COE 328 Lab 6 report: Design of a Simple General-Purpose Processor

Thulasihan Slvarajah

500821937

Date Submission: December 1, 2018

TA: Sajjad Rostami Sani

1. Table of Contents

2. Introduction

This objective of this lab experiment is to design and construct an Arithmetic and Logic Unit (ALU) in VHDL environment and implement it on an FPGA board. This simple GPU contains Latch 1, Latch 2, 4-16 decoder, arithmetic logic unit (ALU), and a finite state machine(FSM). The outputs of each problem set will be displayed on 7-segment displays. The GPU will perform 9 functions from Table 1 of the manual which depends on the microcode input.For problem 2, the ALU will be modified in order to do 9 additional functions based on the assigned ALU from the TA. For problem 3, the ALU will be modified again, in order to take in the student id as an additional input and will perform a comparison functionality based on the assigned ALU from the TA.

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

Register(Latch 1/Latch 2)

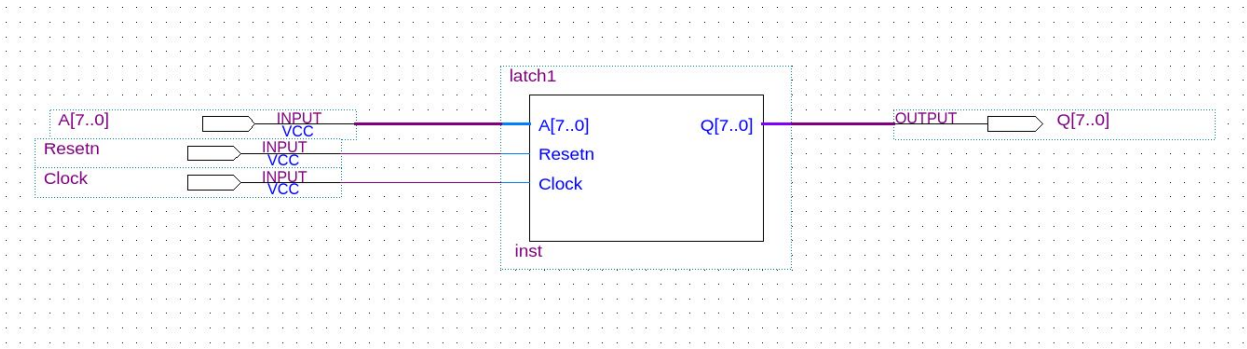
The register is made up of 8 D-flip-flops that are positive edge triggered. Registers are utilized to temporary store the input values and then pass them to the following components in the system. As signified in Figure 1 of the lab manual, two 8-bit register units(Latch 1/ Latch 2) are utilized in the ALU to store inputs A and B . The register reads the bit values on its input on the rising edge of the clock signal and passes those bit values to the output port on the next rising edge of the clock signal.

Truth table of Latch 1/ Latch 2(Single D-flip flop)

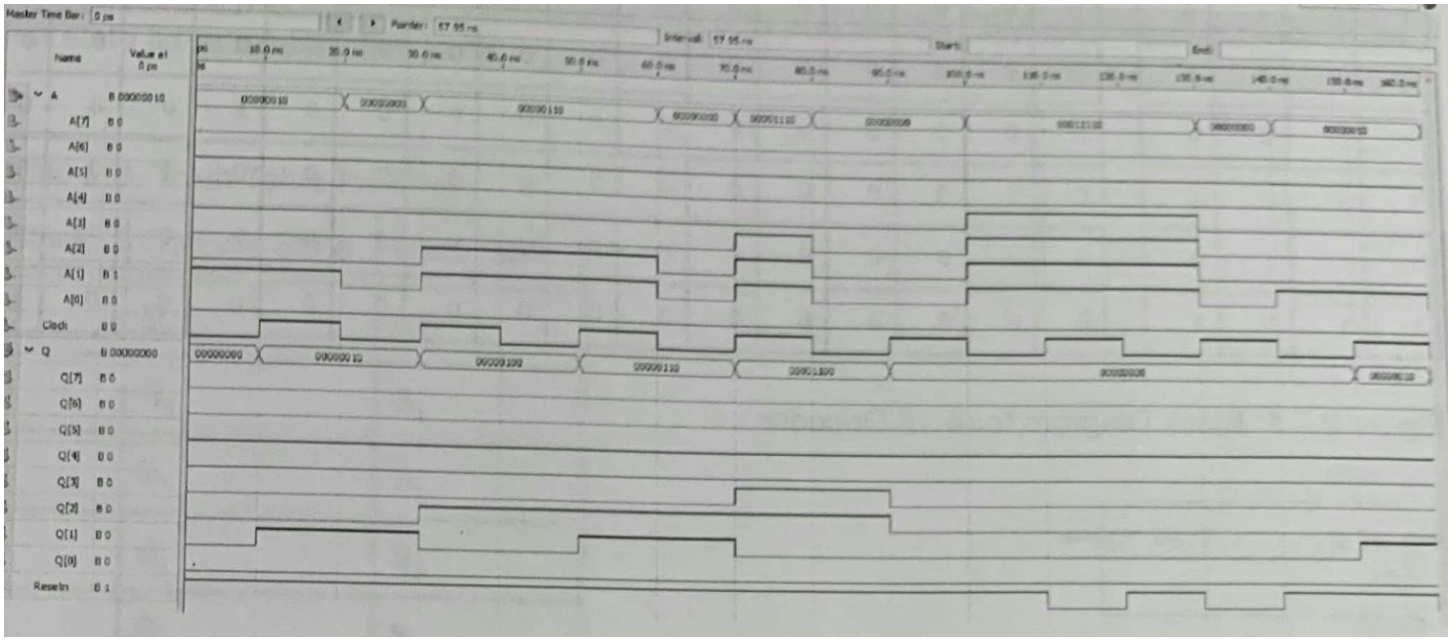
Input			**Output**
Clock	**Reset**	**A**	**Q**
Falling	0	0	0
Falling	0	1	0
Falling	1	0	0
Falling	1	1	0
Rising	0	0	0
Rising	0	1	0
Rising	1	0	0

Rising	1	1	1

Block Schematic of Latch1 / Latch2



WaveForm of Latch 1/ Latch2



Note: Waveform was done on my own laptop(screen shot made it black/white), hard to get all information on the screen shot.

4:16 decoder

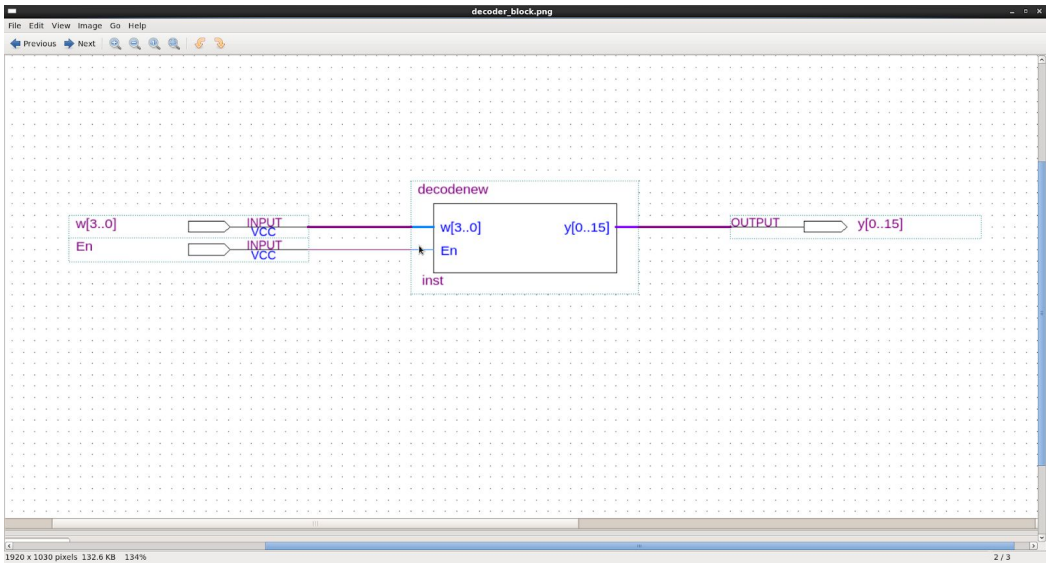
The 4:16 decoder contains 4 inputs and 16 outputs and its used to transform a set of digital input signals into an equivalent decimal code of its output.. The decoder is tasked with receiving the signal current state from FSM and decoding it to the operation-selector microcode. The Decoder unit passes the signal OP to the ALU core, which will then translate to operations selector for the ALU core and follow the functions enlisted in Table 1 of the lab manual.

Truth Table

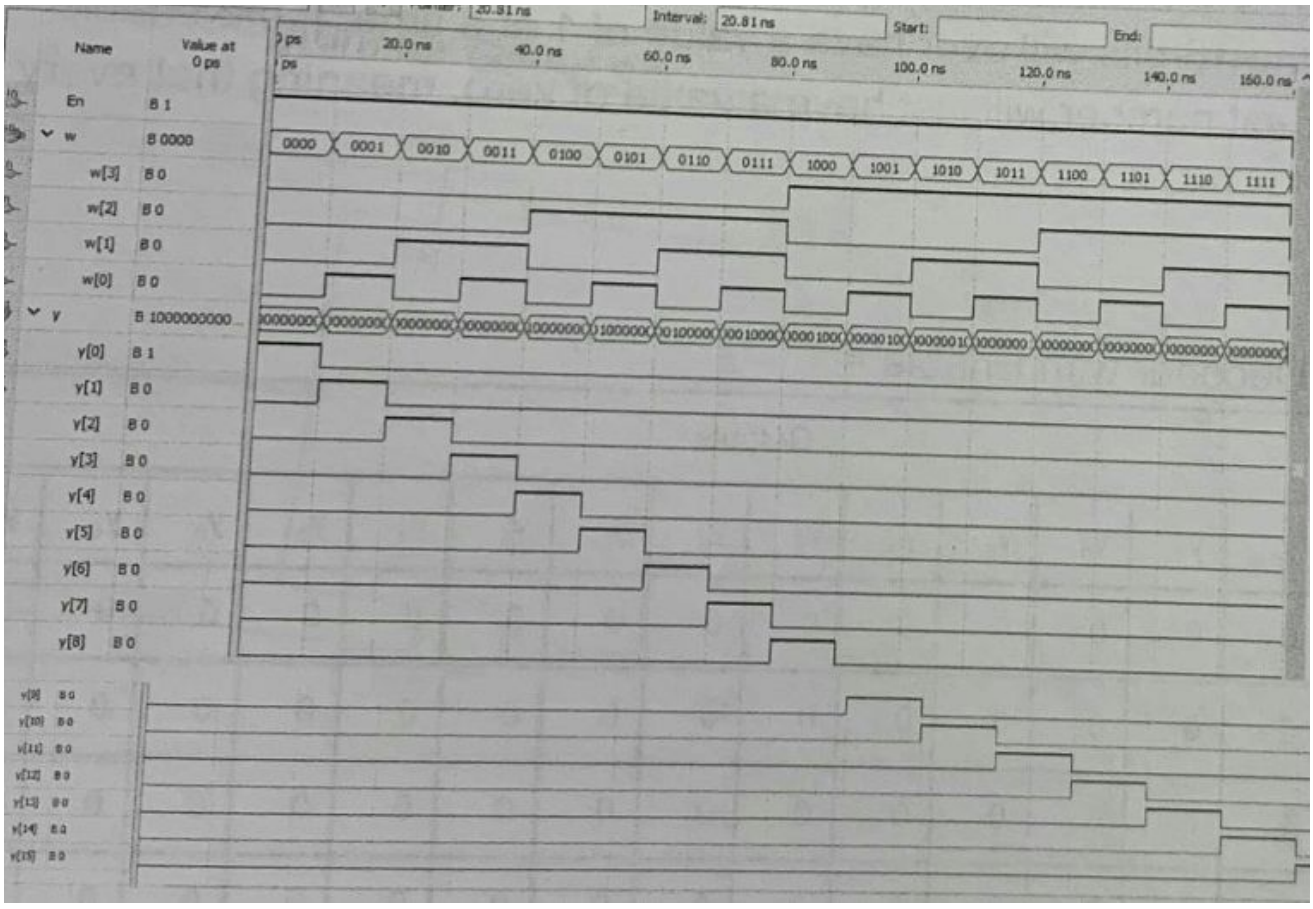
Enable=1

Input: w[3..0]	y[0..15]
0000	100000000000000
0001	010000000000000
0010	001000000000000
0011	000100000000000
0100	000010000000000
0101	000001000000000
0110	000000100000000
0111	000000010000000
1000	000000001000000
1001	000000000100000
1010	000000000010000
1011	000000000001000
1100	0000000000000100
1101	0000000000000010
1111	0000000000000001

Block Schematic



WaveForm of Decoder



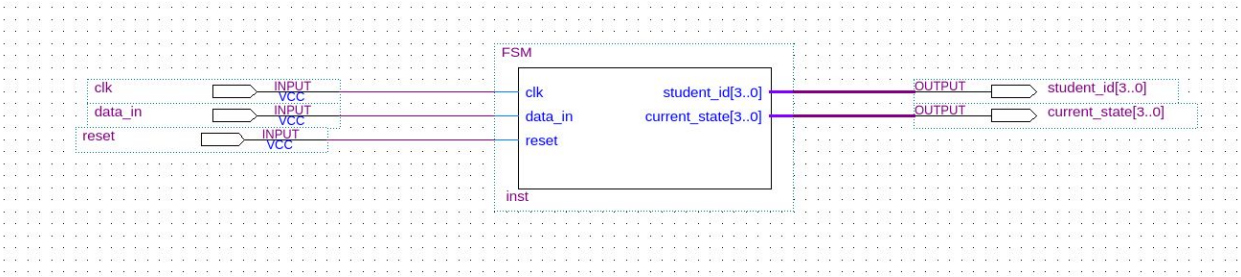
FSM

The FSM contains 8 states and one primary input (data in). The FSM component of the Control unit decides the pattern of the controller sequence which are cycled through using the clock signal. The FSM will only act as an up-counter, cycling through states S0 to S 8 consecutively and back to state 0 while my student id will be displayed on a 7- segment display. The FSM takes the clock signal as the input, and produces the 4-bit output current state and passes it to the decoder.

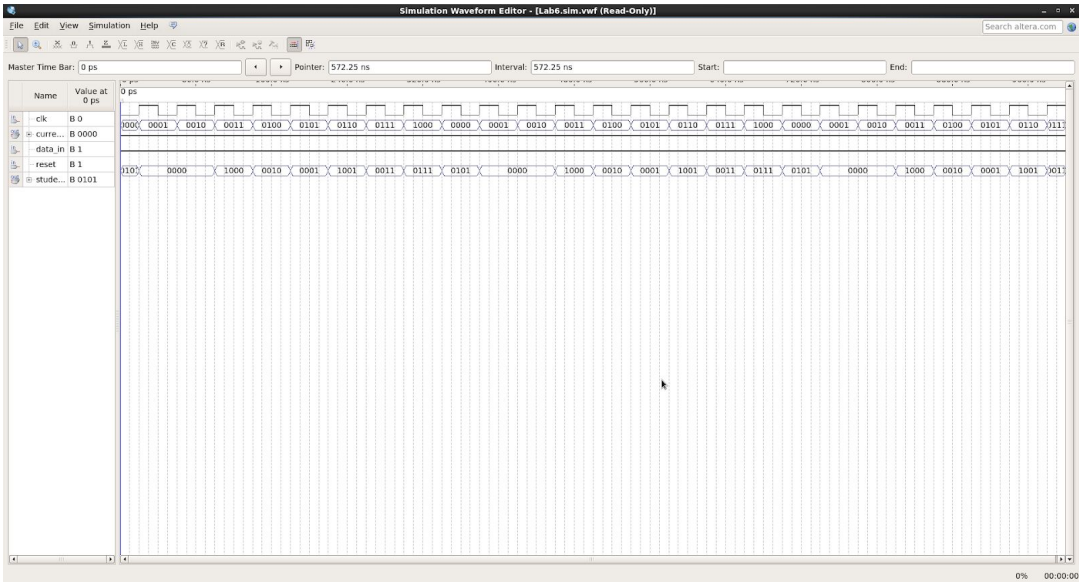
Truth Table

	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	0000
S2	0010	0000	0011	1000
S3	0011	1000	0100	0010
S4	0100	0010	0101	0001
S5	0101	0001	0110	1001
S6	0110	1001	0111	0011
S7	0111	0011	1000	0111
S8	1000	0111	0000	0101

Block Schematic of FSM



WaveForm of FSM



4. ALU_1 for Problem Set 1 of the Lab 6

Problem 1 consists of designing a General Processing Unit (GPU). The FSM output will dictate the operation selector signal to the Arithmetic Logic Unit (ALU). The ALU will represent the outputs of operations shown in the Table of Microcodes below and following the same order. This outputs will be represented on two 7-segment digits on the FPGA board. The major inputs within the ALU core are A and B (two 8- bit values).

Block Schematic

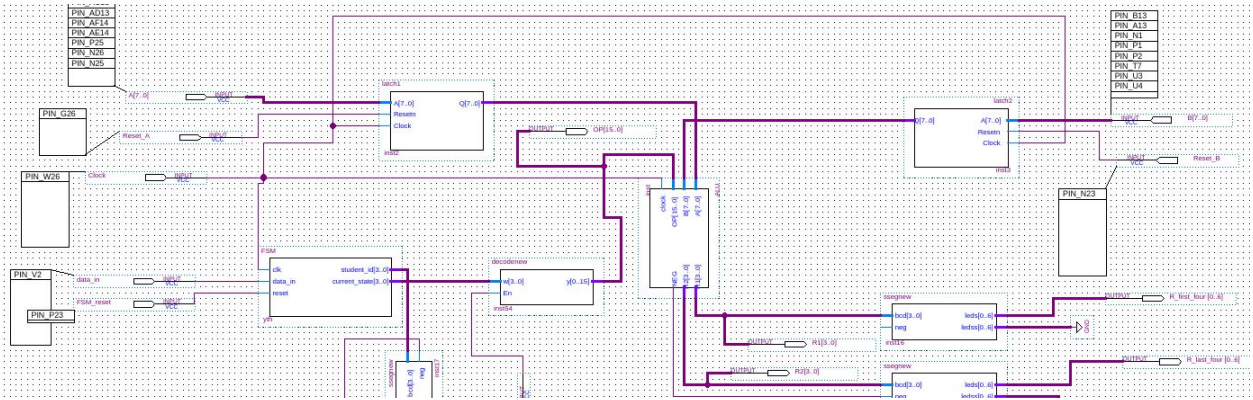
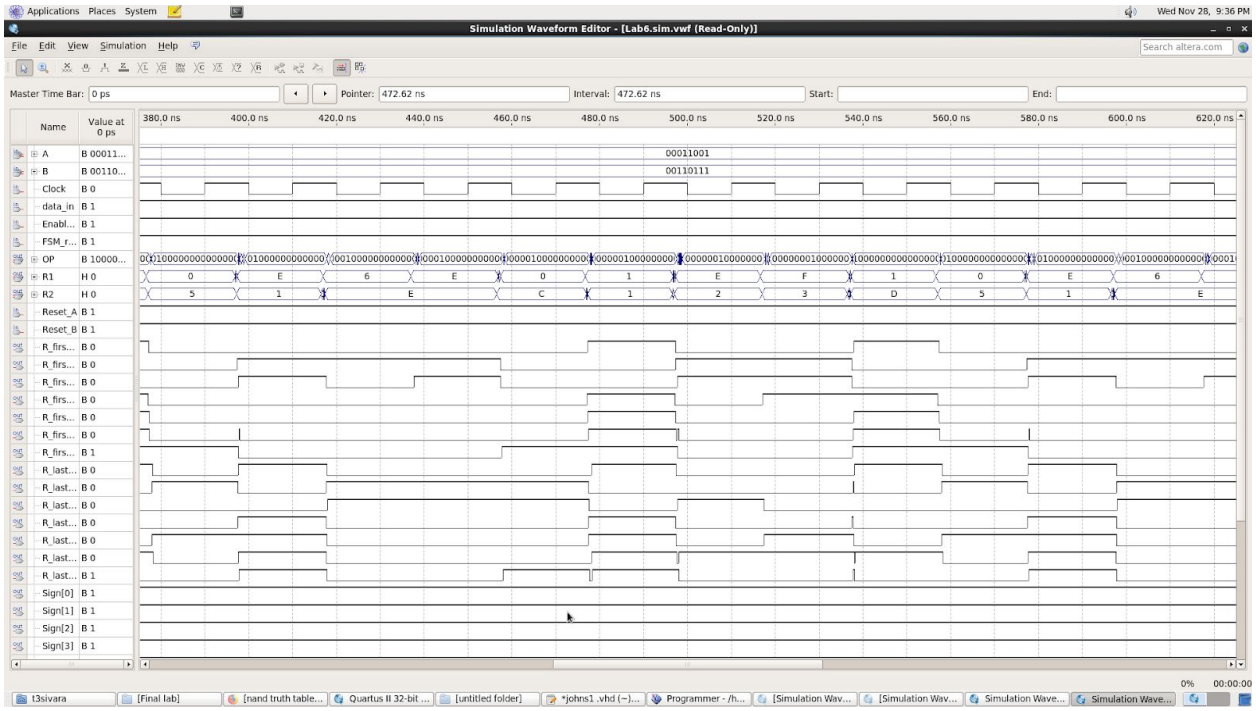


Table of Microcodes

Function #	Microcode	Boolean Operation/ Function
1	1000000000000000	sum(A, B)
2	0100000000000000	diff(A, B)
3	0010000000000000	$\overline{A}$
4	0001000000000000	$\overline{A \bullet B}$
5	0000100000000000	$\overline{A + B}$
6	0000010000000000	$A \bullet B$
7	0000001000000000	$A \oplus B$
8	0000000100000000	$A + B$
9	0000000010000000	$\overline{A \oplus B}$

WaveForm(Tried my best to make it clear as possible)



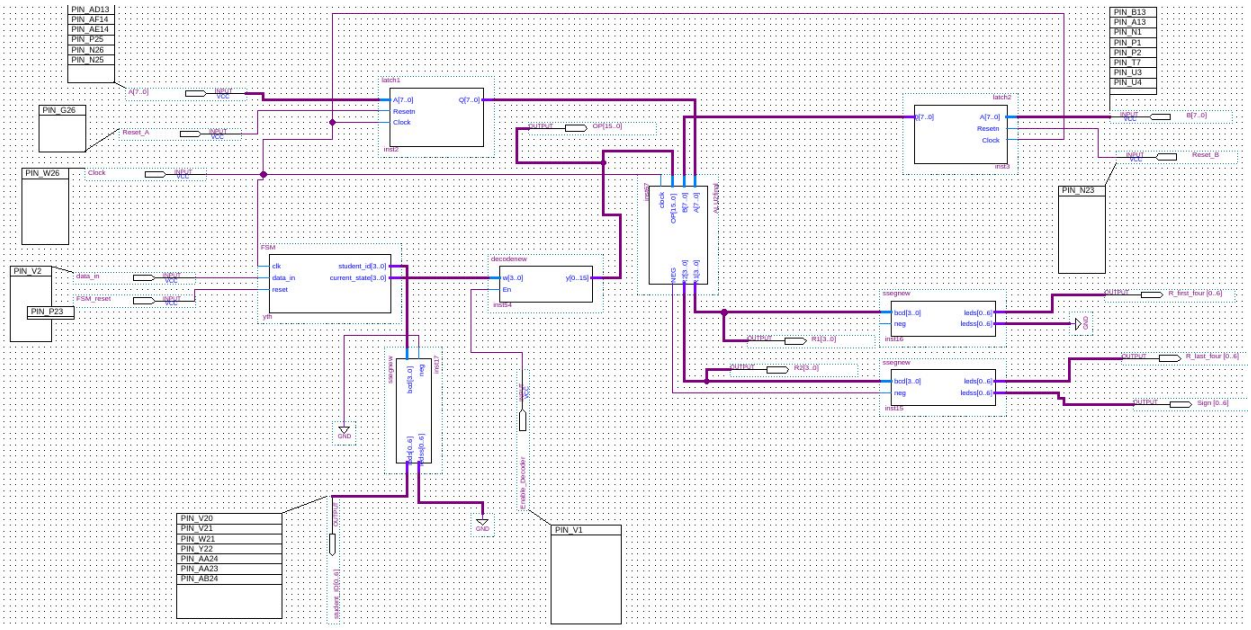
5. ALU_2 for Problem Set 2 of the Lab6

Problem 2 consists of modifying the ALU core and its functionalities. The new functionalities of the ALU are based on what the TA has assigned to me. It is tasked to do operations such as two's complement, shifting bits, and inverting the bits. The major outputs of these operations will be displayed on 7-segment displays and the outputs are represented as hexadecimal. The major inputs are two 8-bit values (A and B) that are used in the operation.

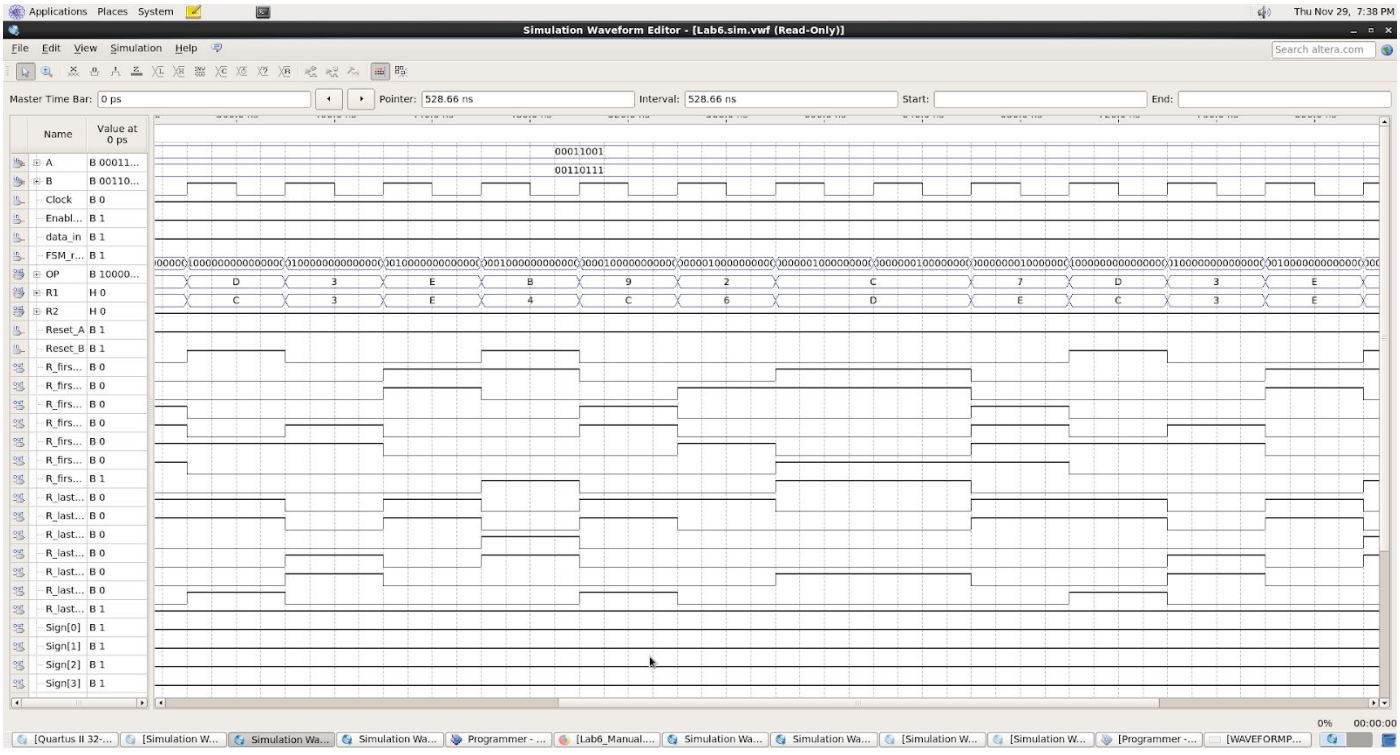
Table of Microcodes

Function #	Microcode	Boolean Operation/ Function
1	100000000000000	Replace the odd bits of A with odd bits of B(Ignore)
2	010000000000000	Produce the result of NANDing A and B
3	001000000000000	Calculate the summation of A and B and decrease it by 5
4	000100000000000	Produce the 2's complement of B
5	000010000000000	Invert the even bits of B
6	000001000000000	Shift A to left by 2 bits, input bit = 1 (SHL)
7	000000100000000	Produce null on the output
8	000000010000000	Produce 2's complement of A
9	000000001000000	Rotate B to right by 2 bits (ROR)

Block Diagram for Problem 2



WaveForm for Problem 2



6. ALU_3 for Problem Set 3 of the Lab6

The task of this problem is to utilize the student_id output from the FSM sub-component of the Control Unit. Problem 3 consists of modifying the ALU core to process 4 inputs and output a "boolean" value (true is y and false is n). The comparison output will be displayed on a 7-segment display and the output is one clock behind. The problem this ALU solves is:

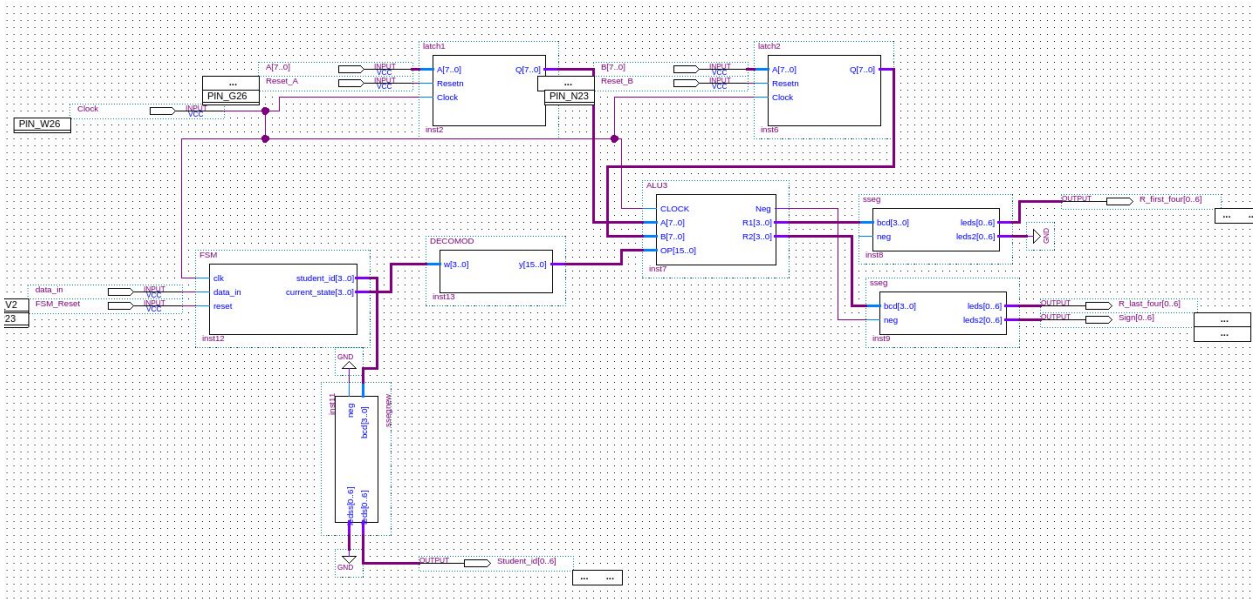
- For each microcode instruction, display 'y' if one of the 2 digits of A are greater than FSM output (student_id) and 'n' otherwise. Use the microcode instruction from part 1 of the lab.

Table of Microcodes

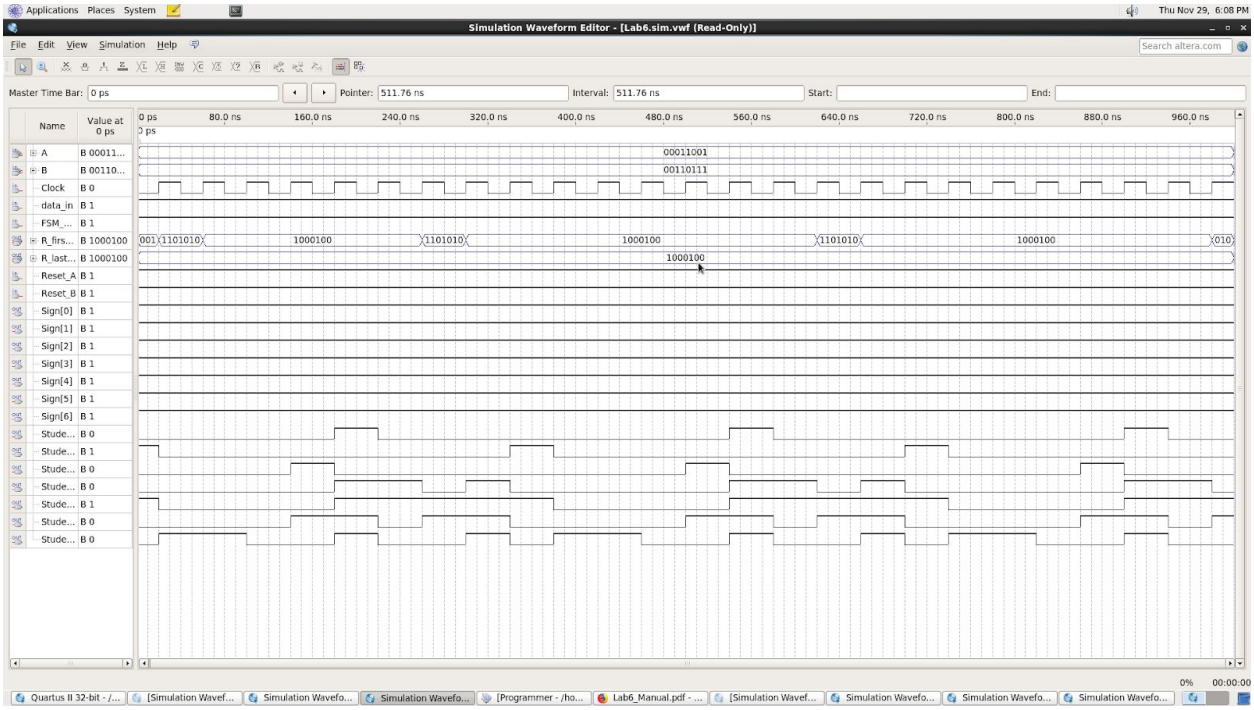
Function #	Microcode	Boolean Operation/ Function
1	100000000000000	Find if A is greater than 5. If it is, output y. Otherwise, output n.
2	010000000000000	Find if A is greater than 0. If it is, output y. Otherwise, output n.
3	001000000000000	Find if A is greater than 0. If it is, output y. Otherwise, output n.
4	000100000000000	Find if B is greater than 8. If it is, output y. Otherwise, output n.
5	000010000000000	Find if A is greater than 2. If it is, output y. Otherwise, output n.
6	000001000000000	Find if A is greater than 1. If it is, output y. Otherwise, output n.
7	000000100000000	Find if A is greater than 9. If it is, output y. Otherwise, output n.
8	000000010000000	Find if A is greater than 3. If it is, output y. Otherwise,

		output n.
9	000000001000000	Find if B is equal to 7. If it is, output y. Otherwise, output n.

Block Diagram for problem 3



Waveform for Problem 3



7. Conclusion

Overall, three problems have been executed on a FBGA where the main concept of an ALU was used. Components such as Latch 1, Latch 2, 4:16 decoder and the FSM have been used within the design. The general schematic consisted of a control unit, bus, registers, and the ALU. The control unit fetches the instructions. The bus controls the data values throughout the processing unit. The registers act as temporary storage units. The ALU performs the specific operations that are instructed through microcode input .