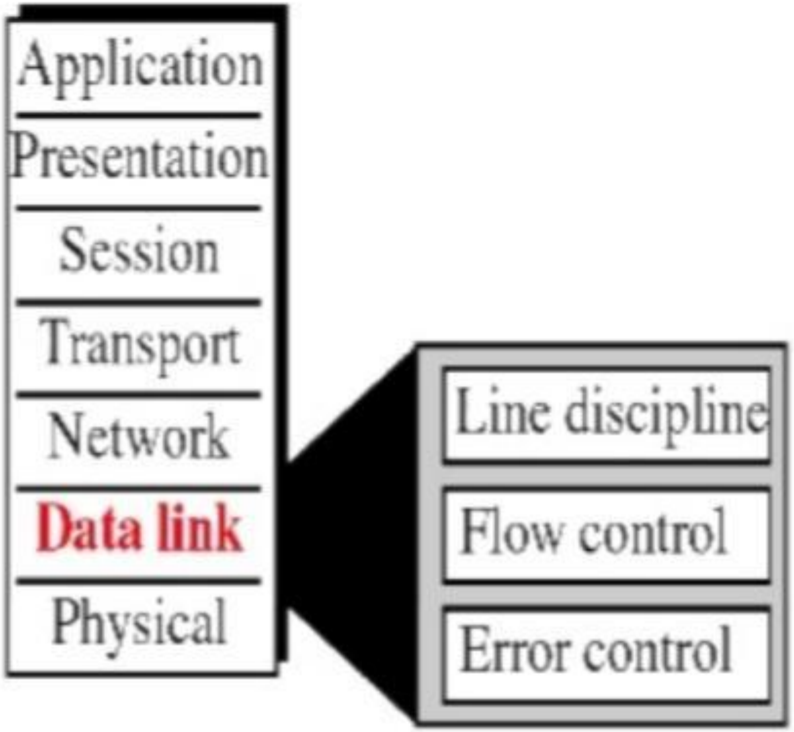
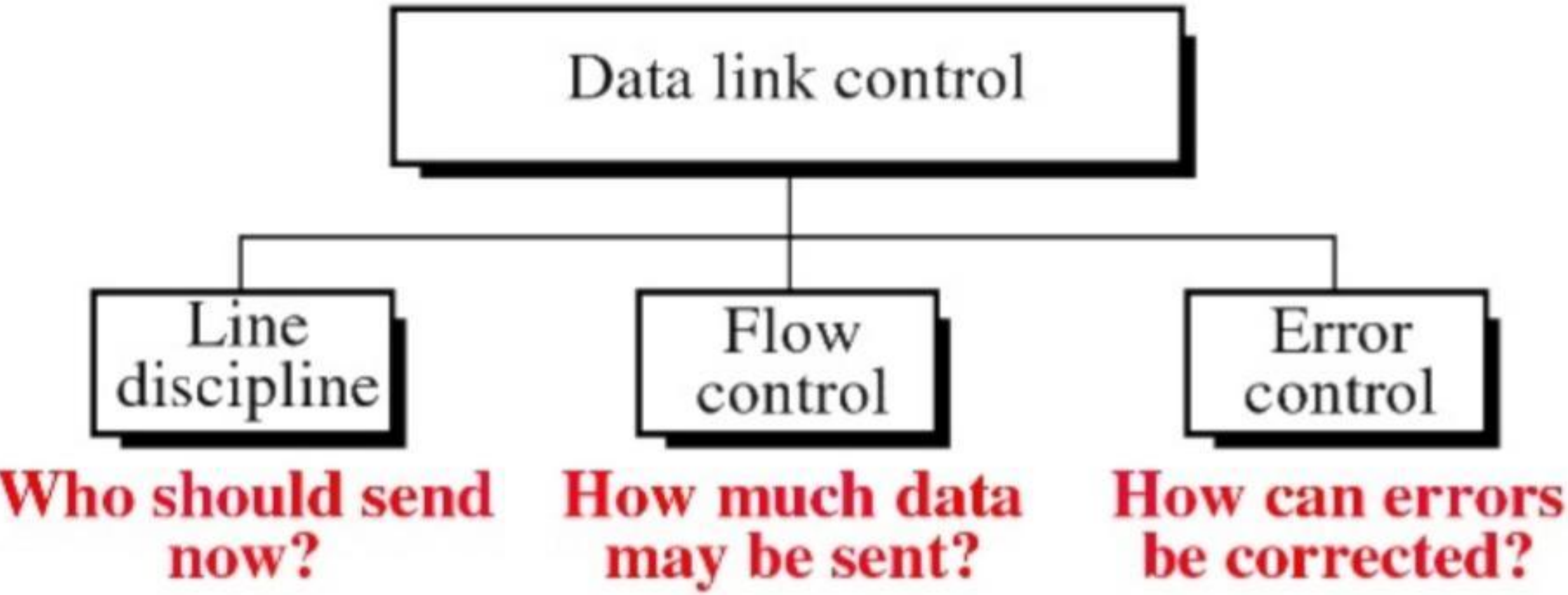
Data Link Layer Design Issues

- Services Provided to the Network Layer
- Framing
- Error Control
- Flow Control



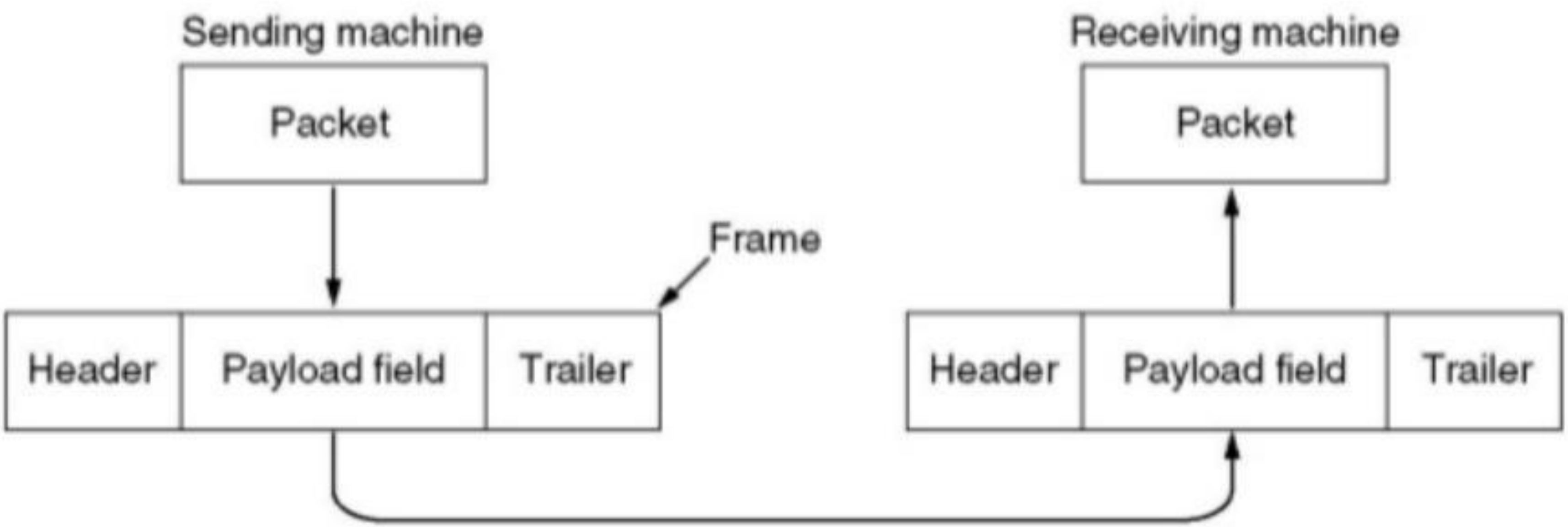

Data link control
Line discipline
Flow control
Error control
Who should send now?
How much data may be sent?
How can errors be corrected?

Functions of the Data Link Layer

- Provide service interface to the network layer
- Dealing with transmission errors
- Regulating data flow
  - Slow receivers not swamped by fast senders

Functions of the Data Link Layer (2)

- Each frame contains a frame header,
- A payload field for holding the packet, and
- A frame trailer



Relationship between packets and frames.

Services Provided to Network Layer

- ❑Unacknowledged connectionless service.
  - ❖ source machine send independent frames to the destination machine without having the destination machine acknowledge them.

- ❑Acknowledged connectionless service.
  - ❖ When this service is offered, there are still no logical connections used, but each frame sent is individually acknowledged. In this way, the sender knows whether a frame has arrived correctly.
- ❑Acknowledged connection-oriented service.

Framing

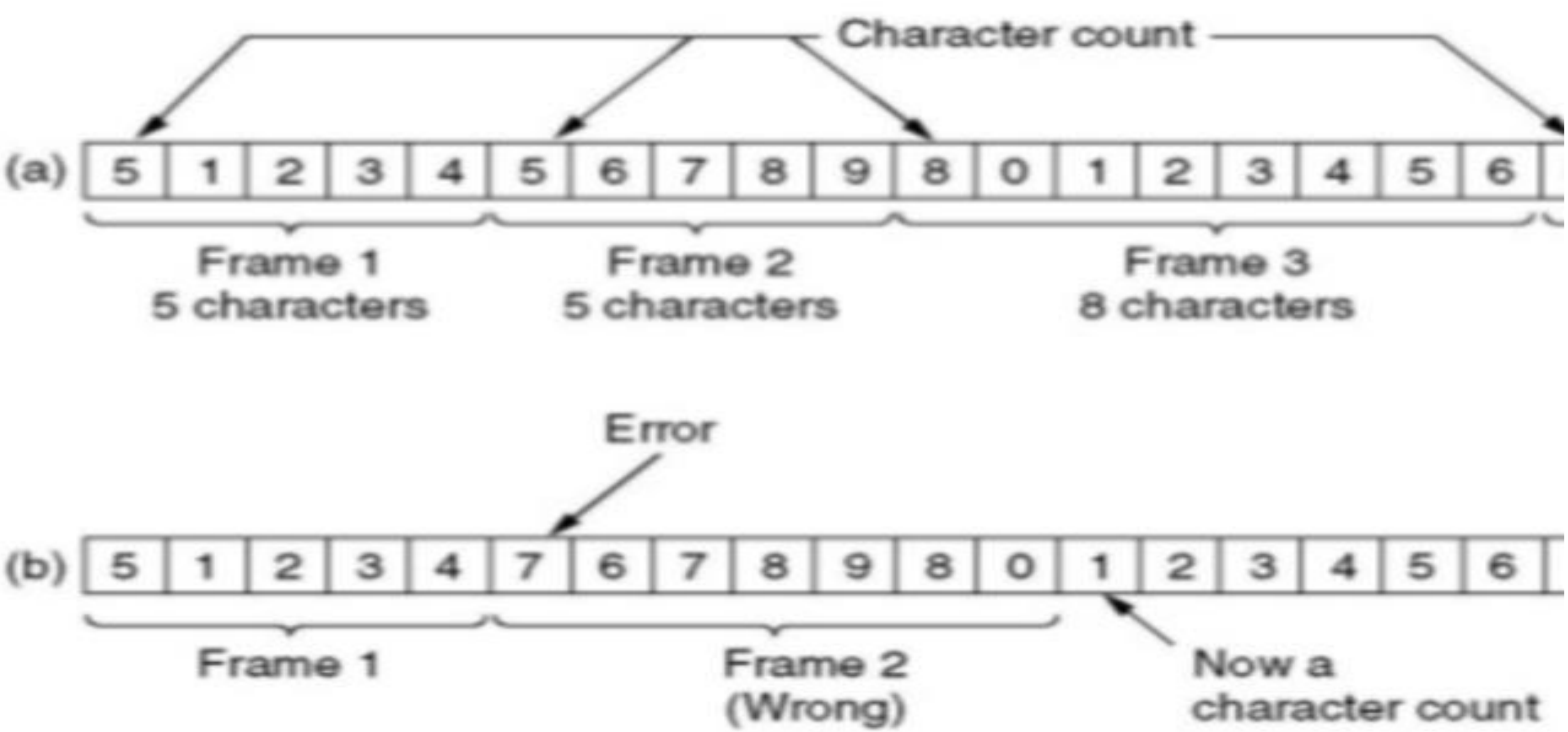
- Data link layer must use the service provided to it by the physical layer.
- This bit stream is not guaranteed to be error free.
- Data link layer to break the bit stream up into discrete frames and compute the checksum for each frame.
- When a frame arrives at the destination, the checksum is recomputed.
- If the newly-computed checksum is different from the one contained in the frame, the data link layer knows that an error has occurred and takes steps to deal with it

Framing

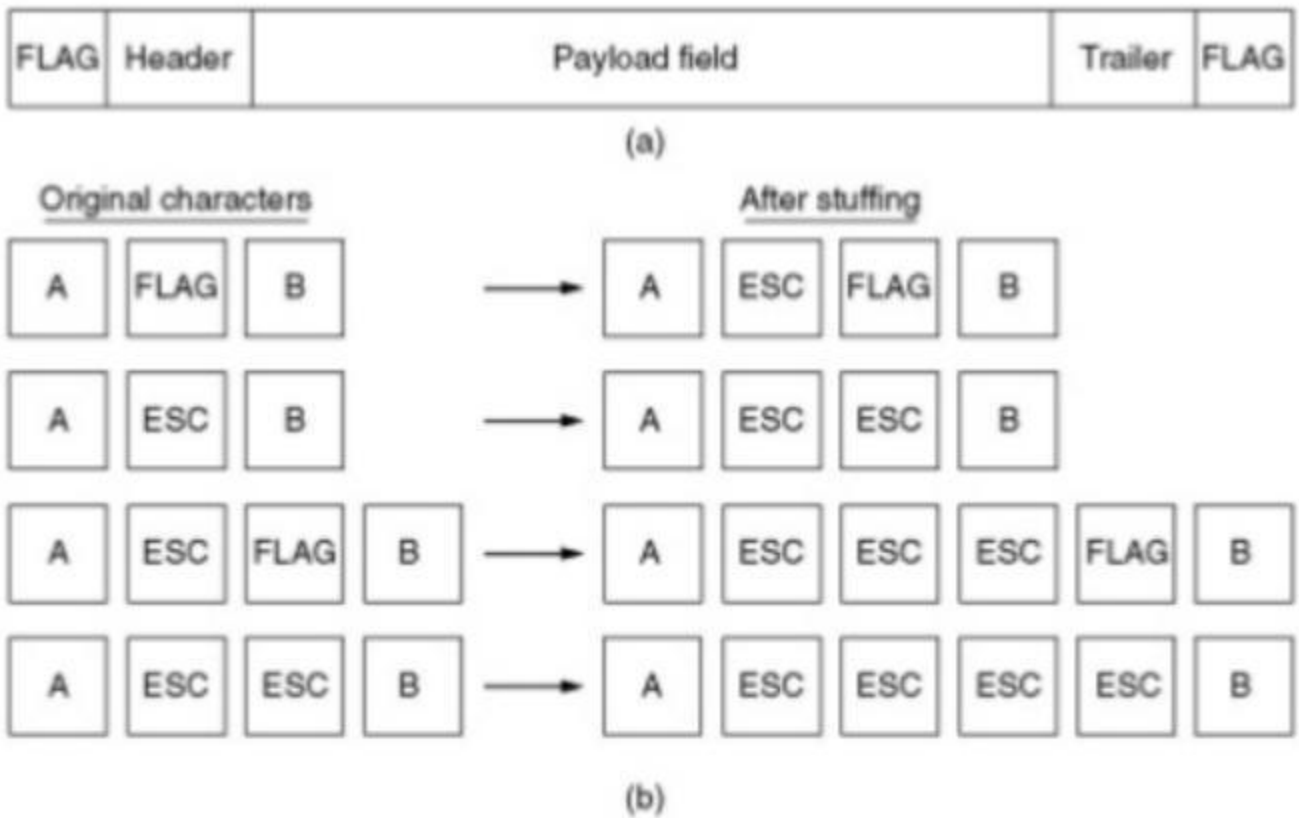
It is too risky to count on timing to mark the start and end of each frame, we will **look at four methods**:

- ❑Character count.
- ❑Flag bytes with byte stuffing.
- ❑Starting and ending flags, with bit stuffing.
- ❑Physical layer coding violations.

Framing Character



Framing Flag Byte



(a) A frame delimited by flag bytes.

(b) Four examples of byte sequences before and after stuffing.

Framing Flag Byte

- A serious problem occurs with this method when binary data, such as object programs or floating-point numbers, are being transmitted.
- It may easily happen that the flag byte's bit pattern occurs in the data.
- This situation will usually interfere with the framing.
- One way to solve this problem is to have the sender's data link layer insert a special escape byte (ESC) just before each "accidental" flag byte in the data.
- The data link layer on the receiving end removes the escape byte before the data are given to the network layer.
- This technique is called byte **stuffing or character stuffing**.
- Thus, a framing flag byte can be distinguished from one in the data by the absence or presence of an escape byte before it.

Framing Bit Stuffing

(a) 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0

(b) 0 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 0 0 1 0

Stuffed bits

(c) 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0

Bit stuffing

(a) The original data.

(b) The data as they appear on the line.

(c) The data as they are stored in receiver's memory after destuffing.

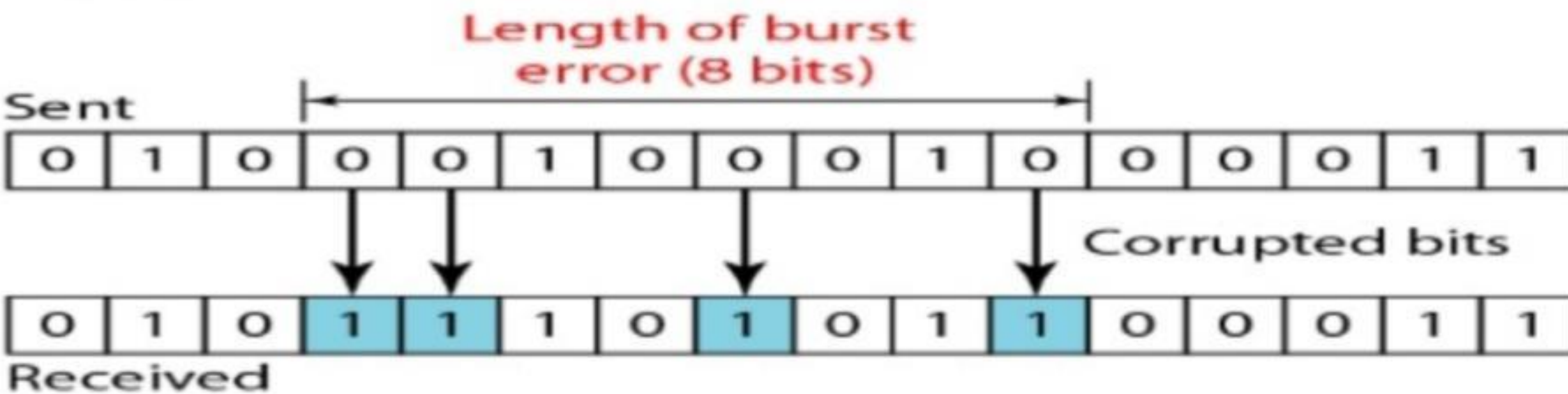
Physical layer coding violations

This method of framing is applicable only to the networks in which the encoding on the physical medium contains some redundancy. The manchester coding is generally used. Normally, a 1 bit is encoded into 10 pair and 0 bit is encoded into a 01 pair.

Types of Error Single Bit Errors

In a single bit error, only 1 bit in the data unit changes:

Burst Errors



Error Detection and Correction

- Error-Correcting Codes
- Error-Detecting Codes

- Computer data always sent in blocks of bits.
- Suppose that the block size is 1000 bits and the error rate is 0.001 per bit.
- If errors were independent, most blocks would contain an error.
- If the errors came in bursts of 100 however, only one or two blocks in 100 would be affected, on average

Definitions

Codeword: n bit

Data bits(k)	Parity bits(n-k)

Code rate : ratio of number of message bits to total bits in code. Code rate = k/n <1.

Hamming weight: number of nonzero elements in the codeword.

Hamming distance: number of locations in which their respective elements differ.

1 1 0 1 0 1 0 0

0 1 0 1 1 1 1 0 : hamming distance is 3.

Minimum hamming distance dmin of linear block code may be defined as smallest hamming distance between any pair of code vectors in the code.

Role of min hamming distance dmin in error detection and error correction:

If dmin >= (s+1) : detect up to s errors

dmin>= (2t+1): correct up to t errors

dmin >=(t+s+1): correct up to t and detect s>t errors

Error detection methods

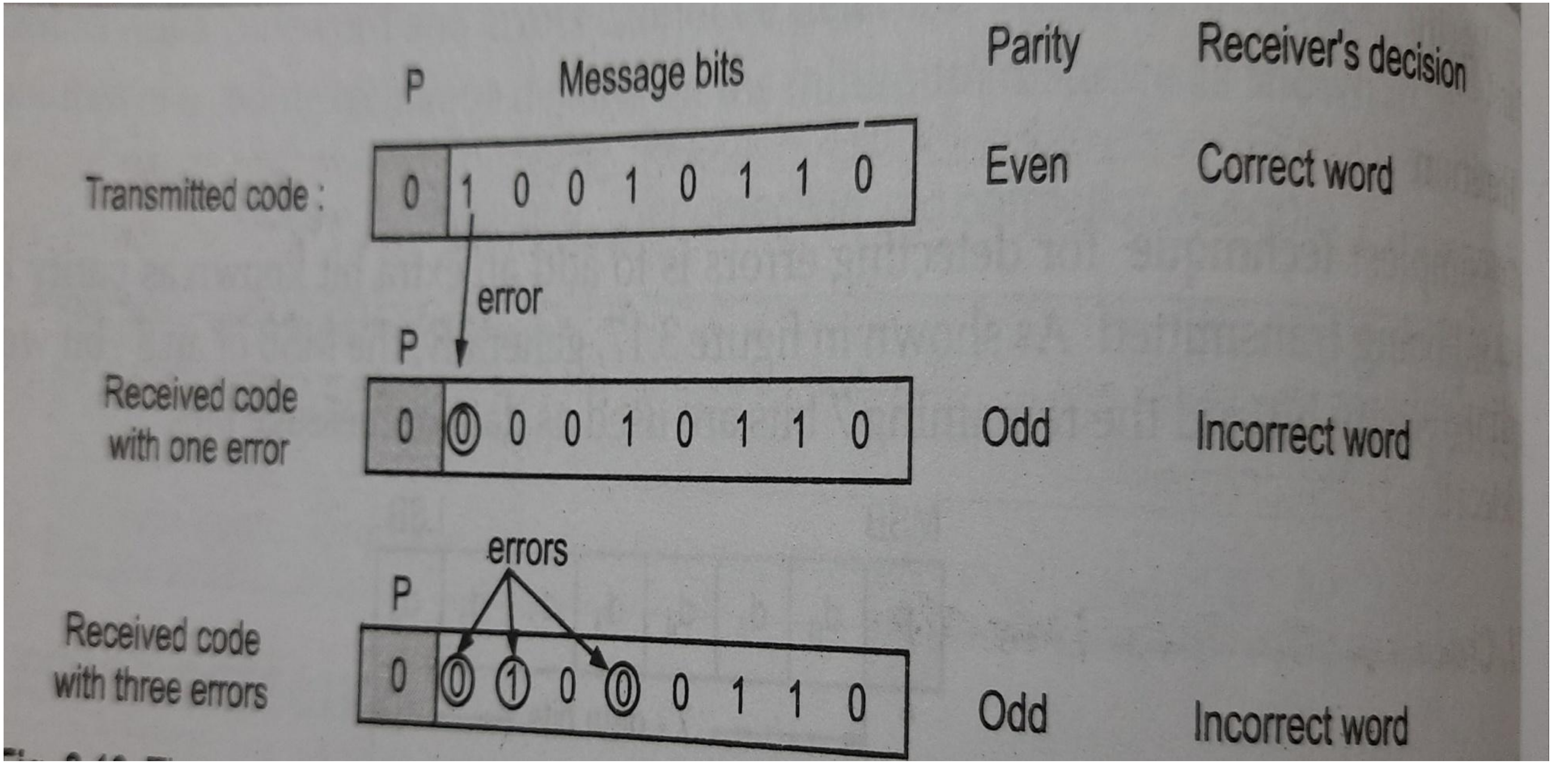
1: parity checking

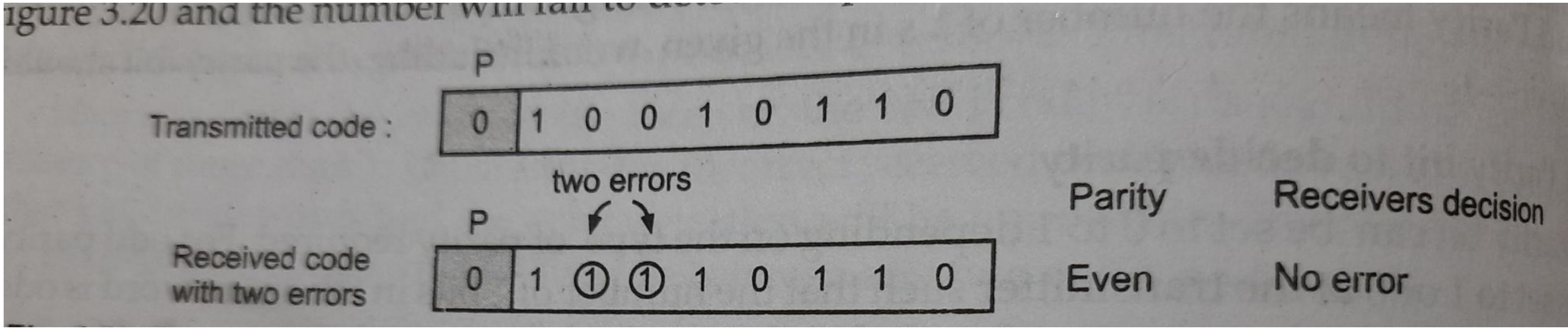
2:checksum

3: cyclic redundancy check.

Parity: simplest technique for detecting error is to add an extra bit known as parity bit to each word being transmitted. Generally the MSB of an 8 bit word is used as parity bit and remaining 7 bits are used as data or message bits. It may be even or odd. even parity means the number of 1's in the given word including parity bit should be even . Odd parity means the number of 1's including parity should be odd.

Parity checking

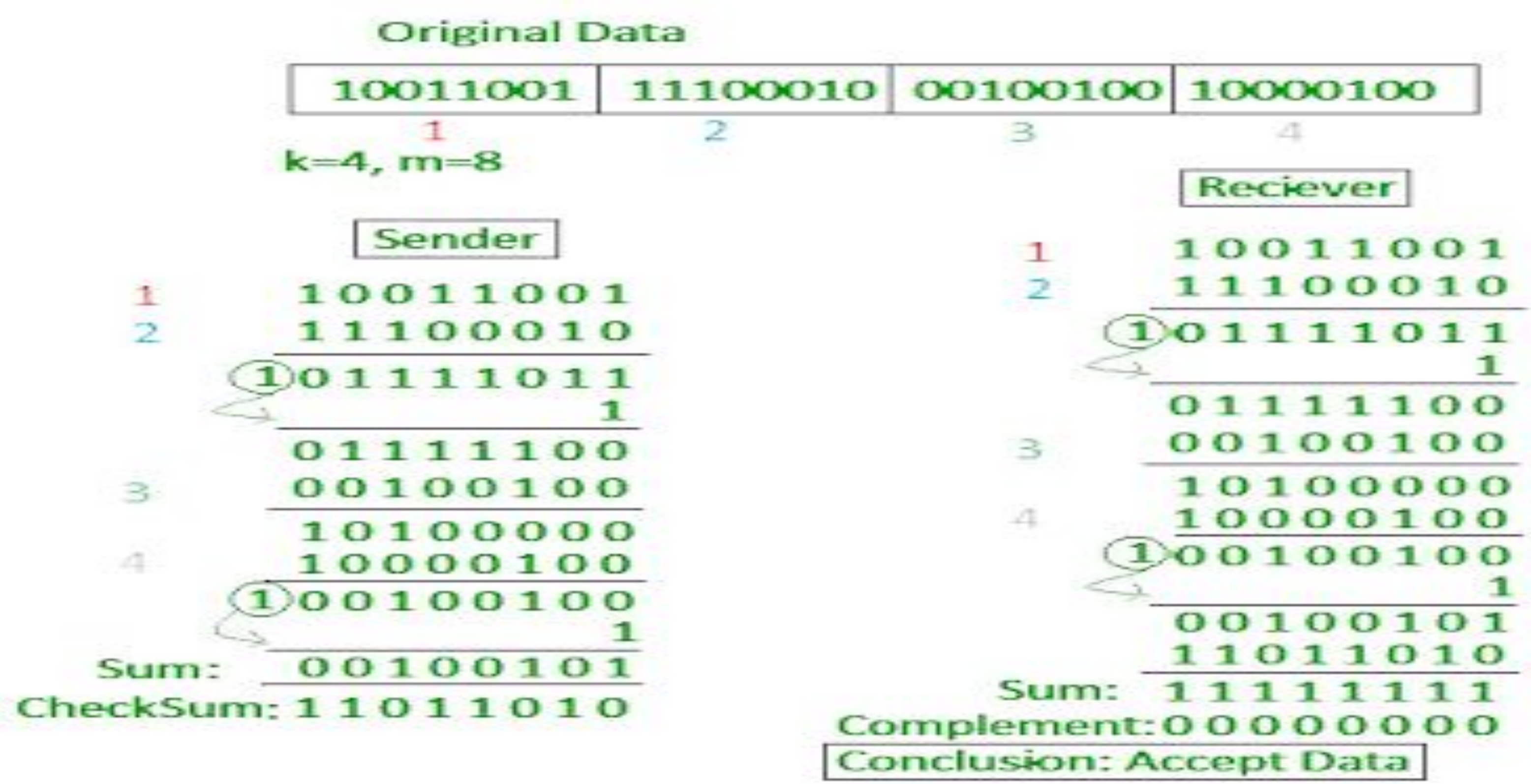




Conclusions: 1) any even number of errors in received word will not change the Parity. Therefore, even number of errors will be unnoticed.
2) Error will be detected only if odd number of bits get errored.

Limitations : 1) it is not suitable for even number of errors.
2) It can not reveal the position of erroneous bit.

Checksum



With a block of data bytes the checksum byte is also transmitted. The checksum is regenerated at receiver. then it is compared with transmitted one. If both are Identical then there is no error. If they are different then the errors are present in The block of data. sometimes 2's compliment of checksum is transmitted then Receiver accumulate the data blocks with checksum then result will be zero if there is no erropr.

Two dimensional parity check

In this parity bits are generated for each row and column of block of data. The two sets of parity bits are known as

1) Longitudinal redundancy check bits (LRC)
2) Vertical redundancy check bits(VRC)

bit in error

LRC bits (even parity)

	byte 1	byte 2	byte 3						LRC
b_1	1	1	1	0	(0)	0	1	0	1 ← Wrong parity
b_2	1	1	0	0	0	0	0	1	1
b_3	0	1	1	0	1	1	1	0	1
Data block → b_4	0	1	1	0	0	0	0	0	0
b_5	0	0	0	1	1	1	0	1	0
b_6	0	0	0	0	0	0	0	0	0
b_7	1	1	1	1	1	1	1	1	0
VRC bits (even parity) →	1	1	0	0	0	1	1	1	1

Wrong parity

First bit of the fifth byte is in error

Even a single error in any bit will result in a non correct LRC in one of the rows and an incorrect VRC in one of the columns. The bit which is common to the row and column is the bit in error. But if there is multiple errors in rows and columns can only be detected.

Cyclic redundancy check (CRC)

It is the type of polynomial code in which a bit string is represented in form of polynomials with coefficient of 0 and 1 only. It uses modulo-2 arithmetic i.e. addition and subtraction are identical to EXOR. For CRC code, the sender and receiver should agree upon a generator polynomial G(x). Codeword can be generator for a given message polynomial M(x) with the help of long division CRC remainder is append at the end of data unit. The resulting data unit after adding CRC remainder becomes exactly divisible . At the receiver, the data unit is divided by the same binary number. But non zero remainder at receiver indicates presence of errors in the received data.

Ex: $X^7 + X^5 + 1$: 10100001

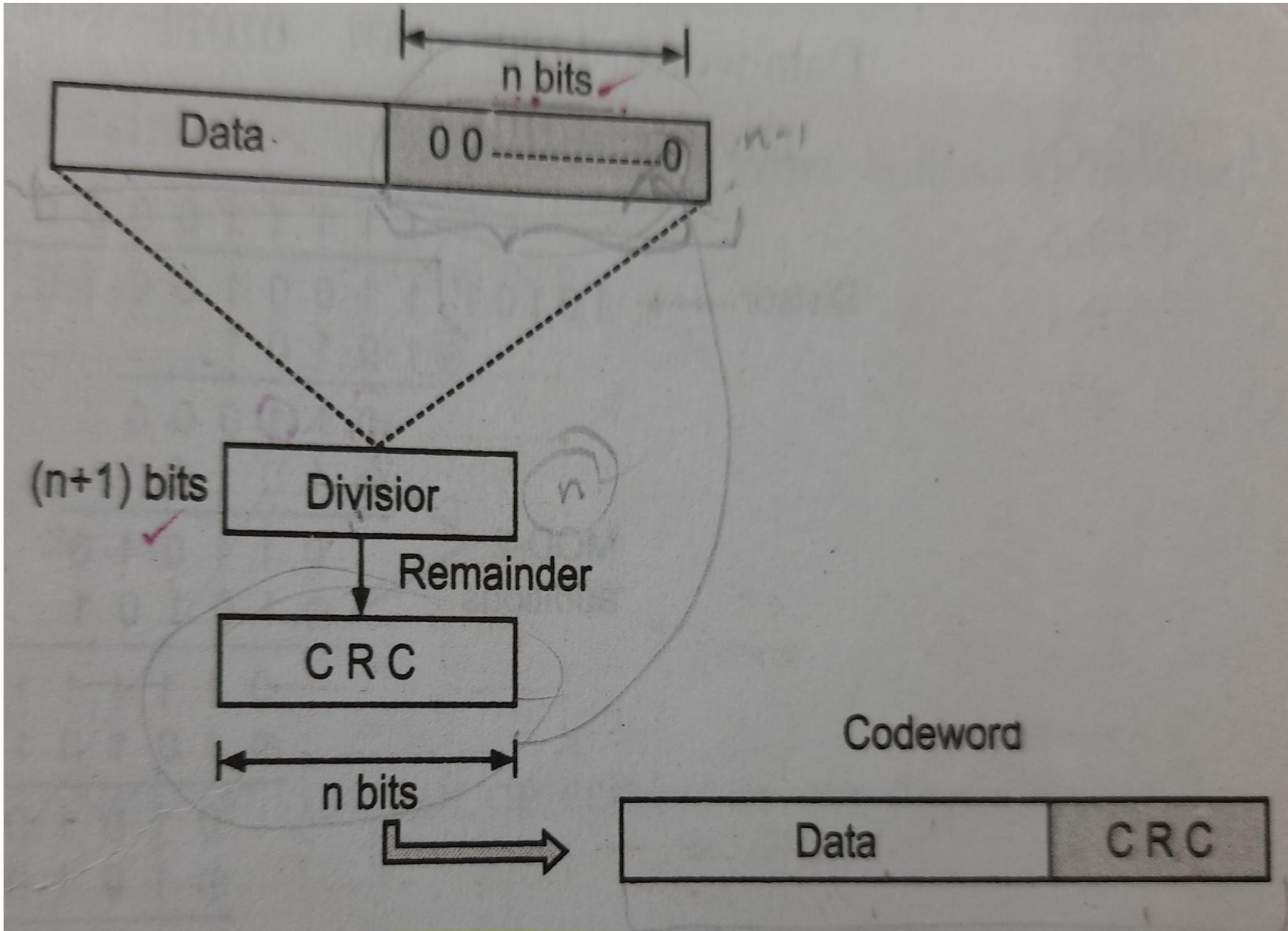


Fig. 3.24 CRC generator

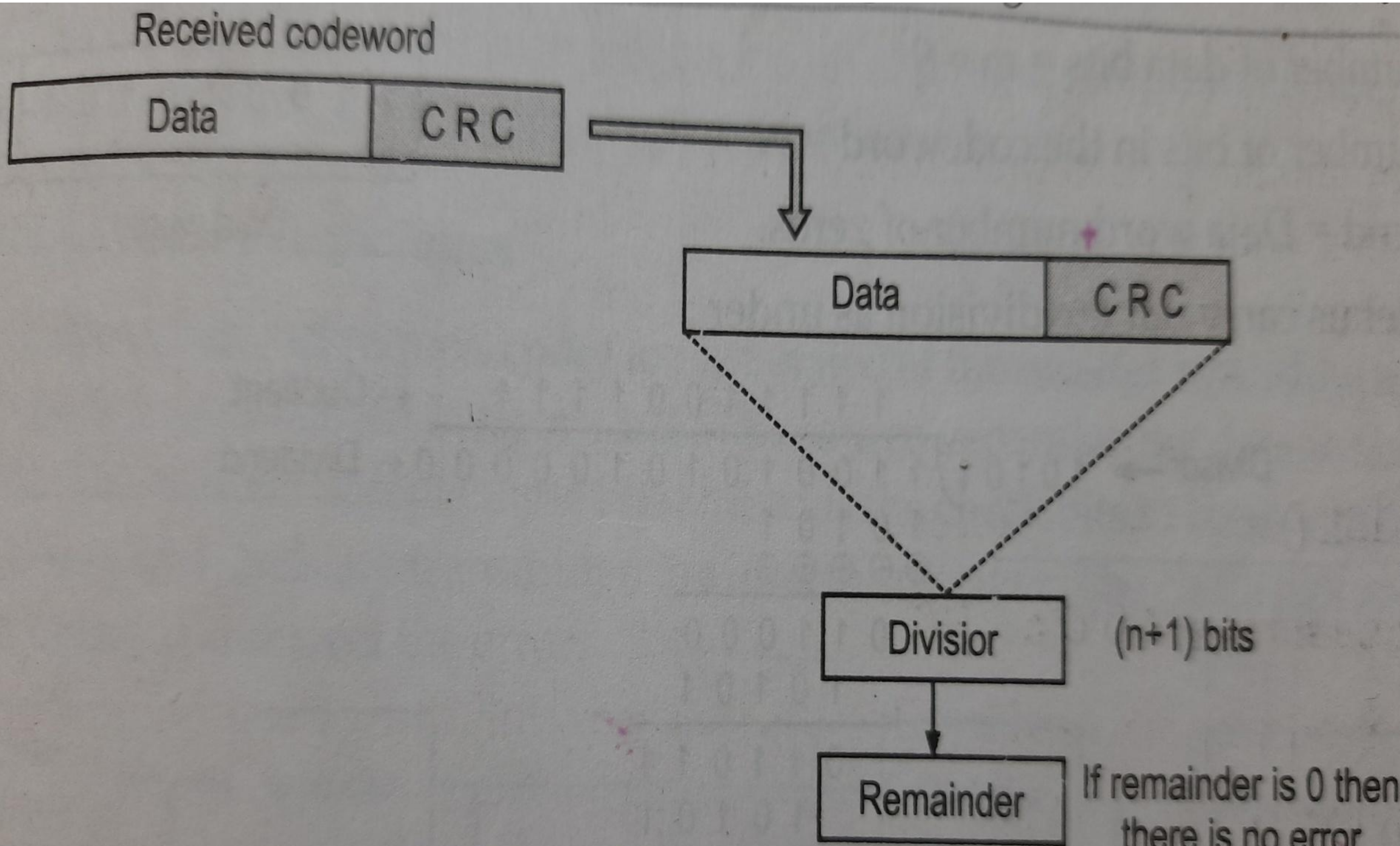


Fig. 3.25 CRC checker

Generate the CRC code for the data word of 1100 10101 . The divisor is 10101.

olution : Given that data word

Divisor : 10101

The number of data bits = m = 9

The number of bits in the codeword = N

Dividend = Data word+number of zeros.

Now, let us carry out the division as under :

1 1 0 0 1 0 1 0 1	0 0 0 0 0
Data word	5 additional zeros

```
                      1 1 1 1 1 0 0 1 1 1 1        ← Quotient
Divisor → 1 0 1 0 1 ) 1 1 0 0 1 0 1 0 1 0 0 0 0 0  ← Dividend
                      1 0 1 0 1
                      ⊕ ⊕ ⊕ ⊕ ⊕
                      0 1 1 0 0 0
                        1 0 1 0 1
                        0 1 1 0 1 1
                          1 0 1 0 1
                          0 1 1 1 0 0
                            1 0 1 0 1
                              1 0 0 1 1
                              1 0 1 0 1
                                  1 1 0 0 0
                                  1 0 1 0 1
                                    1 1 0 1 0
                                    1 0 1 0 1
                                      1 1 1 1 0
                                      1 0 1 0 1
                                      0 0 1 1 0  ← Remainder
```

Codeword

In CRC, the required codeword is obtained by writing the data word followed by the remainder. Thus, we have

$$
\begin{array}{r}
11001010100000 \\
110 \\
\hline
\text{Codeword} = \quad 11001010100110
\end{array}
$$

Undetected Errors in CRC

(*i*) CRC cannot detect all types of errors.

(*ii*) The probability of error detection and the types of detectable errors depends on the choice of divisor.

Assignment - 2

Q1. A bit string 01111011111101111110, needs to be transmitted at the data link layer. What is the string actually transmitted after bit stuffing ?

Q2. Find the checksum of the following message 10110001, 10101011,00110101, 10100001.

Q3. If the frame is 110101011 and generator is $x^4 + x + 1$, what would be the transmitted frame in CRC.

Q4. A bit stream 10011101 is transmitted using the CRC method. The generator polynomial is $x^3 + 1$. show the actual bit string transmitted . Suppose the third bit from left is inverted during transmission. show that this error is detected at the receiver end.

Error correcting techniques

There are two completely different approaches for the error control.

1) The Automatic request for retransmission (ARQ) Technique.
2) Forward error correction (FEC) technique.

In ARQ system, the receiver can request for the retransmission of the complete or a part of message if it finds some error in received message. This requires an additional channel called feedback channel to send the receiver’s request for retransmission.

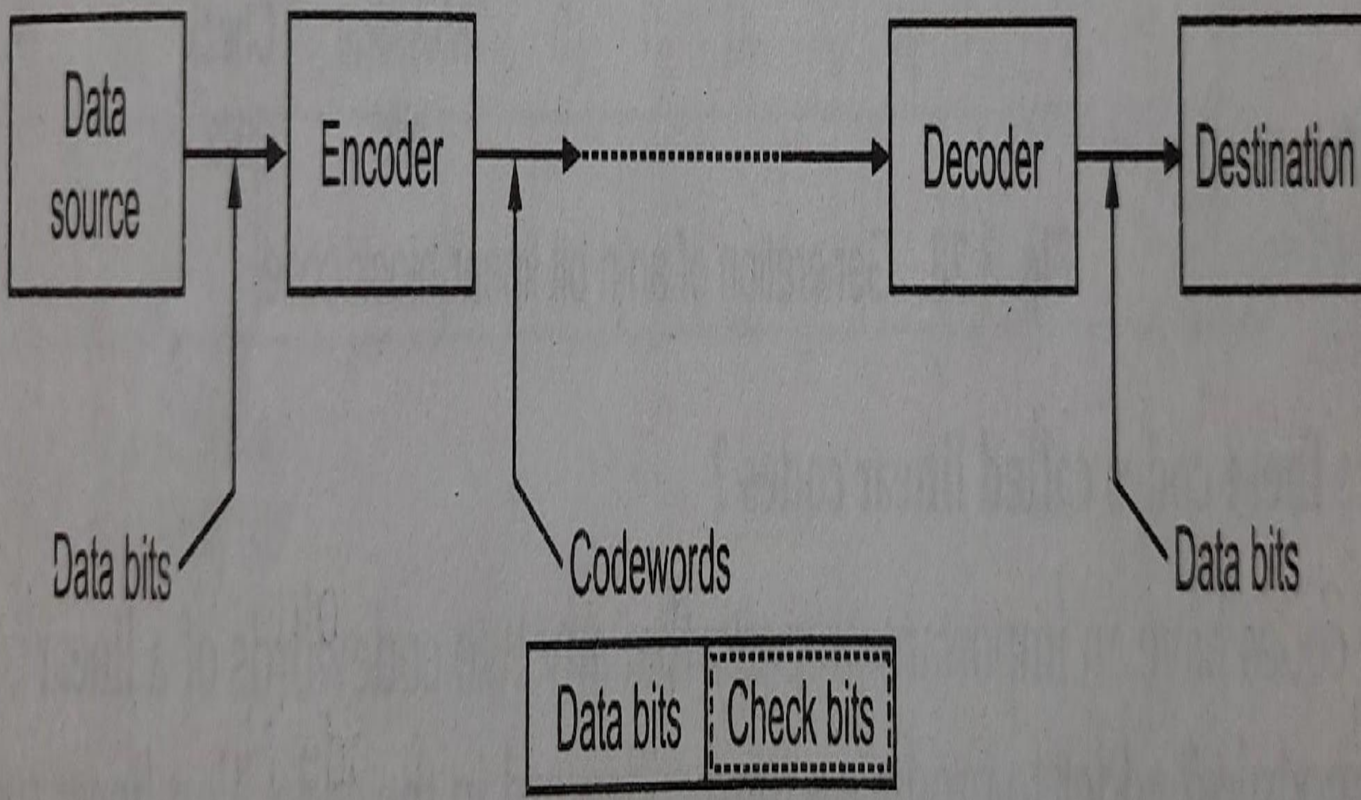
In FEC ,there is no such feedback path and request for retransmission. The coding techniques are related to FEC.

Forward error corection

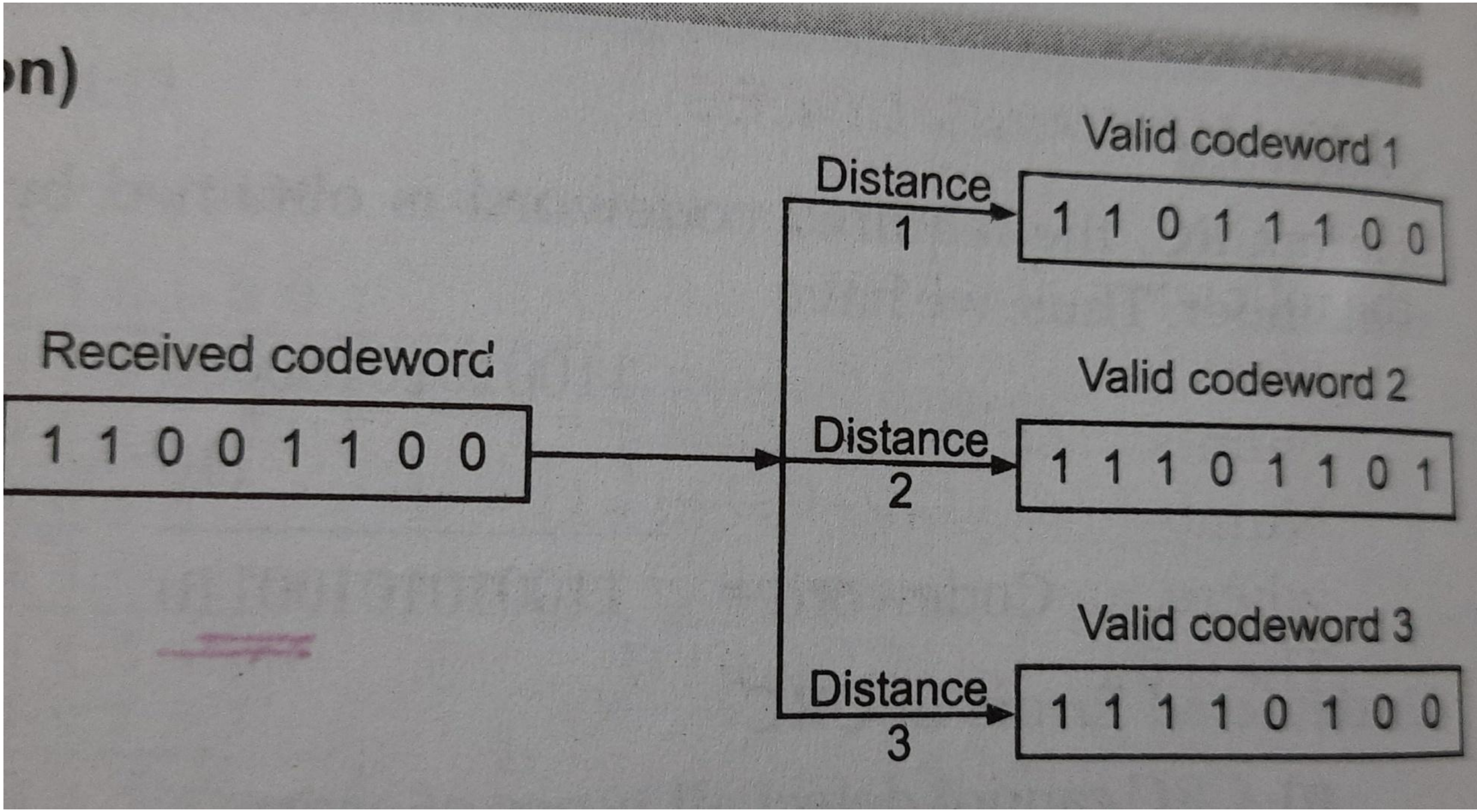
Codes are generated at transmitter by adding a group of parity bits or check bits. source generates the data in form of binary symbols. The encoder accepts these bits and adds the check bits to them to produce the code words. These code words are transmitted towards the receiver. The check bits are used by the decoder to detect and correct the errors.

The decoder separates out the data and check bits. It uses the parity bits to detect and correct errors if they are present in the received code words. The data bits are then applied to the destination.

the decoder to detect and correct the errors.



ERROR CORRECTION TECHNIQUE (FEC)



In FEC, the receiver searches for the most likely correct code word. When an Error is detected, the distance between the received invalid codeword and all the possible code words is measured. The nearest valid code word (the one having minimum distance) is the most likely correct version of the received Code word.

Linear block code

To generate an (n , k) block code, the channel encoder accepts the information in successive k bit blocks. At the end of each such block ,it adds (n - k) check bits. As these bits do not contain any information, they are called as redundant bits. These check bits are related to the k bits. The n bit code word is thus produced.

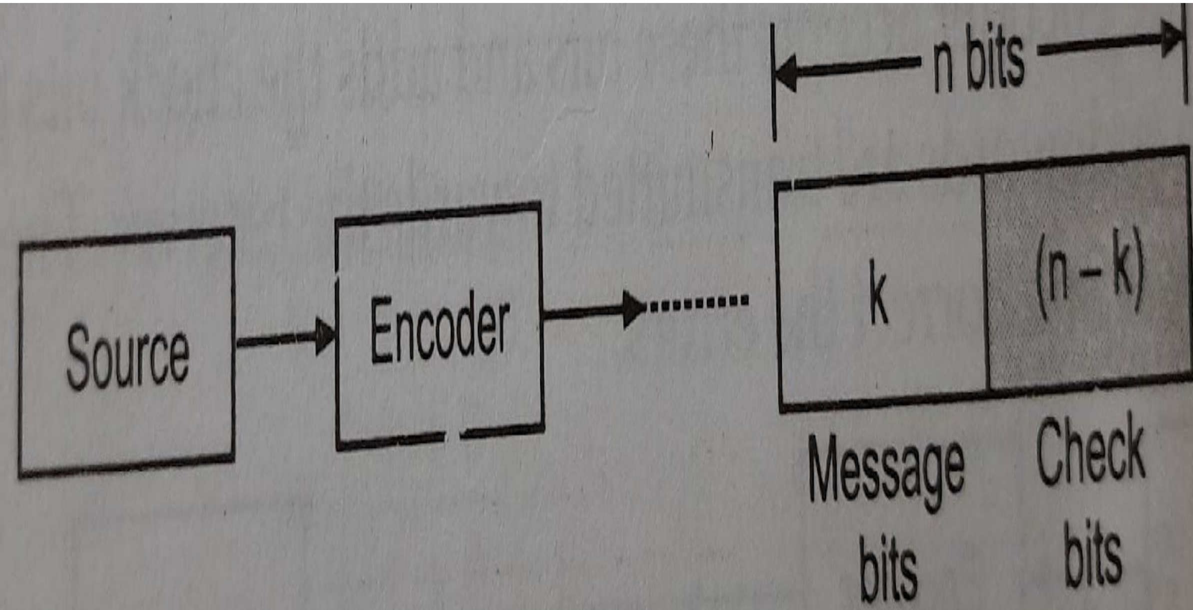
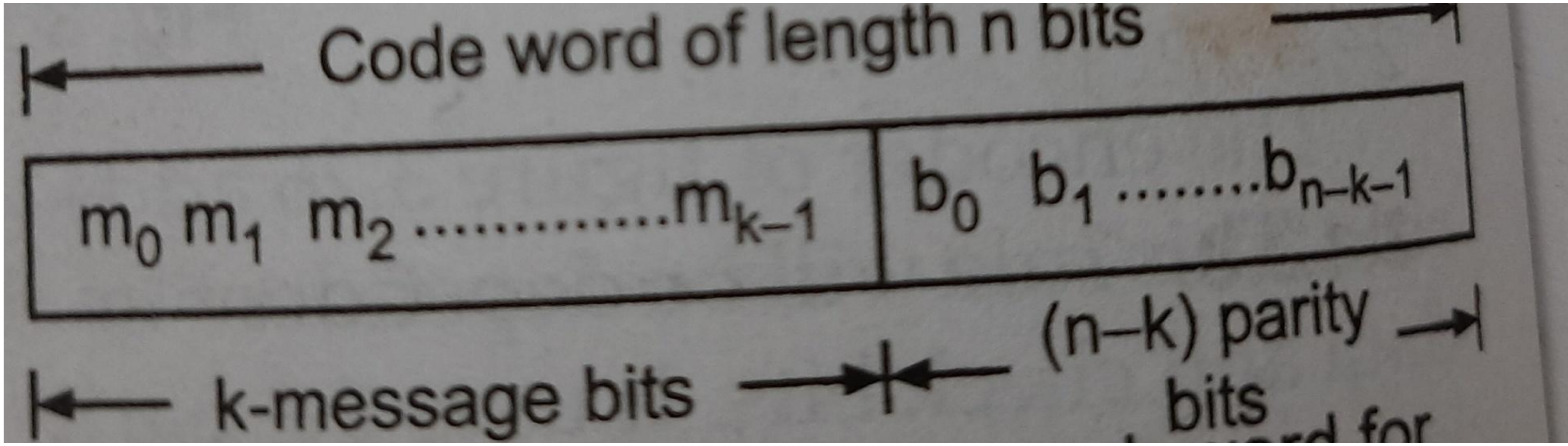


Fig. 3.28 Generation of an n bit linear block code



If message bits then all check bits then it is called systematic(symmetric) linear block code. If they are in between message bits then it is A systematic Asymmetric)linear block code.

Hamming code

Hamming codes are linear block codes. The family of (n, k) Hamming codes for $m \geq 3$ is defined by the following equations:

Block length : $n = 2^m - 1$

Number of message bits : $k = 2^m - m - 1$ ... (3.3)

Number of parity bits : $(n - k) = m$

where, minimum number of parity bits is 3.

The minimum distance $d_{min} = 3$.

$$\text{The code rate or code efficiency} = \frac{k}{n} = \frac{2^m - m - 1}{2^m - 1} = 1 - \frac{m}{2^m - 1} \quad \text{... (3.4)}$$

If $m >> 1$, then, code rate $r \cong 1$.

Error detection and correction capabilities of hamming code

(*i*) The number of errors that can be detected per word = 2

Since $d_{min} \geq (s+1)$, so, $3 \geq s+1$

so, $s \leq 2$

(*ii*) The number of errors that can be corrected per word = 1

Since $d_{min} \geq (2t+1)$, so, $3 \geq (2t+1)$

so, $t \leq 2$

Therefore, with $d_{min} = 3$, it is possible to detect upto 2 errors and it is possible to correct upto only 1 error.

Hamming code structure

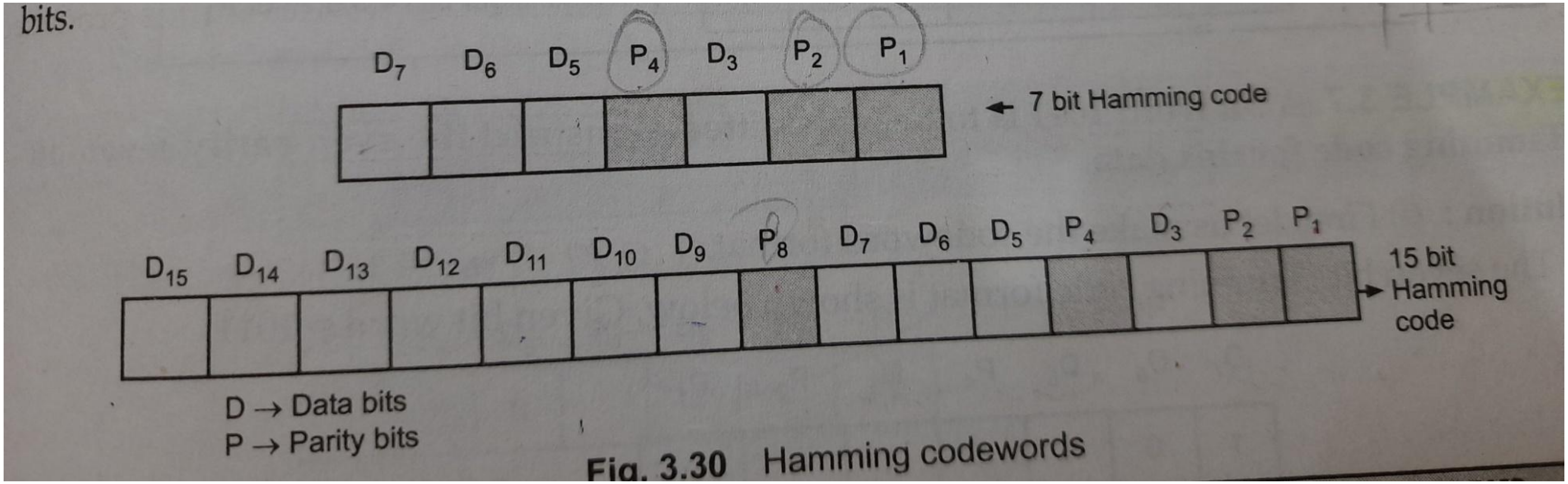
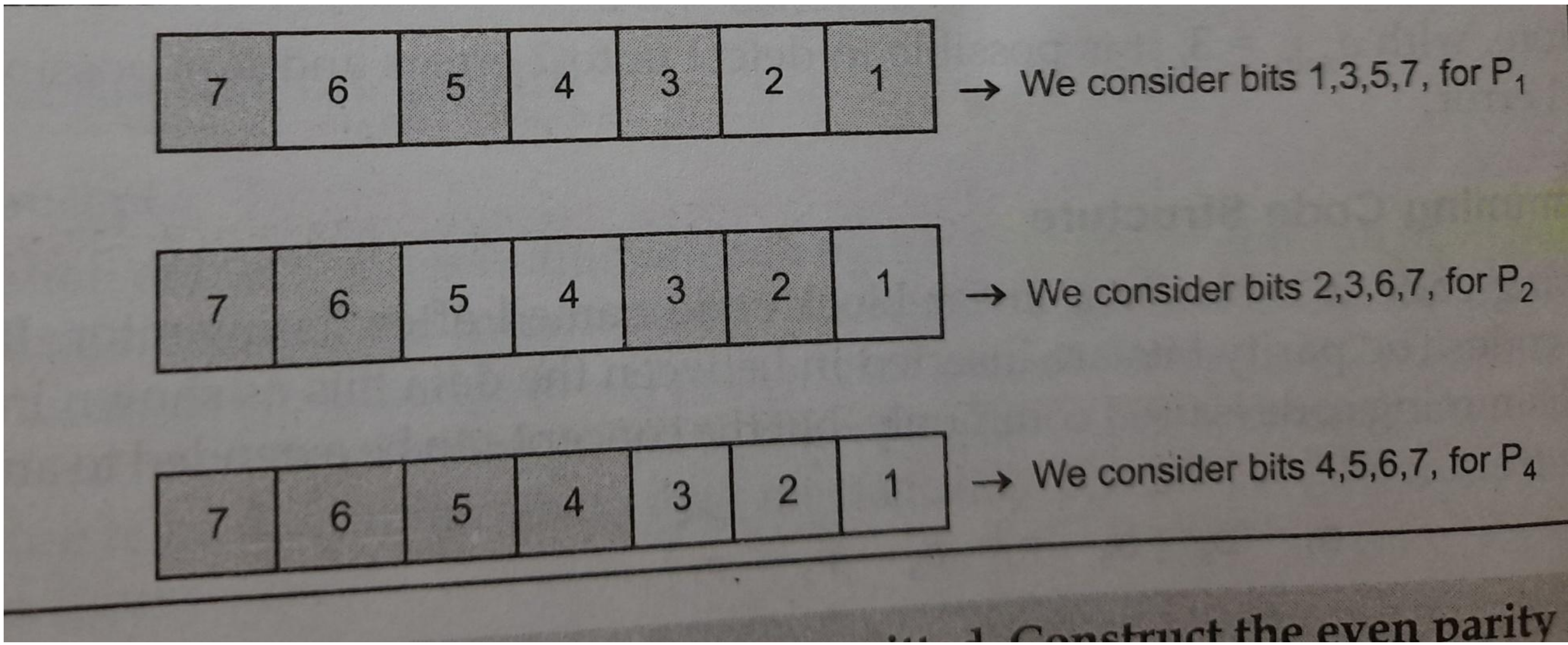


Fig. 3.30 Hamming codewords

7 – bit hamming code



A bit word 0101 is to be transmitted. Construct the even parity hamming code for this data.

Solution : (*i*) First, let us make the codeward format.

D_7	D_6	D_5	P_4	D_3	P_2	P_1
0	1	0		1		

(*ii*) Then, we select P_1 for $P_1\ D_3\ D_5\ D_7$.

D_7	D_6	D_5	P_4	D_3	P_2	P_1
0	1	0		1		1

→ $P_1 = 1$ for $P_1\ D_3\ D_5\ D_7 = 1\,1\,0\,0$

(*iii*) Then, we select P_2 for $P_2\ D_3\ D_6\ D_7$.

D_7	D_6	D_5	P_4	D_3	P_2	P_1
0	1	0		1	0	1

$P_2 = 0$ for $P_2\ D_3\ D_6\ D_7 = 0\,1\,1\,0$

(*iv*) Then, we select P_4.

D_7	D_6	D_5	P_4	D_3	D_2	D_1
0	1	0	1	1	0	1

Set $P_4 = 1$ to have $P_4\ D_5\ D_6\ D_7 = 1\,0\,1\,0$

So code will be 0101101

Parity bits positions

P1 (1,3,5,7,9,11,13,15)

P2 (2,3,6,7,10,11,14,15)

P4 (4,5,6,7,12,13,14,15)

P8 (8,9,10,11,12,13,14,15)