

This convention is called **biased-notation**, with the **bias** being the number that is subtracted from the normal, unsigned representation to determine the real value of the exponent part.

IEEE-754 uses a *bias* of $2^{(\text{number of bits in exponent}) - 1} - 1$ (*) therefore, in our quad-precision data type $\text{bias} = 16383$.

Quad-precision floating point - according to IEEE-754 standard:

Sign	Exponent (15bits)	Mantissa (112bits)	Meaning
0	000000000000000 ₂	0000000...0000000 ₂	+0
1	000000000000000 ₂	0000000...0000000 ₂	-0
0	000000000000000 ₂	Non-Zero	+Denormal
1	000000000000000 ₂	Non-Zero	-Denormal
0	111111111111111 ₂	0000000...0000000 ₂	+Infinity
1	111111111111111 ₂	0000000...0000000 ₂	-Infinity
0	111111111111111 ₂	Non-Zero	NaN
1	111111111111111 ₂	Non-Zero	NaN

A number X in the above (normalized) representation can be calculated as:
 $X = (-1)^{\text{sign}} \times (1 + \text{significand}) \times 2^{\text{exponent} - \text{bias}}$

In the normalized form there are no leading '0' in the fraction part, therefore any number in normalized form has a leading '1' in the integer part of the fraction part and is called *hidden bit* and fraction is computed implicitly by adding 1.0 to the significand. Denormalized floating-point numbers, fill up the gap between 0 and the smallest normalized number, allowing us to extend the system's representable range.

Minimal value: 1.0×2^{-16382}
 Maximal value: $1.0 \times 2^{+16383}$

Accuracy:

References:

- [1]. Computer Organization & Design: The hardware/software interface
- [2]. [IEEE Standard 754 for Binary Floating-Point Arithmetic](#) (W. Kahan)

(*): The exponents 0000000000000000000000000000000₂ and 1111111111111111111111111111111₂ are reserved for special purpose signals.