

# GEORGIA INSTITUTE OF TECHNOLOGY

College of Computing

## CS6290/CS4290 — High-Performance Computer Architecture

Fall 2001

---

CS6290/CS4290  
Homework 3

Issued: September 30, 2001  
Due: October 5, 2001

---

**Purpose:** This homework reviews pipeline operation and instruction-level parallelism.

**Problems:**

1. Pipeline Operation.
2. Instruction Scheduling.
3. Write a paragraph about [Fisher91].

**Reading:** H&P Chapters 3, 4.1.  
[Fisher91]

### Problem 1: Pipeline Operation

**A:** Problem 3.1 in the book. Note that the question describes three different branching strategies. Part A refers to the original pipeline (Figure 3.4) which has a three-cycle branch penalty when you “flush” the pipeline (Figure 3.21). Part B says to use predict-not-taken which means you suffer the three-cycle penalty only if the branch is taken. Part C says to use the standard one-delay slot. The correct circuit diagram for a single delay slot is in the lecture slides (Figure 3.22 has a subtle bug in it).

## Problem 2: Instruction Scheduling

In each problem, show the stall cycles incurred in the original code before you reschedule the loop. I find it really handy to do this sort of work using a text editor instead of paper because I can easily move instructions up and down, etc. There's a text file with the code on the web site.

**A:** Problem 4.4 in the book.

**B:** Problem 4.5 in the book. The book claims you can solve this one with unrolling alone but I've never been able to do it. Feel free to use other tricks like software pipelining or taking advantage of the associativity of addition.

## Problem 3: Reading

[Fisher91] Joseph A. Fisher and B. Ramakrishna Rau, "Instruction-Level Parallel Processing", *Science*, 253, pages 1233-1241, September 13, 1991.

**A:** Read [Fisher91] above and write a one paragraph summary of the key *compiler* techniques that enable instruction-level parallelism. Note, one paragraph means something like 4-5 sentences on a third of a page.