NOTRE DAME UNIVERSITY
BANGLADESH

Digital System Design Lab Task-03

Course Code: CSE-4106
Course Title: Digital System Design Lab
Lab Task Topic: Design 4bit Arithmetic Logic Unit

Submitted by: Group-05

Name: Istiak Alam
ID: 0692230005101005
Name: Nazifa Tasnim
ID: 0692230005101007
Batch: CSE-20

Submission Date: November 9, 2025

Submitted to:

Afsana Sharmin Shanta
Lecturer,
Notre Dame University Bangladesh

Question :

Design a 4-bit Arithmetic Logic Unit (ALU), which generates the following operations. Also show four status registers (Sign flag, Carry flag, Overflow flag, Zero flag).

Operations	Functions
$F = B + 1$	Increment B
$F = B$	Transfer B
$F = A - B$	Subtraction
$F = A - B - 1$	Subtraction with Borrow
$F = A \oplus B$	Exclusive OR
$F = A \odot B$	Exclusive NOR

1 Abstract

This report presents the design and implementation of a 4-bit Arithmetic Logical Unit (ALU) using basic logic gates in Proteus simulation. The ALU performs six operations, both arithmetic and logical, based on select inputs and a carry-in signal. Additionally, status flags such as Sign Flag (SF), Carry Flag (CF), Overflow Flag (OF), and Zero Flag (ZF) are integrated to indicate result conditions.

2 Introduction

An Arithmetic Logical Unit (ALU) is a fundamental component of a CPU that performs arithmetic and logical operations. In this project, a 4-bit ALU was designed to perform operations such as increment, transfer, subtraction, XOR, and XNOR based on control signals.

3 Truth Table

The ALU uses three select lines (S_2, S_1, S_0) and one carry-in input (Cin) to select the required operation. The complete truth table is shown below:

S2	S1	S0	Cin	X	Y	Function
0	0	0	1	0	B	Increment B $\rightarrow$ B+1
0	0	1	0	0	B	Transfer B $\rightarrow$ B
0	1	0	1	A	B'	Subtraction $\rightarrow$ A - B
0	1	1	0	A	B'	Subtraction with Borrow $\rightarrow$ A - B - 1
1	0	0	x	A	B	Logical XOR $\rightarrow$ A $\oplus$ B
1	0	1	x	A	B'	Logical XNOR $\rightarrow$ A $\odot$ B

4 Function Derivations

The internal signals X, Y, and Z are defined using the select inputs as follows:

- $X = A \cdot (S_1 \oplus S_2)$
- $Y = B \oplus (S_1 + (S_2 \cdot S_0))$
- $Z = \overline{S_2} \cdot C_{in}$

These functions determine how the ALU inputs are modified for each operation before being processed by the full adder network.

5 Circuit Diagram

The 4-bit ALU circuit was designed in Proteus using:

- 4-bit Full Adder IC (e.g., 7482)
- Logic gates (XOR, OR, NOT, AND)
- Select-pin for operation selection
- Input switches for X, Y, and Cin
- **The Number of Logic Gates is 27; Excluded F/A, it will be 17**

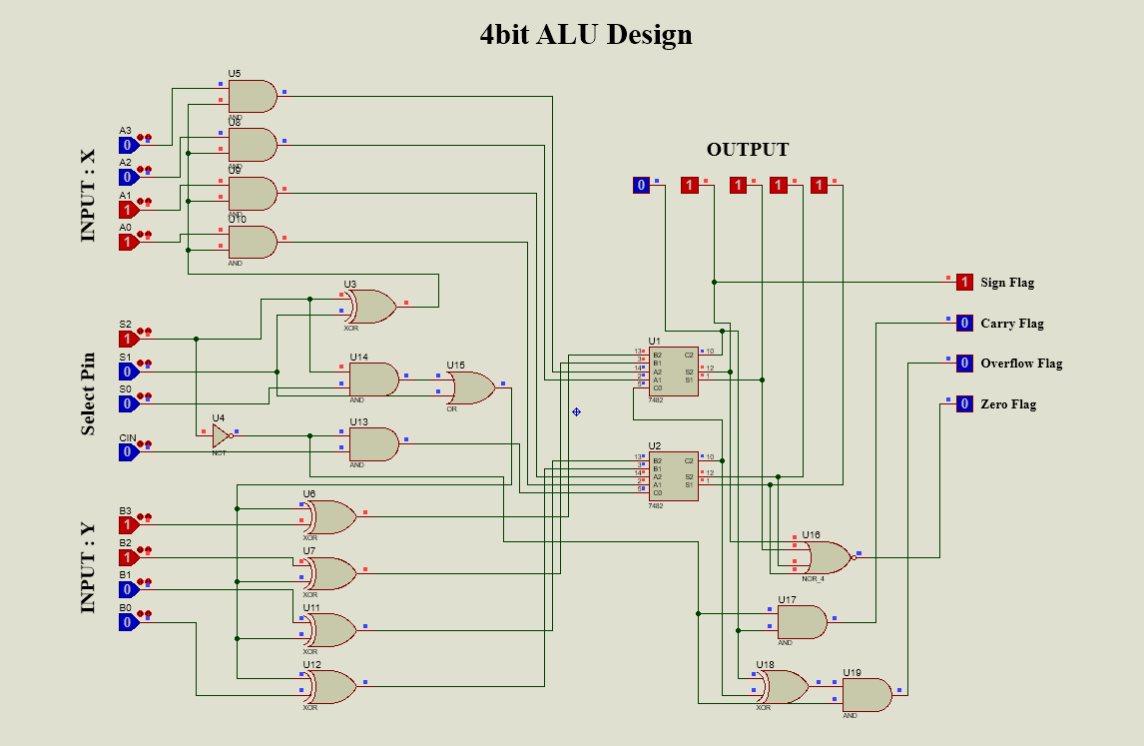

Figure 1: 4-bit ALU Circuit Diagram (Proteus Simulation)

6 Status Flags Implementation

The ALU generates four status flags to indicate the condition of the result:

- **Sign Flag (SF):** Indicates the sign of the result. It is set if the MSB of the result (F3) = 1.

$$SF = F_3$$

- **Zero Flag (ZF):** Set when all output bits are 0.

$$ZF = \overline{(F_0 + F_1 + F_2 + F_3)}$$

- **Carry Flag (CF):** Indicates if there is a carry out from the most significant bit.

$$CF = \overline{S_2} \cdot C_{out}$$

- **Overflow Flag (OF):** Set if there is a signed overflow.

$$OF = (C_3 \oplus C_{out}) \cdot \overline{S_2}$$

In Proteus, these flags can be displayed using Logicprobe connected to the corresponding outputs of logic circuits:

- Use XOR gate for OF
- Use NOT and OR gates for ZF
- Connect MSB directly for SF
- Use the carry output and previous carry from adder for CF

7 Conclusion

The designed 4-bit ALU successfully performs both arithmetic and logic operations as expected based on select input combinations. The inclusion of four status flags allows the ALU to be extended easily for microprocessor-like control applications. This experiment provides a clear understanding of how arithmetic logic units function within CPU architecture.

8 Project File

https://github.com/Istiaq-Alam/Digital-System-Design-Lab/tree/main/Lab%20Test/Lab%20Test-03