Computer Science and Engineering 331
Computer Organization and Design
Spring 2010
Homework #6

Points: 25 (Pair programming assignment)　　Due: Wednesday, Mar 17, 2010 by 5.00pm

1. Design and simulate the VHDL for the 32-bit MIPS ALU with overflow and zero detect depicted in W07&08.24 (slide 24 of week 7&8). The ALU should implement 10 R-type operations (shown in the table in W07&08.24). The first step is to follow the control lines and the functions shown in W07&08.38 to design a 4-bit ALU. Your VHDL description should be *structural* and constructed from 3 *behavioral* VHDL 1-bit ALU blocks, a special *behavioral* VHDL 1-bit ALU block for the most significant bit (also shown in W07&08.38) and a *behavioral* VHDL block for the zero detect logic. The next step is to construct the 32bit ALU by wiring EIGHT 4-bit ALU blocks. The final overflow output of the 32-bit ALU is the overflow output from the most significant 4-bit ALU block. You also need a *behavioral* VHDL block for generating the final zero detection output. Your behavioral blocks should be constructed from CSA's only using the follow "gate library":

inverters (1 ns delay)
2-input NANDs (2 ns delay)
3-input NANDs (3 ns delay)
2-input NORs (2 ns delay)
3-input NORs (3 ns delay)
2-input XORs (3 ns delay

2. What is the best case and worst case delay for your ALU and for what input combinations do these values occur? That is, what is the fastest your ALU will run and for what input (both data and control) combination does this case occur and what is the slowest your ALU will run and for what input combination does this case occur. Assume that all inputs to the ALU arrive at time 0. For the best case and worst case input combinations, what output signal(s) are the last to arrive at their final state?

IMPORTANT NOTE:
Turn in your (completely debugged and simulated) VHDL ALU design *electronically* through the course web page. Note that you are only turning in the appropriately commented *source file* for your ALU (the file containing the VHDL description). It will be re-simulated (in ModelSim) as part of the grading process. The grading simulation will include a combination of and, or, add, subtract, and set on less than operations (signed and unsigned) with both positive and negative input values. Include your answer to question 2 as comments in your ALU VHDL source code. Please name your code files as ALU_lastname1_lastname2.vhd. For full credit, comment your VHDL code appropriately. Each of the partners needs to submit the code, else there would be a penalty imposed on both.