|  |
| --- |
| **Title-Arithmetic Logic Unit (ALU)** |
|  |
| ***DE Lab Project Report*** |
| ***Submitted By*** |
| **Name of Group Member – Yuvraj Koundal(18BCS6114)**  **Name of Group Member – Pranjali Singhal (18BCS6109)**  **Name of Group Member – Akshay Hooda (18BCS6110)** |
| ***Under the Guidance of*** |
| **Mr. Anshul Sharma**  **Mr. Divneet Singh Kapoor**  **Mr. Khushal Thakur** |
| **Assistant Professors, Academic Unit – 1** |
|  |
| **Academic Unit-1**  **Chandigarh University** |

**Table of Contents**

1. Problem Identification .......................................................................................................... 1

2. Features................................................................................................................................. 2

3. Design Flow ......................................................................................................................... 3

4. Outcome ............................................................................................................................... 4

5. Cost Analysis ........................................................................................................................ 6

7. Product Catalogue ................................................................................................................ 7

**1. Problem Identification**

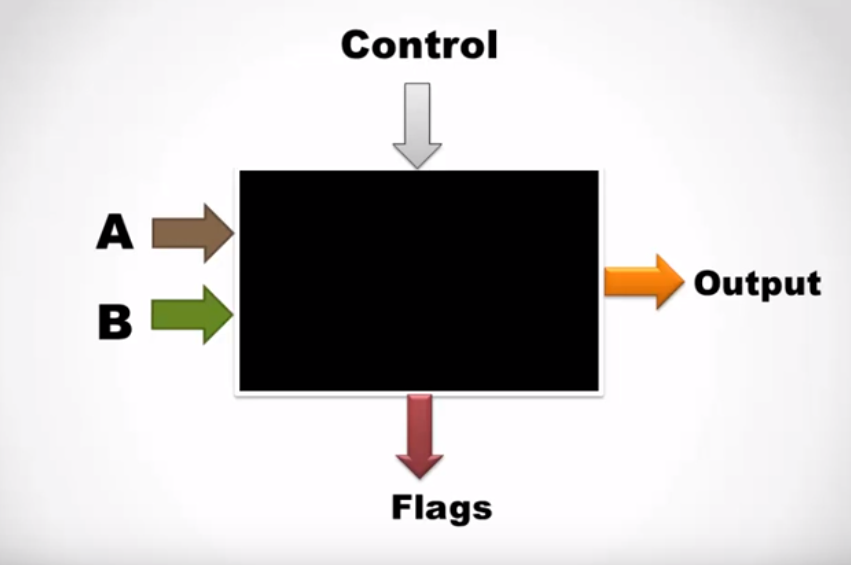
The main agenda of this scenario is to design an ALU with its proper functionality.

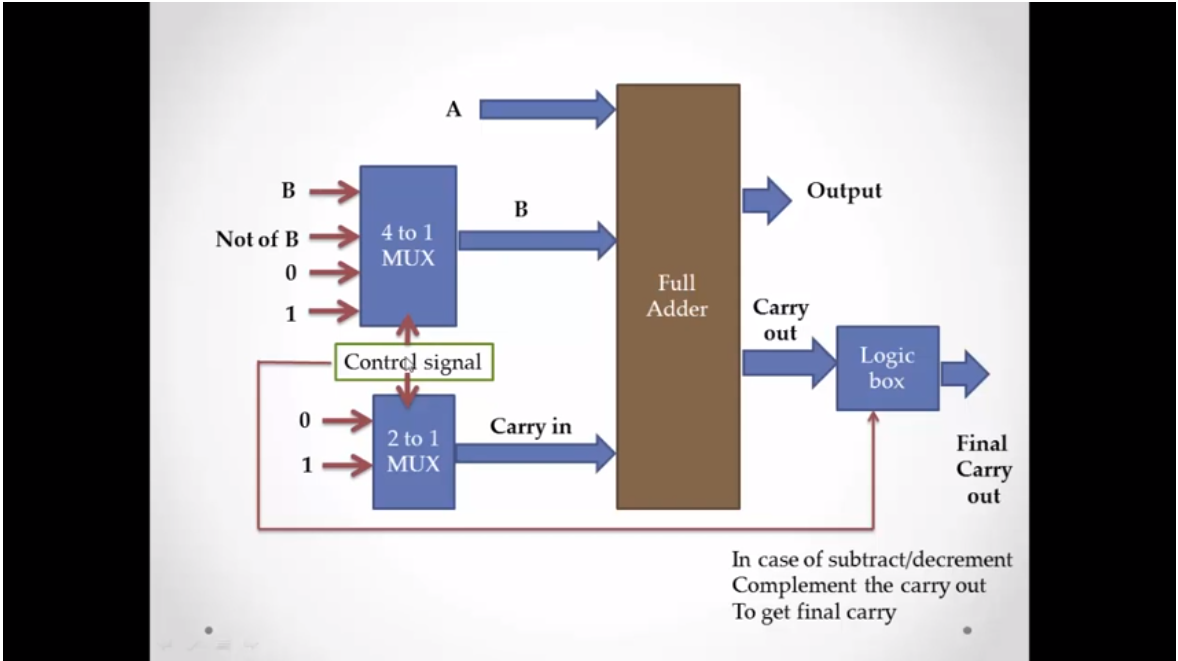
An arithmetic logic unit is a digital circuit used to perform arithmetic and logic operations. It represents the fundamental building block of the central processing unit of a computer. Modern CPUs contain very powerful and complex ALUs. In addition to ALUs, modern CPUs contain a control unit (CU).

Most of the operations of a CPU are performed by one or more ALUs, which load data from input registers. A register is a small amount of storage available as part of a CPU. The control unit tells the ALU what operation to perform on that data, and the ALU stores the result in an output register. The control unit moves the data between these registers, the ALU, and memory.

Hence, the problems solved by this ALU is basic arithmetic problems and conversions.

JUSTIFICATION-

On the basis of our problem identification, we designed a 4-bit ALU on a proteus simulation which works according to the inputs given to it through the help of the selector lines and output provided by the led matrices. 



**2. Features**

An ALU performs basic arithmetic and logic operations. Examples of arithmetic operations are addition, subtraction, multiplication, and division. Examples of logic operations are comparisons of values such as NOT, AND, and OR.

All information in a computer is stored and manipulated in the form of **binary numbers**, i.e. 0 and 1. **Transistor** switches are used to manipulate binary numbers since there are only two possible states of a switch: open or closed. An open transistor, through which there is no current, represents a 0. A closed transistor, through which there is a current, represents a 1.

Operations can be accomplished by connecting multiple transistors. One transistor can be used to control a second one - in effect, turning the transistor switch on or off depending on the state of the second transistor. This is referred to as a **gate** because the arrangement can be used to allow or stop a current.

|  |
| --- |
| The simplest type of operation is a NOT gate. This uses only a single transistor. It uses a single input and produces a single output, which is always the opposite of the input. |

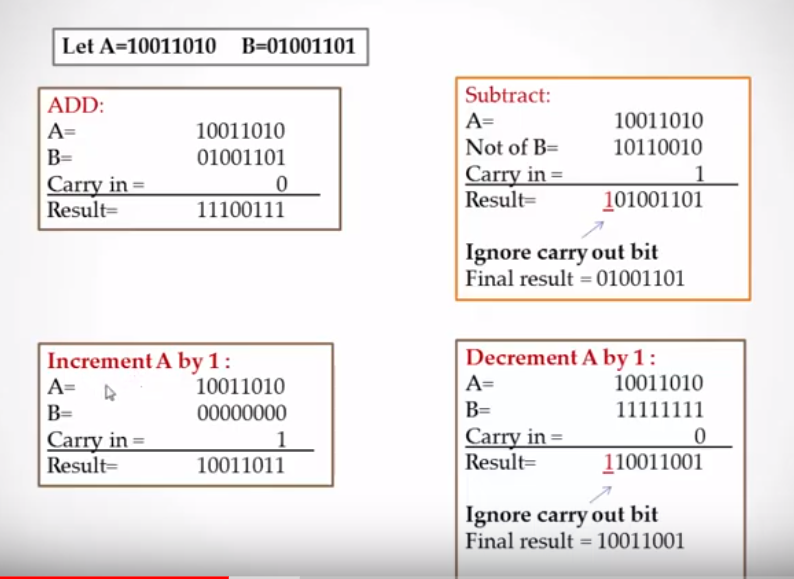
Other gates consist of multiple transistors and use two inputs. The OR gate results in a 1 if either the first or the second input is a 1. The OR gate only results in a 0 if both inputs are 0. This figure shows the logic of the OR gate:

The AND gate results in a 1 only if both the first and second input are 1s

The XOR gate, also pronounced X-OR gate, results in a 0 if both the inputs are 0 or if both are 1. Otherwise, the result is a 1

The overall features of our ALU are:

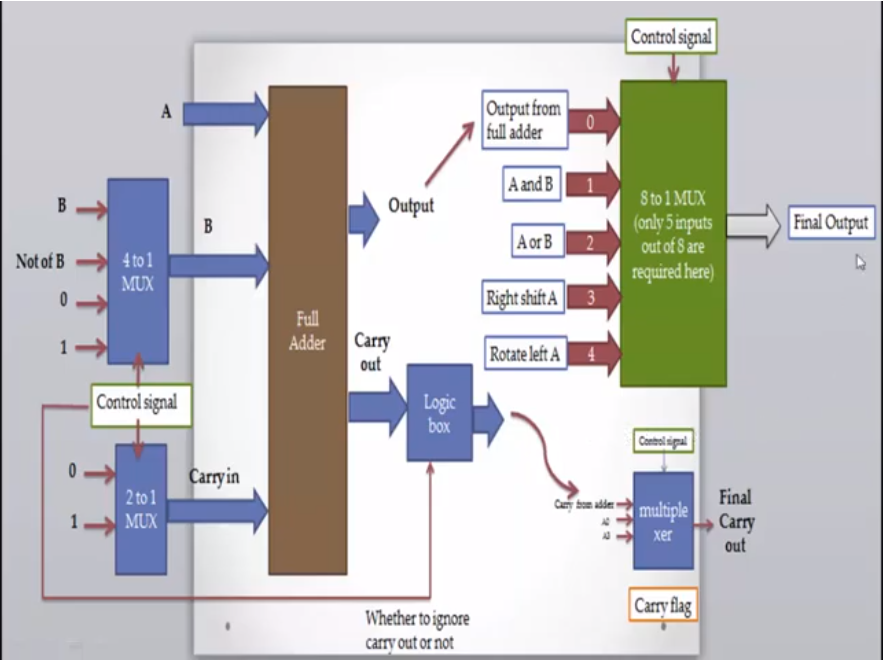
1. It takes data in the form of 4-bit with the help of selector lines designed in the proteus simulation.
2. The output provided by the ALU proteus simulation is in the form of LED’s.
3. The number of digits it takes is 2 i.e. A and B.



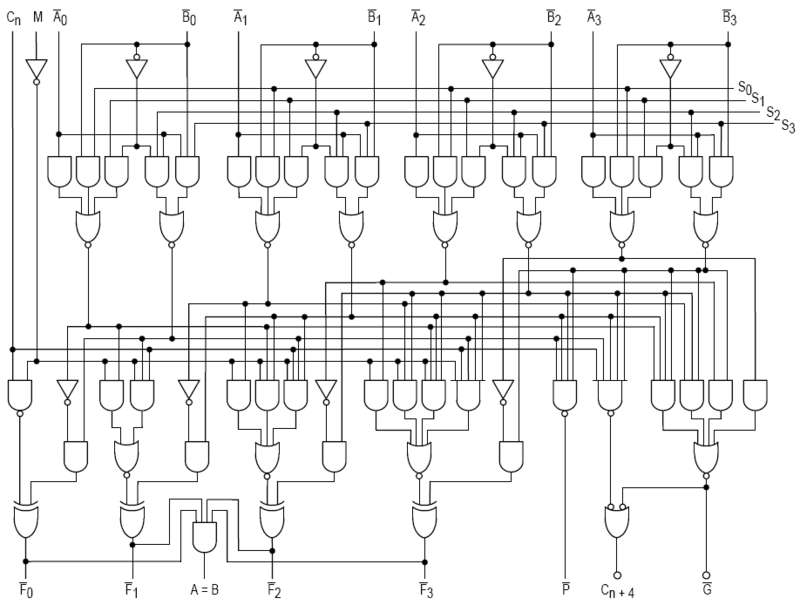
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **S3** | **S2** | **S1** | **S0** | **Carry in** | **Output** | **Carry out(D10)** |
| **0** | **0** | **0** | **0** | **0** | **A+B** | **Consider** |
| **1** | **1** | **1** | **1** | **1** | **A-B** | **Avoid** |
| **0** | **0** | **0** | **0** | **1** | **A+1** | **Consider** |
| **1** | **1** | **1** | **1** | **0** | **A-1** | **Avoid** |

**3. Design Flow**

# Block Diagram



# Circuit Diagram



This is a complete circuit of ALU but we have only covered the arithmetic part of ALU

# Materials:

*This has been performed on proteus so no hardware was required.*

# Steps of Circuit Completion:

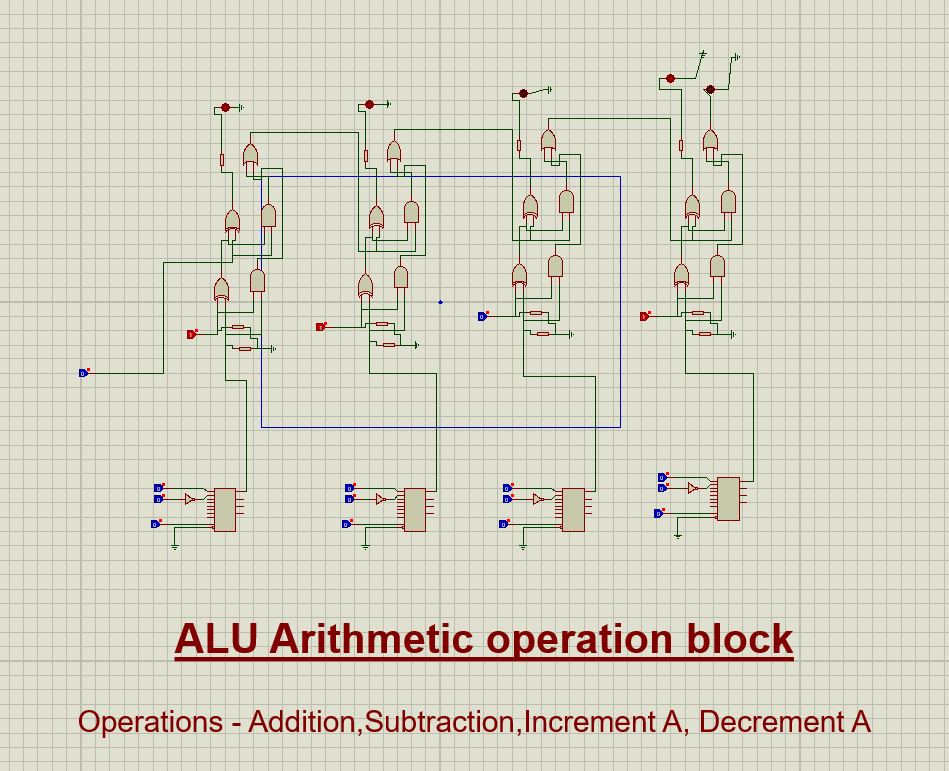
It was a complete virtual proteus simulation.

**4. Outcome**

# Steps of Circuit Completion

Because ALUs can be built in so many ways with wide speciﬁcations and since the objective of the class project is to learn the basic of VLSI design, the speciﬁcations of the ALU were relaxed. The main objective of the project is to have a working ALU that performs different arithmetic and logic functions for all possible combinations of the inputs. The speed of ALU was not an issue and we wanted it to run at low power.

Hence, it is concluded that a virtual proteus simulation has been done.



**5. Cost Analysis**

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Component / Material** | **Price (in Rs.)** |
| 1. | N/A | N/A |
| 2. | N/A | N/A |
| 3. | N/A | N/A |
| **Total**N/A | |  |

Since it was a proteus simulation so no effective amount of money was used during this project.