

Assignment-3

Task:1

Problem-01:

-

A bit stream 1101011011 is transmitted using the standard CRC method. The generator polynomial is x^4+x+1 . What is the actual bit string transmitted?



```
1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  // Function to perform modulo-2 division
6  string xor_division(string data, string generator) {
7      int n = generator.size();
8      string temp = data.substr(0, n);
9
10     for (int i = 0; i < data.size() - n + 1; i++) {
11         if (temp[0] == '1') {
12             for (int j = 0; j < n; j++)
13                 temp[j] = (temp[j] == generator[j]) ? '0' : '1';
14         }
15         if (i + n < data.size())
16             temp = temp.substr(1) + data[i + n];
17         else
18             temp = temp.substr(1);
19     }
20     return temp;
21 }
22
23 int main() {
24     string data = "1101011011";
25     string generator = "10011";
26     int r = generator.size() - 1;
27
28     // Append r zeros
29     string data_appended = data + string(r, '0');
30
31     // Get remainder
32     string remainder = xor_division(data_appended, generator);
33
34     // Transmitted frame
35     string transmitted = data + remainder;
36
37     cout << "Remainder (CRC): " << remainder << endl;
38     cout << "Transmitted Frame: " << transmitted << endl;
39
40     return 0;
41 }
42
```

```
PS C:\Users\welcome\Downloads\OSCN\Assignment-3> cd "c:\Users\welcome\Downloads\OSCN\Assignment-3\" ; if ($?) { g++ Task01.cpp -o Task01 } ; if ($?) { .\Task01 }  
Remainder (CRC): 1110  
Transmitted Frame: 11010110111110  
PS C:\Users\welcome\Downloads\OSCN\Assignment-3> █
```

Task:2

Problem-02:

A bit stream 10011101 is transmitted using the standard CRC method. The generator polynomial is x^3+1 .

1. What is the actual bit string transmitted?
2. Suppose the third bit from the left is inverted during transmission. How will receiver detect this error?

```

1  #include <iostream>
2  #include <string>
3  #include <vector>
4  using namespace std;
5
6  // Function to perform CRC division and return remainder
7  string calculateCRC(string message, string generator) {
8      string dividend = message;
9      int genLength = generator.length();
10
11     // Append zeros to the message equal to generator length-1
12     for(int i = 0; i < genLength - 1; i++)
13         dividend += "0";
14
15     // Perform division
16     for(int i = 0; i <= dividend.length() - genLength; i++) {
17         if(dividend[i] == '1') {
18             for(int j = 0; j < genLength; j++) {
19                 dividend[i + j] = dividend[i + j] == generator[j] ? '0' : '1';
20             }
21         }
22     }
23
24     // Return last genLength-1 bits as remainder
25     return dividend.substr(dividend.length() - (genLength - 1));
26 }
27
28 // Function to check if received message has errors
29 bool hasErrors(string received, string generator) {
30     string remainder = calculateCRC(received, generator);
31     for(char bit : remainder) {
32         if(bit == '1') return true;
33     }
34     return false;
35 }
36
37 int main() {
38     string message = "10011101";
39     string generator = "1001"; //  $x^3 + 1$ 
40
41     cout << "Original Message: " << message << endl;
42     cout << "Generator Polynomial ( $x^3 + 1$ ): " << generator << endl;
43
44     // Calculate CRC
45     string crc = calculateCRC(message, generator);
46     string transmittedMessage = message + crc;
47
48     cout << "\n1. Actual bit string transmitted:" << endl;
49     cout << "CRC bits: " << crc << endl;
50     cout << "Transmitted message: " << transmittedMessage << endl;
51
52     // Simulate error by inverting third bit
53     cout << "\n2. Error Detection:" << endl;
54     cout << "Original transmitted message: " << transmittedMessage << endl;
55
56     // Create corrupted message by inverting third bit
57     string corruptedMessage = transmittedMessage;
58     corruptedMessage[2] = corruptedMessage[2] == '0' ? '1' : '0';
59
60     cout << "Message with error (third bit inverted): " << corruptedMessage << endl;
61
62     // Check for errors
63     if(hasErrors(corruptedMessage, generator)) {
64         cout << "Error detected in the received message!" << endl;
65     } else {
66         cout << "No errors detected in the received message." << endl;
67     }
68
69     return 0;
70 }
71

```

```
PS C:\Users\welcome\Downloads\OSCN\Assignment-3> cd "c:\Users\welcome\Downloads\OSCN\Assignment-3\" ; if ($?) { g++ Task02.cpp -o Task02 } ; if ($?) { .\Task02 }
Original Message: 10011101
Generator Polynomial ( $x_7 + 1$ ): 1001

1. Actual bit string transmitted:
CRC bits: 100
Transmitted message: 10011101100

2. Error Detection:
Original transmitted message: 10011101100
Message with error (third bit inverted): 10111101100
Error detected in the received message!
PS C:\Users\welcome\Downloads\OSCN\Assignment-3>
```