Floating Point Numbers

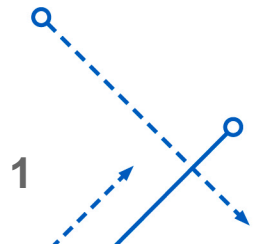
Karthik Dantu

Ethan Blanton

Computer Science and Engineering

University at Buffalo

kdantu@buffalo.edu



Administrivia

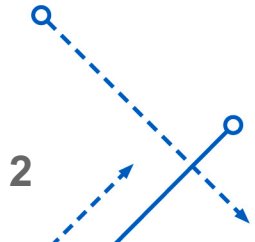
- PA2 is out! Due next Friday at 11:59 PM
- PA2 handout quiz is due THIS WEDNESDAY at 11:59 PM
- Many used magic numbers in PA1 – don't!

  80

  24

  X

  88
- Start PA2 now – handout is tricky



Floating Point Numbers

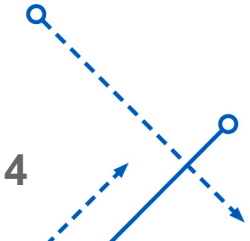
- Counter point to integer representation
- Used to

  Represent rational numbers

  Approximate real numbers
- Binary floating point formats have surprising properties

Fixed Point

- Floating point has a closely related representation – fixed point
- Fixed point is also used to represent rational and real numbers
- However, it is less flexible than floating point
- Lets first look at fixed point



Fixed Point

- A fixed point number has a fixed number of digits
- Fixed point number has a maximum magnitude and minimum fractional portion that do not change
- For example, a fixed point number with 3 digits before and after the decimal point could be

  003.142

  099.440

  107.429

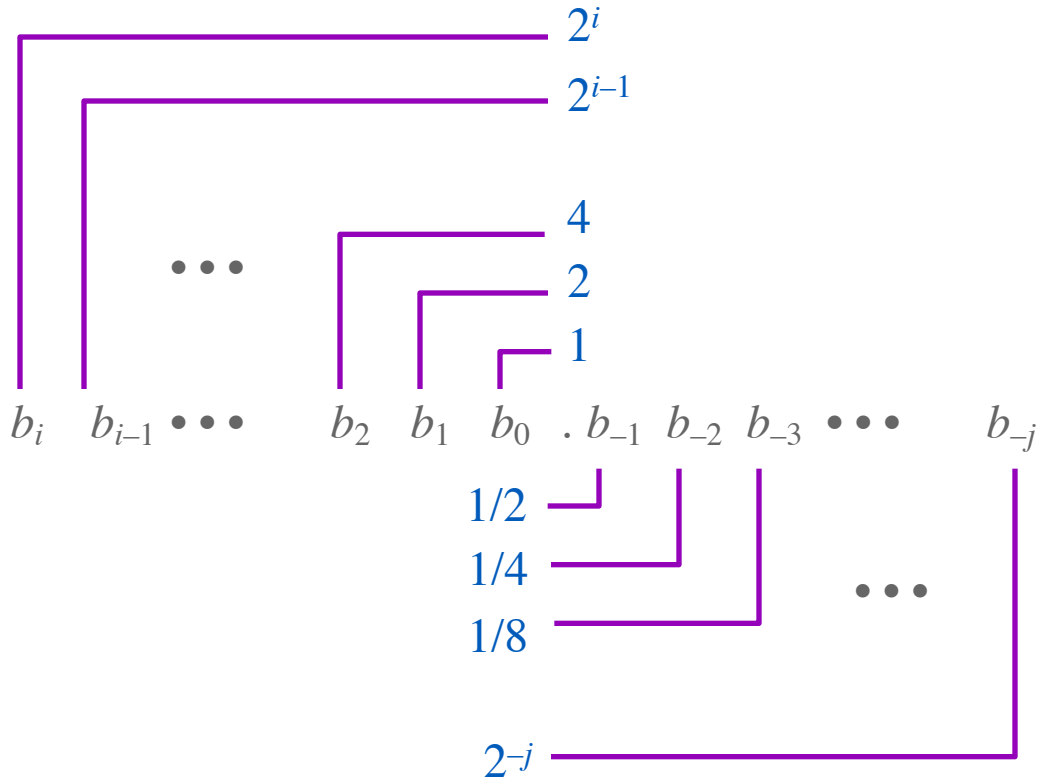
Binary Point

- In binary numbers, we have binary point
- Similar to decimal point, binary point separates 2^0 from 2^{-1}
- Do not confuse decimal digit and decimal point
- Similarly, don't confuse binary digit and binary point

Binary Point

- There are *w* whole-number bits before binary point
- There are *f* fractional bits after the binary point
- The largest bit before the point is b^{w-1}
- The smallest bit before the point is b^0
- The largest bit after the point is b^{-1}
- The smallest bit after the point is b^{-f}

$$b^{w-1},\ldots,b^0,b^{-1},\ldots,b^{-f}$$



A *w.f*-bit Binary Number

- The w whole-number bits are defined as in integers:

$$b_i, i \geq 0 \doteq b_i \cdot 2^i$$

- The f fractional-number bits are defined as follows:

$$b_j, j < 0 \doteq b_j \cdot 2^{-b_j}$$

- The total value is:

$$\sum_{i=0}^{w-1} b_i \cdot 2^i + \sum_{j=1}^{f} b_j \cdot 2^{-j}$$

Example Binary-Point Computation

- Consider 11.101b

$$11.101b = 1.2^{1} + 1.2^{0} + 1.2^{-1} + 0.2^{-2} + 1.2^{-3}$$
$$= 2 + 1 + ½ + 0 + 1/8$$
$$= 3.625$$

Floating Point

- A floating point number, such as a `float` or a `double` is a number with a variable number of digits before and after the decimal point

(On computers, a variable number of bits before and after the binary point!)

Examples:

3.14159

6.022×10^{23}

6.626×10^{-34}

Floating Point numbers

- To represent numbers of very small and very large magnitude, floating point allows the point to move
- Number of digits of precision is fixed
- Terms:
- **Significand**: Meaningful digits of a number
- **Exponent**: “Distance” of those digits from zero in powers of arithmetic base

Floating Point Representation

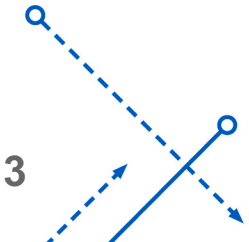
- In base 10, a floating point number is of the form $x \times 10^y$
- For example, 6.022×10^{23} :

  Significand: x is 6.022

  Exponent y is 23
- This requires six digits to store vs 24 digits to store 602200000000000000000000
- In base 2, a float point number is $x \times 2^y$

IEEE 754 Floating Point

- Established in 1985 as a uniform standard for floating point arithmetic
- Supported by all major CPUs
- Driven by numerical concerns
- Nice standards for overflow, underflow, rounding



Floating Point Representation

- Numerical Form
- $-1^s.M.2^E$

  Sign bit *s* determines whether number is negative or positive

  Significand *M* normally a fractional value in range [1.0,2.0).

  Exponent *E* weights value by power of two

- Encoding

  MSB is sign bit

  `exp` field encodes *E*

  `frac` field encodes *M*

Required readings

- B&O 2.4.1-2.4.3,