Appendix C
Xilinx VHDL Entry and Synthesis Procedures

VHDL stands for "VHSIC Hardware Description Language." VHSIC, in turn, stands for "Very High Speed Integrated Circuit," which was a U.S. Department of Defense program to encourage research on high-performance IC technology.

The only real difference between the schematic and VHDL design flows is the method by which the netlist is extracted from the design description. Therefore, this procedure will only discuss the steps of entering the VHDL code for the circuit design and the synthesis of the netlist from the VHDL code. The steps which follow (functional simulation, compiling into a bitstream, downloading the bit stream to an XS board, and testing the downloaded design) are performed exactly as they were in the schematic design flow discussed in the previous section.

1. Double-click on the Xilinx Foundation Project Manager icon.

2. You will get a screen Getting Started. Click on Create a New Project and then click OK. You will get a screen - New Project. Enter your project name in the box Name: and click on HDL. Click OK after this.

3. You will come to a screen that shows you the directory and some default files that have been created. There are three main panes in the Project Manager window. On the upper right-hand pane, you can find a box-Design Entry. Click on the leftmost icon, which has the shape of a paper with HDL (icon for the HDL editor). You now see the HDL editor in a new window.

4. In the window that appears, click on Use HDL Design Wizard and then click on OK.

5. Then click on Next in the Design Wizard window. Here, select the VHDL and click Next to move to the Design Wizard-Name window. Enter your file name and click Next.

6. Now you should be in the Design Wizard - Ports window where you specify inputs and outputs for your design. To add an input, click on New and then type your input variable in the Name field. Click on Input to see the port to be an input. Repeat these steps for all your inputs. To add your output of your circuit, click on New, enter your output variable and click on the Output. Repeat these steps for all your outputs. In the case your input or output is a vector, you can use Bus option. Now all the inputs and outputs are defined so click on the Finish.

7. If you are familiar with HDL editor, you can click on the Advanced button to bring up the Advance Port Settings window. The drop-down menu in this window lets you set the type of any of the inputs or outputs. The default setting is std-logic.

8. At this point, an HDL Editor window will appear with the skeleton program. The first line of the file uses the LIBRARY keyword followed by the names of the libraries you want to use in your design. Libraries are used to encapsulate functions that are generally useful in a wide

variety of designs. You can access the macros, definitions, and functions from the IEEE library. A library can be subdivided into package, which further encapsulate features useful in a certain area of type of application. The USE keyword in line 2 indicates that our design will have access to ALL the features found in the std-logic-1164 package of the IEEE library. The IEEE library and the std_logic_1164 package are standards, which are support by the VHDL tools.

9. Following the library access control lines define the interface to your circuit. A VHDL entity is simply a declaration of a module's inputs and outputs, while a VHDL architecture is a detailed description of the module's internal structure or behavior. The interface declares the inputs and outputs which an external circuit can use to gain access to the features and functions of the circuit.

10. You will use the std_logic type which allows logic signals to take on the standard 1 and 0 Boolean states as well as the undefined, high-impedance, and other states.

11. The interface definition is followed by an architecture definition. The VHDL statement in the architecture section describe how the circuit actually carries out the operations on the input/output values passed through the interface. You should enter your statement here. Like other high-level programming languages, VHDL generally ignores spaces and line breaks, and these may be provided as desired for readability. Comments being with two hyphens and end at the end of a line.

12. Now we have to check to make sure we have not made any mistakes. Select the Synthesis → Check Syntax menu item. A small window below the HDL editor will appear informing you that the VHDL code is being examined for errors. Within a few seconds, it will state Check Successful if there is no error. Click OK in the pop-up window. If there are errors, each error will be highlighted in the HDL Editor window and an error message will appear at the bottom of the window. You can get error-free VHDL code examples for reference by selecting the Tools → Language Assistant menu item.

13. VHDL is not synthesized within the HDL Editor window so the Synthesis → Synthesize menu item is blanked out.

14. Now that the design entry is complete, select File → Save in the HDL Editor window. Then select File → Exit.

15. Upon returning to the Project Manager window, you must make yourfilename.VHD file a part of your project. Select the Document → Add… menu item and list items of type HDL (yourfilename.VHD) in the dialog window. You should then see your VHDL file under your project name.

16. Now, you have to map your input and output variables to the input and output pins on the board. For this , you should double click on the .ucf file under your project name. You will see Report Browser window. Go down to the end of the file and type "NET your *input or output* LOC=p*number*,". For example, if you use "A" for your input (MSD), then type "NET A LOC=p48,". Add all other inputs and outputs in the same manner. Now the pin

assignment is complete, select File → Save and the File → Exit.

17. Once the VHDL source file complete, you next need to extract its netlist. Select the Synthesis → Force Analysis of ALL HDL Source Files in the Project Manger window. This indicates a check of all the VHDL files to detect any errors. If there is no error, a green checkmark will appear by the .VHD file name under your project name.

18. Next, click on the SYNTHESIS button in the right-hand pane of the Project Manager window. This brings up the Synthesis/ Implementation window.

19. When the window first appears, the name of the interface for your file will be listed in the Top Level text box. The Version Name box shows ver 1 and this will be incremented each time you change the source code for the design.

20. In the Target Device are of the window, you will select the family, particular device type, and device speed in the drop-down menus. This lets the synthesis software know the type of chip architecture you are targeting so it can generate a netlist that takes advantage of the features of the chip. Select Family name XC9500 and Device name 95108PC84.

21. You can also use the controls in the Synthesis Settings are to direct the synthesis tools to emphasize high-speed or area-efficient circuitry. There is also an Insert I/O Pads checkbox, which controls whether input and output buffers will be placed on all I/O signals. This box should be checked if your .VHD file is at the top level of design. (This box would not be checked if the .VHD circuitry is included as a macro in a larger design.)

22. Clicking on the Run button starts the synthesis process. If the synthesis is successful, you should see a green mark in the SYTHESIS box of the Flow tab in the Project Manager window. You will see a red cross in event of a failure.

23. At this point, you have extracted a netlist from the VHDL code that describes your circuit. With this netlist you can do functional simulation, compile the netlist into a bitstream, and download and test the bitstream to an XS95 Board in exactly the same way as was done in the previous lab on the schematic design. Refer to appendix A if you have forgotten.

24. Test a functional simulation.

25. Download your circuit onto the prototype board and test it.