Lab # 03: Implementation of Arithmetic Logic Unit (ALU), Register Files and BCD to Seven Segment Decoder in LogiSim

Objectives:

After completing this experiment, students will be able to:

- Understand the basic functions of an ALU.
- Use different components like MUX, ADDER, SUBTRACTOR and basic Logical Operations in LogiSim.
- Design and Implement 1-, 8-, 16- and 32-bit ALU in LogiSim.
- Design and Implement 4-, 8-, bit Register Files in LogiSim.
- Design and Implement BCD to Seven Segment Decoder

Background Theory

An Arithmetic Logic Unit (ALU) is a fundamental component in digital systems, playing a crucial role in performing arithmetic and logical operations. It serves as the computational heart of a microprocessor or a digital circuit, executing mathematical calculations and logical comparisons that are essential for processing data.

Functions of an ALU

Addition

The primary arithmetic function of an ALU is addition. It can add binary numbers, which is a fundamental operation in any digital computing system. The ALU takes two binary inputs and produces their sum as the output.

Subtraction

Subtraction is closely tied to addition in digital systems. An ALU can perform subtraction by employing a mechanism known as two's complement arithmetic. This allows for efficient addition of negative numbers, and the ALU can effectively subtract one binary number from another.

Logical AND Operations

ALUs also perform logical operations, including the AND operation. In the AND operation, each bit of the output is the logical AND of the corresponding bits of the input. It results in a '1' only if both input bits are '1'; otherwise, the output is '0'.

Logical OR Operations

Another essential logical operation is the OR operation. The OR operation produces a '1' if at least one of the corresponding input bits is '1'. It results in '0' only when both input bits are '0'.

Logical XOR Operations

The XOR operation is a logical operation where the output is '1' if the input bits are different and '0' if they are the same. XOR is frequently used in various applications, including data comparison and error detection.

Importance in Digital Systems

Central Processing Unit (CPU)

ALUs are a critical component of a CPU, where they perform the arithmetic and logical operations necessary for executing program instructions. The speed and efficiency of an ALU directly impact on the overall performance of the CPU.

Data Processing

In digital systems, ALUs are responsible for manipulating data, making them indispensable in applications ranging from basic calculators to complex data processing units in computers and embedded systems.

Control Flow

ALUs play a key role in controlling the flow of operations within a digital system. They enable decision-making processes, comparisons, and conditional branching based on the results of arithmetic and logical operations.

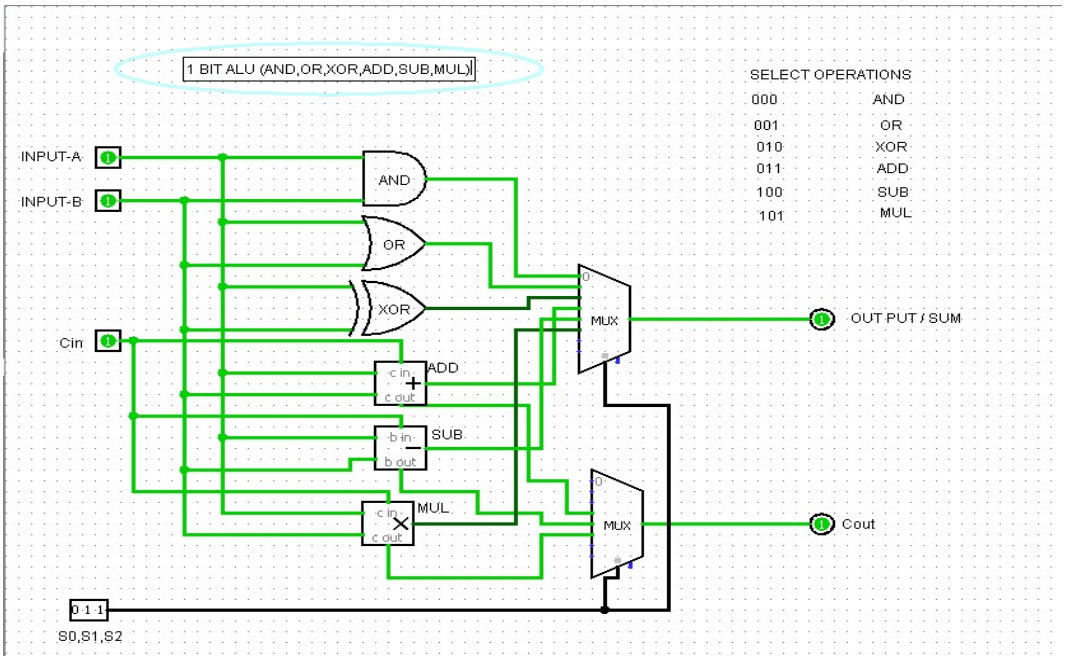
Mathematical Computations

ALUs are crucial for performing mathematical computations, making them integral to applications such as scientific calculations, simulations, and numerical analysis.

So, the Arithmetic Logic Unit is a fundamental building block of digital systems, providing the computational capabilities necessary for a wide range of applications in the field of electronics and computing. Its ability to perform both arithmetic and logical operations makes it a versatile and indispensable component in the world of digital design and information processing.

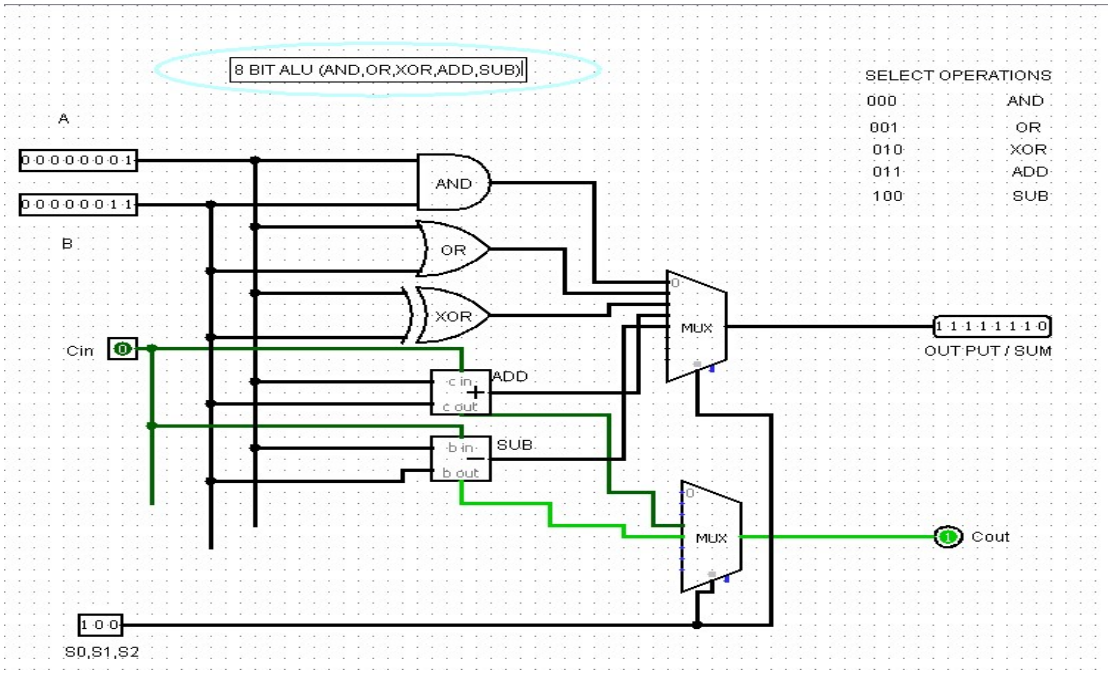
1 Bit ALU Design

Designing a 1-bit Arithmetic Logic Unit (ALU) with basic operations like AND, OR, XOR, addition, subtraction, and multiplication involves careful arrangement of logic components. In the following Figure: 2.1, two multiplexers (MUX) play a crucial role in managing the output and carry-out signals. For each operation, the ALU utilizes logic gates such as AND, OR, and XOR to process the input bits. The addition and subtraction operations employ a 1-bit full-adder for the arithmetic computations. To facilitate the selection of the desired operation, a control line is connected to the input of the MUX responsible for generating the output. Another MUX is incorporated to manage the carry-out signal. During addition or subtraction, the carry-out from the full-adder is selected, while for logical operations, the carry-out is set to zero. By implementing this design, the 1-bit ALU effectively performs various operations, and the multiplexers contribute to the flexibility of choosing the appropriate results based on the operation selected.



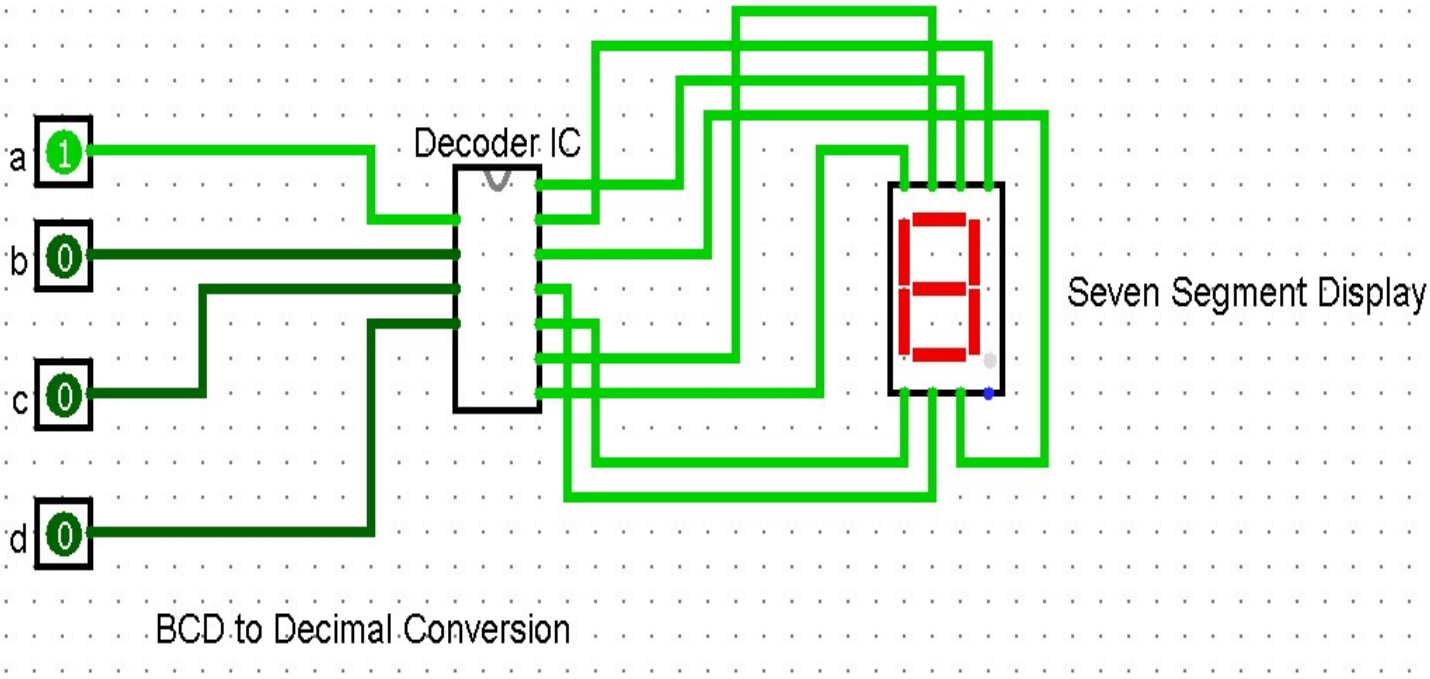
8 Bit ALU Design

Just change the input and output bit size to 8 bits and the rest of the circuit will remain the same as shown in Figure: 2.2.



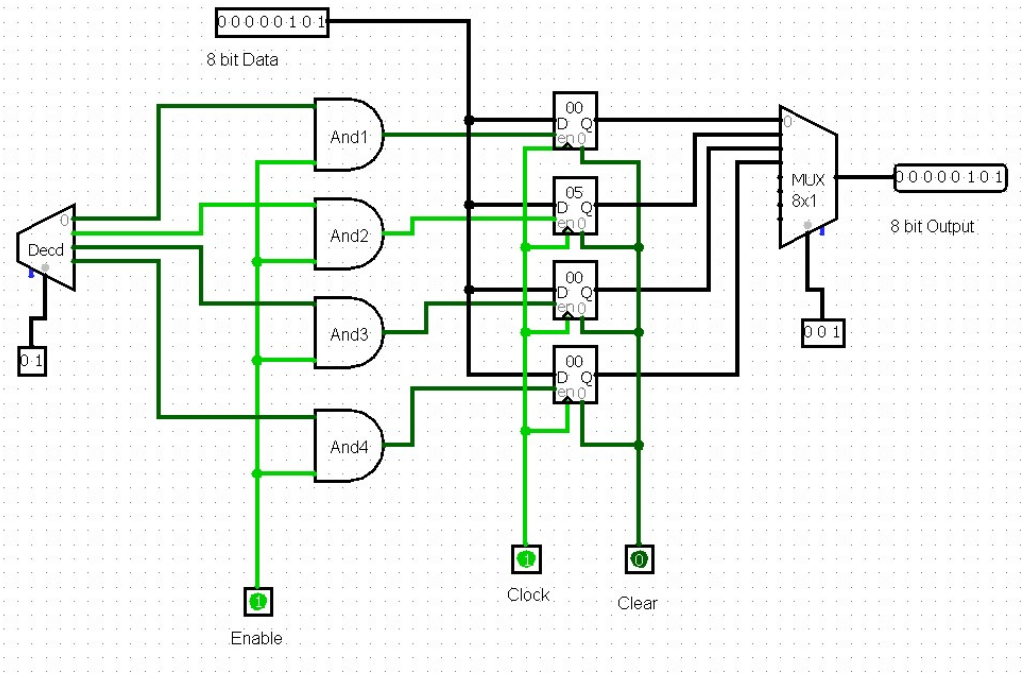
BCD to Decimal and Seven-Segment Display

BCD (Binary-Coded Decimal) is a method of representing decimal numbers in binary form, where each digit of a decimal number is represented by a 4-bit binary code. This allows for easy conversion between binary and decimal systems, especially in digital electronics where decimal values need to be processed. For example, decimal digit 5 is represented in BCD as 0101. A Seven-Segment Display is a popular electronic component used to visually display decimal numbers. It consists of seven LEDs arranged in a figure-eight shape, which can be lit up in various combinations to represent different digits. Each segment of the display can be individually controlled to create the desired digit. When converting a BCD input to a Seven-Segment Display, a decoder circuit is typically used to map the BCD input (a 4-bit binary number) to the corresponding segments that need to be lit for each digit. For instance, a BCD input of 0000 (representing the decimal digit 0) would activate the segments that display the number 0 on the Seven-Segment Display. By using a combination of logic gates, this BCD-to-display conversion can be achieved, making the Seven-Segment Display a key tool in digital electronics for displaying numerical information.



4-Bit Register File in Logisim

Register files are fundamental building blocks in digital systems, serving as small, high-speed memory units within the Central Processing Unit (CPU). They are essentially arrays of registers, each capable of storing a small amount of data, typically a word (e.g., 8-bits, 16-bits, 32-bits, or 64-bits). Organized for fast access, register files enable the CPU to quickly retrieve and store operands during instruction execution. Unlike main memory, which can be relatively slow, register files provide extremely rapid data access, significantly boosting the performance of the CPU. They are crucial for holding temporary results, intermediate values, and frequently used data, minimizing the need to access slower levels of the memory hierarchy. The number and organization of registers within a register file are key architectural features that influence the CPU's capabilities and efficiency. Operations such as arithmetic logic unit (ALU) calculations, data movement, and address generation heavily rely on the fast access provided by register files. Understanding the operation and design of register files is essential for anyone studying computer architecture and digital design.



Lab Activities:

Task-1: Design 16 Bits ALU in LogiSim having (AND, OR, XOR, ADD and SUB) functions.

Task-2: Design 8 Bits Register Files in LogiSim

Task-2: Design and implement a digital circuit that converts a Binary Coded Decimal (BCD) input into its corresponding decimal value and displays the result on a Seven-Segment Display using Logisim.