COE328 Lab 6 Report: Design of a Simple General-Purpose Processor

Ahmed Tabl

Date Submission: Dec 04,2023

1. Table of Contents Pages

2. Introduction:

Lab 6 undertakes the task of designing a Simple General-Purpose Processor over a two-week span. The primary objective is to craft an effective Arithmetic and Logic Unit (ALU) using VHDL, subsequently deploying it on an FPGA board. The lab is structured with distinct components, each essential to the overall functionality of the ALU. Commencing with the acquisition of input data based on the terminal quartet of student IDs, the process sets the stage for the design and simulation of an 8-bit Register Storage Unit. Progressing further, the development extends to a Control Unit, housing a Finite State Machine (FSM) and a 4-to-16 Decoder, critical in governing the operations of the ALU Core. As the central processing nexus, the ALU Core executes specific arithmetic and logical operations, guided by a 16-bit microcode from the Control Unit. The subsequent integration of components into a cohesive circuit design is aimed at fostering seamless interaction. The overarching objective is the implementation of diverse problem sets, rigorously assessing and demonstrating the functionality of the intricately crafted processor. This report endeavors to elucidate the procedural steps, VHDL code implementations, and outcomes derived from the empirical exploration within the realm of Digital Systems Lab 6.

3. Components: Latch 1, Latch 2, 4:16 Decoder, FSM

Latch1/Latch2:

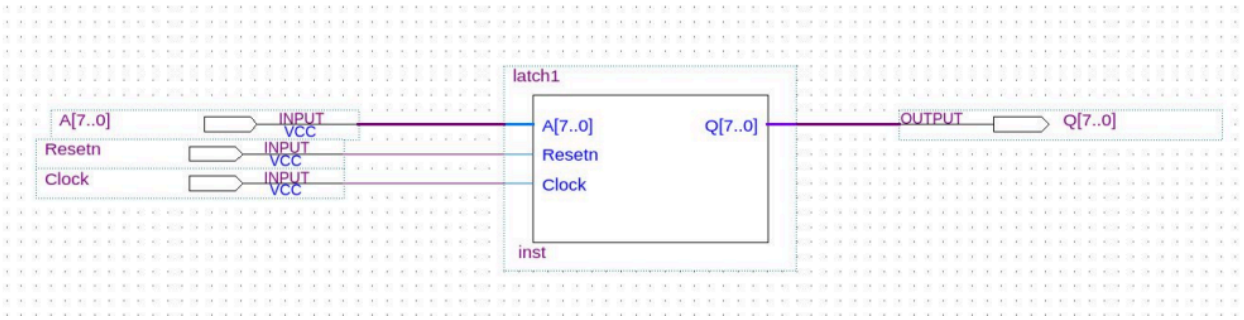
Latch1 and latch2 operate in tandem, capturing and holding 8-bit data on the rising edge of the clock signal. This dual-latch design ensures synchronized data transfer, allowing for efficient and reliable storage of information within the system. The register's fundamental role is to facilitate the orderly passage of input values to subsequent components, contributing to the seamless flow of data in the overall processing unit. The inputs are “A” and “B” for latches 1 and 2 respectively, each holding 2 digits from the last four digits of my student number(501169943).

- $A = (99)_{16} = (10011001)_2$
- $B = (43)_{16} = (01000011)_2$

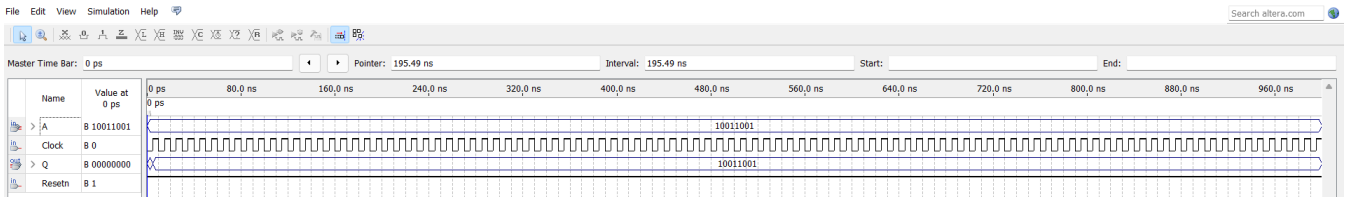
Truth Table:

Input			Output
Clock	Reset	A/B	Q
Falling	0	0	0
Falling	0	1	0
Falling	1	0	0
Falling	1	1	0
Rising	0	0	0
Rising	0	1	0
Rising	1	0	0
Rising	1	1	1

Block Schematic:



Waveform:



4:16 decoder

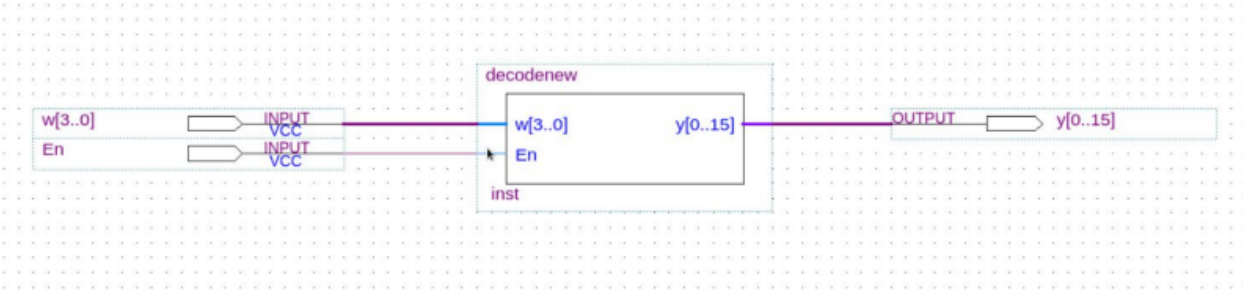
The 4-to-16 decoder plays a pivotal role in the digital system, decoding a 4-bit input signal into one of sixteen possible outputs. Employing a design based on three 3-to-8 decoders, it effectively translates the input signal into a unique combination, selecting a specific output line. This component is integral to the Control Unit, as it determines the operation to be performed by the Arithmetic and Logic Unit (ALU) core based on the microcode received. Its ability to decode and route signals contributes to the precise orchestration of operations within the processing unit, enhancing the overall functionality of the digital system.

Truth Table:

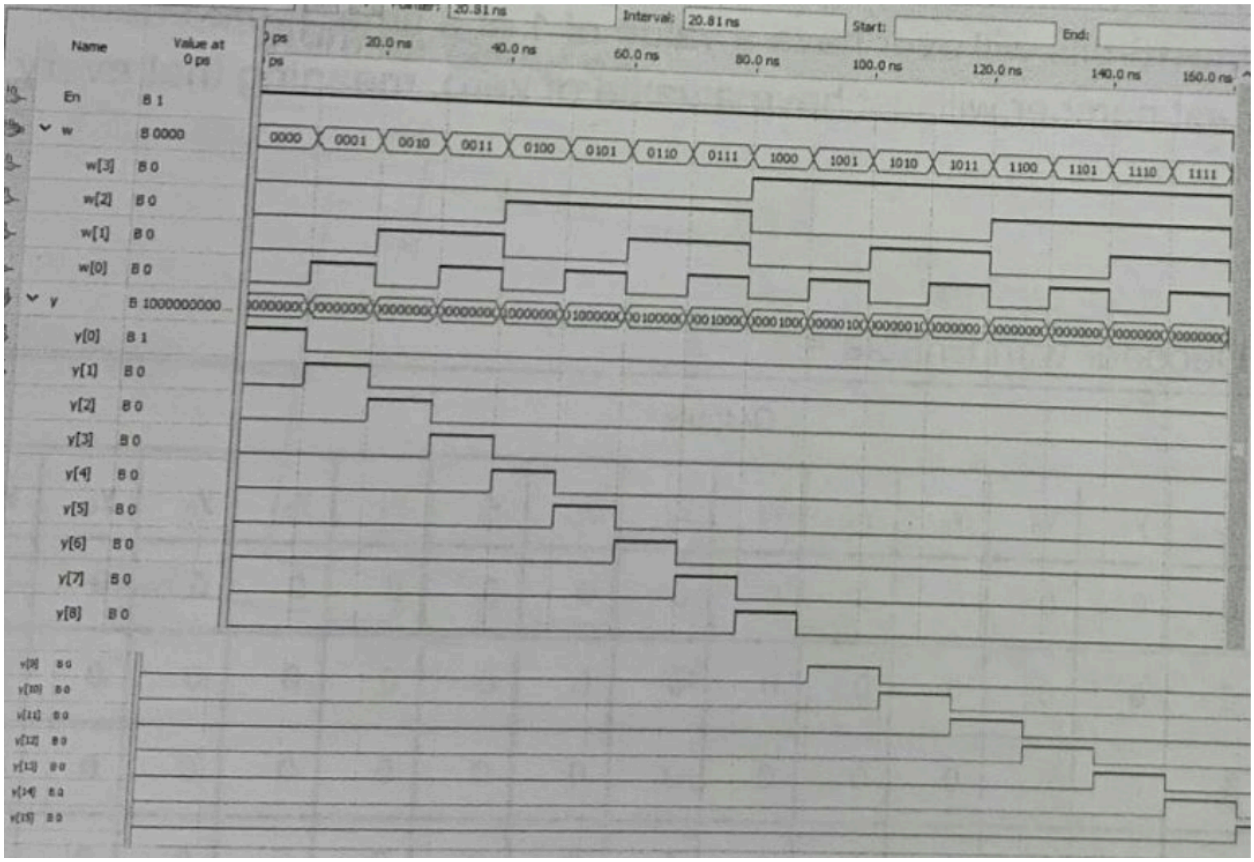
Enable=1

Input: w[3..0]	output: y[0..15]
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

Block Schematic:



Waveform:



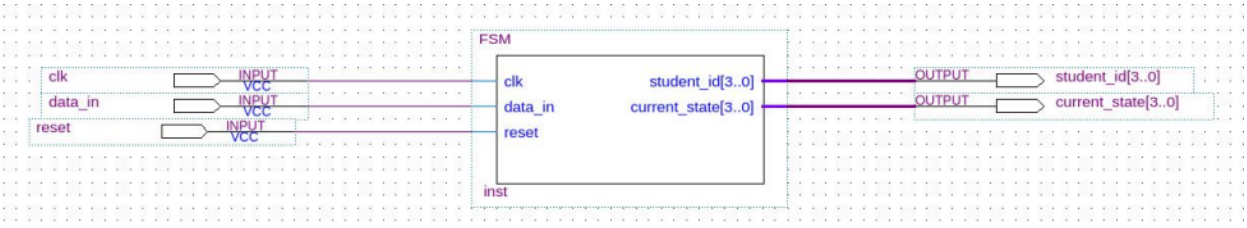
FSM:

The Finite State Machine (FSM) is a vital component within the digital system, functioning as a control unit that dictates the sequence of controller states. Designed with an up-counter mechanism, it cycles through nine different states in response to the clock signal, facilitating sequential transitions. The FSM serves as a crucial element in the Control Unit, with its current state output providing input to the 4-to-16 Decoder. This dynamic interaction enables the precise coordination of microcode delivery to the Arithmetic and Logic Unit (ALU) core, effectively guiding the overall operation of the digital processor.

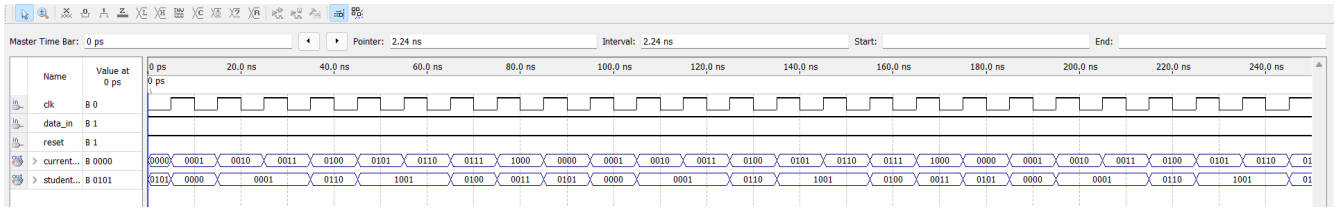
Truth Table:

	Data in = 0		Data in = 1	
State type	Current state	Student ID	Current state	Student ID
S0	0000	0101	0001	0000
S1	0001	0000	0010	0001
S2	0010	0001	0011	0001
S3	0011	0001	0100	0110
S4	0100	0110	0101	1001
S5	0101	1001	0110	1001
S6	0110	1001	0111	0100
S7	0111	0100	1000	0011
S8	1000	0011	0000	0101

Block Schematic:



Waveform:



4. ALU1(Problem set 1):

Problem Set 1 for the Arithmetic and Logic Unit (ALU) involves the initial design of the General Processor Unit (GPU). In this configuration, the Finite State Machine (FSM) output follows an up-counting pattern, influencing the operation selector signal for the ALU core. The resulting ALU core output, termed "Result," reflects the outcomes of operations listed in Table 1. These operations include summation, subtraction, various Boolean operations, and basic arithmetic manipulations. The ALU core, fueled by microcode from the Control Unit, exhibits its prowess in executing these predefined functions, establishing the foundation for subsequent problem sets and comprehensive testing of the digital processor's capabilities.

Block Schematic:

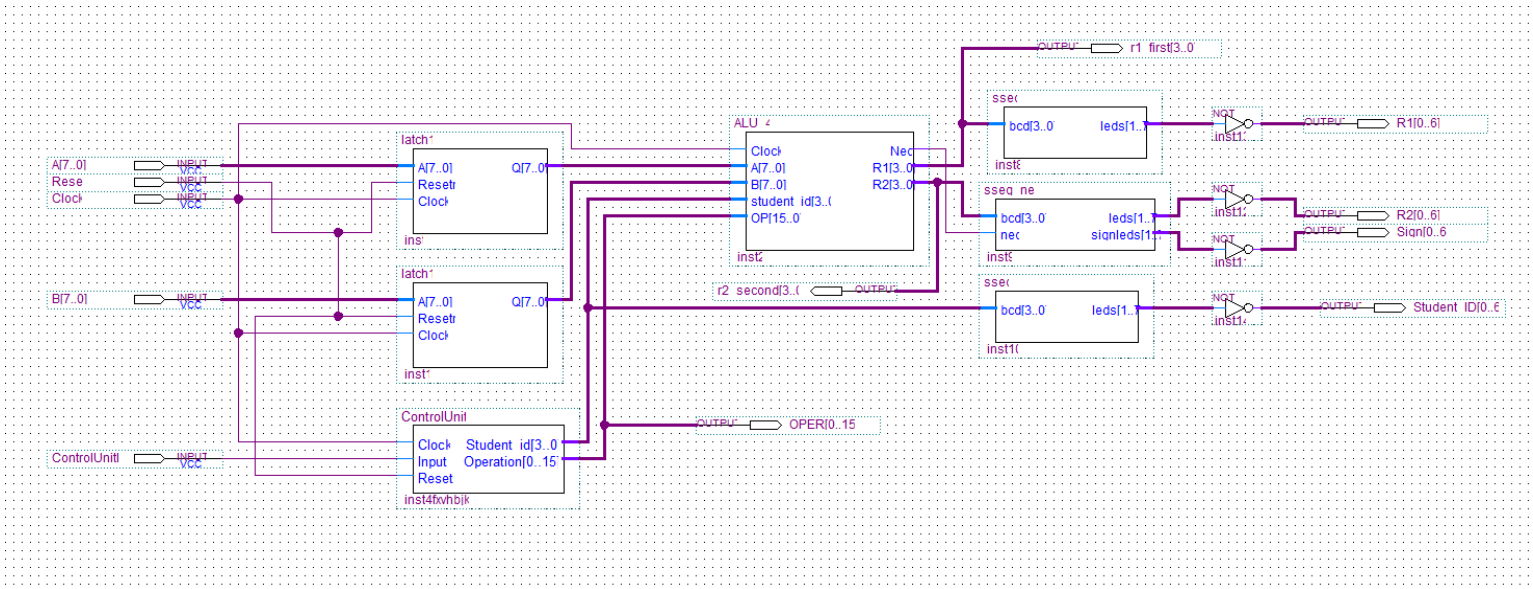
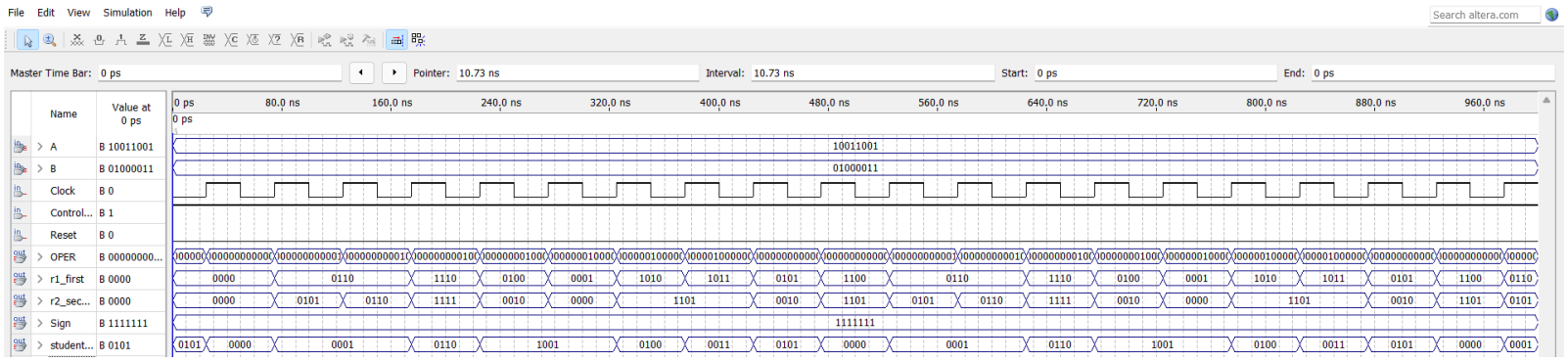


Table of Microcodes:

Function	Microcode	Operation
1	1000000000000000	sum(A, B)
2	0100000000000000	diff(A, B)
3	0010000000000000	NOT(A)
4	0001000000000000	NOT(A • B)
5	0000100000000000	NOT(A + B)
6	0000010000000000	A • B
7	0000001000000000	A ⊕ B
8	0000000100000000	A + B
9	0000000010000000	NOT(A ⊕ B)

Waveform:



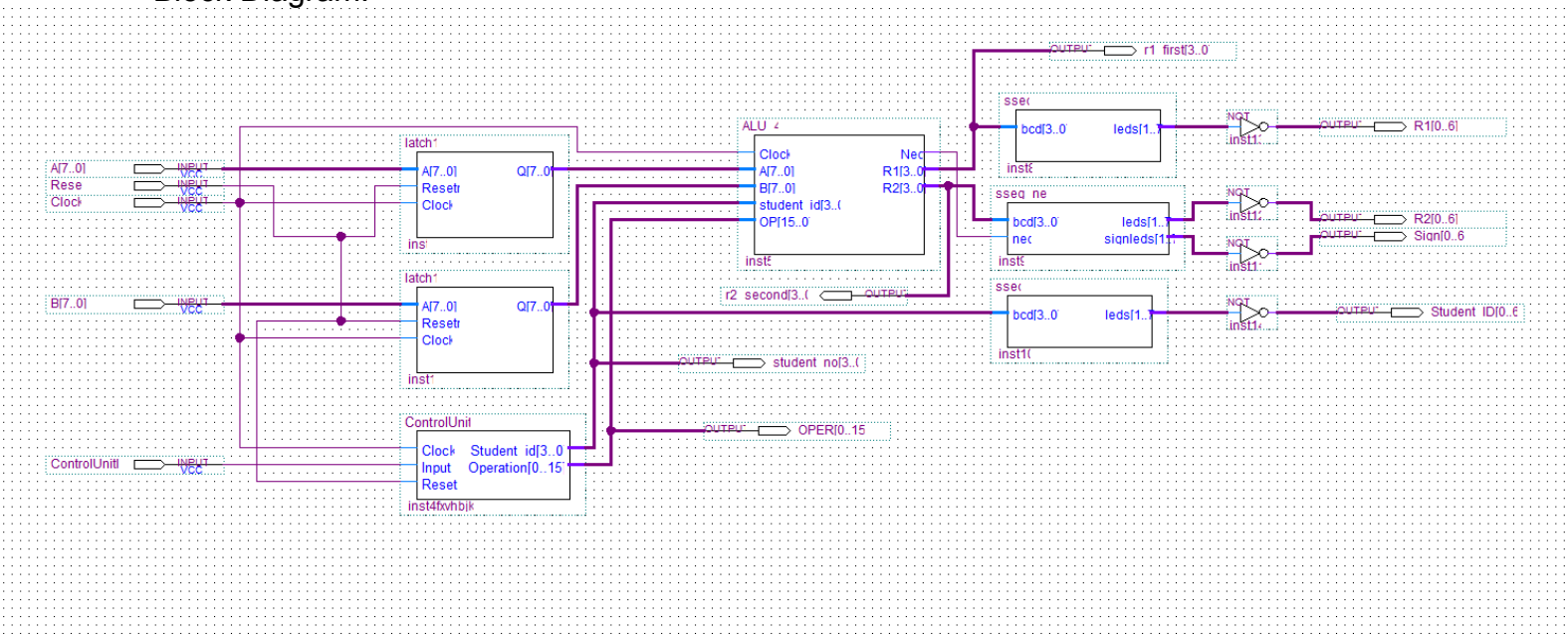
5. ALU2(Problem set 2):

Problem Set 2 for the Arithmetic and Logic Unit (ALU) involves modifying the ALU core to accommodate specific functionalities. We were assigned **set a)** as one of the specified modifications, which ranged from shifting and rotating operations to bitwise manipulations and arithmetic alterations. For instance, tasks include incrementing values, shifting bits, finding minimum and maximum values, as well as performing logical operations.

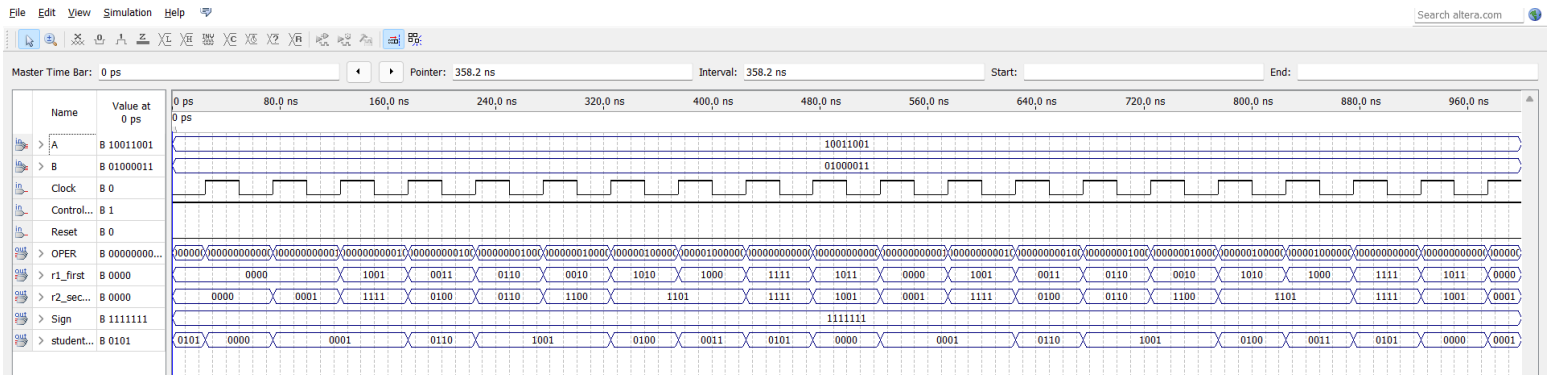
Table of Microcodes:

Function	Microcode	Operation
1	1000000000000000	Increment A by 2
2	0100000000000000	Shift B to right by two bits, input bit = 0 (SHR)
3	0010000000000000	Shift A to right by four bits, input bit = 1 (SHR)
4	0001000000000000	Find the smaller value of A and B and produce the results (Min(A,B))
5	0000100000000000	Rotate A to right by two bits (ROR)
6	0000010000000000	Invert the bit-significance order of B
7	0000001000000000	Produce the result of XORing A and B
8	0000000100000000	Produce the summation of A and B, then decrease it by 4
9	0000000010000000	Produce all high bits on the output

Block Diagram:



Waveform:



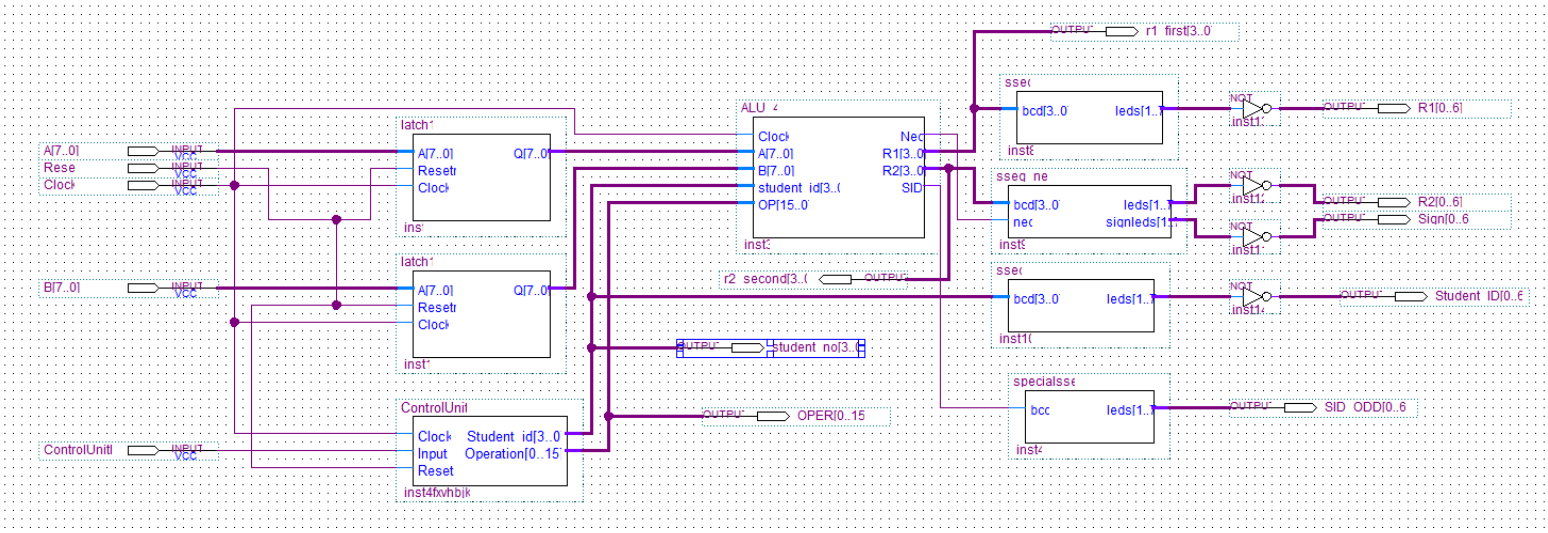
6. ALU3(Problem set 3):

Problem Set 3 for the Arithmetic and Logic Unit (ALU) focuses on modifying the Control Unit, specifically the Finite State Machine (FSM). We were assigned **set a)** as one of the specified tasks related to utilizing the FSM output, denoted as "student_id." The modifications involve processing the FSM output for various conditions, including checking for odd or even values, evaluating parity, and comparing digits between inputs A and B. Each student is tasked with implementing these conditions using the ALU and microcode from the initial design section.

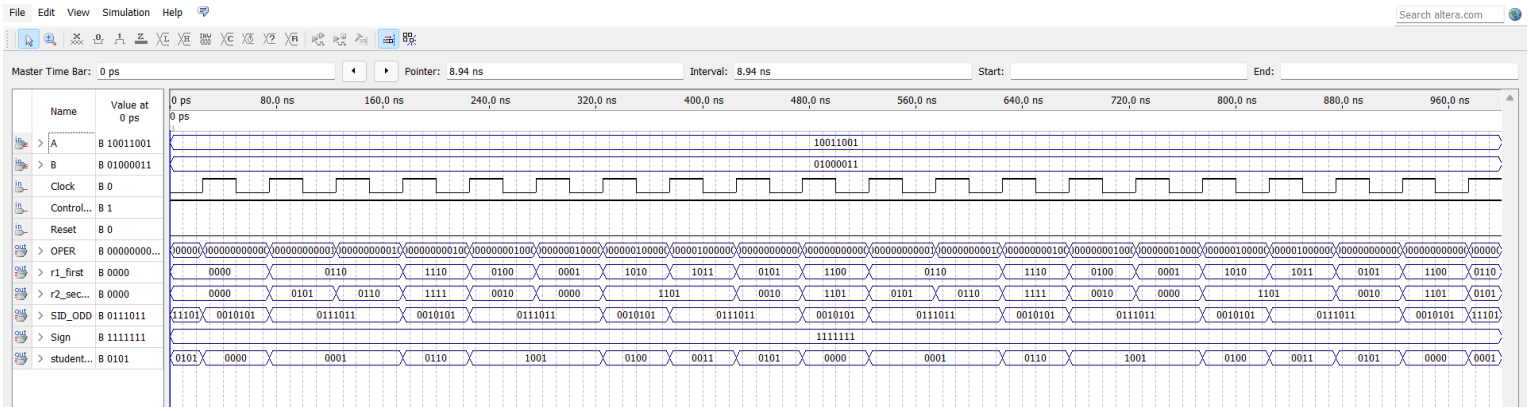
Table of Microcodes:

Function	Microcode	Operation
1	1000000000000000	Find if A is odd. If it is, output y. Otherwise, output n.
2	0100000000000000	Find if A is odd. If it is, output y. Otherwise, output n.
3	0010000000000000	Find if A is odd. If it is, output y. Otherwise, output n.
4	0001000000000000	Find if A is odd. If it is, output y. Otherwise, output n.
5	0000100000000000	Find if A is odd. If it is, output y. Otherwise, output n.
6	0000010000000000	Find if A is odd. If it is, output y. Otherwise, output n.
7	0000001000000000	Find if A is odd. If it is, output y. Otherwise, output n.
8	0000000100000000	Find if A is odd. If it is, output y. Otherwise, output n.
9	0000000010000000	Find if A is odd. If it is, output y. Otherwise, output n.

Block Diagram:



Waveform:



7. Conclusion:

In summary, Digital Systems Lab 6 proved to be an enriching experience in the design and implementation of a Simple General-Purpose Processor. The lab's structured approach guided us through the creation of an efficient Arithmetic and Logic Unit (ALU), integrating components such as registers and a Control Unit. The successful synthesis of a final circuit design demonstrated the ALU's versatility, validated through the implementation of diverse problem sets. The collaborative effort emphasized the importance of teamwork in complex engineering projects. This hands-on exploration deepened our understanding of digital system design, honed VHDL coding skills, and provided a practical application of theoretical concepts. The acquired knowledge positions us well for future endeavors in electrical and computer engineering.

References

Department of Electrical and Computer Engineering. (2023).

Lab 6 Design of a Simple General-Purpose Processor. Coe328 home page. https://www.ecb.torontomu.ca/~courses/coe328/