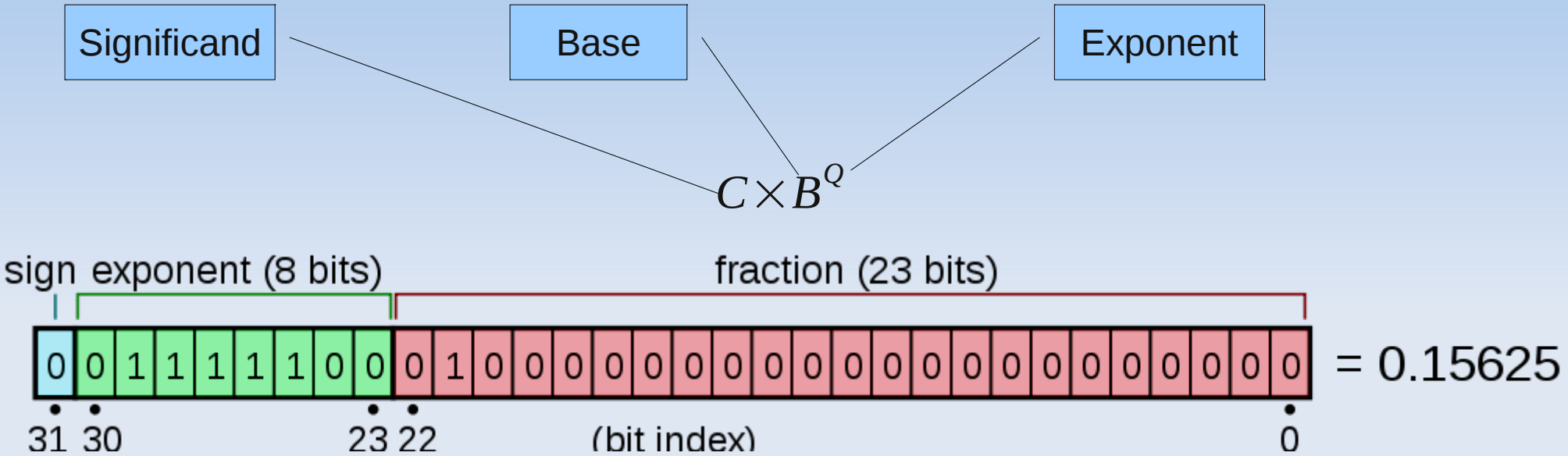
Floating Point Representation in Computers

- Floating Point Numbers - What are they?
- Floating Point Representation
- Floating Point Operations
- Where Things can go wrong

What are Floating Point Numbers?

- Any Number that cannot be completely described using an Integer number.
  - 1, 2, 523. not a floating point number
  - 1.02, e, 0.1. a floating point number
- Floating Point numbers are at their best when describing a continuous variable.
- Floating Point numbers are at their worst when describing discrete values.

Floating Point Number Representation.



- Floating Point numbers constist of a signed significand (C) and a signed integer exponent(Q) of the base(B).
- Most commonly represented according to the IEEE 754 Standard.

Floating Point Number Representation.

- Decimal notation is the sum of fractional powers of 10.

$$0.625 = \frac{6}{10} + \frac{2}{100} + \frac{5}{1000}$$

- Computers store values as the sum of fractional powers of 2.

$$0.625 = \frac{1}{2} + \frac{0}{4} + \frac{1}{8}$$

Floating Point Number Representation.

- Some decimal numbers have no exact representation in base 2

$$0.1$$

$$\frac{1}{2^1}+\frac{1}{2^2}+\frac{0}{2^3}+\frac{0}{2^4}+\frac{1}{2^5}+\frac{1}{2^6}+\frac{0}{2^7}+\frac{0}{2^8}+\frac{1}{2^9}+\frac{1}{2^{10}}+\frac{0}{2^{11}}+\frac{0}{2^{12}}+\frac{1}{2^{13}}+\frac{1}{2^{14}}+\frac{0}{2^{15}}+\frac{0}{2^{16}}+\frac{1}{2^{17}}+\frac{1}{2^{18}}+\frac{0}{2^{19}}+\frac{0}{2^{20}}+\overline{\frac{1}{2^{21}}+\frac{1}{2^{22}}+\frac{0}{2^{23}}+\frac{0}{2^{24}}}$$

- The significand inifinitely repeats 1100 pattern.

$$\frac{1}{2^1}+\frac{1}{2^2}+\frac{0}{2^3}+\frac{0}{2^4}+\frac{1}{2^5}+\frac{1}{2^6}+\frac{0}{2^7}+\frac{0}{2^8}+\frac{1}{2^9}+\frac{1}{2^{10}}+\frac{0}{2^{11}}+\frac{0}{2^{12}}+\frac{1}{2^{13}}+\frac{1}{2^{14}}+\frac{0}{2^{15}}+\frac{0}{2^{16}}+\frac{1}{2^{17}}+\frac{1}{2^{18}}+\frac{0}{2^{19}}+\frac{0}{2^{20}}+\frac{1}{2^{21}}+\frac{1}{2^{22}}+\frac{0}{2^{23}}+\frac{1}{2^{24}}$$

$$0.100000001490116119384765625$$

- Rounded results in a value not quite 0.1

IEEE 754 Types

Name	Common name	Base	Digits (p)	Emin	Emax	Notes	Decimal digits	Decimal E max
binary16	Half precision	2	10+1	−14	15	storage, not basic	3.31	4.51
binary32	Single precision	2	23+1	−126	127		7.22	38.23
binary64	Double precision	2	52+1	−1022	1023		15.95	307.95
binary128	Quadruple precision	2	112+1	−16382	16383		34.02	4931.77

- A IEEE 754 float can represent a range of finite numbers, +∞, -∞, and NaN.
  - The range of finite numbers is determined by the properties of the representation. The range of non-zero magnitudes representable is from 1xB^(Emin - p + 1) to (B^p - 1)^(Emax - p + 1).

IEEE 754 types

- Representable non-zero numbers are split in to two categories, Normal and Subnormal.
  - The smallest magnitude Normal number is B^Emin.
  - Subnormal numbers are number who's magnitude is less then B^Emin.
- Subnormal numbers allow for underflow exceptions to fail gracefully, getting smaller with loss of precision instead of snapping to zero.

Floating Point Rounding

- Rounding occurs when the exact result of an operation requires more precision then available in the significand.
- IEEE 754 Round off modes.
  - round to nearest, where ties round to the nearest even digit in the required position.
  - round to nearest, where ties round away from zero
  - round towards +∞
  - round towards -∞
  - round towards zero

Floating Point Operations

- To Add or Subtract shift numbers left or right so that both have the same exponent, added, and the result is rounded and normalized.

```
  e=5;  s=1.234567     (123456.7)
+ e=2;  s=1.017654     (101.7654)

  e=5;  s=1.234567
+ e=5;  s=0.001017654  (after shifting)
----------------------
  e=5;  s=1.235584654  (true sum: 123558.4654)
  e=5;  s=1.235585     (after rounding)
```

Floating Point Operations

- In extreme cases, the sum of two non-zero numbers may be equal to one of them.

```
  e=5;  s=1.234567
+ e=-3; s=9.876543

########################################

  e=5;  s=1.234567
+ e=5;  s=0.00000009876543 (after shifting)
----------------------
  e=5;  s=1.23456709876543 (true sum)
  e=5;  s=1.234567         (after rounding/normalization)
```

Floating Point Operations

- Catastrophic cancellation can result from the loss of precision when two close numbers are subtracted.

```
  e=5;  s=1.234571
- e=5;  s=1.234567
----------------
  e=5;  s=0.000004
  e=-1; s=4.000000 (after rounding/normalization)
```

Floating Point Operations

- To multiply, the significands are multiplied while the exponents are added, and the result is rounded and normalized.

```
  e=3;  s=4.734612
× e=5;  s=5.417242
-----------------------
  e=8;  s=25.648538980104 (true product)
  e=8;  s=25.64854        (after rounding)
  e=9;  s=2.564854        (after normalization)
```

Floating Point Problems

- Floating point addition and multiplication are not necessarily associative. That is (a + b) + c is not necessarily equal to a + (b + c).

```
  1234.567   (a)
+   45.67834 (b)
_____________
  1280.24534

rounds to   1280.245

  1280.245  (a + b)
+   0.0004 (c)
_____________
  1280.2454
rounds to   1280.245
```

```
a + (b + c):
    45.67834 (b)
+    0.0004  (c)
_____________
    45.67874

    45.67874 (b + c)
+ 1234.567   (a)
_____________
  1280.24574

rounds to 1280.246
```

Floating Point Problems

- Floating point addition and multiplication are not necessarily distributive. That is, (a + b) ×c may not be the same as a×c + b×c.

```
1234.567 × 3.333333 = 4115.223
1.234567 × 3.333333 = 4.115223
                      4115.223 + 4.115223 = 4119.338
but
1234.567 + 1.234567 = 1235.802
                      1235.802 × 3.333333 = 4119.340
```

Floating Point Problems

- Cancellation: subtraction of nearly equal operands may cause extreme loss of accuracy.
- Conversions to integer are not intuitive: converting (63.0/9.0) to integer yields 7, but converting (0.63/0.09) may yield 6.
  - This is because conversions generally truncate rather than round. Floor and ceiling functions may produce results which are off by one from the intuitively expected value.

Floating Point Problems

- Limited exponent range: results might overflow yielding infinity, or underflow yielding a subnormal number or zero.