

Assignment-3

Name : Raj Kumar

Enrollment No : 2503B09910

Class : MCA

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 // 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?
2
3 #include <iostream>
4 #include <string>
5 using namespace std;
6
7 // Function to perform XOR between two binary strings
8 string xor1(string a, string b) {
9     string result = "";
10    for (int i = 1; i < b.length(); i++) {
11        result += (a[i] == b[i]) ? '0' : '1';
12    }
13    return result;
14 }
15
16 // Function to perform Mod-2 division
17 string mod2div(string dividend, string divisor) {
18     int pick = divisor.length();
19     string tmp = dividend.substr(0, pick);
20
21     int n = dividend.length();
22
23     while (pick < n) {
24         if (tmp[0] == '1')
25             tmp = xor1(divisor, tmp) + dividend[pick];
26         else
27             tmp = xor1(string(pick, '0'), tmp) + dividend[pick];
28         pick++;
29     }
30
31     // For last n bits
32     if (tmp[0] == '1')
33         tmp = xor1(divisor, tmp);
34     else
35         tmp = xor1(string(pick, '0'), tmp);
36
37     return tmp;
38 }
39
40 int main() {
41     string data = "1101011011";
42     string divisor = "10011"; //  $x^4 + x + 1$ 
43
44     int m = divisor.length();
45     string appended_data = data + string(m - 1, '0');
46
47     string remainder = mod2div(appended_data, divisor);
48     string codeword = data + remainder;
49
50     cout << "Original Data:      " << data << endl;
51     cout << "Generator:          " << divisor << endl;
52     cout << "Appended Data:      " << appended_data << endl;
53     cout << "CRC Remainder:      " << remainder << endl;
54     cout << "Transmitted Code:    " << codeword << endl;
55
56     return 0;
57 }
58
```

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?

Assignment-3

```
1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  string xor1(string a, string b) {
6      string result = "";
7      for (int i = 1; i < b.length(); i++) {
8          result += (a[i] == b[i]) ? '0' : '1';
9      }
10     return result;
11 }
12
13 string mod2div(string dividend, string divisor) {
14     int pick = divisor.length();
15     string tmp = dividend.substr(0, pick);
16     int n = dividend.length();
17
18     while (pick < n) {
19         if (tmp[0] == '1')
20             tmp = xor1(divisor, tmp) + dividend[pick];
21         else
22             tmp = xor1(string(pick, '0'), tmp) + dividend[pick];
23         pick++;
24     }
25
26     if (tmp[0] == '1')
27         tmp = xor1(divisor, tmp);
28     else
29         tmp = xor1(string(pick, '0'), tmp);
30
31     return tmp;
32 }
33
34 int main() {
35     string data = "10011101";
36     string divisor = "1001"; // x^3 + 1
37
38     // Step 1: Append zeros
39     string appended_data = data + string(divisor.length() - 1, '0');
40
41     // Step 2: Get remainder
42     string remainder = mod2div(appended_data, divisor);
43
44     // Step 3: Form transmitted code
45     string transmitted = data + remainder;
46
47     cout << "Original Data:      " << data << endl;
48     cout << "Generator:      " << divisor << endl;
49     cout << "Appended Data:   " << appended_data << endl;
50     cout << "CRC Remainder:   " << remainder << endl;
51     cout << "Transmitted Codeword: " << transmitted << endl;
52
53     // Step 4: Simulate error (invert 3rd bit)
54     string received = transmitted;
55     received[2] = (received[2] == '0') ? '1' : '0'; // flip 3rd bit
56     cout << "\nReceived (with error): " << received << endl;
57
58     // Step 5: Check remainder at receiver
59     string remainder_receiver = mod2div(received, divisor);
60     cout << "Remainder at Receiver: " << remainder_receiver << endl;
61
62     if (remainder_receiver.find('1') != string::npos)
63         cout << "Error detected during transmission!" << endl;
64     else
65         cout << "No error detected." << endl;
66
67     return 0;
68 }
69
```