

Data Representation: Floating Point Normalized Numbers

Yipeng Huang

Rutgers University

October 16, 2026

Data Representation: Floating Point Normalized Numbers

Announcements

Canvas timed quiz 5 and programming assignment 3

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

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: Summary

Data Representation: Floating Point Normalized Numbers

Announcements

Canvas timed quiz 5 and programming assignment 3

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

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: Summary

Canvas timed quiz 5 and programming assignment 3

Quiz 5

1. Spanning Friday 10/16 - Friday 10/23.
2. 60 minutes.
3. Three tries.
4. The extreme values of floating point numbers.

Programming assignment 3

1. Due Tuesday 10/27.
2. Binary representations of integers and floating point numbers.

Bit shifting

$\ll$ N Left shift by N bits

- ▶ multiplies by 2^N
- ▶ $2 \ll 3 = 0000_0010_2 \ll 3 = 0001_0000_2 = 16 = 2 * 2^3$
- ▶ $-2 \ll 3 = 1111_1110_2 \ll 3 = 1111_0000_2 = -16 = -2 * 2^3$

$\gg$ N Right shift by N bits

- ▶ divides by 2^N
- ▶ $16 \gg 3 = 0001_0000_2 \gg 3 = 0000_0010_2 = 2 = 16/2^3$
- ▶ $-16 \gg 3 = 1111_0000_2 \gg 3 = 1111_1110_2 = -2 = -16/2^3$

Bit Shifting: Logical vs. Arithmetic

- ▶ **Left Shift ($\ll$):** Pads zeros on the right. Mathematically equivalent to multiplying by a power of 2.
- ▶ **Right Shift ($\gg$):** The compiler maps this to different hardware instructions based on data type:
 - ▶ **Unsigned (Logical Shift):** Pads zeros on the left.
 - ▶ **Signed (Arithmetic Shift):** Pads the sign bit on the left to preserve the sign.
- ▶ **Rounding Pitfall:** Arithmetic right-shifting a negative number rounds toward $-\infty$, not zero (e.g., $-3 \gg 1$ yields -2).

Data Representation: Floating Point Normalized Numbers

Announcements

Canvas timed quiz 5 and programming assignment 3

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

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: Summary

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

Data Representation: Floating Point Normalized Numbers

Announcements

Canvas timed quiz 5 and programming assignment 3

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

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: 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

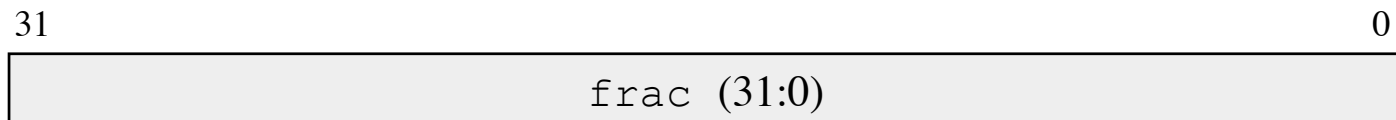


Figure: The two standard formats for floating point data types. Image credit CS:APP

Floats and doubles

textbook
tiny float

8 bits
1 sign
4 exp
3 frac

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

quad

128

1

15

112

The IEEE 754 number line

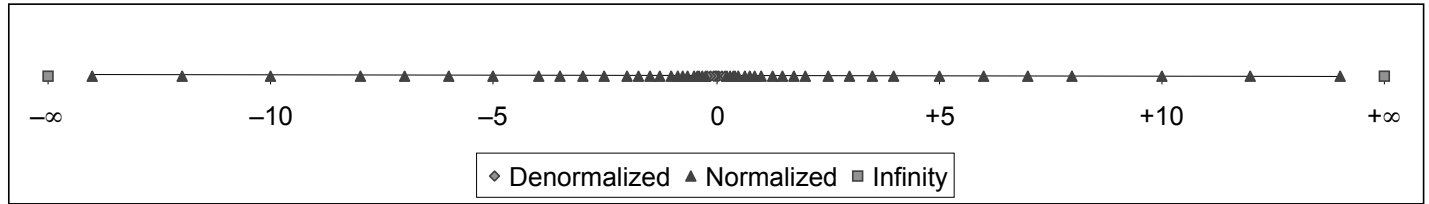


Figure: Full picture of number line for floating point values. Image credit CS:APP

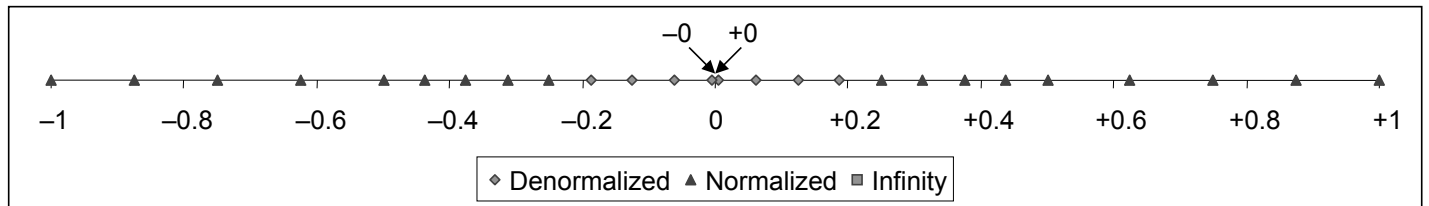


Figure: Zoomed in number line for floating point values. Image credit CS:APP

Different cases for floating point numbers

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

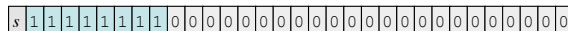
1. Normalized



2. Denormalized



3a. Infinity



3b. NaN



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

Data Representation: Floating Point Normalized Numbers

Announcements

Canvas timed quiz 5 and programming assignment 3

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

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: Summary

Data Representation: Floating Point Normalized Numbers

Announcements

Canvas timed quiz 5 and programming assignment 3

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

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: Summary

PI in tiny float

3.1415926

$s = 0$

$f = 0b00011.001\dots$

$$\begin{array}{r} 3.1415926 \\ 2 \\ \hline 1.1415926 \\ 1.0000000 \\ \hline .1415926 \\ .1250000 \\ \hline \end{array}$$

0	0	0	1	1	0	0	1
16	8	4	2	1	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$

1.1001×2^1

tiny float

1 bit sign

4 bits exp

3 frac

0	1	0	0	0	1	0	0
$\frac{0}{8}$	$\frac{1000}{\text{exp}}$				$\frac{100}{\text{frac}}$		

$E = 1$

$\text{exp} = E + \text{bias} = 1 + 2^{k-1} - 1 = 1 + 2^{4-1} - 1 = 8 = 0b1000$

$$val = (-1)^s \cdot M \cdot 2^E$$

$$M: 1. \text{frac} = 0b 1.100 = 1.5 = \frac{3}{2}$$

$$\begin{aligned} E: \text{exp} - \text{bias} &= 0b 1000 - \text{bias} \\ &= 8 - (2^{k-1} - 1) \\ &= 8 - (2^3 - 1) \\ &= 1 \end{aligned}$$

$$\begin{aligned} val &= (-1)^0 \cdot \frac{3}{2} \cdot 2^1 \\ &= 3. \end{aligned}$$

$$2^{-5}$$

$$E = -5$$

$$\begin{aligned} \text{exp} = E + \text{bias} &= -5 + 2^{k-1} - 1 = -5 + 2^{4-1} - 1 = -5 + 8 - 1 \\ &= 2 \\ &= 010010 \end{aligned}$$

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
- $\text{bias} = 2^{k-1} - 1$
- 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

Data Representation: Floating Point Normalized Numbers

Announcements

Canvas timed quiz 5 and programming assignment 3

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

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: Summary

Normalized: frac field

$$M = 1.\text{frac}$$

Data Representation: Floating Point Normalized Numbers

Announcements

Canvas timed quiz 5 and programming assignment 3

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

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: Summary

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 $1.100011_2 \times 2^3$
- ▶ $\text{exp} = E + \text{bias} = 3 + 127 = 130 = 1000_0010_2$
- ▶ $M = 1.100011_2 = 1.\text{frac}$
- ▶ $\text{frac} = 100011$

Data Representation: Floating Point Normalized Numbers

Announcements

Canvas timed quiz 5 and programming assignment 3

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

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: Summary

The IEEE 754 number line

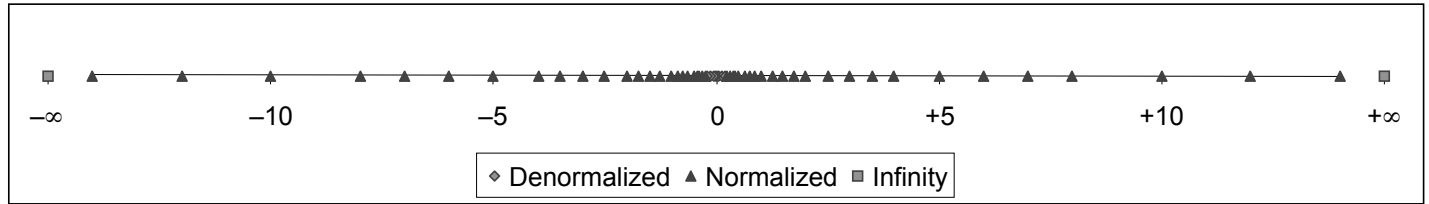


Figure: Full picture of number line for floating point values. Image credit CS:APP

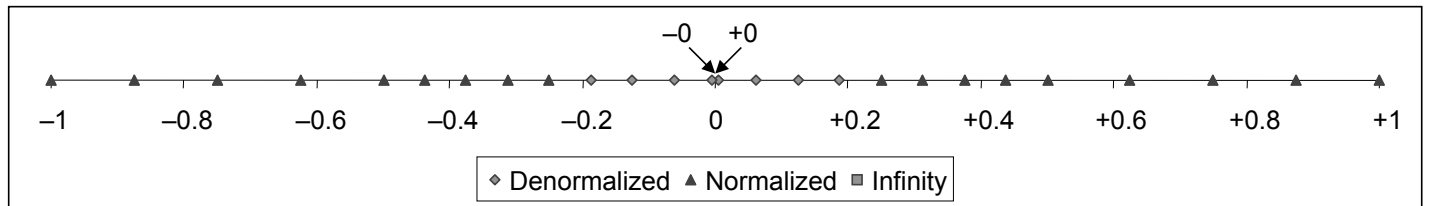


Figure: Zoomed in number line for floating point values. Image credit CS:APP

Data Representation: Floating Point Normalized Numbers

Announcements

Canvas timed quiz 5 and programming assignment 3

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

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: Summary

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

Data Representation: Floating Point Normalized Numbers

Announcements

Canvas timed quiz 5 and programming assignment 3

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

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: Summary

Denormalized: frac field

$M = o.\text{frac}$

value represented leading with 0

Data Representation: Floating Point Normalized Numbers

Announcements

Canvas timed quiz 5 and programming assignment 3

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

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: Summary

Denormalized: examples

Data Representation: Floating Point Normalized Numbers

Announcements

Canvas timed quiz 5 and programming assignment 3

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

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: 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

Data Representation: Floating Point Normalized Numbers

Announcements

Canvas timed quiz 5 and programming assignment 3

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

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: 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	$0 < \text{exp} < (2^k - 1)$	$\text{exp} = 0$
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