

The following questions refer to the paper Avinash Sodani and Gurindar S. Sohi, "Dynamic Instruction Reuse," International Symposium on Computer Architecture, pp. 194-205, 1997.

Problem 1: Consider the configuration used to evaluate DIR, described in Table 1 (page 200) of the paper.

- (a) The branch predictor used was a plain-vanilla (that is, ordinary) bimodal predictor. How might a better branch predictor have affected the results?
- (b) The single-level cache has a 6-cycle miss latency. How might a longer miss latency, say 100 cycles, affected the results?

Problem 2: Both value prediction and DIR (dynamic instruction re-use) determine instruction results. Consider a last value predictor that predicts an instruction result (the value written to the destination register) will be the same as the last two results it produced if the last two results were the same. If the last two results were different then it will make no prediction.

- (a) Describe a situation in which the last value predictor can predict an instruction result more frequently than DIR scheme S_n . Show a code example. Assume comparable table sizes, etc. *Hint: Easy.*
- (b) Describe a situation in which DIR scheme S_n can predict an instruction result more frequently than the last value predictor. Show a code example. Assume comparable table sizes, etc.

Problem 3: Suppose a value predictor gave correct predictions more frequently than DIR (any scheme) determined instructions results. (I'm not saying it does, just suppose that it does.) Also assume the two have about equal cost implementation.

- (a) Why might DIR still be better?
- (b) For which processor configurations would DIR have the largest advantage over last value prediction?

Problem 4: *This problem is optional, I'm not sure how long it will take.* Using PSE, find a section of code that can make good use of S_{n+d} DIR in the following execution of gcc:
/fac/drk/pub/ds/2005/vpc/ds_128854_gcc_ROB_256_tbl_10_conf_0.ds