Recent Developments in Floating Point

James Brakefield

Short float formats for machine learning

https://en.wikipedia.org/wiki/Minifloat
https://uwplse.org/2025/02/17/Small-Floats.html

There are better and more recent lists (than mine) of floating-point **formats for various computers**:

Robert Munafo. *Survey of Floating-Point Formats* (with references)
http://www.mrob.com/pub/math/floatformats.html
John Savard. *Floating-Point Formats* (with detailed descriptions)
http://www.quadibloc.com/comp/cp0201.htm

POSIT by John Gustafson:

John Gustafson. *The End of Error: Unum Computing*, CRC Press 2015
John Gustafson and Posit Working Group. Standard for Posit™ Arithmetic, 2022.
https://posithub.org/docs/posit_standard-2.pdf

PT-Float:

Various exponent sizes and exponential tapers
Jos´e T. de Sousa etal. PT-Float: *A Floating-Point Unit with Dynamically Varying Exponent and Fraction Sizes.* ARITH 2024.
https://www.ac.uma.es/arith2024/program.html See Session 7 for paper and slides

HUB Round half bit to odd:

Javier Hormigo, and Julio Villalba. *New Formats for Computing with Real-Numbers under Round-to-Nearest* 2015
http://www.ac.uma.es/~hormigo/HUB.htm
https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=10226419

Takum Arithmetic

Laslo Hunhold of University of Cologne:
"Beating Posits at Their Own Game: Takum Arithmetic"
https://arxiv.org/pdf/2404.18603
And also covered in John Gustafson's new book:
Every Bit Counts: Posit Computing
https://www.amazon.com/Every-Bit-Counts-Computing-Computational/dp/1032738065

The Takum paper is 72 pages long, e.g. it is comprehensive.
Hunhold has taken the approach of constraining the exponent size to a +/- 10^55 range irrespective of the mantissa/fraction size.and favoring logarithmic "floats"
An earlier paper shows the optimality of using as a radix the square root of "e"

"Low-precision Logarithmic Number Systems: "Beyond Base-2"
https://dl.acm.org/doi/10.1145/3461699
Takums do not use or need gradual underflow.