CSE 240A: Principles of Computer Architecture

Midterm Examination

Fall 2021

1	6	
2	8	
3	8	
4	8	
5	15	
Total	**45**	

Instructions This exam is open book and open notes. Personal calculators *are* allowed. Show your work and insert your answer in the space(s) provided. **Please provide details on how you reach a result unless directed by the question as not to.**

The exam totals 45 points. This exam counts for 45% of your course grade. Please submit typed answers to the following questions as a PDF via Gradescope by **Friday, November 3, 2021 at 11:59 PM**. It would be great if you can finish the midterm by typing rather than handwriting.

1 Performance Evaluation (6 points)

When making changes to optimize part of a processor, it is often the case that speeding up one type of instruction comes at the cost of slowing down something else. For example, if we put in a complicated fast floating-point unit, that takes space, and something might have to be moved farther away from the middle to accommodate it, adding an extra cycle in delay to reach that unit. The basic Amdahl's law equation does not take into account this trade-off.

A If the new fast floating-point unit speeds up floating-point operations by, on average, 2X, and floating-point operations take 20% of the original program's execution time, what is the overall speedup (ignoring the penalty to any other instructions)?

Please insert your solutions here.

B Now assume that speeding up the floating-point unit slowed down data cache accesses, resulting in a 1.5X slowdown (or 2/3 speedup). Data cache accesses consume 10% of the execution time. What is the overall speedup now?

Please insert your solutions here.

2 Cache Performance (8 points)

Assuming the base CPI_{base} (no stall) of a pipeline is 1. A program has 25% of load/store instructions. The processor only has one level of cache, i.e., an L1 instruction cache and an L1 data cache. Cache miss rate and penalty are as following:

- L1 instruction cache: $\%_{miss} = 2\%, t_{miss} = 100$ cycles
- L1 data cache: $\%_{miss} = 30\%, t_{miss} = 100$ cycles

What is the CPI?

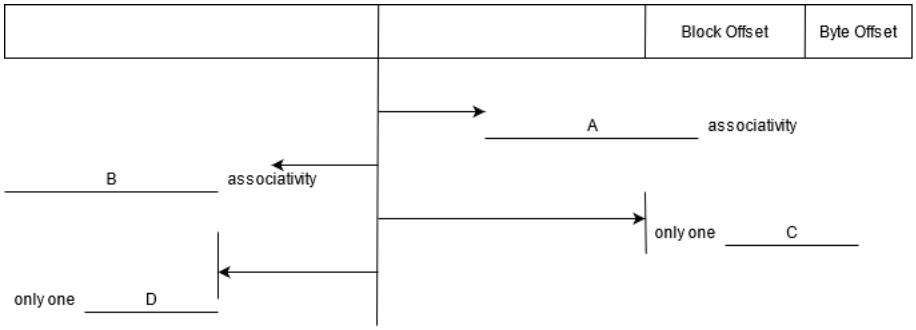
Please insert your solutions here.

3 Caches and Memory Hierarchy

A Allow cache and memory to be inconsistent, i.e., write the data **only** into the cache block. Is this write-back or write through? *insert your answer here* (No further explanation needed)

B Require cache and memory to be consistent, i.e., always write the data into both the cache block and the next level in the memory hierarchy. Is this write-back or write through? *insert your answer here* (No further explanation needed)

C Consider the diagram below that shows the dividing line between the bits used for tag compare and those used to select the cache set. Fill in the lines indicating whether the associativity **increases** or **decreases** and whether the resulting cache has only one **way** or only one **set**. (No further explanation needed)



Please fill your answers to the corresponding blanks below:
A: *insert your answer here*
B: *insert your answer here*
C: *insert your answer here*
D: *insert your answer here*

4 Cache tag overhead (8 points)

Assume:

- A processor has a 64KB 4-way set associative cache
- The cache access uses physical addresses only
- A physical address is 48 bits long
- Each block holds 64 bytes of data
- Tag overhead includes the valid bit and tag bits

How much is the tag overhead in percent?

Please insert your solutions here.

5 Pipelining (15 points)

Assume the following program is running on the 5-stage in-order pipeline processor shown in class. All registers are initialized to 0. Assuming only WX and WD (register file internal forwarding) forwarding, branches are resolved in **Decode** stage, and branches are always predicted **not-taken**. How many cycles will it take to execute the program, if the branch outcome is **actually taken**? Draw a pipeline diagram (table) to show the details of your work. Use arrows to indicate forwarding.

```
      lw $r6 0($r10)
      lw $r7 0($r11)
      add $r2 $r6 $r7
      beq $r2 $r3 label
      sub $r6 $r8 $r4
      sw $r6 0($r10)
label:lw $r1 0($r2)
      or $r4 $r2 $r1
```

Please insert your solutions here.