SANTA CLARA UNIVERSITY	ELEN 21 Fall 2014	Dr. Shoba Krishnan
Laboratory #4: 4-bit Ripple-Carry Adder		

I. OBJECTIVES

- Learn to do hierarchical design.
- Design a 4-bit ripple carry adder using half adders and full adders as building blocks.
- Use 7-segment displays to show inputs and outputs of the design.

II. PRE-LAB

You are to design the logic circuit for a 4-bit ripple carry adder using full adders.
The 4-bit ripple carry adder is to add two 4-bit inputs (X[3] X[2] X[1]X[0] and Y[3] Y[2] Y[1] Y[0]) and a carry in bit, CarryIN. The result is their sum S[3] S[2] S[1]S[0] and a carry out CarryOUT.

(a) Draw the logic gate schematic of a half adder. Clearly show the inputs of the half adder INA and INB and the outputs SUM and CARRY.
Show all connections and internal connections. Label all inputs and outputs.
(b) Draw the schematic of the full adder that uses the half adders from (a). Clearly show the inputs of the full adder A and B and CIN and the outputs SOUT and COUT.
Show all connections and internal connections. Label all inputs and outputs.
(c) Build the 4-bit ripple carry adder using the full adders from (b). Clearly show the inputs (X[3] X[2] X[1]X[0] and Y[3] Y[2] Y[1] Y[0]), CarryIN and the outputs S[3] S[2] S[1]S[0] and CarryOUT.
Show all connections and internal connections. Label all inputs and outputs.

III. PROCEDURE

1. Creating a Half-Adder:

   (i) Draw the schematic of your half-adder
   (ii) Simulate the half adder and confirm that it functions correctly.
   (iii)Once you have designed and simulated your half adder, you will make a symbol for it.
   Creating a symbol: Make sure you have your design file open and click on:
   File/Create/Update/Create Symbol Files for Current File.
   You should see a message confirming that the symbol was created successfully.

2. Creating a Full-Adder:

   (i) Draw the schematic of your full adder.You will be using the symbol that you have created of your half-adder in the schematic of your full-adder. Do this new design in the same project. This full-adder will be the building block that you will be using to create the ripple-carry adder.
   (ii) Simulate the full adder and confirm that it functions correctly.
   (iii)Once you have designed and simulated your full adder, you will make a symbol for it.

3. Creating a Ripple Carry Adder:

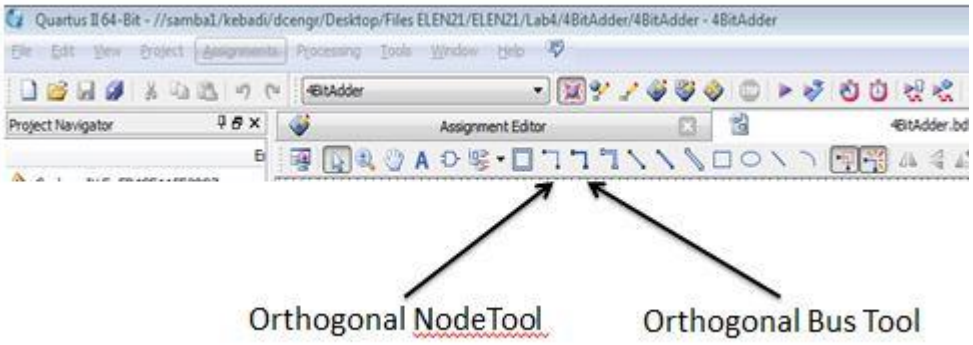
   (i) Draw the schematic of your ripple carry adder. You will be using the symbol that you have created of your full-adder in the schematic. Do this new design in the same project.
   (ii) Simulate the full adder and confirm that it functions correctly.
   (iii) Once you have designed and simulated your full adder, you will make a symbol for it.

4. Assigning Inputs and Outputs:

   (i) When assigning pins for the ripple-carry adder, use switches for X(0) to X(3) and Y(0) to Y(3). Also use a switch for CarryIN, which is the very first carry in. Connect the sum output S(0) to S(3) and CarryOUT to LEDs. Make sure you assign pins in a way that you can see the MSB on the left and the LSB on the right.

   (ii) You will also connect the inputs and outputs to the 7-segment display as shown below. Follow the 7-Segment tutorial and generate the numbers on the FPGA's 7-segment displays.

   (iii) The input of the module is a 4-bit number. Make sure you name the port including **"…[3..0]"**. The input of the module will be connected to a 4-bit number like A, B and S. You will use the **Orthogonal Bus Tool** for wiring. It appears at the top of your page as shown in the following figure.



   (iv) The output of the module is a 7-bit number. Make sure you name the port including **"…[6..0]"**. The output of the module will be connected to a 7-bit number (remember to use the **Orthogonal Bus Tool** for wiring) which goes directly to the pins of the board that correspond to each line of the 7-Segment Display.

The Altera DE2-115 board has eight 7-Segment Displays as described in the tutorial. Choose one for A, one for B and one for S. Cout can be on an LED.

Important! A very useful way to specify connections in the schematic without using a lot of wires is naming the wires as vectors. For example, the input **S** of the **SevenSeg** module for displaying the result of the addition is a wire called **S[3..0]** (4-bit bus of data). Similarly, the inputs and outputs of the **HalfAdder** and **FullAdde**r modules are wires named **S[0]**, **S[1]**, etc.

5. Programming the FPGA:

   Download and demo your design.
   Take a picture of the working FPGA with 7-segment display.

IV. POST-LAB QUESTIONS

What do you expect to happen if you were to add two numbers and have their result be greater than 15? What will be displayed in each bit of the Sum, and what will happen to the Cout LED? How would you interpret these results?

If you had to make an 8-bit adder, how would you use the 4 bit module you have built in this lab to make a 8 bit adder module.

For your lab report, include the schematics and simulation waveforms of all the components you designed.

Note:

The pin assignment file is located at the below link

http://www.ee.scu.edu/classes/2014fall/elen21lab/lab4_pins.qsf