

## **EXPERIMENT: 34**

### **IMPLEMENTING THE SIMULATION OF ERROR CORRECTION CODE – CRC IN JAVA/C**

**Aim:** To implement the simulation of error correction code – CRC in java.

**Steps:**

1. Define the CRC polynomial and bit length:
  - Choose a CRC polynomial. Common choices are CRC-16 (16 bits) or CRC-32 (32 bits).
  - Define the polynomial value as a constant in your code.
2. Implement the CRC calculation function:
  - Write a function that takes the input data and calculates the CRC value based on the chosen polynomial.
  - The function should iterate through each bit of the input data and perform XOR operations with the polynomial.
  - At the end of the calculation, the CRC value should be returned.
3. Test the CRC calculation function:
  - Create test data with known CRC values.
  - Use the CRC calculation function to calculate the CRC for the test data and compare it with the expected CRC value.
  - Verify that the CRC calculation function produces the correct CRC values for the test data.

In this code, the `calculate CRC` function takes an input data array and its length. It performs the CRC calculation by iterating through each bit of the data and performing XOR and shift operations. The function returns the calculated CRC value.

In the `main` function, a test data array is defined along with the expected CRC value. The `calculate CRC` function is then called to calculate the CRC for the test data. The calculated and expected CRC values are printed, and a comparison is made to check if they match.

## Output

```
Enter data bits: 1101011011
Enter generator polynomial: 10011
Remainder (CRC bits): 1110
Transmitted Frame: 11010110111110

Enter received frame: |
```

**Result:** The simulation of error correction code – CRC in C is implementation successfully.