CMPE 415 - Homework #2

Start assignment as soon as possible. Work individually, but you can ask your classmates for help when you get stuck, with consideration to the course collaboration policy (please read it in the course website). Please send me email if something isn't clear and I will update the assignment. Changes are logged at the bottom of this page.

Before getting started, you should go through the verilog notes located under Course Readings on the course home page.

A paper copy of everything and electronic copies of all your code and testing files (all in one zipped file) are due at the beginning of class on the due date.

Notes:

- [15% of points] Clearly state whether your design is fully functional, and state the failing sections if any exist.
- Make sure your design and code are easily readable and understandable (clear and well commented).
  - Up to 5% extra credit will be given for especially thorough, well-documented, or insightful solutions.
- *** Where three '*'s appear in the homework, perform the required test(s) and turn in a printout of either:
  1. a table printed by your verilog testbench module listing all inputs and corresponding outputs,
  2. an Isim waveform plot which clearly shows corresponding inputs and outputs
  3. a section of testing code which clearly compares the designed circuit and a simple reference circuit, and two short cut & paste sections of text from your simulation (one for pass, and one for fail where you purposely make a slight change to your reference code to make it fail) that look something like this:

    Error: input=0101, out_module=11110000, out_ref=11110001

    In all three options, each test case must be marked whether the output is correct or not.

    Keep "hardware" modules separate from testing code. Instantiate a copy of your processing module(s) in your testing module (the highest level module) and drive the inputs and check the outputs from there.
1. [100 pts] Design and write the verilog for a block that adds three 3 numbers (in different fixedpoint formats) into a 2's complement output that is sufficiently large to represent all inputs but with no extra bits. Use one stage of 3:2 carry-save adders (CSA) and one carry-propagate (CPA) adder for the last stage using a "+" in verilog. The three inputs are as follows:
   - a is in 2's complement 3.4 format
   - b is in 2's complement 2.3 format
   - c is in unsigned 5.1 format

   a) [5 pts] What is the output's minimum attainable negative value (most negative)?
   b) [5 pts] What is the output's minimum attainable positive non-zero value?
   c) [5 pts] What is the output's maximum attainable positive value?
   d) [5 pts] For this range of output, how many bits does the output ("out" signal) have and where is its fractional point?
   e) [10 pts] Draw the adder's block diagram with submodules connections and labels that describes your design.

   f) [50 pts] Design this adder in Verilog and in hierarchy. For the CSA, you need to first write Verilog for a submodule named “fulladder.v” and then using the full adder and its multiple instantiations in the “top.v” module make the CSA and full design working.

   g) [20 pts] Write a testbench and test the design over at least 15 input values (including all extreme cases). Turn in ***, option 1 and 2. This means you need to attach a table of your input and output and ISIM waveform clearly show the input and output values.