

## WEEK 13

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

CODE:

```
#include<stdio.h>
int arr[17];

void xor(int x[], int y[])
{
    int k=0;
    for(int i=1;i<16;i++)
    {
        if(x[i]==y[i])
            arr[k++]=0;
        else
            arr[i]=1;
    }
}

void main()
{
    int dd[17],div[33],ze[17],i,k;

    printf("Enter the dataword \n");
    for(i=0;i<17;i++)
        scanf("%d",&div[i]);

    for(i=i;i<33;i++)
        div[i]=0;

    for(i=0;i<17;i++)
        ze[i]=0;
    printf("Enter dividend \n");
```

```

for(i=0;i<17;i++)
    scanf("%d",&dd[i]);

i=0;
k=0;
    for(i=i;i<17;i++)
        arr[k++]=div[i];
while(i<33)
{
    if(arr[0]==0)
        xor(arr,ze);
    else
        xor(arr,dd);

    arr[16]=div[i++];

}
k=0;
for(i=17;i<33;i++)
    div[i]=arr[k++];
printf("Codeword: ");
    for(i=0;i<33;i++)
        printf("%d",div[i]);

for(i=0;i<17;i++)
    arr[i]=0;

printf("\nAt receiver end \n");

k=0;
    for(i=i;i<17;i++)
        arr[k++]=div[i];
while(i<33)
{
    if(arr[0]==0)

```

```

        xor(arr,ze);
    else
        xor(arr,dd);

    arr[16]=div[i++];

}
k=0;
for(i=17;i<33;i++)
    div[i]=arr[k++];

printf("Codeword: ");
for(i=0;i<33;i++)
    printf("%d",div[i]);
}

```

OUTPUT:

```

Enter the dataword
1 0 1 1 0 0 1 1 1 1 0 0 1 0 1 1 1
Enter dividend
1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 1
Codeword: 101100111100101110000000000011011
At receiver end
Codeword: 101100111100101110000000000000000
Process returned 1 (0x1)   execution time : 49.507 s
Press any key to continue.

```

OBSERVATION:

## Lab 13

Q1. Write a program for error detection using CRC (16 bits)

```
#include <stdio.h>
```

```
#include <string.h>
```

```
void binary XOR(char* result, const char* a, const char* b)
```

```
{
```

```
    for(int i=0; i<16; i++)
```

```
    {
```

```
        result[i] = (a[i] - b[i]) ? '0' : '1';
```

```
    }
```

```
    result[16] = '\0';
```

```
}
```

```
void calculateCRC(const char* data, int length, char* crc)
```

```
{
```

```
    char crc[17];
```

```
    for(int i=0; i<16; i++)
```

```
    {
```

```
        crc[i] = '0';
```

```
    }
```

```
    crc[16] = '\0';
```

```
    for(int i=0; i<length; i++)
```

```
    {
```

```
        for(int j=0; j<8; j++)
```

```
        {
```

```
            char msb = crc[0];
```

```

    if (int k = 0, F < 16, k++)
    {
        crc[k] = crc[k+1];
    }
    crc[15] = '0';

    if (msgb == '1')
    {
        char temp[17];
        binaryXOR(temp, crc, "10001000000100001");
        strcpy(crc, temp);
    }
}

crc[15] = (data[i] == '1') ? '1' : '0';
}

strcpy(checksum, crc);
}

int main()
{
    char data[100];
    printf("Enter data in binary:");
    scanf("%s", data);

    int dataLength = strlen(data);
    char checksum[17];
    calculateCRC(data, dataLength, checksum);

    printf("Calculated CRC : '%s'\n", checksum);
}

```

//data

char v  
printf  
scanf

if (S  
{

printf  
}

else  
{

printf  
}

return  
}

Output

Enter  
Calculated  
Enter  
Data

Enter  
Calculated  
Enter

Data



11data [2]^ = 0x01; // Unchecked data trailer error

char receivedCheck[17];

printf("Enter received CRC: ");

scanf("%17s", receivedCheck);

if (strcmp(receivedCheck, checksum) == 0)

{  
printf("Data: error free");

}

else

{

printf("Data has errors");

}

return 0;

}

### Output

Enter data in binary: 10001

Calculated CRC: 0111001001000001

Enter received CRC: 1011100100111100

Data contains errors

Enter data in binary

Calculated CRC: 0111001001000001

Enter received CRC: 0111001001000001

Data is error free