Lab 02

VHDL

Description

This lab will introduce you to VHDL, the hardware descriptive language we will be using to describe hardware. You will use Vivado to simulate, synthesize, and implement this hardware for the Digilent Basys 3 FPGA development board.

Procedure

Part A

- Participate in the group activity and discussion.
  - You are encouraged to take notes. This will be helpful when writing the formal lab report.
  - Scratch paper may also be helpful.

Part B

- When instructed, complete worksheet 02B. Participate in group discussion when applicable.

Part C

Project Creation

1. Download the VHDL file `lab02.vhd` from Canvas.
   - If you are using Windows, it may claim that `lab02.vhd` is a "virtual hard disk" file. This is not the case. It is a VHDL hardware description. You may safely open it with a standard text editor, but do not do this now.
2. Create a folder titled `Lab02` or similar at your desired location (e.g., desktop), the place `lab02.vhd` inside.

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 `lab02.vhd` from where you downloaded it.
    - Ensure `Copy sources into project` is checked. Click `Next`.
    - Click next – constraints are not needed for this lab.
    - Select the following:
        - Category: `General Purpose`
        - Family: `Artix-7`
        - Package: `cpg236`
        - Speed: `-1`
    - Finally, select the middle option: `xc7a35tcpg236-1`.
    - Click `Next`.
    - Click `Finish`.

Code Review

4. Once the project has been created, open the VHDL file by double-clicking its name inside Vivado's `Project Manager` pane, under the `Sources` subpane, and under the `Design Sources` folder.
    - Review the hardware description. How does this hardware work?
    - Relate the hardware description to activity 02A (the black box activity) from earlier.

Simulation

5. Run a behavioral simulation with various input values.
    - From the left side pane, click `Run Simulation`, then `Run Behavioral Simulation`.
    - **Setup**
        - Minimize the two panes on the left with `Scope` and `Objects` tabs to make the wave view bigger.
        - Optionally, minimize the bottom pane (`Tcl Console/Messages/Log`).
        - Optionally, drag the handle next to `Value` to make the columns larger.
        - Select each of the tracked values under the `Name` column, except `N`.
            - This can be done by holding `Shift` and clicking the first and last entries, or holding `Ctrl` and clicking each entry.
                - Holding `Ctrl` and clicking an already selected entry will deselect it.
        - Right-click one of the selected entries
            - Click `Radix`, and select `Binary`.
    - **Running the Simulation**
        - Right-click `input1` and `Force constant`.
            - Change the value radix to `Binary`.
            - Enter a `Force value` of `01000` ($+8$).
            - Click `OK`.
        - Do the same for `input2`, except with a value of `00111` ($+7$).
        - At the top, run for 10 ns by clicking the play button with `(T)` under it. Alternatively, press `Shift+F2`.
            - This is directly left of the text box and dropdown with `10` and `ns`, respectively.
        - At this point, you should see the values you entered along with the intermediate values and the outputs.
    - Now, repeat the steps under "Running the Simulation" several more times, using different values for `input1` and `input2`.
        - While holding `Ctrl`, you can use the mouse scroll wheel to zoom in and out in the waveform view.
        - While holding `Shift`, you can use the mouse scroll wheel to navigate horizontally along the waveform view.
        - If you accidentally close the waveform view, go to `Window` in the toolbar, and click `Waveform`.

- Take a screenshot of the resultant waveform. You do not need to include the startup period where the inputs were all red `X`s or orange `U`s.
  - Scroll to the right using `Shift+Mouse Scroll Down`.
- Close the simulation.
  - If prompted, save the waveform configuration and add to project. This is not necessary, but is helpful if you close the simulation and need to simulate again.

Synthesis

6. Run synthesis for the hardware.
   - From the left side pane, click `Run Synthesis`.
   - Observe the bottom right pane. It may take about a minute for synthesis to complete. Note the category `LUT` and `FF`.

Hardware Analysis and Discussion

7. Discuss the following questions:
   - Based on your simulation what can you say about the piece of hardware that you have just simulated?
   - Will this hardware work for positive and negative numbers?
   - What is the range of numbers that you can deal with using this hardware description?
   - Can you think about a different way to implement the same hardware?
8. Change the hardware description to deal with a larger range of numbers and repeat your simulation to check that it works correctly.

Deliverables

Lab Report

- Submit a **formal report** (see Formal Lab Report Guidelines document).
  - Either in your procedure or as an appendix, attach the code used in the lab (`lab02.vhd`), and **add line comments** (using `--`) to explain what the major sections are doing.
  - In an appendix, attach the completed **worksheet 02B**.

Outcomes

- Practice identifying hardware based on the inputs and outputs.
- Practice working with signed numbers in two's complement.
- Understand overflow and sign extension.
- Understand what VHDL is and why it is useful.
- Understand how to use Vivado for simulation and synthesis.
- Summarize and analyze the work done in the lab in a formal report.