Slide 1

Administrivia

- Notice additions to reading.
- Talk by visiting scholar Prof. Abelson Monday at 4pm in Chapman. Bonus attendance points if you attend.

Slide 2

Floating-Point Representation, Review

- Usual way of representation non-integer numbers is “floating point”, similar to scientific notation. Numbers are represented by sign bit, exponent, and significant/mantissa. IEEE 754 standard defines choices for how many bits for each, some details of arithmetic.

  “Single-precision” format has 8 bits for exponent, biased by 127, 23 bits for significand.
- (Work through another example.)

Slide 3

Floating-Point Multiplication

- How to multiply two floating-point numbers? Approach is also similar to how you'd do this with decimal numbers in scientific notation:
  1. Add exponents and subtract bias.
  2. Multiply fractions.
  3. Normalize (get significand back in proper range).
  4. Check for overflow (exponent too big) or underflow (exponent too small).
  5. Round, and renormalize if necessary.
  6. Set sign bit.

Slide 4

Floating-Point Arithmetic Can Be Strange, Part 1

- Consider the following loop:

```
for (f = 0.0; f != 1.0; f += 0.1)
    printf("f = %g\n", f);
```

  What do you think it does?

  Why?

Slide 5

Floating-Point Arithmetic Can Be Strange, Part 2

- Consider the following code:

```
double fsmall = 1e-10;
double fbig = 1e10;
double temp1 = fbig;
for (int i = 0; i < 10000; ++i)
    temp1 += fsmall;
double temp2 = 0.0;
for (int i = 0; i < 10000; ++i)
    temp2 += fsmall;
temp2 += fbig;
```

  After it runs, is `temp1` equal to `temp2`?

Slide 6

Floating-Point in MIPS Architecture

- Architecture defines 32 floating-point registers (`$f0` through `$f31`), used singly for single-precision, in pairs for double-precision.
- Instruction set includes:
  - Arithmetic instructions:
    `add.s`, `sub.s`, `mul.s`, `div.s`; `add.d`, `sub.d`, `mul.d`, `div.d`
  - Load/store instructions (single-precision):
    `lwc1`; `swc1`
  - Comparisons:
    `c.eq.s`, `c.lt.s`, etc.; `c.eq.d`, `c.lt.d`, etc.
    These set a bit true/false, which can be used by `bc1t`, `bc1f`.

Minute Essay

- Suppose you are given the address of a 32-bit word in the memory of a computer implementing the MIPS architecture. How can you tell whether the 32 bits there are an integer, a single-precision floating point number, or something else? (What are some of the other possibilities?)

Slide 7