# Discussion 10

March 12

# Outline

- Assignment 5 questions
- Overview some of the topics covered after Midterm
- Last week’s quiz!

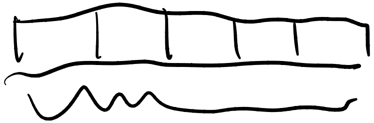
# Cache organizations

offset = $\log_2$ (cache block size)

- Cache types
  - Fully associative, N-way associative, directed-mapped
- Cache parameters
- AMAT
- AMAT with TLBs

HT + MP.

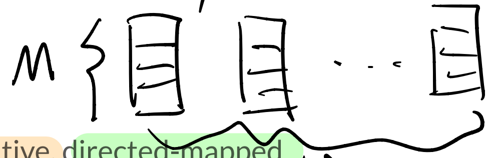
fully associate.



N entries.

PAddr | tag | offset.

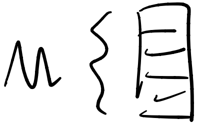
N-way associate



N ways.

| tag | index | offset. |

directed map



| tag | index | offset. |

# bits for offset = $\log_2$(page size).

# Virtual Memory

- Benefits
  - Process isolation
- Overheads
  - Translation cost
- Physical Address / Virtual Address
- Address translation
- Paging, page tables, page walking
- TLB
- TLB organization
- TLB miss

n times of memory access which n is the # of levels of the page table.

VAddr | VPN | offset |

# levels of the page table. → | VPN0 | VPN1 | VPN2 | ... | offset |

PAddr | PPN | offset |

SATP → first level page table base addr.

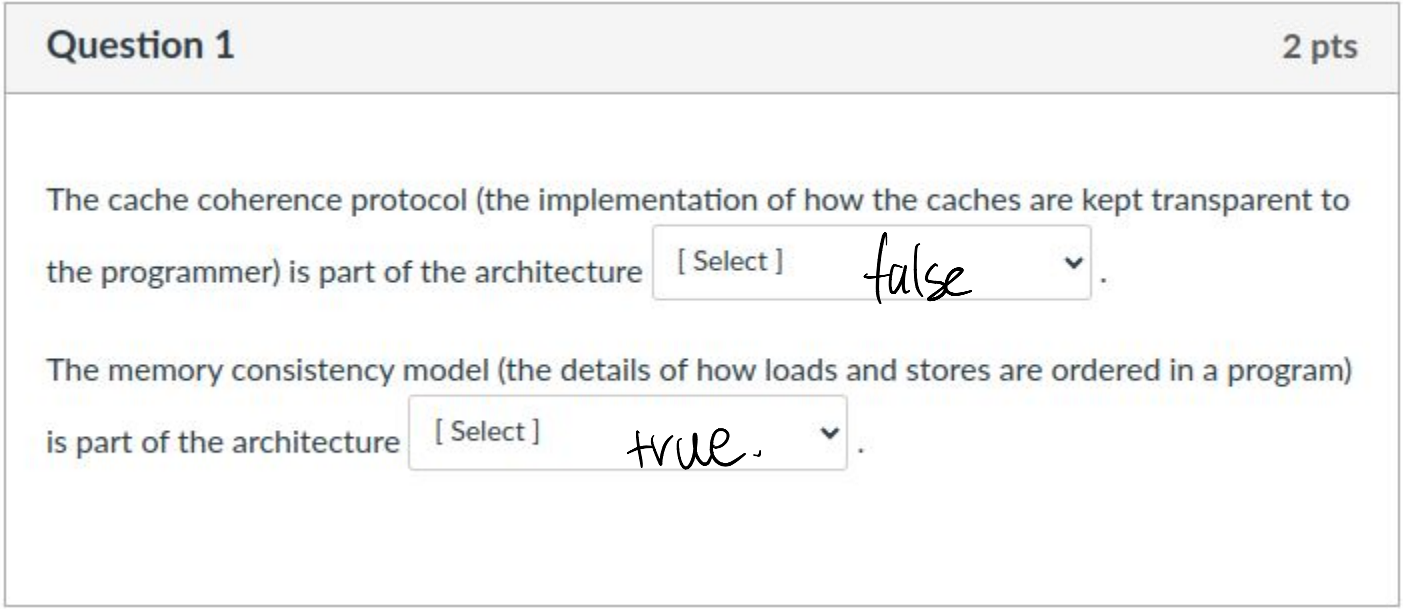
VPN0 → VPN1 → ... → PPN.

| tag | data |
|---|---|
| VPN | PPN |
| | |

| | | |
|---|---|---|
| SISD | SIMD | DLP |
| MISD | MIMD | TLP |

# Parallel Programing/Parallel Architectures

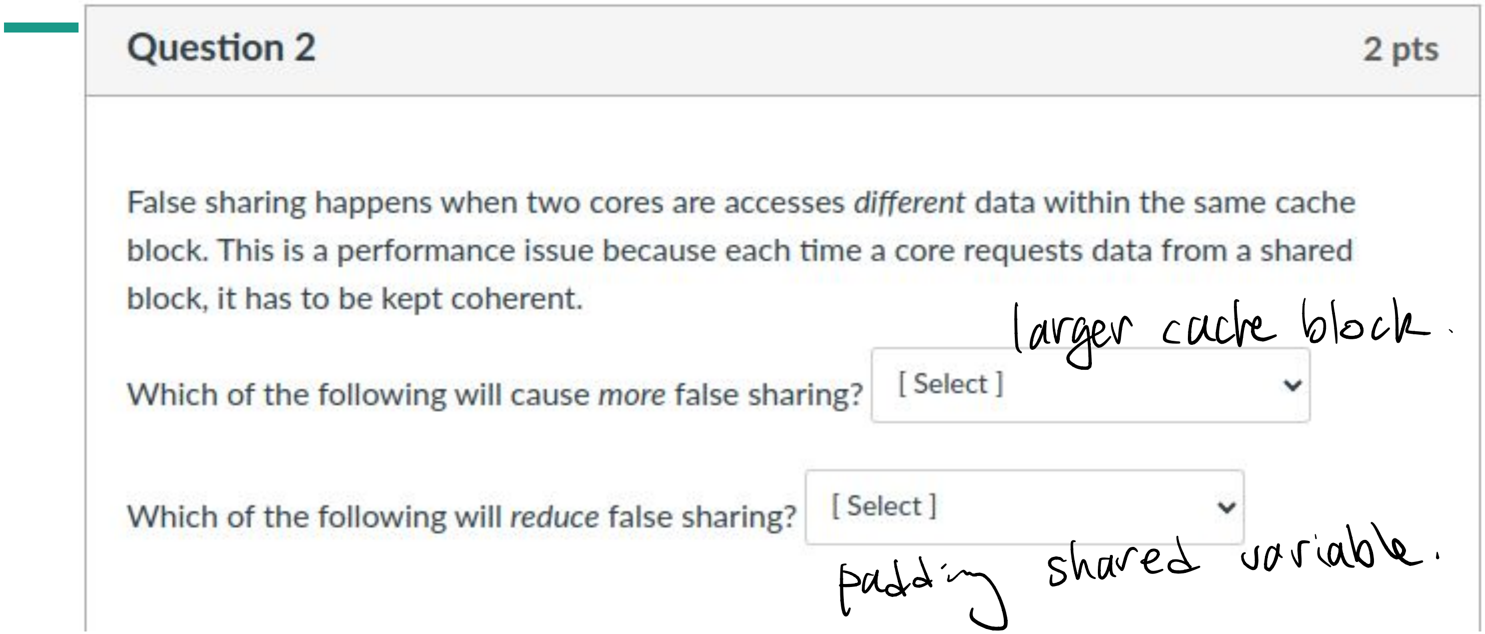
- Flynn’s taxonomy
- Parallel processing and synchronization problem
- Cache coherence/False sharing
- Memory consistency/Sequential consistency
- Accelerators: GPUs, etc.
- Shared memory/Message Passing
- Level of parallelism:
    - ILP, DLP, TLP
- Amdahl’s law



**Question 1** 2 pts

The cache coherence protocol (the implementation of how the caches are kept transparent to the programmer) is part of the architecture [ Select ] false .

The memory consistency model (the details of how loads and stores are ordered in a program) is part of the architecture [ Select ] true. .



## Question 2

2 pts

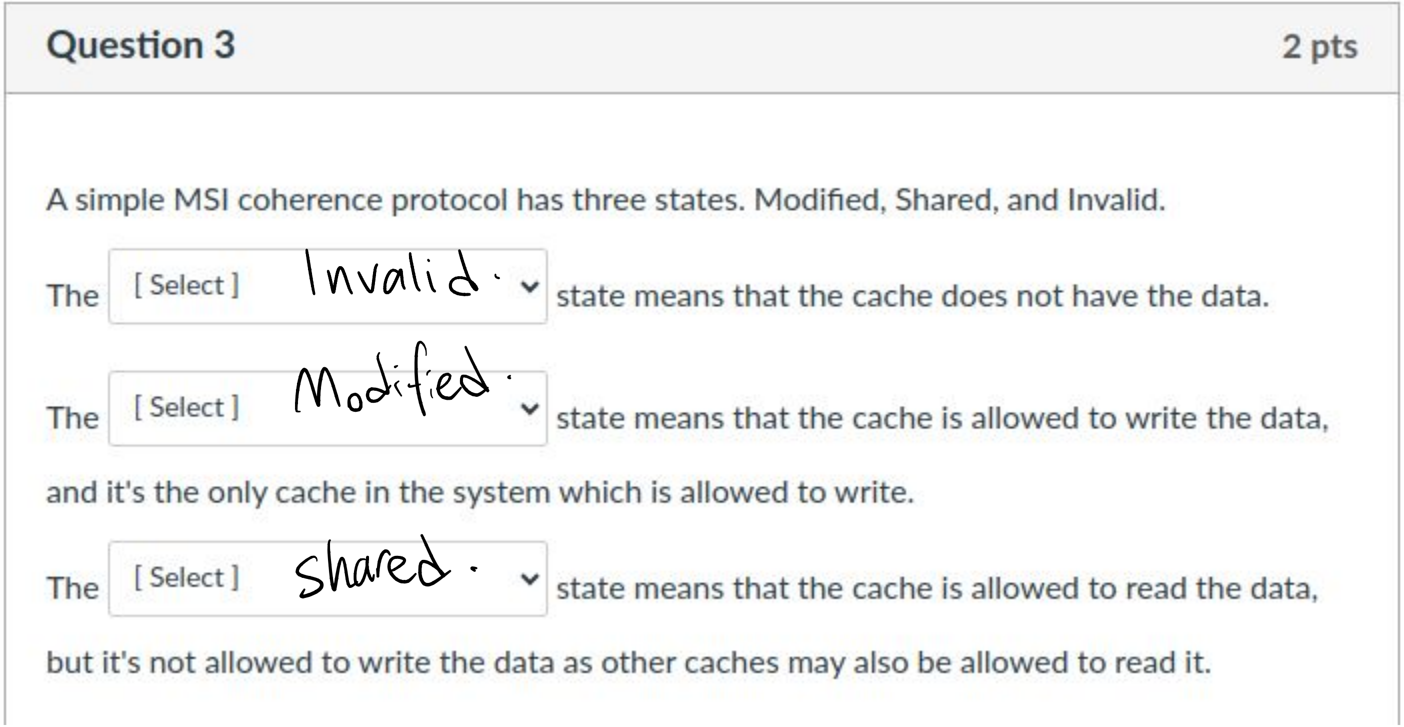
False sharing happens when two cores are accesses *different* data within the same cache block. This is a performance issue because each time a core requests data from a shared block, it has to be kept coherent.

Which of the following will cause *more* false sharing? [ Select ]

larger cache block.

Which of the following will *reduce* false sharing? [ Select ]

padding shared variable.



## Question 3

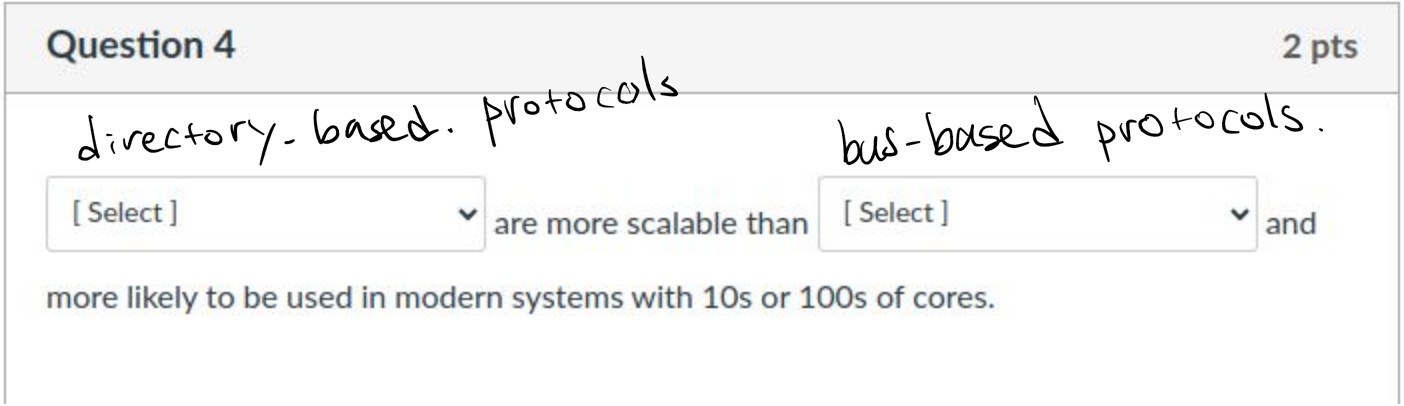
2 pts

A simple MSI coherence protocol has three states. Modified, Shared, and Invalid.

The [ Select ] Invalid. state means that the cache does not have the data.

The [ Select ] Modified. state means that the cache is allowed to write the data, and it's the only cache in the system which is allowed to write.

The [ Select ] Shared. state means that the cache is allowed to read the data, but it's not allowed to write the data as other caches may also be allowed to read it.



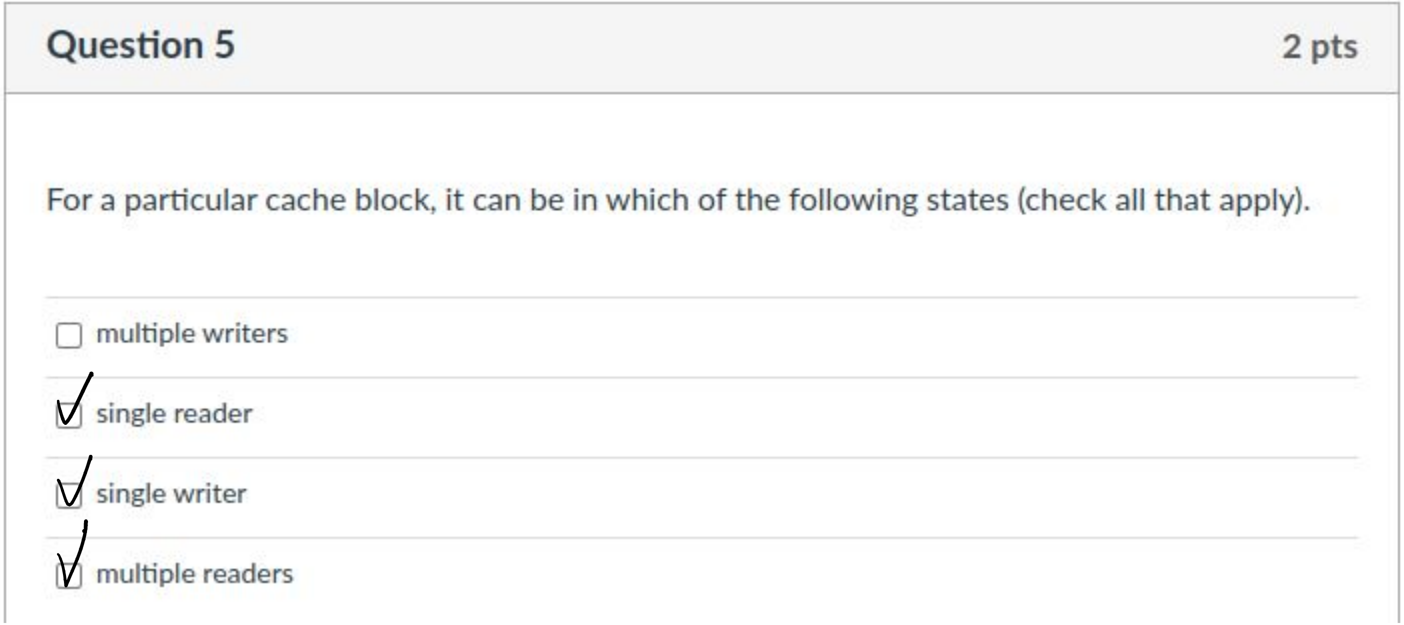
## Question 4

2 pts

directory-based. protocols

bus-based protocols.

[ Select ] ⌄ are more scalable than [ Select ] ⌄ and more likely to be used in modern systems with 10s or 100s of cores.



## Question 5

2 pts

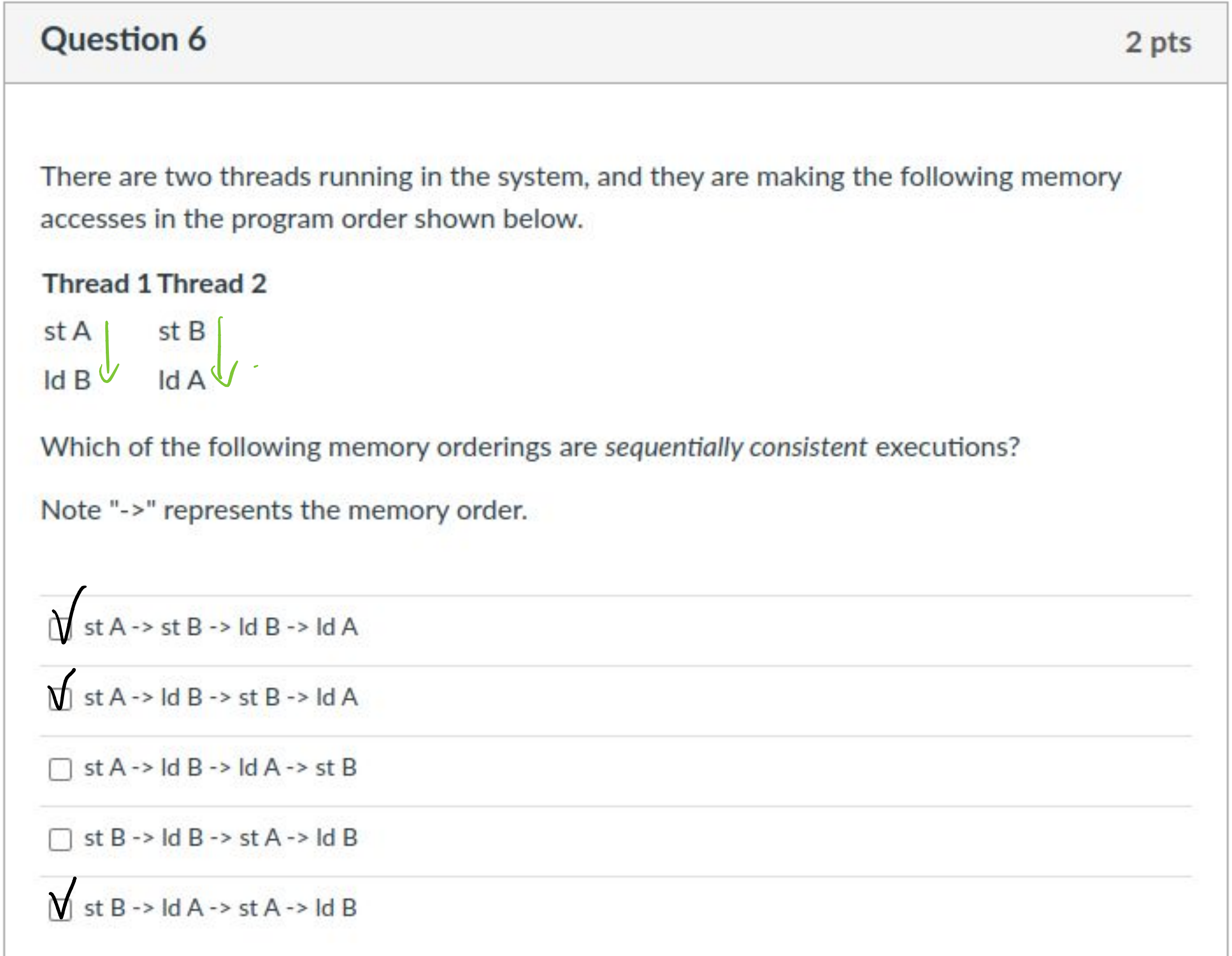
For a particular cache block, it can be in which of the following states (check all that apply).

□ multiple writers

☑ single reader

☑ single writer

☑ multiple readers



## Question 6

2 pts

There are two threads running in the system, and they are making the following memory accesses in the program order shown below.

| **Thread 1** | **Thread 2** |
|---|---|
| st A | st B |
| ld B | ld A |

Which of the following memory orderings are *sequentially consistent* executions?

Note "->" represents the memory order.

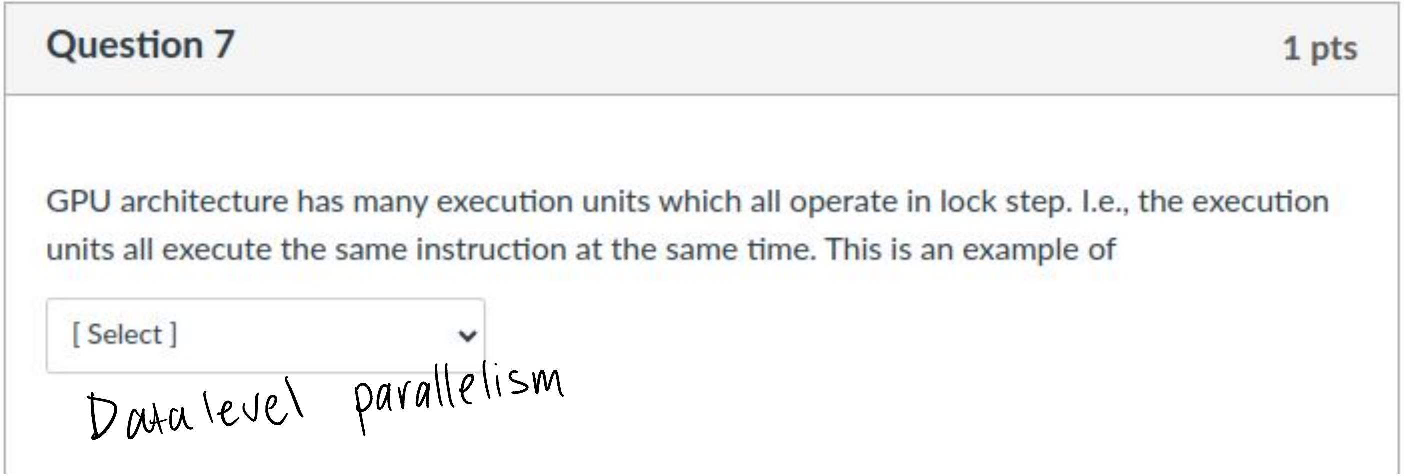
☑ st A -> st B -> ld B -> ld A

☑ st A -> ld B -> st B -> ld A

□ st A -> ld B -> ld A -> st B

□ st B -> ld B -> st A -> ld B

☑ st B -> ld A -> st A -> ld B



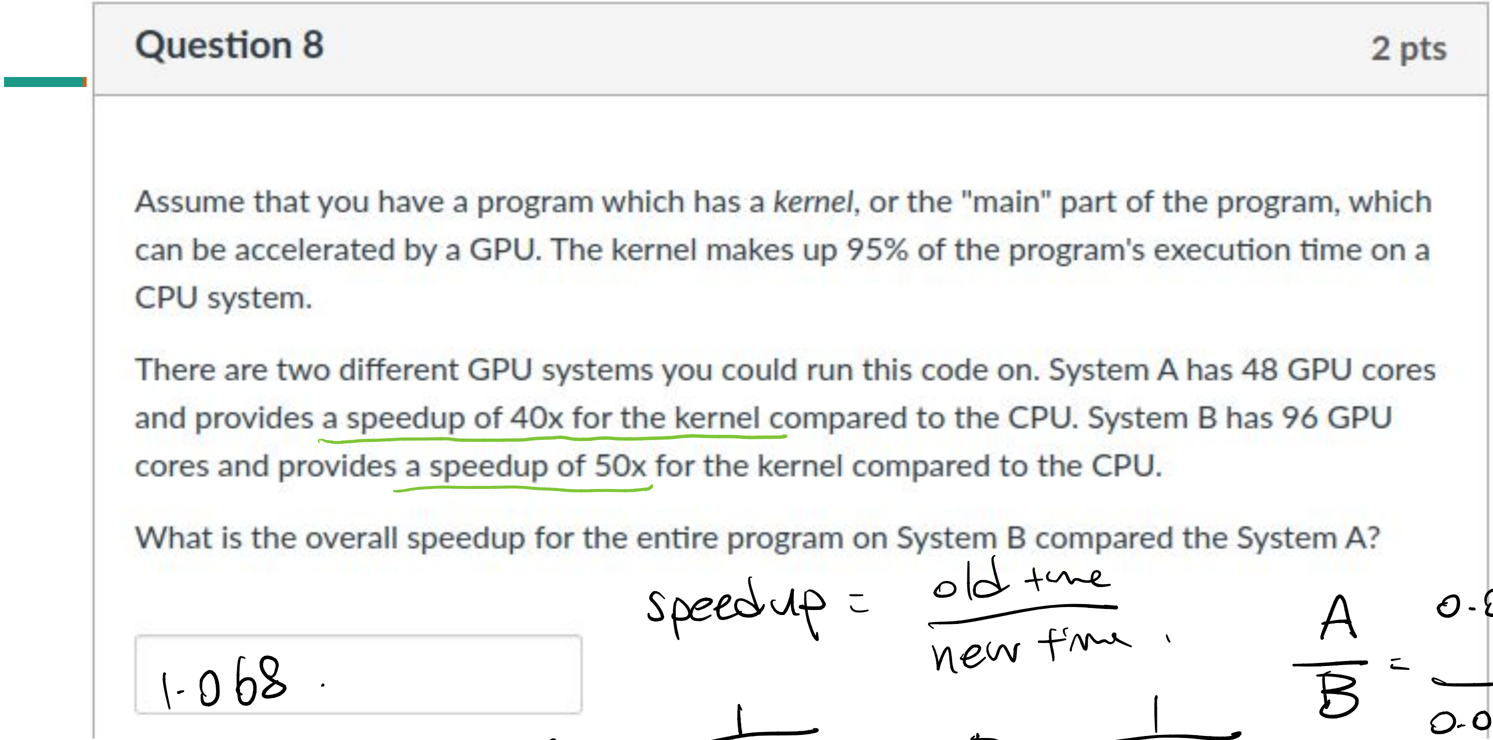
## Question 7

1 pts

GPU architecture has many execution units which all operate in lock step. I.e., the execution units all execute the same instruction at the same time. This is an example of

[ Select ] ⌄

Data level parallelism



## Question 8

2 pts

Assume that you have a program which has a *kernel*, or the "main" part of the program, which can be accelerated by a GPU. The kernel makes up 95% of the program's execution time on a CPU system.

There are two different GPU systems you could run this code on. System A has 48 GPU cores and provides a speedup of 40x for the kernel compared to the CPU. System B has 96 GPU cores and provides a speedup of 50x for the kernel compared to the CPU.

What is the overall speedup for the entire program on System B compared the System A?

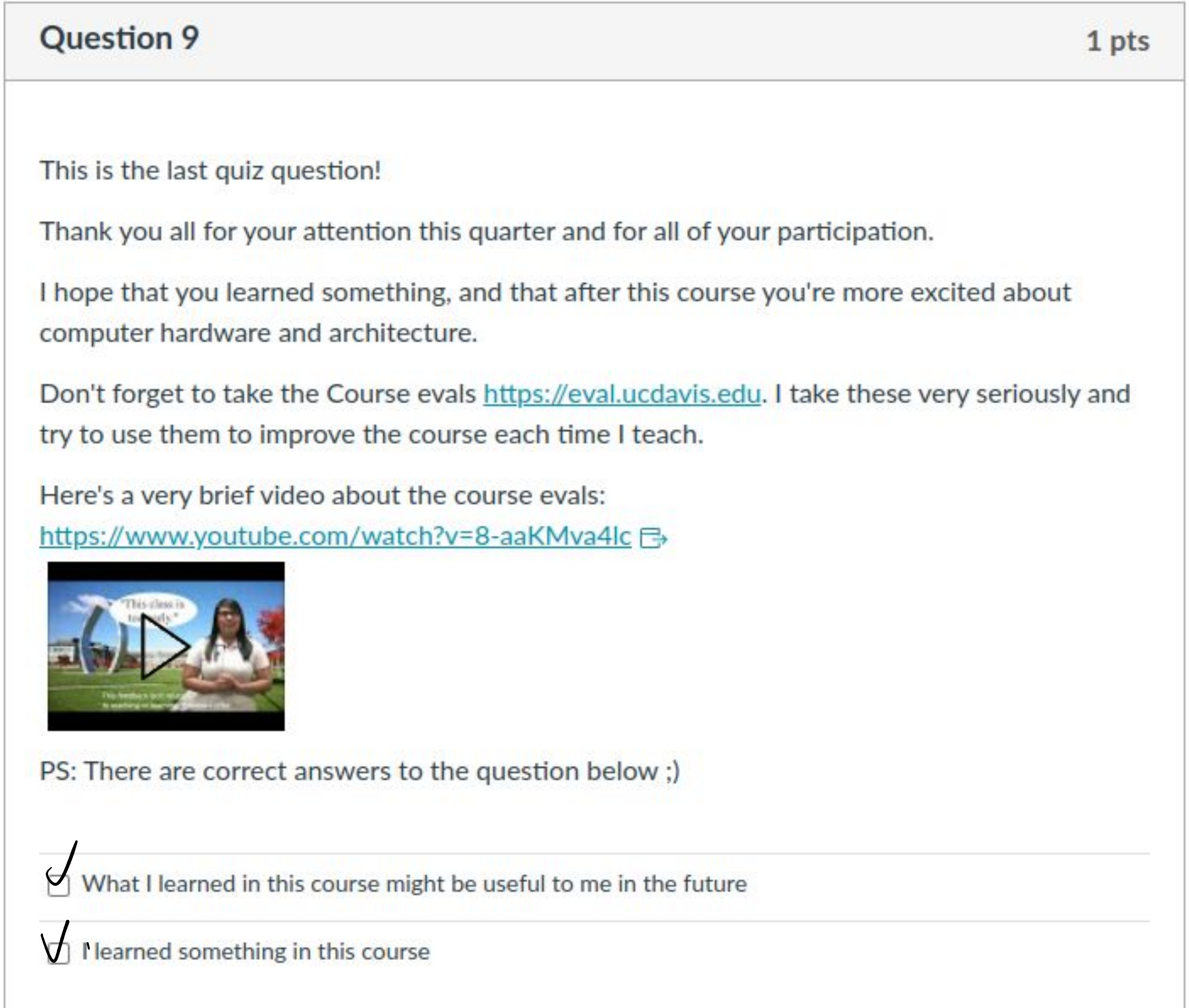
1.068.

$$\text{speedup} = \frac{\text{old time}}{\text{new time}}.$$

$$A : \frac{1}{0.05 + \frac{0.95}{40}} \qquad B : \frac{1}{0.05 + \frac{0.95}{50}}.$$

$$\frac{A}{B} = \frac{0.05 + \frac{0.95}{50}}{0.05 + \frac{0.95}{40}}$$

$$= 1.068.$$



## Question 9

1 pts

This is the last quiz question!

Thank you all for your attention this quarter and for all of your participation.

I hope that you learned something, and that after this course you're more excited about computer hardware and architecture.

Don't forget to take the Course evals https://eval.ucdavis.edu. I take these very seriously and try to use them to improve the course each time I teach.

Here's a very brief video about the course evals: https://www.youtube.com/watch?v=8-aaKMva4lc

PS: There are correct answers to the question below ;)

☑ What I learned in this course might be useful to me in the future

☑ I learned something in this course

GOOD LUCK ON FINAL WEEK!!