FPHUB

Julio Villalba-Moreno, Javier Hormigo-Aguilar

July 2025

1 The format FPHUB

In this document, we propose solutions to some problems that the previous floating-point HUB version had [1]. Specifically, we define the special cases related to 0, 1, and infinity. We also define five standardized floating-point HUB formats with precision similar to their IEEE counterparts. We have named this new version of floating-point HUB numbers as FPHUB.

In order to handle special cases, we introduce a modification in the code of a floating-point HUB number, which affects the exponent field. Moreover, not-a-number (NaN) and subnormal are not considered in FPHUB, allowing us to expand normalized numbers' exponent. In FPHUB, the exponent is now represented in excess–$2^{n_{exp}-1}$, where n_{exp} is the number of bits of the exponent. In the first version of HUB, the exponent was defined as IEEE does, that is, in excess–$(2^{n_{exp}-1}-1)$.

Thus, with the new representation of the exponent, let us define x as a floating-point radix–2 FPHUB number, which is coded by the triple (S_x, E_x, M_x), where S_x is the sign, M_x is the fractional part of the normalized significand (f explicit fractional bits), and E_x is the exponent represented in excess–$2^{n_{exp}-1}$ such that

$$x = (-1)^{S_x}(1 + M_x + 2^{-f-1})2^{E_x - 2^{n_{exp}-1}} \tag{1}$$

where the value 1 is the integer part of the mantissa (implicit integer bit), and 2^{-f-1} represents the characteristic bias of the HUB format — that is, the Implicit Least Significant (fractional) Bit, also called ILSB.

2 Standardized Floating Point HUB formats

We define five standardized FPHUB formats, namely FPHUB16, FPHUB32, FPHUB64, FPHUB128, and FPHUB256, which have similar precision to their counterparts binary16, binary32, binary64, binary128, and binary256, respectively, as defined by IEEE [2–4].

Table 1 shows the size of the different fields. We can see that the number of exponent bits and the precision in bits (rightmost column) are the same as those of their IEEE counterparts.

Table 1: FPHUB-defined formats

Format	Total bits	Exp. bits	Significand bits	Precision (bits)
FPHUB16	16	5	10 (12 with implicit bits)	11
FPHUB32	32	8	23 (25 with implicit bits)	24
FPHUB64	64	11	52 (54 with implicit bits)	53
FPHUB128	128	15	112 (114 with implicit bits)	113
FPHUB256	256	19	236 (238 with implicit bits)	237

3 Special cases

We now define special codes for special cases (in a way similar to how the IEEE does). The special cases are zero, one, and infinity.

3.1 Zero

The zero case is coded as "all 0's" for the exponent and significand fields; the sign field can be 0 or 1. This is similar to the IEEE special case for zero.

$$(S_x, E_x, M_x) = (S_x, 0, 0) \tag{2}$$

3.2 One

The one case is coded using "all 0's" for the significand field, and the exponent is coded as $2^{n_{exp}-1}$:

$$(S_x, E_x, M_x) = (S_x, 2^{n_{exp}-1}, 0) \tag{3}$$

Note that there is both a positive one and a negative one. For example, for 8 bits for the exponent, the code for $+1$ is $(0, 128, 0)$, and for -1 we have $(1, 128, 0)$.

3.3 Infinity

The infinity case is coded as "all 1's" for the significand and exponent fields; the sign field can be 0 or 1.

$$(S_x, E_x, M_x) = (1, 2^{n_{exp}} - 1, 2^{n_m} - 1) \tag{4}$$

where n_m is the number of bits in the field m. For example, for a precision of 24 bits (23 fractional bits, $n_m = 23$) and an exponent of 8 bits ($n_{exp} = 8$), we represent infinity as $\infty = (0, 255, 2^{23} - 1)$.

In Table 2, we summarize the special cases, illustrating them with an example for FPHUB32 (24-bit precision: 8 bits for the exponent, 23 explicit fractional bits).

Case	Code	Example 23-fractional , 8-bit exponent
0	$(0, 0, 0)$	0 00000000 00000000000000000000000
0	$(1, 0, 0)$	1 00000000 00000000000000000000000
+1	$(0, 128, 0)$	0 10000000 00000000000000000000000
-1	$(1, 128, 0)$	1 10000000 00000000000000000000000
$+\infty$	$(0, 255, 2^{23} - 1)$	0 11111111 11111111111111111111111
$-\infty$	$(1, 255, 2^{23} - 1)$	1 11111111 11111111111111111111111

Table 2: Summary of special cases in FPHUB32

3.4 Operation

Let x be a FPHUB number other than one, zero, and infinity. The operations in which these special cases are involved are:

- $x + zero = x$
- $x * zero = zero$
- $x * one = x$
- $x \pm \infty = \pm\infty$
- $zero * one = zero$
- $x/one = x$
- $x/zero = \infty$ $(sign = sign(x))$
- $zero/x = zero$
- $zero/zero = +\infty$
- $x * \infty = \infty$ $(sign = sign(x))$
- $zero/\infty = \infty$
- $\infty/zero = \infty$
- $\infty * zero = \infty$

References

[1] J. Hormigo and J. Villalba-Moreno, "New Formats for Computing with Real-Numbers under Round-to-Nearest," *IEEE Transactions on Computers*, vol. 65, no. 7, pp. 2158–2168, 2016. doi:10.1109/TC.2015.2479623

[2] IEEE Computer Society, *IEEE Standard for Binary Floating-Point Arithmetic*, IEEE Std 754-1985, Aug. 1985. doi:10.1109/IEEESTD.1985.82928

[3] IEEE Computer Society, *IEEE Standard for Floating-Point Arithmetic*, IEEE Std 754-2008, Aug. 2008. doi:10.1109/IEEESTD.2008.4610935

[4] IEEE Computer Society, *IEEE Standard for Floating-Point Arithmetic*, IEEE Std 754-2019, Jul. 2019. doi:10.1109/IEEESTD.2019.8766229