Error Detection And Correction in Computer Network

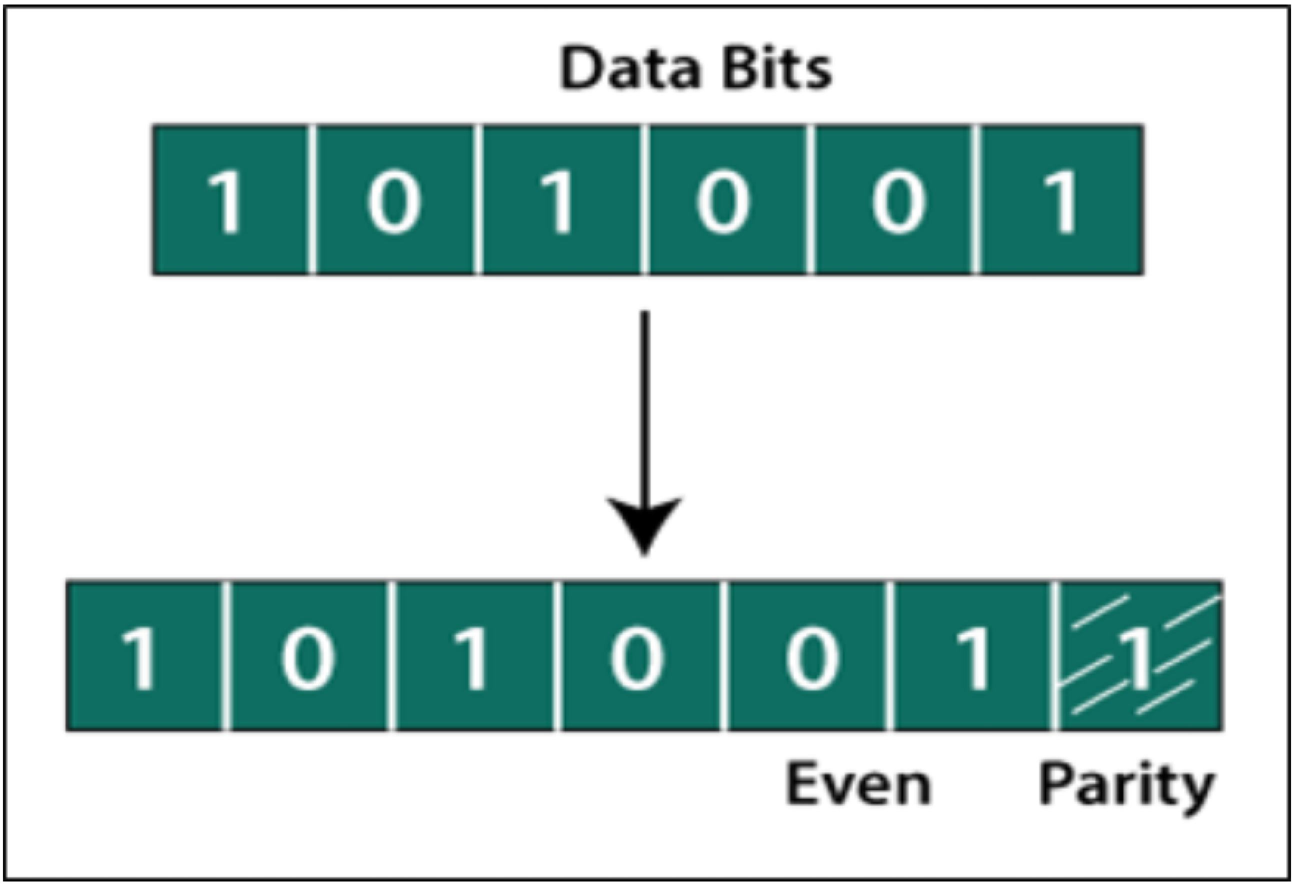
Error Detection Techniques:

1. Simple Parity Check:

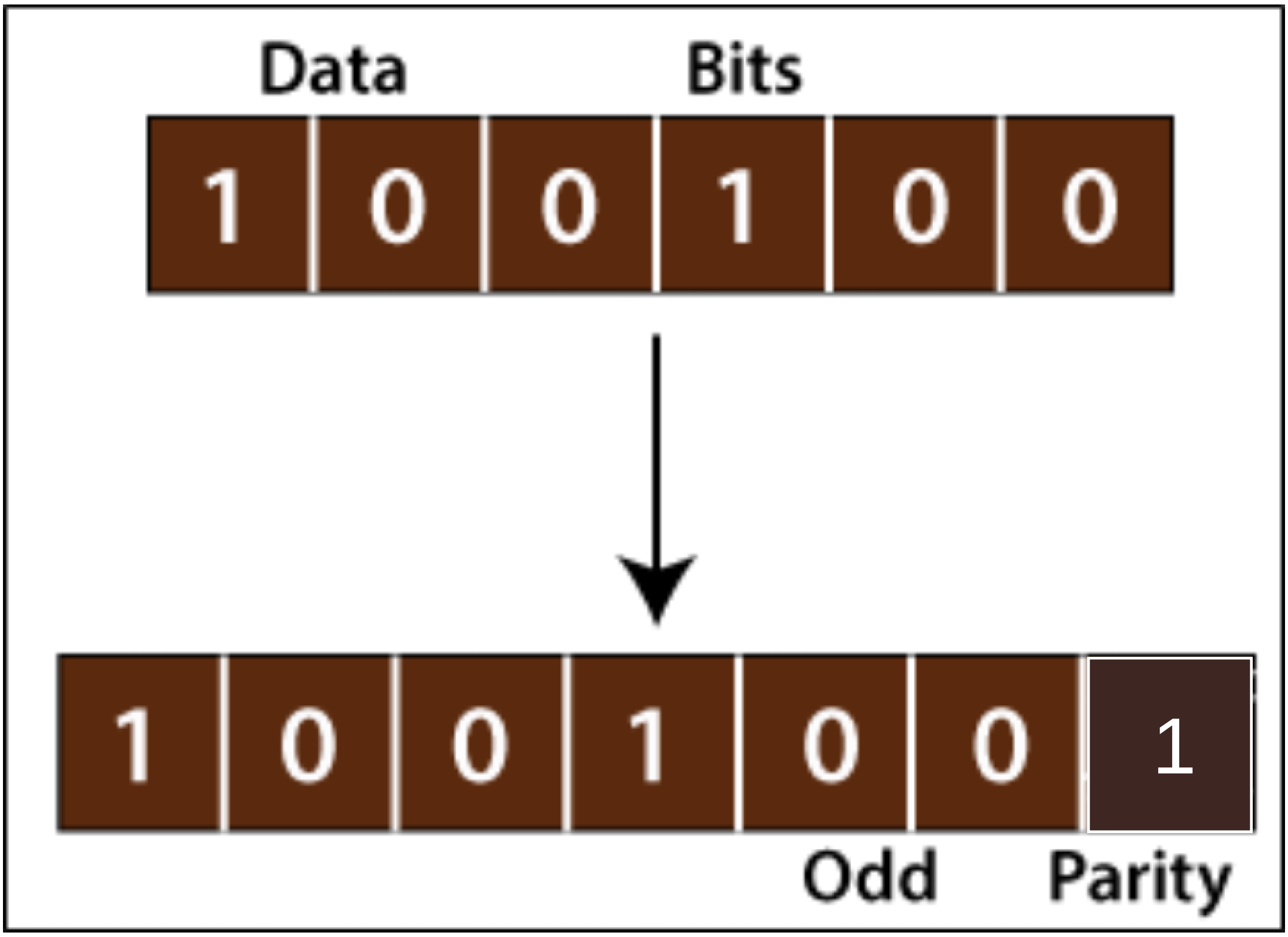
- One extra bit is transmitted in addition to the original bits to make the number of 1s even in the case of even parity or odd in the case of odd parity.
- While creating a frame, the sender counts the amount of 1s in it. If even parity is utilised and the number of 1s is even, one bit with the value 0 is added. In this manner, the number of 1s remains even. If the number of 1s is odd, a value 1 is added to make it even.

- The receiver just counts how many 1s are in a frame. If the number of 1s is even and even parity is utilised, the frame is regarded as uncorrupted and approved. Even if the number of 1s is odd and odd parity is utilised, the frame is not damaged.
- The receiver can identify a single bit flip in transit by counting the number of 1s. However, when more than one bit is incorrect, it is extremely difficult for the receiver to identify the problem.

Even parity: If the number of 1's is even in the frame, 0 is added in the frame. The even-parity example shown in the figure below.



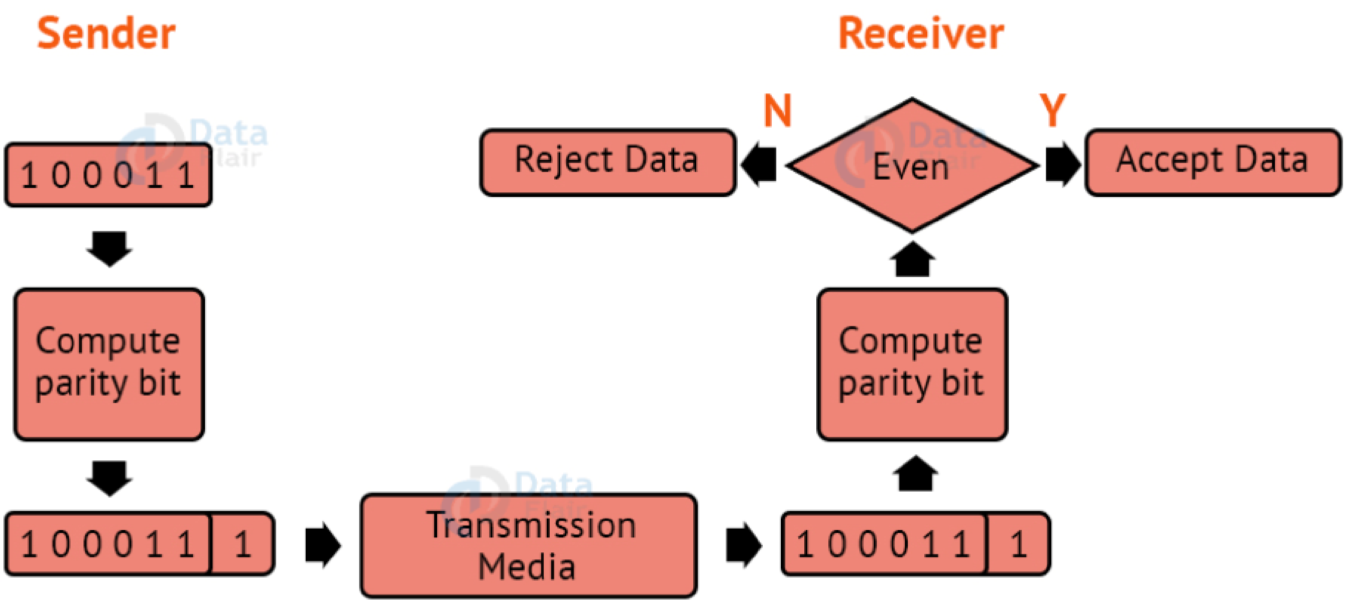
Odd parity: If the number of 1's is odd in the frame, 0 is added in the frame. The odd-parity example shown in the figure below.



Messages with even parity and odd parity

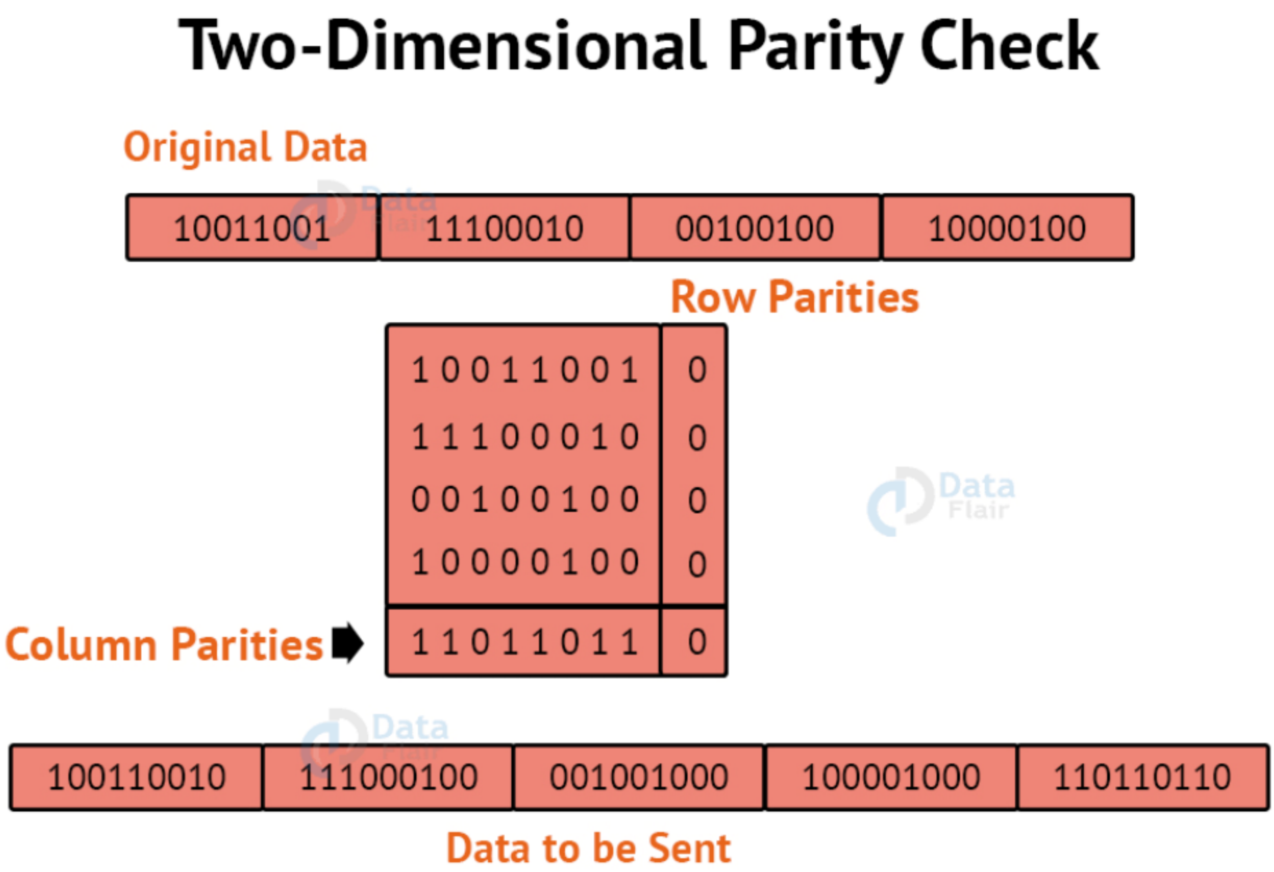
3 bit data			Message with even parity		Message with odd parity	
A	B	C	Message	Parity	Message	Parity
0	0	0	000	0	000	1
0	0	1	001	1	001	0
0	1	0	010	1	010	0
0	1	1	011	0	011	1
1	0	0	100	1	100	0
1	0	1	101	0	101	1
1	1	0	110	0	110	1
1	1	1	111	1	111	0

Example of Simple Even Parity Check



2. Two-Dimensional Parity Check:

For each row, parity check bits are calculated, which is identical to a basic parity check bit. For each column, parity check bits are computed and transmitted together with the data. These are compared with the parity bits calculated on the received data at the receiving end.



3. Checksum:

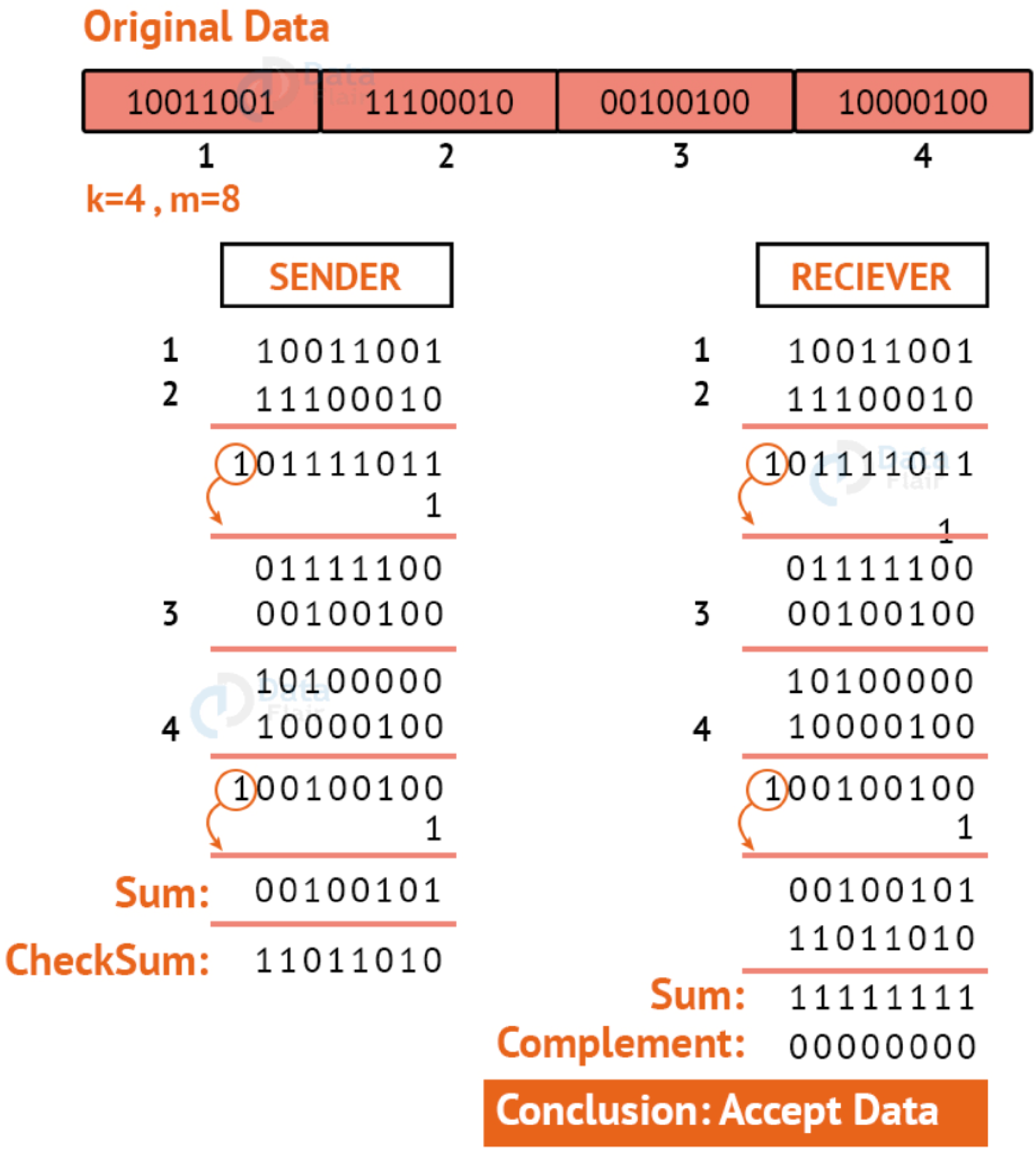
The data is split into k segments of m bits each in the checksum error detection technique.

To get the total, the segments are summed at the sender's end using 1's complement arithmetic. To obtain the checksum, a complement of the sum is taken.

The checksum segment is sent with the data segments.

To obtain the total, all received segments are summed using 1’s complement arithmetic at the receiver’s end. The sum is then calculated.

If the result is 0, the data is accepted; otherwise, it is rejected.


Original Data
10011001
11100010
00100100
10000100
1
2
3
4
k=4 , m=8
SENDER
1 10011001
2 11100010
101111011
1
01111100
3 00100100
10100000
4 10000100
100100100
1
Sum: 00100101
CheckSum: 11011010
RECIEVER
1 10011001
2 11100010
101111011
1
01111100
3 00100100
10100000
4 10000100
100100100
1
00100101
11011010
Sum: 11111111
Complement: 00000000
Conclusion: Accept Data

Cyclic Redundancy Check-

- Cyclic Redundancy Check (CRC) is an error detection method.
- It is based on binary division.

CRC Generator-

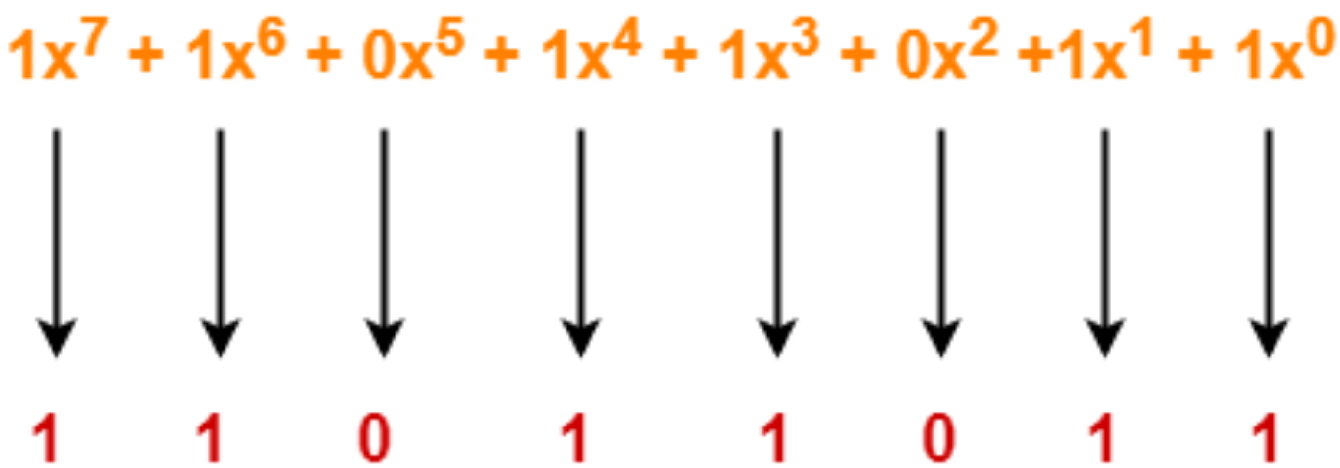
- CRC generator is an algebraic polynomial represented as a bit pattern.
- Bit pattern is obtained from the CRC generator using the following rule-

The power of each term gives the position of the bit and the coefficient gives the value of the bit.

Example-

Consider the CRC generator is $x^7 + x^6 + x^4 + x^3 + x + 1$.

The corresponding binary pattern is obtained as-



Thus, for the given CRC generator, the corresponding binary pattern is 11011011.

Steps Involved-

Error detection using CRC technique involves the following steps-

Step-01: Calculation Of CRC At Sender Side-

At sender side,

- A string of n 0’s is appended to the data unit to be transmitted.
- Here, n is one less than the number of bits in CRC generator.
- Binary division is performed of the resultant string with the CRC generator.
- After division, the remainder so obtained is called as **CRC**.
- It may be noted that CRC also consists of n bits.

Step-02: Appending CRC To Data Unit-

At sender side,

- The CRC is obtained after the binary division.
- The string of n 0's appended to the data unit earlier is replaced by the CRC remainder.

Step-03: Transmission To Receiver-

- The newly formed code word (Original data + CRC) is transmitted to the receiver.

Step-04: Checking at Receiver Side-

At receiver side,

- The transmitted code word is received.
- The received code word is divided with the same CRC generator.
- On division, the remainder so obtained is checked.

The following two cases are possible-

Case-01: Remainder = 0

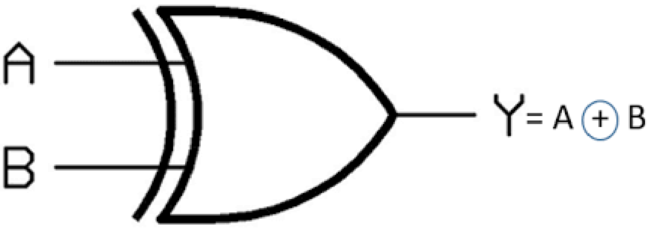
If the remainder is zero,

- Receiver assumes that no error occurred in the data during the transmission.
- Receiver accepts the data.

Case-02: Remainder ≠ 0

If the remainder is non-zero,

- Receiver assumes that some error occurred in the data during the transmission.
- Receiver rejects the data and asks the sender for retransmission.



EX-OR gate symbol

A	**B**	**$Y=A \oplus B$**
0	0	0
0	1	1
1	0	1
1	1	0

EX-OR gate truth table

PRACTICE PROBLEMS BASED ON CYCLIC REDUNDANCY CHECK (CRC)-

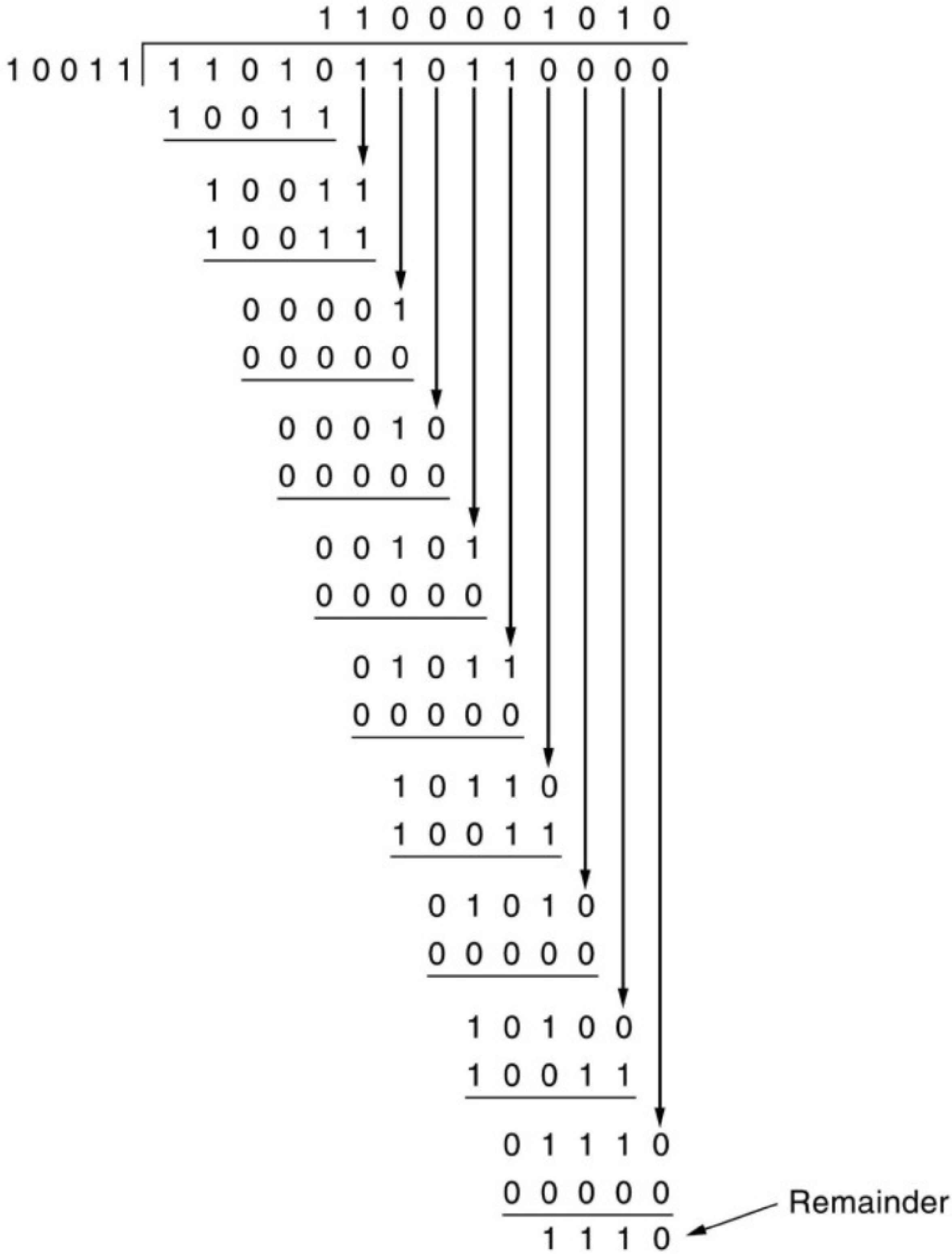
Problem-01:

A bit stream 1101011011 is transmitted using the standard CRC method. The generator polynomial is x^4+x+1. What is the actual bit string transmitted?

Solution-

- The generator polynomial $G(x) = x^4 + x + 1$ is encoded as 10011.
- Clearly, the generator polynomial consists of 5 bits.
- So, a string of 4 zeroes is appended to the bit stream to be transmitted.
- The resulting bit stream is 1101011011**0000**.

Now, the binary division is performed as-



From here, CRC = 1110.

Now,

From here, CRC = 1110.

Now,

- The code word to be transmitted is obtained by replacing the last 4 zeroes of 11010110110000 with the CRC.
- Thus, the code word transmitted to the receiver = 1101011011**1110**.

Problem-02:

A bit stream 10011101 is transmitted using the standard CRC method. The generator polynomial is x^3+1.

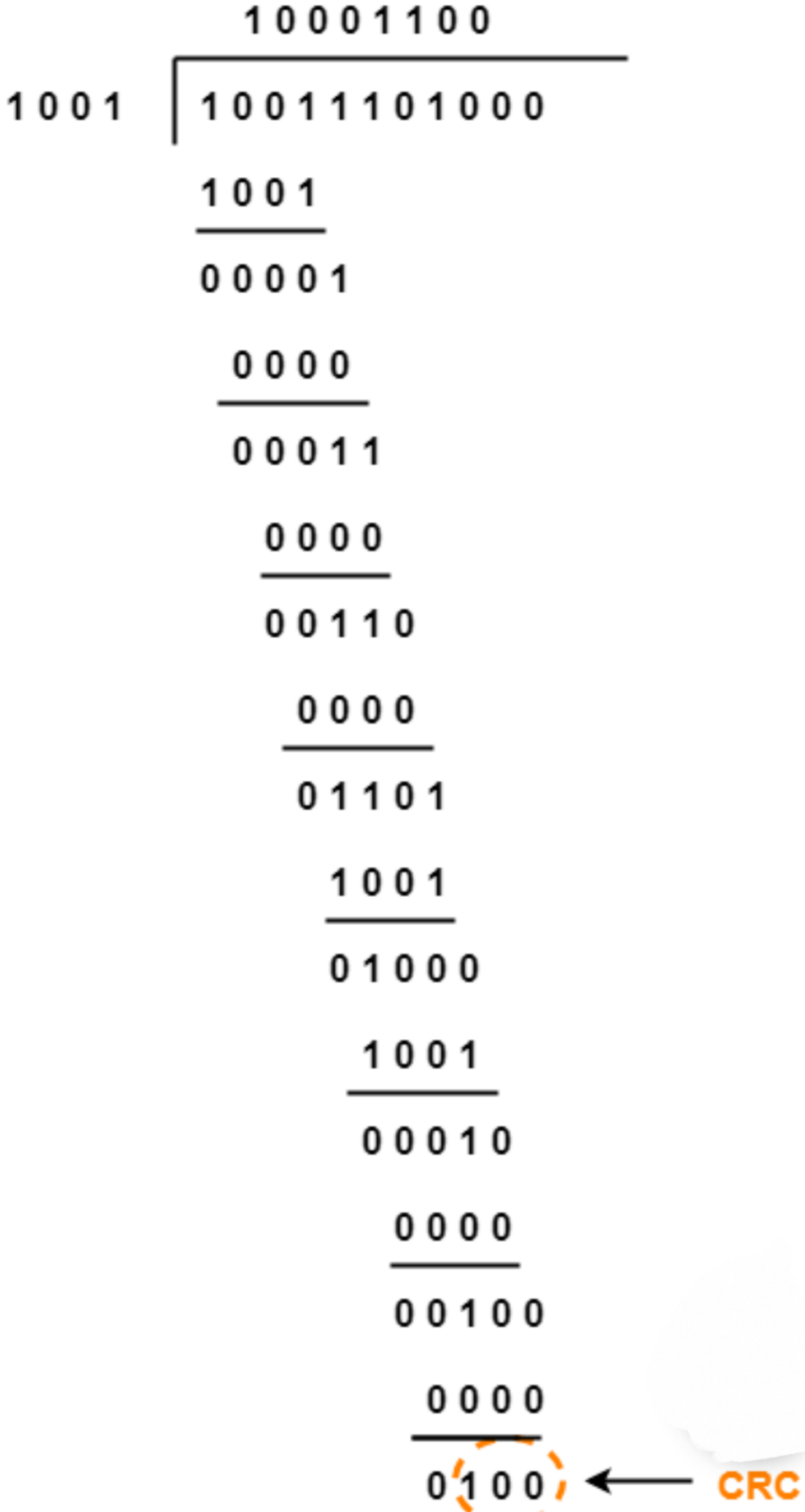
1. What is the actual bit string transmitted?
2. Suppose the third bit from the left is inverted during transmission. How will receiver detect this error?

Solution-

Part-01:

- The generator polynomial $G(x) = x^3 + 1$ is encoded as 1001.
- Clearly, the generator polynomial consists of 4 bits.
- So, a string of 3 zeroes is appended to the bit stream to be transmitted.
- The resulting bit stream is 10011101**000**.

Now, the binary division is performed as-


1 0 0 0 1 1 0 0
1 0 0 1
1 0 0 1 1 1 0 1 0 0 0
1 0 0 1
0 0 0 0 1
0 0 0 0
0 0 0 1 1
0 0 0 0
0 0 1 1 0
0 0 0 0
0 1 1 0 1
1 0 0 1
0 1 0 0 0
1 0 0 1
0 0 0 1 0
0 0 0 0
0 0 1 0 0
0 0 0 0
0 1 0 0
CRC

From here, CRC = 100.

Now,

- The code word to be transmitted is obtained by replacing the last 3 zeroes of 10011101**000** with the CRC.
- Thus, the code word transmitted to the receiver = 10011101**100**.

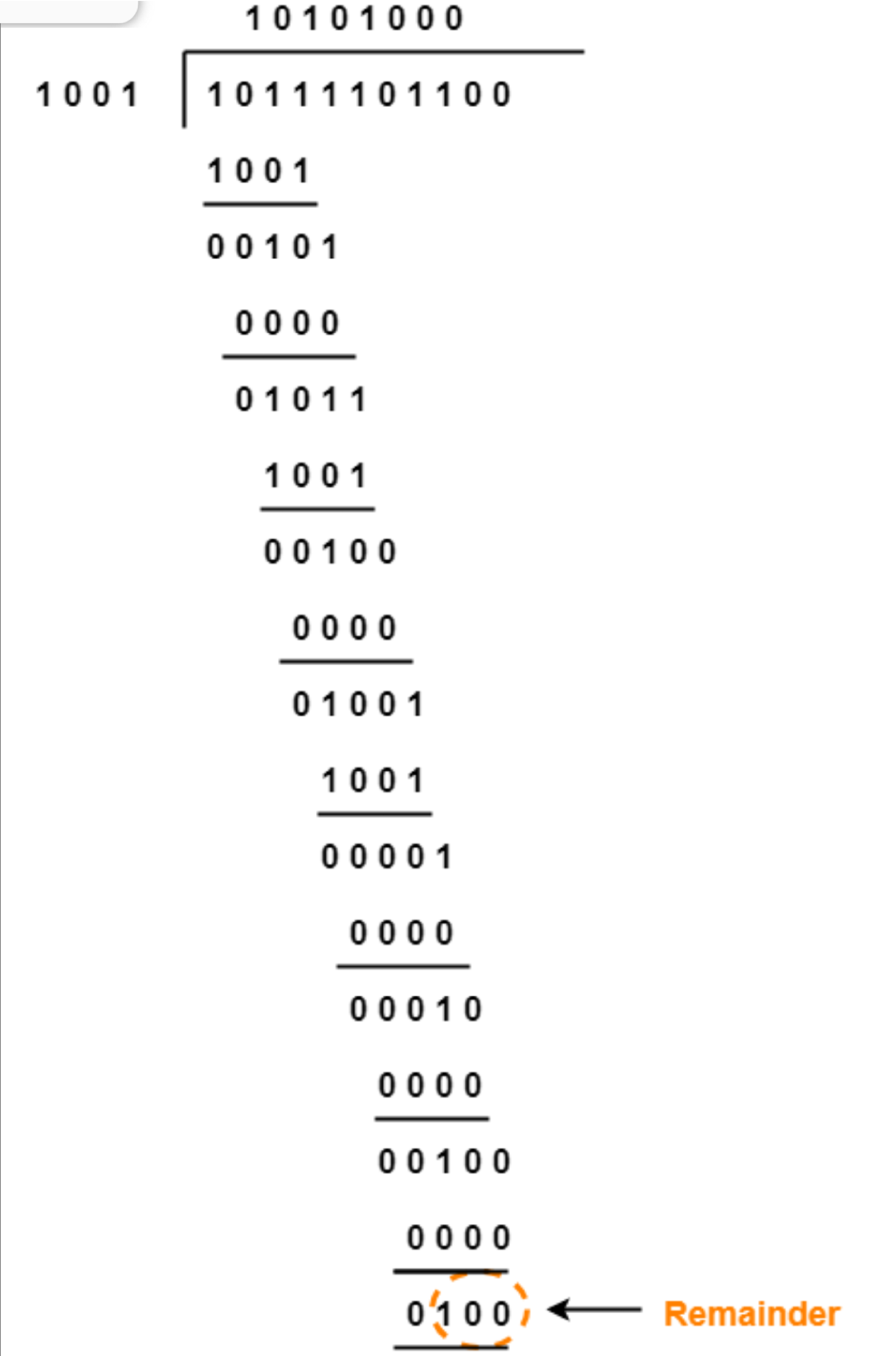
Part-02:

According to the question,

- Third bit from the left gets inverted during transmission.
- So, the bit stream received by the receiver = 10111101100.

Now,

- Receiver receives the bit stream = 10111101100.
- Receiver performs the binary division with the same generator polynomial as-


10101000
1001
10111101100
1001
00101
0000
01011
1001
00100
0000
01001
1001
00001
0000
00010
0000
00100
0000
0100
Remainder

From here,

- The remainder obtained on division is a non-zero value.
- This indicates to the receiver that an error occurred in the data during the transmission.
- Therefore, receiver rejects the data and asks the sender for retransmission.