

CS230-2023@IITB

Name: _____

Exam-III

22th November, 2023

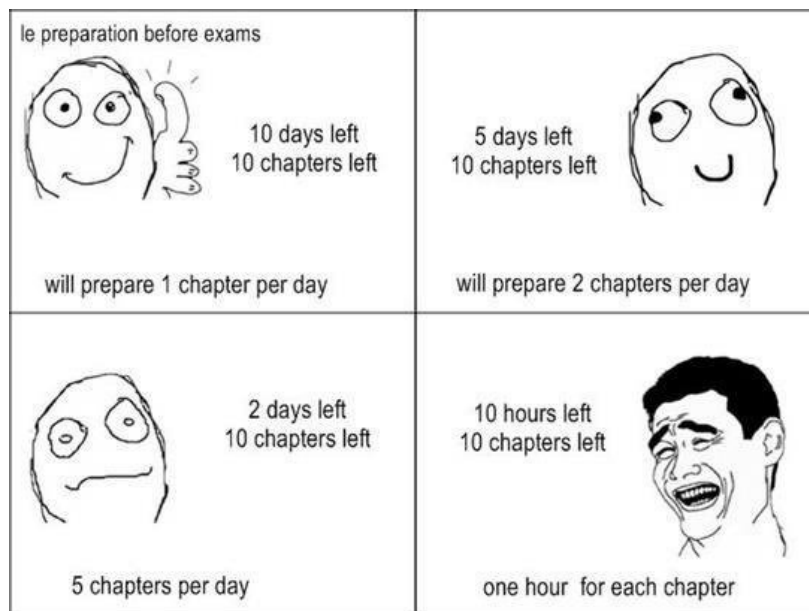
Time Limit: 120 Minutes

Roll No.: _____

Tips:

Be concise and cognizant.

There will be a penalty for verbosity and “it depends” without justification. Your logic and your understanding will not lead to marks unless your logic and understanding respect the design and implementation aspects of Computer Architecture. Do not spend too much or too little time on any particular question. Finally, show your work step-by-step.



“I promise I will write this exam honestly and ethically”. Your Signature:

Exxxxammmmm time!!

1. (48 points) [120 minutes]

- (1.1) (10 points) A processor with 500 MHz clock uses separate data and instruction caches at the first level and a unified second-level cache. The first-level data cache is a direct-mapped, write-through, writes-allocate cache with 8 Kbytes of data and 8-Byte blocks, and has a perfect write buffer (never causes any stalls). The first-level instruction cache is a direct-mapped cache with 4KBytes of data total and 8-byte blocks.

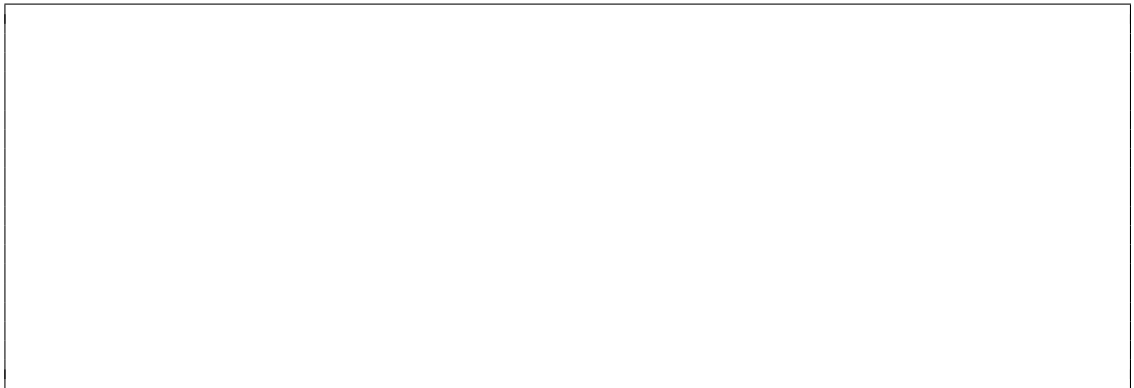
The second-level cache is a two-way set associative, write-back, write-allocate cache with 2MBytes of data total and 32-byte blocks. The first-level instruction cache has a miss-rate of 2%. The first-level data cache has a miss rate of 15%. The unified second-level cache has a local miss rate of 10% (i.e. the miss rate for all accesses going to the second-level cache). Assume that 40% of all instructions are data memory accesses; 60% of those are loads, and 40% are stores. Assume that 50% of the blocks in the second-level cache are dirty at any time. Assume that there is no optimization for fast reads on an L1 or L2 cache miss.

All first-level cache hits cause no stalls. The second-level hit time is 10 cycles (That means that the L1 miss-penalty, assuming a hit in the L2 cache, is 10 cycles). Main memory access time is 100 cycles to the first bus width of data; after that, the memory system can deliver consecutive bus widths of data on each following cycle. Outstanding, non-consecutive memory requests can not overlap; access to one memory location must be completed before access to another memory location can begin. There is a 128-bit bus from memory to the L2 cache and a 64-bit bus from both L1 caches to the L2 cache. Assume that the TLB never causes any stalls.

- (a) What fraction (in %) of all data memory references cause a main memory access (main memory is accessed before the memory request is satisfied)? [2.5 points]



(b) How many bits are used to index each of the caches? [2.5 points]



(c) What is the average memory access time in cycles (including instructions and data memory references)? Hint: don't forget to consider dirty lines in the L2 cache. [5 points]

(a) If you did not treat all stores as L1 misses:

$$= (\text{L1 miss rate}) \times (\text{L2 miss rate})$$

$$= (.15) \times (.1)$$

$$= 1.5\%$$

If you treated all stores as L1 misses:

$$= (\% \text{ of data ref that are stores}) \times (\text{L2 miss rate}) + (\% \text{ of data ref that are loads}) \times (\text{L1 miss rate}) \times (\text{L2 miss rate}) = (.4) \times (.1) + (.6) \times (.15) \times (.1) = 4.9\%$$

$$(b) \text{ Data} = 8K / 8 = 1024 \text{ blocks} = 10 \text{ bits}$$

$$\text{Inst} = 4K / 8 = 512 \text{ blocks} = 9 \text{ bits}$$

$$\text{L2} = 2M / 32 = 64k \text{ blocks} = 32k \text{ sets} = 15 \text{ bits}$$

(c) If you did not treat all stores as L1 misses:

$$\text{AMAT} = (\text{L1 hit time}) + (\text{L1 miss rate}) \times [(\text{L2 hit time}) + (\text{L2 miss rate}) \times (\text{mem transfer time})]$$

$$\text{AMAT}_{\text{inst}} = 1 + 0.02(10 + 0.10 \times 1.5 \times 101) = 1.503$$

$$\text{AMAT}_{\text{data}} = 1 + 0.15(10 + 0.10 \times 1.5 \times 101) = 4.7725$$

$$\text{AMAT} = 2.44$$

If you treat all stores as L1 misses:

AMAT = (L1 hit time) + (L1 miss rate) x [(L2 hit time) + (L2 miss rate) x (mem transfer time)]

AMATinst = 1 + 0.02(10 + 0.10 x 1.5 x 101) = 1.503

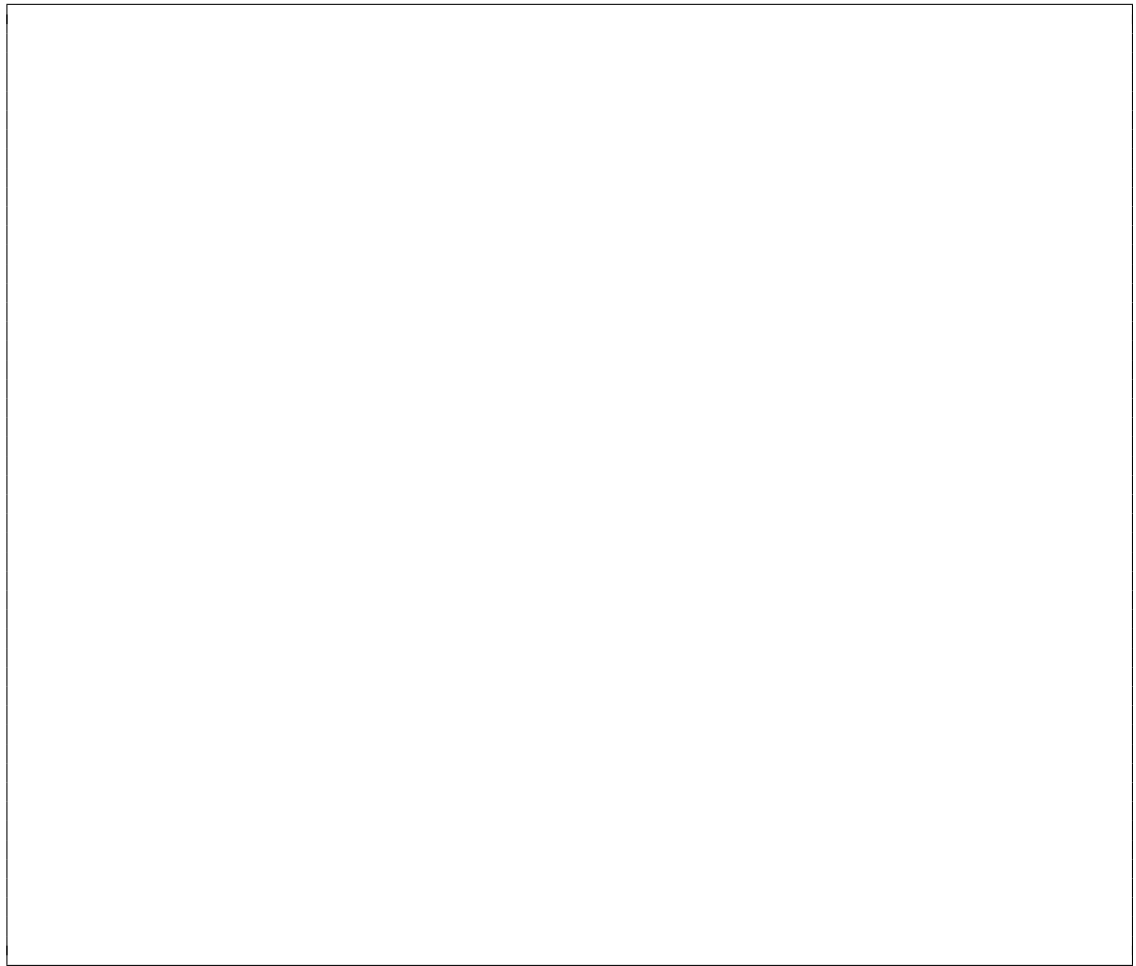
AMATloads = 1 + 0.15(10 + 0.10 x 1.5 x 101) = 4.7725

AMATstores = 1 + 1(10 + 0.10 x 1.5 x 101) = 26.15

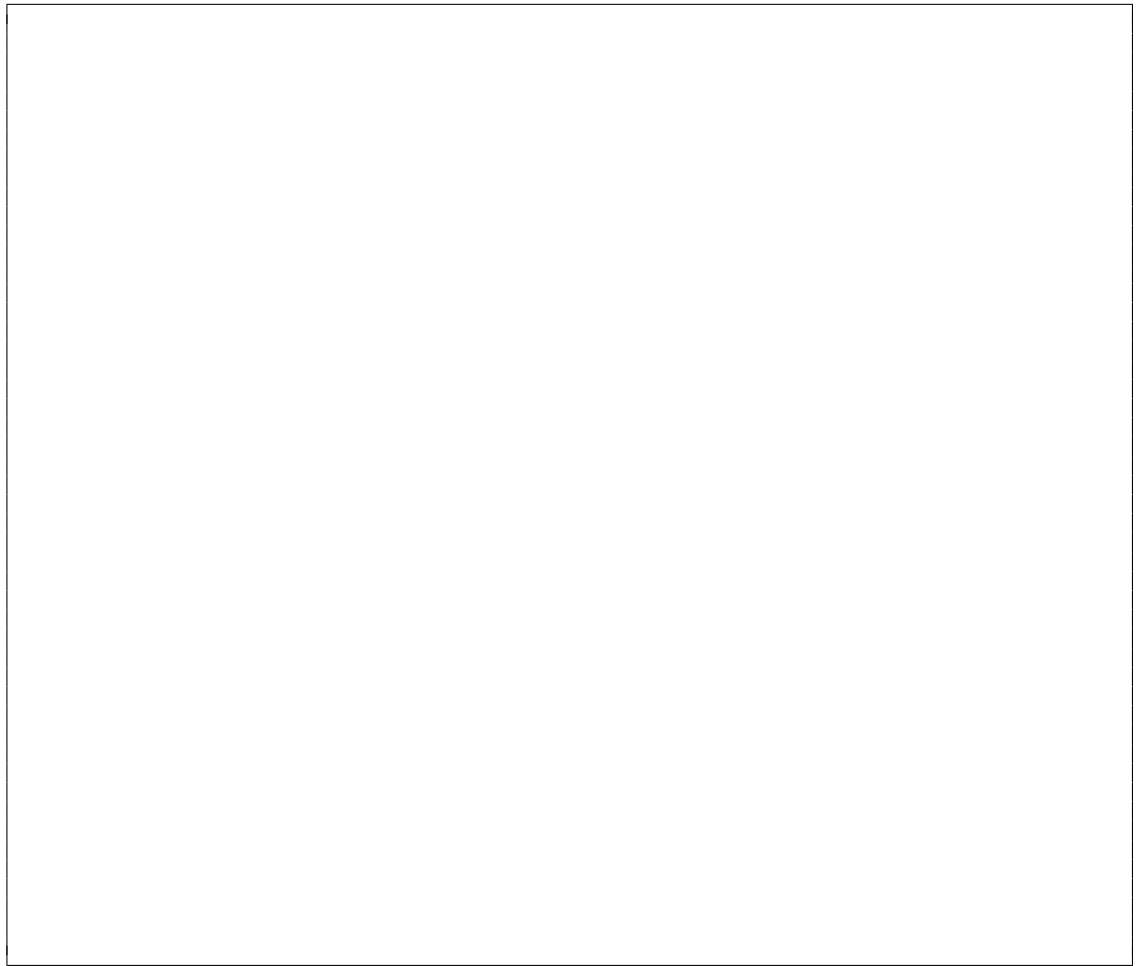
AMAT = 4.88

- (1.2) (10 points) CS230 is designing a next-gen low-power mobile processor codenamed Nano. You and your colleagues are tasked with designing the prefetcher for Nano. Nano has a single core, one level of cache, and a DRAM-based main memory system. You need to examine different prefetcher designs and analyze the trade-offs involved. For all parts of this question, you need to compute the coverage and accuracy. If there is a request to a cache block that has gone to main memory, a new request for the same cache block will not go to main memory as the outstanding request has not yet been completed. Instead the new request will be merged with the already outstanding request in the MSHR. You run an application **oyeoye** that has the following memory access pattern (note that these are cache block addresses): A, A+1, A+2, A+7, A+8, A+9, A+14, A+15, A+16, A+21, A+22, A+23, ...

(a) You first design a stride prefetcher that observes the last three cache block requests. If there is a constant stride S between the last three requests, the prefetcher issues a prefetch to the next cache block using the stride S. In the absence of a constant stride, the prefetcher refrains from prefetching. What is the coverage and accuracy of your stride prefetcher for **oyeoye**? [5 points]



(b) Your colleague designs a new prefetcher that, on a cache block access, prefetches the next N cache blocks. The coverage and accuracy of this prefetcher are 66.67% and 50% respectively for **oyeoye**. What is the value of N ? [2.5 points]



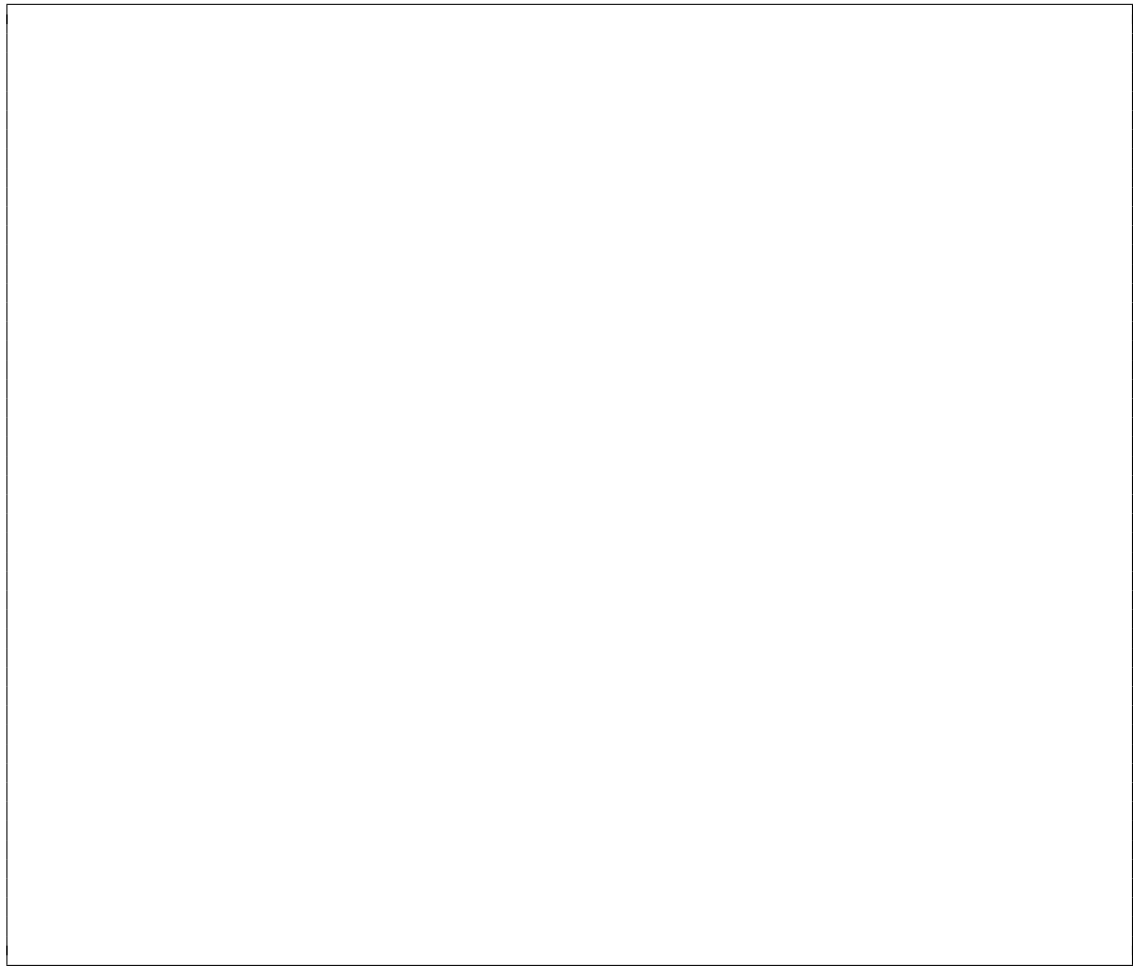
(c) What is the minimum value of N required to achieve a 100% prefetch coverage after the prefetcher reaches the steady state? [2.5 points]



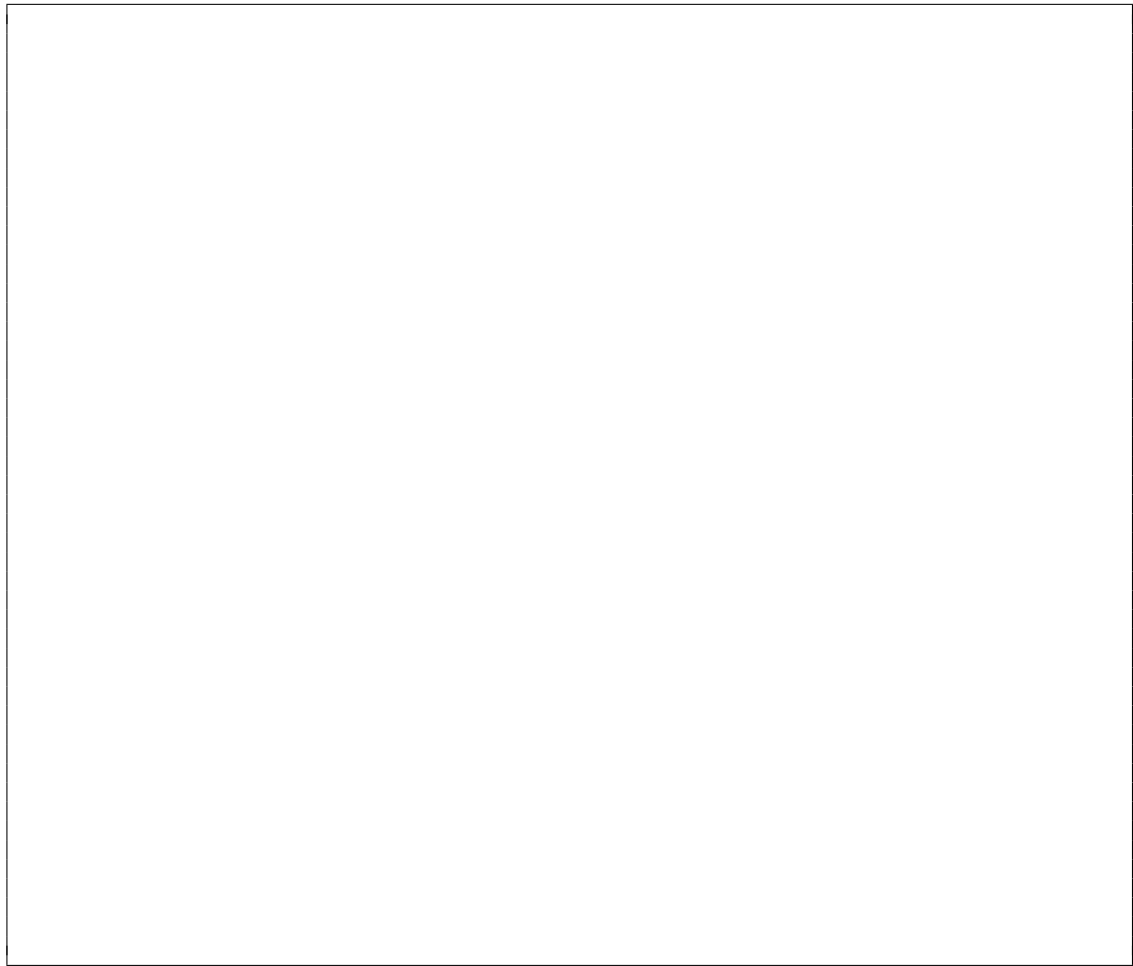
(a) zero and zero, (b) 2, (c) 5

- (1.3) (10 points) A memory system is composed of eight banks, and each bank contains 32K rows. Every DRAM row refresh is initiated by a command from the memory controller, and it refreshes a single row. Each refresh command keeps the command bus busy for 5 ns. We define command bus utilization as the fraction of the total execution time during which the command bus is occupied.

(a) Given that the refresh interval is 64ms, calculate the command bus utilization of refresh commands. [5 points]



(b) If 60% of all rows can withstand a refresh interval of 128 ms, how does the command bus utilization of refresh commands change? Calculate the reduction in bus utilization. [5 points]



(a) Command bus is utilized for $8 \times 2 \text{ pow}(15) \times 5\text{ns}$ at every 64ms.
 Utilization = $(218 \times 5\text{ns}) / (26 \times 106\text{ns}) = 212 / (2 \times 10 \text{ pow}(5)) = 211 \times 10 \text{ pow}(5) = 2.048\%$

(b) At every 128 ms: 60% of the rows are refreshed once. Command bus is busy for: $0.6 \times 8 \times 2 \text{ pow}(15) \times 5\text{ns} = 3 \times 2 \text{ pow}(18) \text{ ns}$

40% of the rows are refreshed twice. The command bus is busy for: $0.4 \times 8 \times 2 \text{ pow}(15) \times 5\text{ns} \times 2 = 4 \times 2 \text{ pow}(18) \text{ ns}$

Utilization = $(3 + 4) \times 2 \text{ pow}(18) \text{ ns} / 128\text{ms} = 0.7 \times 2 \text{ pow}(11) \times 10 \text{ pow}(5)$

Reduction = $1 - (0.7 \times 2 \text{ pow}(11) \times 10 \text{ pow}(5)) / (211 \times 10 \text{ pow}(5)) = 30\%$

- (1.4) (10 points) A processor that uses Tomasulo organization (in-order issue, out-of-order execute and in-order commit with precise exception handling) executes the following instructions with the respective execution latencies. LD F6 34 R2

LD F2 45 R3

MULT F0 F2 F4

SUB F8 F6 F2

DIV F10 F0 F6

ADD F6 F8 F2

LOAD and ADD/SUB take two cycles, MULT takes 10 cycles, and DIV takes 40 cycles. There are three reservation stations for ADD/SUB, and two reservation stations for MULT/DIV. Compute the total number of cycles needed to write the result for all the instructions (show it individually for each instruction) if the first LD is issued on cycle 1. Hint: Just consider ISSUE, EXECUTION COMPLETE, and WRITE RESULT as the stages of interest. [10 points]

4, 5, 16, 17, 57, 58, many of you have answered assuming out of order commit, we will give partial marks based on that

- (1.5) (8 points) For a 4-core system, the following access stream captures the reads and writes to one cache line generated by core-1 to core-4. For an MSI coherence protocol, show the coherence states (fill in the table) at all four cores after each read and write. R1/W1 stands for a read/write request generated by core1. [8

points]

Read/write	State@Core-1	State@Core-2	State@Core-3	State@Core-4
W1				
W1				
R1				
R2				
W3				
W4				
R2				
W4				

M, M, S, M, M, S, M

Ta ta, bye bye, khatam exam !!

Rough sheet

Rough sheet