CS 203 (2021 Fall) Assignment #3

Student ID #:

Name:

Who else you discussed with when finishing the assignment:

(While you may have your partner do all the work, this will only hurt you when the midterm and final come around so don't do it.)

* For your answer to each question, please clearly specify what formula you use to solve the problem before replacing each term with numbers.
* Please show your work as detailed as possible.
* We refuse to give credits for answers with only final results even they are correct.

1. Consider the following RISC-V instructions:

```
Loop:   LD     X1,0(X3)
        ADD    X2,X1,X4
        MUL    X1,X2,X6
        ADD    X1,X1,X5
        ADD    X7,X7,X1
        ADDI   X12,X12,-1
        BNEZ   X12,Loop
        ADDI   X16,X16,4
        LD     X3,0(X16)
```

   Assume the initial value of X12 is 2. Please answer the following question.

   (1) Assume the branch instruction is in the BTB and the branch is always predicted taken. Please list the instruction sequence that will be **executed** (an instruction that does not finish until the end does not count).

(2) Continued from the previous question, if you have a 5-stage RISC-V pipeline as in Figure C.25 of the textbook, can you draw the pipeline diagram that simulates the execution of the above instructions sequence? You may use the given table for the diagram.

WB															
ME															
EX															
ID															
IF															
LD X1,0(X3)															

2. You are asked to evaluate the performance of the following branch prediction schemes:

   A: Static backward taken, forward not taken.

   B: Local predictor with two-bit state machine (Saturating counter/Bimodal).

   C: Global predictor based on 4-bit history and a 2-bit state machine associate with each entry. (4,2) correlating predictor.

   Now, you are given the following code segment. Assume each of the branch PC never cause conflicts/alias with other branches and the predictors are initialized as all zeros. Please evaluate the branch prediction accuracy for the following code snippet with all the give prediction schemes.

```
int i, j;

i = 0;
do {
    j = 0;
    do {
        loop c[i][j] = a[i][j] + b[i][j];
    } while( ++j < 5);
} while ( ++i < 100);
```