CSE 471 Autumn 2001

Computer Design and Organization

Assignment #1

Due: Wednesday October 10

The purpose of this first assignment is to acquaint you with the SimpleScalar environment. You will run simulations using the sim-outorder simulator. sim-outorder is a sophisticated functional simulator that simulates, on a cycle per cycle basis, many of the features found in modern microprocessors. For this assignment, though, we will set-up the parameters of the simulator so that it will simulate a processor and memory hierarchy very similar to that of the MIPS R3000 that was studied in CSE378.

For this assignment, you should work in **2-people teams**. Recall that you will have to change partners for each assignment.

In Section, you will be shown how to use SimpleScalar. Instructions for this assignment will be posted on the Web (in particular, where to find the simulator, what command line to use, the input for the simulation etc.).

Your task consists of:

1. Use a configuration file to reflect a computer system with the following configuration:
   - Issue instructions *inorder*
   - The pipeline can fetch, decode, issue, execute, and commit *one* instruction per cycle, independently of the instruction type.
   - One of each type of functional unit.
   - The L1 data cache is 8KB with 32 bytes blocks. It is 2-way set associative and uses LRU as a replacement algorithm.
   - The L1 instruction cache is 16 KB with 64 bytes blocks. It is 2-way set associative and uses LRU as a replacement algorithm. (We make the L1 instruction cache somewhat "bigger" because SimpleScalar has 64-bit instructions. With these parameters the L1 instruction cache is similar to a MIPS R3000 8KB cache with 32 bytes blocks.)
   - The 256 KB L2 cache is unified, direct-mapped and has 64 bytes blocks.
   - The instruction TLB is 4-way set associative and has 64 entries; Replacement is LRU.
   - The data TLB is 8-way set associative and has 128 entries. Replacement is LRU.

  - The page size is 4KB.
  - Leave the remainder of the specifications (e.g., branch prediction, latencies of the various components of the memory hierarchy etc.) as in the default configuration file. You should also know that all caches use a write-back policy.

2. Run the simulation and report the following measurements:
   - Number of instructions committed (these are the "useful"instructions i.e., those that have gone through all stages of the pipeline including the write-back stage).
   - Number of misses in the instruction and data L1 caches.
   - Miss rates for the instruction and data L1 caches.
   - Number of misses in cache L2.
   - Miss rate for the L2 cache. (Use the "local" miss rate, i.e, the number of misses to L2 divided by the number of references to L2.)
   - CPI of the system.

3. Answer the following questions:

   (a) Why is the CPI not equal to 1?

   (b) Give an *estimate* of the CPI contributed by respectively the L1 instruction cache, the L1 data cache and the L2 cache. I am more interested in the reasoning than in the accuracy of your computations (in fact providing more than two significant digits is not worthwhile).

   List one other major contributor to the CPI (but don't calculate its contribution).

   (c) Why is there a difference between the number of misses and the number of write-backs for the L1 data cache?

You should **hand-out** paper copies of:

1. The configuration file used for the simulation.
2. The results of the measurements asked for in paragraph 2 above.
3. Answers to the 3 questions.