CSE 471 Autumn 2001

Computer Design and Organization

Assignment #5

Due: Wednesday December 5

In this assignment you'll have to be a little creative. Its goal is to help you gain some further understanding on the effects of various cache configuration parameters on performance, and in particular on cache hit rates.

You can do this assignment in groups of two under the same constraints as previously, i.e., with a new partner. Turn in a report with a structure similar to the one you did for Assignment #2 although clearly the methodology section should be longer and the results section shorter.

For this assignment you should use the compiler and simulators found in the directory **HW5**.

Your task is to write applications programs in C that enable you to discover the cache size, associativity, line (block) size, and write update policy of the **L1 data cache** being simulated in a special version of the SimpleScalar toolset. Design the programs so that each one (maybe with variations) determines a particular cache parameter. The order in which you "discover" these cache parameters might be important, so think about it while designing your experiments.

The cache size, associativity, and line size of the L1 data cache will all be powers of 2. The cache will be no larger than 64KB, the associativity will be no larger than 4-way, and the line size is a minimum of 4B and a maximum of 64B.

Since we only need to record cache statistics, we will use a simplified version of SimpleScalar, called *sim-cache* (for how to use *sim-cache* see Section 4.2 of the "The Simple Scalar Tool Set, version 2.0" on the SimpleScalar Web site). In fact, your applications should be tested on a modified version of *sim-cache* named *secret*. *secret* is such that the cache configuration parameters of interest have been set to particular values, but are not printed as part of the output. In other words, your assignment is to discover the cache parameters that are "hidden" in *secret*. Of course, you can use *sim-cache* to test/guide your experiments.

To compile your C programs into SimpleScalar code, use the *ss-gcc* compiler in the directory **HW5**. A typical sequence of compile and run with *secret* would be:

```
HW5/ss-gcc ~/test.c -o ~/test.out
HW5/secret ~/test.out >& ~/result
```

while with *sim-cache* it would be:

```
HW5/ss-gcc ~/test.c -o ~/test.out
HW5/sim-cache -cache:dl1 dl1:256:64:2:l  ~/test.out >& ~/result
```

(Note that the last character in dl1:256:64:2:l is an "el" not a "one".)

secret is read/write protected so you should not be able to read it. Knowing the great hacking ability of many of you, I am sure that you can break this protection. However, even if you do so you will have to "prove" that you have discovered the right configuration in your report via graphs/tables and your methodological reasoning. As a further claim to your honesty and inventiveness, attach a sample copy (source code) of the programs you wrote for discovering each of the configuration parameters.