**SCHOOL OF COMPUTER SCIENCE**

**UNIVERSITY OF PETROLEUM AND ENERGY STUDIES**

**DEHRADUN, UTTARAKHAND**



**DATA COMMUNICATION AND NETWORKS LAB**

**LABORATORY FILE**

**(2024-2025)**

**For**

**Vth Semester**

**Submitted To: Submitted By:**

Prof. Abhishek Yadav Mr. Akshat Negi

Assistant Professor S.S. 500106533(SAP ID)

[Vth Semester] R2142220414(Roll No.)

School of Computer Sciences B.Tech. CSF (Batch-1)

**LAB EXPERIMENT – 2**

**BIT STUFFING AND DE-STUFFING**

**Bit Stuffing** and **De-Stuffing** are techniques used in synchronous data transmission to prevent the occurrence of consecutive ones that could be misinterpreted as control characters. In certain protocols, such as HDLC, a continuous stream of ones is used to indicate the end of a frame. If a series of ones appears unintentionally within the data, it can lead to premature termination of the frame.

**Example**

Consider the following data sequence: 1111111111

**Bit Stuffing:**

111110111110

**Bit De-Stuffing:**

1111011110

As you can see, the extra zero inserted during bit stuffing is removed during de-stuffing, restoring the original data.

**Program for Bit Stuffing**

**Process:**

* The data is transmitted bit by bit.
* If five consecutive ones are encountered, an extra zero is inserted into the data stream.
* This process continues until the end of the frame.

**Purpose:** To ensure that a series of five consecutive ones cannot appear within the data, preventing false termination of the frame.

#include <stdio.h>

#include <string.h>

int main()

{

    char stuff[100];

    int i, count = 0;

    printf("Enter the BIT--> ");

    scanf("%s", stuff);

    printf("Bit-stuffed--> ");

    for (i = 0; stuff[i] != 0; i++)

    {

        printf("%c", stuff[i]);

        if (stuff[i] == '1')

        {

            count++;

            if (count == 5)

            {

                printf("0");

                count = 0;

            }

        }

        else

        {

            count = 0;

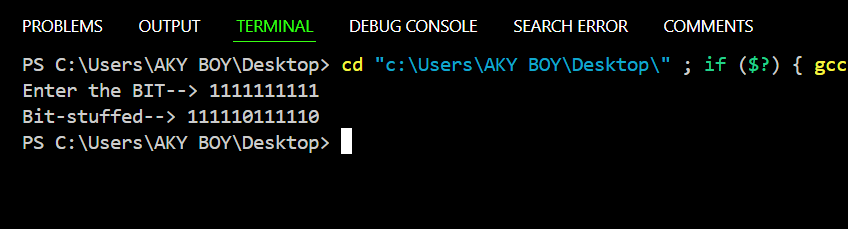
        }

    }

    printf("\n");

    return 0;

}



**Program for Bit De-Stuffing**

**Process:**

* The received data is examined bit by bit.
* If five consecutive ones are encountered followed by a zero, the zero is removed.
* This process continues until the end of the frame.

**Purpose:** To restore the original data by removing the extra zeros inserted during bit stuffing.

#include <stdio.h>

#include <string.h>

int main() {

    char data[100], dest[100];

    int i, j, count = 0;

    printf("Enter the BIT--> ");

    scanf("%s", data);

    printf("Bit De-stuffed--> ");

    for (i = 0, j = 0; data[i] != 0; i++)

    {

        if (data[i] == '1')

        {

            count++;

            if (count == 5 && data[i + 1] == '0')

            {

                count = 0;

                i++;

            }

        }

        else

        {

            count = 0;

        }

        dest[j++] = data[i];

    }

    dest[j] = '\0';

    printf("%s\n", dest);

    return 0;

}

