IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 5, Issue 1, May 2021
Impact Factor: 4.819

A Review on Generations of Various Error Detection Codes Using C Programming in Computer Network

Ms Aphasana Mulla[1], Mrs Shital Deshmukh[2], Ms Wrushali Deshmukh[3]
Lecturer, Department of Electronics and Telecommunication
Bharati Vidyapeeth Institute of Technology, Navi Mumbai, India

Abstract: *In today's world of computer networking, it is necessary to transmit and receive the error free data through any noisy channel in communication system. Because of the advancement in the data transmission the sources of noise and interference has also increased. Lots of efforts have been made by communication engineers to meet the demand for more reliable and efficient techniques for error detection in the received data. To detect the errors in the computer networking various techniques are used. This review paper delivers various error detection code generation techniques being used using C programming.*

Keywords: Error detection, Cyclic Redundancy Check (CRC), Parity check method, Vertical Redundancy Check (VRC), Longitudinal Redundancy Check (LRC)

I. INTRODUCTION

Error is a condition when the receiver's information does not match with the sender's information. An error correcting code (ECC) OR Forward error correction code (FEC) is a process of adding redundant data, or parity data, to a message, such that it can be detected when errors are introduced during the process of data transmission by a receiver. Various error detection methods exist in the communication system. The most common error detecting scheme being employed are parity bit, CRC, VRC and LRC. All these methods are implemented on the second layer of OSI model at Data link layer. The upper layers require error-free transmission between two systems. Almost every application did not work if it receives data with errors. Real time applications like voice and video may not get that much affected and may still function well with some error. Data-link layer uses some error control mechanism to ensure that data bit streams are transmitted with certain level of accuracy.

II. LITERATURE SURVEY

In computer networking error detection can be done with the help of C programming. Error detection codes like VRC, LRC ,CRC codes are generated using simulation software like C programming. [1] Various error detection and correction methods are being used to maintain good level of reliability, to protect memory cells using protection codes. The method used in [2], is based on the hardware and time redundancy, although this technique reduces the number of input and output pins of the combinational logic; it requires additional encoding/decoding circuitry. Process digital data transmission in communication can run safe premises but also there is an error [3] [4] . Error result in changes to the content of data transferred. There is a wide - range logic to detect and correct the error[5].

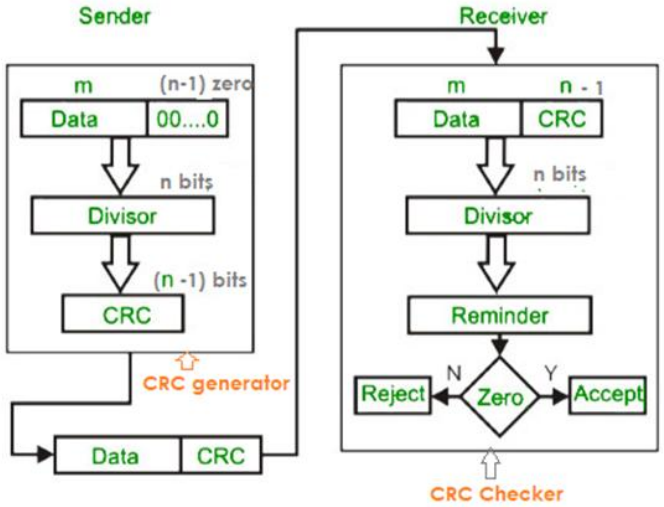
III. METHODS

Error detection code can be generated by the channel encoder with three methods using C programming

1. **CRC:** Cyclic redundancy check - 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.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 5, Issue 1, May 2021
Impact Factor: 4.819

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.



3.1 C Programming code for CRC generation

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
void main() {
        int i,j,keylen,msglen;
        char input[100], key[30],temp[30],quot[100],rem[30],key1[30];
        clrscr();
        printf("Enter Data: ");
        gets(input);
        printf("Enter Key: ");
        gets(key);
        keylen=strlen(key);
        msglen=strlen(input);
        strcpy(key1,key);
        for (i=0;i<keylen-1;i++) {
                input[msglen+i]='0';
        }
        for (i=0;i<keylen;i++)
        temp[i]=input[i];
        for (i=0;i<msglen;i++) {
                quot[i]=temp[0];
                if(quot[i]=='0')
                for (j=0;j<keylen;j++)
                 key[j]='0'; else
                 for (j=0;j<keylen;j++)
                 key[j]=key1[j];
                for (j=keylen-1;j>0;j--) {
```

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 5, Issue 1, May 2021
Impact Factor: 4.819

```
                        if(temp[j]==key[j])
                         rem[j-1]='0'; else
                         rem[j-1]='1';
                }
                rem[keylen-1]=input[i+keylen];
                strcpy(temp,rem);
        }
        strcpy(rem,temp);
        printf("\nQuotient is ");
        for (i=0;i<msglen;i++)
        printf("%c",quot[i]);
        printf("\nRemainder is ");
        for (i=0;i<keylen-1;i++)
        printf("%c",rem[i]);
        printf("\nFinal data is: ");
        for (i=0;i<msglen;i++)
        printf("%c",input[i]);
        for (i=0;i<keylen-1;i++)
         printf("%c",rem[i]);
        getch();
}
```

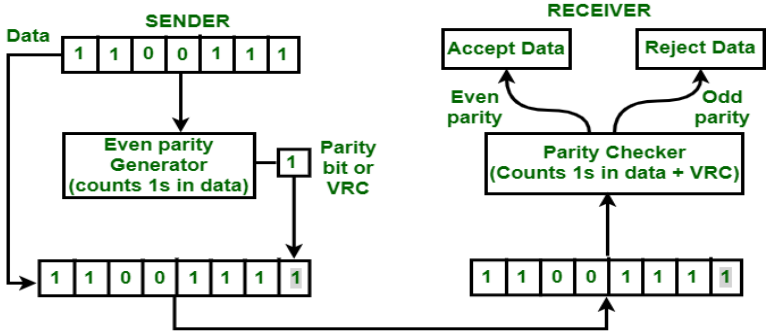
2. **Vertical Redundancy Check (VRC) or Parity Check:** Vertical Redundancy Check is also known as Parity Check. In this method, a redundant bit also called parity bit is added to each data unit. This method includes even parity and odd parity. Even parity means the total number of 1s in data is to be even and odd parity means the total number of 1s in data is to be odd.

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.

Parity generator will count number of 1s in data unit and will add parity bit. In the above example, number of 1s in data unit is 5, parity generator appends a parity bit 1 to this data unit making the total number of 1s even i.e 6 which is clear from above figure.

Data along with parity bit is then transmitted across the network. In this case, 11001111 will be transmitted. At the destination, This data is passed to parity checker at the destination. The number of 1s in data is counted by parity checker. If the number of 1s count out to be odd, e.g. 5 or 7 then destination will come to know that there is some error in the data. The receiver then rejects such an erroneous data unit.



IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 5, Issue 1, May 2021
Impact Factor: 4.819

3.2 C Programming code for VRC Generation for Odd Parity

```
#include<stdio.h>
#include<conio.h>
int main(){
        int data[3][5],i,j,a[4],count=0;
        clrscr();
        printf("Enter 3 numbers in 4 bit binary format e.g:-1100\n");
        for(i=0;i<3;i++){
                printf("%d binary number (4bit with space)\n", i+1);
                for(j=0;j<4;j++){
                        scanf("%d",&data[i][j]);
                }
        }
        for(i=0;i<3;i++){
                for(j=0;j<4;j++){
                        if(data[i][j]==1)
                        count++;
                }
                a[i]=count;
                count=0;
        }
        for(i=0;i<4;i++){
                if(a[i]%2!=0){
                        data[i][4]=0;
                }else{
                        data[i][4]=1;
                }
        }
        printf("\nGiven data\n");
        for(i=0;i<3;i++){
                for(j=0;j<4;j++){
                        printf("%d", data[i][j]);
                }
                printf("\n");
        }
        printf("\nData VRC\n");
        for(i=0;i<3;i++){
                for(j=0;j<5;j++){
                        printf("%d", data[i][j]);
                }
                printf("\n");
        }
        getch();
}
```

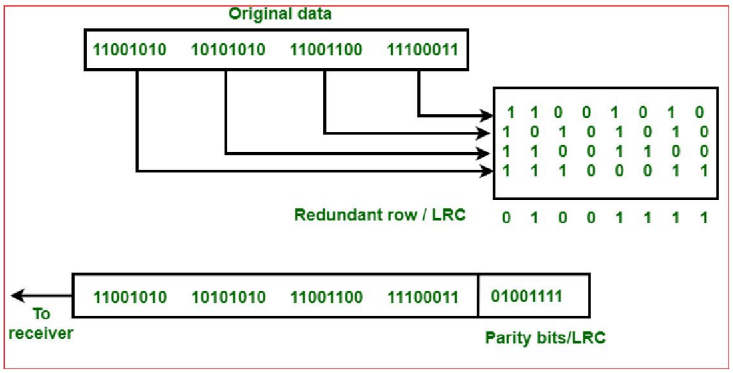
3. **Longitudinal Redundancy Check (LRC)/2-D Parity Check:** It is also known as 2-D parity check. In this method, data which the user want to send is organised into tables of rows and columns. A block of bit is

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 5, Issue 1, May 2021
Impact Factor: 4.819

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.

Example:

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.



3.3 C Programming Code for LRC Generation for Even Parity

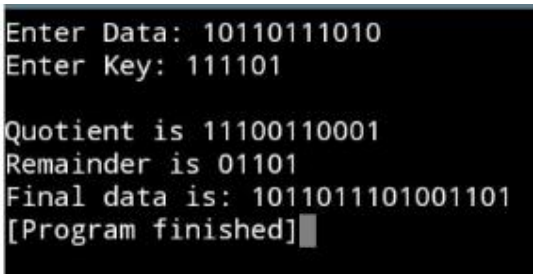
```
#include<stdio.h>
#include<conio.h>
void main(){
        int data[3][4],i,j,a[4],count=0;
        clrscr();
        printf("Enter 3 numbers in 4 bit binary format e.g:-1 1 0 0\n");
        for(i=0;i<3;i++){
                printf("%d binary number (4bit with space)\n",i+1);
                for(j=0;j<4;j++){
                        scanf("%d",&data[i][j]);
                }
        }
        for(i=0;i<4;i++){
                for(j=0;j<3;j++){
                        if(data[j][i]==1)
                        count++;
                }
                a[i]=count;
                count=0;
        }
        for(i=0;i<4;i++){
                if(a[i]%2==0){
                        a[i]=0;
                }else{
                        a[i]=1;
                }
```

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 5, Issue 1, May 2021
Impact Factor: 4.819

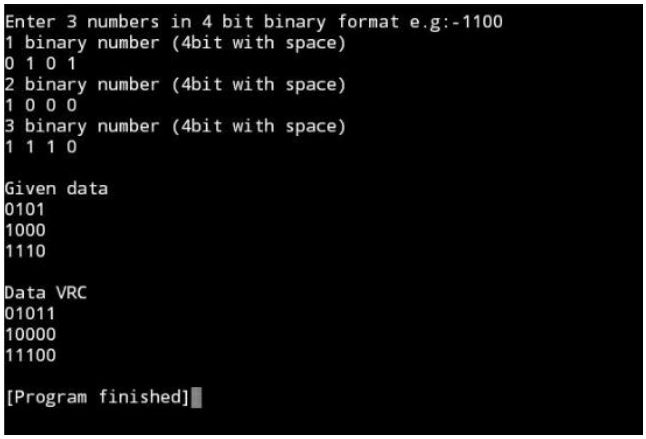
```
        }
        printf("\nGiven Data\n");
        for(i=0;i<3;i++){
                for(j=0;j<4;j++){
                        printf("%d ",data[i][j]);
                }
        printf("\n");
        }
        printf("\nData\n");
        for(i=0;i<3;i++){
                for(j=0;j<4;j++){
                        printf("%d ",data[i][j]);
                }
                printf("\n");
        }
        for(i=0;i<4;i++){
                printf("%d ",a[i]);
        }
        printf(" LRC");
        getch();
}
```

IV. RESULTS

Simulation output for CRC

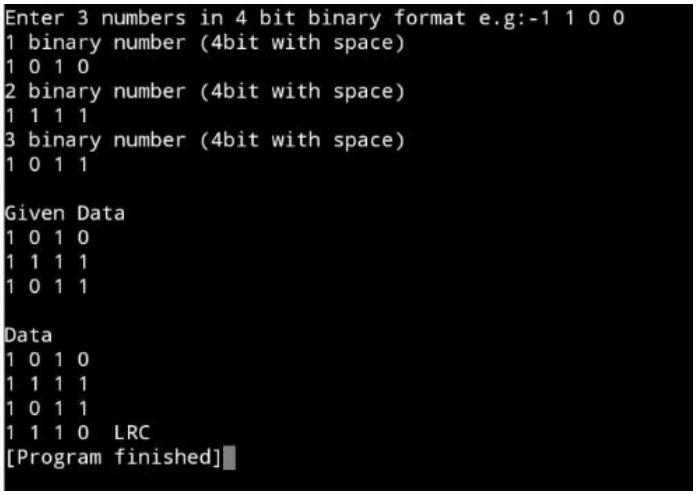


Simulation output for VRC generation for odd parity



Simulation output for LRC generation for even parity

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 5, Issue 1, May 2021
Impact Factor: 4.819



V. CONCLUSION

This paper generates error-detecting code with simulation tools like C programming . we can also generates error-detecting code for checksum method using C programming with help of these code channel decoder detects errors present in received message . so error probability in the received message becomes very less.

REFERENCES

[1]. Varinder Singh, Narinder Sharma "A Review on Various Error Detection and Correction Methods Used in Communication" AIJRSTEM December 2014-February 2015, pp. 252-257

[2]. Fernanda Lima, Luigi Carro, Ricardo Reis "Designing Fault Tolerant Systems into SRAM-based FPGAs" Anaheim,Vol. 3, June. 2003,pp 312-318.

[3]. Deepika, A. Kumar, and Gurusiddayya, "A Study on Error Coding Techniques," International Journal for Research in Applied Science Engineering Technology (IJRASET), vol. 4, no. 4, pp. 825-828, 2016.

[4]. Achmad Fauzi, Nurhayati , Robbi Rahimvon "Bit Error Detection and Correction with Hamming Code Algorithm"2017IJSRSET | Vol 3Issue 1

[5]. Yashveer Singh, Anurag Chandna 'Resolve Error with Detection & Correction Techniques in Computer Networks' International Research Journal of Engineering and Technology (IRJET) Volume: 07 Issue: 01 | Jan 2020