Computer Science and Engineering 431
Computer Architecture
Fall 2014

Homework #3

Distributed: September 21, 2015
Due: October 1, 2015 via Angel by midnight
Points: 18

1. (3 pts) For the following repeating pattern (e.g., in a loop) of branch outcomes: T, T, T, NT, NT

   a. What is the accuracy of always-taken and always-not-taken predictors for this sequence of branch outcomes?

   b. What is the accuracy of a one-bit predictor assuming the predictor starts off in the predict-taken state. What is the accuracy of this predictor if this pattern is repeated forever?

   c. What is the accuracy of a two-bit saturating counter predictor assuming that the predictor starts off in the weakly predict-taken state? What is the accuracy of this predictor if this pattern is repeated forever?

2. (3 pts) Assume the dcache is initially filled with word address 0, 1, 2, …15 memory data (referenced by the datapath in that order). Below is the next sequence of data memory address references given as word addresses.

   1, 18, 19, 3, 4, 5, 18, 21, 5, 33, 32, 19

   a) Assuming a direct-mapped dcache with one-word blocks and a total size of 16 blocks, list if each reference is a hit or a miss. Show the state of the dcache after the last reference (e.g., as in slide Chapter 5B.29 from lecture). What is the hit rate for this reference string?

   b) Now, assuming a direct-mapped dcache with two-word blocks and a total size of 8 blocks, list if each reference is a hit or a miss. Show the state of the dcache after the last reference. What is the hit rate for this reference string?

   c) Now, assuming a set-associative dcache with two ways, two-word blocks and a 4 blocks per set, list if each reference is a hit or a miss (assume LRU replacement). Show the state of the dcache after the last reference. What is the hit rate for this reference string?

3. (3 pts) For the following questions, the icache has a hit rate of 94%, a hit latency of 1 clock cycle, and a miss penalty of 100 clock cycles (the miss

penalty includes both time to get the information from the main memory and the icache miss penalty together). The dcache has a hit rate of 85%, a 2 clock cycle hit latency, a miss penalty of 120 clock cycles (the miss penalty includes both time to get the information from the main memory and the dcache miss penalty together), and memory accesses (`lw` and `sw`) that are 40% of the instruction mix. The core has a 20psec clock.

a) Calculate the CPI taking into account stalls due to icache and dcache and misses broken down into based CPI plus additional CPI due to icache stalls and additional CPI due to dcache stalls. The base CPI using a "perfect" memory system is 1.0. Assume everything else is working perfectly - loads never stall a dependent instruction. Also assume the processor waits for stores to finish when they miss in the dcache, and that icache misses and dcache misses never occur at the same time.

b) What is the AMAT in clock cycles for instructions? For data?

c) What is the AMAT for data if you were to add a 1MB L2 cache with a 90% miss rate and a 10 clock cycle hit latency?

4. (9 pts) In this second set of SimpleScalar experiments you are going to further explore the memory hierarchy. Continue to use a fastforward of 500,000 and simulate 2,500,000 instructions for each benchmark (four integer and two floating point benchmarks).

   Baseline Datapath: For the baseline simulations configure a single-issue datapath as we did in HW#2. Use a memory bus width of 8 bytes (the default – note that the SimpleScalar machine has 64-bit words), a memory latency of 100 and 10 (100 cycle access latency for first word and 10 cycles for the rest), and a TLB latency of 30 cycles.

   Use the default setting for the memory hierarchy, i.e., il1:256:32:1:l (an 8KB L1 instruction cache with 256 blocks, 32 byte blocks (so 4 word blocks) and direct mapped; note that the last character is the letter l (not the number 1)), dl1:256:32:1:l, ul2:1024:64:4:l (a 256KB unified L2 cache with 64 byte blocks and 1024 blocks per way, 4-way set associative and LRU replacement), itlb:16:4096:4:l (a 64 entry (16 per way, 4-way set associative) iTLB with a page size of 4096 and LRU replacement), and dtlb:32:4096:4:l (a 128 entry (32 per way, 4-way set associative) dTLB with a page size of 4096 and LRU replacement), (see page 6 in *The SimpleScalar Tool Set* handout for more details). Use the cache speeds dl1lat = 1, dl2lat = 10, il1lat = 1, and il2lat = 10.

   Larger L2 Cache: Quadruple the size of the unified L2 cache and quadruple its latency (i.e., ul2:4096:64:4:l, dl2lat = 40 and il2lat = 40).

Larger L1 and L2 Caches: Keeping the quadrupled L2 cache, now double the size of the L1 caches and double their latencies. Keep both of the L1 caches direct mapped.

Set Associative L1 Data Cache: Keeping both the quadrupled L2 and the doubled L1 caches, make just the L1 data cache 4-way set associative. As a result, the latency of the L1 data cache is now tripled over the baseline (instead of just doubled). Run two sets of experiments, one using LRU replacement and one using random replacement for the L1 data cache.

For each set of experiments (there are a total of 5: Baseline, Larger L2, Larger L1 and L2, Set Associative L1 Data Cache LRU, and Set Associative L1 Data Cache random), report in tabulated form the sim_IPC, il1 cache hits and misses, dl1 cache hits and misses, and ul2 cache hits and misses. Compare the five different machines and provide an explanation for the changes in performance (sim_IPC) and hit/miss ratios.