

Data Representation: Floating Point Denormalized Numbers

Yipeng Huang

Rutgers University

October 20, 2026

Data Representation: Floating Point Denormalized Numbers

Announcements

Canvas timed quiz 5 and programming assignment 3

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: Understanding its design

Deep understanding 1: Why is exp field encoded using bias?

Deep understanding 2: Why have denormalized numbers?

Deep understanding 3: Why is bias chosen to be $2^{k-1} - 1$?

Floats: Properties

Floating point multiplication

Data Representation: Floating Point Denormalized Numbers

Announcements

Canvas timed quiz 5 and programming assignment 3

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: Understanding its design

Deep understanding 1: Why is exp field encoded using bias?

Deep understanding 2: Why have denormalized numbers?

Deep understanding 3: Why is bias chosen to be $2^{k-1} - 1$?

Floats: Properties

Floating point multiplication

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.

Data Representation: Floating Point Denormalized Numbers

Announcements

Canvas timed quiz 5 and programming assignment 3

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: Understanding its design

Deep understanding 1: Why is exp field encoded using bias?

Deep understanding 2: Why have denormalized numbers?

Deep understanding 3: Why is bias chosen to be $2^{k-1} - 1$?

Floats: Properties

Floating point multiplication

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

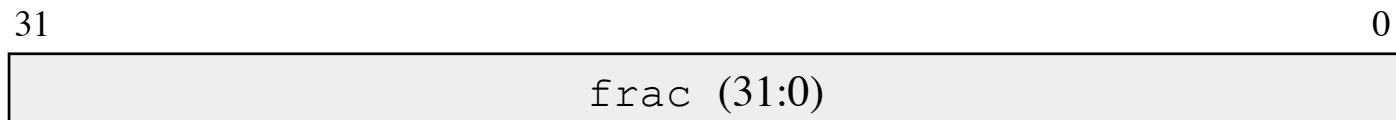


Figure: The two standard formats for floating point data types. Image credit CS:APP

Floats and doubles

Quarter



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"

quad
128
1
13

Table: Properties of floats and doubles

Precision Trade-offs: Scientific vs. AI Era

- ▶ **Relative Precision:** Floats dedicate dense precision near zero and wider spacing at large magnitudes—maintaining near-constant *relative error*.
- ▶ **Historical Trend (Scientific Computing):** Pushed for higher precision (Double 64-bit → Quadruple 128-bit) for exact simulation.
- ▶ **Modern Trend (AI / Machine Learning):** Pushing toward ****lower precision**** (Half 16-bit, Google `bfloat16`).
 - ▶ Biological synaptic signals and deep neural networks are inherently noisy.
 - ▶ LLMs derive capacity from sheer weight count, not individual weight precision.
 - ▶ Dramatically reduces memory bandwidth and TPU/GPU power consumption.

The IEEE 754 number line

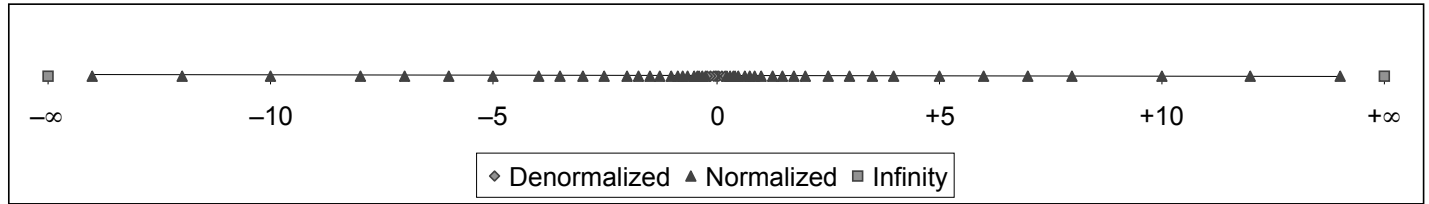


Figure: Full picture of number line for floating point values. Image credit CS:APP

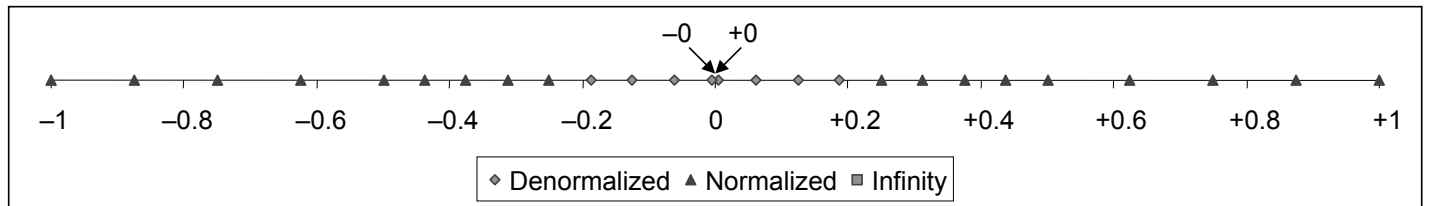


Figure: Zoomed in number line for floating point values. Image credit CS:APP

Data Representation: Floating Point Denormalized Numbers

Announcements

Canvas timed quiz 5 and programming assignment 3

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: Understanding its design

Deep understanding 1: Why is exp field encoded using bias?

Deep understanding 2: Why have denormalized numbers?

Deep understanding 3: Why is bias chosen to be $2^{k-1} - 1$?

Floats: Properties

Floating point multiplication

Data Representation: Floating Point Denormalized Numbers

Announcements

Canvas timed quiz 5 and programming assignment 3

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: Understanding its design

Deep understanding 1: Why is exp field encoded using bias?

Deep understanding 2: Why have denormalized numbers?

Deep understanding 3: Why is bias chosen to be $2^{k-1} - 1$?

Floats: Properties

Floating point multiplication

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 Denormalized Numbers

Announcements

Canvas timed quiz 5 and programming assignment 3

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: Understanding its design

Deep understanding 1: Why is exp field encoded using bias?

Deep understanding 2: Why have denormalized numbers?

Deep understanding 3: Why is bias chosen to be $2^{k-1} - 1$?

Floats: Properties

Floating point multiplication

Normalized: frac field

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

Data Representation: Floating Point Denormalized Numbers

Announcements

Canvas timed quiz 5 and programming assignment 3

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: Understanding its design

Deep understanding 1: Why is exp field encoded using bias?

Deep understanding 2: Why have denormalized numbers?

Deep understanding 3: Why is bias chosen to be $2^{k-1} - 1$?

Floats: Properties

Floating point multiplication

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

Worked Example: Encoding π in Tiny Float (8-bit)

- ▶ **Tiny Float Specs:** 1 Sign bit, 4 Exponent bits ($k = 4$, bias = 7), 3 Fraction bits.
- ▶ **Target Value:** $\pi \approx 3.14159..._{10}$
- ▶ **Step 1: Sign** $\rightarrow$ Positive $\implies S = 0$.
- ▶ **Step 2: Binary Fixed-Point & Normalize**
 - ▶ $3.14159..._{10} = 11.001001..._2 \implies \underline{1.1001001...}_2 \times 2^1$ (so $E = 1$).
Handwritten: 3.125, 2 1 .125, arrow pointing to the 1 in 1.1001001...
- ▶ **Step 3: Biased EXP Field** $\rightarrow \text{EXP} = E + \text{bias} = 1 + 7 = 8 = 1000_2$.
- ▶ **Step 4: Frac Field (Rounding to 3 bits)**
 - ▶ Normalized bits: $1.1001001..._2$
 - ▶ Lower neighbor (100): $1.100_2 \times 2^1 = 3.0$ (Error = $|3.14159 - 3.0| = 0.14159$)
 - ▶ Upper neighbor (101): $1.101_2 \times 2^1 = 3.25$ (Error = $|3.14159 - 3.25| = 0.10841$)
 - ▶ **Round-to-Nearest:** 3.25 is closer $\implies 101_2$.
- ▶ **Encoded Bits:** 0 | 1000 | 101
- ▶ **Decoding Back:** $1 \times (1.101_2) \times 2^1 = 1.625 \times 2 = 3.25$.

infinity

0-1111-000

largest normalized pos number

0-1110-111

$$E = \text{exp} - \text{bias} = 14 - 7 = 7$$

$$1.111_2 = 1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} = \frac{8}{8} + \frac{4}{8} + \frac{2}{8} + \frac{1}{8} = \frac{15}{8}$$

epsilon $(-1)^0 \cdot \frac{15}{8} \cdot 2^7 = \frac{15}{2^1} \cdot 2^7 = 15 \cdot 2^6 = 15 \cdot 16 = 30 \cdot 8 = 240$

0-0111-001

$$(-1)^0 \cdot (1.001_2) \cdot 2^{7-7} = \left(1 + \frac{1}{8}\right) \cdot$$

one $= 1.0_{10} = 1.00000_2 = 1.0 \cdot 2^0$ $E = 0$

$$\text{Exp} = E + \text{bias}$$

0-0111-000

$$= 0 + \text{bias}$$

Smallest positive normalized number

0-0001-000

$$= 0 + 2^{k-1} - 1$$

$$= 0 + 2^{4-1} - 1$$

$$= 8 - 1 = 7_{10} = 0111_2$$

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

$$= 1 - 7 = -6$$

$$(-1)^0 \cdot (1.000) \cdot 2^E$$

$$= 1.0 \cdot 2^{-6} = \frac{1}{64}$$

most neg normalized number

1-1110-111

-240

Data Representation: Floating Point Denormalized Numbers

Announcements

Canvas timed quiz 5 and programming assignment 3

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: Understanding its design

Deep understanding 1: Why is exp field encoded using bias?

Deep understanding 2: Why have denormalized numbers?

Deep understanding 3: Why is bias chosen to be $2^{k-1} - 1$?

Floats: Properties

Floating point multiplication

The IEEE 754 number line

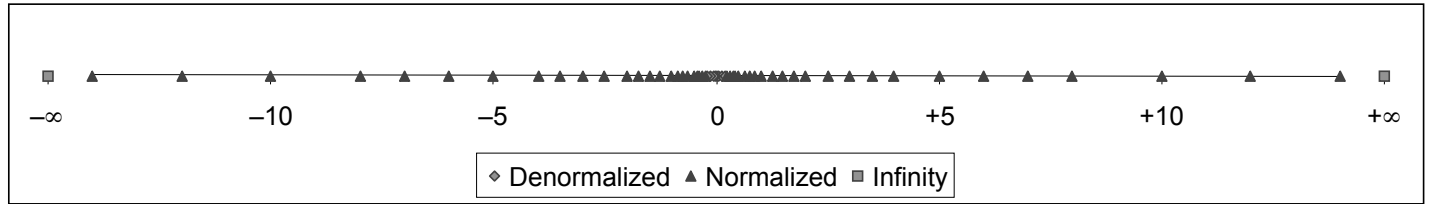


Figure: Full picture of number line for floating point values. Image credit CS:APP

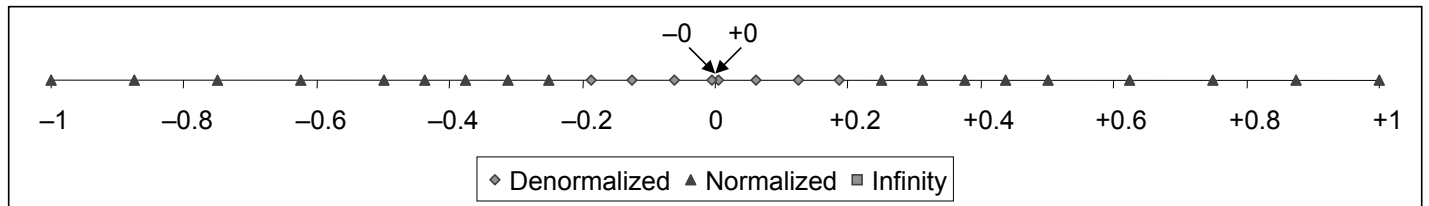


Figure: Zoomed in number line for floating point values. Image credit CS:APP

Data Representation: Floating Point Denormalized Numbers

Announcements

Canvas timed quiz 5 and programming assignment 3

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: Understanding its design

Deep understanding 1: Why is exp field encoded using bias?

Deep understanding 2: Why have denormalized numbers?

Deep understanding 3: Why is bias chosen to be $2^{k-1} - 1$?

Floats: Properties

Floating point multiplication

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 Denormalized Numbers

Announcements

Canvas timed quiz 5 and programming assignment 3

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: Understanding its design

Deep understanding 1: Why is exp field encoded using bias?

Deep understanding 2: Why have denormalized numbers?

Deep understanding 3: Why is bias chosen to be $2^{k-1} - 1$?

Floats: Properties

Floating point multiplication

Denormalized: frac field

$M = o.\text{frac}$

value represented leading with 0

Data Representation: Floating Point Denormalized Numbers

Announcements

Canvas timed quiz 5 and programming assignment 3

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: Understanding its design

Deep understanding 1: Why is exp field encoded using bias?

Deep understanding 2: Why have denormalized numbers?

Deep understanding 3: Why is bias chosen to be $2^{k-1} - 1$?

Floats: Properties

Floating point multiplication

Denormalized: examples

Largest
denorm
number

without special treatment
(not how it is)

$$\begin{aligned}
 &0.0000.111 \\
 &= (-1)^0 \cdot (1.111) \cdot 2^{0-1} \\
 &= \frac{15}{8} \cdot 2^{-1} \\
 &= \frac{15}{2^{10}}
 \end{aligned}$$

with special
treatment.

$$\begin{aligned}
 &0.0000.111 \\
 &= (-1)^0 \cdot (0.111) \cdot 2^{1-1} \\
 &= \frac{7}{8} \cdot \frac{1}{64} \\
 &= \frac{7}{2^9} = \frac{7}{512}
 \end{aligned}$$

Smallest
posim
denorm
number

$$\begin{aligned}
 &0.0000.001 \\
 &= (-1)^0 \cdot (1.001) \cdot 2^{-1} \\
 &= \frac{9}{8} \cdot 2^{-1} \\
 &= \frac{9}{2^{10}}
 \end{aligned}$$

$$\begin{aligned}
 &0.0000.001 \\
 &= (-1)^0 \cdot (0.001) \cdot 2^{1-1} \\
 &= \frac{1}{8} \cdot \frac{1}{64} \\
 &= \frac{1}{512}
 \end{aligned}$$

Zero

$$\begin{aligned}
 &0.0000.000 \\
 &= (-1)^0 \cdot (1.000_2) \cdot 2^{-1} \\
 &= \frac{1}{128}
 \end{aligned}$$

$$\begin{aligned}
 &0.0000.000 \\
 &= (-1)^0 \cdot (0.000) \cdot 2^{1-1} \\
 &= 0.0
 \end{aligned}$$

Data Representation: Floating Point Denormalized Numbers

Announcements

Canvas timed quiz 5 and programming assignment 3

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: Understanding its design

Deep understanding 1: Why is exp field encoded using bias?

Deep understanding 2: Why have denormalized numbers?

Deep understanding 3: Why is bias chosen to be $2^{k-1} - 1$?

Floats: Properties

Floating point multiplication

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 Denormalized Numbers

Announcements

Canvas timed quiz 5 and programming assignment 3

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: Understanding its design

Deep understanding 1: Why is exp field encoded using bias?

Deep understanding 2: Why have denormalized numbers?

Deep understanding 3: Why is bias chosen to be $2^{k-1} - 1$?

Floats: Properties

Floating point multiplication

Data Representation: Floating Point Denormalized Numbers

Announcements

Canvas timed quiz 5 and programming assignment 3

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: Understanding its design

Deep understanding 1: Why is exp field encoded using bias?

Deep understanding 2: Why have denormalized numbers?

Deep understanding 3: Why is bias chosen to be $2^{k-1} - 1$?

Floats: Properties

Floating point multiplication

Deep understanding 1: Why is exp field encoded using bias?

exp field needs to encode both positive and negative exponents.

Why not just use one of the signed integer formats? 2's complement, 1s' complement, signed magnitude?

Deep understanding 1: Why is exp field encoded using bias?

exp field needs to encode both positive and negative exponents.

Why not just use one of the signed integer formats? 2's complement, 1s' complement, signed magnitude?

Answer: allows easy comparison of magnitudes by simply comparing bits.

Deep understanding 1: Why is exp field encoded using bias?

exp field needs to encode both positive and negative exponents.

Why not just use one of the signed integer formats? 2's complement, 1s' complement, signed magnitude?

Answer: allows easy comparison of magnitudes by simply comparing bits.

Consider hypothetical 8-bit floating point format (from the textbook)

1-bit sign, $k = 4$ -bit exp, 3-bit frac.

What is the decimal value of
0b1_0110_111?

What is the decimal value of
0b1_0111_000?

Deep understanding 1: Why is exp field encoded using bias?

exp field needs to encode both positive and negative exponents.

Why not just use one of the signed integer formats? 2's complement, 1s' complement, signed magnitude?

Answer: allows easy comparison of magnitudes by simply comparing bits.

Consider hypothetical 8-bit floating point format (from the textbook)

1-bit sign, $k = 4$ -bit exp, 3-bit frac.

What is the decimal value of
0b1_0110_111?

$$-1.875 \times 2^{-1}$$

What is the decimal value of
0b1_0111_000?

$$-2.000 \times 2^{-1}$$

IEEE 754: Why Use a Biased Exponent?

- ▶ **Biased Exponent Formula:** $EXP = E + \text{bias}$, where $\text{bias} = 2^{k-1} - 1$.
 - ▶ 8-bit Tiny Float ($k = 4$): $\text{bias} = 7$.
 - ▶ 32-bit Single Precision ($k = 8$): $\text{bias} = 127$.
- ▶ **Why NOT Two's Complement for the Exponent?**
 - ▶ Biased encoding makes the EXP field an ****unsigned integer****.
 - ▶ **Hardware Reuse:** Allows floating-point numbers to be compared bit-by-bit using existing ****unsigned integer comparison hardware****.
 - ▶ Larger positive floating-point numbers will always have a 1 appearing earlier in their bit representation.

Data Representation: Floating Point Denormalized Numbers

Announcements

Canvas timed quiz 5 and programming assignment 3

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: Understanding its design

Deep understanding 1: Why is exp field encoded using bias?

Deep understanding 2: Why have denormalized numbers?

Deep understanding 3: Why is bias chosen to be $2^{k-1} - 1$?

Floats: Properties

Floating point multiplication

Deep understanding 2: Why have denormalized numbers?

Why not just continue normalized number scheme down to smallest numbers around zero?

Answer: makes sure that smallest increments available are maintained around zero.

Suppose denormalized numbers NOT used.

What is the decimal value of obo_0000_001?

$$1.125 \times 2^{-7}$$

What is the decimal value of obo_0000_111?

$$1.875 \times 2^{-7}$$

What is the decimal value of obo_0001_000?

$$2.000 \times 2^{-7}$$

Deep understanding 2: Why have denormalized numbers?

Why not just continue normalized number scheme down to smallest numbers around zero?

Answer: makes sure that smallest increments available are maintained around zero.

Suppose denormalized numbers ARE used.

What is the decimal value of obo_0000_001?

$$0.125 \times 2^{-6}$$

What is the decimal value of obo_0000_111?

$$0.875 \times 2^{-6}$$

What is the decimal value of obo_0001_000?

$$1.000 \times 2^{-6}$$

Data Representation: Floating Point Denormalized Numbers

Announcements

Canvas timed quiz 5 and programming assignment 3

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: Understanding its design

Deep understanding 1: Why is exp field encoded using bias?

Deep understanding 2: Why have denormalized numbers?

Deep understanding 3: Why is bias chosen to be $2^{k-1} - 1$?

Floats: Properties

Floating point multiplication

Data Representation: Floating Point Denormalized Numbers

Announcements

Canvas timed quiz 5 and programming assignment 3

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: Understanding its design

Deep understanding 1: Why is exp field encoded using bias?

Deep understanding 2: Why have denormalized numbers?

Deep understanding 3: Why is bias chosen to be $2^{k-1} - 1$?

Floats: Properties

Floating point multiplication

Data Representation: Floating Point Denormalized Numbers

Announcements

Canvas timed quiz 5 and programming assignment 3

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: Understanding its design

Deep understanding 1: Why is exp field encoded using bias?

Deep understanding 2: Why have denormalized numbers?

Deep understanding 3: Why is bias chosen to be $2^{k-1} - 1$?

Floats: Properties

Floating point multiplication

How to multiply scientific notation?

Recall: $\log(x \times y) = \log(x) + \log(y)$

FP Multiplication

- $(-1)^{s1} M1 2^{E1} \times (-1)^{s2} M2 2^{E2}$
- Exact Result: $(-1)^s M 2^E$
 - Sign s: $s1 \wedge s2$
 - Significand M: $M1 \times M2$
 - Exponent E: $E1 + E2$
- Fixing
 - If $M \geq 2$, shift M right, increment E
 - If E out of range, overflow
 - Round M to fit **frac** precision
- Implementation
 - Biggest chore is multiplying significands

Data Representation: Floating Point Denormalized Numbers

Announcements

Canvas timed quiz 5 and programming assignment 3

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: Understanding its design

Deep understanding 1: Why is exp field encoded using bias?

Deep understanding 2: Why have denormalized numbers?

Deep understanding 3: Why is bias chosen to be $2^{k-1} - 1$?

Floats: Properties

Floating point multiplication

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