

CYCLE: 2

19/12/22

LAB PROGRAM: 1

Q. Write an executive program for error detecting code using CRC 16 bits.

$$\text{CRC 16 bits} \therefore x^{16} + x^{12} + x^5 + 1$$

Data = 1011101

→ Data word

→ Divisor

G(x) = 100010000000100000

modified data = 101110100000000000000000

remainder / checkSum : → appended zeros

→ 17 bits = 17 - 1 = 16 bits to append

Final Codeword : 1011101 +

remainder

+ for error detection : yes or no

$$\begin{array}{r}
 1011000 \\
 \hline
 10001000000100001 \downarrow \\
 \text{XOR} \quad 101110100000000000000000 \\
 \hline
 001100100001000010 \downarrow \\
 \text{XOR} \quad 000000000000000000 \\
 \hline
 011001000010000100 \downarrow \\
 \text{XOR} \quad 10001000000100001 \\
 \hline
 10000000101001010 \downarrow \\
 \text{XOR} \quad 10001000000100001 \\
 \hline
 00010001011010110 \downarrow \\
 \text{XOR} \quad 000000000000000000 \\
 \hline
 00100010110101100 \downarrow \\
 \text{XOR} \quad 000000000000000000 \\
 \hline
 001000101101011000 \downarrow \\
 \text{XOR} \quad 000000000000000000 \\
 \hline
 01000101101011000
 \end{array}$$

remainder / checksum : 1000101101011000

Final Codeword : 1011101 + remainder
~~1011101~~

$$\begin{array}{r}
 1000101101011000 \\
 + \quad 1011101 \\
 \hline
 1000101110110101
 \end{array}$$

↳ this become new data word now

Divisor remains the Same

Receiver's Side

→ Already given in ques

$$\begin{array}{r}
 \begin{array}{c} 1011 \\ \hline 1000100000010000 \end{array} \begin{array}{c} \downarrow \\ \text{XOR} \end{array} \begin{array}{c} 1011101100010110101100 \\ \hline 1000100000010000 \end{array} \downarrow \\
 \begin{array}{c} \text{XOR} \end{array} \begin{array}{c} 0110011000001100 \\ \hline 0000000000000000 \end{array} \downarrow \\
 \begin{array}{c} \text{XOR} \end{array} \begin{array}{c} 011001100000110001 \\ \hline 1000100000010000 \end{array} \downarrow \\
 \begin{array}{c} \text{XOR} \end{array} \begin{array}{c} 1000100000010000 \\ \hline 1000100000010000 \\ \hline 0000000000000000 \end{array}
 \end{array}$$

Therefore the remainder is all zeros.
Hence the data received has no error.

Program:

```
#include <stdio.h>
char m[50], g[50], r[50], v[50], temp[50];
void caltrans (int);
void crc (int);
void alram ();
void shufle ();
int main ()
{
    int n, i=0;
    char ch, flag=0;
    printf ("Enter the frame bits :");
    while ((ch=getchar())!='\n')
        m[i++] = ch;
    n=i;
    for (i=0; i<16; i++)
        m[n++] = '0';
    printf ("message after appending 16 zeros : '%s'\n", m);
    for (i=0; i<16; i++)
        g[i] = '0';
    g[0] = g[4] = g[11] = g[16] = '1'; g[17] = '0';
    printf ("\n generator : '%s'\n", g);
    crc(n);
    printf ("\n quotient : '%s'\n", m);
```


Page _____

```

printf("\nEnter transmitted frame:");
scanf("%s", m);
printf("CRC checking\n");
CRC(m);

```

```

printf("\n\n last remainder : %s",
for (i = 0; i < 16; i++)
    if (err[i] != '0')

```

```

        flag = 1;

```

```

    else

```

```

        continue;

```

```

if (flag == 1)

```

```

    printf("Error during transmission\n");
else

```

```

    printf("\n\n Received frame is correct\n");
}

```

```

void crc(int n)

```

```

{
    int i, j;

```

```

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

```

```

        temp[i] = m[i];

```

```

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

```

```

        r[i] = m[i];

```

```

    printf("\n\n intermediate remainder\n");

```

```

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

```

```

        if (r[i] == '1')

```

```

        {
            r[i] = '1';

```

```

    calnam();
}
else
{
    r[i] = '0';
    shift_l();
}
r[16] = m[17+i];
r[17] = '\0';
printf("\n remainder %d: %s", i+1, r);
for(j=0; j<17; j++)
    temp[j] = r[j];
}
v[n-16] = '\0';
}

void calnam()
{
    int i, j;
    for(i=1; i<16; i++)
        r[i-1] = ((int)temp[i]-48)^(1+int)
        g[i]-48+48;
}

void shift_l()
{
    int i;
    for(i=1; i<16; i++)
        r[i-1] = r[i];
}

```


Date _____
Page _____

```
void calcans (int n)
```

```
{
    int j, k = 0;
    for (i = n - 1; i < n; i++)
        m[i] = ((int) m[i] - 48) ^ ((int)
            r[k++] - 48) + 48;
    m[i] = '\0';
}
```

output:-

Enter the frame bits : 1011101
Message after appending 16 zeros:

101110100000000000000000

generator : 100010000000100001

intermediate remainder