Real Numbers

- How to represent:
  - 0.25
  - 1,234,543.00123476

What do they mean

12.125

$\times 10^{1}$ $\times 10^{0}$ $\times 10^{-1}$ $\times 10^{-2}$ $\times 10^{-3}$

Now let's try in binary

- Say we had 8 bits:

1011.1011

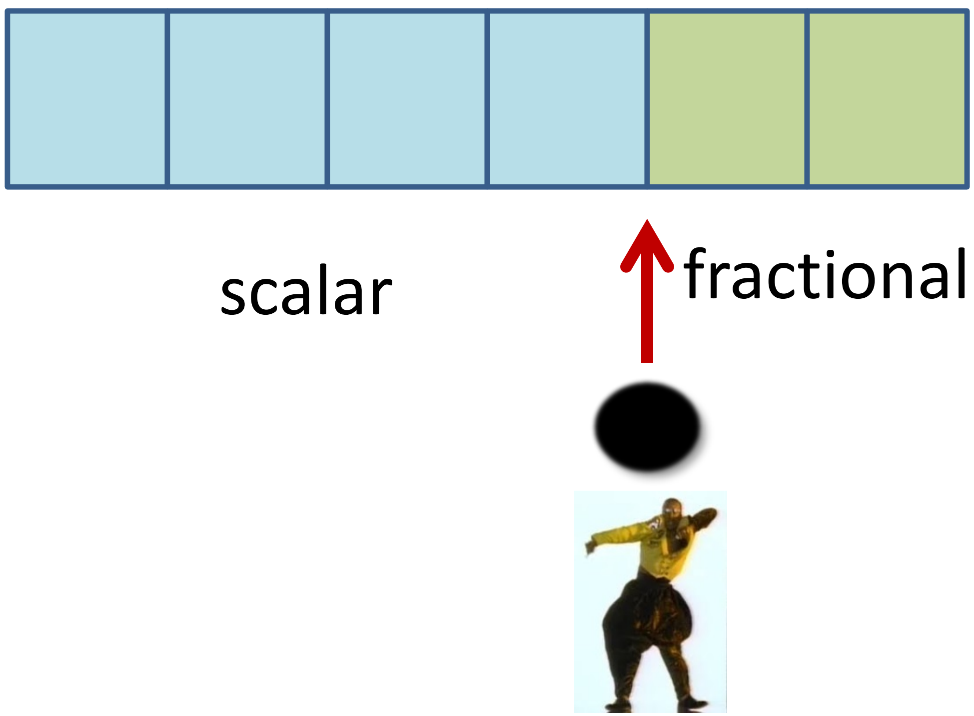
X $x2^{3}$ $x2^{2}$ $x2^{1}$ $x2^{0}$ $x2^{-1}$ $x2^{-2}$ $x2^{-3}$ $x2^{-4}$

= 8 + 0 + 2 + 1 + 0.5 + 0 + 0.125 + 0.0625

11.6875

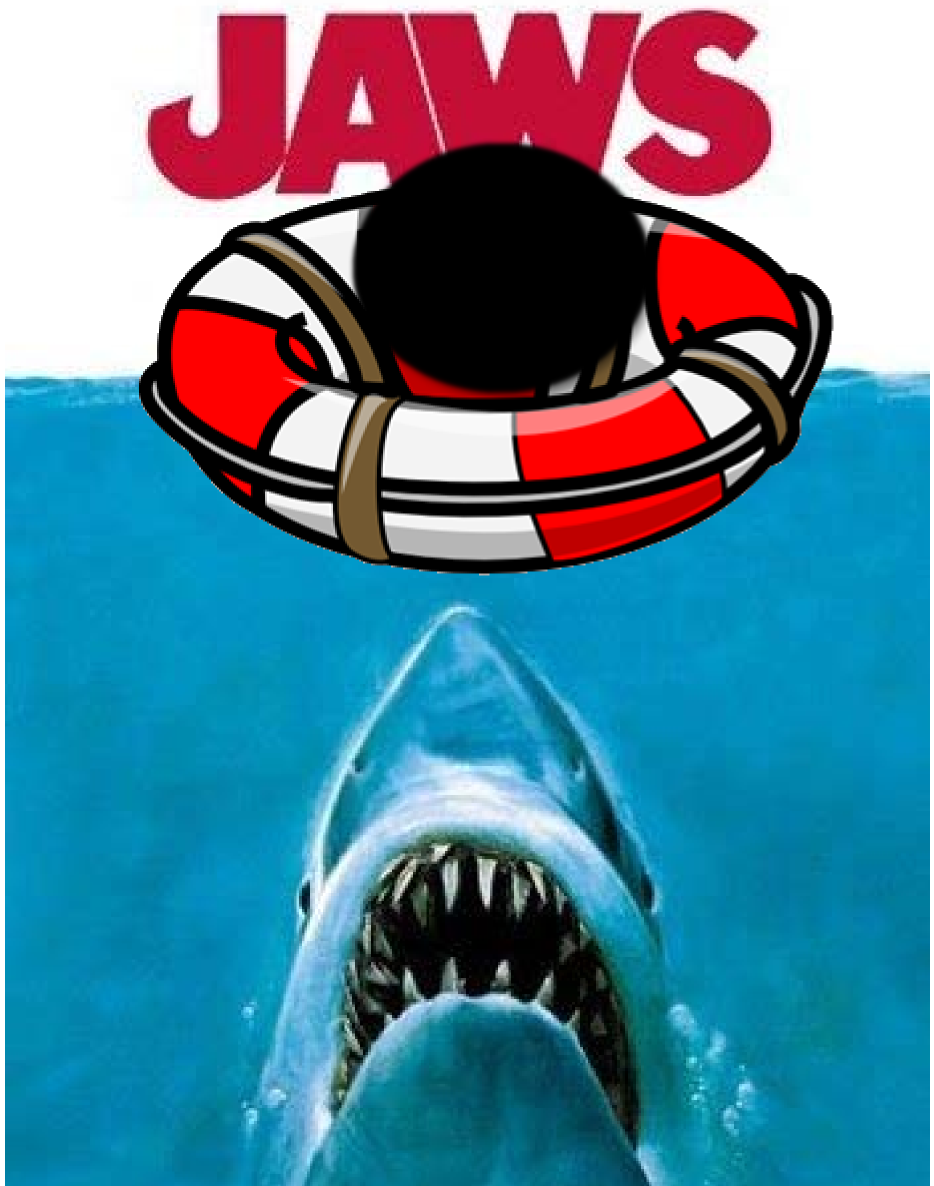
Fixed-Point Representation

- Given N bits to represent real numbers
- The  is fixed by convention between two digits
- e.g., 4.2 representation



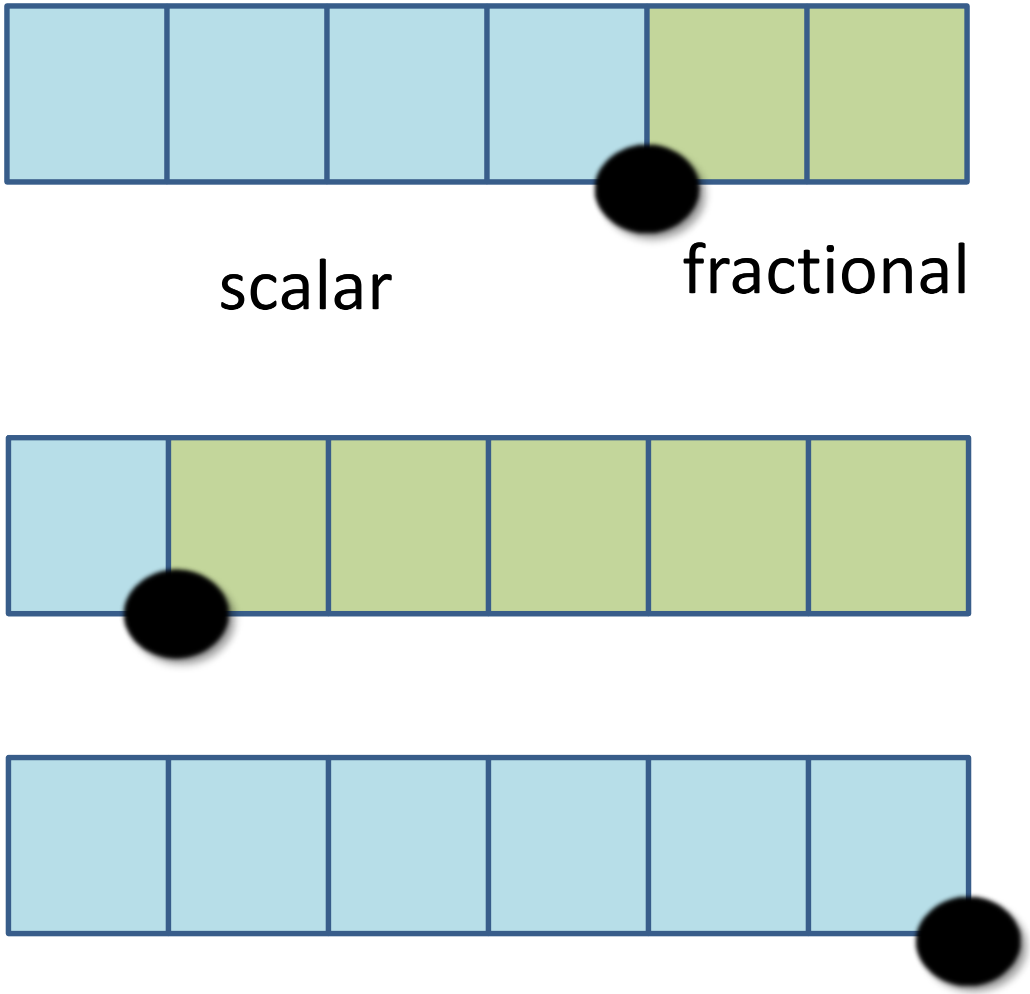
The problem with fixed-point

- Range is small
- Cannot represent very large or very small or mix
- Programmers have to use scaling factors

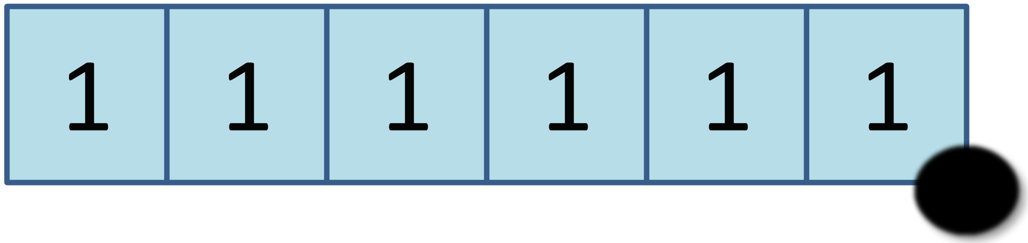

JAWS

Floating Point: Concept

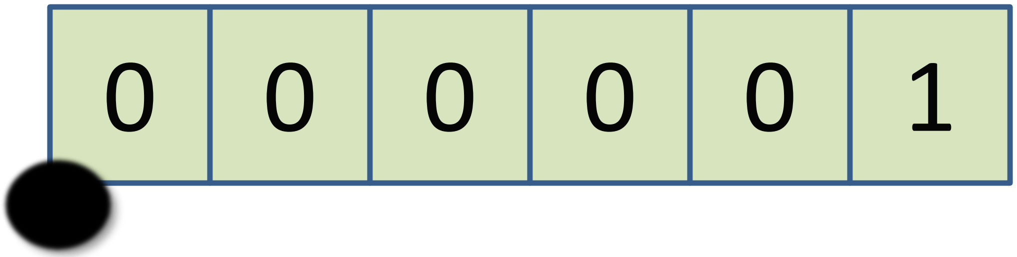
- Point can "float" anywhere we want



Floating point concept contd.



127



0.015625

- Range still small
- Cannot represent very large number or very small ones

Floating-Point Concept Final

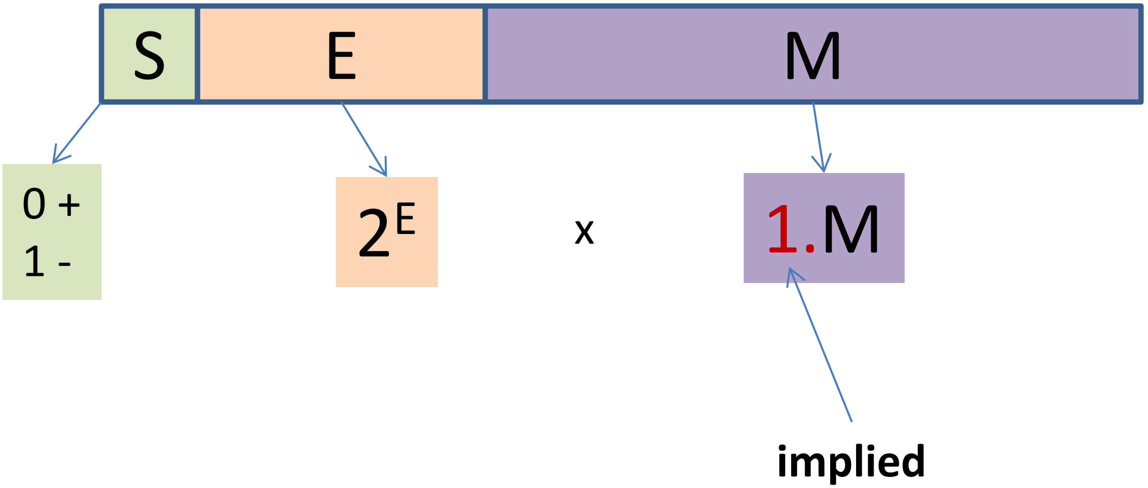
- Given N bits represent as close a number as you can
- E.g., w/ 6 bits

1	1	1	1	1	1	1	0	1	0	1	1	0	1	0	1	0	0	0	0	0	1	1	1

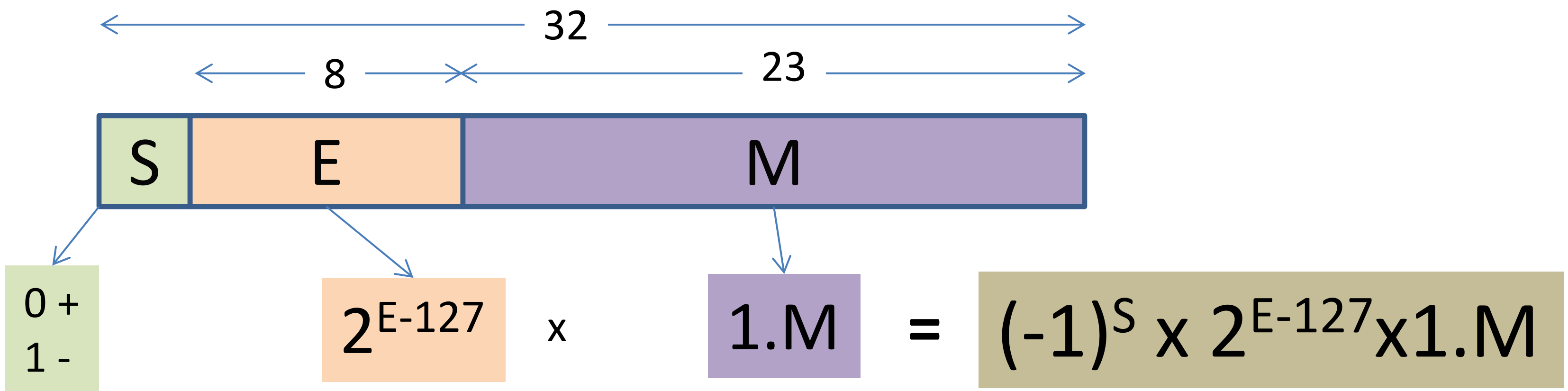
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	0	1	1	1

IEEE 754 Standard for Floating Point

- 16-, 32-, 64-, or 128-bit
- Float = 32-bit, **single precision**
- Double = 64-bit, **double precision**
- In general:



Single-Precision, 32-bit



1 10000001 10000000000000000000000

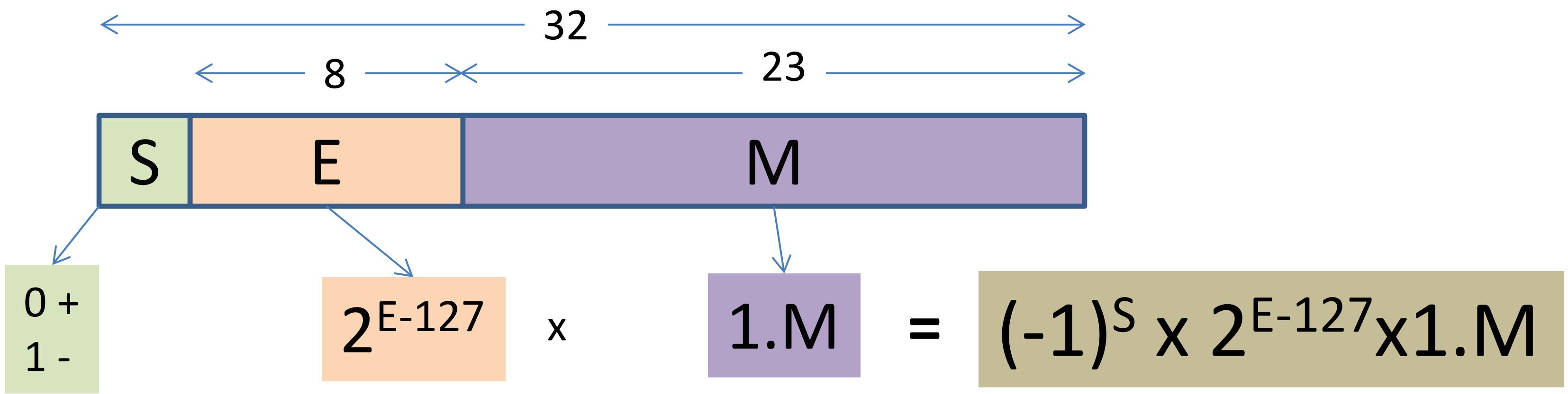
S = -

E = 129 – 127

M = .1

$$-2^2 \times 1.1 = 1100.0 = -6$$

Single-Precision, 32-bit



0 01111110 11000000000000000000000

S = +

E = 126 - 127

M = .11

$$+2^{-1} \times 1.11 = 0.111 = 0.875$$

How to represent a number in IEEE FP

STEP 1: Find most-significant “1”

0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	0	1	1	1

STEP 2: Mantissa: digits to the right

0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	0	1	1	1

STEP 3: Exponent, how many bits till the actual dot

0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	0	1	1	1

13

Example

00011100110101110011.11110011101

000**1**1100110101110011.11110011101

000**1**1100110101110011.11110011101

mantissa

000**1**$\boxed{\text{1100110101110011.1111001}}$1101

16

0 1000111 11001101011100111111001

S 143-127 mantissa

Floating Point is not precise always

- 0001110011010111001 1.11110011101
- Was represented as:
- 0001110011010111001 1.1111001
- The error for SP FP is within 2^{-23}
- In general given a number **x** FP represents:

x’

- **Error:**

x – x’

- **There is a number ε such that:**

$$1 + \varepsilon = 1$$

- **Machine epsilon**

Floating Point is not precise always

- **Relative Error**

$$x - x' / x = \delta$$

- **Number represented is:**

$$x (1 + \delta)$$

- **Error in the units in the last place, ulp**
- Spacing between two successive floating point numbers
  - **Within 0.5 ulp with rounding to nearest**

Got to be careful with calculations

- Say want to calculate:

$$A + B$$

- With FP we'll get this:

$$A\,(1 + \delta_A) + B\,(1 + \delta_B)$$

- **But this may not be possible to represented exactly, so we have:**

$$(A\,(1 + \delta_A) + B\,(1 + \delta_B))(1 + \delta_3)$$

- **Which evaluates to:**

$$A\,B\,[1 + A\,/\,(A + B)\,(\delta_A + \delta_3) + B\,/\,(A + B)\,(\delta_B + \delta_3)]$$

- **What happens when $A \sim B$?**

Got to be careful with calculations

- Say want to calculate:

$$\mathbf{A \times B}$$

- With FP we'll get this:

$$\mathbf{A\ (1 + \delta_A) \times B\ (1 + \delta_B)}$$

- **But this may not be possible to represented exactly, so we have:**

$$\mathbf{(A\ (1 + \delta_A) \times B\ (1 + \delta_B))(1 + \delta_3)}$$

- **Which evaluates to:**

$$\mathbf{A \times B \times [1 + \delta_A + \delta_B + \delta_3]}$$

FP calculations may introduce errors

- Some rules:
  - Be wary of subtracting very close numbers
  - Adding numbers that differ greatly in magnitude

Special Representations

- If E=0, M non-zero, value=(-1)^S x 2^(-126) x 0.M (**denormals**)
  - Mantissa is not normalized
  - Very small numbers close to 0
- If E=0, M zero and S=1, value=-0
- If E=0, M zero and S=0, value=0
- If E=1...1, M non-zero, value=**NaN** “not a number”
- If E=1...1, M zero and S=1, value=-infinity
- If E=1...1, M zero and S=0, value=infinity