CSC205 section 1 Spring 2015
Homework 8

How to Submit:
Please submit your solutions (parts A and B separately) through Blackboard. Remember to put your name and homework number on all the documents that you submit as attachments.

Total possible points in this homework: 5 for Part B
(You receive 1 bonus point towards Part A if you get all Part A questions correct.)

Part A

1. A system is designed to have a 5-stage pipeline, in which every stage is given the same amount of time. If a task at one stage is completed early, the result is stored in a register until it is fed to the next stage in the next clock cycle.

   Stage 1: Fetch and decode instruction,
   Stage 2: Compute operands addresses,
   Stage 3: Fetch operands,
   Stage 4: Execute,
   Stage 5: Write operands.

   The task at Stage 2 takes negligible time. The tasks at all other stages take equal amount of time. Let

   R: Time taken to complete one single instruction without pipeline
   R_p: Time for each stage of the pipeline.
   N: number of instructions

   Suppose the pipeline is almost hazard-free. Why is the theoretical speedup of this system only 4, and not 5?

Part B

2. [Total 5 pts] In this question, we explore synchronous pipeline efficiency in less-than-ideal scenarios. Let t be the typical length of a single task without pipelining. Let P2 be a pipeline with 2 stages, and P4 be a pipeline with 4 stages.
   a. [1pt] Find the maximum efficiency of P2 and of P4 if they are ideal.
   b. [1pt] Suppose the overhead of pipelining is $0.2*t$ per stage, and this cost is independent of the number of stages. Find the maximum efficiency of P2 and of P4 with overhead.
   c. [1pt] Suppose a pipeline hazard requires an idle stage to be added. Evaluate the maximum efficiency of P3 (which is P2 with an idle stage) and P5 (which is P4 with an idle stage) with an overhead of $0.2*t$ per stage.
   d. [1pt] Based on your observations on (b) and (c), provide insight on how to design an effective synchronous pipeline.
   e. [1pt] Pipeline hazards, in the worst case, cause the pipeline to flush and reset. Propose a model to evaluate the efficiency of a pipeline in this scenario.

Though not directly related to this question, the following discussion can cast some interesting light on the devastating effect of conditional branching in a real-world encounter. “Why is processing a sorted array faster than an unsorted array?” (http://stackoverflow.com/questions/11227809/why-is-processing-a-sorted-array-faster-than-an-unsorted-array?rq=1)