Study Guide, Patterson & Hennessy, Sections 3.8 & 3.9

Section 3.8

The main points of this section are the following.

1. Floating point addition is not associative. Hence, if you are going to add a long list of f.p. numbers, in any language, you should sort them and then add them from smallest to largest. In this way, the small numbers will not get swamped by the larger numbers in the roundoff process.

2. A f.p. hardware bug in a Pentium chip cost Intel millions of dollars.

Section 3.9

Although this section is simply review, it's a good time to remind yourself that we have looked at all of the instructions on p. 226. By looking them over again, you can ask yourself whether you know what each of them does.

In addition, from p. 227, you should know the following, even though in some cases we have not formally seen them:

Pseudo-ops:

move, not, mul, mulu, div, divu, la, ld, sd, b, blt, ble, bgt, bge, slt. l.s, l.d, s.s, s.d

Remaining MIPS 32-bit insturctions:

xor, xori (after a lecture on combinational circuits); sllv, srlv, lb, lh, syscall, mfc1, mtc1

The only additional instructions that I'll be coming back to, not because you need to know them, but to remove the mystery from the very strange names are the instructions like:

blel "branch on less than or equal compare likely."

"Likely!" What kind of machine are we dealing with which allows us to write assembly language in which we give the machine a hint: "Hey, machine, just in case you care, I have a feeling that this branch is going to be more likely to be taken than to be skipped!"

The very last topic of our course will be to answer the question implied by these instructions: What could our CPU possibly do with the guess of the author of the MIPS program that one decision in a branch is more likely to take place than the other? The answer: To improve pipelining. Watch for it.