

CS 573
Midterm Exam
March 9, 2007

The following exam is open book and open notes. You may feel free to use whatever additional reference material you wish, but **no electronic aids** are allowed. Please note the following instructions. There will be a ten point deduction for failure to comply with them:

- start each problem on a new sheet of paper
- write your Banner ID number, but not your name, on each sheet of paper you turn in
- show your work whenever appropriate. There can be no partial credit unless I see how you arrived at your answers
- be succinct. You may lose points for facts that, while true, are not relevant to the question at hand

You have until 11:20 to finish the exam. The questions are equally weighted.

1. You have a choice of making exactly one of the following changes to the microarchitecture from HW2.
 - (a) Double the size of the prefetch path: change the size of the prefetch buffer to 16 bytes, and assume you can fetch 8 bytes at a time. The instruction decoders are unchanged.
 - (b) Improve the decoders so you can always decode two instructions per cycle (rather than one in some cases and two in others, as in the HW).
 - (c) Double the number of μ -instruction slots in the decoded instruction buffer.
 - (d) Double the number of physical registers.
 - (e) Double the number of arithmetic pipelines.

In your estimation, which would show the greatest promise for improving the performance of that machine? In this question, your actual answer is almost completely unimportant: what matters is how you think about the problem and *why* you make the choice you do.

2. The VAX instruction set was no fun to decode: the first byte contained the opcode and the following (up to) three bytes each contained a register number and addressing mode specification. Then, any values (such as immediate operands, direct memory addresses, and indexes for indexed addressing) needed for the addressing modes came next.
But this wasn't as bad as it could have been: each register/mode byte could have been followed immediately by its data. Why would this have been even worse for instruction decoding than the actual VAX? Would it have affected the orthogonality, as seen by the programmer?
3. Suppose the program in HW 2 had executed its loop many more times than it actually did.
 - (a) What would be the asymptotic behavior of the branch predictor in the assignment? That is, over time would they converge on some good or bad behavior? If they did, what "ultimate" prediction accuracy would you expect from each of them?
 - (b) Suppose that instead of the branch predictor from the assignment, we use Smith's Strategy 2 (each branch instruction has a simple one-bit predictor so it always predicts every branch behaves the same way on this execution that it did on the last execution). What happens to the asymptotic behavior?
 - (c) Finally, suppose that instead of the predictor from the assignment, each branch instruction had a simple two-bit predictor using saturating counters (an idealized version of Smith's Strategy 7). Now what happens to the asymptotic behavior?