

**Course title: Data Communication Lab**

**Course Code: CSE-308**

**Experiment No: 03**

**Name of the Experiment: Error Detection**

**Date of Performance: 25/02/2021**

**Date of Submission:** **28/02/2021**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Submitted By:** | |  |  | **Submitted To:** |  |
|  | Name :**Mahbub Ali** | | | **Mahmuda Rahman** | | |
|  | Id. | :**192002064** | |  | **Lecturer** | |
|  | Dept. : **CSE** |  |  | **Department of CSE** | | |
|  | **Green University of Bangladesh** | | | **Green University of Bangladesh** | | |
|  |  |  |  |  |  |  |

**Theory**

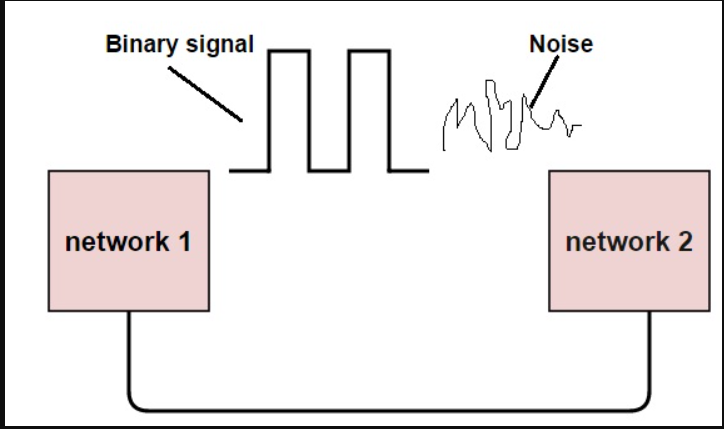
**Introduction:**

In digital systems, the analog signals will change into digital sequence (in the form of bits), This sequence of bits is called as Data stream. The change in position of single bit also leads to catastrophic error in data output t. Almost in all electronic device ,we find error and we use error detection and correction techniques to get the exact or approximate output.

**What is Error:**

Tha data can be corrupted during transmission (from source to receiver). It may be affected by external noise or some other physical imperfections. In this case , the input data is not same as the received output data. This mismatched data is called Error.

The data errors will cause loss of important / secured data. Eben one bit of change in data may affect the whole system’s performance. Generally the data transfer in digital systems will be in the form of Bit transfer. In this case the data error is likely to be changed in positions of 0and 1.



**Types of Errors:**

In a data sequence, if 1 is changed to zero or 0 is changed to 1, It is called Bit error.

There are generally 3 types of errors occur in data transmission from transmitter to receiver .They are

1. Single bit errors
2. Multiple bit errors
3. Burst errors

**Single bit Data Errors:**

The change in one bit in the whole data sequence , is called single bit error.

Occurrence of single bit error is vary rare in serial communication system. This type of error occurs only in parallel communication system , as data is transferred bit wise in single line , there is chance that single line to be rnoisy

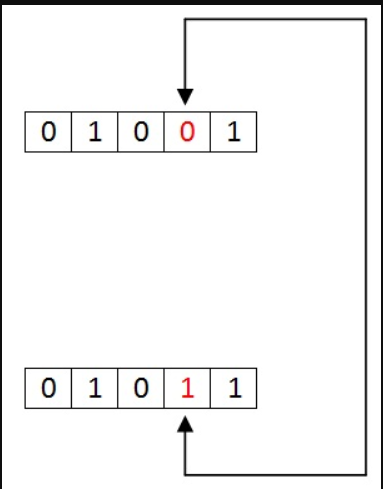
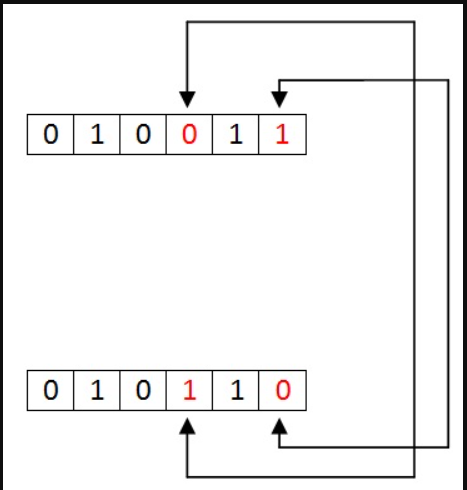


Figure: 1

**Multiple bit data errors:**

If there is change in two or more bits of data sequence of transmitter to receiver, it is called Multiple bit errors. This type of error occurs in both serial type and parallel type data communication networks.



**Burst Errors:**

The change of set of bits in data sequence is called Burst error , the burst error is calculated in from the first bit change to last bit change.



Here we identify the error form fourth bit to 6th bit. The numbers between 4th and 6th bits are also considered as error. These set of bits are called Burst error. This burst bits changes from transmitter to receiver , ehich may cause a major error in data sequence.

**Error Detection codes:**

In digital communication system errors are transferred from one communication system to another , along with the data. If these errors are not detected and corrected , data will be lost. For effective communication, data should be transferred with high accuracy. This can be achieved by first detecting the errors and then correcting therm.

Error detection is the process of detecting the errors that are present in the data transmitted from transmitter to receiver, tin a communication system . We use some redundance code to detect these errors , by adding to the data while it is transmitted from source . These codes are called Error detching codes.

**Types of Error detection:**

1. Parity checking
2. Cyclic Redundancy check (CRC)
3. Longitudinal Redundancy check (LRC)
4. Check sum

**Parity checking:**

Parity bit means nothing but an additional bit adder to the data at the transmitter before transmitting the data. Before adding the parity bit , number of 1’s and 0 is calculated in the data . Based on this calculation of data an extra bit is added to the actual information . The addition of parity bit to the data will result in the change of data string size.

This means if we have an 8 bit data, then after adding bit to the data binary string it will become a 9 bit binary data string.

Parity check is also called as Vertical Redundancy Check (VRC).

There is two types of parity puts in error detection, they are

1. Even parity
2. Odd parity

**Even parity:**

1. If the data has even number of 1’s , the parity bit is 0. Ex: data is10001 >parity bit 0.
2. Odd number of 1’s the parity bit is 1. Ex: data is 11010 > parity bit 1.

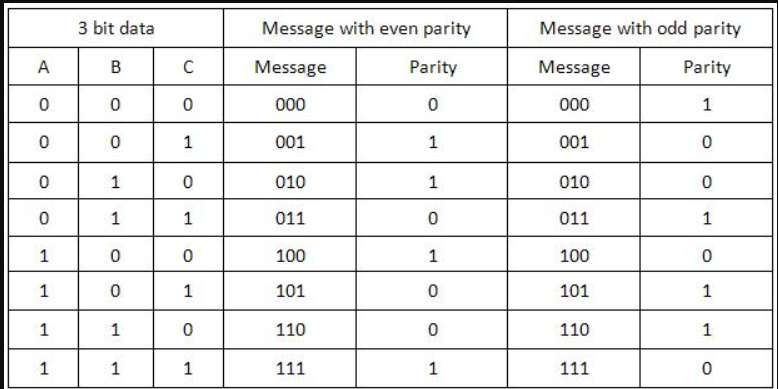
**Odd parity:**

1. If the data has odd number of 1’s the parity bit is 0. Ex: data is 111011 > parity bit is 0.
2. Even number of 1’s the parity bit is 1. Ex: data is 10010 >parity bi is 1.

The circuit which adds a parity bit to the data at transmitter is called Parity generator . The parity bits are transmitter and they are checked at the receiver.

If the parity bits sent at the transmitter and the parity bits received at receiver are not equal then an error is detected. The circuit which checks the parity at receiver is called parity checker.

Messages with even parity and odd parity



**Source Code:**

//1101

#include<stdio.h>

int main()

{

char data[50],even[50],even1[50],deeven[50];

int i,j,count=0,m,n,p, q;

printf("Enter data: ");

scanf("%s",data);

int len=strlen(data);

j=0;

printf("\n Do you use even or odd\n If you use even, then enter 1 \n if you use odd, enter 2: ");

scanf("%d",&m);

if(m==1)

{

for(i=0; i<len; i++)

{

if(data[i]=='1')

{

count++;

}

even[j]=data[i];

j++;

}

//printf("\nCount is : %d",count);

if(count%2==0)

{

even[j]='0';

j++;

}

else

{

even[j]='1';

j++;

}

len=strlen(even);

even[j]='\0';

printf("\nNow data is : %s",even);

strcpy(even1, even);

printf("\nDo you change one data,\nif you want change then enter 1,if you don't change enter 2: ");

scanf("%d",&n);

if(n==1)

{

printf("\n enter positions do you want to change: ");

scanf("%d",&p);

printf("\n Enter 0 or 1: ");

scanf("%d",&q);

even1[p] = q+'0';

printf("\n finally your data is: %s",even1);

}

else

{

printf("\nNow data is : %s",even1);

}

printf("\n your data is");

if(strcmp(even,even1) == 0)

printf(" No error.\n");

else

printf(" Error.\n");

}

if(m==2)

{

for(i=0; i<len; i++)

{

if(data[i]=='1')

{

count++;

}

even[j]=data[i];

j++;

}

//printf("\nCount is : %d",count);

if(count%2==0)

{

even[j]='1';

j++;

}

else

{

even[j]='0';

j++;

}

len=strlen(even);

even[j]='\0';

printf("\nNow data is : %s",even);

strcpy(even1, even);

printf("\n Do you change one data\nif you want change then enter 1 or don't change enter 2: ");

scanf("%d",&n);

if(n==1)

{

printf("\n enter positions do you want to change: ");

scanf("%d",&p);

printf("\n Enter 0 or 1: ");

scanf("%d",&q);

even1[p] = q+'0';

printf("\n finally your data is: %s",even1);

}

else

{

printf("\nNow data is : %s",even1);

}

printf("\n you data is ");

if(strcmp(even,even1) == 0)

printf(" Not error.\n");

else

printf(" Error.\n");

}

}

**Output:**



7