California Institute of Technology
Department of Computer Science
Computer Architecture

CS184b, Winter 2000 Assignment 7: Caching Thursday, February 15

Due: Thursday, February 22, 5:00PM

Remember that the SimpleScalar technical report (`TR_1342.ps` in the directory) describes the various simulation tools including `sim-cache`, `sim-cheetah`, and `sim-outorder`.

The first one should be easy. Two and three just require you to run the simulators and extract numbers. Four is just applying the formulas. Problem 5 will be the hard (and most illustrative) one requiring you to organize your search over the design space, run simulations, and evaluate tradeoffs to find the best design point.

1. For a 1GHz processor issuing up to 4 instructions per cycle and with a 50ns main memory cycle time, give the equi-performance potential curve relating data cache miss-rate to cache latency (assuming you can deal with continuous variations in L1 cache delay); graph your curve. Access rate is 0.3 references per instruction. Curve should pass through the 1ns, 10% miss rate.

2. Use `sim-cheetah` on your application to quantify the miss rate differences among the following (use a 16KB cache as your base, 16B cache lines):

   - fully associative, optimal replacement
   - fully associative, LRU replacement
   - 2-way to 8-way set associative, optimal replacement
   - 2-way to 8-way set associative, LRU replacement
   - direct mapped
   - direct mapped twice the capacity

3. Use `sim-cache` to calculate miss rate under the random and FIFO replacement strategies for the 4-way set associative case above. Compare with the LRU and optimal replacement results which you obtained in the previous problem.

4. Summarize the Mulder/Quach/Flynn area for each of the (different) cache organizations above; that model gives no difference for replacement strategies, so are is per organization ignoring replacement differences.

5. Using your chosen application as the benchmark, and a 2-issue superscalar processor, if you are given 300K rbe of on-chip memory area, how should you organize the on-chip cache? (That is, one or two levels, unified or split instruction and data cache at each level? how much data at each level? blocking factor? associativity?). Use sim-outorder

and the Mulder/Quach/Flynn area model to establish your solution. Pick your favorite parameters for supporting the 2-issue based on your previous experience (that's not the focus of this exercise).

- Assume direct mapped cycles/access = $\lceil \log_2(\text{depth}/512) \rceil$.
- Assume set associative caches take the same time as a direct mapped which is twice as deep (that's effectively one cycle longer, except in the case where the cache is so small that the double size cache can be accessed in a single cycle).