Project 1: Design an 8-bit hybrid (CLA-CPA) adder

CSc 137, Fall 2015, Faroughi

Due in class on Oct. 21

Also refer to Example 2.16 and 3.5 for how to use an "assign" statement and Example 3.1 for how to model an structural model. However, make sure you follow the specific instructions below to complete your design.

1. **Structural and Bahaviroal design of 2-bit CLA**:
   a. Use the "assign" statement to create a behavioral model of a 2-bit PG unit (PGU) and a 2-bit sum unit (SU). For this you need to use the "assign" statement and enter the Boolean expressions for the PGU and the SU unit using the bitwise operators (&, |, ~). Use SOP expresisons.
   b. Use primitive NOT and NAND gates and create a structural model for 2-bit carry-generate unit (CGU). The CGU generates only two carry bits. You can label them c0 and c1. Use SOP expresisons for the carry bits.
   c. Combine a PGU, a CGU, and an SU module to draw the block diagram circuit for the 2-bit CLA and label all the modules and signals.
   d. Using the top-level circuit in part c create an structural model for a 2-bit CLA. Important: Make sure the final module outputs the two generated carry bits (i.e., c0 and c1).
2. **Structural design of an 8-bit CPA using 2-bit CLA modules**:
   a. Draw the block diagram circuit for the 8-bit CLA using four 2-bit CLA modules from part 1. Lable all the modules and signals. Note, some carry out signals will not be connected to anywhere.
   b. Create a structural model for CPA based using the circuit block diagram in part 2.a. The inputs to the adder are 8-bit *A*, 8-bit *B*, and an initial carry-in bit *cin*. Its outputs are 8-bit output *S*. In addition, you would need the adder to also output the last two carryout bits as c_6 and c_7. These carry bits will be used in a future project.
3. Create a testbench to test your design using the following three test vectors.

   A = 8'hFF; B = 8'h00; cin = 1'b1;

   A = 8'h01; B = 8'hFF; cin = 1'b0;

   A = 8'h55; B = 8'hAA; cin = 1'b0;
4. Make sure you add the test vectors by hand to make sure the result of your hand calculations including the final carry out bits also match those generated by simulating your Verilog code.
5. Merge files including the simulation output into a single file each sperated by a dash line for clarity so the single combind file reuiqres only a few pages to print (i.e., save paper:). Sumbit your report containg also the following information:

1. Name (legible)
2. Section number (Sec 1: 3:00-4:15 or Sec 2: 5:30-6:45)
3. Project number and title
4. Semester (Fall 2015)
5. Print out of your Verilog codes and simulation results (in fewer pages)
6. Write comments on the simulation output illustrating that your circuit ineed works or highlight any problems.
7. Include solutions for items 1.c, 2.a, and item 4.

Follow the instructions 1 through 5 in part 5 also for future project reports.