ECE/CS 757: Homework 1

Cores and Multithreading

1. A CPU designer has to decide whether or not to add a new micoarchitecture enhancement to improve performance (ignoring power costs) of a block (coarse-grain) multi-threaded 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 as follows:
   (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 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 the predictor's accuracy requirements before achieving a break-even point in performance?

2. Consider a future 16-way CMP operating in a power-constrained environment. The CMP can automatically configure itself to run as a 1, 2, 4, 8, 16-way core CMP but always using a fixed power budget. For instance, it can run as a single-core processor by grabbing the power from the other 15 cores by putting them to sleep and using the additional power to increase its frequency. Assume that sleep and wakeup times are zero, and that power and frequency have a square relationship. For instance, if one core uses the power of all 16 cores, its frequency can increase four-fold. We call this an EPI-throttled CMP.

   Consider a partially parallel application. The application starts as a single-threaded application and spends 5% of the time in sequential mode. During the following 40% of the time, the application has 16 threads and only four threads for the next following 40% of the execution time. During the remaining execution time, the application has only one thread.

      (a) What is the speedup of this application when it runs on this future CMP compared to running on a single-core machine that uses the same power but operates at a higher frequency using the square relationship?

(b) What is the speedup of this application when it runs on this future CMP compared to running on a traditional 16-way CMP (again using the same power budget) that does not provide the reconfiguration capability?

(c) Due to limitations of voltage-scaling, assume that in the future power depends linearly on frequency. In this case comment on what benefits (if any) an EPI-throttled CMP provides.

3. (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 assumes 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 cache misses is 1. What is the actual CPI, taking into account cache miss latencies?

   (b) Consider the same example as in (a), but assume that hardware is now 2-way multi-threaded. Assume that switching overhead is zero and that there are two threads with identical cache miss behavior as described in the first case. What is the CPI of each of the two programs on the 2-way multi-threaded machine? Did the CPI improve? If yes, explain how. If not, explain why one should bother with the 2-way multi-threaded machine.

   (c) Consider the case as in (a), but the switching overhead is 5 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. Compute the CPI in this machine.

4. The combination of two enhancements is considered to boost the performance of a chip multiprocessor. The enhancements are: (1) adding more cores or (2) adding more shared L2 cache. The base chip has 3 cores and 9 L2 cache banks. The L2 cache size can be incrementally increased by adding cache banks and each cache bank uses 3 times the area of a core. Here is what we also know from all kinds of sources:
   (a) 60% of the workload can be fully parallelized and the rest cannot
   (b) The core stall time due to L2 misses accounts for 30% of each core's execution time in the base configuration with 4 cache banks and 4 cores
   (c) It is suspected that the amount of shared L2 cache per core should remain constant in order to keep the same miss rate
   (d) Simulations have also determined that the miss rate of L2 decreases as the square root of its size per core. A conjecture is that the stall time in each core will also decrease as the square root of L2 size per cores.

   The company that pays your paycheck has acquired a new technology to build large microchips, so that the next generation chips have four times the area of current chips to dedicate to cores and L2 caches. Given what you know, what kind of best "first cut" design would you propose? A design is characterized by the number of cores and the number of L2 cache banks. These numbers

can be any integer. The design should be contained in the new chip. Estimate the speedup of your best design that takes advantage of the new chip real estate.

MP Memory Systems / Coherence

5. In the design exploration of a new chip multiprocessor (CMP) system an important design decision is the choice of cache organization. Assume that a CMP has 4 processor cores. Our options are between using a private or a shared, first-level cache organization, where both of them consume about the same chip resources. The detailed assumptions for each cache organization are given below. The block size for both configurations is 16 bytes.
   Private cache organization: Each private cache contains 8 block entries, is direct-mapped and the cache hit time is 1 cycle. Cache coherence across the private caches is maintained using a simple protocol. The time to carry out a snooping action is 10 cycles. If a block has to be retrieved from memory it takes 100 cycles.
   Shared cache organization: The shared cache organization has as many block entries as the total number of block entries in the private caches. Further, it is partitioned into as many banks as the number of processors and the mapping of memory blocks to banks is round-robin; block address *i* is mapped to bank *i* modulo B, where B is the number of banks. The banks are accessed by the processors using a B x B cross-bar switch. To access a cache bank takes 2 cycles if there is no contention. If a block has to be retrieved from memory it takes 100 cycles.
   a) Determine the average memory-access time for each of the organizations in the case when a *single* processor sequentially accesses memory blocks 0 up to 15 twice in a row. Which organization yields the shortest memory access time and by how much?
   b) Determine the average memory-access time for each of the organizations in the case when *each* processor sequentially accesses memory blocks 0 up to 15 twice in a row. Ignore contention effects. Which organization yields the shortest memory access time and by how much?
   c) Assume that memory blocks 0 to 7 are initially present in both cache organizations. Now assume that processor 1 modifies all blocks and processor 2 reads them subsequently. Ignore contention effects. Which organization yields the shortest memory access time and by how much?

6. Consider a shared-memory multiprocessor that consists of 3 processor/cache units and where cache coherence is maintained by an MSI protocol. The private caches are direct-mapped. The following table shows the access sequence taken by 3 processors to 4 variables (A, B, C and D), where A, B and C belong to the same block and D belongs to a different block. The two blocks map to the same entry in the caches and the cache is full initially.

	Processor 1	Processor 2	Processor 3
1	R_A		
2		R_B	
3			R_C
4	W_A		
5			R_D
6		R_B	
7	W_B		

8			R_C
9		R_B	

a) Classify the misses with respect to cold, replacement, true sharing and false sharing misses.
b) Which of the misses could be ignored and still guarantee that the execution is correct?

7. Assume that a shared-memory multiprocessor using private caches connected to a shared bus uses an MSI cache protocol to maintain cache coherence. The time it takes to carry out various protocol actions is listed in the following table. While a read and write hit take only a single cycle, a read request takes 40 cycles as it has to bring the block from the next level of the cache hierarchy. A bus upgrade request takes less time as it does not involve a block transfer but rather invalidates other shared copies. This action consists of transferring the request on the bus and making a snoop action in each cache; the time for the latter is also shown in the following table.
Determine for each of the cases below how long it takes to carry out the following sequence of reads and writes to blocks X and Y, where the notation R_i / B and W_i / B means a read and write operation respectively by processor/cache unit *i* to block B:

R1/X, R2/X, R3/Y, R4/X, W1/X, R2/X, R3/Y, R4/X.

a) Determine how long it takes to carry out all memory requests under the assumption that snoop actions get a higher priority than processor read/write requests from that same unit; they have to wait until the snoop action is done. The tag directory is not duplicated.
b) Determine how long it takes to carry out the memory requests from each individual processor under the assumption that we duplicate the tag directory to allow concurrency between inbound snoop actions and outgoing processor read/write generated protocol actions. In this case, concurrent tag lookups are only possible when the state of the block in the cache does not change as a result of the snoop action.

Timing and traffic parameters for protocol actions

B is the block size

Request type	Time to carry out protocol action	Traffic
Read hit	1 cycle	N/A
Write hit	1 cycle	N/A
Read request serviced by next level	40 cycles	6 bytes + B
Read request serviced by private cache	20 cycles	6 bytes + B
Read-exclusive request serviced by next level	40 cycles	6 bytes + B
Read-exclusive request serviced by private cache	20 cycles	6 bytes + B
Bus upgrade/update request	10 cycles	10 bytes
Ownership request	10 cycles	6 bytes
Snoop action	5 cycles	N/A

8. Assume a shared-memory multiprocessor with a number of processor/private cache units connected by a shared single-transaction bus. Our baseline cache coherence protocol is an MSI protocol, but we want to see what performance gain can be achieved by adding an exclusive state to make it a MESI protocol. We want to determine the time it takes to execute a sequence of accesses with the same assumptions and notations as in Q7 with an MSI and with a MESI protocol. Consider the following sequence of accesses by the processors:

   R1/X, W1/X, W1/X, R2/X, W2/X, W2/X, R3/X, W3/X, W3/X, R4/X, W4/X, W4/X.

   a) Now suppose that a transition from state E to state M brings no access cost. How many cycles does it take to execute the access sequence under MSI vs. MESI, assuming the access costs for the protocol transactions to be as in the Table given in Q7.
   b) Compare the traffic generated by the MSI and MESI protocol counted in bytes transferred using the data in the Table given in Q7 and assuming that B is 32 bytes.