ECE 452: Computer Organization and Design
Spring 2017

Homework 5: The Memory Hierarchy

Assigned: 17 Apr 2017
Due: 24 Apr 2017

Instructions:

- Please submit your assignment solutions via Canvas in a word or pdf file.
- Some questions might not have a clearly correct or wrong answer. In such cases, grading is based on your arguments and reasoning for arriving at a solution.

Q1 (35 points) In this question we look at memory locality properties of matrix computation. The following code is written in C, where elements within the same row are stored contiguously. Assume each word is a 32-bit integer.

```
for (I=0; I<8; I++)
  for (J=0; J<8000; J++)
    A[I][J]=B[I][0]+A[J][I];
```

a. **(5 points)** How many 32-bit integers can be stored in a 16-byte cache block?
b. **(5 points)** References to which variables exhibit temporal locality?
c. **(5 points)** References to which variables exhibit spatial locality?

Locality is affected by both the reference order and data layout. The same computation can also be written below in Matlab, which differs from C by storing matrix elements within the same column contiguously in memory.

```
for I=1:8
  for J=1:8000
    A(I,J)=B(I,0)+A(J,I);
  end
end
```

d. **(10 points)** How many 16-byte cache blocks are needed to store all 32-bit matrix elements being referenced?
e. **(5 points)** References to which variables exhibit temporal locality?
f. **(5 points)** References to which variables exhibit spatial locality?

Q2 (55 points) For a direct-mapped cache design with a 32-bit address, the following bits of the address are used to access the cache.

Tag	Index	Offset
31–10	9–5	4–0

a. **(5 points)** What is the cache block size (in words)?
b. **(5 points)** How many entries does the cache have?
c. **(5 points)** What is the ratio between total bits required for such a cache implementation over the data storage bits?

Starting from power on, the following byte-addressed cache references are recorded.

Address											
0	4	16	132	232	160	1024	30	140	3100	180	2180

d. **(10 points)** How many blocks are replaced?
e. **(10 points)** What is the hit ratio?
f. **(20 points)** List the final state of the cache, with each valid entry represented as a record of <index, tag, data>.

Q3 (40 points) In this question, we will look at the different ways capacity affects overall performance. In general, cache access time is proportional to capacity. Assume that main memory accesses take 70 ns and that memory accesses are 36% of all instructions. The following table shows data for L1 caches attached to each of two processors, P1 and P2.

	L1 Size	L1 Miss Rate	L1 Hit Time
P1	2 KiB	8.0%	0.66 ns
P2	4 KiB	6.0%	0.90 ns

a. **(5 points)** Assuming that the L1 hit time determines the cycle times for P1 and P2, what are their respective clock rates?
b. **(5 points)** What is the Average Memory Access Time for P1 and P2?
c. **(5 points)** Assuming a base CPI of 1.0 without any memory stalls, what is the total CPI for P1 and P2? Which processor is faster?

For the next three problems, we will consider the addition of an L2 cache to P1 to presumably make up for its limited L1 cache capacity. Use the L1 cache capacities and hit times from the previous table when solving these problems. The L2 miss rate indicated is its local miss rate.

L2 Size	L2 Miss Rate	L2 Hit Time
1 MiB	95%	5.62 ns

d. **(10 points)** What is the AMAT for P1 with the addition of an L2 cache? Is the AMAT better or worse with the L2 cache?
e. **(5 points)** Assuming a base CPI of 1.0 without any memory stalls, what is the total CPI for P1 with the addition of an L2 cache?
f. **(10 points)** Which processor is faster, now that P1 has an L2 cache? If P1 is faster, what miss rate would P2 need in its L1 cache to match P1's performance? If P2 is faster, what miss rate would P1 need in its L1 cache to match P2's performance?

Q4 (45 points) In this question, we will examine how replacement policies impact miss rate. Assume a 2-way set associative cache with 4 blocks. To solve the problems in this exercise, you may find it helpful to draw a table like the one below, as demonstrated for the address sequence "0, 1, 2, 3, 4."

Address of Memory Block Accessed	Hit or Miss	Evicted Block	Contents of Cache Blocks After Reference			
			Set 0	Set 0	Set 1	Set 1
0	Miss		Mem[0]			
1	Miss		Mem[0]		Mem[1]	
2	Miss		Mem[0]	Mem[2]	Mem[1]	
3	Miss		Mem[0]	Mem[2]	Mem[1]	Mem[3]
4	Miss	0	Mem[4]	Mem[2]	Mem[1]	Mem[3]
...						

Consider the following address sequence: 0, 2, 4, 8, 10, 12, 14, 16, 0

a. **(5 points)** Assuming an LRU replacement policy, how many hits does this address sequence exhibit?
b. **(5 points)** Assuming an MRU (most recently used) replacement policy, how many hits does this address sequence exhibit?
c. **(5 points)** Simulate a random replacement policy by flipping a coin. For example, "heads" means to evict the first block in a set and "tails" means to evict the second block in a set. How many hits does this address sequence exhibit?
d. **(10 points)** Which address should be evicted at each replacement to maximize the number of hits? How many hits does this address sequence exhibit if you follow this "optimal" policy?
e. **(10 points)** Describe why it is difficult to implement a cache replacement policy that is optimal for all address sequences.
f. **(10 points)** Assume you could make a decision upon each memory reference whether or not you want the requested address to be cached. What impact could this have on miss rate?

Q5 (35 points) In this exercise, we will examine paging and virtual memory systems. The following list provides parameters of a virtual memory system.

Virtual Address (bits)	Physical DRAM Installed	Page Size	PTE Size (byte)
43	16 GiB	4 KiB	4

a. **(10 points)** For a single-level page table, how many page table entries (PTEs) are needed? How much physical memory is needed for storing the page table?

There are several parameters that impact the overall size of the page table. Listed below are key page table parameters.

Virtual Address Size	Page Size	Page Table Entry Size
32 bits	8 KiB	4 bytes

b. **(10 points)** Given the parameters shown above, calculate the total page table size for a system running 5 applications that utilize half of the memory available.

The following table shows the contents of a 4-entry TLB.

Entry-ID	Valid	VA Page	Modified	Protection	PA Page
1	1	140	1	RW	30
2	0	40	0	RX	34
3	1	200	1	RO	32
4	1	280	0	RW	31

c. **(5 points)** Under what scenarios would entry 2's valid bit be set to zero?
d. **(5 points)** What happens when an instruction writes to VA page 30? When would a software managed TLB be faster than a hardware managed TLB?
e. **(5 points)** What happens when an instruction writes to VA page 200?