

# MERU UNIVERSITY OF SCIENCE AND TECHNOLOGY



## DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

### BACHELOR OF TECHNOLOGY IN ELECTRICAL AND ELECTRONICS ENGINEERING

**TITLE: LAB REPORT**

**UNIT CODE: 3404**

**Description: Digital Circuit Design**

#### **Group members**

1. *Joshua Wambua EG209/109705/22*
2. *Nason Alfred EG209/111030/22*
3. *Abuto Tonny Hillary -EG209/106792/21.*
4. *Collins Kimathi --EG209/109696/22.*
5. *Wilkister Kwamboka -EG209/109739/22*

***INSTRUCTOR: DR. JOSEPH MUGURO***

***DATE:05/12/2025***

## Table of contents

1. Introduction .....	3
2. Objectives.....	3
3. Software and Components Used .....	3
3.1 Software Used .....	3
3.2 Components Used.....	3
4. Procedure.....	4
5. Results .....	5
5.1 Circuit Implementation .....	5
<i>Figure 1. The general circuit.</i> .....	5
<i>Figure 2. The general circuit with the logic analyzer.</i> .....	6
5.2 Truth Table Simulation Results.....	6
<i>Figure 3. The addition circuit. (<math>S_0,S_1=0,0</math>).</i> .....	7
<i>Figure 4. The subtraction circuit. (<math>S_0,S_1=1,0</math>).</i> .....	8
<i>Figure 5. The ORING circuit. (<math>S_0,S_1=1,1</math>).</i> .....	8
<i>Figure 6. The ANDING circuit. (<math>S_0,S_1=0,1</math>).</i> .....	9
5.3 Timing Waveform Analysis .....	9
<i>Figure 7. The arithmetic circuit logic analyzer. (<math>S_0,S_1=0,0</math>).</i> .....	10
<i>Figure 8. The arithmetic circuit logic analyzer. (<math>S_0,S_1=1,1</math>).</i> .....	10
6. Discussion.....	11
(i) Arithmetic Operation.....	11
(ii) Logic Operation.....	11
(iii) Operation Selection .....	11
(iv) Timing Behavior .....	11
7. Conclusion.....	11
8. References.....	12

## 1. Introduction

An Arithmetic Logic Unit (ALU) is a fundamental building block of digital systems and microprocessors. It is responsible for performing arithmetic operations such as addition and subtraction, as well as logical operations such as AND and OR. Understanding the internal construction and operation of an ALU provides insight into how processors execute instructions at the hardware level. In this experiment, a 4-bit ALU was designed and simulated using Multisim. The ALU was constructed using basic digital components including full adders, logic gates, and multiplexers. Different operations were selected using control signals, and the corresponding outputs were observed through simulation. Timing behavior and propagation delays were also analyzed using a logic analyzer.

## 2. Objectives

The objectives of this experiment were to:

1. Design and construct a 4-bit Arithmetic Logic Unit using Multisim.
2. Implement arithmetic operations (addition and subtraction) using full adders.
3. Implement logical operations (AND and OR) using basic logic gates.
4. Use multiplexers to select the desired ALU operation based on control inputs.
5. Simulate various input combinations and observe output behavior.
6. Analyze timing waveforms and observe propagation delays, especially ripple-carry delay.
7. Record and interpret simulation results in the form of truth tables and waveforms.

## 3. Software and Components Used

### 3.1 Software Used

- ❖ **Multisim** – for schematic design, simulation, and waveform analysis.

### 3.2 Components Used

1. Full Adders ( $\times 2$ )
2. XOR Gates
3. AND Gates
4. OR Gates
5. 4-to-1 Multiplexers (one per output bit)
6. Toggle switches/Interactive\_Digital\_Constant (for inputs A, B, and select lines S1, S0)
7. Logic Probes (for output display)
8. Logic Analyzer / Oscilloscope
9. Power supply (Vcc) and Ground

#### 4. Procedure

1. Multisim was launched and a new schematic file was created.
2. Four full adders were connected in cascade to form a 4-bit ripple-carry adder.
3. For subtraction, each bit of input B was inverted using NOT gates, and a logic '1' was applied to the initial carry-in to generate the 2's complement of B.
4. AND and OR gates were connected to form the logic unit for bitwise operations.
5. A 4-to-1 multiplexer was used for each output bit to select between:
  - ✓ Addition result
  - ✓ Subtraction result
  - ✓ AND result
  - ✓ OR result
6. The multiplexer select lines were connected to control inputs S1 and S0.
7. Toggle switches were connected to inputs A3–A0 and B3–B0.
8. LEDs or logic probes were connected to outputs F3–F0 and Cout.
9. A logic analyzer was connected to selected inputs and outputs to observe timing waveforms.
10. Multiple test cases were simulated, and outputs were recorded.

## 5. Results

### 5.1 Circuit Implementation

The complete 4-bit ALU circuit was successfully constructed in Multisim using basic digital components.

	A – B	4-bit subtractor (A + 2's complement of B)

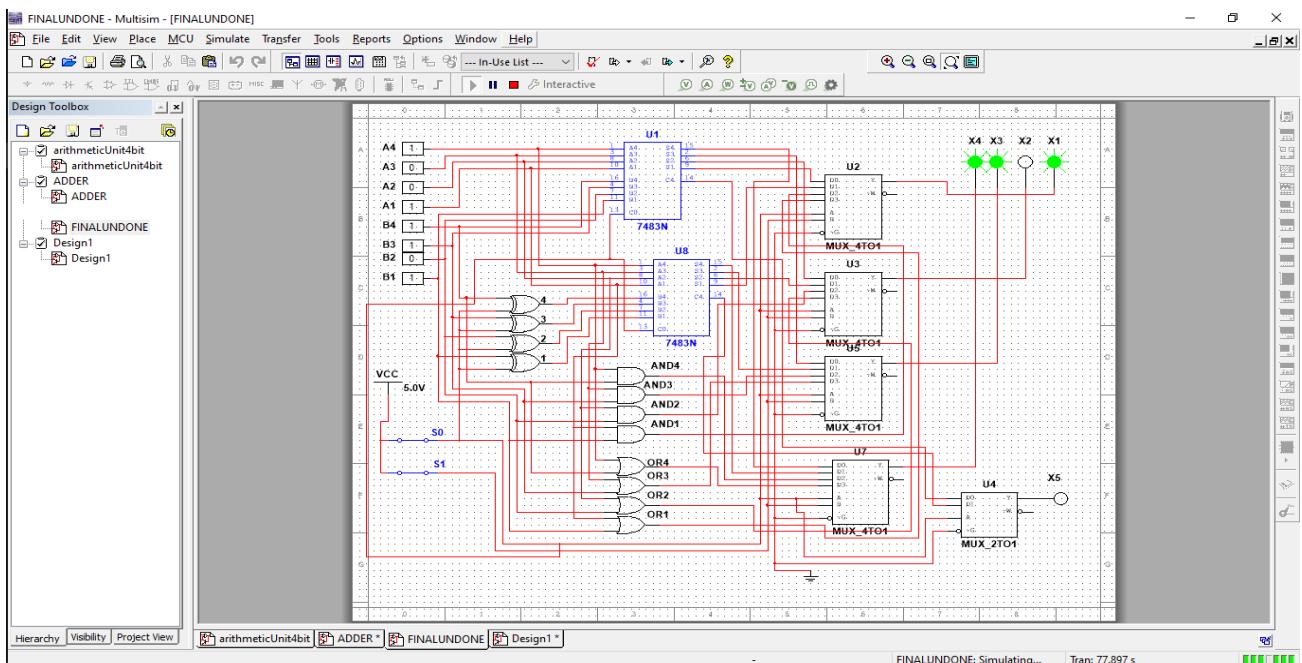
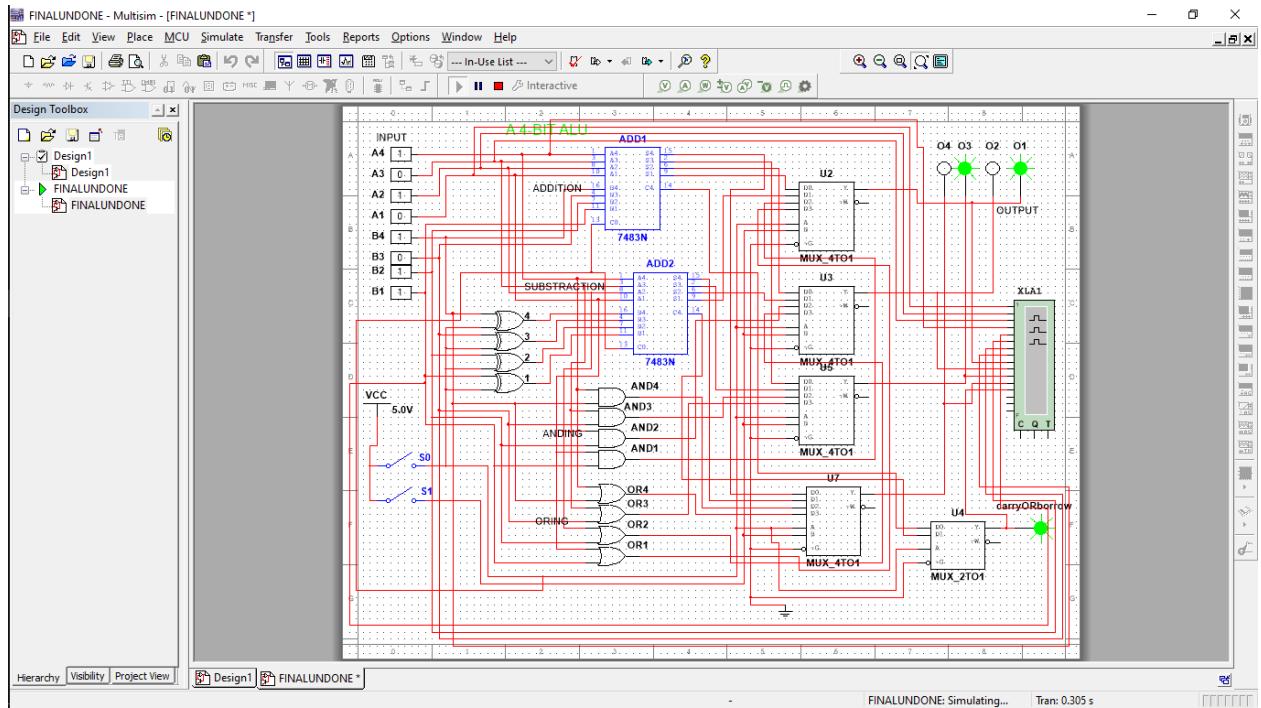


Figure 1. The general circuit.



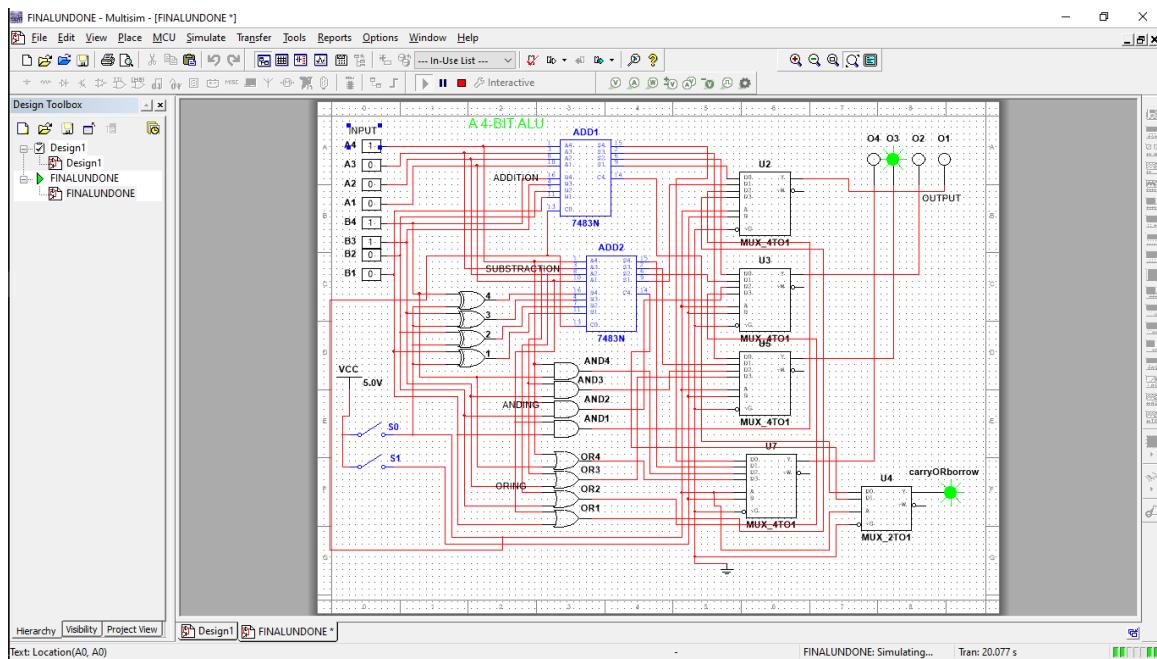
*Figure 2. The general circuit with the logic analyzer.*

## 5.2 Truth Table Simulation Results

The ALU was tested using various combinations of A, B, and select inputs. The outputs observed on the LEDs and logic probes matched the expected results for each operation.

### A. ADDITION TRUTH TABLE

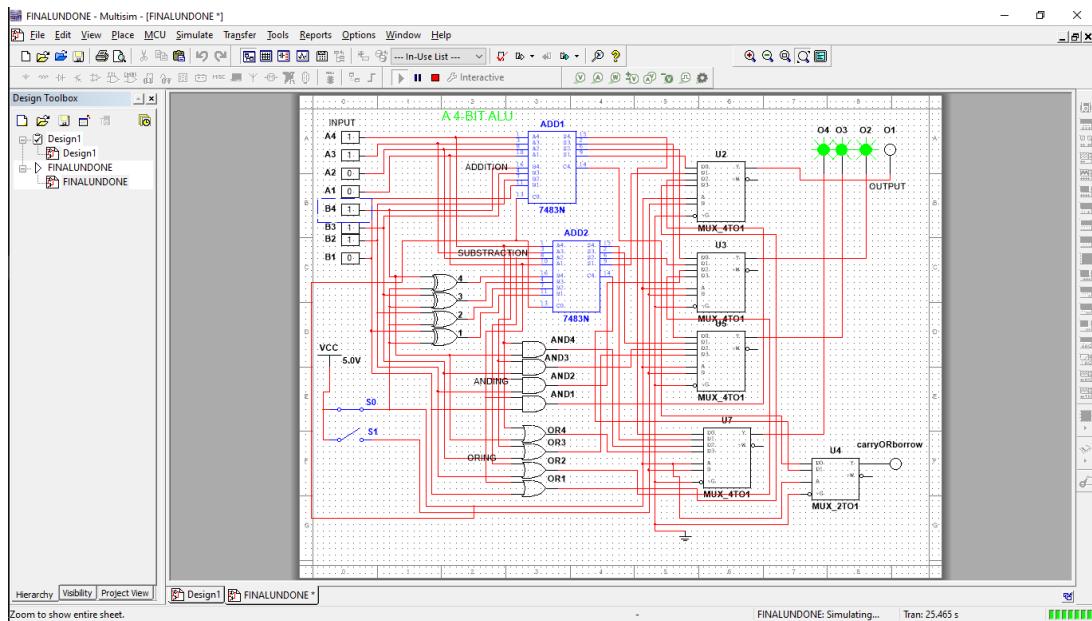
A3	A2	A1	A0	B3	B2	B1	B0	O3	O2	O1	O0	C_out
0	1	0	0	1	0	0	0	1	1	0	0	0
1	0	0	0	1	1	0	0	0	1	0	0	1



*Figure 3. The addition circuit. ( $S_0, S_1 = 0, 0$ ).*

### B. SUBTRACTION (2'S COMPLEMENT) TRUTH TABLE

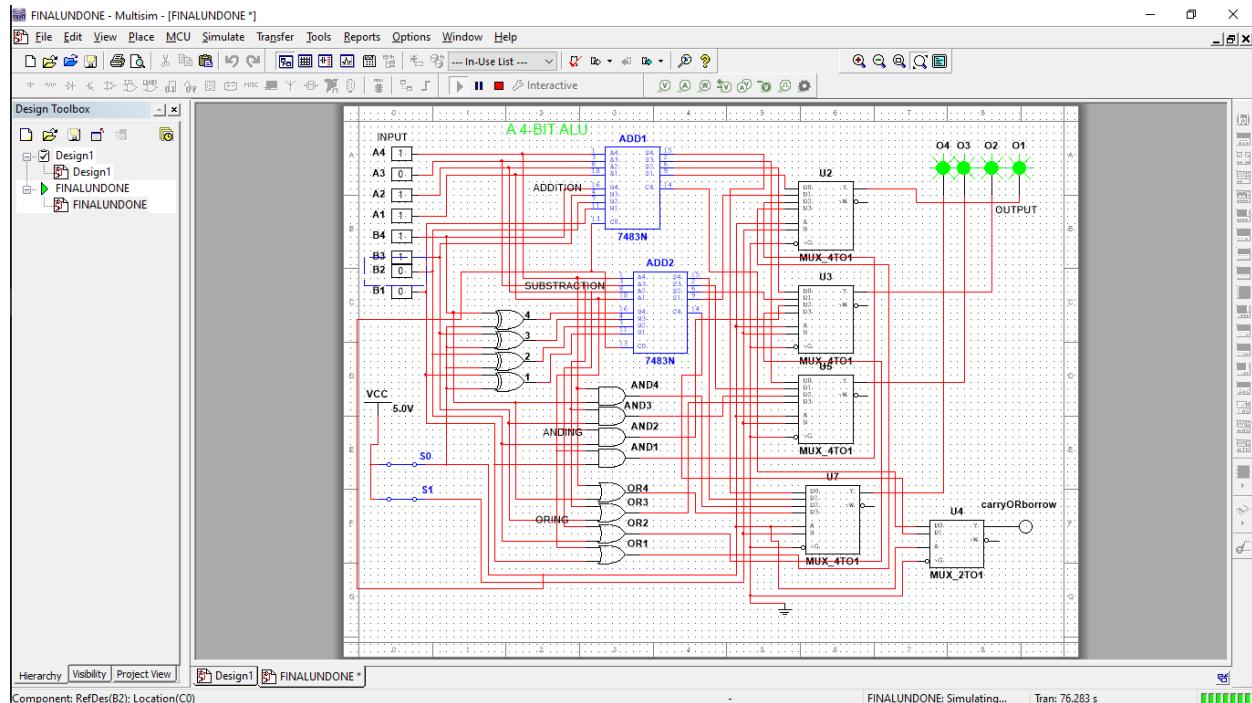
A3	A2	A1	A0	B3	B2	B1	B0	O3	O2	O1	O0	BORROW
0	1	0	1	0	0	0	0	0	1	0	1	1
1	1	0	0	1	1	1	0	1	1	1	0	0



*Figure 4. The subtraction circuit. ( $S_0, S_1 = 1, 0$ ).*

### C. ORING TRUTH TABLE

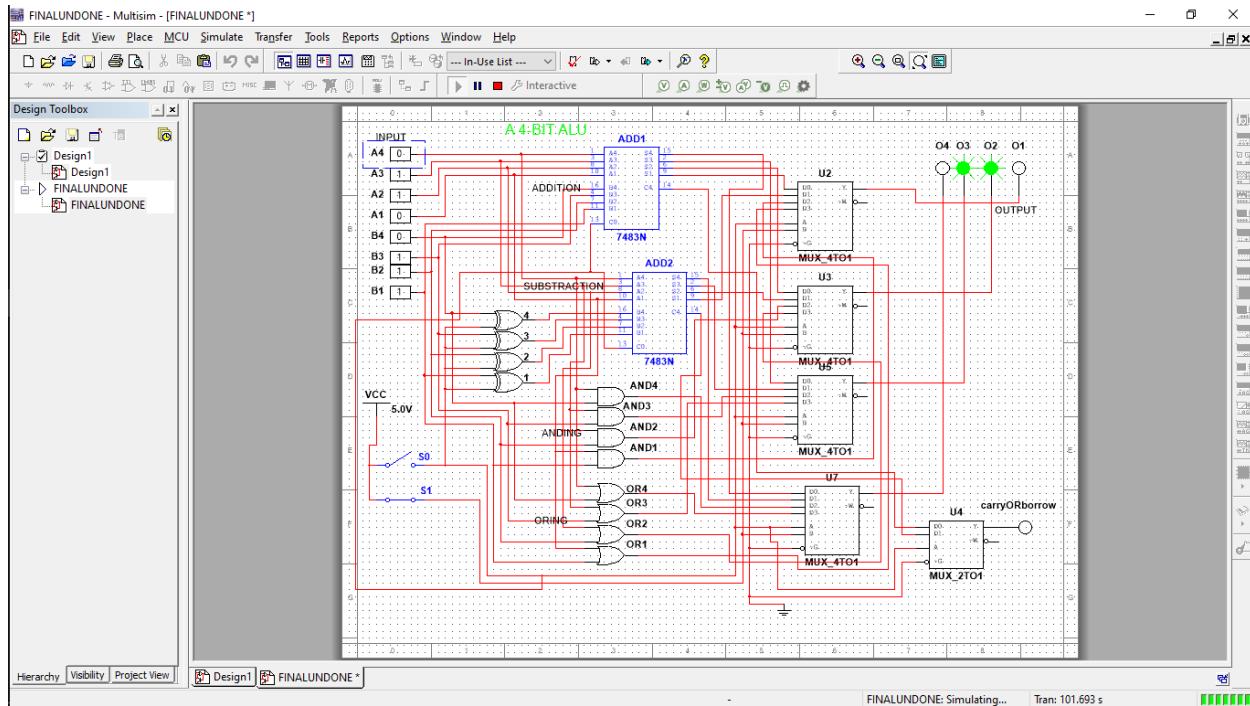
A3	A2	A1	A0	B3	B2	B1	B0	O3	O2	O1	O0
0	1	0	0	1	0	0	0	1	1	0	0
1	0	1	1	1	1	0	0	1	1	1	1



*Figure 5. The ORING circuit. ( $S_0, S_1 = 1, 1$ ).*

### D. ANDING TRUTH TABLE

A3	A2	A1	A0	B3	B2	B1	B0	O3	O2	O1	O0
0	1	1	0	0	1	1	1	0	1	1	0
1	1	1	1	1	1	1	1	1	1	1	1

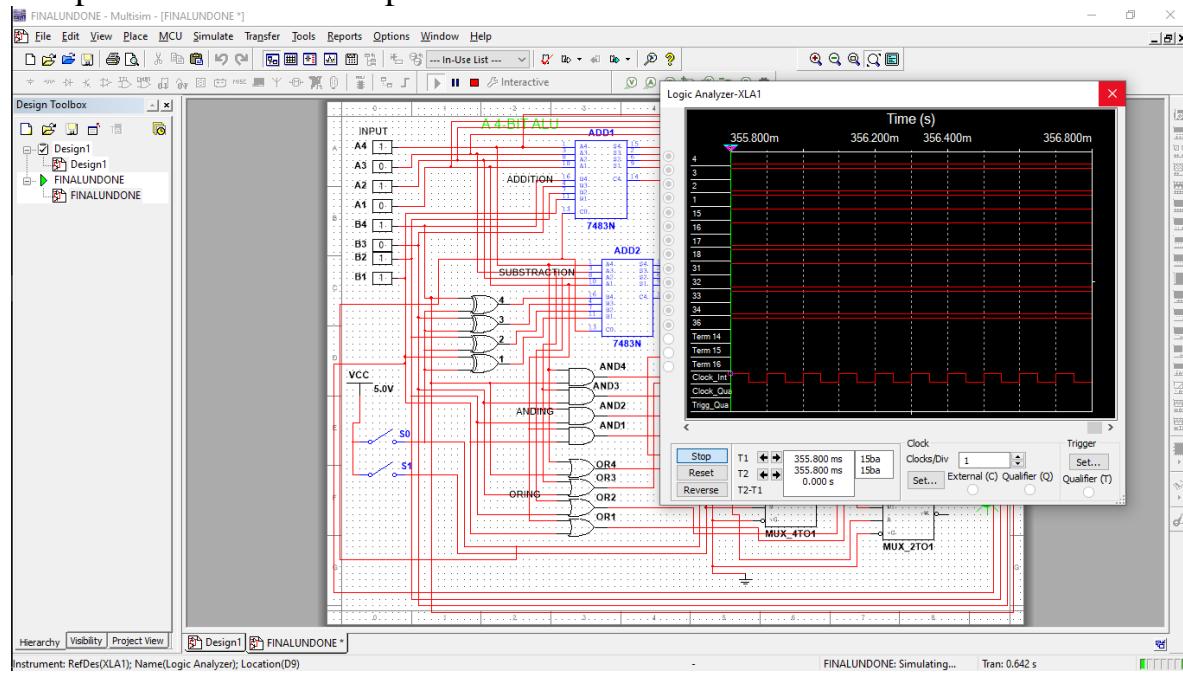


*Figure 6. The ANDING circuit. ( $S_0, S_1=0,1$ ).*

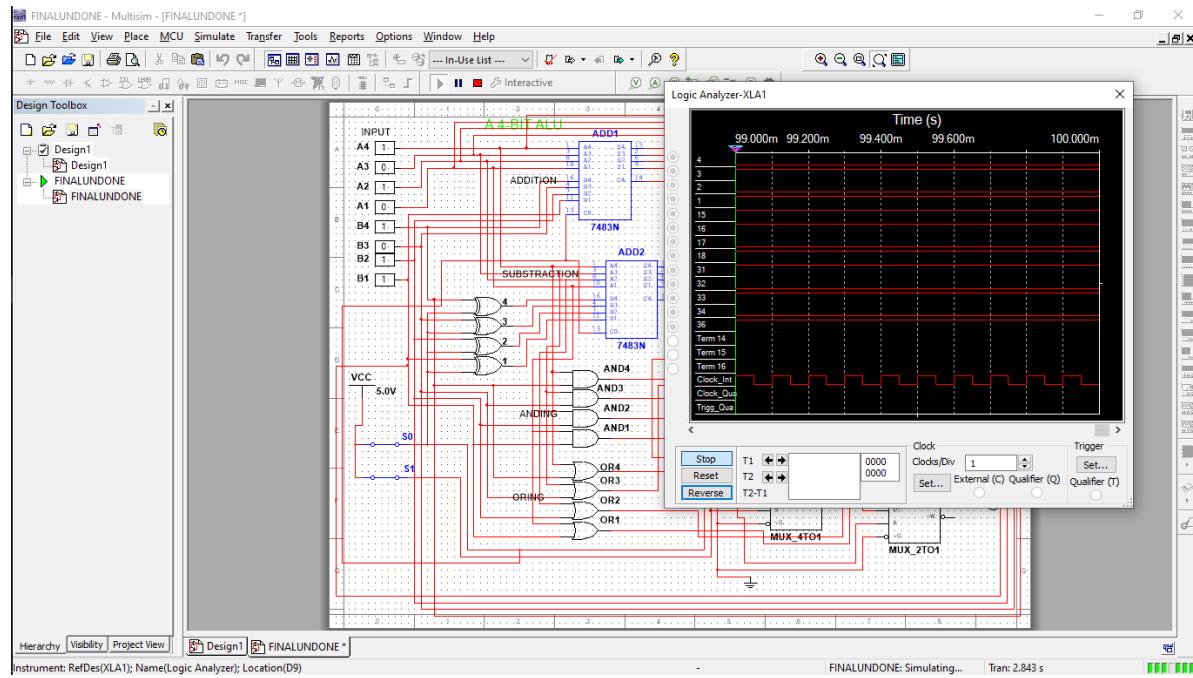
### 5.3 Timing Waveform Analysis

Timing waveforms were captured using the logic analyzer. During addition and subtraction operations, a noticeable ripple-carry delay was observed as the carry propagated through successive full adders. Logic operations showed minimal delay

compared to arithmetic operations.



*Figure 7. The arithmetic circuit logic analyzer. ( $S_0, S_1 = 0, 0$ ).*



*Figure 8. The arithmetic circuit logic analyzer. ( $S_0, S_1 = 1, 1$ ).*

## 6. Discussion

### (i) Arithmetic Operation

The ALU performs addition by using a single 4-bit adder, which internally handles the carry propagation across all four bits. For subtraction, the 2's complement of input B is generated by inverting B and applying a logic '1' to the carry-in of the 4-bit adder. This allows subtraction ( $A - B$ ) to be performed using the same adder hardware, ensuring efficient utilization of the arithmetic unit.

### (ii) Logic Operation

Logical AND and OR operations are implemented using dedicated logic gates for each corresponding bit of inputs A and B. Each bit is processed independently, producing bitwise logical results without carry interaction. Because these operations do not involve arithmetic carry propagation, the outputs respond faster compared to arithmetic operations.

### (iii) Operation Selection

The outputs from the arithmetic and logic units are connected to 4-to-1 multiplexers. The select inputs S1 and S0 control the multiplexers to determine which operation result (addition, subtraction, AND, or OR) is routed to the final output lines F3–F0. This approach allows seamless switching between operations using control signals.

### (iv) Timing Behavior

During addition and subtraction, a ripple-carry delay is observed due to internal carry propagation within the 4-bit adder. Logic operations exhibit near-instantaneous output changes since no carry chain is involved. Minor propagation delays are also visible due to gate and multiplexer delays, as observed in the timing waveform.

## 7. Conclusion

In this experiment, a functional 4-bit Arithmetic Logic Unit was successfully designed and simulated using Multisim. The ALU correctly performed addition, subtraction, AND, and OR operations based on control inputs. Simulation results matched expected theoretical values, and timing analysis revealed propagation delays inherent in ripple-carry adders. This experiment enhanced understanding of digital arithmetic, logic design, and ALU operation in modern computing systems.

## 8. References

1. M. Morris Mano and Michael D. Ciletti, *Digital Design: With an Introduction to the Verilog HDL*, Pearson Education.
2. Thomas L. Floyd, *Digital Fundamentals*, Pearson Education.
3. Multisim User Manual, National Instruments.
4. Lecture Notes – Digital Logic Design, Department of Electrical and Electronics Engineering.