CS/ECE 347, Spring 1996
Recitation Assignment 6: Handling Register Data Hazards

Assigned: Tue., Mar. 5
Due: Mon., Mar. 11

1. Introduction

In this assignment, you will get started on Lab 2 by changing the pipeline simulation code to correctly handle register-register read-after-write data hazards.

You should do this assignment in the same group in which you will do the whole Lab. However, for this recitation assignment, each member should turn in a write-up.

2. Problem 1

Look at the code for the `reg_update` function in `mips_sim.c`, and examine where this function gets called in the other parts of the program. Briefly: what implications does this have for the `MEM/WB.WriteRegister = IF/ID.ReadRegister` hazards that came up in Recitation Assignment 5?

3. Problem 2

Identify all possible data hazards that can occur in the pipelined version of the simulator due to ALU instructions alone (i.e., instructions that operate on registers and immediate values, but don't touch memory). Describe in general English terms what the hazards are, and briefly mention in which parts of `mips_sim.c` they occur.

4. Problem 3

Get the file

`/afs/cs/academic/class/15347-s96/public/recitations/ra6/ra6dtest.s`

This file contains a simple sequence of add instructions that create data hazards in pipelined MIPS implementations.

Compile the `non_pipelined` executable, and convert `ra6dtest.s` into a .O file (see the Lab 2 handout for details). Run `non_pipelined` on the resulting .O file and turn in a printout of the final register values.

5. Problem 4

Modify mips_sim.c to correctly handle register-register read-after-write data hazards. Turn in:

For each of the following aspects of your solution, turn in a brief English description of how you handled it and printouts of the relevant modified pieces of code:

- What additional state did you add to the pipe registers?

- What pipeline state did you use to obtain the control information and data required to implement bypassing?
- How did you implement the MUX's used to implement bypassing?