Computer Networks

Unit 3
Chapter 3:Error Detection, Correction and Wireless Communication

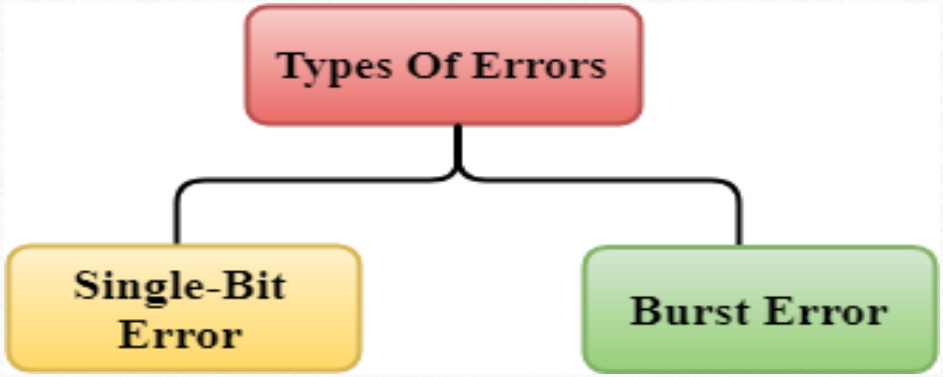
Types Of Error?

When data is transmitted from one device to another device, the system does not guarantee whether the data received by the device is identical to the data transmitted by another device. An Error is a situation when the message received at the receiver end is not identical to the message transmitted.
That means a 0 bit may change to 1 or a 1 bit may change to 0

Types Of Errors

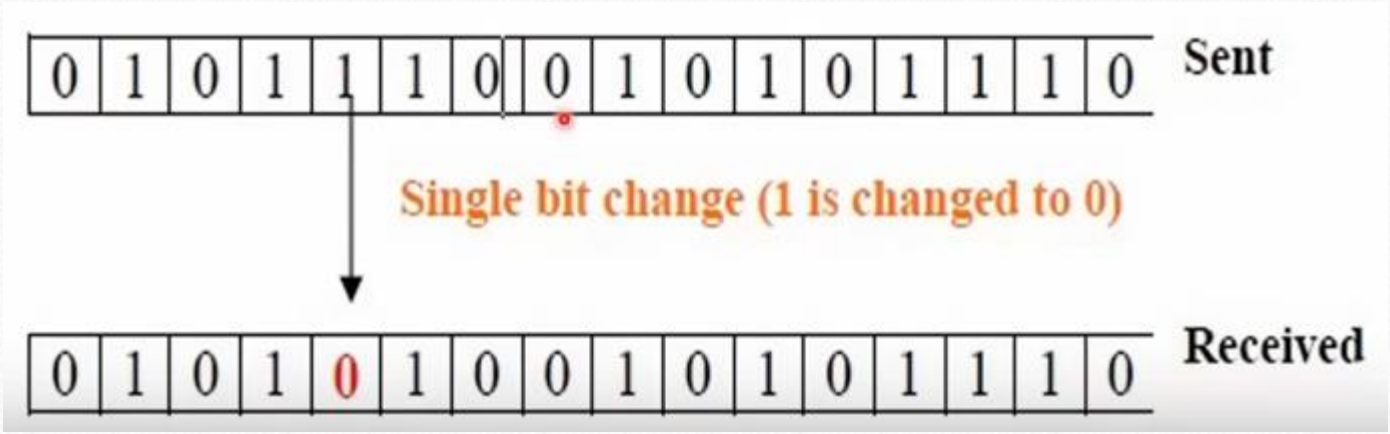
- ✓ Whenever bits flow from one point to another, they are subject to unpredictable changes because of interference. This interference can change the shape of the signals .

1. Single-Bit Error
2. Burst Error



1. Single-Bit Error:

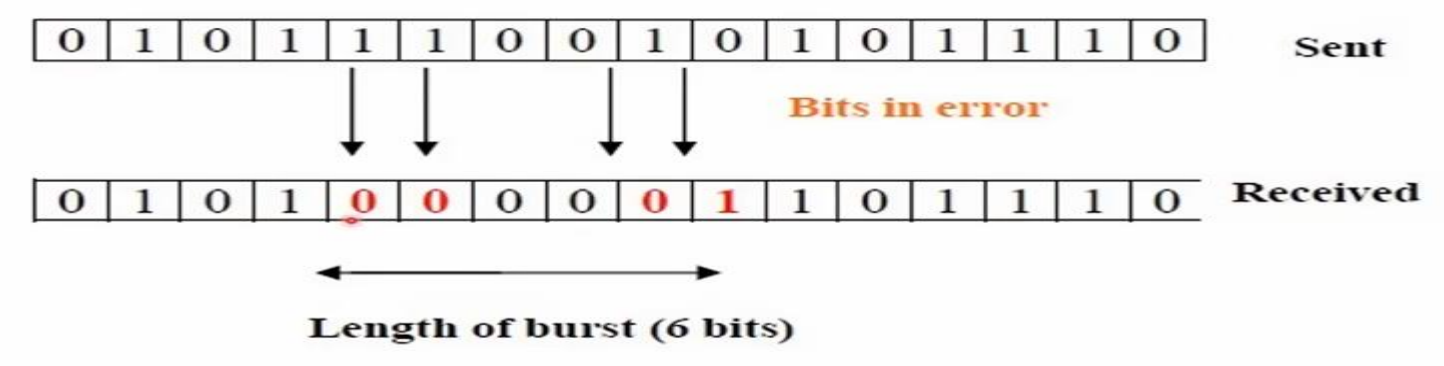
- The only one bit of a given data unit is changed from 1 to 0 or from 0 to 1.



- ✓ In the above figure, the message which is sent is corrupted as single-bit, i.e., 0 bit is changed to 1.
- ✓ Single-Bit Error does not appear more likely in Serial Data Transmission.
- ✓ Single-Bit Error mainly occurs in Parallel Data Transmission. For example, if eight wires are used to send the eight bits of a byte, if one of the wire is noisy, then single-bit is corrupted per byte.

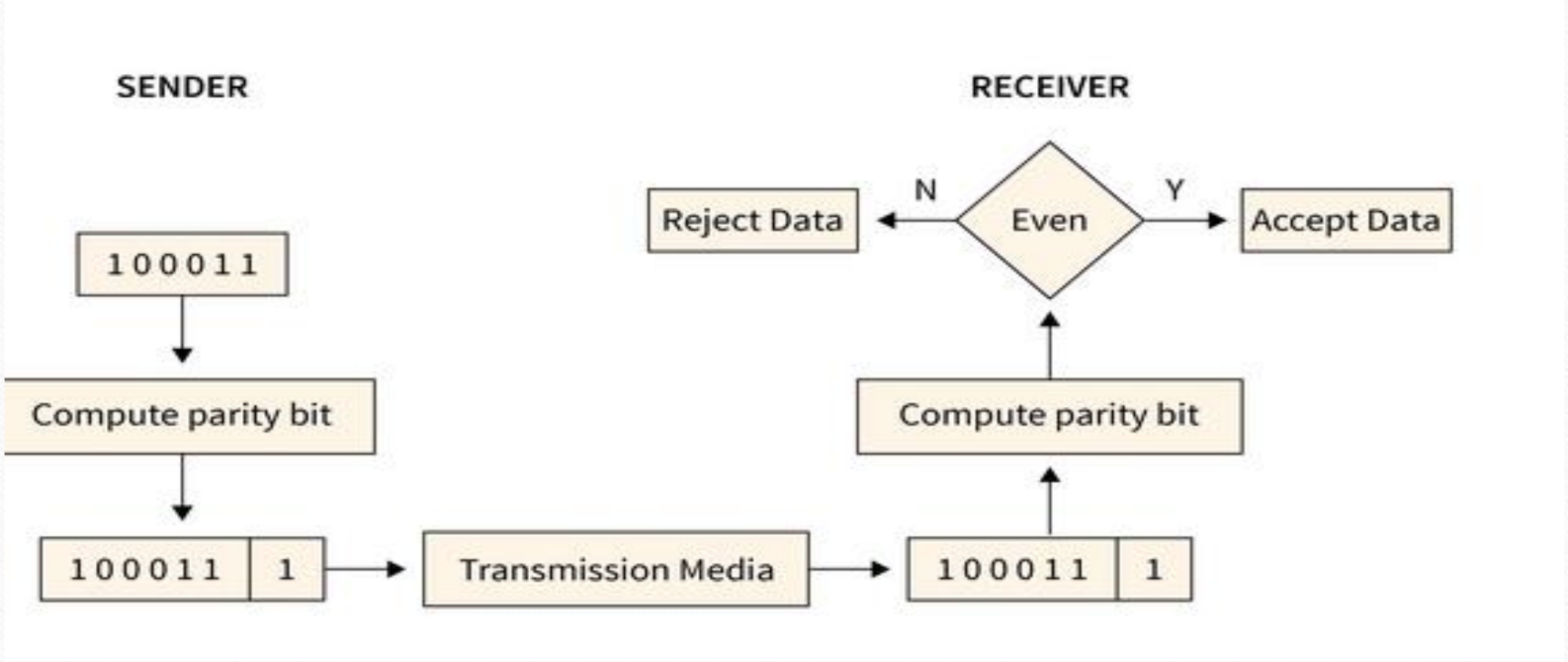
2. Burst Error:

- ✓ The two or more bits are changed from 0 to 1 or from 1 to 0 is known as Burst Error.
- ✓ The Burst Error is determined from the first corrupted bit to the last corrupted bit.
- ✓ The length of the burst error is measured from the first corrupted bit to the last corrupted bit. Some bit in between may not be corrupted.
- ✓ The duration of noise in Burst Error is more than the duration of noise in Single-Bit.
- ✓ Burst Errors are most likely to occur in Serial Data Transmission.
- ✓ The number of affected bits depends on the duration of the noise and data rate.
- ✓ Note that the burst error doesn't necessary mean that error occur in consecutive bits.



Redundancy?

- The central concept in detecting or correction errors is redundancy.
- To be able to detect or correct errors, we need to send some extra bits with our data.
- These redundant bits are added by the sender and removed by the receiver.
- Their presence allows the receiver to detect or correct corrupted bits.
- Redundant bits are called as parity bit. Parity may be odd or even .Odd and even parity signifies that the total of 1s in a bit string is an odd or even number.
- 1 is added as a parity bit to the data block if the data block has an odd number of 1's.
- 0 is added as a parity bit to the data block if the data block has an even number of 1's.



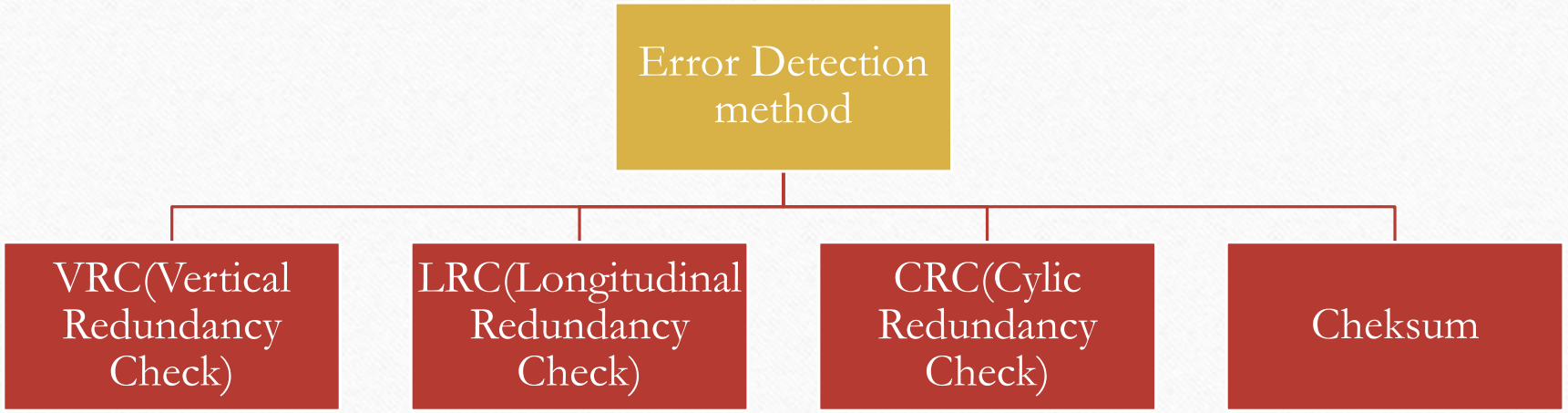
Disadvantage:

•Only **single-bit error** is detected by this method, it fails in multi-bit error detection .

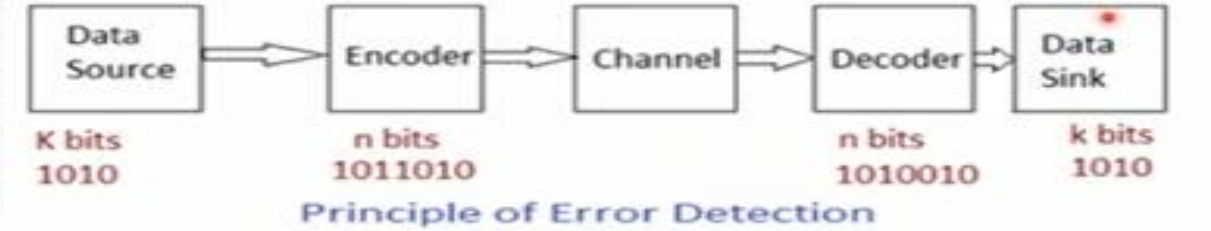
•It can not detect an error in case of an error in **two bits**.

Detection Versus Correction?

- The correction of errors is more difficult than the detection
- In error detection, we check if any errors have occurred or not.
- In error correction, we check the exact number of bits that are corrupted and their location in the message.
- The number of errors and the size of the message are important factors. If we need to correct one single error in an 8-bit data unit, we need to consider eight possible error.



- When frame is transmitted from transmitter to the receiver. There are two possibilities viz. frame is received without error. frame is received in error (i.e. frame is bad).
- Error detection helps in detecting errors in a received block or frame by the receiver.
- Once the error is detected receiver informs the transmitter to re-transmit the same frame again.
- Error detection can be made possible by adding redundant bits in each frame during transmission. Based on all the bits in the frame. data + error check bits). receiver is capable of detecting errors in the frame.



Principle of Error Detection

Error Detection modules

➤ Message (or data) source : Source of information in bits (K bits)

➤ Encoding process : Process of converting block of k-bit information to n-bit codeword. n = k .

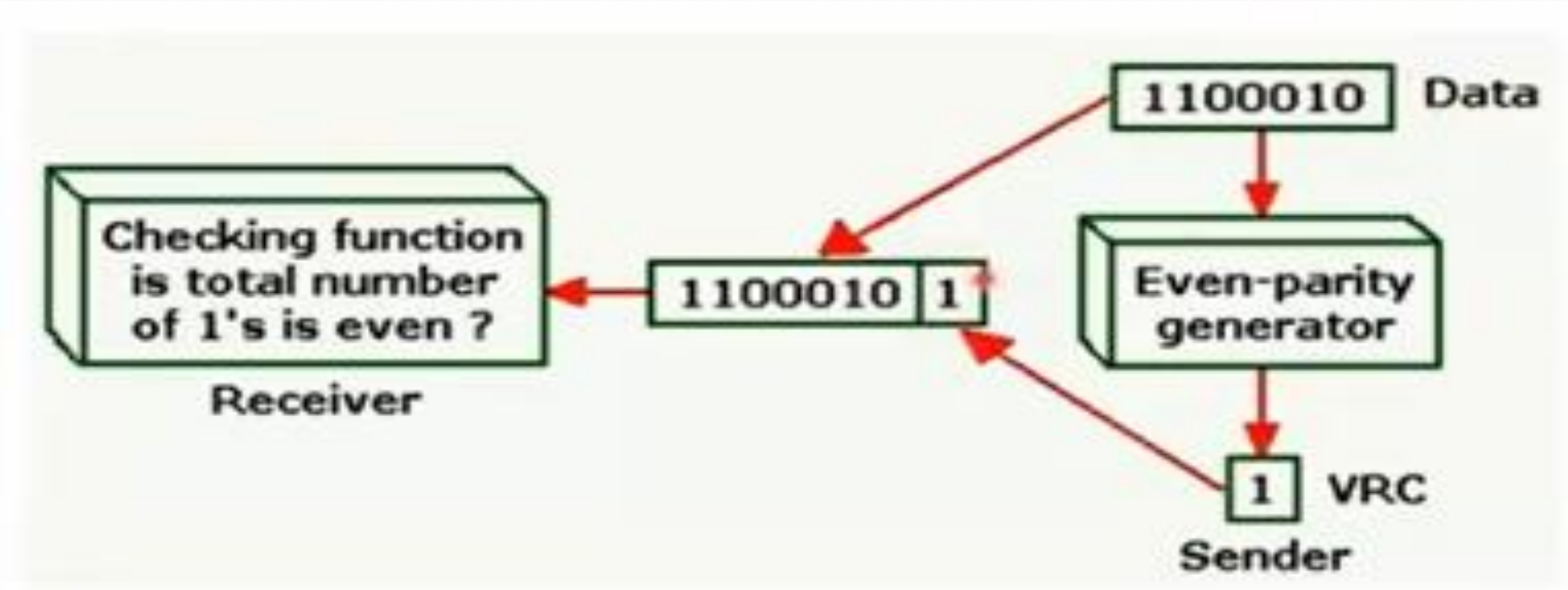
➤ Channel : Medium in which n-bit codewords pass through.

➤ Decoding process : Process of converting n-bit received block to k-bit message.

➤ Message sink : Destination for k-bit information in bits.

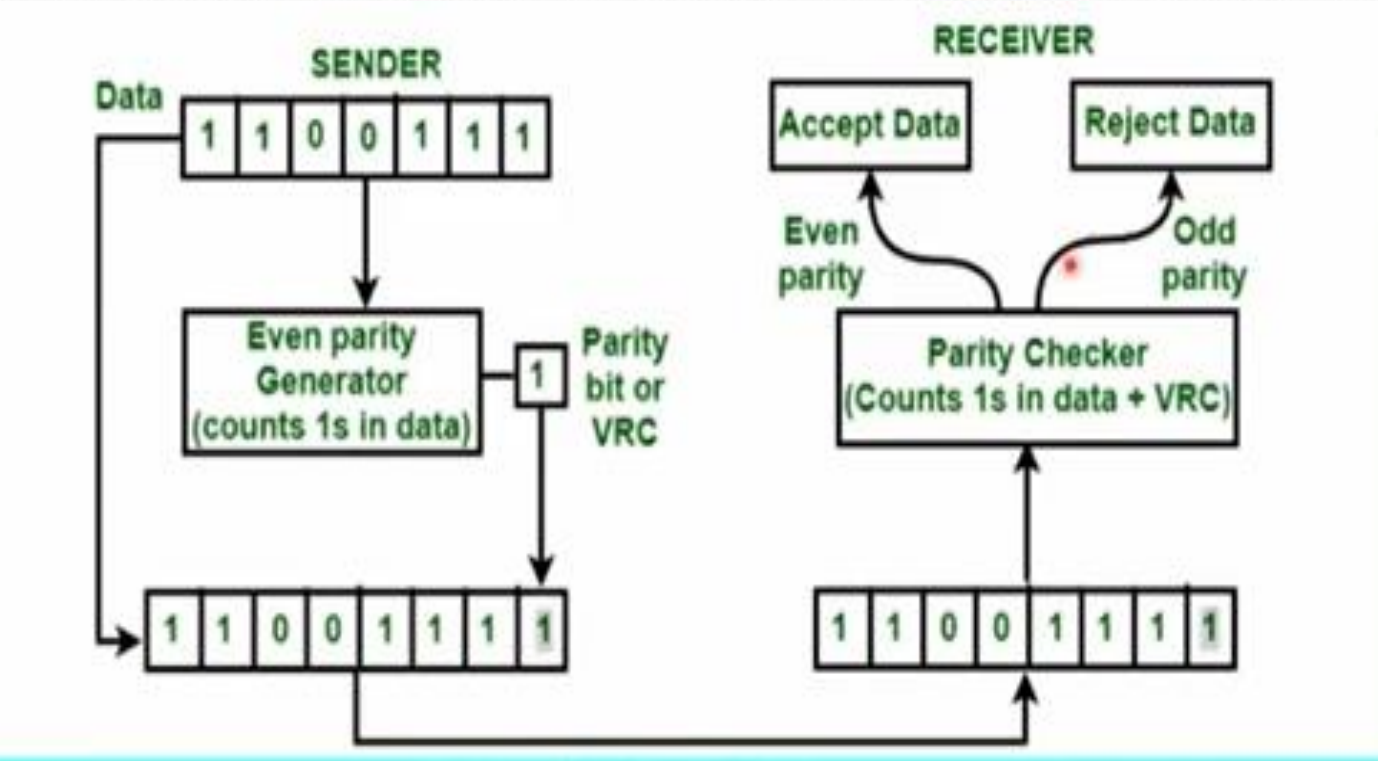
Parity check or vertical redundancy check (VRC) method :

- Vertical redundancy check is an error detecting technique used to detect errors on an eight-bit ASCII character.
- To determine either the transmission is correct, a parity bit is linked to every byte of data.
- Vertical Redundancy Check is also known as Parity Check.
- In this method, a redundant bit also called parity bit is added to each data unit.
- This method includes even parity and odd parity.
- Even parity means the total number of 1s in data is to be even and odd parity means the total number of 1s in data is to be odd.



- Vertical redundancy check is maintenance of parity bit. An additional bit is added with original block to ensure that data transmitted correctly. It is also known as parity check technique.

- In VRC there are two types of parity bit "Even parity bit" and "Odd parity bit'.



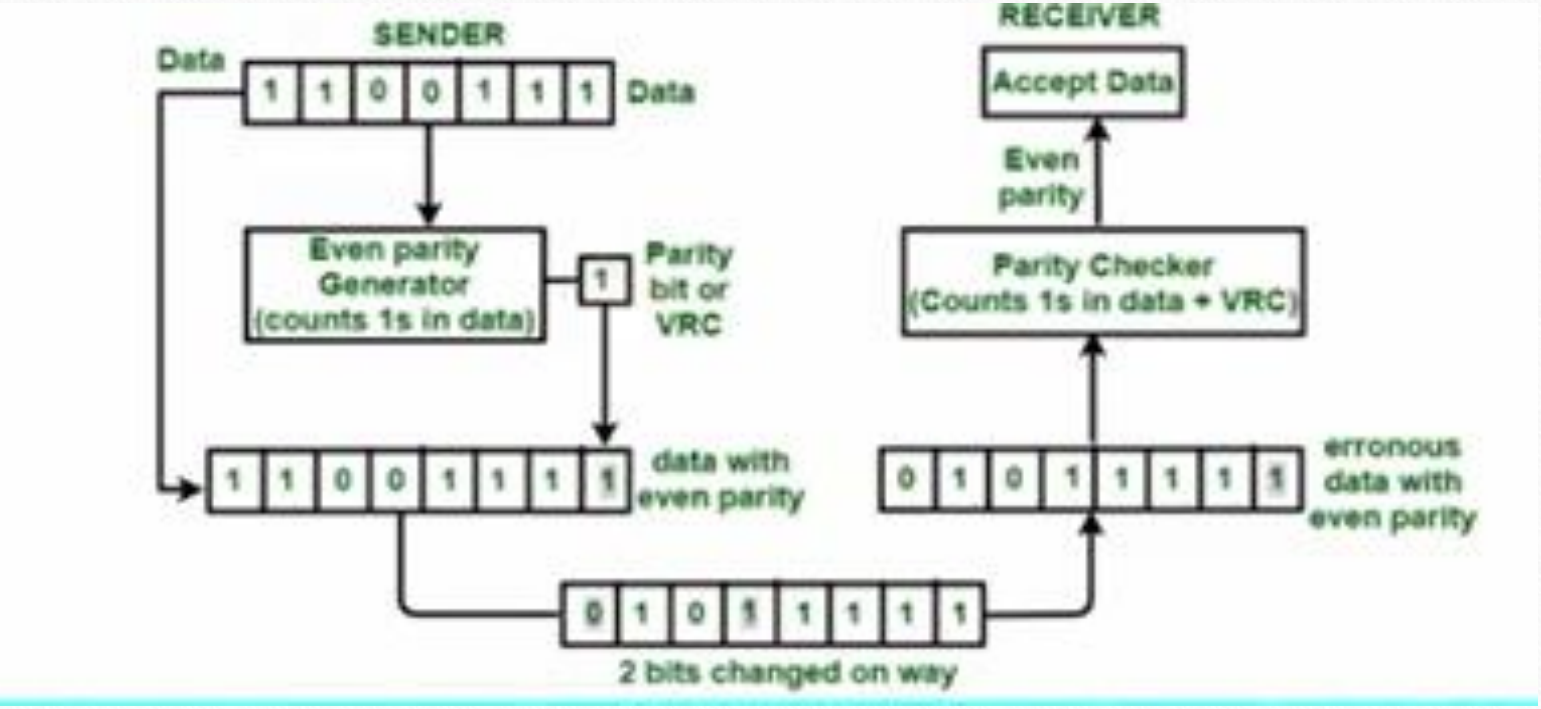
- For Example, if the source wants to transmit data unit 1100111 using even parity to the destination. The source will have to pass through even parity generator.

Advantages

- VRC can detect all single bit error.
- It can also detect burst errors but only in those cases where number of bits changed is odd ie 1,3,5,7….etc

Disadvantages

The major disadvantage of using this method for error detection is that it is not able to detect burst error if the number of bits changed is even ie 2,4,6,8 … etc



Example —If the original data is 1100111. After adding VRC, data unit that will be transmitted is 11001111. Suppose on the way 2 bits arc 01011111. When this data will teach the destination, parity checker will count number of Is in data and that comes out to be even i.e. 8. So. in this case, parity is not changed. it is still even. Destination will assume that there is no error in data even though data is erroneous.

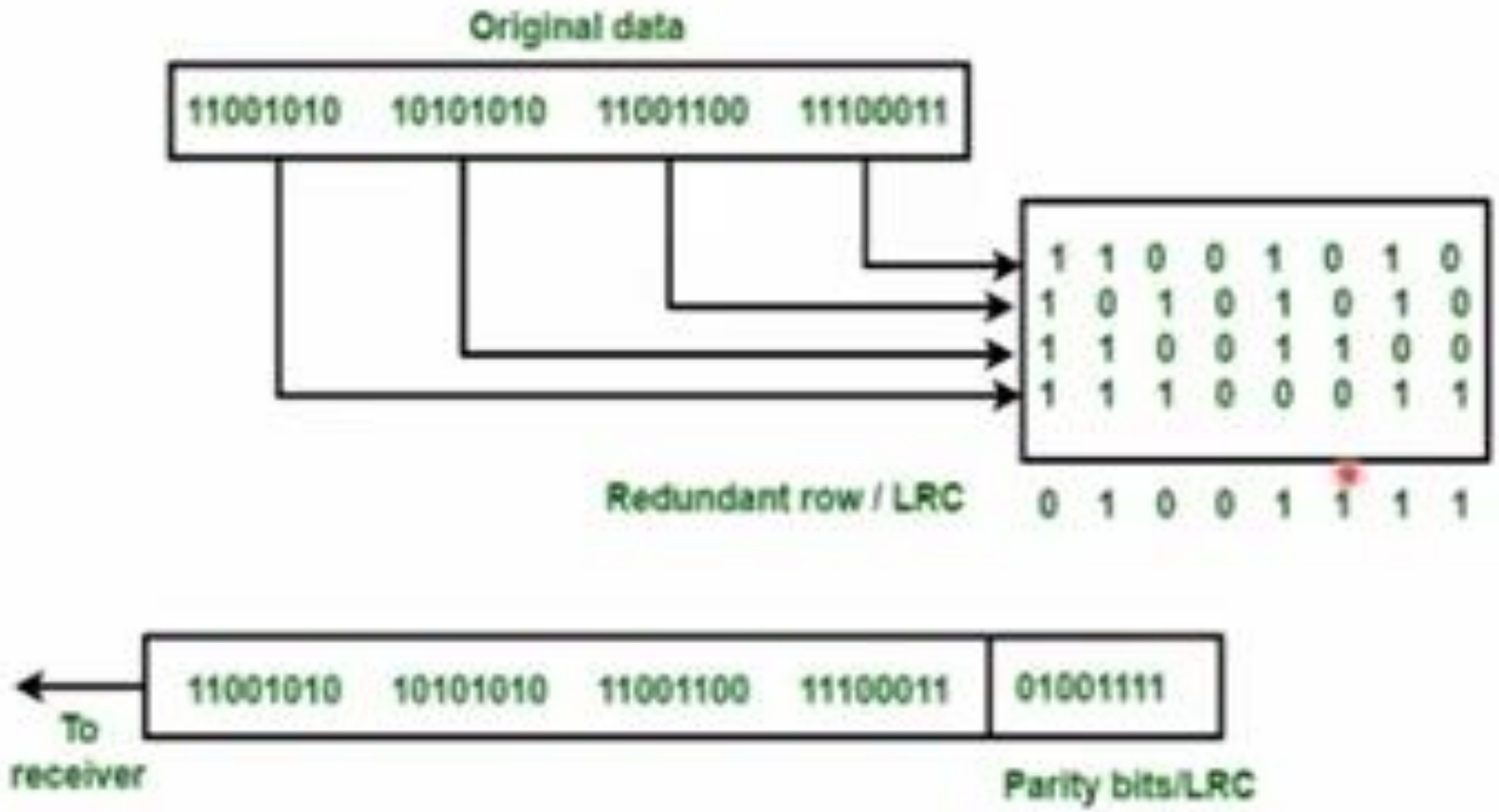
Longitudinal Redundancy Check (LRC)

- In this method, data which the user want to send is organized into tables of rows and columns.
- A block of bit is divided into table or matrix of rows and columns.
- In order to detect an error, a redundant bit is added to the whole block and this block is transmitted to receiver.
- The receiver uses this redundant row to detect error.
- After checking the data for errors, receiver accepts the data and discards the redundant row of bits.

Longitudinal Redundancy Check (LRC)

- If a block of 32 bits is to be transmitted, it is divided into matrix of four rows and eight columns which as shown in the following figure:

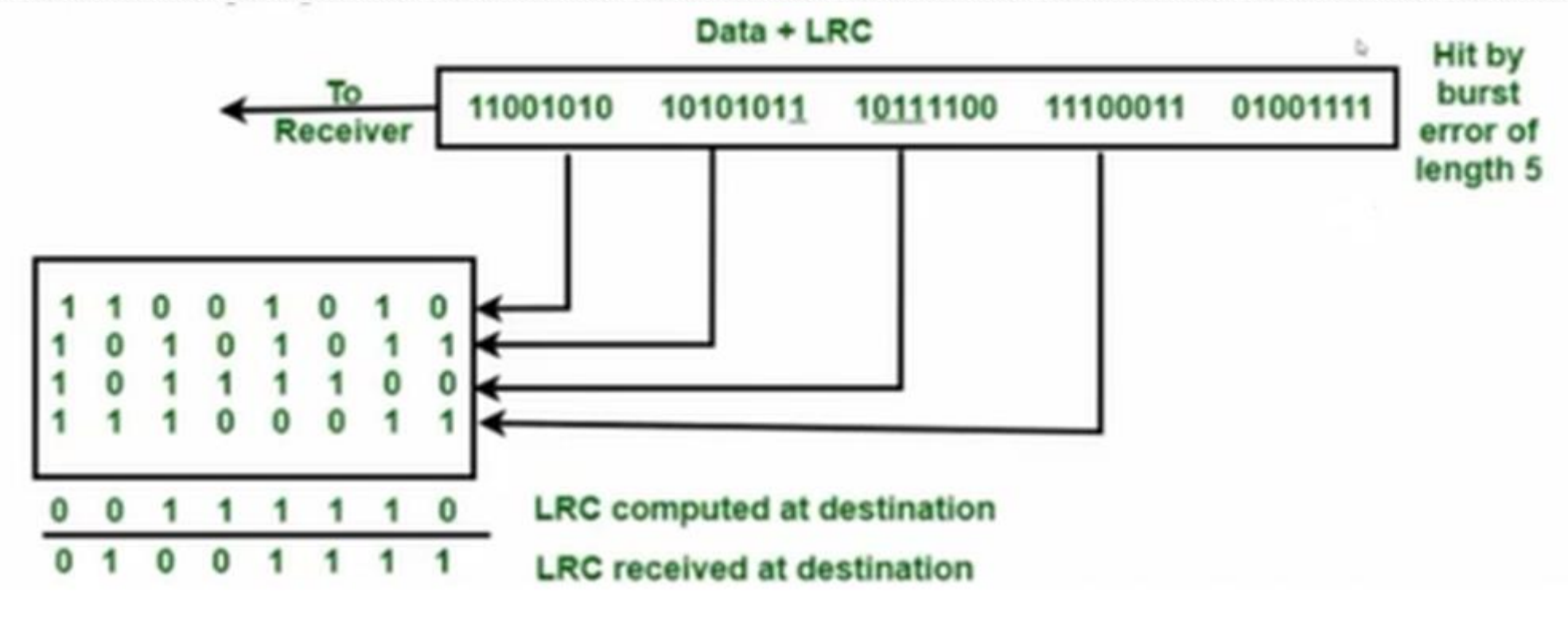
In this matrix of bits, a parity bit (odd or even) is calculated for each column. It means 32 bits data plus 8 redundant bits are transmitted to receiver. Whenever data reaches at the destination. receiver uses LRC to detect error in data.



Advantage :

LRC is used to detect burst errors.

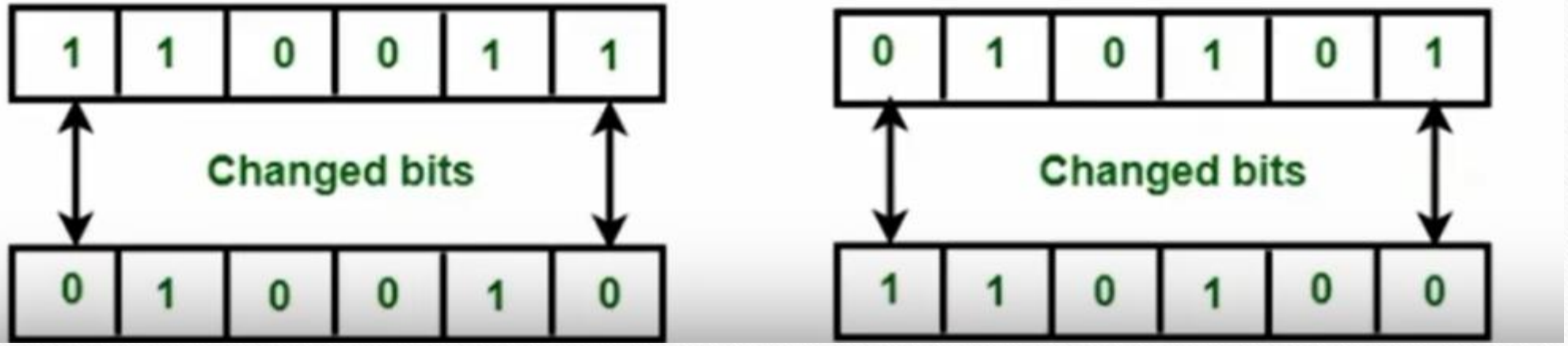
Example : Suppose 32 bit data plus LRC that was being transmitted is hit by a burst error of length 5 and some bits are corrupted as shown in the following figure :



Disadvantage

The main problem with LRC is that, it is not able to detect error if two bits in a data unit are damaged and two bits in exactly the same position in other data unit are also damaged.

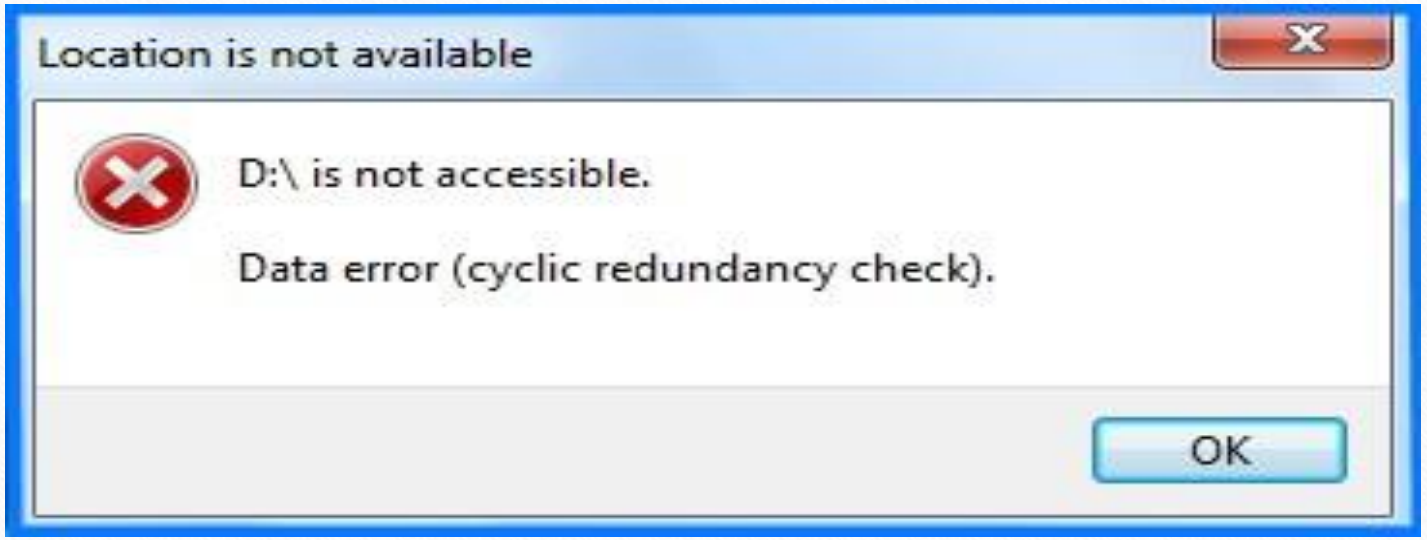
Example : If data 110011 010101 is changed to 010010110100.



Q. Cyclic Redundancy Check (CRC)?

Ans:

- CRC is a redundancy error technique used to determine the error.
- Cyclic Redundancy Check is a number mathematically calculated for a packet by its source computer, and then recalculated by the destination computer.
- If the original and recalculated versions at the destination computer differ, the packet is corrupt and needs to be resent or ignored.

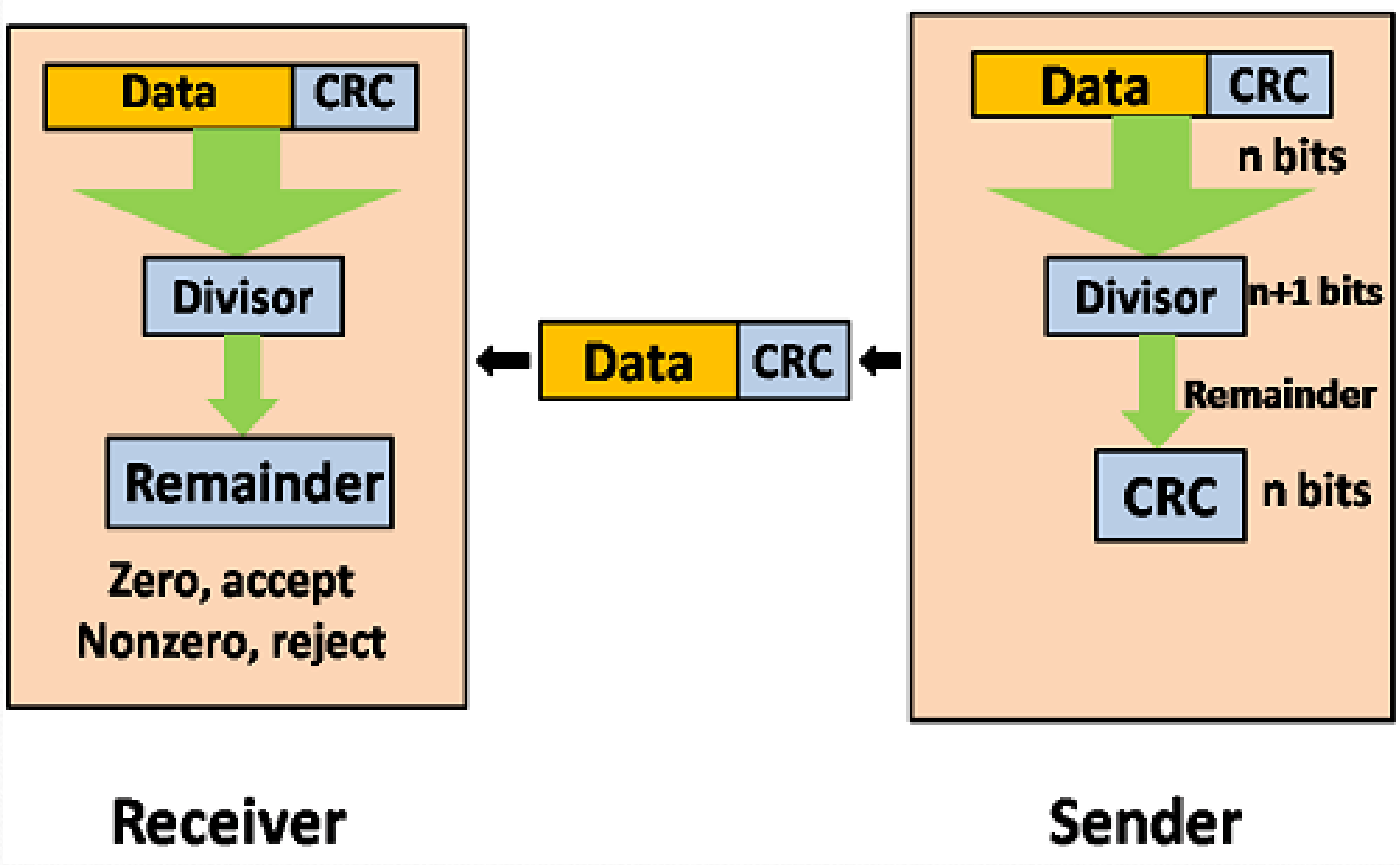


Following are the steps used in CRC for error detection:

- ✓ In CRC technique, a string of n 0s is appended to the data unit, and this n number is less than the number of bits in a predetermined number, known as division which is n+1 bits.
- ✓ Secondly, the newly extended data is divided by a divisor using a process is known as binary division. The remainder generated from this division is known as CRC remainder.
- ✓ Thirdly, the CRC remainder replaces the appended 0s at the end of the original data. This newly generated unit is sent to the receiver.
- ✓ The receiver receives the data followed by the CRC remainder. The receiver will treat this whole unit as a single unit, and it is divided by the same divisor that was used to find the CRC remainder.

If the resultant of this division is zero which means that it has no error, and the data is accepted.

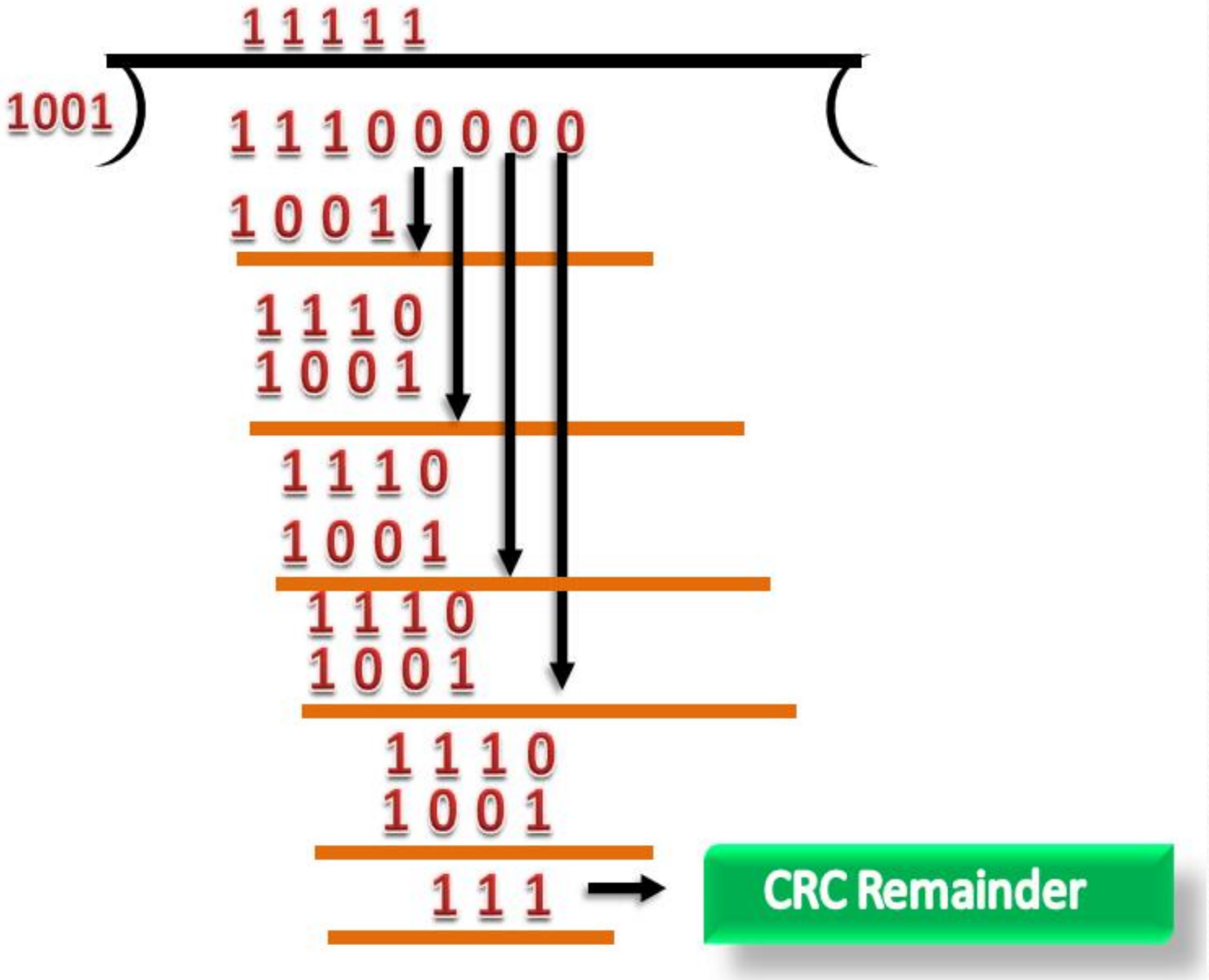
If the resultant of this division is not zero which means that the data consists of an error. Therefore, the data is discarded.


Data
CRC
Divisor
Remainder
Zero, accept
Nonzero, reject
Data
CRC
Data
CRC
n bits
Divisor
n+1 bits
Remainder
CRC
n bits
Receiver
Sender

Suppose the original data is 11100 and divisor is 1001.

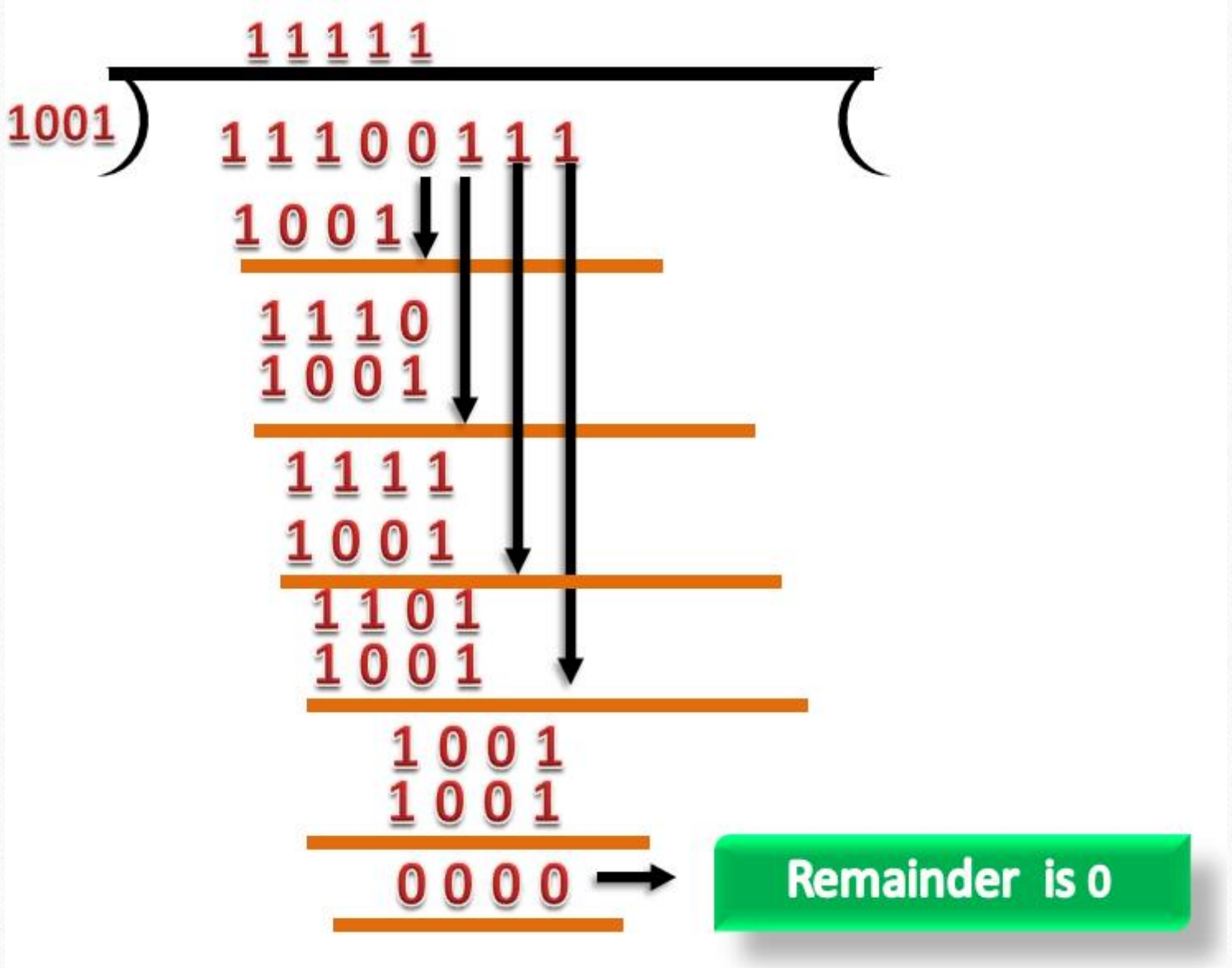
CRC Generator

- A CRC generator uses a modulo-2 division. Firstly, three zeroes are appended at the end of the data as the length of the divisor is 4 and we know that the length of the string 0s to be appended is always one less than the length of the divisor.
- Now, the string becomes 11100000, and the resultant string is divided by the divisor 1001.
- The remainder generated from the binary division is known as CRC remainder. The generated value of the CRC remainder is 111.
- CRC remainder replaces the appended string of 0s at the end of the data unit, and the final string would be 11100111 which is sent across the network.


11111
1001
11100000
1001
1110
1001
1110
1001
1110
1001
111
CRC Remainder

CRC Checker

- The functionality of the CRC checker is similar to the CRC generator.
- When the string 11100111 is received at the receiving end, then CRC checker performs the modulo-2 division.
- A string is divided by the same divisor, i.e., 1001.
- In this case, CRC checker generates the remainder of zero. Therefore, the data is accepted.


11111
1001
11100111
1001
1110
1001
1111
1001
1101
1001
1001
1001
0000
Remainder is 0

Q. Checksums?

Ans:

- This is a block code method where a checksum is created based on the data values in the data blocks to be transmitted using some algorithm and appended to the data.
- When the receiver gets this data, a new checksum is calculated and compared with the existing checksum. A non-match indicates an error.

Error Detection by Checksums

For error detection by checksums, data is divided into fixed sized frames or segments.

Sender's End − The sender adds the segments using 1's complement arithmetic to get the sum. It then complements the sum to get the checksum and sends it along with the data frames.

Receiver's End − The receiver adds the incoming segments along with the checksum using 1's complement arithmetic to get the sum and then complements it.

If the result is zero, the received frames are accepted; otherwise they are discarded.

Example

- Suppose that the sender wants to send 4 frames each of 8 bits, where the frames are 11001100, 10101010, 11110000 and 11000011.
- The sender adds the bits using 1s complement arithmetic. While adding two numbers using 1s complement arithmetic, if there is a carry over, it is added to the sum.
- After adding all the 4 frames, the sender complements the sum to get the checksum, 11010011, and sends it along with the data frames.
- The receiver performs 1s complement arithmetic sum of all the frames including the checksum. The result is complemented and found to be 0. Hence, the receiver assumes that no error has occurred.

Sender's End

Frame 1:	11001100
Frame 2:	+ 10101010
Partial Sum:	1 01110110
	+ 1
	01110111
Frame 3:	+ 11110000
Partial Sum:	1 01100111
	+ 1
	01101000
Frame 4:	+ 11000011
Partial Sum:	1 00101011
	+ 1
Sum:	00101100
Checksum:	11010011

Receiver's End

Frame 1:	11001100
Frame 2:	+ 10101010
Partial Sum:	1 01110110
	+ 1
	01110111
Frame 3:	+ 11110000
Partial Sum:	1 01100111
	+ 1
	01101000
Frame 4:	+ 11000011
Partial Sum:	1 00101011
	+ 1
Sum:	00101100
Checksum:	11010011
Sum:	11111111
Complement:	00000000

Hence accept frames.

Thank You.