Lab 07

Part D – Two-Level 16:4 Priority Encoder

Description

In this lab module, you will work with a 16:4 priority encoder implemented using a two-level hardware structure.

As usual, for this lab module, you will require a PC with a USB-A port to program the FPGA development board.

Procedure

Project Creation

1. Download `lab07d.vhd`, `priority_encoder_4.vhd`, and `lab07d.xdc` from Canvas and place them in a new folder titled `lab07d`.
2. Open Vivado and create a new project titled appropriately.
   - For your convenience, the board identifier is `xc7a35tcpg236-1`.
     - You can copy and paste this directly into the search bar in the device select menu.
     - You can also find the board with the same options as before: `General Purpose`, `Artix-7`, `cpg236`, `-1`.

Hardware Review and Upload

3. Open `lab07d.vhd` in the editor, and follow along with the instructor to review the hardware description.
4. `Open Elaborated Design`, review the schematic alongside the instructor, and take screenshots for your report.
5. Run synthesis and implementation and then generate the bitstream to program the device. Then, open the Hardware Manager.
   - Take a screenshot of the resource utilization (LUTs and FFs) to add to your report.
6. Before proceeding, plug the FPGA development board into your computer using the provided Micro-USB to USB-A cable.

7. At the top, in the green banner (or under the `Open Hardware Manager` dropdown) click `Open Target`, then `Auto Connect`.
8. Once the device is connected, select `Program Device` (in either of the locations where `Open Target` was previously).
   - See 06B-handout for troubleshooting steps, or ask the instructor or TA.
9. Click `Program` to upload the bitstream to the FPGA.

Hardware Testing and Analysis

10. Using the sixteen switches as the input to the priority encoder, test several different cases and observe the output.
    - The output (Q) is displayed on one of the seven-segment displays. The leftmost LED is the valid bit (V).
    - Take at least three pictures showing different input cases. One of these should be the all zeroes input.
    - Verify that the hardware generates the expected outputs for your chosen inputs.
11. As discussed in the lecture, as the input width for a priority encoder grows, it rapidly becomes extremely difficult to find its boolean equations and implement it at the gate level.
    - How does the two-level hardware structure help mitigate the problem of rapid growth in hardware complexity with input size?
    - What size do you think the component encoders should be for a 64:6 two-level priority encoder? What about a 32:5 2LPE?
    - Do you think there is a better way to structure the encoder than the two-level hardware structure introduced in this lab? There is no wrong answer, but be sure to explain your reasoning.

Deliverables

- Include as part of your **formal report**:
  - A screenshot of the Elaborated Design schematic (Step 4)
  - A screenshot of the FPGA resource utilization (LUTs, FFs) (Step 5)
  - 3 pictures of the FPGA development board (Step 10)
  - Answers to handout questions (Step 11)

Outcomes

- Practice working with VHDL.

- Practice using Vivado for hardware synthesis and implementation.
- Practice programming and testing a hardware description on an FPGA development board.
- Understand what a priority encoder does and how it functions.
- Understand how a 16:4 two-level priority encoder is constructed.
- Understand VHDL components, instantiation, and structural connections.