CSE471: Computer Design & Organization
Assignment 1
Due: Tuesday, April 7

The purpose of this assignment is to acquaint you with the sim-outorder simulator that is part of the SimpleScalar tool set and the environment in which it executes. Sim-outorder is an instruction-level simulator that implements many of the architectural features we will study this quarter. But for this assignment, we will use it as though it looks very much like the R3000 you studied in CSE378. Sim-outorder implements the SimpleScalar instruction set architecture, which is very similar to the MIPS architecture.

For this assignment, you should:

1. Pick an application from the application directory to instrument. All these programs are taken from the SPEC95 benchmark suite, which was the standard workload for architecture research a few suite-generations ago (we are now on SPEC2008). They have already been precompiled for sim-outorder and you can use them as input to the simulator. Vince will send you email if it turns out that any of the applications are not appropriate for this assignment.
2. Set sim-outorder configuration parameters to reflect a computer that has the following configuration:
    - a pipeline that fetches, decodes, issues, executes and commits one instruction/cycle, no matter what the instruction type
    - only one of each type of functional unit
    - an 8KB, two-way set-associative L1 instruction and data caches with 32 byte blocks
    - a 256KB, direct-mapped L2 unified cache with 32 byte blocks
    - an 8-way, 128-entry data TLB
    - a 4-way, 64-entry instruction TLB
    - All caches are write-back, write-allocate. They and the TLB have an LRU block replacement policy.
    - The page size is 4KB.

    This means that the rest of the parameters should be left with their default values, except for two which must be set to: fetch:speed 1 and issue:inorder true. The configurations are set by command line arguments or sim-outorder's config file.
3. Run the simulation for 100 million instructions executed (set the max:inst parameter appropriately).

Answer or do the following:

1. Print the output generated by the simulator and highlight the configuration parameters you used, which will either come from a configuration file or the command line. (1 point)
2. Highlight and label the values for the following metrics on your sim-outorder output: (1 point each)
   a. The total number of instructions committed – this is all the instructions that have completed all phases of instruction execution, including writing their results to the register file. Why is this number not equal to the number of instructions executed as reported by the output? Is this latter number the same as the one set up in max:inst (10^8)? (2 points)
   b. The total number of branches and their frequency. Does this seem to be consistent with what you heard about branch occurrences in integer programs? (2 points)
   c. The total number of block replacements for the L1 data cache. (1 point)
   d. The hit ratio for the L2 cache. Or. The number of accesses to the L2 cache. If SimpleScalar had not printed it out, how could you have deduced it from other metrics generated by the simulator? (3 points)
3. Answer the following questions: (3 points each)
   a. In general we say the "L1 I-cache miss ratios are negligible, while L2 D-cache miss ratios are not". Is this true for your experiment? (The answer might vary, depending on the application.) What factors could contribute to the non-negligible I-cache miss ratio? (2 points)
   b. A major contribution to the CPI not being 1 is accesses to the memory hierarchy. Try to quantify the contribution to the CPI due to cache misses by decomposing CPI into contributions due to misses in each of the 3 caches. Be aware that hit ratios are related to accesses to the various caches and that you have to convert that into values related to the frequency of instructions. (3 points)
   c. List 3 other possible contributions to the CPI that will make it greater than 1. (3 points)
   d. Why is there a difference between the number of blocks replaced in the L1 D-cache and the number of blocks written back from the L1 D-cache? (1 point)

There is no electronic turn-in required for this assignment. Instead bring to class on April 7 paper copies of the output generated by sim-outorder and your answers to the questions.