Lab 05

Part B – Challenge

Description

In this lab module, you will design and describe a piece of hardware that can detect when at least 2 bits of a 4 bit input are high.

Procedure

1. Listen to the description of the problem and derivation of the truth table.

Warning: If you are using a laptop that does not have a USB-A port, ensure that you have a USB-C to USB-A adapter at this time. Otherwise, have a partner proceed from here, or use a lab PC.

Project Creation

2. Download `lab05b.vhd` and `lab05.xdc` from Canvas and place it in a new folder titled `Lab05b`.
3. Open Vivado and create a new project.
   - From the Quick Start menu, select `Create Project`.
   - Click `Next`.
   - On the second page, if desired, give the project an appropriate and change the location. Click `Next`.
   - Ensure `RTL Project` is selected. Click `Next`.
   - On the `Add Sources` page, click `Add Files` and add `lab05b.vhd` from where you downloaded it.
   - Ensure `Copy sources into project` is checked. Click `Next`.
   - On the `Add Constraints` page, click `Add Constraints`
   - Select the following:
     - Category: `General Purpose`
     - Family: `Artix-7`
     - Package: `cpg236`
     - Speed: `-1`
   - Finally, select the middle option: `xc7a35tcpg236-1`.
   - Click `Next`.
   - Click `Finish`.

Hardware Development

4. Open the file in the editor, and follow along with the instructor to review the hardware description.
   - To do this: inside the `Project Manager` pane, under the `Sources` subpane, and under the `Design Sources` folder, double-click the `lab05b.vhd` file.
5. Based on the truth table derived earlier, describe the hardware in VHDL using the provided stub program.
   - If stuck, consult the instructor.
6. After you have completed the hardware description, run synthesis and implementation and then generate the bitstream to program the device.
   - From the left side pane, under the `PROGRAM AND DEBUG` dropdown menu, click `Generate Bitstream`.
     - If a popup appears, click `Yes`.
   - This may take a while, since it is running three steps of the development workflow in series.
   - After the operation completes, you may see a popup with the following options: `Open Implemented Design`, `View Reports`, `Open Hardware Manager` and `Generate Memory Configuration File`.
     - Here, select `Open Hardware Manager` and click `OK`.
       - If you clicked `Cancel`, navigate to that menu manually under the `PROGRAM AND DEBUG` dropdown by clicking on the `Open Hardware Manager` text.
7. Before proceeding, plug the FPGA development board into your computer using the provided Micro-USB to USB-A cable.
   - The Micro-USB end should be plugged into the FPGA at the top left.
   - The USB-A end should be plugged into your computer.
8. At the top, in the green banner (or under the `Open Hardware Manager` dropdown) click `Open Target`, then `Auto Connect`.
   - If you are prompted to grant administrator privileges to a program with a name like `hw_server`, do so.
   - This will connect the FPGA to your computer.

9. Once the device is connected, select `Program Device` (in either of the locations where `Open Target` was previously).
    - Ensure that there is a bitstream file selected (there should be a file path in this field). If there is not, you most likely selected the wrong target device (Step 3).
        - If you have selected the correct device, and the bitstream file has not appeared, you may be able to find it by doing the following:
            1. Click the 3 dots to the right of the field.
            2. Near the top left, click the AMD logo (5th from the left) with the tooltip `Jump to Recent Project Directory`.
            3. Navigate to `./lab05b.runs`, then `impl_1`.
            4. You should see `lab05b.bit` – select this, and hit `OK`.
    - A debug probes file is not necessary.
10. Click `Program` to upload the bitstream to the FPGA.
    - Several lights on the FPGA development board may blink during this process.
        - You will know when the program is complete when the `DONE` light has flashed off and on again, and the 7-segment display is no longer looping.

Testing

11. Use the four rightmost switches as the four bit input, and observe the rightmost LED to see the output.
12. Verify each row of the truth table using the FPGA development board.

Deliverables

- As part of your informal report:
    - Completed VHDL description/code
    - Picture(s) of the programmed FPGA development board functioning as a two-high-bit detector

Outcomes

- Practice designing hardware to solve a specific problem.
- Practice working with VHDL.
- Practice using Vivado for hardware synthesis and implementation.
- Understand what a constraint file does.
- Understand how to program an FPGA with a hardware description.
- Understand how to test a hardware description on an FPGA development board.