CENG3420 Homework 3

Due: Apr. 2, 2024

All solutions should be submitted to the blackboard in the format of **PDF/MS Word**.

Q1 (20%) Consider two different implementations of the same instruction set architecture. The instructions can be divided into three classes according to their CPI (classes A, B, and C). P1 with a clock rate of 2.5 GHz and CPIs of 1, 2 and 3, and P2 with a clock rate of 2 GHz and CPIs of 2, 2 and 2. A program with an instruction count of 1×10^7 can be divided into classes as follows: 20% class A, 50% class B, 30% class C.

1. What is the average CPI for each implementation? (10%)
2. What is the CPU execution time of this program for each implementation? (10%)

Q2 (15%) Consider the following sequence of instructions, and assume that it is executed on a five-stage pipelined datapath.

```
lw x12, 4(x11)
lw x14, 4(x13)
and x15, x12, x14
sub x13, x11, x13
```

1. If there is no forwarding or hazard detection, insert NOPs to ensure correct execution. You need to draw a pipeline diagram like we did in the lectures. (5%)
2. Does inserting NOPs change the clock cycle time? (5%)
3. Does inserting NOPs change the excute time of a program containing this block of code. (5%)

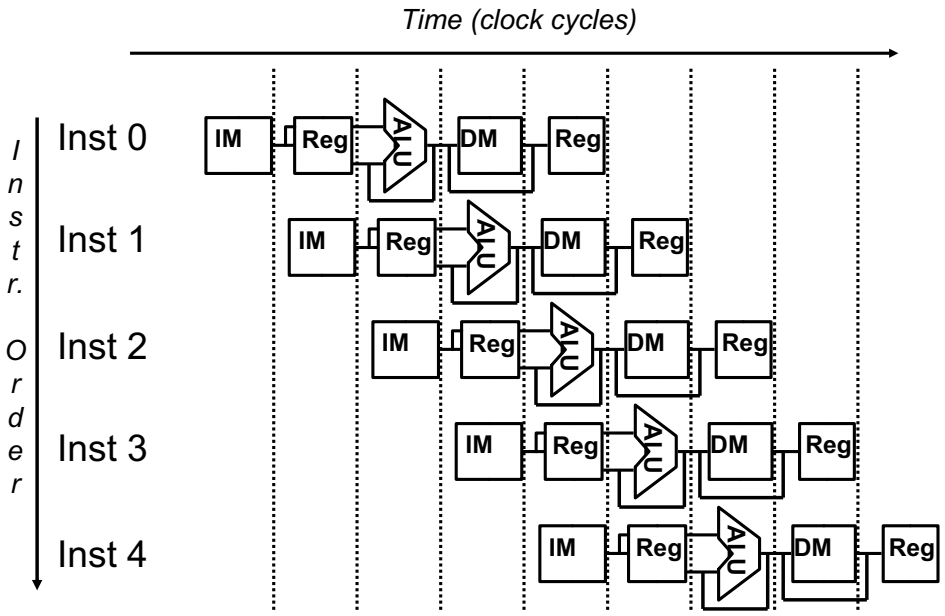
Q3 (25%)

Problems in this exercise refer to the following sequence of instructions, and suppose the instructions are executed on a five-stage pipelined datapath.

```
or x4, x1, x3
add x12, x4, x5
sub x13, x6, x4
```

1. List the data dependencies of the three instructions. (5%)
2. If there is no forwarding or hazard detection, how many NOPs are needed to ensure correct execution. You need to draw a pipeline diagram as we did in the lectures. (10%)
3. If hazard detection and forwarding are allowed, draw a pipeline diagram **only with forwarding** to ensure correct execution. In the diagram, the forwarding need to be specified by an arrow. (10%)

Note: The following figure shows the sketch of a pipeline diagram.



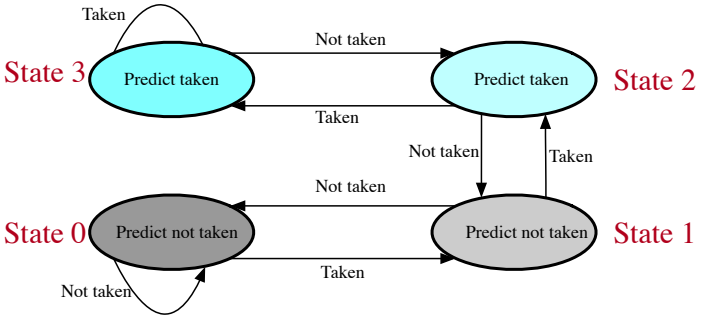
Q4 (20%)

This exercise examines the accuracy of various branch predictors for the following repeating pattern (e.g., in a loop) of branch outcomes: T, NT, NT, T, T. (T means 'Taken' and NT means 'Not taken')

1. What is the accuracy of always-taken and always-not-taken predictors for this sequence of branch outcomes? (5%)
2. What is the accuracy of a 2-bit predictor if this pattern is repeated forever? You should give your explanations rather than just an answer. The table below is recommended.

   The following figure shows the finite-state machine for a 2-bit prediction scheme. Here we assume this predictor starts from "State 0". (15%)

	Iteration1					**Iteration2**					**Iteration3**					...
Actual	T	NT	NT	T	T	T	NT	NT	T	T	T	NT	NT	T	T	
Predictor State																
Prediction																
True or Flase																



Q5 (20%) Assume a program requires the execution of 50×10^6 FP instructions, 110×10^6 INT instructions, 80×10^6 L/S instructions, and 16×10^6 branch instructions. The CPI for each type of instruction is 1, 1, 4, and 2, respectively. Assume that the processor has a 2 GHz clock rate.

1. By how much must we reduce the CPI of L/S instructions if we want the program to run two times faster? (10%)
2. By how much is the execution time of the program improved if the CPI of INT and FP instructions is reduced by 40% and the CPI of L/S and Branch is reduced by 30%? (10%)