EE 4720 **Homework 6** **Due: Not Collected**

If you only have time for one of these problems, do problem three (the one on connecting memory devices to implement a cache). If you have or are hoping to get a job interview with a company that makes processors or which requires strong familiarity with them, do problems one and two. If you want to practice Verilog or VHDL do problem four. It you had the time to read this far you have no excuse for not doing at least one of the problems.

Base answers to the problems below on information in the white paper at `http://www.cpus.hp.com/techreports/PA-8700wp.pdf`. *This is a very concise technical summary of the distinctive features of the Hewlett Packard PA-8700 processor, which implements the Precision Architecture (PA). It was written to sell processors, so though technically informative certain adjectives must be taken in context. Familiarity with the PA ISA is not needed to solve this assignment. If nevertheless you're curious, there is a brief description of the PA ISA (along with others) in Appendix C of the text and a complete description can be found at* `http://www.hp.com/ahp/framed/technology/micropro/architecture/docs/instarch.html`.

Problem 1: Describe how dynamic scheduling on the Hewlett Packard PA-8700 is similar to and different from dynamic scheduling methods 1, 2, and 3 presented in class. The description should include handling of register values and how exception recovery might be performed. The description of dynamic scheduling in the white paper is very brief, get what you can from the description and the figure and make reasonable guesses at the rest. There is additional information on dynamic scheduling at the top of page 12.

Problem 2: How does the dynamic branch prediction performed by the PA-8700 differ from the one-level (2-bit saturating counter) scheme presented in class? How might a compiler (that knows the microarchitecture of the target) use this difference to eliminate the performance penalty of some collisions? Would this work for all collisions?

The problems below are not based on the white paper.

Problem 3: Show how memory devices using 12-bit addresses can be connected to implement a 2^{17}-byte two-way set-associative cache. The memory devices can store 2^{12} items of x bits each. Each memory can have its own width (x), (but of course, all the items in a particular memory are the same width). The system has an address space of $a = 32$ bits, a bus width of $w = 8$ bytes, and a block size of $L = 2^l = 2^4 = 16$ bytes. The character size is one byte. Show which address bits are used to index the memories and which bits are used to determine a hit. Show only the parts used to read data on a cache hit. (This problem would be easier if memory devices with any address size could be used. If your stuck, try that first.)

For students in 4702 or others who know Verilog, VHDL, or some other HDL.

Problem 4: Write a Verilog or VHDL description of a module that will determine which is the least-recently used block in a four-way set-associative cache cache set. This is to be based on information kept in a special memory (like a tag store, but there's only one of them for all the ways). The module will read this information at its inputs. It will generate an updated version of this information at its output for storage. It will also generate an output indicating which set is least recently used. There will be a clock input, inputs to the module must be read on the positive edge. You may specify other inputs that the module might need. (For example, the memory operation.)