Lab 06 Digital Systems

Oscar Wang

Khashayar Bayati

COE328

Section 24

Dr. Asad, Arghavan

12th December 2024

Table of Contents

Introduction

This lab involves using multiple components from previous labs and created new ones on top of it to make an arithmetic and logic unit. The entire circuit begin with two 8-bit input into two latches, where the data is temporarily stored, the data is then proceeded into an ALU unit, where a series of calculations and outputs into two seven-segment displays. The control unit consists of 2 separate components, the finite state machine (FSM) and a 4 to 16 decoder. The FSM is used for determine controller sequence and the decoder decides which operation to make in the ALU.

Components: Latch 1 and 2

Latches act as a temporary storage of the data, these two components take the data and passes it onto the other components. Every time the clock goes up (and down), the data passes on, this also demonstrates that it has an uprising edge structure.

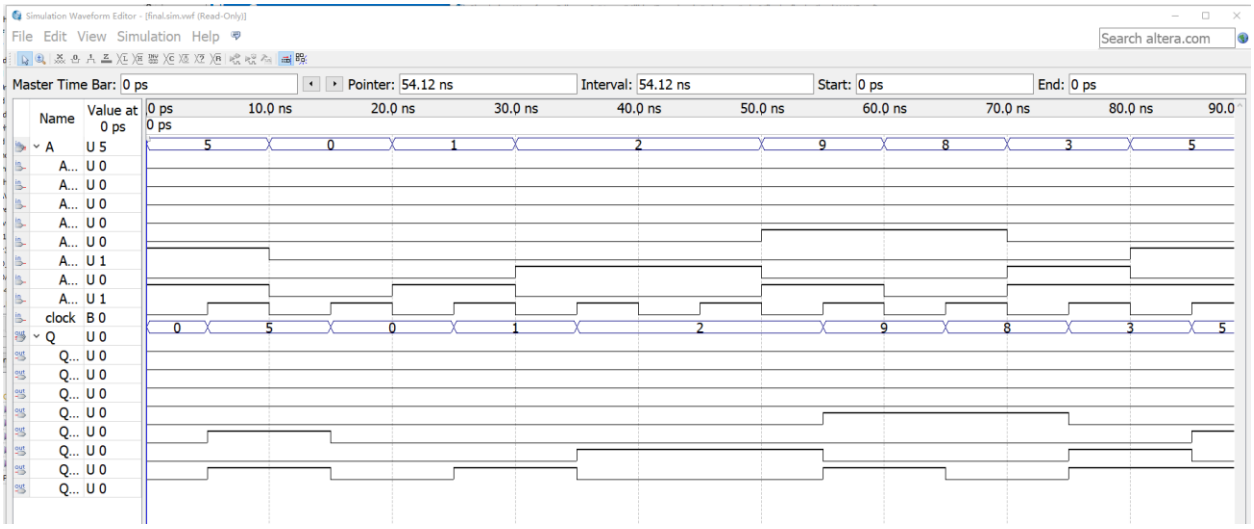

Fig 1.1 Waveform simulation for a latch

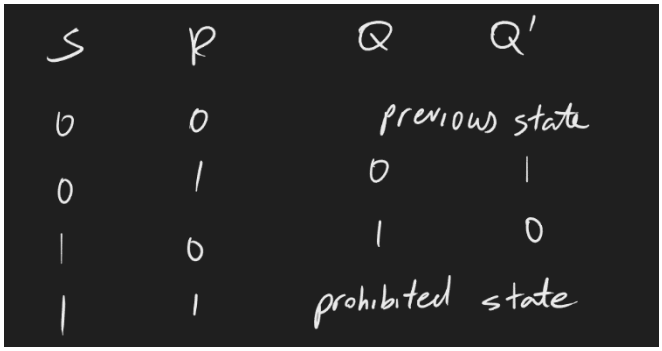

Fig 1.2 truth table for a SR Latch

4 to 16 decoder

This is one of the most important components as it determines the operations for the ALU block, the outputs of decoder act like opcode, the first 8 opcodes from 10000000 to 00000001 tells which operations to make.

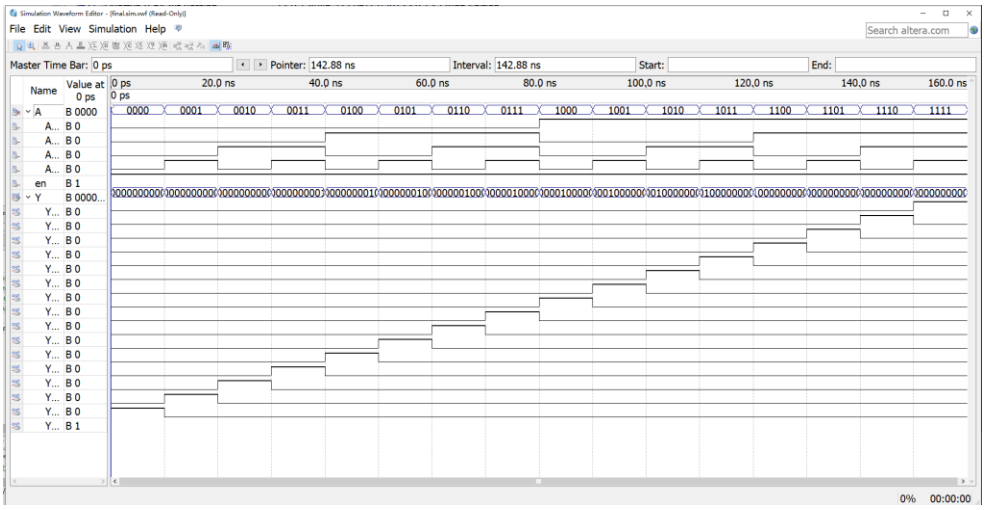

Fig 1.3 Example waveform simulation for a 4 to 16 decoder

W	X	Y	Z	OP0	OP1	OP2	OP3	OP4	OP5	OP6	OP7	OP8	OP9	OP10	OP11	OP12	OP13	OP14	OP15
0	0	0	0	1	0	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	0	1	0	0	0	1	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	1	0	0	0	0	0	0	1	0	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
0	1	1	0	0	0	0	0	0	0	1	0	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	0
1	0	0	0	0	0	0	0	0	0	0	0	1	0	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	0
1	0	1	0	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	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	0
1	1	0	1	0	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	0	1	0
1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Fig 1.4 Truth table for a 4 to 16 decoder

FSM – Finite State Machine

This component takes a clock input and cycles through a set of values, with each pulse sent to the machine it goes to the next state, and outputs the corresponding student number to its current state.

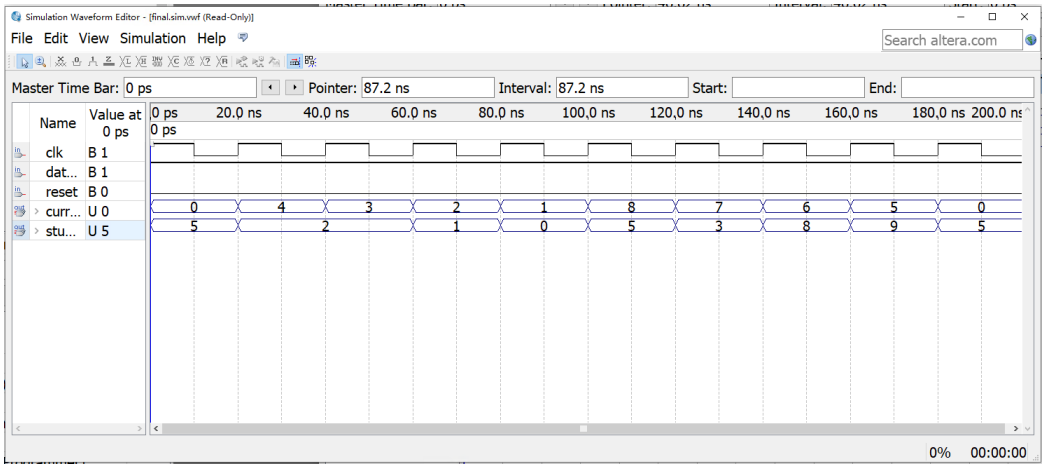

Fig 1.5 Example simulation of FSM

In this particular finite state machine I designed, it follows a 0-4-3-2-1-8-7-6-5

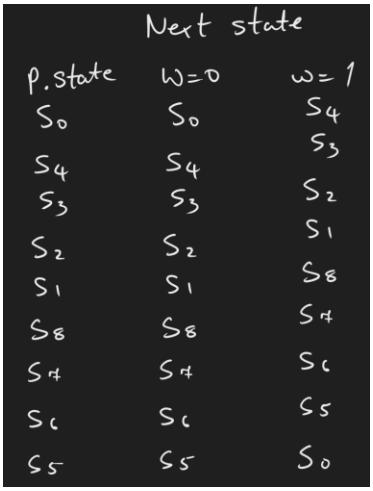

Fig 1.6 State assigned table for FSM

The three structures above are all control units involved in the circuit

ALU #1

The purpose of ALU is set to perform a series of calculations given in the lab manual. The ALU itself has a simple design with two registers connected to the unit, with an opcode for its actions and a clock driving the output of it.

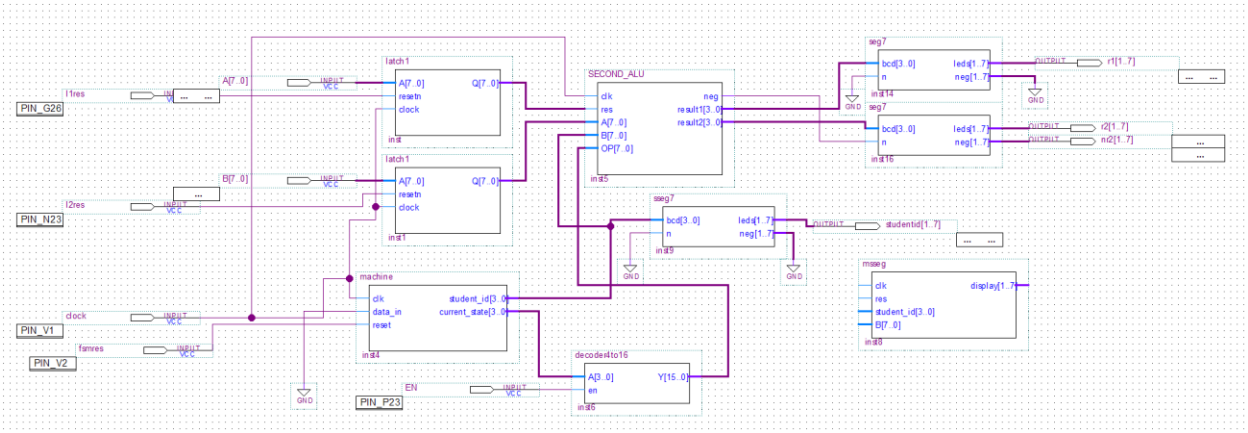

Fig 2.1 block schematic diagram for the general processing unit

The ALU unit is the central of the circuit consist of two 8-bit input A and B and OP for opcodes, it then outputs two 4-bit values r1 and r2 with one being the first four bits of the processed data and the other one for the following four bits of data. On top of that, it also has a neg output to determine the sign of the output. With neg, r1, and r2 together, it displays the hexadecimal representation of the binary number result of the operation.

Function#	Microcode	Function
1	0000000000000001	A + B
2	0000000000000010	A - B
3	0000000000000100	NOT A
4	0000000000001000	A NAND B
5	0000000000010000	A NOR B
6	0000000000100000	A AND B
7	0000000001000000	A XOR B
8	0000000010000000	A OR B
9	0000000100000000	A XNOR B

Fig 2.2 opcode feed into ALU and its action

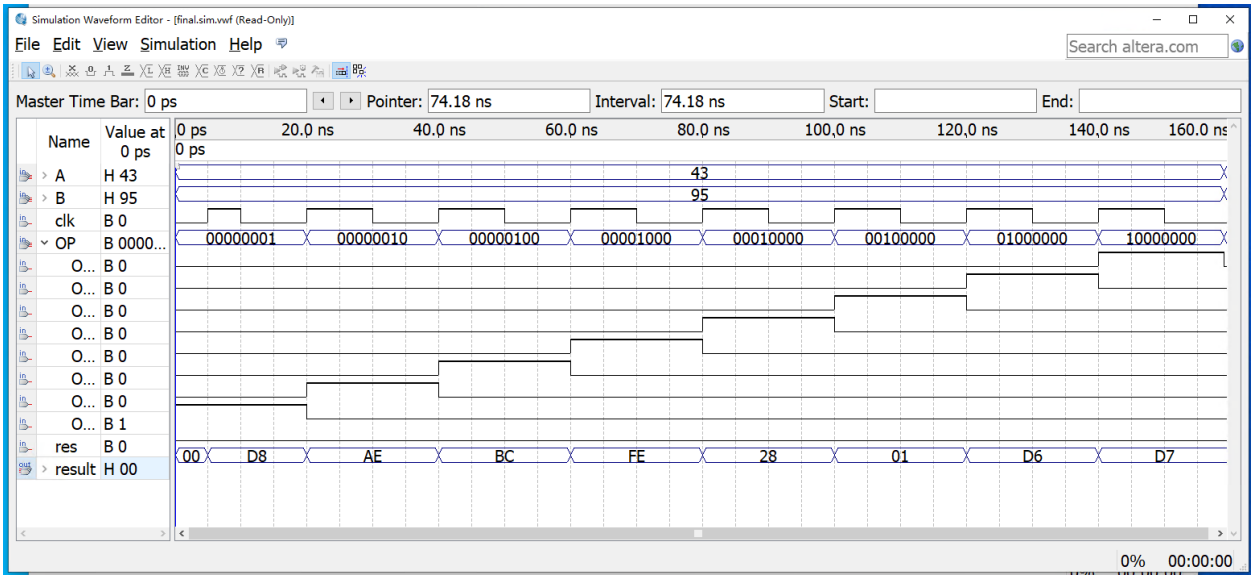

Fig 2.3 Simulation output for ALU #1 (basic logic manipulation)

ALU#2

Function #	Operation / Function
1	Decrement **B** by 9
2	Swap the lower and upper 4 bits of **B**
3	Shift **A** to left by 2 bits, input bit = 0 (SHL)
4	Produce the result of NANDing **A** and **B**
5	Find the greater value of **A** and **B** and produce the results (Max(**A**,**B**))
6	Invert the even bits of **B**
7	Produce null on the output
8	Replace the upper four bits of **B** by upper four bits of **A**

Fig 2.4 Function F from lab manual (assigned by TA)

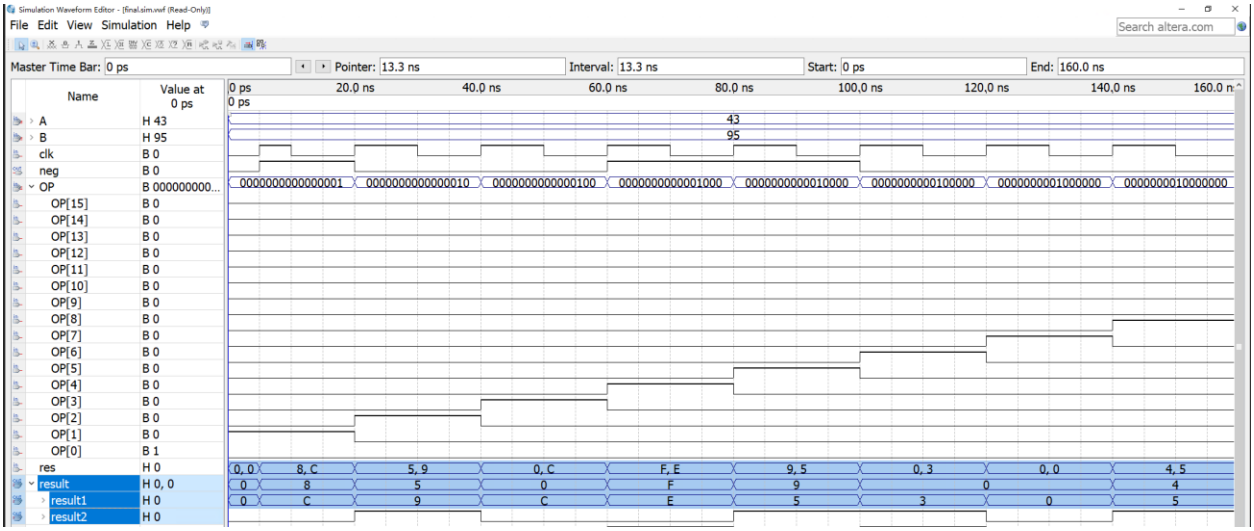

Fig 2.5 Simulation waveform of ALU #2

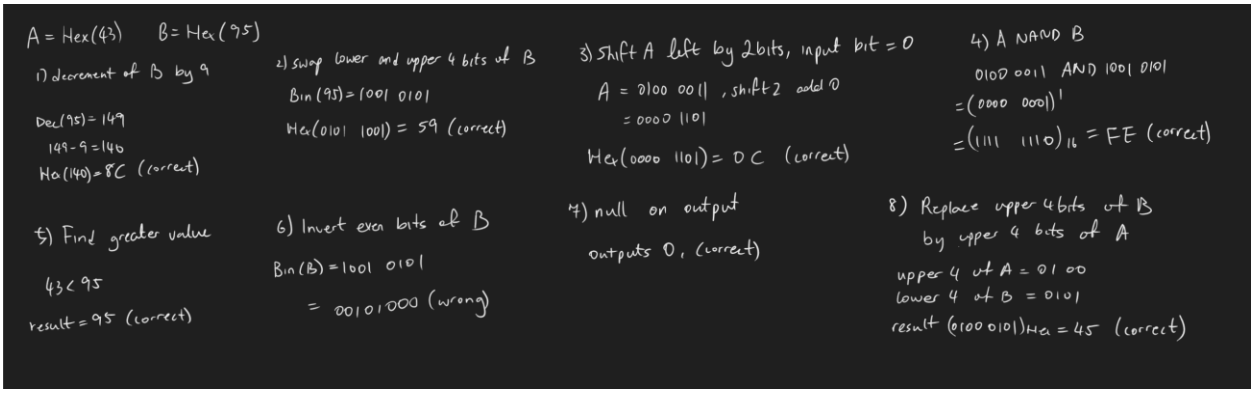

Fig 2.6 Calculation with regards to the simulation

ALU #2 takes in the opcode and do calculations based on it, similar to ALU #1. In the waveform diagram, the opcode has been set to different values in order to simulate the changes, from Fig 2.6, we can see the calculations behind the simulation and its corresponding values.

ALU #3

Function #	Microcode	Operation
1	0000000000000001	Check if B has student id in it, yes (y), no (n)
2	0000000000000010	Check if B has student id in it, yes (y), no (n)
3	0000000000000100	Check if B has student id in it, yes (y), no (n)
4	0000000000001000	Check if B has student id in it, yes (y), no (n)
5	0000000000010000	Check if B has student id in it, yes (y), no (n)
6	0000000000100000	Check if B has student id in it, yes (y), no (n)
7	0000000001000000	Check if B has student id in it, yes (y), no (n)
8	0000000010000000	Check if B has student id in it, yes (y), no (n)
9	0000000100000000	Check if B has student id in it, yes (y), no (n)

Fig 2.7 Microcode for Microcode generated by decoder and the functions performed by ALU given microcode

The third ALU does a comparison with the input value and outputs a result to the seven-segment display.

Fig 2.8 Coded part of ALU #3

The code takes in a 4-bit student id and 8-bit input B from the previous ALU, it compares with the upper and lower four bits of B, a temporary variable is used to determine the comparison made, then it outputs a signal to display “y” when there are matches or “n” when there are not.

Simulated Result

Fig 2.10 shows the block schematic diagram for the entire processing unit, since they are all synchronous clocked, a clock input is connected to all components including latches, FSM, and ALU. An example input of 98 and 35 are used, these are the last four digit of my student id.

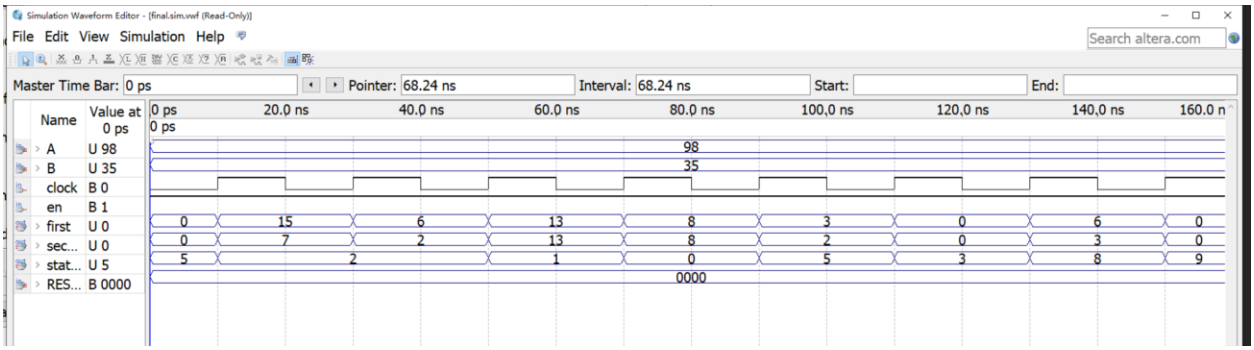

Fig 2.9 simulated waveform output of circuit with two 8-bit input, ALU, and FSM

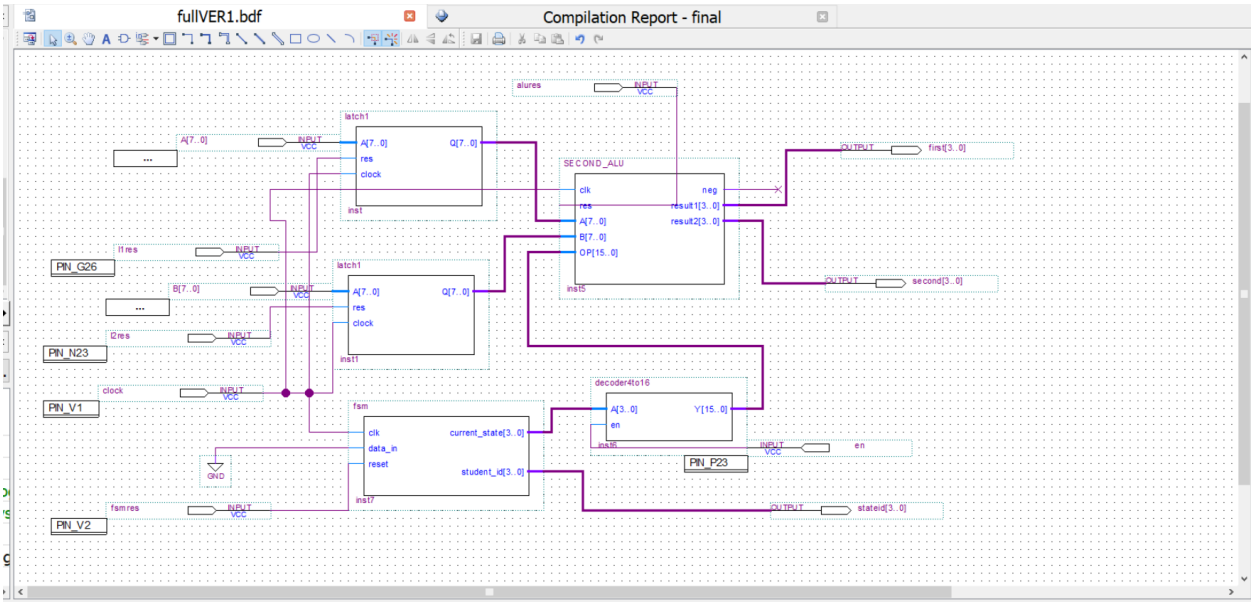

Fig 2.10 block schematic diagram for simulated output

In this diagram, *A* and *B* are two inputs of the system, *first* and *second* outputs the result, enable has been turned on as indicated in Fig 2.9, while every reset is off.

Conclusion

In conclusion, it can be understood from this lab that ALU's can perform a series of process based on its microcode. Then a simple control unit was implemented to output different opcodes to control the ALU's action. Last thing to mention is the S-R latch is used at the beginning of the entire circuit as it stores the

input values and sends it to the ALU and control unit. The lab covered a span of every other lab indicates its complexity and importance.