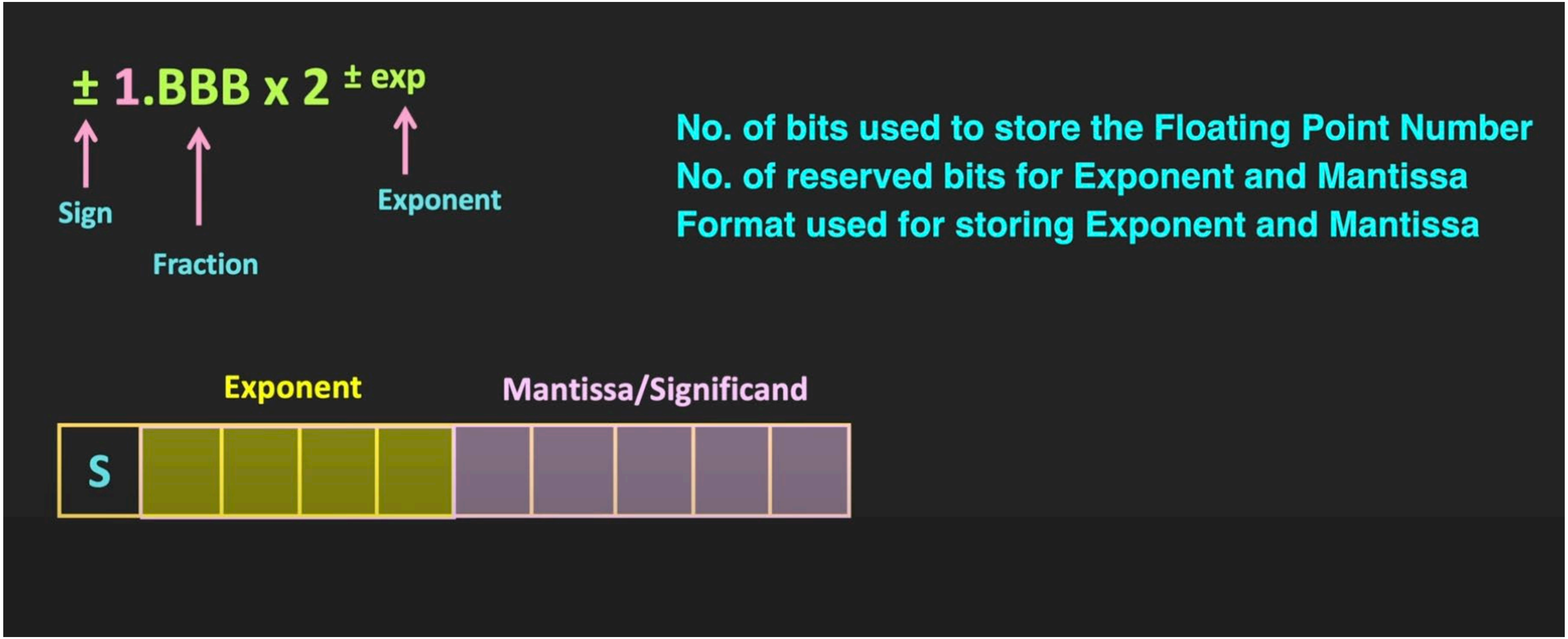
IEEE 754 Standard
Single Precision Format

IEEE 754 Standard



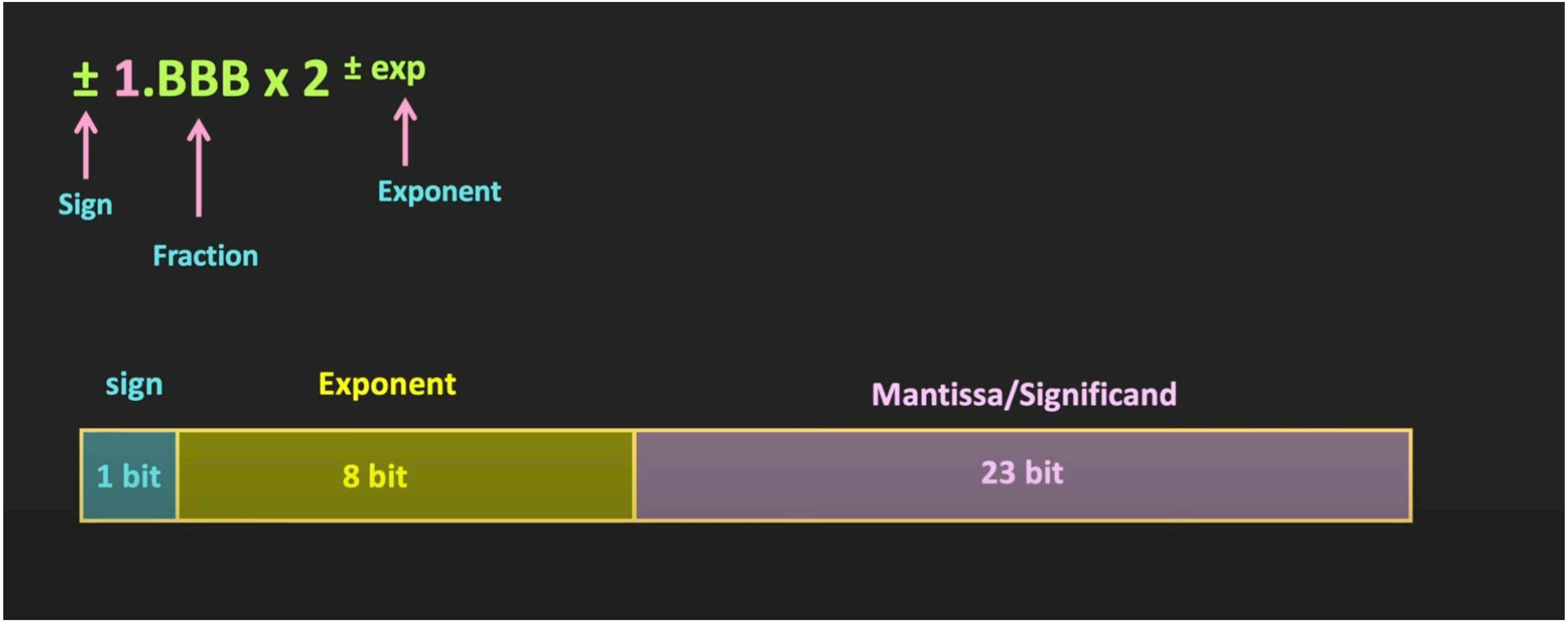
How Floating Point Numbers are Stored in Memory?

- IEEE 754 Standard
  - Half Precision (16 bits)
  - Single Precision (32 bits)
  - Double Precision (64 bits)
  - Quadruple Precision (128 bits)
  - Octaple Precision (256 bits)

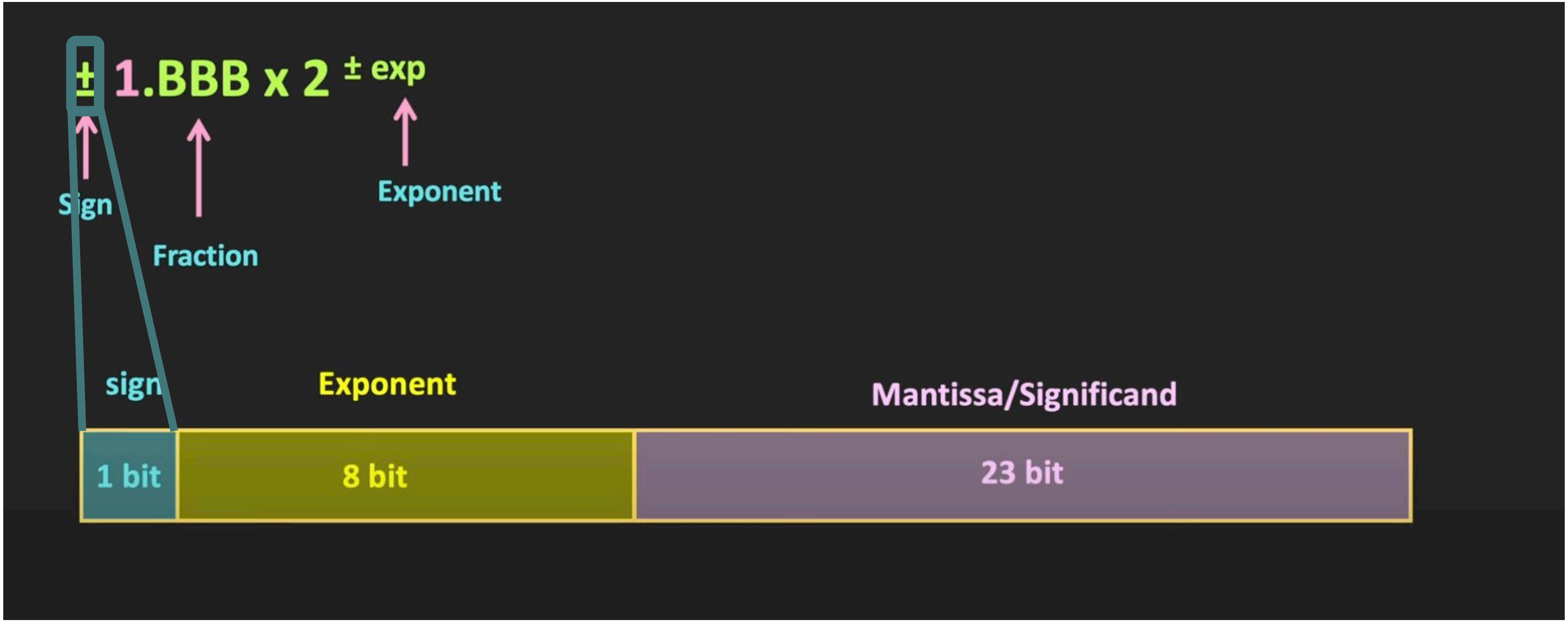
How Floating Point Numbers are Stored in Memory?

- IEEE 754 Standard
  - Half Precision (16 bits)
  - Single Precision (32 bits)
  - Double Precision (64 bits)
  - Quadruple Precision (128 bits)
  - Octaple Precision (256 bits)

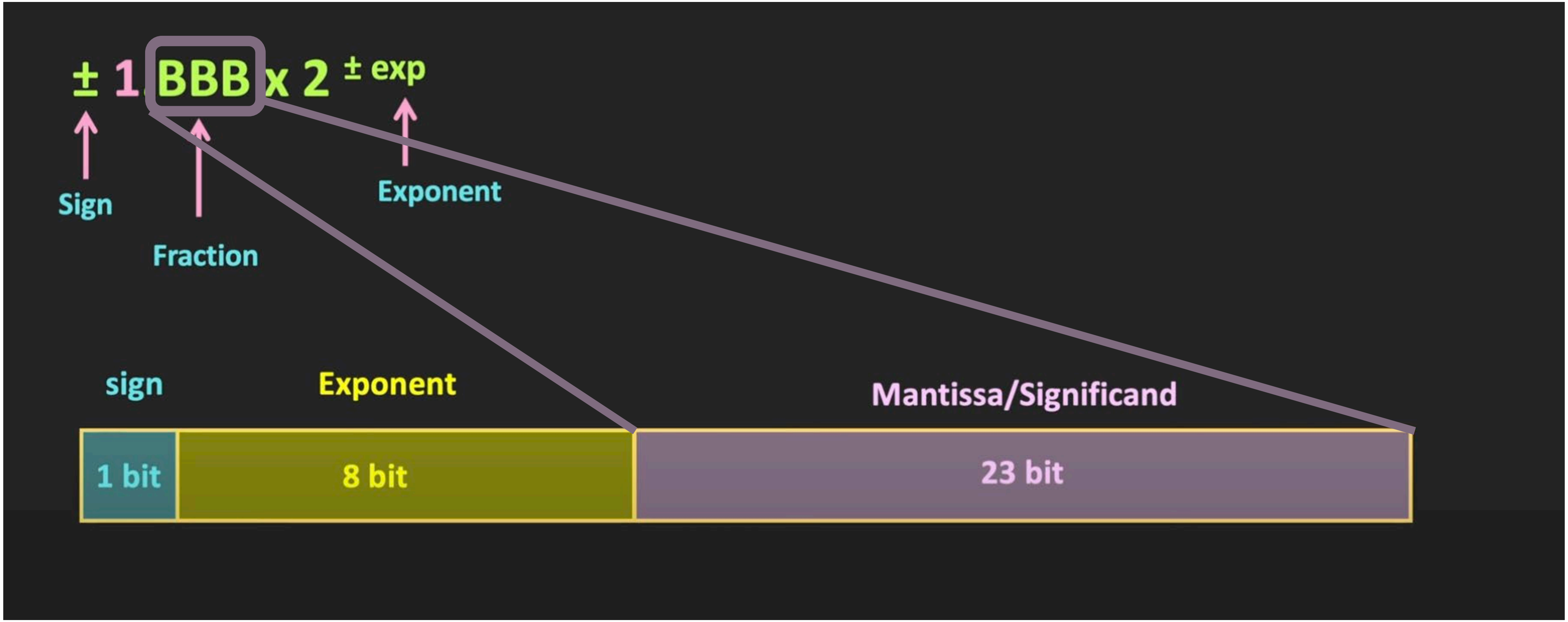
IEEE 754 – Single Precision Format



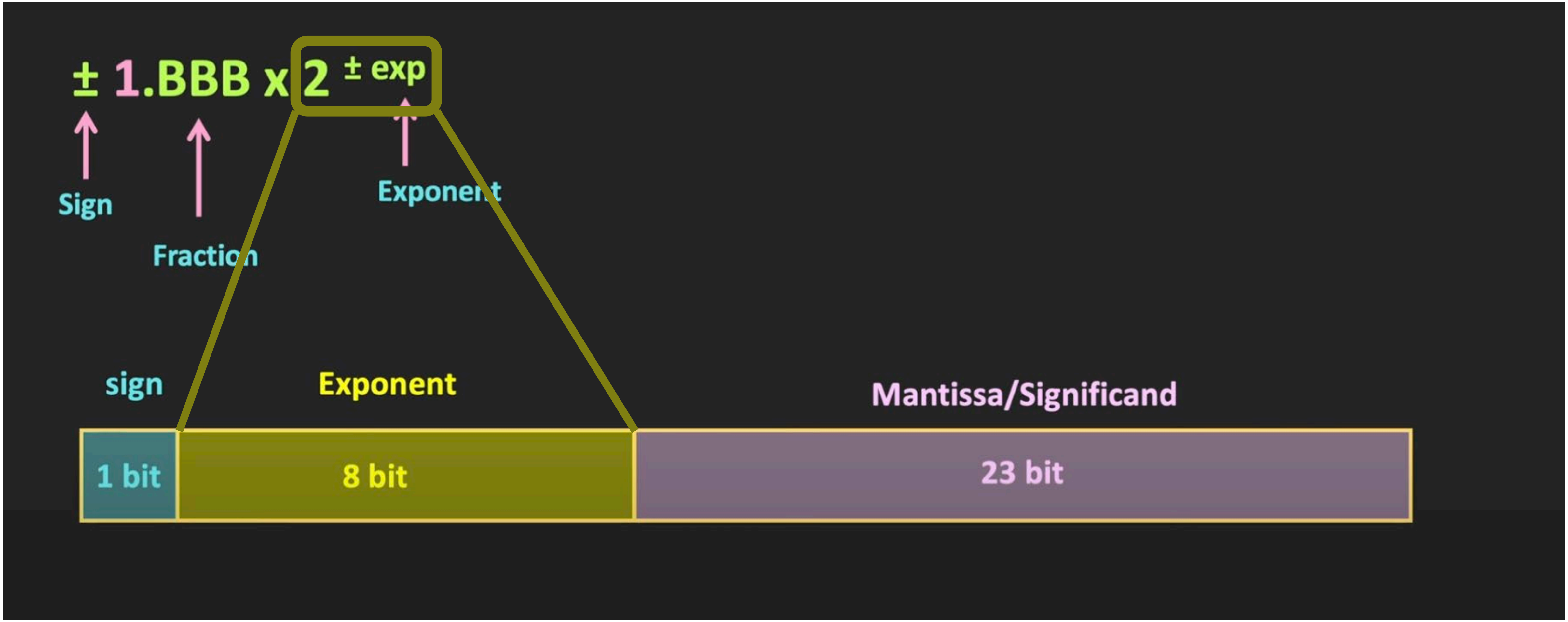
IEEE 754 – Single Precision Format



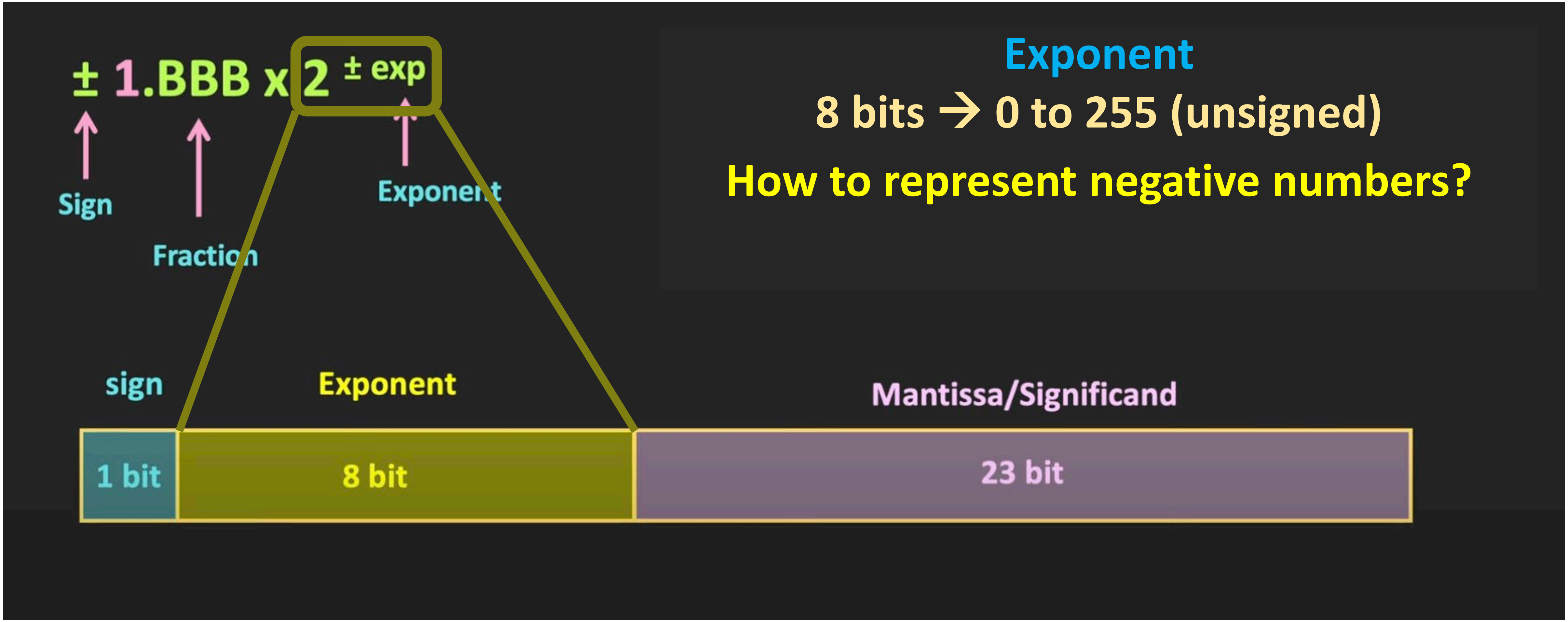
IEEE 754 – Single Precision Format



IEEE 754 – Single Precision Format



IEEE 754 – Single Precision Format



IEEE 754 – Single Precision Format

$\pm 1.BBB \times 2^{\pm exp}$

Sign

Fraction

Exponent

Exponent

8-bits → 0 to 255 (unsigned)

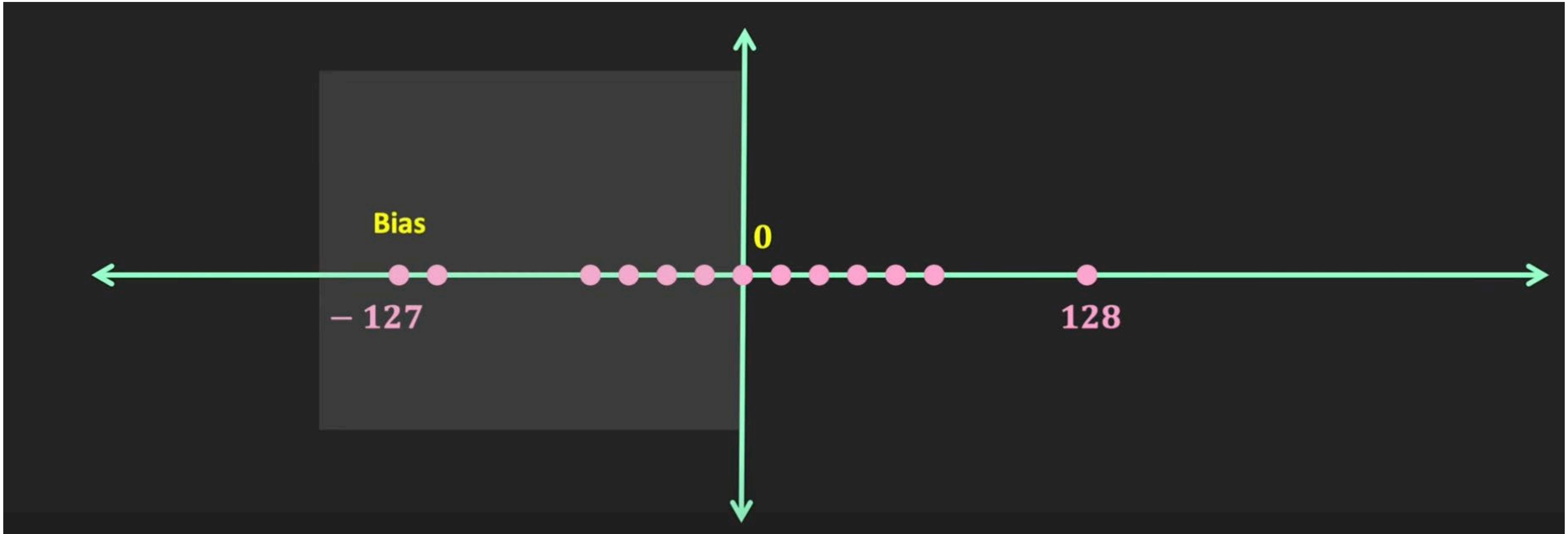
How to represent Negative exponent values ?

2's Complement

Signed Magnitude

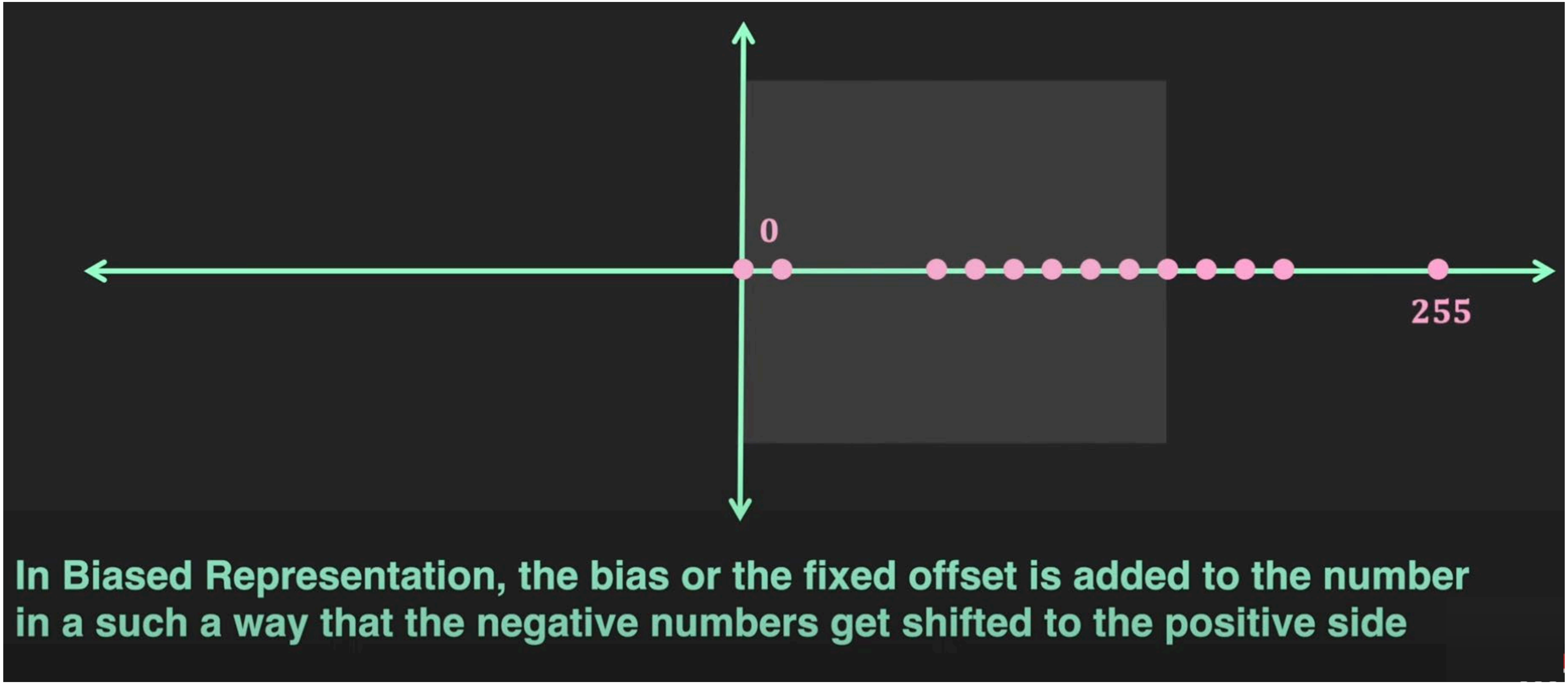
Biased Representation

Biased Representation



In Biased Representation, the bias or the fixed offset is added to the number in a such a way that the negative numbers get shifted to the positive side

Biased Representation



Biased Representation

$$Bias = 2^{n-1} - 1$$

n - no of bits

8 bits → Bias = 127

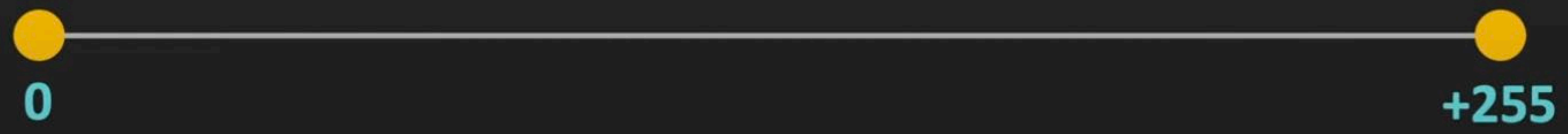
Biased Representation

$$Bias = 2^{n-1} - 1$$

n - no of bits

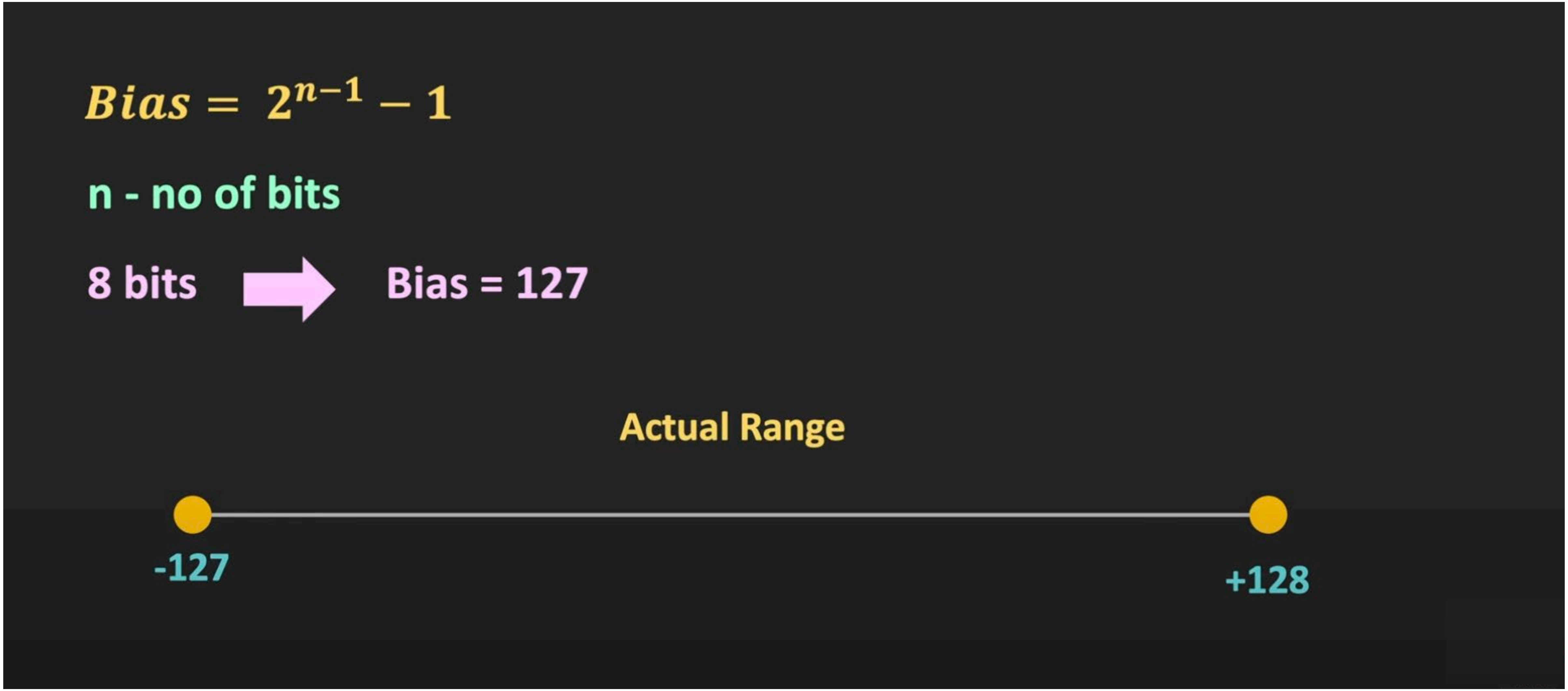
8 bits ⇒ Bias = 127

In biased Representation



Biased Representation

Biased Representation



Biased Representation

Actual Number	Biased Number	Biased Representation
-127	0	0000 0000
-126	1	0000 0001
......	......	....
-1	126	0111 1110
0	127	0111 1111
1	128	1000 0000
......		
127	254	1111 1110
128	255	1111 1111

Biased Representation

Actual Number	Biased Number	Biased Representation
-127	0	0000 0000
-126	1	0000 0001
......	......	....
-1	126	0111 1110
0	127	0111 1111
1	128	1000 0000
......		
127	254	1111 1110
128	255	1111 1111

Special Values

Biased Representation

Exponent Range

-126 to +127

Actual Number	Biased Number	Biased Representation
-127	0	0000 0000
-126	1	0000 0001
......	......	....
-1	126	0111 1110
0	127	0111 1111
1	128	1000 0000
......		
127	254	1111 1110
128	255	1111 1111

Special Values

Biased Representation

Exponent Range

-126 to +127

Actual Number	Biased Number	Biased Representation
-127	0	0000 0000
-126	1	0000 0001
......	......	....
-1	126	0111 1110
0	127	0111 1111
1	128	1000 0000
......		
127	254	1111 1110
128	255	1111 1111

Continuity

Biased Representation

Discontinuity

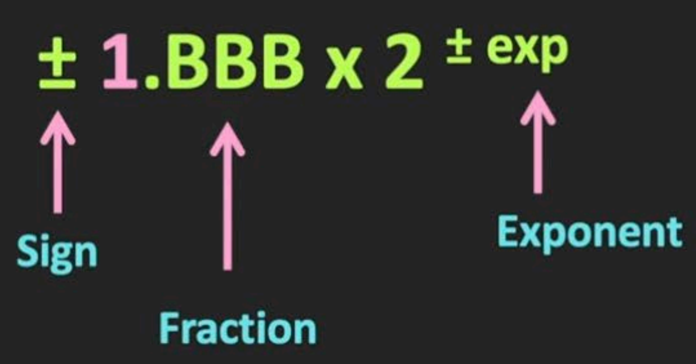
2's Complement

Number	2's Complement
-128	1000 0000
-127	1000 0001
-126	1000 0010
……	
-1	1111 1111
0	0000 0000
1	0000 0001
……	
126	0111 1110
127	0111 1111

Sign Magnitude

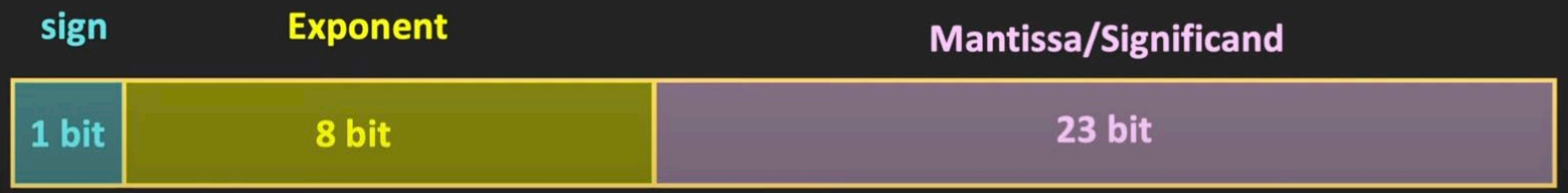
Number	Sign Magnitude
-127	1111 1111
-126	1111 1110
-125	1111 1101
……	
-1	1000 0001
-0	1000 0000
+0	0000 0000
1	0000 0001
……	
126	0111 1110
127	0111 1111

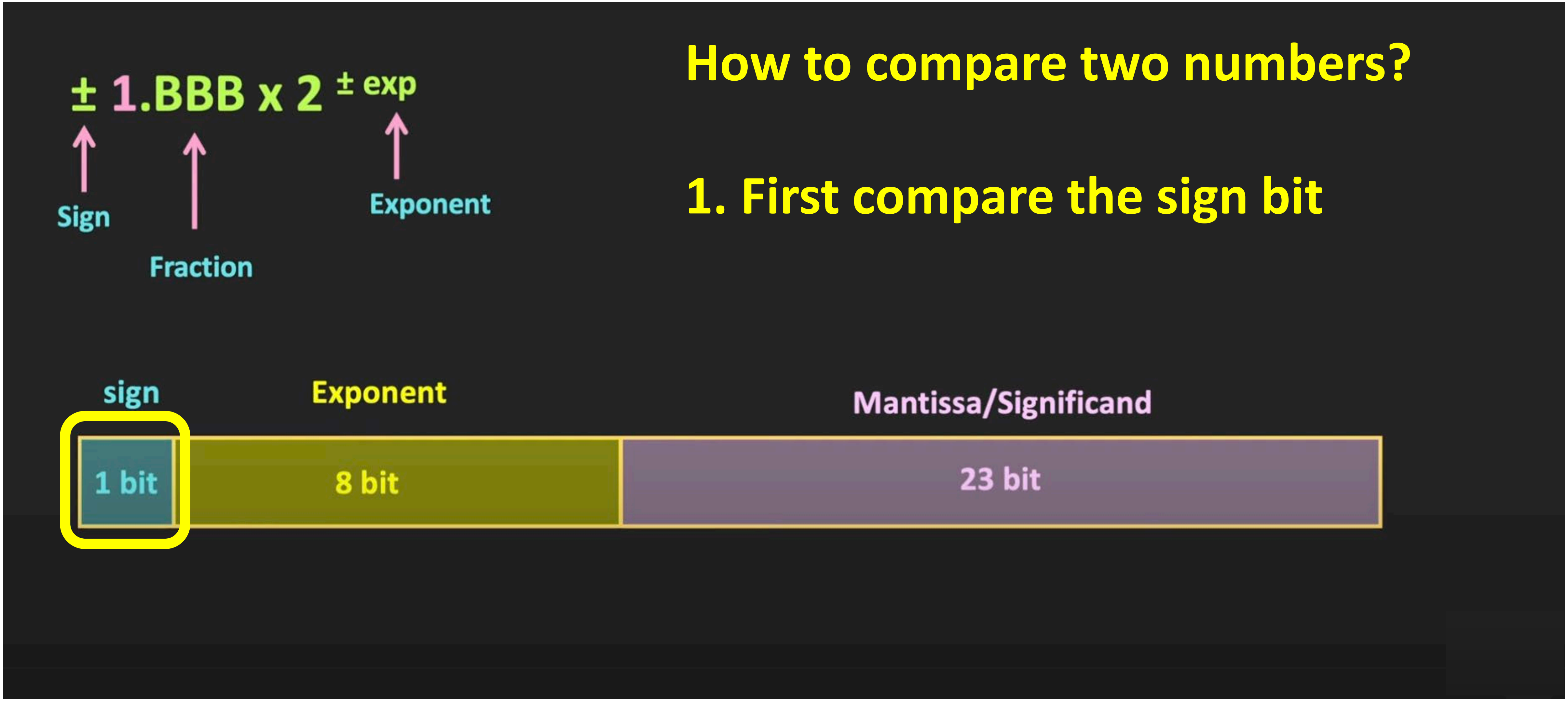
Biased Representation



What is the problem if the number representation is discontinue?

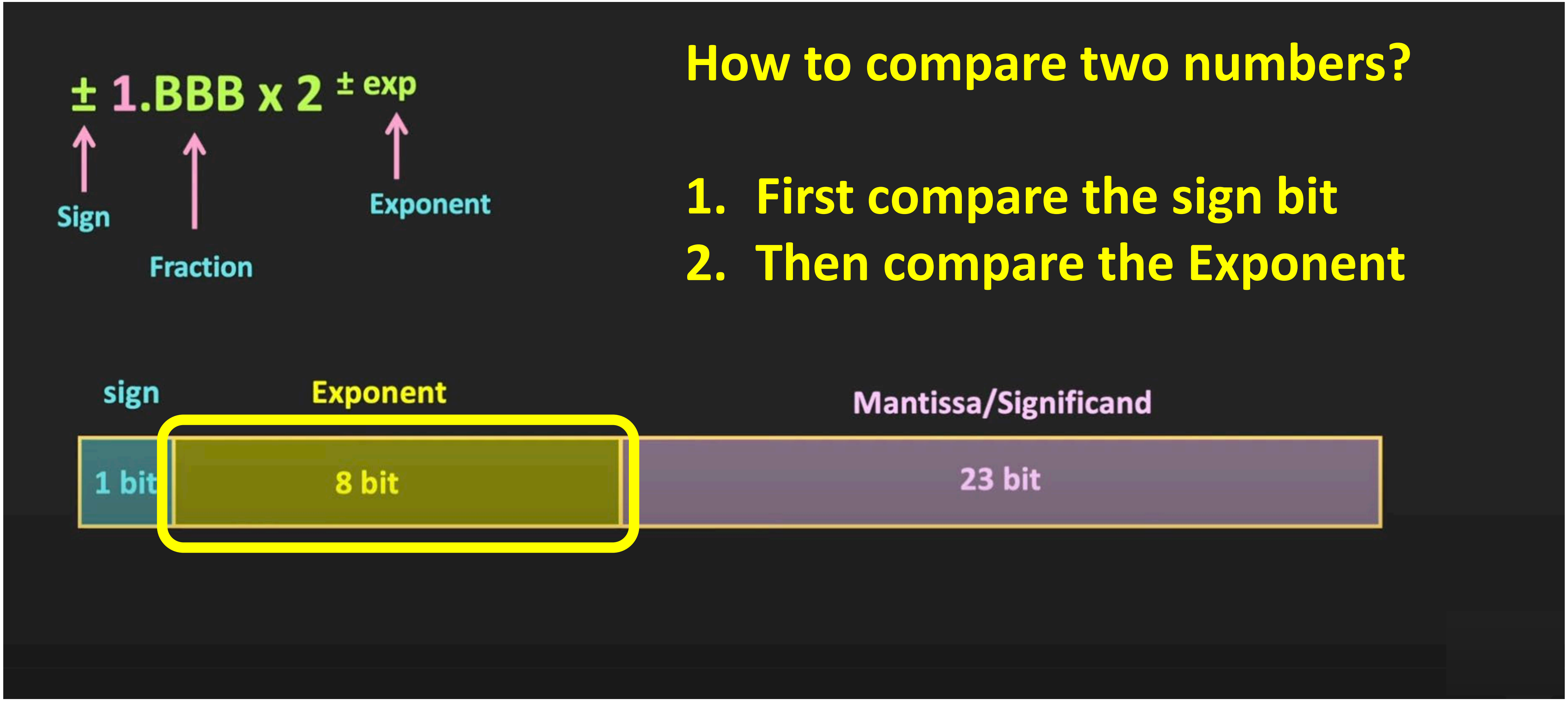
How to compare two numbers?





How to compare two numbers?
1. First compare the sign bit
± 1.BBB x 2 ^{± exp}
Sign
Fraction
Exponent
sign
Exponent
Mantissa/Significand
1 bit
8 bit
23 bit

Biased Representation

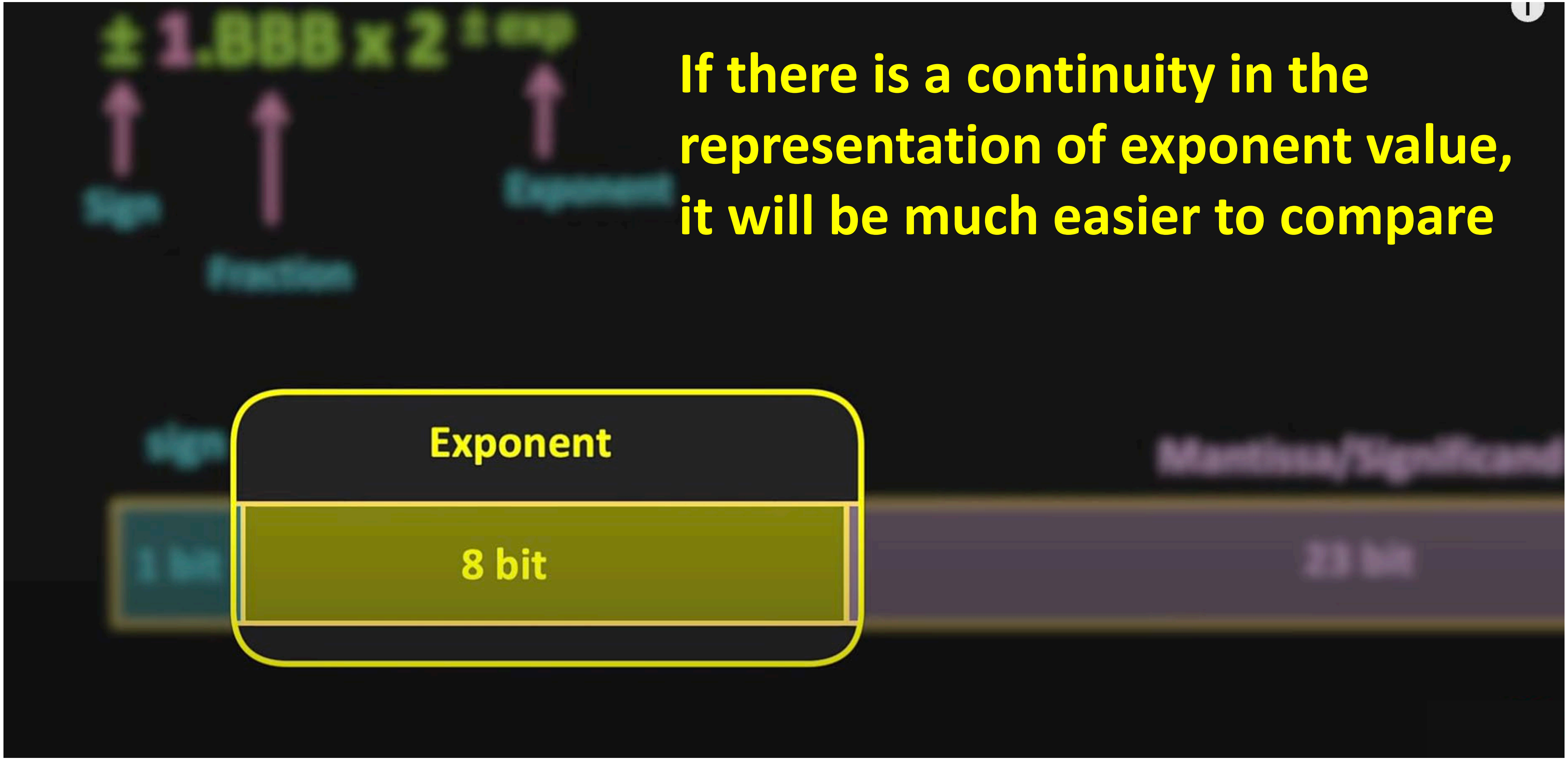


Biased Representation

How to compare two numbers?

1. First compare the sign bit
2. Then compare the Exponent
3. Finally, compare the Mantissa

Biased Representation



Biased Representation

Let’s compare these two floating point numbers

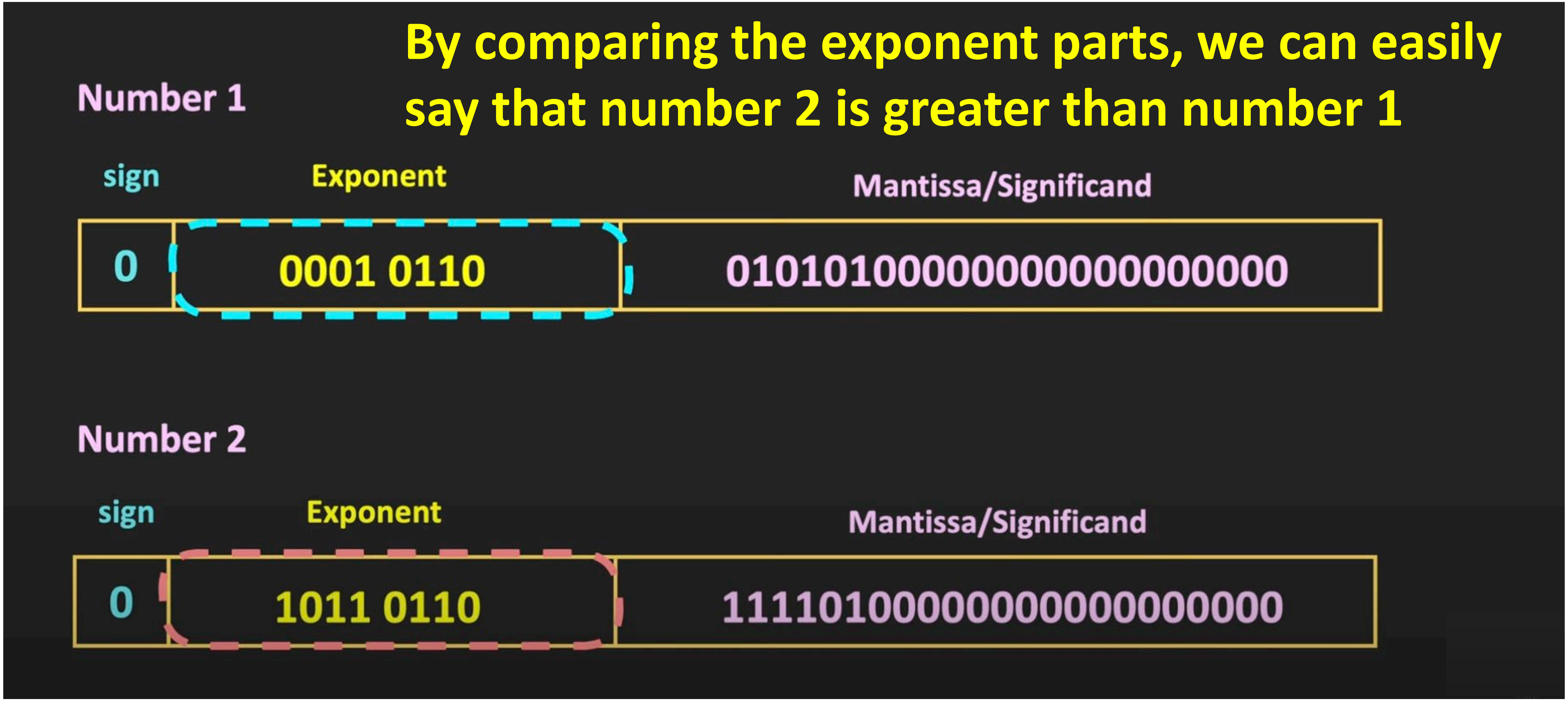
Number 1

sign	Exponent	Mantissa/Significand
0	0001 0110	01010100000000000000000

Number 2

sign	Exponent	Mantissa/Significand
0	1011 0110	11110100000000000000000

Biased Representation



Biased Representation

So, biased representation for is very useful

IEEE 754 uses this for floating point representation

Let's see how to get the actual number from a IEEE 754 Single Precision format

IEEE 754 – Single Precision Format

(Example 1)

Let's say this is a 32 bit number stored in the Single Precision format

sign	Exponent	Mantissa/Significand
0	1000 0101	00111000000000000000000

IEEE 754 – Single Precision Format

(Example 1)

Let's say this is a 32 bit number stored in the Single Precision format

sign	Exponent	Mantissa/Significand
0	1000 0101	00111100000000000000000

The MSB is 0. So this is a positive number

IEEE 754 – Single Precision Format

(Example 1)

Let’s say this is a 32 bit number stored in the Single Precision format

sign	Exponent	Mantissa/Significand
0	1000 0101	00111100000000000000000

Actual value of the exponent

IEEE 754 – Single Precision Format

(Example 1)

Let's say this is a 32 bit number stored in the Single Precision format

sign	Exponent	Mantissa/Significand
0	1000 0101	00111000000000000000000

Actual value of the exponent

1000 0101 → 133

IEEE 754 – Single Precision Format

(Example 1)

Let's say this is a 32 bit number stored in the Single Precision format

sign	Exponent	Mantissa/Significand
0	1000 0101	00111100000000000000000

Actual value of the exponent

1000 0101 $\Rightarrow$ 133

Actual Exponent = 133 – 127 = 6

Since the number is stored using biased format, we need to subtract the bias to get the actual value

IEEE 754 – Single Precision Format

(Example 1)

Let's say this is a 32 bit number stored in the Single Precision format

sign	Exponent	Mantissa/Significand
0	1000 0101	00111100000000000000000

Exponent: 2^6

IEEE 754 – Single Precision Format

(Example 1)

Let's say this is a 32 bit number stored in the Single Precision format

sign	Exponent	Mantissa/Significand
0	1000 0101	00111100000000000000000

Exponent: 2^{6}

In normalized binary form, there is a 1 before the Mantissa:

1. 00111100000000000000000

IEEE 754 – Single Precision Format

(Example 1)

Let’s say this is a 32 bit number stored in the Single Precision format

sign	Exponent	Mantissa/Significand
0	1000 0101	00111100000000000000000

Exponent: 2^6

In this fractional part, we can remove all the zeros from the right side

1. 001111

Significand

IEEE 754 – Single Precision Format

(Example 1)

Let's say this is a 32 bit number stored in the Single Precision format

sign	Exponent	Mantissa/Significand
0	1000 0101	00111100000000000000000

Exponent: 2^6

In this fractional part, we can remove all the zeros from the right side

1. 001111

Significand

Actual normalized binary number:

1.001111×2^6

IEEE 754 – Single Precision Format

(Example 1)

Let's say this is a 32 bit number stored in the Single Precision format

sign	Exponent	Mantissa/Significand
0	1000 0101	00111100000000000000000

Exponent: 2^6

Significand: 1. 001111

Normalized binary number: 1.001111×2^6

Actual binary number: 1001111

IEEE 754 – Single Precision Format

(Example 1)

Let's say this is a 32 bit number stored in the Single Precision format

sign	Exponent	Mantissa/Significand
0	1000 0101	00111100000000000000000

Exponent: 2^6

Significand: 1. 001111

Normalized binary number: 1.001111×2^6

Actual binary number: 1001111 → $(79)_{10}$

IEEE 754 – Single Precision Format

(Example 2)

sign	Exponent	Mantissa/Significand
1	1000 0011	11001100000000000000000

?

IEEE 754 – Single Precision Format

(Example 2)

sign	Exponent	Mantissa/Significand
1	1000 0011	11001100000000000000000
	2^4	1. 110011

$1.110011 \times 2^4 \rightarrow 11100.11 \rightarrow (-28.75)_{10}$

IEEE 754 – Single Precision Format

Let’s try to represent a number

IEEE 754 – Single Precision Format

(Example 3)

$(12.625)_{10}$

IEEE 754 – Single Precision Format

(Example 3)

$(12.625)_{10}$

$(12)_{10} \rightarrow (1100)_2$ **And** $(.625)_{10} \rightarrow (101)_2$

IEEE 754 – Single Precision Format

(Example 3)

$(12.625)_{10}$

$(12)_{10} \rightarrow (1100)_2$ **And** $(.625)_{10} \rightarrow (101)_2$

$(1100.101)_2$

$\pm 1.BBB \times 2^{\pm exp}$

Sign

Fraction

Exponent

IEEE 754 – Single Precision Format

(Example 3)

$(12.625)_{10}$ $(12)_{10} \rightarrow (1100)_2$ And $(.625)_{10} \rightarrow (101)_2$

$(1100.101)_2$

$\pm 1.BBB \times 2^{\pm exp}$

Sign

Fraction

Exponent

$(1100.101)_2 = 1.100101 \times 2^3$

IEEE 754 – Single Precision Format

IEEE 754 – Single Precision Format

IEEE 754 – Single Precision Format

(Example 3)

$+\cancel{1}.100101 \times 2^3$

Sign

sign	Exponent	Mantissa/Significand
0		

IEEE 754 – Single Precision Format

(Example 3)

$+\cancel{1}.100101 \times 2^3$

Sign

Fraction

sign	Exponent	Mantissa/Significand
0		100101

IEEE 754 – Single Precision Format

(Example 3)

$+\cancel{1}.100101 \times 2^3$

Sign

Fraction

sign	Exponent	Mantissa/Significand
0		100101

IEEE 754 – Single Precision Format

(Example 3)

$+\cancel{1}.100101 \times 2^3$

Sign

Fraction

sign	Exponent	Mantissa/Significand
0		10010100000000000000000

IEEE 754 – Single Precision Format

(Example 3)

$+\cancel{1}.100101 \times 2^{3}$

Sign

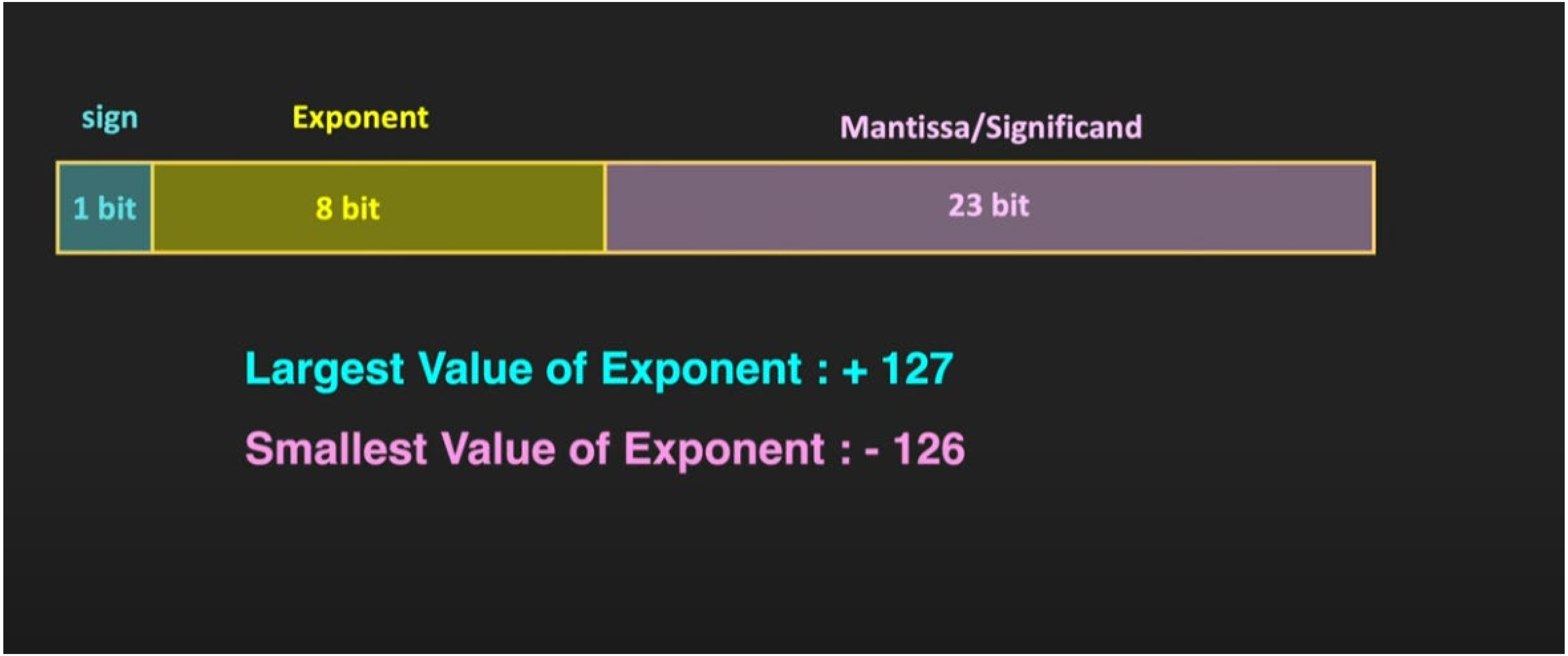
Fraction

Exponent

$$\text{Stored Exponent} = \text{Actual Exponent} + \text{Bias}$$
$$= 3 + 127$$
$$= 130$$

sign	Exponent	Mantissa/Significand
0	1000 0010	10010100000000000000000

IEEE 754 – Single Precision Format Range

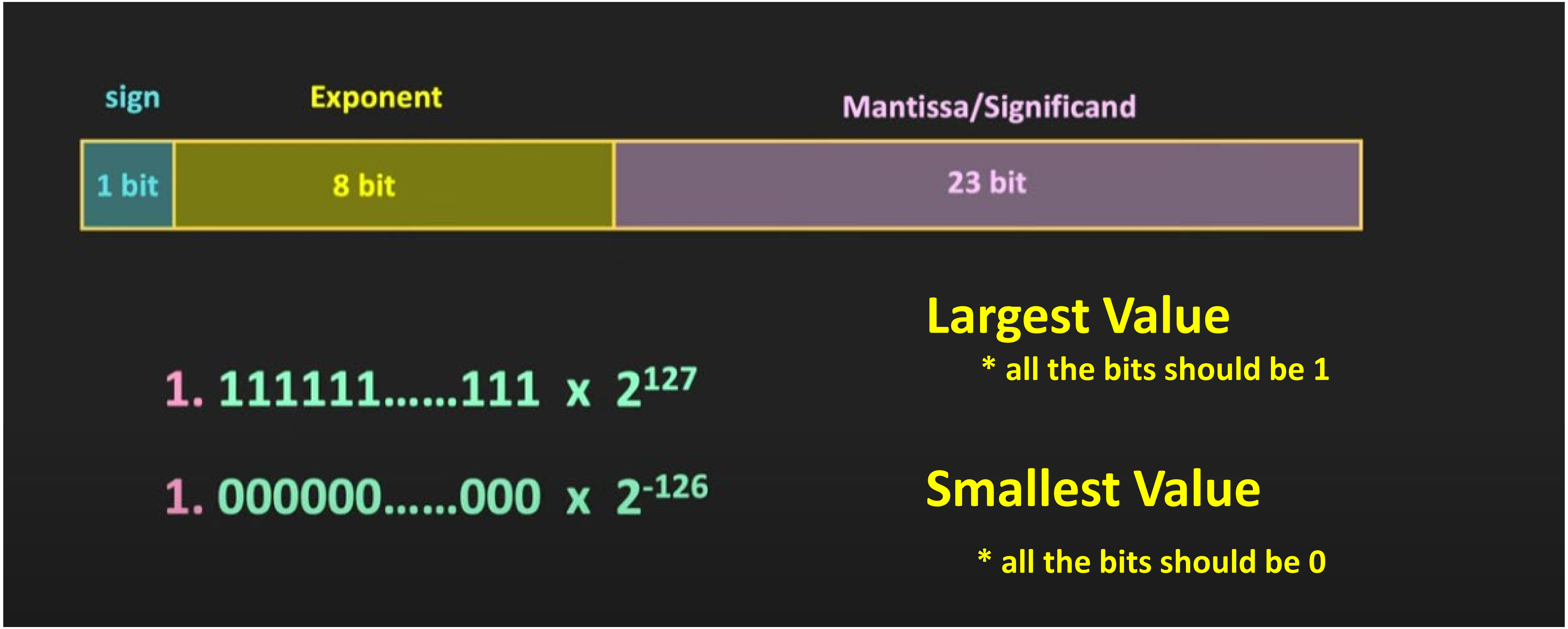


IEEE 754 – Single Precision Format Range

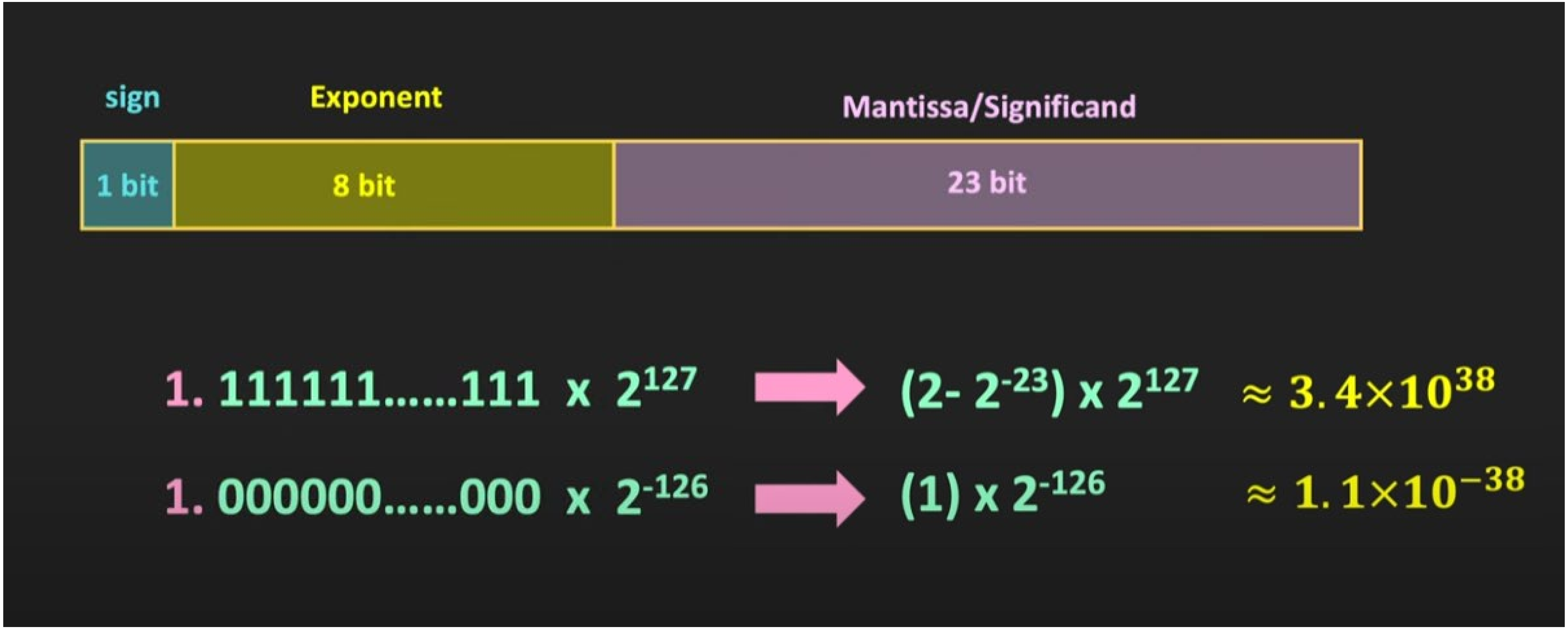
$$1.\text{bbbbbb......bbb} \times 2^{127}$$

$$1.\text{bbbbbb......bbb} \times 2^{-126}$$

IEEE 754 – Single Precision Format Range



IEEE 754 – Single Precision Format Range



IEEE 754 – Single Precision Format vs Fixed Point

Single Precision Format (32 bit)

$Largest\ Number \approx 3.4{\times}10^{38}$

$Smallest\ Number \approx 1.1{\times}10^{-38}$

32 bit Fixed Point Representation (Signed Integer)

$Largest\ Positive\ Number \approx 2.1{\times}10^{9}$

IEEE 754 – Single Precision Format vs Fixed Point

Single Precision Format (32 bit)

$Largest\ Number \approx 3.4{\times}10^{38}$

Why it covers greater range ?

$Smallest\ Number \approx 1.1{\times}10^{-38}$

Floating point allows range at the cost of precision

sign	Exponent	Mantissa/Significand
1 bit	8 bit	23 bit

IEEE 754 – Single Precision Format vs Fixed Point

Single Precision Format (32 bit)

$$Largest\ Number \approx 3.4\times10^{38}$$

$$Smallest\ Number \approx 1.1\times10^{-38}$$

7 Significant Digits in Decimal

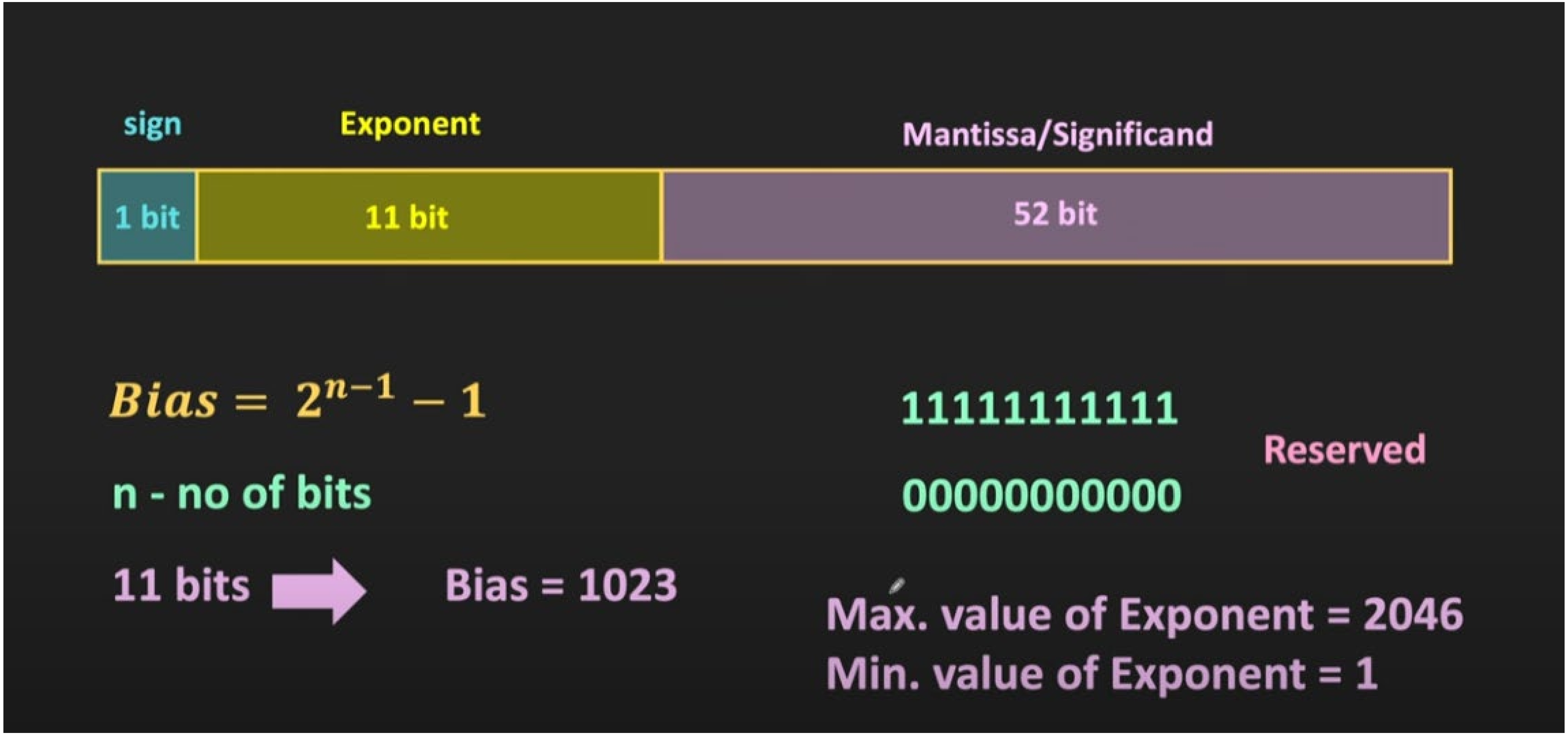
sign	Exponent	Mantissa/Significand
1 bit	8 bit	23 bit

IEEE 754 – Single Precision Format vs Fixed Point

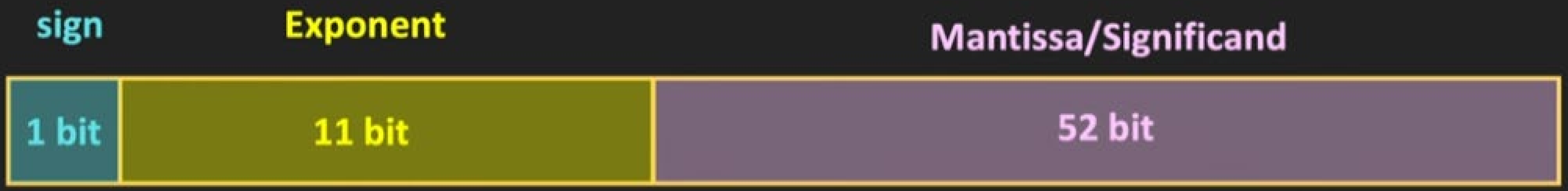
IEEE 754 – Single Precision Format vs Fixed Point

IEEE 754 – Single Precision Format vs Fixed Point

IEEE 754 – Double Precision Format



IEEE 754 – Double Precision Format



Actual max. value of Exponent = 2046- 1023 = 1023

Actual min. value of Exponent = 1- 1023 = -1022

Max. value of Exponent = 2046

Min. value of Exponent = 1

IEEE 754 – Double Precision Format

Actual max. value of Exponent = 2046- 1023 = 1023 $\quad 2^{1023}$

Actual min. value of Exponent = 1- 1023 = -1022 $\quad 2^{-1022}$

$$Smallest\ Positive\ Number \approx 2.2 \times 10^{-308}$$

$$Largest\ Positive\ Number \approx 1.797 \times 10^{308}$$

Thank you

Any Question?