CENG3420 Homework 3

Due: Mar. 29, 2022

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

Q1 (15%) 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?
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%?

Q2 (15%) In this exercise, we examine how pipelining affects the clock cycle time of the processor. For simplicity, we limit our attention to 4 instruction classes: Load (lw), Store (sw), R-type, and Branch(beq). Problems in this exercise assume that individual stages of the datapath have the following latencies:

IF	**ID**	**EX**	**MEM**	**WB**
250ps	350ps	150ps	300ps	200ps

And the stages that each instruction class need to execute is listed as follows:

Instruction Class	**IF**	**ID**	**EX**	**MEM**	**WB**
Load (lw)	✓	✓	✓	✓	✓
Store (sw)	✓	✓	✓	✓	
R-type	✓	✓	✓		✓
Branch	✓	✓	✓		

1. What is the clock cycle time in a pipelined and non-pipelined processor?
2. What is the total latency of an lw instruction in a pipelined and non-pipelined processor? Assuming there is no data harzard.

Q3 (20%)

Problems in this exercise refer to the following sequence of instructions, and assume that it is executed on a five-stage pipelined datapath:

```
add x15, x12, x11
lw x13, 4(x15)
or x13, x15, x13
```

Ps: you can draw the diagrams on paper, take pictures of your answers, and then insert the pictures into your homework document.

1. If there is no forwarding, draw a pipeline diagram to show where to insert NOPs to ensure correct execution. PS: you can follow the example on slide page 5, lec 12.
2. If there is forwarding, draw a pipeline diagram with forwarding to ensure correct execution. You can insert NOPs if necessary. PS: you can follow the example on slide page 10, lec 12.

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, T, T, NT. (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?
2. What is the accuracy of the 2-bit predictor if this pattern is repeated forever, assuming that the predictor starts off in the **top right** state of the FSM on slide page 36, lec 12?

Q5 (20%)

In this exercise we look at memory locality properties of matrix computation. The following code is written in C, **where elements within the same row are stored contiguously.** Assume each word is a 64-bit integer.

```
for (I=0; I<8; I++)
     for (J=0; J<8000; J++)
         A[I][J]=B[I][0]+A[J][I];
```

1. Which variable references exhibit temporal locality?
2. Which variable references exhibit spatial locality?
3. Locality is affected by both the reference order and data layout. The same computation can also be written below in Matlab, which differs from C in that **it stores matrix elements within the same column contiguously in memory.**

```
for I=1:8
    for J=1:8000
        A(I,J)=B(I,0)+A(J,I);
    end
end
```

   Which variable references exhibit temporal locality in the Matlab version?
4. Which variable references exhibit spatial locality in the Matlab version?

Q6 (10%)

How many total bits are required for a direct-mapped cache with 8 KiB of data and four-word blocks, assuming a 32-bit address?