LSU EE 7700-2 **Homework 2** **Due: 10 November 2003**

The first questions are based on the prefetch survey paper, Steven P. Vanderwiel and David J. Lilja, "Data prefetch mechanisms," *ACM Computing Surveys,* vol. 32, no. 2, June 2000, The paper can be found at `http://www.ece.lsu.edu/tca//papers/vanderwiel-00.pdf`.

The other questions are based on the Roth framework paper, Amir Roth and Gurindar Sohi, "A quantitative framework for automated pre-execution thread selection," in *Proc. of the 35th Annual International Symposium on Microarchitecture*, pp. 430-441, 2002. Authorized users can find the paper at `http://www.ece.lsu.edu/tca//s/roth-02.pdf`.

Problem 1: According to the work surveyed in Vanderwiel 2000, how do the performance of tagged prefetch and sequential prefetch compare?

Problem 2: A *prefetch buffer* is an additional cache reserved for data fetched by prefetch instructions. Let system P have prefetch, 64 kiB of cache, but no prefetch buffer (prefetched data goes in the regular cache). Let system PB have prefetch, 48 kiB of cache and a 16 kiB prefetch buffer. The prefetch buffer and ordinary cache have the same hit latency.

(*a*) Under what circumstances would PB be faster? What is the rationale for the prefetch buffer?

(*b*) Under what circumstances would P be faster? What is an argument against a prefetch buffer?

Problem 3: In Roth 02 a p-thread is triggered whenever its trigger instruction is reached. Consider a pre-execution scheme in which a p-thread (or more practically, a p-thread id) is retrieved using a predictor that examines the address of a potential trigger instruction and a global history of branch outcomes (or some other kind of history). Thus a trigger instruction on one occasion might trigger p-thread A and on another occasion might trigger p-thread B.

(*a*) Using an example, show how this might outperform the type of triggering described in Roth 02.

(*b*) Modify the aggregate advantage equation in Roth 02 to account for the predictor.

Problem 4: Execution of p-threads slows the main thread in places (say, before a troublesome load) in addition to speeding it (when a cache miss is avoided). In the paper the amount of slowing was measured two ways. What are the two ways? Why was slowing measured in two ways, why not just measure it one way?