

University of California at Berkeley
College of Engineering
Department of Electrical Engineering and Computer Sciences

Lab 3
FPGA CAD Tool Flow

1 Motivation

In this lab you will be using the Xilinx FPGA CAD tool flow from design entry to programming the hardware. The goal for this lab is to familiarize yourself with the software that you'll be using for the rest of the semester.

2 Prelab

1. Examine the circuit that we have provided for you and find out its function. Pictures of the circuit will be posted on the website <http://inst.eecs.berkeley.edu/~cs150/labs/lab3/>.
2. Come up with a testing strategy for this circuit.
3. Skim over “Foundation Series 3.1i Quick Start Guide” in the Xilinx CAD tool documentations. http://toolbox.xilinx.com/docsan/3_1i/docscan.htm

3 Procedure

3.1 Getting Started

First step in the tool flow after you have conceptualized your circuit is design entry. For this lab we just want to familiarize you with the tool flow so we have already entered a circuit for you. To get the design files:

- ? Extract **Lab3.zip** from <\\fileservice\cs150\cs150\lab3> into your home directory.
(Note your home directory should be a mounted networked drive under My Computer)
- ? Run **Project Manager** (from desk top)
- ? Open the project called lab3 from the directory where you extracted the files.

Open up the schematic of the design by double clicking the **.sch** file under the file tab on the left side of the screen. The design is done hierarchically; this means that many of the symbols are composed of smaller pieces. To view the sub-components of a module right click on it and choose **hierarchical push**.

3.2 Simulate the Design

Once a design is entered, its functionality should be verified. The Xilinx tool suite provides a simulator for the netlist that is generated from your design. Launch the simulator by clicking on the button in the simulation box. Use your testing strategy to verify the functionality of the circuit.

You can control the simulation by typing commands in the command window. Here are some useful commands to help you get started with the simulation. (The full list of commands can be found in **Tools->Script editor, Tools->macro assistant**)

vector or **v**, adds a collection of wires to the simulator.

Usage: v "name" "list of wires"

Example: v input_bus input[7:0]

assign or **a**, assigns a value to a signal or collection of signals.

Usage: a "name" "value"

Example: a input_bus 3fh

watch or **w**, adds an individual signal to the simulation.

Usage: w "name"

Example: w reset

high or **h**, **low** or **l**, assigns a value of 1 or 0 to a single signal.

Usage: h "name"

Example: h reset

clock, assign a signal to repeat a set pattern.

Usage: clock "name" "pattern"

Example: clock clk 0 1

cycle or **c**, advance the simulation time by n clock cycles.

Usage: c "n"

Example: c 10 (note: just typing c defaults to one cycle)

If you wish to reuse the same commands many times you can enter the list of commands in a text file, you can use the script editor provided by the tools or just note pad to do this. Show your TA your simulations of the design.

3.3 Implementing the design

The next step in the tool flow is implementing your design. This involves translating, mapping, placement, routing, and bit-file generation. The Xilinx tool suite merged these steps into a tool called *Flow Engine*. The Xilinx *Flow Engine* takes (EDIF) a netlist generated from your design and compiles into a bit file that can be downloaded to the Xilinx chip.

Start the Flow Engine by clicking on the Implementation box (located under the Design Entry box). In the dialog box that comes up after clicking on Implementation, set the Device to **4010XLPC84**, and Speed of 1. Then run. A new window will pop up showing the progress of the compilation and routing. To look at the results of the compilation, click on the "**Versions**" tab in your project manager. Right click on the revision that you need information on, and select "**View Implementation Log**" or "**Invoke Interactive Flow Engine**". Both will show you a log of the compilation. You can also access the log by clicking on the "**Reports**" tab and double clicking on "**Implementation Log File**". Look through the log files. They provide information such as the amount of FPGA resources that your design uses (LUTs, Flip-Flops, CLBs, IO pads,

Clock buffer...), the estimated clock speed that your design can operate at, and all the error and warning messages.

3.4 Analyzing implementation, Timing Analyzer

Included in the Xilinx tool suite is a “static timing analyzer” that will automatically find the path delays in your implemented design. This tool is useful in a performance sensitive design where knowing the critical path will help in optimizing circuit performance.

Start the timing analyzer by click on the **Timing Analyzer** button under **Verification**. Once the timing analyzer is launched choose **Report path and timing constraints**, select **report unconstrained paths** then proceed. Take a look at the generated timing report. What is the delay of the critical path in this design? How many levels of logic does it have? Try tracing the critical path in the schematic file. (Hint: if you highlight components and wires in the schematic their name will show up at the bottom of the schematic editor. You’ll also need to use hierarchical push to look inside of the modules)

3.5 Analyzing implementation, Floor Planner

The floor planner tool provides a way for the user the manually choose the placement of some or all of the design onto the FPGA. For certain kinds of designs where speed is critical it is preferable to hand place part of your design to obtain higher performance. Select **Tools->Implementation->Floor Planner**, to see the current placement of the resources. (Optional) Try doing your own placement in the editable floor plan window to get a taste of what the computer is doing when it is placing your design.

3.6 Analyzing implementation, FPGA Editor

FPGA editor is a tool that graphically displays the resulting mapping of the logic in the Xilinx chip. This tool also enables modification of the FPGA configuration on the lowest level. Select **Tools->Implementation->FPGA Editor**. Choose **File->Open** and open an implementation to see how it is placed out and routed on the Xilinx chip. This should look like what you saw in the floor planner except routing and other detailed information are included. By selecting nets in the list, you can see how the compiler has routed your wires.

3.7 Download the circuit

Click on the “**Programming**” block in the program manager and select the hardware debugger. If it complains about the cable not being connected correctly, make sure the board has power and the XChecker cable is attached. You can also check the cable configuration by choosing **Cable->Communications**. The XChecker cable is hooked up to **COM1** and the transfer rate should be 115 Kbps. Many of the XChecker cables can be plugged in backwards, please make sure that the two missing pins on the boards matches the ones on the cable. Choose **Download->Download Design** or double-click on the name of the **.bit** file in the diagram to download the design to the Xilinx board.

3.8 Observe the circuit in operation

We will use the digital analyzer to observe the circuit function. You will need to connect to 8 pins that connect the FPGA to the LED bank. The pins that you need to connect to are 60, 59, 58, 57, 66, 65, 62, 61.

4 Acknowledgement

Original lab by J. Wawryznek, N. Zhou.

Prelab Questions

1. What does this circuit take for inputs and output?
2. Describe the function of this circuit
3. Describe your testing strategy for this circuit.

Questions

4. How many 4-LUTs does your design use? Flip-Flops? CLBs?
5. What is the maximum clock frequency of the design?
6. How many level of logic is in the critical path?

5 Checkoffs

Name: _____ Name: _____

Lab Section: **M:** AM PM **Tu:** PM **W:** AM PM**Th:** AM PM

Prelab Questions

Questions TA: _____ (20%)**Simulation of circuit** TA: _____ (20%)**Traced critical path from schematic** TA: _____ (20%)**Working design on Xilinx board** TA: _____ (20%)**Logic analyzer waveforms** TA: _____ (20%)

On time TA: _____ (-0%)**One week late** TA: _____ (-50%)