CS 530 Fall 2014
Computer Architecture

Homework 2 Solutions

1. Problem 2.4

   a. *What are the overall size and block size of the second-level cache?*

      The overall size of the L2 cache corresponds to the array size in L2 cache in which the next largest array accesses the main memory. This behavior can be observed for array size of 1MB, therefore, that is the size of L2 cache. The block size in L2 cache can be obtained via the first stride that makes the main memory access times constant. The L2 cache block size is 128 bytes.

   b. *What is the miss penalty of the second-level cache?*

      The miss penalty of L2 cache implies that data has to be fetched from main memory. The time that takes to access main memory after accessing L2 cache is approximately,

      $$100ns - 6ns = 94ns$$

      The miss penalty is about 94ns.

   c. *What is the associativity of the second-level cache?*

      The set associativity of the L2 cache can be computed by the ratio of the array sizes larger than 1MB and their strides that result in an access time similar to that of an array that fits into the L2 cache. For example, the 128MB array has an access time similar to that of L2 cache when its stride is 16MB, therefore, $\frac{128MB}{16MB} = 8$. Similar calculations can be obtained with array of sizes: 4MB, 8MB, 16MB, 32MB, 64MB, and 256MB. The L2 cache is an 8-way set associative cache.

   d. *What is the size of the main memory?*

      The size of main memory can be deduced by finding the smallest array that does not fit into memory, in other words, the arrays size that has no results because running it was not feasible. The largest array in Figure 2.30, 512MB, has no results, thus we can estimate that the size of main memory is 512MB.

2. Problem 2.8

   a. *What are the relative access times of two-way and four-way set associative caches in comparison to a direct mapped organization?*

Cache organization	Access times (ns)	Relative access times (ns)
Direct mapped	0.862540203138	1
2-way associative	1.12056746501	1.30
4-way associative	1.3713953042	1.59

The data shows a linear increase in access time with the doubling of associativity. The relative access time of the 2-way associative cache is 30% greater than the access time of the direct mapped. The relative access time of the 4-way associative cache is 59% greater than the access time of the direct mapped.

   b. *What are the relative access times of 32KB and 64KB caches in comparison to a 16KB cache?*

Cache size	Access times (ns)	Relative access times (ns)
16KB	1.26889227474	1
32KB	1.34885567595	1.063
64KB	1.3713953042	1.081

The data shows a slight increase in access time with the doubling of cache size. The relative access time of the 32KB cache is 6.3% greater than the access time of the 16KB cache. The relative access time of the 64KB cache is 8.1% greater than the access time of the 16KB cache.

3. Problem B.1

   a. *When you run a program with an overall miss rate of 5%, what will the average memory access time (in CPU cycles) be?*

$$AMAT = 1 + (0.05)(105) = \mathbf{6.25\ clock\ cycles}$$

   b. *You use an array size of 256MB and make accesses using a uniform random number generator. If your data cache size is 64KB, what will the average memory access time be?*

   Since only a portion of the array fits in cache and the accesses are uniformly distributed, we can compute the miss rate as,

$$miss\ rate = 1 - \frac{64KB}{256MB} = 0.99975$$

$$AMAT = 1 + (0.99975)(105) = \mathbf{105.97\ clock\ cycles}$$

   c. *What can you conclude about the role of the principle of locality in justifying the use of cache memory?*

   First let us calculate the average memory access time with the cache disabled,

$$AMAT_{no-cache} = \mathbf{100\ clock\ cycles}$$

   Comparing the AMAT of a system with and without cache shows that the principle of locality is an important factor that affects the miss rate, and consequently the AMAT. Part (b) does not considers the principle of locality resulting in a scenario where cache memory actually degrades the overall performance since most of the time the cache is checked for data that is not resident.

4. Problem B.9: *Now construct a trace of word accesses that would produce more misses in the two-way associative cache.*

   This problem contains an infinite number of solutions based on address traces and number of lines in the cache. The general idea is to show that LRU can produce non-efficient patterns in certain scenarios. Given a direct mapped cache of L lines and a two-way associative cache of $S = \frac{L}{2}$ sets, we can produce a trace of word addresses (or block addresses) that produces more misses in the associative cache by repeating infinitely the generic trace presented in the table below. This trace consists of 3 distinct memory accesses; the last row is the repetition of the first row after wrapping around.

Address	Direct-mapped H/M	Two-way Associative H/M
M[multiple of L]	Miss	Miss
M[multiple of S, not L]	Miss	Miss
M[multiple of S, not L]	Miss	Miss
M[initial multiple of L]	Hit	Miss

   The generic trace can be applied to affect an arbitrary number of sets and their independent accesses can be interleaved. Note that this trace allows symmetry on the address multiples, S can be interchanged with L.

   For example, let us consider $L = 4$ and $S = 2$.

Address	Direct-mapped H/M	Two-way Associative H/M
M[0]	Miss	Miss
M[2]	Miss	Miss
M[6]	Miss	Miss
M[0]	Hit	Miss

5. Problem B.12: *For each access indicate whether it produces a TLB hit/miss and, if it accesses the page table, whether it produces a page hit or fault.*

Virtual page accessed	TLB (hit or miss)	Page table (hit or fault)
1	miss	fault
5	hit	X
9	miss	fault
14	miss	fault
10	miss	hit
6	miss	hit
15	miss	hit
12	miss	hit
7	miss	hit
2	miss	fault

If an entry is invalid in the TLB it is a primary candidate for replacement, valid entries use the LRU replacement policy.

6. *For a direct cache of size 8KB, a word size of 32 bits, and a block size of 8 words, answer the following:*

   a. *Number of lines in the cache?*

$$\#\ of\ lines = \frac{8KB}{(8\ words)(4\ bytes)}$$

$$\#\ of\ lines = 256$$

   b. *Number of bits per tag?*

Assume a virtual address of 32 bits in length,

$$\#\ of\ tag\ bits = 32 - (\log_2 256 + \log_2 32)$$
$$\#\ of\ tag\ bits = 32 - (8 + 5)$$
$$\#\ of\ tag\ bits = 19$$

   c. *Total number of bits required for the cache?*

Assume the cache structure contains a valid bit, tag bits, and a data area for each line in the cache.

$$cache\ size = (256)(1) + (256)(19) + 8KB$$
$$cache\ size = 13KB$$

7. *For a 4-way set associative cache of size 8KB, a word size of 32 bits, and a block size of 8 words, answer the following:*

   a. *Number of lines in the cache?*

$$\#\ of\ lines = \frac{8KB}{(8\ words)(4\ bytes)}$$

$$\#\ of\ lines = 256$$

   b. *Number of bits per tag?*

   Assume a virtual address of 32 bits in length,

$$Number\ of\ sets = \frac{256\ lines}{4 - way} = 64$$
$$\#\ of\ tag\ bits = \ 32 - (\log_2 64 + \log_2 32)$$
$$\#\ of\ tag\ bits = \ 32 - (6 + 5)$$
$$\#\ of\ tag\ bits = 21$$

   c. *Total number of bits required for the cache?*

   Assume the cache structure contains a valid bit, tag bits, and a data area for each line in the cache.

$$cache\ size = (256)(1) + (256)(21) + 8KB$$
$$cache\ size = 13.5KB$$

8. *Assuming a page is 4KB, then how many bits are required for the page table? (Assume all virtual pages are being used.)*

$$\#\ of\ pages = \frac{2^{32}}{4KB} = 2^{20}$$

$$\#\ of\ page\ offset\ bits = \log_2 4KB = 12$$

$$size\ of\ page\ table\ entry = 24 - 12 = 12\ bits$$

Assume the page table contains a present bit and dirty bit for each page table entry.

$$page\ table\ size = \#\ of\ pages\ \times size\ of\ page\ table\ entry$$

$$page\ table\ size = (2^{20})(12 + 1 + 1)$$

$$page\ table\ size = 1.75MB$$

9. *Assume a 2-way set associative cache using LRU with a total of 16 words and a block size of 4 words. For the following word addresses, tell whether each is a hit or a miss. Give the final contents of the cache.*

With a N-word block of data for each cache entry, N words in the entry will have consecutive memory addresses starting with a word address that is a multiple of N.

Word address	Block address	2-way associative (H/M)
4	1	Miss
7	1	Hit
15	3	Miss
11	2	Miss
13	3	Hit
2	0	Miss
9	2	Hit
32	8	Miss
45	11	Miss
1	0	Miss
9	2	Miss
7	1	Miss
8	2	Hit
52	13	Miss
21	5	Miss
12	3	Miss

Cache contents (block addresses)
2 → 0
0 → 8 → 2
1 → 11 → 13 → 3
3 → 1 → 5

10. *Assume a direct cache with a total of 16 words and a block size of 4 words. For the following word addresses, tell whether each is a hit or a miss. Give the final contents of the cache.*

Word address	Block address	Direct-mapped (H/M)
4	1	Miss
7	1	Hit
15	3	Miss
11	2	Miss
13	3	Hit
2	0	Miss
9	2	Hit
32	8	Miss
45	11	Miss
1	0	Miss
9	2	Hit
7	1	Hit
8	2	Hit
52	13	Miss
21	5	Miss
12	3	Miss

Cache contents (block addresses)
0 → 8 → 0
1 → 13 → 5
2
3 → 11 → 3