Terna Engineering College

Computer Engineering Department

Program: Sem V

Course: Computer Network Lab

Faculty: Umesh B Mantale, D V Thombre and Ramesh Shahabade

LAB Manual

PART A

(PART A: TO BE REFERRED BY STUDENTS)

Experiment No. 6

A.1 Objective:

Implementation of a Cyclic Redundancy Code (CRC) generator and checker using any higher level language.

A.2 Prerequisite:

- Knowledge about PAN, LAN and NW Elements.
- Knowledge of Programming Languages.
- Binary arithmetic.
- Error types and their detection and correction.
- Concept of Programming, Analysis, Design, Simulation and Modelling.

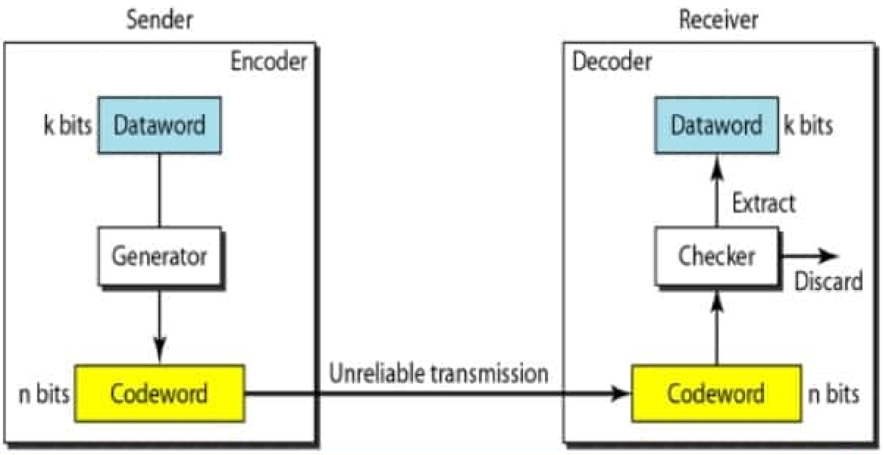
A.3 Outcome:

After successful completion of this experiment students will be able to -

- Ability to select the proper NW Elements required to design NWs.
- Thorough understanding of DLL.
- Error detection methodologies and their implementation.
- Hard coding by applying their programming skills.

A.4 Theory/Tutorial:

Figure 10.6 *Process of error detection in block coding*



Cyclic Redundancy Check (CRC)

- A code added to data which is used to detect errors occurring during transmission, storage, or retrieval.
- **CRC is a redundancy error technique used to determine the error**.
- **Following are the steps used in CRC for error detection:**

Cyclic Redundancy Check (CRC)

- **Sender:**
- 1. 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.**
- 2. 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**.
- 3. **Thirdly, the CRC remainder replaces the appended 0s at the end of the original data.**

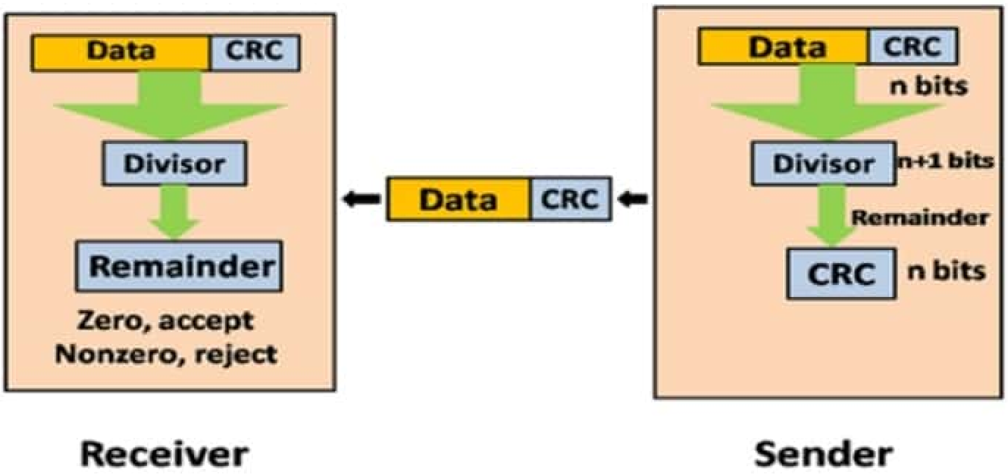
This newly generated unit is sent to the receiver.

Cyclic Redundancy Check (CRC)

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.**

Cyclic Redundancy Check (CRC)



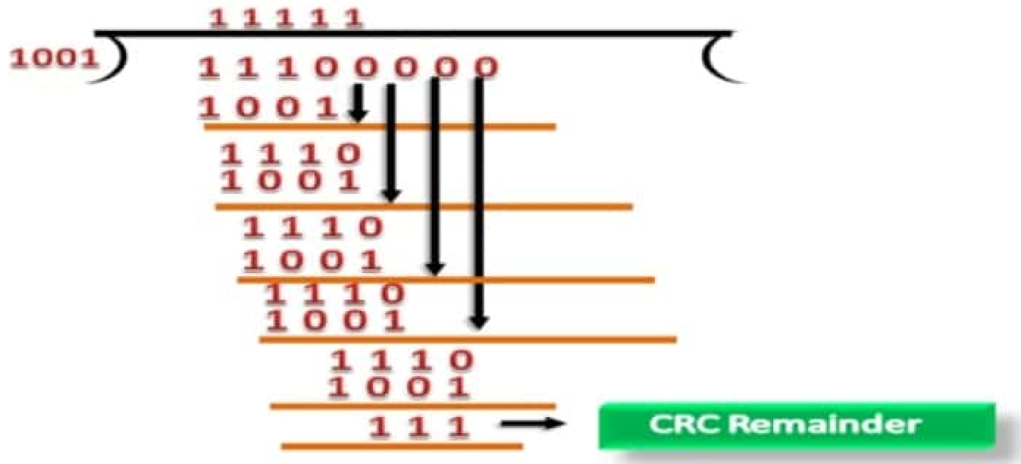
Cyclic Redundancy Check (CRC)

- 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 11100**000**, 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.

Modulo 2 Division:

- **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).

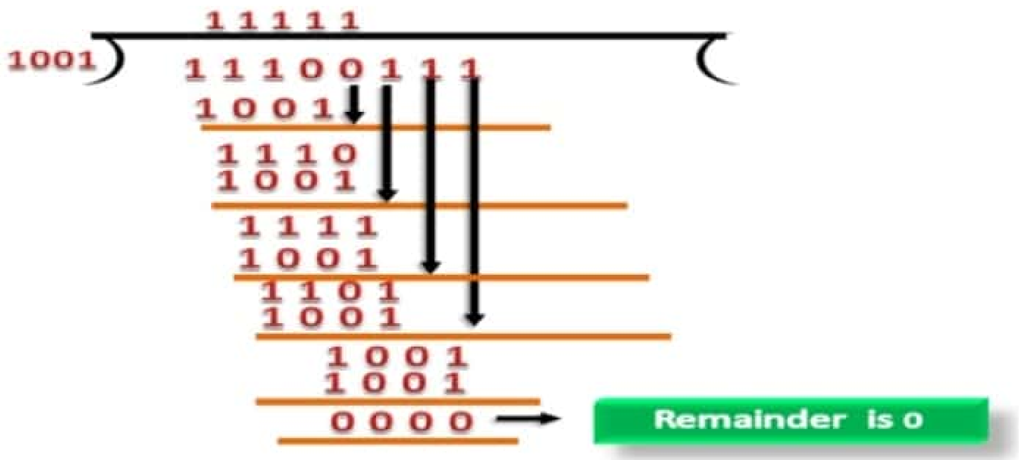
Cyclic Redundancy Check (CRC)



Cyclic Redundancy Check (CRC)

- **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.

Cyclic Redundancy Check (CRC)



Reference:

- https://www.javatpoint.com/computer-network-error-detection

PART B

(PART B: TO BE COMPLETED BY STUDENTS)

(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case the there is no Blackboard access available)

Roll No. 50	Name: Amey Thakur
Class: TE-Comps B	Batch: B3
Date of Experiment: 31/08/2020	Date of Submission: 31/08/2020
Grade :	

B.1 Document created by the student:

(Write the answers to the questions given in section 5.1 during the 2 hours of practical in the lab here)

Refer B.5

B.3 Observations and learning:

(Students are expected to understand the selected topic. Have to list out the components & functionality. Prepare a flow of the algorithm defined in the paper. List the performance metrics that is used)

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.

B.4 Conclusion:

(Students must write the conclusion as per the attainment of individual outcome listed above and learning/observation noted in section B.3)

CRC or Cyclic Redundancy Check is a method of detecting accidental changes/errors in the communication channel. CRC uses Generator Polynomials which are available on both the sender and receiver side.

Computer Networks Laboratory Experiment 6

Amey Thakur D.O.P. - 31.08.2020
TE Comps B-50 D.O.S. - 31.08.2020
B3

Ans:

Roll number: 50
Add : 128
178

Convert it to Binary

Step 1: Divide $(178)_{10}$ successively by 2 until the quotient is zero.

2	178		
2	89	0	LSB
2	44	1	
2	22	0	
2	11	0	
2	5	1	
2	2	1	
2	1	0	
	0	1	MSB

Step 2: Read from the bottom (MSB) to top (LSB)

$$(178)_{10} = (10110010)_2$$

Q 2.

Ans:

Data word to be sent - 10110010

Key - 1010

[Cyclic Redundancy Check and Modulo-2 Division]

Sender's side

```
⇒              10010111
      1010 | 10110010000
             1010
             00010010000
               0000
               0100100000
               0000
                 10010000
                 1010
                 00110000
                   0000
                    110000
                    1010
                     011000
                      1010
                      01100
                       1010
                       [0110]
```

Therefore, the remainder is 110

Hence, the encoded data sent is 10110010110

Receiver's side

⇒

code word received at the receiver side

⇒ 10110010110

```
            10010111000
1010 | 101100101100000
       1010
       00010010110000
        0000
         0100101100000
         0000
          10010110000
          1010
           00110110000
            0000
             110110000
             1010
             011110000
              1010
              01010000
               1010
               00000000
                0000
                 00000
                 0000
                  0000
                  0000
                   000
```

Therefore, the remainder is all zeros.

Hence, data received has no error.

C. Results verified.

B.5. Question of Curiosity

Q.1. What is an error? Name the types of error?

Ans:

- An error is something which is considered to be incorrect or wrong. or which should not have been done.
- There are three types of errors.
  1. Syntax errors
  2. Logical errors / semantic errors
  3. Run time errors

Q.2. Single bit error is found in parallel transmission Give reason.

Ans:

- Single bit error can happen in parallel transmission where all the data bits are transmitted using separate wires.

  Single bits error are therefore found in parallel transmission.

Q.3. Burst error is normally found in serial transmission. Give reason.

Ans:

- Burst errors are most likely to happen in serial transmission because the duration of the noise is normally longer than the duration of a single bit. which means that the noise affects data. It affects a set of bits.
- The length of burst error is measured from first changed bit to last changed bit.

Q4. What are even and odd parities.? State the limitation of single parity check and two dimensional parity check.

Ans:

Even Parity

- Refers to the parity checking mode in which each set of transmitted bits must have an even number of set bits. ~~To parity~~
- The parity checking system on the sending side ensures even parity by setting the extra parity bit if necessary

Odd Parity

- Refers to the mode of parity checking in which each 9-bit combination of a data byte plus a parity bit contains an odd number of set bits.

Limitation of Single parity check
- Its primary disadvantage is that it may fail to catch errors.
- If two data bits are corrupted, for instance, parity will not detect the error.

Limitation of Two dimensional parity check
- In some cases, an only odd number of bit errors can be detected and corrected but even number of errors can only be detected but not corrected.
- In some cases, this method is not able to detect even bit error.

Q5. What are the redundant bit generator and error checker?

Ans:
- Whenever message is transmitted, it may get scrambled by noise or data may get corrupted. To avoid this, we use error-detecting codes or we generate a redundant bit while transmitting message. which are additional data added to a given digital message to help detect if any error has occured during transmission of the message.
- Some popular error checker techniques
    1. Simple parity check
    2. Two dimensional parity check
    3. Checksum
    4. Cyclic Redundancy Check.

Q.6. State the working of CRC error detection with an example of your own.

Ans:

- Unlike checksum scheme, which is based on addition, CRC is based on binary division.
- In CRC, a sequence of redundant bits called cyclic redundancy check bits are appended to the end of data unit so that the resulting data unit becomes exactly divisible by a second, predetermined binary number.
- At the destination, the incoming data unit is divided by the same number. If at this step there is no remainder, the data unit is assumed to be correct and is therefore accepted.
- A remainder indicates that the data unit has been damaged in transit and therefore must be rejected.

- Example:

Data word to be sent - 10110010

Key - 1010

[Cyclic Redundancy Check and Modulo-2 Division]

Sender's side

```
        10010111
      ______________
1010 | 10110010000
       1010
       ----
       00010010000
        0000
        ----
        010010000
        0000
        ----
         10010000
         1010
         ----
         00110000
          0000
          ----
           110000
           1010
           ----
           011000
            1010
            ----
            01100
             1010
             ----
             0110
```

Therefore, the remainder is 110

Hence, the encoded data sent is 10110010110

Receiver's side

→

code word received at the receiver side

⇒ 10110010110

```
                100 10111000
1010 ) 1011 0010 11 0000
       1010
       0001 00 1011 0000
          0000
          0100 1011 00000
          0000
            100 10 110000
            1010
             0011011 0000
              0000
               11 011 0000
               1010
               011110000
                1010
                01010000
                 1010
                 00000000
                  0000
                   00000
                   0000
                    0000
                    0000
                     000
```

Therefore, the remainder is all zeros.

Hence, data received has no error.

Q.7. Which method is used for forward error correction?

Ans:

- Forward error correction is an error correction technique to detect and correct a limited number of errors in transmitted data without the need of retransmission.
- Error correction codes for FEC
  1. Block codes
  2. Convolution Codes
- Methods to find errors in FEC.
  1. Hamming codes
  2. Binary convolution code
  3. Reed - Solomon Code
  4. Low Density Parity Check Code