**4.Binary to Decimal**

Problem Statement:

Write a program to do the following:

Read a string of 0's and 1's of length n and print the decimal equivalent of the string treated as an unsigned integer in the binary representation.

For simplicity, assume that the length of the input string is less than or equal to 32.

Count and print the number of multiplications and additions used.

• Input example :

1. 101111
2. 101205
3. 100011101

• Output example :

1. Decimal equivalent: 47

The number of multiplications: 16

The number of addition: 6

1. Invalid Input
2. Decimal equivalent: 285

The number of multiplications: 37

The number of addition: 9

Proposed C Code:

/\* ------- main.c ------- \*/

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

int main()

{

  printf("Enter binary: ");

  char \*n = (char \*)malloc(32 \* sizeof(char));

  scanf("%s", n);

  int size = strlen(n);

  int b = 1, c = 1;

  unsigned int d = 0;

  for (int i = size - 1; i >= 0; i--)

  {

    if (n[i] - '0' > 1) *// for numbers except binary*

    {

      printf("Invalid Input");

      c = 0;

      break;

    }

    d += (n[i] - '0') \* b; *// taking each element from the right and multiplying it with b*

    b \*= 2; *// each iteration multiply b with 2*

  }

  if (c == 1)

  {

    printf("Decimal equivalent: %d\n", d);

    printf("The number of multiplications: %d\n", (size \* (size - 1) / 2) + 1);

    printf("The number of addition: %d\n", size);

  }

  return 0;

}

/\* ---------------------- \*/

Conclusion:

The proposed algorithm has a runtime of O(n), where n is the size of the string.

Limitations and assumptions for this algorithm include:

1. This program will cause errors for any string which length is above 32.
2. It only the unsigned integer(decimal) value of the binary number.