

ALU Project



Under the supervision of Prof.Ahmed Shalaby

Team Members:

Eyad Ahmed Sabry 23102371

Maryam Ali Elhabal 23102387

Youssef ahmed saad 23101507

Menna Allah Osama Khalil Khalil Ibrahim 23102157

Introduction

An arithmetic logic unit (ALU) is a digital circuit used to perform arithmetic and logic operations. It represents the fundamental building block of the central processing unit (CPU) of a computer.

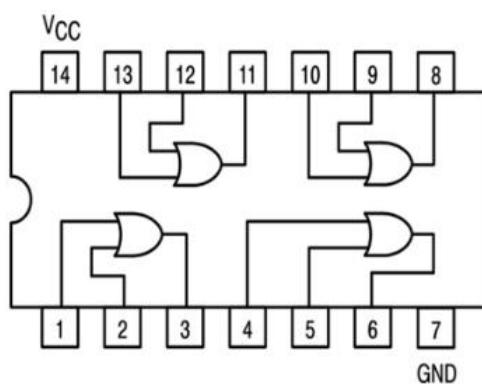
Objective Of The Project:

The primary objective of this project is to understand the design and functionality of an Arithmetic Logic Unit (ALU) by implementing its operations on a breadboard using basic logic gate ICs and simulating the circuit in Logisim software. This hands-on approach aims to bridge theoretical concepts with practical application, enhancing knowledge of digital circuits and logic operations while developing problem-solving and circuit design skills.

Materials and Tools :

1) Logic Unit:

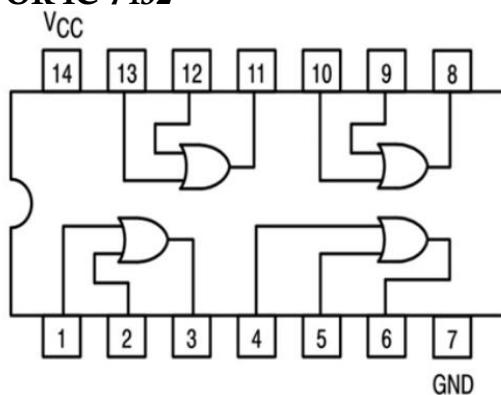
-AND IC 7408 - EQ (AND IC 7408)



AND Truth Table

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

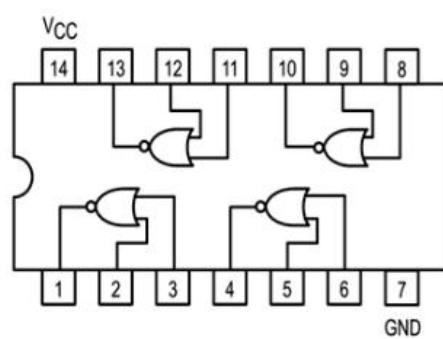
-OR IC 7432



OR Truth Table

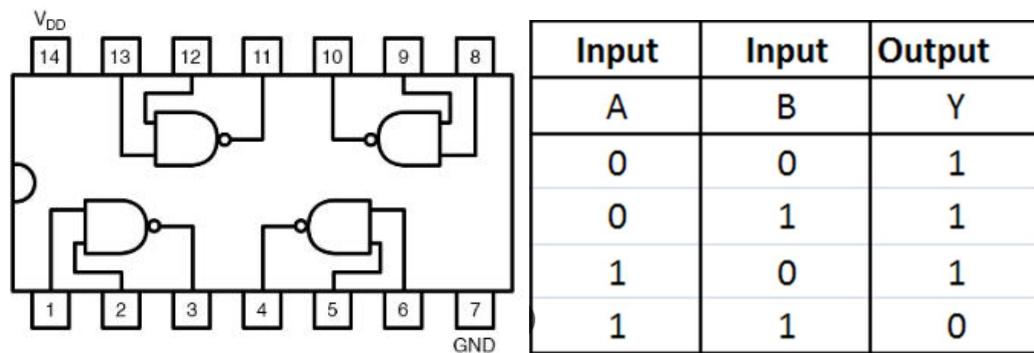
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

-NOR IC 7402

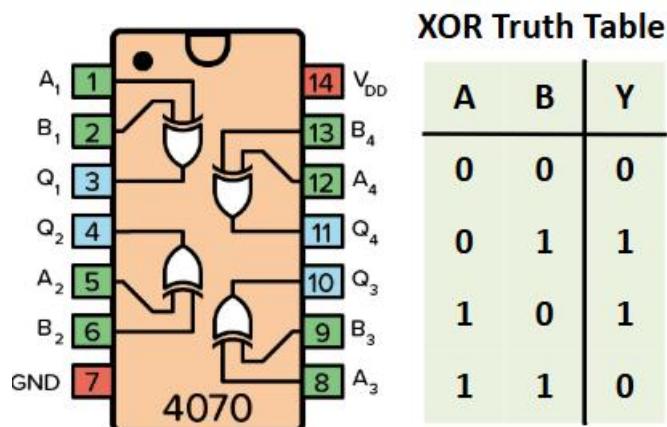


A	B	Output
0	0	1
1	0	0
0	1	0
1	1	0

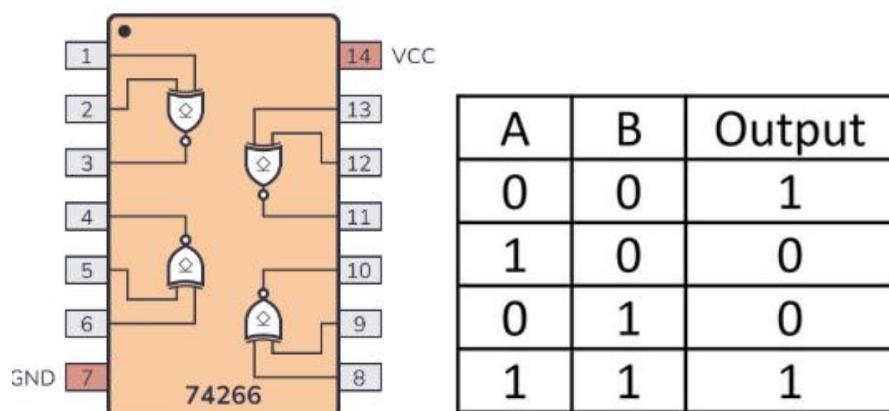
-NAND IC 7400



-XOR IC 4070

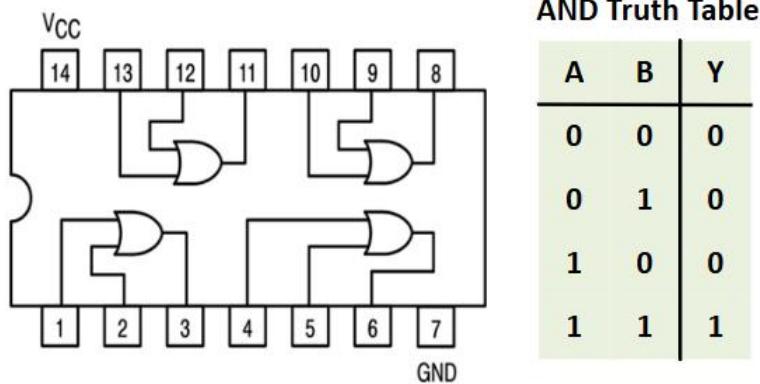


-XNOR IC 74266

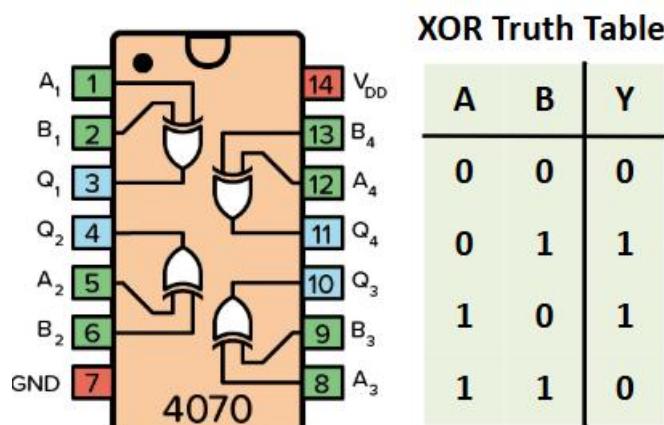


2) Arithmetic Unit:

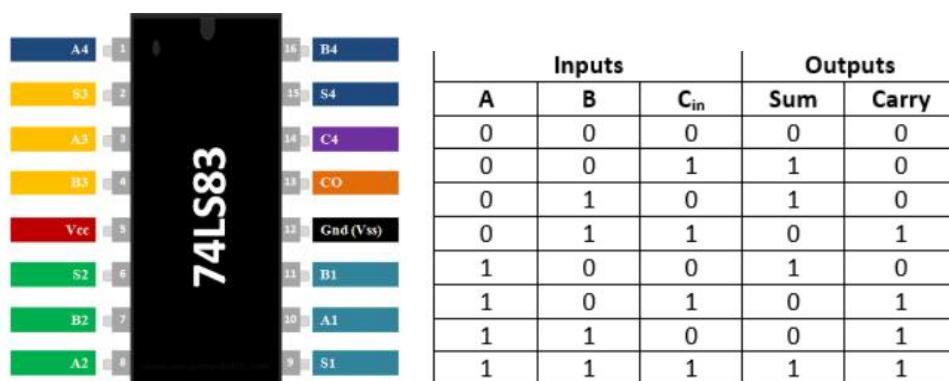
-EQ (AND IC 7408)



-XOR IC 4070

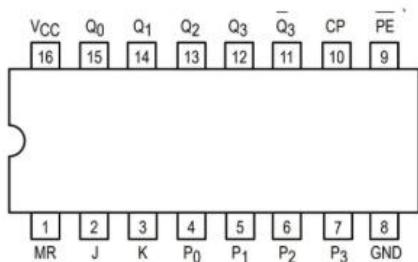


-Full Adder IC 74LS83

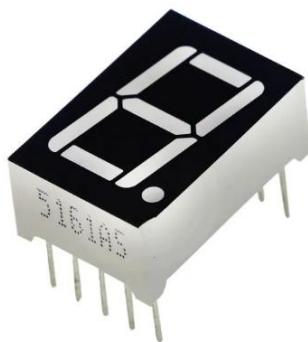


-5-Bit Storage IC 74LS195 (REGISTER)

74LS195 Pinout



-Hexa Digit Display



Inputs	Segments						
	a	b	c	d	e	f	g
0 0 0 0	1	1	1	1	1	1	0
0 0 0 1	0	1	1	0	0	0	0
0 0 1 0	1	1	0	1	1	0	1
0 0 1 1	1	1	1	1	0	0	1
0 1 0 0	0	1	1	0	0	1	1
0 1 0 1	1	0	1	1	0	1	1
0 1 1 0	1	0	1	1	1	1	1
0 1 1 1	1	1	1	0	0	0	0
1 0 0 0	1	1	1	1	1	1	1
1 0 0 1	1	1	1	1	0	1	1
1 0 1 0	1	1	1	0	1	1	1
1 0 1 1	0	0	1	1	1	1	1
1 1 0 0	1	0	0	1	1	1	0
1 1 0 1	0	1	1	1	1	0	1
1 1 1 0	1	0	0	1	1	1	1
1 1 1 1	1	0	0	0	1	1	1

For display 0
For display 1
For display 2
For display 3
For display 4
For display 5
For display 6
For display 7
For display 8
For display 9
For display A
For display b
For display C
For display d
For display E
For display F

-AND / OR / NOT ICs

3) Multiplication

- AND IC 7408
- Full Adder IC 74LS83

4) ALU

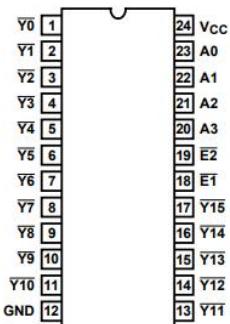
-Logic Unit

- AND IC 7408 - EQ (AND IC 7408)
- OR IC 7432
- NOR IC 7402
- NAND IC 7400
- XOR IC 4070
- XNOR IC 74266

-Arithmetic Unit

- EQ (AND IC 7408)**
- XOR IC 4070**
- Full Adder IC 74LS83**
- 5-Bit Storage IC 74LS195 (REGISTER)**
- Hexa Digit Display**

-DMUX IC CD74HC154



TRUTH TABLE

INPUTS						OUTPUTS															
E1	E2	A3	A2	A1	A0	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8	Y9	Y10	Y11	Y12	Y13	Y14	Y15
L	L	L	L	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	
L	L	L	L	L	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	
L	L	L	L	H	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	
L	L	L	L	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H	
L	L	L	H	L	L	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	
L	L	L	H	L	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	
L	L	L	H	H	L	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H	
L	L	L	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H	
L	L	L	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H	
L	L	L	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H	
L	L	L	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H	
L	L	H	L	L	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	
L	L	H	L	L	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	
L	L	H	L	H	L	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	
L	L	H	L	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	
L	L	H	H	L	L	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H	
L	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H	
L	L	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H	
L	L	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	L	H	H	H	
L	H	X	X	X	X	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	
H	L	X	X	X	X	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	
H	H	X	X	X	X	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	

H = High Voltage Level, L = Low Voltage Level, X = Don't Care

-Multiplication

- AND IC 7408**
- Full Adder IC 74LS83**

Theoretical Background:

The Arithmetic Logic Unit (ALU) is a fundamental component of a computer's central processing unit (CPU). It performs arithmetic operations (such as addition, subtraction, multiplication) and logical operations (such as AND, OR, NOT, NOR, NAND, XOR, XNOR) on binary data. Understanding its functionality is essential for grasping how computers process data at the hardware level.

Arithmetic Operations:

•Addition:

- Implemented using a 4-bit full-adder circuit to add two 4-bit binary numbers, producing a 4-bit result and a carry-out bit.

•Subtraction:

- Performed using the two's complement method, where the subtrahend is inverted, and 1 is added using the adder circuit.

•Multiplication:

- Partial products are generated using AND gates for corresponding bits and summed using multi-bit adders.

Logical Operations:

•AND, OR, NOT, NOR, NAND, XOR, XNOR:

- Each logic operation is performed bit-by-bit across the 4 bits of the input data.

Control Using a Demultiplexer (DMUX)

- A demultiplexer (DMUX) is used to control the operation of the ALU by directing the input signal to one of the operation circuits based on the control inputs.

- The control inputs determine which operation is activated.

For example:

NO	S0	S1	S2	S3	Operation	Function
0	0	0	0	0	A + B	Add A to B
1	0	0	0	1	A - B	Subtract B from A
2	0	0	1	0	A + 1	<u>A</u> Increment by 1
3	0	0	1	1	A - 1	A decrement by 1
4	0	1	0	0	A + B'	Add A to B'
5	0	1	0	1	A's 2's C	2's Complement of A
6	0	1	1	0	A <<	Shift A Left by 1 bit
7	0	1	1	1	Transfer to A	Store Output to A
8	1	0	0	0	A'	A Complement
9	1	0	0	1	B'	B Complement
10	1	0	1	0	A AND B	AND
11	1	0	1	1	A OR B	OR
12	1	1	0	0	A NAND B	NAND
13	1	1	0	1	A NOR B	NOR
14	1	1	1	0	A XOR B	XOR
15	1	1	1	1	A XNOR B	XNOR

The DMUX enables a single control signal to activate only the desired operation circuit, ensuring efficient use of hardware resources.

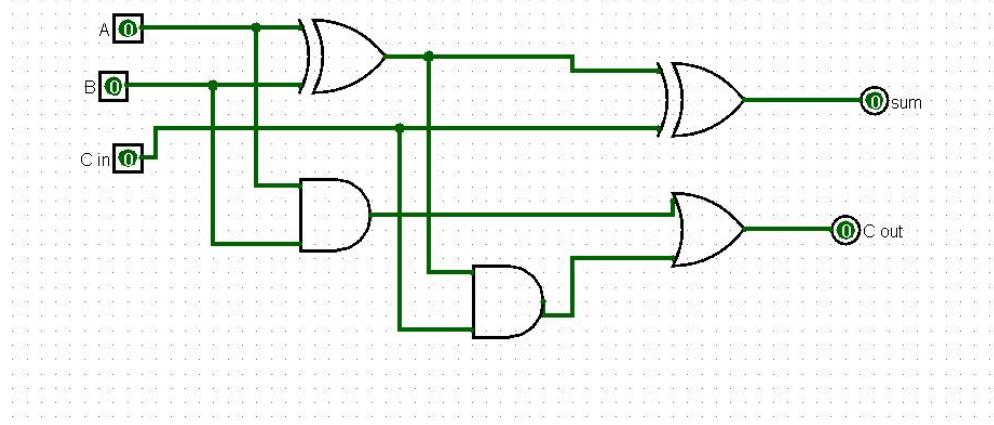
Breadboard and Simulation Design

- The circuit is built on a breadboard using logic gate ICs for arithmetic and logical operations.
- LEDs are used to display the outputs.
- Proteus software is used for initial design and simulation to ensure proper functionality before hardware implementation.

By using a DMUX, the project demonstrates an efficient and scalable method for controlling multiple operations within a single ALU circuit.

Design and Implementation:

1) Full Adder

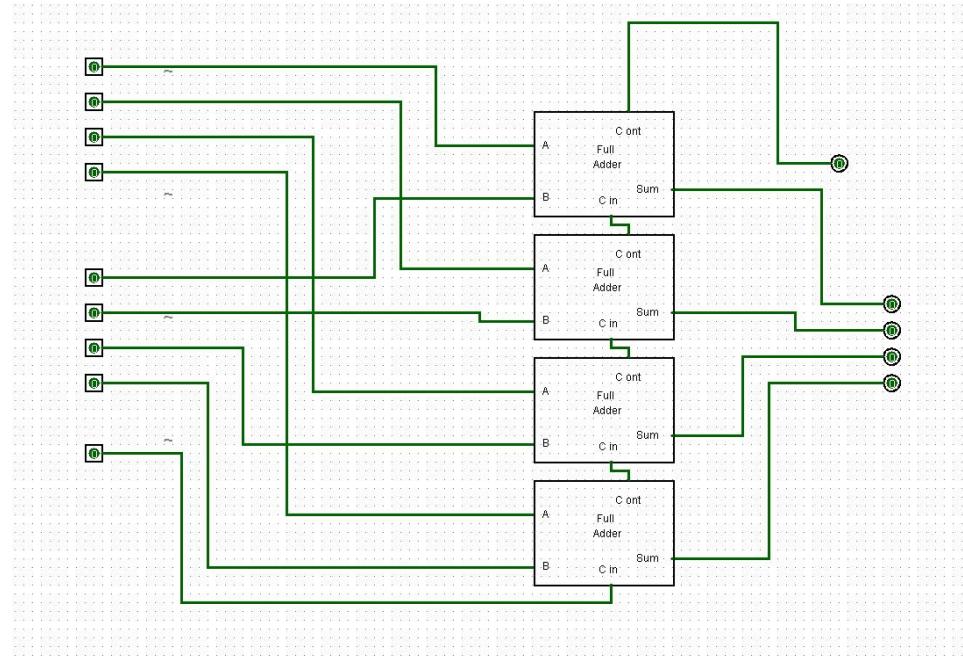


This is the logic circuit for a 1-bit full adder. A full adder computes the sum of three binary inputs: A, B, and Carry-in (C_{in}), producing two outputs: Sum and Carry-out (C_{out}).

-Sum Output: Generated using a series of XOR gates.

-Carry Output: Derived using AND gates and an OR gate to combine partial carry contributions from the inputs.

2) 4-Bit Adder



This is a 4-bit binary adder implemented using four cascaded full adders. It performs the addition of two 4-bit binary numbers along with a Carry-in input.

Inputs:

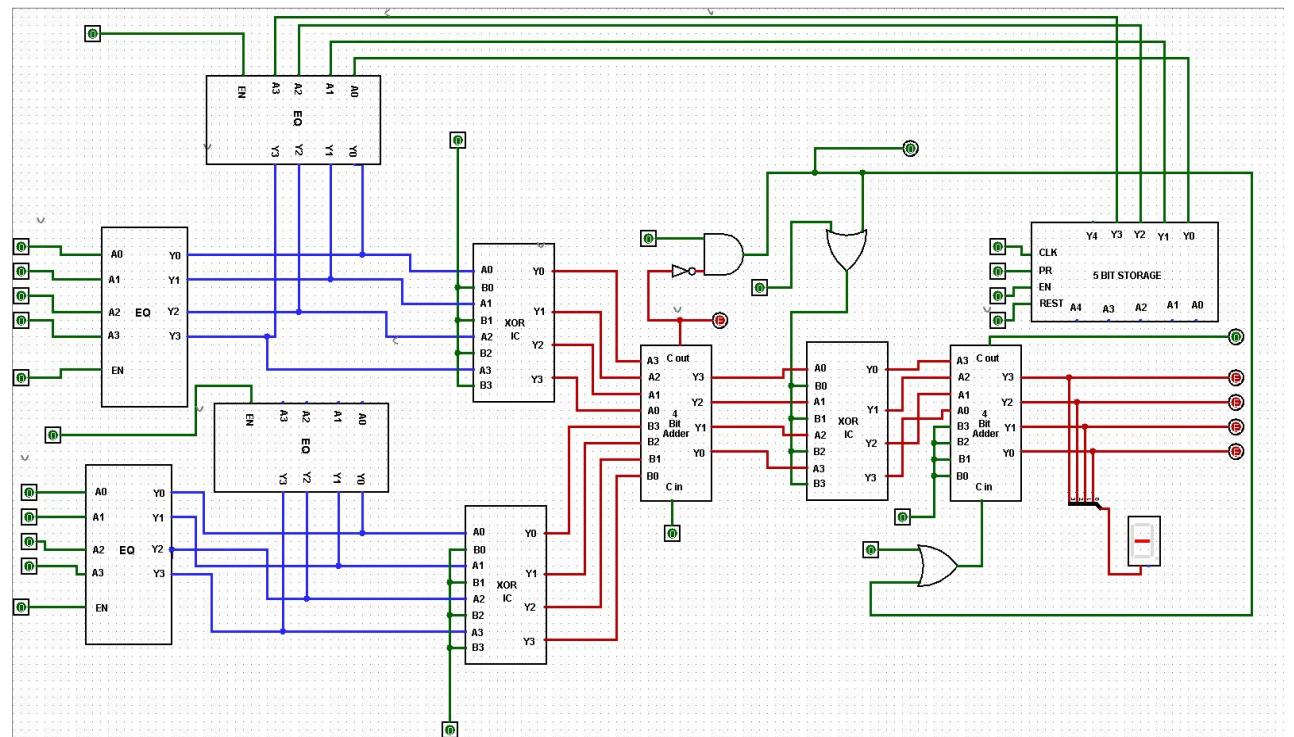
- Two 4-bit binary numbers (A3-A0 and B3-B0).
- A Carry-in for the least significant bit (LSB).

Outputs:

- A 4-bit Sum (S3-S0).
- A Carry-out from the most significant bit (MSB).

- Each full adder computes the sum and carry for a single bit.
- The carry output of one stage serves as the carry input to the next stage.
- The final carry-out provides the overall carry from the 4-bit addition.

3) Arithmetic Logic Unit



This circuit is a 4-bit Arithmetic Logic Unit (ALU) designed to perform arithmetic and logical operations. It uses components like XOR gates, adders, demultiplexers, and a 5-bit storage unit to process binary inputs and control signals.

Key Components:

1. Demultiplexer (DMUX):

- Routes the control signals to enable specific operations (e.g., addition, subtraction).
- The control inputs determine the active path.

2.4-Bit Full-Adder:

- Performs addition of two 4-bit binary numbers (A and B).
- Generates a 4-bit sum and a carry-out bit.

3. XOR Gates:

- Used for operations like subtraction or bitwise XOR.
- For subtraction, XOR inverts bits of the second operand (B) when combined with a carry-in.

4.5-Bit Storage:

- Temporarily stores results from operations.
- Outputs the result for visualization or further processing.

5.Logic Gates:

- Perform basic logical operations like AND, OR, and NOT.

Inputs:

1.Binary Inputs (A and B):

- Two 4-bit numbers provided as input to the ALU.
- Represented as A0–A3 and B0–B3.

2.Control Signals:

- Determine which operation the ALU performs (e.g., addition, subtraction, logical AND/OR).

3.Clock and Enable Signals:

- Manage data flow and synchronization in the storage unit.

Outputs:

1.Result:

- The output of the selected operation (e.g., sum, XOR result).
- Displayed as a 4-bit result (Y0–Y3) or a 5-bit result in the storage unit.

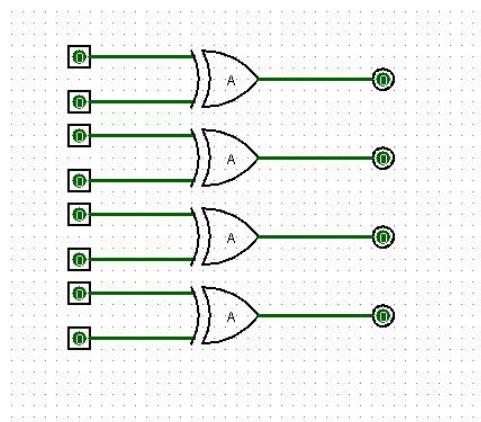
2.Carry-Out (Cout):

- Indicates overflow from the most significant bit during addition.

How It Works

- 1.The user provides two 4-bit binary inputs (A and B).
- 2.Control signals select the desired operation using the DMUX.
- 3.The inputs are processed through the respective circuit (adder, XOR gates, or logical gates).
- 4.The result is stored in the 5-bit storage unit or displayed via LEDs.
- 5.Outputs include the operation's result and a carry-out bit for arithmetic operations.

4) XOR IC



A 4-bit XOR IC is typically used for performing bitwise XOR operations on two 4-bit binary numbers. XOR gates output a high signal (1) when the inputs are different and a low signal (0) when the inputs are the same.

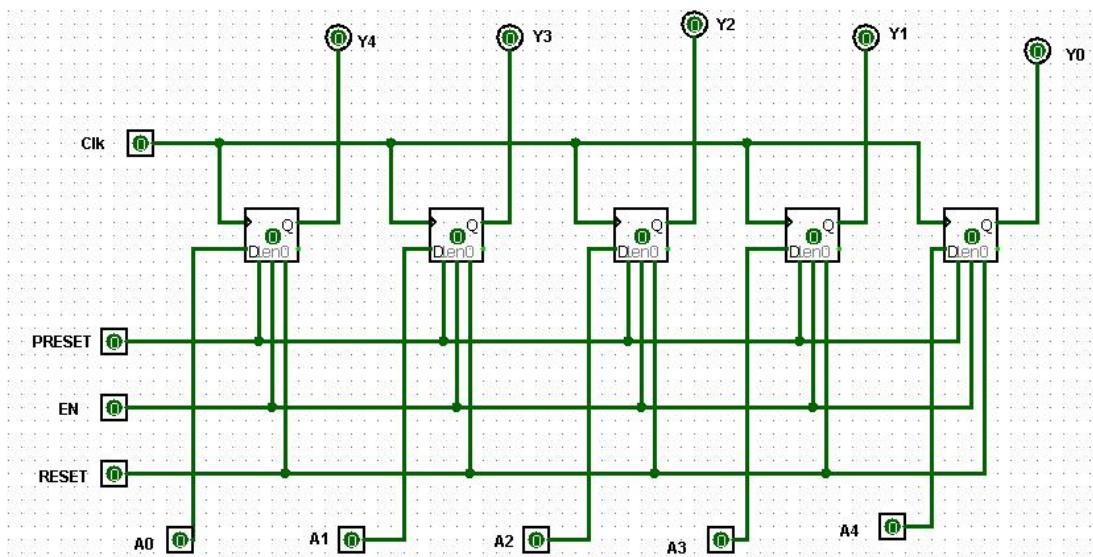
Inputs:

Two 4-bit binary numbers (A0-A3 and B0-B3).

Output:

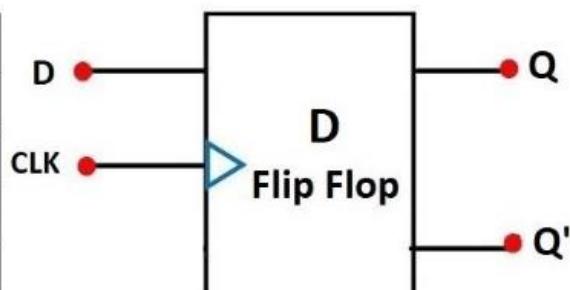
A 4-bit result (R0-R3), where each bit is the XOR of the corresponding input bits.

5) 5-Bit Storage (Register)



Truth Table of DFlip Flop

D	CLK	Q	Q'
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0



Components:

5 D Flip-Flops: Each flip-flop stores 1 bit of binary data.

Clock Signal: A common clock synchronizes all flip-flops to ensure the data is stored or updated simultaneously.

Data Inputs: 5 inputs (A4, A3, A2, A1, A0) to store the binary bits.

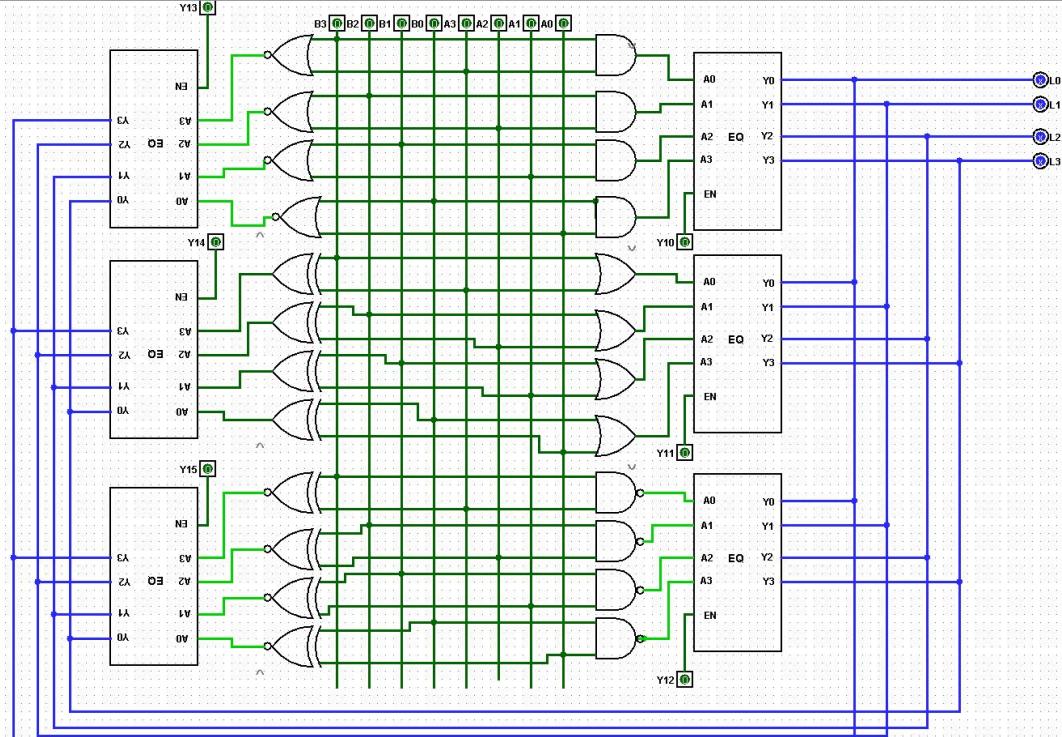
Outputs: 5 outputs (Y4, Y3, Y2, Y1, Y0) to represent the stored data.

Circuit Description:

Each flip-flop is connected to a corresponding bit of the 5-bit input data.

When the clock signal triggers (usually on the rising or falling edge), the data present at the input (A) is latched into the flip-flop and appears at the output (Y).

6) Logic Unit



Inputs:

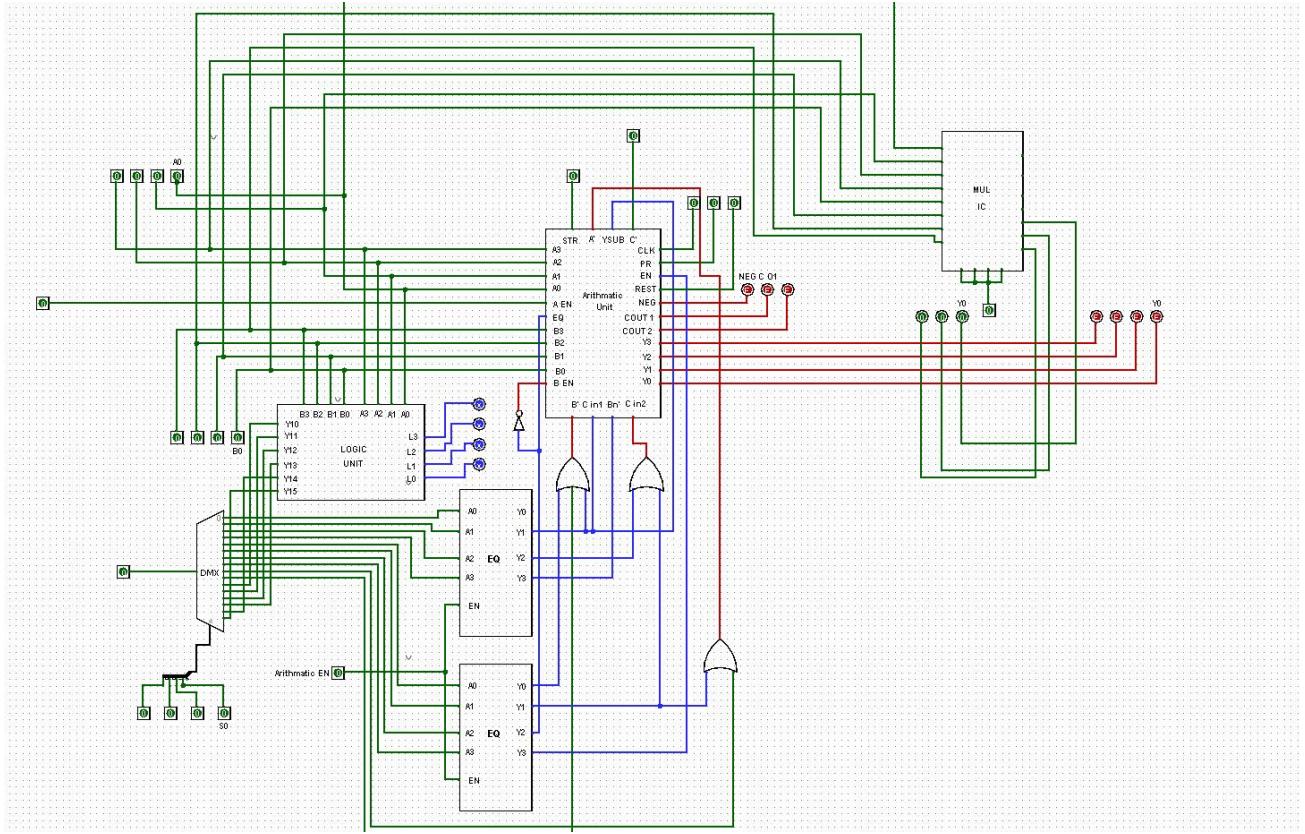
Two binary numbers (A and B), typically of the same bit width.
A control signal or operation selector to determine which logical operation to perform.

Outputs:

A demultiplexer takes a single input (or a set of inputs, such as operation results) and routes it to one of several outputs based on the select signal.

10	1	0	1	0	A AND B	AND
11	1	0	1	1	A OR B	OR
12	1	1	0	0	A NAND B	NAND
13	1	1	0	1	A NOR B	NOR
14	1	1	1	0	A XOR B	XOR
15	1	1	1	1	A XNOR B	XNOR

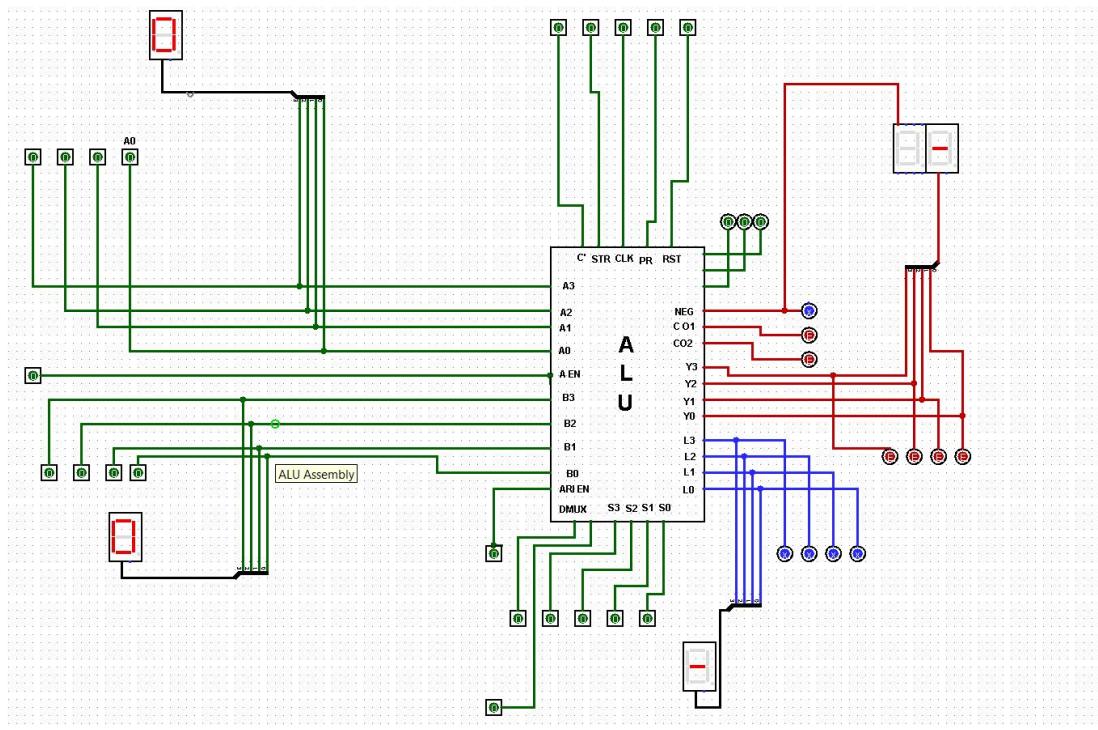
7) Arithmetic Assembly



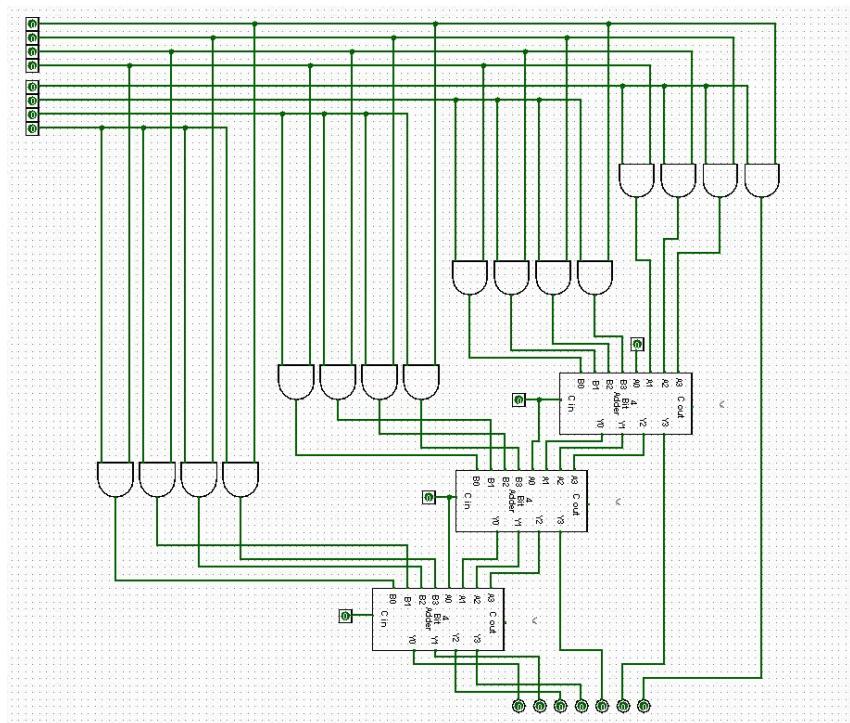
Components:

- ALU (Arithmetic Logic Unit):** This is likely the central processing unit of the circuit, responsible for performing arithmetic and logical operations on data.
- Registers:** These are memory elements used to store data temporarily. We can see several registers labeled A, B, C, and others.
- Control Unit:** This unit is responsible for generating control signals to sequence the operations of the ALU and other components.
- Multiplier:** This component performs multiplication operations on the input data.
- Gates (AND, OR, NOT):** These are basic logic gates used to perform boolean operations.

8) ALU Main Circuit

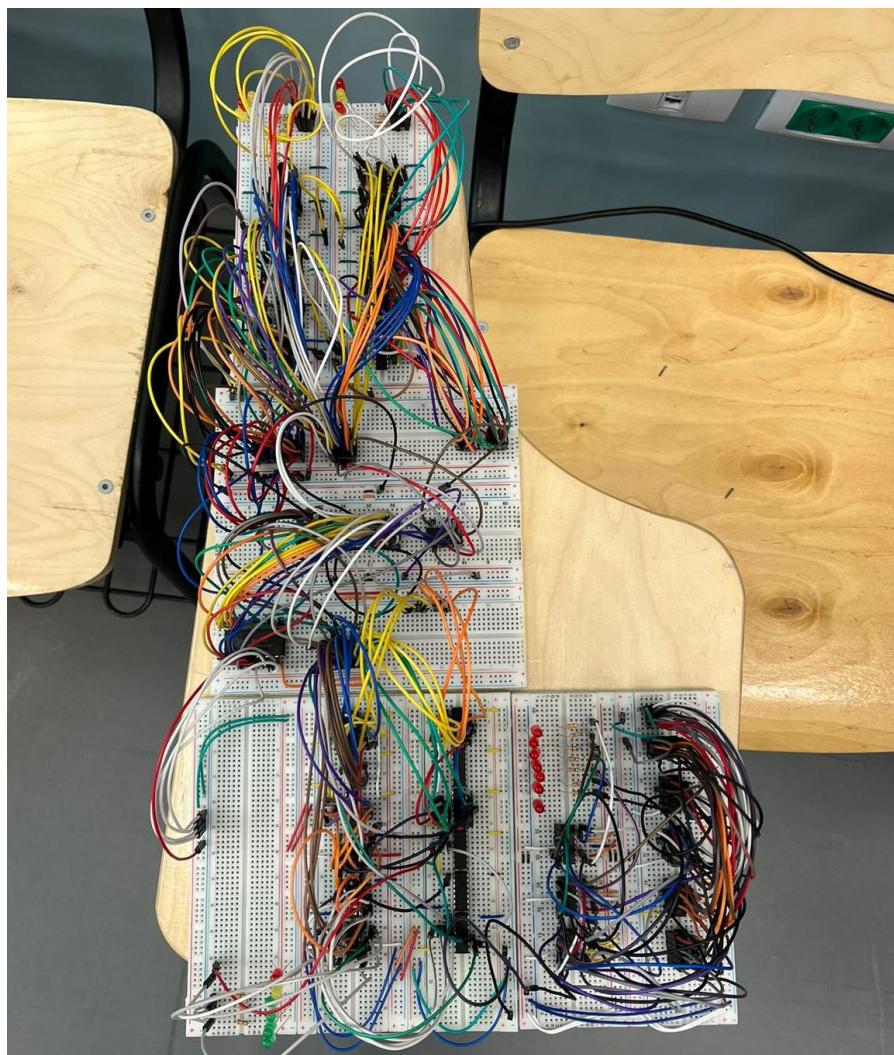


9) Multiplication

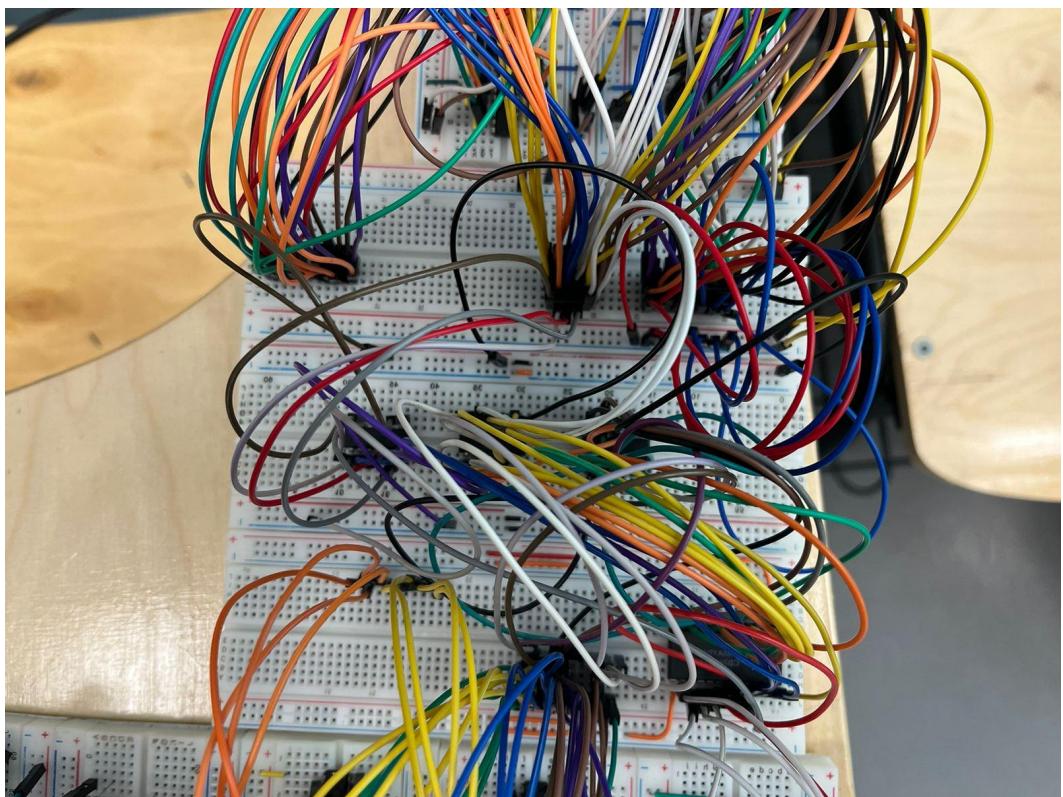
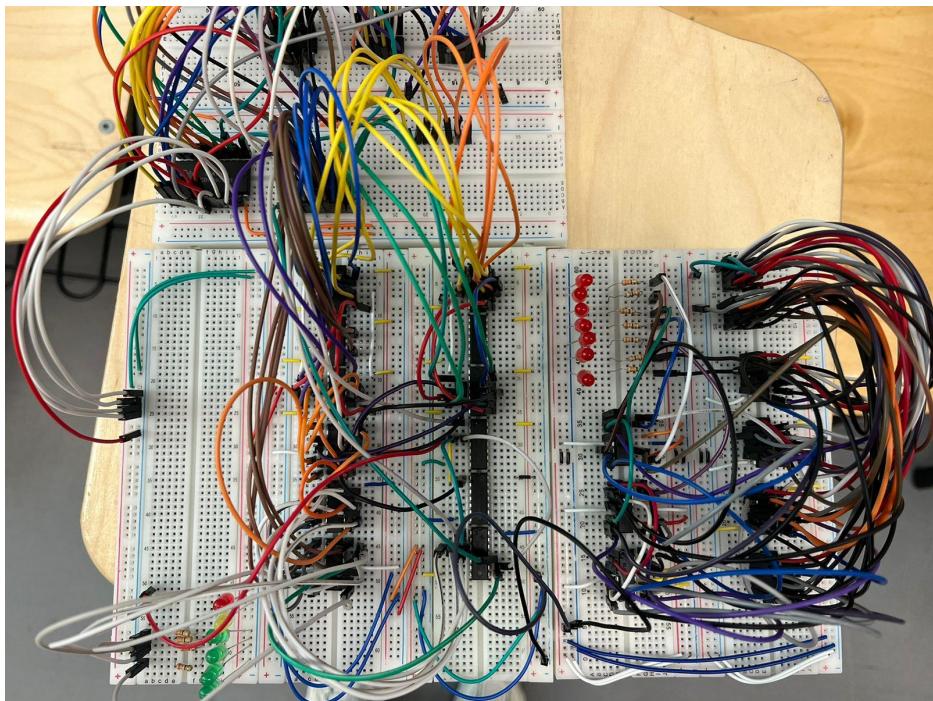


- **Inputs:** The circuit takes two binary numbers as inputs, represented by the lines at the top left.
- **Partial Products:** The AND gates in the upper portion generate the partial products by performing bitwise multiplication of individual bits from the two input numbers.
- **Adder Units:** The partial products are then summed using cascaded full adders (labeled as ICs) to compute the final binary product. The carry bits propagate between stages to ensure accurate addition.
- **Output:** The result of the multiplication is output as a binary number at the bottom-right corner of the circuit.

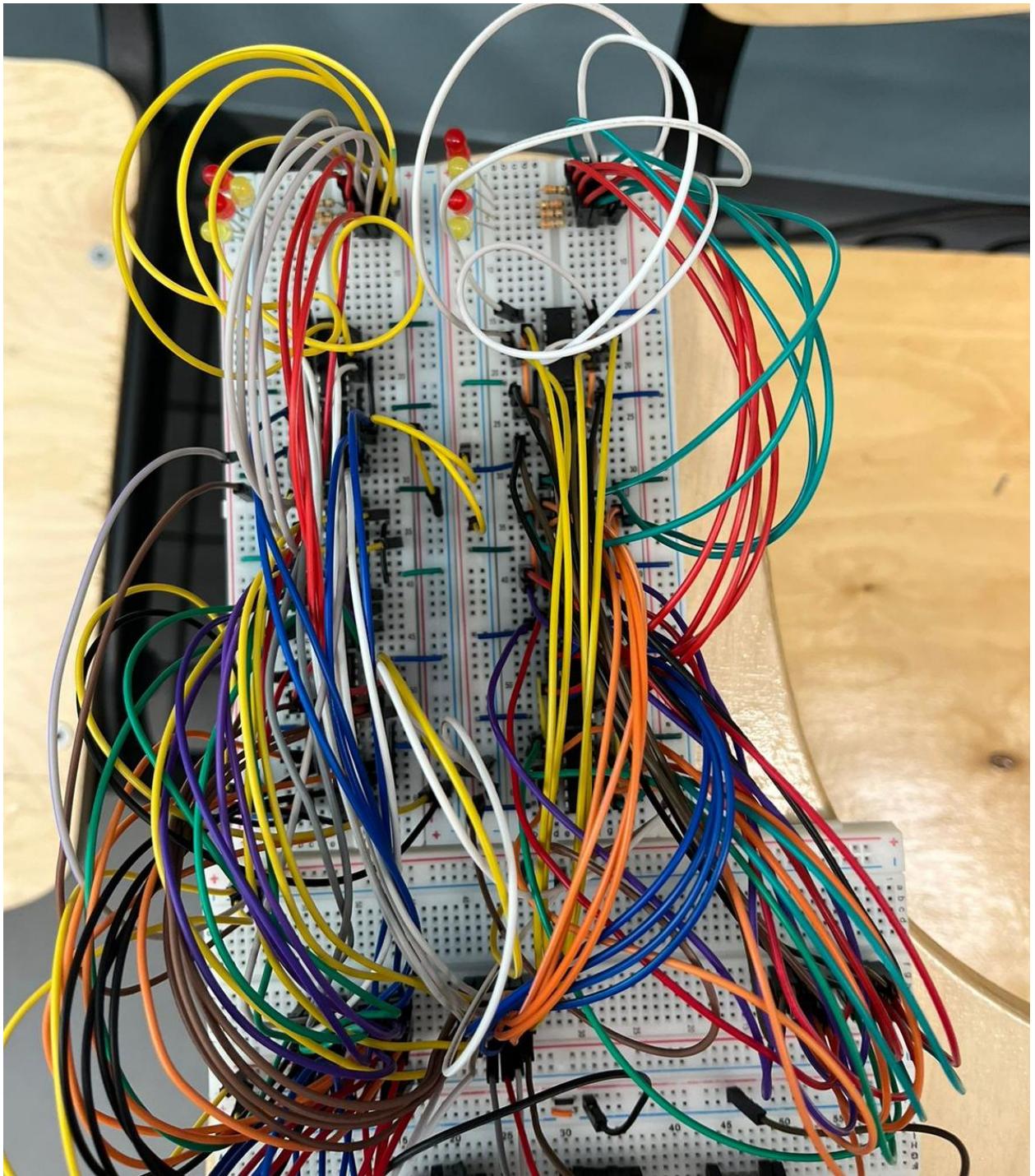
Breadboard Implementation



Multiplication and Arithmetic circuit



Logic Unit



Conclusion

In this project, we successfully designed and implemented a 4-bit Arithmetic Logic Unit (ALU) using a combination of combinational and sequential circuits. By integrating components such as a demultiplexer (DMUX), logic gates, a 4-bit full-adder, and a 5-bit storage unit, we demonstrated how an ALU performs basic arithmetic and logical operations.

The project allowed us to gain hands-on experience with digital circuit design and simulation tools, including breadboards and Logisim software. This practical approach deepened our understanding of how fundamental digital systems function in computing devices. Moreover, it showcased the importance of control signals, synchronization, and storage in digital circuits.

Through this project, we developed problem-solving and debugging skills, improved teamwork, and explored the practical applications of theoretical concepts in computer science. Overall, the implementation of the 4-bit ALU was a valuable learning experience and a step forward in understanding complex digital systems.

References:

1. M. Morris Mano and Michael D. Ciletti, **Digital Design: With an Introduction to the Verilog HDL, VHDL, and SystemVerilog**, 5th Edition, Pearson, 2012.

•For understanding combinational and sequential circuits.

2. “Arithmetic Logic Unit (ALU),” GeeksforGeeks,
https://www.geeksforgeeks.org/introduction-of-alu-and-data-path/?ref=gcse_outind

•Explanation of ALU functionality and design.

3. Class Notes on Digital Logic and Circuit Design.

•Lecture materials from our course, covering ALU design principles.