SEQUENTIAL LOGIC:
TRAFFIC LIGHT REDUX

GOALS

• More Verilog programming in a CAD environment.
• Designing simple sequential circuits and state machines.

OVERVIEW

In this lab you will create digital circuits using Verilog in a CAD environment. You will then program your design onto a field-programmable gate-array (FPGA) and test your circuit in operation.

TASKS

1. STATE MACHINES AND SIMULATOR

• Create a folder named “Lab3” to hold your project somewhere in your network space. Then save the lab 3 files from the lab website [http://www.swarthmore.edu/NatSci/mzucker1/e15/labs-f11.html] into that directory. Save the files from the web site to the folder you created for this project. They are:
  o E15DE2_IO.qsf
  o DetectThreeOnes.v
  o E15Lab3.v

• Launch Quartus from the Start menu (in the “Altera” folder). Select “Create a New Project (New Project Wizard)”.

• Hit “Next >” to get past the "Introduction" page.

• On the next page (“Directory, Name, Top-Level Entity [page 1 of 5]”) choose the working directory that you created in the first step above. For the project name, use “Lab3”. For Top-Level Entity, choose “E15Lab3”. Then hit “Next >”.

• On "Add Files [page 2 of 5]", add the two Verilog files you downloaded (the ones with the .v extensions).

• On the next page ("Family and Device Settings [page 3 of 5]"):
  o Choose Family: "Cyclone II".
  o Choose Available Devices: “EP2C35F672C6” (the number on the large chip on the DE2 Altera board). Note, these are not alphabetical, so you need to scroll through the choices.
2. A SIMPLE COUNTER

Design a Verilog module for a simple counter. Your module should receive a clock and an active low reset as inputs, and provide a 32-bit counter as output. The counter should be cleared when reset is pressed, and otherwise increment by 1 every clock cycle. Simulate your counter and verify that it operates correctly. The counter will be used to slow down the clock in the next task.

3. TRIGGERED TRAFFIC LIGHT

Design a Verilog module for yet another 4-way intersection traffic light. This time, assuming the North-South road has heavier traffic, the light will only change when a car is sensed at the East-West road.

The signals should change according to the following conditions, assuming that one clock cycle is equal about to one second:
• The NS light stays green, and the EW light red, until a car is sensed at the EW light.
• The NS light turns yellow, and stays yellow for about 2 seconds (meanwhile, the EW light is red).
• The NS light turns red, and the EW light turns green for about 4 seconds.
• The EW light turns yellow for about 1 second (the NS remains red).
• Back to initial state (NS green, EW red).

This time, your design and implementation process should use the following steps:

1. Draw a state diagram for a Moore-type state machine.
2. Using Quartus, write a Verilog module for the circuit (assume that reset and the sensor trigger are both active low).
3. Simulate the circuit, making sure that you are testing all cases.
4. Include the counter module from the previous task in your project (copy the file containing the counter module into your project), and instantiate the counter in your top module. Be sure to connect the inputs and outputs appropriately.
5. Modify your main design such that your always block will be triggered by bit 1 of the counter instead of by the positive clock edge. Simulate your design.
6. Connect your clock input to a 27MHz clock, your reset and trigger to push buttons and your outputs to the traffic light cable. Your clock now needs to be slow enough for you to see the lights change, so use bit 24 of the counter of the clock in your main always block to get a clock of about 1Hz.
7. Demonstrate to the instructor that your working circuit works correctly on the traffic light in Hicks 310.

4. EXTRA CREDIT/GOING FURTHER

The solution above is not often used by itself. If the car is not sensed, or if a bicycle comes to the intersection, it is desirable to have the light on the busy road change at some (long) interval. Repeat the steps above according to the following conditions:

• The NS light stays green until a car is sensed at the EW light (use one of the push buttons) or until about 8 seconds pass – whichever comes first,
• The NS light turns yellow, and stays yellow for about 2 seconds,
• The NS light turns red, and the EW light turns green for about 4 seconds,
• The EW light turns yellow for about 1 second,
• Back to initial state (NS green, EW red).

WHAT TO HAND IN

Your lab report should include the following:

• An abstract of what you did in the lab (100 words at most).
• A brief description of the tasks, including a complete description of the inputs, function, and outputs of each circuit.
• A detailed description of the design process for each of the tasks, including your state machines.
• A printout of the well commented Verilog code of your completed designs.
• A section on any problems you encountered and how you solved them.
• A section discussing whether your circuit worked or not, and how you tested it to find this out.
• Detailed descriptions of any extra credit tasks you chose to do.
• A brief conclusion summarizing what you learned.
• A list of references if you used any.