15-213: Introduction to Computer Systems
Written Assignment #4

This written homework covers the Memory Hierarchy and Cache Memories.

Directions

Complete the question(s) on the following pages with single paragraph answers. These questions are not meant to be particularly long! Once you are done, submit this assignment on Canvas.

Below is an example question and answer.

Q: Please describe benefits of two's-complement signed integers versus other approaches.

A: Other representations of signed integers (ones-complement and sign-and-magnitude) have two representations of zero (+0 and -0), which makes testing for a zero result more difficult. Also, addition and subtraction of two's complement signed numbers are done exactly the same as addition and subtraction of unsigned numbers (with wraparound on overflow), which means a CPU can use the same hardware and machine instructions for both.

Grading

Assignments are graded via *peer review:*

1. Three other students will each provide short, constructive feedback on your assignment, and a score on a scale of 1 to . You will receive the maximum of the three scores.
1. You, in turn, will provide feedback and a score for three other students (not the same ones as in part 1). We will provide a *rubric*, a document describing what good answers to each question look like, to assist you. You receive five additional points for completing all of your peer reviews.

Due Date

This assignment is due on July 6 by 11:59pm Pittsburgh time (currently UTC-4). Remember to convert this time to the timezone you currently reside in.

Peer reviews will be assigned roughly 12 hours later, and are due a week after that.

Question 1

a) Explain one thing that cache can do but main memory can't and one thing that main memory can do but a cache can't. If comparing speed or size, be sure to give approximate quantities.
b) Explain one thing that main memory can do but local disks can't and one thing that local disks can do but main memory can't. If comparing speed or size, be sure to give approximate quantities.

a) The latency of cache is much lower than that of main memory. For example, L1 cache can have a 4-cycle latency while main memory can have 100-cycle latency.
The size of main memory is much larger than that of cache. For example, L1 cache can have a 64KB size, while the main memory can have an 8GB size.

b) Main memory access time is much lower than disks. For example, a DRAM's access time is 100 ns but a disk's is 10,000,000 ns.
Disks can store data even if the power is off while main memory loses all of its content after power off.

Question 2

How does a write-back cache handle a write hit differently than a write-through cache? What are the pros and cons of a write-through cache? What are the pros and cons of a write-back cache?

Write-back:
The cache defer writing to memory until the cache line is evicted.
Pros: minimum risk of losing data in memory if the cache fails
Cons: high latency and low throughput

Write-through:
The cache immediately modifies the content of the memory
Pros: low latency and high throughput
Cons: needs extra space for storing dirty bit. And potential data lost in memory if the cache fails

Question 3

Consider a cache of the following configurations

Total Cache Size = 1024 bytes
Block size = 32 bytes
Number of sets = 8
16 bit address space

A. How many block, set, and tag bits are there?
B. How many lines are there per set?
C. Memory is accessed at the address 0x1A23. Write out the tag, set and block bits.
D. Next, memory is accessed at 0x1B32. Is this a hit or a miss? If it is a miss, what kind, and does it evict the other line from the cache? If it is a hit, what kind of locality does it share with the other line?
E. Following this, the next byte of data is accessed with address 0x1B33. Is this a hit or a miss? If it is a miss, what kind, and does it evict the other line from the cache? If it is a hit, what kind of locality does it share with the other line?
F. Finally, the programmer begins a program which loops, accessing 33 distinct places in memory, none of which share a cache block. What kind of misses does this pattern create? How can you differentiate this from any other kinds of misses discussed in (D) or (E)?

A.
# of block bits $b = lg32 = 5$
# of set bits $s = lg8 = 3$
# of tag bits $t = 16 - 3 - 5 = 8$

B.
# of lines per set $= 1024 / 32 / 8 = 4$

C.
0x1A23 = 0b 00011010 001 00011
tag bits = 00011010
Set bits = 001
Block bits = 00011

D.
0x1B32 = 0b 00011011 001 10010
The set bits are the same as 0x1A32 but the tag bits are different. Since a set has 4 cache blocks, this is a compulsory miss without eviction.

E.
0x1B33 has the same tag bits and set bits and is within the same cache block as 0x1B32. This is a hit, and it shares spatial locality.

F.
This creates compulsory misses and possibly conflict misses.
It led to compulsory misses because the cache is still largely empty and these blocks are added to the cache for the first time.
It could lead to conflict misses because it's possible that more than four of the memory addresses accessed have the same set bits and block bits but with different tag bits, causing evictions.