

# Representing and Manipulating Information: Floating point

Yipeng Huang

Rutgers University

March 1, 2022

# Table of contents

## Announcements

Quizzes and programming assignments

Reading and class session plan

`monteCarloPi.c` Using floating point and random numbers to estimate PI

Fractions and fixed point representation

Floats: Overview

Floats: Normalized numbers

Normalized: exp field

Normalized: frac field

Normalized: example

Floats: Denormalized numbers

Denormalized: exp field

Denormalized: frac field

Denormalized: examples

Floats: Special cases

Floats: Properties

Floats: Summary

# Quizzes and programming assignments

## Short quiz 4

- ▶ Has been out as of this morning. Tuesday to Thursday. All about integers.

## Programming assignment 3

- ▶ Has been out, due Thursday before spring break.

# Reading and class session plan

## Reading: CS:APP Chapter 2

- ▶ Chapter 2: Representing and manipulating information
- ▶ Read Chapter 2.4: floating point.

## Recitation update

- ▶ Starting this tomorrow, March 2, Abhilash's recitation will move to Wednesdays 3:50 to 4:45 pm. CoRE 301.

# Table of contents

## Announcements

Quizzes and programming assignments

Reading and class session plan

`monteCarloPi.c` Using floating point and random numbers to estimate PI

Fractions and fixed point representation

Floats: Overview

Floats: Normalized numbers

Normalized: exp field

Normalized: frac field

Normalized: example

Floats: Denormalized numbers

Denormalized: exp field

Denormalized: frac field

Denormalized: examples

Floats: Special cases

Floats: Properties

Floats: Summary

# monteCarloPi.c Using floating point and random numbers to estimate PI

# Table of contents

## Announcements

Quizzes and programming assignments

Reading and class session plan

`monteCarloPi.c` Using floating point and random numbers to estimate PI

Fractions and fixed point representation

Floats: Overview

Floats: Normalized numbers

Normalized: exp field

Normalized: frac field

Normalized: example

Floats: Denormalized numbers

Denormalized: exp field

Denormalized: frac field

Denormalized: examples

Floats: Special cases

Floats: Properties

Floats: Summary

# Unsigned fixed-point binary for fractions

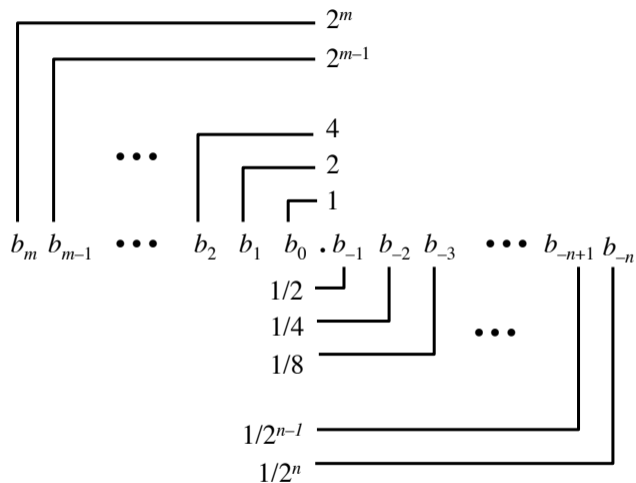


Figure: Fractional binary. Image credit CS:APP



# Unsigned fixed-point binary for fractions

unsigned fixed-point char example	weight in decimal
1000.0000	8
0100.0000	4
0010.0000	2
0001.0000	1
0000.1000	0.5
0000.0100	0.25
0000.0010	0.125
0000.0001	0.0625

**Table:** Weight of each bit in an example fixed-point binary number

- ▶  $.625 = .5 + .125 = 0000.1010_2$
- ▶  $1001.1000_2 = 9 + .5 = 9.5$

## Signed fixed-point binary for fractions

signed fixed-point char example	weight in decimal
1000.0000	-8
0100.0000	4
0010.0000	2
0001.0000	1
0000.1000	0.5
0000.0100	0.25
0000.0010	0.125
0000.0001	0.0625

Table: Weight of each bit in an example fixed-point binary number

- ▶  $-.625 = -8 + 4 + 2 + 1 + 0 + .25 + .125 = 1111.0110_2$
- ▶  $1001.1000_2 = -8 + 1 + .5 = -6.5$

## Limitations of fixed-point

- ▶ Can only represent numbers of the form  $x/2^k$
- ▶ Cannot represent numbers with very large magnitude (great range) or very small magnitude (great precision)

# Table of contents

## Announcements

Quizzes and programming assignments

Reading and class session plan

`monteCarloPi.c` Using floating point and random numbers to estimate PI

Fractions and fixed point representation

Floats: Overview

Floats: Normalized numbers

Normalized: exp field

Normalized: frac field

Normalized: example

Floats: Denormalized numbers

Denormalized: exp field

Denormalized: frac field

Denormalized: examples

Floats: Special cases

Floats: Properties

Floats: Summary

# Floating point numbers

## Avogadro's number

$$+6.02214 \times 10^{23} \text{ mol}^{-1}$$

## Scientific notation

- ▶ sign
- ▶ mantissa or significand
- ▶ exponent

# Floating point numbers

## Before 1985

1. Many floating point systems.
2. Specialized machines such as Cray supercomputers.
3. Some machines with specialized floating point have had to be kept alive to support legacy software.

## After 1985

1. IEEE Standard 754.
2. A floating point standard designed for good numerical properties.
3. Found in almost every computer today, except for tiniest microcontrollers.

## Recent

1. Need for both lower precision and higher range floating point numbers.
2. Machine learning / neural networks. Low-precision tensor network processors.

# Floats and doubles

Single precision



Double precision



Figure: The two standard formats for floating point data types. Image credit CS:APP

## Floats and doubles

property	half*	float	double
total bits	16	32	64
s bit	1	1	1
exp bits	5	8	11
frac bits	10	23	52
C printf() format specifier	None	"%f"	"%lf"

Table: Properties of floats and doubles



# The IEEE 754 number line

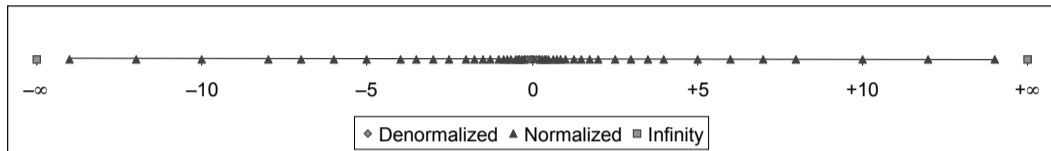


Figure: Full picture of number line for floating point values. Image credit CS:APP

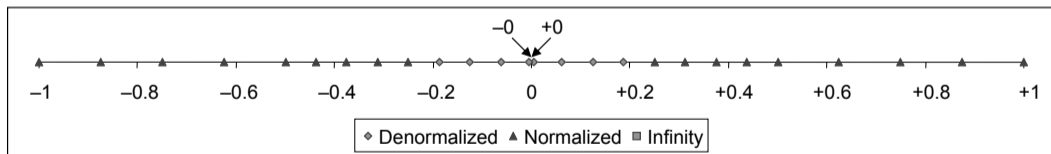


Figure: Zoomed in number line for floating point values. Image credit CS:APP

# Different cases for floating point numbers

$$\text{Value of the floating point number} = (-1)^s \times M \times 2^E$$

- ▶  $E$  is encoded the exp field
- ▶  $M$  is encoded the frac field

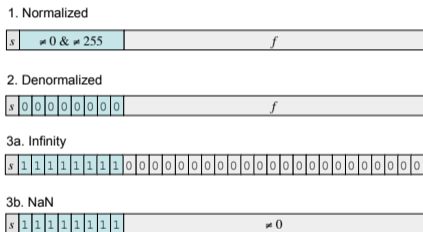


Figure: Different cases within a floating point format. Image credit CS:APP

## Normalized and denormalized numbers

Two different cases we need to consider for the encoding of  $E, M$

# Table of contents

## Announcements

Quizzes and programming assignments

Reading and class session plan

`monteCarloPi.c` Using floating point and random numbers to estimate PI

Fractions and fixed point representation

Floats: Overview

Floats: Normalized numbers

Normalized: exp field

Normalized: frac field

Normalized: example

Floats: Denormalized numbers

Denormalized: exp field

Denormalized: frac field

Denormalized: examples

Floats: Special cases

Floats: Properties

Floats: Summary

## Normalized: exp field

For normalized numbers,

$$0 < \text{exp} < 2^k - 1$$

- ▶ exp is a  $k$ -bit unsigned integer

### Bias

- ▶ need a bias to represent negative exponents
- ▶ bias =  $2^{k-1} - 1$
- ▶ bias is the  $k$ -bit unsigned integer: 011..111

For normalized numbers,

$$E = \text{exp} - \text{bias}$$

In other words,  $\text{exp} = E + \text{bias}$

	property	float	double
	k	8	11
	bias	127	1023
smallest E (greatest precision)		-126	-1022
largest E (greatest range)		127	1023

Table: Summary of normalized exp field

Normalized: frac field

$M = 1.\text{frac}$

## Normalized: example

- ▶ 12.375 to single-precision floating point
- ▶ sign is positive so  $s=0$
- ▶ binary is  $1100.011_2$
- ▶ in other words it is  $1.100011_2 \times 2^3$
- ▶  $\text{exp} = E + \text{bias} = 3 + 127 = 130 = 1000\_0010_2$
- ▶  $M = 1.100011_2 = 1.\text{frac}$
- ▶  $\text{frac} = 100011$

# Table of contents

## Announcements

Quizzes and programming assignments

Reading and class session plan

`monteCarloPi.c` Using floating point and random numbers to estimate PI

Fractions and fixed point representation

Floats: Overview

Floats: Normalized numbers

Normalized: exp field

Normalized: frac field

Normalized: example

Floats: Denormalized numbers

Denormalized: exp field

Denormalized: frac field

Denormalized: examples

Floats: Special cases

Floats: Properties

Floats: Summary

# The IEEE 754 number line

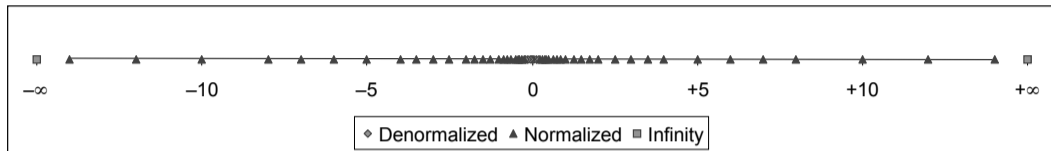


Figure: Full picture of number line for floating point values. Image credit CS:APP

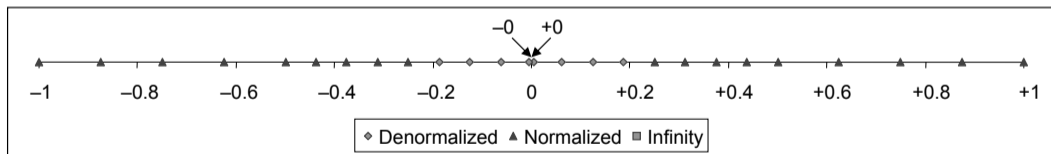


Figure: Zoomed in number line for floating point values. Image credit CS:APP



## Denormalized: exp field

For denormalized numbers,  $\text{exp} = 0$

### Bias

- ▶ need a bias to represent negative exponents
- ▶  $\text{bias} = 2^{k-1} - 1$
- ▶ bias is the  $k$ -bit unsigned integer:  
011..111

For denormalized numbers,  
 $E = 1\text{-bias}$

property	float	double
k	8	11
bias	127	1023
E	-126	-1022

Table: Summary of denormalized exp field

## Denormalized: frac field

$M = 0.\text{frac}$

value represented leading with 0

# Denormalized: examples

# Table of contents

## Announcements

Quizzes and programming assignments

Reading and class session plan

`monteCarloPi.c` Using floating point and random numbers to estimate PI

Fractions and fixed point representation

Floats: Overview

Floats: Normalized numbers

Normalized: exp field

Normalized: frac field

Normalized: example

Floats: Denormalized numbers

Denormalized: exp field

Denormalized: frac field

Denormalized: examples

Floats: Special cases

Floats: Properties

Floats: Summary

## Floats: Special cases

number class	when it arises	exp field	frac field
+0 / -0		0	0
+infinity / -infinity	overflow or division by 0	$2^k - 1$	0
NaN not-a-number	illegal ops. such as $\sqrt{-1}$ , inf-inf, inf*0	$2^k - 1$	non-0

Table: Summary of special cases

# Table of contents

## Announcements

Quizzes and programming assignments

Reading and class session plan

`monteCarloPi.c` Using floating point and random numbers to estimate PI

Fractions and fixed point representation

Floats: Overview

Floats: Normalized numbers

Normalized: exp field

Normalized: frac field

Normalized: example

Floats: Denormalized numbers

Denormalized: exp field

Denormalized: frac field

Denormalized: examples

Floats: Special cases

Floats: Properties

Floats: Summary

# Floats: Properties

fp add is not associattive

# Table of contents

## Announcements

Quizzes and programming assignments

Reading and class session plan

`monteCarloPi.c` Using floating point and random numbers to estimate PI

Fractions and fixed point representation

Floats: Overview

Floats: Normalized numbers

Normalized: exp field

Normalized: frac field

Normalized: example

Floats: Denormalized numbers

Denormalized: exp field

Denormalized: frac field

Denormalized: examples

Floats: Special cases

Floats: Properties

Floats: Summary



## Floats: Summary

	normalized	denormalized
value of number	$(-1)^s \times M \times 2^E$	$(-1)^s \times M \times 2^E$
E	E = exp-bias	E = -bias + 1
bias	$2^{k-1} - 1$	$2^{k-1} - 1$
exp	$0 < exp < (2^k - 1)$	$exp = 0$
M	M = 1.frac M has implied leading 1	M = 0.frac M has leading 0
	greater range large magnitude numbers denser near origin	greater precision small magnitude numbers evenly spaced

Table: Summary of normalized and denormalized numbers