University of Maryland Baltimore County
Department of Computer Science and Electrical Engineering

CMSC 411, Computer Architecture

Assignment #4 Due: Thursday 10/26/17 in class

Question 1: *(40 Points)*

Problems in this exercise refer to the following sequence of instructions:

```
LW    $5, -16($5)
SW    $5, -16($5)
ADD $5, $5, $5
```

A) Indicate dependences and their type.

B) Assume there is not forwarding in this pipeline processor, indicate hazards and add *NOP* instructions to eliminate them.

C) Assume there is full forwarding, indicate hazards and add NOP instructions to eliminate unresolved cases.

The remaining problem in this exercise assumes the following clock cycle times:

Without forwarding	With full forwarding	With ALU-ALU forwarding only
200 ps	250 ps	220 ps

D) What is the total execution time of this instruction sequence without forwarding and with full forwarding? What is the speed-up achieved by adding full forwarding to a pipeline that had no forwarding?

E) What is the total execution time of this instruction sequence with only ALU-ALU forwarding? What is the speed-up over a no-forwarding pipeline?

Question 2: *(30 Points)*

For this problem, assume that all branches are resolved in ID stage and are perfectly predicted (this eliminates all control hazards). For the following fragment of MIPS code:

```
        LW  $5, -16($5)
        SW   $4, -16($4)
        LW   $3, -20($4)
        BEQ $2, $0, Label    ;Assume $2 ≠ $0
        ADD $1, $5, $4
Label:  SUB $2, $1, $3
```

If we only have one memory (for both instruction and data), there is a structural hazard every time we need to fetch an instruction in the same cycle in which another instruction accesses data. To guarantee forward progress, this hazard must always be resolved in favor of the instruction that accesses data.

A) What is the total execution time of this instruction sequence in the 5-stage pipeline that only has one memory?

B) How can the structural hazard be eliminated by adding *NOP* to the code? (Please show a modified version of the program with the added *NOP* instructions)

Question 3: *(30 Points)*

For following code, assume that the loop index ($10) is a multiple of 8:

```
Loop:  LW     $2,    0($10)
       SUB    $4,    $2,    $3
       SW     $4,    0($10)
       LW     $5,    4($10)
       SUB    $6,    $5,    $3
       SW     $6,    4($10)
       ADDI   $10,   $10,   8
       BNE    $10,   $30,   Loop
```

Schedule this code (reorder the instructions and make any necessary changes) for fast execution on the 5-stage MIPS pipeline. Assume data forwarding and not-taken prediction of conditional branching.