Slide 1

Administrivia

- Reminder: Homework 2 due Friday.
- ACM meeting Thursday at 4pm in HAS 329. Find out what this organization is/does. Free refreshments!

Slide 2

Minute Essay From Last Lecture

- Question: Write MIPS assembler code to exchange the values of `a[0]` and `a[1]`. Assume register `$s0` contains the address of `a` (start of the array), and `a` is an array of integers.
- Answer?

Slide 3

Representing Instructions in Binary, Review

- Goal is to represent/encode instructions in binary — which instruction it is, what the operands are.
- Making all instructions the same size (32 bits) simplifies some things.
- However, different instructions have different kinds and numbers of operands, so we define several "formats" — different ways to divide up and use those 32 bits.

Slide 4

R Format

- Meant for instructions such as `add`.
- Fields:
  - `op` — op code, 6 bits (always 0 for this format)
  - `rs` — first source operand, 5 bits
  - `rt` — second source operand, 5 bits
  - `rd` — destination operand, 5 bits
  - `shamt` — "shift amount", 5 bits (not used for `add`)
  - `funct` — "function field", 6 bits (actually tells which instruction)
- Example — find binary representation of

```
add     $t0, $s1, $s2
```

  Look up `op` in table on inside back cover, registers in table on p. A-23.

Slide 5

I Format

- Meant for instructions such as `lw`.
- Fields:
  - `op` — op code, 6 bits
  - `rs` — first source operand, 5 bits
  - `rt` — destination operand, 5 bits
  - `disp` — displacement, 16 bits
- Example — find binary representation of

```
lw      $t0, 1200($t1)
```

  Look up `op` in table on inside back cover, registers in table on p. A-23.
- How can we tell which format is being used? determined by value for `op`.

Slide 6

Flow of Control

- So far we know how to do (some) arithmetic, move data into and out of memory. What about if/then/else, loops?
- We need instructions that allow us to "make a decision" — `beq` ("branch if equal"), `bne` ("branch if not equal").
- Illustrate with an example . . .

Slide 7

Flow of Control Example

- Suppose we have this in C

```
                if (i == j) goto L1:
                f = g + h;
        L1:     f = f - i;
```

- What instructions should compiler produce? Assume we're using `$s0` through `$s4` for for `f`, `g`, `h`, `i`, `j`.
- (For now, punt on how to represent `L1`.)

Slide 8

Another Flow of Control Example

- Of course, we don't usually have `go to` in C. More likely is this:

```
        if (i == j)
                f = g + h
        else
                f = g - h
```

- What to do with this? Rewrite using `go to` ...

Slide 9

Minute Essay

- None — quiz.