Implementation of Cycle Redundancy Check (CRC) for error detection

Done by: Thota Surya Vara Prasad Rao, CSE-G, AP22110011059,
suryavaraprasasdrao_thota@srmap.edu.in
Submitted on: 22-08-2024

Objective:

Cycle Redundancy Check is a method to find error in digital data during the transmission of data.

How it works:

There is a generating divisor(x^8+x^2+x+1) polynomial between sender and receiver.
The message is encrypted and sent. When the receiver receives message, he divides
(binary division) it generates polynomial. If remainder is zero then the message is correct or fully flipped. If not zero then single or more bits are changed.
Limitation:
1. It is designed to detect errors but it cannot correct them.
2. It cannot tell which bit is flipped.

Problem Statement:

Write a program to implement Error Detection Technique using CRC Algorithm.

Code:

```
#include <bits/stdc++.h>
using namespace std;
// Function to perform binary division using CRC
void BinaryDivision(int dividend[], int divisor[], int remainder[], int n) {
```

```
 // Copy initial bits of the dividend to the remainder
 for (int i = 0; i < 9; i++) {
 remainder[i] = dividend[i];
 }
 // Perform division bit by bit
 for (int i = 0; i < n; i++) {
 // If the first bit of the remainder is 1, perform XOR with the divisor
 if (remainder[0] == 1) {
 for (int j = 0; j < 9; j++) {
 remainder[j] = remainder[j] ^ divisor[j];
 }
 }
 // Shift the remainder left and bring down the next bit of the dividend
 if (i + 9 < n + 8) { // Ensure we don't access out of bounds
 for (int j = 0; j < 8; j++) {
 remainder[j] = remainder[j + 1];
 }
 remainder[8] = dividend[i + 9]; // Bring down the next bit
 }
 }
}
int main() {
 int n;
 cout << "Enter the size of the data bits: ";
 cin >> n;
 int re[9] = {0}; // Remainder array for CRC calculation
 int a[n + 8] = {0}; // Dividend array with room for the appended remainder
 cout << "Enter the data bits: ";
 for (int i = 0; i < n; i++) {
```

```
cin >> a[i]; // Input data bits
}
// Divisor for binary division (hardcoded for this example: CRC-8 polynomial)
int divisor[9] = {1, 0, 0, 0, 0, 0, 1, 1, 1}; // Polynomial x^8 + x^7 + x^4 + x^3 + x^2
+ 1
// Perform binary division
BinaryDivision(a, divisor, re, n);
// Append remainder to the end of the original data
int j = 1;
for (int i = n; i < n + 8; i++) {
a[i] = re[j++]; // Append each bit of the remainder
}
// Output the encoded data
cout << "Encoded data bit is: ";
for (int i = 0; i < n + 8; i++) {
cout << a[i] << " "; // Print each bit of the encoded data
}
cout << endl;
// Input received data bits for error checking
int r[n + 8];
cout << "Enter data bit received at receiver's end: ";
for (int i = 0; i < n + 8; i++) {
cin >> r[i]; // Input received data bits
}
// Array to hold the remainder for the received data
int remind[9] = {0};
BinaryDivision(r, divisor, remind, n);
// Check the remainder to detect errors
int k = 0;
```

```
 for (int i = 0; i < 8; i++) {
 if (remind[i] != 0) { // If any bit of the remainder is non-zero
 cout << "ERROR DETECTED!!!!"; // Error detected
 break;
 } else {
 k++; // Increment the counter for non-zero bits
 }
 }
 // If no error detected
 if (k == 8) {
 cout << "NO ERROR DETECTED....";
 }
 return 0;
}
```

Output:

```
Enter the size of the data bits: 9
Enter the data bits: 1 1 1 0 0 1 1 1 0
Encoded data bit is: 1 1 1 0 0 1 1 1 0 0 1 1 1 0 0 0 1
enter data bit received at receivers end:1 1 0 0 0 1 1 1 0 0 1 1 1 0 0 0 1
ERROR DETECTED!!!!
```

In above output screen, one bit is flipped so it gave the output as ERROR DETECTED!!!!

```
Enter the size of the data bits: 9
Enter the data bits: 1 1 1 0 0 1 1 1 0
Encoded data bit is: 1 1 1 0 0 1 1 1 0 0 1 1 1 0 0 0 1
enter data bit received at receivers end:1 1 1 0 0 1 1 1 0 0 1 1 1 0 0 0 1
NO ERROR DETECTED....
```

In above output screen, no bit is flipped so it gave the output as NO ERROR DETECTED….

Problems faced:

1. I faced difficulty to understand the logic and to write the code for binary division.

Conclusion:

Knowledge: learnt how to do binary division and CRC algorithm and how it works.

Skill: Revised concepts of C++ language.