Submission Instructions:

1. Prepare a PDF document with your solutions to the questions below. You may use the LaTeXtemplate that is uploaded to Canvas, or any other tool. Please do not submit handwritten solutions since it will be hard to grade.
2. Ensure that each answer is clearly marked, and starting on a new page. This includes subparts of a question (e.g., Q2.1 on P3 and Q2.2 on P4, or Q1 on P1-2, Q2.1 on P3-4 and so on). Your answer may use multiple pages, but ensure that no single page has multiple answers.
3. Include the following acknowledgement on the first page of your submission, and sign it with your full name and date:

   "I affirm that the work I am submitting is my own. I have not copied or used any portion of another student's work. I have not used external sources without proper acknowledgement. If I have used AI tools, I will acknowledge the version used and explain how I used it."

   Full Name:

   Date:
4. Submit the PDF on Gradescope, matching each question to the corresponding page.

1 Cache Writing [5]

You purchased a computer with the following features:

- 95% of all memory accesses (i.e., reads or writes) are found in the cache.
- Each cache block is two words, and the whole block is read on any miss.
- The processor sends references to its cache at the rate of 10^9 words per second.
- 25% of those references are writes.
- Assume that the memory system can support 10^9 words per second, reads or writes.
- The memory bus transfers one word at a time. Therefore, moving a two-word cache block requires two word transfers. (And moving a word requires a one word transfer.)
- Assume at any one time, 30% of the blocks in the cache have been modified.
- The cache uses write allocate on a write miss.
- A word is 4 Bytes.

You are considering adding a peripheral to the system, and you want to know how much of the memory system bandwidth is already used. Calculate the percentage of memory system bandwidth used on the average in the two cases below. **Be sure to state your assumptions.**

(a) The cache is write-through.

(b) The cache is write-back.

2 Encore: More Cache Writing [5]

One difference between a write-through cache and a write-back cache can be in the time it takes to write. In this problem, we will compare the performance of both policies.

System Specification

You are given a processor with the following characteristics:

- A base CPI of 1 (assuming perfect caches).
- Instruction cache miss rate: 0.5%.
- Data cache miss rate: 1%.
- The workload consists of 26% loads and 9% stores.
- Cache read hits take 1 clock cycle, and **cache write hits take 2 clock cycles.**
- The cache miss penalty (for both reads and writes) is 50 clock cycles.
- A block write from the cache to main memory (for write-backs) takes 50 clock cycles.
- For a write-back cache, 50% of the blocks are dirty on average.
- **Both caches use a write-allocate policy on a write miss.**
- For the write-through cache, a write buffer is used to send data to main memory. You can assume **this buffer never fills up, so there are no *additional* stalls from this process.**

Task

Compare the performance of the two data cache designs specified below. To determine which performs better, **calculate the final CPI for each system.** State any assumptions you make.

(a) A system with a **write-through** data cache.

(b) A system with a **write-back** data cache.

3 Cache Inclusion Analysis [6]

Consider a two-level cache hierarchy (L1 and L2). Both caches use a Least Recently Used (LRU) replacement policy, if applicable.

(a) Direct-Mapped Caches with Different Block Sizes

Let the cache parameters be:

- **L1 Cache:** Direct-mapped ($a_1 = 1$), 4 sets ($n_1 = 4$), 4-byte block size ($b_1 = 4$).
- **L2 Cache:** Direct-mapped ($a_2 = 1$), 8 sets ($n_2 = 8$), 8-byte block size ($b_2 = 8$).

For these parameters, is the L2 cache guaranteed to be inclusive of the L1 cache? If you answer yes, provide a rigorous argument. If you answer no, provide a specific sequence of memory address accesses that serves as a counter-example and explain exactly how it breaks the inclusion property.

(b) Set-Associative Caches with Same Block Size

Now, let the cache parameters be:

- **L1 Cache:** 2-way set-associative ($a_1 = 2$), 4 sets ($n_1 = 4$), 8-byte block size ($b_1 = 8$).
- **L2 Cache:** 2-way set-associative ($a_2 = 2$), 8 sets ($n_2 = 8$), 8-byte block size ($b_2 = 8$).

For these parameters, is the L2 cache guaranteed to be inclusive of the L1 cache? If you answer yes, provide a rigorous argument. If you answer no, provide a specific sequence of memory address accesses that serves as a counter-example and explain exactly how it breaks the inclusion property.