ECE 485/585 – Computer Organization and Design

HOMEWORK #5

Due date: Friday December 2nd 2022 11:59PM

Solve the following exercises from the textbook (Chapter 4 and 5)

1. Exercise 4.14.1, 4.14.2s

4.14 This exercise is intended to help you understand the relationship between delay slots, control hazards, and branch execution in a pipelined processor. In this exercise, we assume that the following MIPS code is executed on a pipelined processor with a 5-stage pipeline, full forwarding, and a predict-taken branch predictor:

```
        lw r2,0(r1)
label1: beq r2,r0,label2 # not taken once, then taken
        lw r3,0(r2)
        beq r3,r0,label1 # taken
        add r1,r3,r1
label2: sw r1,0(r2)
```

4.14.1 [10] <§4.8> Draw the pipeline execution diagram for this code, assuming there are no delay slots and that branches execute in the EX stage.

4.14.2 [10] <§4.8> Repeat 4.14.1, but assume that delay slots are used. In the given code, the instruction that follows the branch is now the delay slot instruction for that branch.

2. Exercise 4.16.1, 4.16.2, 4.16.3

4.16 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

4.16.1 [5] <§4.8> What is the accuracy of always-taken and always-not-taken predictors for this sequence of branch outcomes?

4.16.2 [5] <§4.8> What is the accuracy of the two-bit predictor for the first 4 branches in this pattern, assuming that the predictor starts off in the bottom left state from Figure 4.63 (predict not taken)?

4.16.3 [10] <§4.8> What is the accuracy of the two-bit predictor if this pattern is repeated forever?

ECE 485/585 – Computer Organization and Design

3. Consider the following C code

```
for (i=1000;i>0;i=i-1)
   x[i] = x[i] + s;
```

   a) Convert above C code to MIPS code
   b) Show any hazard on your MIPS code, and insert *stall* if needed. How many clock cycles needed for unscheduled code? (Use delayed branches)
   c) Reschedule your MIPS code to reduce its clock cycles as much as possible. Show any hazard if needed
   d) Use Loop Unrolling technique to reschedule/rearrange your code

4. Exercise 5.3

5.3 For a direct-mapped cache design with a 32-bit address, the following bits of the address are used to access the cache.

Tag	Index	Offset
31–10	9–5	4–0

5.3.1 [5] <§5.3> What is the cache block size (in words)?

5.3.2 [5] <§5.3> How many entries does the cache have?

5.3.3 [5] <§5.3> What is the ratio between total bits required for such a cache implementation over the data storage bits?

Starting from power on, the following byte-addressed cache references are recorded.

Address											
0	4	16	132	232	160	1024	30	140	3100	180	2180

5.3.4 [10] <§5.3> How many blocks are replaced?

5.3.5 [10] <§5.3> What is the hit ratio?

5.3.6 [20] <§5.3> List the final state of the cache, with each valid entry represented as a record of <index, tag, data>.