Homework 04

Jiarui Huang 202283890036 IoT

Exercise 5.1 In this exercise we look at memory locality properties of matrix computation. The following code is written in C, where element 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][J]+A[I][J];
```

5.1.1 How many 32-bit integers can be stored in a 16-byte cache block?

5.1.2 Which variable references exhibit temporal locality?

5.1.3 Which variable references exhibit spatial locality?

SOLUTION:

5.1.1 4

5.1.2 I, J and A[I][J]

5.1.3 A[I][J] B[I][J]

Exercise 5.2 Caches are important to providing a high-performance memory hierarchy to processors. Below is a list of 32-bit memory address references, given a word addresses.
0x03, 0xb4, 0x2b, 0x02, 0xbf, 0x58, 0xbe, 0x0e, 0xb5, 0x2c, 0xba, 0xfd

5.2.1 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

5.2.2 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.

5.2.3 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 8 words of data:
- C1 has 1-word blocks,
- C2 has 2-word blocks, and

- C3 has 4-word blocks

SOLUTION:

5.2.1

Address	Binary address	Tag	Index	Hit or miss
0x03	0000 0011	0000	0000	miss
0xb4	1011 0100	1011	0100	miss
0x2b	0010 1011	0010	1011	miss
0x02	0000 0010	0000	0010	miss
0xbf	1011 1111	1011	1111	miss
0x58	0101 1000	0101	1000	miss
0xbe	1011 1110	1011	1110	miss
0x0e	0000 1110	0000	1110	miss
0xb5	1011 0101	1011	0101	miss
0x2c	0010 1100	0010	1100	miss
0xba	1011 1010	1011	1010	miss
0xfd	1111 1101	1111	1101	miss

5.2.2

Address	Binary address	Tag	Index	Store block	Hit or miss
0x03	0000 0011	00000	01	1	miss
0xb4	1011 0100	10110	10	0	miss
0x2b	0010 1011	00101	01	1	miss
0x02	0000 0010	00000	01	0	hit
0xbf	1011 1111	10111	11	1	miss
0x58	0101 1000	10111	00	0	miss
0xbe	1011 1110	10111	11	0	hit
0x0e	0000 1110	00001	11	0	miss
0xb5	1011 0101	10110	10	1	hit
0x2c	0010 1100	00101	10	0	miss
0xba	1011 1010	10111	01	0	miss
0xfd	1111 1101	11111	10	1	miss

5.2.3

Address	Binary address	Cache 1 (1-word)			Cache 2 (2-word)			Cache 3 (4-word)		
		Tag	Index	Hit/miss	Tag	Index	Hit/miss	Tag	Index	Hit/miss
0x03	0000 0011	00000	000	miss	00000	01	miss	00000	0	miss
0xb4	1011 0100	10110	100	miss	10110	10	miss	10110	0	miss
0x2b	0010 1011	00101	011	miss	00101	01	miss	00101	1	miss
0x02	0000 0010	00000	010	miss	00000	01	hit	00000	0	hit

0xbf	1011 1111	10111	111	miss	10111	11	miss	10111	1	miss
0x58	0101 1000	10111	000	miss	10111	00	miss	10111	1	miss
0xbe	1011 1110	10111	110	miss	10111	11	hit	10111	1	hit
0x0e	0000 1110	00001	110	miss	00001	11	miss	00001	1	miss
0xb5	1011 0101	10110	101	miss	10110	10	hit	10110	0	hit
0x2c	0010 1100	00101	100	miss	00101	10	miss	00101	1	miss
0xba	1011 1010	10111	010	miss	10111	01	miss	10111	1	miss
0xfd	1111 1101	11111	101	miss	11111	10	miss	11111	1	miss

Exercise 5.3 By convention, a cache is named according to the amount of data it contains (i. e.,a 4 KiB cache can hold 4 KiB of data); however, caches also require SRA M to store metadata such as tags and valid bits. For this exercise, you will ex amine how a cache's configuration affects the total amount of SRAM needed To implement it, as well as the performance of the cache. For all parts, assume th at the caches are byte addressable, and that addresses and words are 64 bits.

5.3.1 Calculate the total number of bits required to implement a 32 KiB cache with two-word blocks.

5.3.2 Calculate the total number of bits required to implement a 64 KiB cache with 16_ word blocks. How much bigger is this cache than the 32 KiB cache described in Exercise 5.3.1? (Notice that, by changing the block size, we doubled the amount of data without doubling the total size of the cache.)

5.3.2 Explain why this 64 KiB cache, despite its larger data size, might provide slower performance than the first cache.

SOLUTION:

5.3.1 The total cache contain is :

$$32\text{KiB} = 32 \times 2^{10} = 2^{15}\,\text{bytes}$$

Of each block has two words, thus the numbers of line is:

$$\log_2(2^{15} \div (2^3 \times 2)) = log_2 2^{11} = 11$$

The 64-bits addresses are divided into : 1-bit for block offset, 11-bits for index offset, 3-bit for word offset and 49-bits for tag (64-1-3-11=49),

The cache is composed with:

$$2^{15} \times 8 + 2^{11} \times (49 + 1) = 364544\ bits$$

5.3.2 The total cache contain is:

$$64\text{KiB} = 64 \times 2^{10} = 2^{16}\text{bytes}$$

With each block is 16 words blocks, for each block, the number of data can store is:

$$16 \times 2^3 = 2^7\text{bytes}$$

The lines of cache needs is:

$$\log_2(2^{16} \div 2^7) = log_2 2^9 = 9\ lines$$

The 64-bits addresses are divided into: 4-bits for block offset, 9-bits for index offse

t, 3-bit for word offset and 48-bits for tag (64-4-3-9=48),
The cache is composed with:

$$2^{16} \times 8 + 2^9 \times (1 + 48) = 549376\ bits$$

5.3.3 The larger block size may require an increased hit time and an increased miss pen alty than the original cache. Th e fewer number of blocks may cause a higher con flict miss rate than the original cache.

Exercise 5.8 Media applications that play audio or video files are part of a class of workloads called "streaming" workloads (i.e., they bring in large amounts of data but do not reuse much of it). Consider a video streaming workload that accesses a 512 KiB working set sequentially with the following address stream:
0, 1, 2, 3, 4, 5, 6, 7, 8, 9 …

5.8.1 Assume a 64 KiB direct-mapped cache with a 32-byte block. What is the miss rate for the address stream above? How is this miss rate sensitive to the size of the cache or the working set? How would you categorize the misses this workload is experiencing, based on the 3C model?

5.8.2 Re-compute the miss rate when the cache block size is 16 bytes, 64 bytes, and 128 bytes. What kind of locality is this workload exploiting?

5.8.3 "Prefetching" is a technique that leverages predictable address patterns to speculativ ely bring in additional cache blocks when a particular cache block is accesseed. One example of prefetching is a stream buffer that prefetches sequentially adjacent cache blocks into a separate buffer when a particular cache block is brought in. If the data is found in the prefetch buffer, it is considered as a hit and moved into the cache and the next cache block is prefetched. Assume a two-entry stream buffer and assume that the cache latency is such that a cache block can be loaded before the computation on the previous cache block is completed. What is the miss rate for the address stream above?

SOLUTION:

5.8.1 Each 32-byte contains 4 words, thus every fourth access will occur one miss, the miss rate is: $\frac{1}{4}$

The miss rate is not sensitive to the size of the cache or the size of the working set.
All misses are compulsory miss.

5.8.2 When the cache block size is 16 bytes, each 16-btyes contains 2 word to store data. Thus every twice access will occur a miss.

The miss rate is: $\frac{1}{2}$

The same as above, the miss rate of the 64-bytes block cache is: $\frac{1}{8}$

The miss rate of 128-bytes is: $\frac{1}{16}$

5.8.3 In this case without including the compulsory miss on the first access, the miss rate is 0.
The pre-fetch buffer always has the next request ready

Exercise 5.9 Cache block size (B) can affect both miss rate and miss latency. Assuming a 1-CPI machine with an average of 1.35 references (both instruction and data) per instruction, help find the optimal block size given the following miss rates for various block sizes.

8: 4%	16: 3%	32: 2%	64: 1.5%	128: 1%

5.9.1 What is the optimal block size for a miss latency of 20 × B cycles?

5.9.2 What is the optimal block size for a miss latency of 24 + B cycles?

5.9.3 For constant miss latency, what is the optimal block size?

SULOTION:

5.9.1 AMAT for B=8: $0.04 \times (20 \times 8) = 6.4$
AMAT for B=16: $0.03 \times 20 \times 16 = 9.6$
AMAT for B=32: $0.02 \times 20 \times 32 = 12.8$
AMAT for B=64: $0.015 \times 20 \times 64 = 19.2$
AMAT for B=128: $0.01 \times 20 \times 128 = 25.6$
B=8 is optimal

5.9.2 AMAT for B=8: $0.04 \times (24 + 8) = 1.28$
AMAT for B=16: $0.03 \times (24 + 16) = 1.2$
AMAT for B=32: $0.02 \times (24 + 32) = 1.12$
AMAT for B=64: $0.015 \times (24 + 64) = 1.32$
AMAT for B=128: $0.01 \times (24 + 128) = 1.52$
Thus, B=32 is optimal

5.9.3 Because miss latency is constant, the optimal is B=128, the miss rate is the lowest.

Exercise 5.10 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 70ns 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 KB	8.0%	0.66 ns
P2	4 KB	6.0%	0.90 ns

5.10.1 Assuming that the L1 hit time determines the cycle times for P1 and P2, what are their respective clock rates?

5.10.2 What is the Average Memory Access Time for P1 and P2?

5.10.3 Assuming a base CPI of 1.0 without any memory stalls, what is the total CPI for P1 and P2? Which processor is faster? (When we say a "base CPI of 1.0", we mean that instructions complete in one cycle, unless either the instruction access or the data access causes a cache miss.) 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 MB	95%	5.62 ns

5.10.4 What is the AMAT for P1 with the addition of an L2 cache? Is the AMAT better Or worse with the L2 cache

5.10.5 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?

5.10.6 What would the L2 miss rate need to be in order for P1 with an L2 cache to be faster than P1 without an L2 cache?

5.10.7 What would the L2 miss rate need to be in order for P1 with an L2 cache to be faster than P2 without an L2 cache?

SOLUTION:

5.10.1 The hit time determines the clock cycle time of p1 and p2,

And clock rate $= \frac{1}{clock\ cycle\ time}$

Thus, the clock rate of p1= $\frac{1}{0.66ns} = \frac{1}{0.66\times10^{-9}} = 1.515 \times 10^{9}Hz = 1.151GHz$

The clock rate of p2= $\frac{1}{0.9ns} = \frac{1}{0.9\times10^{-9}} = 1.111 \times 10^{9}Hz = 1.111GHz$

5.10.2 AMAT of P1: AMAT of P1 =p1 hit time+ L1 miss rate×miss penalty
AMAT of P1 =0.66ns+0.08×70
AMAT of P1 =6.26ns

AMAT of P2: AMAT of P2 =p2 hit time+ L1 miss rate×miss penalty
=0.9ns+0.06×70
=5.1ns

5.10.3 For P1, every instruction require one cycle, additionally, 8% of data access will be missed, thus, it require additional 0.08×(70/0.66)≈ 9 cycles
Furthermore, 36% of the instructions are data accesses. 8% of these 36% are cache misses, which adds an additional 107 cycles.
The total CPI is:

$$1 + 9 + 0.08 \times 0.36 \times 107 = 12.96 \text{ cycles}$$

For P2, every instruction require on cycle, additionally, 8% of data access will be missed, thus, it require additional 0.06×(70/0.9)≈ 5 cycles
Furthermore, 36% of the instructions are data accesses. 8% of these 36% are cache misses, which adds an additional 107 cycles.
The total CPI is:

$$1 + 5 + 0.06 \times 0.36 \times （70 \div 0.9） \approx 7.68 \text{ cycles}$$

5.10.4 For each L1 access, it will take 0.66ns, and for L1 miss, it will take access to L2 to find data, it will take 5.62ns, if L2 also miss, it will access main memory.
Thus, the AMAT of P1 is: $0.66 + 0.08 \times (5.62 + 0.95 \times 70) \approx 6.42\text{ns}$
It will be worse with L2.

5.10.5 One access for L1 requires one cycle, 8% of memory miss and access L2, it will require extra

$$5.62 \div 0.66 \approx 8.5\text{cycles}$$

And 95% of memory in L2 miss and access main memory, and 36% of instruction will access main memory.
Thus, the total CIP of P1 is:

$$1 + 0.08 \times (9 + 0.95 \times (70 \div 0.66)) + 0.36 \times 0.08 \times (9 + 0.95 \times (70 \div 0.66)) \approx 13\text{cycles}$$

5.10.6 Assume AMAT of P2 without of L2<AMAT of P1 with L2

$$6.26 < 0.66 + 0.08 \times (5.62 + \text{x} \times 70)$$

x>0.919

Thus, the highest miss rate of L2 is 92% if P1 with an L2 cache to be faster than P1 without an L2 cache.

5.10.7 Assume AMAT of P1 with of L2<AMAT of P1 with L2

$$5.1 < 0.66 + 0.08 \times (5.62 + \text{x} \times 70)$$

x>0.71

Thus, the highest miss rate of L2 is 71% if P1 with an L2 cache to be faster than P1 without an L2 cache.

Exercise 5.12 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%

**First Level Cache miss rate is per instruction. Assume the total number of L1 cache misses (instruction and data combined) is equal to 7% of the number of instructions.

5.12.1 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? (Give each change as both an absolute CPI and a percent change.) Notice the extent to which an L2 cache can hide the effects of a slow memory.

5.12.2 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 have a 13% miss rate. Would this provide better performance? In general, what are the advantages and disadvantages of adding a third level cache?

SOLUTION:

5.12.1 1) For each access will require 1.5 cycle, 7% of access will miss, the extra cycle require is: $0.07 \times (100 \div 0.5) = 14$
Thus, the total time access cost is:

$$1.5+14=15.5 \text{ cycles}$$

2) With second cache, for each access will require 1.5 cycles, and 7% access will miss and access cache second level cache.
Thus, the total CPI is:

$$1.5 + 0.07 \times (12 + 0.035 \times (100 \div 0.5)) = 2.83\text{cycles}$$

3) For the 8-way set associated case, the total CPI is:

$$1.5 + 0.07 \times (28 + 0.0015 \times 200) = 3.67$$

For the main memory access time is doubled
1) The total CPI of single cache is: $1.5+0.07 \times (200 \div 0.5) = 29.5$
2) The total CPI of a second level direct-mapped cache is:

$$1.5 + 0.07 \times (12 + 0.035 \times (200 \div 0.5)) = 3.32\text{cycles}$$

3) The total CPI of second level 8-ways set is:

$$1.5 + 0.07 \times (28 + 0.035 \times (200 \div 0.5)) = 3.88\text{cycles}$$

5.12.2 The total CPI of the third level cache is:

$$1.5 + 0.07 \times (12 + 0.035 \times (50 + 0.13 \times 200)) = 2.52\text{cycles}$$

For the second level cache is 2.83cycles, the performance is better.

Exercise 5.19 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

5.19.1 Under what scenarios would entry 2's valid bit be set to zero?

5.19.2 What happens when an instruction writes to VA page 30? When would a software managed TLB be faster than a hardware managed TLB?

5.19.3 What happens when an instruction writes to VA page 200?

SOLUTION:

5.19.1 It would be invalid if it was paged out to disk

5.19.2 A write to VA page 30 will occur a miss. When software can pre-fetch TLB entry, it will be faster than hardware.

5.19.3 An interrupt would be generated because the page is marked as read only.

Exercise 5.20 In this exercise, we will examine how replacement policies impact miss rate. Assume a 2-way set associative cache with 4 one word blocks. Consider the following word address sequence: 0, 1, 2, 3, 4, 2, 3, 4, 5, 6, 7, 0, 1, 2, 3, 4, 5, 6, 7, 0.

5.20.1 Assuming an LRU replacement policy, which accesses are hits?

5.20.2 Assuming an MRU (most recently used) replacement policy, which accesses are hits?

5.20.3 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?

5.20.4 Describe an optimal replacement policy for this sequence. Which accesses are hits using this policy?

5.20.5 Describe why it is difficult to implement a cache replacement policy that is optimal for all address sequences.

5.20.6 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?

SOLUTION:

5.20.1

Highlight in yellow is the LRU block.

Address		Index	Tag	Block1		Block2		Miss or hit
0	000	0	0	0	Empty	Empty	Empty	Miss
1	001	0	0	0	1	Empty	Empty	Miss
2	010	1	0	0	1	2	Empty	Miss
3	011	1	0	0	1	2	3	Miss
4	100	0	1	4	1	2	3	Miss
2	010	1	0	4	1	2	3	Hit
3	011	1	1	4	1	2	3	Hit
4	100	0	1	4	1	2	3	Hit
5	101	0	1	4	5	2	3	Miss
6	110	1	1	4	5	6	3	Miss
7	111	1	1	4	5	6	7	Miss
0	000	0	0	0	5	6	7	Miss
1	001	0	0	0	1	6	7	Miss
2	010	1	0	0	1	2	7	Miss
3	011	1	0	0	1	2	3	Miss
4	100	0	1	4	1	2	3	Miss
5	101	0	1	4	5	2	3	Miss
6	110	1	1	4	5	6	3	Miss
7	111	1	1	4	5	6	7	Miss
0	000	0	0	0	5	6	7	Miss

5.20.2

Highlight in yellow is the MRU block.

Address		Index	Tag	Block1		Block2		Miss or hit
0	000	0	0	0	Empty	Empty	Empty	Miss
1	001	0	0	0	1	Empty	Empty	Miss
2	010	1	0	0	1	2	Empty	Miss
3	011	1	0	0	1	2	3	Miss
4	100	0	1	0	1	2	4	Miss
2	010	1	0	0	1	2	4	Hit
3	011	1	1	0	1	3	4	Miss
4	100	0	1	0	1	3	4	Hit
5	101	0	1	0	1	3	5	Miss
6	110	1	1	0	1	3	6	Miss
7	111	1	1	0	1	3	7	Miss
0	000	0	0	0	1	3	7	Hit
1	001	0	0	0	1	3	7	Hit
2	010	1	0	0	2	3	7	Miss

3	011	1	0	0	2	3	7	Hit
4	100	0	1	0	2	4	7	Miss
5	101	0	1	0	2	5	7	Miss
6	110	1	1	0	2	6	7	Miss
7	111	1	1	0	2	6	7	Hit
0	000	0	0	0	2	6	7	Hit

5.20.3 The following is a Simulation results execute by myself."Number" evict the first block in a set and "flower" means to evict the second block in a set.

Address		Block1		Block2		Miss or hit
0	Number	0	Empty	Empty	Empty	Miss
1	Flower	0	Empty	1	Empty	Miss
2	Flower	0	Empty	1	2	Miss
3	Flower	0	Empty	3	2	Miss
4	Number	0	4	3	2	Miss
2	Flower	0	4	3	2	Hit
3	Number	0	4	3	2	Hit
4	Flower	0	4	3	2	Hit
5	Flower	0	4	3	5	Miss
6	Flower	0	4	6	5	Miss
7	Number	7	4	6	5	Miss
0	Flower	7	4	6	0	Miss
1	Number	7	1	6	0	Miss
2	Number	2	1	6	0	Miss
3	Flower	2	1	3	0	Miss
4	Flower	2	1	3	4	Miss
5	Flower	2	1	5	4	Miss
6	Number	2	6	5	4	Miss
7	Flower	2	6	5	7	Miss
0	Number	0	6	5	7	Miss

The miss rate is $\frac{17}{20} \approx 0.85$

5.20.4 MRU is an optimal policy

5.20.5 The best block to evict is the one that will cause the fewest misses in the future. Unfortunately, a cache controller cannot know the future! Our best alternative is to make a good prediction.

5.20.6 If you knew that an address had limited temporal locality and would conflict with another block in the cache, choosing not to cache it could improve the miss rate. On the other hand, you could worsen the miss rate by choosing poorly which addresses to cache.