King
CS 331
08 December 2020

Homework Assignment 6

Objectives

The purpose of this assignment is to:

- Understand key cache concepts such as choosing a block size, cache size, block placement, block replacement, associative vs. direct-mapped, and multi-level caches,
- Understand the impact of cache design and memory hierarchy on performance.

Guidelines

All question numbers refer to exercises at the end of chapter 5 of the textbook (Computer organization and design: the Hardware/Software interface, 5th edition). Solutions for the following problems are to be done by you and only you.

Questions

Exercise 5.2.1 to 5.2.2

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.

3, 180, 43, 2, 191, 88, 190, 14, 181, 44, 186, 253

1. 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.
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 8 blocks. Also list if each reference is a hit or a miss, assuming the cache is initially empty.

```
Given:
0000 0011, 1011 0100, 0010 1011, 0000 0010,
1011 1111, 0101 1000, 1011 1110, 0000 1110,
1011 0101, 0010 1100, 1011 1010, 1111 1101.

32-bit memory address references.
16 blocks, 1 word/block. 2^0 words = no offset.
```

16 blocks = 2^4 blocks, so the 4 LSB represent the cache index.
Cache index = (block address) % (number of cache blocks)

Word address	Binary address	Tag	Index	Hit/miss
3	0000 0011	0	3	Miss
180	1011 0100	11	4	Miss
43	0010 1011	2	11	Miss
2	0000 0010	0	2	Miss
191	1011 1111	11	15	Miss
88	0101 1000	5	8	Miss
190	1011 1110	11	14	Miss
14	0000 1110	0	14	Miss
181	1011 0101	11	3	Miss
44	0010 1100	2	12	Miss
186	1011 1010	11	10	Miss
253	1111 1101	15	13	Miss

32-bit memory address references.
8 blocks, 2 words/block. 2^1 words = 1 bit offset.
blocks = 2^3 blocks, so the 3 LSB from the offset represent the cache index.

Word address	Binary address	Tag	Index	Hit/miss
3	0000 0011	0000	001	Miss
180	1011 0100	1011	010	Miss
43	0010 1011	0010	101	Miss
2	0000 0010	0000	001	**Hit**
191	1011 1111	1011	111	Miss
88	0101 1000	0101	100	Miss
190	1011 1110	1011	111	**Hit**
14	0000 1110	0000	111	Miss
181	1011 0101	1011	010	**Hit**
44	0010 1100	0010	110	Miss
186	1011 1010	1011	101	Miss
253	1111 1101	1111	110	Miss

Exercise 5.6.1 to 5.6.5 (Note: don't forget to take into account Instruction and Data cache misses.)

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
P1	2 KiB	8.0%	0.66 ns
P2	4 KiB	6.0%	0.90 ns

1. Assuming that the L1 hit time determines the cycle times for P1 and P2, what are their respective clock rates?
2. What is the Average Memory Access Time for P1 and P2?
3. 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

4. What is the AMAT for P1 with the addition of an L2 cache? Is the AMAT better or worse with the L2 cache?
5. 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?

Given:
main memory access takes 70 ns, memory accesses are 36% of all instructions

P1:
L1 size = 2 KiB, L1 miss rate = 8.0%, L1 hit time = 0.66 ns
clock rate = 1/(cycle time)
clock rate = 1/(0.66 ns) = 1/(0.66e-9 s) = 1.515e+9 Hz = 1.52 GHz

P2:
L1 size = 4 KiB, L1 miss rate = 6.0%, L1 hit time = 0.90 ns
clock rate = 1/(0.90 ns) = 1/(0.90e-9 s) = 1.111e+9 Hz = 1.11 GHz

P1:
AMAT = Hit time + (Miss rate × Miss penalty)
AMAT = 0.66 ns + (0.08 × 70 ns) = 6.26 ns

P2: AMAT = 0.90 ns + (0.06 × 70 ns) = 5.1 ns
P1: base CPI of 1.0 MCPI = accesses/instruction × miss rate × miss penalty MCPI = (0.36) × (0.08 × 70 ns) = 5.96 CPI = base CPI + MPCI = 1.0 + 5.96 = 6.96 cycles/instruction - faster processor **P2:** MCPI = (0.36) × (0.06 × 70 ns) = 4.56 CPI = 1.0 + 4.56 = 5.56 cycles/instruction
AMAT = Hit time + Miss rate × Miss penalty AMAT = (5.62 ns) + (0.95 × 70 ns) = 72.12

Exercise 5.7.3 to 5.7.5

(For part 5.7.3: Show the entire table and use only LRU.

For part 5.7.4: Calculate the CPI for the processor in the table using: 1) Only a first level cache, and 2) A second level direct mapped cache.

Then, calculate how these numbers change if main memory access time is doubled.)

This exercise examines the impact of different cache designs, specifically comparing associative caches to the direct-mapped caches from Section 5.4. For these exercises, refer to the address stream shown in Exercise 5.2.

1. Using the references from Exercise 5.2, what is the miss rate for a fully associative cache with two-word blocks and a total size of 8 words, using LRU replacement? 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 MissRate 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
1.5	2 GHz	100 ns	7%	12 cycles	3.5%	28 cycles	1.5%

2. 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?
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?

Fully Associative, block size: 2 words, size: 8 words.

Number of cache blocks = cache size/block size = 8/2 = 4
Offset = 2^1 words = 1 bit. Offset bit of 1 = [a-1, a], 0 = [a, a+1].

Replacement rule: Least recently used

Block address	Binary address	Cache index	Hit/miss	Block 0	Block 1	Block 2	Block 3
3	0000 0011	1	Miss	Mem[2,3]	“	“	“
180	1011 0100	0	Miss	“	Mem[180,181]	“	“
43	0010 1011	1	Miss	“	“	Mem[42,43]	“
2	0000 0010	0	**Hit**	“	“	“	“
191	1011 1111	1	Miss	“	“	“	Mem[190,191]
88	0101 1000	0	Miss	Mem[88,89]	“	“	“
190	1011 1110	0	**Hit**	“	“	“	“
14	0000 1110	0	Miss	“	Mem[14,15]	“	“
181	1011 0101	1	Miss	“	“	Mem[180,181]	“
44	0010 1100	0	Miss	“	“	“	Mem[44,45]
186	1011 1010	0	Miss	Mem[186,187]	“	“	“
253	1111 1101	1	Miss	Mem[186,187]	Mem[252,253]	Mem[180,181]	Mem[44,45]

Final memory references:
3 (m), 180 (m), 43 (m), 2 (h), 191 (m), 88 (m), 190 (h), 14 (m), 181 (m), 44 (m), 186 (m), 253 (m).

Final cache contents:

Block 0	Block 1	Block 2	Block 3
Mem[186,187]	Mem[252,253]	Mem[180,181]	Mem[44,45]

Miss rate = 10/12 = 83.3%

Given:

Base CPI, no memory stalls	Processor speed	Main memory access time	1st-level cache, miss rate per instruction	2nd-level cache, direct-map speed	Global miss rate w/ 2nd-level cache, direct-map	2nd-level cache, 8-way set assoc. speed	Global miss rate w/ 2nd-level cache, 8-way set assoc.
1.5	2 GHz	100 ns	7%	12 cycles	3.5%	28 cycles	1.5%

<u>Only a first level cache:</u>

miss penalty = memory access time/CT
miss penalty = (100.0e-9 sec)/(0.5e+9 cycles/sec) = 200 cycles

MCPI = miss rate × miss penalty
MCPI = (0.07) × (200 cycles) = 14 cycles

total CPI = base CPI + MCPI
total CPI = (1.5 cycles/instruction) + (14 cycles) = 15.5 cycles/instruction

<u>A second level direct mapped cache:</u>

miss penalty = (12 cycles/sec)/(0.5 cycles/ns) = 24 cycles

Total CPI = 1.5 + Primary stalls per instruction + Secondary stalls per instruction

<u>If main memory access time is doubled:</u>

first level cache: CPI = (1.5) + (0.07 × 400) = 29.5 cycles/instruction
second level direct mapped cache:

<u>If main memory access time is cut in half:</u>

first level cache: CPI = (1.5) + (0.07 × 100) = 8.5 cycles/instruction
second level direct mapped cache:

Given:
Third level cache that takes 50 cycles to access and will reduce the global miss rate to 1.3%.

Adding an additional cache level will improve the performance of the processor by increasing the CPI.

Extra Credit

Research the cache organization of a recent Processor (Intel or ARM… different from what are in the textbook), and fill out a table as in Figure 5.44. List references you used for the answer.

Given:
Figure 5.44: Caches in the ARM Cortex-A8 and Intel Core i7 920.

Characteristic	ARM Cortex-A8	Intel Nehalem
L1 cache organization	Split instruction and data caches	Split instruction and data caches
L1 cache size	32 KiB each for instructions/data	32 KiB each for instructions/data per core
L1 cache associativity	4-way (I), 4-way (D) set associative	4-way (I), 8-way (D) set associative
L1 replacement	Random	Approximated LRU
L1 block size	64 bytes	64 bytes
L1 write policy	Write-back, Write-allocate(?)	Write-back, No-write-allocate
L1 hit time (load-use)	1 clock cycle	4 clock cycles, pipelined
L2 cache organization	Unified (instruction and data)	Unified (instruction and data) per core
L2 cache size	128 KiB to 1 MiB	256 KiB (0.25 MiB)
L2 cache associativity	8-way set associative	8-way set associative
L2 replacement	Random(?)	Approximated LRU
L2 block size	64 bytes	64 bytes
L2 write policy	Write-back, Write-allocate(?)	Write-back, Write-allocate
L2 hit time	11 clock cycles	10 clock cycles
L3 cache organization	-	Unified (instruction and data)
L3 cache size	-	8 MiB, shared
L3 cache associativity	-	16-way set associative
L3 replacement	-	Approximated LRU
L3 block size	-	64 bytes
L3 write policy	-	Write-back, Write-allocate
L3 hit time	-	35 clock cycles

Table:

Characteristic	
L1 cache organization	
L1 cache size	
L1 cache associativity	
L1 replacement	
L1 block size	
L1 write policy	
L1 hit time (load-use)	
L2 cache organization	
L2 cache size	
L2 cache associativity	
L2 replacement	
L2 block size	
L2 write policy	
L2 hit time	
L3 cache organization	
L3 cache size	
L3 cache associativity	
L3 replacement	
L3 block size	
L3 write policy	
L3 hit time	

p. 472

References: