ENGN 2910A Homework 07 (50 points) – Due Date: Nov 14th 2013

Professor: Sherief Reda

School of Engineering, Brown University

1. (25 points)

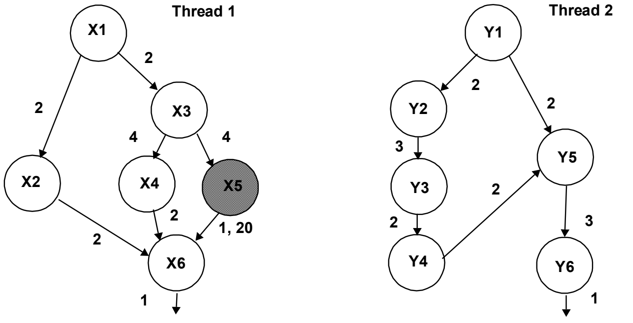
   a. Consider a simple 5-stage pipeline that is single-threaded. The pipeline treats every cache miss as a hazard and freezes the pipeline. While executing a benchmark assume that a L1 cache miss occurs every 100 cycles, and that each L1 cache miss takes 10 cycles to satisfy if the block is found in L2 or 50 cycles if L2 misses as well. A L2 cache miss occurs after 200 cycles of computation. Assume that the CPI in the absence of the cache misses is 1. What is the actual CPI, taking into account the cache miss latencies?

   b. Consider the same example as in (a.), but assume that hardware is block multi-threaded with support for two HW threads (similar to slide 9 in lecture). Assume that switching overhead is zero, and that there are two threads with identical cache miss behavior as in in part (a.). What is the CPI of each of the two programs on the two-way multi-threaded machine? Did the CPI improve? If yes, explain how. If not, explain why one should bother with the two-way multi-threaded machine.

   c. Consider the case in (a), but the switching overhead is five cycles. Again compute the CPI of each thread, and explain why it increases, decreases, or stays the same.

   d. Consider the case for which the L2 miss latency jumps from 50 to 500 cycles and the switching overhead jumps from 5 to 50 cycles. Compare the CPI in this machine.

2. (25 points) Assume the example we taken in the class with the given data flow graph except but we will now assume X5 is a cache hit and hence it always takes one cycle to complete. For the following questions, please use the same assumption about for the superscalar OoO organization as in class.



a. Redo the speculative scheduling using interleaved multi-threading and block multi-threading. Note that when C5 is a cache hit, block multi-threading is identical to a single-threaded core, where X executes first and then Y executes.

b. How much faster (in clock cycles) is interleaved multi-threading over block multi-threading in this case?

c. Interleaved multi-threading required more hardware support for selective flossing and using thread ID for tag matching dependencies. Due to the design complexity, the clock cycle of the inter-leaned multi-threading processor is 20% slower. Does it still make sense to use interleaved multi-threading for the modified threaded core where X5 is a cache hit?