

ECE 327 Synthesis Lab Tutorial

June, 2004

1 What's this about?

This is basically a beginners guide to using digital design design tools through the steps of design entry, synthesis, and hardware implementation. This will require you to become familiar with and use several different software and hardware components. Working with these tools will provide you with hands on experience with the design concepts that you learn in ECE 327.

2 Useful Terminology

- *Verilog*- A hardware description language (HDL) used to represent digital designs (both structural and behavioral) in a programming language.
- *Simulation*- The process of checking a design with software tools so that you can verify it's correctness and see its behavior before actually implementing the design.
- *Synthesis*- The process of translating a HDL representation of a design into a form that can actually be implemented with physical logic devices.
- *Download*- This is the process of transferring a physical hardware design from the PC to the FPGA device and loading it.

3 Lab Hardware

3.1 Workstations

Most of this work can now be performed on ordinary PC workstations. For this lab it will be helpful to be familiar with the Linux operating system environment. Those of you who have taken ECE L272 with Linux based computers will find these machines to be very similar for the most part.

3.2 XESS Corp. FPGA kit

This is the fun part. Each lab machine is equipped with a programmable FPGA board and prototyping kit. What is an FPGA? Look in your textbook or lecture notes for a technical description, but for the purposes of this tutorial we will just say that it is a reprogrammable digital device that lets you implement the digital designs you create in class. It allows you to quickly (and inexpensively) prototype a design. It can be physically tested without having to manufacture a traditional integrated circuit.

Each FPGA board is attached to a prototyping board that provides several physical interfaces to the device for experimentation. If you look at the circuit board you will see that it includes the following: 3 buttons, 8 DIP switches, 3 LED displays, 2 VGA (video) connectors, 1 ps/2 (keyboard) connector, 1 parallel port, 2 audio connectors, and many connector pins. We have the potential to create designs to control all of these things in hardware. For example, we could create an audio processor, graphic equalizer, or video controller, and actually implement them to see the results.

The FPGA board and PC are connected by a parallel port cable so that we can download new designs into the FPGA as needed.

4 Software Tools

4.1 Synopsys (Simulation)

This is a large software package which lets you create HDL designs and simulate them. It is available on the Riggs 10 workstations. You can use it to analyze files, and view waveforms of the design.

4.2 Xilinx ISE Series (Synthesis)

This is another related software package made by Xilinx. It also allows you to enter designs and simulate them, although we prefer to use Synopsys for simulation at this time. The reason we use the Xilinx software is it supports the specific model of FPGA hardware that we wish to use. Its main role will be to perform the device specific synthesis and optimizations.

4.3 XESS XStools version 4 (Download, Test, etc)

This is the software toolset available under Linux that allow us to download our designs onto the actual FPGA devices. Xsload configures the FPGA. Xsport allows us to send signals down the parallel port cable for testing purposes. Xstest is a diagnostics program.

5 Tutorial

5.1 Logging in and Setting up the Tutorial Environment

1. The TA will supply you with a new account and password for the 327 machines. These accounts are NOT your university or engineering account! These machines are independent and have different files, passwords, etc. Your password for these machines must contain at least one letter, one number, six characters, and cannot be based on a dictionary word. Once you receive your new account, login in with your supplied username and password.
2. Next right click on the desktop and select new -> terminal and Type the following command at the prompt:

```
cp -r ~walt/ece327/xstools4 ~<your user id>
```

You should now have a new sub directory in your user id home titled xstools4. This directory should contain the following two files, xsbrdinf.txt and xsparam.txt. These files are used by xsload when downloading to the FPGA. In addition there should be two other directories, docs and xsa. The docs directory contains all the documentation related to the XESS FPGA prototyping boards you will be using in the lab. It also contains tutorials and information about the Xilinx ISE and xsload tools. The remaining directory is used by other XESS applications in particular the xstest utility that you will be using shortly.

3. The next thing that needs to be done is to set an ENV variable to point to the config files in the newly created xstools4 directory. Type the following command at the terminal:

```
setenv XSTOOLS_BIN_DIR "/users/<your user id>/xstools4"
```

Now type "env" at the command prompt and verify that the XSTOOLS_BIN_DIR variable is now set. At some point you will also want to include this command in your .cshrc file so you don't have to type it in every terminal window you want to run xsload out of.

4. Now that all the files are copied and the environment variables are set, it is time to use the xstest command to verify the XSA-50 FPGA board is in good working order. First check to make sure that the 9v power and the parallel cable are plugged into the XSA-50 board then type the following command in the terminal window:

```
xstest
```

If the previous steps were completed correctly you should see a string of configuration messages ending with "Your XSA50 passed the test!" and a zero on the led display next to the 9v power connection.

5. Finally, the last step before running the Xilinx ISE application is to make a tutorial directory and copy some verilog files to it. You will use these files with the ISE software to create a bitstream file that can be downloaded to the XSA-50 with xsload utility.

Type the following commands at the prompt:

```
mkdir ~<your user id>/tutorial
cp ~walt/ece327/synth-tutorial/leds.v ~<your user id>/tutorial
cp ~walt/ece327/synth-tutorial/leds-xsa50.ucf ~<your user id>/tutorial
```

Your tutorial directory should now contain two files, leds.v and leds-xsa50.ucf.

5.2 Running Xilinx ISE 6.1i

1. Launch the Xilinx Project Manager by typing “ise &” at the terminal command prompt.

Note: It can take several moments to launch.

2. From the top menu bar choose “File– >New Project”.
3. Enter the project name as “tutorial”, the project location as “users/< youruserid >/tutorial”, and select the “HDL” option for the top level module type. Then click “Next”.
4. The next panel in the configuration wizard is setting the HDL parameters and FPGA type. For the device family line left click on the value field and select “Spartan2”. This is the device family name of the FPGA. Verify or set the rest of the fields to the following values. Be very careful with this part or the generated bit file will not work correctly.

```
Device Family = Spartan2
Device = xc2s50
Package = tq144
Speed Grade = -5
```

```
Top Level Module Type = HDL
Synthesis Tool = XST(VHDL/Verilog)
Simulator = other
Generated Simulation Language = Verilog
```

5. The next panel is titled “Create a New Source”. This will create a new verilog source file template. For this tutorial we are using an existing file so just click on “Next”.
6. For the “Add Existing Sources” panel left click on “Add Source”. You should see the files you copied from the previous section. Hold down the “Ctrl” key and select “leds.v” and “leds-xsa50.ucf” then click “Open”. Then choose “Verilog Design File” in the “Choose Source Type” window that pops up. Click “Ok” then “Next”.
7. You should now see a panel titled “New Project Information”, verify that everything is correct and click “Finish”.
8. You now have a fresh project area to work with in the ISE tools. The main project navigator window has 4 sections. The bottom most one (Console) logs messages and errors from the current operation. The top left side (Module View) shows files active within your project. The middle left side (Process View) provides control over the design process. The field on the upper right is used to display files.

9. Left click on the “Module View” tab in the upper left window and you should see a source tree with your project name, FPGA model number, and the verilog files added in the previous section. Double click on the leds.v file and the contents should be displayed in an editor window on the right hand side of the field.
10. The next step is to synthesize the design. Select the leds.v file in the “Module View” window. Then right click on “Synthesize XST” in the “Process View” window and select “Run” from the resulting menu. You should then see the message “Started Process Synthesize.” in the console window at the bottom of the screen. A completed process message will appear in the console window when synthesis is completed and a green check will appear beside “Synthesize XST” in the “Process View” window.
11. Next, the synthesized verilog code must be mapped to the generic components on the FPGA. Right click on “Implement Design” in the “Process View” window and select “Run” from the resulting menu. When complete, a green check will appear next to Implement Design. Expand the Implement Design branch and verify that the individual steps of Translate, Map, and Place & Route are also checked green.
12. Finally, the bitstream used to program the FPGA needs to be generated. Right click on “Generate Programming File” in the “Process View” window and select “Run” from the resulting menu. When the “completed process” message appears in the console window, expand “Generate Programming File”, right click on the “ Programming File Generation Report”, and select “Open Without Updating”. In the report window that will open up, scroll to the bottom and verify the the last lines are as follows:

```
Creating bit map...
Saving bit stream in ‘test.bit’.
BitStream generation is complete.
```

The file xsload will use to configure to the FPGA is test.bit.

13. If everything has completed successfully, then your design is complete. Go on to the next step to download it to the device.

5.3 Downloading the design

1. In order to download the design you must switch back to the Linux terminal you set up with the xstools ENV variable.
2. Type the following commands at the terminal prompt:

```
cd ~/tutorial
xsload -p 1 -b XSA-50 -fpga test.bit
```

Note: The “-b” and “-p” options will be saved in your xsparam.txt file and will not need to be repeated.

3. You will see a progress indicator go across the prompt to indicate how long it will take to download the design. Wait until it is completed.
4. To prove that the design was downloaded correctly, press the button on the prototype board labeled "SPARE" (button is the leftmost of three). The left seven segment LED panel on the prototyping board should light up "0" when this button is pressed.
5. Congratulations! You have completed all of the steps involved in pushing a design through the synthesis tools. Show the TA that your prototype board works before you leave.