LAB #5: Combinational Blocks and VHDL Simulation

This lab is due electronically by midnight on Tuesday, March 6, 2001. NO LATE HOMEWORK WILL BE ACCEPTED.

1 Laboratory Objectives

This laboratory covers chapters 6 of the book on combinational building blocks. In addition to a few book problems, learn to simulate VHDL.

2 Setup

Startup PowerView using the pvcade or pvcs script. Next, you must create a project. To do this, pull down the project menu and select create. Note that when powerview prompts you for a name for the new project, it is assuming that the current project is the first part of the name. This is almost never what you want! So, if a current project is selected, you should type the entire path name to the new directory for the new project. For example, if I am going to call this new project lab5, I would type /home/ee/prof/cmyers/3700-pv/lab5. Of course, your home directory will be different, but you get the idea. You can use the browse button in the project-create dialog box to help get the directory figured out for the new project. The ViewDraw type for the project is fine.

Next, you must tell the tool where your library files will be. To do this, double-click on the VhdlMngr icon in your toolbox. After it brings up a new window, click on the List system lib button. This will list several system libraries including the ieee one. Double-click on ieee.lib/ which will add this library to your search path. If you click on 44 elements in the righthand window, it will tell you what is in the library. If you click on 1 element next to std_logic_1164 and then click on view it will show you the actual VHDL code used to make up that package in the library.

In order to dismiss this window or any other window, use the red button to pull up a menu in which the last option is Dismiss Window. Also, all the help is on-line and is available under red button->help->ViewDoc.

Now, while still in the VHDL manager, you need to create your own library. Pull down the file menu and select create library. The default name is user.lib which is fine. Select a symbolic name for it (ex. LAB5) and press ok.

Next, you need to make your library the working library. Do this by pulling down the edit menu and select set working. LAB5 will go to the top in the righthand window.

Last, you must save your library search path. Do this by pulling down the file menu and select save INI file. You can now dismiss the VHDL manager by pulling down the red button menu and selecting dismiss window.

Now, exit powerview and go back to your shell window. Create a directory for your VHDL files by typing:

mkdir ~/3700-pv/lab5/behv

All VHDL files from the book are on the CD in the back of the book. You can also find them in:
acmyers/ee3700/VHDLcode.
For this tutorial, you should type:

cmp ~cmyers/ee3700/VHDLcode/Chap5/Figure5.23/FULLADD.VHD ~/3700-pv/lab5/behv/fulladd.vhd
3 Preparing your VHDL code

Initially, you should skip this step, and compile and simulate the example files. Once you are comfortable with them, using your favorite editor you should prepare your VHDL files to complete the homework problems above. At first, the simplest approach is to put everything which you need into one file for each problem.

4 Analyzing your VHDL code

Once you have a VHDL model, you need to compile it (i.e., analyze it) before you can simulate it. This step will check for syntax errors and type mismatches as well as create some intermediate files used by the simulation tool.

To analyze your code, start powerview (i.e., type pv-cade or pv-cs). To analyze one of your VHDL files, double click on the VhdlAnalz button in your toolbox. In the left side of the window brought up, double click on the behv directory and on the right side double click on the file name you wish to analyze (ex. adder.vhd). The VHDL file is then analyzed and any errors will be displayed in a new window. If there are errors, go edit your file and try again. If not, dismiss the window.

5 Simulating your VHDL model

To simulate your VHDL, you will use the Fusion SpeedWave tool which is entered by double clicking on its icon in your toolbox. You can simulate either a schematic, configuration, or entity/architecture pair. Click on the VHDL Entity/Architecture button. There is a field for, Configuration/Entity at the Top of Hierarchy, enter that name (ex. fulladd) and press OK.

The new window will show your source code in the top half and give you a shell like window to enter simulation commands. You can type help to get a list of simulation commands. The simulation commands can also be run from the buttons and pulldown menus in the bottom half of the screen. For example, you can pull down the set menu and select L to set a wire low, H to set a wire high, X to set a wire to unknown, and Assign to assign a value (for example to a std_logic_vector). In the first three cases, it will ask for a node name. The name consists of a path and the signal name. For example, in the adder, the nodes are /Cin /z /y /s /Cout if it is the top element in the design being analyzed. If it is a component in 4-bit adder being analyzed and the name of the first fulladder component is stage0, then the signal names are /stage0/x /stage0/y, etc. You can also assign a value to a node which is useful if the node is a std_logic_vector, string, integer, etc. No assignments actually take effect until you press the sim button. If you have a testbench to exercise your design, you can simply press the sim button without making any particular node assignments.

To display the values in the shell window, use the display menu and value submenu. If you select node, it will ask you for which node to display. If you select all, then all wires will be displayed. To see waveforms, use the Trace menu and select in wave window. It will prompt for which nodes (ex. /x /y /Cin /s /Cout).

6 What to turn in for lab/homework #5

You should copy into this directory your VHDL files and also the examples from ~cmyers/ee3700/lab5. This directory includes 4 files:

1. multiplier.vhd (entity declaration for problem 1).
2. prob2.vhd (entity declaration for problem 2).
3. prob3.vhd (entity declaration for problem 3).
4. prob4.vhd (entity declaration for problem 4).

This homework will be turned in electronically by typing: handin cs3700 lab5 filename. You should handin four files, one for each design using the names given above and entities provided. In problem 1, you will have several entity/architectures, and they should all be included in the one file. Since we will be testing your designs automatically, pay close attention to getting the top-level entity declaration and file names correct. Be sure to comment your code to receive full credit.

7 Problems

1. Simulate your multiplier from Homework/Lab 4.

2. Show how the function $f(w_1, w_2, w_3) = \Sigma m(0, 2, 3, 4, 5, 7)$ can be implemented using a 3-to-8 binary decoder and an OR gate. Write VHDL code that represents this implementation of this function using a selected signal assignment.

3. Consider the function $f = \overline{w_1} w_3 + w_2 \overline{w_3} + \overline{w_1} w_2$. Use a truth table to derive a circuit for $f$ that uses a 2-to-1 multiplexer. Write VHDL code that represents this implementation of this function using a conditional signal assignment.

4. For the function $f(w_1, w_2, w_3) = \Sigma m(0, 2, 3, 6)$ use Shannon’s expansion to derive an implementation using a 2-to-1 multiplexer and any other necessary gates. Write VHDL code that represents this implementation of this function using an if-then-else statement.