EXERCISES

Problem 8.1

A CPU designer has to decide on whether or not to add a new microarchitecture enhancement to improve performance (ignoring power costs) of a block (coarse-grain) multithreaded processor. In this processor a thread switch occurs only on a L2 cache miss. The cost of a thread switch is 60 cycles (time before a new thread can start executing). Assume that there are always enough ready threads to switch to on a cache miss. Also, it is given that the current L2 cache hit rate is 50%. The new microarchitectural block is a cache hit/miss predictor. The new predictor predicts whether a memory reference is going to hit or miss in L2 (note not L1) cache. The predictor is used to decide when to switch threads. If the predictor predicts a cache miss thread switching is initiated early. There are four scenarios to consider:

(a) The predictor predicts a L2 cache miss and the true outcome is also a L2 cache miss. In this case thread switching is initiated early and the thread switching cost is reduced to 20 cycles (from 60 cycles in the baseline).

(b) The predictor predicts a L2 cache miss and the true outcome is a L2 cache hit. In this case an unnecessary thread switch has been initiated which increases the thread switching overhead to 120 cycles due to unnecessary pipeline flushes.

(c) The predictor predicts a L2 cache hit and the true outcome is also a L2 cache hit. In this case no thread switching is initiated and there is no gain or loss.

(d) The predictor predicts a L2 cache hit and the true outcome is also a L2 cache miss. This is a case of lost opportunity for an early thread switch and the machine pays the 60 cycle baseline switching penalty.

Given these four scenarios what should be the predictor accuracy before the designer can be certain that this new microarchitectural block leads to a break-even point in performance. If the L2 cache hit rate of the base machine is improved from 50% to 80% how does that impact predictor's accuracy requirements before achieving break-even point in performance?

Problem 8.2

For this problem we use the dependence graph shown in Figure 8.4 with the following modification. First, we assume that X5 is a cache hit and hence takes only one cycle to compute always.

(a) Recalculate the speculative scheduling using interleaved multithreading and block multithreading (similar to Table 8.1 and all assumptions remain the same as what was used for generating Table 8.1). Note that when X5 is a cache hit block multithreading is identical to a single-threaded core where X executes first and then Y executes.

(b) How much faster (in clock cycles) is interleaved multithreading over block multithreading in this case?

(c) Interleaved multithreading requires more hardware support for selective flushing and using thread id for tag matching dependencies. Due to the design complexity if the clock cycle of interleaved multithreading processor is 20% slower. Does it still makes sense to use interleaved multithreading for the modified threaded code in Figure 8.4 where X5 is a cache hit?