EE 3613: Computer Organization

Homework #5

Due Date: Monday Nov 9, 2020 by 11:59 PM

Section 1: HARDCOPY SUBMISSION (100 Points) – 5

1. (**20 Points**) Show the forwarding paths needed to execute the following 4 instructions:
   add $3, $4, $6
   sub $5, $3, $2
   lw $7, 100($5)
   add $8, $7, $2

2. (**20 Points**) Identify all of the data dependencies in the following code. Which dependencies are data hazards that will be resolved via forwarding? Which dependencies are data hazards that will cause a stall?
   add $3, $4, $2
   sub $5, $3, $1
   lw $6, 200($3)
   add $7, $3, $6

3. (**20 Points**) Consider executing the following code on the pipelined datapath:
   add $2, $3, $1
   sub $4, $3, $5
   add $5, $3, $7
   add $7, $6, $1
   add $8, $2, $6
   At the end of the 5th cycle of execution, which registers are being read and which are being written? What are the *forwarding unit* and *hazard detection* units doing during the 5th cycle of execution, if any comparisons are being made, mention them.

4. (**15 Points**) Write the control logic to implement forwarding when we have a memory-memory instruction i.e. lw followed by sw. In which stage will the forwarding be implemented and what is the control logic code (similar to what we studied in class)?

5. (**25 Points**) We have a program core consisting of five conditional branches. The program core will be executed thousands of times. Below are the outcome of each branch for one execution of the program core (T for taken, N for Not taken)

   Branch 1: T-T-T
   Branch 2: N-N-N-N
   Branch 3: T-N-T-N-T-N
   Branch 4: T-T-T-N-T
   Branch 5: T-T-N-T-T-N-T

   Assume the behavior of each branch remains the same for each program core execution. For dynamic schemes assume each branch has its own prediction buffer and each buffer is initialized to the same state before each execution. List the predictions for the following branch prediction schemes:

   a. Always taken

b. Always not taken
c. 1-bit predictor, initialized to predict taken
d. 2-bit predictor, initialized to weakly predict taken

6. (**10 Points**) **BONUS**: We have a program of 103 instructions in the format of "lw, add, lw, add, …". The add instruction depends (and only depends) on the lw instruction right before it. The lw instruction also depends (and only depends) on the add instruction right before it. If the program is executed on the pipelined datapath:
(a) What would be the actual CPI?
(b) Without forwarding, what would be the actual CPI?