16.482 / 16.561: Computer Architecture and Design
Fall 2014

Homework #3
Due **Thursday, 9/25/14**

Notes:
- While typed submissions are preferred, handwritten submissions are acceptable.
- Any electronic submission must be in a single file. Archive files will not be accepted.
- Electronic submissions should be e-mailed to Dr. Geiger at Michael_Geiger@uml.edu.
- This assignment is worth a total of 100 points.

1. (20 points) A 5-stage pipeline contains four staging registers, one between each pair of stages in the datapath; these registers are used to hold information about each instruction as it proceeds through the pipeline. Describe the contents of each register.

Hints: Look at each stage of the pipeline, and think about (a) what information is needed for that stage, and (b) when is it generated? It's probably easiest to think of the last register first and work your way backwards through the pipeline.

2. (80 points) Consider the following sequence of instructions:

```
add     $t1, $t2, $t3
add     $t4, $t5, $t6
lw      $t7, 0($t1)
lw      $t8, 0($t4)
sub     $t9, $t7, $t8
lw      $s0, 4($t1)
sw      $t9, 4($t4)
slt     $s1, $s0, $t9
```

Assume that each datapath stage requires the following minimum amount of time to complete:
- o Instruction fetch (IF): 45 ns
- o Instruction decode (ID): 25 ns
- o Execute / address calculation (EX): 35 ns
- o Memory access (MEM): 45 ns
- o Register write back (WB): 25 ns

a. (15 points) How long will this code take in a single-cycle datapath?

b. (15 points) If we assume ideal pipelining (i.e., no hazards and therefore no stalls), how long will the code take in a pipelined datapath?

Question 2 is continued on the next page.

2 (continued)

c. (20 points) If we now assume a more realistic pipelined datapath **without** forwarding, how long will the code take? Show a revised code sequence that includes all necessary no-ops to support your answer.

d. (15 points) If we now assume a pipelined datapath **with** forwarding, how long will the code take?

e. (15 points) Calculate the speedup for each of the pipelined datapaths over the single-cycle case. In other words, determine how much faster each of the pipelined cases are than the single-cycle case.