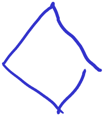
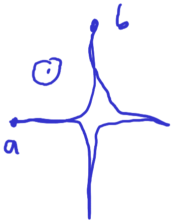
- Forum invites
- Exam lab 0 (→ schedule, online exam same period)
  - 1h 50
  - 19 quiz — 4 min → short-answer/mc → no fb
  - 3 code — 10 min →



- Forum posts w/ solutions: avoid
- Today: Floating point / inclass / tp
- unit balls $p=0.5$



$\|a\|_{0.5}=1$

$\|b\|_{0.5}=1$

$\|0.5a+0.5b\|_{0.5}$
$\le 0.5\|a\|_{0.5}+0.5\|b\|_{0.5}=1$

Wanted: Real Numbers... in a computer

Computers can represent *integers*, using bits:

$$23 = 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = (10111)_2$$

How would we represent fractions?

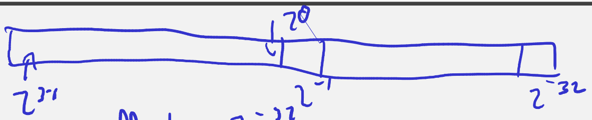
$(110.11)_2$

$$23.625 = \ldots + \underbrace{1}\cdot \underset{\frac{1}{2}}{2^{-1}} + \underbrace{0}\cdot \underset{\frac{1}{4}}{2^{-2}} + \underbrace{1}\cdot \underset{\frac{1}{8}}{2^{-3}}$$

"fixed point"

Fixed-Point Numbers

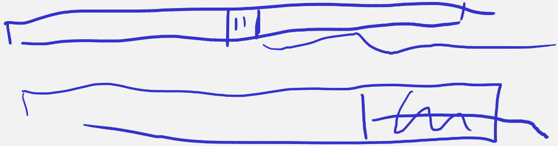
Suppose we use units of 64 bits, with 32 bits for exponents $\geqslant 0$ and 32 bits for exponents < 0. What numbers can we represent?



smallest: 2^{-32}

biggest $\approx 2^{32}$

How many ‘digits’ of relative accuracy (think relative rounding error) are available for the smallest vs. the largest number?



Floating Point Numbers

Convert $13 = (1101)_2$ into floating point representation.

$$\underbrace{1.\underline{101}}_{\text{significand}} \cdot 2^{\overbrace{3}^{\text{Exponent}}}$$

What pieces do you need to store an FP number?

significand Exponent

Floating Point: Implementation, Normalization

Previously: Consider *mathematical* view of FP. (via example: $(1.101)_2$)
Next: Consider *implementation* of FP in hardware.
Do you notice a source of inefficiency in our number representation?

Idea: Don't store the leading one in the sig.

For $(1.\underbrace{101}_{})_2 \cdot 2^3$, only store 101

FP exponent doesn't use 2's complement

actual exp = -1023 + the integer from the stored bit pattern

Unrepresentable numbers?

Can you think of a somewhat central number that we cannot represent as

$$x = (1.\_\_\_\_\_\_\_\_\_\_\_\_)_2 \cdot 2^{-p}?$$

~~zero~~ can't be represented : oops

"special exponent" : -1023

↳ turn off the implicit leading 1 on the significand

Demo: Picking apart a floating point number [cleared]

Subnormal Numbers

What is the smallest representable number in an FP system with 4 stored bits in the significand and an exponent range of $[-7, 7]$?

$1.0000 \times 2^{-7} \longrightarrow 0$

$\cdots\cdots$

Subnormal Numbers II

What is the smallest representable number in an FP system with 4 stored bits in the significand and an exponent range of $[-7, 7]$? (Attempt 2)

= special exponent" -7

0.001×2^{-7}

1.000×2^{-7}

Why learn about subnormals?

Underflow

- FP systems without subnormals will *underflow* (return 0) as soon as the exponent range is exhausted.
- This smallest representable *normal* number is called the *underflow level*, or *UFL*.
- Beyond the underflow level, subnormals provide for *gradual underflow* by 'keeping going' as long as there are bits in the significand, but it is important to note that subnormals don't have as many accurate digits as normal numbers.
- Analogously (but much more simply–no 'supernormals'): the overflow level, *OFL*.

Rounding Modes

How is rounding performed? (Imagine trying to represent π.)

$$\left(\underbrace{1.1101010}_{\text{representable}}11\right)_2$$

- throwing away 1.1101010
- nearest number 1.1101011

What is done in case of a tie? $0.5 = (0.1)_2$ ("Nearest"?)

- "round-to-even"

Demo: Density of Floating Point Numbers [cleared]

Demo: Floating Point vs Program Logic [cleared]