ECE 252 / CPS 220 Homework #3  
Due in class on Monday, October 6  
Total Points 100

**Dynamic ILP**

1) (10 points) If predication enables us to remove conditional branch instructions, by converting control dependences to data dependences, why don’t we just use predication to get rid of ALL conditional branches? Explain when predication might pose a problem. Use an example, if necessary.

2) (10 points) (a) When does dynamic scheduling help the most? When does it help the least? (b) Compare Tomasulo’s algorithm to the MIPS R10000 style of register renaming. What are the advantages and disadvantages of each?

3) (10 points) Compare DIVA’s approach to reliability to the approach of the IBM mainframes that provide high reliability by: (a) replicating the processor three times (i.e., triple modular redundancy) (b) operating the three processors in lock-step synchrony, and (c) using a voter circuit on their outputs to choose the majority data values (which can mask a fault/defect in any one of the three processors). Compare them in terms of (a) hardware cost, (b) fault coverage, (c) performance, and (d) power. Justify your answers, since there may not be one absolutely correct answer to any of these aspects.

**Static ILP**

4) (10 points) H&P 4.6

5) (10 points) H&P 4.21

**Dynamic Scheduling with SimpleScalar**

6) (50 points) Start with the sim-R10K simulator and just use the gcc and go benchmarks. You will NOT have to modify the sim-R10K.c code for this assignment, but you will have to feed it different command line parameters to configure it. If you run sim-R10K without any input parameters, it will spit out all of the possibilities, which should help you to figure out how to specify the configurations in the following experiments.

Experiment #1: Compare in-order versus out-of-order execution (hint: the default is out-of-order, and there’s a flag that can change this). Don’t change any other flags.

Experiment #2: Evaluate the importance of the number of physical registers by comparing a system with 256 vs. a system with 512. Don’t change anything else from default settings.

Experiment #3: Evaluate the importance of the issue window size, by comparing a size of 32 vs. a size of 64. As with all experiments here, don’t change anything else.

Experiment #4: Evaluate the importance of superscalar width by comparing a 4-wide to an 8-wide. Remember that you want to balance the widths of fetch, rename, and commit.

Experiment #5: Evaluate the impact of changing the number of integer ALUs from 2 to 4.

Analysis: Given what you learned from these 5 experiments, explain (a) which issues are most important for performance and (b) where you might choose a lesser performing design point for reasons of power-efficiency and cost-effectiveness. Justify your answers!