Ryerson University Faculty of Science

Department of Physics

Course Number	COE 328
Course Title	Digital Systems
Semester/ Year	Fall 2023
Instructor	Professor R. Sedaghat
TA Name	Tien Loc Le

Lab/Tutorial Report No.	6

Report Title	Design of a Simple General-Purpose Processor

Section No.	7
Submission Date	12/04/2023
Due Date	12/04/2023

Student Name	Student ID	Signature
Ayoub Jibril Said	****09845	A.J.S

Table of contents

Introduction 3

Components 3

ALU_1 8

ALU_2 10

ALU_3 13

Conclusion 14

Introduction

The goal of this lab was to display one’s student number at the same time as displaying the result of various boolean functions involving the last 4 digits of one’s student number.

During this lab, the circuit components used were two latches, an ALU, and a control unit which consisted of a FSM of type mealy and a 4:16 decoder made of 2 3:8 decoders. The latches act as storage units, which are utilized to temporarily store input values. These input values are passed to the ALU to be used in the desired boolean operation.

The ALU is the component where all arithmetic and logical operations are to be implemented and applied based on the microcode assigned to the state that it is in. Its 8-bit output is then displayed on two 7-segment displays.

The control unit, consisting of a Mealy FSM and a 4:16 decoder, acts as a selector for the ALU, controlling which microcode/function is to be applied/conducted. The Mealy FSM acts as an up-counter, cycling through states 0-8 consecutively and feeding the current states to the 4x16 decoder, where the 16-bit output is fed to the ALU.

Components

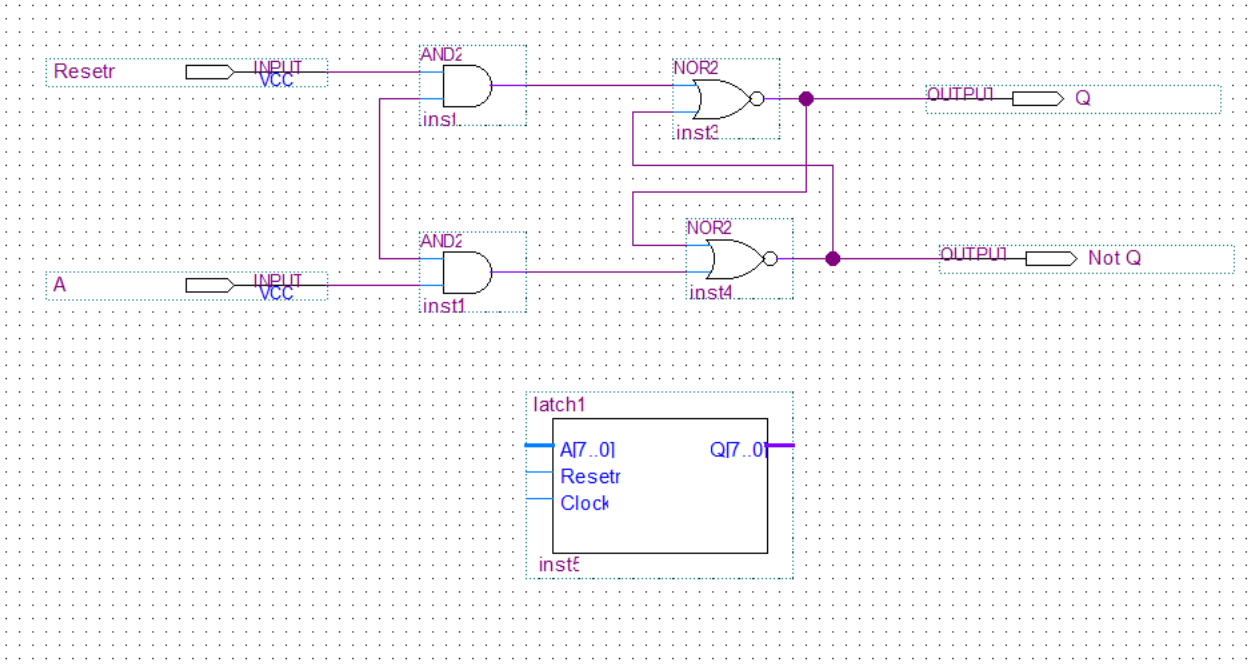
Latch1 and Latch2 act as storage units which temporarily store input values that will be passed to the ALU and used in the boolean operations. The 4:16 decoder outputs 16 bits to the ALU, determining which operation is to be conducted. The Mealy FSM up-counts from 0-8, and feeds its current state value to the 4:16 decoder.

Latch1 and Latch2

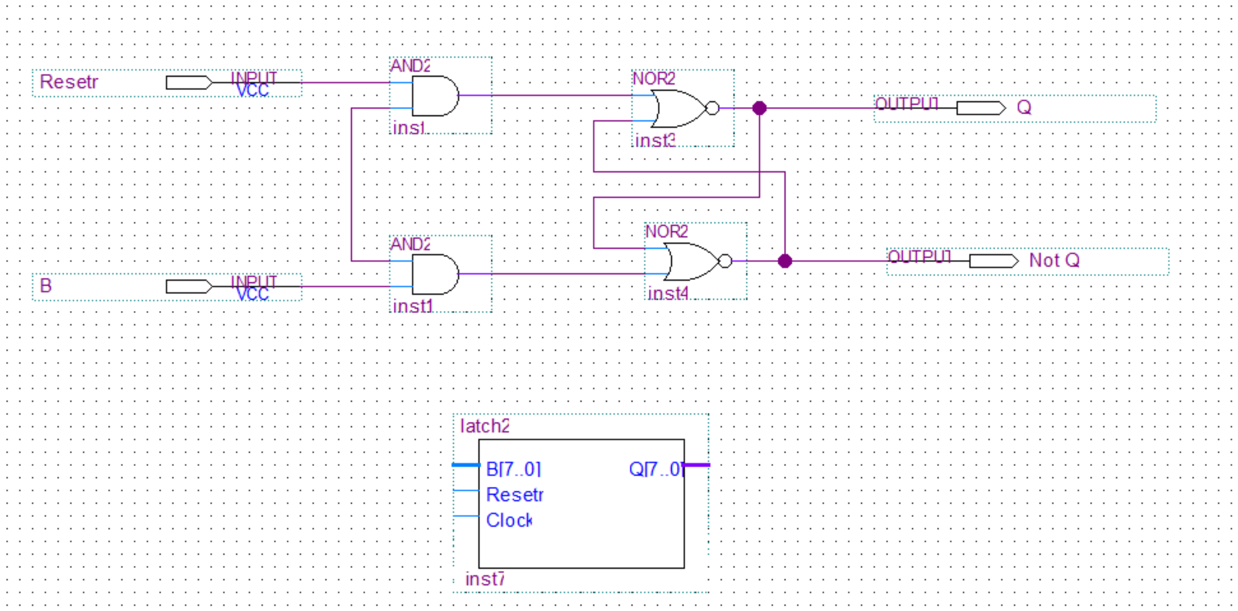
Truth Table of Latch1 and Latch2

Clock	A	Resetn	Q
0	x	x	Q(no change)
1	0	0	Q(no change)
1	0	1	0
1	1	0	1
1	1	1	x

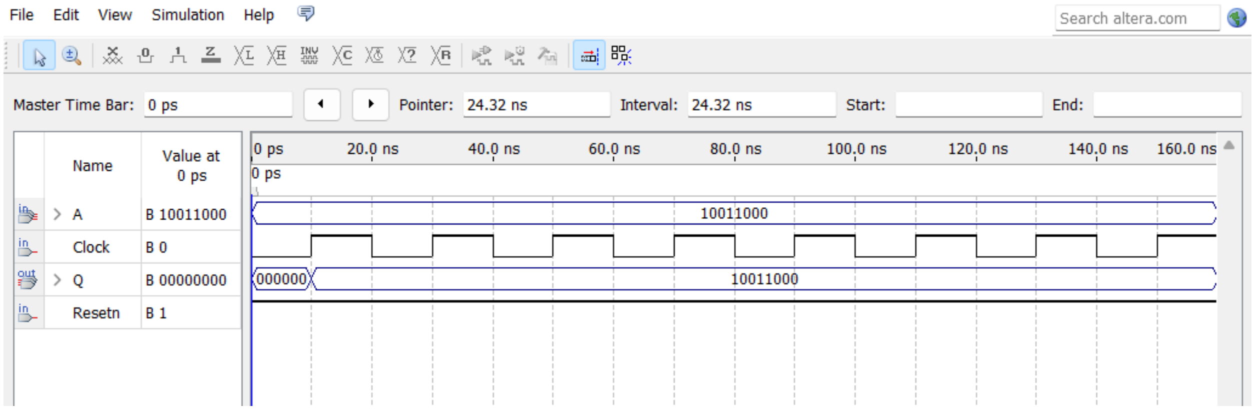
Block Diagram for Latch1



Block Diagram for Latch2



Waveform of Latch1



Master Time Bar: 0 ps

Pointer: 24.32 ns

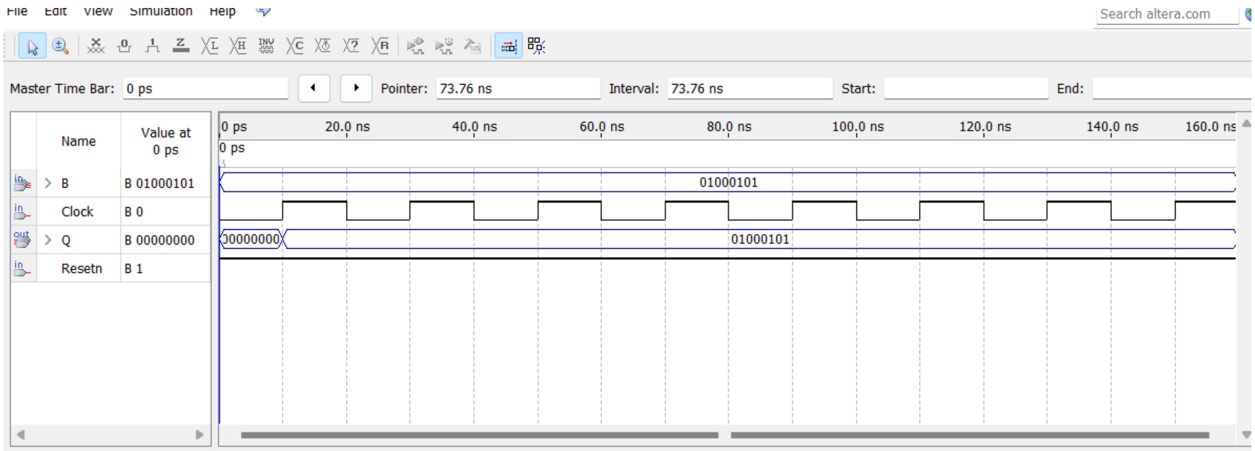
Interval: 24.32 ns

Start:

End:

Name	Value at 0 ps
A	B 10011000
Clock	B 0
Q	B 00000000
Resetn	B 1

Waveform of Latch2



Master Time Bar: 0 ps

Pointer: 73.76 ns

Interval: 73.76 ns

Start:

End:

Name	Value at 0 ps
B	B 01000101
Clock	B 0
Q	B 00000000
Resetn	B 1

4:16 Decoder

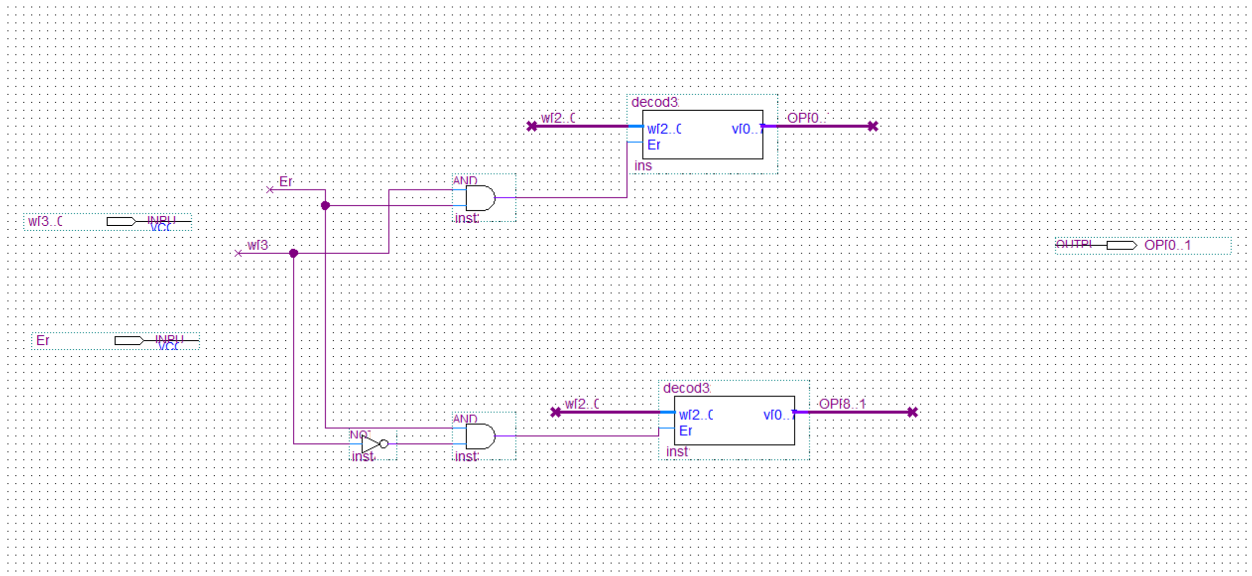
Truth Table for Decoder

En	w3	w2	w1	w0	Count															
					y15	y14	y13	y12	y11	y10	y9	y8	y7	y6	y5	y4	y3	y2	y1	y0
1	0	0	0	0	1	0	0	0	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	0
1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

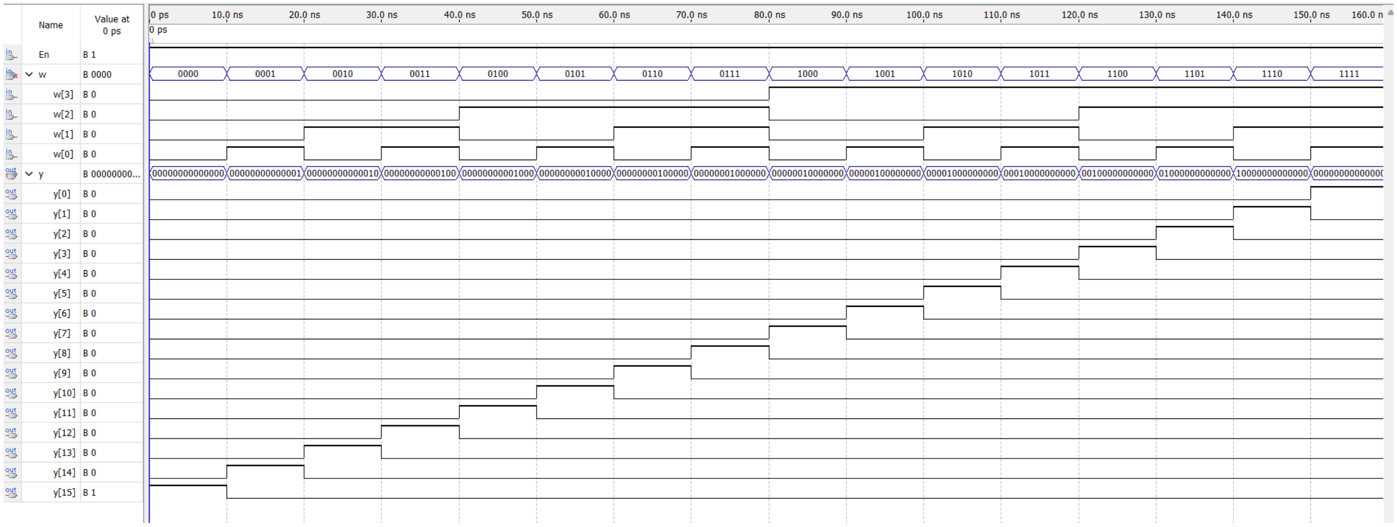
1	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
1	0	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
1	1	1	0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

Block Diagram of Decoder

The following 4:16 Decoder was made out of two 3:8 Decoders:



Waveform of Decoder

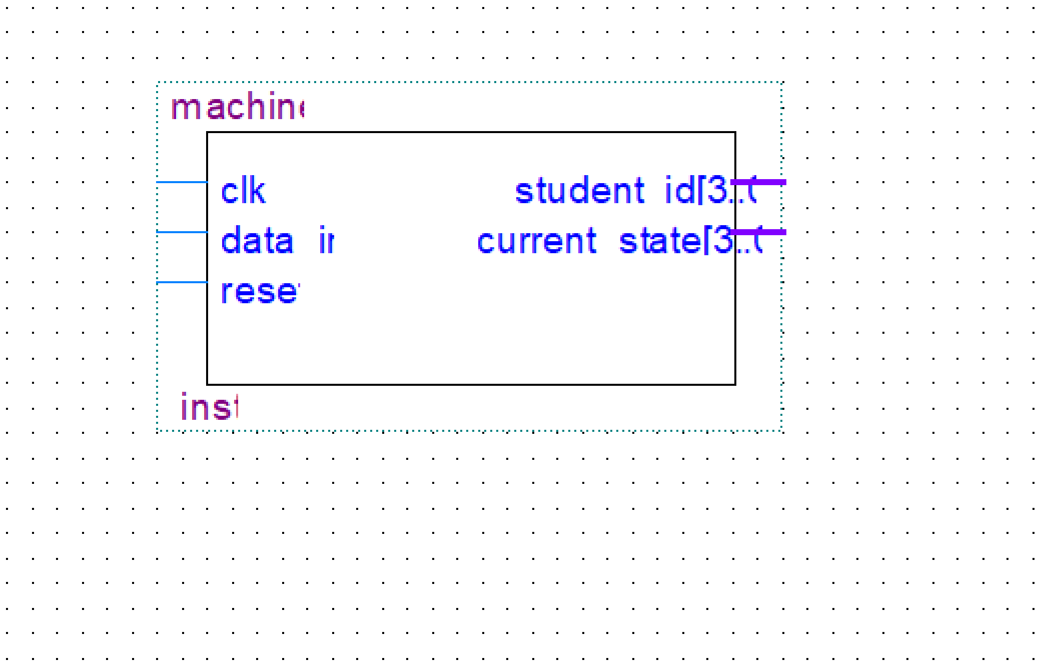


Finite State Machine (FSM)

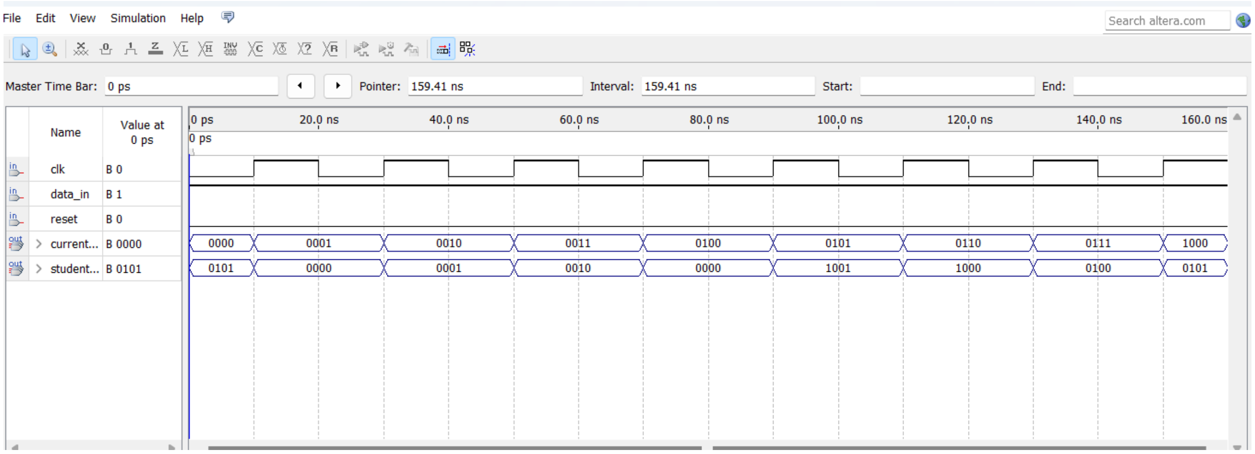
Truth Table for FSM

Current State	Next State		Output
	w=0	w=1	
0000	0000	0001	5
0001	0001	0010	0
0010	0010	0011	1
0011	0011	0100	2
0100	0100	0101	0
0101	0101	0110	9
0110	0110	0111	8
0111	0111	1000	4
1000	1000	0000	5

Block Diagram of FSM



Waveform of FSM



Master Time Bar: 0 ps
Pointer: 159.41 ns
Interval: 159.41 ns
Start:
End:

```
Name        Value at 0 ps   0 ps   20.0 ns  40.0 ns  60.0 ns  80.0 ns  100.0 ns  120.0 ns  140.0 ns  160.0 ns
clk         B 0
data_in     B 1
reset       B 0
current...  B 0000          0000   0001     0010     0011     0100     0101      0110      0111      1000
student...  B 0101          0101   0000     0001     0010     0000     1001      1000      0100      0101
```

ALU-1

The Arithmetic and Logical Unit (ALU) is the component of the GPU unit which implements all the arithmetic and logical operations. The ALU for this lab had a total of 5 inputs - Clk, A, B, student_id, and OP, and 3 outputs - Neg, R1, and R2.

Inputs “A” and “B” represent the value of the desired values of A and B set on the FPGA board. The input “OP” is the selector of the desired boolean operation. Lastly, “Clock” provides a time frame for how long each output is active.

Outputs “R1” and “R2” split the 8 bit result of the boolean operation into two 4-bit numbers. The values are passed to the SSEG component for display on the FPGA board. Output “Neg” was used to display a negative sign on the FPGA board if the output of the boolean operation/function yielded a negative value.

The goal of this problem set was to modify the ALU such that it conducts the correct boolean operation between the 8-bit values A and B for the microcode that it was assigned to, and outputs the correct result. To do so, the ALU uses the 16-bit value of the “current state” of the FSM as a selector for the operation that is to be conducted between A and B.

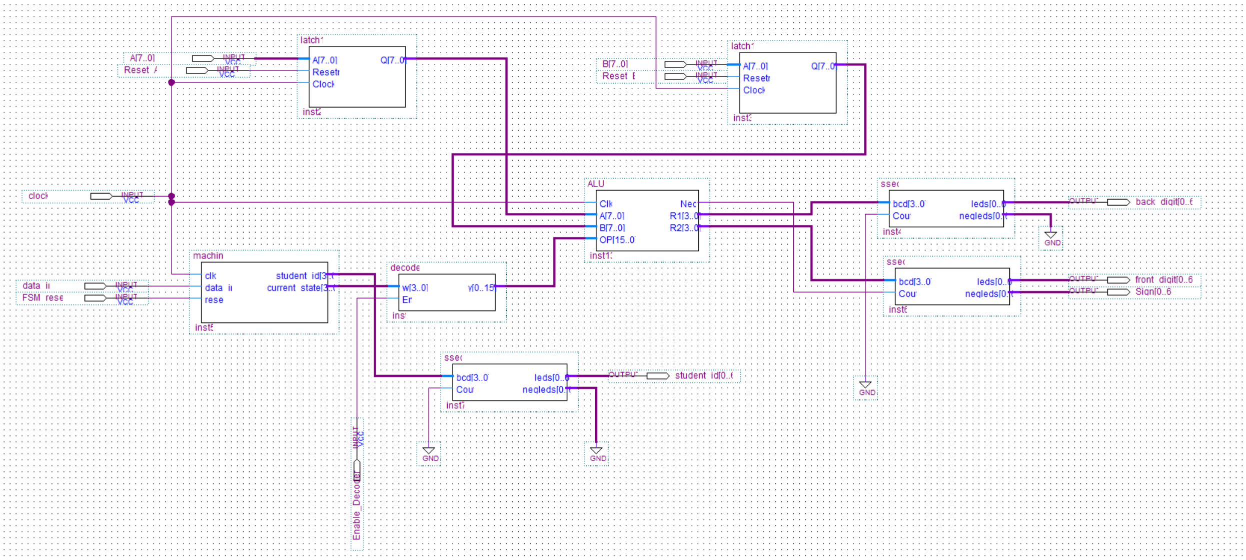


Table of Microcode generated by the decoder for ALU_1:

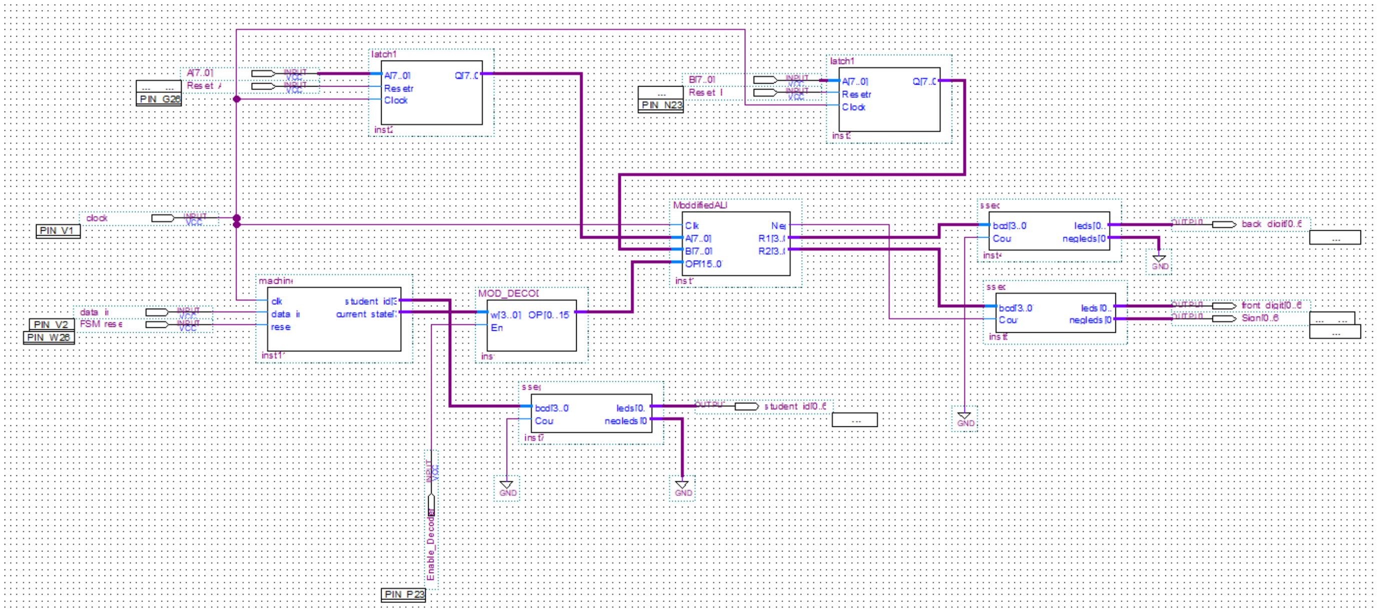
A	B	Boolean Functions	Output	Hexadecimal
10011000	01000101	sum(A,B)	11011101	dd
10011000	01000101	diff(A,B)	01010011	53
10011000	01000101	NotA	01100111	67
10011000	01000101	Not(A.B)	11111111	FF
10011000	01000101	Not(A+B)	00100010	22
10011000	01000101	A.B	00000000	00

10011000	01000101	A xor B	11011101	dd
10011000	01000101	A+B	11011101	dd
10011000	01000101	Not(A xor B)	00100010	22

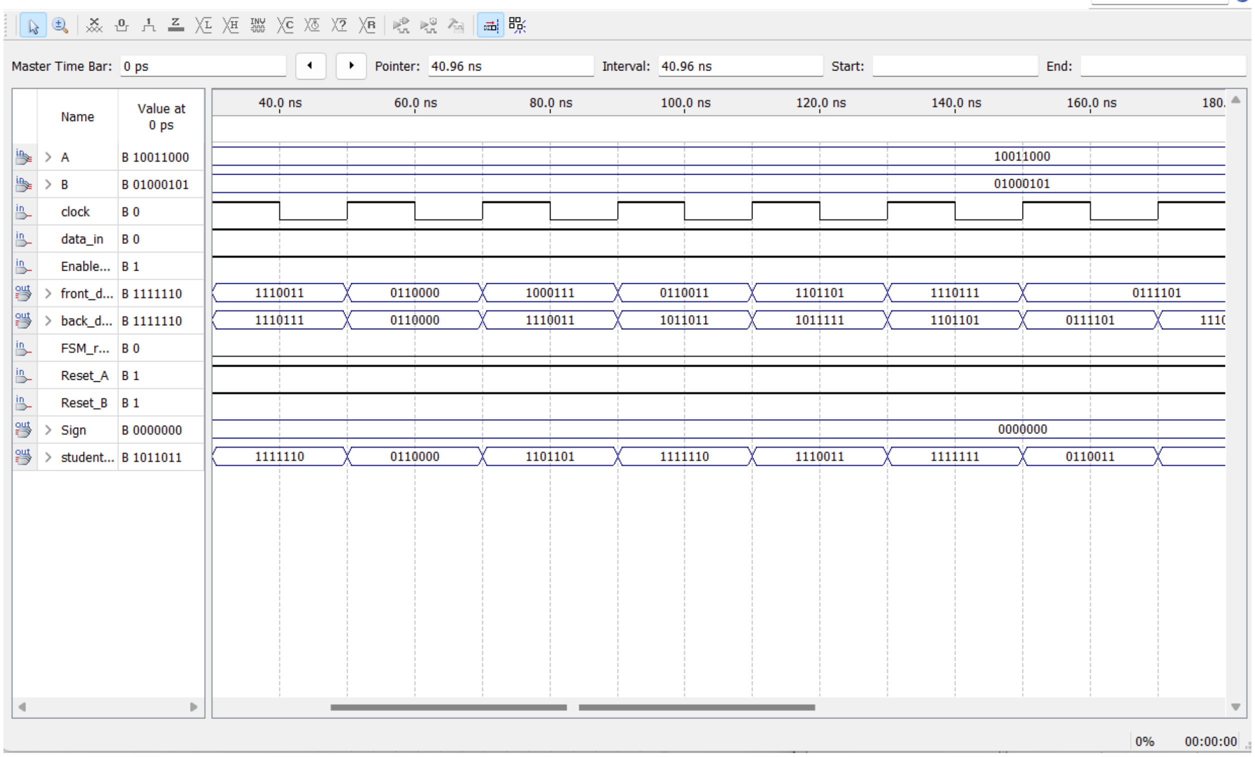
ALU-2

The design of the ALU component in this problem set is the same as the design for the first problem set. The purpose of the ALU component in this problem set is to perform a different set of boolean functions to inputs A and B corresponding to the assigned microcode.The purpose of the inputs and outputs are the same as in ALU_1. The assigned microcode is Table a.

Block Diagram



Waveform



Master Time Bar: 0 ps Pointer: 40.96 ns Interval: 40.96 ns Start: End:

Name	Value at 0 ps
A	B 10011000
B	B 01000101
clock	B 0
data_in	B 0
Enable...	B 1
front_d...	B 1111110
back_d...	B 1111110
FSM_r...	B 0
Reset_A	B 1
Reset_B	B 1
Sign	B 0000000
student...	B 1011011

Table of Microcode generated by the decoder for ALU_2:

A	B	Boolean Functions	Output	Hexadecimal
10011000	01000101	Increment A by 2	10011010	9A
10011000	01000101	Shift B to right by two bits, input bit = 0 (SHR)	00010001	11
10011000	01000101	Shift A to right by four bits, input bit = 1 (SHR)	11111001	F9
10011000	01000101	Find the smaller value of A and B and produce the results (Min(A,B))	01000101	45
10011000	01000101	Rotate A to right by two bits (ROR)	00100110	26

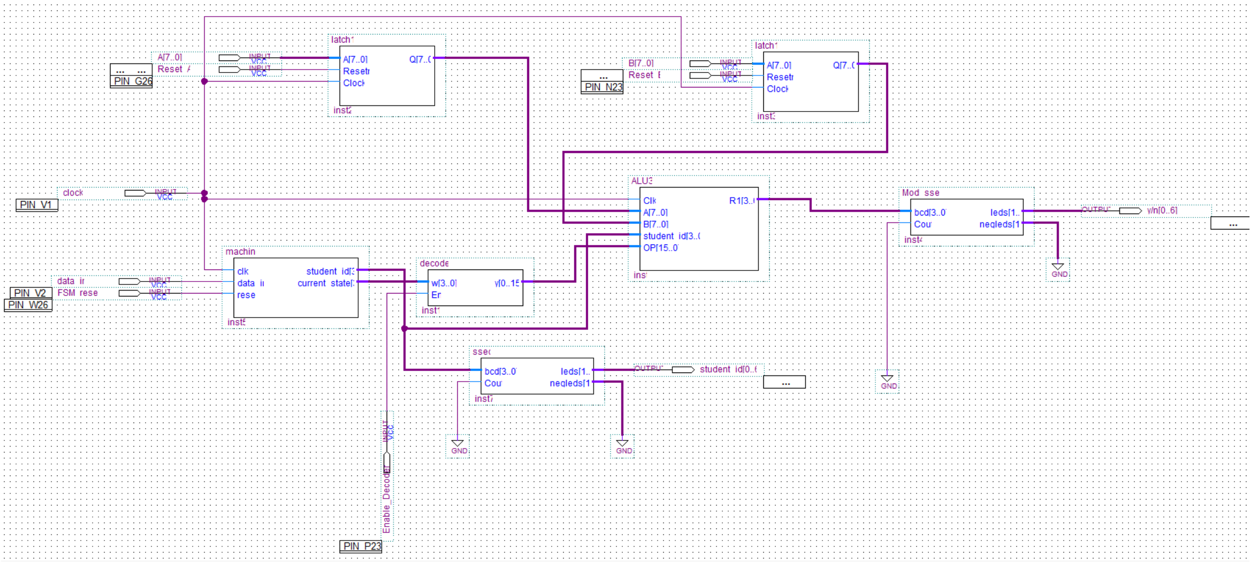
10011000	01000101	Invert the bit-significance order of B	10100010	A2
10011000	01000101	Produce the result of XORing A and B	11011101	DD
10011000	01000101	Produce the summation of A and B, then decrease it by 4	11011001	D9
10011000	01000101	Produce all high bits on the output	11111111	FF

ALU-3

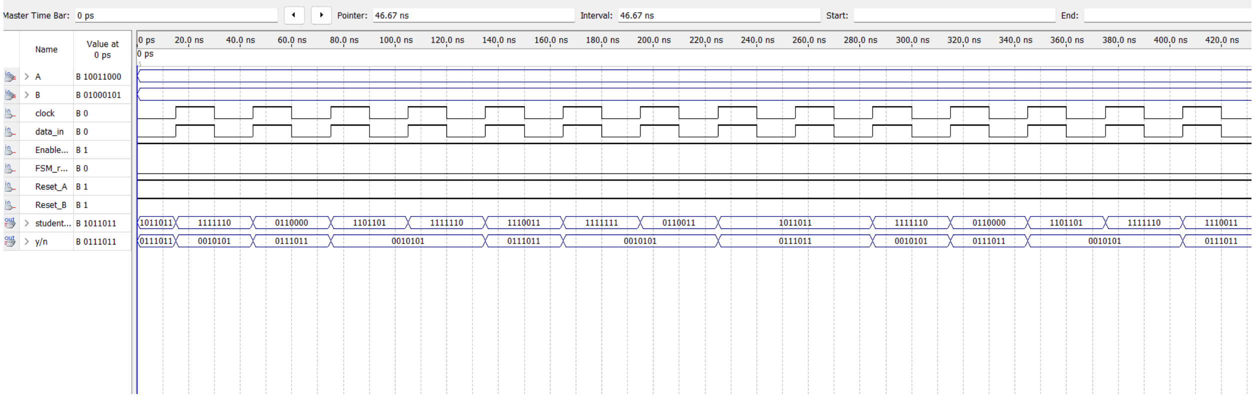
The task to be accomplished in problem set 3 was to modify the ALU such that it would be able to check the digits of the student number for being either odd or even over 9 clock cycles. If the digit of the student number is odd, then ‘y’ is displayed on a single seven-segment display, and if it is even, then ‘n’ is displayed. To do so, an input “student_id” was added to the ALU, and conditional statements were added to the ALU code. Instead of conducting a boolean function assigned to the current state that is passed to the ALU between A and B, the ALU simply checks the input value of “student_id” for being either odd or even.

Inputs “A” and “B” represent the desired values based on the last 4 digits of the student number. The input “OP” selects the student number to be checked for either odd or even. Lastly, “Clk” provides a time frame for how long each output is active. Output “R1” is used to display ‘y’ or ‘n’ on the seven-segment display.

Block Diagram



Waveform



Master Time Bar: 0 ps Pointer: 46.67 ns Interval: 46.67 ns Start: End:

Name	Value at 0 ps
A	B 10011000
B	B 01000101
clock	B 0
data_in	B 0
Enable...	B 1
FSM_r...	B 0
Reset_A	B 1
Reset_B	B 1
student...	B 1011011
y/n	B 0111011

```
student...  1011011 | 1111110 | 0110000 | 1101101 | 1111110 | 1110011 | 1111111 | 0110011 | 1011011 | 1111110 | 0110000 | 1101101 | 1111110 | 1110011
y/n         0111011 | 0010101 | 0111011 | 0010101 | 0111011 | 0010101 | 0111011 | 0010101 | 0111011 | 0010101 | 0111011
```

Table of Microcode generated by the decoder for ALU_3:

Student number	Output
5	y
0	n
1	y
2	n
0	n
9	y

8	n
4	n
5	y

Conclusion

The objective of this laboratory experiment was to construct basic general-purpose processors with specifications delineated in three sets of problems. The completion of these problem sets necessitated the implementation of diverse Arithmetic Logic Unit (ALU) cores capable of processing inputs "A" and "B" to execute Boolean functions, modifications, and display results on a seven-segment display. Serving as the concluding laboratory session, it assessed comprehension of concepts acquired throughout the entire semester.

The finalized designs employed in this experiment incorporated elements from prior labs, including the seven-segment display, finite state machine, and decoder. Additionally, new components like latches were introduced. Successful completion of this lab demanded a comprehensive understanding of how each individual component functions and interconnects with others.