

OSCN LAB – 3

Name: VIQUAR FATHIMA

Date: 21/10/2025

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?

Ans: Given:

- Data bits: 1101011011
- Generator polynomial: $x^4 + x + 1 \rightarrow$ generator bits: 10011

Code:

```
#include <iostream>

#include <string>

using namespace std;

// Function to perform XOR operation between two binary strings
string xorOperation(string a, string b) {
    string result = "";
    for (size_t i = 1; i < b.size(); i++)
        result += (a[i] == b[i]) ? '0' : '1';
    return result;
}

// Function to perform modulo-2 division
string mod2Division(string dividend, string divisor) {
    int pick = divisor.size();
    string tmp = dividend.substr(0, pick);
    int n = dividend.size();
```

```

while (pick < n) {
    if (tmp[0] == '1')
        tmp = xorOperation(divisor, tmp) + dividend[pick];
    else
        tmp = xorOperation(string(pick, '0'), tmp) + dividend[pick];
    pick++;
}

// For the last n bits
if (tmp[0] == '1')
    tmp = xorOperation(divisor, tmp);
else
    tmp = xorOperation(string(pick, '0'), tmp);

return tmp;
}

int main() {
    string data = "1101011011";
    string generator = "10011";

    int m = generator.size();
    string appendedData = data + string(m - 1, '0'); // Append (m-1) zeros

    cout << "Data bits: " << data << endl;
    cout << "Generator: " << generator << endl;
    cout << "Appended data: " << appendedData << endl;

    string remainder = mod2Division(appendedData, generator);
    cout << "Remainder (CRC bits): " << remainder << endl;
}

```

```

// XOR remainder with appended data to get transmitted codeword

string transmitted = data + remainder;

cout << "Transmitted frame (data + CRC): " << transmitted << endl;

return 0;
}

```

Output:

```

Lab-3.cpp > ...
1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  // Function to perform XOR operation between two binary strings
6  string xorOperation(string a, string b) {
7      string result = "";
8      for (int i = 0; i < a.length(); i++) {
9          result += a[i] ^ b[i];
10     }
11     return result;
12 }
13
14
PS C:\Users\91832\OneDrive\Documents\Desktop\OSCN ASSIGNMENTS> cd "c:\Users\91832\OneDrive\Documents\Desktop\OSCN ASSIGNMENTS"
PS C:\Users\91832\OneDrive\Documents\Desktop\OSCN ASSIGNMENTS> cd "c:\Users\91832\OneDrive\Documents\Desktop\OSCN ASSIGNMENTS"
g++ Lab-3.cpp -o Lab-3 } ; if ($?) { .\Lab-3 }
Data bits: 1101011011
Generator: 10011
Appended data: 11010110110000
Remainder (CRC bits): 1110
Transmitted frame (data + CRC): 11010110111110
PS C:\Users\91832\OneDrive\Documents\Desktop\OSCN ASSIGNMENTS>

```

Explanation

1. **Data bits** = 1101011011
2. **Generator polynomial** = $x^4 + x + 1 \rightarrow$ **Generator bits** = 10011
3. Append 4 zeros (degree of generator = 4) to data \rightarrow 11010110110000
4. Perform **mod-2 division** (XOR based) \rightarrow remainder = **1110**
5. **Actual transmitted frame** = 1101011011 + 1110 = **11010110111110**

Therefore, the **actual bit string transmitted** is: 11010110111110

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?

Code:

```
#include <iostream>

#include <string>

using namespace std;

// XOR operation between two binary strings (skip the first bit)
string xorOp(string a, string b) {
    string result = "";
    for (int i = 1; i < b.size(); i++) {
        if (a[i] == b[i])
            result += '0';
        else
            result += '1';
    }
    return result;
}

// Perform Modulo-2 Division
string mod2Division(string dividend, string divisor) {
    int pick = divisor.size();
    string tmp = dividend.substr(0, pick);
    int n = dividend.size();
```

```

while (pick < n) {
    if (tmp[0] == '1')
        tmp = xorOp(divisor, tmp) + dividend[pick];
    else
        tmp = xorOp(string(pick, '0'), tmp) + dividend[pick];
    pick++;
}

if (tmp[0] == '1')
    tmp = xorOp(divisor, tmp);
else
    tmp = xorOp(string(pick, '0'), tmp);

return tmp;
}

// Flip (invert) a bit at a given position
string flipBit(string s, int pos) {
    if (pos < 0 || pos >= s.size())
        return s;
    if (s[pos] == '0')
        s[pos] = '1';
    else
        s[pos] = '0';
    return s;
}

int main() {
    string data = "10011101"; // Given data bits
    string generator = "1001"; // Given generator polynomial ( $x^3 + 1$ )
    int genDegree = generator.size() - 1;

```

```

cout << "Data bits:    " << data << endl;
cout << "Generator bits: " << generator << endl;

// Step 1: Append (m-1) zeros
string appended = data + string(genDegree, '0');
cout << "Appended data: " << appended << endl;

// Step 2: Perform division to find remainder (CRC)
string remainder = mod2Division(appended, generator);
cout << "Calculated CRC (remainder): " << remainder << endl;

// Step 3: Form transmitted frame
string transmitted = data + remainder;
cout << "Transmitted frame (data + CRC): " << transmitted << endl;

// Step 4: Simulate bit error (flip 3rd bit from left)
int flipIndex = 2; // 0-based index
string received = flipBit(transmitted, flipIndex);
cout << "\nSimulate error: flip 3rd bit from left" << endl;
cout << "Received frame: " << received << endl;

// Step 5: Receiver checks CRC again
string recvRemainder = mod2Division(received, generator);
cout << "Receiver remainder: " << recvRemainder << endl;

// Step 6: Check if remainder is all zeros
bool isZero = true;
for (int i = 0; i < recvRemainder.size(); i++) {
    if (recvRemainder[i] != '0') {
        isZero = false;
    }
}

```

```

        break;
    }
}

if (isZero)
    cout << "Receiver: No error detected (remainder is zero)." << endl;
else
    cout << "Receiver: Error detected (non-zero remainder)." << endl;

return 0;
}

```

Output:

The screenshot shows a C++ IDE with a code editor and a terminal window. The code in the editor defines a function `xorOp` for XORing two binary strings. The terminal shows the execution of a program that calculates the CRC of a data string '1101011011' using a generator polynomial '10011'. The output shows the transmitted frame '1101011011110' and the receiver's calculation of the remainder '100', indicating an error was detected.

```

1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  // XOR operation between two binary strings (skip the first bi
6  string xorOp(string a, string b) {
7      string result = "";
8      for (int i = 1; i < b.size(); i++) {
9          if (a[i] != b[i])
10             result += "1";
11         else
12             result += "0";
13     }
14     return result;
15 }
16
17 int main() {
18     string data = "1101011011";
19     string gen = "10011";
20     string frame = data + xorOp(data, gen);
21     cout << "Data bits: " << data << endl;
22     cout << "Generator: " << gen << endl;
23     cout << "Appended data: " << frame << endl;
24     string remainder = xorOp(frame, gen);
25     cout << "Remainder (CRC bits): " << remainder << endl;
26     cout << "Transmitted frame (data + CRC): " << frame << endl;
27     return 0;
28 }

```

```

PS C:\Users\91832\OneDrive\Documents\Desktop\OSCN ASSIGNMENTS> cd "c:\Users\91832\OneDrive\Documents\Desktop\OSCN ASSIGNMENTS"
PS C:\Users\91832\OneDrive\Documents\Desktop\OSCN ASSIGNMENTS> cd "c:\Users\91832\OneDrive\Documents\Desktop\OSCN ASSIGNMENTS\"
g++ Lab-3.cpp -o Lab-3 } ; if ($?) { .\Lab-3 }
Data bits: 1101011011
Generator: 10011
Appended data: 11010110110000
Remainder (CRC bits): 1110
Transmitted frame (data + CRC): 11010110111110
PS C:\Users\91832\OneDrive\Documents\Desktop\OSCN ASSIGNMENTS> cd "c:\Users\91832\OneDrive\Documents\Desktop\OSCN ASSIGNMENTS"
PS C:\Users\91832\OneDrive\Documents\Desktop\OSCN ASSIGNMENTS> cd "c:\Users\91832\OneDrive\Documents\Desktop\OSCN ASSIGNMENTS\"
g++ lab-3.2.cpp -o lab-3.2 } ; if ($?) { .\lab-3.2 }

```

1. Actual transmitted bit string: 10011101100

2. After 3rd bit inversion, receiver remainder = 100 → Error detected