ECE 176 – Computer Aided Engineering in Digital Design
Fall 2014

The 8 x8 Multiplier Instantiations
Assignment 10

Objectives

- *Complete the code to implement the 8x8 multiplier using Verilog structural modeling and instantiations*
- *Synthesize and verify its operation*

Introduction

You will have all of the building blocks necessary to complete the 8 x 8 multiplier. Making use of the knowledge you have gained up to this point, you must now instantiate each components in a top-level design and connect all signals as shown in the block diagram. Once your design compiles and simulates successfully, you have completed the multiplier instantiations assignments project in ECE 176.

Procedure

Step 1: Use Quartus II and Develop Verilog Codes to Implement 8 x 8 Multiplier

Create a top-level design project "lab10" using Quartus II wizard menu for **Cyclone II FPGA** board, and **Verilog** and **Altera ModelSim** design environments.

Write a source code to connect the **8 x 8 multiplier** using **Verilog structural instantiations.** Use the following information as a guide:

- Declare a top-level module named mult8x8 with the top-level ports as shown in the block diagram.
- Create instantiations for **mux4** (2 copies), **mult4x4**, **shifter**, **counter**, **mult_control**, **reg16**, **adder**, and **seven_segment_cntrl** based on the connections shown in Block diagram. Use **u6** as the instance name for mult_control. For other modules use any instance name you wish.
- This will require declaring new internal wires for connecting the blocks. Use the names from Block diagram.

Save the file as mult8x8.v. From the **Quartus II File** menu, select **Save** and save your Verilog file as **mult8x8.v**.

Step 2: Synthesis the Design and Check the Code for Correctness

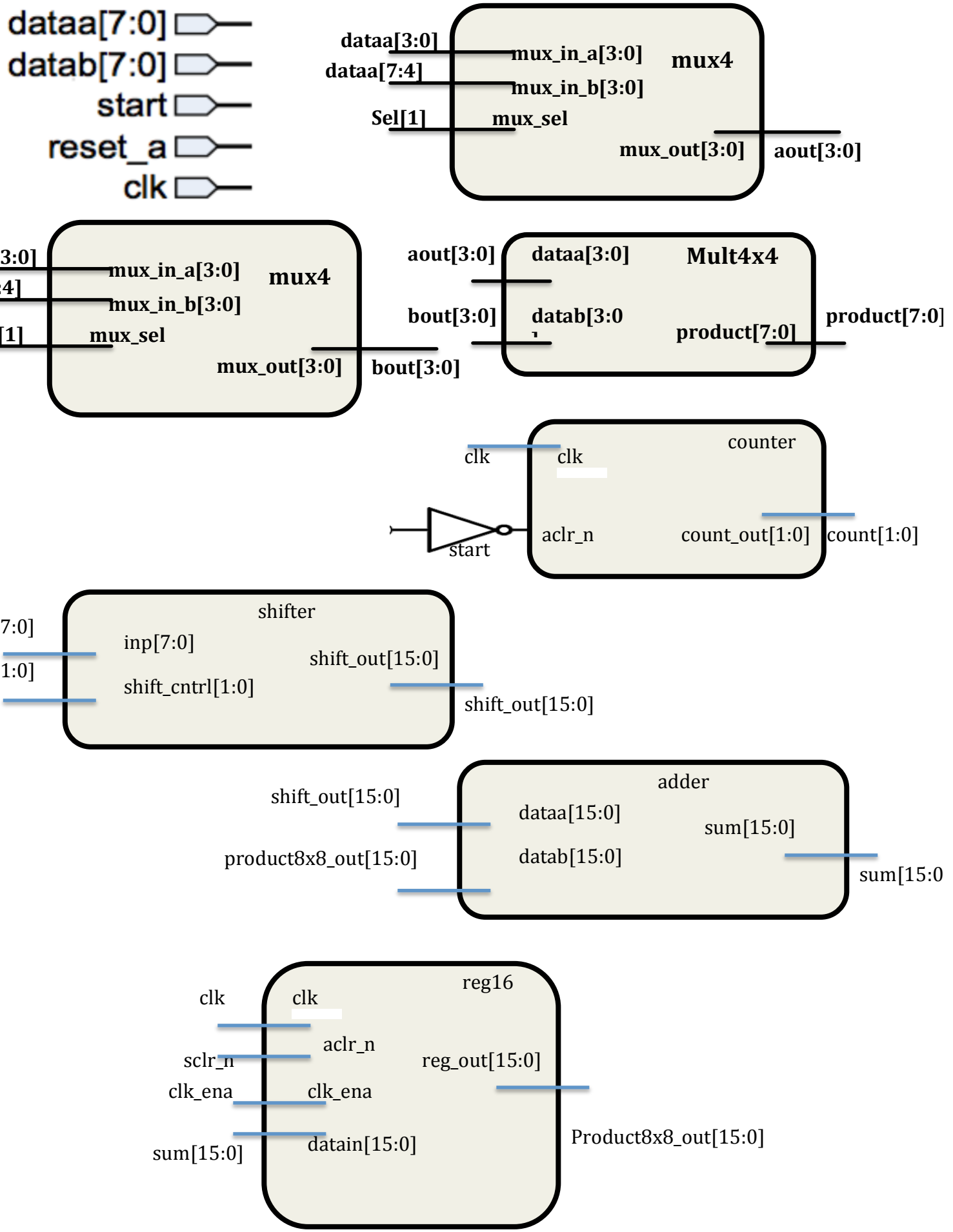
Since the other source files are in different directories, before synthesizing, you must directly add these other source files to this Quartus II project.

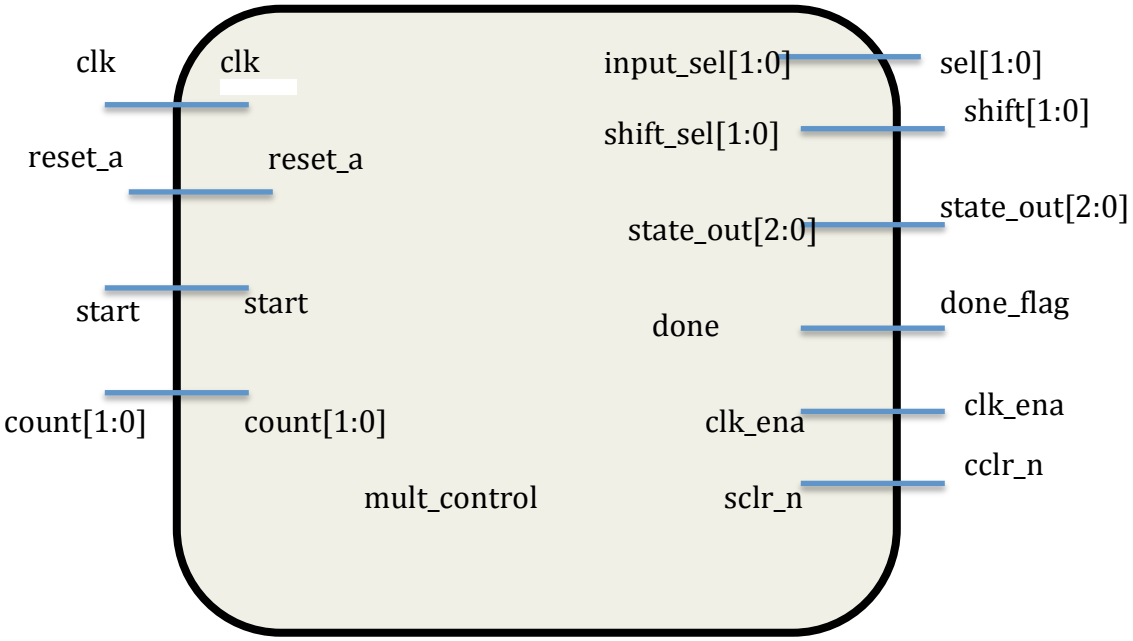
1. Add the source files from the other projects. From the **Quartus II Project** menu, select **Add/Remove Files in Project**. Click on the button  and browse to each of the subdirectories used for previous assignments to add each of the Verilog design files to this project. This includes: **adder.v**, **mult4x4.v**, **mux4.v**, **shifter.v**, **seven_segment_cntrl.v**, **reg16.v**, **counter.v** and **mult_control.v**. MAKE SURE to click on the **Add** button each time so that the file appears in the **File name** list.
2. Synthesize the design. From the **Quartus II Processing** menu, select **Start ⇒ Start Analysis & Synthesis** OR click on the  button.
3. Correct any warnings and errors. Check the **Messages** window of the **Quartus II** software for any warning or error messages. Repeat synthesis and error checking until the **Quartus II** software reports that the "**Analysis & Synthesis was successful**."

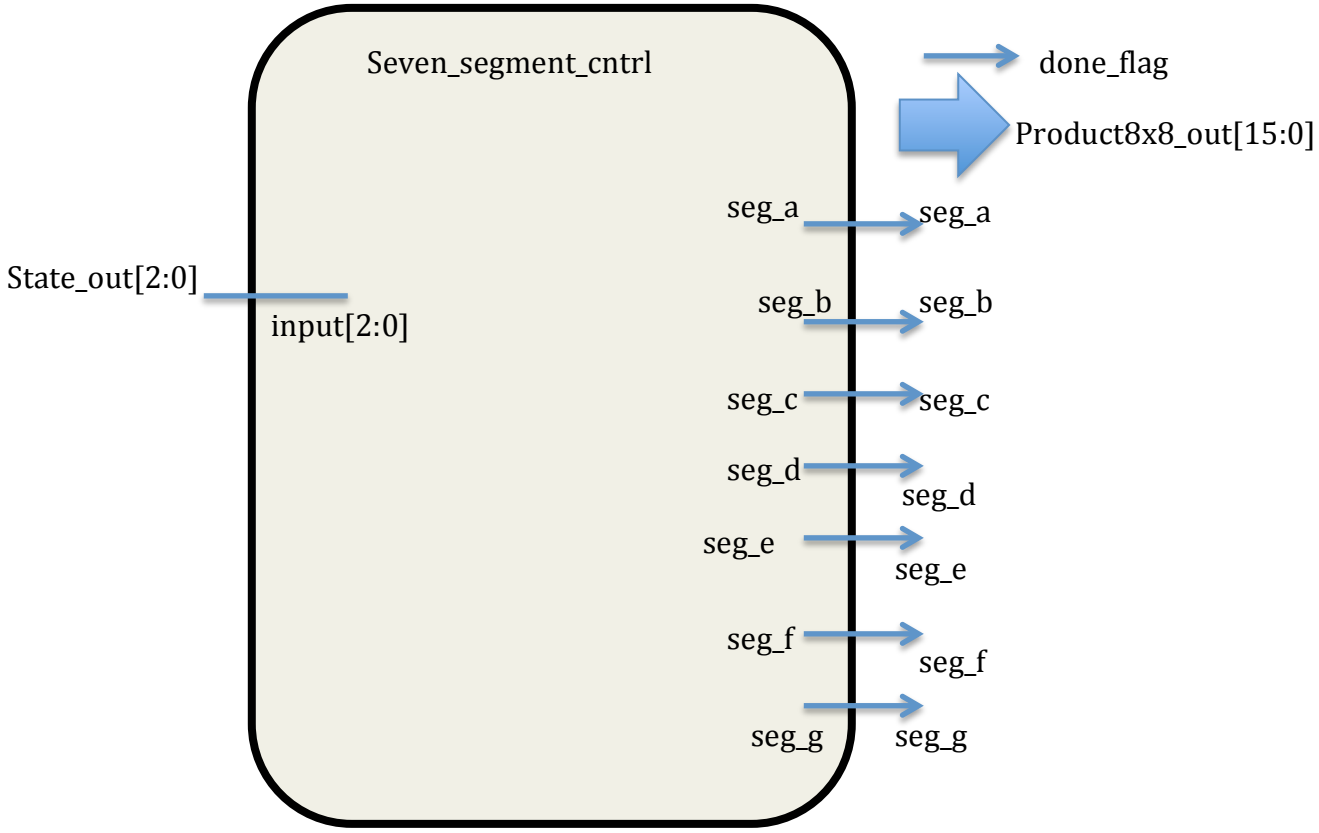
Step 3: Perform an RTL Simulation

*Use the **ModelSim-Altera Starter Edition** simulation tool to verify the functionality of your design. A Verilog testbench file (**mult8x8_tb.v**) has been created for you to provide the test vectors for your RTL simulation.*

1. Set the project directory. From the **ModelSim File** menu to where your current project is located. Run the ModelSim tool and peform simulation.
2. The **ModelSim** tool will now compile all of the Verilog files and start simulation. The waveform window will open (as a separate window) with all of the multiplier input and output signals added so you can verify that your Verilog code is functioning correctly.
3. Check simulation results for correct functionality. Bring the **Wave** window to the foreground. Use the Zoom In tool to zoom until you can see the product output calculations. Demonstrate the simulation results to your instructor.


dataa[7:0]
datab[7:0]
start
reset_a
clk
dataa[3:0]
dataa[7:4]
Sel[1]
mux_in_a[3:0]
mux_in_b[3:0]
mux_sel
mux4
mux_out[3:0]
aout[3:0]
datab[3:0]
datab[7:4]
Sel[1]
mux_in_a[3:0]
mux_in_b[3:0]
mux_sel
mux4
mux_out[3:0]
bout[3:0]
aout[3:0]
bout[3:0]
dataa[3:0]
datab[3:0
Mult4x4
product[7:0]
product[7:0]
clk
clk
start
aclr_n
counter
count_out[1:0]
count[1:0]
product[7:0]
shift[1:0]
inp[7:0]
shift_cntrl[1:0]
shifter
shift_out[15:0]
shift_out[15:0]
shift_out[15:0]
product8x8_out[15:0]
dataa[15:0]
datab[15:0]
adder
sum[15:0]
sum[15:0
clk
clk
sclr_n
clk_ena
sum[15:0]
aclr_n
clk_ena
datain[15:0]
reg16
reg_out[15:0]
Product8x8_out[15:0]





8 x 8 multiplier top-level design block diagram