CE/CZ 3001: Advanced Computer Architecture

Module 5: Memory Systems

- review

Asst Prof Liu Weichen

School of Computer Science and Engineering

Nanyang Technological University, Singapore

Summary: Cache System

- Cache design
  - Organization schemes: direct mapped, set associative, fully associative
  - Write policies: write through, write back
  - Replacement policies: random, FIFO, LRU, etc.
  - Cache miss reasons and optimization
- Cache performance
  - Cache/memory address and size related calculations
  - CPI, Average Memory Access Time (AMAT)

E1

Consider a Byte-addressable memory with the address space of 32 bits. A 128 KB (1 KB = 1024 Bytes) eight-way set-associative cache is used for this memory system with the cache block size of 128 Bytes. Determine the number of bits for the tag, set index and block offset fields of the address, respectively.

Offset bits = log (128 Bytes / 1 Byte) = 7 bits

Number of sets = 128 KBytes / 128 Bytes / 8 = 128

Index bits = log (128) = 7 bits

Tag bits = 32 – 7 – 7 = 18 bits

E2.

Tables Q4a and Q4b shows the actual miss rates and average memory access time for different cache block sizes of a given cache. Answer the following questions according to the data in the tables.

Table Q4a: Actual Miss Rate v.s. Block Size

Block Size (Bytes)	Cache Size			
	4KB	16KB	64KB	256KB
16	8.57%	3.94%	2.04%	1.09%
64	7.00%	2.64%	X%	0.51%
256	9.51%	3.29%	1.15%	0.49%

Table Q4b: Average Memory Access Time v.s. Block Size

Block Size (Bytes)	Miss Penalty	Cache Size			
		4KB	16KB	64KB	256KB
16	82	8.03	4.23	2.67	1.89
64	88	7.16	Y	1.93	Z
256	112	11.65	4.69	2.29	1.55

a. Assume that the hit time takes 1 clock cycle and the memory system takes 80 clock cycles of overhead and then delivers 16 bytes every 2 clock cycles. For example, it can supply 16 bytes in 82 clock cycles, 32 bytes in 84 clock cycles, and so on. Compute the missing entries X, Y and Z in Tables Q3a and Q3b.

Average memory access time = Hit time + Miss rate * Miss penalty

1.93	= 1 + X% * 88	=>	X = 1.06
Y	= 1 + 2.64% * 88	=>	Y = 3.32
Z	= 1 + 0.51% * 88	=>	Z = 1.45

E2.

Tables Q4a and Q4b shows the actual miss rates and average memory access time for different cache block sizes of a given cache. Answer the following questions according to the data in the tables.

<u>Table Q4a: Actual Miss Rate v.s. Block Size</u>

Block Size (Bytes)	Cache Size			
	4KB	16KB	64KB	256KB
16	8.57%	3.94%	2.04%	1.09%
64	7.00%	2.64%	X = 1.06%	0.51%
256	9.51%	3.29%	1.15%	0.49%

<u>Table Q4b: Average Memory Access Time v.s. Block Size</u>

Block Size (Bytes)	Miss Penalty	Cache Size			
		4KB	16KB	64KB	256KB
16	82	8.03	4.23	2.67	1.89
64	88	7.16	Y = 3.32	1.93	Z = 1.45
256	112	11.65	4.69	2.29	1.55

b. Which block sizes have the smallest average memory access time for the cache sizes of 16K, 64K and 256K Bytes, respectively? Which block sizes have the minimum miss rate for each of them?

Block sizes for smallest average memory access time: 64 bytes for all the three cases.

Block sizes for minimum miss rates: 64 bytes for 16k and 64k, and 256 bytes for 256k.

E2.

Tables Q4a and Q4b shows the actual miss rates and average memory access time for different cache block sizes of a given cache. Answer the following questions according to the data in the tables.

Table Q4a: Actual Miss Rate v.s. Block Size

Block Size (Bytes)	Cache Size			
	4KB	16KB	64KB	256KB
16	8.57%	3.94%	2.04%	1.09%
64	7.00%	2.64%	X = 1.06%	0.51%
256	9.51%	3.29%	1.15%	0.49%

Table Q4b: Average Memory Access Time v.s. Block Size

Block Size (Bytes)	Miss Penalty	Cache Size			
		4KB	16KB	64KB	256KB
16	82	8.03	4.23	2.67	1.89
64	88	7.16	Y = 3.32	1.93	Z = 1.45
256	112	11.65	4.69	2.29	1.55

c. Observe from Table Q4a that the miss rate first decreases when the block size increases, but then, the miss rate increases when the block size increases further in most of the cases. Combining the reasons for cache miss, explain the phenomenon shortly.

(1) Increasing block size can take advantage of spatial locality. It reduces compulsory misses and thus reduces miss rate. This explains the decrease when increasing the block size.
(2) However, increasing block size will gain less in temporal locality. It can increase capacity misses and conflict misses, especially when the cache is small. This explains the increased miss rate.

E2.

Tables Q4a and Q4b shows the actual miss rates and average memory access time for different cache block sizes of a given cache. Answer the following questions according to the data in the tables.

Table Q4a: Actual Miss Rate v.s. Block Size

Block Size (Bytes)	Cache Size			
	4KB	16KB	64KB	256KB
16	8.57%	3.94%	2.04%	1.09%
64	7.00%	2.64%	X = 1.06%	0.51%
256	9.51%	3.29%	1.15%	0.49%

Table Q4b: Average Memory Access Time v.s. Block Size

Block Size (Bytes)	Miss Penalty	Cache Size			
		4KB	16KB	64KB	256KB
16	82	8.03	4.23	2.67	1.89
64	88	7.16	Y = 3.32	1.93	Z = 1.45
256	112	11.65	4.69	2.29	1.55

d. Between miss rate and average access time, which one should be emphasized to maximize the cache performance?

Average access time.

E3

You are trying to appreciate how important the principle of locality is in justifying the use of a cache memory, so you experiment with a computer having an L1 data cache and a main memory (you exclusively focus on data accesses). The latencies (in CPU cycles) of the different kinds of accesses are as follows: cache hit, 1 cycle; cache miss, 105 cycles; main memory access with cache disabled, 100 cycles.

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

Average access time = (1 – miss rate) × hit time + miss rate × miss time

(Note: the miss time includes both the time for confirming a miss and the miss penalty)

Average access time = 0.95 × 1 + 0.05 × 105 = 6.2 cycles.

b. Next, you run a program specifically designed to produce completely random data addresses with no locality. Toward that end, you use an array of size 256 MB (all of it fits in the main memory). Accesses to random elements of this array are continuously made (using a uniform random number generator to generate the elements indices). If your data cache size is 64 KB, what will the average memory access time be?

Hit rate = 64 Kbytes/256 Mbytes ≈1/4000 = 0.00025.

average access time = 0.00025 × 1 + (1 – 0.00025) × 105 = 104.974 cycles.

c. If you compare the result obtained in part (b) with the main memory access time when the cache is disabled, what can you conclude about the role of the principle of locality in justifying the use of cache memory?

The access time when the cache is disabled is 100 cycles which is less than the average access time when the cache is enabled and almost all the accesses are misses.

If there is no locality at all in the data stream then the cache memory will not only be useless but also it will be a liability.

d. You observed that a cache hit produces a gain of 99 cycles (1 cycle vs. 100), but it produces a loss of 5 cycles in the case of a miss (105 cycles vs.100). In the general case, we can express these two quantities as G (gain) and L (loss). Using these two quantities (G and L), identify the highest miss rate after which the cache use would be disadvantageous.

Assuming the memory access time with no cache is T_{off}, with cache is T_{on} and the miss rate is m, the average access time (with cache on)

T_{on} = (1 – m) (T_{off} – G) + m (T_{off} + L) ………………………………(1)

The cache becomes useless when the miss rate is high enough to make T_{off} less than or equal to T_{on}.

$T_{off} \leq T_{on}$ ……………………………………………………………………(2)

$\frac{G}{G+L} \leq m$ (G=99 and L=5)

$m \geq 0.9519 = 95.19\%$