Submission Instructions:

1. Prepare a PDF document with your solutions to the questions below. You may use LaTeX(template), or any other tool.
2. Ensure that each answer is clearly marked, and starting on a new page. This includes subparts of a question (e.g., Q2(a) on P3 and Q2(b) on P4, or Q1 on P1-2, Q2(a) 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 honor code on the first page of your submission, and sign it with your full name and date:

   "I affirm that the work I am submitting for this assignment is entirely my own. I have not copied or used any portion of another student's work, external sources, or AI-generated tools without proper acknowledgement. I understand that academic honesty is essential to the integrity of my education, and I have adhered to the highest standards of academic conduct in completing this assignment."
   Full Name: _____
   Date: _____

4. Submit the PDF on Gradescope, matching each question to the corresponding page.
5. Templates for making the pipeline tables for Q2 are available in `.xls`, `.csv`, and LaTeX in the above linked LaTeX template.

---

1 Branch Prediction [2]

Can the 1-bit branch predictor ever perform better than the 2-bit dynamic branch predictor? Justify your answer with reasons. *No marks for the correct answer without the reasoning.* [2]

2 Memory Loop [10]

Consider the following code:

```
loop:   R1 <- Mem[R2]
```

```
R1 <- R1 + 1
Mem[R2] <- R1
R2 <- R2 + 4
R4 <- R3 - R2
BNEZ R4, loop
```

Assume that the initial value of `R3` is `R2 + 196`. Throughout this exercise use the classic five-stage pipeline taught in class. Specifically, assume that:

(a) Branches are resolved in the second stage of the pipeline, unless stated otherwise.

(b) There are separate instruction and data memories.

(c) All memory accesses take 1 clock cycle.

Q2.1 Show the timing of this instruction sequence for the pipeline without any forwarding or bypassing hardware but assuming a register read and a write in the same clock cycle "forwards" through the register file. Assume that the branch is resolved in the 4th stage of the pipeline, and handled by predicting not taken. If all memory references take 1 cycle, how many cycles does this loop take to execute? [4]

Q2.2 Show the timing of this instruction sequence for the pipeline with normal forwarding and bypassing hardware, and assuming that branches are resolved in the second stage. Assume that the branch is handled by predicting it as not taken. If all memory references take 1 cycle, how many cycles does this loop take to execute? [3]

Q2.3 Assume the pipeline with a single-cycle delayed branch and normal forwarding and bypassing hardware. Schedule the instructions in the loop including the branch delay slot. You may reorder instructions and modify the individual instruction operands, but do not undertake other loop transformations that change the number or opcode of the instructions in the loops. Show a pipeline timing diagram and compute the number of cycles needed to execute the entire loop. [3]

3 Branch Target Buffer [4]

A particular 32-bit ISA has `call` and `return` instructions. The `call` instruction jumps to the address specified as part of the `call` instruction, and saves the next instruction address after the call instruction

(i.e. `PC+4`) in a specified memory location (say M). The `return` instruction loads the saved address stored in M and transfers control to that address.

Q3.1 How well do you expect the Branch Target Buffer to perform on `return` instructions for a typical program? A program typically has lots of functions (e.g., a function for printing out to the screen) that are called from multiple call sites in the program. State your other assumptions, if any, and justify your answer with reasons. *No marks for the correct answer without the reasoning.* [2]

Q3.2 Develop a microarchitectural technique to improve the ability to predict the target address of return instructions, explaining how your technique would improve upon the baseline Branch Target Buffer. *No marks for the correct answer without the reasoning.* [2]