

## Multiplication and Division, Recap

- Algorithms for both operations are based on how you do things "by hand", with some modifications to permit simpler hardware. It's not critical to understand the details, but probably useful to work through an example to believe that it works. (See last problem in Homework 4.)
- Slide 3
- Required hardware is a 32-bit ALU (which we already talked about how to design), a 64-bit "work area", something to do right and left shifts of the 64-bit area, and some control logic.
- MIPS architecture uses "special registers" lo and hi for the 64-bit work area. This is where the results end up. There are instructions to multiply, to divide, and to move from the special registers.

## Representing Real (Non-Integer) Numbers

Approach is based on a binary version of "scientific notation":

In base 10, we can write numbers in the form  $+/-x.yyyy \times 10^{z}$ . E.g.,  $428 = 4.28 \times 10^{2}$ , or  $-.0012 = -1.2 \times 10^{-3}$ .

• We can do the same thing in base 2. Examples:

$$32 = 1.0_2 \times 2^5$$
  
-3 = -1.1\_2 × 2<sup>1</sup>

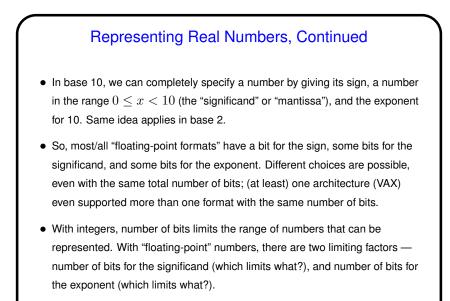
 $1/2 = 1.0_2 \times 2^{-1}$ 

$$3/8 = 1.1_2 \times 2^{-1}$$

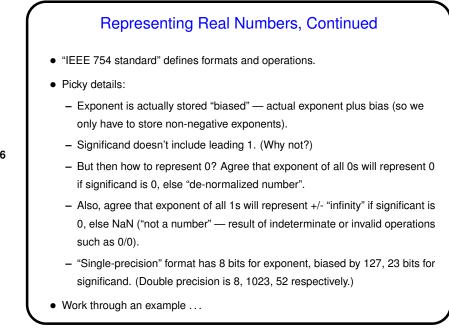
 This is "floating point" (as opposed to "fixed point", which would allow for non-integers but wouldn't allow as much flexibility — wide range, all with reasonable precision).

Slide 4

Slide 5



(Does this suggest why the VAX designers offered two formats?)



Slide 6

