Lab 3 – Introduction to Xilinx Design Tool and VHDL Review

Overview

In this lab, you will be introduced to Xilinx Design tools. The lab computers have Xilinx ISE 14.7 installed, however the latest design suite from Xilinx is called Vivado. Both software operate in similar fashion with some GUI differences. You can download "WebPack" version of Xilinx's Vivado 2019.2 CAD tools in your personal computer. The Webpack version of the tools are free and can be downloaded from Xilinx [Vivado 2020.2]. The tools run on both Windows and Linux but not natively on a Mac. If you have a Mac, you will have to run the tools in a virtual machine. Oracle's Virtualbox VM is a free VM that works quite well and is the recommended VM.

Creating a project

1. Open “ISE Design Suite” from your desktop or start menu.
2. Click “New project…” from the left-hand menu. Enter a project name and specify a location of your choice. Click **Next**.
3. Next window is project settings. As we will be using the simulator only, there is no need to choose any specific product (keep defaults). Select VHDL from “Preferred Language”. Click **Next**.
4. Next window shows the project summary. Review and click **Finish**.

Adding Files to the Project

1. Download *or_word.vhd* and *or_tb.vhd* from your class website.
2. Select **Project => Add Source** from the file menu.
3. Select the two VHDL files you just downloaded and click **Open**.
4. The "Adding Source Files..." window shows the files you are adding, their association, and which library they are added to (the default is the "work" library). Click **OK**.

Run the Simulator

1. In the "hierarchy" window, select the testbench.
2. Select *Run 'Behavior Simulation'* from the menu on the left.
3. Click the "Run for Specified Time" button to extend the simulation. You can zoom in and out using the zoom buttons on the left of the simulation window Note that when the

simulation first opens the time step may be set to "picoseconds". You will need to change this to "microseconds" to actually see the results of the simulation.

In the future, you will want to organize the signals in the wave window. Try these with the current simulation

1. The 'tick' signal is just a constant and not useful to view. Right-click it and select "Delete" to remove it.
2. You can create dividers to help organize waveforms. Right-click on wave window and select "New Divider." Give it the name "Output". Drag this divider between the b and q signals.
3. You can use grouping as another way to organize waveforms. Control-click to select both the "a" and "b" signals. Right-click and select "Add Group." Give it the name "Inputs."
4. To save your changes, click the save icon, and save a file name *my_waves.wcfg* in the *lab1* directory. You can reload this config file for subsequent simulations.

Now, take your time and try to go back and understand all the details about this project. o*r_word*.vhd is a VHDL code for a simple 4-bit OR gate, and *or_tb.vhd* is a testbench for this design.

Assignment

The next step would be designing and testing a **4-bit wide AND gate**.

1. Create a new project or add new source to existing project by clicking Project -> New source, then choose "VHDL Module" and write a code for a 4-bit AND gate.

2. Click Project -> New Source, then choose "VHDL Test bench" and choose the 4-bit AND gate as the associate source. Then complete the testbench and run the simulation. You can use the testbench for OR Gate for help.

The next one would be designing and testing a **1-bit wide 2-to-1 multiplexer**. A testbench file is given in beachboard 1bit_2to1mux_tb.vhd .

1. Create a new project or add new source to existing project by clicking Project -> New source, then choose "VHDL Module" and write a code for a 1-bit wide 2to1 multiplexer.

2. Click Project -> New Source, then choose "VHDL Test bench" and chose the 1-bit wide 2-to-1 mux as the associate source. Then use the downloaded testbench file and run the simulation to verify your design.

The next one would be designing and testing a **1-bit full adder**. You will write your own testbench for this one.

1. Create a new project or add new source to existing project by clicking Project-> New source, then choose "VHDL Module" and write a code for a 1-bit adder.

2. Click Project -> New Source, then choose "VHDL Test bench" and chose the 1-bit adder as the associate source. Then complete the testbench and run the simulation to verify your design.

Demonstration

Submit video via beachboard.

Lab Write-up

You will turn in a hard copy of your laboratory writeup. Your lab report should include the following –

- VHDL code for 4-bit AND, 1-bit 2-to-1 mux, and 1-bit adder
- VHDL code for adder testbench
- Screenshots of the simulator for aforementioned programs showing different states