ENEE 759M Spring 2005

April 7, 3:45 P.M. - 4:45 P.M.

Open Book, Filtering Mind, Closed Communication

I have neither received aid nor given aid in this exam: _____

If your answer is unlegible, you will probably loose a lot of points. **If you perceive any ambiguity in any of the questions, state your assumptions explicitly.**

NOTE: All work on this exam is to be wholly your own. Consulting (copying or possessing) from other students' answers, or aiding other students (by verbal communication or by showing your answers) will be considered a violation of the academic honor code. Violations will result in a grade of F for the course.

Please don't keep your answers too brief or too long. Make sure that all relevant points are covered.

1. What do you see as the major hardware trends that affect processor design in the next decade? Indicate how these trends affect processor design.

2. A skeptic in the 759M class is not convinced that microarchitectural techniques provide any noteworthy benefit in power consumption. His viewpoint is that power consumption could be easily reduced by reducing the clock frequency or supply voltage. How would you convince him that microarchitectural techniques are very helpful? (Assume that you are not that skeptic!)

3. What are the potential drawbacks of dynamic IPC/clock rate optimization? What kind of code will benefit from each end of the IPC/clock rate spectrum (one end is high IPC-low clock rate, and the other end is low IPC-high clock rate)?

4. Explain clearly why a CMP processor might provide a faster clock rate than an SMT processor? Explain clearly why the SMT approach might give better performance, for a small number of PEs.

5. What are the major differences between the CMP approach and the SMT approach? Which, in your opinion, is likely to perform better? Justify your answer.

6. What is meant by a *critical instruction*? What are the potential reasons for an instruction to become critical? What are some of the ways in which we can exploit instruction criticality?

7. One enthusiastic designer says that helper thread (as we know it) is an overdesign, and its effect can be achieved by temporarily suspending the main thread, and letting the helper thread run in its hardware context. What do you see as the potential advantages/disadvantages of this technique?