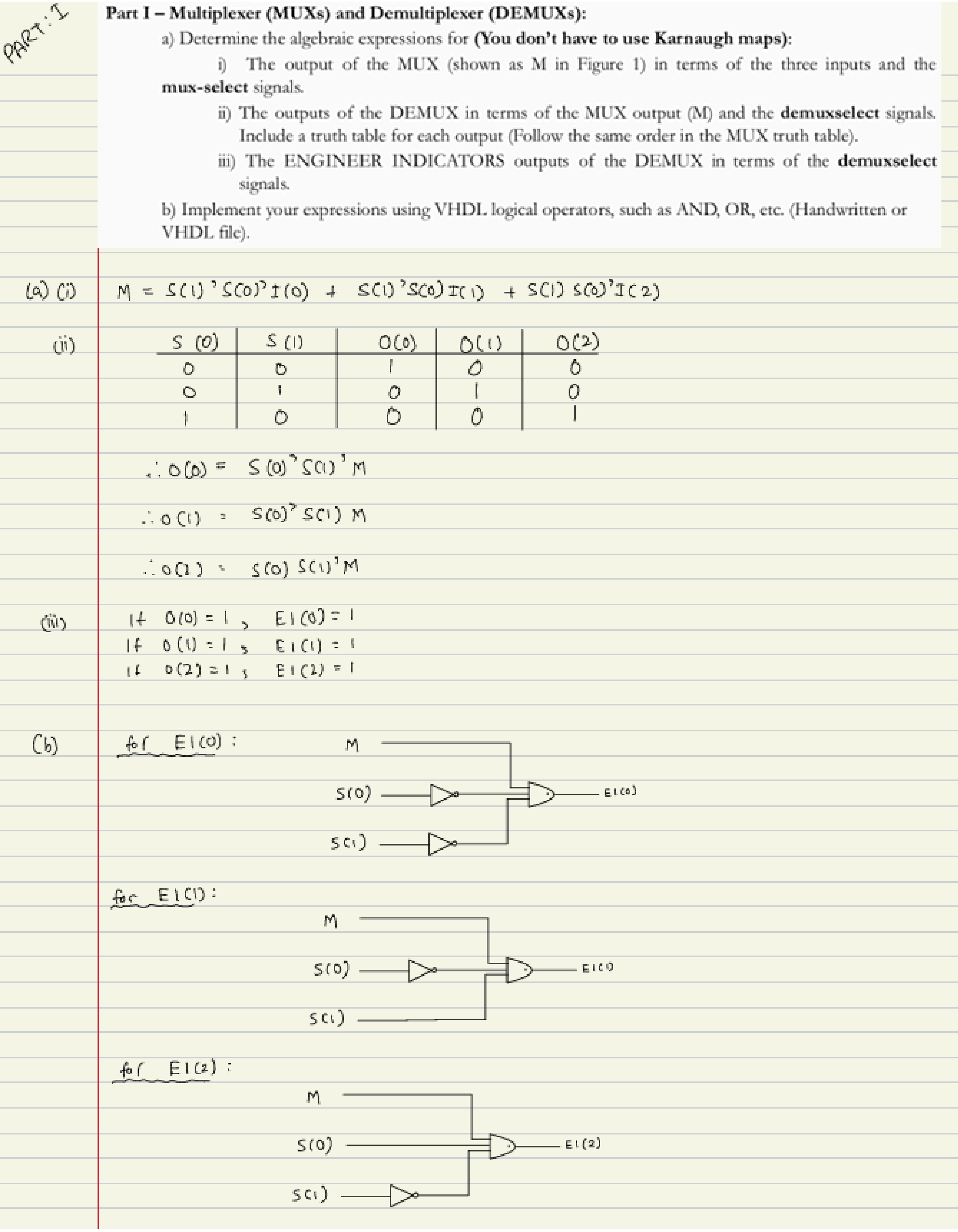
ECE 210 LAB-2 REPORT

Name: Misbah Ahmed Nauman
Student ID: 1830574
CCID: misbahah

Name: Muhammad Abdur Rab Siddiqui
Student ID: 1804733
CCID: msiddiq6

Section Number: D25
Lab Number: 2
Lab Date: 10.22.2024

PRE-LAB ASSIGNMENT



PART: I

Part I – Multiplexer (MUXs) and Demultiplexer (DEMUXs):

a) Determine the algebraic expressions for **(You don't have to use Karnaugh maps)**:

i) The output of the MUX (shown as M in Figure 1) in terms of the three inputs and the **mux-select** signals.

ii) The outputs of the DEMUX in terms of the MUX output (M) and the **demuxselect** signals. Include a truth table for each output (Follow the same order in the MUX truth table).

iii) The ENGINEER INDICATORS outputs of the DEMUX in terms of the **demuxselect** signals.

b) Implement your expressions using VHDL logical operators, such as AND, OR, etc. (Handwritten or VHDL file).

(a) (i) $M = S(1)'S(0)'I(0) + S(1)'S(0)I(1) + S(1)S(0)'I(2)$

(ii)

S (0)	S (1)	O(0)	O(1)	O(2)
0	0	1	0	0
0	1	0	1	0
1	0	0	0	1

$\therefore O(0) = S(0)'S(1)'M$

$\therefore O(1) = S(0)'S(1)M$

$\therefore O(2) = S(0)S(1)'M$

(iii) If O(0) = 1, EI(0) = 1
If O(1) = 1, EI(1) = 1
If O(2) = 1, EI(2) = 1

(b) for EI(0) :

for EI(1) :

for EI(2) :

PART : II

Part II – Lab Access Control:

a) Refer to Part II of BACKGROUND section (later in this document) to help understand design requirements and complete TRUTH TABLE below comprising five inputs (**C0, C1, K0, K1, K2**) and three outputs (**Alarm, Lab0_Unlock, Lab1_Unlock**) signals relating to the problem discussed. **To derive the logical expressions, you may either use karnaugh maps (provided in the appendix) or use the direct method. A valid expression and circuit will be accepted even if it is not in the minimal form.**

Note: Some of the entries in the table are completed to establish clarity on requirements.

b) Provide the algebraic expressions and draw the logic circuit for the following outputs:

1. **Lab0_Unlock**
2. **Lab1_Unlock**
3. **Alarm**

Hints:
- → You may use the "**Lab0_Unlock**" and "**Lab1_Unlock**" signals in your expression for the "**Alarm**" signal.
- → A valid expression and circuit will be accepted even if it is not in the minimal form.

Workings:

Lab 0 Unlock

K0=0		K1K2			
		00	01	11	10
C0C1	00	0	0	0	0
	01	0	0	0	0
	11	0	0	0	0
	10	0	0	0	0

K0=1		K1K2			
		00	01	11	10
C0C1	00	0	0	0	0
	01	1	0	0	1
	11	0	0	0	0
	10	0	0	0	0

Alarm:

K0=0		K1K2			
		00	01	11	10
C0C1	00	0	0	0	0
	01	1	1	1	1
	11	1	1	1	1
	10	1	0	0	1

K0=1		K1K2			
		00	01	11	10
C0C1	00	0	0	0	0
	01	0	1	1	0
	11	1	1	1	1
	10	1	1	1	1

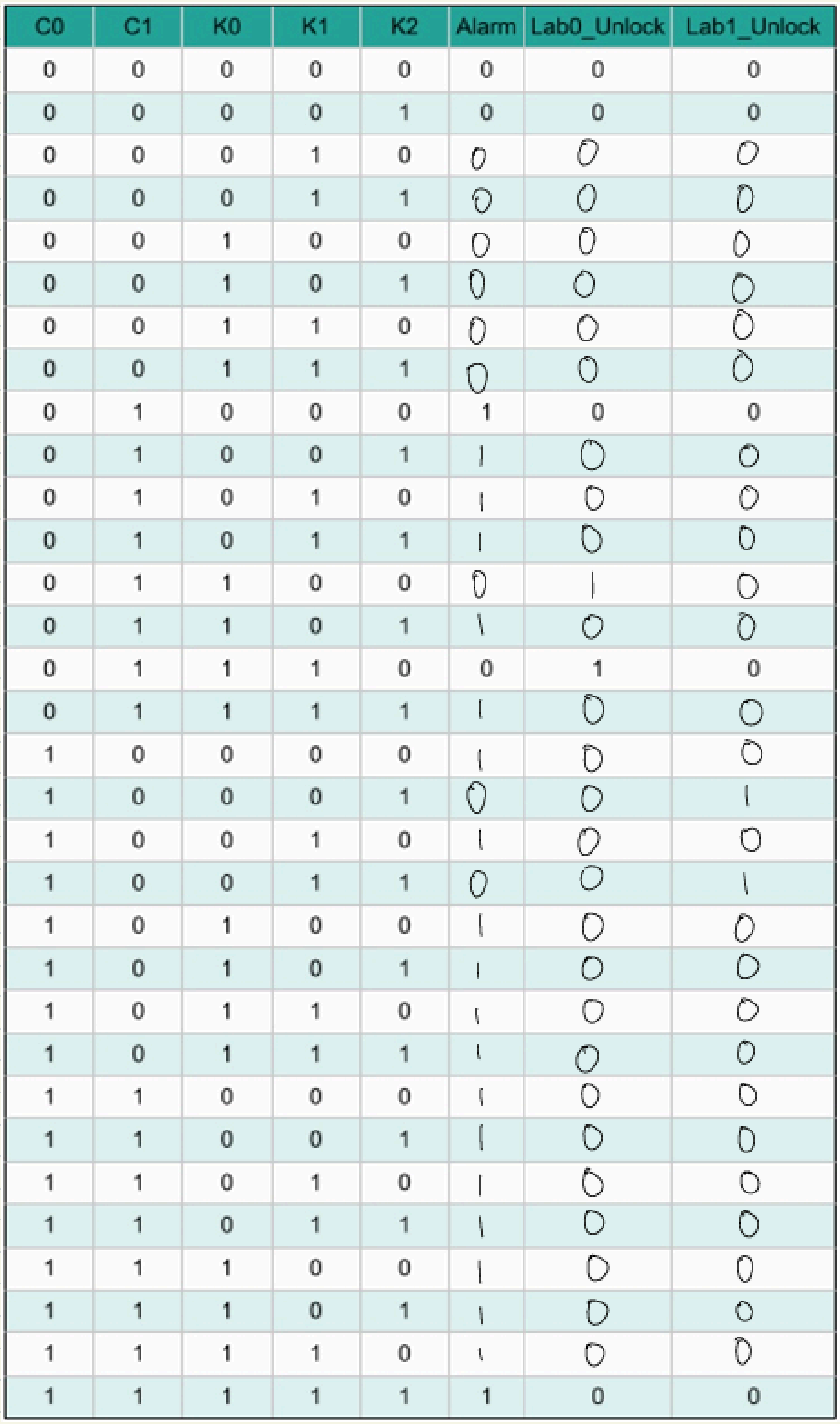
Lab 1 Unlock

K0=0		K1K2			
		00	01	11	10
C0C1	00	0	0	0	0
	01	0	0	0	0
	11	0	0	0	0
	10	0	1	1	0

K0=1		K1K2			
		00	01	11	10
C0C1	00	0	0	0	0
	01	0	0	0	0
	11	0	0	0	0
	10	0	0	0	0

PART: II

(a)

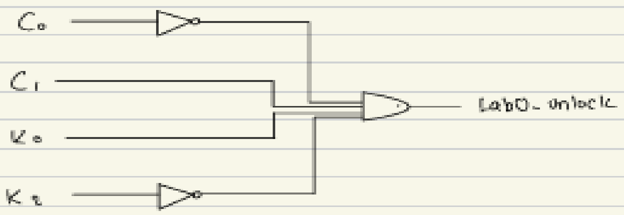


C0	C1	K0	K1	K2	Alarm	Lab0_Unlock	Lab1_Unlock
0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0
0	0	0	1	0	0	0	0
0	0	0	1	1	0	0	0
0	0	1	0	0	0	0	0
0	0	1	0	1	0	0	0
0	0	1	1	0	0	0	0
0	0	1	1	1	0	0	0
0	1	0	0	0	1	0	0
0	1	0	0	1	1	0	0
0	1	0	1	0	1	0	0
0	1	0	1	1	1	0	0
0	1	1	0	0	0	1	0
0	1	1	0	1	1	0	0
0	1	1	1	0	0	1	0
0	1	1	1	1	1	0	0
1	0	0	0	0	1	0	0
1	0	0	0	1	0	0	1
1	0	0	1	0	1	0	0
1	0	0	1	1	0	0	1
1	0	1	0	0	1	0	0
1	0	1	0	1	1	0	0
1	0	1	1	0	1	0	0
1	0	1	1	1	1	0	0
1	1	0	0	0	1	0	0
1	1	0	0	1	1	0	0
1	1	0	1	0	1	0	0
1	1	0	1	1	1	0	0
1	1	1	0	0	1	0	0
1	1	1	0	1	1	0	0
1	1	1	1	0	1	0	0
1	1	1	1	1	1	0	0

(b) 1. for Lab0 _unlock:

$$\text{Lab0 - Unlock} = C_0' C_1 K_0 (K_1' K_2' + K_1 K_2')$$

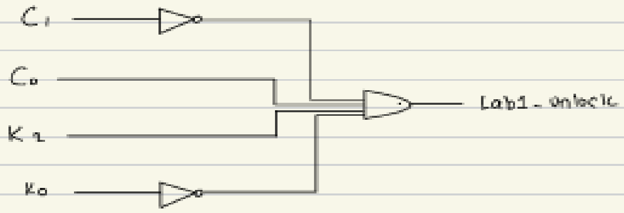
$$= C_0' C_1 K_0 K_2'$$



2. for Lab1_Unlock:

$$\text{Lab1 - Unlock} = C_0 C_1' K_0' (K_1' K_2 + K_1 K_2)$$

$$= C_0 C_1' K_0' K_2$$



3. For Alarm:

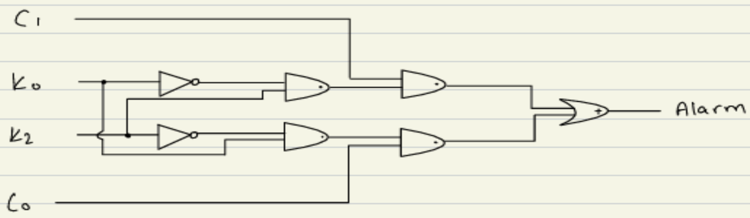
$$\text{Alarm} = C_1 K_0' + C_0 K_0' K_2' + C_0 K_0 + C_1 K_0 K_2$$

$$= C_1 (K_0' + K_0 K_2) + C_0 (K_0 + K_0' K_2')$$

$$= C_1 (K_0' + K_0)(K_0' + K_2) + C_0 (K_0 + K_0')(K_0 + K_2')$$

$$= C_1 K_0' + C_1 K_2 + C_0 K_0 + C_0 K_2'$$

$$= C_1 (K_0' + K_2) + C_0 (K_0 + K_2')$$



The aforementioned solution serves as the basis for the pre-lab assignment for this experiment, with all circuits and calculations in this lab derived from these results. During the lab, we received the correct pre-lab solutions and identified mistakes in our own work. However, the lab was conducted using the accurate solutions from the pre-lab.

ABSTRACT

This lab report presents the design and implementation of a multiplexer (MUX) and a demultiplexer (DEMUX) using the Zybo Z7 FPGA board, along with a lab access control system. The objective was to demonstrate combinational logic design principles through both simulation and hardware implementation. The MUX was constructed to select one of three input signals based on two select lines, while the DEMUX directed the selected input to one of three output channels based on a data select line. Additionally, the lab access control utilized push buttons to control access, simulating a secure entry system. Simulation results were obtained using the provided test-bench files, demonstrating the correct functionality of the designs. The hardware implementation utilized the Analog Discovery 2 to provide external input signals, allowing real-time signal monitoring through the Zybo board's LEDs.

This report concludes that both designs effectively achieved their intended functionality, with hardware successfully replicating the simulation waveforms and validating the theoretical principles of combinational logic.

INTRODUCTION

The design and implementation of multiplexers (MUX) and demultiplexers (DEMUX) are fundamental in digital logic design, with significant applications in communication systems, data routing, and signal processing. This lab aims to demonstrate the practical application of these components using a Zybo Z7 FPGA board, emphasizing the principles of combinational logic design. The objectives include simulating and implementing a MUX and DEMUX to manage data signals effectively, showcasing the versatility of digital circuits in engineering applications.

The experiment involves two primary components: the MUX, which selects one of several input signals based on control signals, and the DEMUX, which distributes the selected input signal to one of several outputs. This operation relies on the principles of Boolean algebra and logic gate functionality. The design utilizes three inputs (I) for the MUX, with two select lines (S) determining which input is sent to the output. The DEMUX then takes this output and directs it to one of its outputs based on its data select (DS) inputs.

This lab will also explore lab access control systems, further illustrating the practical application of logic design in enhancing security measures. By comparing traditional discrete component methods with modern FPGA implementations, this experiment highlights the advantages of using programmable logic devices for rapid prototyping and flexibility in design.

Overall, this experiment aims to solidify the understanding of MUX and DEMUX operations, their practical applications, and their importance in the realm of digital logic design.

DESIGN SECTION

The goal of this lab was to design and implement a multiplexer (MUX) and demultiplexer (DEMUX) system using the Zybo Z7 FPGA board. This system is intended to effectively route and control digital signals based on select inputs, demonstrating the principles of combinational logic design in a practical context.

CONSTRAINTS

- **Input Signals**: The MUX will accept three input signals (I0, I1, I2), which will be provided through the Analog Discovery 2 (AD2) tool via PMOD ports. The select lines (S0, S1) will be connected to the first two switches on the Zybo board.
- **Output Signals**: The DEMUX will route the selected input signal to one of three output lines (O0, O1, O2), which will be mapped to the corresponding LEDs on the Zybo board.
- **Software Constraints**: The design must comply with the requirements of the provided test-bench files for simulation and validation of the design.

METHODOLOGY

- **Circuit Design:** A MUX was designed with three data inputs (I0, I1, I2) and two select lines (S0, S1). The output of the MUX is connected to the input of the DEMUX, which has its own select lines (DS0, DS1) to determine which output (O0, O1, O2) receives the signal.
- **Simulation**: A test-bench was provided to simulate the MUX and DEMUX functionalities, validating the design logic under various input conditions. This helped ensure the expected behavior before physical implementation.
- **Implementation**: The VHDL design was synthesized and implemented on the Zybo Z7 FPGA. The constraint file was applied to map the physical pins on the FPGA to the appropriate switches and LEDs.
- **Testing**: The hardware was tested by providing input signals via the AD2 and using the switches to control the MUX and DEMUX operation. Observations were made on the LED outputs to confirm that the correct signals were being routed as intended.

EXPERIMENTAL PROCEDURE AND EQUIPMENTS

EQUIPMENTS

- Zybo Z7 FPGA Board
- Analog Discovery 2 (Digilent)
- VHDL Development Software (Vivado)
- Waveform Software
- Resistors

PROCEDURE

1. **Setup**: Begin by assembling the Zybo Z7 FPGA board and connecting it to the computer. Ensure that the Analog Discovery 2 is also connected properly to the board using the appropriate PMOD interface.
2. **Circuit Configuration**: Using the schematic design, connect the input signals (I0, I1, I2) to the corresponding PMOD pins on the FPGA board. Connect the two select lines (S0, S1) to the push-button switches located on the Zybo board. Set up the output pins to connect to the three LEDs (O0, O1, O2) through the resistors to limit the current.
3. **VHDL Design**: Open Xilinx Vivado and create a new project. Import the test-bench file and write the VHDL code for the MUX and DEMUX designs. Ensure that the constraints file correctly maps the inputs and outputs to the appropriate FPGA pins.
4. **Simulation**: Before programming the FPGA, simulate the design using the test-bench files provided. Verify that the outputs react correctly to the changes in inputs and select lines.
5. **Synthesis and Implementation**: After successful simulation, proceed to synthesize the design. Once the synthesis is completed without errors, implement the design onto the Zybo Z7 FPGA.
6. **Testing**: After programming the FPGA, use the Analog Discovery 2 to generate input signals. Activate the select switches and observe the LED outputs. Record the behavior of the system, ensuring that each input corresponds correctly to the expected output as defined by the MUX and DEMUX functionality.

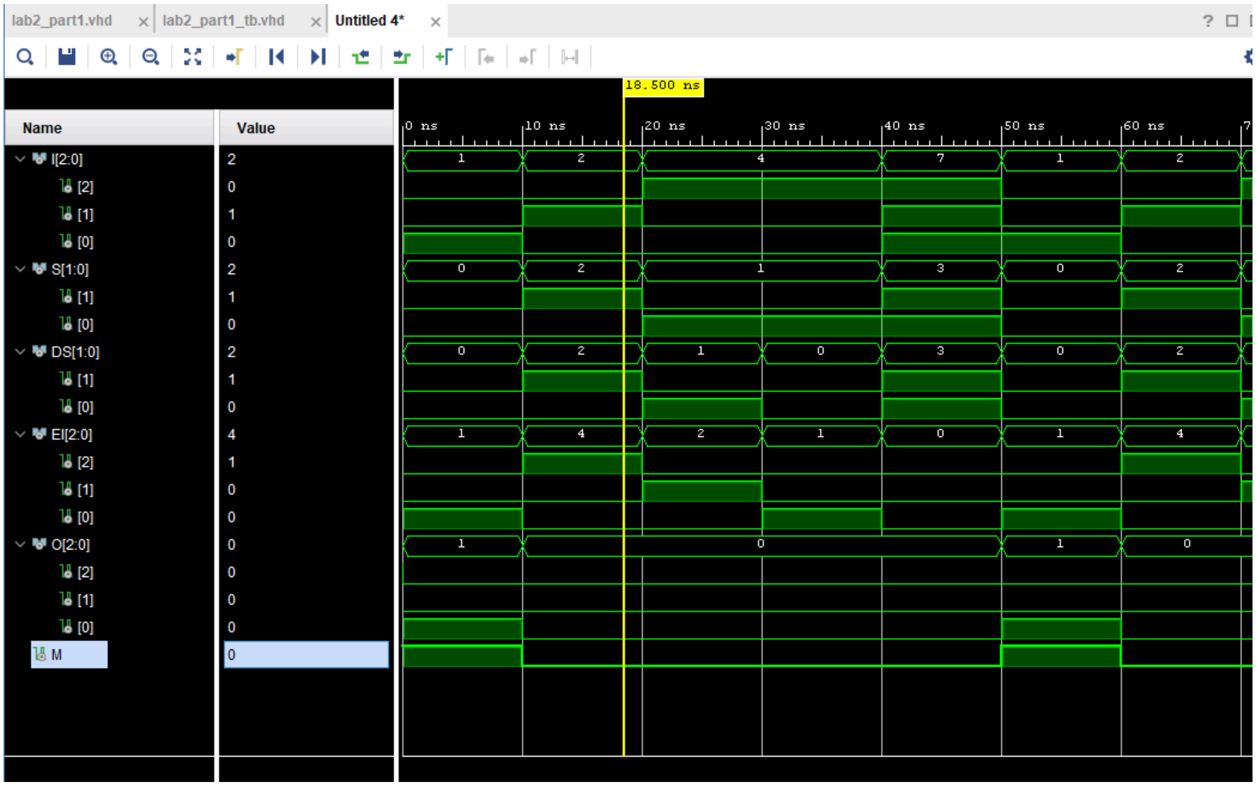
RESULTS

The experiment successfully demonstrated the functionality of the multiplexer (MUX) and demultiplexer (DEMUX) designs as implemented on the Zybo Z7 FPGA board. The following observations and results were recorded during the testing phase.

Key Observations

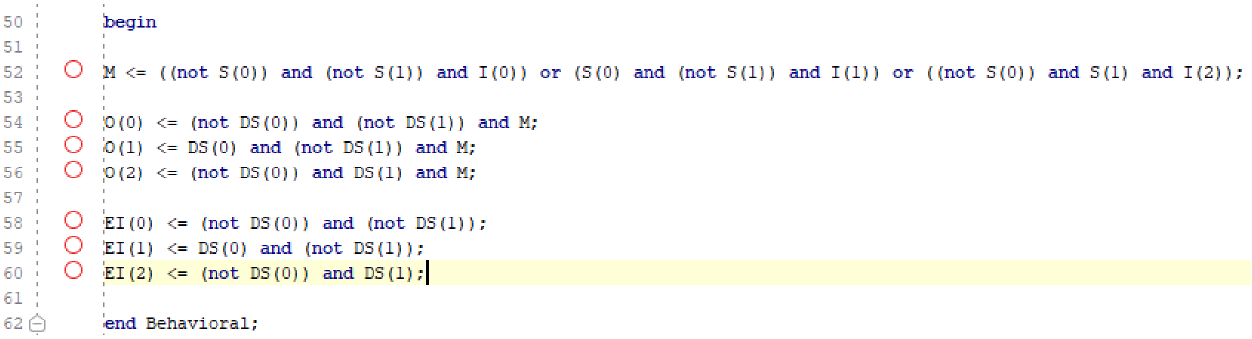
- **MUX Output Behavior**: The MUX was tested with various combinations of the input signals (I0, I1, I2) while altering the select lines (S0, S1). The expected output (O0, O1, O2) matched the theoretical results, confirming that the MUX correctly selected the desired input based on the select line configuration.

- **LED Indicator**: Each output LED corresponded to the active input, illuminating correctly as per the input and select line states. The observed outputs were as follows:
    - When S0 = 0, S1 = 0, I0 was selected, and O0 was lit.
    - When S0 = 1, S1 = 0, I1 was selected, and O1 was lit.
    - When S0 = 0, S1 = 1, I2 was selected, and O2 was lit.



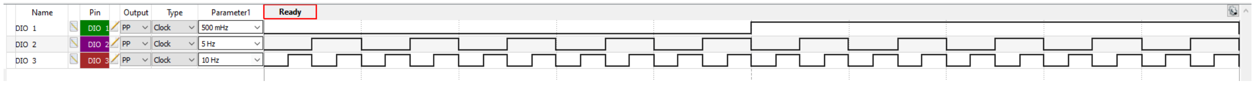
Name	Value
I[2:0]	2
[2]	0
[1]	1
[0]	0
S[1:0]	2
[1]	1
[0]	0
DS[1:0]	2
[1]	1
[0]	0
EI[2:0]	4
[2]	1
[1]	0
[0]	0
O[2:0]	0
[2]	0
[1]	0
[0]	0
M	0

Fig 2.1 : Screen-shot of a simulation window taken in the Vivado software for Part I.



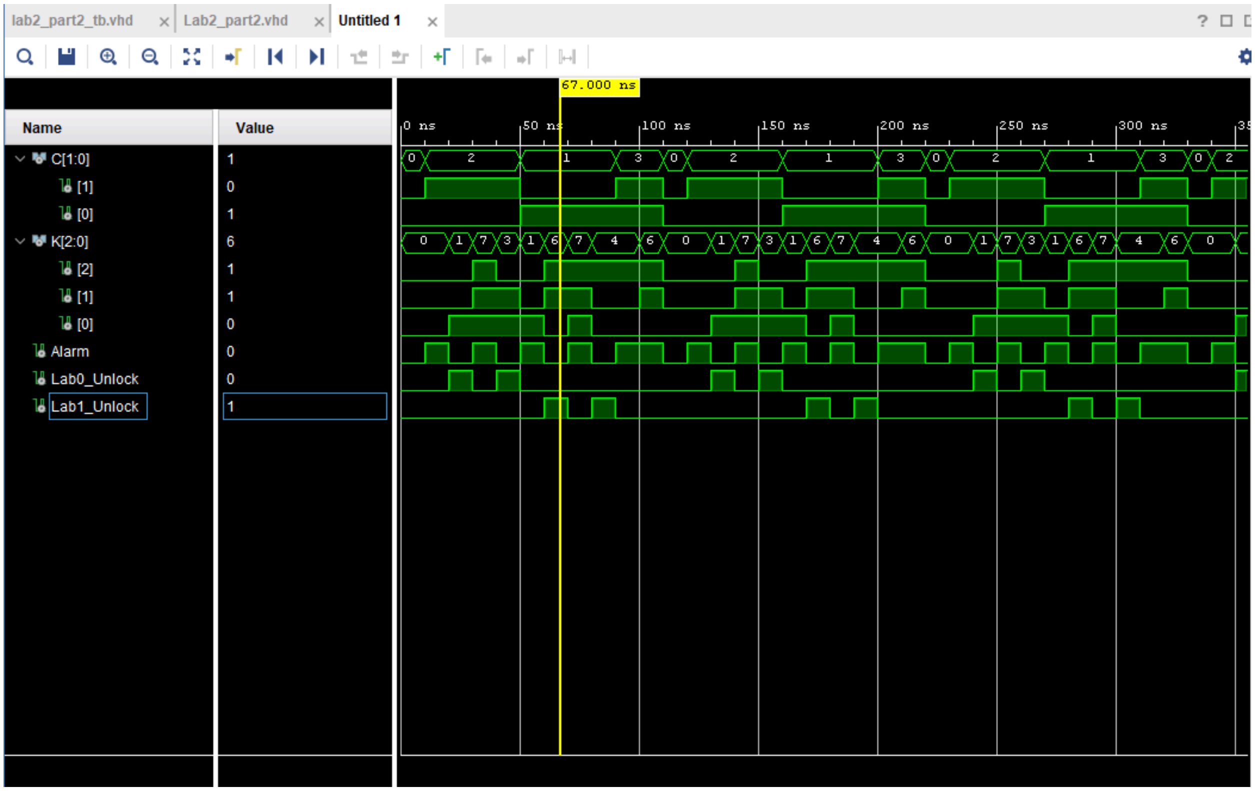
```
begin

M <= ((not S(0)) and (not S(1)) and I(0)) or (S(0) and (not S(1)) and I(1)) or ((not S(0)) and S(1) and I(2));

O(0) <= (not DS(0)) and (not DS(1)) and M;
O(1) <= DS(0) and (not DS(1)) and M;
O(2) <= (not DS(0)) and DS(1) and M;

EI(0) <= (not DS(0)) and (not DS(1));
EI(1) <= DS(0) and (not DS(1));
EI(2) <= (not DS(0)) and DS(1);

end Behavioral;
```

Fig 2.2: Code written in VHDL on the Vivado software for Part I.



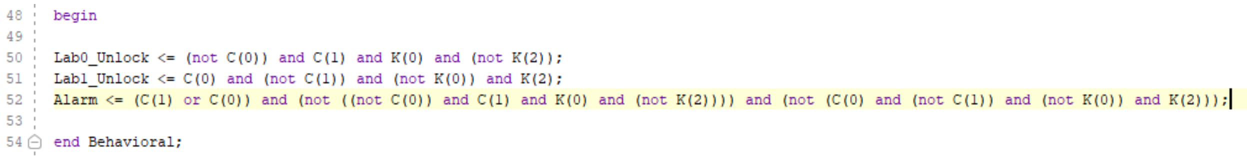
Name	Pin	Output	Type	Parameter1	Ready
DIO 1	DIO 1	PP	Clock	500 mHz	
DIO 2	DIO 2	PP	Clock	5 Hz	
DIO 3	DIO 3	PP	Clock	10 Hz	

Fig 2.3: Screenshot of a logic window taken in the Waveform software for Part I.



Name	Value
C[1:0]	1
[1]	0
[0]	1
K[2:0]	6
[2]	1
[1]	1
[0]	0
Alarm	0
Lab0_Unlock	0
Lab1_Unlock	1

Fig 2.4 : Screen-shot of a simulation window taken in the Vivado software for Part II.



```
begin

Lab0_Unlock <= (not C(0)) and C(1) and K(0) and (not K(2));
Lab1_Unlock <= C(0) and (not C(1)) and (not K(0)) and K(2);
Alarm <= (C(1) or C(0)) and (not ((not C(0)) and C(1) and K(0) and (not K(2)))) and (not (C(0) and (not C(1)) and (not K(0)) and K(2)));

end Behavioral;
```

Fig 2.5: Code written in VHDL on the Vivado software for Part II.

DISCUSSION

MUXandDEMUX

a. What happens when the mux-select and demux-select signals are in the unused state?

Ans. The output channels do not receive the input signal, meaning all outputs (O(0), O(1), O(2)) remain in an inactive state.

b. Justify your design for the demux output (provide simulation waveform, indicate 2 time moments on the waveform, and justify the output condition based on the input signals).

Ans. At 18.500 ns DS(0)=0, DS(1)=1, and M=1. Applying these values in our equations we get O(0)=0,O(1)=0,O(2)=0 which matches the values in the simulation.

Similarly, at 5.00 ns DS(0)=0, DS(1)=0, and M=1. Applying these values in our equations we get O(0)=1,O(1)=0,O(2)=0 which matches the values in the simulation.

c. Discuss the differences between the design approach in this lab using the Xilinx FPGA board, and the design approach in lab1 using the discrete ICS and breadboard. List the pros and cons of each method and describe which one you would prefer and why.

Ans. The Xilinx FPGA board in Lab 2 offers flexibility, handling complex designs and enabling quick modifications via Vivado software, but requires VHDL knowledge and has a higher cost. Lab 1's discrete ICs and breadboard setup is hands-on and affordable for simple circuits but lacks scalability, is time-consuming, and limited by physical space and component availability. I would prefer the FPGA approach because of its flexibility, scalability, and efficiency in testing and implementing changes, especially for more advanced digital logic projects.

Lab Access Control

a. Justify your design using the functional simulation (provide simulation waveform, indicate 2 time moments on the waveform, and justify the output condition based on the input signals).

Ans. Alarm = (C(1)+C(0)).(Lab0_Unlock).(Lab1_Unlock)

At 67.000 ns, C(1)=0 ,C(0)=1 ,Lab0_Unlock=0 ,Lab1_Unlock=1 which gives Alarm=0 . This matches the Alarm value in simulation.

At 45.000 ns, C(1)=1 ,C(0)=0 ,Lab0_Unlock=1 ,Lab1_Unlock=0 which gives Alarm=0 . This matches the Alarm value in simulation.

b. How would you rate the system’s effectiveness?

Ans. The system is effective for basic access control, as it uses push-button signals to simulate secure entry. However, it has limitations in complexity, relying solely on button presses instead of more secure methods like keypads or biometric inputs.

c. Can you come up with a better user access system? If so then describe your system.

Ans. A more advanced system could incorporate an RFID reader or keypad for multi-factor authentication. For example, requiring both a unique RFID tag and a personal PIN would increase security. Adding a display and implementing time-based access restrictions could further enhance the system, making it more applicable in practical situations

CONCLUSION

In conclusion, the experiment successfully demonstrated the operation of multiplexers (MUX) and demultiplexers (DEMUX) as implemented on the Zybo Z7 FPGA board. The circuit functioned as expected, with the MUX accurately selecting inputs based on the state of the select lines, as evidenced by the correct illumination of the output LEDs.

Key Findings

- The experiment confirmed the theoretical principles of digital circuit design, with reliable output from the MUX.
- The lab access control system effectively ensured that only authorized individuals could operate the equipment, maintaining a safe and secure environment.

Suggestions for Future Work

For future experiments, it would be beneficial to:

- Explore more complex MUX/DEMUX configurations.
- Investigate the effects of varying input signal frequencies.

Overall, the lab provided valuable insights into digital logic design and its practical applications.

REFERENCES

1. *ECE 210 Lab Manual 1*, University of Alberta, 2024. [Online]. Available: eClass. [Accessed: Nov. 4, 2024].

ACKNOWLEDGEMENTS

- We would like to express our sincere gratitude to the lab teaching assistants for their invaluable guidance and continuous support throughout this experiment. Their willingness to assist with even the simplest questions significantly enhanced our understanding of the concepts and procedures involved. Their dedication to helping students succeed is greatly appreciated.