Design and CMOS Implementation of a 4-Bit Arithmetic Logic Unit (ALU) Using Electric VLSI

Abstract

This paper presents the design, schematic capture, CMOS layout, and transient simulation of a 4-bit Arithmetic Logic Unit (ALU) implemented using the Electric VLSI Design System. The ALU is constructed from custom-designed CMOS logic cells including XOR, NAND, multiplexers, tri-state buffers, and full adders. The project demonstrates the complete VLSI workflow—from transistor-level design to layout and SPICE simulation—highlighting issues encountered during LVS/DRC verification and incorrect simulation results caused by missing well taps and unconnected bulk terminals.

I. Introduction

The purpose of this laboratory project is to design a functional 4-bit Arithmetic Logic Unit (ALU) using CMOS VLSI design techniques. The ALU is a fundamental digital component enabling arithmetic operations such as addition and subtraction. Implementing such a system at the transistor level provides practical experience with CMOS logic design, Electric VLSI CAD tools, layout verification, and SPICE simulation.

II. Background and Theory

The ALU is designed using standard CMOS logic gates. Full adders are constructed from XOR and NAND gates. A multiplexer determines whether the ALU performs addition or subtraction, with subtraction implemented using two's complement inversion of the A input coupled with a controlled carry-in. Tri-state output buffers control data flow depending on an enable signal.

In CMOS processes, NMOS transistors must be placed in p-substrate regions tied to GND through NTAPs, while PMOS transistors must be placed in n-wells tied to VDD through PTAPs. Failure to include proper substrate/well contacts results in floating bulk terminals, causing threshold shifts, body effect, and unpredictable output behavior.

III. Methodology

The design was implemented hierarchically. Individual cells such as inverters, XOR gates, NAND gates, and multiplexers were designed first. These were combined to form a full-adder cell, and four full adders were interconnected to implement a 4-bit ripple-carry ALU. The schematic was then translated to transistor-level layout, ensuring proper design rule checking (DRC), layout vs schematic (LVS), and well-tap insertion.

IV. Figures

Alu Input signal

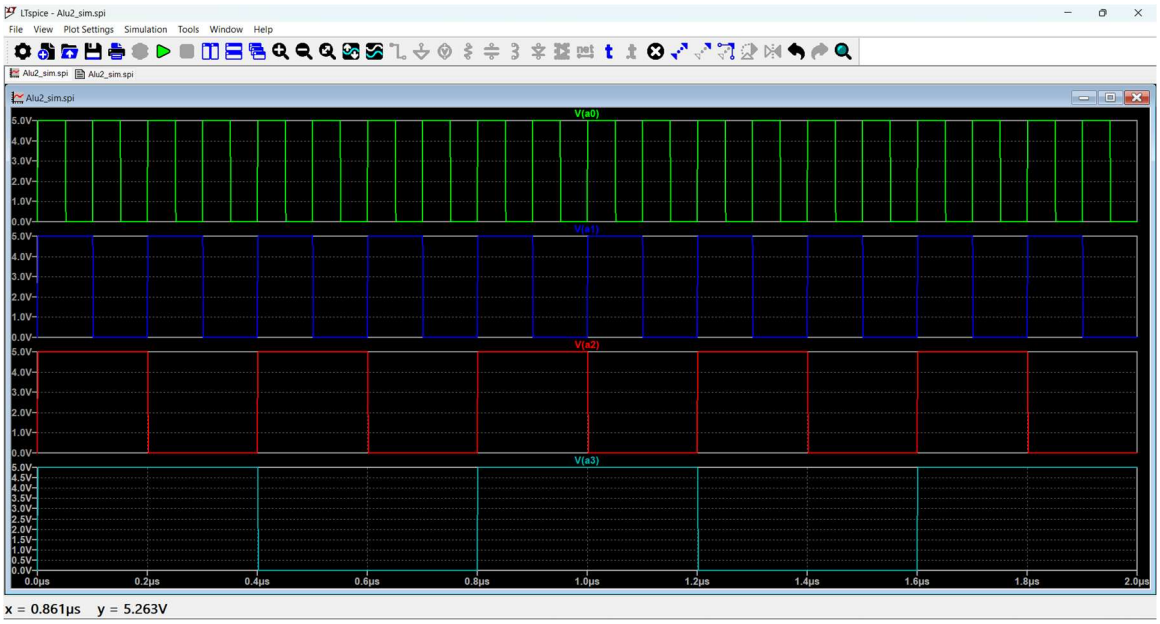

Figure: Screenshot 2025-11-14 102537.png

Alu Output signal

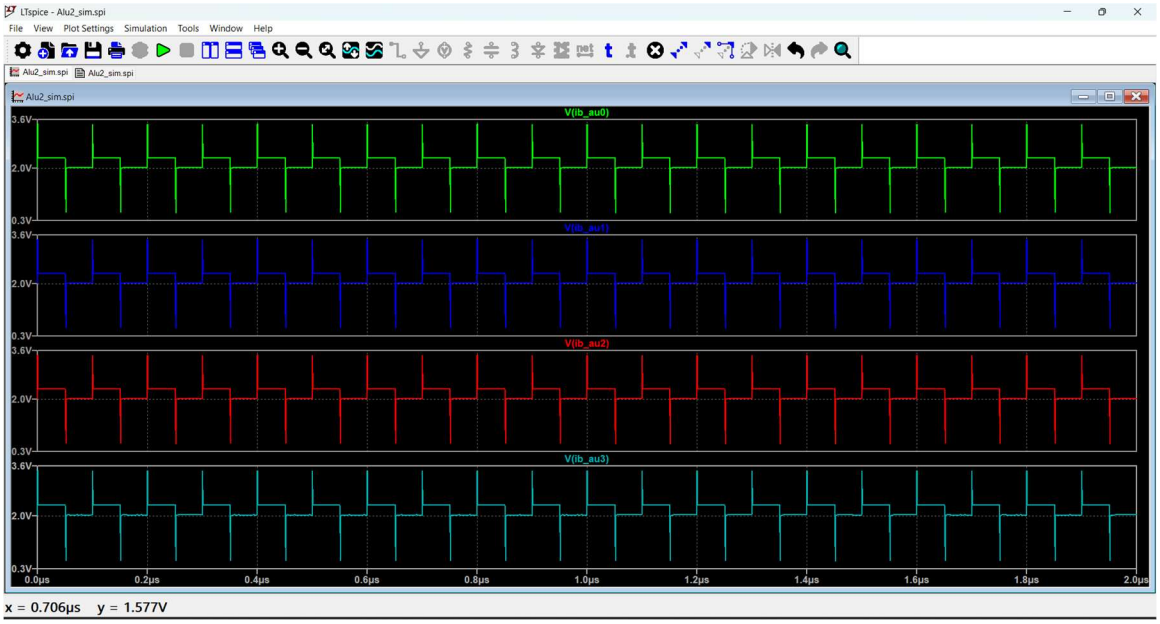

Figure: Screenshot 2025-11-14 102617.png

ALU Simulation

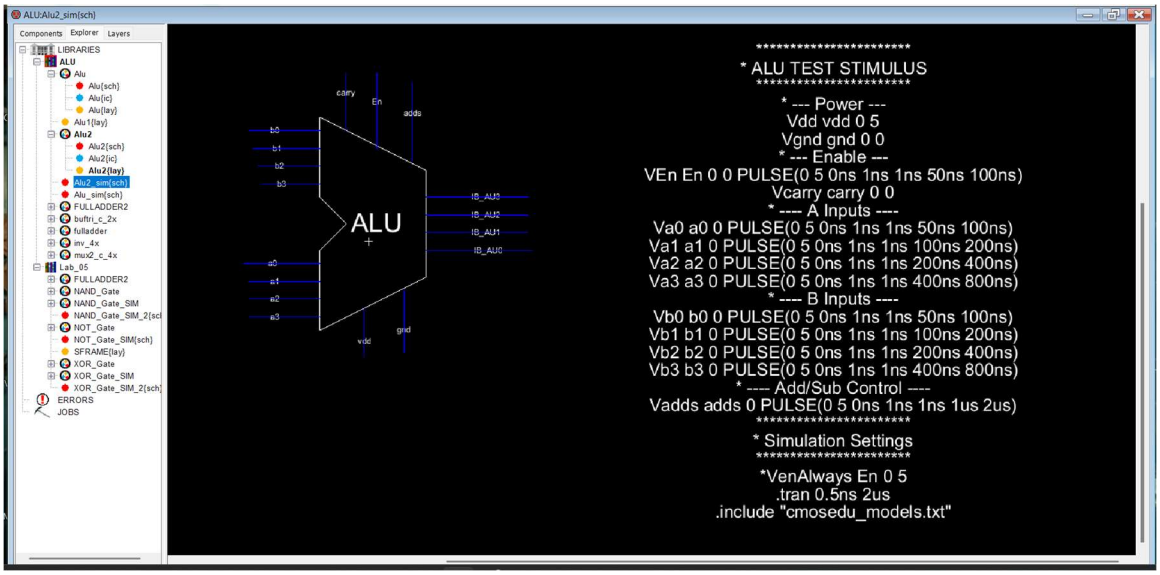

Figure: Screenshot 2025-11-14 102636.png

ALU Schematics

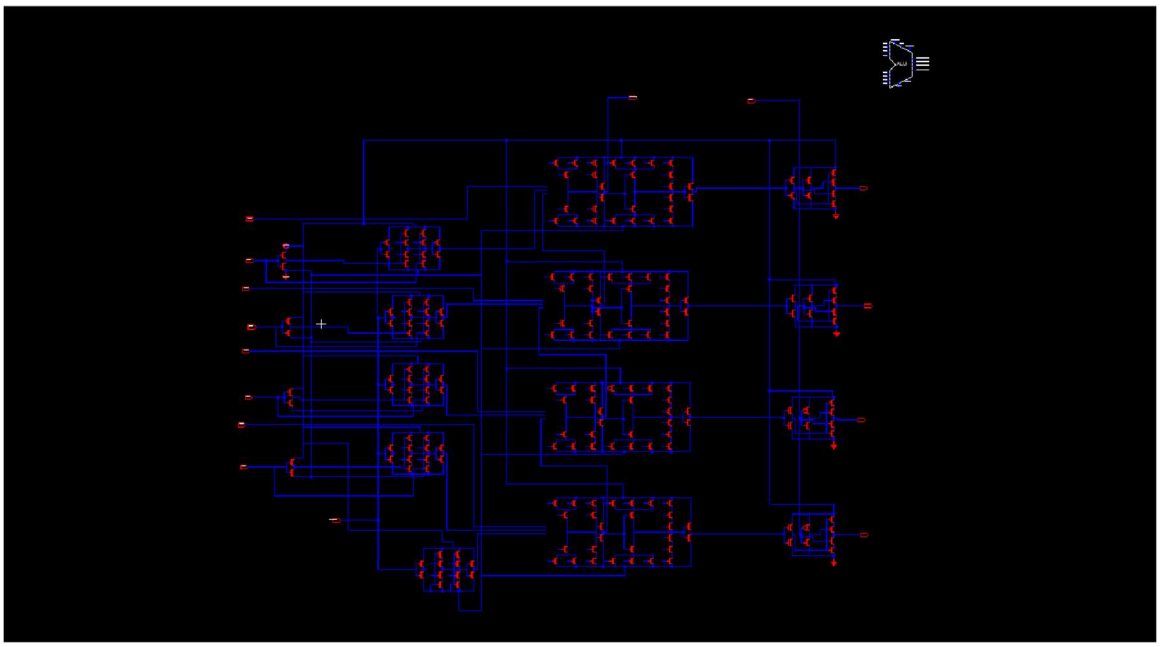

Figure: Screenshot 2025-11-14 102713.png

ALU Layout

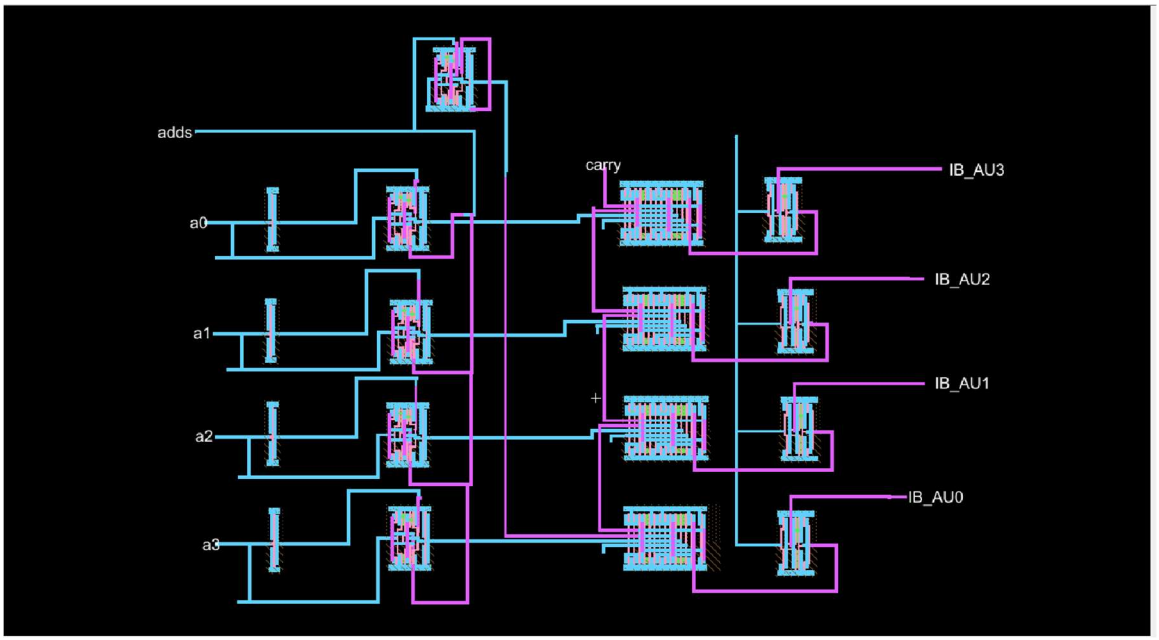

Figure: Screenshot 2025-11-14 102722.png

ALU Icon

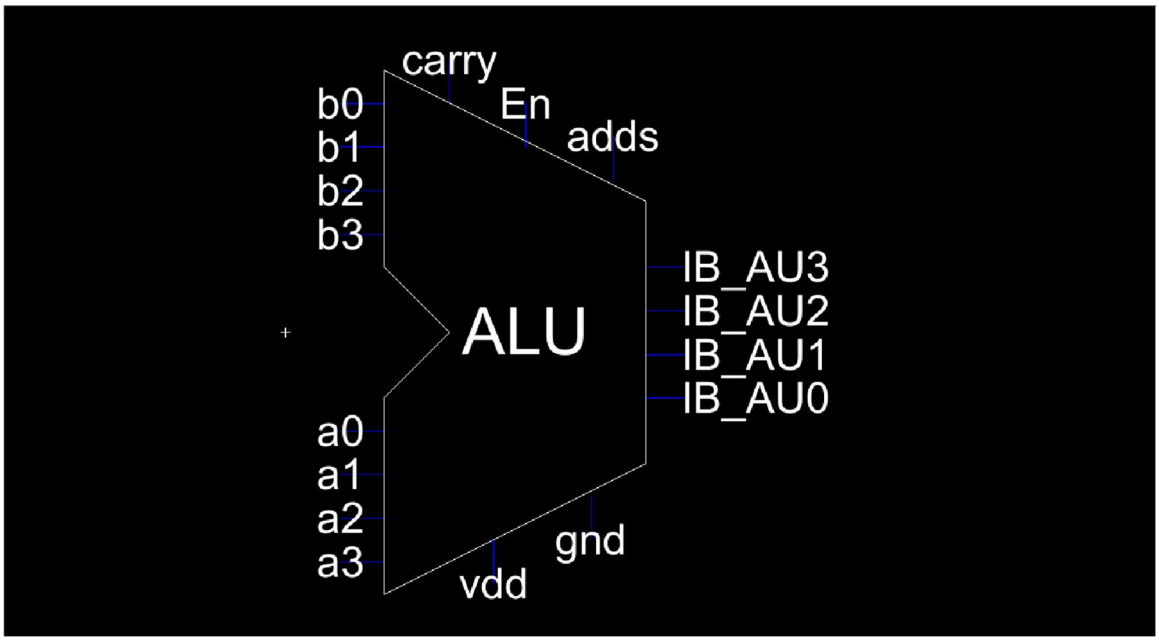

Figure: Screenshot 2025-11-14 102732.png

3d View of the ALU Layout

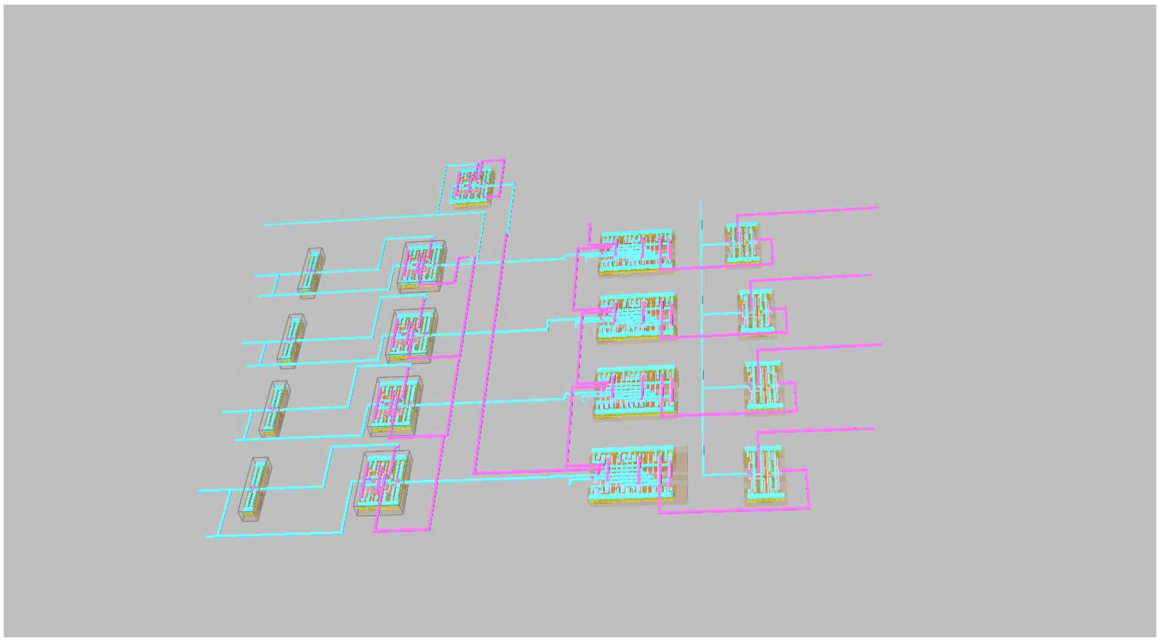

Figure: Screenshot 2025-11-14 102756.png

V. Simulation Results

Transient SPICE simulation was performed using the extracted SPICE deck generated by Electric. The input stimulus consisted of pulsed waveforms applied to the A and B buses, with add/sub control driven by a 1 µs periodic signal. Although the waveforms for A and B inputs appeared correct, the ALU outputs exhibited degraded and incorrect behavior. This was determined to be caused by missing PTAP/NTAP connections and floating bulk nodes in the layout.

VI. Discussion

The incorrect ALU output waveforms illustrate the importance of ensuring correct body-bias connections in CMOS layouts. Without proper NTAPs and PTAPs, threshold voltages drift due to body effect, resulting in intermediate or unstable logic levels. Another issue identified was improper labeling of global VDD and GND, which caused inconsistencies in the extracted SPICE netlist. Correcting these issues is essential for achieving a functional ALU.

VII. Conclusion

This laboratory successfully demonstrated the CMOS VLSI workflow for designing a 4-bit ALU. While the schematic was correct, layout issues—specifically missing well taps and bulk connections—produced incorrect simulation results. The project highlights the critical role of proper layout practices, DRC/LVS verification, and transistor-level understanding in digital VLSI design.