COMP206 **Homework 3** Assigned: October 6, 2003
Prof. Montek Singh Due: October 13, 2003

Note: This homework assignment is in two parts, Part I and Part II. Part I consists of problems from the textbook (Hennessy/Patterson 3rd ed., or "HP3") whose solutions appear at the end of the book. These problems are for self-study only, and will not be graded, and therefore need not be submitted as part of your written work. Part II consists of problems that will count towards your grade for this homework.

Part I (not graded):

1. **Basic Tomasulo Algorithm.** Do HP3 Problem 3.5.
2. **Branch Prediction Buffers.** Do HP3 Problem 3.9.
3. **Branch Target Buffers.** Do HP3 Problem 3.14.

Part II (to be graded):

1. (25 points) **Scoreboarding: RAW vs. WAR hazards.** This problem explores in detail how the scoreboarding approach is able distinguish between RAW and WAR hazards.
   (a) Write a short program (5 machine instructions or fewer) which has both an RAW hazard and a WAR hazard inherent in it, such that both the hazards can arise *simultaneously.* Clearly identify the instructions which cause these hazards.
   (b) Show a snapshot of the contents of the entire scoreboard (*i.e.,* all fields of the three-part data structure) which shows both the hazards "in action" simultaneously.
   (c) Discuss whether or not it is always possible to distinguish between RAW and WAR hazards by only looking at the "register result status" and "functional unit status" parts of the scoreboard (*i.e.,* without looking at the "instruction status").
   (d) Now assume that there are only two machine instructions in question, and that there is either an RAW dependency between the two, or a WAR dependency, not both. Write precise conditions which the hardware must check for in the scoreboard data structure, in order to be able to infer which of the two dependecies exists.
2. (30 points) **Tomasulo's Algorithm.** Do HP3 Problem 3.6, *parts (a) and (b) only.*
3. (20 points) **Branch Prediction.** Do HP3 Problem 3.10, but assume that the 1-bit predictors are initialized to T, the correlation bit is initialized to NT.