CSSE2010

# CSSE2010/CSSE7201
# Lecture 24

# Floating Point Numbers

School of Information Technology and Electrical Engineering
The University of Queensland

# Admin/Reminders

- Assignment 2 is due on Monday (01/11/21) 4:00PM Brisbane time.
- No labs on Friday (29/10/21) due to holiday. EX students on Friday can join other online sessions for the week.
- Late submissions will incur penalties unless you have an approved extensions.
- Any extension requests must be submitted through my-UQ portal.
- Exam review session on Wed 9-10am and another 2 hour review session during the revision week.

Please take 5 minutes provide your feedback for course CSSE2010/7201 and teaching SECAT surveys

(https://eval.uq.edu.au/)

CSSE2010

# Fractions – fixed-point

- How do we represent fractions?
  - We can arbitrarily specify the location of a binary point (fixed point notation), e.g.:

| $2^3$ | $2^2$ | $2^1$ | $2^0$ | • | $2^{-1}$ | $2^{-2}$ | $2^{-3}$ | $2^{-4}$ |
|---|---|---|---|---|---|---|---|---|
| 8 | 4 | 2 | 1 | | $^1/_2$ | $^1/_4$ | $^1/_8$ | $^1/_{16}$ |

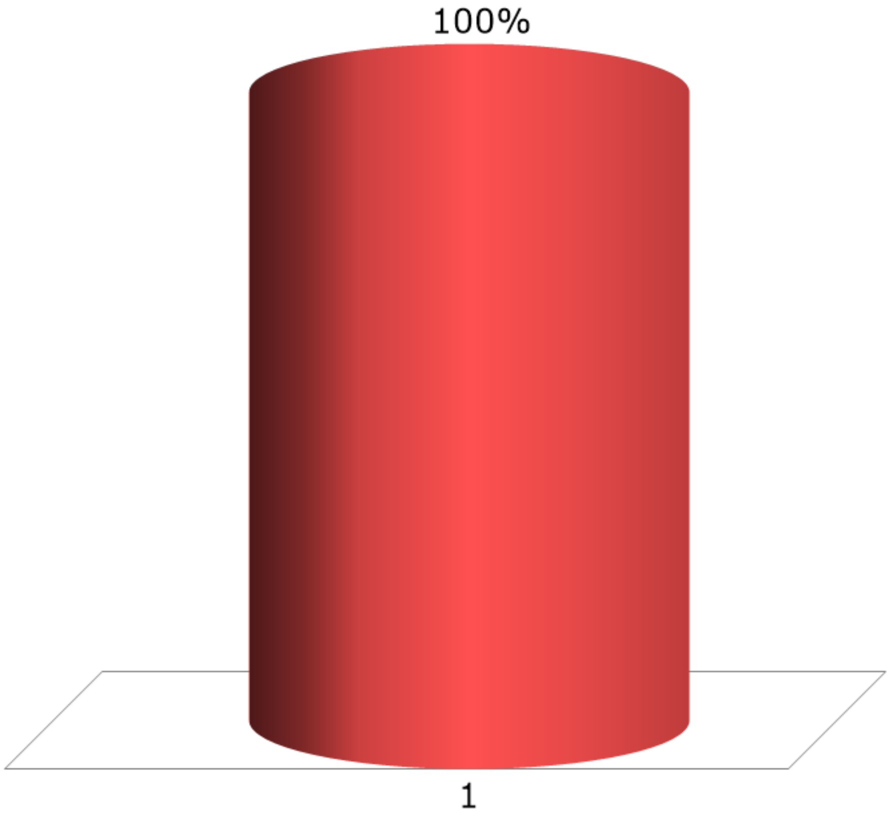
- Example: How do we represent 4.375?

| | | | | • | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |

- Problem: Not all numbers can be represented – how do we represent $4^1/_3$

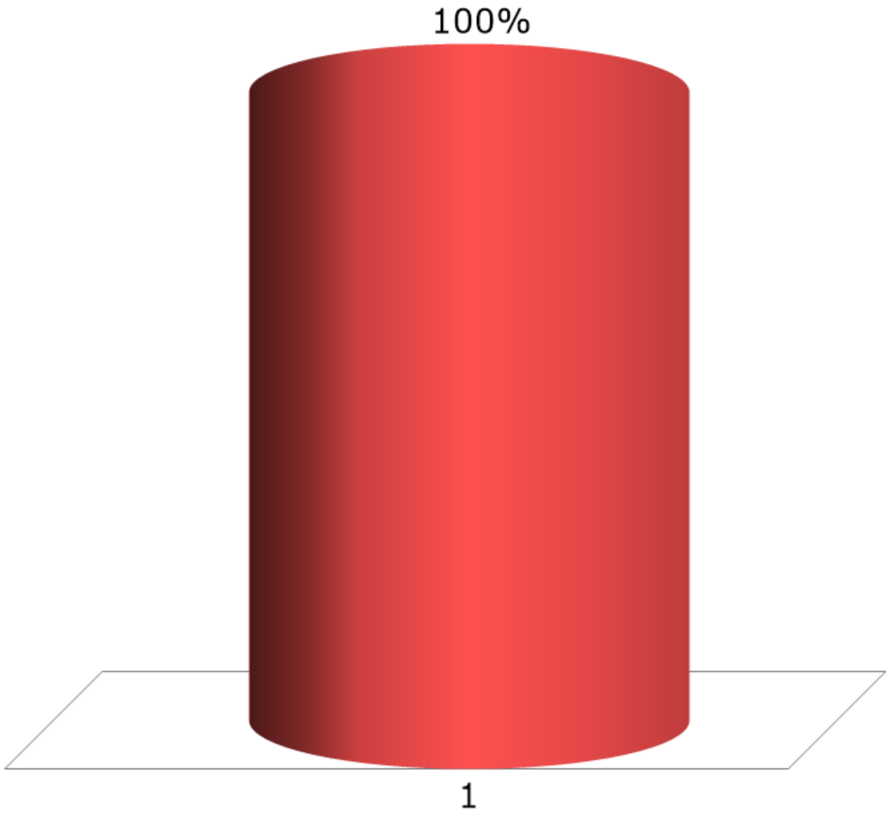
# What does the 8-bit number 10110101 represent in unsigned fixed-point notation with 5 integer bits and 3 fractional bits?

| Rank | Responses |
|---|---|
| 1 | |
| 2 | |
| 3 | |
| 4 | |
| 5 | |
| 6 | Other |



## What does the 8-bit number 10110101 represent in 2's complement fixed-point notation with 5 integer bits and 3 fractional bits?

| Rank | Responses |
|---|---|
| 1 | |
| 2 | |
| 3 | |
| 4 | |
| 5 | |
| 6 | Other |



# Real Numbers

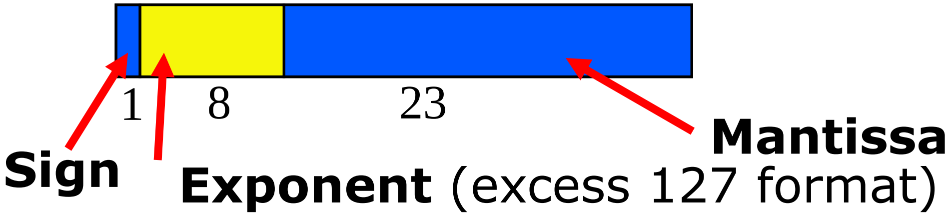
- Examples:
  - Mass of an electron: $9 \times 10^{-28}$ grams
    - 0.0000000000000000000000000009g
  - Mass of the sun: $2 \times 10^{33}$ grams
    - 2000000000000000000000000000000000g
- Scientific or Floating Point Notation:
  - number = $f \times 10^{e}$
    - f = fraction (or mantissa)
    - e = exponent
- Same applies to binary, e.g.
  - $2.0 \times 10^{33} = 1.1000101.. \times 2^{1101110}$

Fraction

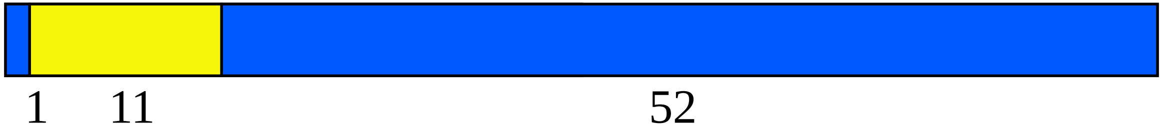
Exponent: Decimal 110

# IEEE Standard Floating Point

- Single Precision (32 bits)




- Double Precision (64 bits)
  - Exponent is excess 1023 format




- Mantissa always normalised (begins with 1)
  (so it doesn't actually need to be stored…)

# Floating Point Example

- Hexadecimal IEEE single precision representation of -23.25

# Finite Precision

- Not every number can be represented precisely
  - e.g. 10/3
  - So (10.0/3.0) * 3.0 may not equal 10.0

# IEEE Floating Point Number Characteristics

| Item | Single precision | Double precision |
|---|---|---|
| Bits in sign | 1 | 1 |
| Bits in exponent | 8 | 11 |
| Bits in fraction | 23 | 52 |
| Bits, total | 32 | 64 |
| Exponent system | Excess 127 | Excess 1023 |
| Exponent range | $-126$ to $+127$ | $-1022$ to $+1023$ |
| Smallest normalized number | $2^{-126}$ | $2^{-1022}$ |
| Largest normalized number | approx. $2^{128}$ | approx. $2^{1024}$ |
| Decimal range | approx. $10^{-38}$ to $10^{38}$ | approx. $10^{-308}$ to $10^{308}$ |
| Smallest denormalized number | approx. $10^{-45}$ | approx. $10^{-324}$ |

[Tanenbaum]

# IEEE Numerical Types

| | Sign | Exponent | Fraction |
|---|---|---|---|
| Normalized | ± | 0 < Exp < Max | Any bit pattern |
| Denormalized | ± | 0 | Any nonzero bit pattern |
| Zero | ± | 0 | 0 |
| Infinity | ± | 1 1 1 … 1 | 0 |
| Not a number | ± | 1 1 1 … 1 | Any nonzero bit pattern |

Sign bit

[Tanenbaum]

# Not A Number (NANs)

- Used to represent the result of functions that have no solution
  - e.g. infinity divided by infinity, taking the square root of a negative number
- When NaN results from a computation, it propagates as a NaN through all subsequent operations