Homework 8 (Mandatory)

Total Points: 100

Deadline: Sat, 4/20/2024.

Since this is a mandatory assignment, we will keep the submission open until Wed, 4/24/2024.

An Excel file in XLSX format is provided. Put answers in the file and submit it in Gradescope. Round the answers to the nearest hundredth if necessary.

The file has multiple sheets. Pay attention to the sheet names. Do not add or remove cells.

The submitted file will be automatically graded. Please check the feedback from the auto grader. We will not grade submission manually. So it is important to make attempts early and correct mistakes in your work. HW8 solutions will not be provided.

We assume the following three processors in this homework assignment.

Processor A is a RISC-V processor with a 5-stage pipeline that does NOT have any forwarding paths. The branch is resolved in the MEM stage and the correct instruction is fetched in the WB stage of the branch instruction. The pipeline stalls until the branch is resolved. The instructions following the branch are flushed. They are fetched again if the branch is not taken.

Processor B is a RISC-V processor with a 5-stage pipeline that has all necessary forwarding paths. The branch is resolved in the MEM stage and the correct instruction is fetched in the WB stage of the branch instruction. The processor also uses a static branch predictor that always predicts not taken.

Processor C is similar to Processor B. The only difference is that Processor C has dynamic branch predictors.

1 Loop on Processor A.

Assume we use processor A.

The following code adds integers in two arrays (whose address is in s0 and s1, respectively) and stores the results back into the first array. s2 is initialized to 0. s3 is a large number.

```
I1:        lw         t0, 0(s0)
I2:        lw         t1, 0(s1)
I3:        add        t0, t0, t1
I4:        sw         t0, 0(s0)
I5:        addi       s0, s0, 4
```

```
I6:        addi       s1, s1, 4
I7:        addi       s2, s2, 1
I8:        bne        s2, s3, I1
I9:
```

a) Write the pipeline diagram of running the loop, from the IF stage of I1 to the IF stage of I1 in the next iteration. Submit the spreedsheet separately in HuskyCT.

b) What is the number of cycles required for each iteration? This number is the difference of cycle times where the same instruction is fetched. For example, if the first instruction in a loop is fetched in cycle 1 and fetched again in cycle 11 in the following iteration, each iteration takes 10 cycles. How many stall cycles are due to data hazards and how many are due to control hazards?

c) Scheduling instructions can remove some hazards that cannot be handled by hardware. Schedule the instructions to minimize pipeline stalls. Do not change any instructions. What is the number of cycles required for each iteration in the scheduled code? How many stall cycles are due to data hazards and how many are due to control hazards?

d) What is the speedup achieved on the loop by scheduling the instructions?

2 Loop on Processor B.

We study the loop in Problem 1 on processor B.

Repeat all the questions in Problem 1.

3 Performance.

We study the performance of an application on processors B. Suppose the instructions executed in the application break down into the following categories.

Type	**Percentage**
Arithmetic and Logic	30%
Load	32%
Store	20%
Branch	18%

The ideal CPI of the processor is 1. However, the actual CPI is higher because of hazards. For this application, 10% of the load instructions cause pipeline stalls because of the (load-use) data hazards. 60% of the executed branches are taken.

a) What is the CPI overhead from data hazards?

b) What is the CPI overhead from control hazards?

c) What is the average CPI of this application on Processor B?

d) What percentage of the execution time is spent on hazards?

4 Performance.

We study the performance of an operation Processor B. The two instructions below perform the **operation**: increment register x11 by 1 if and only if x10 is non-zero (not necessarily 1).

```
        beq   x10, x0, skip
        addi  x11, x11, 1
skip:
```

a) How many cycles does the operation take if x10 is 0?

b) How many cycles does the operation take if x10 is not 0?

c) If x10 is 0 70% of the time, how many cycles does the operation take on average?

d) Compilers can optimize code by removing some branch instructions. For example, one of the pseudoinstructions listed on the reference card can help to perform the same operation without a branch. Identify the pseudoinstruction and the corresponding (real) instruction. Using the new method, how many cycles does the operation take?

5 Dynamic branch predictor.

This exercise examines the accuracy of branch predictors for the following repeating pattern of branch outcomes:

T, T, T, T, NT, T, NT

Suppose the pattern repeats many times. The initial state of the predictor does not affect the accuracy much.

a) What is the accuracy of a single-bit predictor? What is the average CPI of this instruction on Processor C?

b) What is the accuracy of a two-bit predictor? What is the average CPI of this instruction on Processor C?