scheme, which transmits 000 to → Symbol '0'
111 to → Symbol '1'

→ there are two redundant bits in every message being transmitted.

→ Decoder checks the received triplets and takes the decision in favour of majority of bits

→ Ex If triplet is → 110 → 2 1's → So → '1' will tx.
If triplet is → 100 or 010 → 2 0's → So → '0' will tx.

→ Message symbol is received correctly if no more than one bit in each triplet is in error.

→ If the message would have been transmitted without coding, then it is difficult to recover the original transmitted symbols.

→ Thus the redundancy in the transmitted message reduces probability of error at the receiver

→ **Note**

Error control coding has following aspects.

1. The redundancy bits in the message are called check bits, Error can be detected and corrected with the help of these bits.
2. It is not possible to detect & correct all the errors in the message. Error up to certain limit can be detected & corrected.
3. The check bits reduces the the data rate through the channel.

→ So the no. of errors introduced due to channel noise are minimized by encoder & decoder.

→ Due to redundant bits the system becomes slightly complex because of coding techniques.

Types of Codes

Mainly classified into 2

1. Blocks codes

→ These blocks consists of 'n' no. of bits in one blocks or codeword

→ codeward consists of 'K' message bits & (n-K) redundant bits.

→ Such blocks codes are called (n, K) blocks codes.

2. Convolutional Codes:-

→ Coding operation is discrete time convolution of input sequence with the impulse response of encoder.

→ The convolution encoder accepts the message bit continuously and generates the encoded sequence continuously.

→ Codes can also be classified as.

i) linear

ii) Non-linear.

i) linear codes

If the two code words are added by Modulo-2 arithmatic, then it Produces third code word (other codeward can be obtained by addition of existing codewords).

ii) Non-linear Codes:-

→ Addition of non-linear codewords does not necessarily Produce third codeword.

Methods of Controlling Errors

→ There are mainly two main methods used for error control coding.

i) Forward acting error correction

ii) Error detection with transmission

1. Forward Acting error Correction :-

→ In this method, the errors are detected and corrected by proper coding techniques at the Receiver. (decoder)

→ check bits are used by the Receiver to detect and correct errors.

→ Error detection & correction capability of the Rx. depends upon the no. of redundant bits in transmission message.

→ Forward acting error correction is faster, but overall Probability of errors is higher.

→ This is because some of the errors may not be corrected.

2. Error detection with transmission

→ In this, the decoder checks the input sequence.

→ when it detects any error, it discard that part of the sequence & requests the transmitter for retransmission.

→ The transmitter then again transmit the part of sequence in which error was detected.

→ Detector does not correct that error.

→ It just detect the error and sends requests to transmitter

→ This method has lower Probability of errors, but it is slow.

Types of errors

There are basically two types of errors introduced during transmission on the data.

1. **Random error**
   - → These errors are created due to white gaussian noise in the channel
   - → These errors are uncorrelated.

2. **Burst error** :-
   - → Generated due to impulsive noise in the channel
   - → These impulsive noise are generated due to lightning & switching transients.
   - → These noise (burst) effect several successive symbols.
   - → Such errors are called burst errors.
   - → Burst error are dependent on each other in successive message intervals.

Some terms used in Error Control Coding

1. **Codeword** :- * Encoder block of n bits is called Codeword
   * It contain message bit & redant bits
2. **Blocks length** * The no. of bits after coding is called blocks length of the code
3. **Code Rate** * The ratio of message bit(k) and encoder output bit(n) is called code Rate.

Code rate = $r = \frac{k}{n}$.

$0 < r < 1$

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,

$$\text{Channel Data Rate} = R_0 = \frac{n}{k} R_s$$

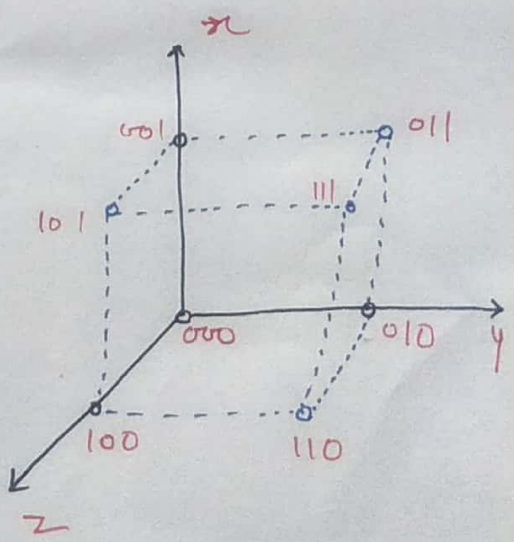
Code Vector :-

→ 'n' bit code word can be visualized in an n-dimensional space as a vector whose elements or co-ordinates are the bits in the code word.

→ For a 3 bit code vector ⇒ 8 distinct code words.
(no. of code words = 2^k)

→ Suppose if b_0 = on x-axis
b_1 = on Y-axis
b_2 = on z-axis

S.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



Hamming Distance

→ Hamming Distance between two code vectors is the Min. distance in which they are differ.

→ Let $X = 101$

$Y = 110$

X & Y are differ in 2nd and 3rd bits.

→ So hamming distance between X & Y is = 2.

$d(X, Y) = d = 2$.

Note Max. distance b/w X & Y in fig 1.

$X = 100$

$Y = 011$

Max. $d = 3$

Minimum distance

→ It is the smallest hamming distance between the valid code vectors

→ Some of the requirements of error control capability of code :-

S. No	error detected/~~upto~~ Corrected	Distance Required
1.	detect upto 's' error per word	$d_{min} \geq s+1$
2.	Correct upto 't' error per word	$d_{min} \geq 2t+1$
3	Correct upto 't' error & detect $s \geq t$ errors per word.	$d_{min} \geq t+s+1$

For (n, k) blocks code, the minimum distance is given

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

Code efficiency

→ Code efficiency is the ratio of message bit in a blocks to the transmitted bits for that blocks by the encoder. i.e.

→ for (n, k) blocks code, there are 'k' message bits & 'n' transmitted bit.

$$\text{Code efficiency} = \frac{k}{n}$$

→ Code rate is $= \frac{k}{n}$

So $$\text{Code efficiency} = \text{Code rate} = \frac{k}{n}$$

weight of code

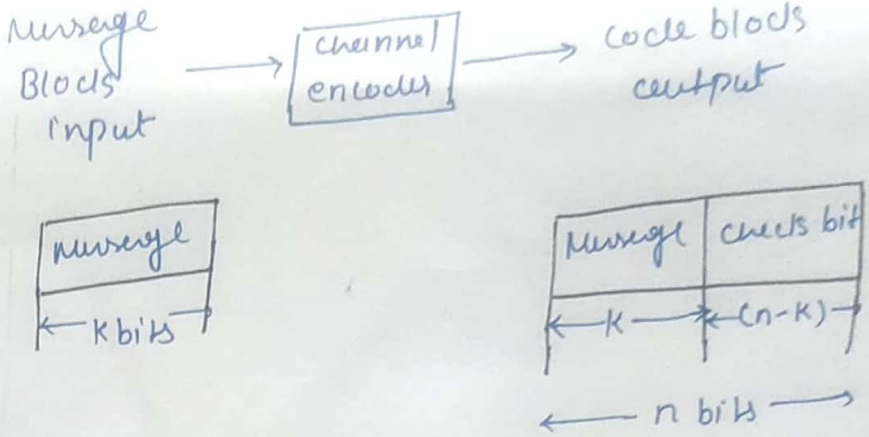
→ No. of non zero elements in the transmitted code vector is called vector weight.

→ And denoted by $w(x)$, where x = code vector.

Ex $x = 0111010$

$w(x) = 5$

Linear Blocks Codes



→ For a blocks of k message bits, (n-K) parity bits or checks bits are added.

→ So total bits at the o/p of channel encoder = 'n'

→ Such codes are called (n, K) blocks codes.

Symmetric Codes

→ In this message bits appears in the beginning of the code word

→ Message bit appears first, and then checks bits are transmitted in a blocks.

→ This type of code is called Symmetric code

Note In non-Symmetric code it is difficult to/not Possible, to find Message bit & checks bits. They are mixed in the blocks.

Linear Codes

→ A code is linear, when Sum of any two code vectors produces another code vectors.

→ Consider a code vector consists of $m_1, m_2, \ldots m_K$ message bits and $c_1, c_2 \ldots c_q$ checks bits.

Then this code vector can be written as

$$X = (m_1, m_2 \ldots\ldots m_{15}, c_1, c_2 \ldots\ldots c_q) \quad \text{---(1)}$$

$$q = n - 15 \quad \text{---(2)}$$

→ Where q = no. of redundant bits added by encoder.

→ Code vector can also be written as -

$$X = (M|C) \quad \text{---(3)}$$

M = k- bit message vector

C = q- bit check vector.

Note: check bit plays a role of error detection & correction.

→ Linear Block code, generates that 'check bits'. then code vector can be written as

$$\boxed{X = MG} \quad \text{---(4)}$$

X = code vector $[1 \times n]$, n bits

M = Message vector $[1 \times K]$, 15 bits

G = Generator Matrix $[K \times]$,

$$\boxed{[X]_{1\times n} = [M]_{1\times k} [G]_{k\times n}} \quad \text{---(5)}$$

→ Generator Matrix depends upon - LBC used & Represented by

$$\boxed{G = [I_k \mid P_{k\times q}]_{k\times n}} \quad \text{---(6)}$$

$I = K \times K \Rightarrow$ Identity Matrix

$P = K \times q$ Submatrix

→ The checks vector can be obtained as

$$C = MP$$

$$[C_1, C_2 \cdots C_q]_{1\times q} = [M_1, M_2 \cdots M_K]_{1\times K} \begin{bmatrix} P_{11} & P_{12} & \cdots & P_{1q} \\ P_{21} & P_{22} & \cdots & P_{2q} \\ \vdots & \vdots & & \vdots \\ P_{K1} & P_{K2} & \cdots & P_{Kq} \end{bmatrix}_{K\times q}$$

Summary

→ code vector $X = (m_1, m_2 \cdots m_K \; c_1, c_2 \cdots c_q)$

q = Redundant bit

→ (or)

$X = (M \mid C)$

M = k-bit message vector

C = q-bit checks vector

↳ for error detection / correction

→

$X = MG$

$G = [I_K \mid P_{K\times q}]_{K\times n}$

$I_K = K \times K$

$P_{K\times q}$ = Submatrix.

$$C_1 = m_1 P_{11} \oplus m_2 P_{21} \oplus m_3 P_{31} \oplus \cdots \oplus m_K P_{K1}$$

$$C_2 = m_1 P_{12} \oplus m_2 P_{22} \oplus m_3 P_{32} \oplus \cdots \oplus m_K P_{K2}$$

$$C_3 = m_1 P_{13} \oplus m_2 P_{23} \oplus m_3 P_{33} \oplus \cdots \oplus m_K P_{K3}$$

Note — Here addition is Modulo-2

Q Generator Matrix for (6,3) block code is given. Find all code vectors of this code.

$$G = \left[\begin{array}{ccc:ccc} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{array}\right] \quad \text{——— (1)}$$

Sol^{n} $C = MP$

Step-1 To obtain 'P' sub Matrix.

$$G = [I_K : P_{K\times q}] \quad \text{——— (2)}$$

→ Comparing this eq(2) with given G (eq^{n} 1)

$$I_K = I_{3\times 3} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$P_{K\times q} = P_{3\times 3} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

Step-2 To obtain check bits

$$[G] = \underset{\downarrow\ \downarrow}{K\times n}$$
$$3\ \ 6$$

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

→ Now blocks size of message vector = 3, So total Combinations will be = 8

S.No.	Bits of message vector in one blocks		
	m_1	m_2	m_3
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

$[C] = MP$.

$$[C_1, C_2, C_3]_{1\times 3} = [m_1, m_2, m_3] \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

$C_1 = (0 \times m_1) \oplus (m_2) \oplus m_3$

$C_2 = m_1 \oplus (0 \times m_2) \oplus m_3$

$C_3 = m_1 \oplus m_2 \oplus (0 \times m_3)$

→ By above three eq^{n}s we obtain

$C_1 = m_2 \oplus m_3$ ——— (3)

$C_2 = m_1 \oplus m_3$ ——— (4)

$C_3 = m_1 \oplus m_2$ ——— (5)

Step-3 To determine checks bits & code vector for every message Vector:-

→ first blocks $(m_1\ m_2\ m_3) = 000$

$C_1 = 0 \oplus 0 = 0$

$C_2 = 0 \oplus 0 = 0$

$C_3 = 0 \oplus 0 = 0$

i.e. $(C_1, C_2, C_3) = 000$

→ Second blocks $(m_1\ m_2\ m_3) = 001$ ($m_1 m_2 m_3$)

$C_1 = m_2 \oplus m_3 = 0 \oplus 1 = 1$

$C_2 = m_1 \oplus m_3 = 0 \oplus 1 = 1$

$C_3 = m_1 \oplus m_2 = 0 \oplus 0 = 0$

$(C_1, C_2, C_3) = 110$

Similarly for $m_1\ m_2\ m_3 =$ 010
011
100
101
110
111

S.No.	Bits of Message vector in 1 Blocks.			Checks bits		Complete Code vector.		
	m_1	m_2	m_3	$c_1 = m_2 \oplus m_3$		$c_2 = m_1 \oplus m_3$	$c_3 = m_1 \oplus m_2$	
1	0	0	0	0		0	0	
2	0	0	1	1		1	0	
3	0	1	0	1		0	1	
4	0	1	1	0		1	1	
5	1	0	0	0		1	1	
6	1	0	1	1		0	1	
7	1	1	0	1		1	0	
8	1	1	1	0		0	0	

Parity check Matrix

→ For every blocks code there is a $q \times n$ Parity check matrix (H). And defined as

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

→ P^T = transpose of P - Sub-Matrix.

$$P = \begin{bmatrix} P_{11} & P_{12} & P_{13} & \cdots & P_{1q} \\ P_{21} & P_{22} & P_{23} & \cdots & P_{2q} \\ \vdots & & & & \\ P_{K1} & P_{K2} & P_{K3} & \cdots & P_{Kq} \end{bmatrix}_{K \times q}$$

The transpose of this matrix can

$$P^T = \begin{bmatrix} P_{11} & P_{21} & P_{31} & \cdots & P_{K1} \\ P_{12} & P_{22} & P_{32} & \cdots & P_{K2} \\ \vdots & \vdots & \vdots & & \\ P_{1q} & P_{2q} & P_{3q} & \cdots & P_{Kq} \end{bmatrix}_{q \times K}$$

→ So $H = [P^T : I]_{q \times n}$ can be written as

$$H_{q \times n} = \left[\begin{array}{cccc:cccc} P_{11} & P_{21} & P_{31} & \cdots P_{K1} & 1 & 0 & 0 & \cdots 0 \\ P_{12} & P_{22} & P_{32} & \cdots P_{K2} & 0 & 1 & 0 & \cdots 0 \\ P_{13} & P_{23} & P_{33} & \cdots P_{K3} & 0 & 0 & 1 & \cdots 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\ P_{1q} & P_{2q} & P_{3q} & \cdots P_{Kq} & 0 & 0 & 0 & \cdots 1 \end{array}\right]_{q \times n}$$

→ <u>Note</u> we know

$$G = [I_K : P_{K \times q}]_{K \times n}.$$

$$H = [P^T : I_K]$$

So if Generator Matrix is given then Parity checks Matrix 'H' can be obtained & vice-versa

Hamming Code

→ These are (n, k) Linear blocks codes.

→ Those satisfy the following conditions:-

i) No. of checks bits $q \geq 3$

ii) Blocks length $n = 2^q - 1$

iii) No. of message bit $k = n - q$

iv) Minimum distance $d_{min} = 3$.

→ We know $\boxed{r = \frac{k}{n}}$ *

$$r = \frac{n-q}{n}$$

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

$\because q = n - k$

$k = n - q$

→ Putting the value of $n = 2^q - 1$, we get

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

→ By above eq? we observed that $r \simeq 1$ if $q >> 1$

Error detection & correction using Hamming Codes:-

→ Because the Min. distance (d_{min}) of hamming Code = 3

→ So it can detect $d_{min} = s + 1$

$3 = s + 1 \Rightarrow s = 2$.

So it can detect 2-errors.

→ & correct 1 error $d_{min} = 2t + 1$

$3 = 2t + 1$

$2t = 2$

$t = 1$ = errors which can be corrected

Q. The Parity checks matrix of a Particular (7,4) LBC 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 the generator matrix (G)

ii) List all the code vectors.

iii) what is the minimum distance between code vectors?

iv) How many errors can be detected? How many errors can be corrected?

Sol^n here $n = 7$

$k = 4$

$q = n - k$

$q = 7 - 4 = 3$

Thus $n = 2^q - 1 = 2^3 - 1 = 7$

To determine the P Matrix

→ The Parity checks matrix is of $q \times n$ size and is given by ($q = 3$ & $n = 7$ & $k = 4$)

$$[H]_{3\times 7} = \begin{bmatrix} P_{11} & P_{21} & P_{31} & P_{41} & \cdots & 1 & 0 & 0 \\ P_{12} & P_{22} & P_{32} & P_{42} & \cdots & 0 & 1 & 0 \\ P_{13} & P_{23} & P_{33} & P_{43} & \cdots & 0 & 0 & 1 \end{bmatrix}$$

$$[H]_{3\times 7} = [P^T : I_3]$$

→ on Comparing Parity checks matrix

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

$$[P] = \begin{bmatrix} P_{11} & P_{12} & P_{13} \\ P_{21} & P_{22} & P_{23} \\ P_{31} & P_{32} & P_{33} \\ P_{41} & P_{42} & P_{43} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}_{4\times 3}$$

To obtain the Generator Matrix (G)

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

→ with $k=4$, $q=3$

$n=7$

$$G = [\, I_4 : P_{4\times 3} \,]_{4\times 7}$$

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & : & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & : & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & : & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & : & 0 & 1 & 1 \end{bmatrix}_{4\times 7}$$

$I_{4\times 4}$ $\quad$ $P_{4\times 3}$

This is the required generator Matrix

iii) To find all the code words :

To obtain equations for check bits

The check bits can be obtained using equation (3.2.7), i.e.,

$$C = MP$$

In the more general form we can use equation (3.2.8) i.e. (with $q = 3$, $k = 4$)

$$[C_1\ C_2\ C_3]_{1\times 3} = [m_1\ m_2\ m_3\ m_4]_{1\times 4}\,[P]_{4\times 3}$$

$$[C_1\ C_2\ C_3] = [m_1\ m_2\ m_3\ m_4]\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 0\\ 1 & 0 & 1\\ 0 & 1 & 1\end{bmatrix}_{4\times 3}$$

Solving the above equation with mod-2 addition we get,

$$C_1 = (1\times m_1) \oplus (1\times m_2) \oplus (1\times m_3) \oplus (0\times m_4)$$

$$C_2 = (1\times m_1) \oplus (1\times m_2) \oplus (0\times m_3) \oplus (1\times m_4)$$

and $$C_3 = (1\times m_1) \oplus (0\times m_2) \oplus (1\times m_3) \oplus (1\times m_4)$$

Thus the above equations are,

$$\left.\begin{aligned} C_1 &= m_1 \oplus m_2 \oplus m_3\\ C_2 &= m_1 \oplus m_2 \oplus m_4\\ \text{and}\quad C_3 &= m_1 \oplus m_3 \oplus m_4 \end{aligned}\right\} \quad \ldots (3.2.20)$$

To determine the code vectors

Consider for example $(m_1\ m_2\ m_3\ m_4) = 1\ 0\ 1\ 1$ we get,

$$C_1 = 1 \oplus 0 \oplus 1 = 0$$

$$C_2 = 1 \oplus 0 \oplus 1 = 0$$

and $$C_3 = 1 \oplus 1 \oplus 1 = 1$$

Thus for message vector of (1 0 1 1) the check bits are $(C_1\ C_2\ C_3) = 0\,0\,1$. Therefore the systematic block code of the code vector (code word) can be written as,

$$(m_1\ m_2\ m_3\ m_4\ C_1\ C_2\ C_3) = (1\ 0\ 1\ 1\ :\ 0\ 0\ 1)$$

Using the same procedure as given above, we can obtain the other code words or code vectors. Table 3.2.2 lists all the code vectors (code words). Table also lists the weight of each code word.

Sr. No.	Message vector M				Check bits (C) by eq. 3.2.20			Code vector or code word X							Weight of code vector w(X)
	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

Table 3.2.2 Code vectors of Ex. 3.2.2

iii) Minimum distance between codevectors

The Table 3.2.2 lists $2^k = 2^4 = 16$ code vectors along with their weights. The smallest weight of any non-zero code vector is 3. We know that the minimum distance is $d_{min} = 3$. Therefore we can write :

The minimum distance of a linear block code is equal to the minimum weight of any non zero code vector i.e.

$$d_{min} = [w(X)]_{min} \; ; X \neq (0\ 0\ \ldots\ 0) \qquad \ldots (3.2.21)$$

iv) Error detection and correction capabilities

Since $d_{min} = 3$,

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

$$3 \geq s + 1$$

or $$s \leq 2$$

Thus two errors will be detected.

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

$$3 \geq 2t + 1$$

or $$t \leq 1$$

Thus one error will be corrected.

The hamming code ($d_{min} = 3$) always two errors can be detected and single error can be corrected by its property.

3.2.3 Encoder of (7, 4) Hamming Code

Fig. 3.2.2 shows the encoder of (7, 4) Hamming code. This encoder is implemented for generator matrix of the example 3.2.2. The lower register contains check bits C_1, C_2 and C_3. These bits are obtained from the message bits by mod-2 additions. These additions are performed according to equation (3.2.20). The mod-2 addition operation is nothing but exclusive-OR operation.

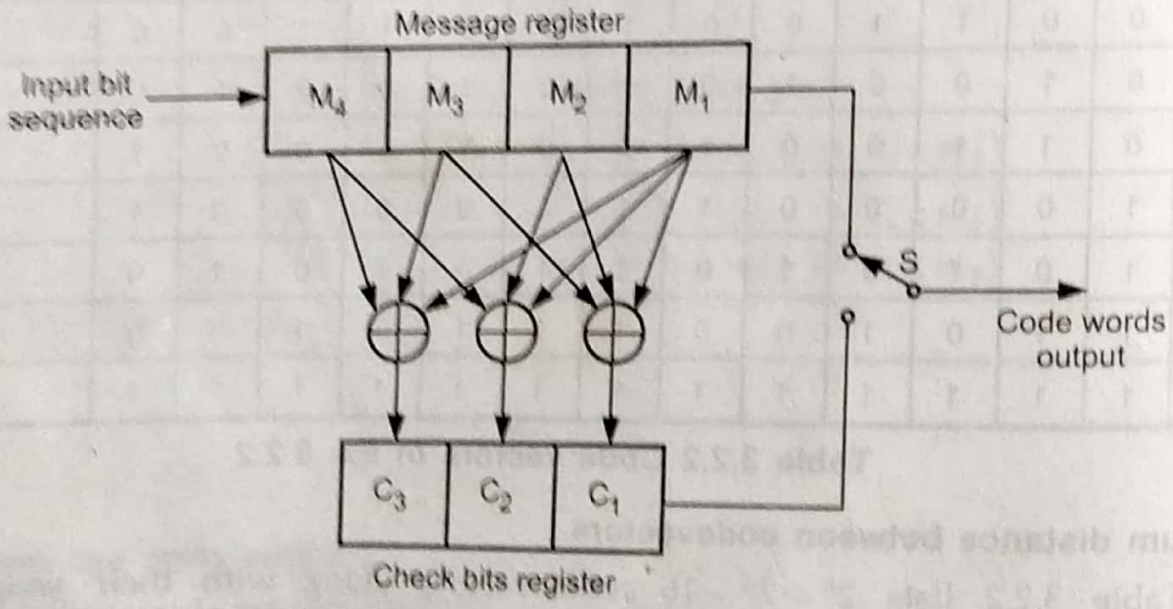


Fig. 3.2.2 Encode for (7, 4) hamming code or (7, 4) linear block code

The switch 'S' is connected to message register first and all message bits are transmitted. The switch is then connected to the check bit register and check bits are transmitted. This forms a block of '7' bits. The input bits are then taken for next block.

3.2.4 Syndrome Decoding

In this section we will see the method to correct errors in linear block coding. Let the transmitted code vector be 'X' and corresponding received code vector be represented by 'Y'. Then we can write,

$X = Y$ if there are no transmission errors

$X \neq Y$ if there are errors created during transmission

and

The decoder detects or corrects those errors in Y by using the stored bit pattern in the decoder about the code. For larger block lengths, more and more bits are required to be stored in the decoder. This increases the memory requirement and adds to the complexity and cost of the system. To avoid these problems, syndrome decoding is used in linear block codes. This method is illustrated in the subsequent paragraphs.

We know that with every (n, k) linear block code, there exists a parity check matrix (H) of size $q \times n$. From equation (3.2.11) it is defined as,

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

The transpose of the above matrix can be obtained by interchanging the rows and the columns, i.e.

$$H^T = \begin{bmatrix} P \\ \cdots \\ I_q \end{bmatrix}_{n \times q} \quad \ldots (3.2.22)$$

Here P is the submatrix of size $k \times q$ and I_q is the identity matrix of size $q \times q$. We have defined P submatrix in equation (3.2.12) earlier.

Important property used in syndrome decoding

The transpose of parity check matrix (H^T) has very important property as follows,

$$XH^T = (0\ 0\ 0\ \ldots\ 0) \quad \ldots (3.2.23)$$

or $$\boxed{[H]_{1 \times n}\ \left[H^T\right]_{n \times q} = (0\ 0\ 0\ \ldots\ 0)_{1 \times q}} \quad \ldots (3.2.24)$$

This is true for all code vectors.

Explanation with example

For example consider the parity check matrix and code vectors obtained in example 3.2.2. The parity check matrix is given by equation (3.2.18). The transpose of this matrix can be readily obtained as follows -

$$H^T = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}_{7 \times 3} \quad (n = 7 \text{ and } q = 3) \quad \ldots (3.2.25)$$

Table 3.2.2 lists all the code vectors for this parity check matrix. Consider the third code vector in this table.

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

Now let's apply the property of equation (3.2.23),

$$XH^T = [0010101]_{1\times 7} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}_{7\times 3}$$

Solving the above two matrices with the rules of mod-2 addition (Exclusive-OR operation) we get,

$$XH^T = (0 \oplus 0 \oplus 1 \oplus 0 \oplus 1 \oplus 0 \oplus 0\ 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0\ 0 \oplus 0 \oplus 1 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 1)$$

$$= (0\quad 0\quad 0)$$

This proves the property. It can be proved for other code vectors also. Thus X belongs to the valid code vector at the transmitter. At the receiver, the received code vector is Y. Then we can write,

$YH^T = (0\ 0\ \ldots .0)$, if $X = Y$ i.e. no errors or Y is valid code vector

YH^T = Non-zero, if $X \neq Y$ i.e. some errors.

Definition of syndrome (S)

When some errors are present in received vector Y, then it will not be from valid code vectors and it will not satisfy the property of equation (3.2.23). This shows that whenever YH^T is non-zero, some errors are present in Y. The non-zero output of the product YH^T is called *syndrome* and it is used to detect the errors in Y. Syndrome is represented by 'S' and can be written as,

$$S = YH^T \qquad \ldots (3.2.26)$$

or

$$\boxed{[S]_{1\times q} = [Y]_{1\times n}\,[H^T]_{n\times q}} \qquad \ldots (3.2.27)$$

Detecting error with the help of syndrome and error vector (E)

The non-zero elements of 'S' represent error in the output. When all elements of 'S' are zero, the two cases are possible.

i) No error in the output and Y = X

ii) Y is some other valid code vector other than X. This means the transmission errors are undetectable.

Lets consider on n-bit error vector E. Let this vector represent the position of transmission errors in Y. For example consider,

$X = (1\ 0\ 1\ 1\ 0)$ be a transmitted vector

$\uparrow \quad \uparrow$

and $Y = (1\ 0\ 0\ 1\ 1)$ be a received vector

$\uparrow \quad \uparrow$

Then $E = (0\ 0\ 1\ 0\ 1)$ represents the error vector

The non-zero entries represent errors in Y.

Using the mod-2 addition rules we can write,

$$Y = X \oplus E \qquad \ldots (3.2.28)$$

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

Bit by bit mod-2 addition

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

or we can write,

$$X = Y \oplus E \qquad \ldots (3.2.29)$$

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

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

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

From equation (3.2.26) we know that syndrome vector is given as,

$$S = YH^T$$

Putting the value of $Y = X \oplus E$. From equation (3.2.28) above

$$S = (X \oplus E)\, H^T$$

$$= XH^T \oplus EH^T$$

From the property of equation (3.2.23) we know that $XH^T = 0$, then above equation will be,

$$S = EH^T \qquad \ldots (3.2.30)$$

This relation shows that syndrome depends upon the error pattern only. It does not depend upon a particular message. Syndrome vector 'S' is of size $1 \times q$. Thus q bits of syndrome can only represent 2^q syndrome vectors. Each syndrome vector corresponds to a particular error pattern.

3.3 Cyclic Codes

Cyclic codes are the subclass of linear block codes. Cyclic codes can be in systematic or nonsystematic form. In systematic form, check bits are calculated separately and the code vector is $X = (M:C)$ form. Here 'M' represents message bits and 'C' represents check bits.

3.3.1 Definition of Cyclic Code

A linear code is called cyclic code if every cyclic shift of the codevector produces some other codevector. This definition includes two fundamental properties of cyclic codes. They are discussed next.

3.3.2 Properties of Cyclic Codes

As defined above, cyclic codes exhibit two fundamental properties :

1. Linearity and 2. Cyclic property

3.3.2.1 Linearity Property

This property states that sum of any two codewords is also a valid codeword. For example let X_1 and X_2 are two codewords. Then,

$$X_3 = X_1 \oplus X_2$$

Here X_3 is also a valid codeword. This property shows that cyclic code is also a linear code.

3.3.2.2 Cyclic Property

very cyclic shift of the valid code vector produces another valid codevector. Because of this property, the name 'cyclic' is given. Consider an n-bit codevector as shown below :

$$X = \{x_{n-1}, x_{n-2}, \ldots\ldots, x_1, x_0\} \quad \ldots (3.3.1)$$

Here $x_{n-1}, x_{n-2}, \ldots, x_1, x_0$ etc. represent individual bits of the codevector 'X'. Let us shift the above codevector cyclically to left side. i.e.,

One cyclic shift of X gives, $X' = (x_{n-2}, x_{n-3}, \ldots, x_1, x_0, x_{n-1})$ ... (3.3.2)

Here observe that every bit is shifted to left by one position. Previously x_{n-1} was MSB but after left cyclic shift it is at LSB position. Here the new code vector is X' and it is valid codevector. One more cyclic shift yields another codevector X''. i.e.,

$$X'' = (x_{n-3}, x_{n-4}, \ldots, x_1, x_0, x_{n-1}, x_{n-2}) \quad \ldots (3.3.3)$$

Here observe that x_{n-3} is now at MSB position and x_{n-2} is at LSB position.

3.3.3 Representation of Codewords by a Polynomial

The codewords can be represented by a polynomial.

For example, consider the n-bit codeword,

$$X = (x_{n-1}, x_{n-2}, \ldots\ldots, x_1, x_0)$$

This codeword can be represented by a polynomial of degree less than or equal to $(n-1)$. i.e.,

$$X(p) = x_{n-1}\, p^{n-1} + x_{n-2}\, p^{n-2} + \ldots + x_1\, p + x_0 \quad \ldots (3.3.4)$$

Here $X(p)$ is the polynomial of degree $(n-1)$.

p is the arbitrary variable of the polynomial.

The power of 'p' represent the positions of the code word bits. i.e.,

p^{n-1} represents MSB

p^0 represents LSB

p^1 represents second bit from LSB side.

Why to represent codewords by a polynomial ?

Polynomial representation is used due to following reasons :

i) These are algebraic codes. Hence algebraic operations such as addition, multiplication, division, subtraction etc. becomes very simple.

ii) Positions of the bits are represented with the help of powers of p in a polynomial.

3.3.4 Generation of Codevectors in Nonsystematic Form

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

$$M(p) = m_{k-1}\, p^{k-1} + m_{k-2}\, p^{k-2} + \ldots + m_1\, p + m_0 \qquad \ldots (3.3.5)$$

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

$$\boxed{X(p) = M(p)\, G(p)} \qquad \ldots (3.3.6)$$

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,

$$G(p) = p^q + g_{q-1}\, p^{q-1} + \ldots + g_1\, p + 1 \qquad \ldots (3.3.7)$$

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

If M_1, M_2, M_3 .... etc are the other message vectors, then the corresponding codevectors can be calculated as,

$$\begin{aligned} X_1(p) &= M_1(p)\, G(p) \\ X_2(p) &= M_2(p)\, G(p) \\ X_3(p) &= M_3(p) G(p) \text{ and so on} \end{aligned} \qquad \ldots (3.3.8)$$

All the above codevectors X_1, X_2, X_3 .... are in nonsystematic form and they satisfy cyclic property. Note the generator polynomial $G(p)$ remains the same for all codevectors.

Example 3.3.1 : *The generator polynomial of a (7, 4) cyclic code is* $G(p) = p^3 + p + 1$. *Find all the code vectors for the code in nonsystematic form.*

Solution : Here $n = 7$ and $k = 4$ therefore $q = n - k = 3$.

There will be total $2^k = 2^4 = 16$ message vectors of 7 bits each. Consider any message vector as,

$$M = (m_3\ m_2\ m_1\ m_0) = (0\ 1\ 0\ 1)$$

Then the message polynomial will be ($k = 4$ in equation (3.3.5)),

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

$$\therefore \quad M(p) = p^2 + 1 \qquad \ldots (3.3.9)$$

And given generator polynomial is,

$$G(p) = p^3 + p + 1 \qquad \ldots (3.3.10)$$

To obtain non-systematic code vectors

The non systematic cyclic code is given by equation (3.3.6) as,

$$\begin{aligned} X(p) &= M(p)\ G(p) \\ &= (p^2 + 1)(p^3 + p + 1) \\ &= p^5 + p^3 + p^2 + p^3 + p + 1 \\ &= p^5 + p^3 + p^3 + p^2 + p + 1 \\ &= p^5 + (1 \oplus 1)p^3 + p^2 + p + 1 \\ &= p^5 + p^2 + p + 1 \qquad (\text{since } (1 \oplus 1)p^3 = 0p^3 = 0) \\ &= 0p^6 + p^5 + 0p^4 + 0p^3 + p^2 + p + 1 \end{aligned}$$

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

$$\begin{aligned} X &= (x_6\ x_5\ x_4\ x_3\ x_2\ x_1\ x_0) \\ &= (0\ 1\ 0\ 0\ 1\ 1\ 1) \end{aligned}$$

This is the code vector for message vector 0101. This code vector is non systematic cyclic code vector. Similarly other code vectors can be obtained using the same procedure. Table 3.3.1 lists the codevectors in nonsystematic form.

Sr. No.	Message bits $M = m_3\ m_2\ m_1\ m_0$	Nonsystematic code vectors $X = x_6\ x_5\ x_4\ x_3\ x_2\ x_1\ x_0$
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

Table 3.3.1 Code vectors of a (7, 4) cyclic code for $G(p) = p^3 + p + 1$

3.3.5 Generation of Codevectors in Systematic Form

Now let us study systematic cyclic codes. The systematic form of the block code is,

$$X = \text{(k message bits : } (n-k) \text{ check bits)} \quad \ldots (3.3.11)$$

$$= (m_{k-1}\, m_{k-2} \ldots m_1\, m_0 : c_{q-1}\, c_{q-2} \ldots c_1\, c_0) \quad \ldots (3.3.12)$$

Here the check bits form a polynomial as,

$$C(p) = c_{q-1}\, p^{q-1} + c_{q-2}\, p^{q-2} + \ldots c_1\, p + c_0 \quad \ldots (3.3.13)$$

The check bit polynomial is obtained by,

$$\boxed{C(p) = rem\left[\frac{p^q\, M(p)}{G(p)}\right]} \quad \ldots (3.3.14)$$

Above equation means -

i) Multiply message polynomial by p^q.

ii) Divide $p^q M(p)$ by generator polynomial.

iii) Remainder of the division is $C(p)$.

Example 3.3.2 : *The generator polynomial of a (7, 4) cyclic code is* $G(p) = p^3 + p + 1$.

Find all the code vectors for the code in systematic form.

Solution : Here $n = 7$ and $k = 4$ therefore $q = n - k = 3$.

There will be total $2^k = 2^4 = 16$ message vectors of 7 bits each. Consider any message vector as,

$$M = (m_3\ m_2\ m_1\ m_0) = (0\ 1\ 0\ 1)$$

Then the message polynomial will be ($k = 4$ in equation (3.3.5)),

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

$$\therefore \quad M(p) = p^2 + 1 \qquad \ldots (3.3.15)$$

And given generator polynomial is,

$$G(p) = p^3 + p + 1 \qquad \ldots (3.3.16)$$

To obtain $p^q M(p)$

Since $q = 3$, $p^q M(p)$ will be,

$$p^q M(p) = p^3 M(p)$$
$$= p^3 (p^2 + 1)$$
$$= p^5 + q^3$$
$$= p^5 + 0p^4 + p^3 + 0p^2 + 0p + 0 \qquad \text{(for message vector of 0101)}$$

and $$G(p) = p^3 + p + 1$$
$$= p^3 + 0p^2 + p + 1$$

To perform the division $\dfrac{p^q M(p)}{G(p)}$

We now have $p^q M(p)$ and $G(p)$. Now let's perform the division to find remainder of this division.

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

Thus the remainder polynomial is $p^2 + 0p + 0$ in the division of $p^3\, M(p)$ by $G(p)$. Therefore equation (3.3.14) can be written as,

$$C(p) = rem\left[\frac{p^3\, M(p)}{G(p)}\right] = p^2 + 0p + 0$$

With $q = 3$ the polynomial $C(p)$ from equation 3.3.13 is,

$$C(p) = c_2 p^2 + c_1 p + c_0$$

Thus $c_2 p^2 + c_1 p + c_0 = p^2 + 0p + 0$

Therefore the check bits are

$$C = (c_2 c_1 c_0) = (1\ 0\ 0)$$

The code vector is written in system form as given by equation (3.3.12) i.e.,

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

In this example $X = (m_3 m_2 m_1 m_0 : c_2 c_1 c_0) = (0\ 1\ 0\ 1 : 1\ 0\ 0)$

This is the required cyclic code vectors in systematic form. The other code vectors can be obtained using the same procedure.

Table 3.3.2 lists all the systematic cyclic codes.

Sr. No.	Message bits $M = m_3\ m_2\ m_1\ m_0$	Systematic code vectors $X = m_3\ m_2\ m_1\ m_0\ c_2\ c_1\ c_0$
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

Table 3.3.2 Code vectors of a (7, 4) cyclic code for $G(p) = p^3 + p + 1$

3.3.6 Generator and Parity Check Matrices of Cyclic Codes

3.3.6.1 Nonsystematic Form of Generator Matrix

Since cyclic codes are subclass of linear block codes, generator and parity check matrices can also be defined for cyclic codes. The generator matrix has the size of $k \times n$. That means there are 'k' rows and 'n' columns. Let the generator matrix $G(p)$ be given by equation (3.3.7) as,

$$G(p) = p^q + g_{q-1}p^{q-1} + \ldots + g_1 p + 1 \quad \ldots (3.3.32)$$

Multiply both the sides of this polynomial by p^i i.e.,

$$\boxed{p^i\, G(p) = p^{i+q} + g_{q-1}\, p^{i+q-1} + \ldots + g_1\, p^{i+1} + p^i} \quad \ldots (3.3.33)$$

and $i = (k-1), (k-2), \ldots, 2, 1, 0$.

The above equation gives the polynomials for the rows of a generating polynomials. This procedure will be clear after the discussion of next example.

Example 3.3.4 : *Obtain the generator matrix corresponding to* $G(p) = p^3 + p^2 + 1$ *for a (7, 4) cyclic code.*

Solution : Here $n = 7$, $k = 4$ and $q = 7 - 4 = 3$ $p^i\, G(p)$ will be,

$$p^i\, G(p) = p^{i+3} + p^{i+2} + p^i \quad \text{for given } G(p)$$

Since $k - 1 = 3$; $i = 3, 2, 1, 0$

Thus we will obtain four polynomials corresponding to 4 values of 'i'. These four polynomials represent rows of generator matrix,

$$\left.\begin{aligned}
&\text{For row 1 : } i=3 \Rightarrow p^3\,G(p) = p^6+p^5+p^3\\
&\text{For row 2 : } i=2 \Rightarrow p^2\,G(p) = p^5+p^4+p^2\\
&\text{For row 3 : } i=1 \Rightarrow p\,G(p) = p^4+p^3+p\\
&\text{For row 4 : } i=0 \Rightarrow G(p) = p^3+p^2+1
\end{aligned}\right\} \quad \ldots (3.3.34)$$

The generator matrix for (n, k) code is of size $k \times n$. For this $(7, 4)$ cyclic code the size will be 4×7. Corresponding to four rows we have obtained four polynomials given by above equation. Let's write each polynomial in the following way.

$$\left.\begin{aligned}
&\text{Row 1} \Rightarrow p^3\,G(p) = p^6+p^5+0p^4+p^3+0p^2+0p+0\\
&\text{Row 2} \Rightarrow p^2\,G(p) = 0p^6+p^5+p^4+0p^3+p^2+0p+0\\
&\text{Row 3} \Rightarrow p\,G(p) = 0p^6+0p^5+p^4+p^3+0p^2+p+0\\
&\text{Row 4} \Rightarrow G(p) = 0p^6+0p^5+0p^4+p^3+p^2+0p+1
\end{aligned}\right\} \quad \ldots (3.3.35)$$

Let's transform the above set of polynomials into a matrix of 4×7

$$G_{4\times 7} = \begin{array}{c} \\ \text{Row 1} \\ \text{Row 2} \\ \text{Row 3} \\ \text{Row 4} \end{array} \begin{array}{c} \begin{array}{ccccccc} p^6 & p^5 & p^4 & p^3 & p^2 & p^1 & p^0 \end{array} \\ \left[\begin{array}{ccccccc} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{array}\right]_{4\times 7} \end{array} \quad \ldots (3.3.36)$$

This is the generator matrix for given generator matrix.

Example 3.3.5 : *Find out the possible generator polynomials (7, 4) cyclic code. Find out the code vectors corresponding to these generator polynomials.*

Solution : For this (7, 4) cyclic code,

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

We know that the generator polynomial is the factor of p^{n+1}. For this example, generator polynomial is the factor of $p^7 + 1$. The factors of $p^7 + 1$ are

$$p^7 + 1 = (p \oplus 1)(p^3 \oplus p^2 \oplus 1)(p^3 \oplus p \oplus 1)$$

The valid generating polynomial is given by, ... (3.3.37)

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

Thus the degree of the generating polynomial should be 'q'. For this example $q = 3$. Therefore the valid generator polynomials for $p^7 + 1$ will be $p^3 + p^2 + 1$ and $p^3 + p + 1$. $p+1$ will not be a generator polynomials. Since its degree is not q (i.e. 3). Thus generator polynomials for (7, 4) cyclic code are,

$$G_1(p) = p^3 + p^2 + 1 \quad \ldots (3.3.38)$$

and

$$G_2(p) = p^3 + p + 1 \quad \ldots (3.3.39)$$

Example 3.3.6 : *Find out the generator matrix corresponding to* $G(p) = p^3 + p + 1$ *and find out the code vectors for (7, 4) cyclic code.*

Solution : (i) To obtain generator matrix

The rows of a generator matrix are given by $p^i\, G(p)$. Here,

$$p^i\, G(p) = p^{i+3} + p^{i+1} + p^i$$

and $i = 3, 2, 1, 0$ since $k - 1 = 3$

$$\left.\begin{aligned} &\text{For row 1 : } i = 3 \Rightarrow p^3\, G(p) = p^6 + p^4 + p^3 \\ &\text{For row 2 : } i = 2 \Rightarrow p^2\, G(p) = p^5 + p^3 + p^2 \\ &\text{For row 3 : } i = 1 \Rightarrow p\, G(p) = p^4 + p^2 + p^1 \\ &\text{For row 4 : } i = 0 \Rightarrow G(p) = p^3 + p + 1 \end{aligned}\right\} \quad \ldots (3.3.40)$$

The above set of polynomials is transformed into a generator matrix of size 4×7 (i.e. $k \times n$) as shown below.

$$G = \begin{array}{c} \\ \text{Row 1} \\ \text{Row 2} \\ \text{Row 3} \\ \text{Row 4} \end{array} \begin{array}{c} \begin{array}{ccccccc} p^6 & p^5 & p^4 & p^3 & p^2 & p^1 & p^0 \end{array} \\ \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}_{4 \times 7} \end{array}$$

Since cyclic code is a subclass of linear block code, its code vectors can be obtained by equation (3.3.4) i.e.

$$X = MG \quad \ldots (3.3.41)$$

ii) To obtain the codevectors

Here M is the $1 \times k$ message vector and G is generator matrix. Here $k = 4$. Let's consider any 4 bit message vector

$$M = (m_3\ m_2\ m_1\ m_0) = (1\ 0\ 0\ 1)$$

The code vector corresponding to this message vector will be,

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

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

(**Note** : Here we perform matrix multiplication and additions are performed by mod-2 rules. i.e. $1 \oplus 1 = 0$, $1 \oplus 0 = 1$, $0 \oplus 1 = 1$ and $0 \oplus 0 = 0$).

This code vector we have already obtained in example 3.3.1 and is listed in Table 3.3.1. This code vector is in non systematic form. Also observe that generator matrix is also in nonsystematic form. Similarly other code vectors for cyclic code can be obtained.

Note : Here note that generator matrix is not in systematic form hence parity check matrix cannot be obtained using direct method.

3.3.6.2 Systematic Form of Generator Matrix

The systematic form of generator matrix is given by equation (3.3.6) as,

$$G = \left[I_k : P_{k \times q}\right]_{k \times n} \quad \ldots (3.3.42)$$

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

$$t^{th} \text{ row of } G = p^{n-t} + R_t(p) \quad \text{where } t = 1, 2, 3, \ldots k \quad \ldots (3.3.43)$$

Let's divide p^{n-t} by a generator matrix $G(p)$. Then we can express the result of this division in terms of quotient and remainder. i.e.,

$$\frac{p^{n-t}}{G(p)} = \text{Quotient} + \frac{Remainder}{G(p)} \quad \ldots (3.3.44)$$

Here remainder will be a polynomial of degree less than 'q', since degree of $G(p)$ is 'q'. The degree of quotient will depend upon value of t.

Let's represent Remainder $= R_t(p)$

and Quotient $= Q_t(p)$

Generator Matrix for cyclic code Systematic form

$$[G] = [I : P]$$

I → Identity, P → submatrix.

→ Row of Parity Matrix based on generator Polynomial $G(P)$ is given by

$$1^{st} \text{ row} = \text{Rem.} \left[\frac{P^{n-1}}{G(P)}\right]$$

$$2^{nd} \text{ Row} = \text{Rem.} \left[\frac{P^{n-2}}{G(P)}\right]$$

$$3^{rd} \text{ Row} = \text{Rem} \left[\frac{P^{n-3}}{G(P)}\right]$$

$$\vdots$$

$$k^{th} \text{ Row} = \text{Rem.} \left[\frac{P^{n-k}}{G(P)}\right]$$

Q If Generator Polynomial of cyclic code is (7,4) is given by $g(P) = P^3 + P + 1$. then Construct generator matrix for Systematic form.

Sol^n (7,4) Code.

$n = 7$

$k = 4$.

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

1^{st} row of P-sub Matrix $= \text{Rem}\left[\frac{P^{n-1}}{G(P)}\right] = \text{Rem}\left[\frac{P^6}{G(P)}\right]$

$$= \text{Rem.}\left[\frac{P^6}{P^3+P+1}\right]$$

2^{nd} Row of P $= \text{Rem.}\left[\frac{P^{n-2}}{G(P)}\right] = \text{Rem.}\left[\frac{P^5}{P^3+P+1}\right]$

3rd Row of P. Matrix = Rem. $\left[\frac{p^{7-3}}{p^3+p+1}\right]$

4th Row of P " = Rem. $\left[\frac{p^{7-4}}{p^3+p+1}\right]$

1st Row

$x^3+x+1 \;\big|\; x^6$

$x^6 + x^4 + x^3$ (⊕ ⊕ ⊕)

$x^4 + x^3$

$x^4 + x^2 + x$ (⊕ ⊕ ⊕)

$x^3 + x^2 +$

$p^3+p+1 \;\big|\; p^6$

$p^6 + p^4 + p^3$ (⊕ ⊕ ⊕)

$p^4 + p^3$

$p^4 + p^2 + p$ (⊕ ⊕ ⊕)

$p^3 + p^2 + p$

$p^3 + p + 1$ (⊕ ⊕ ⊕)

$p^2 + 1$

Rem. → p^2+1

2nd Row

$p^3+p+1 \;\big|\; p^5$

$p^5 + p^3 + p^2$ (⊕ ⊕ ⊕)

$p^3 + p^2$

$p^3 + p + 1$ (⊕ ⊕ ⊕)

$p^2 + p + 1$

Rem. → p^2+p+1

3rd Row

$p^3+p+1 \;\big|\; p^4$

$p^4 + p^2 + p$ (⊕ ⊕ ⊕)

$p^2 + p$

→ p^2+p

4th Row

$$\begin{array}{r|l} & 1 \\ p^3+p+1 & \overline{p^3} \\ & p^3+p+1 \\ & \overline{\oplus\ \ 0\ \ \oplus} \\ & p+1 \end{array}$$

1st Row Rem. = p^2+1 = [1 0 1]

2nd " " = p^2+p+1 = [1 1 1]

3rd " " = p^2+p = [1 1 0]

4th " " = $p+1$ = [0 1 1]

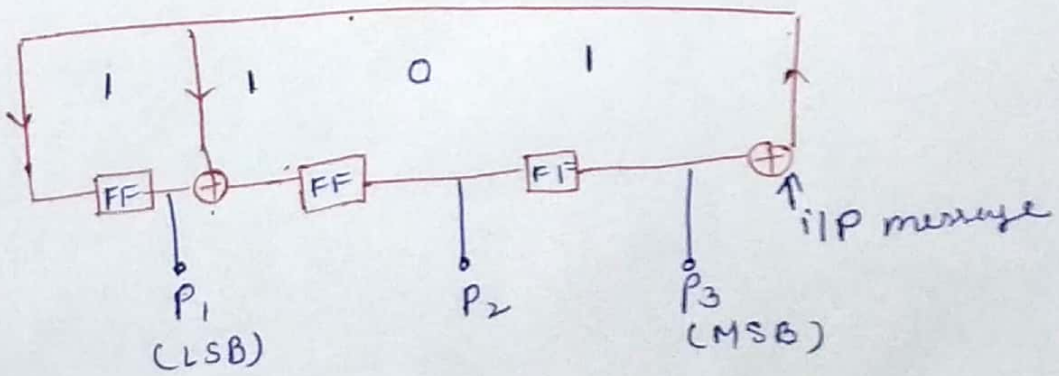
So $G = \left[\begin{array}{cccc:c} 1 & 0 & 0 & 0 & 1\,0\,1 \\ 0 & 1 & 0 & 0 & 1\,1\,1 \\ 0 & 0 & 1 & 0 & 1\,1\,0 \\ 0 & 0 & 0 & 1 & 0\,1\,1 \end{array}\right]$

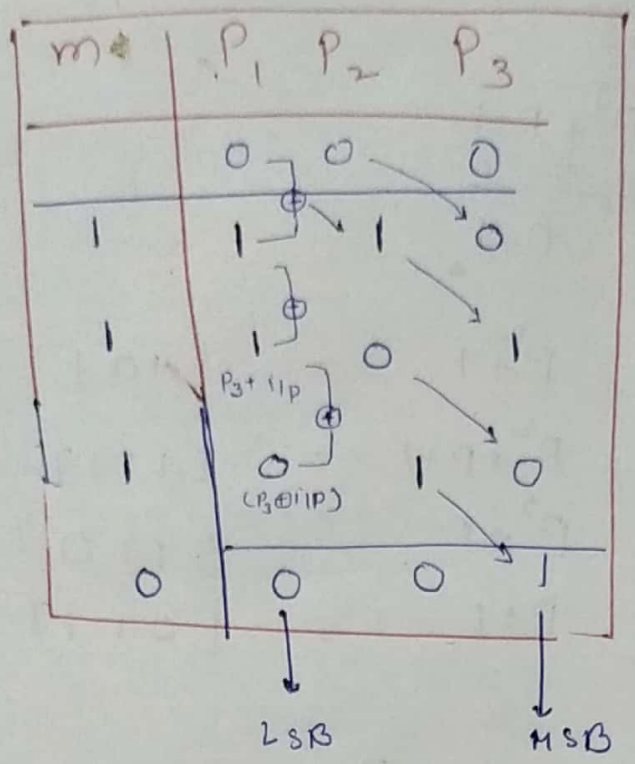
Cyclic Encoder

→ If Generator polynomial is $g(p) = 1+p+p^3$

calculate – 1. Cyclic Encoder
2. Codeword of message is (1110)

$$g(p) = 1+p+p^3$$
$$= 1+p+0\cdot p^2+p^3$$



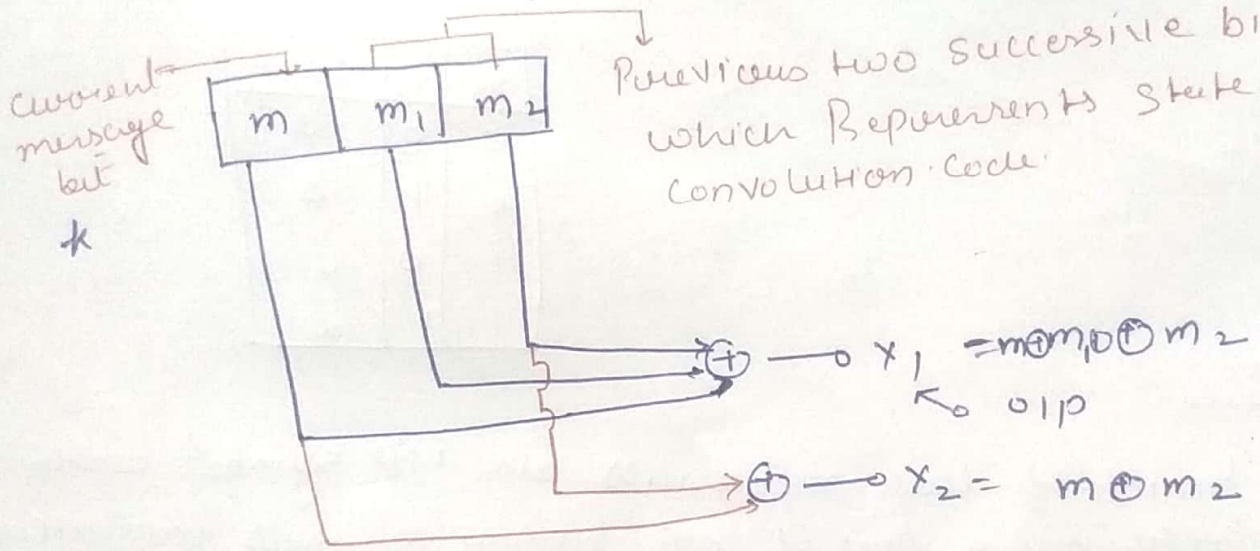


Codeword = [M : P]

= [1110 : 100]

Convolution Code Basic, Parameter & design :-

→ In convolution codes, blocks of 'n' code digits generated by the encoder in time unit depends on not only blocks of 'k' message digits with that time unit but also on the preceding (m-1) blocks of message digits.



k = no. of message bits. = 1

n = no. of encoded o/p bits = 2

K = constrain length. = 3

m_1	m_2	state
0	0	a
0	1	b
1	0	c
1	1	d

Code Rate = r

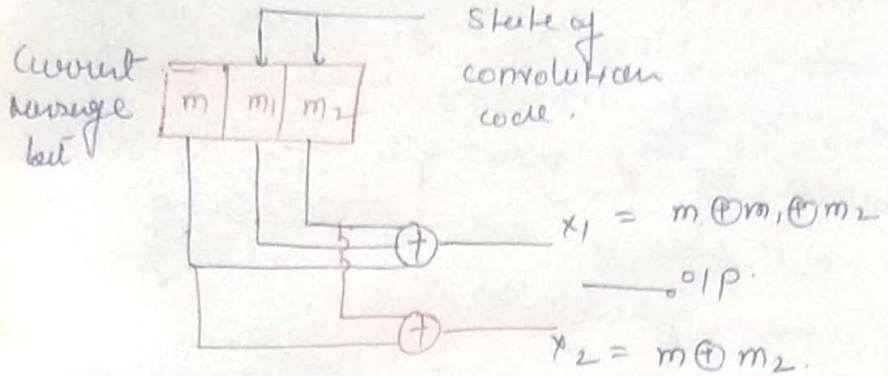
$$r = \frac{k}{n} = \frac{1}{2}$$

Constraint length

Single message bit influences encoder o/p for different successive shifts

Code dimensions (n, k) = (2, 1)

Convolution Code States & Code tree :-

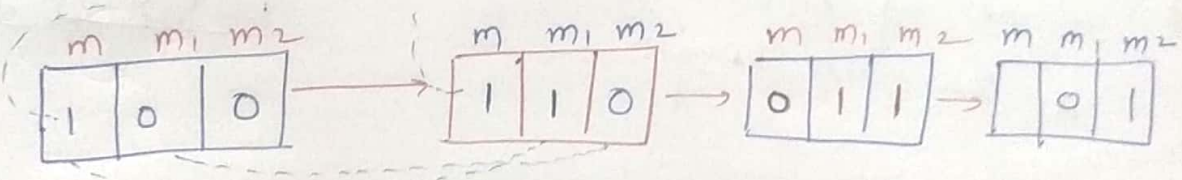


m_1	m_2	State
0	0	a
0	1	b
1	0	c
1	1	d

Code tree

Each branch of tree represents an i/p symbol with the corresponding pair of o/p binary symbols indicating on the branch.

→ let give i/p = 110



$x_1 = m \oplus m_1 \oplus m_2$
$= 1 \oplus 0 \oplus 0 = 1$
$x_2 = 1 \oplus 0 = 1$

$x_1 x_2 = 11$

State = a
$(m_1 m_2 = 00)$

$x_1 = 0$
$x_2 = 1$

$x_1 x_2 = 01$

State → c

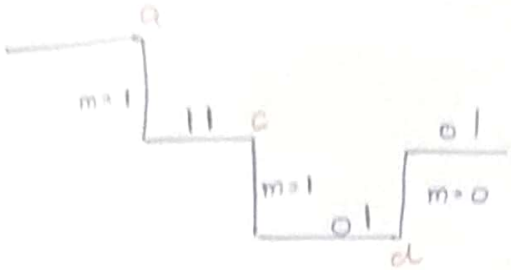
$x_1 = 0$
$x_2 = 1$

$x_1 x_2 = 01$

State → d

State → b

down → i/p = 1
upstep → i/p = 0



Code trellis & State Diagram of Convolution Code

m	m_1 →	m_2 →	X_1 $m \oplus m_1 \oplus m_2$	X_2 $m \oplus m_2$	Current State	Next State
0	0	0	0	0	a	a
1	0	0	1	1	a	c
0	0	1	1	1	b	a
1	0	1	0	0	b	c
0	1	0	1	0	c	b
1	1	0	0	1	c	d
0	1	1	0	1	d	b
1	1	1	1	0	d	d

Code trellis

Code Analysis

——— i/p bit = 0
- - - - - - i/p bit = 1

State Diagram

——→ i/p = 0
- - - → i/p = 1