

# CMPE 110 Computer Architecture

## Fall 2016, Homework #3

Computer Engineering  
UC Santa Cruz

November 3, 2016

Name: \_\_\_\_\_

Email: \_\_\_\_\_

### Submission Guidelines:

- This homework is due on **Friday 11/16/16**.
- The homework must be submitted to ecommons by 11:59 pm.
  - Anything later is a late submission
- Please write your **name** and your UCSC **email address**
- The homework should be “readable” without too much effort
  - The homework must be typed and submitted as a single file in PDF format
  - Please name your homework file cmpe110-hw3-yourcruzid.pdf
  - Please keep your responses coherent and organized or you may lose points
- Provide details on how to reach a solution. **An answer without explanation gets no credit. Clearly state all assumptions.**
- Points:  $64 = 24 + 12 + 12 + 16$

Question	Part A	Part B	Part C	Part D	Part E	Part F	Total
1							
2				-	-	-	
3					-	-	
4					-	-	
Total							

## Question 1. Cache Mapping and Access (24 points)

Consider a 512-KByte cache with 64-word cachelines (a cacheline is also known as a cache block, each word is 4-Bytes). This cache uses write-back scheme, and the address is 32 bits wide.

### Question 1.A Direct-Mapped, Cache Fields (4 points)

Assume the cache is direct-mapped. Fill in the table below to specify the size of each address field.

Field	Size (bits)
Cacheline Offset	
Cacheline Index	
Tag	

### Question 1.B Fully-Associative, Cache Fields (4 points)

Assume the cache is fully-associative. Fill in the table below to specify the size of each address field.

Field	Size (bits)
Cacheline Offset	
Cacheline Index	
Tag	

### Question 1.C 16-Way Set-Associative, Cache Fields (4 point)

Assume the cache is 16-way set-associative. Fill in the table below to specify the size of each address field.

Field	Size (bits)
Cacheline Offset	
Cacheline Index	
Tag	

### Question 1.D Direct-Mapped, Cache Transactions (4 points)

Assume the cache is direct-mapped. Fill in the table on the next page to identify the content of the cache after each of the following memory accesses. Assume the cache is initially empty (also called a “cold cache”). Specify if an entry causes another line to be replaced from the cache, and if an entry has to write its data back to memory. For the data column, specify the data in the block by referring to its address like M[address]. Write accesses will modify the data, so let’s indicate the data after a write access with D[address].

Address	Request Type	Cacheline Index	Hit or Miss?	Modified	Tag	Data	Caused Replace?	Write-back to Memory?
0x128	read					M[0x100]		
0xF40	write					D[0xF00]		
0xA00051	read							
0x093	write							
0x4000B44	read							

### Question 1.E 16-Way Set-Associative, Cache Transactions (4 points)

Assume the cache is 16-way set-associative. Fill the table below to identify the content of the cache after each of the following memory accesses. Assume the cache is empty in the beginning (also known as cold cache). Specify if an entry causes another line to be replaced from the cache, and if an entry has to write its data back to memory. For the data column, specify the data in the block by referring to its address like M[address]. Write accesses will modify the data, so lets indicate the data after a write access with D[address].

Address	Request Type	Cacheline Index	Hit or Miss?	Modified	Tag	Data	Caused Replace?	Write-back to Memory?
0x128	read					M[0x100]		
0xF40	write					D[0xF00]		
0xA00051	read							
0x093	write							
0x4000B44	read							

### Question 1.F Overhead (4 points)

What is the overhead and actual size of the direct-mapped cache? What is the overhead and actual size of the 16-way set-associative cache? Does the structure change the overhead in terms of number of memory bits?

## Question 2. Average Memory Access Time (12 points)

Considered two pipelined machines A and B, described in the table below.

Property	Computer A	Computer B
ISA	MIPS	x86
Clock Frequency	2 GHz	2.5 GHz
Base CPI	1 cycle/instruction	2 cycles/instruction
L1 Instruction Cache Hit Time	1 cycle	1 cycle
L1 Instruction Cache Miss Rate	2%	2%
L1 Data Cache Hit Time	1 cycle	2 cycles
L1 Data Cache Miss Rate	8%	4%
L2 Hit Time	15 cycles	12 cycles
L2 Global Miss Rate	3%	4%
Main Memory Access Time	250 cycles	300 cycles

**Figure 1: Table containing properties of two different machines**

### Question 2.A Ideal System (4 points)

Assume a perfect memory system (100% of memory accesses hit in the L1 caches) and perfect branch prediction. Assume that MIPS programs execute 1.25x as many instructions as x86 programs. Which machine has the faster execution time and what is the speedup?

### Question 2.B No L2 Cache (4 points)

Now assume each machine has only L1 caches (split iL1 cache and dL1 cache), no L2 cache, perfect branch prediction, and that 30% of the instructions are loads/stores. Which machine has the faster execution time and what is the speedup?

### Question 2.C Full System (4 points)

Now assume each machine has both L1 caches and a unified L2 cache, and that 30% of the instructions are loads/stores. Which machine has the faster execution time and what is the speedup?

### Question 3. Cache and AMAT (12 points)

A processor has two levels of caches:

- A 2-way set-associative L1 cache that can house 4 blocks in total. The access latency to this cache is 1 cycle.
- A 16-way set-associative L2 cache that can house 128 blocks in total. The access latency to this cache is 20 cycles.

Both L1 and L2 caches employ LRU replacement policy. The processor does not employ any prefetching mechanism.

A programmer writes a test program that repeatedly accesses only the following data cache blocks in a loop (assume billions of iterations are executed):

A, B, C, D, E, F

where A, . . . , F are different cache block addresses.

In the steady state (i.e., after the loop has executed for a large number of iterations), the programmer observes that the average memory access time (AMAT) is 1 cycle.

Then, the programmer writes another program that repeatedly accesses only the following data cache blocks in a loop:

A, B, C, D, E, F, G, H

In the steady state, the programmer observes that the AMAT is 20 cycles.

#### Question 3.A (3 points)

Why do the two programs have different AMAT? Please explain.

#### Question 3.B (3 points)

Based on the above information, what do you expect the average memory access time of yet another program that repeatedly accesses only the following data cache blocks in a loop?

A, B, C, D, E

Please explain.

### Question 3.C (3 points)

Again, based on the above information, what do you expect the average memory access time of yet another program that repeatedly accesses only the following data cache blocks in a loop?

A, B, C, D, E, F, G

Please explain.

### Question 3.D (3 points)

Finally, based on the above information, what do you expect the average memory access time of yet another program that repeatedly accesses only the following data cache blocks in a loop?

A, B, C, D, E, F, G, H, I

Please explain.

## Question 4. Caching Basics (16 points)

Below, we have given you four different sequences of addresses generated by a program running on a processor with a data cache. Cache hit ratio for each sequence is also shown below.

Sequence No.	Address Sequence	Hit Ratio
1	0, 2, 4, 8, 16, 32, 64, 128	0.50
2	0, 1024, 2048, 3072, 4096, 3072, 2048, 1024, 0	0.33
3	0, 256, 512, 1024, 2048, 1024, 512, 256, 0	4/9
4	0, 512, 2048, 0, 1536, 0, 1024, 512	0.25

Assumptions: all memory accesses are one byte accesses. All addresses are byte addresses. Assuming that the cache is initially empty at the beginning of each sequence, find out the following parameters of the processors data cache:

- 1) Associativity (1, 2 or 4 ways) **(4 points)**
- 2) Block size (1, 2, 4, 8, 16, or 32 bytes) **(4 points)**
- 3) Total cache size (1024 B or 2048 B) **(4 points)**
- 4) Replacement policy (LRU or FIFO) **(4 points)**

Please justify (explain) your answers.