

## ECE 327 Spring 2005

### Assignment 6 - Due 04/29/05

#### Synthesize a Verilog Circuit for a Xilinx FPGA

1. Follow the Verilog Synthesis tutorial to synthesize, load, and run a simple design onto the Xilinx demo boards in the lab in Riggs 215/217. The tutorial document is in the documents section of the class web page.
2. Design a new circuit and synthesize, load, and run it on the Xilinx demo boards. The circuit should have the following specification:

Using the bar LEDs on the demo board (resembles an LED volume meter) light each LED, one at a time, in order from the first to the last, and when the last LED is lit, reverse direction and light each LED, one at a time, in order back to the first. When the first is lit again, repeat the process forever. The effect should be of a "bouncing" LED. The time between LED changes is based on the 100MHz clock provided to the FPGA. Initially, the LED should "move" about once a second. Thus you will need a clock divider, which is simply a counter that counts clock cycles and toggles an output signal once every  $\frac{1}{2}$  second (thus producing a rising edge once a second). This signal should feed into a second clock divider that divides the 1Hz clock by 1, 2, 4, or 8, as determined by a state register. Each time the SPARE button on the demo board is pressed, the state register cycles between 1, 2, 4, and 8 and then back again. The first 7 segment LED display on the demo board shows the current state – thus the LED can "bounce" at one of 4 different speeds.

A few notes: only the first 8 LEDs in the bar are actually connected, though there are 10 of them. The LEDs are active LOW, so when you output a 0 they turn on. Finally, in order to connect Verilog signals to devices you have to determine which pin they are on and assign them to different signals using the "ucf" file. Look at the example for the tutorial design. This shows the 7 segment LED and the SPARE button. All you will need to add is the 8 signals for the bar LEDs. The XSA manual (in the document section) has a table near the back that shows all of the pins and their connections. The blue area to the right is the demo board – the various LEDs and switches are noted there.

This should be a fairly simple design with two counters (the clock dividers) two state machines (one for the LEDs and one for the divider state), the output logic for the 7 segment LEDs, maybe a comparator for the clock divider (just combinational logic) that's it. If you find yourself writing a very complex design, something is probably wrong. REMEMBER, you CAN build a testbench and SIMULATE your design before you try to run it on the hardware. You can do that using iverilog, or using the Xilinx simulator (but you'd have to figure that out on your own).