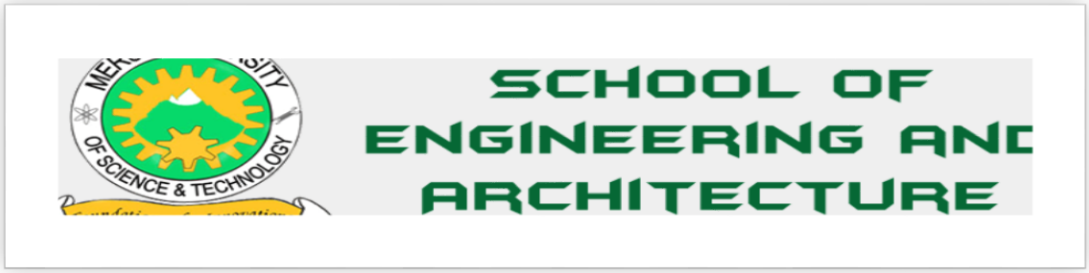
MERU UNIVERSITY OF SCIENCE AND TECHNOLOGY



DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

BACHELOR OF TECHNOLOGY IN ELECTRICTRAL AND ELECTRONIS ENGINEERING

TITLE: LAB REPORT

UNIT CODE: 3404

Description: Digital Circuit Design

https://github.com/joshuawambua/digitalCircuitDesign.git

Group members

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

INSTRUCTOR: DR. JOSEPH MUGURO

DATE:05/12/2025

Table of contents

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. Finally, a hardware implementation was done using an ALU IC SN74LS181N.

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.
8. To build the circuit on a bread board and perform simple addition.

3. Software and Components Used

3.1 Software Used

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

3.2 Components Used

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

3.3 Hardware used;

1. ALU IC 74LS181N
2. JUMBERS
3. LEDS
4. BREADBOARD
5. POWER SUPPLY UNIT

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.
11. Finally, the circuit of the ALU IC addition was done on a breadboard and basic addition done.

5. Results

5.1 Circuit Implementation

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

S1 S0	Operation	Description
00	A + B	4-bit addition (ripple-carry adder)
01	A – B	4-bit subtraction (A + 2's complement of B)
10	A AND B	Bitwise AND
11	A OR B	Bitwise OR

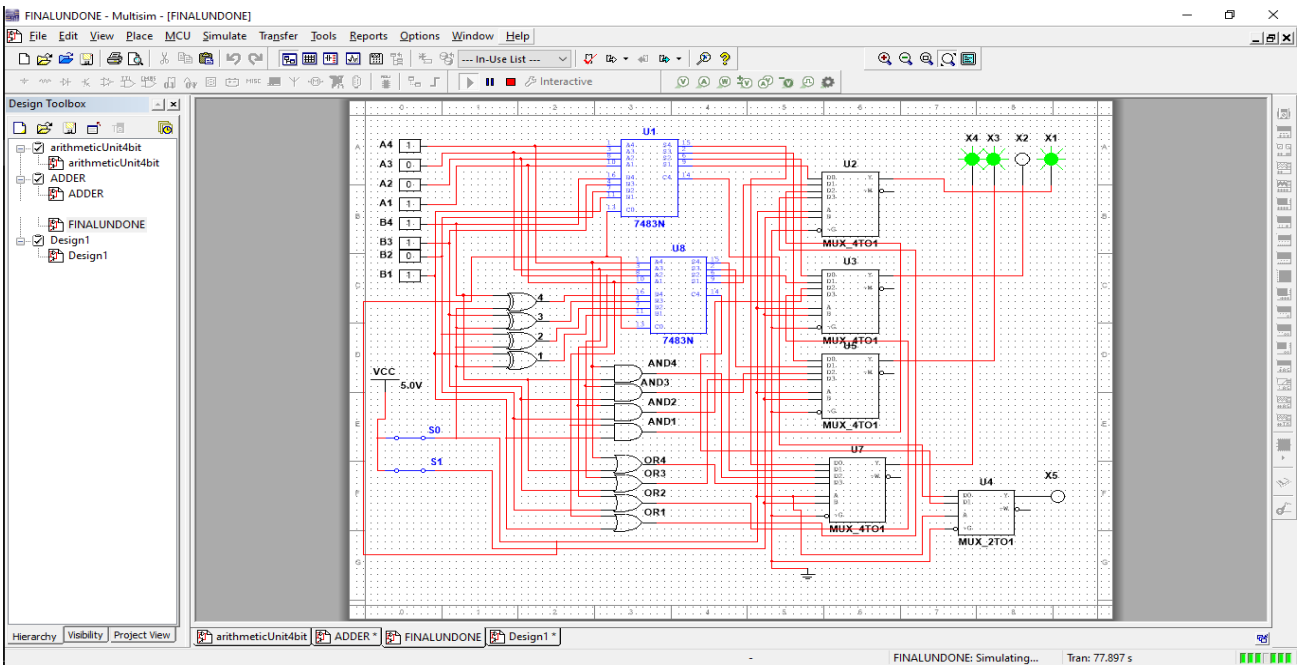


Figure 1. The general circuit.

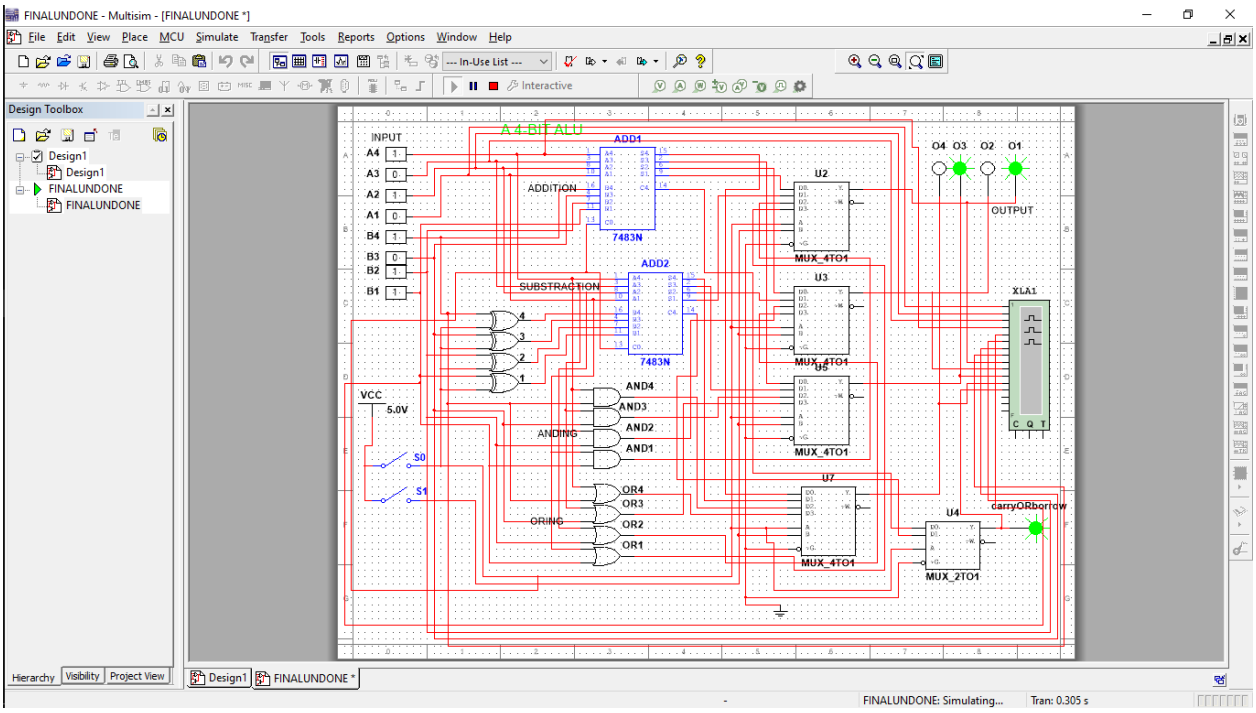


Figure 2. The general circuit with the logic analyzer.

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

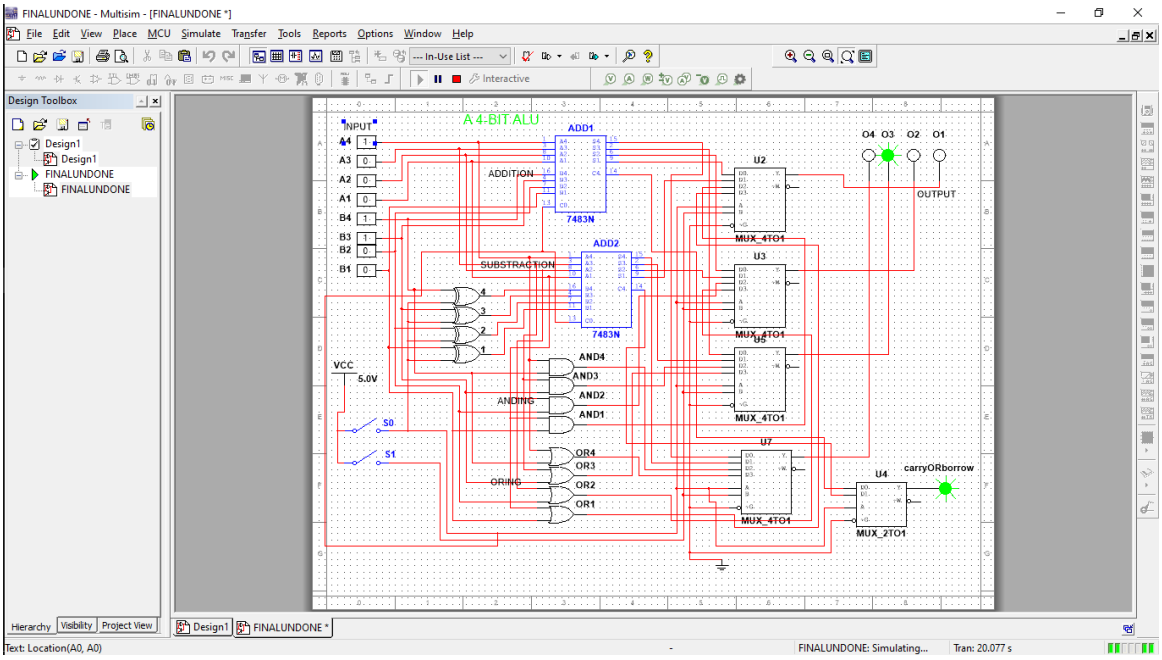


Figure 3. The addition circuit. (S0,S1=0,0).

B. *SUBSTRACTION (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

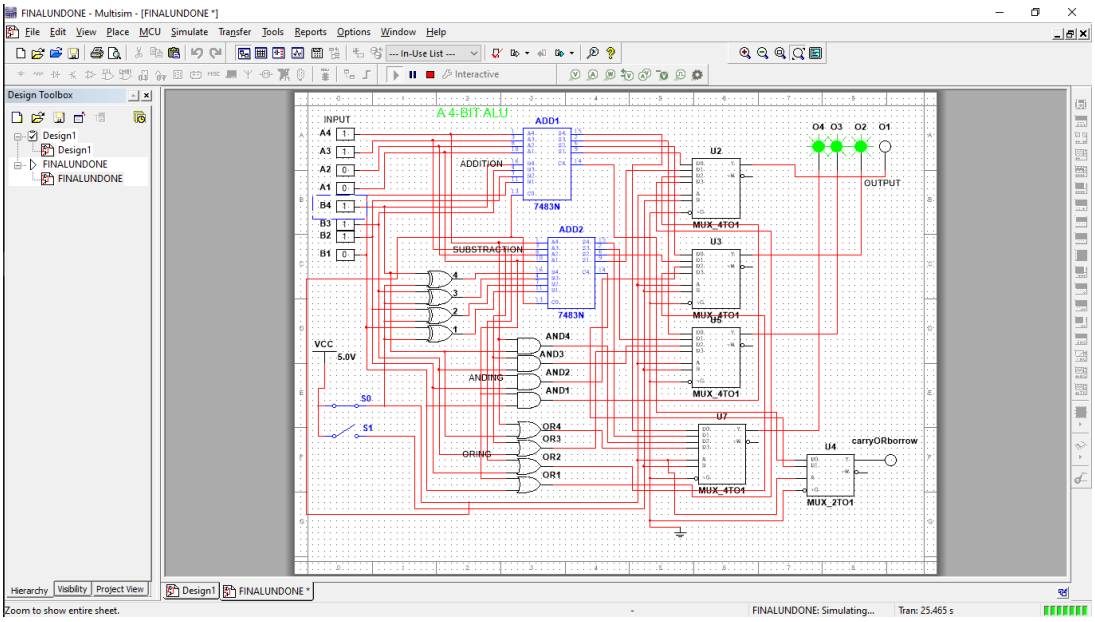


Figure 4. The substraction circuit. (S0,S1=1,0).

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

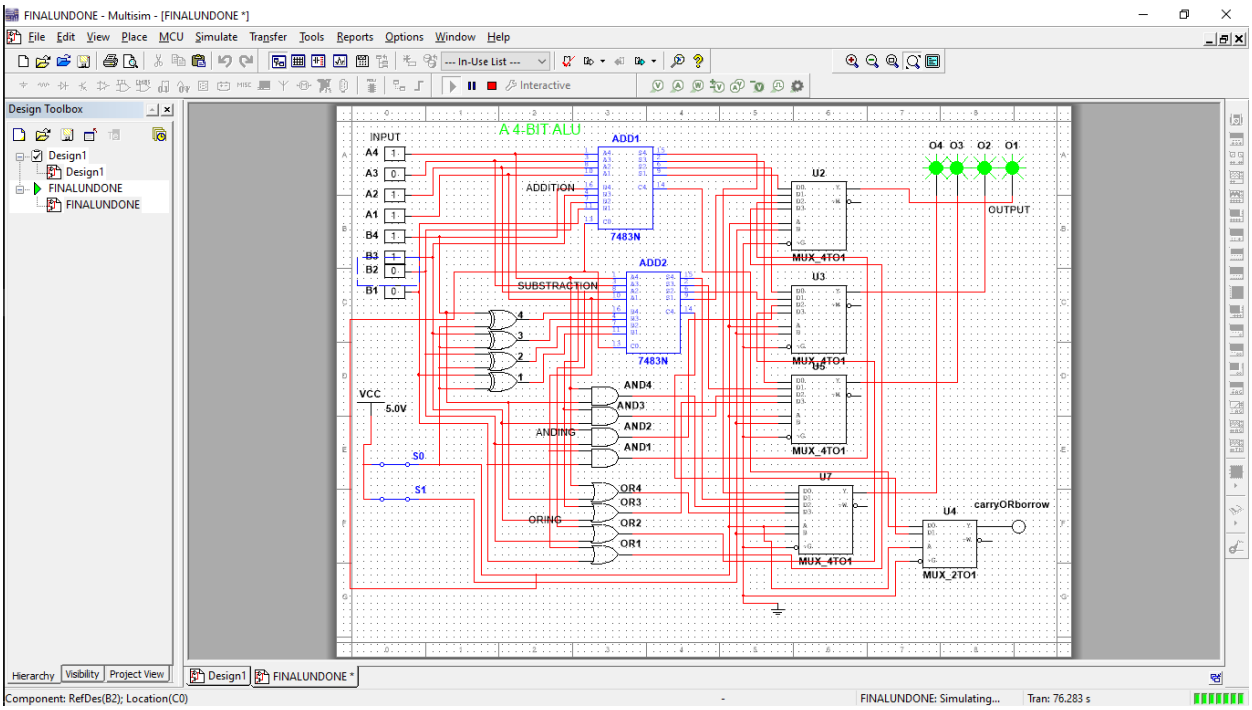


Figure 5. The ORING circuit. (S0,S1=1,1).

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

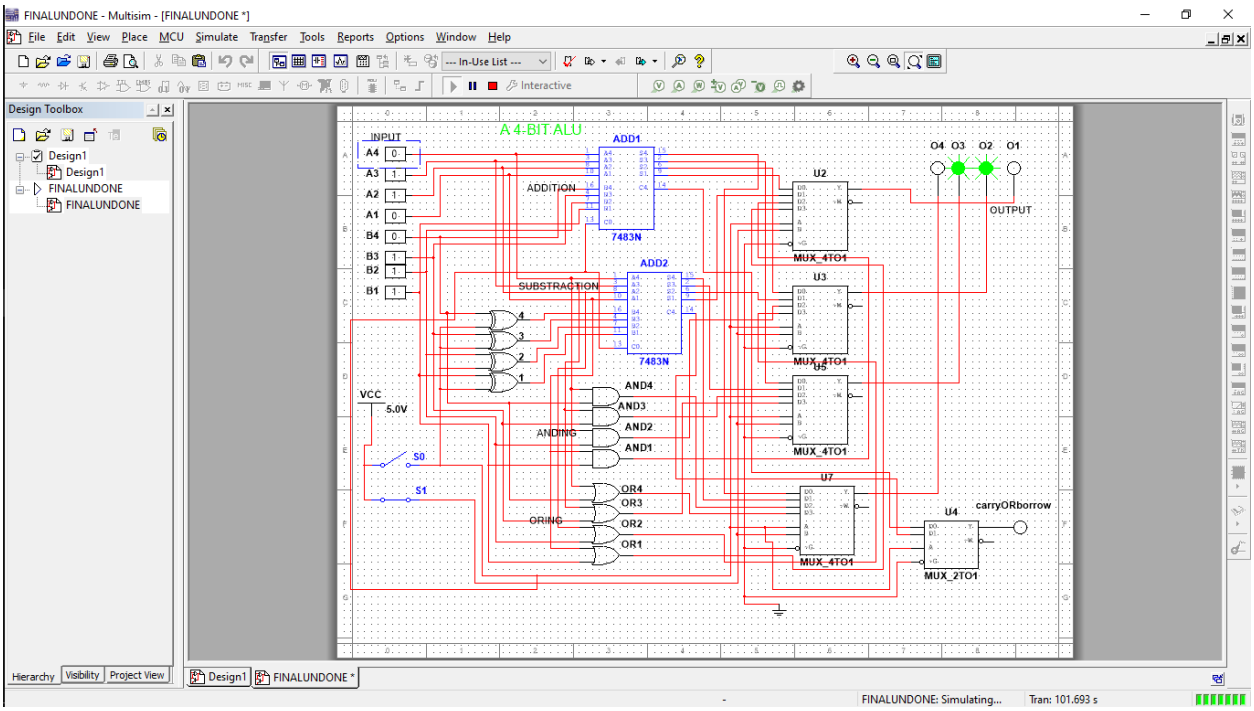


Figure 6. The ANDING circuit. (S0,S1=0,1).

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.

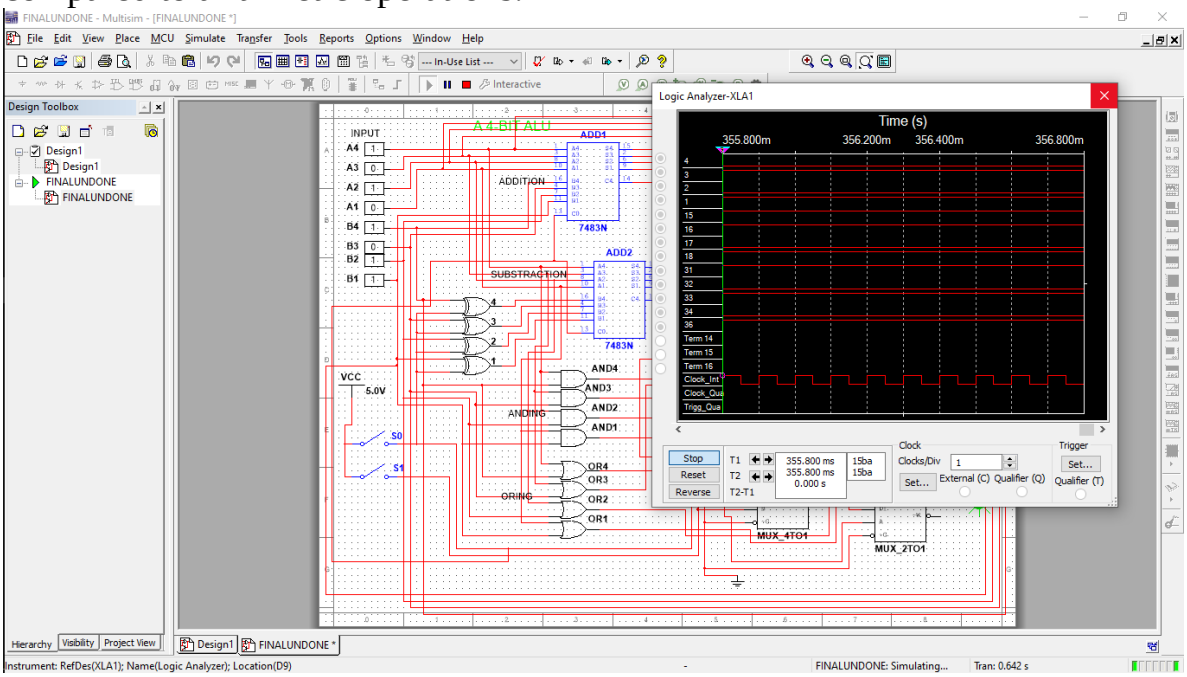


Figure 7. The arithmetic circuit logic analyzer. (S0,S1=0,0).

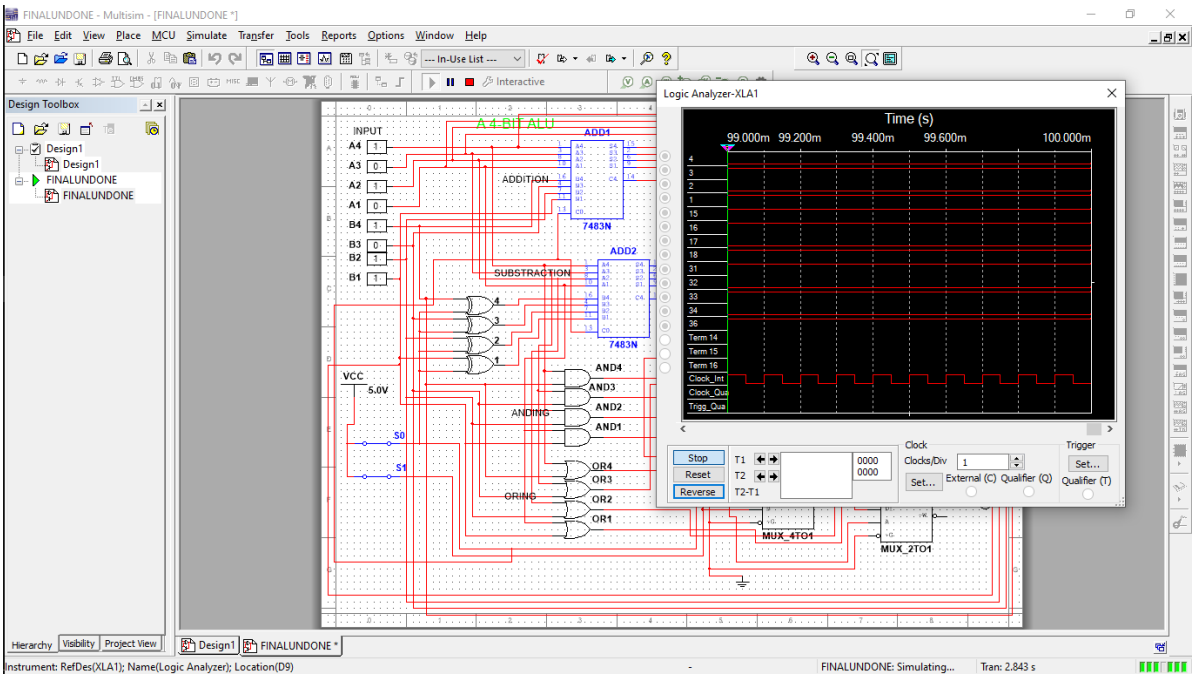


Figure 8. The arithmetic circuit logic analyzer. (S0,S1=1,1).

5.4: The ALU IC Implementation (SN74LS181N).

The basic addition was done as shown in the figure below.

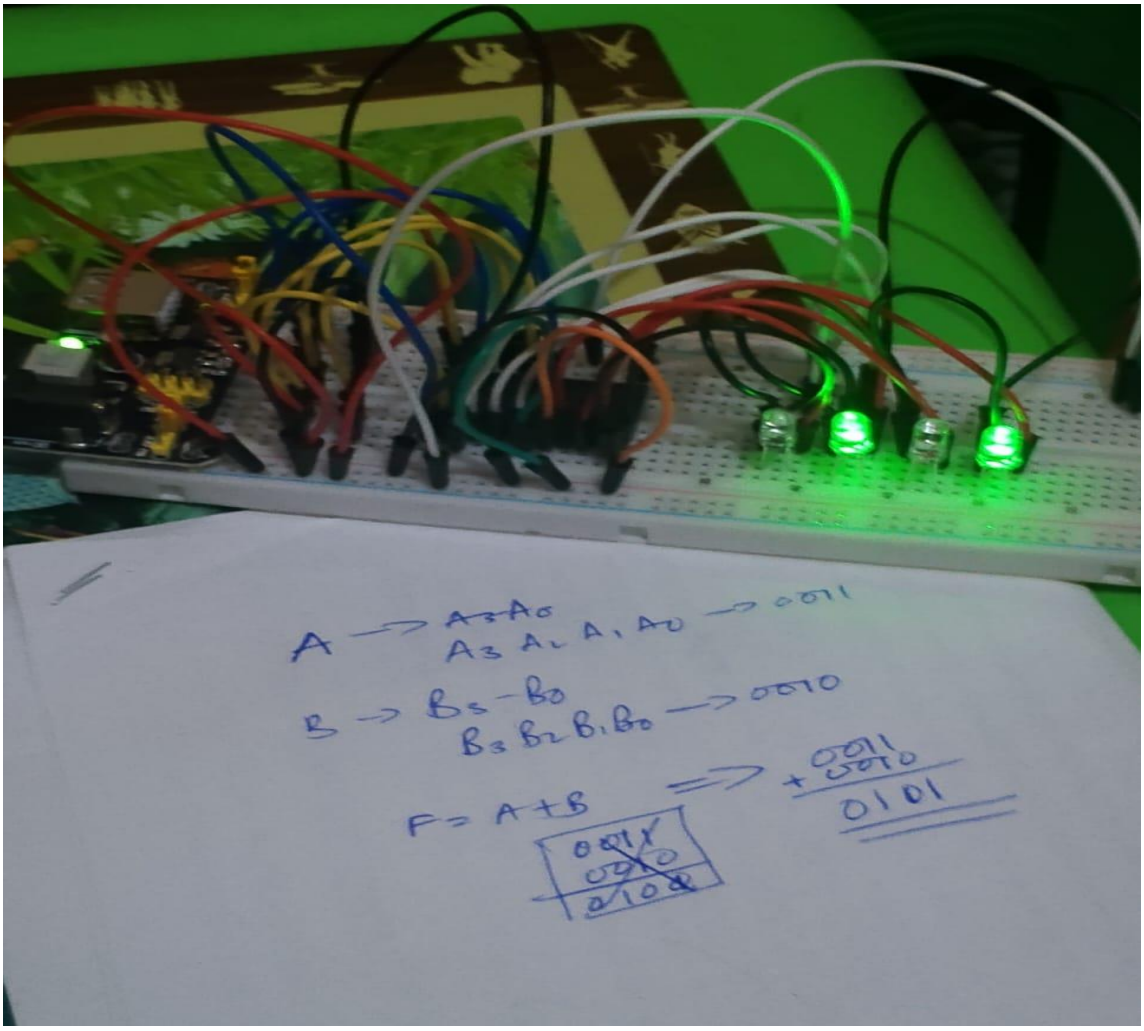


Figure 9: An ALU performing normal addition on a breadboard at a HOMELAB

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. The hardware implementation of the same was done using IC SN74LS181N. This experiment enhanced understanding of digital arithmetic, logic design, and ALU operation in modern computing systems.

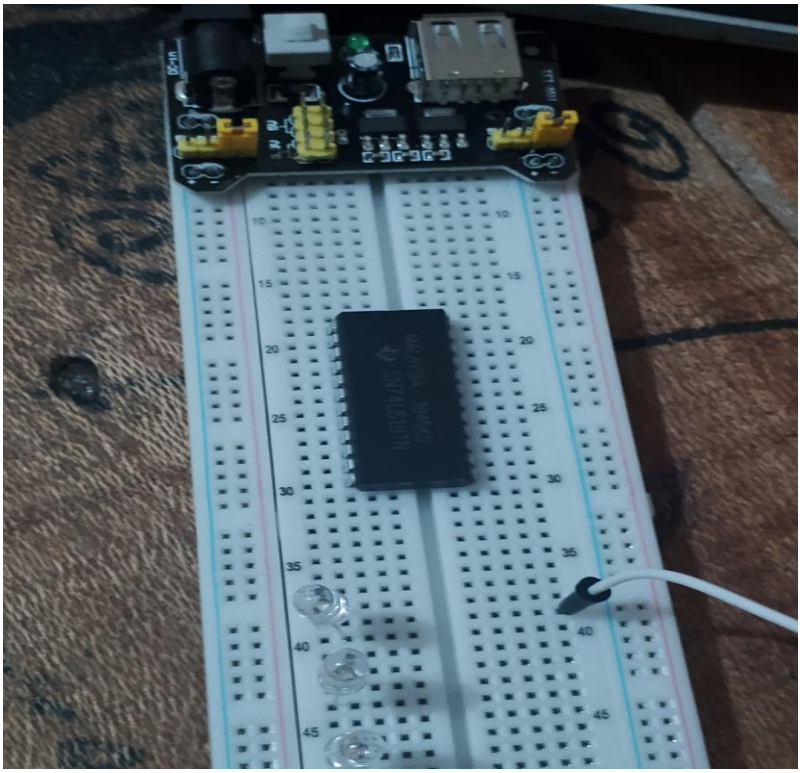
8. References

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.

9. Appendices

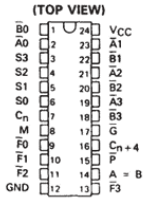
Appendix 1: Pin configuration of the ALU IC used.



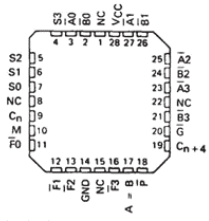
SN54LS181, SN54S181
SN74LS181, SN74S181
ARITHMETIC LOGIC UNITS/FUNCTION GENERATORS
SDLS136 – DECEMBER 1972 – REVISED MARCH 1988

- Full Look-Ahead for High-Speed Operations on Long Words
- Input Clamping Diodes Minimize Transmission-Line Effects
- Darlington Outputs Reduce Turn-Off Time
- Arithmetic Operating Modes:
  - Addition
  - Subtraction
  - Shift Operand A One Position
  - Magnitude Comparison
  - Plus Twelve Other Arithmetic Operations
- Logic Function Modes:
  - Exclusive-OR
  - Comparator
  - AND, NAND, OR, NOR
  - Plus Ten Other Logic Operations

SN54LS181, SN54S181 . . . J OR W PACKAGE
SN74LS181, SN74S181 . . . DW OR N PACKAGE
(TOP VIEW)

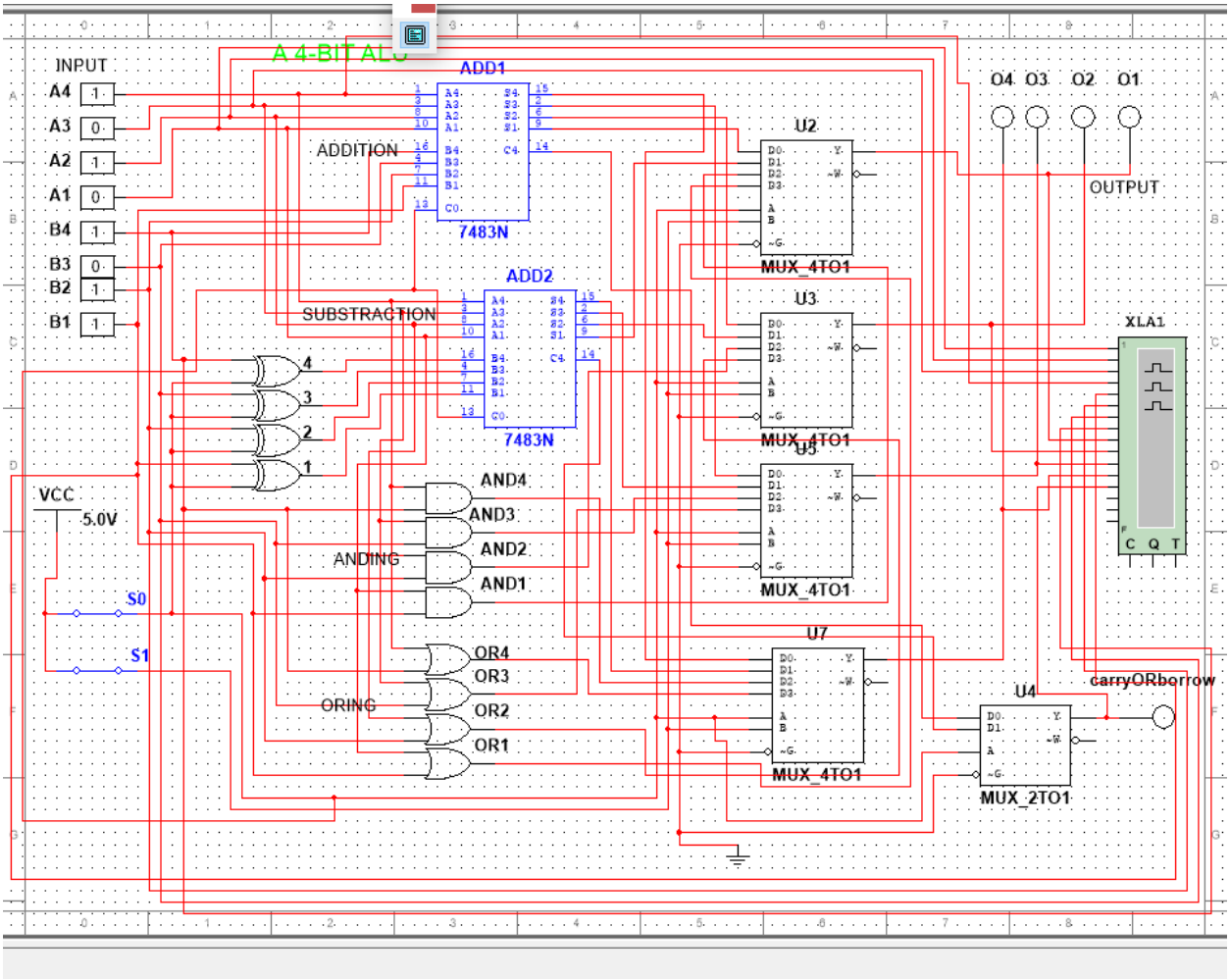


SN54LS181, SN54S181 . . . FK PACKAGE
(TOP VIEW)



NC - No internal connection

Appendix 2: Complete Circuit On Multism



Appendix 3: GitHub repository for more resources of the report.

https://github.com/joshuawambua/digitalCircuitDesign.git