COMP 1521

WEEK 7!

CONTENT

to write -x in n bits of data:

$$\text{binary} = 2^{n} - X$$

2's complement

The result of 2^n-X is identical to an operation called 2's complement.

0000 0101 = 5

1111 1011 =-5 =2^8 - 5 = 251

2's complement

The result of 2^n-X is identical to an operation called 2's complement.

0000 0101 = 5

1111 1011 =-5 =2^8 - 5 = 251

translation:

the negative of a number is just the 2's complement!

2's complement

The result of 2^n-X is identical to an operation called 2's complement.

0000 0101 = 5

1111 1011 =-5 =2^8 - 5 = 251

To do 2's compliment:

2's complement

The result of 2^n-X is identical to an operation called 2's complement.

0000 0101 = 5

1111 1011 =-5 =2^8 - 5 = 251

To do 2's compliment:

1.take the ‘not’ of the binary

0000 0101

1111 1010

invert, or ‘not’

2's complement

The result of 2^n-X is identical to an operation called 2's complement.

0000 0101 = 5

1111 1011 =-5 =2^8 - 5 = 251

To do 2's compliment:

1.take the ‘not’ of the binary

0000 0101

1111 1010

invert, or ‘not’

<- this is known as the 1’s compliment!

2. add 1 to the number

1111

1010

+

1

1111 1011

Taking 2's compliment is much faster than calculating 2^n-X for larger n’s!

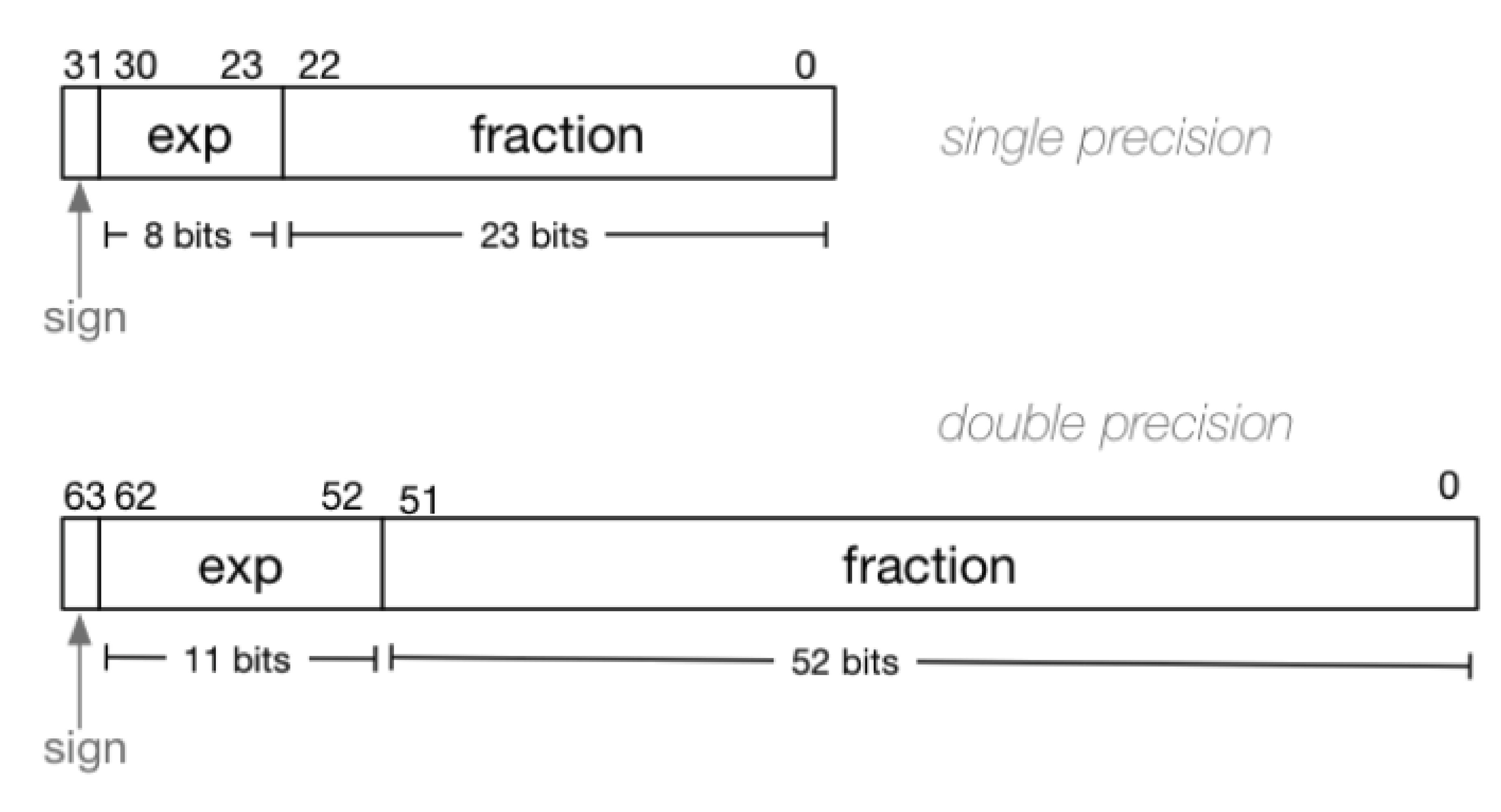
it is equivalent to *-1

Q1 iii, iv, v
Q2 ii, v, vi,
vii

Floats (decimals)

- also known as IEEE 754 standard or IEEE 754-1985
- IEEE 754 single -> float
- IEEE 754 double -> double

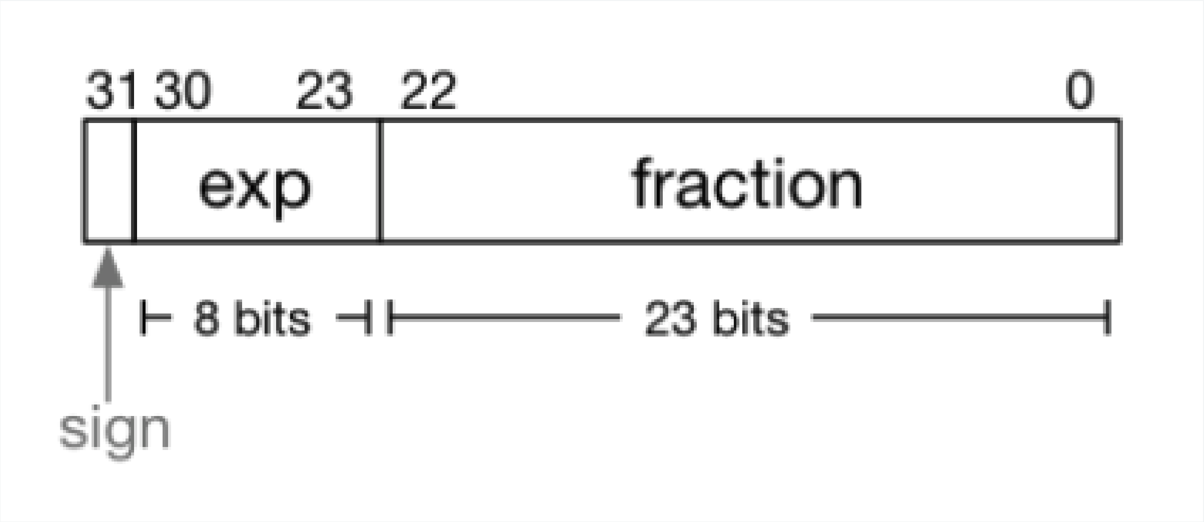
- `float` ... typically 32-bit (lower precision, narrower range)
- `double` ... typically 64-bit (higher precision, wider range)
- `long double` ... typically 128-bits (but maybe only 80 bits used)



Floats (decimals)

Overall Formula:

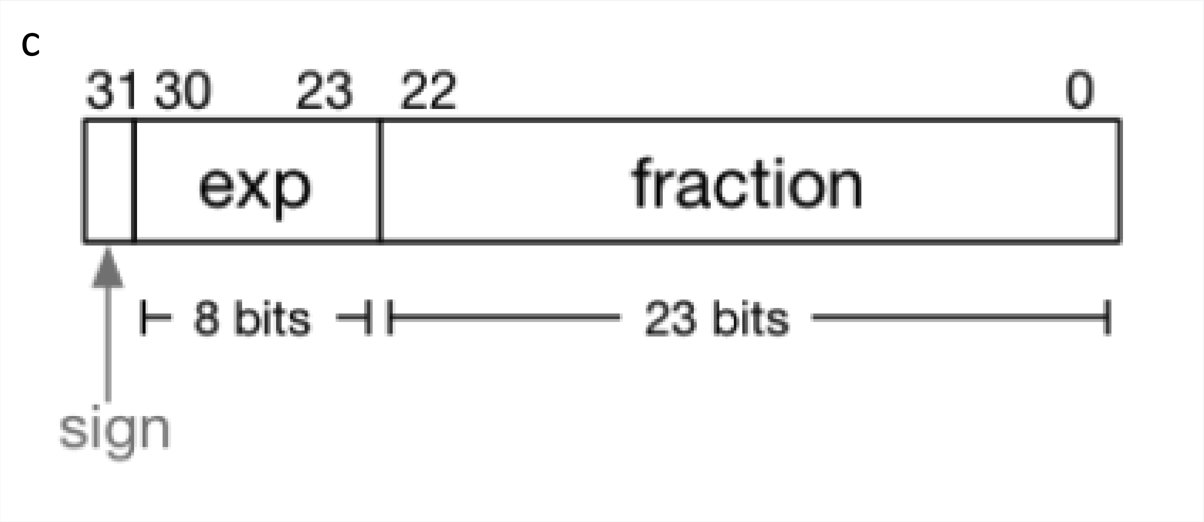
$(-1)^{sign} * (1.frac) * 2^{exp - 127}$



Floats (decimals)

1.Sign

- The sign is the easiest .
- 0 -> positive number
- 1 -> negative number



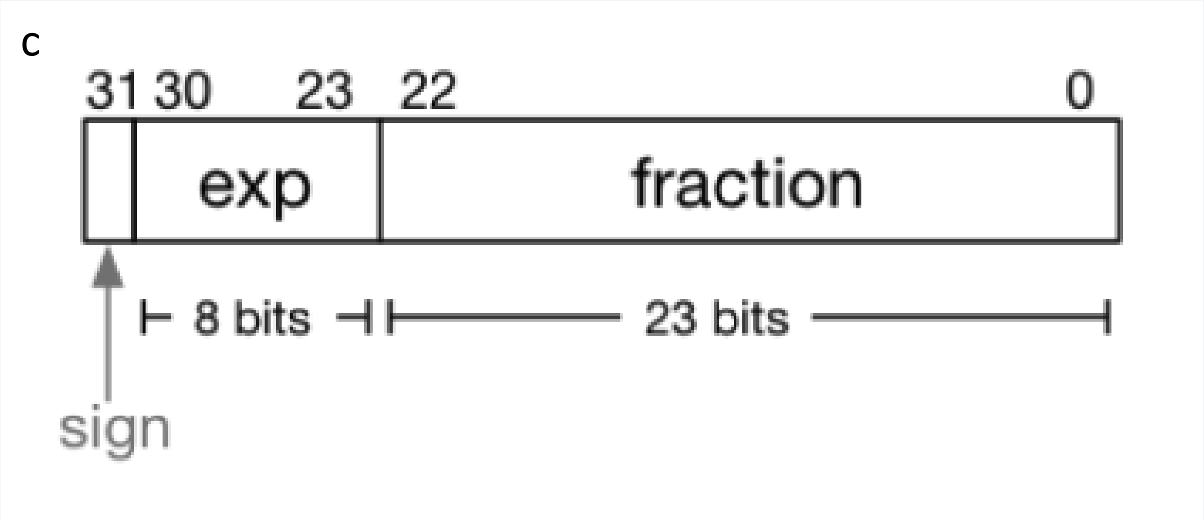
Overall Formula:

$(-1)^{sign} * (1.frac) * 2^{exp - 127}$

Floats (decimals)

1.Sign

- The sign is the easiest .
- 0 -> positive number
- 1 -> negative number



Overall Formula:

$(-1)^{sign} * (1.frac) * 2^{exp - 127}$

sign = 0 ⇒ (-1)^0 = 1 （positive）

sign = 1 ⇒ (-1)^1 = -1 （negative）

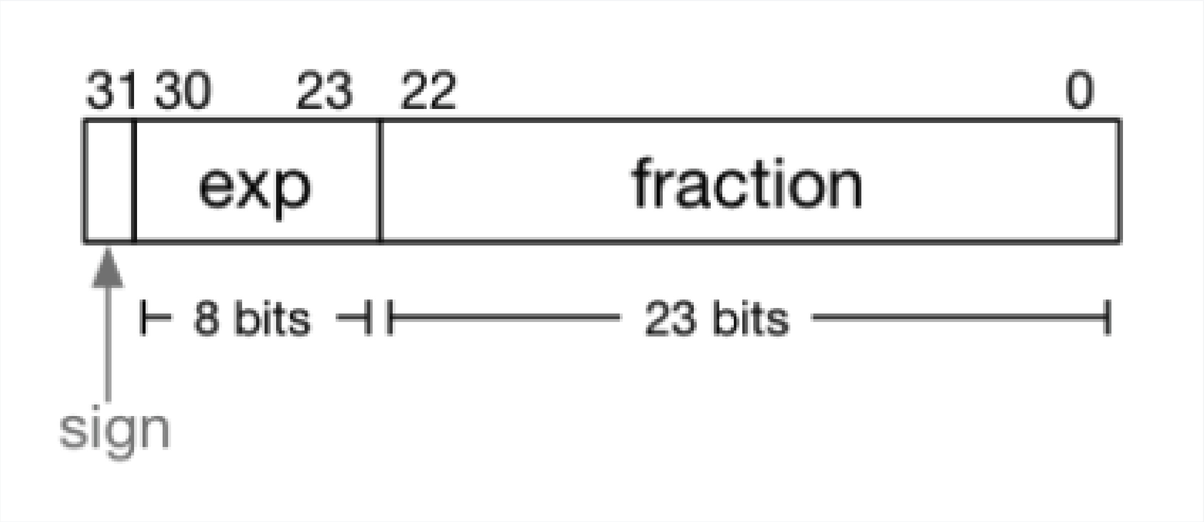
Floats (decimals)

1.Sign

- The sign is the easiest .
- 0 -> positive number
- 1 -> negative number

2**. exponent**

- exp has a bias of -127 added to it
- $2^{exp - 127}$



Overall Formula:

$(-1)^{sign} * (1.frac) * 2^{exp - 127}$

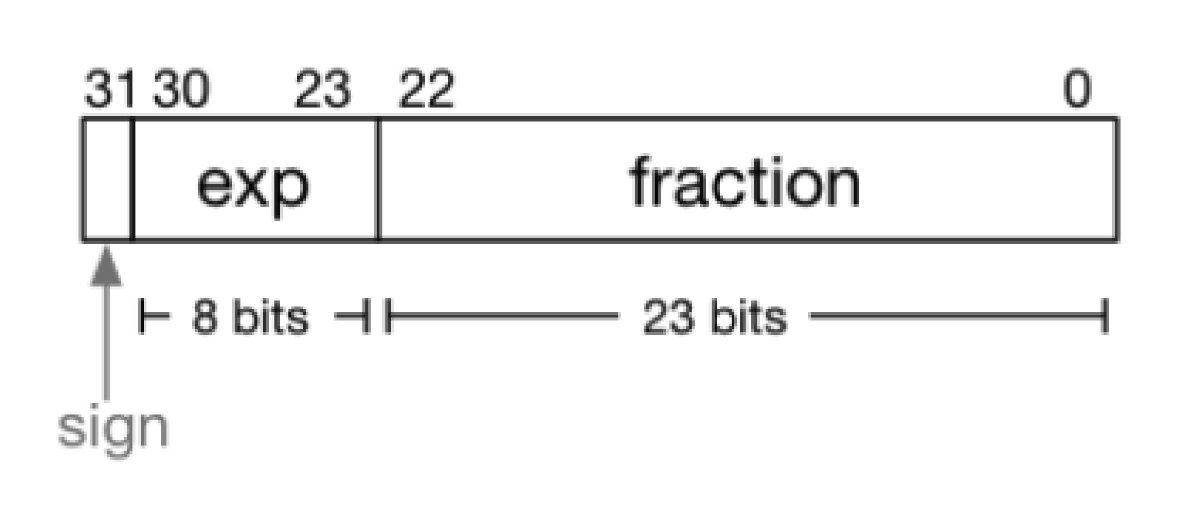
Floats (decimals)

1.Sign

- The sign is the easiest .
- 0 -> positive number
- 1 -> negative number

2**. exponent**

- exp has a bias of -127 added to it
- $2^{exp - 127}$



Overall Formula:

$(-1)^{sign} * (1.frac) * 2^{exp - 127}$

Exp is an 8-bit **unsigned integer**
The **raw exponent field (exp)** ranges from 0 to 255.
The **actual exponent (e = exp - 127)** would *mathematically* range from: -127 to 128

Floats (decimals)

1.Sign

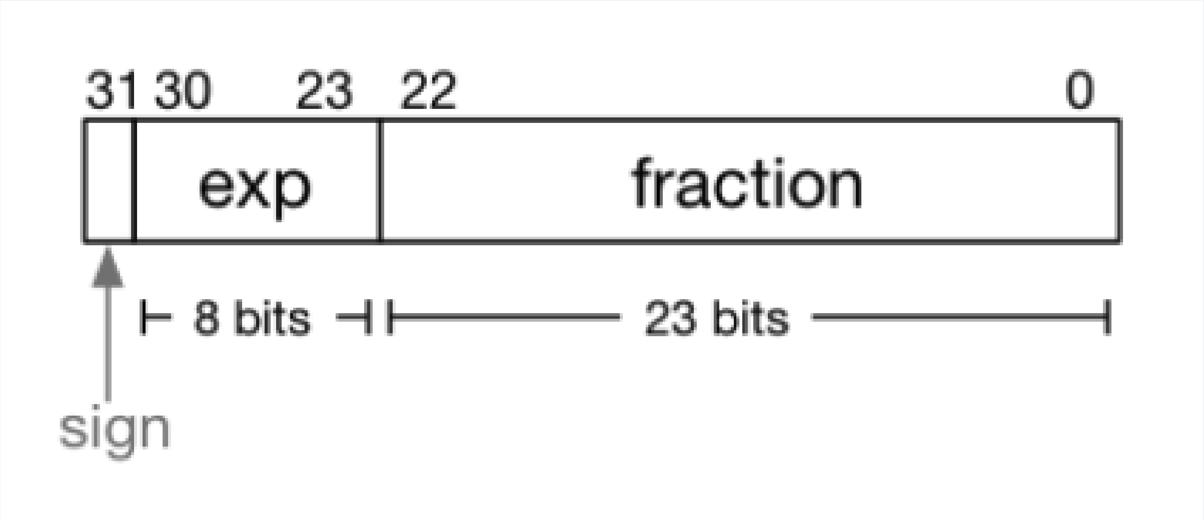
- The sign is the easiest .
- 0 -> positive number
- 1 -> negative number

2. **exponent**

- exp has a bias of -127 added to it
- $2^{exp - 127}$

3. **fraction**

- the fraction is concatenated to the end of 1
- e.g. for a fraction of “0101 1010”,
- 1.frac means 1.0101 1010 (binary decimal)



Overall Formula:

$$(-1)^{sign} * (1.frac) * 2^{exp - 127}$$

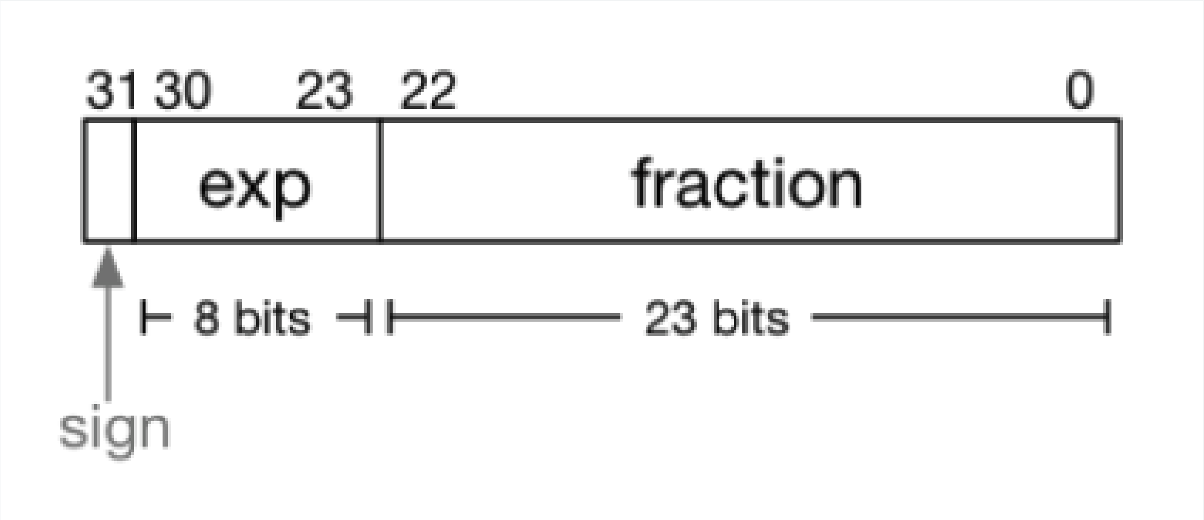
Floats (decimals)

1.Sign

- The sign is the easiest .
- 0 -> positive number
- 1 -> negative number

2. **exponent**
   - exp has a bias of -127 added to it
   - $2^{\text{exp - 127}}$

3. **fraction**
   - the fraction is concatenated to the end of 1
   - e.g. for a fraction of “0101 1010”,
   - 1.frac means 1.0101 1010 (binary decimal)



Overall Formula:

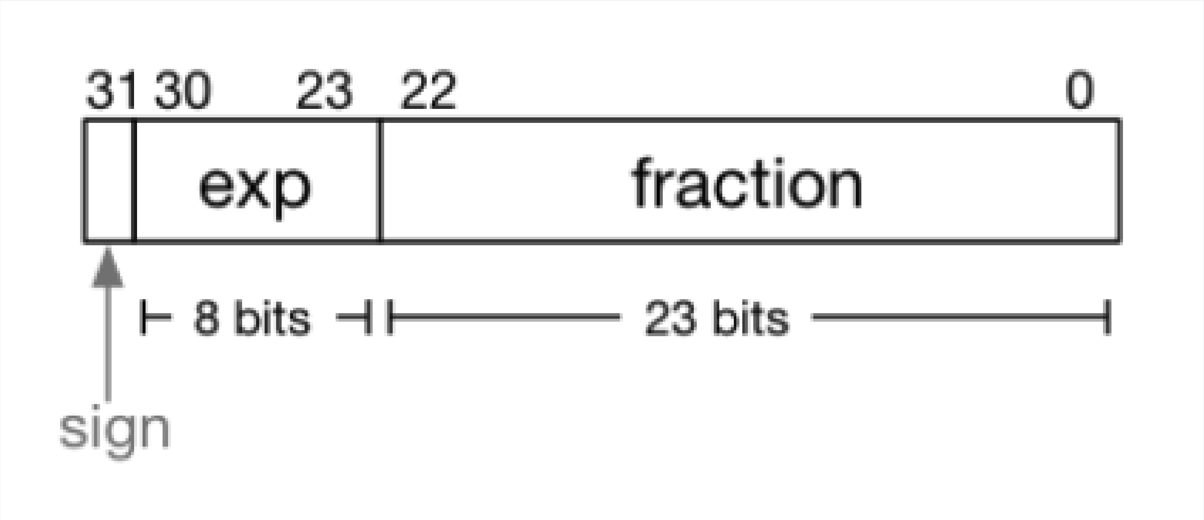
$$(-1)^{\text{sign}} * (1.\text{frac}) * 2^{\text{exp - 127}}$$

1.101 is

$$1 * 2^0 + 1 * 2^{-1} + 0 * 2^{-2} + 1 * 2^{-3}$$

Floats (decimals)

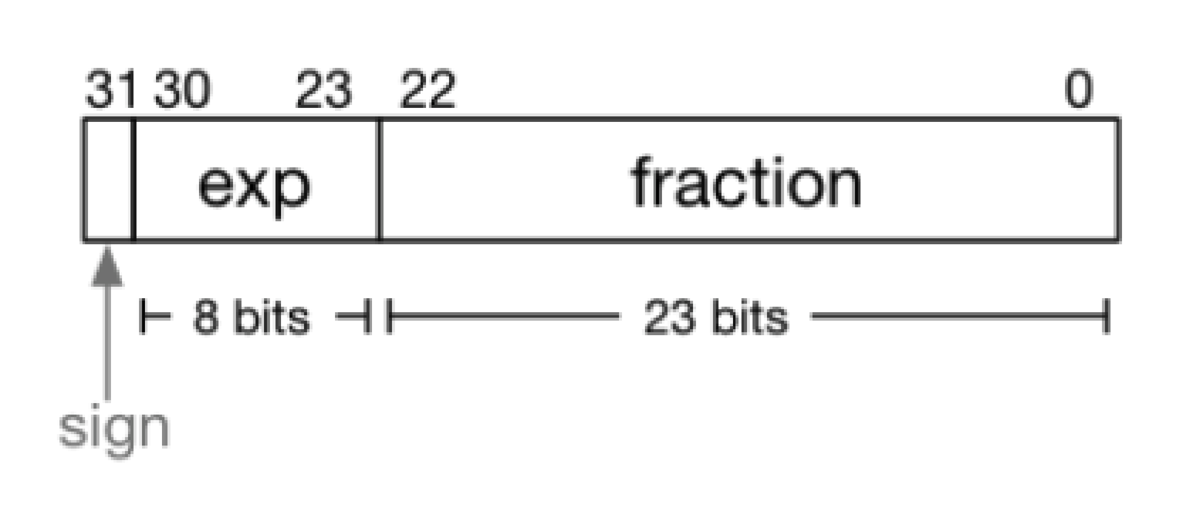
```
$ ./explain_float_representation -96.125
-96.125 is represented in IEEE-754 single-precision by these bits:
11000010110000000100000000000000
sign | exponent | fraction
   1 | 10000101 | 10000000100000000000000
sign bit = 1
sign = -
raw exponent    = 10000101 binary
                = 133 decimal
actual exponent = 133 - exponent_bias
                = 133 - 127
                = 6
number = -1.10000000100000000000000 binary * 2**6
       = -1.50195 decimal * 2**6
       = -1.50195 * 64
       = -96.125
```



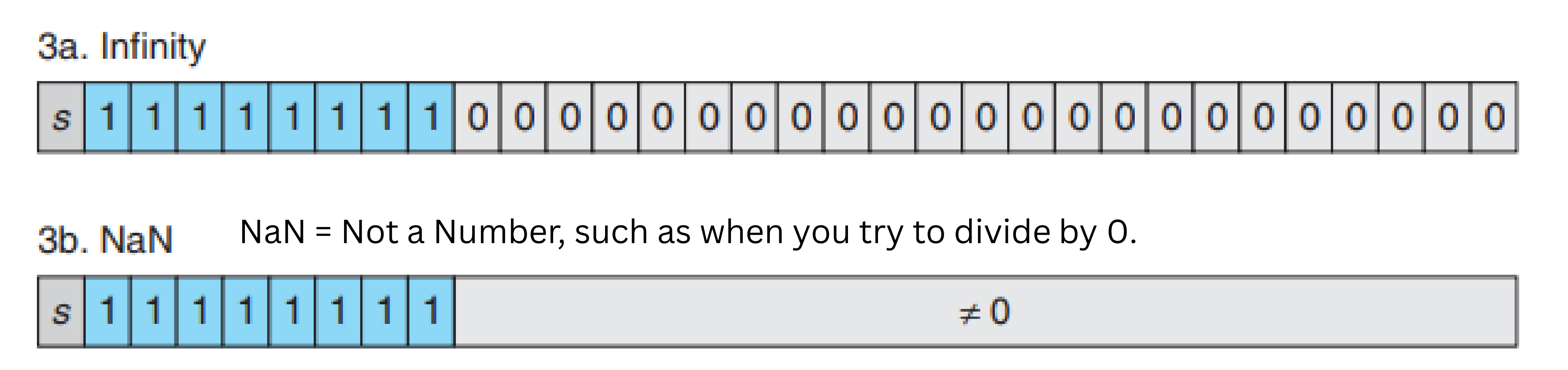
Overall Formula:

$$(-1)^{sign} * (1.frac) * 2^{exp - 127}$$

Floats (decimals)



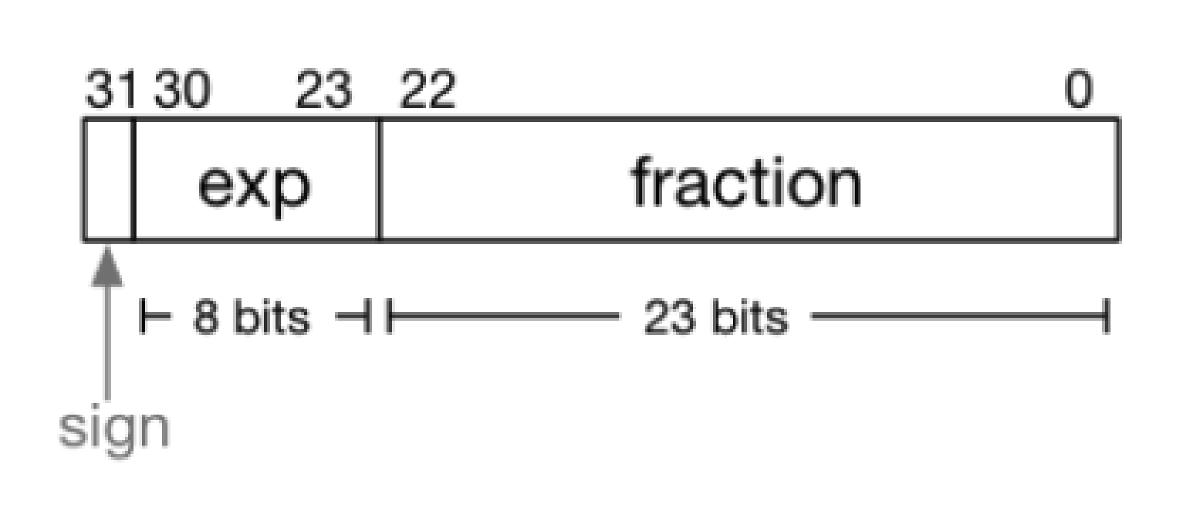
Special Cases:



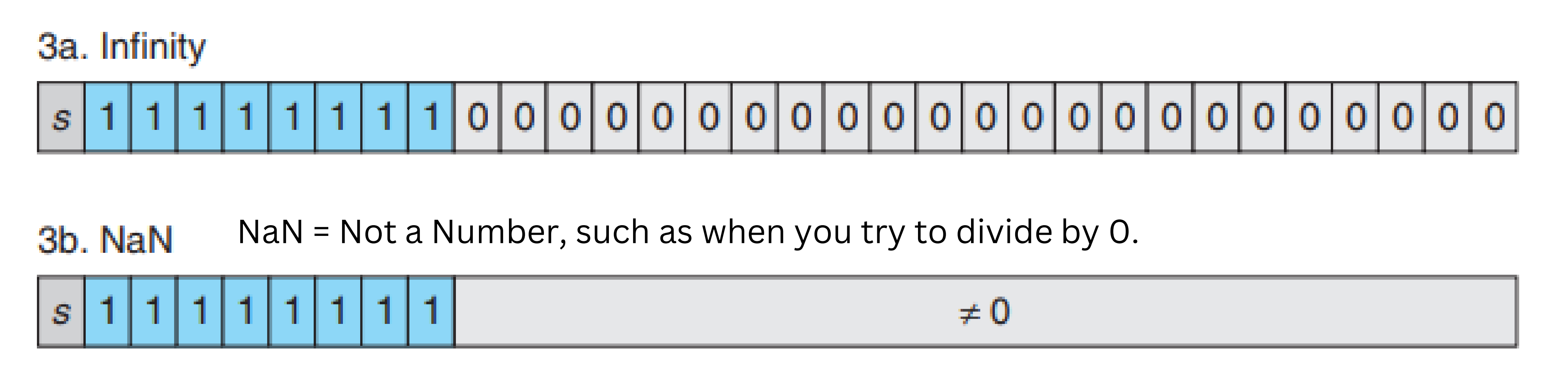
+ and - inf are both defined! (so pay attention to the sign)

NaN is usually not defined as + or -

Floats (decimals)



Special Cases:



+ and - inf are both defined! (so pay attention to the sign)

NaN is usually not defined as + or -

The trick is to always check if the exp bits >= 0xFF!

q4 a->f

working backwards

how to make a number K into float form?

> We need to first express the number k as $(1 + \mathbf{frac}) \times 2^n$. To work out the fraction, we divide k by the largest 2^n that is smaller than k.

q5

Labs - extract the components of a float