# ECE550 Cache Simulator Testing - due 12/8/17

We will test our cache simulators for accuracy and performance.

The first test is a simple accuracy test on a short trace file of 10000 addresses. This is a good way to verify your basic cache structure before implementing multilevel caches.

Use a single, unified, direct-mapped L1 cache only, total size 8192 bytes, linesize=32, write-allocate. Using the 10000 entry file linked below, my reference simulator gives the following results:

| Metrics | Total | Instrn | Data | Read | Write | Misc |
|---|---|---|---|---|---|---|
| Demand Fetches | 10000 | 7637 | 2363 | 1027 | 1336 | 0 |
| Fraction of total | 1.0000 | 0.7637 | 0.2363 | 0.1027 | 0.1336 | 0.0000 |
| | | | | | | |
| Demand Misses | 493 | 251 | 242 | 48 | 194 | 0 |
| Demand miss rate | 0.0493 | 0.0329 | 0.1024 | 0.0467 | 0.1452 | 0.0000 |
| Compulsory misses | 417 | 216 | 201 | 22 | 179 | 0 |
| Capacity misses | 5 | 2 | 3 | 1 | 2 | 0 |
| Conflict misses | 71 | 33 | 38 | 25 | 13 | 0 |

In this output, "fetches" is the total number of addresses processed; you see that there are about 3/4 instruction references and 1/4 data references, with the various miss rates and types of misses given (classifying the misses like this is optional, but will earn you extra credit). Your output should be identical or extremely close to this for the same situation. Since this case is direct mapped, there is no variation from using, say, RND replacement policy.

You will probably want to make a local copy of the test file linked below.

1. Turn in your own output to this test case and a brief report assessing the accuracy of your simulator. Be prepared to run this test for your TAs come demo time.

Link to 10000 entry dinero-format trace file

Our next test is a performance test. Move on to this test once you are happy with the accuracy of your simulator as tested above. The objective is simply to measure the performance of your code as it stands.

Be sure to report what kind of computer and operating system you ran your tests on.

2. Run your program on the full length trace file (still a short one - only 7.5M references!) linked below, for the following case: 2 level cache: Level 1: Harvard, 16K I-cache, 16K D-cache, both direct mapped, both linesize=32, write-allocate. Level 2: Unified, 256K, 4-way set associative, LRU replacement, linesize=64, write-allocate. Report the cpu and wallclock time required for this case, and be sure to tell me which computer you used.

Include in your report a summary of your timing results as well as the hit/miss/access time data for each cache.

Link to full-length (but still short) dinero-format trace file

On demo day we may have additional trace files we want you to run.

.