CS110: Computer Architecture I

Spring 2026

Homework #5

Due: _____ at _____

Total Points: 100

Gradescope submission format. Enter hexadecimal answers as `0x...` with uppercase `A-F` and no leading zeros. Use English commas with no spaces after commas, for example `B1,B4` and `0x0,0x1,0x2`. Enter only integers unless a problem explicitly asks for CPI, round to two decimal places for CPI. When a problem asks for a choice token, enter exactly `1` or `2`. Do not add units or ending punctuation in Gradescope answer boxes.

Problem 1 (*Cache Geometry and Address Breakdown*) **[20 pts]**

Given a 32-bit byte-addressed system and a direct-mapped L1 data cache with total size 8 KiB (8192 B) and block size 64 B.

(a) Derive the cache geometry and address format for this cache. Report the number of cache lines and the number of tag, index, and block-offset bits. **[8 pts]**
cache lines = _____, tag bits = _____, index bits = _____, offset bits = _____.

(b) For address `0x1234ABCD`, write the tag in hexadecimal (use `0x` with uppercase `A-F` and no leading zeros) and the index and offset in decimal. **[6 pts]**
tag = _____, index = _____, offset = _____.

(c) Reuse the numerical tag, index, and offset values from part (b). Now suppose the cache were instead direct-mapped with total size 16 KiB and block size 32 B. Reconstruct the byte address in hexadecimal (use `0x` with uppercase `A-F` and no leading zeros). **[6 pts]**
resulting address = _____.

Problem 2 (*Direct-Mapped Mapping and Conflicts*) **[20 pts]**

Consider a 32-bit byte-addressed system with a 128 B direct-mapped cache and 16 B block size. Use the following five block-aligned byte addresses. In Problems 2 and 3, write block labels in answers as `B1`, `B2`, ..., `B5` (Do not add period at the end of your answer).

$$B_1 = \texttt{0x00000000},\ B_2 = \texttt{0x00000080},\ B_3 = \texttt{0x00000100},$$
$$B_4 = \texttt{0x00000010},\ B_5 = \texttt{0x00000090}$$

(a) Partition the five listed addresses into the two direct-mapped conflict classes. Write labels in ascending order within each class, and list the class containing B_1 first. Example format: `B1,B4,B5`. **[6 pts]**
class 1 = _____, class 2 = _____.

(b) For the conflict class containing B_1 and the conflict class containing B_4, give the common cache line index and list the tag values in ascending hexadecimal order, separated by English

commas with no spaces (use `0x` with uppercase `A-F` and no leading zeros). Example format: `0x0,0x1,0x2`. **[6 pts]**
class of B_1: line = _____, tags = _____;
class of B_4: line = _____, tags = _____.

(c) Starting from an empty cache, consider the access trace `B1,B4,B1,B2,B4,B5,B2,B3`. Assume hit time is 2 cycles and miss penalty is 18 cycles. What is the total access time of this trace? **[4 pts]**
total access time = _____ cycles.

(d) Keeping the cache direct-mapped and the block size fixed at 16 B, what minimum total cache size would allow all five listed blocks to coexist simultaneously? Give the answer in bytes. **[4 pts]**
minimum total size = _____ bytes. *[Hint: Think in terms of how binary address bits map onto hardware structure.]*

Problem 3 (*Associativity and LRU*) **[20 pts]**

Use the same five labeled block-aligned byte addresses from Problem 2. Keep the total cache size 128 B and the block size 16 B, but now change the mapping to a 2-way set associative cache with LRU replacement.

(a) Give the number of sets and list the blocks that map to the same set as B_1. Write labels in ascending order, separated by English commas with no spaces. Example format: `B1,B4`. **[6 pts]**
number of sets = _____, same-set blocks = _____.

(b) Among the five listed blocks, what is the maximum number that can reside simultaneously in this 2-way cache? **[4 pts]**
maximum resident blocks = _____.

(c) Keeping the total cache size and block size fixed, what minimum power-of-two associativity allows all five listed blocks to coexist simultaneously? **[4 pts]**
minimum associativity = _____.

(d) Starting from an empty cache, compare the total number of hits under the 2-way cache from this problem and under a 4-way set associative cache with the same total size and block size. Use LRU in both caches for this byte-address trace. **[6 pts]**

`0x00000004,0x00000088,0x0000000C,0x00000100,`
`0x00000084,0x00000108,0x00000008`

hits in 2-way cache = _____, hits in 4-way cache = _____.

Problem 4 (*Memory Performance and Data Layout*) **[40 pts]**

Write CPI answers to **two decimal places with rounding**. All other numeric answers in this problem should be written as integers.

Part I: Two-level cache and CPI model

Use this machine model for parts (a) and (b):

- L1 hit time = 1 cycle
- L1 miss rate = 12%
- On an L1 miss, the access goes to L2
- L2 hit time = 6 cycles
- L2 local miss rate = 20%
- On an L2 miss, the additional main-memory penalty = 90 cycles
- Base CPI (without memory stalls) = 1.0
- Memory accesses per instruction = 0.40

Part II: Data layout and streaming kernels

For parts (c) to (e), keep the same base CPI, memory-access rate, and two-level cache hierarchy. Also assume cache block size is 64 B, `double` is 8 B, `Particle` has no extra padding, and:

```
typedef struct {
   double x;
   double y;
   double z;
   double vx;
   double vy;
   double vz;
   double mass;
   double charge;
} Particle;
Particle p[N];
double x[N], mass[N];
double acc = 0;
```

Compare the following kernels:

Kernel 1

```
for (i = 0; i < N; i++)
   acc += p[i].x * p[i].mass;
```

Kernel 2

```
for (i = 0; i < N; i++)
   acc += x[i] * mass[i];
```

(a) Compute the effective stall penalty of an L1 miss (i.e., hit paths touch L1 hit are not considered) and the final CPI. **[8 pts]**
effective miss penalty = _____ cycles, final CPI = _____.

(b) Consider two independent optimizations. Optimization 1 reduces the L1 miss rate from 12% to 9%. Optimization 2 reduces the L2 local miss rate from 20% to 10%. Compute the resulting CPI of each optimization, then identify the better optimization using exactly `1` or `2`. **[8 pts]**
CPI with opt 1 = _____, CPI with opt 2 = _____,
better optimization = _____.

(c) Give the size of `Particle`, the stride between consecutive `x` values in each kernel, and the number of loop iterations whose `x` values are supplied by one 64 B cache block in each kernel. **[8 pts]**
Particle size = _____ B, x-stride in Kernel 1 = _____ B, x-stride in Kernel 2 = _____ B,
iterations/block in Kernel 1 = _____, iterations/block in Kernel 2 = _____.

(d) Assume $N = 8192$ and the referenced data arrays are cold before the loop. Count only compulsory data-block fetches for the arrays read in the loop body; ignore conflict and capacity effects. Compute the total number of 64 B blocks fetched by each kernel. **[8 pts]**
blocks in Kernel 1 = _____, blocks in Kernel 2 = _____.

(e) Suppose the measured L1 miss rates are 18% for Kernel 1 and 6% for Kernel 2. Using the same two-level hierarchy, compute the CPI of each kernel and identify the better kernel using exactly `1` or `2`. **[8 pts]**
CPI of Kernel 1 = _____, CPI of Kernel 2 = _____, better kernel = _____.