Using Xilinx ISE Tools

Tom Kelliher, CS 220

Fall, 2003

This document will take you through some of the basic steps involved in using the Xilinx ISE Tools:

- A typical design flow
- Creating a new VHDL project
- Adding a library to a project (not always necessary)
- Creating VHDL modules
- Synthesizing a VHDL design
- Simulating a design
- Printing simulation results

I assume you will be able to open an existing project yourself. For all of the following, run the *Project Navigator* tool from the *Xilinx ISE 5* tab. **Your first step, before doing anything else, should be to create a Xilinx folder on your G: drive.**

1 A Typical Design Flow

These are the steps you will typically follow in creating and testing a design:

1. Start *Project Navigator* and create a new project.
2. Copy any necessary pre-existing source files to the project directory. (This should be underneath your Xilinx directory.)
3. From the *Project* menu, use *Add Source* to add any pre-existing source files.
4. Create any necessary libraries and move any appropriate VHDL package files to them.
5. Use *New source* under the *Project* menu to create any new source files, whether they be VHDL or Test Bench Waveform. For VHDL files, check for syntax errors and correct. Save your work.
6. Synthesize your design. Correct any errors.
7. Perform a behavioral simulation.
8. As required, print VHDL source and simulation waveforms.

2 Creating a New VHDL Project

1. From the *File* menu, choose *New Project*. A dialog box will open.

2. Fill-in the requested information. **"Project Location" should be your Xilinx directory on your G: drive.** "Project Name" should be a descriptive project name. For "Device Family" select `Spartan2`. For "Device" select `xc2s50`. For "Package" select `tq144`. For "Speed Grade" select `-5`. For "Design Flow" select `XST VHDL`.

3 Adding a Library to a Project

You won't need to do this very often. One situation where you would have to do this is if you use the `LCDF_VHDL` library with structural VHDL.

1. From *Project* menu, choose *New Source*. Select "VHDL Library" and fill in a library name. For example: `lcdf_vhdl`.

2. The new library will be visible in "Library View."

3. Move the appropriate VHDL packages to the library.

For example, before creating the `lcdf_vhdl` library, go to the course home page and download `func_prims.vhd` into your project folder. From the *Project* menu, select *Add Source* and select this file. If prompted, set the "Source Type" to "VHDL Package." Next, in "Sources in Project," right click on the file and select *Move to Library*. Select the `lcdf_vhdl` library.

4 Creating VHDL Modules

1. Once you've opened your project, you can double-click any .vhd file in the Files tab window to open it in the editor.

2. To create a new VHDL file and edit it, open the *Project* menu and choose *New Source*. Select `VHDL module` and fill in a value for the "File Name."

3. The next dialog box may save you some time when entering an entity's port information. If you don't want to use this feature, move on.

4. To check a module's syntax, select the module in the "Sources in Project" window, Expand the "Synthesize" item in the "Processes for Current Source Window" and double-click on "Check Syntax." Scroll through the "Console" window to locate any syntax errors and correct.

5 Synthesizing a VHDL Design

Each time you change your source VHDL, you need to re-synthesize before running simulation. (Synthesizing a design is similar to compiling a program.)

1. Select the appropriate VHDL module in the "Sources in Project" window.

2. Double-click on "Synthesize" in the "Processes for Current Source" window.

3. Correct any errors.

6 Simulating a Design

First, we use HDL Bencher to specify test vectors, then we use ModelSim to simulate the design, using the test vectors as inputs.

If you haven't separately installed ModelSim, you won't be able to use it. Once you have installed it, you have to let the Xilinx ISE tools know where to find it. You do this by opening the *Edit* menu, selecting *Preferences*, and choosing the *Partner Tools* tab. The entry for "Model Tech Simulator" should be set to something like `C:\Modeltech_xe_starter\win32xoem` (depending upon where you chose to install ModelSim). Be sure to follow the instructions regarding restarting ISE once you set this entry.

ModelSim doesn't appear to like more than one library per `use` statement. Check your VHDL to make sure this is the case, otherwise you'll get a vague error message.

Okay, here's where you test your design to determine if it behaves as intended. Just as proper selection of test cases is critical in debugging a program, proper selection of test vectors is critical in debugging a hardware design.

1. First, if you've made any VHDL changes, re-synthesize your design.
2. From the *Project* menu, choose *New Source*. Select "Test Bench Waveform" and choose a file name.
3. In the next dialog box, make sure the waveform is associated with the correct piece of VHDL.
4. Accept the defaults in the *Initialize Timing* dialog box.
5. HDL Bencher should now be running. You use this tool to specify your test vectors. Click on the blue fields to set the input waveforms.

   Once you're finish, save your work and close HDL Bencher.
6. Back in Project Navigator, select your test bench waveform file from the "Sources in Project" window. In the "Processes for Current Source" window, double-click on "Simulate Behavioral VHDL Model."
7. ModelSim will begin to run and open several windows. Watch the "ModelSim XE II" window for messages. Assuming all went well, the input and output waveforms will be visible in the "Wave" window. In this window, select the *View* menu, choose *Zoom*, and select *Zoom Full* to see the entire simulation.
8. When you're finished, close the "ModelSim XE II" window to close all the ModelSim windows.
9. Back in *Project Navigator*, you can double-click on the Test Bench Waveform file in order to edit the test vectors.

7 Printing Simulation Results

In the ModelSim "Wave" window, open the *File* menu and choose *Print*. It's customary to print waveforms in landscape mode. This can be chosen by clicking the *Setup* button. You also generally want your waveform to occupy as few pages as possible, but still be readable.