## INSTRUCTIONS:

1. The project report should be neatly typed.
2. Avoid using Abbreviations.
3. The text should be justified and typed in the Font style ‘Times New Roman’ and Font size ‘12’.
4. Heading and subheading should be bold.
5. The length of the report may be about 10 to 15 pages.

# DON BOSCO INSTITUTE OF TECHNOLOGY

**Skill Lab: C++ and Java Programming**
**MINI PROJECT REPORT**

**On**

## "Title of mini-project"
## 2021-22

**Submitted By:**


| | |
|---|---|
| Student_1: Jaipreet Singh | Student_1: 11 |
| Student_2: Anto ridon | Student_2: 05 |
| Student_3: Nadhanial Antony | Student_3: 16 |

Under the guidance of
Ms. Deepali Kayande

**Mini Project Title** : **4 BIT BINARY ADDER AND SUBTRACTOR**

**Institute Name** : Don Bosco Institute of Technology.

**Institute Address** : Premier Automobiles Road,
Kurla (West), Mumbai – 400070

**Department** : **EXTC**

**Class** : 3rd SEM

**Project Group Members :**

| | Names of students | Roll No. |
|---|---|---|
| 1. | Jaipreet Singh | 11 |
| 2. | Anto Ridon | 05 |
| 3. | Nadhanial Antony | 16 |
| 4. | | |
| 5. | | |

**Date of Submission** : **10/12/2021**

**Guide : Ms. Deepali Kayande**

# TABLE OF CONTENTS

# CHAPTER 1

# INTRODUCTION

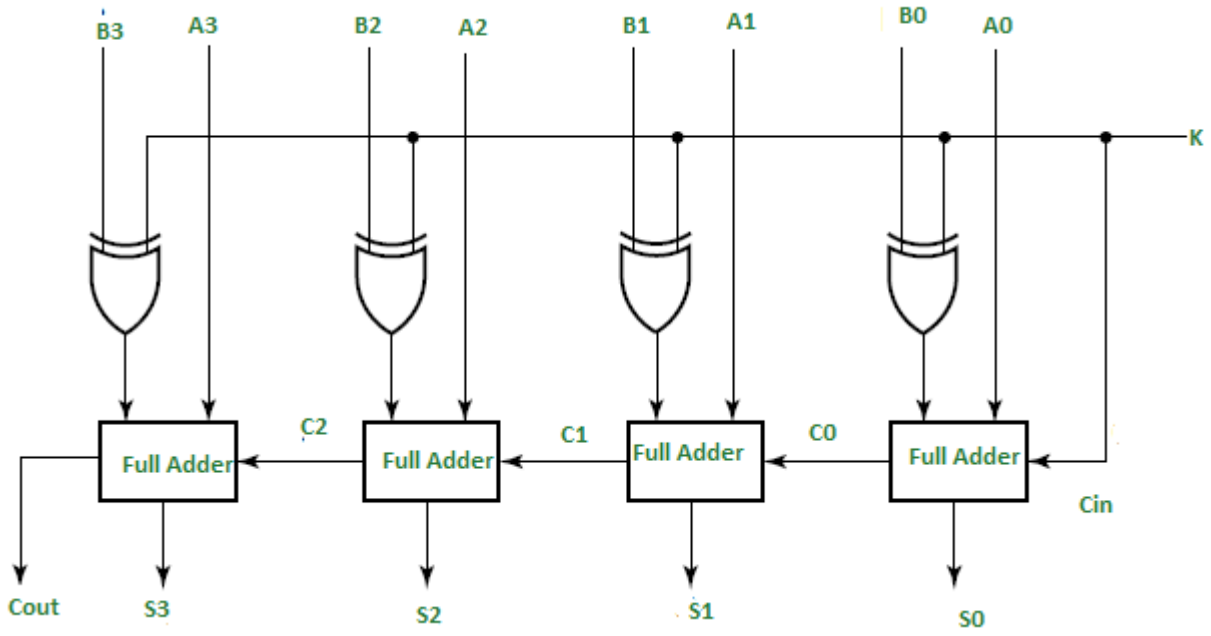
In Digital Circuits, A **Binary Adder-Subtractor** is one which is capable of both addition and subtraction of binary numbers in one circuit itself. The operation being performed depends upon the binary value the control signal holds. It is one of the components of the ALU (Arithmetic Logic Unit).
This Circuit Requires prerequisite knowledge of Exor Gate, Binary Addition and Subtraction, Full Adder.

Lets consider two 4-bit binary numbers A and B as inputs to the Digital Circuit for the operation with digits

`A0 A1 A2 A3 for A`

`B0 B1 B2 B3 for B`

The circuit consists of 4 full adders since we are performing operation on 4-bit numbers. There is a control line K that holds a binary value of either 0 or 1 which determines that the operation being carried out is addition or subtraction.




As shown in the figure, the first full adder has control line directly as its input(input carry Cin), The input A0 (The least significant bit of A) is directly input in the full adder. The third input is the exor of B0 and K. The two outputs produced are Sum/Difference (S0) and Carry (C0).

If the value of K (Control line) is 1, the output of B0(exor)K=B0′(Complement B0). Thus the operation would be A+(B0′). Now 2's complement subtraction for two

numbers A and B is given by A+B’. This suggests that when K=1, the operation being performed on the four bit numbers is subtraction.

Similarly If the Value of K=0, B0 (exor) K=B0. The operation is A+B which is simple binary addition. This suggests that When K=0, the operation being performed on the four bit numbers is addition.

Then C0 is serially passed to the second full adder as one of it’s outputs. The sum/difference S0 is recorded as the least significant bit of the sum/difference. A1, A2, A3 are direct inputs to the second, third and fourth full adders. Then the third input is the B1, B2, B3 EXORed with K to the second, third and fourth full adder respectively. The carry C1, C2 are serially passed to the successive full adder as one of the inputs. C3 becomes the total carry to the sum/difference. S1, S2, S3 are recorded to form the result with S0.

For an n-bit binary adder-subtractor, we use n number of full adders.

**Example:**
Lets take two 3 bit numbers A=010 and B=011 and input them in the full adder with both values of control lines.

For K=0:

B0(exor)K=B0 and C0=K=0

Thus from first full adder

= A0+B0

= 0+1

= 1,

S0=1

C1=0

Similarly,

S1=0 with C2=1

S2=1 and C2=0

Thus,

A = 010 =2

B = 011 = 3

Sum = 0101 = 5

For K=1

B0(exor)K=B0' and C0=k=1

Thus

S0=1 and C1=0

Similarly

S1=1 and C2=0

S2=1 and c3=0

Thus,

A = 010 = 2

B = 011 = 3

Sum(Difference) = 1111 = -1

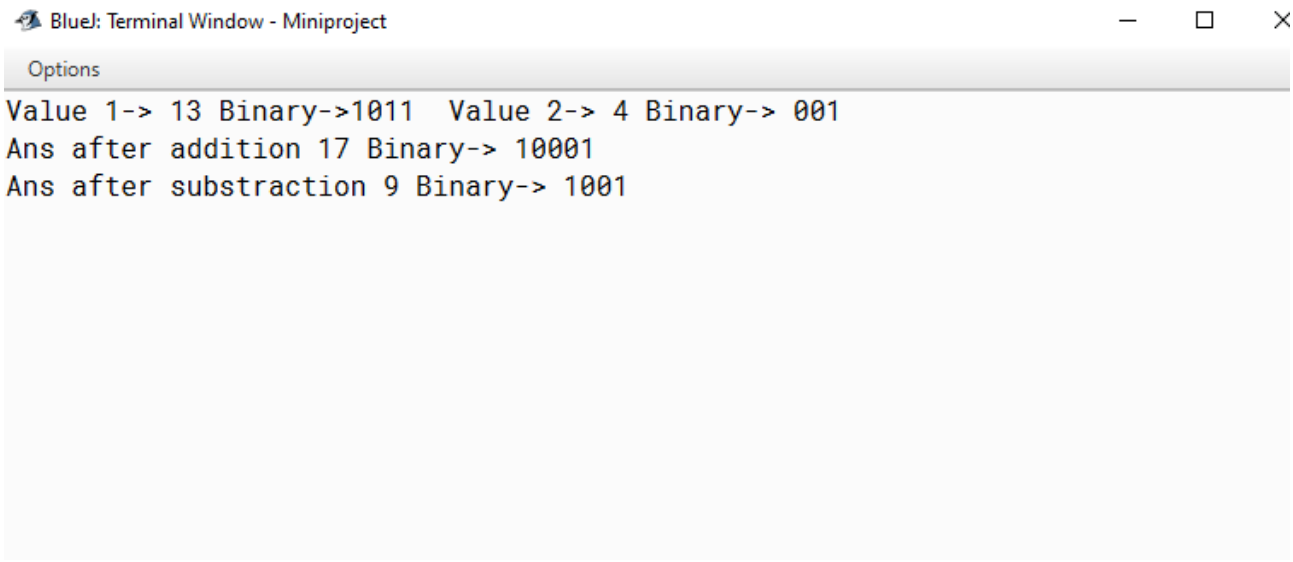
# Implementation: Our code

```
public class MyClass {
   public static int get_adder(int value1,int value2){
      // value 1 take care of addition
      // value 2 take care of carry

      int carry=0;
      while(value2!=0){
         carry = value1&value2;
         value1 = (value1^value2);
         value2=carry<<1;
      }
      int result = value1;
      return result;
   }
   public static int get_substracter(int value1,int value2){
      // 2's complement of value 2
      // add value 1 and value 2
      value2 = ~value2;
      value2+=1;
      return value1+value2;
   }
   public static void main(String args[]) {
     int value1 = 13;
     int value2 = 4;
     String binary_value1 = get_binary_value(value1);
     String binary_value2 = get_binary_value(value2);
     System.out.println("Value 1-> "+value1+" " +"Binary->" + binary_value1+ " "+" Value
2-> "+value2+  " "+ "Binary-> "+ binary_value2);
     int addition = get_adder(value1,value2);
     System.out.println("Ans after addition "+addition + " " + "Binary->
"+get_binary_value(addition));
     int substraction = get_substracter(value1,value2);
     System.out.println("Ans after substraction "+substraction + " " + "Binary->
"+get_binary_value(substraction));
```

```
    }
    public static String get_binary_value(int value){
    int len =0;
    int temp = value;
    String answer = "";
    while(temp != 0){
        len+=1;
        temp>>=1;
    }
    temp = len;
    while(temp!=0){
        int pointer = 1<<(len - temp);
        if ((pointer & value )==0){
            answer += "0";
        }
        else{
            answer += "1";
        }
        temp-=1;
    }
    String result="";
    int space = 32-answer.length();
    for(int i=0;i<space;i++){
        result+="0";
    }
    result+=answer;
    return answer;
}
}
```

# Output:




# Conclusion:

Here we conclude that we have find the 4 bit binary adder & subtractor and the code was verified and the output was obtained.
Blue J software was used for this implementation.

# References:


- https://www.geeksforgeeks.org/4-bit-binary-adder-subtractor/
- https://www.javatpoint.com/coa-binary-adder-subtractor