ECE410-Advanced Digital Logic Design

LAB 1: Priority Encoder Design Using VHDL

INTRODUCTION

In this lab exercise you will use the divide-and-conquer strategy to design priority encoders of increasing size, as measured by the number of inputs. You will start by designing a small 3:2 priority encoder from simplified Boolean expressions that are derived from the priority encoder's truth table specification. Then you will use the gained knowledge to design two 7:3 priority encoders (first using boolean expressions and then using instances of the smaller 3:2 priority encoder).

LEARNING OBJECTIVES

- To learn how to design priority encoders using a truth table specification and logic simplification from a Karnaugh map. The resulting simplified designs are to be expressed in VHDL.
- To learn how to synthesize, implement, and program a 3:2 priority encoder design that will operate in the FPGA fabric of the Zybo Z7 board.
- To learn how to design a 7:3 priority encoder using a reduced truth table specification and logic simplification. Two VHDL designs are to be created: (1) one will use the built-in VHDL logic functions, and (2) the second will use structural-level VHDL that employs instances of the 3:2 priority encoder.
- To implement the second 7:3 priority encoder design on the Zybo Z7 board.

DATES

section	date
D11	Sept-19-2022 14:00 to 16:50
D31	Sept-21-2022 14:00 to 16:50
D51	Sept-23-2022 14:00 to 16:50

Pre-Lab Marks-5%

- Carefully read this document and study the instructions before arriving at your lab session.
- Review the material on priority encoders in the lecture notes**.** Note that the priority encoders to be implemented in this lab exercise use positive logic for both the inputs and outputs.
- Complete the VHDL design of the 3:2 priority encoder before coming to the lab. **This design will be checked as part of the pre-lab work.** It would also be a good idea to start working on the two 7:3 priority encoder designs before coming to the lab, but this work will not be checked at the lab.
- Consult the lecture slides, and perhaps also one of the recommended VHDL textbooks, to understand the difference between the structural-level and behavioral-level modelling styles in VHDL.

- Note that 5% of the Lab 1 mark will be allocated by the lab teaching staff for the pre-lab work. Skipping the pre-lab work may prevent you from finishing the exercises within the 3-hour lab time.

LAB DEMONSTRATION REQUIREMENTS

Students must demonstrate the successful functioning of the priority encoder designs for Parts 1 and 2 to a Teaching Assistant (TA) or Lab Instructor (LI). These demonstrations can be done at the start of the first session of Lab 2, but you should aim to complete both of your Lab 1 demonstrations in your scheduled 3-hour Lab 1 session.

PART 1: 3:2 Priority Encoder Marks 20%

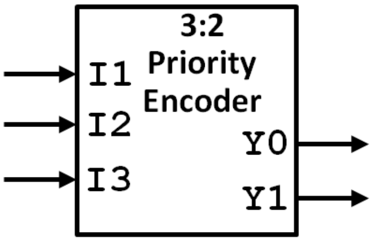

Figure 1: 3:2 Priority Encoder

The block diagram shown in Figure 1 specifies the external interfaces of a 3:2 priority encoder. The encoder inputs are to be labelled I1 (lowest priority), I2 and I3 (highest priority), and the outputs are to be labelled Y1 (the more significant bit) and Y0. Following the positive logic convention, when a '0' is applied to all three inputs, outputs Y1 and Y0 should both produce a '0', etc. As a first step you are to construct the 8-row truth table that fully specifies the behaviour of the 3–input priority encoder. Then you are to do a Karnaugh map simplification to obtain suitable Boolean functions for the two encoder outputs. You are then to implement this design using the VHDL hardware description language for this lab exercise. The implementation should use the built-in **and**, **or** and **not** VHDL functions. **This design, including the intermediate work, will be checked as your pre-lab work.**

- Create a project with project name lab1_part1 and design file name priority_encoder. Refer to the Lab 0 tutorial to review how to create a project in Vivado. Use the Zybo-z7-master.xdc constraint file from eClass (available in the Lab 1 resources).
- Download the resources for Part 1 from the Lab 1 section in eClass. Insert your VHDL code into the commented section of the skeleton VHDL file.
- A VHDL testbench is not provided in the project. You must verify the design yourself by writing your own testbench for it that includes a reasonable number of test cases to verify the correct functionality of the circuit. In your lab report you are to carefully justify your choice of test cases.
- Use Vivado to generate the bitstream and then program the FPGA.

- **Demonstrate your working design to a TA or LI.**

Your lab report section for Part 1 is to include a gate-level diagram of your design, which shows the labelled input and output signals as well as the gate-level implementations of the two output functions. Your report is to also include the 8-row truth table specifications for the two output functions, the Karnaugh map(s), the two simplified Boolean functions, the commented VHDL file for your design, and the commented VHDL testbench that you used to verify the design in simulation. As mentioned above, you should provide a brief justification for your testbench design.

PART 2-1: 7:3 Priority Encoder Design, Verification and Implementation Marks-20%

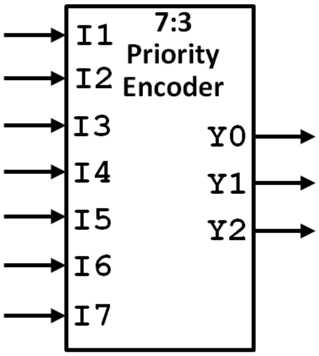

Figure 2: 7:3 Priority Encoder

In this exercise you are to build two different designs of the 7:3 priority encoder. Figure 2 shows the common external interface for both designs.

The first design is to use only the built-in VHDL logic functions, like you used for the 3:2 priority encoder in Part 1. The truth table for a 7-input function normally contains 128 rows; however, the truth table for the priority encoder can in fact be reduced in size to only 8 rows by taking advantage of *don't cares (x on the truth table)*, as shown in lecture slide 1-42. (Unlike in the example in the course notes, your truth table for this lab is to use 0s, 1s and *don't care* values, and is to assume the positive logic signal encoding.) By studying the pattern of *don't care* entries in the truth table you will be able to produce simplified Boolean functions for the three encoder outputs, Y2 (the most significant bit), Y1 and Y0 (the least significant bit). As noted above, the simplified functions are to use built-in VHDL logic functions, like **and**, **or** and **not**.

Use your simplified Boolean functions to encode your first implementation of the 7:3 priority encoder in a suitably commented VHDL model, which you are to include in your lab report. **You will need to include an image of your simulation waveform that shows that your 7:3 priority encoder is working properly.**

PART 2-2: 7:3 Priority Encoder Design, Verification and Implementation Marks-35%

The second design of the 7:3 priority encoder, which you will synthesize and implement on the Zybo Z7 board, is to use two instances of the 3:2 priority encoder design from Part 1 along with some simple glue logic. To gain insight into how this can be done, first go back to the 3:2 priority encoder and construct for it a reduced 4-row truth table that takes advantage of don't care values. Note how, if you only look at outputs Y1 and Y0, the four rows of the reduced truth table for the 3:2 priority encoder appear in both the first four rows and the last four rows of the reduced truth table for the 7:3 priority encoder. This leads to the idea of connecting inputs I0, I1 and I2 to one 3:2 encoder, while inputs I5, I6 and I7 could be connected to a second 3:2 encoder. The challenge now is to find a way of connecting the two 3:2 priority encoders, using a bit of logic, so that they operate together to form a 7:3 priority encoder. **Hint:** You may find it useful to construct a simplified Boolean expression for the decoder output Y2.

Use your simplified Boolean functions to encode your second implementation of the 7:3 priority encoder in a suitably commented VHDL model. As in Part 1, design a VHDL testbench so that you can verify that the second 7:3 priority encoder design works correctly in simulation. **Your VHDL model, the testbench, and an image of your simulated waveform signals are to be included in your lab report.**

Synthesize your verified design and then download the resulting bitstream to the Zybo Z7 board using the steps shown below. Verify that your design works correctly in the hardware implementation.

- Create a project with project name lab1_part2 and design file name priority_encoder_7_3. Refer to the Lab 0 tutorial for the instructions for creating a project. Use the Zybo-z7-master.xdc constraint file from eclass (available among the Lab1 resources posted to eClass).
- Download the resource files for Part 2 from e-class. Add the required VHDL code in the commented section.
- A testbench is not provided in the project. You must verify the design yourself by writing your own testbench for it and include a reasonable number of test cases to verify the correct working of the circuit.
- The constraint file provided on the e-class is the Zybo-Z7-Master.xdc. You must modify it on your own to verify the design on the hardware.
- Use Vivado to generate the bitstream and program the FPGA. Since the number of inputs is bigger than the available number of switches on the board, map the push buttons as inputs as well. ***The output of the 7x3 priority encoder can be shown on any of the three white leds. The .xdc constraint file is fully commented on the eclass. Students will have to uncomment the required lines to map the inputs and outputs of the encoder.***
- **Demonstrate your working design to a TA or LI.**

Lab Report: Marks for lab report quality-20%

Submit all simulation test benches, waveforms and synthesis results in your lab report. Suggested reports layout(s) are provided on e-Class site to clarify the report submission requirements. Upload your report as one .pdf to use the link in the section for Lab 1 in eClass by the deadline for your section.

The lab reports for students in sections LAB D11, D31 and D51 are to be uploaded to eClass by 6 pm (18:00) on Oct 3, 5 and 7, respectively.

References:

- Zybo Z7 Documentation, Tutorials and Example Projects,
- Zybo Z7 Reference Manual
- ECE410 VHDL Reference from the Lectures