**COMPUTER NETWORK LAB**

**VIVEK SRIVASTAV**

**2K19/CO/437**

**PROGRAM 1**

**AIM : To implement a program on bit stuffing**

**Theory :** Bit stuffing is the mechanism of inserting one or more non-information bits into a message to be transmitted, to break up the message sequence, for synchronization purpose.

* **Bit - Stuffing** − A pattern of bits of arbitrary length is stuffed in the message to differentiate from the delimiter. This is also called bit - oriented framing.

**Frame in a Bit - Oriented Protocol**

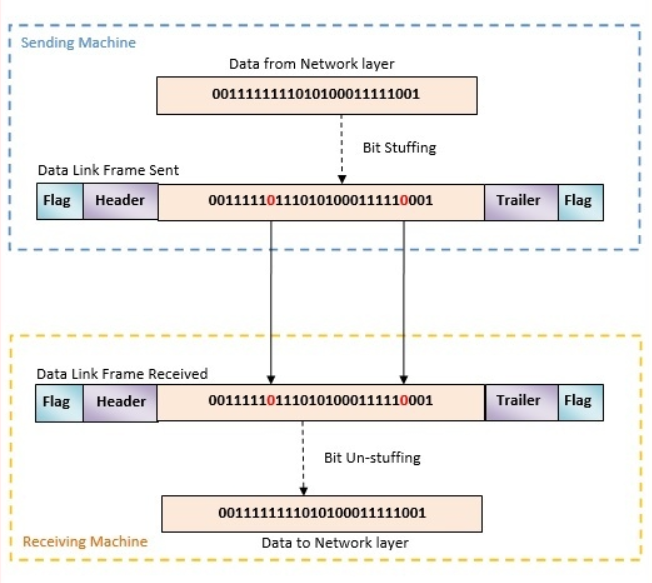
In bit-oriented protocols, the message is coded as a sequence of bits, which are interpreted in the upper layers as text, graphics, audio, video etc. A frame has the following parts −

* Frame Header − It contains the source and the destination addresses of the frame.
* Payload field − It contains the message to be delivered.
* Trailer − It contains the error detection and error correction bits.
* Flags − A bit pattern that defines the beginning and end bits in a frame. It is generally of 8-bits. Most protocols use the 8-bit pattern 01111110 as flag.

**Bit Stuffing Mechanism**

In a data link frame, the delimiting flag sequence generally contains six or more consecutive 1s. In order to differentiate the message from the flag in case of the same sequence, a single bit is stuffed in the message. Whenever a 0 bit is followed by five consecutive 1bits in the message, an extra 0 bit is stuffed at the end of the five 1s.

When the receiver receives the message, it removes the stuffed 0s after each sequence of five 1s. The un-stuffed message is then sent to the upper layers.



**Code :**

#include<bits/stdc++.h>

using namespace std;

int main()

{

    int a[20],b[30],i,j,k,count,n;

    cout<<"Enter frame size:"<<endl;

    cin>>n;

    cout<<"Enter the frame in the form of 0 and 1 :"<<endl;

    for(i=0; i<n; i++)

        cin>>a[i];

    i=0;

    count=1;

    j=0;

    while(i<n)

    {

        if(a[i]==1)

        {

            b[j]=a[i];

            for(k=i+1; a[k]==1 && k<n && count<5; k++)

            {

                j++;

                b[j]=a[k];

                count++;

                if(count==5)

                {

                    j++;

                    b[j]=0;

                }

                i=k;

            }

        }

        else

        {

            b[j]=a[i];

        }

        i++;

        j++;

    }

    cout<<"After Bit Stuffing : ";

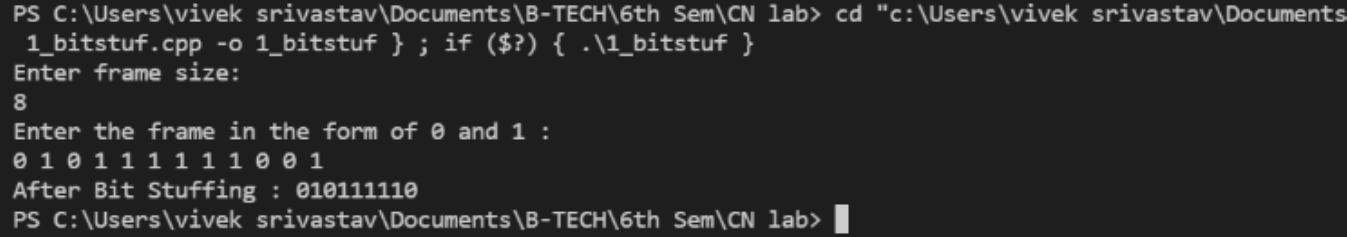
    for(i=0; i<j; i++)

        cout<<b[i]<<"";

    return 0;

}

**Output:**



**Result :**

I have successfully compile this code and In this code of Bit Stuffing I have to inserting an extra bit as 0, once the frame sequence encountered 5 consecutive 1’s.