ITC - UNIT 4 (Error Control Coding) - J S Chitode.

The statement of above theorem can also be written as,

A continuous time signal can be completely represented in its samples and recovered back if the sampling frequency $f_s \geq 2W$. Here f_s is the sampling frequency and W is the maximum frequency present in the signal.

Q.5 Mention two merits of DPCM.

Ans. : i) Bandwidth requirement of DPCM is less compared to PCM.

ii) Quatization error is reduced because of prediction filter.

iii) Number of bits used to represent one sample value are also reduced compared to PCM.

Q.6 What is the main difference in DPCM and DM ?

Ans. : DM encodes the input sample by only one bit. It sends the information about $+\delta$ or $-\delta$, i.e. step rise or fall. DPCM can have more than one bit for encoding the sample. It sends the information about difference between actual sample value and predicted sample value.

Q.7 How the message can be recovered from PAM ?

Ans. : The message can be recovered from PAM by passing the PAM signal through reconstruction filter. The reconstruction filter integrates amplitudes of PAM pulses. Amplitude smoothing of the reconstructed signal is done to remove amplitude discontinuities due to pulses.

Q.8 Write an expression for bandwidth of binary PCM with N messages each with a maximum frequency of f_m Hz.

Ans. : If 'v' number of bits are used to code each input sample, then bandwidth of PCM is given as,

$$B_T \geq N \cdot v \cdot f_m$$

Here $v \cdot f_m$ is the bandwidth required by one message.

Q.9 How is PDM wave converted into PPM systems ?

Ans. : The PDM signal is given as a clock signal to monostable multivibrator. The multivibrator triggers on falling edge. Hence a PPM pulse of fixed width is produced after falling edge of PDM pulse. PDM represents the input signal amplitude in the form of width of the pulse. A PPM pulse is produced after this 'width' of PDM pulse. In other words, the position of the PPM pulse depends upon input signal amplitude.

Q.10 Mention the use of adaptive quantizer in adaptive digital waveform coding schemes.

Ans. : Adaptive quantizer changes its step size according to variance of the input signal. Hence quantization error is significantly reduced due to adaptive quantization. ADPCM uses adaptive quantization. The bit rate of such schemes is reduced due to adaptive quantization.

Q.11 What do you understand from adaptive coding ?

Ans. : In adaptive coding, the quantization step size and prediction filter coefficients are changed as per properties of input signal. This reduces the quantization error and number of bits used to represent the sample value. Adaptive coding is used for speech coding at low bit rates.

Q.12 What is meant by quantization ?

Ans. : While converting the signal value from analog to digital, quantization is performed. The analog value is assigned to the nearest digital level. This is called quantization. The quantized value is then converted to equivalent binary value. The quantization levels are fixed depending upon the number of bits. Quantization is performed in every Analog to Digital Conversion.

Q.13 The signal to quantization noise ratio in a PCM system depends on ...

Ans. : The signal to quantization noise ratio in PCM is given as,

$$\left(\frac{S}{N}\right)_{dB} \leq (4.8 + 6\,v)\text{ dB}$$

Here v is the number of bits used to represent samples in PCM. Hence signal to quantization noise ratio in PCM depends upon number of bits or quantization levels.

Q.14 For the transmission of normal speech signal in the PCM channel needs the B.W. of ...

Ans. : Speech signals have the maximum frequency of 3.4 kHz. Normally 8 bits PCM is used for speed. The transmission bandwidth of PCM is given as,

$$B_T \geq vW$$

$$\geq 8 \times 3.4 \text{ kHz} \qquad \text{i.e. } 27.2 \text{ kHz}$$

Q.15 It is required to transmit speech over PCM channel with 8-bit accuracy. Assume the speech in baseband limited to 3.6 kHz. Determine the bit rate ?

Ans. : The signaling rate in PCM is given as,

$$r = vf_s$$

Here v number of bits i.e. 8

The maximum signal frequency is W = 3.6 kHz. Hence minimum sampling frequency will be,

$$f_s = 2\text{ W}$$

$$= 2 \times 3.6 \text{ kHz}$$

$$= 7.2 \text{ kHz}$$

$$\therefore \quad r = 8 \times 7.2 \times 10^3$$

$$= 57.6 \text{ kbits/sec}$$

Q.16 What is meant by adaptive delta modulation ?

Ans. : In adaptive delta modulation, the step size is adjusted as per the slope of the input signal. Step size is made high if slope of the input signal is high. This avoids slope overload distortion.

Q.17 What is the advantage of delta modulation over pulse modulation schemes ?

Ans. : Delta modulation encodes one bit per sample. Hence signaling rate is reduced in DM.

Q.18 What should be the minimum bandwidth required to transmit a PCM channel ?

Ans. : The minimum transmission bandwidth in PCM is given as,

$$B_T = vW$$

Here v is number of bits used to represent one pulse.

W is the maximum signal frequency.

Q.19 What is the advantage of delta modulation over PCM ?

Ans. : Delta modulation uses one bit to encode one sample. Hence bit rate of delta modulation is low compared to PCM.

□□□

UNIT-IV

3

Error Control Coding

3.1 Introduction

Errors are introduced in the data when it passes through the channel. The channel noise interferes the signal. The signal power is also reduced. Hence errors are introduced. In this chapter we will study various types of error detection and correction techniques.

3.1.1 Rationale for Coding and Types and Codes

The transmission of the data over the channel depends upon two parameters. They are transmitted power and channel bandwidth. The power spectral density of channel noise and these two parameters determine signal to noise power ratio. The signal to noise power ratio determine the probability of error of the modulation scheme. For the given signal to noise ratio, the error probability can be reduced further by using coding techniques. The coding techniques also reduce signal to noise power ratio for fixed probability of error.

Fig. 3.1.1 shows the block diagram of the digital communication system which uses channel coding.

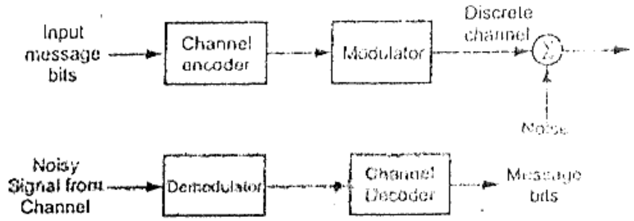


Fig. 3.1.1 Digital communication system with channel encoding

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 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. The systems become slightly complex because of coding techniques.

3.1.2 Types of Codes

The codes are mainly classified as block codes and convolutional codes.

i) 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.

ii) 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.

i) Linear code : If the two code words of the linear code are added by modulo-2 arithmetic, then it produces third codeword in the code.

This is very important property of the codes, since other codewords can be obtained by addition of existing codewords.

ii) Nonlinear code : Addition of the nonlinear codewords does not necessarily produce third codeword.

3.1.3 Discrete Memoryless Channels

We have defined discrete memoryless channels in the previous chapters. Consider the digital communication system of Fig. 3.1.1. Let the output of channel decoder depends only on the present transmitted signal, and it does not depend on any previous transmission. Then the modulator, discrete channel and demodulator of Fig. 3.1.1 can be combinely modeled as a discrete memoryless channel. We know that such channel is completely described by transition probabilities $P(y_j / x_i)$. Here x_i is the input symbol to modulator from channel encoder. And y_j is the output of demodulator and input to the channel decoder. $P(y_j / x_i)$ represents the probability of receiving symbol y_j, given that symbol x_i was transmitted.

3.1.4 Examples of Error Control Coding

Let us consider the error control coding scheme which transmits 000 to transmit symbol '0' and 111 to transmit symbol '1'. Here note that there are two redundant bits in every message (symbol) being transmitted. The decoder checks the received triplets and takes the decision in favour of majority of the bits. For example if the triplet is 110, then there are two 1's. Hence decision is taken in favour of 1. Here note that there is centainly error introduced in the last bit. Similarly if the received triplet is 001 or 100 or 010, then the decision is taken in favour of symbol '0'. The 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. Error control coding has following important aspects :

i. The redundancy bits in the message are called check bits. Errors can be detected and corrected with the help of these bits.

ii. It is not possible to detect and correct all the error in the message. Errors upto certain limit can only be detected and corrected.

iii. The check bits reduce the data rate through the channel.

3.1.5 Methods of Controlling Errors

There are two main methods used for error control coding : Forward acting error correction and Error detection with transmission.

I) Forward acting error correction

In this method, the errors are detected and corrected by proper coding techniques at the receiver (decoder). The check bits or redundant bits are used by the receiver to detect and correct errors. The error detection and correction capability of the receiver depends upon number of redundant bits in the transmitted message. The forward acting error correction is faster, but over all probability of errors is higher. This is because some of the errors cannot be corrected.

II) Error detection with retransmission

In this method, the decoder checks the input sequence. When it detects any error, it discards that part of the sequence and requests the transmitter for retransmission. The transmitter then again transmits the part of the sequence in which error was detected. Here note that, the decoder does not correct the errors. It just detects the errors and sends requests to transmitter. This method has lower probability of error, but it is slow.

3.1.6 Types of Errors

There are mainly two types of errors introduced during transmission on the data, random errors and burst errors.

i) Random errors : These errors are created due to white gaussian noise in the channel. The errors generated due to white gaussian noise in the particular interval does not affect the performance of the system in subsequent intervals. In other words, these errors are totally uncorrelated. Hence they are also called as random errors.

ii) Burst errors : These errors are generated due to impulsive noise in the channel. These impulse noise (bursts) are generated due to lightning and switching transients. These noise bursts affect several successive symbols. Such errors are called burst errors. The burst errors are dependent on each other in successive message intervals.

3.1.7 Some of the Important Terms Used in Error Control Coding

The terms which are regularly used in error control coding are defined next.

Code word : The encoded block of '*n*' bits is called a code word. 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 ratio of message bits (*k*) and the encoder output bits (*n*) is called code rate. Code rate is defined by '*r*' i.e.,

$$r = \frac{k}{n} \qquad \dots (3.1.1)$$

we find that $0 < r < 1$.

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,

$$\text{Channel data rate } (R_o) = \frac{n}{k} R_s \qquad \dots (3.1.2)$$

Code vectors : An '*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. It is simpler to visualize the 3-bit code words. Fig. 3.1.2 shows the 3-bit code vectors. There will be distinct '8' code words (since number of code words = 2^k). If we let bits b_0 on x-axis, b_1 on y-axis and b_2 on z-axis, then the following table gives various points as code vectors in the 3-dimensional space.

Sr.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 3.1.1 Code vectors in 3-dimensional space

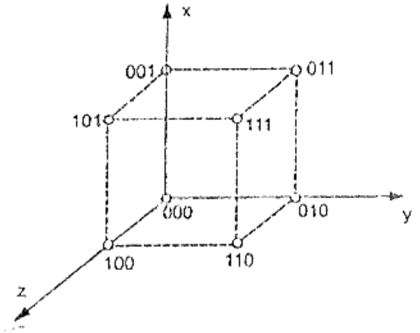


Fig. 3.1.2 Code vectors representing 3-bit code words

Hamming distance : The hamming distance between the two code vectors is equal to the number of elements in which they differ. For example let $X = (101)$ and $Y = (110)$. The two code vectors differ in second and third bits. Therefore hamming distance between X and Y is 'two'. Hamming distance is denoted as $d(X, Y)$ or simply 'd'. i.e.

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

Thus we observe from Fig. 3.1.2 that the hamming distance between (100) and (011) is maximum i.e. 3. This is indicated by the vector diagram also.

Minimum distance (d_{min}) : It is the smallest hamming distance between the valid code vectors.

Error detection is possible if the received vector is not equal to some other code vector. This shows that the transmission errors in the received code vector should be less than minimum distance d_{min}. The following table lists some of the requirements of error control capability of the code.

Sr.No.	Name of errors detected / corrected	Distance requirement
1	Detect upto 's' errors per word	$d_{min} \geq s+1$
2	Correct upto 't' errors per word	$d_{min} \geq 2t+1$
3	Correct upto 't' errors and detect $s > t$ errors per word	$d_{min} \geq t+s+1$

Table 3.1.2 Error control capabilities

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

$$d_{min} \leq n-k+1 \quad \text{... (3.1.3)}$$

Code efficiency : The code efficiency is the ratio of message bits in a block to the transmitted bits for that block by the encoder i.e.,

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

We know that for an (n, k) block code, there are 'k' message bits and 'n' transmitted bits. Therefore code efficiency becomes,

$$\text{Code efficiency} = \frac{k}{n} \quad \text{... (3.1.4)}$$

If we compare the above expression with the code rate (r) of equation (3.1.1) we find that,

$$\text{Code efficiency} = \text{code rate} = \frac{k}{n} \quad \text{... (3.1.5)}$$

Weight of the code : The number of non-zero elements in the transmitted code vector is called vector weight. It is denoted by $w(X)$ where X is the code vector. For example if $X = 01110101$, then weight of this code vector will be $w(X) = 5$.

Review Questions

1. *Briefly discuss the classification of codes.*
2. *Explain the following terms.*

   i) Hamming distance ii) Code rate iii) Free distance iv) Weight of code.
3. *What is error control coding ? Which are the functional blocks of a communication system that accomplish this ? Indicate the function of each block.*

3.2 Linear Block Codes

Principle of block coding :

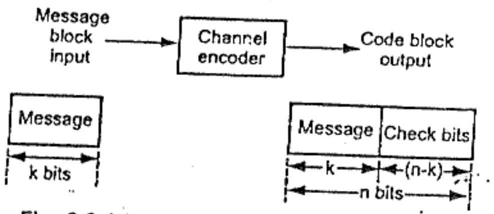


Fig. 3.2.1 Functional block diagram of block coder

For the block of k message bits, $(n-k)$ parity bits or check bits are added. Hence the total bits at the output of channel encoder are 'n'. Such codes are called (n, k) block codes. Fig.3.2.1 illustrates this concept.

Systematic codes : In the systematic block code, the message bits appear at the beginning of the code word. Thus as shown in Fig. 3.2.1, the message bits appear first and then check bits are transmitted in a block. This type of code is called systematic code. In *nonsystematic code* it is not possible to identify message bits and check bits. They are mixed in the block.

In this section we will consider binary codes. That is all transmitted digits are binary.

Linear code : A code is 'linear' if the *sum of any two code vectors produces another code vector. This shows that any code vector can be expressed as a linear combination of other code vectors. Consider that the particular code vector consists of $m_1, m_2, m_3, \dots, m_k$ message bits and $c_1, c_2, c_3, \dots, c_q$ check bits. Then this code vector can be written as,

$$X = (m_1, m_2, \dots m_k \; c_1, c_2, \dots c_q) \quad \text{... (3.2.1)}$$

Here $q = n-k$ ... (3.2.2)

i.e. q are the number of redundant bits added by the encoder. The above code vector can also be written as,

$$X = (M|C) \quad \text{... (3.2.3)}$$

Here M = k - bit message vector and

C = q - bit check vector

The check bits play the role of error detection and correction. The job of the linear block code is to generate those 'check bits'. The code vector can be represented as,

$$X = MG \quad \text{... (3.2.4)}$$

Here X = Code vector of $1 \times n$ size or n bits

* Here 'sum' is performed according to mod-2 addition rules. i.e. $1 \oplus 1 = 0$; $1 \oplus 0 = 1$; $0 \oplus 1 = 1$ and $0 \oplus 0 = 0$.

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

and G = Generator matrix of $k \times n$ size.

Thus equation (3.2.4) above represents matrix form i.e.

$$\boxed{[X]_{1\times n} = [M]_{1\times k}\,[G]_{k\times n}} \quad \text{... (3.2.5)}$$

The generator matrix depends upon the linear block code used. Generally it is represented as,

$$G = \left[I_k \,|\, P_{k\times q} \right]_{k\times n} \quad \text{... (3.2.6)}$$

Here I_k = $k \times k$ identity matrix and

P = $k \times q$ submatrix

The check vector can be obtained as,

$$\boxed{C = MP} \quad \text{... (3.2.7)}$$

Thus in the expanded form we can write above equation as,

$$[C_1, C_2, ..., C_q]_{1\times q} = [m_1, m_2, ... m_k]_{1\times k} \begin{bmatrix} P_{11} & P_{12} & .... & P_{1q} \\ P_{21} & P_{22} & .... & P_{2q} \\ : & & & \\ : & & & \\ P_{k1} & P_{k2} & .... & p_{kq} \end{bmatrix}_{k\times q} \quad \text{... (3.2.8)}$$

By solving the above matrix equation, check vector can be obtained. i.e.

$$\left.\begin{aligned} C_1 &= m_1\,P_{11} \oplus m_2\,P_{21} \oplus m_3\,P_{31} \oplus .... \oplus m_k\,P_{k1} \\ C_2 &= m_1\,P_{12} \oplus m_2\,P_{22} \oplus m_3\,P_{32} \oplus .... \oplus m_k\,P_{k2} \\ C_3 &= m_1\,P_{13} \oplus m_2\,P_{23} \oplus m_3\,P_{33} \oplus .... \oplus m_k\,P_{k3} \\ & \text{.... and so on} \end{aligned}\right\} \quad \text{... (3.2.9)}$$

Here note that all the additions are mod-2 additions.

➡ Example 3.2.1 : *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 : The code vectors can be obtained through following steps :

i) Determine the P submatrix from generator matrix.

ii) Obtain equations for check bits using $C = MP$.

iii) Determine check bits for every message vector.

The above steps are explained next :

i) To obtain P' sub matrix :

From equation (3.2.6) we know that,

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

Comparing this equation with the given matrix, we find that,

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

and

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

ii) To obtain the equations for check bits :

Here $k = 3$, $q = 3$ and $n = 6$.

That is, the block size of the message vector is 3 bits. Hence there will be total 8 possible message vectors as shown below in the table.

Sr.No.	Bits of message vector in one block		
	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

The P submatrix is given in the example as,

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

For the check bit vector, there will be three bits. They can be obtained using equation (3.2.7) or (3.2.8) i.e.

$$[C_1\; C_2\; C_3] = [m_1\; m_2\; m_3] \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

From the above matrix multiplication we obtain,

$$C_1 = (0\times m_1)\oplus(m_2)\oplus(m_3)$$
$$C_2 = (m_1)\oplus(0\times m_2)\oplus(m_3)$$

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

From the above three equations we obtain,

$$\left.\begin{aligned} C_1 &= m_2\oplus m_3 \\ C_2 &= m_1\oplus m_3 \\ C_3 &= m_1\oplus m_2 \end{aligned}\right\} \qquad \ldots(3.2.10)$$

The above three equations give check bits for each block of m_1, m_2, m_3 message bits.

iii) To determine check bits and codevectors for every message vector :

Consider the first block of $(m_1, m_2, m_3) = 000$ we have,

$$C_1 = 0\oplus 0 = 0$$
$$C_2 = 0\oplus 0 = 0$$
$$C_3 = 0\oplus 0 = 0 \qquad \text{i.e.} \quad (C_1, C_2, C_3) = 000$$

For second block of $(m_1, m_2, m_3) = 001$ we have,

$$C_1 = 0\oplus 1 = 1$$
$$C_2 = 0\oplus 1 = 1$$
$$C_3 = 0\oplus 0 = 0 \qquad \text{i.e.} \quad (C_1, C_2, C_3) = 110$$

The following Table 3.2.1 lists all the message bits, their check bits and code vectors calculated as above.

Sr. No.	Bits of message vector in one block			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

Table 3.2.1 Code vectors of (6, 3) block code of example 3.2.1

Parity check matrix (H)

For every block code there is a $q\times n$ parity check matrix (H). It is defined as,

$$H = \left[P^T : I_q\right]_{q\times n} \qquad \ldots(3.2.11)$$

Here P^T is the transpose of P sub-matrix. The P submatrix is defined earlier in equation (3.2.8) as,

$$P = \begin{bmatrix} P_{11} & P_{12} & P_{13} & \ldots & P_{1q} \\ P_{21} & P_{22} & P_{23} & \ldots & P_{2q} \\ P_{31} & P_{32} & P_{33} & \ldots & P_{3q} \\ \vdots & \vdots & \vdots & & \vdots \\ \vdots & \vdots & \vdots & & \vdots \\ P_{k1} & P_{k2} & P_{k3} & \ldots & P_{kq} \end{bmatrix}_{k\times q} \qquad \ldots(3.2.12)$$

The transpose of this submatrix become (by changing rows to columns),

$$P^T = \begin{bmatrix} P_{11} & P_{21} & P_{31} & \ldots & P_{k1} \\ P_{12} & P_{22} & P_{32} & \ldots & P_{k2} \\ P_{13} & P_{23} & P_{33} & \ldots & P_{k3} \\ \vdots & \vdots & \vdots & & \vdots \\ \vdots & \vdots & \vdots & & \vdots \\ P_{1q} & P_{2q} & P_{3q} & \ldots & P_{kq} \end{bmatrix}_{q\times k} \qquad \ldots(3.2.13)$$

With the above equation, we can write equation (3.2.11) as

$$H_{q\times n} = \begin{bmatrix} P_{11} & P_{21} & P_{31} & \ldots & P_{k1} & : & 1 & 0 & 0 & .. & 0 \\ P_{12} & P_{22} & P_{32} & \ldots & P_{k2} & : & 0 & 1 & 0 & .. & 0 \\ P_{13} & P_{23} & P_{33} & \ldots & P_{k3} & : & 0 & 0 & 1 & .. & 0 \\ \vdots & \vdots & \vdots & & \vdots & : & : & : & : & .. & : \\ \vdots & \vdots & \vdots & & \vdots & : & : & : & : & .. & : \\ P_{1q} & P_{2q} & P_{3q} & \ldots & P_{kq} & : & 0 & 0 & 0 & .. & 1 \end{bmatrix}_{q\times n} \qquad \ldots(3.2.14)$$

If we compare equation (3.2.6) and equation (3.2.11). We find that if generator matrix (G) is given, then parity check matrix (H) can be obtained and vice-versa.

3.2.1 Hamming Codes

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

Those codes satisfy the following conditions,

$$\left.\begin{array}{ll} (1) & \text{Number of check bits } q \geq 3 \\ (2) & \text{Block length } n = 2^q - 1 \\ (3) & \text{Number of message bits } k = n - q \\ (4) & \text{Minimum distance } d_{\min} = 3 \end{array}\right\} \qquad \ldots (3.2.15)$$

We know that the code rate is given as,

$$\begin{aligned} r &= \frac{k}{n} \\ &= \frac{n-q}{n} \quad \text{for hamming code} \quad k = n - q \\ &= 1 - \frac{q}{n} \qquad \ldots (3.2.16) \end{aligned}$$

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

$$r = 1 - \frac{q}{2^q - 1} \qquad \ldots (3.2.17)$$

From the above equation we observe that $r \approx 1$ if $q >> 1$.

3.2.2 Error Detection and Correction Capabilities of Hamming Codes

Since the minimum distance $(d_{\min})$ of Hamming code is 3, it can be used to detect double errors or correct single errors. This can also be obtained from the generalized Table 3.2.2

For detecting double (2) errors $\Rightarrow d_{\min} \geq 2+1$ i.e. $d_{\min} \geq 3$

and for correcting upto one (1) errors $\Rightarrow d_{\min} \geq 2(1)+1$ i.e. $d_{\min} \geq 3$

Example 3.2.2 : *The parity check matrix of a particular (7, 4) linear block code is given by*

$$[H] = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \qquad \ldots (3.2.18)$$

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 ?

Solution : Here n = 7 and k = 4

∴ Number of check bits are n − k = 7 − 4 i.e. q = 3

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

This shows that the given code is hamming code.

To determine the P submatrix :

The parity check matrix is of $q \times n$ size and is given by equation (3.2.14). It can be written as, (with q = 3 and n = 7 and k = 4)

$$[H]_{3\times 7} = \begin{bmatrix} P_{11} & P_{21} & P_{31} & P_{41} & \ldots & 1 & 0 & 0 \\ P_{12} & P_{22} & P_{32} & P_{42} & \ldots & 0 & 1 & 0 \\ P_{13} & P_{23} & P_{33} & P_{43} & \ldots & 0 & 0 & 1 \end{bmatrix} \qquad \ldots (3.2.18\ (a))$$

$$= [P^T : I_3] \qquad \text{from equation 3.2.11}$$

On comparing parity check matrices of equation (3.2.18) and equation (3.2.17) we get,

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

Therefore the P submatrix can be obtained as,

$$P = \begin{bmatrix} P_{11} & P_{12} & P_{13} \\ P_{21} & P_{22} & P_{23} \\ P_{31} & P_{32} & P_{33} \\ P_{41} & P_{42} & P_{43} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}_{4\times 3} \qquad \ldots (3.2.18\ (b))$$

i) To obtain the generator matrix (G) :

From equation (3.2.6) the generator matrix G is given as,

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

with $k = 4$, $q = 3$ and $n = 7$ the above equation becomes,

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

Putting the identity m[illegible], of size 4 × 4 and parity submatrix $P_{4\times 3}$ of size 4 × 3 as obtained in equation (3.2.18) we get,

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & : & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & : & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & : & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & : & 0 & 1 & 1 \end{bmatrix}_{4\times 7} \qquad \text{... (3.2.19)}$$

$$\underbrace{\qquad}_{I_{4\times4}} \quad \underbrace{\qquad}_{P_{4\times3}}$$

This is the required generator matrix.

ii) 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 $q = 3$, $k = 4$)

$$[C_1\ C_2\ C_3]_{1\times3} = [m_1\ m_2\ m_3\ m_4]_{1\times4}\,[P]_{4\times3}$$

$$[C_1\ C_2\ C_3] = [m_1\ m_2\ m_3\ m_4]\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}_{4\times3}$$

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,

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

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 $(C_1\ C_2\ C_3) = 001$. 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.

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

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.

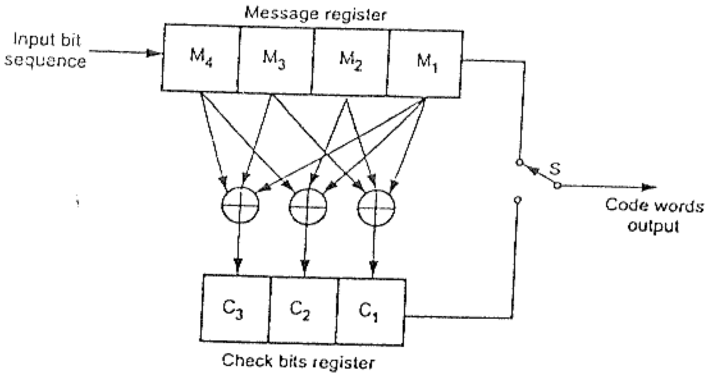


Fig. 3.2.2 Encode for (7, 4) hamming code or (7, 4) linear block code

The switch 'S' is connected to message register 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

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

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.

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

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 $\left(H^T\right)$ has very important property as follows.

$$XH^T = (0\ 0\ 0\ \dots\ 0) \qquad \dots (3.2.23)$$

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

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 (3.2.16). The transpose of this matrix can be readily obtained as follows -

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

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,

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

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

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,

$$X = (1\ 0\ 1\ 1\ 0) \quad \text{be a transmitted vector}$$

and

$$Y = (1\ 0\ 0\ 1\ 1) \quad \text{be a received vector}$$

Then

$$E = (0\ 0\ 1\ 0\ 1) \quad \text{represents the error vector}$$

The non-zero entries represent errors in Y.

Using the mod-2 addition rules we can write,

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

$$= (1 \oplus 0 \quad 0 \oplus 0 \quad 1 \oplus 1 \quad 1 \oplus 0 \quad 0 \oplus 1) \quad \text{Bit by bit mod-2 addition}$$

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

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.

➡ Example 3.2.3 : *The parity check matrix of a (7, 4). Hamming code is given as follows*

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

calculate the syndrome vector for single bit errors.

Solution : This is a (7, 4) linear block code.

This is $n=7$ and $k=4$

$$q = n-k=3$$

i) To determine error pattern for single bit errors

We know that syndrome vector is a q bit vector. For this example syndrome will be a 3 bit vector. Therefore there will be $2^3-1=7$ non-zero syndromes. This shows that '7' single bit error patterns will be represented by these '7' non-zero syndromes. Error vector E is a n bit vector representing error pattern. For this example E is '7' bit vector. Following Table 3.2.3 shows the single error patterns in a 7 bit error vector (Note that only single bit error patterns are shown).

Sr. No.	Bit in error	Bits of Error 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

Table 3.2.3 Single error pattern of 7 bit error vector

ii) Calculation of syndromes

From equation (3.2.30) the syndrome vector is given as,

$$S = EH^T$$

S is q bit, E is n bit and H^T is $n \times q$ bits size. For this example $n=7$, $q=3$ we can write above equation as,

$$[S]_{1\times 3} = [E]_{1\times 7}[H^T]_{7\times 3} \qquad \text{... (3:2.31)}$$

From the given parity check matrix H, we can obtain its transpose (H^T) by interchanging rows to columns, i.e.

$$H^T = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \text{... (3.2.32)}$$

Syndrome for first bit in error

Let's calculate syndrome for first bit error vector i.e.

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

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

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

This is the syndrome vector for first bit in error.

Syndrome for second bit in error

Let's calculate syndrome for 2nd bit in error. In Table 3.2.3 the error vector for error in 2nd bit is given. We can write,

$$S = EH^T = [0\,1\,0\,0\,0\,0\,0]\begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

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

$$= [1 \quad 1 \quad 1]$$

This is the syndrome vector for second bit in error.

Syndrome vectors are rows of H^T

The Table 3.2.4 lists the error vector with single bit error and corresponding syndromes. Other syndromes can be calculated using the same procedure as above. The table also lists the syndrome for no error vector i.e. $E=(0000000)$. Observe that the corresponding syndrome is $S=(0\ 0\ 0)$.

Sr. No.	Error vector 'E' showing single bit error patterns							Syndrome Vector 'S'			
1	0	0	0	0	0	0	0	0	0	0	
2	1	0	0	0	0	0	0	1	0	1	← 1st row of H^T
3	0	1	0	0	0	0	0	1	1	1	← 2nd row of H^T
4	0	0	1	0	0	0	0	1	1	0	← 3rd row of H^T
5	0	0	0	1	0	0	0	0	1	1	← 4th row of H^T
6	0	0	0	0	1	0	0	1	0	0	← 5th row of H^T
7	0	0	0	0	0	1	0	0	1	0	← 6th row of H^T
8	0	0	0	0	0	0	1	0	0	1	← 7th row of H^T

Table 3.2.4 Syndromes for (7, 4) Hamming code of single bit error

The following table shows that error in the first bit corresponds to a syndrome vector of $S=(101)$. This syndrome vector is same as the first row of H^T. The syndrome vector $(S=111)$ for error in second bit is same as second row of H^T. This is same for remaining syndromes.

3.2.4.1 Error Correction Using Syndrome Vector

Let's see how single bit errors can be corrected using syndrome decoding. We will see this for (7, 4) block code. Let the transmitted code vector be,

$$^{*}X = (1\ 0\ 0\ 1\ 1\ 1\ 0)$$

Let there be error created in the 3rd bit in the received code vector Y. Then Y will be

$$Y = (1\ 0\ \textcircled{1}\ 1\ 1\ 1\ 0) \quad \text{encircled bit shows it is in error.}$$

Now error correction can be done by adopting following steps :

i) Calculate the syndrome $S=YH^T$

ii) Check the row of H^T which is same as 'S'.

iii) For p^{th} row of H^T, p^{th} bit is in error. Hence write corresponding error vector (E).

* Here note that the assumed code vector X is derived for the parity check matrix of example 3.2.3. Students can verify this using the regular procedure explained in example 3.2.2. The code vectors depend upon the parity check matrix 'H'. Therefore in $S=YH^T$. We have used the same

iv) Obtain correct vector by $X=Y\oplus E$.

Above procedure is illustrated next :

i) To obtain syndrome vector (S)

Let's use the parity check matrix and syndrome vectors of example 3.2.2 for this illustration. The receiver calculates $S=YH^T$ i.e.

$$S = YH^T = \underbrace{[1\ 0\ 1\ 1\ 1\ 1\ 0]}_{Y} \underbrace{\begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1' & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{H^T} \qquad \text{From equation 3.2.26}$$

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

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

From equation (3.2.26) and equation (3.5.30) we can write

$$S = YH^T = EH^T \qquad \text{Here} \quad S=YH^T=EH^T=(1\ 1\ 0)$$

ii) To determine row of H^T which is same as 'S'

and (iii) To determine 'E'

On comparing this syndrome with H^T, we observe that $(S=1\ 1\ 0)$ is the 3rd row of H^T. From Table 3.2.4 we can obtain the error pattern corresponding to this syndrome as,

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

This shows that there is an error in the third bit of Y. We have also verified that, if syndrome vector is equal to 3rd row of H^T, then third bit of Y is in error.

iv) To obtain correct vector

The correct vector can be obtained from equation (3.2.29) as,

$$X = Y\oplus E$$

i.e.

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

$$= (1\ 0\ 0\ 1\ 1\ 1\ 0) \quad \text{which is same as transmitted code vector}$$

Thus a single bit errors can be corrected using syndrome decoding.

What happens if double error occurs in Y ?

Let's see the case of double error in Y. Consider the same message vector X. i.e.,

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

Let's consider that error is present in 3^{rd} and 4^{th} bits. Then Y will be

$$Y = 1\ 0\ \textcircled{1}\textcircled{0}\ 1\ 1\ 0 \quad \text{encircled bits are in error.}$$

Then $S = YH^T$ gives,

$$S = YH^T = [1\ 0\ 1\ 0\ 1\ 1\ 0]\begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$S = 101$$

From Table 3.2.4 we observe that the syndrome $S = 101$ corresponds to an error pattern of $E = 1000000$. This shows that there is an error in the first bit. Thus the error detection and correction goes wrong. The probability of occurrence of multiple errors is less compared to single errors. To correct multiple errors, extended hamming codes are used. In these codes one more extra bit is provided to correct double errors.

We know that for (n, k) block code, there are $2^q - 1$ distinct non-zero syndromes. There are ${}^nC_1 = n$ single error patterns. ${}^nC_2$ double error patterns, ${}^nC_3$ tripple error patterns and so on. Therefore to correct 't' errors per word the following relation should be satisfied,

$$2^q - 1 \geq {}^nC_1 + {}^nC_2 + {}^nC_3 + \dots + {}^nC_t \qquad \dots (3.2.33)$$

3.2.5 Hamming Bound

We can write equation 3.2.33 as,

$$2^q \geq 1 + {}^nC_1 + {}^nC_2 + \dots + {}^nC_t$$

$$\geq \sum_{i=0}^{t} {}^nC_i$$

We know that $q = n - k$. Then the above equation becomes,

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

By taking logarithm to base 2 on both sides we get,

$$n - k \geq \log_2 \sum_{i=0}^{t} {}^nC_i$$

Dividing both sides by n we get,

$$1 - \frac{k}{n} \geq \frac{1}{n} \log_2 \sum_{i=0}^{t} {}^nC_i$$

Since code rate $r = \frac{k}{n}$ the above equation will be,

$$1 - r \geq \frac{1}{n} \log_2 \sum_{i=0}^{t} {}^nC_i \qquad \dots (3.2.34)$$

This equation relates code rate 'r' with the error correction capability of 't' errors per code vector in a block of 'n' bits. We know that the error correcting capability of the code (i.e. 't' errors per code vector) is related to the minimum distance. This minimum distance is also called hamming distance. Equation (3.2.34) gives the relation between code rate, number of errors to be corrected and number of bits in a block. This equation is also called hamming bound.

➡ **Example 3.2.4 :** *For a linear block code, prove with examples that :*

i) The syndrome depends only on error pattern and not on transmitted codeword.

ii) All error patterns that differ by a codeword have the same syndrome

Solution : (i) Syndrome depends only on error pattern.

Equation (3.2.30) gives the relationship between error pattern and syndrome i.e.,

$$S = EH^T$$

Above equation shows that syndrome (S) depends only on the error pattern (E). It doesnot depend on codeword (X).

In example 3.2.3 we have obtained the error patterns and corresponding syndromes for a particular code. Table 3.2.4 lists these error patterns. It is clear from this table that the syndrome depends only on error pattern and not on the codeword.

(ii) All error patterns that differ by a codeword have the same syndrome

Let us consider the two code vectors X_1 and X_2. Let an error be introduced in the first bit (MSB). Then the error pattern for both of these code vectors will be same. i.e.,

$$E = (1\,0\,0\,0\quad 0\,0\,0\,0) \quad \text{For 8 bits codevector}$$

Then the syndrome for first received code word will be,

$$\begin{aligned} S_1 &= Y_1 H^T \\ &= (X_1 \oplus E) H^T && \text{Here } Y_1 = X_1 \oplus E \\ &= X_1 H^T \oplus E H^T \\ &= E H^T && \text{Since } X_1 H^T = 0 \end{aligned}$$

Similarly syndrome for second received codeword will be,

$$\begin{aligned} S_2 &= Y_2 H^T \\ &= (X_2 \oplus E) H^T && \text{Here } Y_2 = X_2 \oplus E \\ &= X_2 H^T \oplus E H^T \\ &= E H^T && \text{Since } X_2 H^T = 0 \end{aligned}$$

Here observe that, $S_1 = S_2 = EH^T$. This shows that the error pattern differs by the codeword have the same syndrome. This confirms that syndrome is independent of the codeword.

Example :

In example 3.2.2, the parity check matrix is,

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

For this parity check matrix, codewords are given in table 3.2.2. Consider the two codewords,

$$X_2 = 0001 \quad 011$$

and $$X_3 = 0010 \quad 101$$

Let an error be introduced in first (MSB) bit of above codewords. Then we get,

$$Y_2 = \textcircled{1}\ 001 \ 011$$
$$Y_3 = \textcircled{1}\ 010 \ 101$$

Here encircled bit is in error. Let us calculate syndrome for Y_2. i.e.,

$$S_2 = Y_2 H^T = [1\,0\,0\,1\,0\,1\,1] \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

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

Similarly let us calculate the syndrome for Y_3. i.e.,

$$S_3 = Y_3 H^T = [1\,0\,1\,0\,1\,0\,1] \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

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

Thus the syndrome $S_2 = S_3 = [111]$ even if two codewords are different. This proves that for a particular error pattern syndrome is same even if codewords are different.

3.2.6 Syndrome Decoder for (n, k) Block Code

Fig. 3.2.3 shows the block diagram of a syndrome decoder for linear block code to correct errors. The received n-bit vector 'Y' is stored in an n-bit register. From this vector a syndrome is calculated using,

$$S = YH^T$$

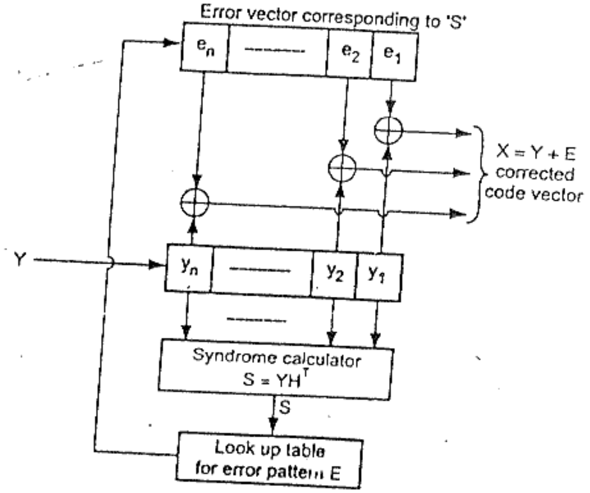


Fig. 3.2.3 Syndrome decoder for linear block code

Thus H^T is stored in the syndrome calculator. The q-bit syndrome vector is then applied to a look up table of error patterns. Depending upon the particular syndrome an error pattern is selected. This error pattern is added (mod - 2 addition) to the vector Y. The output is thus,

$$Y \oplus E = X \qquad \text{from equation (3.2.29)}$$

The block diagram shown above can correct only single errors in the above vectors.

Maximum likelyhood decoding for linear block codes :

We know that there are 2^q different syndromes. These syndromes can only represent $2^q - 1$ error patterns. But in an n-bit vector, there can be 2^n error patterns. Hence syndrome doesnot uniquely represent error vector (E). With the help of syndrome we can correct only $2^q - 1$ error patterns and remaining patterns are uncorrectable. Single errors are more common than double and higher errors. Therefore single error patterns are *most likely* compared to double and higher error patterns. Therefore syndrome decoding corrects single errors which are most likely. Hence syndrome decoding is called *maximum likelyhood decoding*. The maximum likelyhood decoding selects the code vector that has the smallest hamming distance from received vector. Such code vector is obtained by

$X = Y + E$, Here Y is received vector

and 'E' is the most likely error pattern. This error pattern is selected based on calculated syndrome. The maximum likelyhood decoding minimizes the word error probability.

3.2.7 Other Linear Block Codes

3.2.7.1 Single Parity Check Bit Code

If there are $m_1, m_2, m_3, \ldots, m_k$ are the bits of the k-bit message word; then,

$$m_1 \oplus m_2 \oplus m_3 \oplus \ldots \oplus m_k \oplus C_1 = 0$$

In the above equation C_1 is the parity check bit added to the message bit. The above equation shows that if there are even number of 1s in the message word, then parity check bit $C_1 = 0$. If there are odd number of 1s in the message word, then parity check bit $C_1 = 1$. Thus for this code,

$$\left.\begin{aligned} n &= k+1 \\ \text{and} \quad q &= 1 \end{aligned}\right\} \qquad \ldots (3.2.35)$$

Note that this code only detects single error but does not correct it.

3.2.7.2 Repeated Codes

3-111

In this code, a single message bit is transmitted and $q = 2t$ bit are parity bits. [illegible] $t \geq 1$. Then the transmitted bits are,

$$n = 2t + 1 \qquad \ldots (3.2.36)$$

This code is called repeated code since many redundant check bits are transmitted along with a single message bit. This code can correct 't' errors per block. Since this code uses many redundant check bits, it requires a larger bandwidth.

3.2.7.3 Hadamard Code

The hadamard code is derived from hadamard matrix. The hadamard matrix is the $n \times n$ square matrix. Rows of this hadamard matrix represent code vectors. Thus a $n \times n$ hadamard matrix, represents 'n' code vectors of 'n' bits each. If the block of message vector contains 'k' bits, then

$$n = 2^k \qquad (3.2.37)$$

This equation shows the relationship between number of bits in the code vector and number of bits in the message vector. We know that number of check bits (q) in (n,k) block code are

$$q = n - k$$

$$\therefore \quad q = 2^k - k \qquad (3.2.38)$$

This equation shows that as number of bits in the message block (k) increases, the parity bits become very large. This reduces the code rate. The code rate is given as,

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

$$= \frac{k}{2^k} \qquad \ldots (3.2.39)$$

This shows that with increase in 'k', the code rate becomes very small.

Hadamard Matrix and Code Words :

There are some following important points as follows :

i) One code vector represented by hadamard matrix contains all zero elements. That is one row of hadamard matrix contains all zero elements.

ii) The other code vectors contain $\frac{n}{2}$ 1's and $\frac{n}{2}$ 0's. That is other rows of hadamard matrix contains half number of 1's and half number of 0's.

iii) Every code vector differs from other code vectors at $\frac{n}{2}$ places. This means every row of hadamard matrix differs with other rows at $\frac{n}{2}$ places (i.e. half number of places). Consider the hadamard matrix with single message bit i.e. $k = 1$. Hence,

$$n = 2^k = 2^1 = 2$$

Thus hadamard matrix for single message bit will be of $n \times n$ (i.e. 2×2) size. The first row will be all zero elements. This matrix is shown below.

$$H_2 = \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}}_{\uparrow} \leftarrow \text{All zero row} \qquad \text{... (3.2.40)}$$

These elements satisfy the points (ii) and (iii) mentioned above.

Here H_2 is 2×2 size hadamard matrix. Observe that the second row contains half number of elements as zero and half as 1's. We know that the code words are the row of hadamard matrix. Here the code words are 00 and 01. Consider the hadamard matrix for two message bits (i.e. $k = 2$). Then we have,

$$n = 2^k = 2^2 = 4$$

Thus the hadamard matrix will be of size 4×4. This matrix is shown below.

$$H_4 = \begin{bmatrix} H_2 & H_2 \\ H_2 & \overline{H}_2 \end{bmatrix} \qquad \text{... (3.2.41)}$$

Here H_2 is the matrix given in equation (3.2.40) above $\overline{H}_2$ is the complement of matrix H_2 will be,

$$\overline{H}_2 = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \qquad \text{... (3.2.42)}$$

Thus in the above matrix every element of H_2 is complemented. Then the matrix H_4 given by equation 3.2.41 above will be,

$$H_4 = \left[\begin{array}{cc} \overset{H_2}{\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}} & \overset{H_2}{\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}} \\ \underset{H_2}{\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}} & \underset{\overline{H}_2}{\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}} \end{array} \right] \qquad \text{... (3.2.43)}$$

The above matrix is of size 4×4 and it gives four code words. These code vectors are (0000), (0101), (0011) and (0110) observe that the above code vectors and hadamard matrix satisfies all the three points discussed earlier. Since every code word differs by $\frac{n}{2}$ places with the other code words, these code words are orthogonal to each other over complete 'n' bits. Equation (3.2.41) can be generalized further to higher number of bits as follows.

$$H_{2n} = \begin{bmatrix} H_n & H_n \\ H_n & \overline{H}_n \end{bmatrix} \qquad \text{... (3.2.44)}$$

Here $\overline{H}_n$ is the complement matrix of H_n.

Since every code word in hadamard matrix differs every other code word by $\frac{n}{2}$, the minimum distance between the code words will be,

$$d_{\min} = \frac{n}{2} \qquad \text{... (3.2.45)}$$

$$= \frac{2^k}{2}$$

$$= 2^{k-1} \qquad \text{... (3.2.46)}$$

We know that, the correct upto t errors per words,

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

Putting the value of $d_{\min} = 2^{k-1}$ in above equation,

$$2^{k-1} \geq 2t + 1$$

$$t \leq \frac{2^{k-1} - 1}{2} \qquad \text{... (3.2.47)}$$

Drawback : Since hadamard code uses many check bits, its code rate is very low. Hence to transmit the signal at higher rates, higher bandwidths are required.

3.2.7.4 Extended Codes

We know that every (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.

$$H_e = \left[\begin{array}{c|c} \boxed{\begin{matrix} \text{Parity check} \\ \\ \text{matrix } H \end{matrix}} & \begin{matrix} 0 \\ 0 \\ \vdots \end{matrix} \\ 1 \ 1 \ 1 \cdots & 1 \end{array} \right]_{(q+1) \times (n+1)} \qquad \text{... (3.2.48)}$$

The code turned by such parity check matrix is called extended code. Thus the code will be described as $(n+1, k)$ linear block code. The newly formed parity check matrix H_e will be of size $(q+1)$ by $(n+1)$. Consider the parity check matrix of (7, 4) hamming code of equation (3.2.18). It is reproduced below

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

The parity check matrix for extended code will be (using equation (3.2.48))

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

Newly added row and column.

With the above extended parity check matrix, the minimum distance for extended code will be,

$$d_{e(\min)} = d_{\min} + 1 \qquad \text{... (3.2.49)}$$

Therefore for extended hamming code $d_{e\,(min)} = 4$.

Advantage :

This code can detect more number of errors compared to normal (n, k) block code. But it does not have any advantage of error correction.

3.2.7.5 Dual Code

We know that for (n, k) block code,

Generator matrix, $$[G]_{k\times n} = \left[I_{k\times k} \middle| P_{k\times q}\right]_{k\times n} \qquad \text{... (3.2.50)}$$

Parity check matrix, $$[H]_{q\times n} = \left[P^T_{q\times k} \middle| I_{q\times q}\right]_{q\times n} \qquad \text{... (3.2.51)}$$

Here $q = n - k$. Now consider the matrix product HG^T. i.e.,

$$HG^T = \left[P^T_{q\times k} \middle| I_{q\times q}\right]_{q\times n} \begin{bmatrix} I_{k\times k} \\ P^T_{q\times k} \end{bmatrix}$$

$$= \left[P^T \oplus P^T\right]_{q\times k} = 0$$

i.e. $$\boxed{HG^T = 0} \qquad \text{... (3.2.52)}$$

To illustrate above property, consider the generator and parity check matrices of Ex.3.2.2. i.e.,

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

and

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

Then HG^T will be,

$$HG^T = \begin{bmatrix} 1 & 1 & 1 & 0 & : & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & : & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & : & 0 & 0 & 1 \end{bmatrix}_{3\times 7} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \cdot\cdot & \cdot\cdot & \cdot\cdot & \cdot\cdot \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}_{7\times 4}$$

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

Definition of dual code :

Consider an (n, k) block code. As illustrated above, this code satisfies,

$$HG^T = 0$$

Then the $(n, n-k)$ i.e. (n, q) block code is called dual code. Thus for every (n, k) block code, there exists a dual code of size (n, q).

We have defined the generator matrix of (n, k) block code in equation (3.2.50). Similarly for (n, q) block code generator matrix will be,

$$[G]_{q\times n} = \left[I_{q\times q} \middle| P_{q\times k}\right]_{q\times n} \qquad \text{... (3.2.53)}$$

Similarly from equation (3.2.52), we can write the parity check matrix for (n, q) block code as,

$$[H]_{k\times n} = \left[P^T_{k\times q} \middle| I_{k\times k}\right]_{k\times n} \qquad \text{... (3.2.54)}$$

We know that (n, q) code is dual code of (n, k) block code. Then parity check matrix (H) and generator matrix (G) given above are matrices of dual code.

Example 3.2.5 : *Consider an (n, k) linear block code with generator matrix G and parity check matrix H. The (n, n – k) code generated by H is called the dual code of (n, k) code. Show that the matrix G is the parity check matrix of dual code.*

Solution : (i) Consider (n, k) block code

For this code generator matrix and parity check matrix are defined as,

$$[G]_{k\times n} = \left[I_{k\times k} | P_{k\times q}\right]_{k\times n} \quad \text{... (3.2.55)}$$

$$[H]_{q\times n} = \left[P^T_{q\times k} | I_{q\times q}\right]_{q\times n} \quad \text{... (3.2.56)}$$

The generator matrix and parity check matrix satisfy following property,

$$\left[HG^T\right]_{q\times k} = \left[P^T_{q\times k} | I_{q\times q}\right]\begin{bmatrix} I_{k\times k} \\ P^T_{q\times k}\end{bmatrix}_{n\times k}$$

$$= \left[P^T \oplus P^T\right]_{q\times k} = 0 \quad \text{... (3.2.57)}$$

(ii) Consider (n, q) block code

We know that (n, q) code is dual code of (n, k) block code. The generator and parity check matrices of this (n, q) code can be written from equation (3.2.53) and equation (3.2.54) as,

$$[G_{dual}]_{q\times n} = \left[I_{q\times q} | P_{q\times k}\right]_{q\times n} \quad \text{... (3.2.58)}$$

$$[H_{dual}]_{k\times n} = \left[P^T_{k\times q} | I_{k\times k}\right]_{k\times n} \quad \text{... (3.2.59)}$$

Here we have written G_{dual} and H_{dual} so that they can be differentiated from G and H of (n, k) code.

Now let us check whether (n, q) code also satisfies the property of equation (3.2.52), i.e.,

$$\left[H_{dual}\, G^T_{dual}\right] = \left[P^T_{k\times q} | I_{k\times k}\right]_{k\times n}\begin{bmatrix} I_{q\times q} \\ P^T_{k\times q}\end{bmatrix}_{n\times q}$$

$$= \left[P^T \oplus P^T\right]_{k\times q} = 0$$

i.e. $\left[H_{dual}\, G^T_{dual}\right]_{k\times q} = [0]_{k\times q}$

Taking transpose of both the sides of above equation,

$$\left[H_{dual}\, G^T_{dual}\right]^T_{k\times q} = [0]^T_{k\times q}$$

Here let us use the property of matrix : $[AB]^T = B^T A^T$. i.e.,

$$\left[G^T_{dual}\right]^T H^T_{dual} = [0]_{q\times k}$$

Here $\left[G^T_{dual}\right]^T = G_{dual}$ and $[0]^T$ will be zero matrix. Hence above equation become,

$$\left[G_{dual}\, H^T_{dual}\right]_{q\times k} = [0]_{q\times k}$$

(iii) Conclusion of equation (3.2.57) and above equation

We derived two results,

For (n, k) code : $\left[H_{q\times n}\, G^T_{n\times k}\right]_{q\times k} = [0]_{q\times k}$

For (n, q) code : $\left[\underset{q\times n}{G_{dual}}\; \underset{n\times k}{H^T_{dual}}\right]_{q\times k} = [0]_{q\times k}$

From above equations we can conclude,

$$\left[G^T\right]_{n\times k} = \left[H^T_{dual}\right]_{n\times k}$$

Taking transpose of both the sides,

$$[G]_{k\times n} = [H_{dual}]_{k\times n} \quad \text{... (3.2.60)}$$

Above equation shows that generator matrix of (n, k) block code is the parity check matrix of its dual code (i.e. (n, q) code).

This is the required proof of the given statement.

Example 3.2.6 : *For a linear block code which corrects single error per code vector, prove that,*

$n \geq k = \log_2(n+1)$

And hence design a linear block code with a minimum distance of three and a message block size of eight bits.

Solution : i) Proof of the equation

For linear block code we know from equation (3.2.33) that,

$$2^{q-1} \geq {}^nC_1 + {}^nC_2 + \ldots + {}^nC_t$$

This equation gives the condition for correction of 't' errors per word. For correction of single error per code vector ($t = 1$) the above equation will have only first term on RHS; i.e.

$$2^{q-1} \geq {}^nC_1$$

Since $q = n - k$ and ${}^nC_1 = n$

$$2^{(n-k)} - 1 \geq n$$

$$\therefore \quad (n-k) \geq \log_2 (n+1)$$

$$\therefore \quad n \geq k + \log_2 (n+1) \qquad \ldots (3.2.61)$$

which is proved.

ii) To determine linear block code

We have to design a linear block code with $d_{\min} = 3$ and $k = 8$. We know that the code with $d_{\min} = 3$ is a single error correcting code. Hence we can use the relation given by equation (3.2.61) above.

i.e. for single error correction $(d_{\min} = 3)$

$$n \geq k + \log_2 (n+1)$$

Putting $k = 8$ as given,

$$n \geq 8 + \log_2 (n+1)$$

On solving we get $n = 12$. Thus the code will be (12, 8). Therefore $q = 12 - 8 = 4$. The parity check matrix will be of size $q \times n$, i.e. parity check matrix size will be 4×12 i.e.

$$[H]_{4 \times 12} = [P^T : I_4]_{4 \times 12}$$

The matrix P^T will be of size 4×8 and I_4 will be an identity matrix of size 4×4. Select the P^T matrix such that,

i) Its size should be 4×8. This is P submatrix will be of size 8×4.

ii) No row should be zero, and

iii) All rows should be distinct.

Such matrix is given below

$$P^T = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}_{4 \times 8}$$

Here note that you have the freedom of which combination should form a particular column. Therefore parity check matrix will be,

$$H = \left[\underbrace{\begin{matrix} 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \end{matrix}}_{P^T} \begin{matrix} : \\ : \\ : \\ : \end{matrix} \underbrace{\begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}}_{I_4} \right]_{4 \times 12}$$

From the above matrix generator matrix and code vectors can be obtained.

Example 3.2.7 : *For a systematic linear block code, the three parity check digits, C_4, C_5 and C_6 are given by,*

$$\left.\begin{aligned} C_4 &= d_1 \oplus d_2 \oplus d_3 \\ C_5 &= d_1 \oplus d_2 \\ C_6 &= d_1 \oplus d_3 \end{aligned}\right\} \qquad \ldots (3.2.62)$$

i) Construct generator matrix

ii) Construct code generated by this matrix

iii) Determine error correcting capability

iv) Prepare a suitable decoding table

v) Decode the received words 101100 and 000110.

Solution : **i) To obtain the generator matrix :**

We know that the check bits, message bits and parity matrix are related as,

$$[C_4\, C_5\, C_6]_{1 \times 3} = [d_1\ d_2\ d_3]_{1 \times 3}\ [P]_{3 \times 3} \qquad \ldots (3.2.63)$$

The above equation can be written as

$$[C_4\, C_5\, C_6] = [d_1\ d_2\ d_3] \begin{bmatrix} P_{11} & P_{12} & P_{13} \\ P_{21} & P_{22} & P_{23} \\ P_{31} & P_{32} & P_{33} \end{bmatrix}$$

Hence,

$$\left.\begin{aligned} C_4 &= d_1\, P_{11} \oplus d_2\ P_{21} \oplus d_3\ P_{31} \\ C_5 &= d_1\, P_{12} \oplus d_2\ P_{22} \oplus d_3\ P_{32} \\ C_6 &= d_1\, P_{13} \oplus d_2\ P_{23} \oplus d_3\ P_{33} \end{aligned}\right\} \qquad \ldots (3.2.64)$$

Comparing the above equations with equation (3.2.62) we get parity matrix as,

$$P = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \qquad \ldots (3.2.65)$$

The generator matrix is given as,

$$\begin{aligned} G &= [I_k \ : \ P_{k \times q}] \\ &= [I_3 \ : \ P_{3 \times 3}] \\ &= \begin{bmatrix} 1 & 0 & 0 & : & 1 & 1 & 1 \\ 0 & 1 & 0 & : & 1 & 1 & 0 \\ 0 & 0 & 1 & : & 1 & 0 & 1 \end{bmatrix} \end{aligned} \qquad \ldots (3.2.66)$$

ii) To obtain the code vectors :

In this code, there are 3 message bits and 3 check bits. Hence this is (6, 3) block code. Table 3.2.5 shows the message bits, check bits and code vectors for this code.

Sr. No.	Message vector M			Check bits as per equation 3.2.51 C			Code vector or code word X						Weight of code vector w(X)
	d_1	d_2	d_3	C_4	C_5	C_6	d_1	d_2	d_3	C_4	C_5	C_6	
1	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	1	1	0	1	0	0	1	1	0	1	3
3	0	1	0	1	1	0	0	1	0	1	1	0	3
4	0	1	1	0	1	1	0	1	1	0	1	1	4
5	1	0	0	1	1	1	1	0	0	1	1	1	4
6	1	0	1	0	1	0	1	0	1	0	1	0	3.
7	1	1	0	0	0	1	1	1	0	0	0	1	3
8	1	1	1	1	0	0	1	1	1	1	0	0	4

Table 3.2.5 : Code vectors

iii) To obtain error correcting capability :

The minimum distance between the code vector is,

$$d_{\min} = [w(X)]_{\min} \; ; \qquad X \neq (00 \ldots\ldots 0)$$

From table 3.2.5, it is clear that

$$d_{\min} = 3$$

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

$$\therefore \quad 3 \geq s+1$$

$$\therefore \quad s \leq 2$$

Thus two errors will be detected

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

$$\therefore \quad 3 \geq 2t+1$$

$$\therefore \quad t \leq 1$$

Thus one error will be corrected.

iv) To prepare the decoding table :

The parity check matrix (H) is given as,

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

Hence transpose of above matrix becomes,

$$H^T = \begin{bmatrix} P \\ \ldots \\ I_q \end{bmatrix}_{n \times q}$$

From equation 3.2.65, above matrix will be,

$$H^T = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \ldots (3.2.67)$$

The syndrome vector (S) can be calculated from error vector (E) and H^T by equation (3.2.30) as,

$$S = EH^T$$

Here E is the 1×6 size error vector. Let us calculate syndrome for 2^{nd} bit in error. The E will be,

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

Hence syndrome will be (from $S = EH^T$),

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

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

Thus the above syndrome vector corresponds to 2^{nd} row of H^T. Similarly other syndromes can be obtained directly from rows of H^T. Table 3.2.6 shows the error patterns and corresponding syndrome vectors.

Sr. No.	Error vector 'E' showing single bit error patterns						Syndrome vector 'S'			Comments
1	0	0	0	0	0	0	0	0	0	
2	1	0	0	0	0	0	1	1	1	← First row of H^T
3	0	1	0	0	0	0	1	1	0	← Second row of H^T
4	0	0	1	0	0	0	1	0	1	← Third row of H^T

5	0	0	0	1	0	0	1	0	0	← Fourth row of H^T
6	0	0	0	0	1	0	0	1	0	← Fifth row of H^T
7	0	0	0	0	0	1	0	0	1	← Sixth row of H^T

Table 3.2.6 : Decoding table

v) To decode received words

To decode 101100 :

Here observe that the received word 101100 is not standard codevector from Table3.2.5. Hence there is an error in received word. Let,

$$Y = [1\ 0\ 1\ 1\ 0\ 0]$$

From equation (3.2.26), the syndrome can be calculated for this word. i.e.,

$$S = YH^T$$

Putting the values,

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

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

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

Note that [1 1 0] is second syndrome in Table 3.2.6, and the corresponding error pattern is,

$$E = [0\ 1\ 0\ 0\ 0\ 0]$$

The correct word is obtained as,

$$X = Y\oplus E$$

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

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

This is the correct word.

To decode 0 0 0 1 1 0

This also contains an error since it is not valid codeword from table 3.2.5. Let,

$$Y = [0\ 0\ 0\ 1\ 1\ 0]$$

Hence syndrome can be obtained as

$$S = YH^T$$

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

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

From Table 3.2.6, the corresponding error pattern is,

$$E = [0\ 1\ 0\ 0\ 0\ 0]$$

The correct codeword is given as,

$$X = Y\oplus E$$

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

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

This is the correct word.

Example 3.2.8 : *An error control code has the following parity check matrix :*

$$H = \begin{bmatrix}1 & 0 & 1 & 1 & 0 & 0\\1 & 1 & 0 & 0 & 1 & 0\\0 & 1 & 1 & 0 & 0 & 1\end{bmatrix} \qquad \ldots (3.2.68)$$

i) Determine the generator matrix G.

ii) Find the codeword that begins with 101 ....

iii) Decode the received codeword 110110. Comment on error detection and correction capability of this code.

iv) What is the relation between G and H ? Verify the same.

Solution : This is (6, 3) code. Hence n = 6, k = 3 and q = 3.

i) To obtain the generator matrix :

The parity check matrix is given as equation (3.2.11)

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

Hence equation 3.2.68 can be written as,

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

$$\underbrace{}_{P^T_{3\times 3}}\qquad\underbrace{}_{I_{3\times 3}}$$

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

Hence the matrix P will be,

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

The generator matrix is given as (equation 3.2.6)

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

$$\therefore \quad G = \begin{bmatrix} 1 & 0 & 0 & : & 1 & 1 & 0 \\ 0 & 1 & 0 & : & 0 & 1 & 1 \\ 0 & 0 & 1 & : & 1 & 0 & 1 \end{bmatrix} \qquad \text{... (3.2.69)}$$

This is the required generator matrix.

ii) To obtain the code word that begins with 101

Here the codeword begins with 101. This means first three bits of the codeword are 101. Length of the message bits is $k=3$. In systematic code, first 'k' bits of codeword are message bits. Hence first '3' bits in every codeword will be message bits. Thus 101 are message bits. i.e.,

$$M = [1\ 0\ 1]$$

This is (6, 3) code. The three check bits can be obtained by the equation,

$$C = MP$$

Putting appropriate matrices,

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

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

Hence the code vector is,

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

Thus the codeword that begins with 1 0 1 .... is $X = 1\ 0\ 1\ 0\ 1\ 1$.

iii) To decode 1 1 0 1 1 0 :

Let the received codeword be,

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

Then the syndrome is given as,

$$S = YH^T$$

Putting the matrices in above equation,

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

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

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

This is the second row of H^T. Hence there is an error in an error in the second bit. Hence the codeword is,

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

Here note that the second bit is made 0 to remove an error.

Error correction capability

It can be verified for this code that $d_{\min} = 3$. And we have seen in the earlier examples that such codes can detect upto two errors and correct single error. This is supported by following equations :

$$d_{\min} \geq s+1 \quad \text{for} \quad d_{\min} = 3, \quad s \leq 2$$

and

$$d_{\min} \geq 2t+1 \quad \text{for} \quad d_{\min} = 3, \quad t \leq 1$$

Here s is the number of errors detected and t is the number of errors corrected.

➡ **Example 3.2.9 :** *The generator matrix of a (6, 3) systematic block code is given by*

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

i. Find the code vectors

ii. Find the parity check matrix

iii. Find the error syndrome

Solution : i) To obtain code vectors :

Refer to example (3.2.1). The code vectors are obtained in this example.

ii) To obtain parity check matrix (H) :

The P submatrix is obtained in example (3.2.1) as,

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

Hence the transpose of this matrix will be,

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

The parity check matrix is given by equation (3.2.11) as,

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

Here I is the $q \times q$ identity matrix. In this example n = 6, k = 3 and q = 3. Hence above equation becomes,

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

iii) To find error syndrome :

We obtained parity check matrix just now. The transpose of the parity check matrix will be,

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

The syndrome is q = 3 bits in length. The error pattern vector will consists of n = 6 bits. The syndrome vector is given by equation (3.2.30) as,

$$S = EH^T$$

Let there be error in the first bit. Hence error pattern will be, E = [1 0 0 0 0 0]. Putting values in above equation,

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

$$[0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \quad 1 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \quad 1 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0]$$

$$= 0\;1\;1$$

Thus the syndrome for error in first bit corresponds to first row of H^T. Similarly it can be shown for other error patterns. Table 3.2.7 shows all the single bit error patterns and their corresponding error syndromes.

Sr. No.	Error vector E showing single bit error patterns						Syndrome S			Comments
1	0	0	0	0	0	0	0	0	0	
	1	0	0	0	0	0	0	1	1	← 1st row of H^T
	0	1	0	0	0	0	1	0	1	← 2nd row of H^T
	0	0	1	0	0	0	1	1	0	← 3rd row of H^T
	0	0	0	1	0	0	1	0	0	← 4th row of H^T
	0	0	0	0	1	0	0	1	0	← 5th row of H^T
	0	0	0	0	0	1	0	0	1	← 6th row of H^T

Table 3.2.7 Error Syndromes

iv) Relation between G and H

The relation between G and H is given by equation (3.2.66) i.e.,

$$HG^T = 0$$

Taking transpose of both the sides

$$(HG^T)^T = (0)^T$$

We know that $(AB)^T = B^T A^T$. Hence above equation becomes,

$$(G^T)^T H^T = (0)^T$$

Here $(0)^T = 0$. Therefore above equation will be,

$$GH^T = 0$$

Thus the relation between G and H is,

$$HG^T = GH^T = 0$$

Example 3.2.10 : *The parity check bits of a (8,4) block code are generated by,*

$$\begin{aligned} C_5 &= d_1 + d_2 + d_4 \\ C_6 &= d_1 + d_2 + d_3 \\ C_7 &= d_1 + d_3 + d_4 \\ C_8 &= d_2 + d_3 + d_4 \end{aligned}$$

Where d_1, d_2, d_3 and d_4 are the message bits.

i) Find the generator matrix and the parity check matrix for this code.

ii) List all code vectors

iii) Find the errors detecting and correcting capabilities of this code.

iv) Show through an example that this code detects upto 3 errors.

Solution : i) To obtain the generator matrix and parity check matrix

We know that the check bits, message bits and parity matrix are related as,

$$\begin{aligned} [C_5\,C_6\,C_7\,C_8]_{1\times4} &= [d_1\,d_2\,d_3\,d_4]_{1\times4}\,[P]_{4\times4} \\ &= [d_1\,d_2\,d_3\,d_4]\begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \\ P_{41} & P_{42} & P_{43} & P_{44} \end{bmatrix} \end{aligned}$$

Hence,

$$\begin{aligned} C_5 &= P_{11}\,d_1 \oplus P_{21}\,d_2 \oplus P_{31}\,d_3 \oplus P_{41}\,d_4 \\ C_6 &= P_{12}\,d_1 \oplus P_{22}\,d_2 \oplus P_{32}\,d_3 \oplus P_{42}\,d_4 \\ C_7 &= P_{13}\,d_1 \oplus P_{23}\,d_2 \oplus P_{33}\,d_3 \oplus P_{43}\,d_4 \\ C_8 &= P_{14}\,d_1 \oplus P_{24}\,d_2 \oplus P_{34}\,d_3 \oplus P_{44}\,d_4 \end{aligned}$$

Comparing above equations with the given check bit equations we find that,

$$\begin{array}{llll} P_{11} = 1 & P_{21} = 1 & P_{31} = 0 & P_{41} = 1 \\ P_{12} = 1 & P_{22} = 1 & P_{32} = 1 & P_{42} = 0 \\ P_{13} = 1 & P_{23} = 0 & P_{33} = 1 & P_{43} = 1 \\ P_{14} = 0 & P_{24} = 1 & P_{34} = 1 & P_{44} = 1 \end{array}$$

Hence P matrix can be formed as,

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

Generator matrix is given as,

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

$$\therefore \quad G = \begin{bmatrix} 1 & 0 & 0 & 0 & : & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & : & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & : & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & : & 1 & 0 & 1 & 1 \end{bmatrix}_{8\times4}$$

The parity check matrix is given as,

$$\begin{aligned} H &= [P^T : I_q] \\ &= \begin{bmatrix} 1 & 1 & 0 & 1 & : & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & : & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & : & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & : & 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

ii) To Obtain all Code Vectors

All the code vectors in systematic form can be obtained from message bits and check bits. The check bits are given as,

$$\begin{aligned} C_5 &= d_1 \oplus d_2 \oplus d_4 \\ C_6 &= d_1 \oplus d_2 \oplus d_3 \\ C_7 &= d_1 \oplus d_3 \oplus d_4 \\ C_8 &= d_2 \oplus d_3 \oplus d_4 \end{aligned}$$

Then the code vector can be expressed as,

$$X = (d_1\,d_2\,d_3\,d_4\,c_5\,c_6\,c_7\,c_8)$$

Table 3.2.8 lists all the messages, check bits, codevectors and weights.

Sr. no.	Message vector				Check bits				Code vector X								Weight of code
	d_1	d_2	d_3	d_4	C_5	C_6	C_7	C_8	d_1	d_2	d_3	d_4	C_5	C_6	C_7	C_8	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	1	0	1	1	0	0	0	1	1	0	1	1	4
2	0	0	1	0	0	1	1	1	0	0	1	0	0	1	1	1	4
3	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	4
4	0	1	0	0	1	1	0	1	0	1	0	0	1	1	0	1	4
5	0	1	0	1	0	1	1	0	0	1	0	1	0	1	1	0	4
6	0	1	1	0	1	0	1	0	0	1	1	0	1	0	1	0	4
7	0	1	1	1	0	0	0	1	0	1	1	1	0	0	0	1	4
8	1	0	0	0	1	1	1	0	1	0	0	0	1	1	1	0	4
9	1	0	0	1	0	1	0	1	1	0	0	1	0	1	0	1	4
10	1	0	1	0	1	0	0	1	1	0	1	0	1	0	0	1	4
11	1	0	1	1	0	0	1	0	1	0	1	1	0	0	1	0	4
12	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	4
13	1	1	0	1	1	0	0	0	1	1	0	1	1	0	0	0	4
14	1	1	1	0	0	1	0	0	1	1	1	0	0	1	0	0	4
15	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	8

Table 3.2.8 Codevectors of Ex. 3.2.9

iii) Error detecting and correcting capabilities

From table 3.2.8 it is clear that minimum weight of the code is 4. Hence minimum distance is $d_{\min} = 4$. Hence 's' errors are detected if,

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

$$4 \geq s + 1 \quad \text{or} \quad s \leq 3$$

Thus three errors can be detected.

Similarly 't' error will be corrected if,

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

i.e.,
$$4 \geq 2t + 1 \quad \text{i.e.,} \quad t \leq \frac{3}{2}$$

Thus one error can be corrected by this code.

(v) To show that this code detects '3' errors

Consider the codevector X = 0 0 0 1 1 0 1 1 from table 3.2.8. Let three errors be introduced in this codevector as follows :

$$x = \underbrace{1\ 0\ 1\ 1\ 1\ 0\ 0\ 1}_{\text{These bits are in error}}$$

Let us calculate the syndrome for this received vector. i.e,

$$S = YH^T$$

$$= [10111001]\begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 \oplus 0 \oplus 0 \oplus 1 \oplus 1 \oplus 0 \oplus 0 \oplus 0 \\ 1 \oplus 0 \oplus 1 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \\ 1 \oplus 0 \oplus 1 \oplus 1 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \\ 0 \oplus 0 \oplus 1 \oplus 1 \oplus 0 \oplus 0 \oplus 0 \oplus 1 \end{bmatrix}$$

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

The syndrome is nonzero. This means the code detects up to three errors.

Example 3.2.11 : *A generator matrix of (6, 3) linear block code is given as,*

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

Determine the '$d_{\min}$' for the above code. Comment on error correction and detection capabilities. If the received sequence is 1 0 1 1 0 1, determine the message bit sequence.

Solution : For this code n = 6 and k = 3

$$\therefore \quad q = 6 - 3 = 3$$

(i) To obtain '$d_{\min}$' for this code

To determine '$d_{\min}$' we have to findout all the codewords.

To obtain P submatrix

We know that,

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

$$= [I_{3\times3} : P_{3\times3}]$$

$$= \begin{bmatrix} 1 & 0 & 0 & : & P_{11} & P_{12} & P_{13} \\ 0 & 1 & 0 & : & P_{21} & P_{22} & P_{23} \\ 0 & 0 & 1 & : & P_{31} & P_{32} & P_{33} \end{bmatrix}$$

Comparing above matrix with that given in the problem,

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

To obtain equations for check bits

We know that the check bits are given as,

$$C = MP$$

i.e.

$$[C_1 \, C_2 \, C_3]_{1\times3} = [m_1 \, m_2 \, m_3]_{1\times3} \, [P]_{3\times3}$$

$$= [m_1 \, m_2 \, m_3] \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

$$= [m_1 \oplus m_2 \quad m_1 \oplus m_2 \oplus m_3 \quad m_1 \oplus m_3 \,]$$

Thus the equations for check bits are,

$$C_1 = m_1 \oplus m_2$$

$$C_2 = m_1 \oplus m_2 \oplus m_3$$

$$C_3 = m_1 \oplus m_3$$

To determine the codevectors

Since there are three message bits, there will be nine message vectors. Hence there will be nine codevectors. Table 3.2.9 lists the codevectors of this table. The check bits are calculated as per above equations.

Sr. No.	Message vector M			Check bits C			Code vector X						Wt. of the code W(X)
	m_1	m_2	m_3	c_1	c_2	c_3	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	0
2	0	0	1	0	1	1	0	0	1	0	1	1	3
3	0	1	0	1	1	0	0	1	0	1	1	0	3
4	0	1	1	1	0	1	0	1	1	1	0	1	4
5	1	0	0	1	1	1	1	0	0	1	1	1	4
6	1	0	1	1	0	0	1	0	1	1	0	0	3
7	1	1	0	0	0	1	1	1	0	0	0	1	3
8	1	1	1	0	1	0	1	1	1	0	1	0	4

Table 3.2.9 : Calculations of Ex. 3.2.11

As shown in above table, consider the message vector of $m_1 m_2 m_3 = 001$. Then check bits are calculated as,

$$c_1 = m_1 \oplus m_2 = 0 \oplus 0 = 0$$

$$c_2 = m_1 \oplus m_2 \oplus m_3 = 0 \oplus 0 \oplus 1 = 1$$

$$c_3 = m_1 \oplus m_3 = 0 \oplus 1 = 1$$

Hence $c_1 c_2 c_3 = 011$, and the code vector will be,

$$X = (m_1 m_2 m_3 \; c_1 c_2 c_3) = 0\,0\,1\,0\,1\,1$$

Weight of the code and $d_{\min}$

As shown in table 3.2.9, the minimum weight of the code is 3. Hence,

Since

$$d_{\min} = 3$$

$$d_{\min} = [w(X)]_{\min} = 3$$

ii) Error correction and detection capabilities

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

$$3 \geq s + 1$$

$$\therefore \quad s \leq 2$$

Thus two errors will be detected.

and

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

$$3 \geq 2t + 1$$

$$\therefore \quad t \leq 1$$

Thus one error will be corrected.

This is hamming code ($d_{min} = 3$) and it always detects double errors and corrects single errors.

(iii) To obtain message bits, if Y = 1 0 1 1 0 1

We have to determine whether the received vector is a valid codevector. This can be done by calculating syndrome.

To obtain syndrome (S)

Syndrome vector is given as,

$$S = YH^T$$

We know that $H^T = \begin{bmatrix} P \\ \dots\dots \\ I_q \end{bmatrix}_{n \times q}$

Putting the value of P submatrix in above equation,

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

Hence syndrome becomes,

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

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

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

Here the syndrome is nonzero. Hence there is an error in the received codevector.

To locate the error

On comparing syndrome S = 0 0 1 with the rows of H^T, we find that 6^{th} row matches with syndrome. Hence 6^{th} bit is in error. Error vector can be written as,

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

To correct the error

Correct vector can be obtained by

$$X = Y \oplus E$$
$$= (1\ 0\ 1\ 1\ 0\ 1) \oplus (0\ 0\ 0\ 0\ 0\ 1)$$
$$= 1 \oplus 0\ \ 0 \oplus 0\ \ 1 \oplus 0\ \ 1 \oplus 0\ \ 0 \oplus 0\ \ 1 \oplus 1$$
$$= 1\,0\,1\,1\,0\,0$$

This is transmitted vector.

We know that,

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

and $X = (1\ 0\ 1\ 1\ 0\ 0)$ as calculated above.

On comparing above two equations, the message bits are,

$$m_1 m_2 m_3 = 1\,0\,1$$

Example 3.2.12 : *The parity digits of a (6, 3) linear block code are given as,*

$$c_4 = m_1 \oplus m_2, \quad c_5 = m_1 \oplus m_2 \oplus m_3 \quad and \quad c_6 = m_1 \oplus m_3$$

i) Determine the generator and parity check matrices for the systematic code.

ii) Comment on error detection and error correction, capabilities of the code.

iii) If the received sequence is 1 0 1 1 0 1, determine the message word.

Solution : (i) To obtain the generator matrix (G) and parity check matrix (H).

We know that the check bits, the P-sub matrix and message bits are related as,

$$[c_4\ c_5\ c_6]_{1 \times 3} = [m_1\ m_2\ m_3][P]_{3 \times 3}$$

The above equation can be written as,

$$[c_4\ c_5\ c_6] = [m_1\ m_2\ m_3]\begin{bmatrix} P_{11} & P_{12} & P_{13} \\ P_{21} & P_{22} & P_{23} \\ P_{31} & P_{32} & P_{33} \end{bmatrix}$$

i.e.,

$$c_4 = m_1 P_{11} \oplus m_2 P_{21} \oplus m_3 P_{31}$$
$$c_5 = m_1 P_{12} \oplus m_2 P_{22} \oplus m_3 P_{32}$$
$$c_6 = m_1 P_{13} \oplus m_2 P_{23} \oplus m_3 P_{33}$$

Comparing above equations with the given equations of c_4, c_5 and c_6 we get values of P matrix.i.e.,

$P_{11} = 1$ $\quad P_{12} = 1$ $\quad P_{13} = 1$

$P_{21} = 1$ $\quad P_{22} = 1$ $\quad P_{23} = 0$

$P_{31} = 0$ $\quad P_{32} = 1$ $\quad P_{33} = 1$

$$\therefore \quad P = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}_{3\times 3}$$

The generator matrix is given as,

$$\begin{aligned} G &= [I_k : P_{k\times q}] \\ &= [I_3 : P_{3\times 3}] \\ &= \begin{bmatrix} 1 & 0 & 0 & : & 1 & 1 & 1 \\ 0 & 1 & 0 & : & 1 & 1 & 0 \\ 0 & 0 & 1 & : & 0 & 1 & 1 \end{bmatrix}_{3\times 6} \end{aligned}$$

And the parity check matrix is given as,

$$\begin{aligned} H &= [P^T : I_q]_{q\times n} \\ &= [P^T : I_3]_{3\times 6} \\ &= \begin{bmatrix} 1 & 1 & 0 & : & 1 & 0 & 0 \\ 1 & 1 & 1 & : & 0 & 1 & 0 \\ 1 & 0 & 1 & : & 0 & 0 & 1 \end{bmatrix}_{3\times 6} \end{aligned}$$

(ii) and (iii)

The generator matrix obtained in part (i) is same as that given in Ex. 3.2.11. Hence part (ii) and (iii) are same as that given in Ex. 3.2.11.

Review Questions

1. *What is the difference between systematic codes and non systematic codes.*
2. *What are the functions of parity check matrix and generator matrix in linear block codes ? How they are used to generate code vectors from message block?*
3. *What are Hamming codes ? What are their properties ?*
4. *How error correction and detection capabilities of block codes are related to minimum distance d_{min} ?*
5. *What is the use of syndromes ? Explain syndrome decoding.*
6. *What are the hadamard and extended block codes ?*

Unsolved Examples

1. *Consider a (6, 3) linear code whose generator matrix is*

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

i) Find all the codevectors

ii) Find all hamming weights and distances

iii) Find minimum weight parity check matrix. *iv) Draw the encoder circuit.*

2. *Consider a (7, 4) linear block code whose generator matrix is given below*

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

a) Find all code vectors of this code.

b) Find the party check matrix of this code.

c) Find minimum weight of this code.

3. *The parity check bits of a (8, 4) block code are given by,*

$$C_1 = m_1 + m_2 + m_4$$
$$C_2 = m_1 + m_2 + m_3$$
$$C_3 = m_1 + m_3 + m_4$$
$$C_4 = m_2 + m_3 + m_4$$

Here m_1, m_2, m_3 and m_4 are the 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.

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 :

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

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.,

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

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.,

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

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,

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

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.

BU

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)),

$$M(p) = m_3\,p^3 + m_2\,p^2 + m_1 p + m_0 \qquad \text{... (3.3.9)}$$

$$M(p) = p^2 + 1$$

And given generator polynomial is, ... (3.3.10)

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

To obtain non-systematic code vectors

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

$p^3 + p + 1$

$+ 0\ 1\ 1$

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

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

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

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

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.

0 0 0 1 0

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

To check whether cyclic property is satisfied :

Let us consider codevector X_g which is given in above table as,

$$X_g = (1011000)$$

Let us shift this codevector cyclically to left side by 1 bit position. Then we get,

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

From table, observe that

$$X' = X_8 = (0110001)$$

Thus cyclic shift of X_9 produces X_8. This can be varified for other codevectors also.

3.3.5 Generation of Codevectors in Systematic Form

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

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

$$= \left(m_{k-1}\ m_{k-2} \dots m_1\ m_0 : c_{q-1}\ c_{q-2} \dots c_1\ c_0\right) \qquad \text{... (3.3.12)}$$

Here the check bits form a polynomial as,

$$C(p) = c_{q-1}\,p^{q-1} + c_{q-2}\,p^{q-2} + \dots c_1\,p + c_0 \qquad \text{... (3.3.13)}$$

The check bit polynomial is obtained by,

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

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)),

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

$$M(p) = p^2 + 1 \qquad \cdots (3.3.15)$$

And given generator polynomial is,

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

To obtain $p^q M(p)$

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

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

and

$$\begin{aligned} G(p) &= p^3 + p + 1 \\ &= p^3 + 0p^2 + p + 1 \end{aligned}$$

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

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

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

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 $\quad c_2 p^2 + c_1 p + c_0 = p^2 + 0p + 0$

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.,

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

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

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

We have obtained nonsystematic codevectors for the same generating polynomial in Ex. 3.3.1. They are listed in table 3.3.1.

Example 3.3.3 : *An 'n' digit code polynomial $X(p)$ is obtained as,*

$$X(p) = C(p) + p^{(n-k)} M(p)$$

where $M(p)$ represents message polynomial for k digit data and $C(p)$ is remainder polynomial obtained by dividing $p^{(n-k)} M(p)$ by proper generator polynomial $G(p)$, in modulo-2 sense. Prove that $X(p)$ represents a systematic cyclic code if $G(p)$ is the factor of $p^n + 1$ in modulo-2 sense.

Solution : (I) To prove that $G(p)$ must be a factor of p^{n+1} :

Consider the codeword, $X = (x_{n-1}, x_{n-2}, \ldots\ldots x_1, x_0)$ polynomial of this codeword will be of degree less than or equal to $(n-1)$, and it can be expressed as,

$$X(p) = x_{n-1} p^{n-1} + x_{n-2} p^{n-2} + \ldots + x_1 p + x_0 \qquad \ldots (3.3.17)$$

Now let us shift the codevector 'X' cyclically to left side. We get,

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

The polynomial for this codevector can be written as,

$$X'(p) = x_{n-2} p^{n-1} + x_{n-3} p^{n-2} + \ldots + x_1 p^2 + x_0 p + x_{n-1} \qquad \ldots (3.3.18)$$

Multiplying the polynomial of equation 3.3.17 by p,

$$pX(p) = x_{n-1} p^n + x_{n-2} p^{n-1} + \ldots + x_1 p^2 + x_0 p$$

Let us add above equation and equation 3.3.18 as per mod-2 rules. We get,

$$pX(p) + X'(p) = x_{n-1} p^n + (x_{n-2} \oplus x_{n-2}) p^{n-1} + \ldots + (x_1 \oplus x_1) p^2 + (x_0 \oplus x_0) p + x_{n-1}$$

We know that in mod-2 addition, if both the bits are same, then result is zero. i.e. $x_{n-2} \oplus x_{n-2} = 0$, $x_1 \oplus x_1 = 0$ and so on. Then above equation becomes,

$$pX(p) + X'(p) = x_{n-1} p^n + x_{n-1}$$

i.e. $$pX(p) + X'(p) = x_{n-1} (p^n + 1)$$

We know that by mod-2 addition rules, there is no addition and subtraction. That is if $x \oplus y = z$ then $x = y \oplus z$ or $y = x \oplus z$. This is because mod-2 addition and subtraction is same operation. Applying this rule to above equation,

$$\boxed{X'(p) = pX(p) \oplus x_{n-1} (p^n + 1)} \qquad \ldots (3.3.19)$$

Thus new codevector polynomial $X'(p)$ is obtained with the help of $X(p)$ and $(p^n + 1)$. The generator polynomial $G(p)$ is of the degree $q = n - k$. It is expressed as,

$$G(p) = p^q + g_{q-1} p^{q-1} + \ldots\ldots + g_1 p + 1 \qquad \ldots (3.3.20)$$

Let $M(p)$ be the message vector polynomial of degree $(k-1)$. It is expressed as,

$$M(p) = m_{k-1} p^{k-1} + m_{k-2} p^{k-2} + \ldots + m_1 p + m_0 \qquad \ldots (3.3.21)$$

Then the product of generating polynomial and message polynomial gives codevector i.e.,

$$X(p) = M(p)\, G(p) \qquad \ldots (3.3.22)$$

Important conclusions :

1. Here note that $G(p)$ becomes a factor of $X(p)$.
2. Similarly $X'(p)$ of equation (3.3.19) can be generated with the help of $G(p)$ and some other message vector.
3. Under this condition $G(p)$ will be a factor of $X'(p)$ also.
4. Then in equation (3.3.19) observe that $G(p)$ is factor of $X(p)$ as well as $X'(p)$. Both $X(p)$ and $X'(p)$ are valid cyclic codevectors. For above statements to be true, $G(p)$ must be a factor of $(p^n + 1)$ also.

   If $G(p)$ is a factor of $(p^n + 1)$, then $X'(p)$ will be a polynomial of degree less than 'n' and it will satisfy cyclic shift property. If $G(p)$ is not a factor of $(p^n + 1)$, then $X'(p)$ will not be valid cyclic codevector.

(II) To prove that $X(p) = C(p) + p^{(n-k)} M(p)$:

The systematic form of a codevector is given as,

$$X = (\text{'k' message bits} \mid \text{'q' check bits})$$

Here $q = n - k$ are number of check bits. Above codevector can also be written as,

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

The above code vector can be written in polynomial form as,

$$X(p) = m_{k-1}p^{n-1} + m_{k-2}p^{n-2} + \ldots + m_1 p^{n-k+1} + m_0 p^{n-k}$$
$$+ c_{q-1}p^{n-k-1} + c_{q-2}p^{n-k-2} + \ldots + c_1 p + c_0 \qquad \ldots (3.3.23)$$

We know that $n-k=q$ or $n=k+q$. Putting those values of 'n' and '$n-k$' in above equation we get,

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

Let's rearrange the above equation as,

$$X(p) = p^q[m_{k-1}p^{k-1} + m_{k-2}p^{k-2} + \ldots + m_1 p + m_0] + c_{q-1}p^{q-1} + c_{q-2}p^{q-2} + \ldots + c_1 p + c_0 \qquad \ldots (3.3.24)$$

In the above equation $m_{k-1}p^{k-1} + m_{k-2}p^{k-2} + \ldots + m_1 p + m_0 = M(p)$. Therefore above equation becomes,

$$X(p) = p^q M(p) + c_{q-1}p^{q-1} + c_{q-2}p^{q-2} + \ldots + c_1 p + c_0 \qquad \ldots (3.3.25)$$

Let's define the check bit polynomial of check bits

$$C = (c_{q-1}c_{q-2} \ldots c_1 c_0) \text{ as}$$
$$C(p) = c_{q-1}p^{q-1} + c_{q-2}\,p^{q-2} + \ldots + c_1 p + c_0 \qquad \ldots (3.3.26)$$

The above equation is check bit polynomial of degree less than q. From above equation (3.3.25) we obtain,

$$X(p) = p^q M(p) + C(p) \qquad \ldots (3.3.27)$$

The above equation gives a code word polynomial in systematic form. For this code vector to be cyclic, then the above equation should be same as equation (3.3.22). Thus for a cyclic code vector we can equate above equation and equation (3.3.22) i.e.

$$p^q M(p) + C(p) = M(p)\ G(p)$$
$$\frac{p^q M(p)}{G(p)} \oplus \frac{C(p)}{G(p)} = M(p)$$

The above equation has the form of $z \oplus t = l$. We know that mod-2 addition and subtraction operation is same i.e. if $z \oplus t = l$ then we can write $z = t \oplus l$ or $t = z \oplus l$ or $z \oplus t \oplus l = 0$. Thus there is no mod–2 subtraction as such. Mod–2 addition and subtraction yields same result. With these conclusions we can write above equation as,

$$\frac{p^q M(p)}{G(p)} = M(p) \oplus \frac{C(p)}{G(p)} \qquad \ldots (3.3.28)$$

This equation has the form of

$$\frac{\textit{Numerator}}{\textit{Denominator}} = \textit{Quotient} \oplus \frac{\textit{Remainder}}{\textit{Denominator}} \left\{ \text{For example } \frac{14}{3} = 4 + \frac{2}{3} \right\}$$

Thus the check bit polynomial is obtained as remainder in dividing $p^q M(p)$ by $G(p)$ i.e.

$$\boxed{C(p) = rem\left[\frac{p^q\ M(p)}{G(p)}\right]} \qquad \ldots (3.3.29)$$

Here $C(p)$ is check bit polynomial for systematic code

$M(p)$ is message bit polynomial.

and $G(p)$ is generating polynomial, which is the factor of $p^n - 1$.

Thus equation (3.3.27) represents the cyclic code in systematic form i.e.,

$$X(p) = p^q M(p) + C(p) \qquad \ldots (3.3.30)$$

Here $q = n - k$, hence above equation becomes,

$$X(p) = C(p) + p^{(n-k)}M(p) \qquad \ldots (3.3.31)$$

Observe that this equation is same as the given equation. Here $C(p)$ is given by equation (3.3.29).

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,

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

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

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

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,

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

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

$$p^7+1 = (p \oplus 1)(p^3 \oplus p^2 \oplus 1)(p^3 \oplus p \oplus 1) \quad \ldots (3.3.37)$$

The valid generating polynomial is given by,

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

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

$$i = 3, 2, 1, 0 \qquad \text{since } k-1=3$$

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

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

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

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,

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

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

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

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.,

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

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 $= Q_t(p)$

Then equation (3.3.44) will be,

$$\frac{p^{n-t}}{G(p)} = Q_t(p) + \frac{R_t(p)}{G(p)} \qquad \text{... (3.3.45)}$$

i.e.,

$$p^{n-t} = Q_t(p)\,G(p) \oplus R_t(p) \quad \text{and} \quad t = 1, 2, \dots k \qquad \text{... (3.3.46)}$$

We know that if $z = y \oplus t$, then $z \oplus y = t$ or $z \oplus t = y$. That is mod-2 addition and subtraction yields same results. Then we can write above equation as,

$$p^{n-t} \oplus R_t(p) = Q_t(p)\,G(p) \qquad \text{... (3.3.47)}$$

As we have stated in equation (3.3.43) the above equation represents t^{th} row of systematic generator matrix. The above procedure is illustrated in next example.

3.3.6.3 Parity Check Matrix

Once the generator matrix in systematic form is obtained then parity check matrix can be obtained as per the procedure discussed in last section. Next example illustrate this.

➡ **Example 3.3.7 :** *Find out the generator matrix for a systematic (7, 4) cyclic code if $G(p) = p^3 + p + 1$. Also find out the parity check matrix.* (Nov./Dec.-2003, 4 Marks)

Solution : (I) To obtain generator polynomial

The t^{th} row of generator matrix is given by equation (3.3.47) as,

$$p^{n-t} + R_t(p) = Q_t(p)\,G(p) \quad \text{and} \quad t = 1, 2, \dots k$$

We are given that $n = 7$, $k = 4$ and $q = n - k = 3$.

The above equation will be,

$$p^{7-t} + R_t(p) = Q_t(p)\,(p^3 + p + 1) \quad \text{and} \quad t = 1, 2, 3, 4 \qquad \text{... (3.3.48)}$$

With $t = 1$, the above equation becomes,

$$p^6 + R_t(p) = Q_t(p)\,(p^3 + p + 1) \qquad \text{... (3.3.49)}$$

I) To obtain $R_t(p)$ and $Q_t(p)$ for 1st row

The RHS or LHS of this equation represents 1st row of systematic generator matrix. We have to find $Q_t(p)$. From equation (3.3.45) we know that $Q_t(p)$ is obtained by dividing p^{n-t} by $G(p)$. Here to obtain $Q_t(p)$ we have to divide p^6 by $G(p) = p^3 + p + 1$

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

Denotes mod-2 addition

Here $Q_l(p) = p^3+p+1$

and $R_l(p) = p^2+1$

Putting those values in equation (3.3.49) we get,

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

The RHS or LHS (actually both are same) of the above equation represents 1st row of generator matrix i.e.

$$1^{st} \text{ row polynomial } = p^6+p^2+1 \qquad \ldots (3.3.50)$$

ii) Other row polynomials

Using the same procedure as discussed above, other row polynomials are obtained and they are given below

$$\left.\begin{array}{l}
l=2 \Rightarrow 2^{nd} \text{ row polynomial} = p^5+p^2+p+1 \\
l=3 \Rightarrow 3^{rd} \text{ row polynomial} = p^4+p^2+p \\
l=4 \Rightarrow 4^{th} \text{ row polynomial} = p^3+p+1
\end{array}\right\} \qquad \ldots (3.3.51)$$

iii) Conversion of row polynomials into matrix

The above equation can be transformed into generator matrix as shown below

$$G = \begin{array}{l} \\ Row\,1 \\ Row\,2 \\ Row\,3 \\ Row\,4 \end{array} \begin{array}{c} \begin{array}{ccccccc} p^6 & p^5 & p^4 & p^3 & p^2 & p^1 & p^0 \end{array} \\ \underbrace{\left[\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right.}_{I_3} \begin{array}{c} : \\ : \\ : \\ : \end{array} \underbrace{\left.\begin{array}{ccc} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{array}\right]_{4\times 7}}_{P_{4\times 3}} \end{array} \qquad \ldots (3.3.52)$$

This is the required generator matrix in systematic form. The code vector can be obtained from equation (3.3.4) as

$$X = MG$$

Let's take any 4 bit message vector and find corresponding code vector. Let's take

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

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

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

This code vector is obtained by performing matrix multiplication and mod-2 additions. Observe that the same systematic code vector is listed in table 3.3.2.

Using the same procedure other code vectors can be obtained.

II) To obtain parity check matrix (H)

We know that $G = [I_k : P_{k\times q}]_{k\times n}$ from equation (3.3.6)

The P submatrix can be obtained from equation (3.3.52) as

$$P = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}_{4\times 3} \qquad \ldots (3.3.53)$$

The parity check matrix is given by equation (3.3.11) as,

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

Here P^T is the transpose of P submatrix and

I_q is the $q \times q$identity matrix.

By taking transpose of P submatrix of equation (3.3.53), the parity check matrix will be,

$$H = \underbrace{\left[\begin{array}{cccc} 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{array}\right.}_{P^T} \begin{array}{c} : \\ : \\ : \end{array} \underbrace{\left.\begin{array}{ccc} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array}\right]_{3\times 7}}_{I_3}$$

This is the required parity check matrix for (7, 4) cyclic code in systematic form.

3.3.7 Encoders for Cyclic Codes

In this section we will discuss the encoders for systematic cyclic codes. Fig. 3.3.2 shows the block diagram of a generalized (n, k) cyclic code. The symbols used to draw encoders are shown in Fig. 3.3.1.

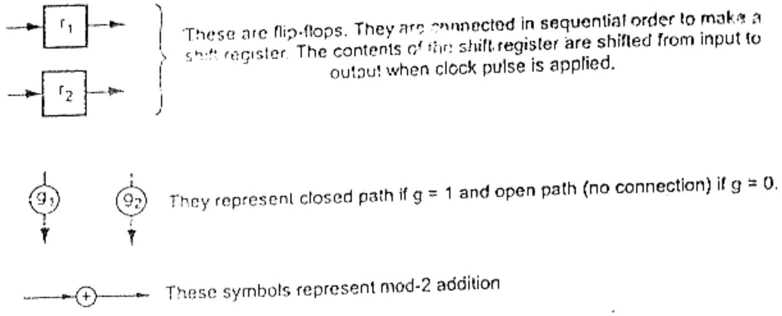

r1
r2
These are flip-flops. They are connected in sequential order to make a shift register. The contents of the shift register are shifted from input to output when clock pulse is applied.
g1
g2
They represent closed path if g = 1 and open path (no connection) if g = 0.
These symbols represent mod-2 addition

Fig. 3.3.1 Various symbols used in encoder

Operation : The feedback switch is first closed. The output switch is connected to message input. All the shift registers are initialized to all zero state. The *k* message bits are shifted to the transmitter as well as shifted into the registers.

After the shift of '*k*' mesage bits the registers contain '*q*' check bits. The feedback switch is now opened and output switch is connected to check bits position. With the every shift, the check bits are then shifted to the transmitter.

Here we observe that the block diagram performs the division operation and generates the remainder (i.e. check bits). This remainder is stored in the shift register after all message bits are shifted out.

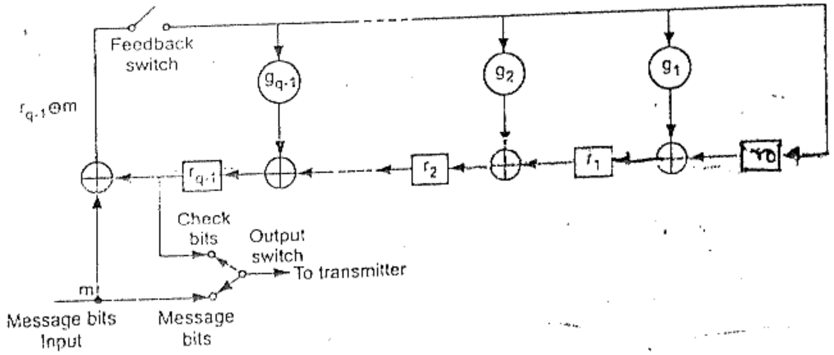


Fig. 3.3.2 Encoder for systematic (n, k) cyclic code

Example 3.3.8 : *Design the encoder for the (7, 4) cyclic code generated by $G(p) = p^3 + p + 1$ and verify its operation for any message vector.*

(Nov./Dec.-2003, 4 Marks)

Solution : The generator polynomial is,

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

and
$$G(p) = p^3 + g_2\, p^2 + g_1\, p + 1$$

On comparison of the two equation we obtain,

$$g_1 = 1 \quad \text{and} \quad g_2 = 0$$

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

With these values the block diagram of Fig. 3.3.2 will be as shown in Fig. 3.3.3 below.

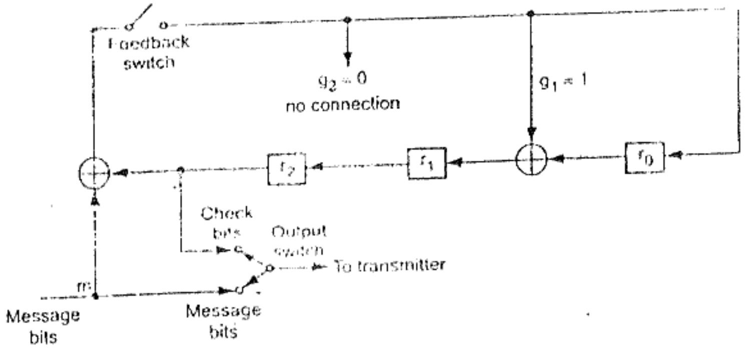


Fig. 3.3.3 Encoder for (7, 4) cyclic code for $G(p) = p^3 + p + 1$

Since $q = 3$, there are '3' flip-flops in shift register to hold check bits c_1 c_2 and c_0. Since $g_2 = 0$, its link is not connected. $g_1 = 1$, hence its link is connected. Now let's verify the operation of this encoder for message vector $M = (m_3\ m_2\ m_1\ m_0) = (1100)$. Table 3.3.3 shows the contents of shift registers before and after shifts.

Table (3.3.3) shows that at the end of last message bit the register bit outputs are $r_2' = 0$, $r_1' = 1$ and $r_0' = 0$. The feedback switch is opened and output switch is closed to check bits position. The check bits are then shifted to the transmitter. The check bits

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

Table 3.3.3 Shift register bits positions for input message M = (1100)

are shifted as $c_2 = r'_2$, $c_1 = r'_1$ and $c_0 = r'_0$. The following table illustrates the shift operation of message and check bits. We know that the code vector is,

$$X = (m_3\ m_2\ m_1\ m_0\ c_2\ c_1\ c_0) = (1\ 1\ 0\ 0\ 0\ 1\ 0)$$

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)

Table 3.3.4 Operation of (7, 4) cyclic code encoder

The above table illustrated how the bits are transmitted when input message is (1100).

3.3.8 Syndrome Decoding for Cyclic Codes

In cyclic codes also during transmission some errors may occur. Syndrome decoding can be used to correct those errors. Let's represent the received code vector by Y. If 'E represents an error vector then the correct code vector can be obtained as,

$$X = Y \oplus E \quad \text{(from equation 3.3.29)} \qquad \dots (3.3.54)$$

or we can write the above equation as,

$$Y = X \oplus E \qquad \dots (3.3.55)$$

We can write the above equation since it is mod-2 addition.

In the polynomial form we can write the above equation as,

$$Y(p) = X(p) + E(p) \qquad \dots (3.3.56)$$

Since $X(p) = M(p)\,G(p)$ the above equation will be,

$$Y(p) = M(p)\,G(p) + E(p) \qquad \dots (3.3.57)$$

Let the received polynomial $Y(p)$ be divided by $G(p)$ i.e.

$$\frac{Y(p)}{G(p)} = \text{Quotient} + \frac{\textit{Remainder}}{G(p)} \qquad \dots (3.3.58)$$

In the above equation if $Y(p) = X(p)$ i.e. if it does not contain any error then,

$$\frac{X(p)}{G(p)} = \text{Quotient} + \frac{\textit{Remainder}}{G(p)}$$

Since $X(p) = M(p)\,G(p)$, Quotient will be equal to $M(p)$ and remainder will be zero. This shows that if there is no error, then remainder will be zero. Here $G(p)$ is factor of code vector polynomial. Let's represent Quotient by $Q(p)$ and Remainder by $R(p)$ then equation (3.3.58) becomes,

$$\frac{Y(p)}{G(p)} = Q(p) + \frac{R(p)}{G(p)} \qquad \dots (3.3.59)$$

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

$$Y(p) = Q(p)\,G(p) + R(p) \qquad \dots (3.3.60)$$

On comparing equation 3.3.57 and above equation 3.3.60 we obtain,

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

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

The above equation has all mod-2 additions. Therefore subtraction and addition is same.

$$E(p) = [M(p) + Q(p)]\,G(p) + R(p) \qquad \dots (3.3.61)$$

This equation shows that for a fixed message vector and generator polynomial, an error pattern or error vector 'E depends on remainder R. For every remainder 'R' there will be specific error vector. Therefore we can call the remainder vector 'R' as syndrome vector 'S', or $R(p) = S(p)$. Therefore equation (3.3.59) will be,

$$\frac{Y(p)}{G(p)} = Q(p) + \frac{S(p)}{G(p)} \qquad \dots (3.3.62)$$

Thus the syndrome vector is obtained by dividing received vector $Y(p)$ by $G(p)$, i.e.

$$\boxed{S(p) = rem\left[\frac{Y(p)}{G(p)}\right]} \qquad \dots (3.3.62(a))$$

3.3.8.1 Block Diagram of Syndrome Calculator

Fig. 3.3.4 shows the generalized block diagram of a syndrome calculator.

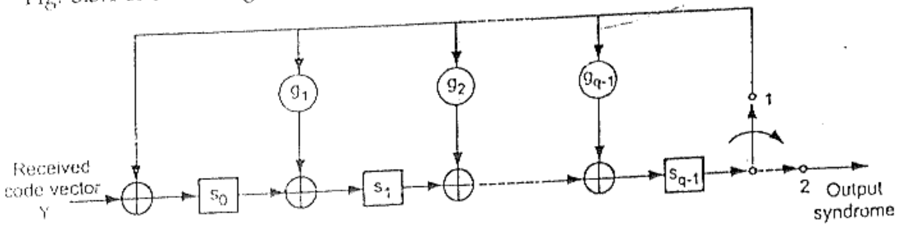


Fig. 3.3.4 Computation of syndrome for an (n, k) cyclic code

In above figure observe in figure that there are 'q' stage shift register to generate 'q' bit syndrome vector. The operations as follows -

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. The contents of flip flops keep on changing according to input bits of Y and values of g_1, g_2 etc. After all the bits of Y are shifted, the 'q' flip-flops of shift register contains the q - bit syndrome vector. The switch is then closed to position 2 and clocks are applied to the shift register. The output is a syndrome vector $S = (s_{q-1}, s_{q-2}, \dots s_1\ s_0)$

Example 3.3.9 : *Design a syndrome calculator for a (7, 4) cyclic Hamming code generated by the polynomial $G(p) = p^3 + p + 1$. Calculate the syndrome for Y = (1 0 0 1 1 0 1)* **(Nov./Dec.-2003, 4 Marks)**

Solution : For the given code $n = 7, k = 4, q = n - k = 7 - 4 = 3$

The given generator polynomial is,

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

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

On comparison of the above two equations we obtain,

$$g_1 = 1 \text{ and } g_2 = 0$$

With these values the block diagram of a syndrome calculator for (7, 4) cyclic code will be as shown in Fig. 3.3.5 .

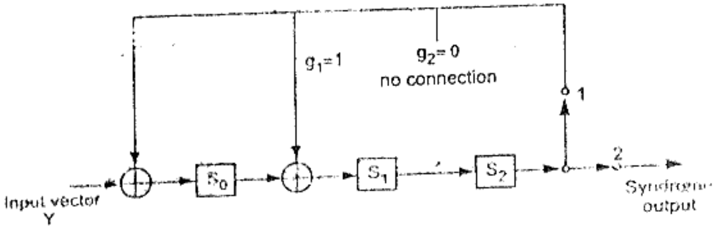


Fig. 3.3.5 Block diagram of a syndrome calculator for (7, 4) cyclic code with $G(p) = p^3 + p + 1$

Operation and explanation

The switch is kept in position 1 until all the '7' bits of received vector Y are shifted into the shift register. The flip-flops of the shift register contain syndrome vector when all bits of 'Y' are shifted. The switch is then closed to position 2 and clock pulses are applied to shift register. This gives syndrome vector at the output. The following table illustrates the operation of this syndrome calculator for received vector Y = (1 0 0 1 1 0 1). The table shows the contents of flip-flops with every shift.

The table shows that at the end of last shift the register contents are $(s_0\ s_1\ s_2) = (1\ 1\ 0)$.

Shift	Received vector i.e. bits of Y	Contents of flip flops in shift register		
		$s_0 = y \oplus s_2$	$s_1 = s_0 \oplus s_2$	$s_2 = s_1$
-	-	0	0	0
1	1	$1 \oplus 0 = 1$	0	0
2	0	0	$1 \oplus 0 = 1$	0
3	0	0	0	1
4	1	0	1	0
5	1	1	0	1
6	0	1	0	0
7	1 Syndrome	1	1	0

Table 3.3.5 Calculation of syndrome for Y = (1001101)

Hence, the calculated syndrome is,

$$S = (s_2\ s_1\ s_0) = (0\,1\,1)$$

3.3.9 Decoder for Cyclic Codes

Once the syndrome is calculated, then an error pattern is detected for that particular syndrome. When this error vector is added to the received vector Y, then it gives corrected code vector at the output. This decoding operation can be performed by the scheme shown in Fig. 3.3.6.

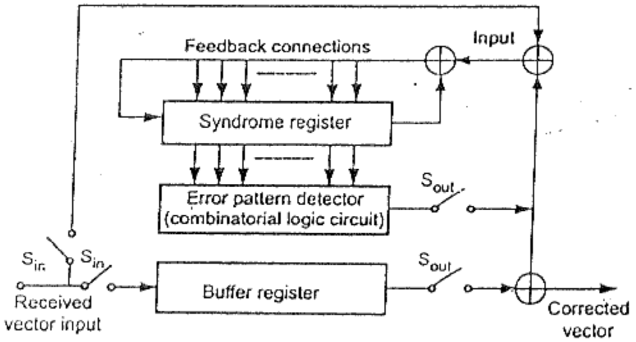


Fig. 3.3.6 Generalized block diagram of decoder for cyclic codes

Operation of the decoder

The switches named S_{out} are opened and S_{in} are closed. The bits of the received ector Y are shifted into the buffer register as well as they are shifted into the ndrome calculator. When all the 'n' bits of the received vector Y are shifted into uffer register and syndrome calculator the syndrome register holds a syndrome ector. The syndrome vector is given to the error pattern detector. A particular ndrome detects a specific error pattern. The switches S_{in} are opened and S_{out} are closed. The shifts are then applied to the flip-flops of buffers register, error register (which holds error pattern) and syndrome register. The error pattern is then added bit by bit to the received vector (which is stored in buffer register). The output is the corrected error free vector.

3.3.10 Advantages and Disadvantages of Cyclic Codes

As we have seen that cyclic codes are the subclass of linear block codes, they have some advantages over noncyclic block codes as given below -

Advantages :

1) The error correcting and decoding methods of cyclic codes are simpler and easy to implement. These methods eliminate the storage needed for lookup table decoding. Therefore the codes becomes powerful and efficient.

2) The encoders and decoders for cyclic codes are simpler compared to noncyclic codes.

3) Cyclic codes also detect error burst that span many successive bits.

4) Cyclic codes have well defined mathematical structure. Hence very efficient decoding schemes are possible.

Inspite of these advantages cyclic codes also have some disadvantages.

Disadvantages :

1) The error detection in cyclic codes is simpler but error correction is little complicated since the combinational logic circuits in error detector are complex.

To avoid such complex circuits some special cyclic codes are used which are discussed next.

3.3.11 BCH Codes (Bose - Chaudhri - Hocquenghem Codes)

BCH codes are most extensive and powerful error correcting cyclic codes. The decoding of BCH codes is comparatively simpler. For any positive integer m and t (where $t < 2^{m-1}$) there exists a BCH code with following parameters

Block length : $n = 2^{m-1}$

Number of parity check bits : $n - k \leq mt$

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

The decoding schemes of BCH codes can be implemented on digital computer. Because of software implementation of decoding schemes they are quite flexible compared to hardware implementation of other schemes.

3.3.12 Reed-Soloman (RS) Codes

These are nonbinary BCH codes. The encoder of RS codes operate on multiple bits simultaneously. The (n, k) RS code takes the groups of m-bit symbols of the incoming binary data stream. It takes such 'k' number of symbols in one block. Then the encoder adds (n – k) redundant symbols to form the codeword of 'n' symbols. Thus there are 'n' symbols in the codeword. Each symbol contains 'm' number of bits. Normally m = 8 is taken. The 't' error correcting RS code has the following parameters :

Block length : $n = 2^m - 1$ symbols

Message size : k symbols

Parity check size : $n - k = 2t$ symbols

Minimum distance : $d_{\min} = 2t + 1$ symbols.

Here observe that the minimum distance is greater than the number of parity symbols. Hence this code is maximum distance separable code. These codes provide wide range of code rates. Efficient decoding techniques are available with RS codes.

3.3.13 Golay Codes

Golay code is the (23, 12) cyclic code whose generating polynomial is,

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

This code has minimum distance of $d_{\min} = 7$. This code can correct upto 3 errors. But Golay code cannot be generalized to other combinations of n and k.

3.3.14 Shortened Cyclic Codes

For the (n, k) cyclic code, the generator polynomials are divisors of $x^n + 1$. The polynomial $x^n + 1$ has very few divisors. Hence there are very few generator polynomials available. This difficulty can be overcome by shortened cyclic codes. In shortened cyclic codes, the last 'i' bits out of 'n' bits of the codeword are padded with zeros. This 'i' bits are not transmitted. Only (n – i) bits of the codeword are transmitted. The decoder pads 'i' zeros to the received codeword. Thus for (n, k) cyclic code, (n – i, k – i) shortend cyclic code is generated. This code has all the advantages of original (n, k) code. Its error detection and correction capabilities are same as the original (n, k) cyclic code.

3.3.15 Burst Error Correcting Codes

In the preceeding sections we discussed the codes which detect and correct errors occurring independently at different bit positions. Burst errors occur as a cluster of errors. Cyclic and shortened codes can be used to detect these burst errors.

The burst of length q is defined as the vectors whose nonzero components are confined to 'q' consecutive digit positions with nonzero first and last digits. For example the vector x = [0 0 1 0 1 1 1 0 1 0 1 0 0 0] has the burst of length 9. The q-burst error correcting code is capable of correcting the bursts length q or less. The following theorem gives the number of parity bits required by burst error correcting code :

The q-burst error correcting code must have at least 2q parity check digits i.e.,

$$n - k \geq 2q \qquad \text{... (3.3.63)}$$

Thus we can say that the (n, k) burst error correcting code can correct the bursts of length upto $\frac{n-k}{2}$. This becomes the upper bound on the burst error correcting capability of (n, k) code i.e.,

$$q \leq \frac{n-k}{2} \qquad \text{.... (3.3.64)}$$

The burst error correcting efficiency is denoted by z. It is given as,

$$z = \frac{2q}{n-k} \qquad \text{... (3.3.65)}$$

To detect the burst of length d, then the check bits must be,

$$n - k \geq d \qquad \text{... (3.3.66)}$$

Thus the check bits must be at least equal to d.

➡ **Example 3.3.10** *Consider the (15, 9) cyclic code generated by*

$$G(p) = p^6 + p^5 + p^4 + p^3 + 1$$

This code has a burst error correcting ability of $q = 3$. Find burst error correcting efficiency of this code.

Solution : The given code has q = 3

It is (15, 9) code. Hence

$$n = 15, \quad k = 9$$

The burst error correcting efficiency is given by equation (3.3.65) i.e.,

$$z = \frac{2q}{n-k}$$

Putting values in above equation ,

$$z = \frac{2 \times 3}{15 - 9} = 1 \quad \text{or} \quad 100\%$$

Thus the burst error correcting efficiency of this code is 100%

3.3.16 Interleaving of Coded Data for Burst Error Correction

The block codes like hamming, cyclic, etc are effective when the errors in the channel are statistically independent. For example, the errors in Additive white gaussian noise (AWGN) channel are statistically independent. But there are some channels which produce burst errors. For example the channels having multipath fading. Because of multipath propagation the signal fading occurs and it creates errors at the receiver. This phenomena depends upon time characteristics of the channel. The burst error can be produced when the data is stored on magnetic tape or disk. The defects in the magnetic material creates clusters of errors. If the block codes are optimally designed for statistically independent errors, then they cannot correct the burst errors. In this section we will see how interleaving of coded data is used to correct burst errors.

A burst error of length 'b' is a sequence of b-bit errors. A systematic (n,k) block is capable of correcting the error bursts of length $b \le \frac{1}{2}(n-k)$. The burst errors are converted to statistically independent errors by interleaving the coded data. Then the code designed for independent errors can be used to correct these errors.

Fig. 3.3.7 shows the block diagram of the system which uses the interleaving technique to correct burst errors. The channel encoder encodes the data by some (n,k) block code. The coded data thus has codewords of length 'n'. This coded data from the channel encoder is given to the block interleaver.

The block interleaver has 'm' rows and 'n' columns as shown in Fig. 3.3.7.

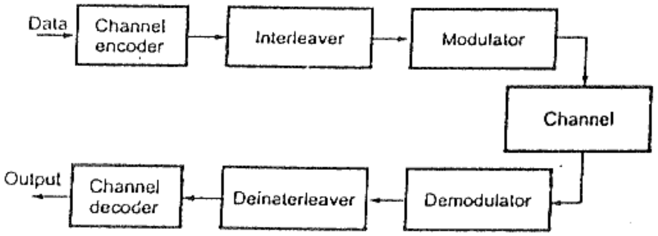
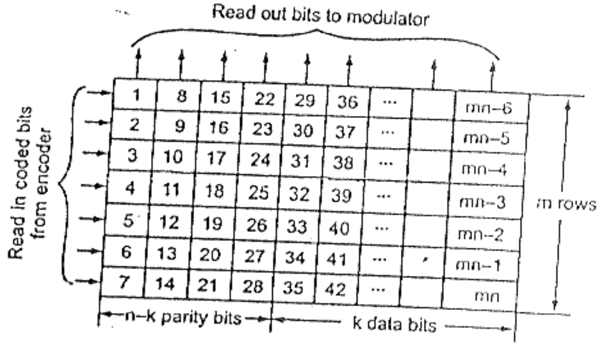


Fig. 3.3.7 Interleaving of coded data to correct burst errors

Thus the codeword bits are stored in the interleaver rowwise. The numbers indicate the actual bit numbers as they come from encoder. The interleaver in the above figure thus stores 'm' codewords of 'n' bits length, or total 'mn' bits. As shown in Fig. 3.3.8, the bits are given to the modulator column wise. The modulator then transmits these bits on modulated carrier over the channel. At the receiver the demodulator gets these bits back by demodulation and soft or hard decision decoding. The deinterleaver then stores these bits in the same format as shown in Fig. 3.3.8. The channel decoder then reads the bits row wise with one codeword of length 'n' at a time. Because of this recording of coded data by the interleaver, the error burst of length 'mb' is broken into 'm' bursts of length 'b'. The (n,k) block code then has to correct these small bursts of length 'b'. By increasing 'm', the length of the bursts can be further reduced. The block code which uses an interleaver of size $m \times n$, is also called as interleaved (mn, mk) block code.



1	8	15	22	29	36	...		mn–6
2	9	16	23	30	37	...		mn–5
3	10	17	24	31	38	...		mn–4
4	11	18	25	32	39	...		mn–3
5	12	19	26	33	40	...		mn–2
6	13	20	27	34	41	...	,	mn–1
7	14	21	28	35	42	...		mn

Fig. 3.3.8 A data recording in the block interleaver

In this section we discussed a block interleaver. Another type of interleaver called convolutional interleaver. Such convolutional interleavers or encoders are discussed in the next section.

3.3.17 Interlaced Codes for Burst and Random Error Correction

Definition of interlaced code

Consider an (n,k) block code. When λ number of codewords of this code are interlaced, then the code becomes $(\lambda n, \lambda k)$. This code is called interlaced code.

How an interlacing takes place ?

Consider (15, 8) code. Let its three code vectors be as follows :

$$x = (x_1\ x_2\ x_3\ \dots\ x_{14}\ x_{15})$$

$$y = (y_1\ y_2\ y_3\ \dots\ y_{14}\ y_{15})$$

$$z = (z_1\ z_2\ z_3\ \dots\ z_{14}\ z_{15})$$

If these three code vectors are transmitted directly, then we get the sequence as,

$$x_1\ x_2\ x_3\ \dots x_{14}\ x_{15}\ y_1\ y_2\ y_3\ \dots y_{14}\ y_{15}\ z_1\ z_2\ z_3\ \dots z_{14}\ z_{15} \qquad \dots (3.3.67)$$

If individual bits of the three code vectors are interlaced, then we get the sequence

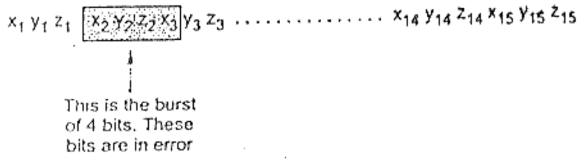
$$x_1\ y_1\ z_1\ x_2\ y_2\ z_2\ x_3\ y_3\ z_3\ \dots\dots\dots\dots x_{14}\ y_{14}\ z_{14}\ x_{15}\ y_{15}\ z_{15} \qquad \dots (3.3.68)$$

This is interlaced sequence. Here we have interlaced three codevectors. Hence = 3. Therefore the interlaced code will be of dimension,

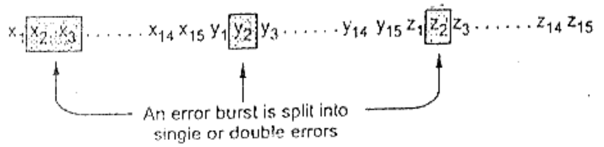
$$(\lambda n\ \lambda k) = (3 \times 15,\ 3 \times 8) = (45,\ 24)$$

How burst errors are corrected by interlaced codes ?

Consider the interlaced sequence of equation 3.3.68,


$x_1\,y_1\,z_1\,\boxed{x_2\,y_2\,z_2\,x_3}\,y_3\,z_3 \ldots\ldots\ldots\ldots\ldots x_{14}\,y_{14}\,z_{14}\,x_{15}\,y_{15}\,z_{15}$
This is the burst of 4 bits. These bits are in error

During transmission of the interlaced sequence, a burst of error takes place as shown above. Four successive bits are in error. At the receiver, when this sequence is received, it is converted to its noninterlaced natural form. That is given by equation (3.3.67). It is shown below.


$x_1\,\boxed{x_2\,x_3} \ldots\ldots x_{14}\,x_{15}\,y_1\,\boxed{y_2}\,y_3 \ldots\ldots y_{14}\,y_{15}\,z_1\,\boxed{z_2}\,z_3 \ldots\ldots z_{14}\,z_{15}$
An error burst is split into single or double errors

As shown above, the single burst of 4 digits is split into single or double errors.

Error detecting and correcting capabilities of $(\lambda n, \lambda k)$ code :

Let the (n,k) code corrects 't' digits. Then the interlaced code can correct any combination of 't' bursts of length λ or less.

Why cyclic codes are more suitable for burst error correction ?

If the code (n,k) is cyclic, then its interlaced version $(\lambda n, \lambda k)$ is also cyclic. If $G(p)$ is the generating polynomial of (n,k) code, then $G(p^{\lambda})$ is the generating polynomial of $(\lambda n, \lambda k)$ code.

Therefore encoding and decoding of interlaced code is also possible using shift registers. To obtain the decoder of interlaced code, each shift register stage of (n,k) cyclic code is replaced with λ stages without changing other connections. Because of all the above reasons, cyclic codes are more suitable for detecting and correcting error bursts.

3.3.18 Cyclic Redundancy Check (CRC) Codes

Definition : A cyclic code which is used for *error detection* purpose only is called cyclic redundancy check (CRC) code.

Why CRC codes ?

Cyclic codes are very much suitable for error detection because of two reasons :

i) Many combinations likely errors can be detected with the help of cyclic codes.

ii) Implementation of encoding and error detection circuits is practically possible.

Error detection capabilities of binary (n,k) CRC codes

The CRC codes are capable of detecting -

i) All error bursts of length $(n-k)$ or less.

ii) Fraction of error bursts of length equal to $(n-k+1)$.

iii) Fraction of error bursts of length greater than $(n-k+1)$.

iv) All error combinations of $(d_{\min}-1)$ or less.

v) If generator polynomial $G(p)$ has even number of coefficients, than all error patterns with odd number of errors can also be detected.

Commonly used CRC codes

Three commonly used CRC codes are given below :

CRC - 12 : $G(p) = 1 + p + p^2 + p^3 + p^{11} + p^{12}$, with $n-k = 12$

CRC - 16 : $G(p) = 1 + p^2 + p^{15} + p^{16}$, with $n-k = 16$

CRC - ITU : $G(p) = 1 + p^5 + p^{12} + p^{16}$, with $n-k = 16$

All the above codes contain $1+p$ as a prime factor. CRC-12 code is used for 6-bit characters. CRC-16 and CRC-ITU are used for 8-bit characters.

Applications :

1) CRC codes are used mainly in ARQ systems for error detection.

2) They are also used in digital subscriber lines.

3.3.19 Concatenated Block Codes

Nonbinary codes

Till now we have discussed linear block codes which are binary in nature. Nonbinary codes also exists. The nonbinary code consists of the set of fixed length codewords. The individual elements of the codeword are selected from the alphabet of q symbols, {0, 1, 2, ... q–1}. Normally $q = 2^k$, means k information bits can generate 'q' different symbols. The length of the nonbinary codeword is represented by N. The number of information symbols in nonbinary codes are represented by K. These K information symbols are encoded into 'N' number of symbols by the nonbinary code.

Definition of concatenated block code

The concatenated code is obtained by combining two separate codes. Normally one code is binary and other is nonbinary to form the combined concatenated code. Fig. 3.3.9 shows the block diagram of the system which uses concatenated block code.

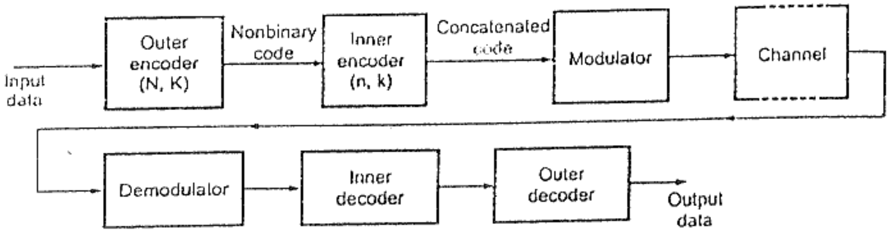


Fig. 3.3.9 A communication system which uses concatenated code

As shown in figure, the nonbinary code (N, K) is the outer code and binary code (n, k) is the inner code. The codewords of the concatenated code are formed by subdividing the block of kK information bits into K groups. Each such group is called symbol and it consists of k bits. The outer encoder encodes the K symbols into N symbols. The inner encoder encodes the k-bit symbol into n bit codeword. Thus the final codeword is made up of N symbols of n bits each. This is called concatenated block code of length Nn bits. This concatenated block code consists of Kk number of information bits. This is equivalent to (Nn, Kk) binary code. The concatenated codewords then modulate some carrier in the modulator and transmitted over the channel. At the receiver side, the demodulator generates the transmitted codewords back from the received signal. The inner decoder then makes hard decision on the group of every n bits. These 'n' bits are then converted to k information bits using minimum distance decoding. These k information bits represent one symbol of the nonbinary outer code. The group of N such symbols is used by the outer decoder to get K information symbols. The outer decoder also uses hard decision minimum distance decoding. Soft decision decoding can also be used for concatenated codes if the number of codewords are small.

Minimum distance and code rate

The minimum distance of the concatenated block code is $d_{min} D_{min}$. Here d_{min} is the minimum distance of the inner code and D_{min} is the minimum distance of the outer code. Similarly the rate of the concatenated code is Kk/Nn. This is equivalent to product of code rates of inner code and outer code.

Example 3.3.11 : *The generator polynomial of a (15, 11) Hamming code is given by $g(x) = 1+x+x^4$. Develop encoder and syndrome calculator for this code using systematic form.*

Solution : This is (15, 11) block code. Hence,

$$n = 15$$

$$k = 11$$

and

$$q = n - k = 15 - 11 = 4$$

i) To develop encoder

The generator polynomial is,

$$G(p) = 1+p+p^4 = p^4+0p^3+0p^2+p+1$$

and

$$G(p) = p^4+g_3p^3+g_2p^2+g_1p+1$$

Comparing the above two equations,

$g_3 = 0, \quad g_2 = 0, \quad g_1 = 1$

Fig. 3.3.10 shows the generalized encoder for (n, k) cyclic code. Based on this, the encoder of this example is shown in Fig. 3.3.10. In this figure observe that there are four flip-flops to hold four check bits. Since $g_1 = 1$, its link is connected. But $g_2 = g_3 = 0$, hence their links are not connected.

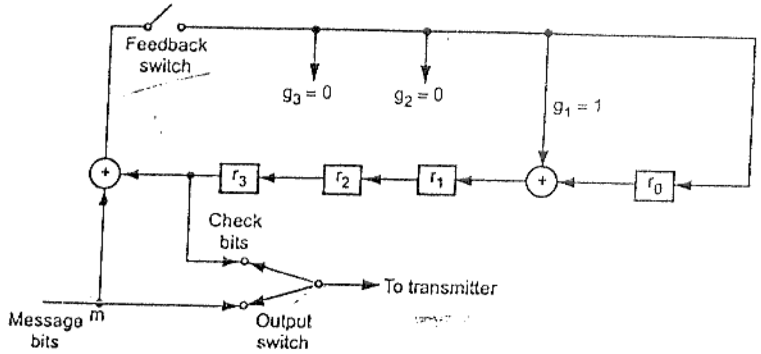


Fig. 3.3.10 Encoder for cyclic code $G(p) = 1+p+p^4$

ii) To develop syndrome calculator

The generalized syndrome calculator for (n, k) cyclic code is shown in Fig. 3.3.3. The syndrome calculator for this example is shown based on Fig. 3.3.4. The syndrome calculator is shown in Fig. 3.3.11.

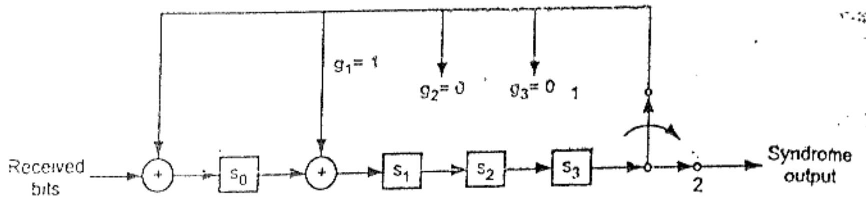


Fig. 3.3.11 Syndrome calculator for cyclic code $G(p) = 1 + p + p^4$

In the above figure observe that the links of g_2 and g_3 are not connected. Only g_1 is connected. The four flip-flops contain the four bit syndrome vector. The switch is kept in position '1' till all the bits of the received vector are shifted into shift register ($s_0 - s_3$). Then the flip-flops of the shift register contain 4-bit syndrome. The switch is then moved to position '2' to transmit the syndrome vector.

Example 3.3.12 : *Construct a systematic (7, 4) cyclic code using the generator polynomial $g(x) = x^3 + x + 1$. What are the error correcting capabilities of this code ? Construct the decoding table and for the received codeword 1101100, determine the transmitted data word.*

Solution : For this code n = 7, k = 4 and q = 3.

i) To determine generator matrix (G)

The t^{th} row of the generator matrix is given as, (equation 3.3.47),

$$p^{n-t} + R_t(p) = Q_t G(p) \quad \text{and} \quad t = 1, 2, \ldots\ldots k$$

We have obtained the generator matrix based on the above equation in example 3.3.5 for the same generating polynomial. It is given by equation (3.3.52) as,

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & : & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & : & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & : & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & : & 0 & 1 & 1 \end{bmatrix} \quad \ldots (3.3.69)$$

ii) To construct codevectors

The codevector can be obtained from the generator matrix as,

$$X = MG$$

Let us take the message vector as M = 0101. Then the codevector for this message will be,

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

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

Thus the check bits are 100. We know that,

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

Hence P submatrix can be obtained from equation 3.3.69 as,

$$P = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \quad \ldots (3.3.70)$$

Hence the check bits can be obtained by

$$C = MP$$

$$\therefore \quad [C_1\ C_2\ C_3] = [m_0\ m_1\ m_2\ m_3] \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

$$\therefore \quad C_1 = m_0 \oplus m_1 \oplus m_2$$

$$C_2 = m_1 \oplus m_2 \oplus m_3$$

$$C_3 = m_0 \oplus m_1 \oplus m_3$$

The check bits can be obtained for all the codevectors with the help of above equations. Table 3.3.2 lists all the systematic codevectors. We will get the same check bits of Table 3.3.2 from above equations.

iii) Error correcting capability

It is clear from Table 3.3.2 that,

$$d_{min} = [w(X)]_{min} = 3$$

Hence this code can detect upto two errors and correct one error.

iv) To obtain parity check matrix

The parity check matrix is given as,

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

$$\therefore \quad H^T = \begin{bmatrix} P \\ \cdots \\ I_q \end{bmatrix}$$

Hence from equation (3.3.76) we can write above matrix as follows :

$$H^T = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \text{... (3.3.71)}$$

v) To obtain decoding table

The decoding table can be easily prepared from H^T. For the block code, each row of H^T represents a syndrome and unique error pattern. This we have discussed earlier in linear block codes. Table 3.3.6 shows the error patterns and the syndrome vectors.

Sr. No.	Error vector 'E' showing single bit error patterns							Syndrome vector			Comments
1	0	0	0	0	0	0	0	0	0	0	
2	1	0	0	0	0	0	0	1	0	1	← 1st row of H^T
3	0	1	0	0	0	0	0	1	1	1	← 2nd row of H^T
4	0	0	1	0	0	0	0	1	1	0	← 3rd row of H^T
5	0	0	0	1	0	0	0	0	1	1	← 4th row of H^T
6	0	0	0	0	1	0	0	1	0	0	← 5th row of H^T
7	0	0	0	0	0	1	0	0	1	0	← 6th row of H^T
8	0	0	0	0	0	0	1	0	0	1	← 7th row of H^T

Table 3.3.6 Decoding table

vi) To decode 1 1 0 1 1 0 0

Determine syndrome

Let the received codeword be,

$$Y = [1\ 1\ 0\ 1\ 1\ 0\ 0]$$

$$\therefore \quad Y(p) = p^6 + p^5 + p^3 + p^2$$

The syndrome vector is given by equation (3.3.62).

i.e.,

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

We know that $G(p) = p^3 + p + 1$. Hence let us perform the division of above equation. $Y(p)$ can be written as,

$$Y(p) = p^6 + p^5 + 0p^4 + p^3 + p^2 + 0p + 0$$

And $G(p)$ can be written as,

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

The division is as shown below :

$$\begin{array}{r l}
 & p^3 + p^2 + p + 1 \\ \cline{2-2}
p^3 + 0p^2 + p + 1 \,\big) & p^6 + p^5 + 0p^4 + p^3 + p^2 + 0p + 0 \\
 & p^6 + 0p^5 + p^4 + p^3 \\
 & \oplus \quad \oplus \quad \oplus \quad \oplus \\ \cline{2-2}
 & \quad p^5 + p^4 + 0p^3 + p^2 \\
 & \quad p^5 + 0p^4 + p^3 + p^2 \\
 & \oplus \quad \oplus \quad \oplus \quad \oplus \\ \cline{2-2}
 & \quad p^4 + p^3 + 0p^2 + 0p \\
 & \quad p^4 + 0p^3 + p^2 + p \\
 & \oplus \quad \oplus \quad \oplus \quad \oplus \\ \cline{2-2}
 & \qquad p^3 + p^2 + p + 0 \\
 & \qquad p^3 + 0p^2 + p + 1 \\
 & \oplus \quad \oplus \quad \oplus \quad \oplus \\ \cline{2-2}
\text{Remainder} \rightarrow & \qquad p^2 + 0p + 1
\end{array}$$

Thus the remainder is,

$$S(p) = p^2 + 0p + 1$$

i.e. $S = [1\ 0\ 1]$

The syndrome is non zero. Hence there is an error in the received codeword.

Determine error pattern for $S = 101$ and correct the codeword

Table 3.3.5 indicates that there is error in the first bit, i.e.,

$$E = [1\ 0\ 0\ 0\ 0\ 0\ 0]$$

Hence correct codevector is,

$$X = Y \oplus E$$

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

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

Thus the transmitted codeword is X = 0 1 0 1 1 0 0. It is also one of the codevector in Table 3.3.2.

Example 3.3.13 : *Determine the encoded message for the following 8-bit data codes using the following CRC generating polynomial $P(x) = x^4 + x^3 + x^0$.*
i) 11001100 ii) 01011111

Solution : Here $G(P) = x^4 + x^3 + x^0$ hence q = 4

$$= p^4 + p^3 + p^0$$

and length of message bits is k = 8

$$q = n - k \quad \text{or} \quad n = k + q$$

$$= 8 + 4 = 12 \text{ bits.}$$

i) Consider the first 8-bit data 11001100

∴ Message polynomial will be,

$$M(p) = p^7 + p^6 + p^3 + p^2$$

Let us find $p^q M(p)$. Since q = 4,

$$p^4 M(p) = p^4(p^7 + p^6 + p^3 + p^2)$$

$$= p^{11} + p^{10} + p^7 + p^6$$

Now let us divide $p^4 M(p)$ by $G(p)$. Then we get,

$$p^7 + p^2 + p + 1$$

$$p^4 + p^3 + 1 \overline{)p^{11} + p^{10} + 0p^9 + 0p^8 + 0p^7 + p^6 + 0p^5 + 0p^4 + 0p^3 + 0p^7 + 0p + 0}$$

$$\underline{p^{11} + p^{10} + 0p^8 + 0p^7}$$

$$0 \quad 0 \quad 0 \quad 0 \quad p^6 + 0p^5 + 0p^4 + 0p^3 + 0p^2$$

$$\underline{p^6 + p^5 + 0p^4 + 0p^3 + p^2}$$

$$0 + p^5 + 0p^4 + 0p^3 + p^2 + 0p$$

$$\underline{p^5 + p^4 + 0p^3 + 0p^2 + p}$$

$$0 + p^4 + 0p^3 + p^2 + p + 0$$

$$\underline{p^4 + p^3 + 0p^2 + 0p + 1}$$

$$0 + p^3 + p^2 + p + 1$$

Thus the remainder is

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

Therefore check bits are,

$$C = (1\ 1\ 1\ 1)$$

Therefore the encoded message in systematic form is given as,

$$X = (\text{Message bits : Check bits})$$

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

Thus there are 12 bits in encoded message.

ii) The given 8-bit data is 01011111

The corresponding message polynomial is,

$$M(p) = p^6 + p^4 + p^3 + p^2 + p^1 + 1$$

Therefore $p^2 M(p)$ will be,

$$p^4 M(p) = p^4(p^6 + p^4 + p^3 + p^2 + p + 1)$$

$$= p^{10} + p^8 + p^7 + p^6 + p^5 + p^4$$

Now let us divide $p^2 M(p)$ by $G(p)$, then we get,

$$\begin{array}{r}
p^6+p^5+p^3+p^2+p \\
p^4+p^3+1\overline{)p^{10}+0p^9+p^8+p^7+p^6+p^5+p^4+0p^3+0p^2+0p^1+0} \\
\underline{p^{10}+p^9+0p^8+0p^7+p^6} \qquad\qquad\qquad\qquad\qquad\qquad \\
0+p^9+p^8+p^7+0p^6+p^5 \qquad\qquad\qquad\qquad\qquad \\
\underline{p^9+p^8+0p^7+0p^6+p^5} \qquad\qquad\qquad\qquad\qquad \\
0 \quad 0 \quad p^7+0p^6+0p^5+p^4+0p^3 \qquad\qquad\qquad \\
\underline{p^7+p^6+0p^5+0p^4+p^3} \qquad\qquad\qquad \\
0+p^6+0p^5+p^4+p^3+0p^2 \qquad\qquad \\
\underline{p^6+p^5+0p^4+0p^3+p^2} \qquad\qquad \\
0+p^5+p^4+p^3+p^2+0p \qquad \\
\underline{p^5+p^4+0p^3+0p^2+p} \qquad \\
0+0+p^3+p^2+p \qquad
\end{array}$$

Thus the remainder is,

$$C(p) = p^3+p^2+p$$

Therefore the check bits are,

$$C = 1\,1\,1\,0$$

Therefore message in systematic form will be,

$$X = 0\,1\,0\,1\,1\,1\,1\,1 : 1\,1\,1\,0.$$

Example 3.3.14 : *Suggest a suitable generator polynomial for a (7, 4) systematic cyclic code and find codevectors for the following data words :*

(i) 1010 (ii) 1111 (iii) 0001 (iv) 1000.

Draw an encoder arrangement for the above code and explain its operation. Construct the decoding table for all single bit error patterns and determine the data vectors transmitted for the following received vectors.

(i) 1101101 (ii) 0101000

Solution : Given (7, 4) cyclic code. Hence

$$n = 7, \; k=4$$

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

(i) To obtain the generator polynomial :

The generator polynomial will be the factor of (p^n+1), i.e.

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

The generator polynomial must be of the degree 'q'. As given by above equation, two generator polynomials are possible : p^3+p+1 and p^3+p^2+1. Hence let us use the generator polynomial,

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

(ii) To obtain the generator matrix in systematic form :

We have obtained the generator matrix in systematic form for $G(p)=p^3+p+1$ in Ex. 3.3.4. It is calculated (see equation 3.3.52) as,

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

(iii) To determine the code vectors :

1) Codevector for M = 1010

We know that X = M G. Therefore,

$$X = [1 \;\; 0 \;\; 1 \;\; 0]\begin{bmatrix} 1 & 0 & 0 & 0 & : & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & : & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & : & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & : & 0 & 1 & 1 \end{bmatrix} = [1\,0\,1\,0\,0\,1\,1]$$

Observe that the same codevector is obtained for M = 1010 in table 3.3.2, since the generating polynomial is same.

2) Codevector for $M=[1\,1\,1\,1]$

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

3) Codevector for $M=0001$

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

4) Codevector for $M=1000$

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

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

iv) Encoder arrangement :

Encoder for $G(p)=p^3+p+1$ is shown in Ex. 3.3.6. Its operation is also explained in this example.

v) Decoding table :

The decoding table for $G(p)=p^3+p+1$ is given in Ex. 3.3.10 and table 3.3.4.

vi) To decode the given vectors :

i) To decode $y=1101101$

To determine the syndrome

Hence the polynomial of received vector will be,

$$Y(p) = p^6+p^5+p^3+p^2+1$$

Syndrome can be calculated by equation 3.3.62 (a). i.e.,

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

Let us divide $Y(p)$ by $G(p)$.

$$\begin{array}{r l}
 & p^3+p^2+p+1 \\
p^3+0p^2+p+1 \big) & \overline{p^6+p^5+0p^4+p^3+p^2+0p+1} \\
 & p^6+0p^5+p^4+p^3 \\
 & \oplus \quad \oplus \quad \oplus \\
\hline
 & p^5+p^4+0p^3+p^2 \\
 & p^5+0p^4+p^3+p^2 \\
 & \oplus \quad \oplus \quad \oplus \quad \oplus \\
\hline
 & p^4+p^3+0p^2+0p \\
 & p^4+0p^3+p^2+p \\
 & \oplus \quad \oplus \quad \oplus \quad \oplus \\
\hline
 & p^3+p^2+p+1 \\
 & p^3+0p^2+p+1 \\
\hline
 & p^2
\end{array}$$

Thus the remainder is, $S(p)=p^2$. It can be written as,

$$S(p) = p^2+0p+0$$

$$S = 1\,0\,0$$

Since the syndrome is nonzero, there is an error in the received vector.

To determine the error vector

From the decoding table 3.3.5, observe that for the syndrome of S = 100, an error vector is,

$$E = 0\,0\,0\,0\,1\,0\,0$$

To determine the correct codevector

Correct codevector is given as,

$$\begin{aligned} X &= Y \oplus E \\ &= [1\,1\,0\,1\,1\,0\,1] \oplus [0\,0\,0\,0\,1\,0\,0] \\ &= 1\,1\,0\,1\,0\,0\,1 \end{aligned}$$

Note that this is one of the systematic codevector in table 3.3.2 for M = 1101.

2) To decode Y = 0101000

To determine the syndrome

Polynomial for this received vector becomes,

$$Y(p) = p^5+0p^4+p^3+0p^2+0p+0$$

Let us divide $Y(p)$ by $G(p)$. i.e.,

$$\begin{array}{r l}
 & p^2 \\
p^3+0p^2+p+1 \big) & \overline{p^5+0p^4+p^3+0p^2+0p+0} \\
 & p^5+0p^4+p^3+p^2 \\
 & \oplus \quad \oplus \quad \oplus \\
\hline
 & p^2+0p+0
\end{array}$$

Thus the remainder is, $S(p)=p^2+0p+0$

$\therefore \qquad S=100$

Since the syndrome is nonzero, there is an error in the received vector.

To determine error vector

From the decoding table 3.3.5, observe that for the syndrome of S = 100, an error vector is,

$$E = 0\,0\,0\,0\,1\,0\,0$$

To determine the correct codevector

Correct codevector is given as,

$$X = Y \oplus E$$

$$= (0101000) + (0000100)$$

$$= 0\,1\,0\,1\,1\,0\,0$$

Note that this is one of the systematic code vector in table 3.3.2 for $M = 0101$.

➡ **Example 3.3.15 :** *Why are cyclic codes effective in detecting error bursts ? The message 1001001010 is to be transmitted in a cyclic code with a generator polynomial $g(x) = x^2 + 1$.*

i) How many check bits does the encoded message contain ?

ii) Obtain the transmitted codeword.

iii) Draw encoding arrangement to obtain remainder bits.

iv) After the received word is clocked into the decoder input, what should be the content of the register stores ?

Solution : i) To determine number of check bits :

To determine size of the code

The given message is,

$$M = (1001001010) \quad \text{i.e.} \quad k = 10$$

The generator polynomial is,

$$G(p) = p^2 + 1, \quad \text{hence} \quad q = 2$$

Therefore $n = k + q = 10$. Thus this is $(12, 10)$ cyclic code.

Number of check bits

The encoded message contains 'q' number of check bits. Here $q = 2$ check bits will be present.

ii) To obtain transmitted code word :

To obtain transmitted codeword we have to perform following steps :

a) Divide $p^q M(p)$ by $G(p)$.

b) From remainder determine check bits.

c) Transmitted codeword will be $X = (M : C)$.

a) To divide $p^q M(p)$ by $G(p)$

We know that $M = (1001001010)$

Hence message polynomial will be,

$$M(p) = p^9 + 0p^8 + 0p^7 + p^6 + 0p^5 + 0p^4 + p^3 + 0p^2 + p + 0$$

Here $q = 2$. Hence $p^q M(p)$ will be,

$$p^q M(p) = p^2 (p^9 + 0p^8 + 0p^7 + p^6 + 0p^5 + 0p^4 + p^3 + 0p^2 + p + 0)$$

$$= p^{11} + 0p^{10} + 0p^9 + p^8 + 0p^7 + 0p^6 + p^5 + 0p^4 + p^3 + 0p^2$$

The generator polynomial is, $G(p) = p^2 + 1$

$$= p^2 + 0p + 1$$

$$\begin{array}{r l}
 & p^9 + p^7 + p^6 + p^5 + p^4 + p^2 + p + 1 \\
p^2 + 0p + 1 \,\big) & \overline{p^{11} + 0p^{10} + 0p^9 + p^8 + 0p^7 + 0p^6 + p^5 + 0p^4 + p^3 + 0p^2 + 0p + 0} \\
 & p^{11} + 0p^{10} + p^9 \\
 & \underline{\oplus \quad \oplus \quad \oplus} \\
 & \quad p^9 + p^8 + 0p^7 \\
 & \quad p^9 + 0p^8 + p^7 \\
 & \quad \underline{\oplus \quad \oplus \quad \oplus} \\
 & \qquad p^8 + p^7 + 0p^6 \\
 & \qquad \underline{p^8 + 0p^7 + p^6} \\
 & \qquad\quad p^7 + p^6 + p^5 \\
 & \qquad\quad \underline{p^7 + 0p^6 + p^5} \\
 & \qquad\qquad p^6 + 0p^5 + 0p^4 \\
 & \qquad\qquad \underline{p^6 + 0p^5 + p^4} \\
 & \qquad\qquad\quad p^4 + p^3 + 0p^2 \\
 & \qquad\qquad\quad \underline{p^4 + 0p^3 + p^2} \\
 & \qquad\qquad\qquad p^3 + p^2 + 0p \\
 & \qquad\qquad\qquad \underline{p^3 + 0p^2 + p} \\
 & \qquad\qquad\qquad\quad p^2 + p + 0 \\
 & \qquad\qquad\qquad\quad \underline{p^2 + 0p + 1} \\
 & \qquad\qquad\qquad\qquad p + 1
\end{array}$$

From above division, the remainder is,

$$C(p) = p + 1$$

b) To determine check bits

The remainder is $C(p) = p + 1$

Hence check bits, $C = (1\ 1)$

c) To obtain code word

The systematic cyclic code is given as,

$$X = (M : C)$$

$$= \underbrace{1001001010}_{\text{Message bits}} : \underbrace{11}_{\text{Check bits}}$$

iii) To draw encoder for obtaining remainder bits :

The remainder bits are check bits, that are generated by the encoder. The generator polynomial is given as,

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

and $$G(p) = p^2 + g_1 p + 1$$

On comparing above two equations we obtain,

$$g_1 = 0$$

An encoder can be obtained from Fig. 3.3.2 with $q = 2$ and $g_1 = 0$. It is shown below :

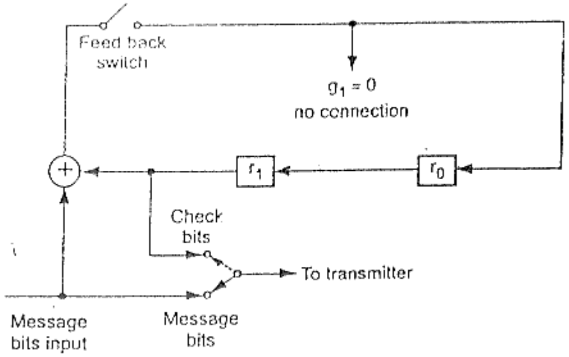


Fig. 3.3.12 Encoder for $G(p) = p^2 + 1$

The shift register $r_1\, r_0$ contains remainder bits i.e. check bits after all 10 bits of message are entered.

iv) Contents of register at decoder :

The syndrome register contains syndrome after all bits of received vector are clocked into the decoder input. Depending upon the received vector, syndrome is calculated.

Example 3.3.16 : *For a systematic (7, 4) cyclic code with generator polynomial $(x^3 + x^2 + 1)$, determine the data vectors transmitted for the following received vectors*

i) 1101101 ii) 0101000 iii) 0001100

using syndrome decoding technique. Compare the technique with maximum likelihood decision rule based decoding.

Solution : This example can be solved through following steps :

i) Determine generator matrix (G).

ii) Determine parity check matrix (H).

iii) Determine decoding table.

iv) Determine syndromes for received vectors and obtain correct transmitted vectors.

i) To obtain generator matrix (G) :

The t^{th} row of the systematic generator matrix is given as,

$$p^{n-t} + R_t(p) = Q_t(p)\, G(p) \quad \text{and} \quad t = 1, 2, \ldots k$$

Here, it is given that $n = 7$, $k = 4$ and $q = n - k = 3$.

We can write above equation as,

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

or $$\frac{p^{n-t}}{G(p)} = Q_t(p) + \frac{R_t(p)}{G(p)} \qquad \ldots (3.3.72)$$

Note that additions in above equations are mod-2. Hence $x \oplus y = z$ is same as $x = y \oplus z$. Here note that $R_t(p)$ is the remainder obtained by dividing p^{n-t}.

a) To obtain polynomial for Row 1 (t = 1)

With $t = 1$, equation 3.3.78 becomes,

$$\frac{p^{n-1}}{G(p)} = Q_1(p) + \frac{R_1(p)}{G(p)}$$

With $n = 7$ and putting for $G(p) = p^3 + p^2 + 1$

$$\frac{p^6}{p^3 + p^2 + 1} = Q_1(p) + \frac{R_1(p)}{p^3 + p^2 + 1}$$