Objective :

① To explore the function of an Arithmetic Logic Unit (ALU) and understand how it performs arithmetic and logical operations.

② To understand the use of combinational logic in the design of an ALU.

③ To design a basic 1-bit ALU as the foundation for wider ALUs.

④ To construct a 16-bit ALU by connecting sixteen 1-bit ALUs.

⑤ To observe how modular design simplifies the creation of multi-bit ALUs like 16 bit or 32 bit versions.

Equipment :

① Logisim Software

Block Diagram :

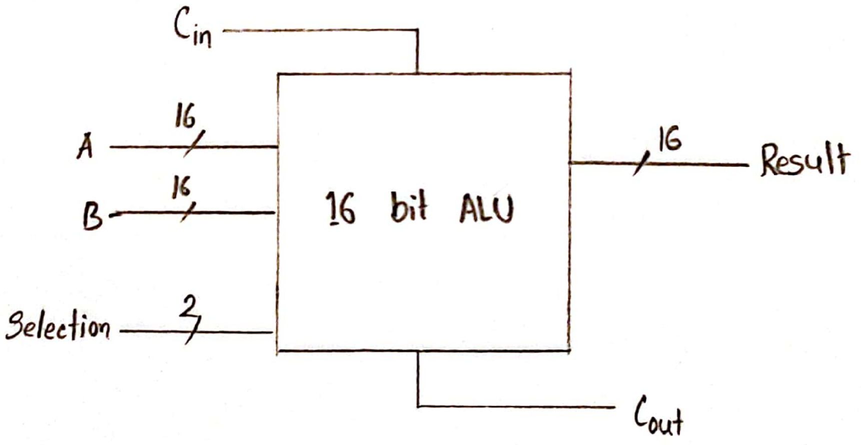

Fig : Block Diagram of 16 bit ALU

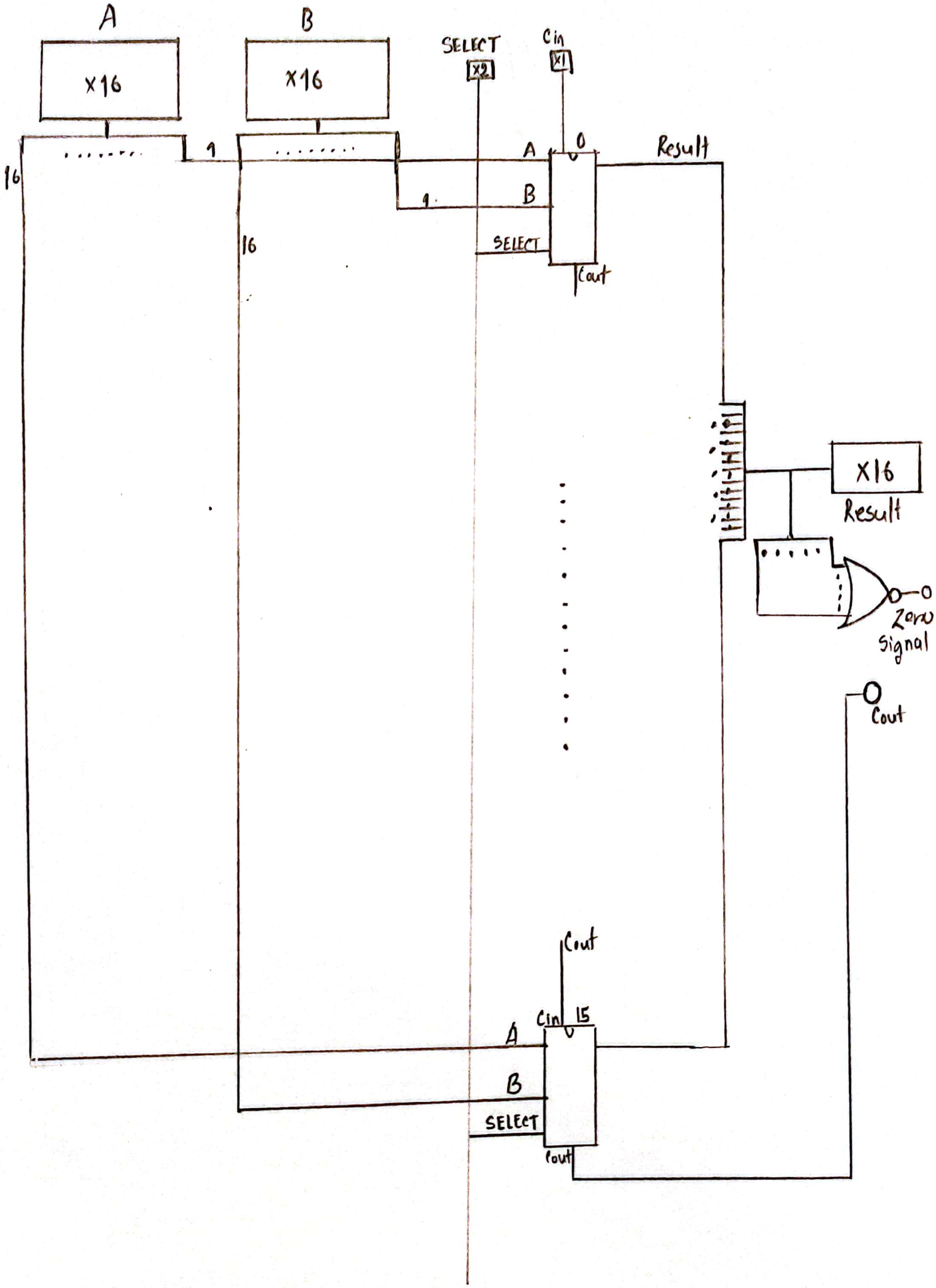

Fig - 16 - bit ALU Design

Discussion :

In this experiment, I designed a 1-bit ALU using basic combinational logic components in Logisim. After successfully implementing the 1-bit wider version, I created a 16-bit ALU by connecting 1-bit ALUs together. I also incorporated key features such as addition, subtraction, AND, OR operations, as well as zero signal logic, which are essential in real-world ALU functionality.

Although the lab was fairly straightforward, one of the difficulties I faced was connecting the wires properly within the limited screen space of Logisim and Laptop also. It was hard to see where each connection was going, especially when managing multiple bits and control signals, which made the circuit abit cluttered and confusing at times.

To overcome this challenge, we relied heavily on visual cues and paid close attention to wire alignment. I also made good use of Logisim's subcircuit feature, which helped simplify the wiring and reduced screen clutter.