♢ General format

IEEE (the Institute for Electrical and Electronic Engineers) has defined the following standard for representing a number V in floating-point format. The representation consists of

1. A *sign* bit s, which determines whether the number is positive ($s = 0$) or negative ($s = 1$).
2. A k-bit *exponent field* e, which encodes an exponent E in the range $\left[1 - \left(2^{k-1} - 1\right), 2^{k-1} - 1\right]$.
3. An n-bit *significand field* m, which encodes a significand M in the range $[0, 2)$.

All together, these bits represent a number $V = (-1)^s \times M \times 2^E$.

♢ The exponent field

The true exponent E is encoded in the bits e in what is known as *biased form*. If we treat e as an unsigned integer, then we can find E by subtracting the *bias*: $E = e - \left(2^{k-1} - 1\right)$. Conversely, we can put some exponent E in biased form by adding the bias.

Two values of e in IEEE are reserved and have a special interpretation:

- If e is all-bits 0, then we say that $E = 1 - \left(2^{k-1} - 1\right)$ and the significand is *denormalized* (see below).
- If e is all-bits 1, then V represents a special value determined by the remaining bits:
  - if m is all-bits 0, then $V = \pm\infty$ (depending on the value of s)
  - if m is nonzero, then $V =$ *NaN* ("not a number"). *NaN* results from operations such as $\sqrt{-1}$, which are mathematically legal but whose result does not have any representation in the IEEE format.

♢ The significand field

Except when e is all-bits 0 or all-bits 1, the significand M is interpreted to be in *normalized* form, which means that there must be exactly one digit to the left of the radix point, and that digit must be nonzero. However, in binary there is only one nonzero digit (1), so we take it as "assumed" and do not store it in the significand field m. (This trick gets us an additional bit of precision for free.) Since the leading one is implicit, m is actually a *fraction* in the range $[0, 1)$. Therefore m represents a significand $M = 1 + m$, which is in the range $[1, 2)$.

If the significand M were always in the range $[1, 2)$, we would not be able to represent the number 0, or other very small numbers. To get around this, we use a *denormalized* significand when e is all-bits 0. This means that with such an exponent field, we do not assume that there is a leading 1. The significand field m still encodes a fraction in the range $[0, 1)$. In this case $M = m$.

♢ Summary

To convert a number from IEEE floating-point $s\,e\,m$ to fixed-point V:

- If e is all-bits 1, then V is *NaN* or $\pm\infty$.
- If e is all-bits 0, then
$$V = (-1)^s \times m \times 2^{1-\left(2^{k-1}-1\right)}.$$
- Otherwise,
$$V = (-1)^s \times (1+m) \times 2^{e-\left(2^{k-1}-1\right)}.$$

To convert most numbers[1] from fixed-point V to IEEE floating point $s\,e\,m$:

- If $V > 0$, let $s = 0$.
  If $V < 0$, let $s = 1$.
- Convert V to binary and normalize the significand so that it is in the range $[1, 2)$. You end up with a number $1.v_1v_2v_3\ldots \times 2^E$
- Let $m = v_1v_2v_3\ldots$. Note that m should be exactly n bits long.
  - If m is too short, pad 0s to the end.
  - If m is too long, chop off all digits after the nth.
- Add the bias to get $e = E + \left(2^{k-1} - 1\right)$. Convert e to binary.

♢ Real-life examples

Single-precision IEEE uses an 8-bit exponent field and a 23-bit significand field. Therefore the bias is $2^{8-1} - 1 = 127$.

s	e	m			V
0	00000000	00000000000000000000000		$=$	0
1	00000000	00000000000000000000000		$=$	0
0	11111111	00000000000000000000000		$=$	$+\infty$
1	11111111	00000000000000000000000		$=$	$-\infty$
0	11111111	00000100000000000000000		$=$	*NaN*
0	01111111	00000000000000000000000	$= (-1)^0 \times 1.000_2 \times 2^{127-127}$	$=$	1
0	10000000	00000000000000000000000	$= (-1)^0 \times 1.000_2 \times 2^{128-127}$	$=$	2
0	10000001	10100000000000000000000	$= (-1)^0 \times 1.101_2 \times 2^{129-127}$	$=$	6.5
1	10000001	10100000000000000000000	$= (-1)^1 \times 1.101_2 \times 2^{129-127}$	$=$	-6.5
0	00000000	10000000000000000000000	$= (-1)^0 \times 0.100_2 \times 2^{1-127}$	$\approx$	5.877×10^{-39}
0	11111110	00100000000000000000000	$= (-1)^0 \times 1.001_2 \times 2^{254-127}$	$\approx$	1.914×10^{38}

[1] To simplify things, I am omitting the case where we end up with a denormalized significand.