

COE 328 Lab 6 Report

Name: Kirolos Youssef

Student ID: 500968175

Class: COE 328

TA Name: Arghavan Asad

Section Number: 04

1. Cover Page

2. Table of Contents

- ♣ Include all the chapters/Sections with Page Numbers.

3. Introduction

- ♣ Description of the components in not more than 10 sentences.

4. Components: Latch1, Latch2, 4:16 Decoder, FSM

- ♣ Description of the components in not more than 5 sentences.
- ♣ Truth Tables
- ♣ Circuit Diagram and/or Block Diagram as required for the component. ♣
- Waveform File for each component.

5. ALU_1 for Problem Set 1 of the Lab 6 procedure

- ♣ Description outlining purpose and design of the component.
- ♣ Screen shot of the Block schematic file BDF (either hand written or using software).
- ♣ Mention the purpose of all inputs and outputs of this component.
- ♣ Table of Microcode's generated by decoder for ALU_1.
- ♣ Complete Waveform File (using FSM student ID).

6. ALU_2 for Problem Set 2 of the Lab6 procedure

- ♣ Description outlining purpose and design of the component. The
- ♣ Screen shot of the Block schematic file BDF (either hand written or using software).
- ♣ Mention the purpose of all inputs and outputs of this component.
- ♣ Table of Microcode's generated by decoder for ALU_2.
- ♣ Complete Waveform File (using FSM student ID).

7. ALU_3 for Problem Set 3 of the Lab6 procedure

- ♣ Description outlining purpose and design of the component.
- ♣ Screen shot of the Block schematic file BDF (either hand written or using software).
- ♣ Mention the purpose of all inputs and outputs of this component.
- ♣ Table of Microcode's generated by decoder for ALU_3.
- ♣ Complete Waveform File (using FSM student ID).

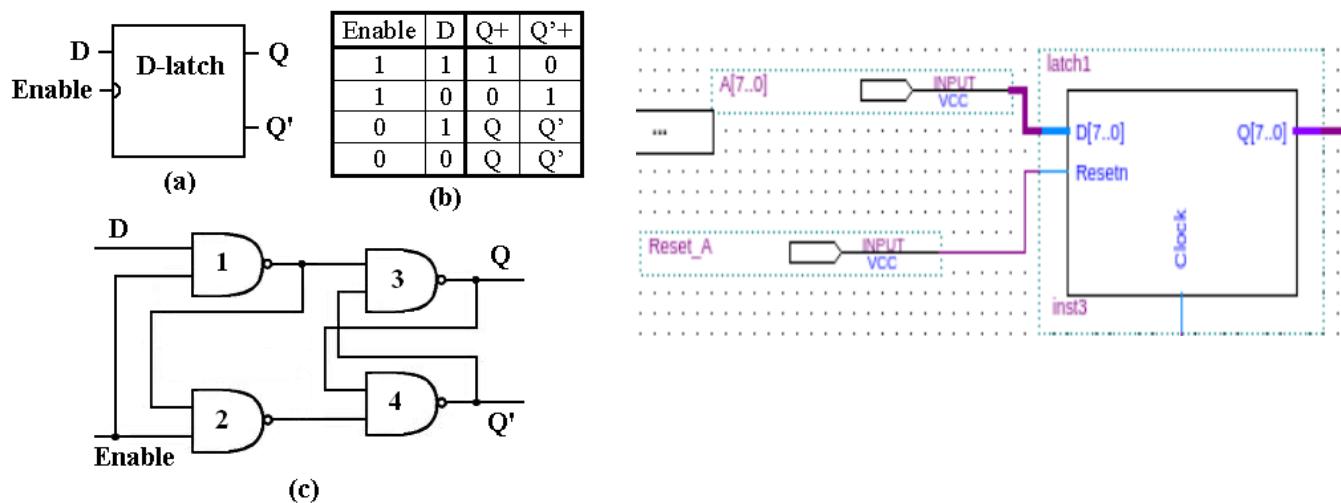
8. Conclusion

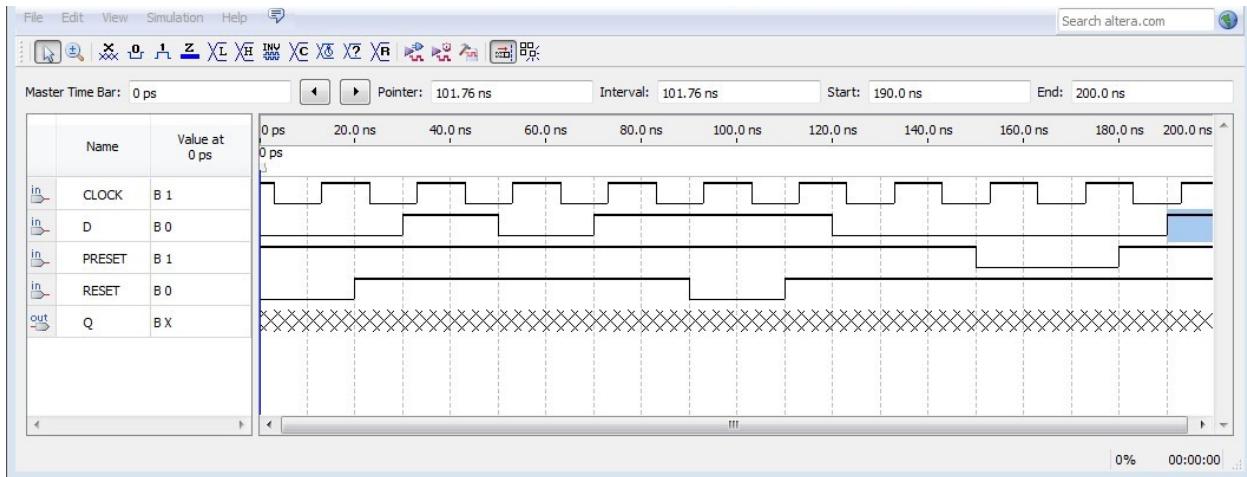
Introduction

Lab 6 combined all the smaller components we have worked with and learned about into 1 big chip that can perform multiple functions. There were 4 main elements in the overall design, the storage elements, the combinational logic elements, the control elements and the output display elements. The storage elements include the latches used to store and vary the input of the combinational logic element which in this instance was the ALU (Arithmetic Logic Unit). The controller unit which consisted of a Finite State Machine and a 4-16 decoder also sent an input to the ALU which acted like a function selector. Finally, the ALU relayed the output of the function to the output element which was the Seven Segment Display.

Components

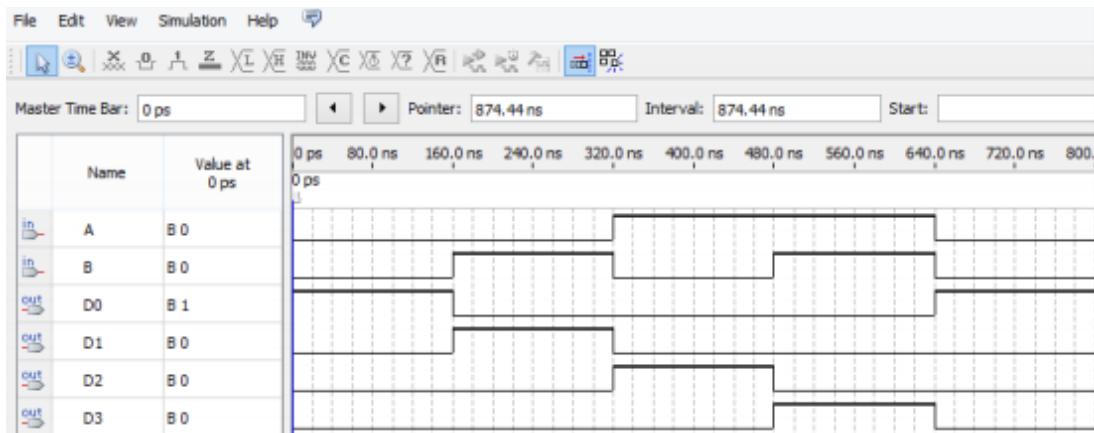
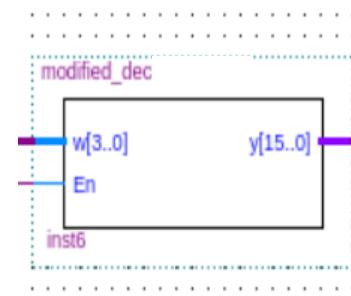
Latch: A latch is a storage element that stores a set binary value and updates that value with a clock trigger which in our case was a rising edge trigger.





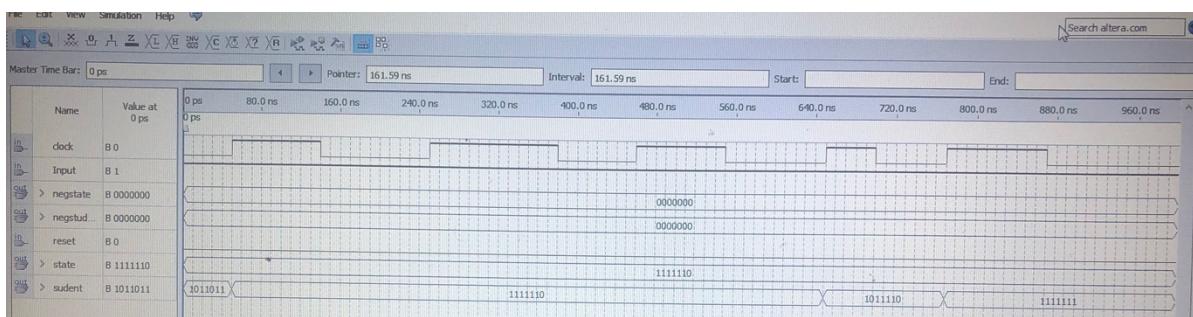
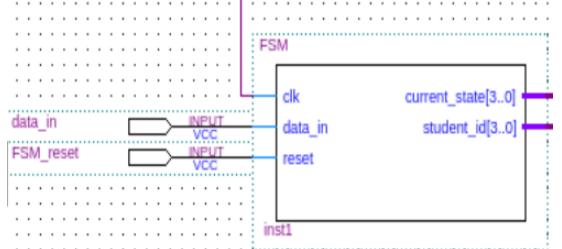
4:16 Decoder: A Decoder is a component that takes n number of inputs and using their values decodes the signal into 2^n number of outputs which acts like a selector. In this lab we required a 4 to 16 Decoder

INPUTS				OUTPUTS															
A	B	C	D	X ₀	X ₁	X ₂	X ₃	X ₄	X ₅	X ₆	X ₇	X ₈	X ₉	X ₁₀	X ₁₁	X ₁₂	X ₁₃	X ₁₄	X ₁₅
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	
0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	
0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	
0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	
0	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	
1	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	
1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	
1	0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	
1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	
1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	
1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	



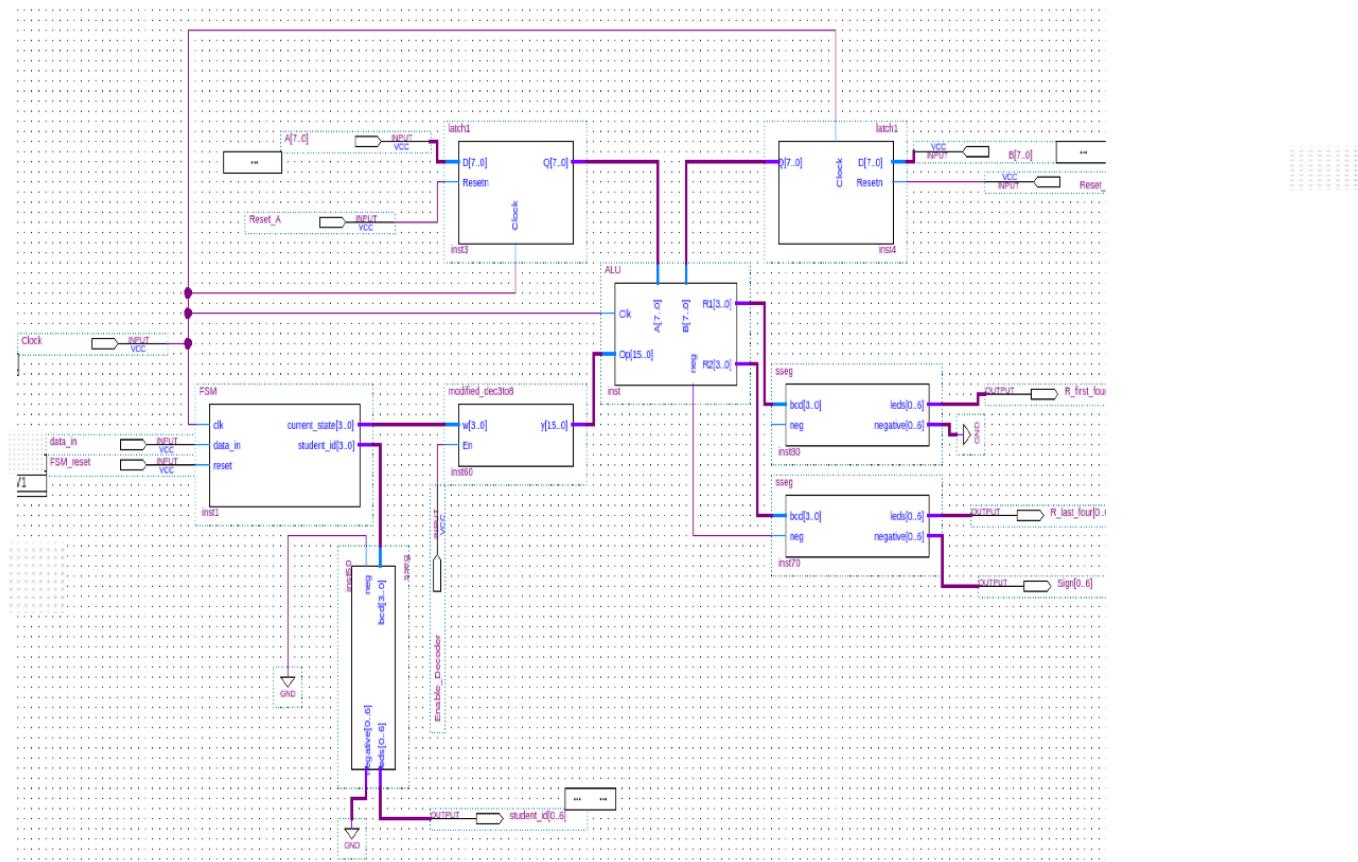
FSM: A Finite State Machine is a component that uses an input and a present state to determine the next state and the output. Throughout the lab the FSM was interchanged between Mealy and Moore Machines.

Current State		Input I	Next State		Outputs Y	Flip Flop Inputs	
A	B		A _{next}	B _{next}		D _A	D _B
0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	1
0	1	0	0	0	1	0	0
0	1	1	1	0	1	1	0
1	0	0	0	0	0	0	0
1	0	1	1	0	0	1	0
1	1	0	X	X	X	X	X
1	1	1	X	X	X	X	X



ALU 1 for Problem Set 1 of the Lab 6 procedure

ALU 1: The purpose of ALU1 is to receive 2 inputs A and B and pick a certain function from the list of microcodes and apply that operation to A and B and relay it to the Seven Segment Display.



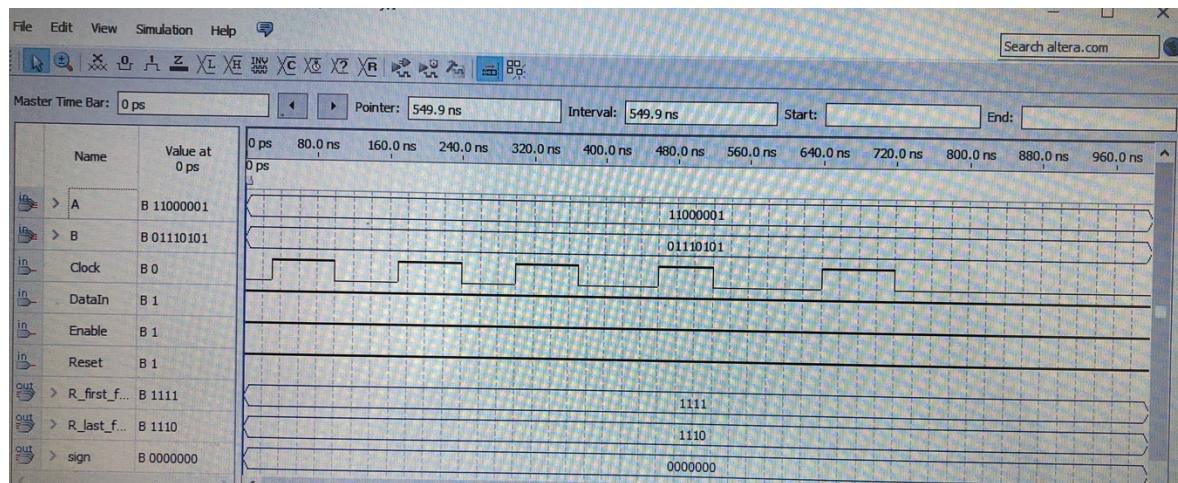
Inputs:

- **Clock:** The clock that updates the value with every rising edge
 - **OP:** The Function Selector
 - **A:** Binary Input Value
 - **B:** Binary Input Value

Outputs:

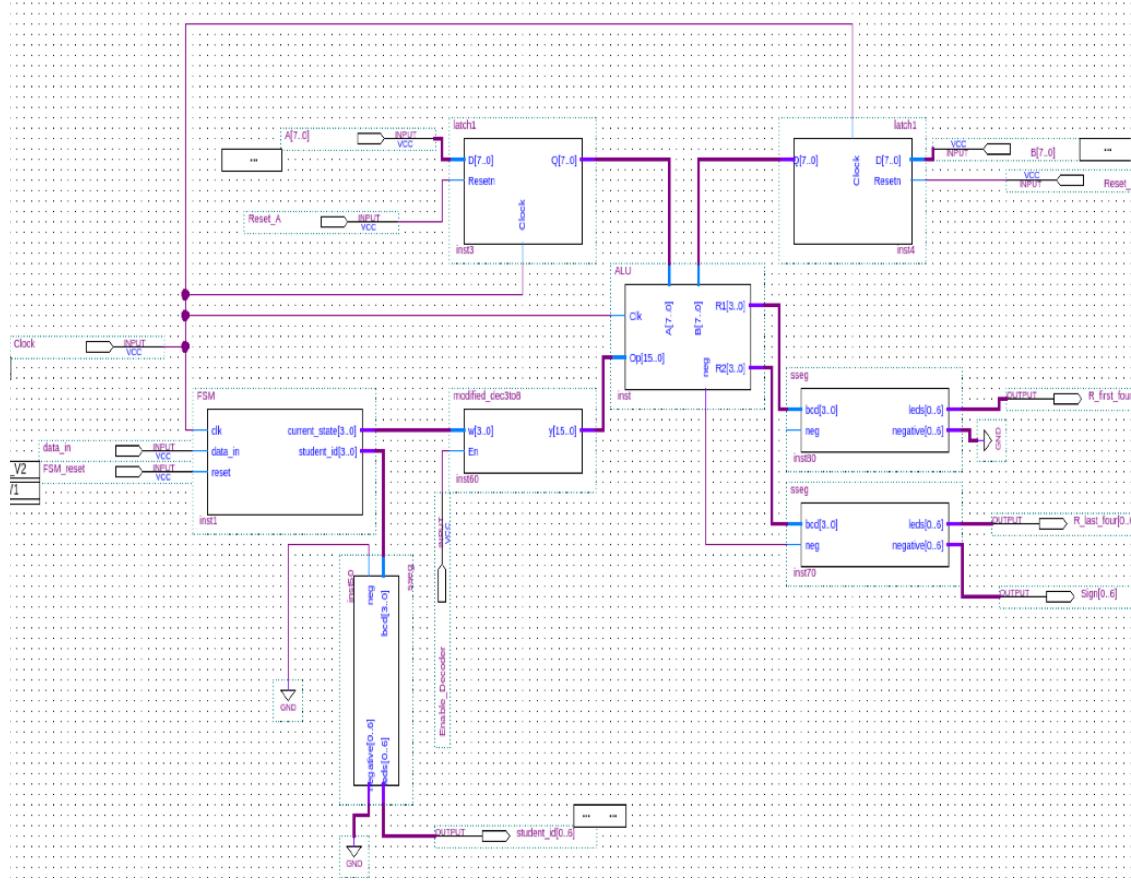
- R1: First 4 Binary Values of Output
 - R2: Second 4 Binary Values of Output
 - Neg: Only on when the Result is negative

Function #	Microcode	Boolean Operation / Function
1	0000000000000001	sum(A, B)
2	0000000000000010	diff(A, B)
3	00000000000000100	\bar{A}
4	0000000000001000	$\overline{A \cdot B}$
5	00000000000010000	$\overline{A + B}$
6	00000000000100000	$A \cdot B$
7	00000000001000000	$A \oplus B$
8	00000000010000000	$A + B$
9	0000000100000000	$\overline{A \oplus B}$



ALU 2 for Problem Set 2 of the Lab6 procedure

ALU 2: The purpose of ALU2 is to receive 2 inputs A and B and pick a certain function using a certain microcode and apply that operation to A and B and relay it to the Seven Segment Display.



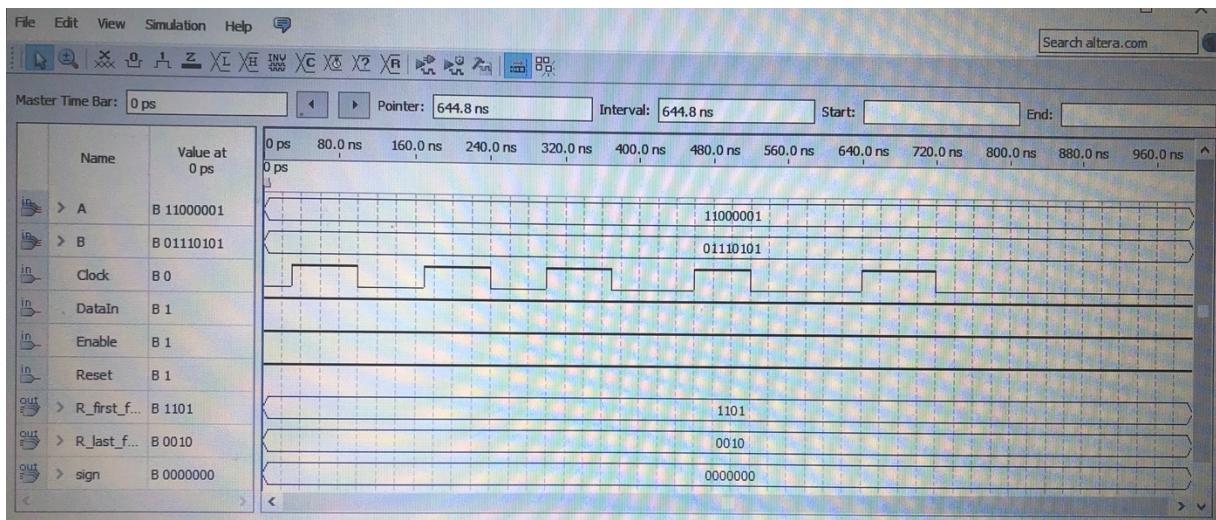
Inputs:

- **Clock:** The clock that updates the value with every rising edge
- **OP:** The Function Selector
- **A:** Binary Input Value
- **B:** Binary Input Value

Outputs:

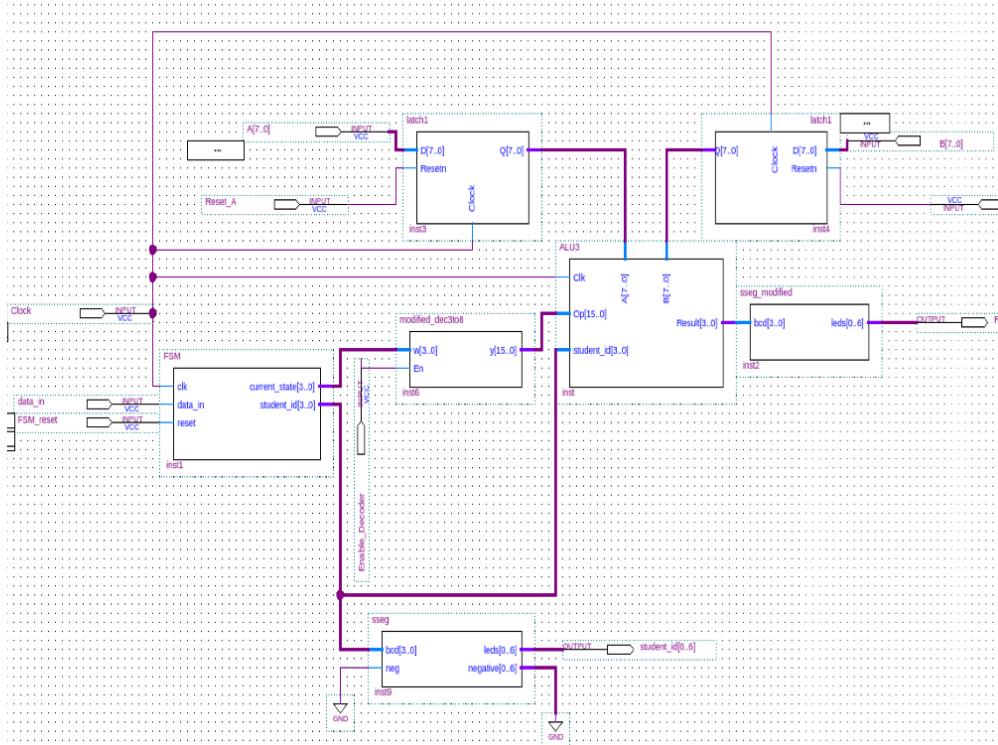
- **R1:** First 4 Binary Values of Output
- **R2:** Second 4 Binary Values of Output
- **Neg:** Only on when the Result is negative

Microcode	Function #	Operation / Function
0000000000000001	1	Increment A by 2
0000000000000010	2	Shift B to right by two bits, input bit = 0 (SHR)
0000000000000100	3	Shift A to right by four bits, input bit = 1 (SHR)
00000000000001000	4	Find the smaller value of A and B and produce the results (Min(A,B))
00000000000010000	5	Rotate A to right by two bits (ROR)
0000000000100000	6	Invert the bit-significance order of B
0000000001000000	7	Produce the result of XORing A and B
0000000010000000	8	Produce the summation of A and B , then decrease it by 4
0000000100000000	9	Produce all high bits on the output



ALU 3 for Problem Set 3 of the Lab6 procedure

ALU 3: The purpose of ALU3 is to compare the FSM output (student_id) with the values of A and B, if a certain parameter is true relay a 'Y' to the Seven Segment Display otherwise relay a 'N'.



Inputs:

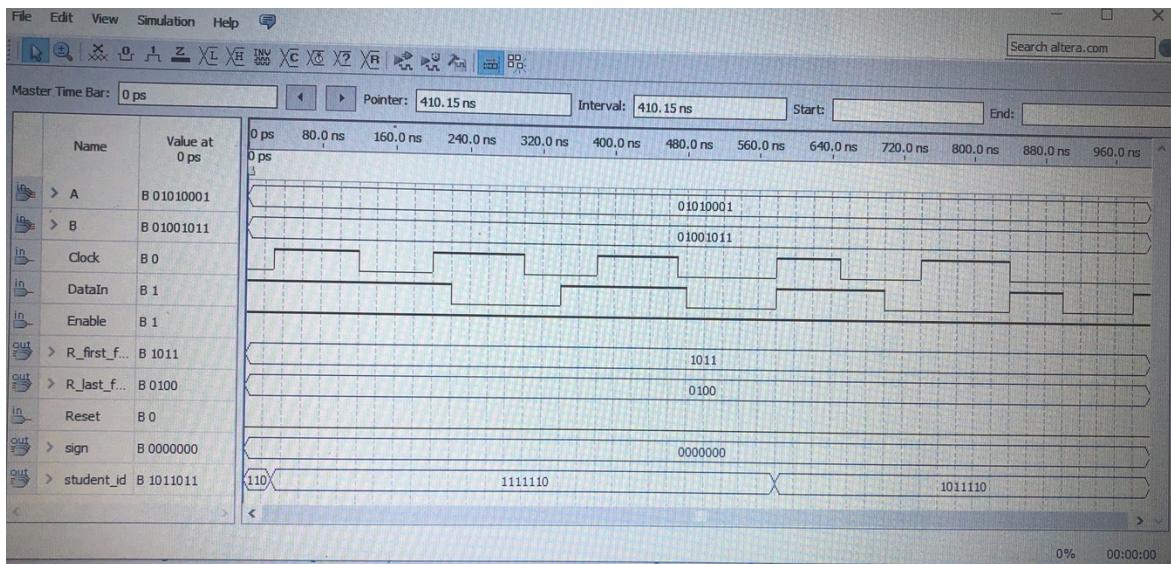
- **Clock:** The clock that updates the value with every rising edge
- **OP:** The Function Selector
- **Student Id:** Binary Value of Student ID
- **A:** Binary Input Value
- **B:** Binary Input Value

Outputs:

- **Result:** 'Y' or 'N'

Microcode
0000000000000001
0000000000000010
00000000000000100
000000000000001000
00000000000010000
000000000000000000
0000000000000000000
00000000000000000000

- a) For each microcode instruction, display 'y' if the FSM output (**student_id**) is odd and 'n' otherwise
- b) For each microcode instruction, display 'y' if the FSM output (**student_id**) is even and 'n' otherwise
- c) For each microcode instruction, display 'y' if the FSM output (**student_id**) had an odd parity and 'n' otherwise
- d) For each microcode instruction, display 'y' if the FSM output (**student_id**) had an even parity and 'n' otherwise
- e) For each microcode instruction, display 'y' if one of the 2 digits of A are greater than FSM output (**student_id**) and 'n' otherwise. Use the microcode instruction from part 1 of the lab.
- f) For each microcode instruction, 'y' if one of the 2 digits of A are less than FSM output (**student_id**) and 'n' otherwise. Use the microcode instruction from part 1 of the lab.
- g) For each microcode instruction, 'y' if one of the 2 digits of A are equal to FSM output (**student_id**) and 'n' otherwise. Use the microcode instruction from part 1 of the lab.
- h) For each microcode instruction, display 'y' if one of the 2 digits of B are greater than FSM output (**student_id**) and 'n' otherwise. Use the microcode instruction from part 1 of the lab.
- i) For each microcode instruction, 'y' if one of the 2 digits of B are less than FSM output (**student_id**) and 'n' otherwise. Use the microcode instruction from part 1 of the lab.
- j) For each microcode instruction, 'y' if one of the 2 digits of B are equal to FSM output (**student_id**) and 'n' otherwise. Use the microcode instruction from part 1 of the lab.



Conclusion

This lab was the most complex and practical one yet because the design acts like a very basic computer chip with a storage unit, a controller unit, a combinational logic unit, and finally an output display unit. Combining all these smaller seemingly insignificant components and allowing them to relay signals with one another results in a very useful chip that can be programmed to perform any function. This design is the very foundation of computing chips and can be altered in functionality and complexity to perform different operations.