CSARCH Lecture Series: Binary Floating-Point format for Single Precision (special cases)

Sensei RL Uy

College of Computer Studies

De La Salle University

Manila, Philippines

Copyright Notice

Overview

Reflect on the following questions:

- How are zeros and infinity represented in the memory?
- How large should a single-precision floating-point number be to considered an infinity?

Overview

- This sub-module introduces the IEEE-754 single-precision floating-point format involving special cases
- The objective is as follows:
  - ✓Describe the process of representing special cases such as zero, infinity, denormalized and NaN using IEEE-754 standard

Special cases

Sign Bit	E’	Significand	Value
0	0000 0000	000 0000 0000 0000 0000 0000	+0 (Positive Zero)
1	0000 0000	000 0000 0000 0000 0000 0000	-0 (Negative Zero)
0/1	0000 0000	$\neq 0$	Denomalized
0	1111 1111	000 0000 0000 0000 0000 0000	+ Infinity
1	1111 1111	000 0000 0000 0000 0000 0000	- Infinity
x	1111 1111	01x xxxx xxxx xxxx xxxx xxxx	sNaN
x	1111 1111	1xx xxxx xxxx xxxx xxxx xxxx	qNaN

Special cases use smallest (00000000) and the largest (11111111) exponent representation (e’)

Special case (Denormalized)

- Denormalized are numbers so small (approaching 0) that it cannot be represented normally
- What is the smallest positive normal number?

Sign	Exponent representation	Fraction part of significand
0	0000 0001	000 0000 0000 0000 0000 0000

The smallest possible e' is 1. Thus e=1-127 = -126.

The smallest positive normal number is $+1.0\text{x}2^{-126}$ (or $1.18\text{x}10^{-38}$)

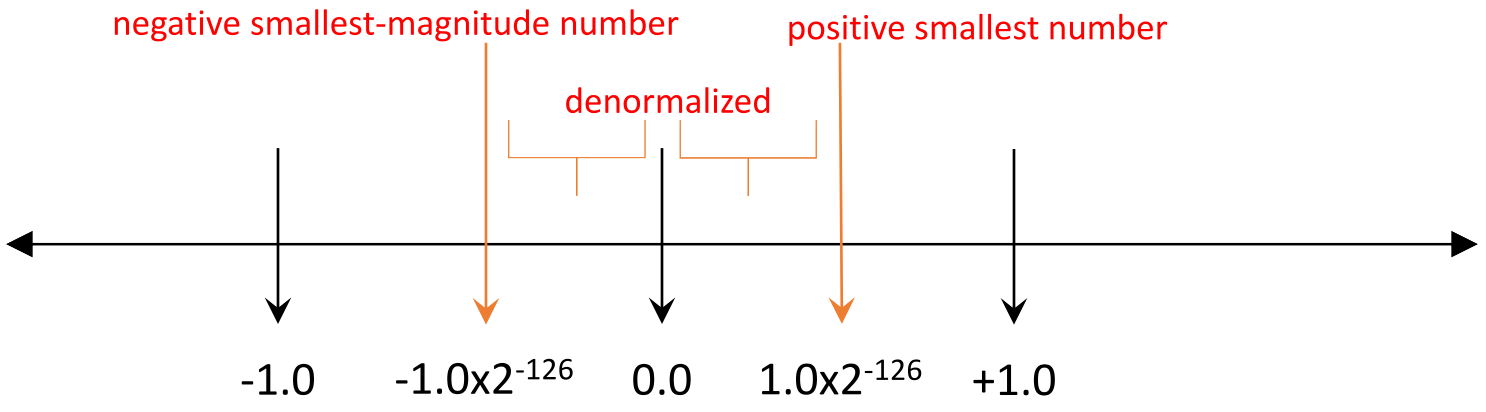
- What is the smallest-magnitude negative normal number?

The smallest-magnitude negative normal number is $-1.0\text{x}2^{-126}$ (or $-1.18\text{x}10^{-38}$)

Special case (Denormalized)

The smallest positive normal number is $+1.0\text{x}2^{-126}$ (or $1.18\text{x}10^{-38}$)

The smallest-magnitude negative normal number is $-1.0\text{x}2^{-126}$ (or $-1.18\text{x}10^{-38}$)



To represent denormal number

- peg the exponent to -126 and denormalized the significand
- e' = 0
- significand is the denormalized significand

Special case (Denormalized)

Example: $-1.1110_2 \text{x } 2^{-130}$

normalized format: $-0.0001111_2 \text{x } 2^{-126}$

Significand in binary?	Yes
Base-2?	Yes
Normalized?	Yes. But special case, need to denormalized
Sign bit	1
Exponent representation	special case: 0000 0000

Answer:

Sign	Exponent representation	Fraction part of significand
1	0000 0000	000 1111 0000 0000 0000 0000

Hex: 0x800F0000

Special case (infinity)

- Infinity are very big numbers (approaching infinity) that it cannot be represented normally
- What is the largest positive normal number?

Sign	Exponent representation	Fraction part of significand
0	1111 1110	111 1111 1111 1111 1111 1111

The largest possible e’ is 254. Thus e=254-127 = 127

The largest positive normal number is +1.11111111111111111111111x2^{127} (or 3.4x10^{38})

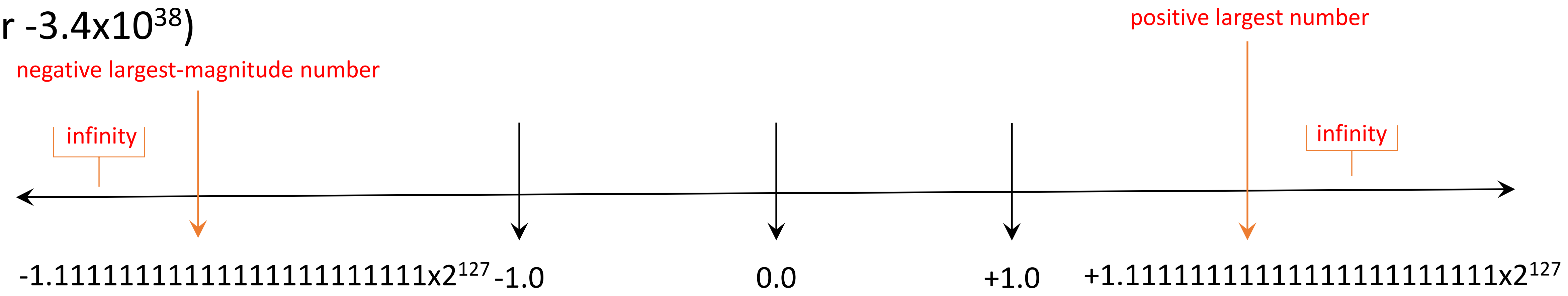
- What is the largest-magnitude negative normal number?

The largest-magnitude negative normal number is -1. 11111111111111111111111x2^{127} (or -3.4x10^{38})

Special case (infinity)

The largest positive normal number is $+1.11111111111111111111111 \times 2^{127}$ (or 3.4×10^{38})

The largest-magnitude negative normal number is $-1.11111111111111111111111 \times 2^{127}$ (or -3.4×10^{38})



To represent infinity number

- e' = 11111111
- significand is 000 0000 0000 0000 0000 0000

Special case (Infinity)

Example: $+1.111_2 \times 2^{999}$

normalized format: $+1.111_2 \times 2^{999}$ (Same)

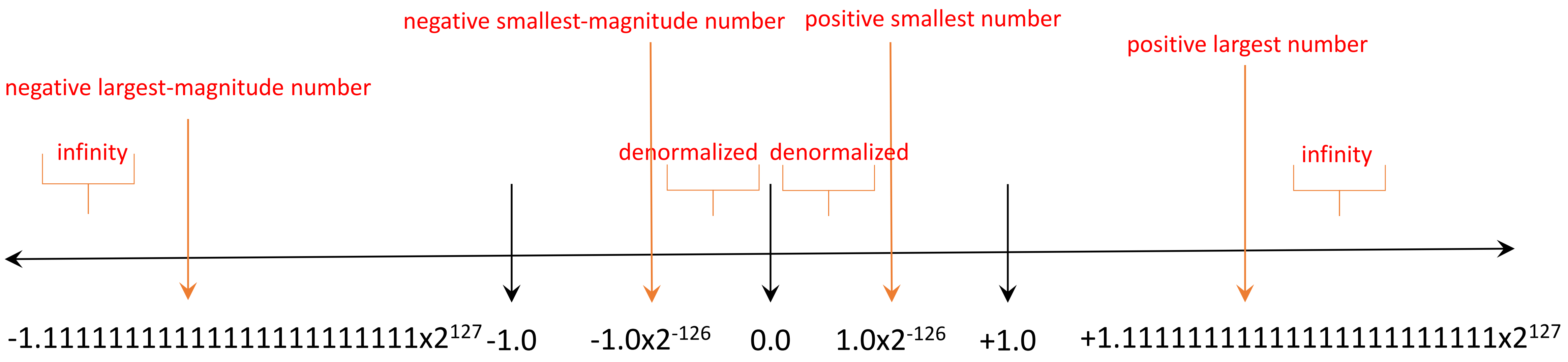
Significand in binary?	Yes
Base-2?	Yes
Normalized?	Yes
Sign bit	0
Exponent representation	special case: 1111 1111

Answer:

Sign	Exponent representation	Fraction part of significand
0	1111 1111	000 0000 0000 0000 0000 0000

Hex: 0x7F80000

Special case (number line)



Special case (NaN)

- Indeterminate numbers are example of Not a Number (NaN)
- Sign bit is don't care
- there are 2 types of NaN representation:
  - Signaling NaN (sNaN)
    - Two most significant bit of the significand is 01
    - floating-point result using sNaN signals the invalid operation exception
  - Quiet NaN (qNaN)
    - most significant bit of the significand is 1
    - floating-point result using qNaN allows the result to be propagated

Sign Bit	E'	Significand	Value
x	1111 1111	01x xxxx xxxx xxxx xxxx xxxx	sNaN
x	1111 1111	1xx xxxx xxxx xxxx xxxx xxxx	qNaN

To recall …

- What have we learned:
  - ✓Describe the process of representing special cases such as zero, infinity, denormalized and NaN using IEEE-754 standard