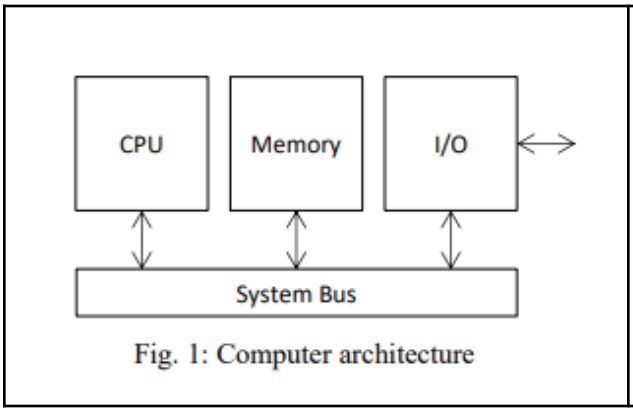


Fig. 1: Computer architecture

(1)

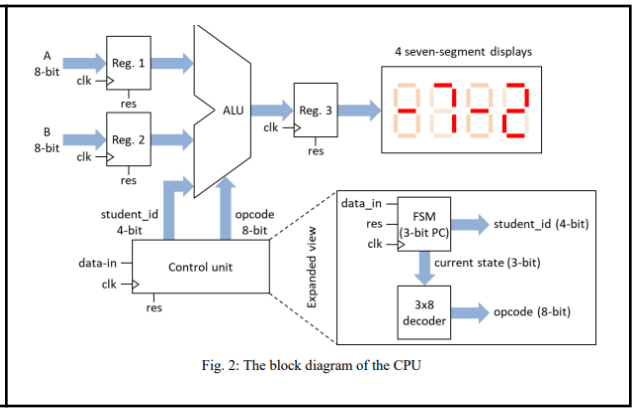


Fig. 2: The block diagram of the CPU

(2)

Project: Design of a Simple CPU

By: Rayon Dhali

Table of Contents

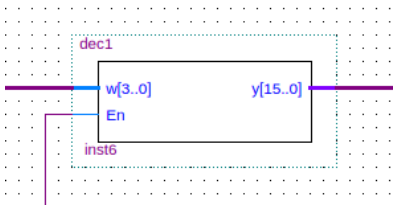
Introduction:

The purpose of the lab involves designing and implementing a simple central processing unit (CPU) (1) using VHDL on an FPGA board, with a focus on building and simulating an Arithmetic and Logic Unit (ALU). The CPU also includes a control unit that manages system operations and a program counter (PC) designed as a finite state machine (FSM) to handle instruction sequencing. Memory simulation is achieved through a 3-to-8 decoder that provides operation codes for the ALU, with operands supplied by the FPGA board switches and student ID digits. This project aims to demonstrate the fundamentals of CPU architecture (2) and the practical application of VHDL programming, bridging the gap between theoretical concepts and real-world hardware implementation.

Components: Latch1, Latch 2, 4:16 Decoder, FSM

4:16 Decoder Info:

Circuit component diagram:



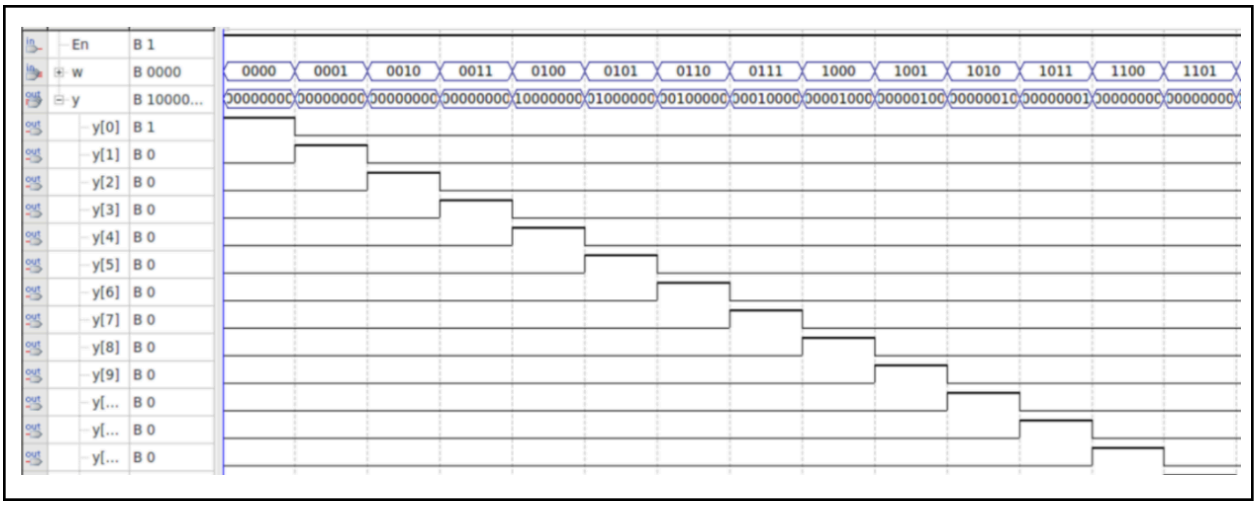
Truth Table

Inputs	Outputs
A B C D	D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 D10 D11 D12 D13 D14 D15
0000	1000000000000000
0001	0100000000000000
0010	0010000000000000
0011	0001000000000000
0100	0000100000000000
0101	0000010000000000
0110	0000001000000000
0111	0000000100000000
1000	0000000010000000
1001	0000000001000000
1010	0000000000100000
1011	0000000000010000
1100	0000000000001000
1101	0000000000000100
1110	0000000000000010
1111	0000000000000001

Brief description:

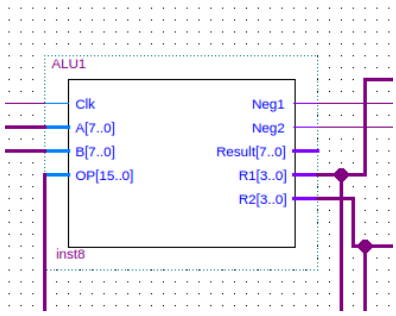
A 4-to-16 decoder is a digital logic component that converts a 4-bit binary input into one of 16 outputs. Each output corresponds uniquely to one of the 16 possible combinations of the binary input, with only one output active at any time. This device is commonly used in memory address decoding and multiplexing applications to simplify the design of digital systems.

Waveform:



ALU1 Info:

Circuit component diagram:



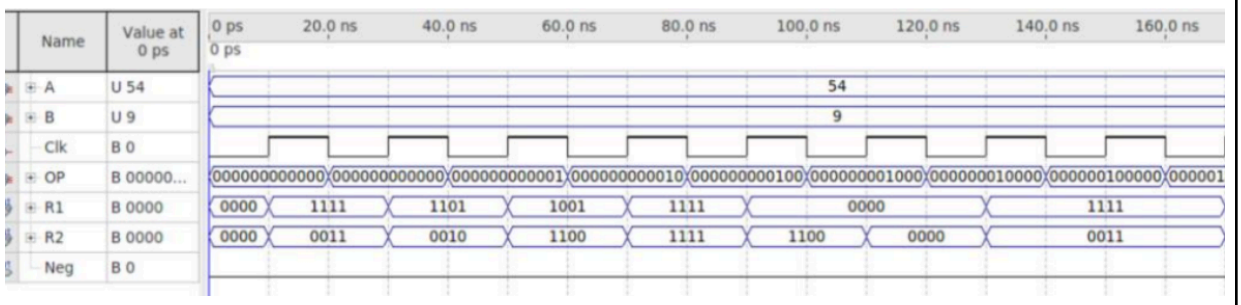
Truth Table:

Op Code (Operation)	A	B	Result (Out)
00 (Addition)	0	0	0
00 (Addition)	0	1	1
00 (Addition)	1	0	1
00 (Addition)	1	1	10
01 (Subtraction)	0	0	0
01 (Subtraction)	0	1	1
01 (Subtraction)	1	0	1
01 (Subtraction)	1	1	0
10 (AND)	0	0	0
10 (AND)	0	1	0
10 (AND)	1	0	0
10 (AND)	1	1	1
11 (OR)	0	0	0
11 (OR)	0	1	1
11 (OR)	1	0	1
11 (OR)	1	1	1

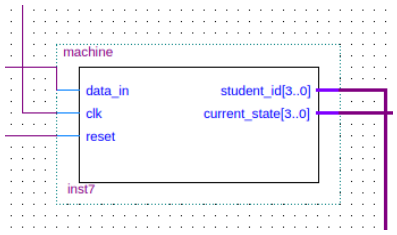
Brief description:

An Arithmetic Logic Unit (ALU) is a critical component of a CPU that handles all arithmetic and logical operations. It can perform basic functions like addition, subtraction, bitwise operations, and compare values based on input instructions. The ALU is essential for the processing power of computers, enabling them to perform complex computations efficiently.

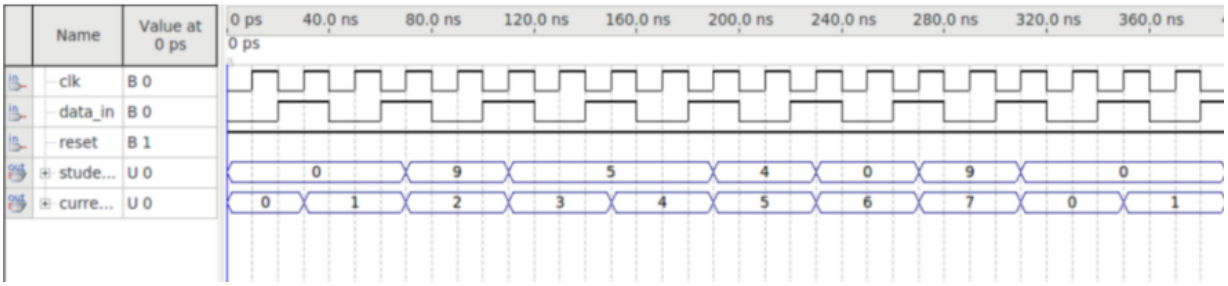
Waveform ALU1:



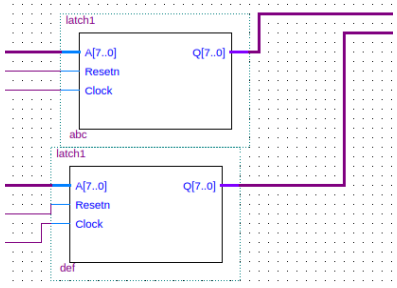
FSM Info:

Circuit component diagram:	Truth table:	Brief description:
		

Waveform FSM:

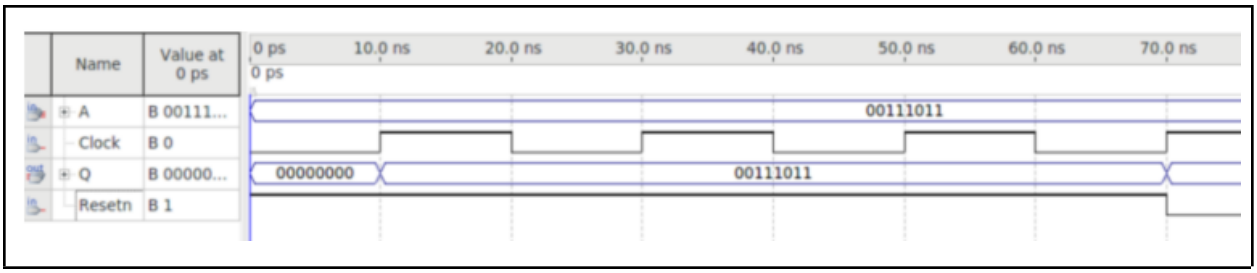


<u>Latch 1 and Latch 2:</u>

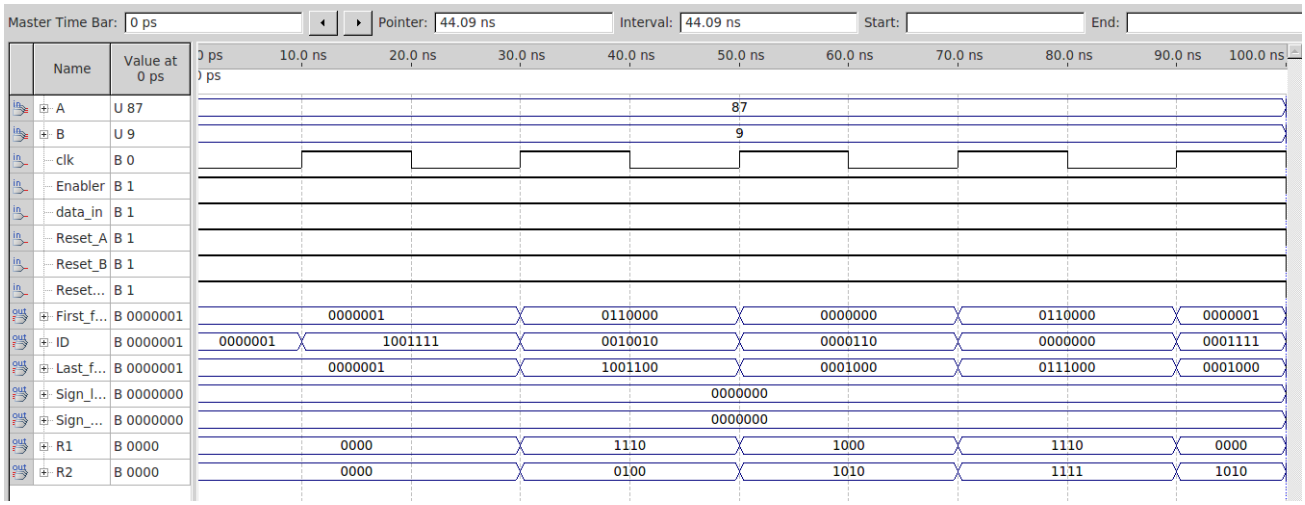
Circuit component diagram:	Truth table:	Brief description:
		In this lab, Latch 1 and Latch 2 are fundamental digital components that store binary values based on clock signals and reset conditions. They are integral in demonstrating how data is temporarily held and manipulated within a CPU's operational cycle. This setup provides practical insight into memory and processing dynamics essential for understanding computer architecture

Resetn	Clock	A (Input)	Q (Output)
0	X	XXXXXXX	00000000
1	↑	1010101	1010101
1	↑	0101010	0101010
1	↓	XXXXXXX	(no change)
1	↑	1111111	1111111

Waveform: Latch 1 and 2:



ALU 1- problem 1:



Waveform Part 1:

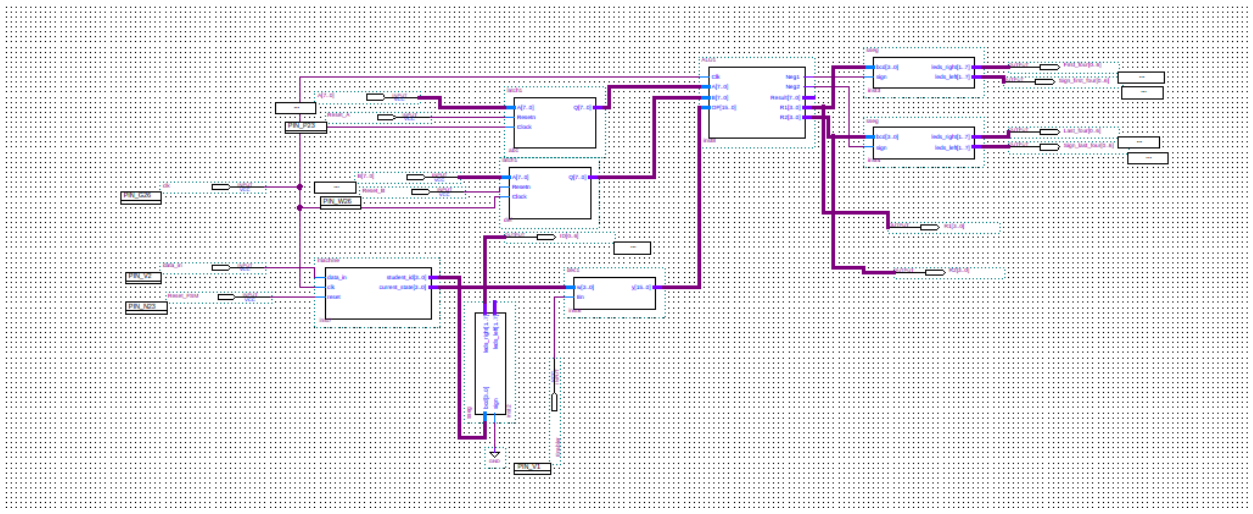


Diagram for part 1 containing Latch1 , Latch 2, 4:16 Decoder, FSM, ALU-3

Outline/design of component:

The waveform simulation shows the ALU processing inputs `Reg1` and `Reg2` based on the `current_state` from the FSM, which dictates operations via a decoder. The output `Result` displays computed values, highlighting the ALU's role in executing arithmetic and logical instructions within the CPU, as it responds dynamically to changes in control signals. This setup emphasizes the essential function of inputs and outputs in managing real-time data processing.

Conclusion:

In this lab, we successfully designed and implemented a simple central processing unit (CPU) using VHDL on an FPGA board, with a focus on the arithmetic and logic unit (ALU) and its various functions. Our tasks included modifying ALU operations such as incrementing, logical shifts, rotations, and basic arithmetic, and integrating these into the CPU framework. Through extensive procedural steps and functional simulations with Quartus software, we tested the behavior of the CPU under different input conditions and validated the functionality of each component, including registers, the finite state machine (FSM), and decoders. This lab reinforced our understanding of CPU architecture and VHDL programming, highlighting the importance of careful design and simulation in digital system development, and equipped us with practical skills and theoretical knowledge essential for designing complex digital systems in real-world applications.