Slide 1

Administrivia

- None.

Slide 2

Minute Essay From Last Lecture

- Question: Suppose `$t0` contains `0xffffffff` and `$t1` contains `0x000000ff`. What is in `$t2`, `$t3`, `$t4` after the following instructions are executed? Answers in either binary or hexadecimal are fine.

```
sll $t2, $t0, 4
and $t3, $t0, $t1
or $t4, $t0, $t1
```

- Answer?

Slide 3

Constructing an ALU — Building Blocks

- Next goal is to build an ALU (arithmetic logic unit) to perform addition, subtraction, and logical operations.
- Building blocks will be "gates", as shown in figure 4.8: AND gate, OR gate, inverter, multiplexor. (What's "below" these? Transistors acting as switches, as discussed very briefly earlier.)
- A word about notation: We'll use the textbook's notation, which alas is different from what you used in CS 1323.

CS 2321	*CS 1323*
$a \cdot b$	$a \wedge b$
$a + b$	$a \vee b$
$\overline{a}$	a'

- How to use these to build ALU?

Slide 4

Constructing an ALU — `and`, `or`

- How to implement "black box" with inputs a, b, op (0 for `and`, 1 for `or`)? See figure 4.9.

  We could then put 32 of these side by side for 32-bit operands, no?

Slide 5

Constructing an ALU — Addition

- Now consider addition. Two bits in, one bit out? Not exactly, right? also need carry-in and carry-out bits. See figure 4.10.
- We could write out a table showing inputs/outputs. See figure 4.11.
- If we can build something to implement this, we can connect 32 of them to get something to add 32-bit numbers. See figure 4.15.

Slide 6

Constructing an ALU — Addition, Continued

- Can we build a "1-bit adder" that behaves as described, using AND and OR gates and inverters?
- Carry-out bit is 1 exactly when at least two inputs are 1, right? How to write that in terms of AND, OR, NOT?

  And then we can build it; see figure 4.13
- Sum bit is a little more complicated, but doable.

Slide 7

Constructing an ALU — Subtraction

- What about subtraction? First we remember that $a - b$ is just $a + (-b)$, and then we remember how to compute $-b$ in two's complement notation.
- So, we can make an "adder" subtract by inverting one set of inputs and setting first carry-in bit to 1. (!)

Slide 8

Constructing an ALU — Summary

- Combining what we have so far, we get figure 4.16.

Slide 9

Minute Essay

- For the logic block drawn on the board, what results does it give for all possible inputs (combinations of a and b)?