HW10
EECS150, Fall 2010
Due Friday, 2:10 pm, November 19, outside 125 Cory

1. Design a datapath to add or subtract single-precision floating point numbers. Note that there are many special cases when doing this right. You don't need to worry about those. Just handle the basics.
    a. Show the blocks and interconnection. You can use adders, shifters, etc., but clearly indicate exactly what the inputs and outputs are (bit widths, format, etc.)
    b. Label the control inputs to those blocks
    c. Show the control block, clearly indicating its inputs and outputs. Is the control block an FSM, or can it be implemented with combinational logic?
2. Design a datapath to implement digital division of unsigned 63 bit numbers using the Newton-Raphson method discussed in class (e.g. http://en.wikipedia.org/wiki/Division_(digital)).
    a. Show the blocks and interconnection.
    b. Label the control inputs to those blocks
    c. Show the control block, clearly indicating its inputs and outputs. Is the control block an FSM, or can it be implemented with combinational logic?
    d. Draw the state transition diagram for the control FSM (I guess that kind of gives away the answer to part of 2c, doesn't it?)
    e. How many cycles will it take to arrive at the correct solution?
    f. If you added an 8 bit LUT to your solution, how would that affect the area and execution time? How about a 16 bit LUT?
3. Using the solution to problem 2 as a building block, design a datapath to implement division of signed (2's complement) 64 bit numbers.
4. With a signed 8 bit floating point number using the IEEE754 style covered in class, what are the smallest (most negative), largest, and smallest magnitude (closest to zero) numbers that can be represented, assuming
    a. the mantissa is 2 bits
    b. the mantissa is 4 bits
    c. the mantissa is 6 bits
5. In wireless sensor networks, some communication routing algorithms rely on a comparison of remaining energy in the routing nodes. To make a general-purpose algorithm, you'd like to be able to represent energies from one microJoule (uJ) to one megaJoule (MJ). To make meaningful comparisons, you'd like to be able to have a resolution of no worse than 2%. For example, if node A has 1mJ (milliJoule), node B has 0.99mJ, and node C has 1.02mJ, it's ok if the representation of the energy in nodes A and B are indistinguishable, but node C should be different. What is the minimum number of bits necessary to represent the energy remaining in a node, assuming
    a. unsigned integer representation
    b. unsigned floating point in the same style as IEEE 754 (covered in class)