November 3, 2003

- Working With Real Numbers
  - Must use approximations
    - Infinite number of reals
    - Finite number of bits
  - Floating Point
  - Fixed Point

Floating Point

- Based on scientific notation
  - Value = ±fraction * base $^{\pm exponent}$
  - Base is a power of 2 (2 and 16 most common)
  - Example: IEEE-754
    - Short (Used for Java's *float* type)
      - 32 bits: 1 fraction sign, 8 signed exponent, 23 fraction value
    - Long (Used for Java's *double* type)
      - 64 bits: 1 fraction sign, 11 signed exponent, 52 fraction value

Floating Point ALU

- Parse operands
  - Sign of fraction
  - Value of fraction
    - IEEE-754: Restore hidden 1.
  - Exponent
- Align fractions
  - Shift value and adjust exponent.
- Perform calculation
- Normalize result
  - Fraction must be binary 01.xxx
  - Shift fraction and adjust exponent as necessary
- Construct floating point result
- Not to be implemented casually!

Fixed Point

- Fixed number of bits to the right of the binary point.
  - Integers (and unsigned integers) have zero bits to the right of the binary point.
- Hardware is the same as for signed and unsigned integers, no matter where the binary point is.
- Gives up dynamic range possible with floating-point.

Handel-C Fixed Point Library

- See me for a copy of the manual.
- fixed.hch and fixed.hcl are in the usual places.
- Manual says you can encode arbitrary values, but compilation fails if the real value cannot be represented precisely in the bit structure specified.
  - For example 0.1_{10} can not be used.
  - Sample Code