Data representation: Floating Point Normalized Numbers and Denormalized Numbers

Yipeng Huang

Rutgers University

October 16, 2025

Table of contents

Floating point numbers

Avogadro's number

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

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

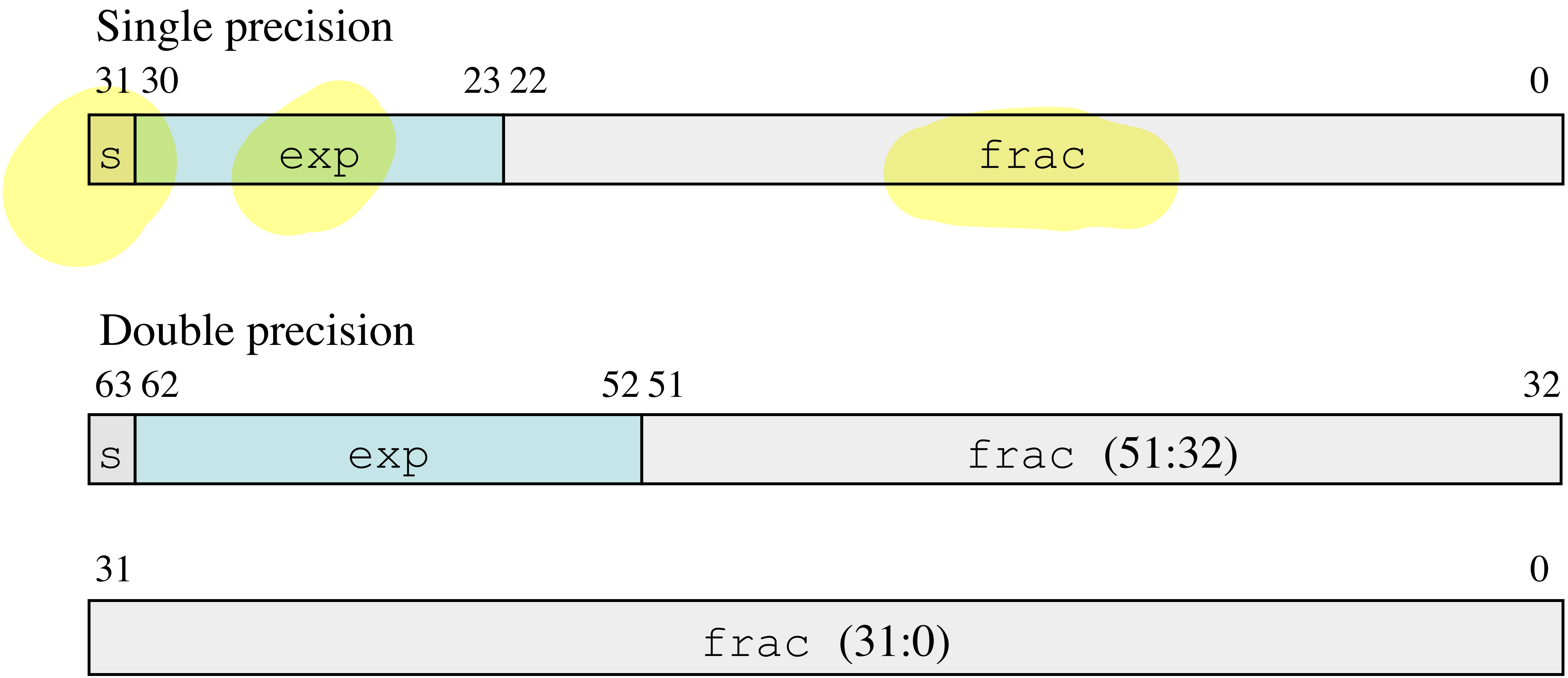

Figure: The two standard formats for floating point data types. Image credit CS:APP

Floats and doubles

textbook tiny float	property	half*	float	double	quad
8	total bits	16	32	64	128
1	s bit	1	1	1	1
4	exp bits	5	8	11	15
3	frac bits	10	23	52	112
	C printf() format specifier	None	"%f"	"%lf"	

Table: Properties of floats and doubles

The IEEE 754 number line

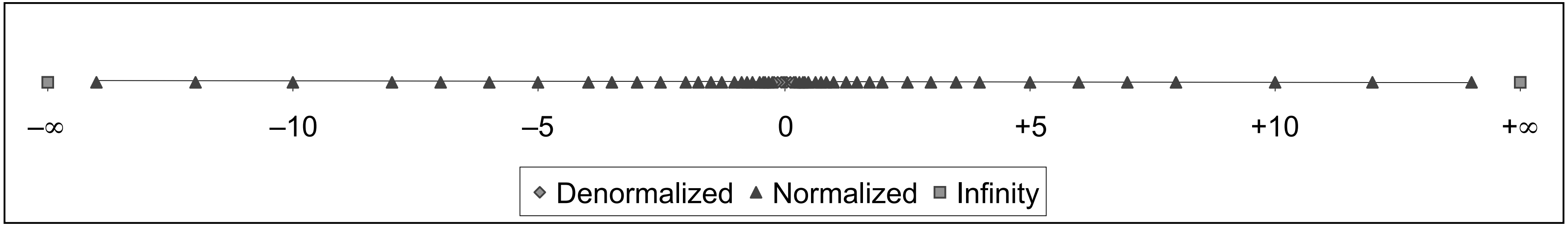

Figure: Full picture of number line for floating point values. Image credit CS:APP

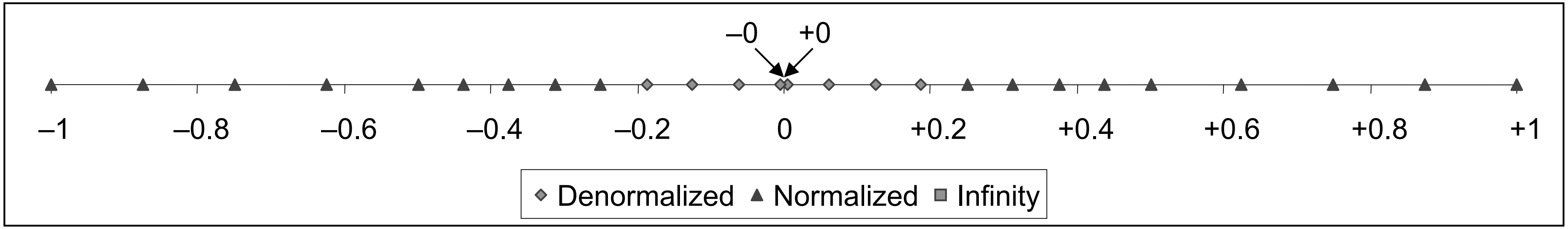

Figure: Zoomed in number line for floating point values. Image credit CS:APP

Different cases for floating point numbers

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

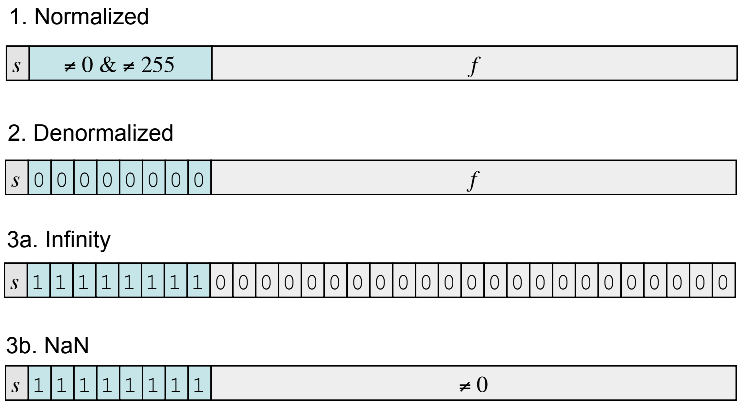

Figure: Different cases within a floating point format. Image credit CS:APP

Normalized and denormalized numbers

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

Table of contents

Normalized: exp field

For normalized numbers,
$0 < \exp < 2^k - 1$

- 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 = exp-bias

In other words, exp = E+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.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.frac
- frac = 100011

infamily

0_1111_000

largest (normalized) number

s exp frac

0_1110_111

$$(-1)^S \times (1.frac) \times 2^E = (-1)^0 \cdot (1.frac) \times 2^7$$

$$= +1 \cdot 1.875 \times 128$$

$$= \frac{15}{8} \times 128 = 15 \times 16$$

$$= 30 \times 8$$

$$= 240$$

$$E = exp - bias$$

$$= 14 - bias.$$

$$= 14 - (2^{(k-1)} - 1)$$

$$= 14 - (2^{(4-1)} - 1)$$

$$= 14 - (2^3 - 1) \quad 0111_b$$

$$= 14 - 7 = 7$$

Mantissa: $1.frac$

$$= \underset{1}{1}.\underset{\frac{1}{2}}{1}\underset{\frac{1}{4}}{1}\underset{\frac{1}{8}}{1}_b = 1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} = \frac{15}{8} = 1.875$$

-239

128	64	32	16	8	4	2	1	
1	1	1	0	1	1	1	1	.

$$\begin{array}{r} 239 \\ -128 \\ \hline 111 \\ -64 \\ \hline 47 \\ -32 \\ \hline 15 \end{array}$$

Mantissa: $1.\underbrace{111}_{\text{frac}}1111$

$239_{10} = 1.1110000 \times 2^7$

$E = exp - bias$

$7 = exp - (2^{(k-1)} - 1)$

$7 = exp - 7$

$exp = 14$

$239 \rightarrow 0\_1110\_111 \rightarrow (-1)^s \cdot (1.frac) \cdot 2^E$

$= (1.110_2) \cdot 128$

$= (1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8}) \cdot 128 = \frac{15}{8} \cdot 128 = 32 \cdot 7 = 240$

epsilon

One

s exp frac

0 _ 0111 _ 000

Mantissa = 1.000

$1.0 = (1.0) \cdot 2^0$

$E = exp - bias$

$0 = exp - 7$

$exp = 7 = 0b111$

smallest (normalized) number <non zero>

exp frac

0 _ 0001 _ 000

$$(-1)^s \cdot (1.\text{frac}) \cdot 2^{E}$$

$(\text{exp}-\text{bias})$

$$= (+1) \cdot (1.000) \cdot 2^{(1-7)}$$

$$= 1 \cdot 2^{(1-7)}$$

$$= 1 \cdot 2^{-6} = \frac{1}{64}$$

Zero

s	exp	frac.
0_	0000_	000

	$+$	$\times$
identity	0	1
commutative	$A+B=B+A$ ✓	$AB=BA$ ✓
associativity	$(A+B)+C=A+(B+C)$ ✗	$(AB)C=A(BC)$ ✗

$240+1-64$

$(240+1)-64=\infty-64=\infty$

$240+(1-64)=240-63=177$

distributative

$A(B+C)=AB+AC$

$240(64-64)=240\times 0=0$

$240\cdot 64-240\cdot 64$

$=\infty-\infty$

$=\text{NaN}$

inverse

Yes, negation always exists.

Table of contents

The IEEE 754 number line

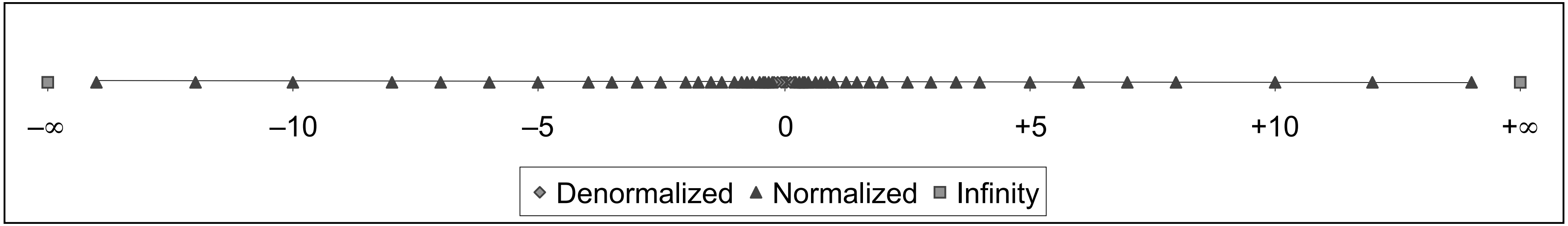

Figure: Full picture of number line for floating point values. Image credit CS:APP

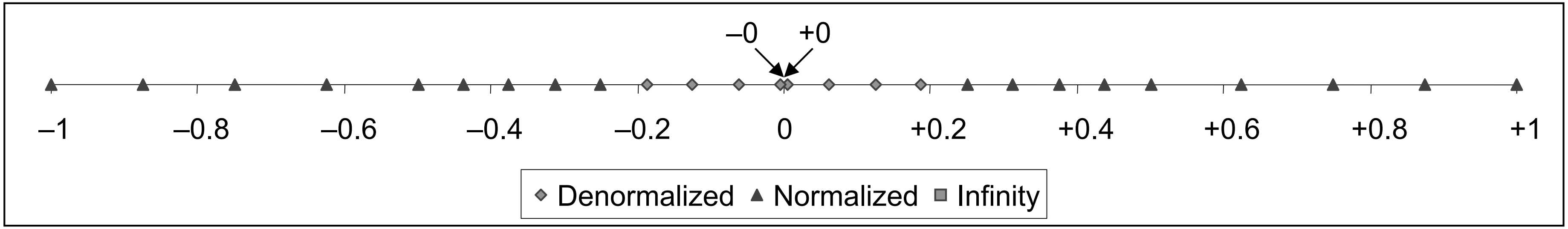

Figure: Zoomed in number line for floating point values. Image credit CS:APP

Denormalized: exp field

For denormalized numbers, exp $= 0$

Bias

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

For denormalized numbers,
E = 1-bias

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

Table: Summary of denormalized exp field

Denormalized: frac field

M = 0.frac

value represented leading with 0

Denormalized: examples

largest denormalized number

smallest denormalized number

Table of contents

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

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 = 0.frac
	M has implied leading 1	M has leading 0
	greater range	greater precision
	large magnitude numbers	small magnitude numbers
	denser near origin	evenly spaced

Table: Summary of normalized and denormalized numbers

I. Why use bias encoding?

II. Why have denormalized numbers?

PROPERTIES.