Lab 6 - Design of a Simple Central Processing Unit Report

COE328

Professor: Dr. Sedaghat Reza

TA: Menglu Li

Section: 12

Student Name: Kevin Karunaratna

Student ID #: 501242608

Submitted: December 12, 2024

Due: December 12, 2024

Page Count : 11

Table of Contents

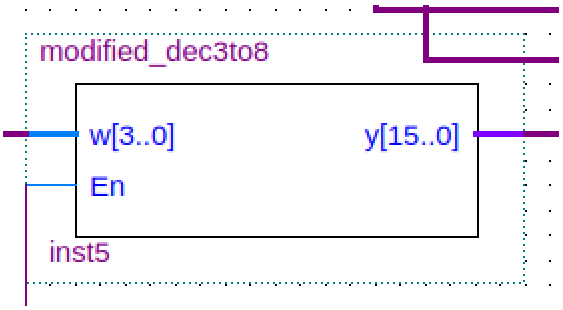
Introduction

This lab required several components to be created and integrated into the circuit for its 3 part problem set.The ASU is a component that completes operations based on several input conditions. Those input conditions are: A and B 8bit data sets, a clock, a reset, and an opcode input which determines the index for the operations. The output is the result of the calculations. The 4 to 16 decoder is a component that takes the last 4 digits of student number and outputs opcodes for the ASU to take as input. The latch component acts as a storage device for the inputs where they can be enabled and clocked to control the output to the ASU. The FSM (finite state machine) is a component that consists of states that it transitions between whenever it is clocked. It is used to control the behavior of a system. When all of these components are combined, a central processing unit is created where tasks can be performed based on given data and conditions.

Components

<u>4 to 16 Decoder</u>

The 4-16 Decoder takes 4 input values, and an enable signal to produce 16 unique values of output. In this application, the decoder takes the last 4 digits of my student number and outputs the signals as an opcode for the ASU. It acts as a converter from the student number digits to the interpretable opcodes for operation selection.

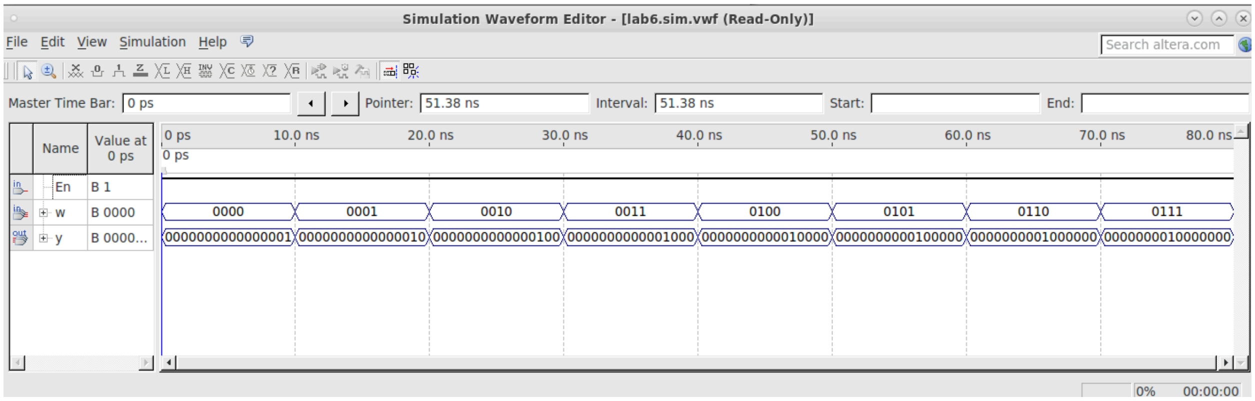


Truth Table

E	Inputs				Outputs															
1	0	0	0	0	0	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	0	0	1	0
1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

1	0	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
1	1	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Waveform



Simulation Waveform Editor - [lab6.sim.vwf (Read-Only)]

File Edit View Simulation Help

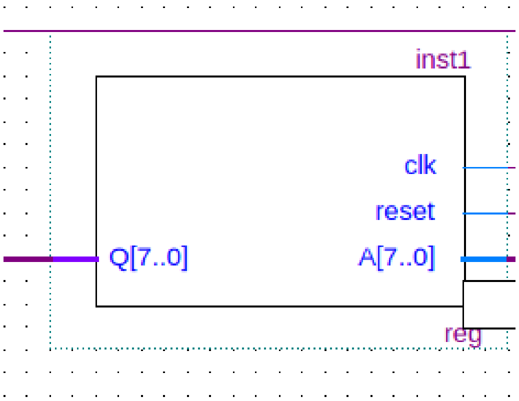
Search altera.com

Master Time Bar: 0 ps Pointer: 51.38 ns Interval: 51.38 ns Start: End:

Name	Value at 0 ps	0 ps	10.0 ns	20.0 ns	30.0 ns	40.0 ns	50.0 ns	60.0 ns	70.0 ns	80.0 ns
En	B 1									
w	B 0000	0000	0001	0010	0011	0100	0101	0110	0111	
y	B 0000...	0000000000000001	0000000000000010	0000000000000100	0000000000001000	0000000000010000	0000000000100000	0000000001000000	0000000010000000	

0% 00:00:00

Latch



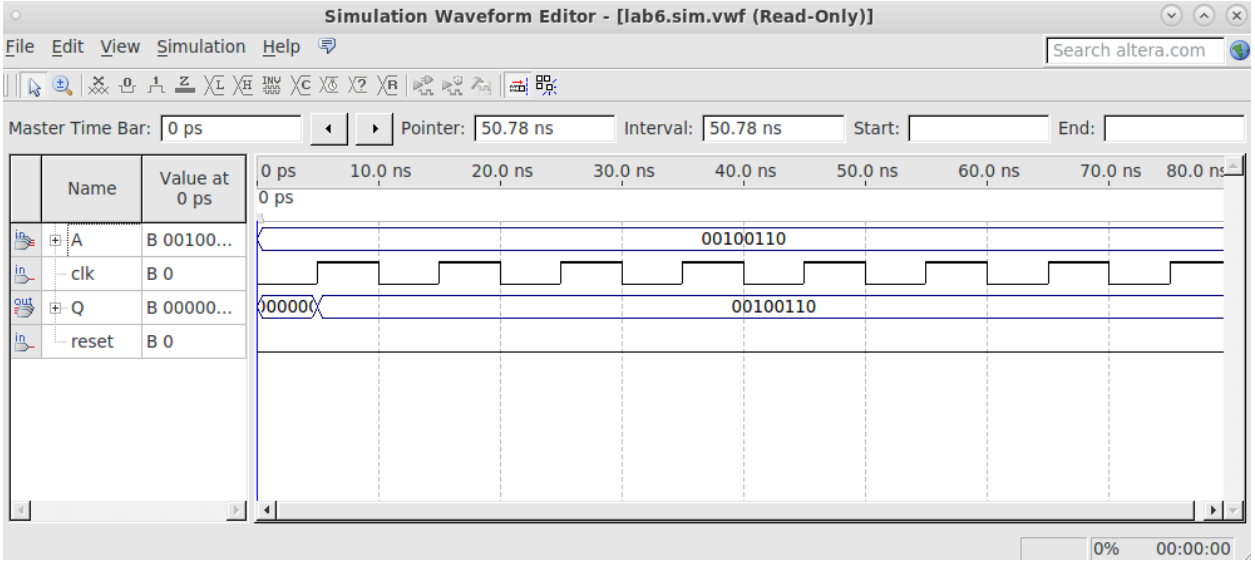
The purpose of the latch is to act as a storage element as it retains the data between clock cycles. Besides the main 8bit data input, the latch also has an input for clock and reset. The output is 0 if there is no clock event. If there is a clock event, it sends the input to the ASU. This application contains 2 latch components.

Truth Table

Enable	Clock	Input	Output	Desc.

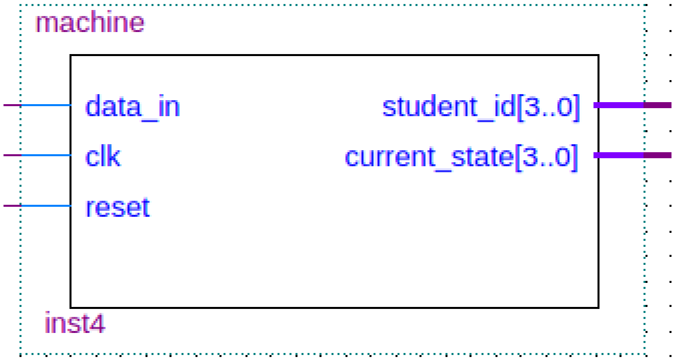
0	X	X	00000000	Device not on.
1	-	D1D2D3…	00000000	Output 00000000 if no clock.
1	1	D1D2D3…	D1D2D3…	Output the input D1D2D3… if clock event.

Waveform



FSM

The FSM (Finite State Machine) is a component which consists of states and outputs based on those states which change when clocked. In this application, the inputs are the data_in, clock, and high active reset. The outputs are the student ID and the current state.

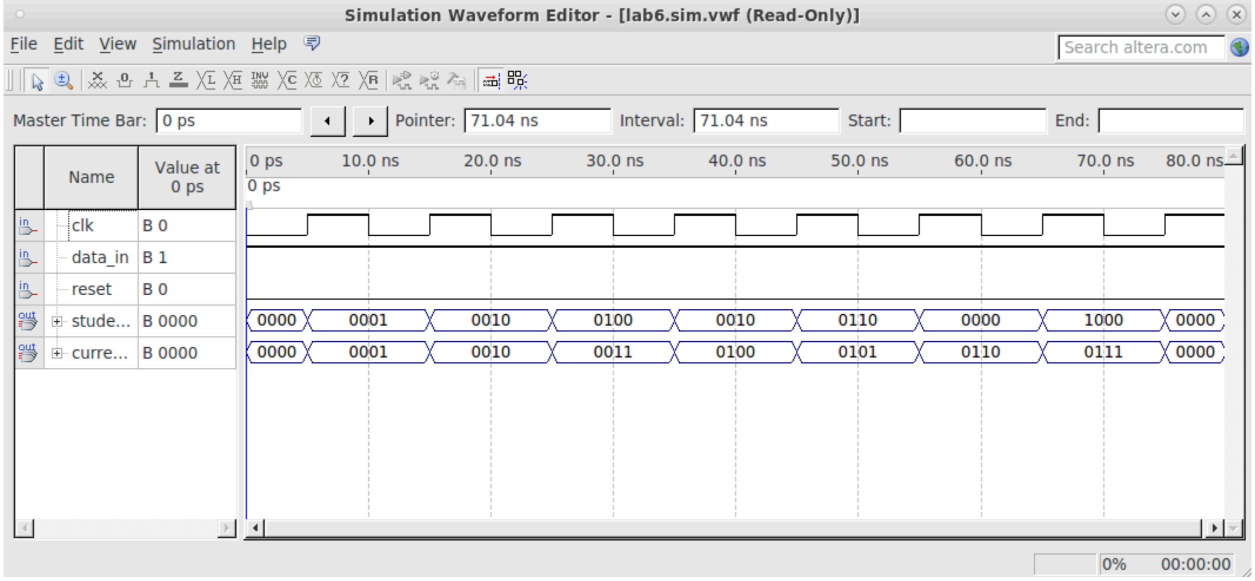


Truth Table (State table)

Present State	Next State		Output
	w = 0	w = 1	
S_0	S_0	S_1	0000

S_1	S_1	S_3	0001
S_3	S_3	S_5	0010
S_5	S_5	S_7	0100
S_7	S_7	S_2	0010
S_2	S_2	S_4	0110
S_4	S_4	S_6	0000
S_6	S_6	S_0	1000

Waveform



Simulation Waveform Editor - [lab6.sim.vwf (Read-Only)]

Master Time Bar: 0 ps   Pointer: 71.04 ns   Interval: 71.04 ns   Start: End:

Name	Value at 0 ps
clk	B 0
data_in	B 1
reset	B 0
stude...	B 0000
curre...	B 0000

```
stude...  0000 0001 0010 0100 0010 0110 0000 1000 0000
curre...  0000 0001 0010 0011 0100 0101 0110 0111 0000
```

ALU 1

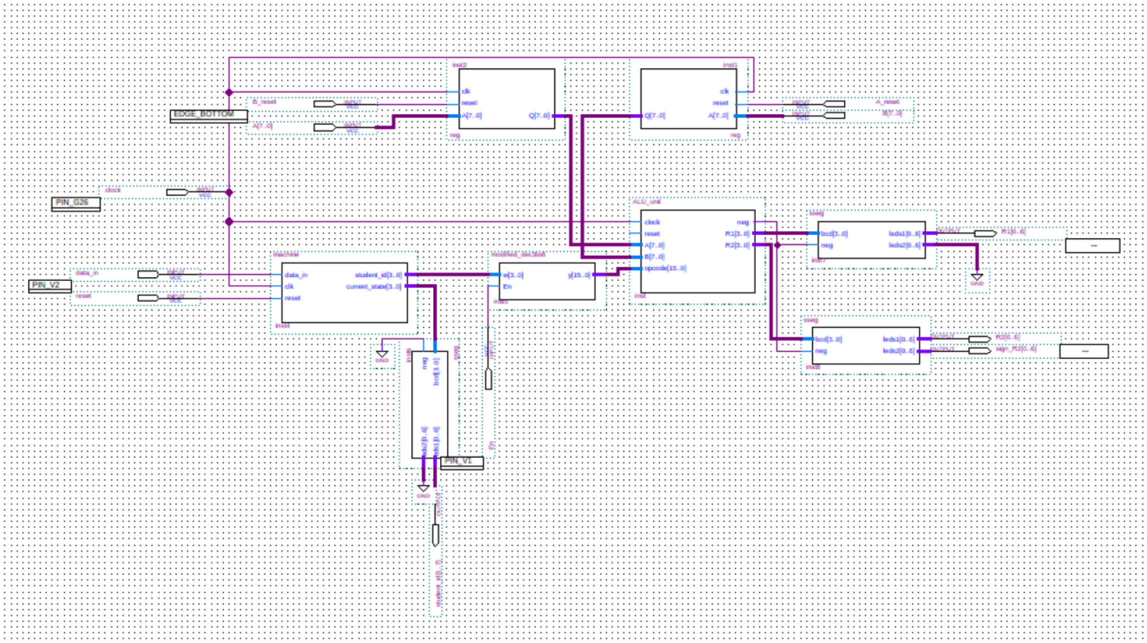
The purpose of this ALU to produce outputs based on given data and opcodes. The inputs are two 8bit data sets containing the last 4 digits of my student number (2608), the opcodes coming from the decoder, a clocker, and a low active reset. The opcodes determine which operations are performed within the ASU. The outputs are the results of these operations.

Opcode Table

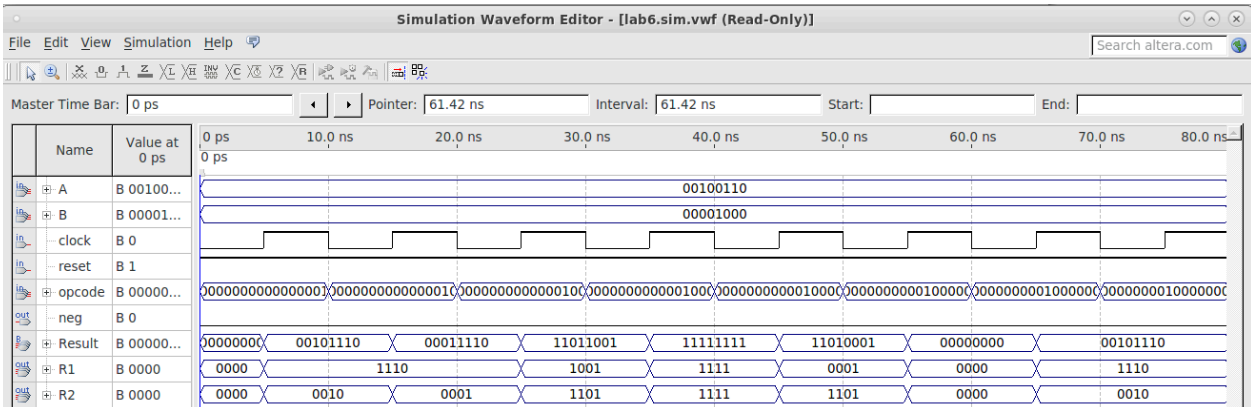
Function #	Opcode	Function
1	00000001	sum(A,B)

2	00000010	dif(A,B)
3	00000100	Ā
4	00001000	(A•B)!
5	00010000	(A+B)!
6	00100000	A•B
7	01000000	A⊕B
8	10000000	A+B

ALU 1 Block Diagram



Waveform



Simulation Waveform Editor - [lab6.sim.vwf (Read-Only)]

Master Time Bar: 0 ps
Pointer: 61.42 ns
Interval: 61.42 ns
Start:
End:

Name	Value at 0 ps	0 ps	10.0 ns	20.0 ns	30.0 ns	40.0 ns	50.0 ns	60.0 ns	70.0 ns	80.0 ns
A	B 00100...	00100110								
B	B 00001...	00001000								
clock	B 0									
reset	B 1									
opcode	B 00000...	000000000000001	000000000000001	00000000000010	00000000000100	000000000001000	00000000010000	000000001000000	000000010000000	
neg	B 0									
Result	B 00000...	0000000	00101110	00011110	11011001	11111111	11010001	00000000	00101110	
R1	B 0000	0000	1110		1001	1111	0001	0000	1110	
R2	B 0000	0000	0010	0001	1101	1111	1101	0000	0010	

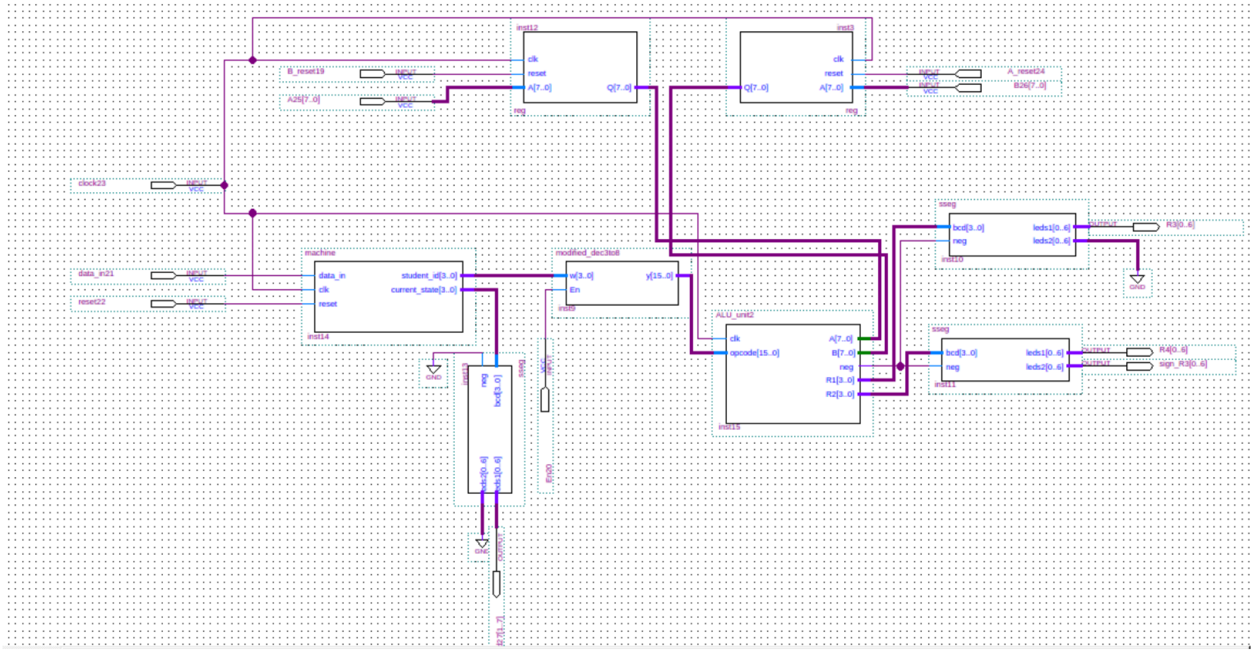
ALU 2

Like the previous ALU, the inputs are the two 8bit data sets for the last 4 digits of my student number, the opcodes, the clock, and the low active reset. Again, the opcodes determine the operations used. The results are output in separate 4bit sets as well as a separate ‘Result’. The difference here is that the opcodes point to different operations but the main idea is the same.

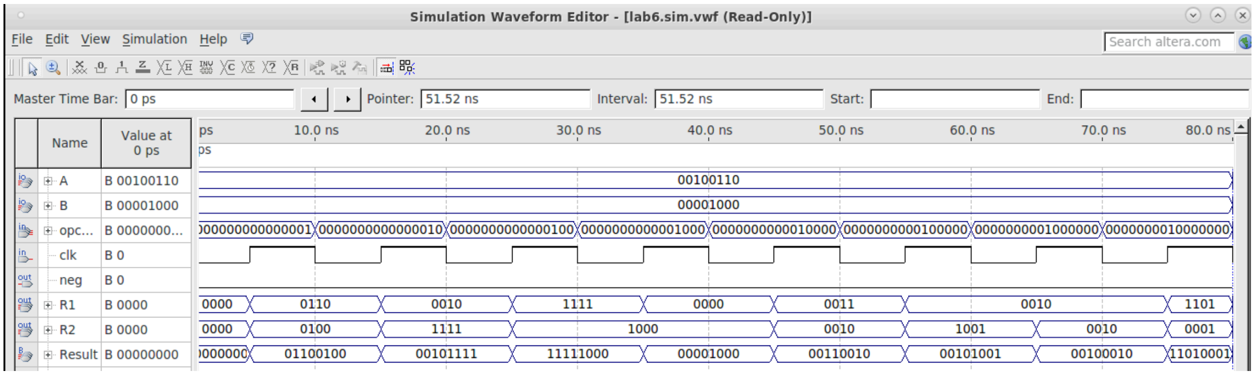
Opcode Table

Function #	Opcode	Function
1	00000001	Invert the bit-significance order of A
2	00000010	Shift A to left by 4 bits, input bit = 1 (SHL)
3	00000100	Invert upper four bits of B
4	00001000	Find the smaller value of A and B and produce the results (Min(A,B))
5	00010000	Calculate the summation of A and B and increase it by 4
6	00100000	Increment A by 3
7	01000000	Replace the even bits of A with even bits of B
8	10000000	Produce the result of XNORing A and B

ALU 2 Block Diagram



Waveform



Simulation Waveform Editor - [lab6.sim.vwf (Read-Only)]

Master Time Bar: 0 ps Pointer: 51.52 ns Interval: 51.52 ns Start: End:

Name	Value at 0 ps	ps	10.0 ns	20.0 ns	30.0 ns	40.0 ns	50.0 ns	60.0 ns	70.0 ns	80.0 ns
A	B 00100110	00100110								
B	B 00001000	00001000								
opc...	B 0000000...	000000000000001	0000000000000010	0000000000000100	0000000000001000	0000000000010000	0000000000100000	0000000001000000	0000000010000000	
clk	B 0									
neg	B 0									
R1	B 0000	0000	0110	0010	1111	0000	0011	0010		1101
R2	B 0000	0000	0100	1111	1000		0010	1001	0010	0001
Result	B 00000000	0000000	01100100	00101111	11111000	00001000	00110010	00101001	00100010	11010001

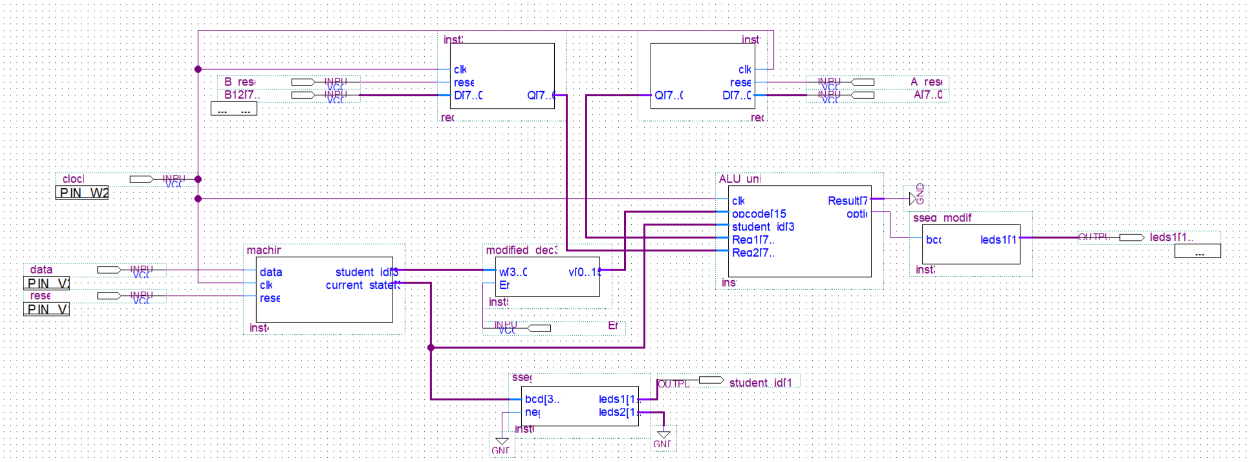
ALU 3

This ALU slightly differs from the others as it is based off ALU 1, but is modified to take 3 data inputs instead of 2. The student number pin has been added which takes input from the FSM. The reg values are compared with the student number and will change the display if one of the two digits of A equal to the ID signal value. The option output pin has also been implemented to reflect this.

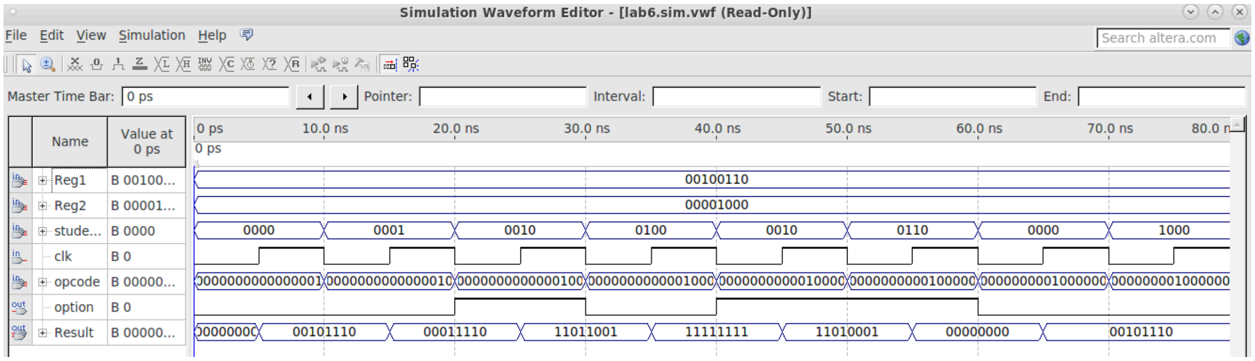
Opcode Table

Function #	Opcode	Function
1	00000001	Invert the bit-significance order of A
2	00000010	Shift A to left by 4 bits, input bit = 1 (SHL)
3	00000100	Invert upper four bits of B
4	00001000	Find the smaller value of A and B and produce the results (Min(A,B))
5	00010000	Calculate the summation of A and B and increase it by 4
6	00100000	Increment A by 3
7	01000000	Replace the even bits of A with even bits of B
8	10000000	Produce the result of XNORing A and B

ALU 3 Block Diagram



Waveform



Name	Value at 0 ps
Reg1	B 00100...
Reg2	B 00001...
stude...	B 0000
clk	B 0
opcode	B 00000...
option	B 0
Result	B 00000...

Conclusion

This lab explored many of the components and processes learned in previous labs. The 7-segment displays, registers, ASUs, decoders, and FSM had to be integrated and meshed together to create a functioning central processing unit (CPU). This lab also solidified the teachings from lectures and built-up experience working with Quartus II 13.0 and the VHDL programming language. This lab serves as a solid conclusion to learning real skills in the field concerning digital systems and FPGAs.

References

- COE328 Homepage, Toronto Metropolitan University (2024), Lab6_report https://www.ee.torontomu.ca/~courses/coe328/

- COE328 Homepage, Toronto Metropolitan University (2024), Lab #6 https://www.ee.torontomu.ca/~courses/coe328/