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
- **Number of Logic Gates are 27**

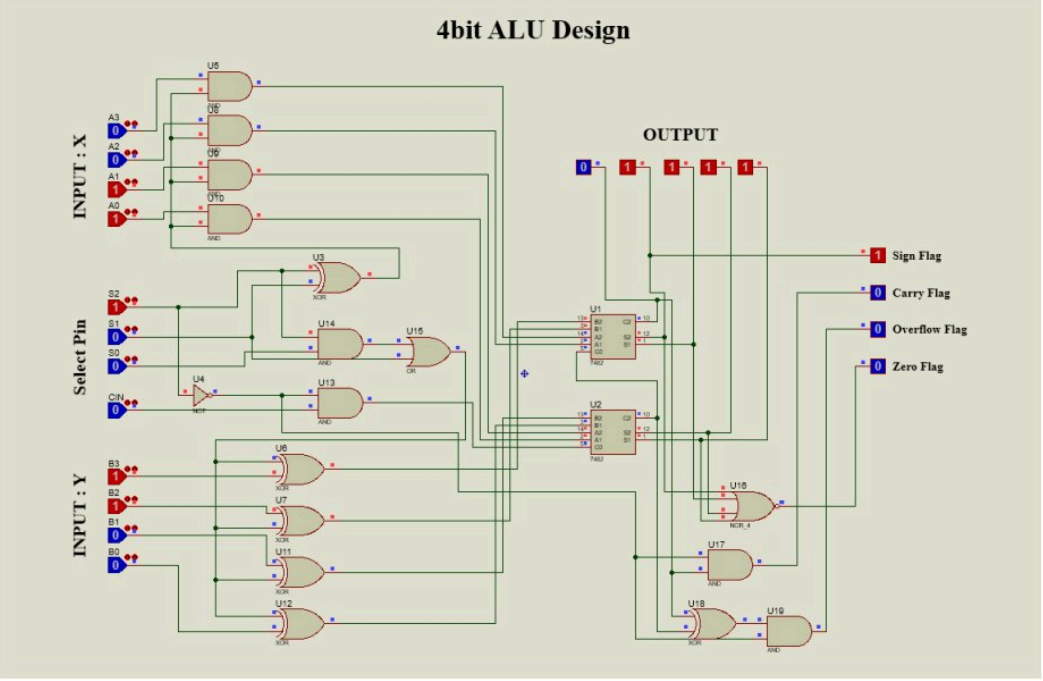


Figure 1: 4-bit ALU Circuit Diagram (Proteus Simulation)

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

NOTRE DAME UNIVERSITY BANGLADESH

Project Report

Course Title: Digital System Design Lab
Course Code: CSE 4106
Date of Submission: 09/11/25

Submitted by:
Name: Maria Fardus
ID: 0692230005101003
Name: Tanvir Hossain Ratul
ID: 0692230005101004
Batch: CSE20

Submitted To:
Afsana Sharmin Shanta
Lecturer, Department of CSE
NDUB

Abstract

This report details the design, optimization, and simulation of a 4-bit Arithmetic Logic Unit (ALU), a critical component of any digital system or Central Processing Unit (CPU). The primary objective was to implement an ALU capable of executing six fundamental arithmetic and logic operations. A core constraint of this project was the strict prohibition of multiplexer components, necessitating an optimized, purely combinatorial design constructed exclusively from basic logic gates. The methodology involved deriving simplified Boolean equations for both the functional units and the control logic. Furthermore, the design was engineered to accurately generate four status flags—Sign, Carry, Overflow, and Zero—to support sequential control and conditional branching in a microarchitecture. The optimized design was implemented and verified using the Proteus simulation software, confirming full functionality and providing a comprehensive count of the total logic gates used.

Introduction:

The objective of this project is to develop a complete and optimized **4-bit ALU** capable of processing two 4-bit operands (A and B). The design must be capable of generating the following six specific operations based on control signal inputs:

Operation	Function
F = A	Transfer A
F = A + 1	Increment A
F = A - B - 1	Subtract with Borrow
F = A - B	Subtraction
F = A OR B	OR
F = A NAND B	NAND

State Table :

S2	S1	S0	Cin	X	Y	F
0	0	0	0	A	0	Transfer A
0	0	0	1	A	0	Increment A
0	0	1	0	A	B’	Subtract with Borrow
0	0	1	1	A	B’	Subtraction
1	0	0	x	A+B	0	OR
1	0	1	x	A+B’	B	NAND

Functions:

$X = S_2 S_1'S_0'(A+B) + S_2S_1'S_0'(A+B') + S_2S_1'S_0'(A+B) + S_2S_1'S_0'(A+B')$

$= A + AS_2S_1'S_0' + BS_2S_1'S_0' + AS_2S_1'S_0' + B'S_2S_1'S_0'$

$= A + S_2S_1'(S_0'B+S_0B')$

$= S_1'A + S_2S_1'(S \oplus B)$

$Y = S_1'S_0B' + S_1'S_0B$

$= S_1'S_0(B+B')$

$Z = S_2'$ Cin

Circuit :

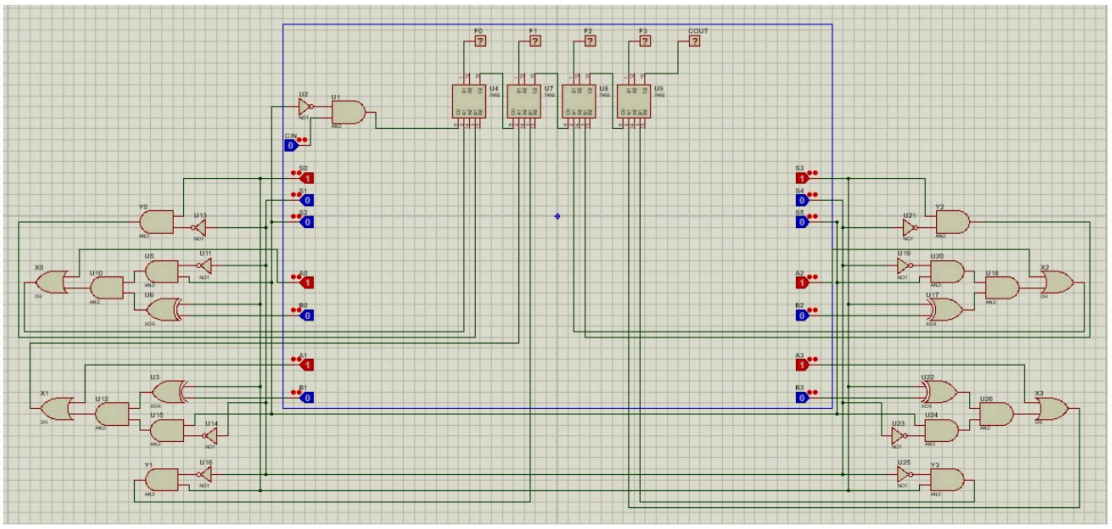


Figure : 4 bit Arithmetic Logic Unit

NOTRE DAME UNIVERSITY BANGLADESH

Lab Project Report

Course title : **Digital System Design**

Course code : **CSE 4106**

Submitted by ,

Group 11

Name : Md. Hemal Ahmed. _ **ID :** 0692230005101013

Name : Prima Saha. _ **ID :** 0692230005101014

Name : Md. Rasin Hasan Shad. _ **ID :** 0692230005101015

Batch : CSE 20

Department of Computer Science & Engineering.

Submitted to ,

Afsana Sharmin Shanta

Lecturer, NDUB

Department of Computer Science & Engineering

Submission date :

Tables :

S0	S1	S2	Cin	x	y	Functions
0	**0**	**0**	**0**	**A**	**B**	Addition, **B+A**
0	**0**	**0**	**1**	**A**	**B**	Add with carry, **B+A+1**
0	**0**	**1**	**0**	**All 1**	**B**	Decrement B, **B-1**
0	**1**	**0**	**0**	**0**	**B**	Transfer B, **B**
1	**0**	**0**	**X**	**A**	**B**	Exclusive-OR, **A XOR B = $A \oplus B$**
1	**0**	**1**	**X**	**1**	**A+B**	NOR, **A NOR B = $\overline{A + B}$**

Equations :

$$y = B + S2\,\overline{S1}\,S0\,(A+B)$$

$$= B + S2\,\overline{S1}\,S0\,A + S2\,\overline{S1}\,S0\,B$$

$$= B\,(1 + S2\,\overline{S1}\,S0) + S2\,\overline{S1}\,S0\,A$$

$$= B + S2\,\overline{S1}\,S0\,A$$

$\mathbf{x} = \overline{S1}\ \overline{S0}\ \mathrm{A} + \overline{S1}\ S0$

$\mathbf{z} = \overline{S2}\ \mathrm{Cin}$

Total number of gates :

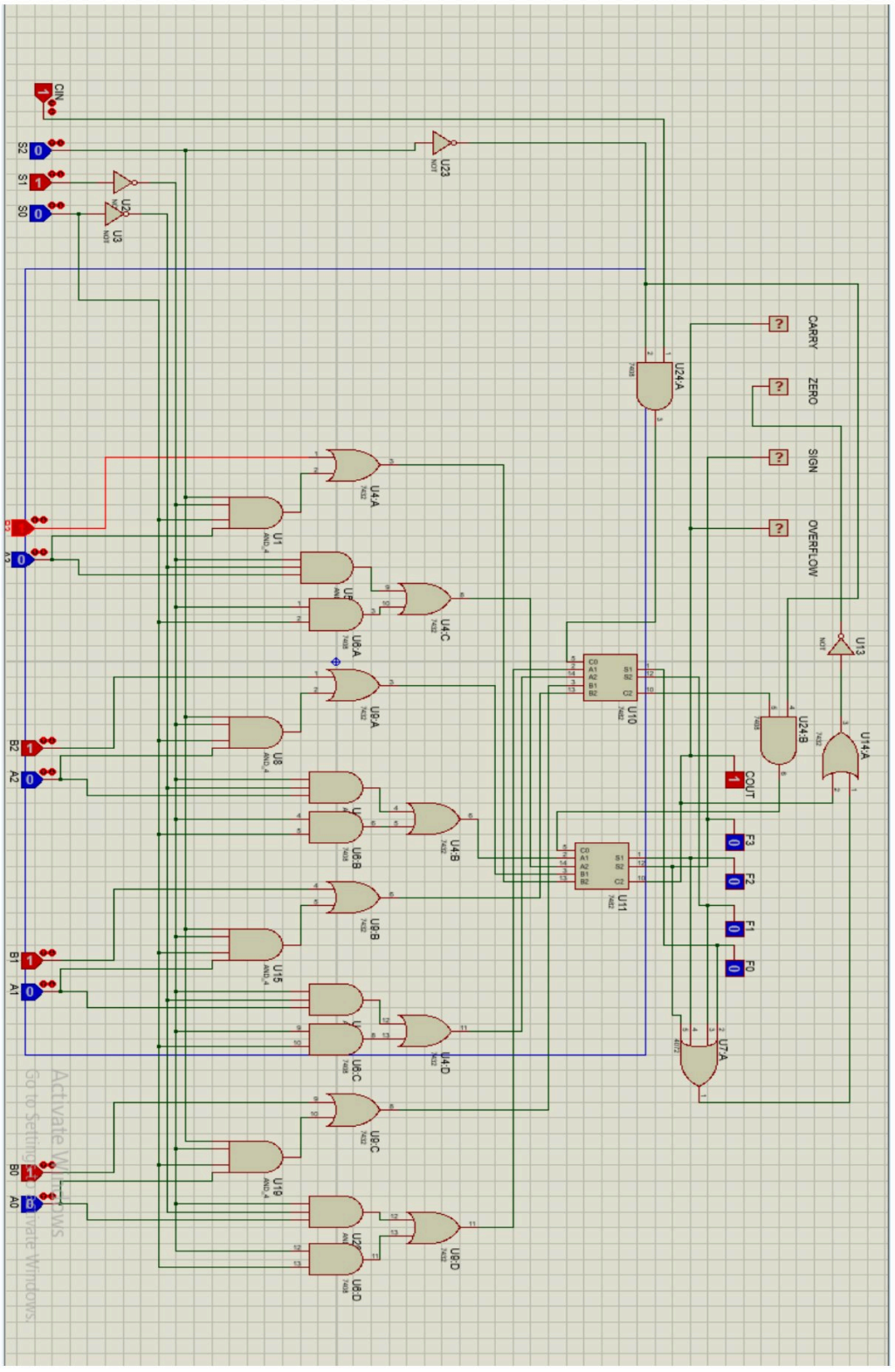
AND gate : 14

OR gate : 10

NOT gate : 4

:. Total number of 28 gates we used.

Circuit diagram :



NOTRE DAME UNIVERSITY
B A N G L A D E S H

Project-01 (Group 08)

Course Title : Digital System Design Lab

Course Code: CSE-4106

Submitted by:

Raisha Anjum (0692230005101019)

Ruhi Tahmidul Al Rashedi (0692230005101020)

Batch: CSE 20

Date: 9/11/25

Submitted To:

Afsana Sharmin Shanta
Lecturer
Dept. of CSE
NDUB

Problem :

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).

Operation	Function
F=B–A–1	Subtract with borrow
F=B–A	Subtraction
F=B–1	Decrement B
F = B	Transfer B
F=A XOR B	Exclusive-OR
F=B'	Complement B

State Table:

S2	S1	S0	Cin	X	Y	Function
0	0	0	0	A’	B	Sub with borrow
0	0	0	1	A’	B	Subtraction
0	0	1	0	All 1	B	Decrement B
0	1	0	0	0	B	Transfer B
1	0	0	0(don’t care)	A	B	Exclusive OR
1	0	1	0(don’t care)	All 1	B	Complement B)

Equations:

X=S2’S1’S0’A’+S1’S0+S2S1’S0’A

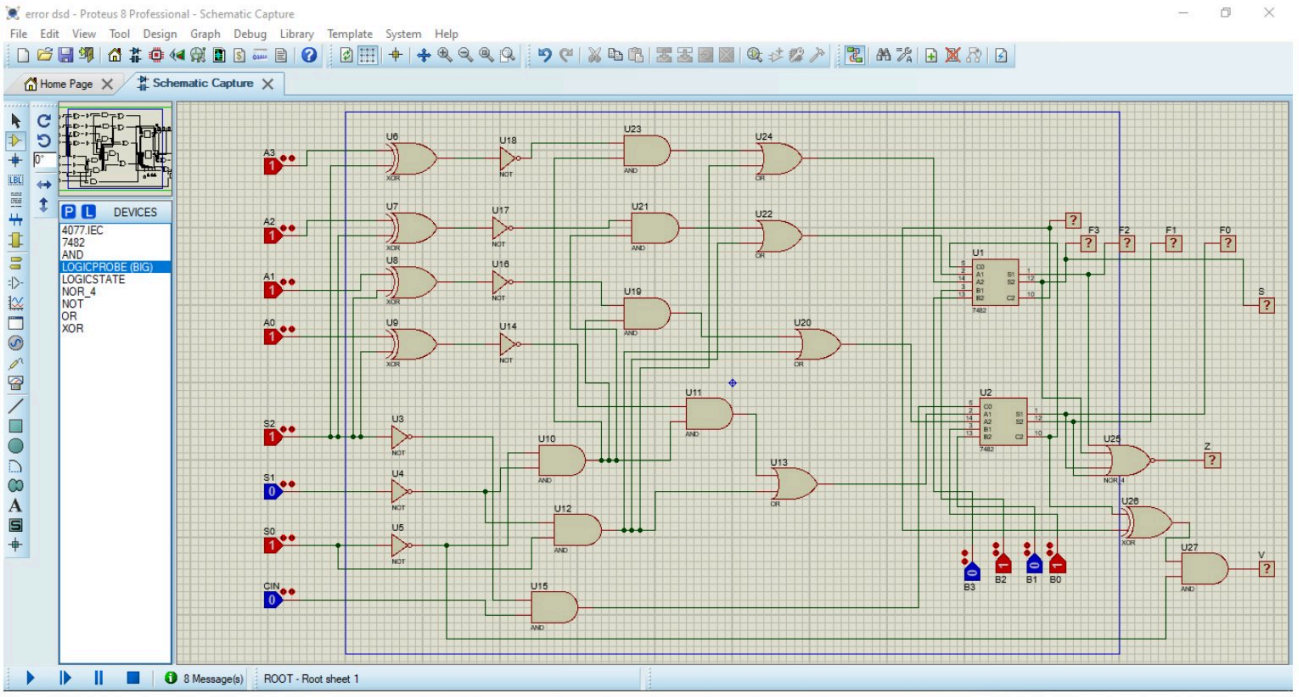
= S1’S0’(S2’A’+S2A) +S1’S0

=S1'S0'(S2 XNOR A)+S1'S0

Y=B

Z=S2'Cin

Circuit Design:



NOTRE DAME UNIVERSITY BANGLADESH

Project Report

Course Title: Digital System Design Lab
Course Code: CSE 4106
Date of Submission: 09/11/25

Submitted by:
Name: Maria Fardus
ID: 0692230005101003
Name: Tanvir Hossain Ratul
ID: 0692230005101004
Batch: CSE20

Submitted To:
Afsana Sharmin Shanta
Lecturer, Department of CSE
NDUB

Abstract

This report details the design, optimization, and simulation of a 4-bit Arithmetic Logic Unit (ALU), a critical component of any digital system or Central Processing Unit (CPU). The primary objective was to implement an ALU capable of executing six fundamental arithmetic and logic operations. A core constraint of this project was the strict prohibition of multiplexer components, necessitating an optimized, purely combinatorial design constructed exclusively from basic logic gates. The methodology involved deriving simplified Boolean equations for both the functional units and the control logic. Furthermore, the design was engineered to accurately generate four status flags—Sign, Carry, Overflow, and Zero—to support sequential control and conditional branching in a microarchitecture. The optimized design was implemented and verified using the Proteus simulation software, confirming full functionality and providing a comprehensive count of the total logic gates used.

Introduction:

The objective of this project is to develop a complete and optimized **4-bit ALU** capable of processing two 4-bit operands (A and B). The design must be capable of generating the following six specific operations based on control signal inputs:

Operation	Function
F = A	Transfer A
F = A + 1	Increment A
F = A - B - 1	Subtract with Borrow
F = A - B	Subtraction
F = A OR B	OR
F = A NAND B	NAND

State Table :

S2	S1	S0	Cin	X	Y	F
0	0	0	0	A	0	Transfer A
0	0	0	1	A	0	Increment A
0	0	1	0	A	B’	Subtract with Borrow
0	0	1	1	A	B’	Subtraction
1	0	0	x	A+B	0	OR
1	0	1	x	A+B’	B	NAND

Functions:

$X = S_2 S_1'S_0'(A+B) + S_2S_1'S_0'(A+B') + S_2S_1'S_0'(A+B) + S_2S_1'S_0'(A+B')$

$= A + AS_2S_1'S_0' + BS_2S_1'S_0' + AS_2S_1'S_0' + B'S_2S_1'S_0'$

$= A + S_2S_1'(S_0'B+S_0B')$

$= S_1'A + S_2S_1'(S \oplus B)$

$Y = S_1'S_0B' + S_1'S_0B$

$= S_1'S_0(B+B')$

$Z = S_2'$ Cin

Circuit :

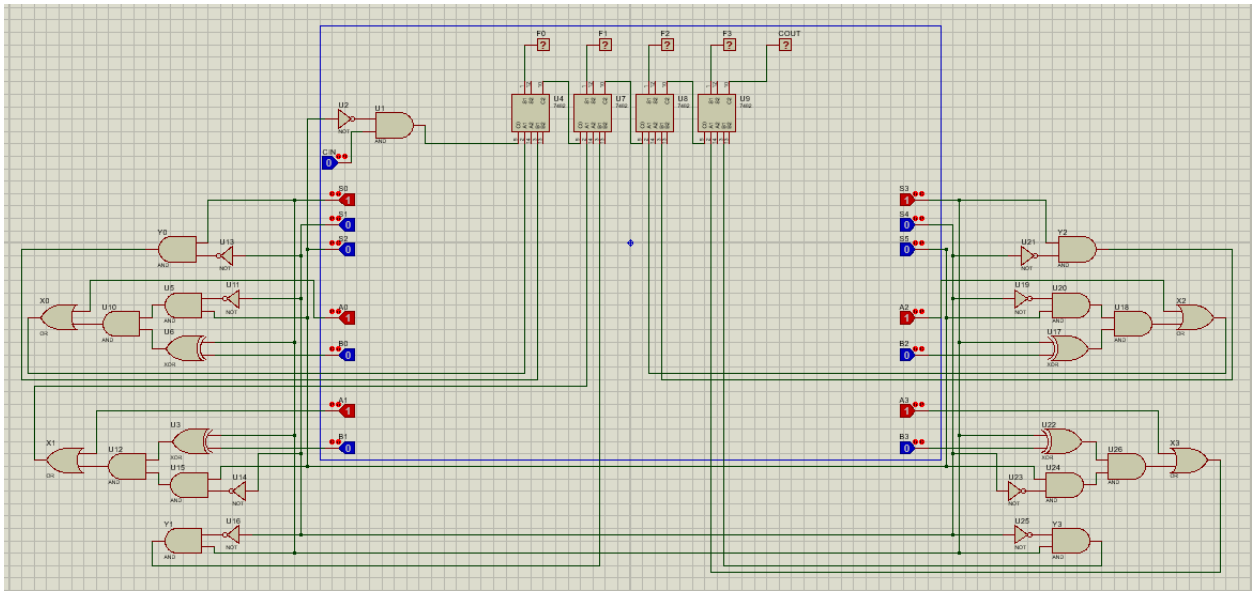


Figure : 4 bit Arithmetic Logic Unit

NOTRE DAME UNIVERSITY
BANGLADESH

Project-01 (Group 08)

Course Title : Digital System Design Lab

Course Code: CSE-4106

Submitted by:

Raisha Anjum (0692230005101019)

Ruhi Tahmidul Al Rashedi (0692230005101020)

Batch: CSE 20

Date: 9/11/25

Submitted To:

Afsana Sharmin Shanta
Lecturer
Dept. of CSE
NDUB

Problem :

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).

Operation	Function
F=B–A–1	**Subtract with borrow**
F=B–A	**Subtraction**
F=B–1	**Decrement B**
F = B	**Transfer B**
F=A XOR B	**Exclusive-OR**
F=B'	**Complement B**

State Table:

S2	S1	S0	Cin	X	Y	Function
0	0	0	0	A’	B	Sub with borrow
0	0	0	1	A’	B	Subtraction
0	0	1	0	All 1	B	Decrement B
0	1	0	0	0	B	Transfer B
1	0	0	0(don’t care)	A	B	Exclusive OR
1	0	1	0(don’t care)	All 1	B	Complement B)

Equations:

X=S2’S1’S0’A’+S1’S0+S2S1’S0’A

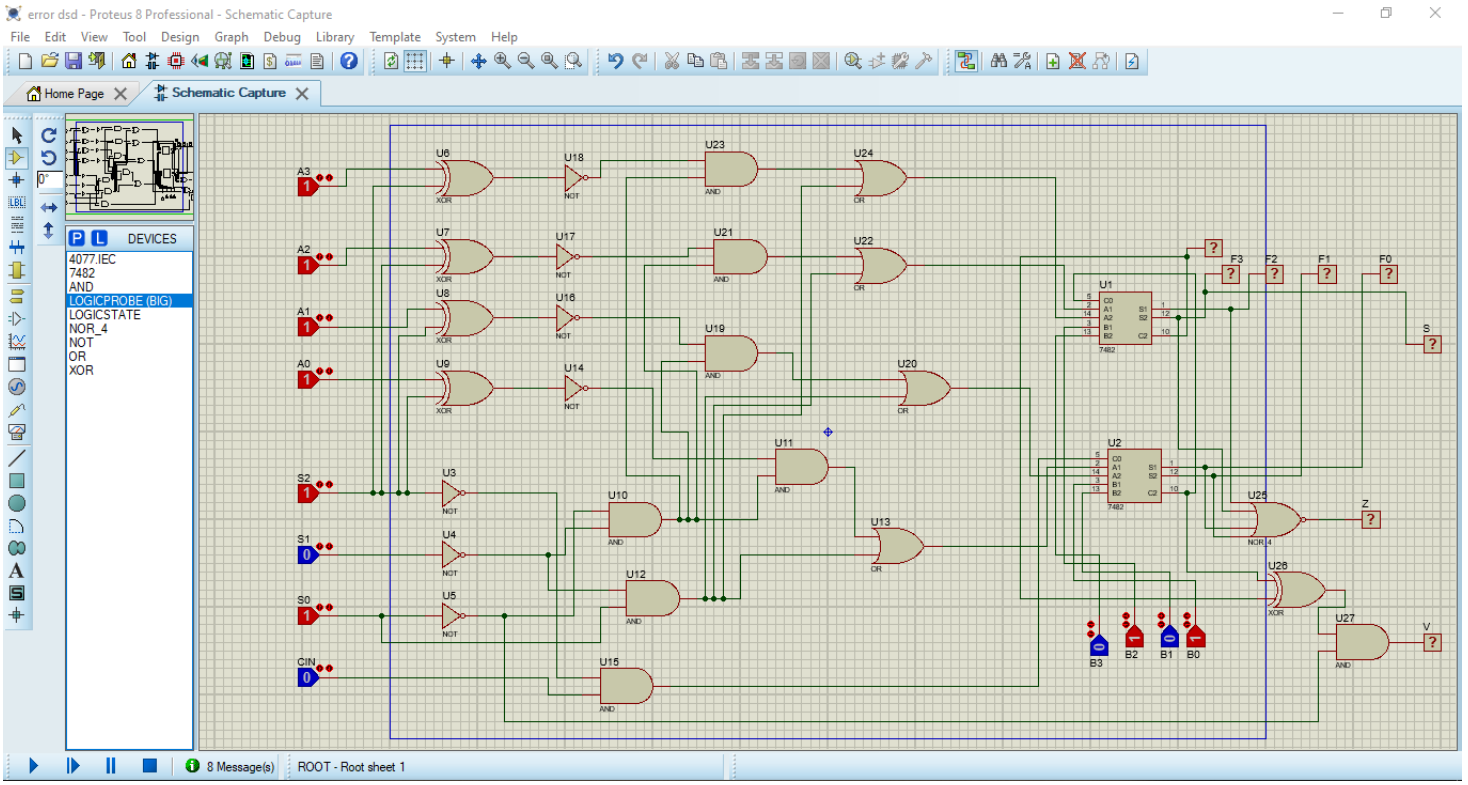
= S1’S0’(S2’A’+S2A) +S1’S0

=S1'S0'(S2 XNOR A)+S1'S0

Y=B

Z=S2'Cin

Circuit Design:



NOTRE DAME UNIVERSITY BANGLADESH

Lab Project Report

Course title : **Digital System Design**

Course code : **CSE 4106**

Submitted by ,

Group 11

Name : Md. Hemal Ahmed. _ **ID :** 0692230005101013

Name : Prima Saha. _ **ID :** 0692230005101014

Name : Md. Rasin Hasan Shad. _ **ID :** 0692230005101015

Batch : CSE 20

Department of Computer Science & Engineering.

Submitted to ,

Afsana Sharmin Shanta

Lecturer, NDUB

Department of Computer Science & Engineering

Submission date :

Tables :

S0	S1	S2	Cin	x	y	Functions
0	**0**	**0**	**0**	**A**	**B**	Addition, **B+A**
0	**0**	**0**	**1**	**A**	**B**	Add with carry, **B+A+1**
0	**0**	**1**	**0**	**All 1**	**B**	Decrement B, **B-1**
0	**1**	**0**	**0**	**0**	**B**	Transfer B, **B**
1	**0**	**0**	**X**	**A**	**B**	Exclusive-OR, **A XOR B =** $A \oplus B$
1	**0**	**1**	**X**	**1**	**A+B**	NOR, **A NOR B =** $\overline{A + B}$

Equations :

$$y = B + S2\,\overline{S1}\,S0\,(A+B)$$

$$= B + S2\,\overline{S1}\,S0\,A + S2\,\overline{S1}\,S0\,B$$

$$= B\,(1 + S2\,\overline{S1}\,S0) + S2\,\overline{S1}\,S0\,A$$

$$= B + S2\,\overline{S1}\,S0\,A$$

$\mathbf{x} = \overline{S1}\,\overline{S0}\,A + \overline{S1}\,S0$

$\mathbf{z} = \overline{S2}\,Cin$

Total number of gates :

AND gate : 14

OR gate : 10

NOT gate : 4

:. Total number of 28 gates we used.

Circuit diagram :