Chapter 5 Exercises

2013-12-17/410 Points

Pls. mail your homework (converted into PDF File) to coa.2012.assignment@gmail.com.

File name: ID_YourName_HW_Sequence.pdf, ex: **5XXXXXXX_Obama_HW_01.pdf**（You could save the doc/docx as PDF in word 2007 or later, using ‘Save as…’）

Exercise 5.3

Exercise 5.3

Caches are important to providing a high performance memory hierarchy to processors. Below is a list of 32-bit memory address references, given as word addresses.

a.	1, 134, 212, 1, 135, 213, 162, 161, 2, 44, 41, 221
b.	6, 214, 175, 214, 6, 84, 65, 174, 64, 105, 85, 215

5.3.1 [10] <5.2> For each of these references, identify the binary address, the tag, and the index given a direct-mapped cache with 16 one-word blocks. Also list if each reference is a hit or a miss, assuming the cache is initially empty.

5.3.2 [10] <5.2> For each of these references, identify the binary address, the tag, and the index given a direct-mapped cache with two-word blocks and a total size of eight blocks. Also list if each reference is a hit or a miss, assuming the cache is initially empty.

5.3.3 [20] <5.2, 5.3> You are asked to optimize a cache design for the given references. There are three direct-mapped cache designs possible, all with a total of eight words of data: C1 has one-word blocks, C2 has two-word blocks, and C3 has four-word blocks. In terms of miss rate, which cache design is the best? If the miss stall time is 25 cycles, and C1 has an access time of 2 cycles, C2 takes 3 cycles, and C3 takes 5 cycles, which is the best cache design?

There are many different design parameters that are important to a cache's overall performance. The table below lists parameters for different direct-mapped cache designs.

	Cache data size	Cache block size	Cache access time
a.	64 KB	1 word	1 cycle
b.	64 KB	2 word	2 cycle

5.3.4 [15] <5.2> Calculate the total number of bits required for the cache listed in the table, assuming a 32-bit address. Given that total size, find the total size

of the closest direct-mapped cache with 16-word blocks of equal size or greater. Explain why the second cache, despite its larger data size, might provide slower performance than the first cache.

5.3.5 [20] <5.2, 5.3> Generate a series of read requests that have a lower miss rate on a 2 KB two-way set associative cache than the cache listed in the table. Identify one possible solution that would make the cache listed in the table have an equal or lower miss rate than the 2 KB cache. Discuss the advantages and disadvantages of such a solution.

5.3.6 [15] <5.2> The formula shown on page 457 shows the typical method to index a direct-mapped cache, specifically (Block address) modulo (Number of blocks in the cache). Assuming a 32-bit address and 1024 blocks in the cache, consider a different indexing function, specifically (Block address[31:27] XOR Block address[26:22]). Is it possible to use this to index a direct-mapped cache? If so, explain why and discuss any changes that might need to be made to the cache. If it is not possible, explain why.

Exercise 5.4

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

	Tag	Index	Offset
a.	31–10	9–4	3–0
b.	31–12	11–15	4–0

5.4.1 [5] <5.2> What is the cache line size (in words)?

5.4.2 [5] <5.2> How many entries does the cache have?

5.4.3 [5] <5.2> 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

5.4.4 [10] <5.2> How many blocks are replaced?

5.4.5 [10] <5.2> What is the hit ratio?

5.4.6 [20] <5.2> List the final state of the cache, with each valid entry represented as a record of <index, tag, data>.

Exercise 5.7

In this exercise, 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
a.	P1	1 KB	11.4%	0.62 ns
	P2	2 KB	8.0%	0.66 ns
b.	P1	8 KB	4.3%	0.96 ns
	P2	16 KB	3.4%	1.08 ns

5.7.1 [5] <5.3> Assuming that the L1 hit time determines the cycle times for P1 and P2, what are their respective clock rates?

5.7.2 [5] <5.3> What is the AMAT for each of P1 and P2?

5.7.3 [5] <5.3> Assuming a base CPI of 1.0, what is the total CPI for each of 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
a.	512 KB	98%	3.22 ns
b.	4 MB	73%	11.48 ns

5.7.4 [10] <5.3> What is the AMAT for P1 with the addition of an L2 cache? Is the AMAT better or worse with the L2 cache?

5.7.5 [5] <5.3> Assuming a base CPI of 1.0, what is the total CPI for P1 with the addition of an L2 cache?

5.7.6 [10] <5.3> 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?

Exercise 5.8

Exercise 5.8

This exercise examines the impact of different cache designs, specifically comparing associative caches to the direct-mapped caches from Section 5.2. For these exercises, refer to the table of address streams shown in Exercise 5.3.

5.8.1 [10] <5.3> Using the references from Exercise 5.3, show the final cache contents for a three-way set-associative cache with two-word blocks and a total size of 24 words. Use LRU replacement. For each reference identify the index bits, the tag bits, the block offset bits, and if it is a hit or a miss.

5.8.2 [10] <5.3> Using the references from Exercise 5.3, show the final cache contents for a fully associative cache with one-word blocks and a total size of eight words. Use LRU replacement. For each reference identify the index bits, the tag bits, and if it is a hit or a miss.

5.8.3 [15] <5.3> Using the references from Exercise 5.3, what is the miss rate for a fully associative cache with two-word blocks and a total size of eight words, using LRU replacement? What is the miss rate using MRU (most recently used) replacement? Finally what is the best possible miss rate for this cache, given any replacement policy?

Multilevel caching is an important technique to overcome the limited amount of space that a first level cache can provide while still maintaining its speed. Consider a processor with the following parameters:

	Base CPI, no memory stalls	Processor speed	Main memory access time	First-level cache miss rate per instruction	Second-level cache, direct-mapped speed	Global miss rate with second-level cache, direct-mapped	Second-level cache, eight-way set associative speed	Global miss rate with second-level cache, eight-way set associative
a.	2.0	3 GHz	125 ns	5%	15 cycles	3.0%	25 cycles	1.8%
b.	2.0	1 GHz	100 ns	4%	10 cycles	4.0%	20 cycles	1.6%

5.8.4 [10] <5.3> Calculate the CPI for the processor in the table using: 1) only a first-level cache, 2) a second-level direct-mapped cache, and 3) a second-level eight-way set-associative cache. How do these numbers change if main memory access time is doubled? If it is cut in half?

5.8.5 [10] <5.3> It is possible to have an even greater cache hierarchy than two levels. Given the processor above with a second-level, direct-mapped cache, a designer wants to add a third-level cache that takes 50 cycles to access and will reduce the global miss rate to 1.3%. Would this provide better performance? In general, what are the advantages and disadvantages of adding a third-level cache?

5.8.6 [20] <5.3> In older processors such as the Intel Pentium or Alpha 21264, the second level of cache was external (located on a different chip) from the main processor and the first-level cache. While this allowed for large second-level caches, the latency to access the cache was much higher, and the bandwidth was typically lower because the second-level cache ran at a lower frequency. Assume a 512 KB off-chip second-level cache has a global miss rate of 4%. If each additional 512 KB of cache lowered global miss rates by 0.7%, and the cache had a total access time of 50 cycles, how big would the cache have to be to match the performance of the second-level direct-mapped cache listed in the table? Of the eight-way set-associative cache?

Exercise 5.10

Exercise 5.10

As described in Section 5.4, virtual memory uses a page table to track the mapping of virtual addresses to physical addresses. This exercise shows how this table must be updated as addresses are accessed. The following table is a stream of virtual addresses as seen on a system. Assume 4 KB pages, a four-entry fully associative TLB, and true LRU replacement. If pages must be brought in from disk, increment the next largest page number.

a.	4095, 31272, 15789, 15000, 7193, 4096, 8912
b.	9452, 30964, 19136, 46502, 38110, 16653, 48480

TLB

Valid	Tag	Physical Page Number
1	11	12
1	7	4
1	3	6
0	4	9

Page table

Valid	Physical page or in disk
1	5
0	Disk
0	Disk
1	6
1	9
1	11
0	Disk
1	4
0	Disk
0	Disk
1	3
1	12

5.10.1 [10] <5.4> Given the address stream in the table, and the shown initial state of the TLB and page table, show the final state of the system. Also list for each reference if it is a hit in the TLB, a hit in the page table, or a page fault.

5.10.2 [15] <5.4> Repeat Exercise 5.10.1, but this time use 16 KB pages instead of 4 KB pages. What would be some of the advantages of having a larger page size? What are some of the disadvantages?

5.10.3 [15] <5.3, 5.4> Show the final contents of the TLB if it is two-way set-associative. Also show the contents of the TLB if it is direct-mapped? Discuss the importance of having a TLB to high performance. How would virtual memory accesses be handled if there were no TLB?

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

	Virtual address size	Page size	Page table entry size
a.	32 bits	4 KB	4 bytes
b.	64 bits	16 KB	8 bytes

5.10.4 [5] <5.4> Given the parameters in the table above, calculate the total page table size for a system running five applications that utilize half of the memory available.

5.10.5 [10] <5.4> Given the parameters in the table above, calculate the total page table size for a system running five applications that utilize half of the memory available, given a two-level page table approach with 256 entries. Assume each entry of the main page table is 6 bytes. Calculate the minimum and maximum amount of memory required.

5.10.6 [10] <5.4> A cache designer wants to increase the size of a 4 KB virtually indexed, physically tagged cache. Given the page size listed in the table above, is it possible to make a 16 KB direct-mapped cache, assuming two words per block? How would the designer increase the data size of the cache?

Exercise 5.16

Cache coherence concerns the views of multiple processors on a given cache block. The following table shows two processors and their read/write operations on two different words of a cache block X (initially X[0] = X[1] = 0).

	P1	P2
a.	`X[0] ++; X[1] = 4;`	`X[0] = 2; X[1] ++;`
b.	`X[0] ++; X[1] += 3;`	`X[0] = 5; X[1] =2;`

5.16.1 [15] <5.8> List the possible values of the given cache block for a correct cache coherence protocol implementation. List at least one more possible value of the block if the protocol doesn't ensure cache coherency.

5.16.2 [15] <5.8> For a snooping protocol, list a valid operation sequence on each processor/cache to finish the above read/write operations.

5.16.3 [10] <5.8> What are the best-case and worst-case numbers of cache misses needed to finish the listed read/write instructions.

Memory consistency concerns the views of multiple data items. The following table shows two processors and their read/write operations on different cache blocks (A and B initially 0).

	P1	P2
a.	`A = 1; B = 2; A++; B++;`	`C = B; D = A;`
b.	`A = 1; B += 2; A++; B=4;`	`C = B; D = A;`

5.16.4 [15] <5.8> List the possible values of C and D for an implementation that ensures the consistency assumptions on page 535.

5.16.5 [15] <5.8> List at least one more possible pair of values for C and D if such assumptions are not maintained.

5.16.6 [15] <5.2, 5.8> For various combinations of write policies and write allocation policies, which combinations make the protocol implementation simpler?