Networking Essentials
ITC 2243

KAVINDU CHETHIYA YAKUPITIYA

PHD (IS-READING), MSC (CS), PGD (CS), BSC (IT), DIP (TECH, IT), CCNA, NSE (CERT)

Errors

Data communications refers to the transmission of digital data between two or more computers using various channels/media. Many communication channels are subject to channel noise, and thus errors may be introduced during transmission from the source to a receiver.

In order to understand how errors can be controlled, it is essential to know the types of errors.

Types of Errors

- **Single bit error**

In a frame, there is only one bit, anywhere though, which is corrupt.

Errors

- **Multiple bits error**

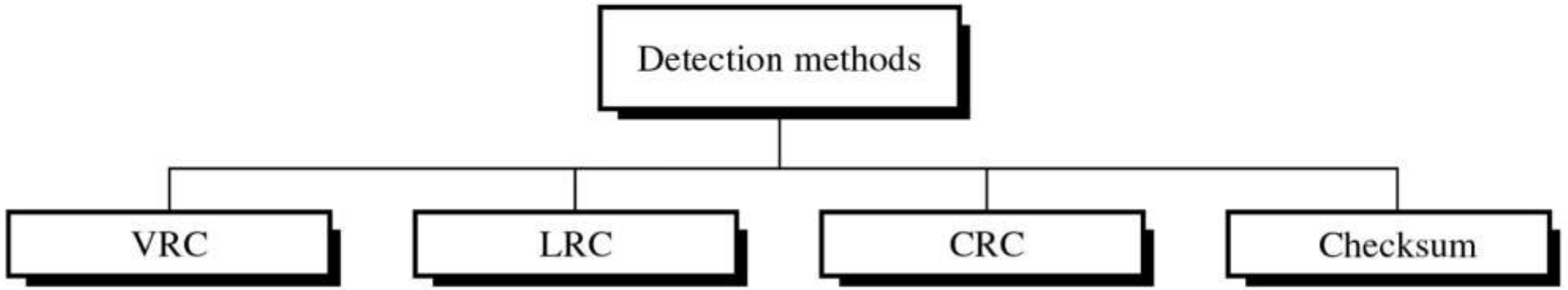
Frame is received with more than one bits in corrupted state.

- **Burst error**

Frame contains more than1 consecutive bits corrupted.

Error Detection Methods

- Error detection uses the concept of redundancy, which means adding extra bits for detecting errors at the destination.
- Error detection methods:
  - ✔ VRC(Vertical Redundancy Check)
  - ✔ LRC(Longitudinal Redundancy)
  - ✔ CRC(Cyclic redundancy Check)
  - ✔ Checksum



Vertical Redundancy Check (VRC)

VRC (Vertical Redundancy Check)

A parity bit is added to every data unit so that the total number of 1s(including the parity bit) becomes even for even-parity check or odd for odd-parity check

VRC can detect all single-bit errors.

Example: even parity

- 1000000(1)
- 1111101(0)
- 1001001(1)

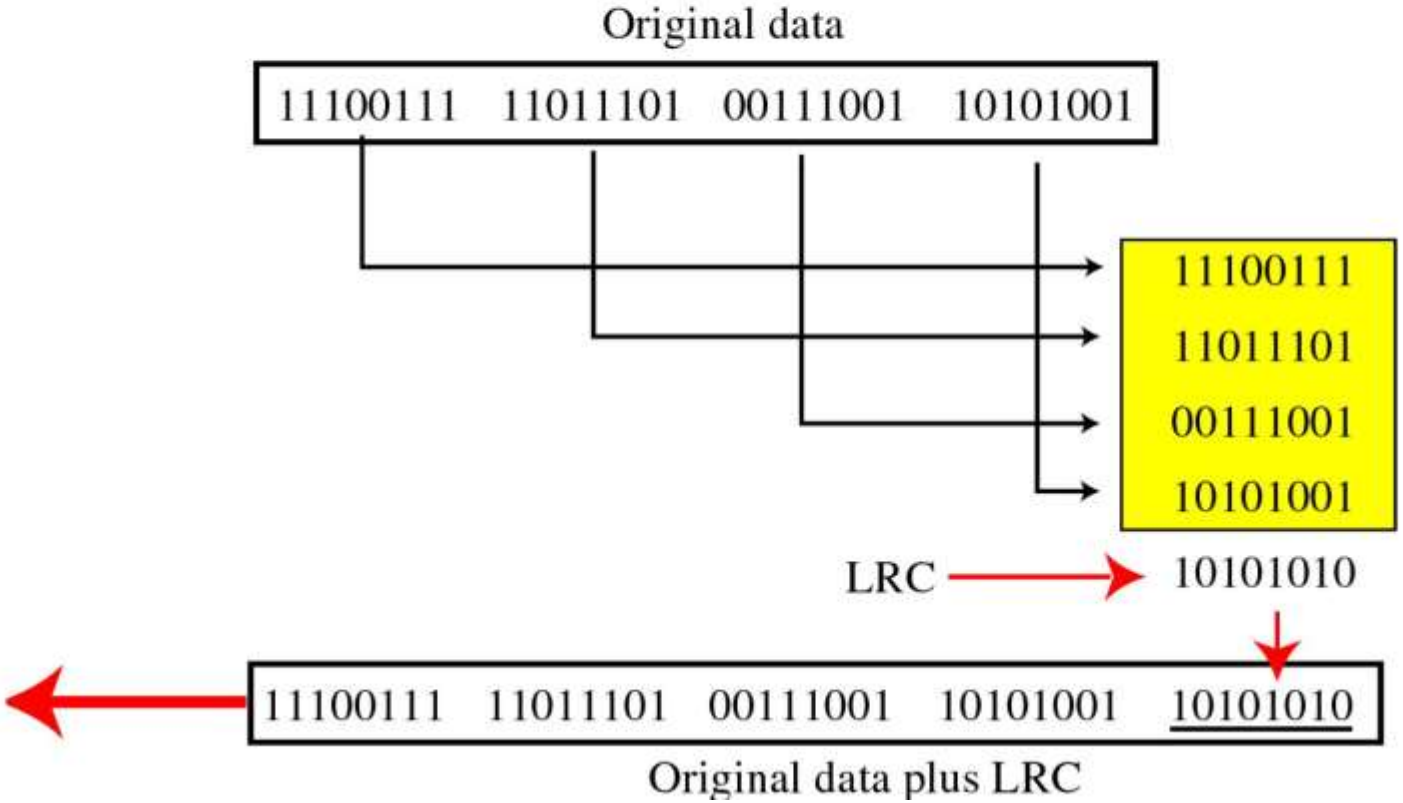
Example: odd parity

- 1000000(0)
- 1111101(1)
- 1001001(0)

Longitudinal Redundancy Check (LRC)

LRC (Longitudinal Redundancy)

Parity bits of all the positions are assembled into a new data unit, which is added to the end of the data block



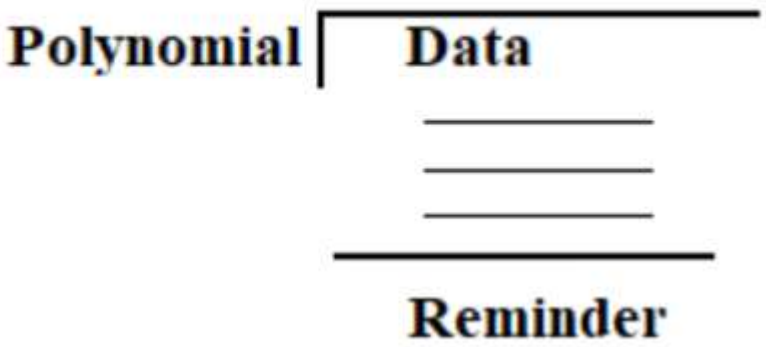
Cyclic Redundancy Check

One of the most popular methods of error detection for digital signals is the Cyclic Redundancy Check (CRC). CRCs are useful because they are capable of detecting all single and double errors and many multiple errors with a small number of bits.

Communications protocols often use two CRCs in a packet - one to protect the header of the packet and another to protect the data portion of the packet.

CRC is based on binary division. The remainder number is appended to the message and sent. At the destination, the incoming data unit is divided by the same number. If at this step there is no remainder, the data unit received is correct and accepted. A remainder indicates that the data unit has been damaged.

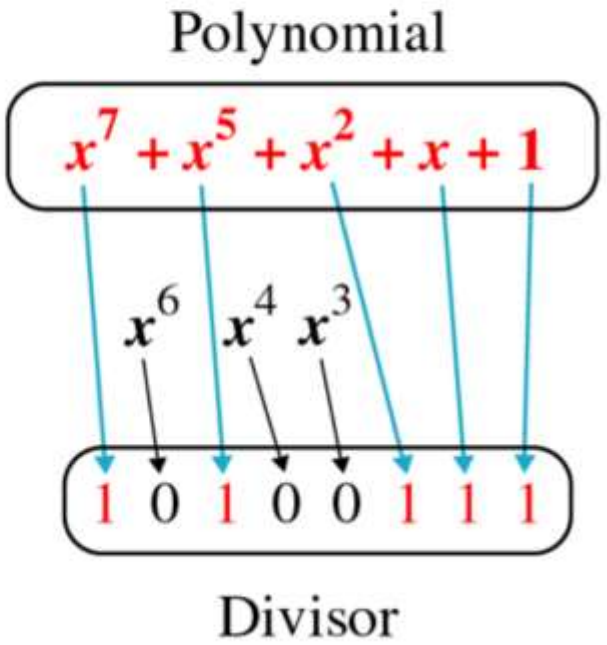
Cyclic Redundancy Check

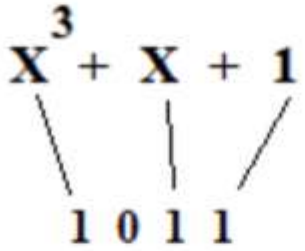


Transmitted frame = DataReminder

Cyclic Redundancy Check

Calculating a polynomial:





Cyclic Redundancy Check

Polynomial arithmetic is done using modulo 2:

$$\begin{array}{r} 10110011 \\ +\ 11010110 \\ \hline 01100101 \end{array} \qquad \begin{array}{r} 01110001 \\ +\ 10011010 \\ \hline 11101011 \end{array}$$

Cyclic Redundancy Check

What is the transmitted frame for data 1101011011 using generator $x^4 + x + 1$:

Frame : 1 1 0 1 0 1 1 0 1 1
Generator: 1 0 0 1 1
Message after 4 zero bits are appended: 1 1 0 1 0 1 1 0 1 1 0 0 0 0

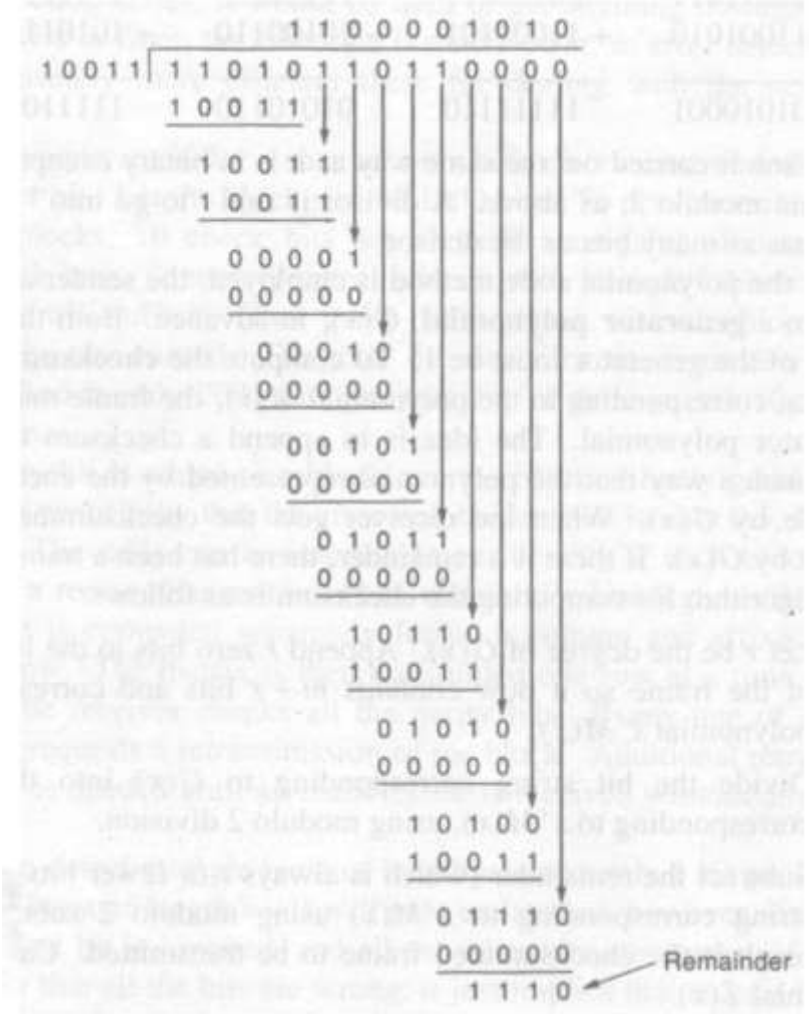
Cyclic Redundancy Check

Frame : 1 1 0 1 0 1 1 0 1 1

Generator: 1 0 0 1 1

Message after 4 zero bits are appended: 1 1 0 1 0 1 1 0 1 1 0 0 0 0

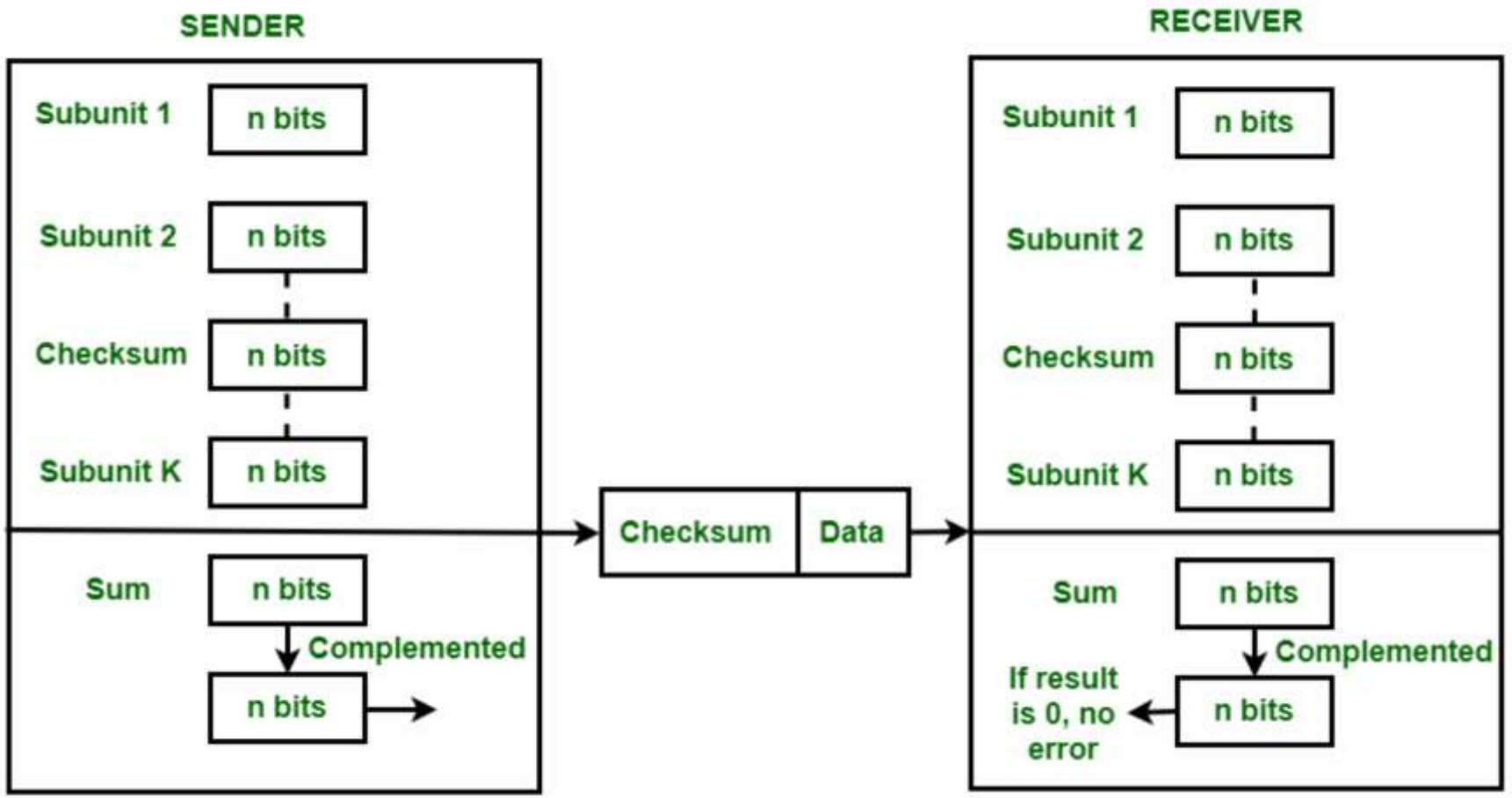
Transmitted frame: 1 1 0 1 0 1 1 0 1 1 1 1 1 0



Checksum

- In checksum error detection scheme, the data is divided into k segments each of m bits.
- In the sender's end the segments are added using 1's complement arithmetic to get the sum. The sum is complemented to get the checksum.
- The checksum segment is sent along with the data segments.
- At the receiver's end, all received segments are added using 1's complement arithmetic to get the sum. The sum is complemented.
- If the result is zero, the received data is accepted; otherwise discarded.

Checksum


SENDER
Subunit 1
n bits
Subunit 2
n bits
Checksum
n bits
Subunit K
n bits
Sum
n bits
Complemented
n bits
Checksum
Data
RECEIVER
Subunit 1
n bits
Subunit 2
n bits
Checksum
n bits
Subunit K
n bits
Sum
n bits
Complemented
If result is 0, no error
n bits

Checksum

Example –

If the data unit to be transmitted is 10101001 00111001, the following procedure is used at Sender site and Receiver site.

Sender Site :

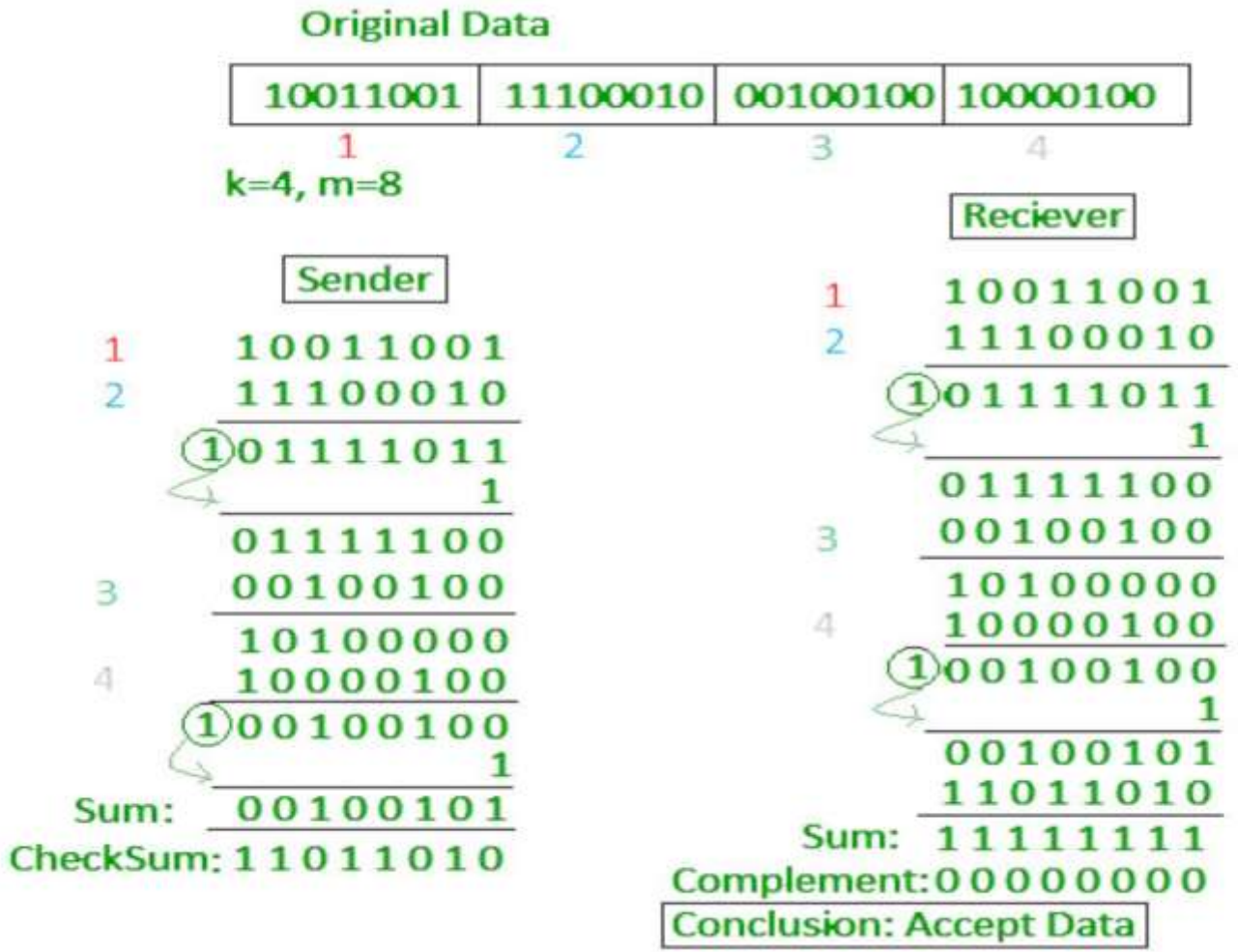
```
10101001        subunit 1
00111001        subunit 2
--------
11100010        sum (using 1s complement)
00011101        checksum (complement of sum)
```

Receiver Site :

```
10101001        subunit 1
00111001        subunit 2
00011101        checksum
--------
11111111        sum
00000000        sum's complement

Result is zero, it means no error.
```

Checksum


Original Data
10011001 11100010 00100100 10000100
1 2 3 4
k=4, m=8
Sender
1 10011001
2 11100010
10111011
1
01111100
3 00100100
10100000
4 10000100
100100100
1
Sum: 00100101
CheckSum: 11011010
Reciever
1 10011001
2 11100010
10111011
1
01111100
3 00100100
10100000
4 10000100
100100100
1
00100101
11011010
Sum: 11111111
Complement: 00000000
Conclusion: Accept Data

Error Correction

Error Correction codes are used to detect and correct the errors when data is transmitted from the sender to the receiver.

Error Correction can be handled in two ways:

- Backward error correction:

  Once the error is discovered, the receiver requests the sender to retransmit the entire data unit.

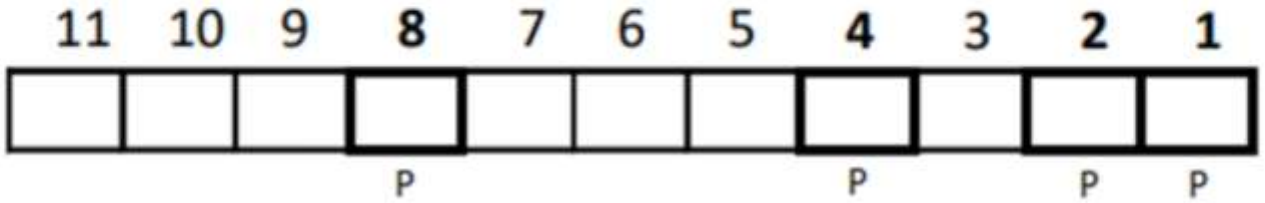
- Forward error correction:

  In this case, the receiver uses the error-correcting code which automatically corrects the errors.

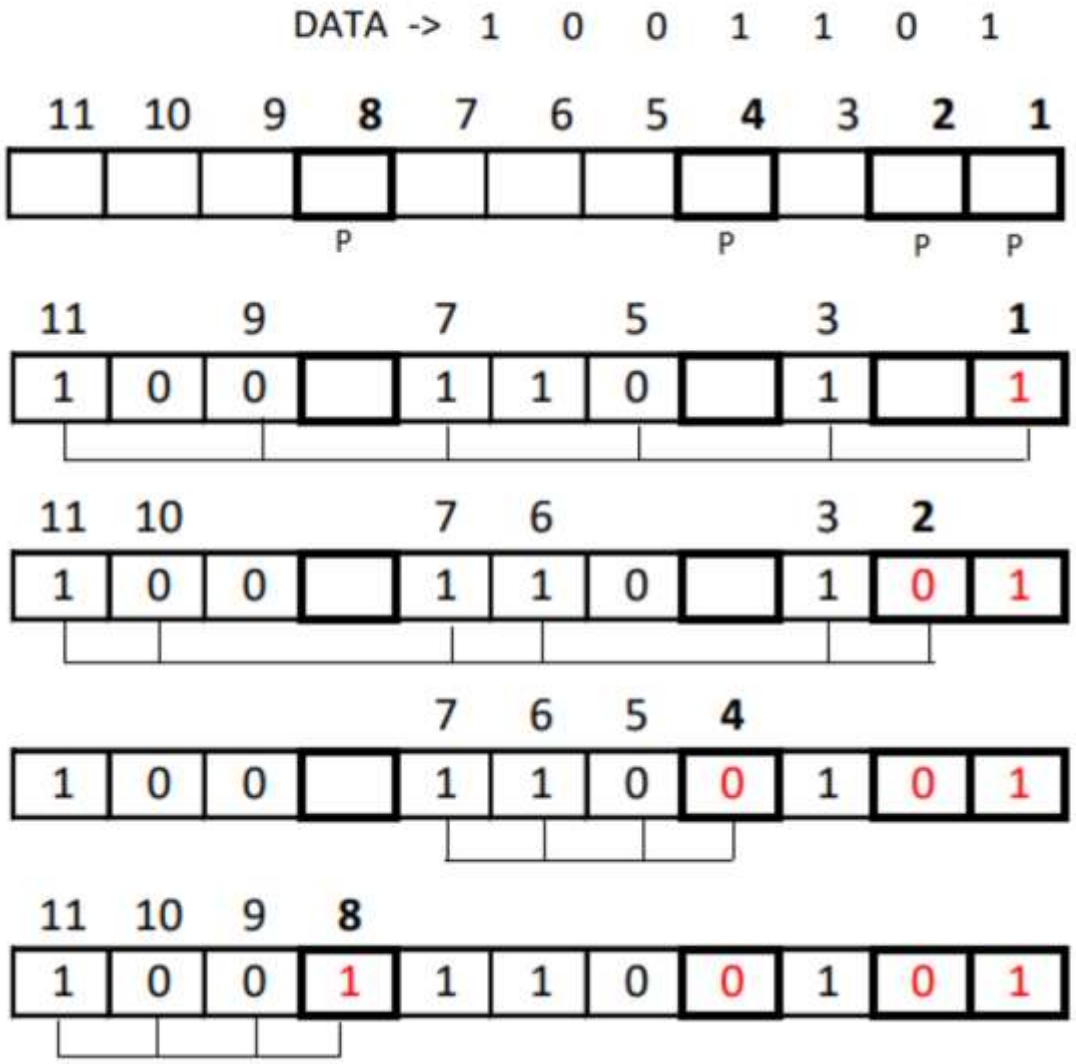
Hamming Code

Hamming Code

- Hamming code is capable of detecting two bit errors and correcting single-bit errors. It was developed by R.W. Hamming for error correction.
- In this coding method, the source encodes the message by inserting redundant bits within the message. These redundant bits are extra bits that are generated and inserted at specific positions in the message itself to enable error detection and correction. When the destination receives this message, it performs recalculations to detect errors and find the bit position that has error.



Hamming Code



Compute VRC/Even parity for 1st Parity

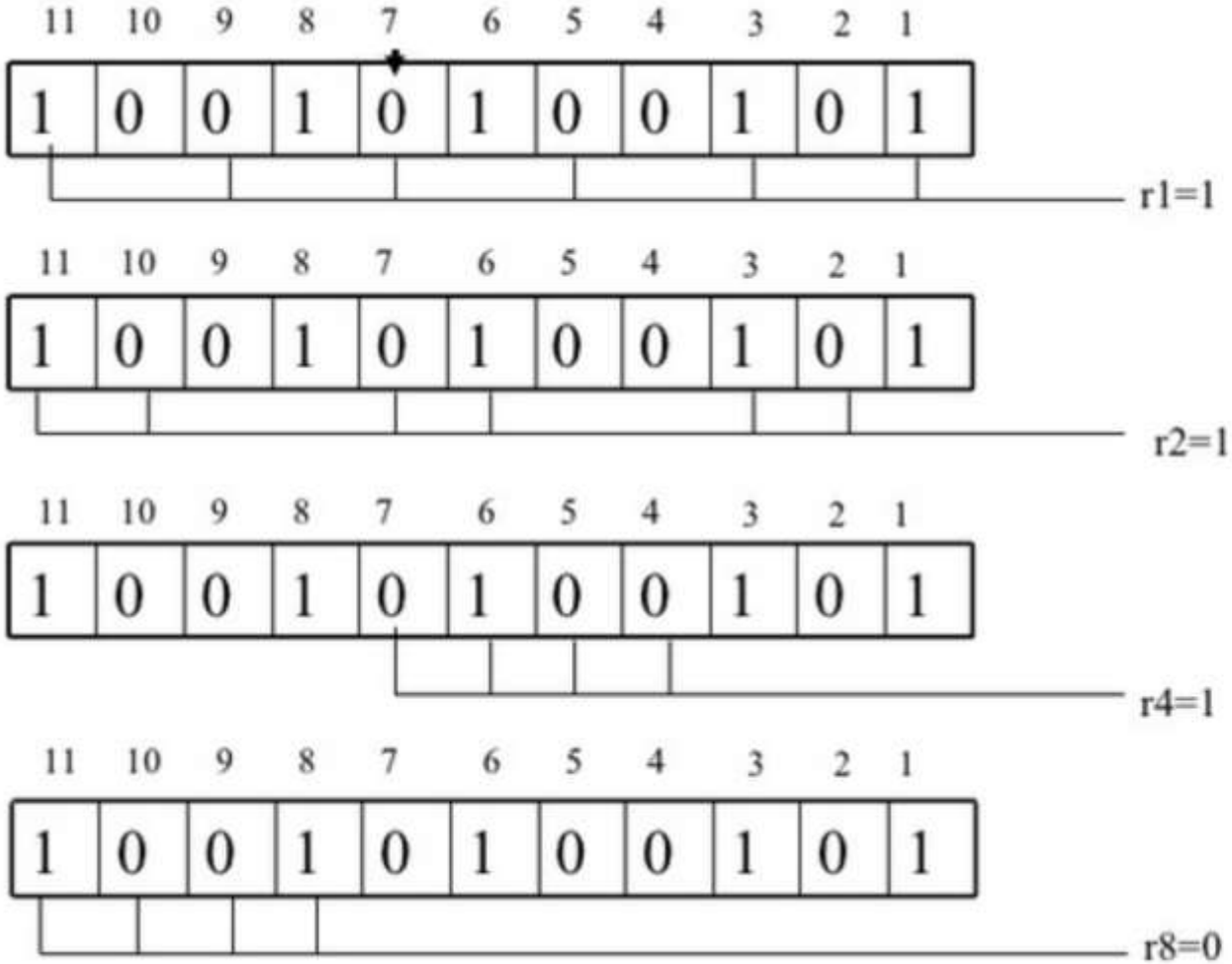
Compute VRC/Even parity for 2nd Parity

Compute VRC/Even parity for 4th Parity

Transmitted frame:

10011100101

Hamming Code



Received frame:
10010100101

8 4 2 1

0 1 1 1

7

It mean the 7 bit is corrupted

Thank You!!!