Discussion 8
Floating Point Numbers

Zheng Junren

- IEEE 754
- The *sign* determines the sign of the number (0 for positive, 1 for negative)
- The *exponent* is in **biased notation** with a bias of 127
- The *significand* is akin to unsigned, but used to store a fraction instead of an integer.

32bits

Sign	Exponent	Significand
1 bit	8 bits	23 bits

64bits

uses 11 bits for the exponent (and thus a bias of 1023) and 52 bits for the significand.

For normalized floats:

$$\text{Value} = (-1)^{\text{Sign}} \times 2^{(\text{Exponent} - \text{Bias})} \times 1.\text{significand}_2$$

For denormalized floats:

$$\text{Value} = (-1)^{\text{Sign}} \times 2^{(\text{Exponent} - \text{Bias} + 1)} \times 0.\text{significand}_2$$

Exponent	Significand	Meaning
0	Anything	Denorm
1-254	Anything	Normal
255	0	Infinity
255	Nonzero	NaN

Exercise

- (a) IEEE Standard Single-Precision Floating-Point Numbers
- A. 1000 0000 0100 1101 1010 0100 1111 0101_{two}
- B. 1000 0000 0111 0111 0111 0010 1011 0101_{two}
- Question: A ? B

Answer

- (a) IEEE Standard Single-Precision Floating-Point Numbers
- A. 1000 0000 0100 1101 1010 0100 1111 0101_{two}
- B. 1000 0000 0111 0111 0111 0010 1011 0101_{two}
- Answer: A > B

Exercise

- (b) IEEE Standard Single-Precision Floating-Point Numbers
- A. 1110 1111 0101 1101 1110 0100 1111 0101_{two}
- B. 1110 1100 0111 0111 0111 0000 1011 0101_{two}
- Question: A ? B

Answer

- (b) IEEE Standard Single-Precision Floating-Point Numbers
- A. 1110 1111 0101 1101 1110 0100 1111 0101_{two}
- B. 1110 1100 0111 0111 0111 0000 1011 0101_{two}
- Answer: A < B

Exercise

- (c) IEEE Standard Single-Precision Floating-Point Numbers
- A. 0x9A2F 6AC0
- B. 0x9CBC BCBB
- Question: A ? B

Exercise

- (c) IEEE Standard Single-Precision Floating-Point Numbers
- A. 0x9A2F 6AC0
- B. 0x9CBC BCBB
- Answer: A > B

Exercises

1. How many zeroes can be represented using a float?

2. What is the largest finite positive value that can be stored using a single precision float?

3. What is the smallest positive value that can be stored using a single precision float?

4. What is the smallest positive normalized value that can be stored using a single precision float?

5. Convert the following numbers from binary to decimal or from decimal to binary:

0x00000000 8.25 0x00000F00 39.5625 0xFF94BEEF $-\infty$

1. How many zeroes can be represented using a float? 2

2. What is the largest finite positive value that can be stored using a single precision float?

   $0x7F7FFFFF = (2 - 2^{-23}) \times 2^{127}$

3. What is the smallest positive value that can be stored using a single precision float?

   $0x00000001 = 2^{-23} \times 2^{-126}$

4. What is the smallest positive normalized value that can be stored using a single precision float?

   $0x00800000 = 2^{-126}$

5. Convert the following numbers from binary to decimal or from decimal to binary:

   0x00000000 8.25 0x00000F00 39.5625 0xFF94BEEF -∞

$0x00000000 = 0$

$8.25 = 0x41040000$

$0x000000F0 = (2^{-12} + 2^{-13} + 2^{-14} + 2^{-15}) \times 2^{-126}$

$39.5625 = 0x421E4000$

$0xFF94BEEF = NaN$

$-\infty = 0xFF800000$

Exercise

- For the following questions, we will be referring to the IEEE 32-bit floating point representation ***except*** with a 7 bit exponent (bias of 2^7/2 - 1 = 63) and a denorm implicit exponent of -62.

a) Convert -32.588 to floating point format. Write your answer in hexadecimal.

b) Convert the floating point number 0xC1102000, which follows the 7 bit exponent representation as described above, to decimal. Please specify infinities as +inf or -inf, and not a number as NaN.

c) What's the smallest non-infinite positive integer (an integer has nothing to the right of the decimal) it **CANNOT** represent? Leave your answer in decimal (ex: 12).

d) What's the smallest positive value it can represent that is **not a denorm**? Leave your answer as a power of 2 (ex: 2^x).

e) What's the smallest positive value it can represent? Leave your answer as a power of 2 (ex: 2^x).

a. -32.588(dec) = -10 0000.1001 0110 1000 0111 0010 1011(bin)
= -1. 0000 0100 1011 0100 0011 1001 * 2^(68-63)
bin : 1 1000100 0000 0100 1011 0100 0011 1001
hex : 0xC404 B439

b.0xC1102000(hex)=1100 0001 0001 0000 0010 0000 0000 0000(bin)
sign : minus, exponent:1000001(bin)=65(dec)=2+63,
significand : 0001 0000 0010 0000 0000 0000(bin)=2^(-4)+2^(-11)(dec)=0.0630(dec)
result : (-1)*(1+ 0.0630)*2^2= -4.252

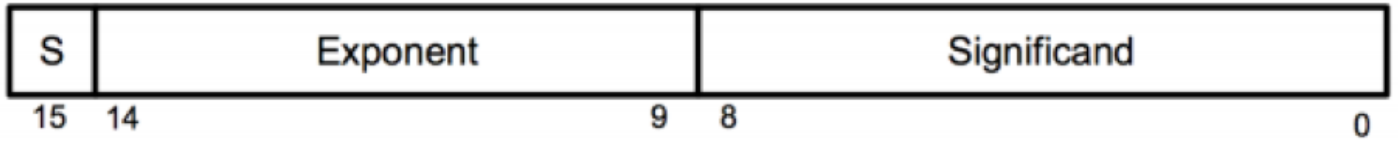
c.max integer it can represent :
0 1111110 1111 1111 1111 1111 1111 1111 = (1+1-2^(-24))*2^63=2^64-2^39
min integer it cannot represent :
2^64-2^39+1
decimal : 18446743523953737729

d.2^(-62)

e.2^(-62-24)=2^-86

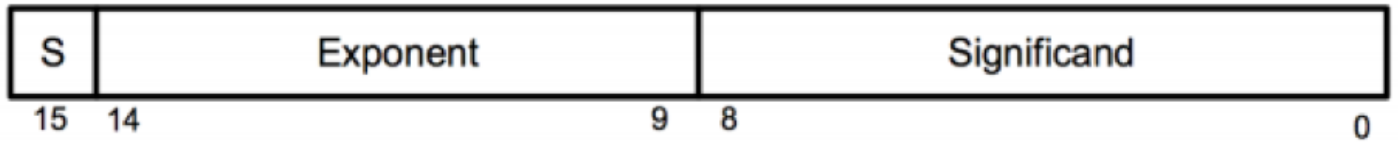
Exercise

- Consider the following 16-bit representation for floating point numbers:



Question: how many positive real numbers can be represented?

Answer



Exponent	Significand	Object
0	0	0
0	Nonzero	Denorm
1-62	anything	+/- fl. pt. #
63	0	+/- ∞
63	nonzero	NaN

let Sign = 0, there are 15 bits left: 2^{15}
except exponent = 63: 2^{9}
except 0: 1
answer: $2^{15}-2^{9}-1$