**Lab Program 3 :**

**Write a program for error detecting code using CRC-CCITT (16- bits).**

import java.util.Scanner;

public class CRC

{

public static int n;

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

CRC crc=new CRC();

String copy,rec,code,zero="0000000000000000";

System.out.println("enter the dataword to be sent");

code=sc.nextLine();

System.out.println("The dataword is "+code);

n=code.length();

System.out.println("Length of the dataword="+n);

copy=code;

code+=zero;

System.out.println("The modified codeword is"+code);

code=crc.divide(code);

copy=copy.substring(0,n)+code.substring(n);

System.out.print("CRC=");

System.out.println(code.substring(n));

System.out.println("transmitted frame is="+copy);

System.out.println("enter received data:");

rec=sc.nextLine();

if(zero.equals(crc.divide(rec).substring(n)))

System.out.println("correct bits received");

else

System.out.println("received frame contains one or more error");

sc.close();

}

public String divide(String s)

{

String div="10001000000100001";

int i,j;

char x;

for(i=0;i<n;i++)

{

x=s.charAt(i);

for(j=0;j<17;j++)

{

if(x=='1')

{

if(s.charAt(i+j)!=div.charAt(j))

s=s.substring(0,i+j)+"1"+s.substring(i+j+1);

else

s=s.substring(0,i+j)+"0"+s.substring(i+j+1);

}

}

}

return s;

}

}

Output:

# java CRC

enter the dataword to be sent

1100

dataword=1100

CRC=1100000110001100

transmitted frame is=11001100000110001100

enter received data:

11001100000110001101

***received frame contains one or more error***

# java CRC

enter the dataword to be sent

1100

dataword=1100

CRC=1100000110001100

transmitted frame is=11001100000110001100

enter received data:

11001100000110001100

***correct bits received***

**Explanation**

* Class and Variables:
  + The class is named **CRC.**
  + There is a static variable **n representing the length of the dataword.**
  + The main method is the entry point of the program.
* ***Input:***
  + The program takes user input for the **dataword to be sent.**
* ***CRC Calculation:***
  + The divide method takes a **string s (representing the dataword concatenated with zeros for CRC)** and performs the CRC division.
  + The polynomial used for CRC is represented by the **string div.**
  + The method iterates through each bit of the dataword, **performing XOR operations** based on the CRC polynomial.
* ***Transmission:***
  + The **original dataword** is displayed.
  + **Zeros are appended to the dataword** to accommodate the CRC bits.
  + The CRC bits are **calculated** using the divide method.
  + The transmitted frame is displayed, **consisting of the original dataword followed by the CRC bits.**
* **Error Detection:**
  + The program then takes user input for the **received data.**
  + The received data is **divided** using the divide method.
  + If the remainder obtained after division is **all zeros**, it is assumed that **no errors occurred,** and the message indicates correct bits received. Otherwise, an error is assumed.
* Scanner Closure:
  + The Scanner is closed to release resources.
* Note:
  + The CRC polynomial used in this example is "10001000000100001," which corresponds to the CRC-CCITT polynomial.
* **The divide method performs the CRC division.**
* The method takes a string s as input, **representing the dataword concatenated with zeros for CRC.**
* **It initializes a string** div representing the CRC polynomial "10001000000100001."
* The **outer loop** (for (i = 0; i < n; i++)) iterates through **each bit of the dataword.**
* Inside the outer loop, the variable x is assigned the i-th bit of the dataword.
* The inner loop (for (j = 0; j < 17; j++)) **iterates through each bit of the CRC polynomial.**
* If the **current bit of the dataword (x) is '1', it enters the conditional block.**
* Inside the conditional block, **it checks if the corresponding bits of the dataword and the CRC polynomial are equal.**
* **If they are not equal, it performs an XOR operation** ('1' XOR '1' = '0', '1' XOR '0' = '1') and updates the string s.
* **If they are equal, it performs an XOR operation and updates the string s** accordingly.
* The method returns the **modified string s after completing the CRC division.**
* This process essentially simulates the CRC division operation, where XOR operations are performed based on the CRC polynomial, and the result is updated in the input string. The final remainder is contained in the modified string s.

**Java String charAt() Method**

The charAt() method **returns the character at the specified index in a string.**

The index of the first character is 0, the second character is 1, and so on.

Syntax: public char charAt(int index)

Parameter Values ParameterDescription

index An int value representing the index of the character to return