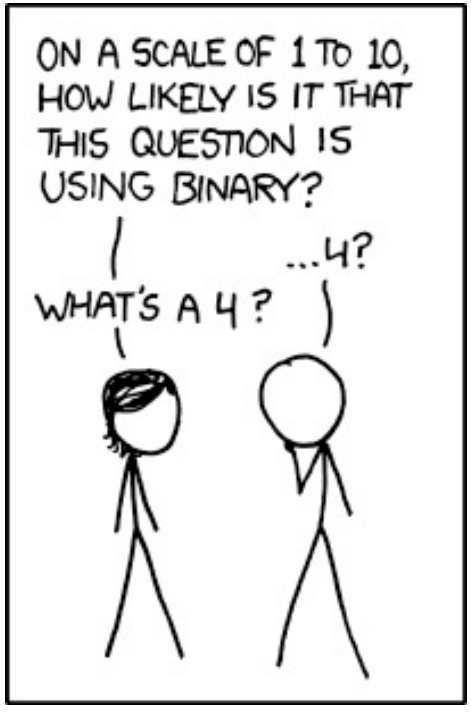
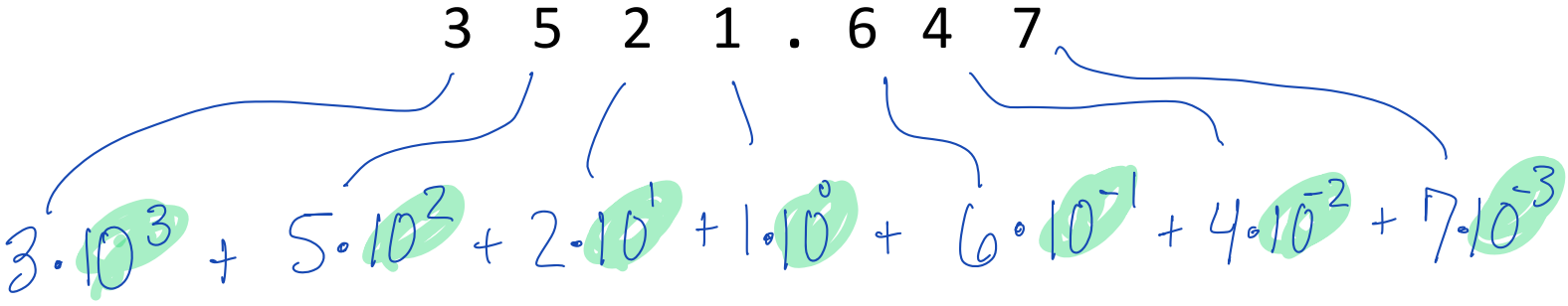


Floating-Point Numbers (the sequel)

Appendix B

“Decimal” points (points in radix 10)



Radix points: converting to decimal point

$1 \cdot 16^3 + 9 \cdot 16^2 + 4 \cdot 16^1 + 8 \cdot 16^0 + 11 \cdot 16^{-1} + 6 \cdot 16^{-2}$

Example 1

Hexadecimal: 1 9 4 8 . B 6

Binary: 0 0 0 1 1 0 0 1 0 1 0 0 1 0 0 0 . 1 0 1 1 0 1 1 0 0

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 -1 -2 -3 -4 -5 -6 -7 -8 -9

$2^{12} + 2^{11} + 2^8 + 2^6 + 2^3 + 2^{-1} + 2^{-3} + 2^{-4} + 2^{-6} + 2^{-7}$

Converting from decimal point to binary point

Convert 45.45 to binary

$45 = 22 \times 2 + 1$
$22 = 11 \times 2 + 0$
$11 = 5 \times 2 + 1$
$5 = 2 \times 2 + 1$
$2 = 1 \times 2 + 0$
$1 = 0 \times 2 + 1$

$45_{10} = 101101_2$

$0.45 \times 2 = 0.9$ *
$0.9 \times 2 = 1.8$
$0.8 \times 2 = 1.6$
$0.6 \times 2 = 1.2$
$0.2 \times 2 = 0.4$
$0.4 \times 2 = 0.8$

$0.45_{10} = 0.0\overline{011100}_2$

$45.45_{10} = 101101.\overline{011100}$

$= 1.01101\overline{011100} \times 2^5$

IEEE 754: A standard for binary floating point

- Different formats existed until the 1980s
- Late 1970s: a committee forms to standardize floating point arithmetic!
- Mainly the work of Berkeley Math Prof. William Kahan

Formats

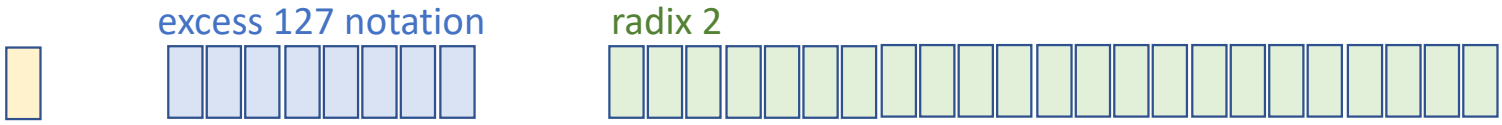
- **Single precision (32 bits)**
- Double precision (64 bits)
- Extended precision (80 bits) – mainly used in ALUs

sign (1 bit)

exponent (8 bits)

fraction (23 bits)

IEEE 754: A standard for binary floating point



significand = **1 .** fraction

notable implicit 1 and implicit binary point

Representing in traditional binary point:

$$number = significand \times 2^{exponent}$$

Example:

Convert 45.45 to IEEE 754 single precision format.

$45.45_{10} = 1.\underbrace{01101\,01\overline{1100}}_{\text{fraction}} \times 2^{(5)}$ exponent

0 — sign

10000100 — exponent (5) = 132 in excess-127

0110 1011 1001 1001 1001 1001

= 0 100 0010 0011 0101 1001 1001 1001 1001 in IEEE 754 form.

OR, 4 2 3 5 9 9 9 9 in hex.

Example 1: Converting 3F00000 (hex) from IEEE 754

$3\ F\ 0\ 0\ 0\ 0\ 0_{16}$

$0011\ 1111\ 0000\ 0000\ 0000\ 0000\ 0000_{2}$

sign (1 bit)	exponent (8 bits) — excess 127 notation	fraction (23 bits) — radix 2
0	0 1 1 1 1 1 1 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

significand = **1 .** fraction

notable implicit 1 and implicit binary point

Representing in traditional binary point:

$$number = significand \times 2^{exponent}$$

sign = positive

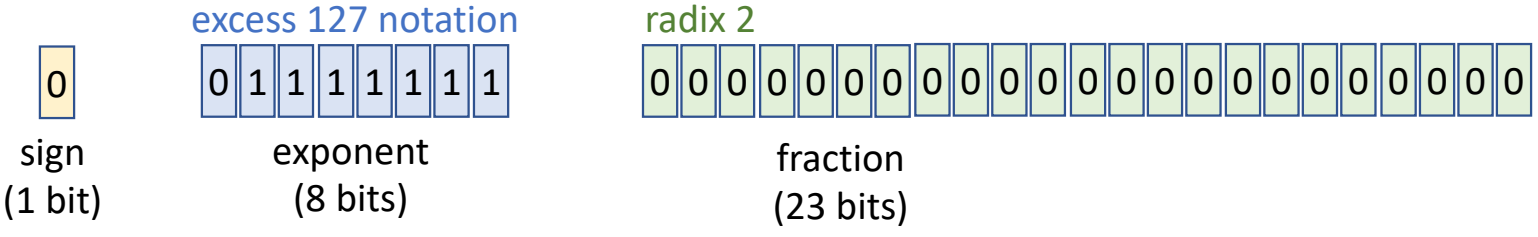
exponent = 126 in excess-127 ⇒ −1

fraction = 0

$+1.0 \times 2^{-1}$ ← exponent $= 2^{-1} = \boxed{0.5_{10}}$

↑ fraction

Example 2: Converting 3F80000 (hex) from IEEE 754



significand = **1 .** fraction

notable implicit 1 and implicit binary point

Representing in traditional binary point:

$$number = significand \times 2^{exponent}$$

Sign = positive

exponent = 127 in excess-127 ⇒ 0

fraction = 0

$+1.0 \times 2^{0}$ ← exponent

↑ fraction

$= 1_{10}$

Group work: Converting to IEEE 754

Convert 18.25 to IEEE 754 single precision format.