Representing and Manipulating Information: Floating point mastery

Yipeng Huang
Rutgers University
March 2, 2023
Table of contents

Announcements
  Quizzes and programming assignments

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

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 $2^{k-1} - 1$?
Quizzes and programming assignments

Short quiz 4
- Due Friday. All about integers.

Programming assignment 3
- Has been out, due Friday before spring break.
Table of contents

Announcements

Quizzes and programming assignments

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

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 $2^{k-1} - 1$?
monteCarloPi.c Using floating point and random numbers to estimate PI
Table of contents

Announcements
Quizzes and programming assignments

`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

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 $2^{k-1} - 1$?
Floating point numbers

Avogadro’s number

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

Scientific notation

- sign
- mantissa or significand
- exponent
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

- $s \neq 0 \& \neq 255$

2. Denormalized

- $s = 0$

3a. Infinity

- $s = 1$

3b. NaN

- $s = 1$

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

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

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 $2^{k-1} - 1$?
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
- \(\text{bias} = 2^{k-1} - 1\)
- Bias is the \(k\)-bit unsigned integer: 011..111

For normalized numbers,  \(E = \exp - \text{bias}\)

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

<table>
<thead>
<tr>
<th>property</th>
<th>float</th>
<th>double</th>
</tr>
</thead>
<tbody>
<tr>
<td>(k)</td>
<td>8</td>
<td>11</td>
</tr>
<tr>
<td>bias</td>
<td>127</td>
<td>1023</td>
</tr>
<tr>
<td>smallest (E) (greatest precision)</td>
<td>-126</td>
<td>-1022</td>
</tr>
<tr>
<td>largest (E) (greatest range)</td>
<td>127</td>
<td>1023</td>
</tr>
</tbody>
</table>

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 1100.011₂
- in other words it is 1.100011₂ × 2³
- $\exp = E + \text{bias} = 3 + 127 = 130 = 1000_0010₂$
- $M = 1.100011₂ = 1.\text{frac}$
- $\text{frac} = 100011$
Table of contents

Announcements
  Quizzes and programming assignments

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

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 $2^{k-1} - 1$?
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

<table>
<thead>
<tr>
<th>property</th>
<th>float</th>
<th>double</th>
</tr>
</thead>
<tbody>
<tr>
<td>k</td>
<td>8</td>
<td>11</td>
</tr>
<tr>
<td>bias</td>
<td>127</td>
<td>1023</td>
</tr>
<tr>
<td>E</td>
<td>-126</td>
<td>-1022</td>
</tr>
</tbody>
</table>

Table: Summary of denormalized exp field
Denormalized: frac field

\[ M = 0.frac \]
value represented leading with 0
Denormalized: examples
## Floats: Summary

<table>
<thead>
<tr>
<th></th>
<th>normalized</th>
<th>denormalized</th>
</tr>
</thead>
<tbody>
<tr>
<td>value of number</td>
<td>$(-1)^s \times M \times 2^E$</td>
<td>$(-1)^s \times M \times 2^E$</td>
</tr>
<tr>
<td>E</td>
<td>$E = \text{exp-bias}$</td>
<td>$E = \text{-bias} + 1$</td>
</tr>
<tr>
<td>bias</td>
<td>$2^{k-1} - 1$</td>
<td>$2^{k-1} - 1$</td>
</tr>
<tr>
<td>exp</td>
<td>$0 &lt; \text{exp} &lt; (2^k - 1)$</td>
<td>$\text{exp} = 0$</td>
</tr>
<tr>
<td>M</td>
<td>$M = 1.\text{frac}$</td>
<td>$M = 0.\text{frac}$</td>
</tr>
<tr>
<td></td>
<td>M has implied leading 1</td>
<td>M has leading 0</td>
</tr>
<tr>
<td></td>
<td>greater range</td>
<td>greater precision</td>
</tr>
<tr>
<td></td>
<td>large magnitude numbers</td>
<td>small magnitude numbers</td>
</tr>
<tr>
<td></td>
<td>denser near origin</td>
<td>evenly spaced</td>
</tr>
</tbody>
</table>

**Table:** Summary of normalized and denormalized numbers
Table of contents

Announcements
  Quizzes and programming assignments

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

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 $2^{k-1} - 1$?
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 $0b1_0111_000$?
$-2.000 \times 2^{-1}$
Table of contents

Announcements
  Quizzes and programming assignments

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

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 $2^{k-1} - 1$?
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 0b0_0000_001?
1.125 \times 2^{-7}

What is the decimal value of 0b0_0000_111?
1.875 \times 2^{-7}

What is the decimal value of 0b0_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 0b0_0000_001?
0.125 \times 2^{-6}

What is the decimal value of 0b0_0000_111?
0.875 \times 2^{-6}

What is the decimal value of 0b0_0001_000?
1.000 \times 2^{-6}
Table of contents

Announcements

Quizzes and programming assignments

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

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: Special cases

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

**Table:** Summary of special cases