COMBINATIONAL LOGIC:
TRAFFIC LIGHT AND SEVEN SEGMENT DISPLAY

GOALS

• Practice using Karnaugh Maps to simplify Boolean expressions.
• Introduction to Quartus and programming the Altera DE2 board.
• Verilog programming in a CAD environment.

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. GET STARTED USING QUARTUS

• Create a directory named “Lab2” to hold your project. You should put it somewhere in your network folder, so that you can access it from around campus. Then save the lab 2 files from the lab website [http://www.swarthmore.edu/NatSci/mzucker1/e15/labs-f11.html] into that directory. They are Save the files from the web site to the folder you created for this project. They are:
  o E15ClockDivider.v
  o E15Counter.v
  o E15DE2_IO.qsf
  o E15Lab2.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 “Lab2”. For Top-Level Entity, choose “E15Lab2”. Then hit “Next >”.

• On "Add Files [page 2 of 5]", add the three 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.

o Hit “Next >”.

- On the next page ("EDA Tool Settings [page 4 of 5]"), hit “Next >”.
- Hit “Finish” ("Summary [page 5 of 5]”)
- Now, import the assignments for the DE2 board: from the “Assignments” menu, pick “Import Assignments…”, and select the “E15DE2_I0.qsf” file that you downloaded earlier.
- Plug the E15 breakout board into the rightmost GPIO slot of the Altera DE2 board. Then connect the DE2 board to the computer via USB (use the port that says “Blaster” on the DE2 side), and plug the AC adapter in to power up the board.
- Click the purple triangular “Compile” button on the Quartus toolbar. You should get a message after a few seconds saying that “Full compilation was successful”.
- When compilation is finished, click the “Programmer” button on the toolbar (it looks like a blue diamond with steam coming out of it). Then, click the “Start” button in the Programmer window. The Verilog code should start running and the LEDs should begin to blink.
- Carefully read over the contents of the E15Lab2.v file to see what has been implemented. Play with the switches on the DE2 board and see what happens to the red LEDs above them.

2. TRAFFIC LIGHT

The first task is to design a traffic light controller using Quartus and testing it by programming the Altera board. To do so, you will design a combinational circuit that takes as input a 4-bit binary number and outputs the proper traffic light signals for a 4-way intersection (6 outputs).

Your inputs should come from the counter already defined in E15Lab2.v (you will need to modify the code to get it to count to 15 instead of 9). You will first test your design on the Altera board’s LEDs. You will then connect your circuit to the traffic light.
The signals should change according to the following table:

<table>
<thead>
<tr>
<th>Input Values</th>
<th>Light Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>North-South</td>
</tr>
<tr>
<td>0</td>
<td>Red</td>
</tr>
<tr>
<td>1-5</td>
<td>Green</td>
</tr>
<tr>
<td>6-7</td>
<td>Yellow</td>
</tr>
<tr>
<td>8</td>
<td>Red</td>
</tr>
<tr>
<td>9-13</td>
<td>Red</td>
</tr>
<tr>
<td>14-15</td>
<td>Red</td>
</tr>
</tbody>
</table>

Your design and implementation process should use the following steps:

- Create a truth table for all of the possible inputs and outputs (This may be in an abbreviated form such as above).
- Use a Karnaugh map method to determine the minimal circuit design for each of the desired outputs. Write a minimal expression for each output as a function of the inputs.
- Using Quartus, write a behavioral Verilog module for your design called StopLight.v, and add an instantiation into the E15Lab2.v file.
- Connect your inputs to switches and your outputs to red and green LEDs on the Altera board, and make sure the circuit operates correctly.
- Connect the Altera board to the traffic light in Hicks 310. You will need to program the FPGA in Active Serial mode to make sure it retains its configuration when the power is turned off, so that you could move it to the traffic light. See the Quartus Manual linked from the website for instructions.
- Demonstrate to the instructor that design works correctly on the traffic light in Hicks 310.

### 2. SEVEN SEGMENT DISPLAY

Using Verilgo, design a circuit that takes as input a 4-bit unsigned binary number and outputs a decimal representation of this number on one of the Altera board’s seven segment displays. The segments are active low. Segment zero is on the top, and one through five continue around clockwise, with six in the middle.

Your design and implementation process should use the following steps:

- Create a truth table for all of the possible inputs and outputs.
- Use a Karnaugh map to determine the minimal circuit design for each of the desired outputs. Since we are using 4 bits to represent 10 decimal digits, you will have “don’t care’s” in your Karnaugh map. Remember that your outputs...
should be active low – how does this affect the way you derive a minimal expression from the maps?

• Write a minimal expression for each output as a function of the inputs.
• Using Quartus, write a data-flow/behavioral Verilog module for your design called SevenSegment.v, and add the instantiation into the E15Lab2.v file.
• Connect your inputs to switches and your outputs to one of the 7-segment displays on the Altera board, and make sure the circuit operates correctly. Demonstrate your working circuit to the instructor.

3. EXTRA CREDIT

Talk with the instructor to see what you can do for extra credit. Here are some possibilities:

• Design a circuit that flashes up the familiar patterns from a six-sided die on the breakout board when a button is pressed. Each button press should give a different, “random” number.
• Design a circuit that takes as input two 8-bit 2’s complement numbers (from the switches) and outputs their hexadecimal sum using two 7-segment displays and an LED as the sign.

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 (1 and 2), including your truth table, Karnaugh maps, and minimal expressions for the outputs.
• A print out of your nicely indented and well commented Verilog code.
• 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 you used, if any.