WRITEUP:

Assignment - 2

Title: Implementation of hamming code and CRC.

Problem Statement: Write a program in C/C++ for error detection and correction for 7/8 bits ASCII codes using Hamming codes or CRC. Demonstrate the packets captured traces using wireshark packet analyzer tool for peer to peer mode.

Objective: Understand and implement methods for error detection and correction.

Outcome: Ability to implement error detection and correction methods.

S/W and H/W Packages:
64 bit Fedora 20
Eclipse
Wireshark

Theory:
What is error?
The case in which the input data is not same as the received output data because of the exteral noise or any other physical imperfections, the data is called error.
The data error will cause loss of important secured data.

Types of errors:

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

Three types of errors are:-

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

1. Single bit errors:-

The change in one bit in the whole data sequence, is called single bit error. Occurrence of single bit error is very rare in serial communication systems. This type of error only occurs in parallel communication systems. As data is transferred bit wise in single line, there is a chance that single line is noisy.

2. Multiple bit errors:

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

3. 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.

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 added to the data at the transmitter before transmitting the data before adding the parity bit, number of 1s or 0s is calculated in the data based on the calculation of data an extra bit is added to the actual information/data. The addition of parity bit to the data will result in the change of data string size.

Even parity:

If the data has even number of 1s, the parity bit is 0.
ex. data is 10000 → parity bit is 0

Odd parity:

If the data has odd number of 1s, the parity bit is 1 ex. data is 10100 → parity bit is 1.

2. Cyclic Redundancy Check

A cyclic code is a linear block code with the property that every cyclic shift of a codeword results in another code word there K, indicates the length of the message at transmitter (the number of information bits), n is the total length of the message of information bits. K is the number of check bits. The codes used for cyclic redundancy check thereby error detection are known as CRC codes. CRC codes are shortened cyclic codes. These types of codes are used for error detection and encoding.

3. Error Correcting code

The codes which are used for both error detecting and error correction are called error correction codes.

Single bit error correction

The process of correcting single bit error is called single bit error correction.

Burst error correction

The method of detecting and correcting burst errors in the data sequence is called Burst error correction.

- Algorithm for encoding using CRC

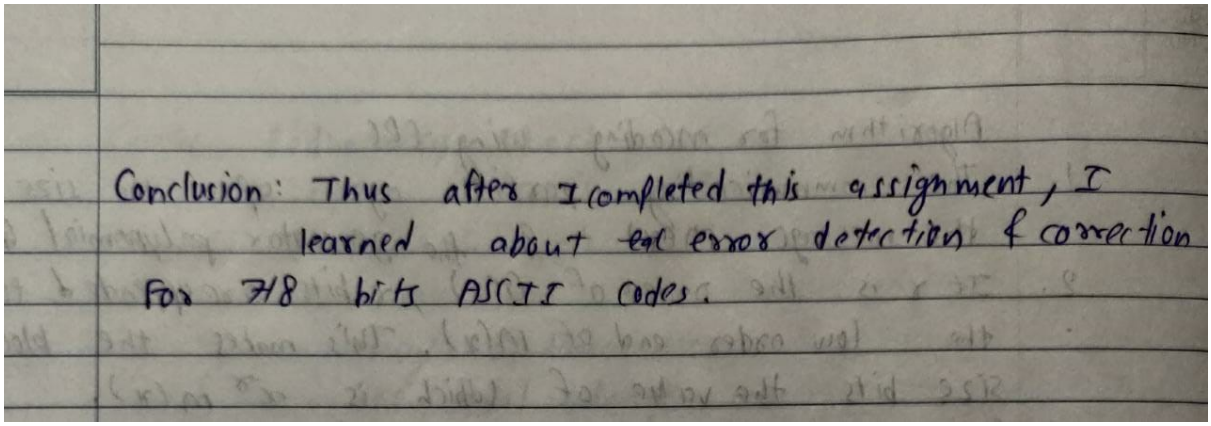
1. The communicating parties agree upon the size of the message $M(x)$ & the generator polynomial $G(x)$.
2. If r is the order of $G(x)$, r bits are appended to the low order end of $M(x)$. This makes the block size bits the value of which is $x^r m(x)$.
3. The block $x^r m(x)$ is divided by $G(x)$ using modulo 2 division
4. The remainder after division is added to $x^r m(x)$ using modulo 2 addition. The result is the frame to be transmitted. $T(x)$ The encoding procedure makes it exactly divisible by $G(x)$.

- Algorithm for decoding using CRC.

1. The receiver divides the incoming dataframe $T(x)$ unit by $G(x)$ using modulo 2 division. Mathematically if $E(x)$ is the error, then modulo 2 division of $[m(x) + E(x)]$ by $G(x)$ is done.

2. If there is no remainder then it implies that $E(x)$ the data frame is accepted.

3. A remainder indicates a non-zero value of $E(x)$ or in other words, presence of an error. So the dataframe is rejected. The receiver may then send an erroneous acknowledgement back to the sender for retransmission.



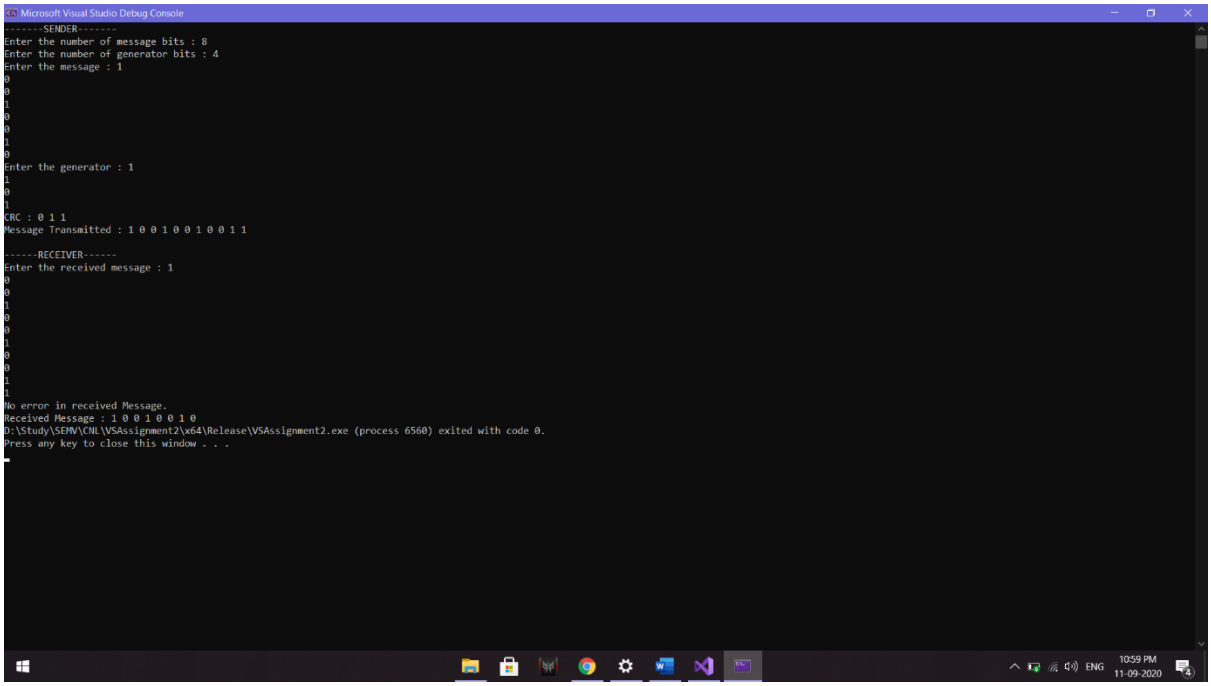
SAMPLE CODE:

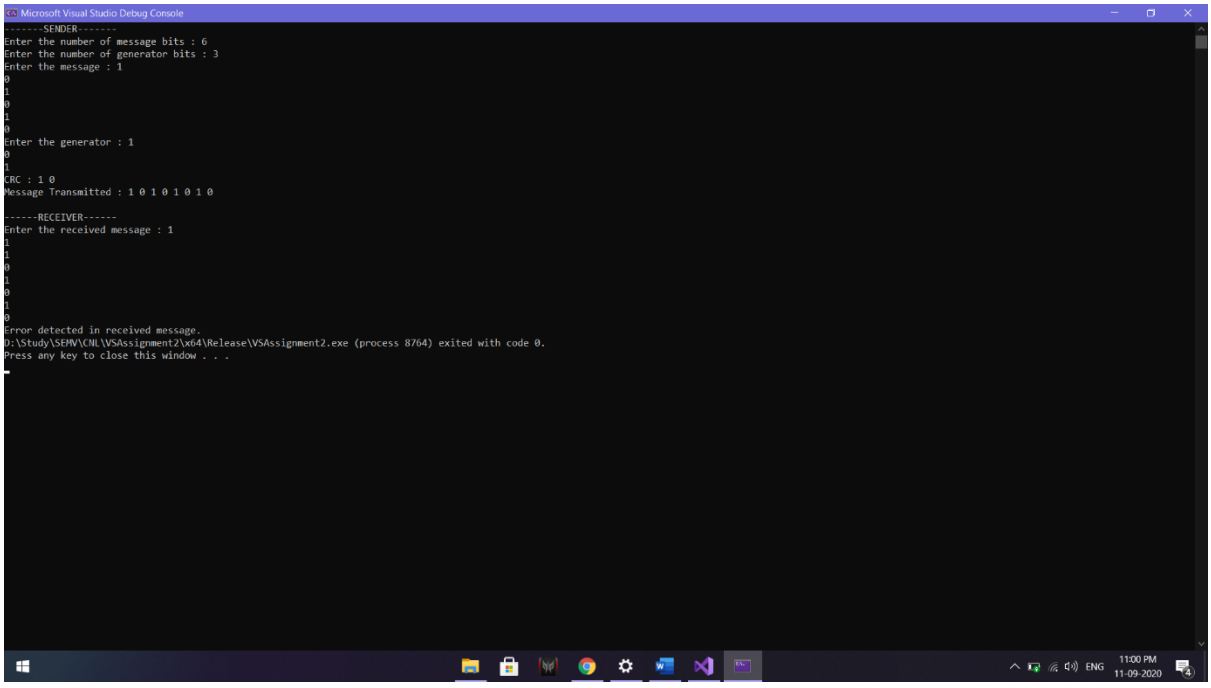
```
#include<iostream>
using namespace std;
void division(int temp[], int gen[], int n, int r)
{
	for (int i = 0; i < n; i++)
	{
		if (gen[0] == temp[i])
		{
			for (int j = 0, k = i; j < r + 1; j++, k++)
			{
				if (!(temp[k] ^ gen[j]))
					temp[k] = 0;
				else
					temp[k] = 1;
			}

		}
	}
}
int main()
{
	int n, r, message[50], gen[50], temp[50];
```

```
cout << "-------SENDER-------" << endl;
cout << "Enter the number of message bits : ";
cin >> n;
cout << "Enter the number of generator bits : ";
cin >> r;
cout << "Enter the message : ";
for (int i = 0; i < n; i++)
        cin >> message[i];
cout << "Enter the generator : ";
for (int i = 0; i < r; i++)
        cin >> gen[i];
r--;
for (int i = 0; i < r; i++)
        message[n + i] = 0;
for (int i = 0; i < n + r; i++)
        temp[i] = message[i];
division(temp, gen, n, r);
cout << "CRC : ";
for (int i = 0; i < r; i++)
{
        cout << temp[n + i] << " ";
        message[n + i] = temp[n + i];
}
cout << endl << "Message Transmitted : ";
for (int i = 0; i < n + r; i++)
        cout << message[i] << " ";
cout << endl << endl << "------RECEIVER------ " << endl;
cout << "Enter the received message : ";
for (int i = 0; i < n + r; i++)
        cin >> message[i];
for (int i = 0; i < n + r; i++)
```

```
		temp[i] = message[i];
	division(temp, gen, n, r);
	for (int i = 0; i < r; i++)
	{
		if (temp[n + i])
		{
			cout << "Error detected in received message.";
			return 0;
		}
	}
	cout << "No error in received Message.\nReceived Message : ";
	for (int i = 0; i < n; i++)
		cout << message[i] << " ";
	return 0;
}
```

OUTPUT:


Microsoft Visual Studio Debug Console
-------SENDER-------
Enter the number of message bits : 8
Enter the number of generator bits : 4
Enter the message : 1
0
0
1
0
0
1
0
Enter the generator : 1
1
0
1
CRC : 0 1 1
Message Transmitted : 1 0 0 1 0 0 1 0 0 1 1
------RECEIVER------
Enter the received message : 1
0
0
1
0
0
1
0
0
1
1
No error in received Message.
Received Message : 1 0 0 1 0 0 1 0
D:\Study\SEMV\CNL\VSAssignment2\x64\Release\VSAssignment2.exe (process 6560) exited with code 0.
Press any key to close this window . . .
ENG
10:59 PM
11-09-2020


Microsoft Visual Studio Debug Console
-------SENDER-------
Enter the number of message bits : 6
Enter the number of generator bits : 3
Enter the message : 1
0
1
0
1
0
Enter the generator : 1
0
1
CRC : 1 0
Message Transmitted : 1 0 1 0 1 0 1 0
------RECEIVER------
Enter the received message : 1
1
1
0
1
0
1
0
Error detected in received message.
D:\Study\SEMV\CNL\VSAssignment2\x64\Release\VSAssignment2.exe (process 8764) exited with code 0.
Press any key to close this window . . .
ENG
11:00 PM
11-09-2020