An Update Regarding the Exponent Bias of the P3109 8-bit Floating Point Formats

Michael L. Overton

July 20, 2025

The second release of the IEEE P3109 Working Group interim report, dated 29 October 2024, defines several 8-bit floating point formats called *binary8pp*, where the second p in the name[1] refers to the precision p, for $p = 1, \ldots, 7$. Among other parameters, the report defines the exponent bias as 2^{w-1} for precisions p from 2 to 7, where w is the width of the exponent field, so that, taking account of the sign bit and the hidden significand bit, $w = 8 - p$. (This formula does not apply to $p = 1$ which is a special case.) One of the motivations was that as a result, $E_{\min} = -E_{\max}$, where $E_{\min}$ and $E_{\max}$ are respectively the minimum and maximum possible exponents in the floating point format. This choice for the bias is in contrast with the IEEE 754 exponent bias $2^{w-1} - 1$.

On 28 April 2025, the P3109 Working Group voted to change the exponent bias to $2^{w-1} - 1$, in part to be more consistent with the value set provided by the widely used OCP formats with the same precision. The second edition of my book *Numerical Computing with IEEE Floating Point Arithmetic* was by this time nearing final production by SIAM. However, because of the decision to change the bias, I asked SIAM for a few days to make the necessary changes to incorporate the new bias into the discussion in Chapter 15. This was granted, and soon afterward, the book went to press, stating that the P3109 bias was $2^{w-1} - 1$, and noting in a footnote that *The presentation here mostly follows the interim report, but also recent decisions by the working group which are not yet incorporated into the report.*

However, a few weeks later, on 7 July 2025, the decision was reversed by the Working Group, changing the bias back to 2^{w-1}.

As a result, Table 15.2 in the book should be replaced by the one shown

[1]The name in the report is actually *binary8pP*, where P is the precision, but we use *binary8pp* to be consistent with the widespread usage of p for precision.

Table 15.2: Parameters for three 8-bit floating point formats recommended by the P3109 interim report: precision p, exponent bit width w, maximum exponent, minimum exponent, exponent bias, maximum normalized number, minimum positive normalized number, minimum positive subnormal number. Notice the different formulas for $E_{\min}$, *bias* and $N_{\max}$ compared to those given in Table 4.3 for the three basic formats in the 2019 754 IEEE standard and in Table 15.1 for the commonly used 16-bit formats. The same formulas apply to $p = 2$, $p = 6$ and $p = 7$, but not to $p = 1$: see the discussion on p. 106.

Format	*binary8p3*	*binary8p4*	*binary8p5*
p	3	4	5
$\epsilon_{\text{mch}} = 2^{-(p-1)}$	0.25	0.125	.0625
$w = 8 - p$	5	4	3
$E_{\max} = 2^{w-1} - 1$	15	7	3
$E_{\min} = -2^{w-1} + 1$	-15	-7	-3
$bias = 2^{w-1}$	16	8	4
$N_{\max} = 2^{E_{\max}}\left(2 - 2^{-(p-2)}\right)$	49152	224	15
$N_{\min} = 2^{E_{\min}}$	$\approx 3.1 \times 10^{-5}$	$\approx 7.8 \times 10^{-3}$	$\approx 1.3 \times 10^{-1}$
$S_{\min} = 2^{E_{\min}-(p-1)}$	$\approx 7.6 \times 10^{-6}$	$\approx 9.8 \times 10^{-4}$	$\approx 7.8 \times 10^{-3}$

here, and the penultimate sentence in the long paragraph on p. 105 should be replaced by:

> Noting that there is a lack of symmetry in the IEEE 754 formats in the sense that $E_{\max} = 2^{w-1} - 1$ but $E_{\min} = -2^{w-1} + 2$, it is recommended that for the 8-bit format representation, $E_{\max}$ remains equal to $2^{w-1} - 1$ but $E_{\min}$ is changed to $-2^{w-1} + 1$. This is accomplished by setting the exponent bias to 2^{w-1}.

On p. 106, in the third paragraph, the two sentences discussing *binary8p7* should be replaced by:

> In the case *binary8p7*, with $p = 7$ and $w = 1$, the formulas given in Table 15.2 give $E_{\max} = E_{\min} = 0$, and so all normalized and subnormal numbers have the exponent 0. Hence, the format becomes essentially a scaled sign-and-magnitude integer format, with subnormal values $\pm 1/64, \ldots, \pm 63/64$ and normalized values $\pm 64/64, \ldots, \pm 126/64$, as well as $0, \pm\infty$ and NaN.