LAB 2 – Combinational Logic Design with Basic Sequential Elements

Learning Objectives

- Use top-down design approach with sub-systems
- Implement basic combinational blocks and registers in VHDL
- Use the seven-segment display and multiplexing

This lab contains two parts:

1. Part 1: A semi-guided tutorial to be familiar with some of the building blocks required for the task. The Blackboard tutorial provide example VHDL descriptions for an 8-bit register, hex-to-seven segment decoder, multiplexer, and a counter, which can be used as sub-systems (with appropriate modifications) when implementing the task given in part 2. You are strongly advised to implement and test the required sub-systems first by following the tutorial. The tutorial also introduces parameterised VHDL using generics, which can be a useful coding style.
2. Part 2: An individual lab task which is assessed.

Part 1: A semi-guided tutorial for some sub-systems for the task in part 2

Go through the tutorial activities from 2.1-2.5 on Blackboard under week 3 >> prac 2 folder. Teaching staff will provide explanations as needed. It is recommended that you draw a block diagram for the lab task in part 2 and identify which sub-systems you will need to implement it. The purpose of this part 1 tutorial is to identify and individually test the sub-systems required so that you can integrate them in part 2, when implementing the required lab task.

Part 2: Lab Task – 4-BCD digit combinational lock (this task is assessed)

You are required to design a simple password detector where the user inputs a ***4-digit BCD number*** which is compared against the ***last four digits of your 8-digit student number*** to indicate a match.

Four BCD digits are entered to the system, one at a time, using four slide switches on the board to set the BCD digit and a push button(s) to trigger the input processing. That is, 4 button presses are required to enter all four BCD digits (i.e., 16 bits) to the system. The 4-digit BCD input should be displayed on four seven segment displays all the time. If the 4-digit BCD input matches the last four digits of your 8-digit student number, then the lock opens (the output *lock* is low and *unlock* is high). If there is no match, then *lock* is high and *unlock* is low. The two outputs *lock* and *unlock* are to be displayed on two LEDs on the board. The system is not responsive to changes of slide switches unless the values are entered by pressing the input processing push button. When the reset button is pressed, the display is cleared to 'EEEE' and the system is locked once again (i.e., *lock* is high and *unlock* is low). You can use either a ***synchronous*** design approach where all the sequential elements are driven by the 100 MHz system clock or an ***asynchronous*** design approach where sequential elements can be driven by different clocks such as the 100 MHz system clock and push buttons. The difference between these two approaches is the choice for the clock signal for the registers used to hold the 16-bit input, as mentioned below.

- Synchronous approach – Any registers used to hold the user input and any other sequential blocks used for seven segment multiplexing should be driven by the same 100 MHz system clock. Here, four 4-bit registers can be used to hold the 4-digit BCD input, and four distinct push buttons can be used as distinct clock enable signals for these registers, which are driven by the system clock.
- Asynchronous approach - Any registers used to hold the user input and any other sequential blocks used for seven segment multiplexing can be driven by different clocks. Here, a four stage 4-bit shift register can be used to capture the 4-digit BCD inputs and a single push button can be used as the clock signal for this shift register.

When implementing the seven-segment multiplexing, you may also need to use a clock divider (a counter can be used to achieve this) to slow down the clock signal as needed to avoid any flicker on the display.
You can make assumptions about what happens after the lock is open – as to whether it goes back to the *lock* state after certain amount of clock edges, or the system remains unlocked until the next incorrect code is inputted via switches and button. Mention such assumptions in your report.

Following steps are suggested:

- Read the description and draw a top-level block diagram to identify the inputs and outputs.
- Use a top-down design approach to complete this block diagram by identifying all the sub-systems required and their details. You must have a detailed block diagram before you start attempting any VHDL coding. You can show this to teaching staff and seek feedback.
- It is recommended that you use dataflow/behavioural abstraction for the sub-systems and structural abstraction at the top-level to integrate the sub-systems. There is no need to use gate level descriptions for any of the sub-systems or for the main system.
- It is also recommended that you test the sub-systems individually, which is partially achieved by following the part 1 tutorial. When testing the sub-systems, you do not need to synthesise them individually. Just a functional simulation is sufficient to ensure the sub-system is working as intended. However, you do not need to show the individual testing of the sub-systems in your report. This step is mainly suggested to minimise errors and facilitate troubleshooting.
- Integrate the sub-systems using a structural approach at the top level, following your block diagram.
- Complete functional simulation for the overall system with a suitable testbench to generate the input signals. Make sure the simulation covers at least one successful unlock, one un-successful unlock and reset functionality. You may use a self-checking testbench, although this is not essential. Also, the seven-segment display multiplexing is not essential for the simulation part, however you will need that for implementation on board.
- Complete synthesis, implementation, and on-board testing.

Submission and Assessment

You will be assessed based on a report submission followed by oral assessment. Submit a short report (electronically typeset) in PDF format via Blackboard containing the following:

- Short introduction to the task [1 mark]
- Block diagram of the design with properly labelled inputs, outputs, and sub-systems. Gate level descriptions are not needed. Details of the sub-systems should be provided up to the level of VHDL abstraction used in your design. That is, if a sub-system is further divided into sub-sub-systems in your VHDL description, then this must be shown on the block diagram. The purpose of the block diagram is to plan the design before you described it in VHDL. [2 marks]
- VHDL descriptions for the main design blocks (must include the lock/unlock part) - to be included in appendix. [4 marks]
- Functional simulation showing one example of unlock, lock, and reset functionality including seven segment decoder outputs. [5 marks]
- RTL and Synthesis schematics with some interpretations. [2 marks]
- Synthesis and implementation output – i.e., the FPGA resource consumption [1 mark]
- Evidence of on-board testing including lock, unlock, display and reset functionality (subject to demonstration during the oral assessment) [4 marks]
- Conclusion identifying any issues, potential improvements, and your reflection [1 mark]
- References if any

The report should not be just a collection of screenshots, but with proper explanation of figures/results with some discussion. The report is **due on Monday 19/08/2024 4:00 pm AEST**.

Oral assessment: Conducted during scheduled lab sessions in week 5. You need to explain your design to teaching staff and ***demonstrate on-board testing of your design*** followed by some Q&A to check the understanding of your work. Marks for the oral assessment will be distributed as follows:

0: No knowledge of the design **(total mark capped at 0)**
1: Very little knowledge of the design **(total mark capped at 50%)**
2: Reasonable knowledge of the design
3: Good knowledge of the design
4: Excellent knowledge of the design

Total mark for this lab is out of 24, subject to oral assessment caps as above and any late penalties as outlined in the ECP. If you do not attend/complete the oral assessment, your marks for this lab will be capped at 0, as indicated above.