EECE 7352: Computer Architecture
HW2 Due March 2 11:59pm Eastern

Assume our traditional 5-stage pipeline (single instruction issue, in-order pipeline) with multi-port register file and memory. You can assume that bypassing is always enabled, and the register file is written in the first half of the cycle and read in the second half. The branch instruction (denoted with BR) is resolved in the second stage of the pipeline (ID stage) if the operands are ready. You can assume that registers and memory locations are appropriately initialized/modified such that the BR instruction is executed exactly three times in the following two questions (that is, the branch is taken the first two times and then, it "falls through"). The ground truth is that the branch is taken twice and falls through the third time.

Code Segment for Q1 and 2

```
LOCATION: SUB   R5, R2, R5
          ADD   R1, R5, R6
          BR    R1,  LOCATION
          MUL   R6, R1, R5
          LD    R2, #0(R6)
```

Q1. (15 points) Assume that we do not have any branch prediction capability (i.e., the branch is by default assumed to fall through until its condition has been fully evaluated). Do not assume any other branch related execution support (e.g., fetching from both paths, issuing multiple instructions, etc.) How many cycles does the given code segment take to execute?

Q2. (15 points) Assume that the branch instruction is predicted correctly every time. How many cycles does the given code segment take to execute?

Assume our traditional 5-stage pipeline (single instruction issue, in-order pipeline) with multi-port register file and memory. You can assume that the register file is written in the first half of the cycle and read in the second half.

Q3. (10 point) How many cycles will the following code segment take to execute? You should assume that the bypassing mechanism is enabled.

```
LD   R2, #0(R1)
MUL R4, R1, R2
LD   R3, #0(R4)
SUB  R4, R3, R1
```

Q4. (10 point) How many cycles will the following code segment take to execute? You should assume that the bypassing mechanism is enabled. Assume that the branch instruction is resolved in the second stage of the pipeline (ID stage) if the operands are ready. You can assume that registers and memory locations are appropriately initialized/modified such that the BR instruction is executed exactly once – that is, the BR instruction is not taken. Assume there is no branch prediction unit in the processor.

```
LOC: LD    R2, #0(R1)
     SUB   R3, R2, R3
     BR    R2, LOC
     MUL   R3, R3, R2
```