Slide 1

Administrivia

- Homework 5 on the Web (finally!). Due in a week. Problems may also give some hints about topics for next quiz.
- Homework 2 graded (for almost everyone). Updated solution in hardcopy (still not quite right — next slide).

Slide 2

Homework 2 Solution Errata

- In solution to problem 2.6.1, references to `B[j]` should be `A[j]`, and code to compute address to load this value should use `$s6` not `$s7`.

Slide 3

Implementing (Part Of) MIPS ISA — Review/Recap

- Previously we started sketching out something that implements a representative subset of the MIPS architecture (i.e., the definition of instructions, etc., used in the MIPS assembler programs we wrote), using building blocks discussed in Appendix C.
- Figure 4.2 shows most pieces of overall design; we discussed briefly how the various things we want to do correspond to parts of this figure. Section 4.3 gives more detail, and fills in a few more pieces . . .

Slide 4

The "Datapath"

- As discussed previously in class (and in more detail in section 4.2), we will need instruction memory, data memory, register file, PC, a full ALU, and a couple of adders.
- Did we leave anything out? yes:
  - Input to ALU / adder is two 32-bit quantities, but for some instructions what we have in the instruction is 16 bits — so we need something to extend that to 32 bits by extending the sign.
  - Both control-flow instructions include something that needs to be shifted two bits before being used to compute a target address, so we need to support that.
- Combine with "datapath" part of Figure 4.2 to get Figure 4.11, which leaves out the "control" part, substituting not-connected-yet control inputs (blue in the text).

Slide 5

Control Logic

- So we have a “datapath” that can do things, but there are some inputs that aren’t connected to anything. An analogy — the datapath is a puppet, and these inputs are its strings.
- Who/what pulls the strings? the “control logic” — combinational logic whose input is the current instruction plus any other needed information and whose output is those disconnected inputs to datapath.
- As mentioned in Appendix C, tools exist to transform truth tables into combinational logic, so our job is to come up with ones that will generate the signals we need for the datapath.
- Section 4.4 works through details. A lot of it should seem like common sense (viewed from the right angle?). Only potentially tricky part is input to ALU “which operation?” field (a bit more next time).

Slide 6

Minute Essay

- The design sketched so far has two separate memory blocks, one for instructions and one for data. This turns out to be needed for the simplest implementation, one in which each instruction executes in a single cycle. Why? is there something different about the types of values to be stored, or is there some other reason?

Slide 7

Minute Essay Answer

- This is one of the textbook's “check yourself” questions (p. 315), and the answer is at the end of the chapter.