	ECE210-Intro. Digital Logic Design
Fall 2024	LAB2: Combinational Logic Design

DATES

Date	Section	Time
Oct 22-2024	D21	08:00 - 10:50 AM
	D25	02:00 - 04:50 PM
Oct 23-2024	D31	08:00 - 10:50 AM
Oct 24-2024	D45	02:00 - 04:50 PM
Oct 29-2024	D22	08:00 - 10:50 AM
	D26	02:00 - 04:50 PM
Oct 30-2024	D32	08:00 - 10:50 AM
Oct 31-2024	D46	02:00 - 04:50 PM

INTRODUCTION

In this laboratory exercise, you will delve into the implementation of combinational logic circuits, leveraging the Xilinx Zybo Z7 FPGA development board and the Xilinx Vivado software suite. The lab tasks will involve designing and constructing a Multiplexer (MUX) and Demultiplexer (DEMUX) circuit, alongside a Lab Access Control circuit. The practical application of these tasks aims to enhance your understanding of digital electronics and FPGA programming.

LEARNING OBJECTIVES

Upon completion of this laboratory exercise, students will be able to:

1. Demonstrate a comprehensive understanding of the fundamental concepts and operations of combinational logic circuits.
2. Apply knowledge of Karnaugh Maps (K-maps) in the design and optimization of logic circuits.
3. Integrate the theoretical knowledge of digital logic design with practical skills in FPGA development.
4. Utilize VHDL programming to implement and validate combinational circuit logic.

PRE-LAB

You are expected to first read the lab instructions and background and then complete the pre-lab assignment and submit it to eClass before the pre-lab deadline. Refer to the Tutorial Lab (Lab 0) as well to refresh lessons learnt.

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).

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.

Department of Electrical and Computer Engineering
11-203 Donadeo Innovation Centre for Engineering, 9211-116 Street NW,
University of Alberta, Edmonton, Alberta, Canada T6G 1H9   2/17

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

BACKGROUND

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

The definition of multiplexing is combining several signals for transmission on some shared medium (e.g. a telephone wire). The signals are combined at the transmitter by a multiplexer ("mux") and split up at the receiver by a demultiplexer. The communications channel may be shared between the independent signals in one of several different ways: time division multiplexing (TDM), frequency division multiplexing (FDM), or code division multiplexing (CDM).

Generally, MUXs and DEMUXs are available as standard integrated circuit (IC's the 74150 16 to 1 mux) packages allowing for 2, 4, 8, or 16 inputs/outputs. Sometimes, however, a non-standard input number is made from discrete AND and OR gates as required.

Let's discuss a practical scenario to help understand MUXs and DEMUXs. This is the case for a small radio telescope array.

You have been hired by the NASA (UofA employees union) as a co-op student. One of your tasks is to design, build, and test a 3- input/1-output MUX and an accompanying 1- input/3-output DEMUX using AND, OR, and inverter gates. The layout of the system is shown in Figure 1. We see that there are three parabolic antennas, each with a remote data logger. Each data logger sends important digital data back to the control center, via a shared transmission medium, where the radio astronomers analyse them.

The MUX's output will be fed into the DEMUX, and each of the three data outputs from the DEMUX (indicated by O) will be directed to the offices of three different radio engineers. O(0) is the output for the first engineer whereas O(1) and O(2) are the outputs for the second and third engineers. The MUX select inputs (S) will be employed to choose which antenna will receive data and the DEMUX select inputs (DS) will determine the receiving engineer.

An RGB LED will be used as an ENGINEER INDICATOR (EI) where it being ON provides them with visual confirmation that they are receiving data **(even if the received data be "0")**. The color of the RGB color indicates which engineer is receiving the data with red, green and blue corresponding to the first, second and third engineers. **Each color of the RGB color is separately controlled with EI(0), EI(1) and EI(2) being for red, green and blue colors.** Only one of these should be in the ON state "1" for our particular application here.
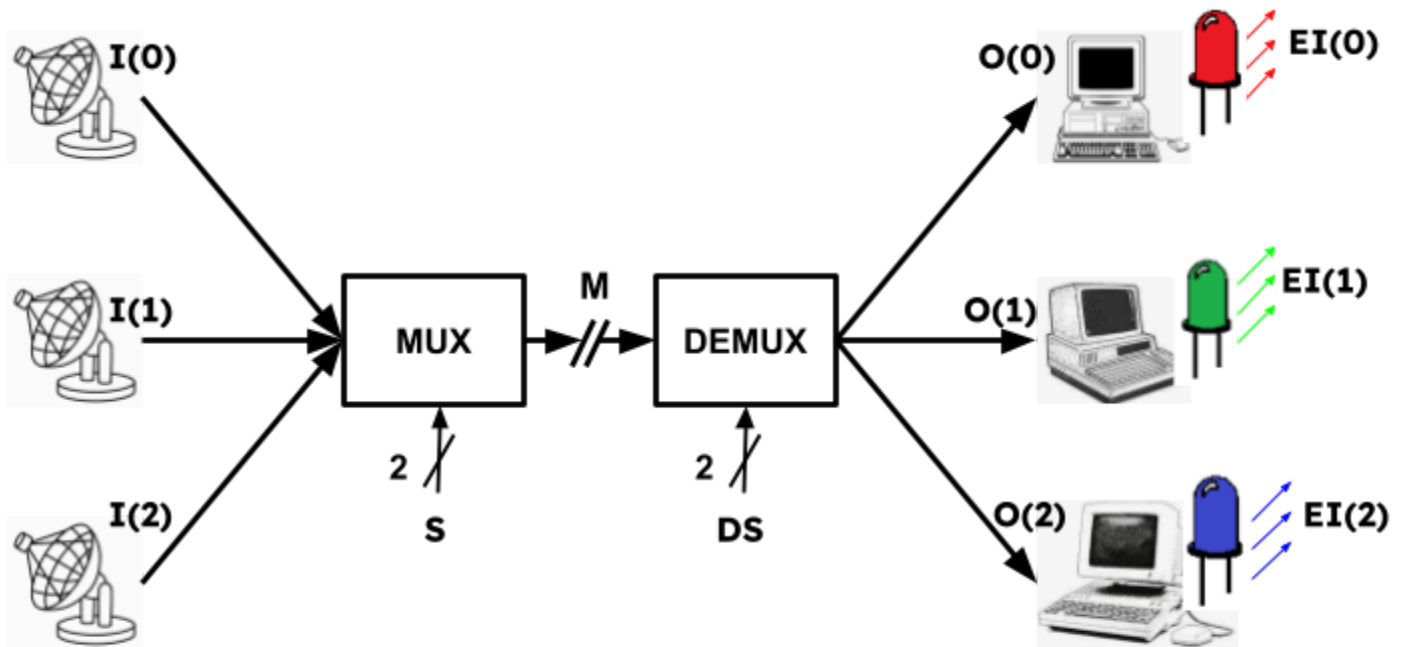
Figure 1

The MUX passes one of the three data signals to the transmission line. The desired data stream is selected using the mux-select signal S(0) and S(1) based on the following table:

S(1)	S(0)	MUX Output (M)
0	0	I(0)
0	1	I(1)
1	0	I(2)

On the receiving end, the target engineer is selected using the demux-select signals DS(0) and DS(1).
This is an example of Time-Division Multiplexing as the signals are separated on the transmission medium by time (only one signal at a time).

Part II – Lab Access Control Circuit:

The ECE department has decided to increase the access security for two of its new labs (Lab0, Lab1). There will be a card reader and a keypad to manage access to the labs. To gain entry into a particular lab, the user must have a OneCard with the correct code, and enter an authorized 3-bit code using the keypad.
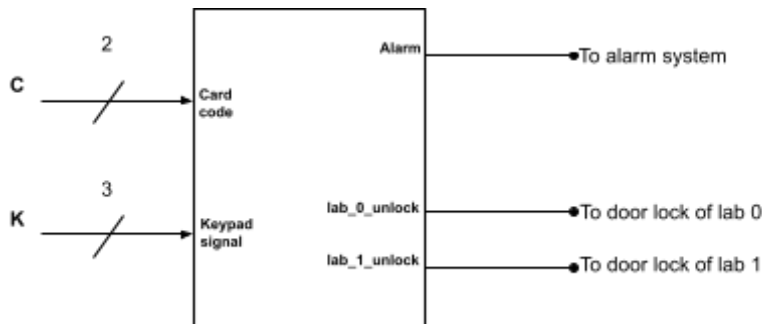
In this part of the lab, you will design and implement a circuit to control the access to the two new labs (Lab0, Lab1). The door access to each of these labs is controlled by the circuit shown below.

The control circuit accepts two input signals: Card-code (2-bits), and Keypad (3- bits). The output signals are Alarm, Lab0_unlock, and Lab1_unlock. The description of the input/ output signal is as follows:

1. **Card-code**: This is a 2-bit input signal (C0, C1) which represents the output of the card reader.

C(0)	C(1)	Card reader output
0	0	No card in reader
0	1	Valid for Lab 0
1	0	Valid for Lab 1
1	1	Invalid card - Turn ON red LED

2. **Keypad**: This is a 3-bit input signal (K0, K1, K2) which represents the access code the user entered into the keypad

Lab	Authorized codes (K0 K1 K2)
Lab 0	Either (100) or (110)
Lab 1	Either (011) or (001)

3. **Lab0_Unlock**: This is a one-bit output signal that is applied to the door lock of lab0. The door of lab 0 will be unlocked if (Lab0_Unlock = '1').
4. **Lab1_Unlock**: This is a one-bit output signal that is applied to the door lock of lab1. The door of lab 1 will be unlocked if (Lab1_Unlock = '1').
5. **Alarm**: This is a one-bit output signal that triggers the alarm system.

Function Description:

A. To gain access to a particular lab (Lab0, Lab1), the user must use a valid OneCard and enter an authorized keypad code for that lab. When the combination of card code and keypad code is correct, the appropriate unlock signal should be turned ON "1", otherwise the unlock signal will be OFF "0".

B. The Alarm signal should go high if :
   1. There is a card in the reader (i.e C_0 C_1 is not equal to "00")
   2. The combination of the card code (C_0 C_1) and keypad pressed keys (K_0 K_1 K_2) is not correct.

LAB COMPLETION REQUIREMENTS

1. **(Part I: Simulation)** Consider the system demonstrated in Figure 1. Implement a MUX with 3 inputs, I, and S as the signal select. Connect the output of the MUX to your DEMUX with DS as data select and O as its output. Simulate your design and show the results using the test-bench file provided, "**lab2_part1_tb**". In writing the test-bench file we supposed your entity name is "lab2_part1". **Test-bench imports your design as a component. Therefore, it needs your entity name. If you have used another name for the entity, open the test-bench file in Vivado and change the name in the component and port map sections.**

2. **(Part I: Hardware Implementation)** Add the constraint file for Part I from the lab 2 files (lab2_part1_constraint.xdc) on eclass to your project. This provided constraint file will map the MUX select inputs (S(0) and S(1)) to the first two switches on the Zybo board and the DEMUX select inputs (DS(0) and DS(1)) to the last two switches. Figure 2 below shows this mapping. The DEMUX outputs (the received data by engineers, O(0), O(1) and O(2)) will be mapped to the three LEDs indicated.

    The data (I(0), I(1), I(2)) must be externally provided by the Analog Discovery 2 (AD2) tool to the Zybo board through its PMOD ports (See figure 2, Pmod ports are a type of input/output interface). To do this, you have to connect the provided adapter cable to the left Zybo Pmod port as shown in Figure 3 (a). On the other side of this adapter cable, you need to use breadboard jumper wires to access 1, 2 and 3 on the cable. This is shown in Figure 3(b). These wires have to be connected to the AD2 DIOs 1 to 3, respectively. You need to connect the Zybo board ground to the AD2 ground (black wire in Figure. 3b). The connection can be conveniently implemented on a breadboard.
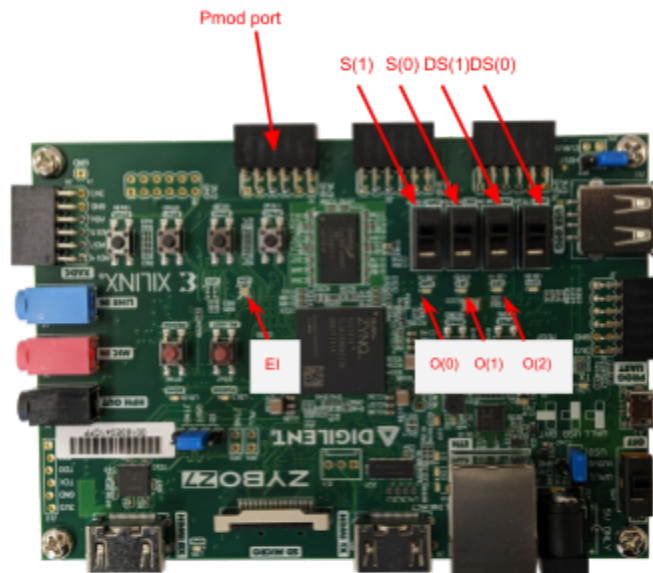

Figure 2

Department of Electrical and Computer Engineering
11-203 Donadeo Innovation Centre for Engineering, 9211-116 Street NW,
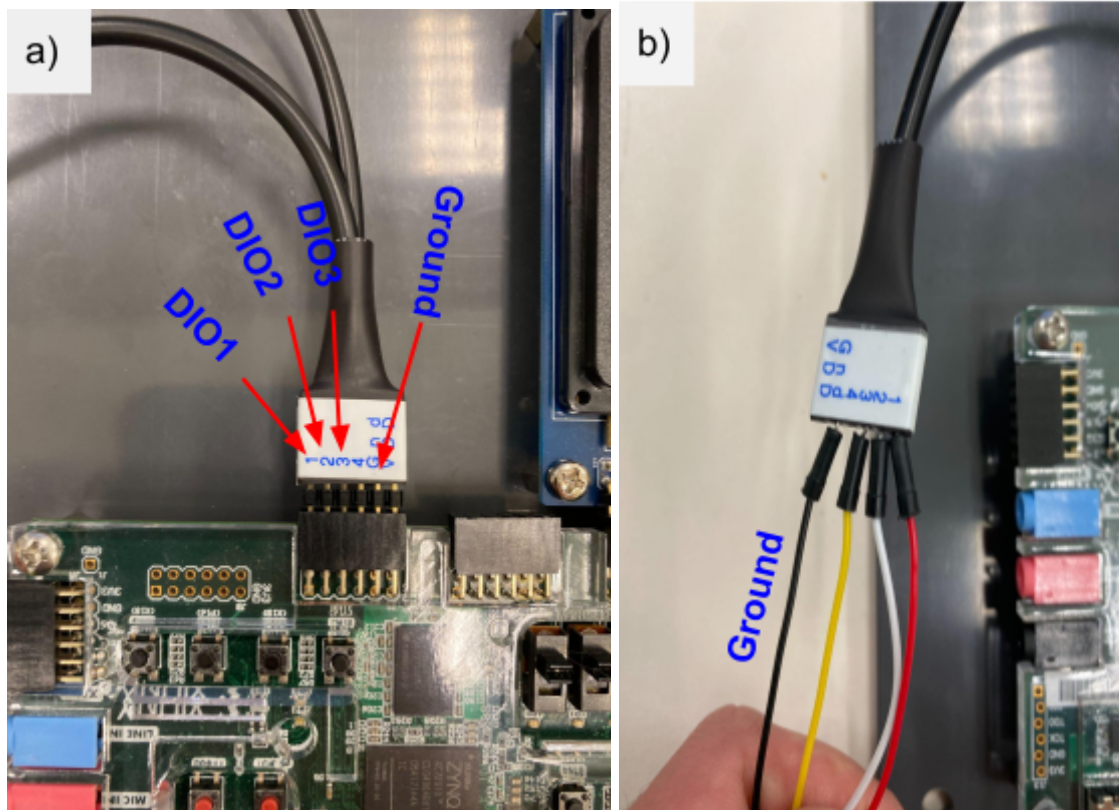University of Alberta, Edmonton, Alberta, Canada T6G 1H9

Figure 3

As you recall from lab 1, we have to use the software WaveForms to control the AD2. To set up the three-channel data, open the patterns tab and set up DIOs 1 to 3 as shown in the following picture:

By clicking on the run button, AD2 will generate digital signals (clock pulse signals) with the corresponding frequencies in the figure on its DIOS 1 to 3 which then will be fed to the Zybo board as I(0), I(1) and I(2). After programming your board, you may now choose which of these data channels to be sent to which of the radio engineers. You can clearly distinguish the messages by the blinking frequencies on the LEDs. Toggle the first two switches to multiplex between the messages. Use the other two switches to choose which radio engineer gets to see the message.

> **(Part II: Simulation)** Implement your design using VHDL behavioural approach and show simulation results using the provided "**lab2_part2_tb**" test-bench file. Change the component and port map name in the test-bench file if necessary.

3. **(Part II: Hardware Implementation)** Demo Part II on Zybo Z7 board using the Lab II constraints file on eclass. Description of the Zybo Z7 input-outputs for this design is as follows:
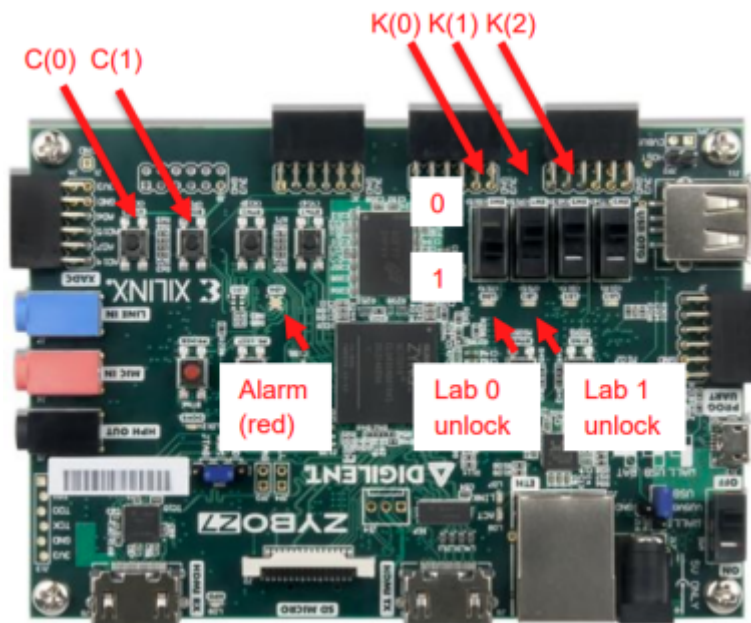
If you push the C(0) or C(1) buttons, then their values are 1 otherwise 0. We will be using LED[0-1], SW[0-2], Button [0-1] and Led[6] (red) as illustrated in the above image. Each of the corresponding signals for part 2 will be displayed using the LEDs and the input signals c[0-1] and k[0-2] are assigned to the buttons and switches respectively. To demonstrate your design, mimic the simulation waveforms from the previous section and show a TA/LI.

DOCUMENTATION

This lab exercise requires a formal report. Don't forget to include your name(s), section number, and lab number on the title page, and follow the guidelines described on the website. Handwritten reports will not be accepted.

Answer the following questions in your report.

MUX and DEMUX

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

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).
b. How would you rate the system's effectiveness?
c. Can you come up with a better user access system? If so then describe your system.

REFERENCES

- [Zybo Z7 Documentation, Tutorials and Example Projects](),
- [Zybo Z7 Reference Manual]()
- "Digital System Design with FPGA: Implementation Using Verilog and VHDL" ; Cem Ünsalan, Ph.D., Bora Tar, Ph.D; Pub. Date: 2017 McGraw-Hill Education; ISBN: 9781259837906

APPENDIX

Worksheets that can help complete lab work. C1 C0   K2 K1 K0

C0	C1	K0	K1	K2	Alarm	Lab0 Unlock	Lab1 Unlock
0	0	0	0	0			
0	0	0	0	1			
0	0	0	1	0			
0	0	0	1	1			
0	0	1	0	0			
0	0	1	0	1			
0	0	1	1	0			
0	0	1	1	1			
0	1	0	0	0			
0	1	0	0	1			
0	1	0	1	0			
0	1	0	1	1			
0	1	1	0	0			
0	1	1	0	1			
0	1	1	1	0			
0	1	1	1	1			
1	0	0	0	0			
1	0	0	0	1			
1	0	0	1	0			
1	0	0	1	1			
1	0	1	0	0			
1	0	1	0	1			
1	0	1	1	0			
1	0	1	1	1			
1	1	0	0	0			
1	1	0	0	1			
1	1	0	1	0			
1	1	0	1	1			
1	1	1	0	0			
1	1	1	0	1			
1	1	1	1	0			
1	1	1	1	1			

Lab 0 Unlock

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

KO=1		K1K2 00	01	11	10
C0C1	00				
	01				
	11				
	10				

Lab 1 Unlock

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

KO=1		K1K2 00	01	11	10
C0C1	00				
	01				
	11				
	10				

Department of Electrical and Computer Engineering
11-203 Donadeo Innovation Centre for Engineering, 9211-116 Street NW,
University of Alberta, Edmonton, Alberta, Canada T6G 1H9

Alarm:

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

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

MARKING GUIDELINES

Student Names:_____

_____

Pre-Lab

(10) Derivation of MUX and DEMUX Output Equations

(07) Derivation of Lab0_Unlock, Lab1_Unlock and Alarm Equations

(10) Schematic for part II (hand drawn or using any tool of your choice)

(06) Completed Truth Tables for both parts

(05) Handwritten or electronic version of VHDL solution for part I.

Sub-total_____/38

In-Lab

(10) Mux Functionality

(10) Demux Functionality

(05) Engineer Indicators

(05) Lab0_Unlock Signal

(05) Lab1_Unlock Signal

(05) Alarm Signal

TA Signatures:_____ Sub-total_____/40

Department of Electrical and Computer Engineering

11-203 Donadeo Innovation Centre for Engineering, 9211-116 Street NW,

University of Alberta, Edmonton, Alberta, Canada T6G 1H9

Lab Report

(02) Title page, Abstract and Introduction

(10) Design section and procedure

      Part I:

            i. Design (MUX and DEMUX logic expressions) (2)

            ii. MUX/DEMUX VHDL code (2)

      Part II:

            iv Design (Logic Expression) (1)

            ii. Schematic (1)

            iii. Lab Door Lock/Unlock VHDL code (4)

(10) Results, Discussion and Conclusion

      i. Snapshot of the simulation result for PART I (1.5)

      ii. Snapshot of the simulation result for PART II (1.5)

      iii. Answers to questions in the documentation part of the manual (1 for each) (6)

      iv. Conclusion (1)

Sub-total_____/22

Lab Report Total_____/100