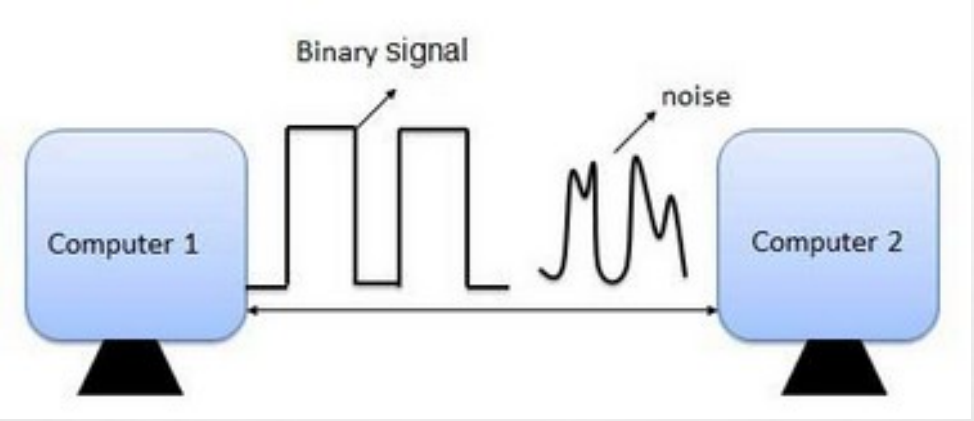
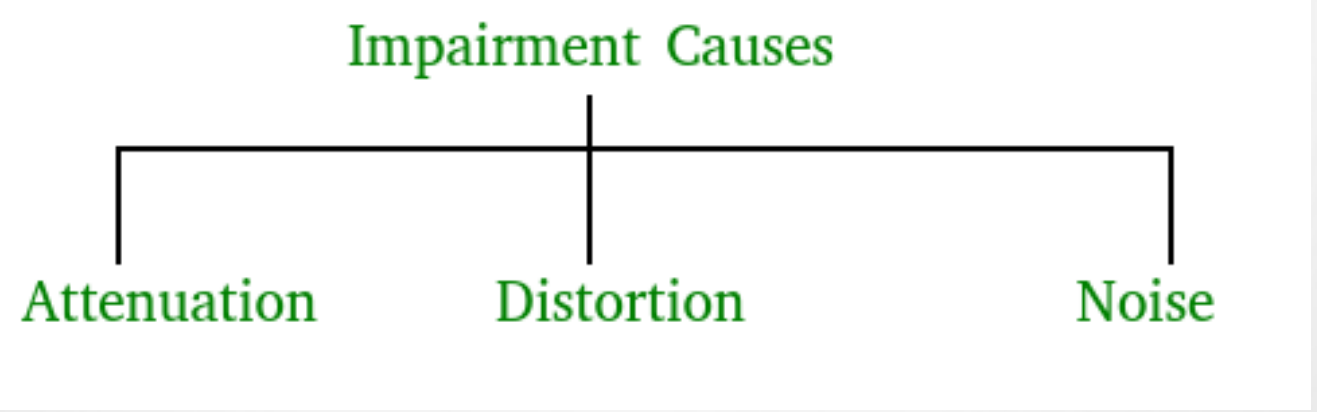
Transmission Errors: Detection and correction

- In Data Communication Networks **electromagnetic signals** can cause incorrect delivery of data. Due to this data in the communication can be received incorrectly or data can be lost or unwanted data can be generated. Any of these problems are called transmission errors in communication networks.
- Error is a condition when the output information does not match with the input information. During transmission, digital signals suffer from noise that can introduce errors in the binary bits travelling from one system to other. That means a 0 bit may change to 1 or a 1 bit may change to 0.



Transmission Errors: Detection and correction

- The errors can be classified in three basic categories as given below:
  - **Delay distortion**
  - **Attenuation**
  - **Noise**

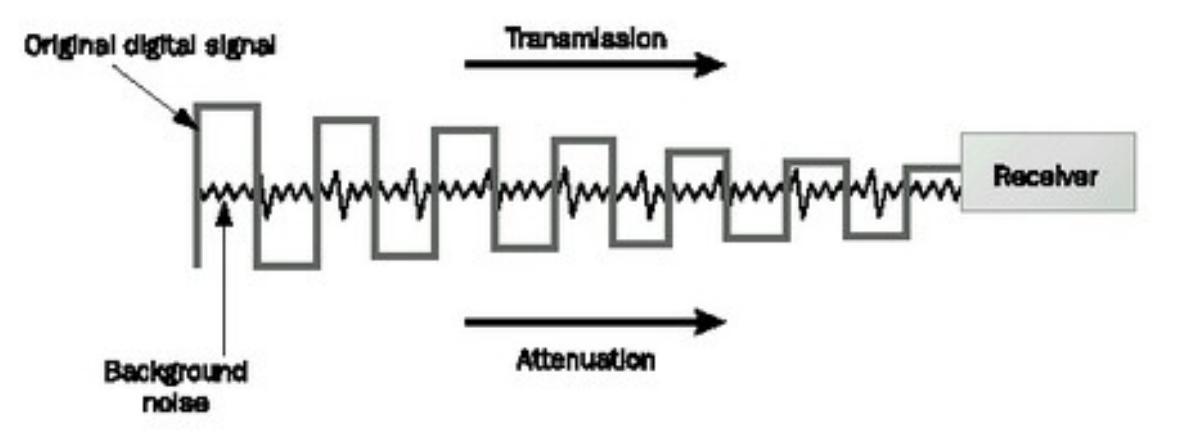


Transmission Errors: Detection and correction

- The errors can be classified in three basic categories as given below:
  - **Delay distortion**
  - **Attenuation**
  - **Noise**
- **Delay Distortion**
- Delay Distortion is caused because signals at different frequencies travel at different speeds along the medium.
- Any complex signal can be decomposed into different sinusoidal signals (Component signals) of different frequencies resulting in a frequency bandwidth for every signal.
- The property of signal propagation is such that the speed of travel of frequency at the center of this bandwidth is highest and this speed is low at both ends of the frequency bandwidth.
- At the receiving end, signals with different frequency in a given bandwidth will arrive at different times. Hence at the receiver, if the receiving frequencies are measured at a specific time, they will not measure up to the original signal resulting in its **misinterpretation**.

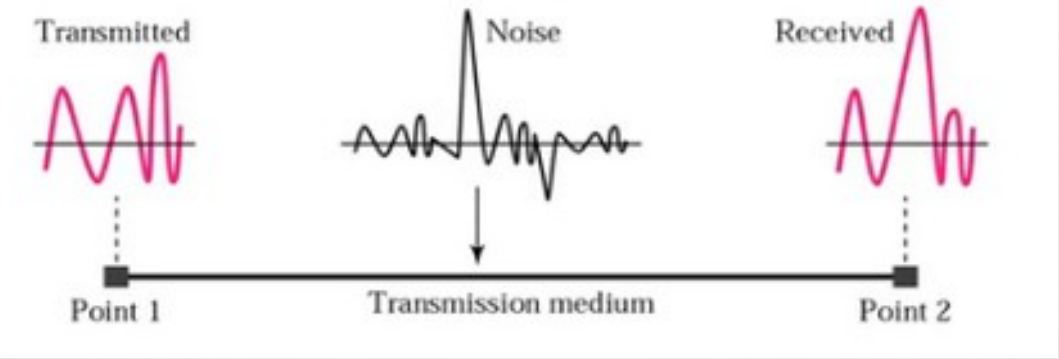
Transmission Errors: Detection and correction

- **Attenuation:**
- Attenuation is another form of **distortion**.
- In attenuation , as a signal travels through any medium, its strength decreases. For ex. Just like our voice becomes weak over a distance and loses its contents beyond a certain distance.
- Attenuation means loss of energy. When any signal travels over a medium or channel, it loses some of its energy in the form of heat in the resistance of the medium. Attenuation decides the signal to noise ratio hence the quality of received signal.
- Attenuation is very small at short distances, it increases with distance.



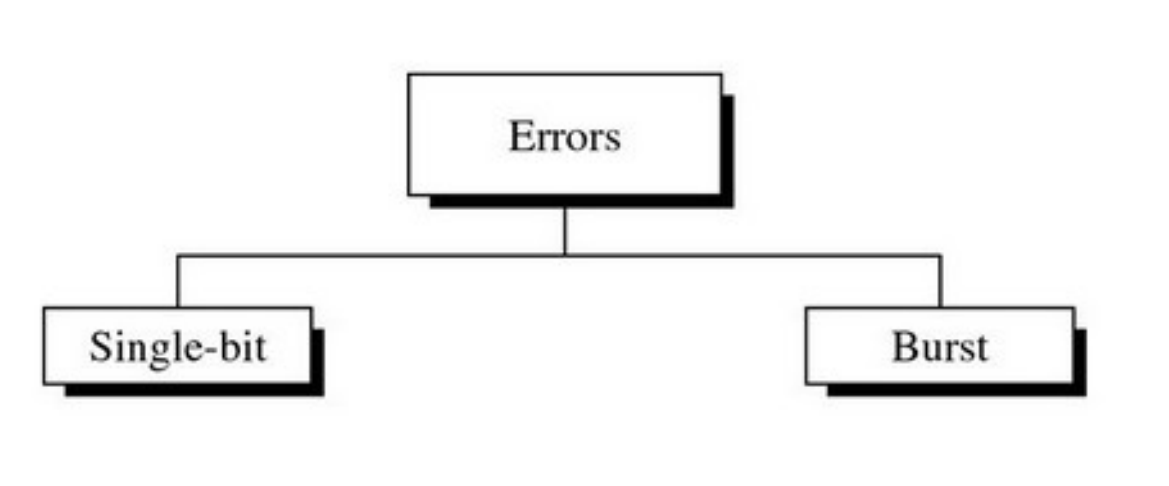
Transmission Errors: Detection and correction

- **Noise:**
- The random/unwanted signal mixes up with the original signal is called noise.
- When signals travel as electromagnetic signals through any medium, some electromagnetic energy get inserted somewhere during transmission **– called Noise**
- There are several types of noise such as induced noise, crosstalk noise, thermal noise and impulse noise which may corrupt the signal.
- **Induced noise** comes from sources such as motors and appliances. These devices act as sending antenna and transmission medium act as receiving antenna. **Thermal noise** is movement of electrons in wire which creates an extra signal. **Crosstalk noise** is when one wire affects the other wire**. Impulse noise** is a signal with high energy that comes from lightning or power lines.



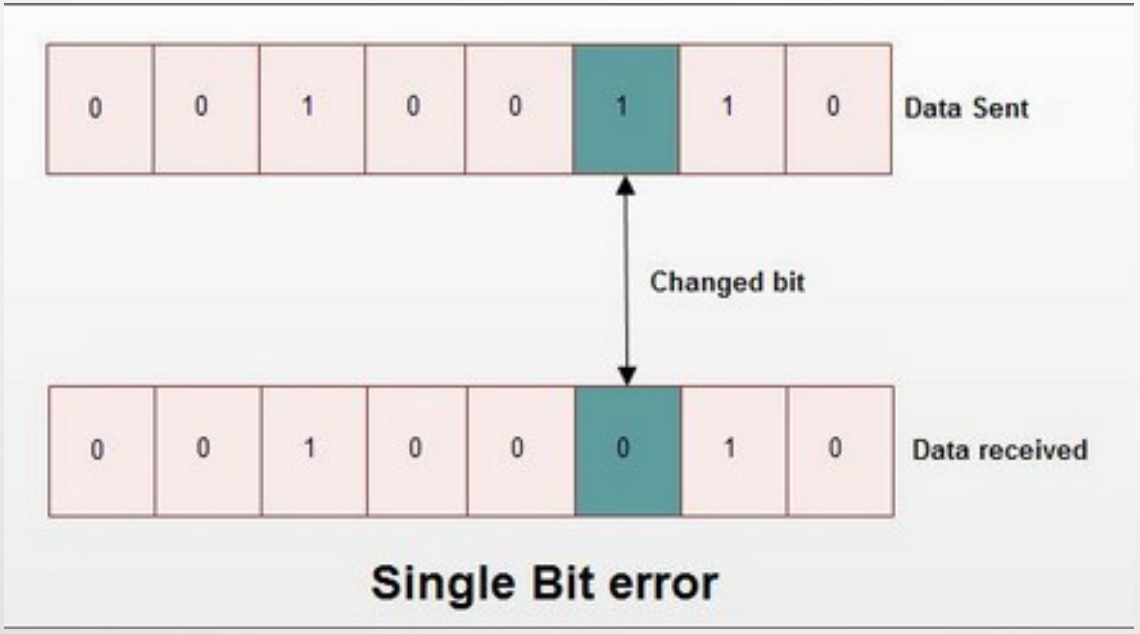
Types of Errors

- **Types of Errors:**
  - If the signal is carrying binary data there can be two types of errors. **1. Single bit errors and 2. burst errors.**
  - In single bit errors a bit value of 0 changes to 1 and vice versa.
  - In burst error, multiple bits of binary value are changed.



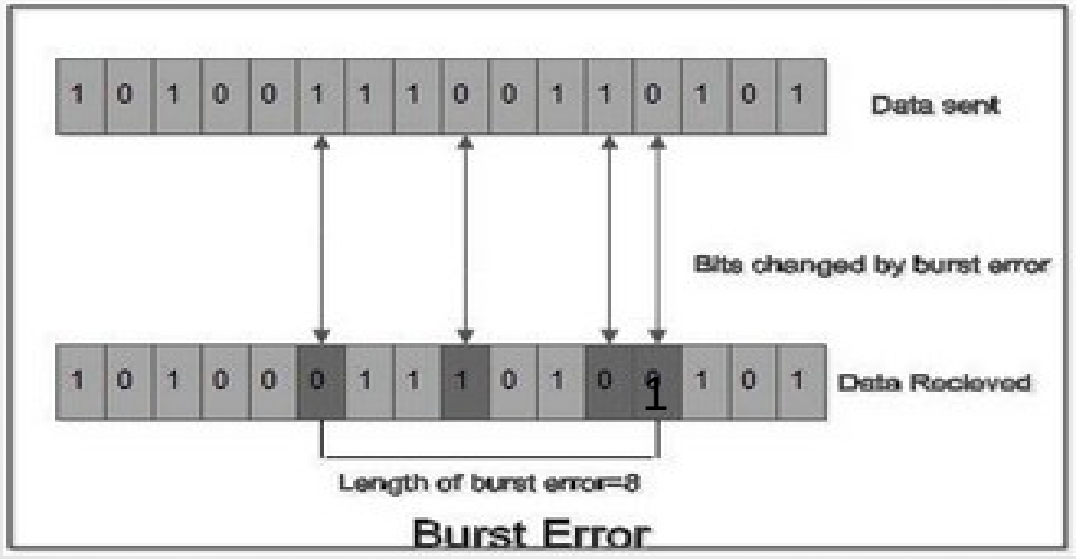
Types of Errors

- **Single Bit Error:**
- It means only one bit of data unit is changed from 1 to 0 or from 0 to 1 as shown in fig:
- Single bit error can happen in **parallel transmission** where all the data bits are transmitted using separate wires.



Types of Errors

- **Burst Error:**
- It means two or more bits in data unit are changed from 1 to 0 from 0 to 1 as shown in fig:
- Burst error does not necessarily mean that errors occur in consecutive bits.
- Most likely to happen in **serial transmission.**
- The length of burst error is measured from first changed bit to last changed bit.
- As shown in fig. length of burst error is 8, although some bits are unchanged in between.
- Number of bits affected depends on the data rate and duration of noise.



Types of Errors

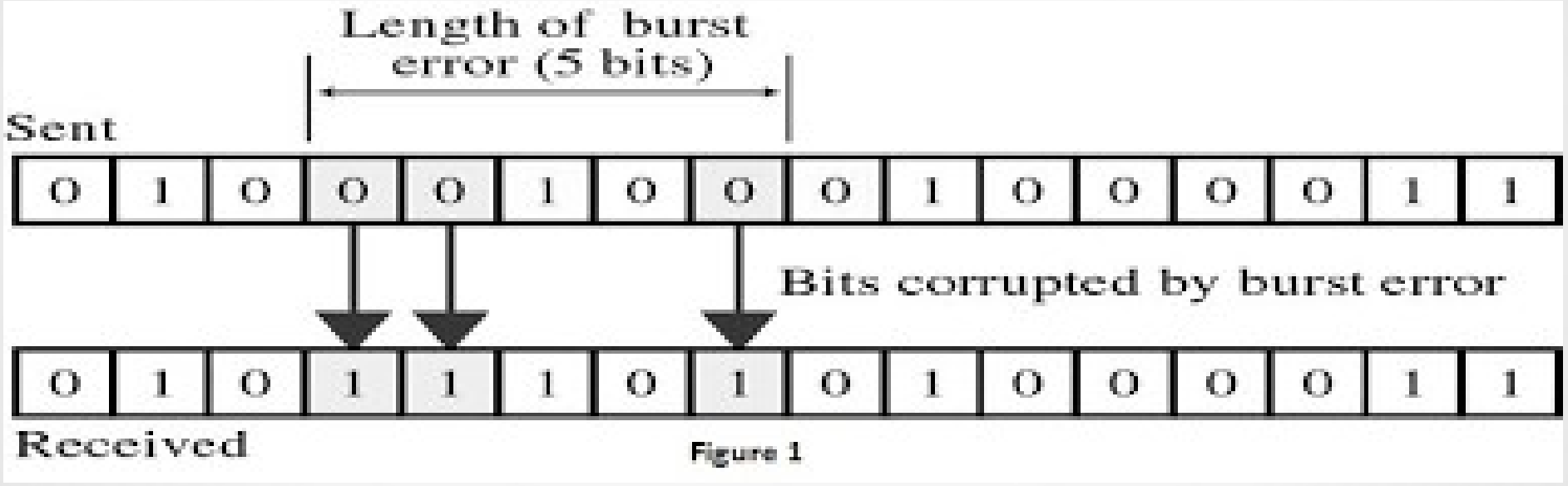


Figure 1

Error Detection

- **Error Detection:**
- Whenever a message is transmitted, it may get scrambled by noise or data may get corrupted. To avoid this, we use error-detecting codes which are additional data added to a given digital message to help us detect if any error has occurred during transmission of the message.
- Basic approach used for error detection is the use of redundancy bits, where additional bits are added to facilitate detection of errors.
- **Some popular techniques for error detection are:**
  - **Checksum**
  - **Vertical Redundancy Check(VRC) or Parity Check**
  - **Longitudinal Redundancy Check(LRC)**
  - **Cyclic Redundancy Check(CRC)**

Error Detection

- **Checksum:**
- A **checksum** also called **hash sum** is fixed-length data that is the result of performing certain operations on the data to be sent from the sender to the receiver.
- The sender runs the appropriate checksum algorithm to compute the checksum of the data, appends it as a field in the packet that contains data to be sent as well as headers.
- When the receiver receives the data the receiver runs the same checksum algorithm to compute a fresh checksum.
- The receiver compares freshly computed checksum with the checksum that was computed by the sender.
- If the two checksums match, the receiver of the data is assured that the data has not changed during transmission.
- Various checksum algorithms are popular. Most common are **parity check, modular sum, position-dependent checksum, etc.**

Error Detection

- **Checksum:**
- In **modular sum**, the data that the sender is sending is arranged into smaller blocks called **words.** For example, if the data stream that needs to be sent is 1100100001111100110, then we can make up 5 words, each containing 4 bits. The 5 words are:
  - 1100 1000 0111 1110 0110
  - This is the 5 words are added.

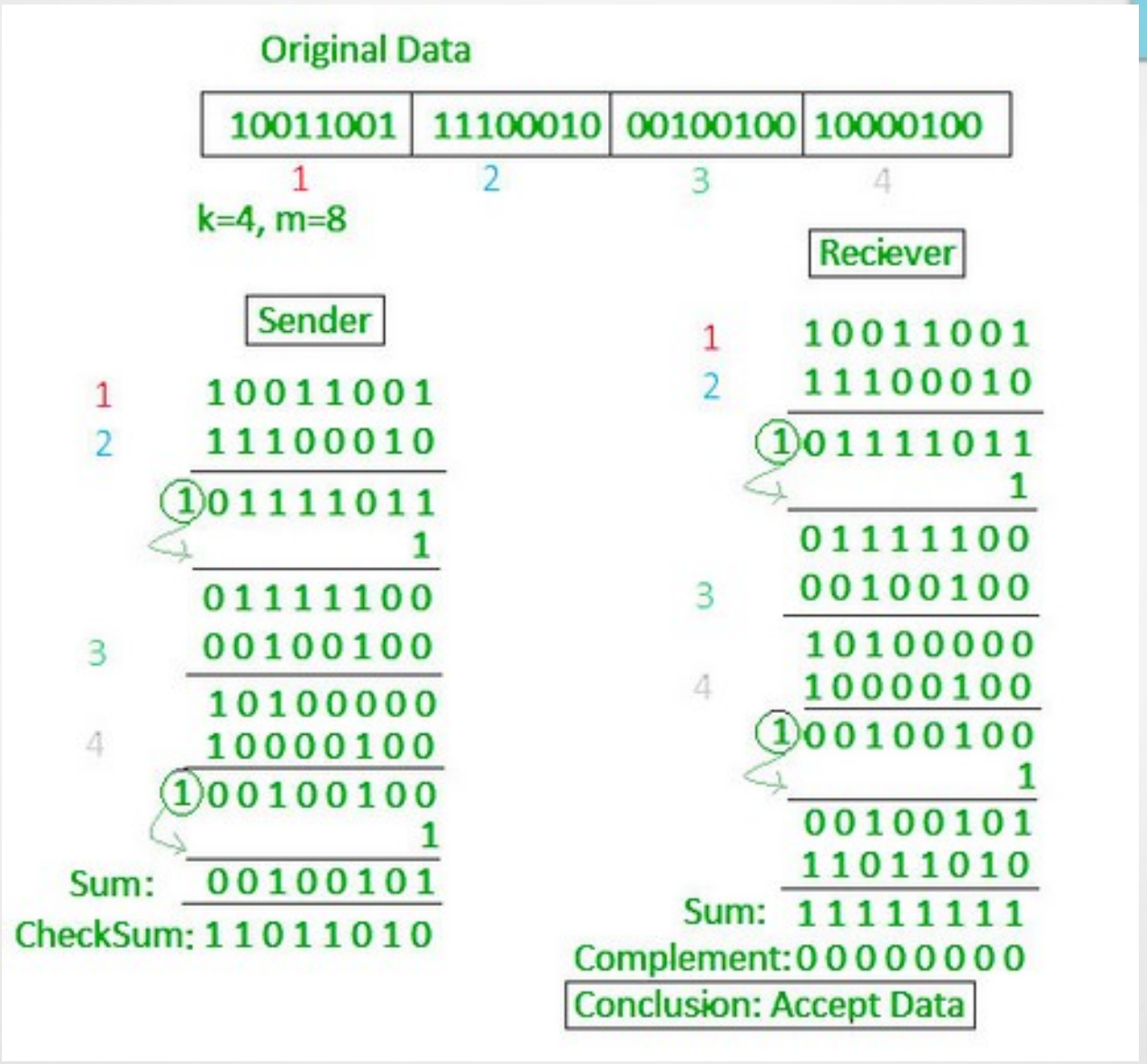
```
  1100
  1000
  0111
  1110
  0110
 101111
```

- The result is 101111. Find complement – 010000. Then 2's complement of the result is found, which is equal to 010001. This value is considered as the checksum and is sent along with the data. The receiver computes a fresh checksum and compares it with received checksum.

Error Detection

- **Checksum:**
- Example- 4-bit numbers are: 7, 11, 12, 0, 6
- Message to be sent:7,11,12,0,6,36 [7+11+12+0+6=36]
- Receiver adds actual nos. and compares with the sum(36)
- If the two are the same receiver assumes no error, nos. accepted and sum discarded.
- Else an error somewhere and data are not accepted.

Error Detection- checksum

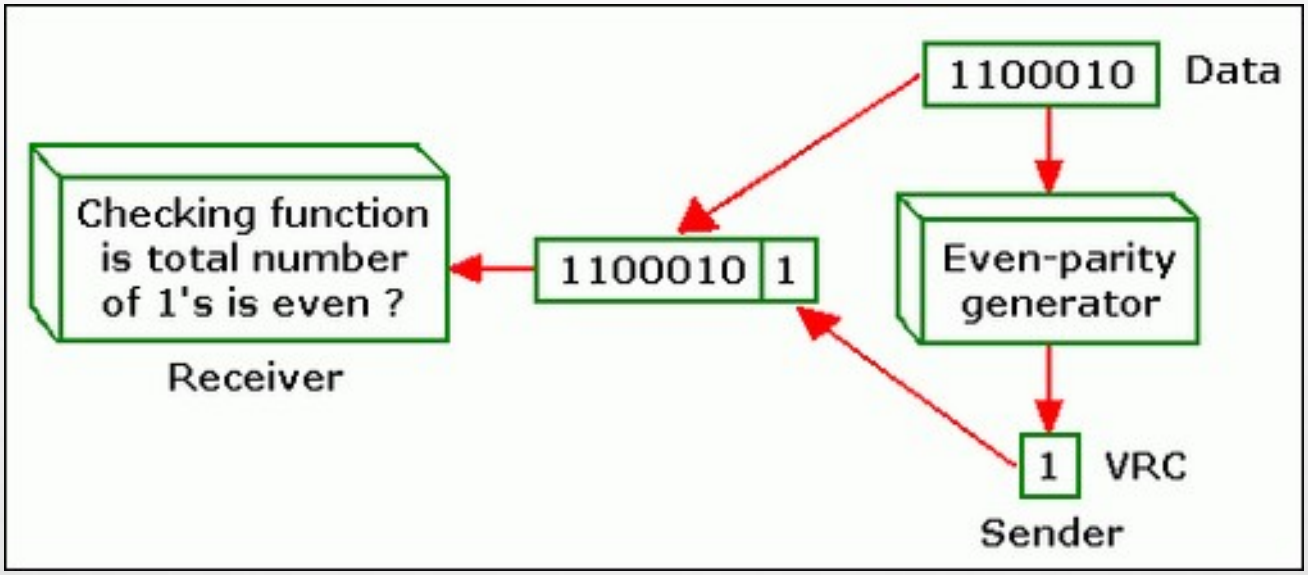


Error Detection

- **Vertical Redundancy Check(VRC) or Parity Check:**
- **Vertical Redundancy Check(VRC)** also known as **parity check**.
- It is least expensive technique.
- In this method, the sender appends a single additional bit called the **parity bit,** to the message before transmitting it.
- There are two types i.e. **odd parity and even parity.**
- In **odd parity,** given some bits, an additional bit is added in such a way that the number of 1s in the bits inclusive of the parity bit is **odd**.
- In **even parity**, the parity bit is added such that the number of 1s inclusive if the parity bit is **even**.

Error Detection

- **Vertical Redundancy Check(VRC) or Parity Check:**
- Suppose we want to transmit the binary data unit 1100001, adding the number of 1s gives us 3, an odd number. Before transmitting, a parity generator counts the 1s and appends the parity bit (a 1 in this case) to the end. The total number of 1 becomes 4 now (even number). The system now transmits the entire appended unit across the network link.



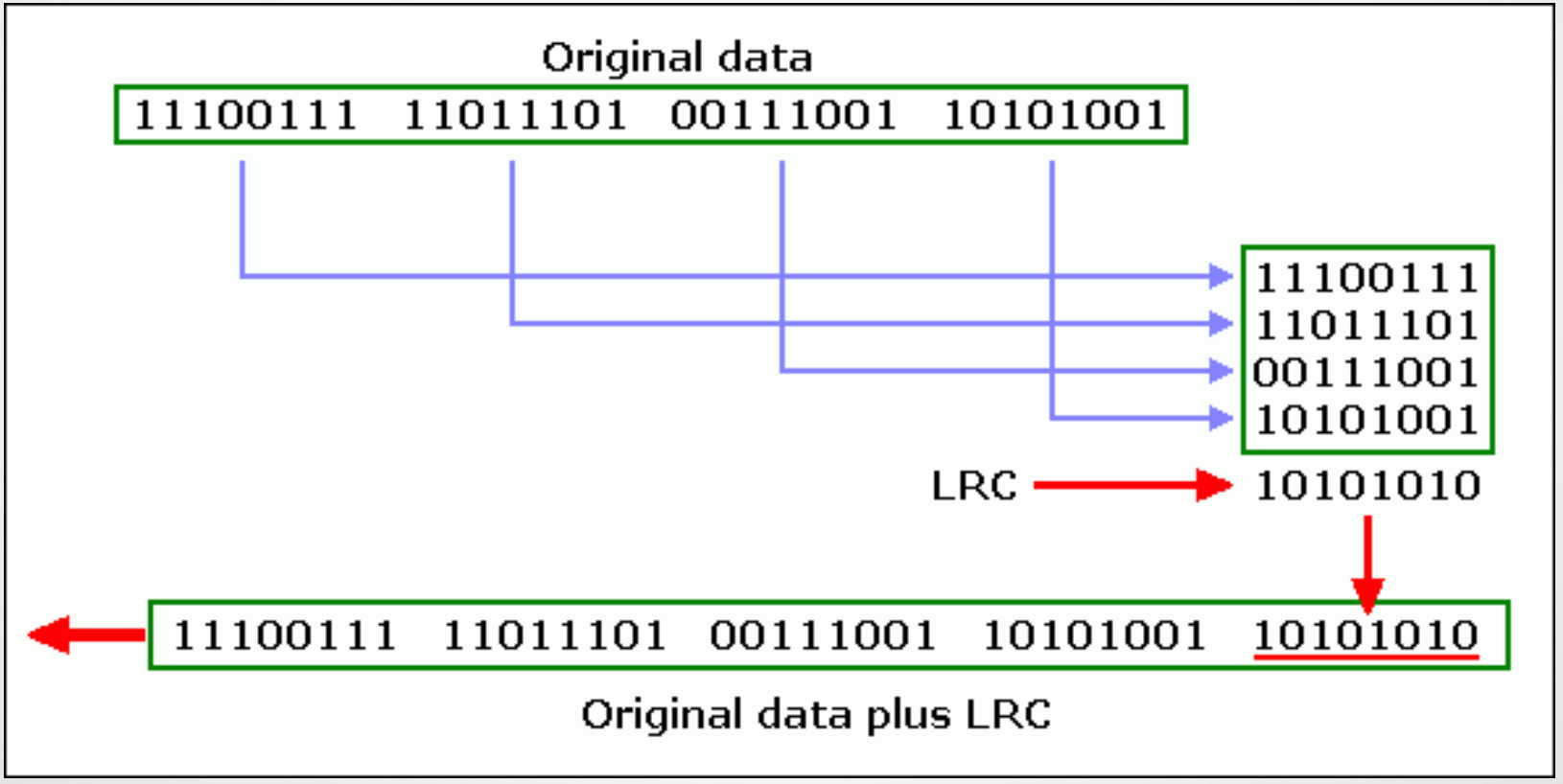
Error Detection

- **Vertical Redundancy Check(VRC) or Parity Check:**
- When the data unit is reached its destination, the receiver puts all eight bits through an even-parity checking function. If the receiver sees 11100001, it counts and gets four 1s, an even number.
- But if the receiver sees 11100101, or total number of 1s is odd. The receiver knows that an error has been occurred into the data somewhere and therefore rejects the whole unit.
- **This method VRC can detect only single-bit errors.**

Error Detection

- **Longitudinal Redundancy Check(LRC):**
- In this error detection method, a block of bits is organized in a table with rows and columns.
- For instance, if we want to send 32 bits we arrange them into a list of four rows. Then parity bit for each column is calculated and a new row of eight bits is created. These become the parity bits for the whole block.
- After that the new calculated parity bits are attached to the original data and sends to the receiver.
- LRC increases the likelihood of detecting burst error. An LRC of n bits can easily detects a burst error of n bits.

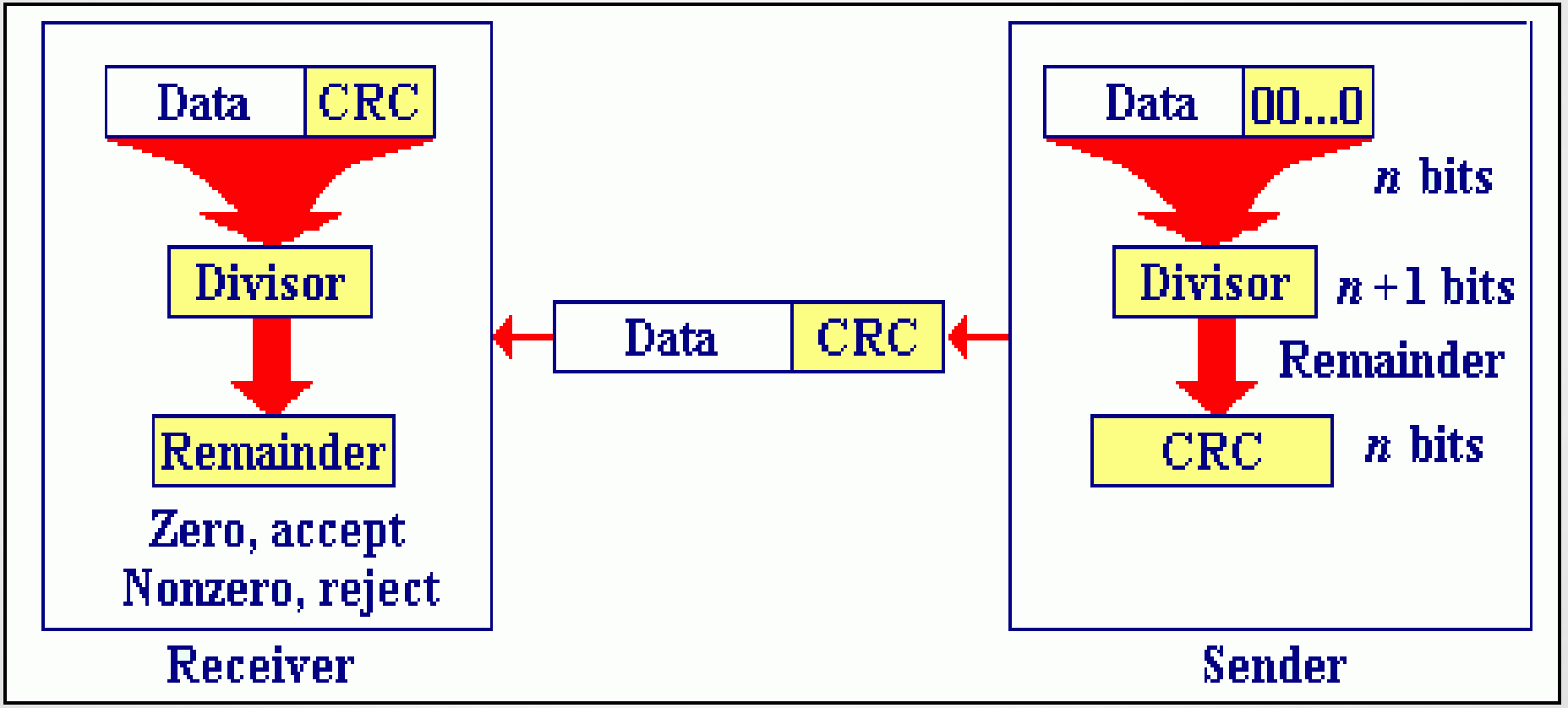
Error Detection



Error Detection

- **Cyclic Redundancy Check(CRC):**
- In Cyclic Redundancy Check(CRC) a sequence of redundant overhead bits called CRC or CRC remainder is added to the end of the data to be transmitted.
- The CRC is so calculated that it can be perfectly divided by a second predecided number. At the receiver, the arriving data is divided by the same predecided number.
- If this division produces a zero remainder, the transmission is considered as error-free.
- In such a case, the incoming data is accepted by the receiver. If there is a remainder, it means that the transmission is in error and therefore the arriving data must be rejected.

Error Detection



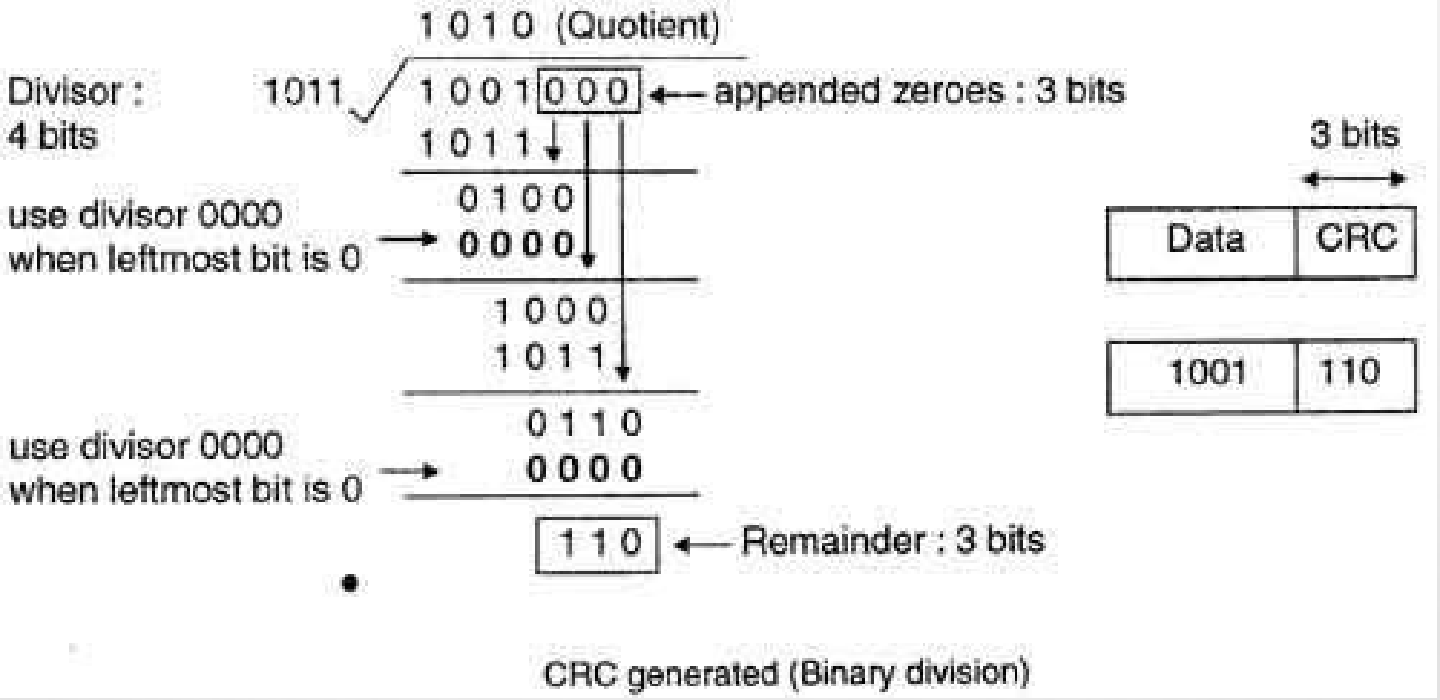
Error Detection

Cyclic Redundancy Check(CRC):

1)Data unit 1001000 is divided by 1011.

2)During this process of division, whenever the leftmost bit of dividend or remainder is 0, we use a string of Os of same length as divisor. Thus in this case divisor 1011 is replaced by 0000.

3)At the receiver side, data received is 1001110.



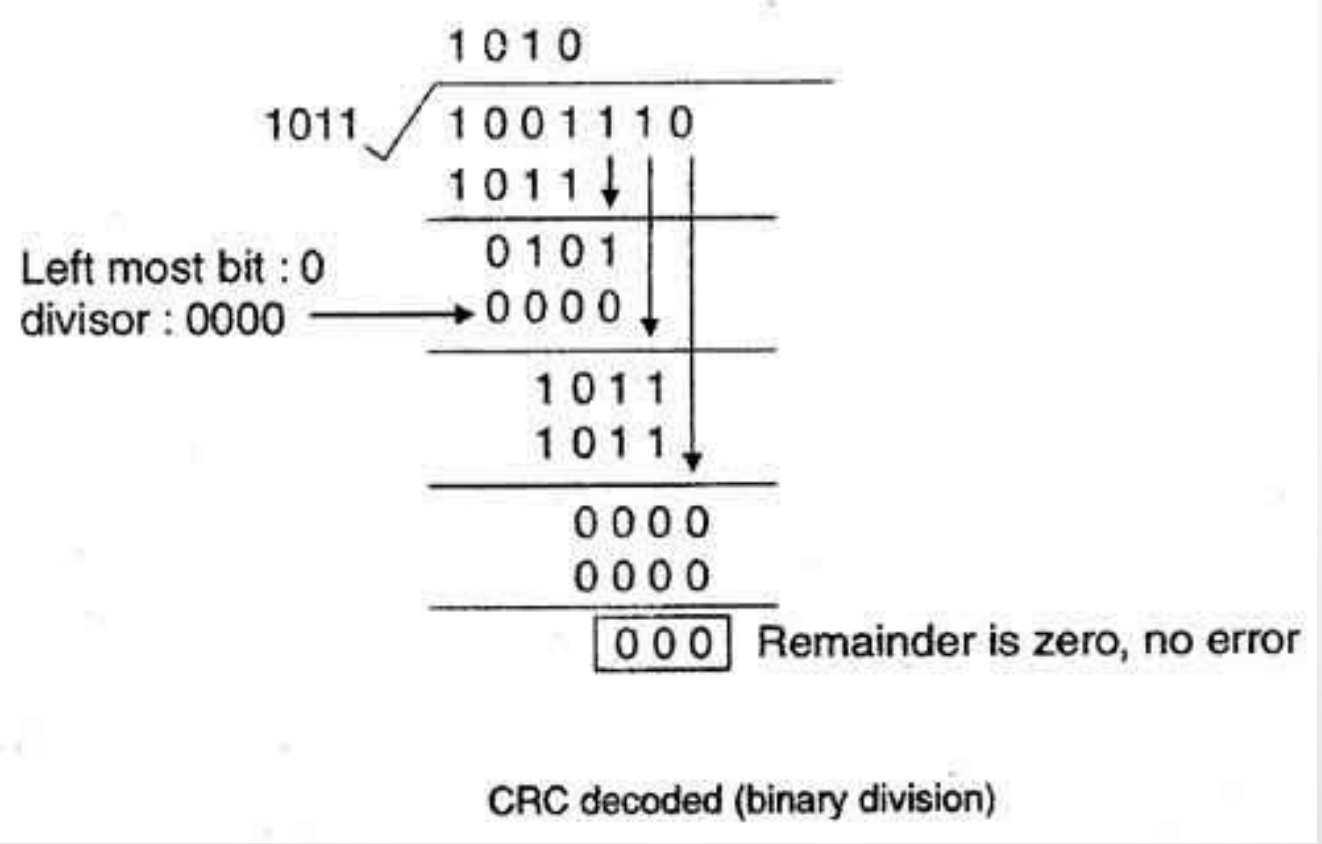
Error Detection

- **Cyclic Redundancy Check(CRC):**

4) This data is again divided by a divisor 1011.

5) The remainder obtained is 000; it means there is no error.

Next example: message: 1100101, polynomial: 11011



CRC decoded (binary division)

Steps:

- **Cyclic Redundancy Check(CRC):**

n : Number of bits in data to be sent from sender side.
k : Number of bits in the key obtained from generator polynomial.

Sender Side (Generation of Encoded Data from Data and Generator Polynomial (or Key)):

The binary data is first augmented by adding k-1 zeros in the end of the data
Use modulo-2 binary division to divide binary data by the key and store remainder of division.
Append the remainder at the end of the data to form the encoded data and send the same

Receiver Side (Check if there are errors introduced in transmission)

Perform modulo-2 division again and if the remainder is 0, then there are no errors.

Steps:

Modulo 2 Division:

- The process of modulo-2 binary division is the same as the familiar division process we use for decimal numbers. Just that instead of subtraction, we use XOR here.
- In each step, a copy of the divisor (or data) is XORed with the k bits of the dividend (or key).
- The result of the XOR operation (remainder) is (n-1) bits, which is used for the next step after 1 extra bit is pulled down to make it n bits long.
- When there are no bits left to pull down, we have a result. The (n-1)-bit remainder which is appended at the sender side.

Error Detection