Technical Report

Circuit Name: 4-bit Arithmetic Logic Unit (ALU)

Team members	**Member ID**
Abdulrahman Elsaied Waheeb Alhadek	172314
Abdullah Omar Elazhary	168462

Under Supervision: Dr. Mohamed Youssef

1. Abstract

This project implements a 4-bit Arithmetic Logic Unit (ALU) using VHDL. The ALU performs multiple arithmetic and logical operations such as addition, subtraction, multiplication, and comparison (greater than, less than). The ALU is modularly designed and tested through simulation using a testbench. The timing diagram and RTL schematic validate the functionality of the circuit.

---

2. Introduction

An ALU (Arithmetic Logic Unit) is a critical component of a digital system responsible for performing arithmetic and logical operations. This project focuses on designing a simple yet functional 4-bit ALU using VHDL. The design is divided into smaller reusable blocks for each operation, which are integrated into a single top-level ALU module. This modular approach improves readability, testing, and scalability.

The ALU supports the following operations:

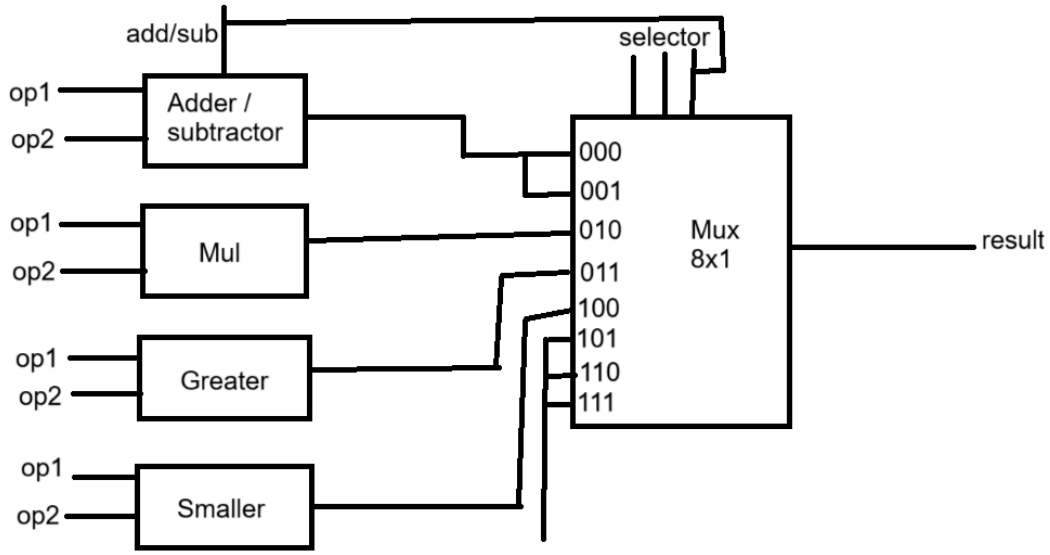
- 4-bit Addition and Subtraction
- 4-bit Multiplication
- Greater Than Comparison
- Smaller Than Comparison

Each block was individually coded in VHDL, tested through a testbench, and then connected in a hierarchical structure within the ALU entity.

---

3. Design

Block Diagram



The top-level ALU receives two 4-bit inputs (A and B) and a 3-bit selector that determines the operation to perform. Internal modules are instantiated for different operations.

Truth Table

Selector	Operation
000	A + B
001	A - B
010	A * B
011	A > B
100	A < B
101-111	No operation

Output Equations

Operations are implemented using behavioral or structural VHDL, and the selection is done using an 8-to-1 multiplexer.

Component Descriptions

3.1 FULL ADDER

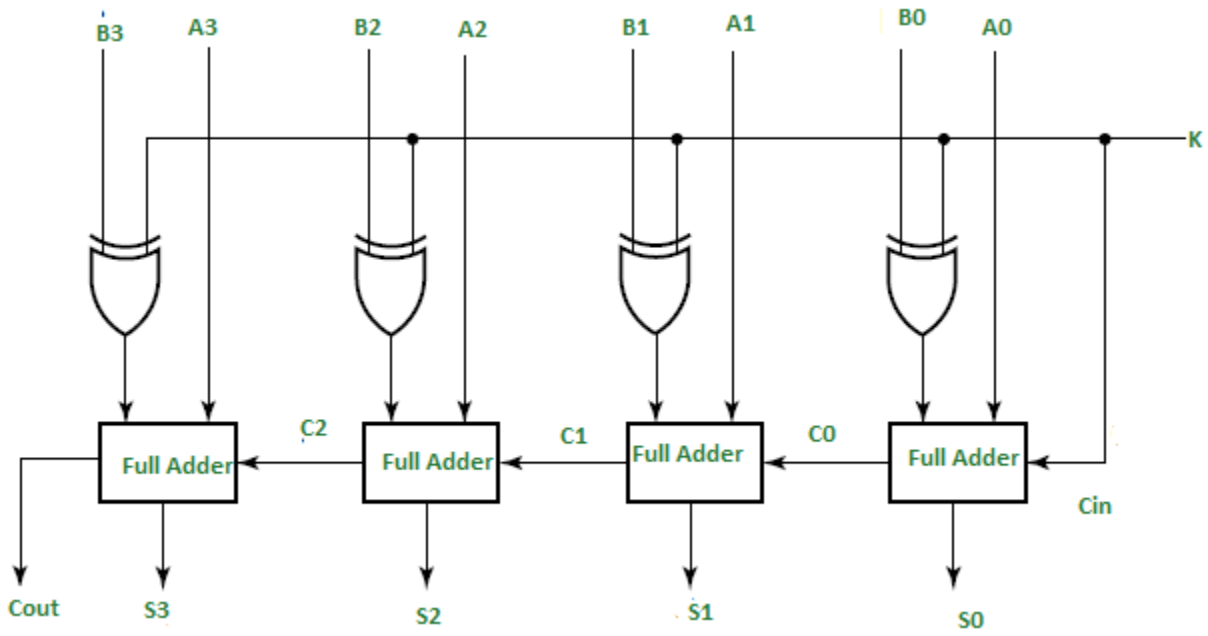
A single-bit full adder adds two bits and a carry-in, generating a sum and a carry-out. It is the fundamental building block for the multi-bit adder.

3.2 FULL ADDER 4 BIT

This module chains four full adders to construct a 4-bit ripple carry adder. It is responsible for performing the addition of two 4-bit numbers.

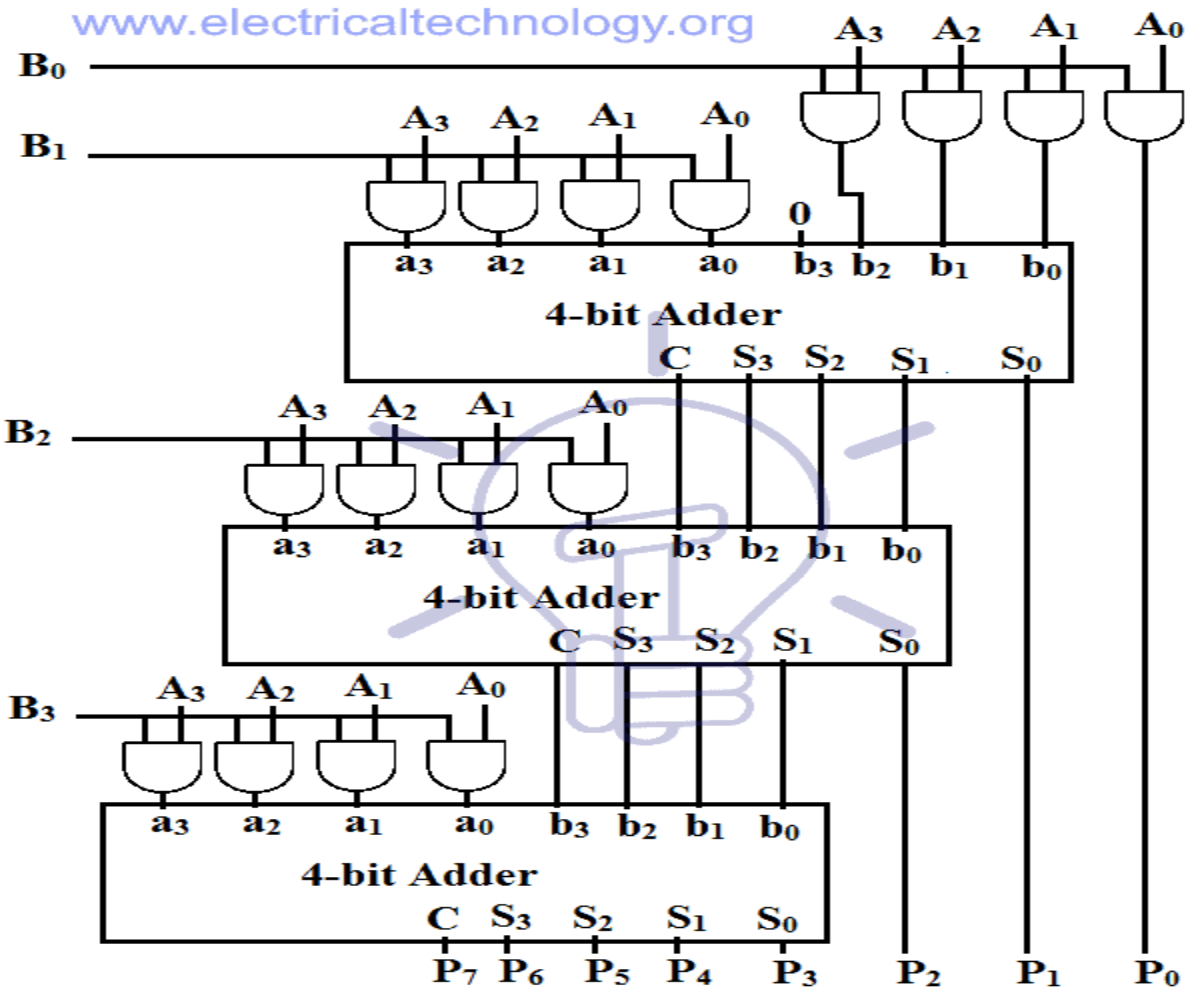
3.3 ADDER_SUBTRACTOR_4_BIT

This module extends the adder to support subtraction by applying 2's complement logic to the second operand when subtraction is selected. It handles both addition and subtraction operations based on a control bit.



3.4 MULTIPLIER_4_BIT

Implements a 4-bit binary multiplication using an array multiplier structure. Each bit of the multiplier is ANDed with the multiplicand, and the partial products are shifted and added to form the result.



3.5 GREATER_THAN

Compares two 4-bit inputs and outputs a logic high if the first input is greater than the second. Implemented using comparator logic.

3.6 SMALLER_THAN

Similar to the GREATER_THAN module, but outputs logic high when the first input is less than the second.

3.7 MUX_8_TO_1_8_BIT

An 8-to-1 multiplexer selects the output of one of the functional modules based on the 3-bit selector. This output is routed as the final result of the ALU.

3.8 ALU TOP MODULE

The top-level module integrates all functional blocks and connects them to the multiplexer. It defines the operation mode and ensures proper routing of inputs and outputs.

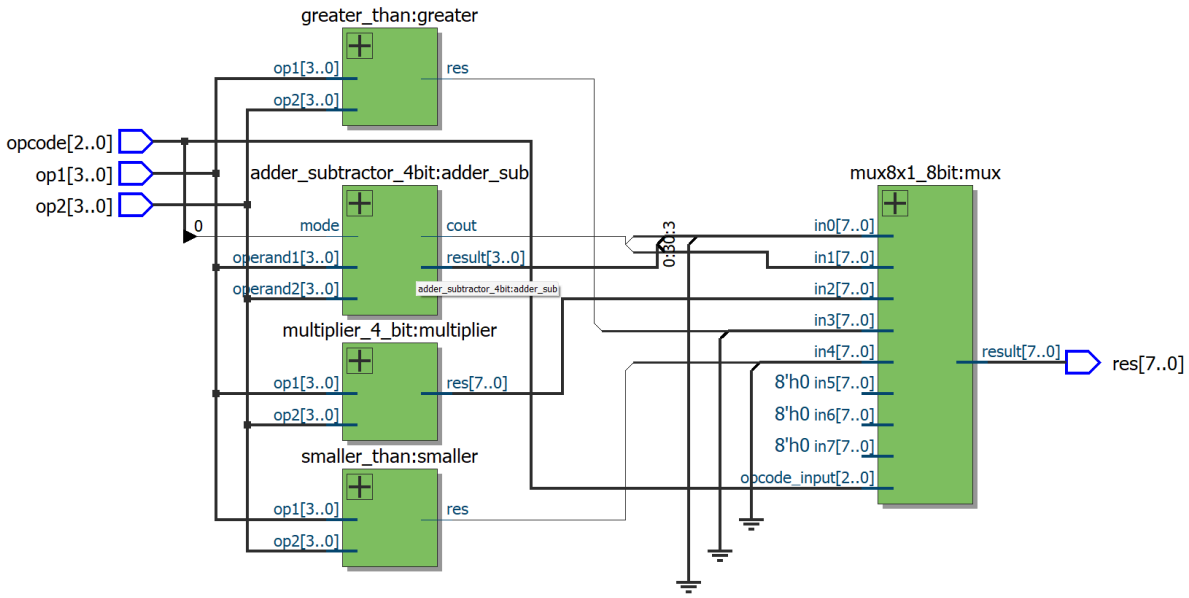
---

5. VHDL Code and Simulation

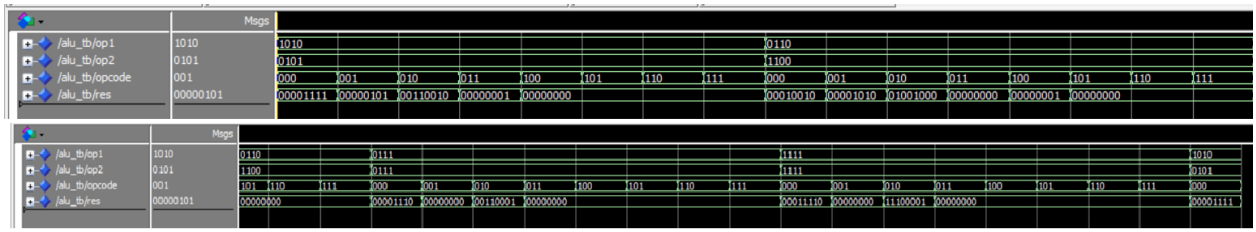
a. VHDL and Testbench Code

- `Mini Project`

c. RTL Schematic



d. Timing Diagram



---

8. References

- Binary Multiplier - Types & Binary Multiplication Calculator
- 4-bit binary Adder-Subtractor - GeeksforGeeks
- How to Build Your Own Discrete 4-Bit ALU - Projects

---

9. Conclusion and Future Work

The 4-bit ALU was successfully designed and tested using VHDL. The modular design approach made it easier to verify each component independently. In the future, the ALU can be extended to support additional operations like division, bitwise AND/OR, or shift operations. Moreover, the bit-width can be extended from 4-bit to 8-bit or 16-bit to handle larger data sizes.