DESIGN OF A SIMPLE GENERAL-PURPOSE PROCESSOR

Table of Contents

1.0 Introduction

ALU (Arithmetic Logic Unit) is the part of a central processing unit that performs arithmetic and logical operations. The ALU performs its functions by utilizing of 2 latches and a control unit. A latch (also called a storage unit) has two inputs and one output; it samples the inputs continuously when it's enable signal is on. Two latches are used to temporarily store the input values to pass it on to the ALU in the system. Each of the two latches are used to store inputs A and B, which are 8-bit numbers representing the last four digits of a nine digit student ID.

The purpose of the control unit is to determine the microcode that has to be passed on to the ALU, acting as an operations-selector for the core. It has two sub-components, including FSM (Finite State Machine) and 4 to 16 decoder. FSM is a computational model used to simulate sequential logic and decoder is a circuit that changes a code into a set of signals.

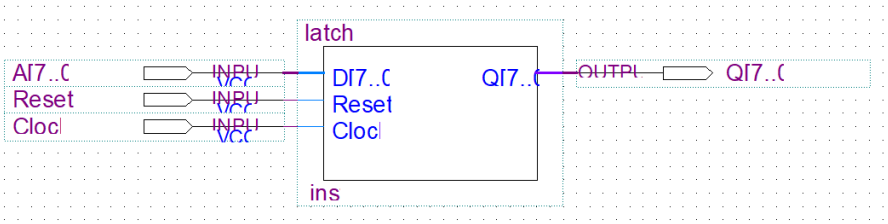
2.0 Components: Latch1, Latch2, 4:16 Decoder, FSM

2.1 Description of the components

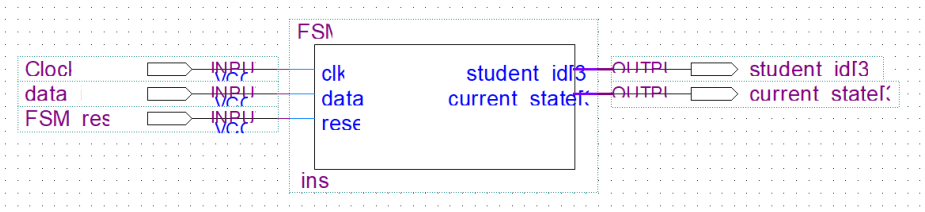
The two 8-bit registers (latches), used for the arithmetic logic unit, temporarily stores inputs A and B and reads the input values on the rising edge of the clock and then passes these two inputs to the output port on the next rising edge of the clock. The control unit, which is composed of a finite state machine and a 4 to 16 decoder, functions as an operations-selector for the arithmetic logic unit. The task of the finite state machine is to find the pattern of the controller sequence. The task of the decoder unit is to pass the signal output to the arithmetic logic unit and to receive the current state signal from the finite state machine in order to decode it to the operation-selector.

2.2 Block Diagram of Each Component

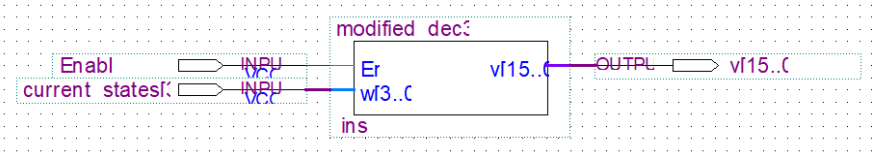
Latch



FSM

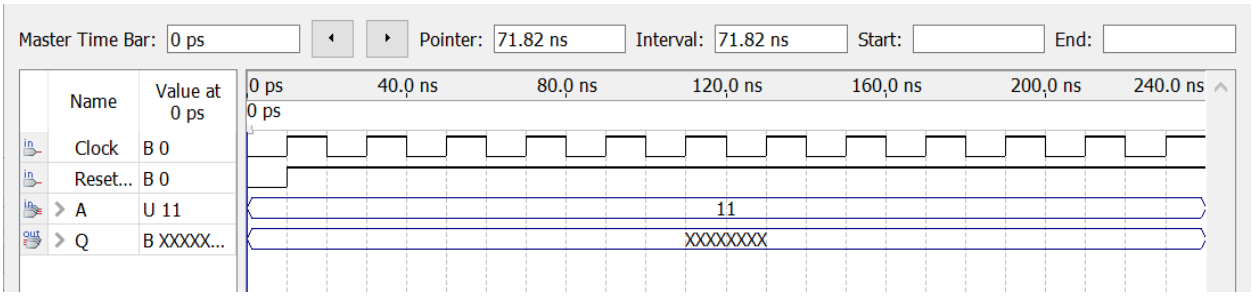


4 to 16 Decoder

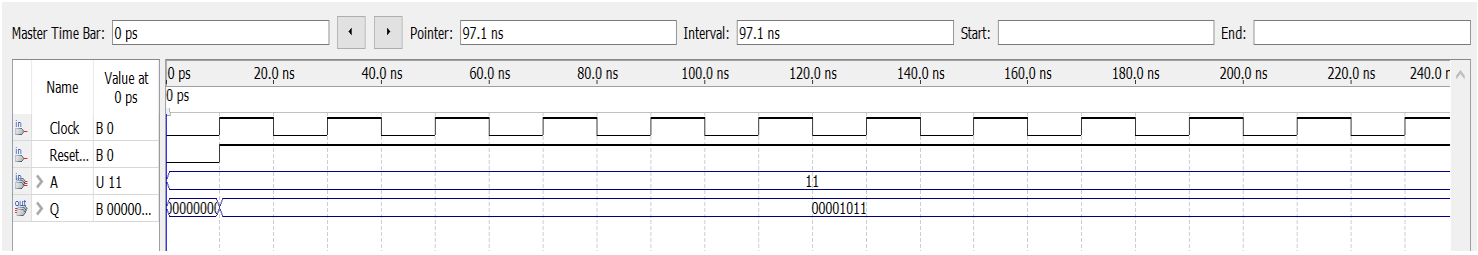


2.3 Waveform of Each Component

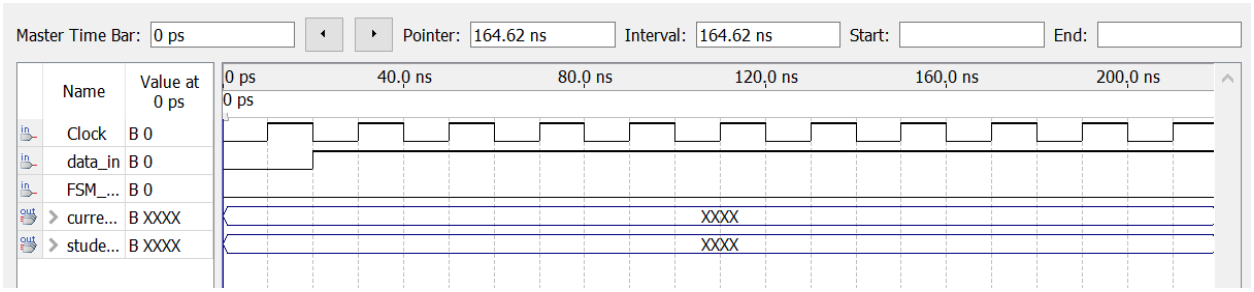
Latch



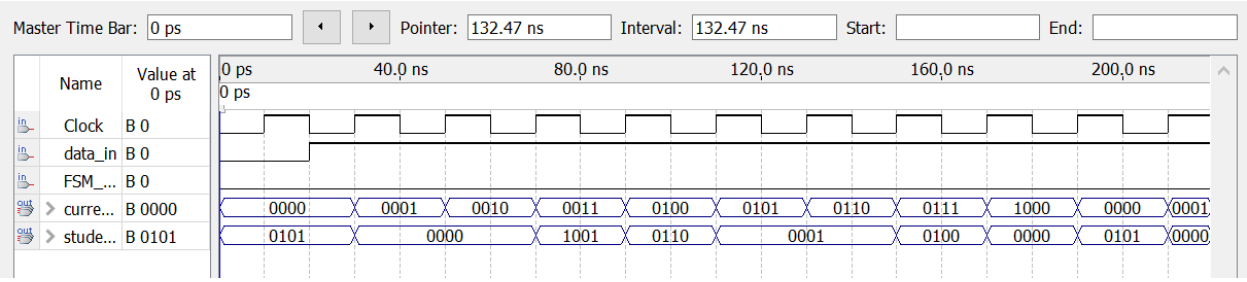
After simulation:



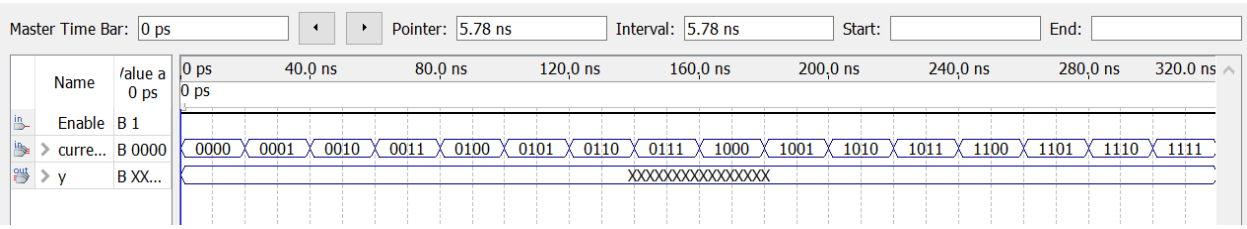
FSM



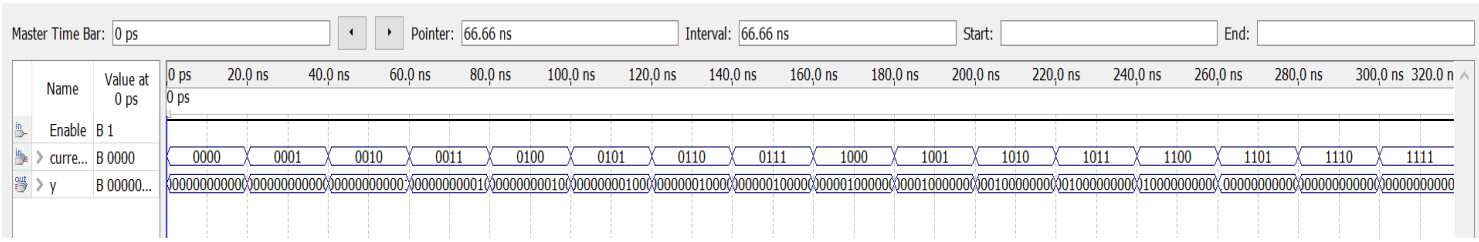
After simulation:



4 to 16 Decoder



After simulation:

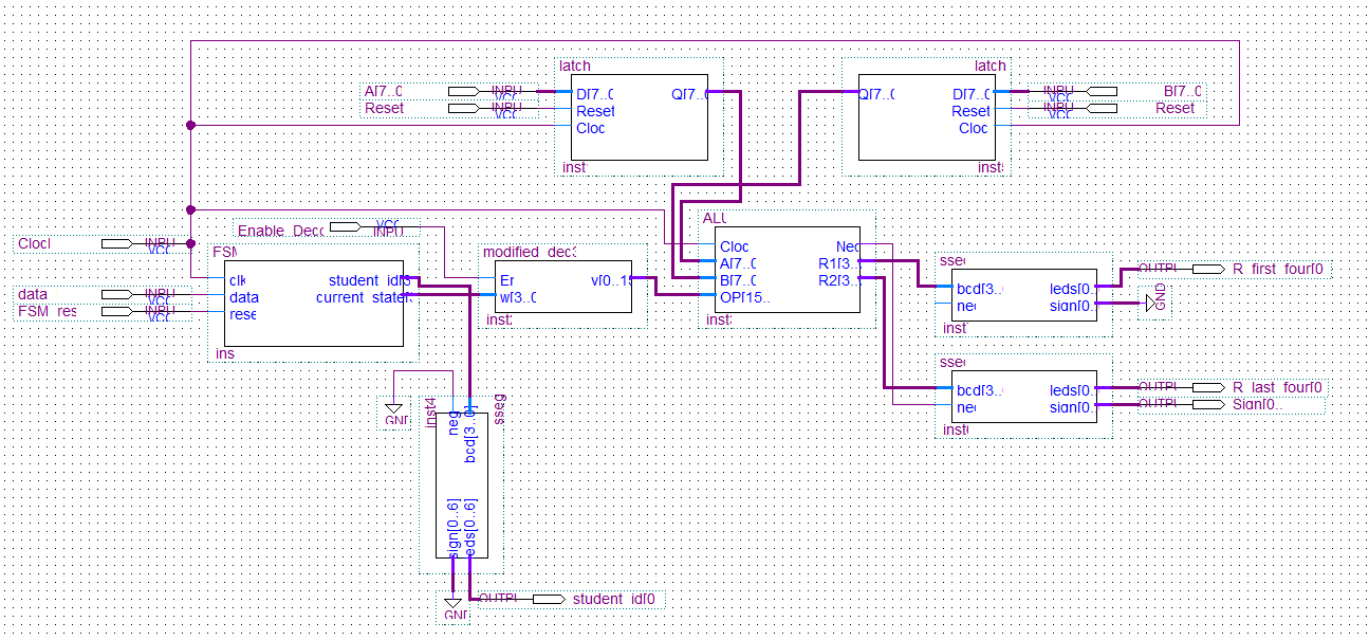


3.0 ALU 1

3.1 Purpose and Design of The Component

The purpose of the ALU is to perform mathematical operations on binary numbers. The ALU core takes two 8-bit inputs (A and B) from the storage unit as well as a 16-bit input from the decoder. Nine distinct operations are implemented in the ALU using microcode that is received from the control unit. Each of the nine microcode determines the operations of the inputs. The ALU is designed to perform addition, subtraction, inverse, and Boolean operations using A and B. The result of the ALU core is an 8-bit output that is displayed using two 7-segment display.

3.2 Block Schematic



3.3 Purpose of All Inputs and Outputs of This Component

Input

- Clock: It causes the ALU to either change or keep its output signal based on the values of the input signals during transition, which are passed on the rising edge of the clock cycles.
- A[7..0]: It represents the 8-bit input, A from the storage unit. A is the binary representation of the 6th and 7th digit of the student ID.
- B[7..0]: It represents the 8-bit input, B from the storage unit. B is the binary representation of the last two digits of student ID.
- OP[15..0]: OP is passed to ALU from the control unit. The ALU uses this input as the operations selector to select a specific function to perform, based on the microcode from the decoder.

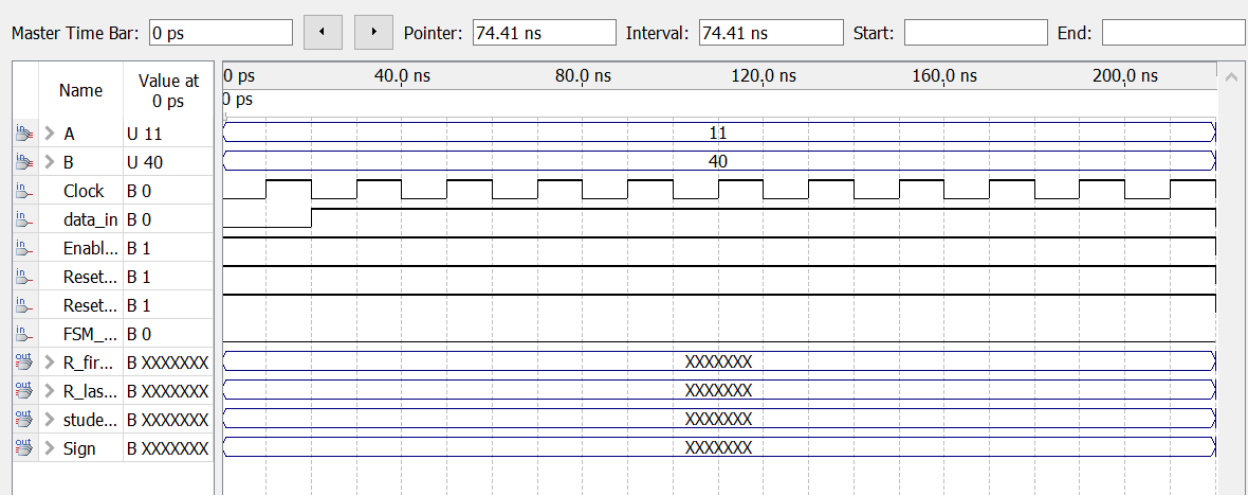
Output

- Neg: It displays any possible overflows.
- R1[3..0]: It passes the first 4-bits of the 8-bit number, which resulted from the Boolean operations of A and B to the 7-segment display.
- R2[3..0]: It passes the last 4-bits of the 8-bit number, which resulted from the Boolean operations of A and B to the 7-segment display.

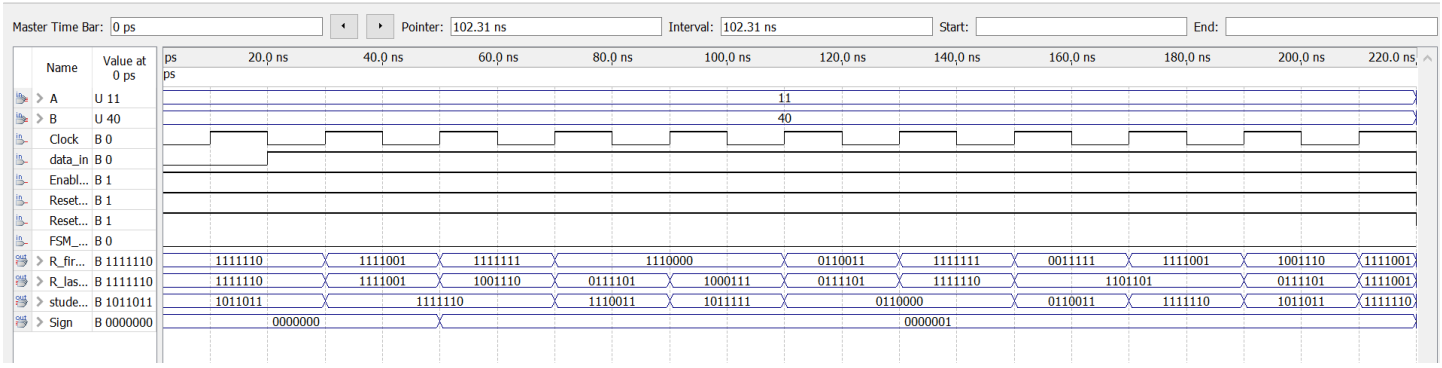
3.4 Table of Microcode's generated by decoder for ALU 1

Function #	Microcode	Boolean Operation (in ALU)
1	0000000000000001	sum(A,B)
2	0000000000000010	diff(A,B)
3	0000000000000100	$\bar{A}$
4	0000000000001000	$\overline{A \cdot B}$
5	0000000000010000	$\overline{A + B}$
6	0000000000100000	$A \cdot B$
7	0000000001000000	A⊕B
8	0000000010000000	A+B
9	0000000100000000	$\overline{A \oplus B}$

3.5 Waveform



After simulation:

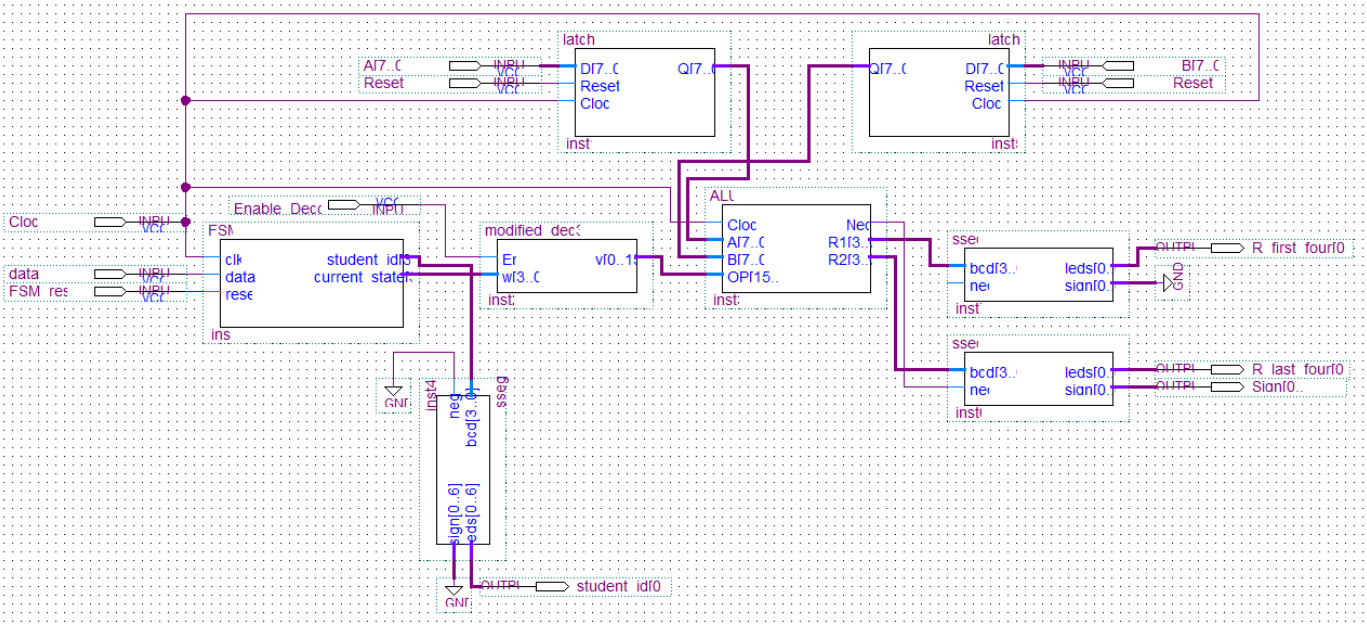


4.0 ALU 2

4.1 Purpose and Design of The Component

The purpose of the ALU is to perform mathematical operations on binary numbers. The ALU core takes two 8-bit inputs (A and B) from the storage unit as well as a 16-bit input from the control unit. The microcode from the control unit determines the operations of the inputs. The ALU design from the previous problem set was modified to perform a different set of operations that increments the number, shifts it to the right, finds the smaller value of the two number, rotates it to the right, inverts the bit significance order, and produces all high bits on the output. The result of the ALU core is an 8-bit output that is displayed using two 7-segment display.

4.2 Block Schematic



4.3 Purpose of All Inputs and Outputs of This Component

Input

- Clock: It causes the ALU to either change or keep its output signal based on the values of the input signals during transition, which are passed on the rising edge of the clock cycles.
- A[7..0]: It represents the 8-bit input, A from the storage unit. A is the binary representation of the 6th and 7th digit of the student ID.
- B[7..0]: It represents the 8-bit input, B from the storage unit. B is the binary representation of the last two digits of student ID.

- OP[15..0]: OP is passed to ALU from the control unit. The ALU uses this input as the operations selector to select a specific function to perform, based on the microcode from the decoder.

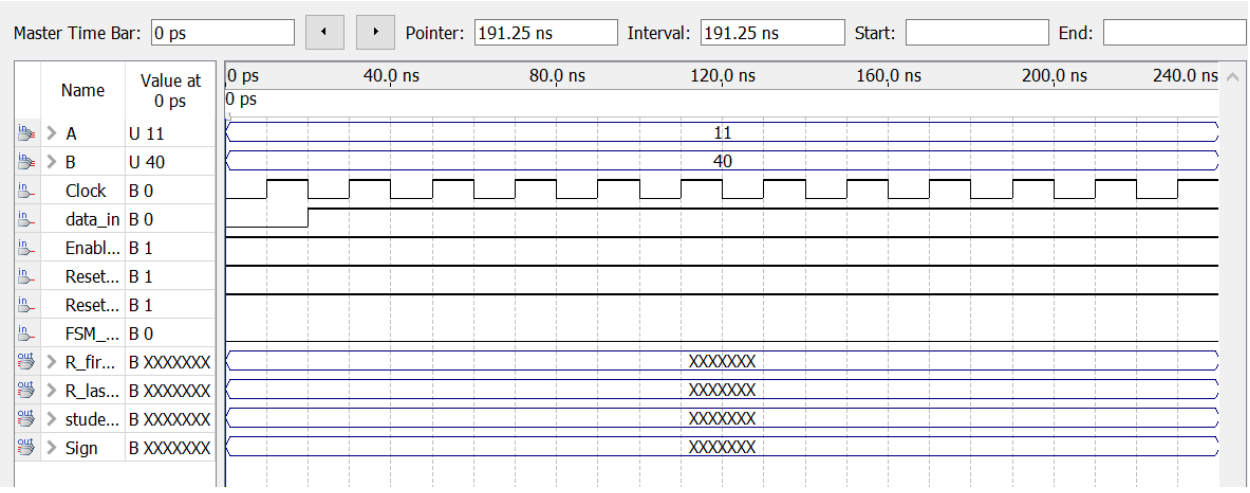
Output

- Neg: It displays any possible overflows.
- R1[3..0]: It passes the first 4-bits of the 8-bit number, which resulted from the Boolean operations of A and B to the 7-segment display.
- R2[3..0]: It passes the last 4-bits of the 8-bit number, which resulted from the Boolean operations of A and B to the 7-segment display.

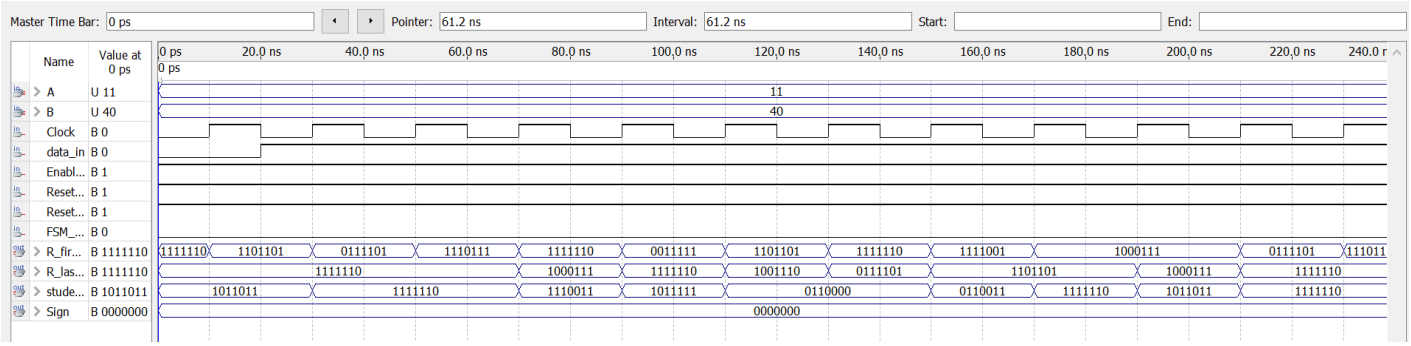
4.4 Table of Microcode's generated by decoder for ALU 2

Function #	Microcode	Boolean Operation (in ALU)
1	0000000000000001	Increment A by 2
2	0000000000000010	Shift B to right by two bits, input bit = 0 (SHR)
3	0000000000000100	Shift A to right by four bits, input bit = 1 (SHR)
4	0000000000001000	Find the smaller value of A and B and produce the results (Min(A,B))
5	0000000000010000	Rotate A to right by two bits (ROR)
6	0000000000100000	Invert the bit-significance order of B
7	0000000001000000	Produce the result of XORing A and B
8	0000000010000000	Produce the summation of A and B, then decrease it by 4
9	0000000100000000	Produce all high bits on the output

4.5 Waveform



After simulation:

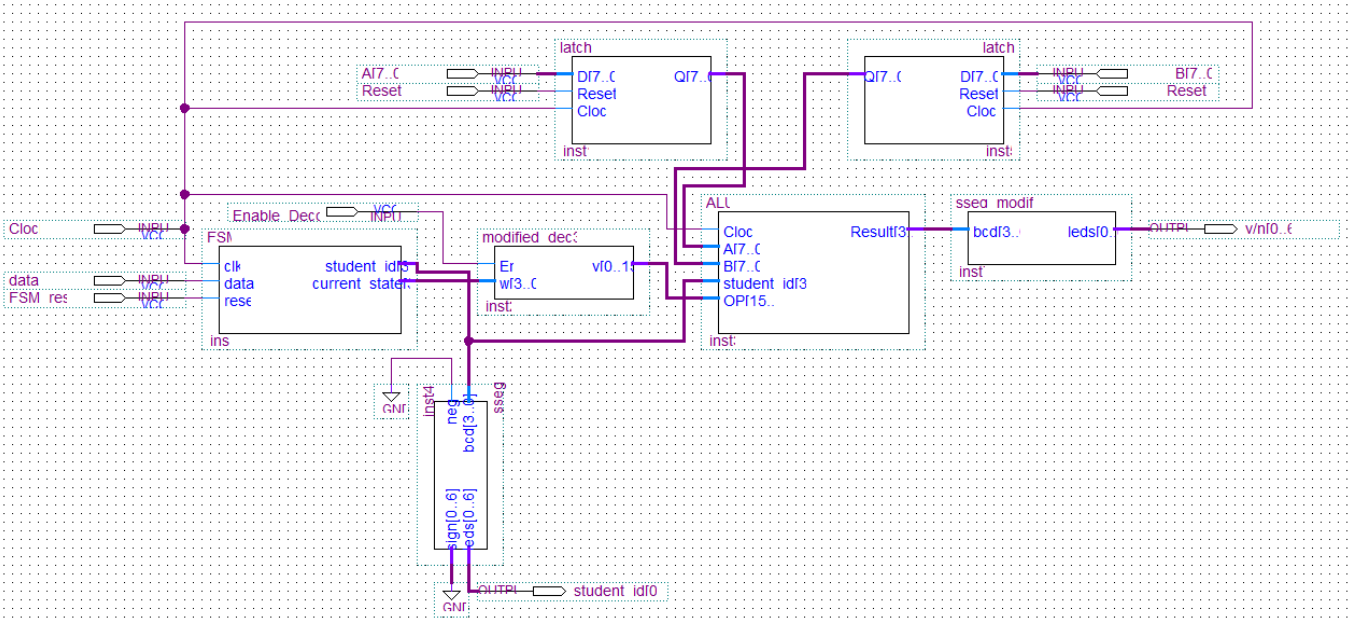


5. ALU 3

5.1 Purpose and Design of The Component

The purpose of the ALU is to perform mathematical operations on binary numbers. The ALU core takes two 8-bit inputs (A and B) from the storage unit and student ID output from the FSM, sub-component of the control unit. The ALU design was modified so that for each microcode, it displays 'y' if one of the two digits in the 8-bit input, A are less than the student ID output from FSM and 'n' otherwise. The 7-segment design was also modified so that 'y' or 'n' from the ALU gets displayed on a 7-segment display.

5.2 Block Schematic



5.3 Purpose of All Inputs and Outputs of This Component

Input

- Clock: It causes the ALU to either change or keep its output signal based on the values of the input signals during transition, which are passed on the rising edge of the clock cycles.
- A[7..0]: It represents the 8-bit input, A from the storage unit. A is the binary representation of the 6^{th} and 7^{th} digit of the student ID.
- B[7..0]: It represents the 8-bit input, B from the storage unit. B is the binary representation of the last two digits of student ID.
- OP[15..0]: OP is passed to ALU from the control unit. The ALU uses this input as the operations selector to select a specific function to perform, based on the microcode from the decoder.
- student_id[3..0]: The FSM outputs the student ID, which gets passed on to the ALU as its input to be used to perform the assigned function.

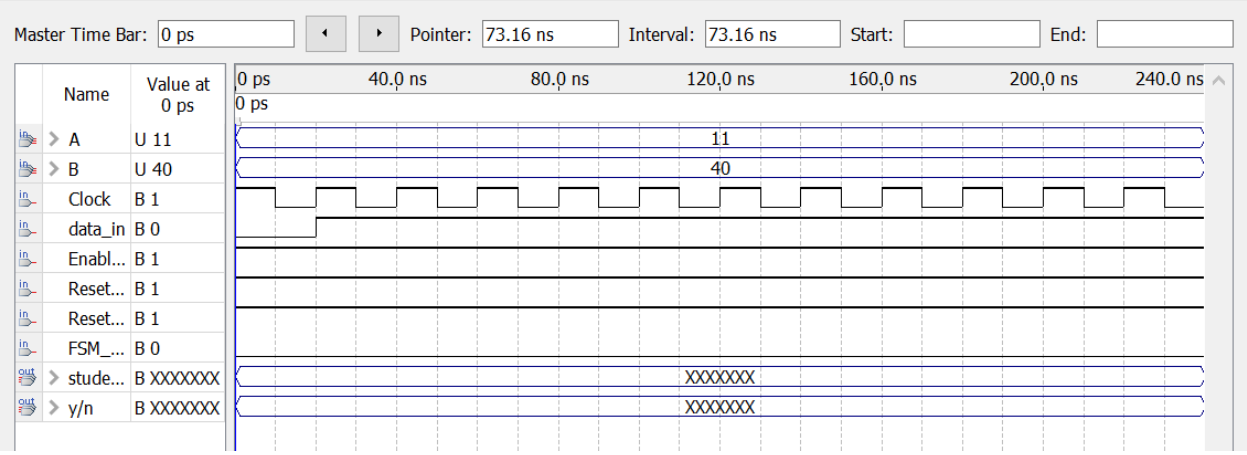
Output

- Result [3..0]: It passes a 4-bit output to the modified 7-segment display. The 4-bit number represents ‘y’ or ‘n’ that resulted by performing operations in the ALU, based on the assigned function.

5.4 Table of Microcode’s generated by decoder for ALU 3

Function #	Microcode	Boolean Operation (in ALU)
1	0000000000000001	'y' if one of the 2 digits of A are less than FSM output(student_id) and 'n' otherwise.
2	0000000000000010	
3	0000000000000100	
4	0000000000001000	
5	0000000000010000	
6	0000000000100000	
7	0000000001000000	
8	0000000010000000	
9	0000000100000000	

5.5 Waveform



After simulation: