

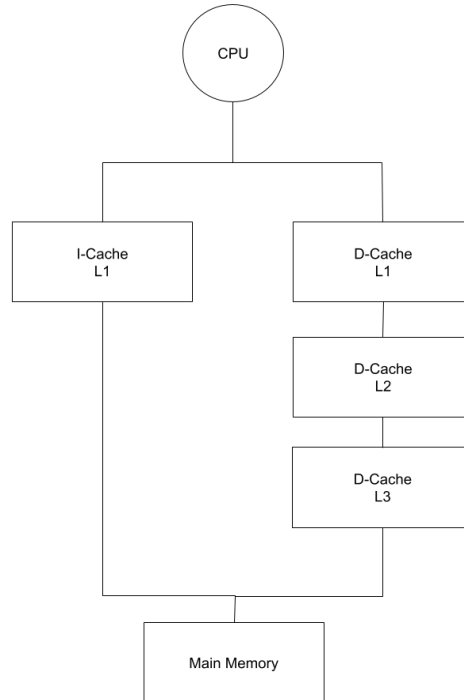
CS 2200 Spr24 Exam 3: Released

Question	Topic	Points
1	Cache EMAT and CPI	14
2	Paging and Page Size	10
3	Memory Pressure	4
4	Cache Miss Types	12
5	Cache Design	10
6	Cache Calculations	11
7	Page Replacement	9
Total	-----	70

Note: Released questions will not change qualitatively, but can change quantitatively. For example, numbers, instructions, order of events, etc. may be different in the exam. **In the actual exam, any such changes will be bolded.** The exam uses the LC-2200 instruction set, with the addition of the LEA and BGT instructions.

Question 1 - (14 points)

Sanskriti's pipelined processor features separate instruction and data caches. The instruction cache (I-cache) has a **single** level and the data cache (D-cache) has **three** levels, as shown in the following diagram.



	I-Cache L1	D-Cache L1	D-Cache L2	D-Cache L3
Hit Ratio	97%	88%	75%	72%
Access Time	4 cycles	4 cycles	14 cycles	24 cycles
Block Size	1 word	1 word	1 word	1 word

You may assume the following information about this processor and its caches:

- The processor achieves an average CPI of 1.2, without accounting for memory stalls.
- Memory accesses (LWs and SWs) account for 25% of all the instructions executed.
- Out of these memory accesses, 65% are reads and 35% are writes.
- On a cache miss, the CPU experiences a latency of 120 cycles to read from memory, and a latency of 7 cycles to write to memory.
- All caches use the write-through and write-allocate policies.
- There is no write-buffer in the datapath between the CPU and memory bus, hence the CPU must stall until each write reaches memory.

For each of the following questions (1.1 to 1.9), show your work to receive credit. Round your final answers to **2 decimal places**. For EMAT calculations, give your answer in terms of **CPU cycles**.

Q1.1 (1 points)

What percentage of all instructions are loads? What percentage of all instructions are stores?

Q1.2 (2 points)

Calculate the EMAT for an I-Cache read.

Q1.3 (1 point)

Calculate the EMAT for a D-cache L3 read.

Q1.4 (1 point)

Calculate the EMAT for a D-cache L2 read.

Q1.5 (2 points)

Calculate the EMAT for a D-cache L1 read.

Q1.6 (1 point)

Calculate the EMAT for a D-cache L3 write.

Q1.7 (1 point)

Calculate the EMAT for a D-cache L2 write.

Q1.8 (2 points)

Calculate the EMAT for a D-cache L1 write.

Q1.9 (3 points)

Calculate the effective CPI of the processor, accounting for all the memory stalls.

Question 2 (10 points)

Consider a memory system with 24-bit virtual addresses, 20-bit physical memory addresses, and a page size of 2048 B. Calculate the following for questions 2.1 - 2.4.

Q2.1 (2 points)

Size of VPN in bits.

Q2.2 (2 points)

Size of PFN in bits.

Q2.3 (1 point)

Number of page table entries (answer should be in the format: 2^n).

Q2.4 (1 point)

Number of frame table entries (answer should be in the format: 2^n).

Q2.5 (2 points)

A standard page size for virtual memory is 4 KB, but some systems support page sizes of 2 MB or larger. How does the page size affect the frequency of page faults in a virtual memory system? Explain your answer.

Q2.6 (2 points)

How does the page size affect internal fragmentation? Explain your answer.

Question 3 (4 points)

Kaylia notices the following virtual page accesses are recorded for three independent processes P1, P2, and P3 respectively over a time interval.

P1: 0, 1, 0, 21, 1, 1, 4, 0, 0

P2: 91, 62, 1, 65, 55, 55, 62, 1, 91, 91, 62

P3: 45, 10, 3, 3, 5, 7, 28, 5

What is the **cumulative memory pressure** on the system during this time interval?

Question 4 (12 points)

Q4.1 (6 points)

Consider a direct-mapped cache with **5 sets**. Sets are indexed from 0 to 4. The cache block size is a **word**. Assume that the cache is initially empty and that each memory access operates at word granularity (i.e., all addresses are word addresses).

- a) For the memory accesses shown below starting from an empty cache, fill in the rows. If there is a miss, please specify the type of miss in the table. (3 points total)

Memory Address	Cache Set Accessed	Hit or Miss?
0x01		
0x04		
0x04		
0x05		
0x00		
0x04		

- b) For the memory accesses shown below starting from an empty cache, fill in the rows. If there is a miss, please specify the type of miss in the table. (3 points total)

Memory Address	Cache Set Accessed	Hit or Miss?
0x03		
0x04		
0x07		
0x03		
0x02		
0x04		

Q4.2 (6 points)

Now, consider a **3-way** set associative cache with 3 sets. The cache uses an LRU replacement policy. Sets are indexed from 0 to 2 and ways are numbered from 0 to 2. The cache block size is a word. Assume that the cache is initially empty and that each memory access operates at word granularity (i.e., all addresses are word addresses). If the cache set is empty, use cache way 0.

- c) For the memory accesses shown below starting from an empty cache, fill in the rows. If there is a miss, please specify the type of miss in the table. (3 points total)

Memory Address	Cache Set Accessed	Cache Way Accessed	Hit or Miss?
0x00			
0x03			
0x05			
0x06			
0x09			
0x00			

- d) For the memory accesses shown below starting from an empty cache, fill in the rows. If there is a miss, please specify the type of miss in the table. (3 points total)

Memory Address	Cache Set Accessed	Cache Way Accessed	Hit or Miss?
0x04			
0x07			
0x05			
0x07			
0x04			
0x02			

Question 5 (10 points)

The CS 2200 TA team is tasked with designing a 4KB (1KB = 1024 bytes) cache. (Note: Only descriptive answers are expected for this question)

Q5.1 (3 points)

Nityam decides to design the cache with 256 sets and 16B cache blocks. What are some drawbacks of his design?

Q5.2 (3 points)

Ayush decides to design the cache with 256 ways and 16B cache blocks. What is another name for Ayush's cache design? Why might it not perform as well as Nityam's design?

Q5.3 (4 points)

Kaylia designs a direct mapped cache, but with 256B cache blocks. What are some of the advantages and disadvantages of this design compared to Nityam's cache?

Question 6 (11 points)

Prasad is an intern at Intel. His job is designing a **4-way** set-associative cache with the following characteristics:

- Total cache capacity = 128 KB (1KB = 1024 bytes).
- CPU uses 64-bit byte-addressable memory addresses.
- The cache block size is 64 bytes.
- The cache has one valid bit per cache block.
- The cache uses a write-back policy and uses one dirty bit per 8-byte word to minimize the amount of data that must be written back when an eviction occurs.
- The cache's replacement policy guarantees that the two most-recently used cache blocks will not be evicted. The MRU fields uniquely identify the two most recently used cache blocks in that cache set. For replacement, one of the remaining 6 ways is chosen at random.

However, Prasad never took CS 2200 and is thus completely lost! Help him by answering some questions to design the cache.

Q6.1 (2 points)

How many sets are in the cache?

Q6.2 (3 points)

How many bits are needed for the following (1 pt each):

- a) Block Offset
- b) Index
- c) Tag

Q6.3 (2 points)

How many dirty bits are needed per cache block?

Q6.4 (2 points)

How many MRU bits are needed per set?

Q6.5 (2 points)

How many bits are required in total for metadata storage (including MRU)? You don't need to compute the final sum.

Question 7 (9 points)

Consider a paged memory system with **5 physical frames** that the OS has reserved strictly for housing application pages.

Figure 1 shows a timeline of virtual page accesses with associated timestamps. For example, at timestamp 2, virtual page 10 is accessed.

Figure 1: Timeline of Page Accesses

Timestamp	Virtual Page Accessed
1	1
2	10
3	7
4	3
5	4
6	5
7	1
8	8
9	10

The memory system has a hardware component in the CPU called the Tracker. The Tracker records the access order of all the pages currently in memory. That is, entry 1 holds the most recently accessed page, entry 2 holds the second most recently accessed page, and the last entry holds the least recently accessed page.

Figure 2 shows the contents of the Tracker at timestamp 5. Its contents are [4, 3, 7, 10, 1]. The first entry in the Tracker is the most recently accessed page: page 4. The Tracker is accessible to the memory management system in the OS.

Figure 2: Contents of the Tracker at timestamp 5

Entry 1	Entry 2	Entry 3	Entry 4	Entry 5
4	3	7	10	1

Q7.1 (2 points)

Explain how the operating system can use the Tracker component to make page replacement decisions.

Q7.2 (2 points)

What are the contents of the Tracker at timestamp 9?

Q7.3 (3 points)

How many total page faults occur in total in this scenario of 9 page accesses. Assume that none of the pages are in physical memory in the beginning. Show your work for credit.

Q7.3 (2 points)

Would implementing a Tracker component be feasible for a memory system with thousands of physical frames? Explain why or why not.