

Slide 1

Administrivia

- None?

Slide 2

Minute Essay From Last Lecture

- Question: Fill in the following table (using figure 5.19):

| | add | lw | sw | beq |
|----------|-----|----|----|-----|
| RegWrite | ? | ? | ? | ? |
| MemRead | ? | ? | ? | ? |
| MemWrite | ? | ? | ? | ? |

- Answer?

How It Works

- We can see how this all works by tracing through what happens for some representative instructions. To do this, start with state element(s) whose outputs don't depend on inputs — here, PC.
- You'll do this for an `add` and a `lw` as homework. Let's try a `beq`.

Slide 3

Implementing the `j` Instruction

- Recall what's supposed to happen: Instruction contains a 26-bit "absolute" address. New value for PC is top 4 bits from PC+4, concatenated with 26 bits from instruction, and 2 bits of zero.
So, we need to compute this value, and use it when opcode says we have a `j`.
- This requires a little extra wiring, another multiplexor, and one more control signal, as in figure 5.29.

Slide 4

Minute Essay

- None — quiz.

Slide 5