HW10: Cache Coherence
CSE 30321 Computer Architecture
Names: Aaron Wang and Ethan LIttle

Important reminder: homework assignments are weighted equally (35% total/10 assignments = 3.5% each). This assignment is short, but worth the same as other assignments – use this (and the small extra credits) as an opportunity to boost your Homework grade as we enter finals week. (If you end up with > 35% in the Homework category ***that will count, it is not capped****.)*

Preamble:

1. ***Copy this assignment from our Google Drive directory, not from Gradescope. Downloading the PDF from Gradescope may not preserve all answer boxes.***

2. Enter your answers in the boxes provided. Each empty box should be filled in with an answer. You can either type your answers directly or insert images as long as they are legible. For solutions that require code, use a **fixed-width font (Consolas preferred) no smaller than 10 points with proper indentation**. Save your solutions as a single PDF file and upload them to Gradescope.

3. You are encouraged to work in groups of 2. Please type your names at the top of this document.

Problem 1: Cache Coherence

Consider the multiprocessor cache state shown in the tables below.

P0's Cache:

Index	Coherence State	Tag	Word 1	Word 0
0	M	400	–	--
1	E	500	–	–
2	S	600	–	–
3	S	700	–	–

P1's Cache:

Index	Coherence State	Tag	Word 1	Word 0
0	S	000	–	–
1	S	100	–	–
2	S	600	–	–
3	E	1100	–	–

P2’s Cache:

Index	Coherence State	Tag	Word 1	Word 0
0	S	000	–	–
1	S	100	–	–
2	M	1000	–	–
3	S	700	–	–

Instructions:
For each question below, describe a request that results in the listed coherence action. Explain what the request is, any generated bus traffic, and the effect(s) on the other processors and memory (if any). Write your answers as a bulleted list, you do not need to update the tables above. Treat each question **independently** - as you begin a new question, use the initial state given above.

Note that in some cases, some cache blocks in a given processor may not change at all, no bus traffic may be generated, etc. You should assume a MESI protocol, a centralized shared memory machine, a write invalidate protocol, and write-back caches for this problem – this is the same setup in our examples from class. As in class, the “–” indicates there is some data in each word of the 2-word block, but we don’t care what that data is for this problem. Finally, assume there are **no other processors in the system**.

Question 1
Describe a request that will result in a **read hit**. List the request in this format: “PN requests data via a lw instruction. The address maps to index X, tag Y.” Where N = 0, 1 or 2.

Request that generates the read hit:
P0 requests data via a lw instruction. The address maps to index 1, tag 500.

Bus traffic generated on the read hit:
None

Changes to other caches and memory:
None

Question 2
Describe a request that will result in a **read miss** that is satisfied **from memory**. List the request in this format: “PN requests data via a lw instruction. The address maps to index X, tag Y.” Where N = 0, 1 or 2.

Request that generates the read miss satisfied from memory:
P2 requests data via a lw instruction. The address maps to index 2, tag 200.

Bus traffic generated on the read miss:
Read Miss → request I2T200 on bus; P0 and P1 do not have so memory provides.

Changes to other caches and memory:
P2 sets the state to E (exclusive). No other caches change.

Question 3
Describe a request that will result in a **read miss** that is satisfied **from another processor**. List the request in this format: "PN requests data via a lw instruction. The address maps to index X, tag Y." Where N = 0, 1 or 2.

Request that generates the read miss satisfied from another processor:
P2 requests data via a lw instruction. The address maps to index 0, tag 400
Bus traffic generated on the read miss:
Read Miss → request I0T400 on bus; P0 provides the block to P2.

Changes to other caches and memory:
P0 sets the state to S (Shared). P2 sets the state to S (Shared). Memory gets a copy.

Question 4
Describe a request that will result in a **write miss** (can be satisfied from anywhere). List the request in this format: "PN requests data via a sw instruction. The address maps to index X, tag Y." Where N = 0, 1 or 2.

Request that generates the read miss satisfied from memory:
P2 requests data via a sw instruction. The address maps to index 2, tag 200.

Bus traffic generated on the read miss:
Write Miss → request I2T200 on bus

Changes to other caches and memory:
Memory provides the block to P2. P2 sets the state to M (Modified). Sends broadcast for I but no one cares.

Question 5
Describe a request that will result in a **write hit**. List the request in this format: “PN requests data via a sw instruction. The address maps to index X, tag Y.” Where N = 0, 1 or 2.

Request that generates the read miss satisfied from another processor:
P1 requests data via a sw instruction. The address maps to index 0, tag 000
Bus traffic generated on the read miss:
Write Hit → P1 broadcast I for I0T000

Changes to other caches and memory:
P2 S → I; P1 → M

What to Turn In
Fill in the boxes on the previous pages with answers to all the questions. Save your Google Doc as a PDF and upload it to **Gradescope** for grading. Note Gradescope can be accessed via our Canvas site, or by visiting gradescope.com. Below is a checklist for this assignment:

Problem 1 (50 points)

	Deliverable	Points
1.	10 points each for Q1-Q5	50