Lab 06

Part B – Hexadecimal to 7-Segment

Description

In this lab module, you will implement a simple hexadecimal to 7-segment converter using 4 switches and one digit of the four-digit 7-segment display module on the FPGA development board.

Note: For this lab module, you will need to connect a USB-A to Micro-USB cable from your computer to the FPGA development board. 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.

Procedure

Project Creation

1. Download `lab06b.vhd` and `lab06b.xdc` from Canvas and place them in a new folder titled `lab06b`.

2. Open Vivado and create a new project titled appropriately.
   - From the Quick Start menu, select `Create Project`, then click `Next`.
   - On the second page, if desired, give the project an appropriate name and change the location. Click `Next`.
   - Ensure `RTL Project` is selected. Click `Next`.
   - On the `Add Sources` page, click `Add Files` and add `lab06b.vhd` from where you downloaded it.
   - Ensure `Copy sources into project` is checked. Click `Next`.
   - On the `Add Constraints` page, click `Add Constraints` and add `lab06b.xdc` from where you downloaded it.
     - Note that unlike Lab 05, there are different constraint files for each module of this lab.
   - Select the following:
     - Category: `General Purpose`
     - Family: `Artix-7`
     - Package: `cpg236`
     - Speed: `-1`
   - Finally, select the middle option: `xc7a35tcpg236-1`.
   - Click `Next`, then `Finish`.

Hardware Review and Upload

3. 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 `lab06b.vhd` file.
4. Under the `RTL ANALYSIS` dropdown menu, click `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.
    - 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.
    - 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 of the Vivado window, 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.
8. 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 `./lab06b.runs`, then `impl_1`.
            4. You should see `lab06b.bit` – select this, and hit `OK`.
9. Click `Program` to upload the bitstream to the FPGA.

Hardware Testing and Analysis

10. The leftmost four switches (`switches(15 downto 12)`) are your inputs for the 7-segment display.
    - Which switch provides the most significant bit of the input?
    - Choose a set of inputs and take a picture of the board.
11. The 7-segment representations for the decimal digits 0-9 are fairly straightforward.
    - Why do you think the 7-segment representations for hexadecimal digits A-F are the way that they are?
    - Could we implement any of them differently while still ensuring each digit is distinct from the others?

Deliverables

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

Outcomes

- Practice describing truth tables of arbitrary hardware.
- 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 how a seven-segment display functions.