Friedrich-Alexander-Universität
Technische Fakultät

Exploring Tapered Precision:

From theory to custom RTL implementation

Felix Blenk, Thomas Schlögl

Introduction to Tapered Precision

Standard IEEE 754 floating-point formats use rigid, fixed-size fields for exponent and mantissa. While robust, this fixed structure often leads to wasted precision or premature underflow/overflow in modern data-driven workloads.
Tapered Precision addresses this limitation by introducing dynamic bit-field boundaries.
By shifting bits dynamically between the exponent and fraction, tapered formats achieve higher accuracy and a wider dynamic range using fewer total bits (e.g., 16-bit Posit vs. 32-bit Float).

- **Dynamic Scaling:** A specialized *regime* field dynamically scales the exponent size based on the magnitude of the represented value.
- **Golden Zone Accuracy:** It concentrates maximum precision and accuracy around 1.0, which is the critical range for most real-world DSP and Deep Learning computations.
- **Graceful Degradation:** Instead of abrupt underflows to zero, precision decreases gradually (*"tapers"*) as numbers approach extreme values.

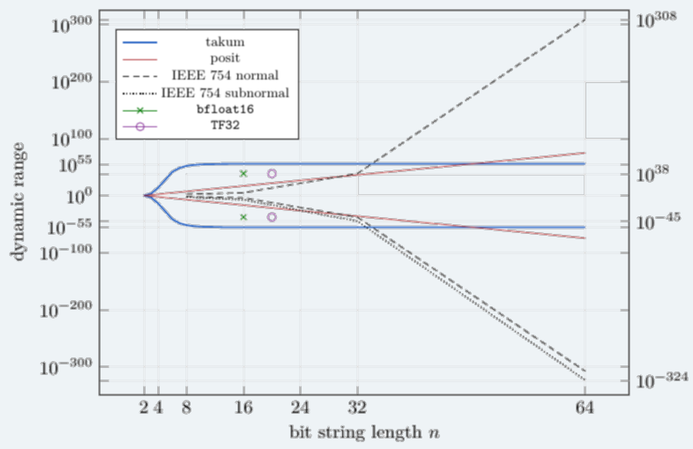
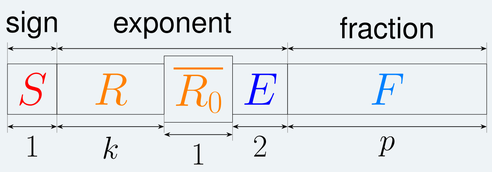


Figure: Dynamic range and precision of tapered formats. [Hun24]

What are Posits?

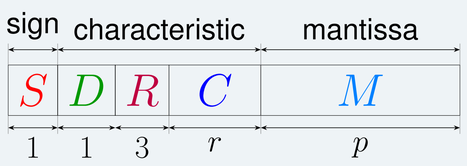


Proposed by John **Gustafson** [GY], **Posits** serve as an alternative to the standard IEEE 754 format. By dynamically adjusting internal field boundaries (Figure 4), they offer distinct advantages:

$$
\begin{aligned}
&S \in \{0,1\} && \text{: Sign}\\
&R := (R_{k-1},\dots,R_0),\ r := \begin{cases} -k & R_0 = 0 \\ k-1 & R_0 = 1 \end{cases} && \text{: Regime}\\
&E := (E_1, E_0),\ e := 2E_1 + E_0 && \text{: Exponent}\\
&f := 2^{-p}\sum_{i=0}^{p-1} F_i 2^i \in [0,1) && \text{: Fraction } (p = n-k-4)\\
&\hat{e} := (-1)^S(4r + e + S) && \text{: Actual Exponent}
\end{aligned}
$$

$$
\pi(S,R,E,F) := \begin{cases} 0 & S=0 \wedge R=E=F=0 \\ \text{NaR} & S=1 \wedge R=E=F=0 \\ \left[(1-3S)+f\right]\cdot 2^{\hat{e}} & \text{otherwise} \end{cases}
$$

What are Takums?



Takums [Hun24] are a closely related extension of the Posit format, designed to minimize redundant bit strings. By utilising a unique base of $\sqrt{e}$, takums reduce the excessive dynamic range of posits, and provide interesting mathematical properties of logarithmic number systems.
The takum value is defined with: Let $R_{\text{val}} = 4R_2 + 2R_1 + R_0 \in [0,7]$ and $C_{\text{val}} = \sum_{i=0}^{r-1} C_i 2^i$. The n-bit string decodes to $\tau = (-1)^S\sqrt{e}^{\ell}$ via:

$$
\begin{aligned}
&S, D \in \{0,1\} && C := (C_{r-1},\dots,C_0)\\
&R := (R_2, R_1, R_0) && m := 2^{-p}\textstyle\sum_{i=0}^{p-1} M_i 2^i\\
&r := \begin{cases} 7 - \sum R_i 2^i & D=0 \\ \sum R_i 2^i & D=1 \end{cases} && \ell := (-1)^S(c+m)\\
&c := \begin{cases} -2^{r+1} + 1 + \sum C_i 2^i & D=0 \\ 2^r - 1 + \sum C_i 2^i & D=1 \end{cases} && p := n - r - 5
\end{aligned}
$$

$$
\tau := \begin{cases} 0 & S=0 \wedge D=R=C=M=0 \\ \text{NaR} & S=1 \wedge D=R=C=M=0 \\ (-1)^S \cdot \sqrt{e}^{\ell} & \text{otherwise} \end{cases}
$$

Comparison

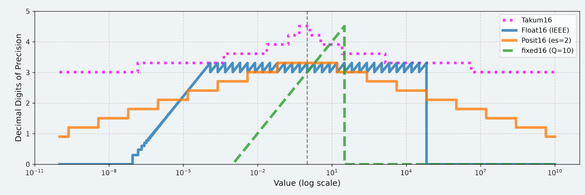


Figure: Dynamic Range of different number formats [Hun24]

Physical Implementation

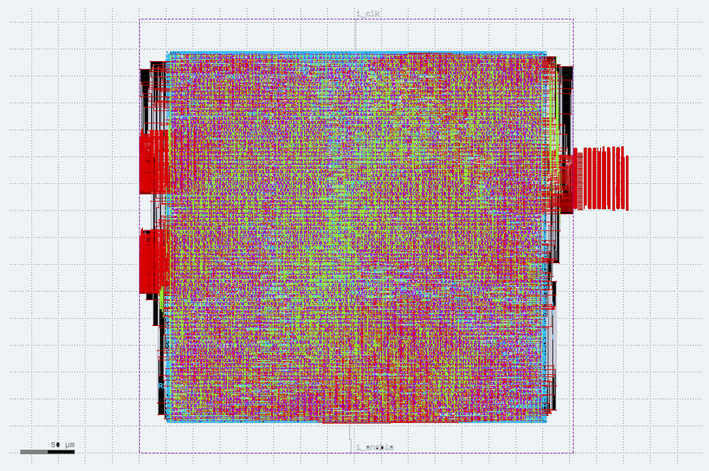


Figure: Posit<32,2> Makro implemented using openroad with ihp sg13g2 pdk

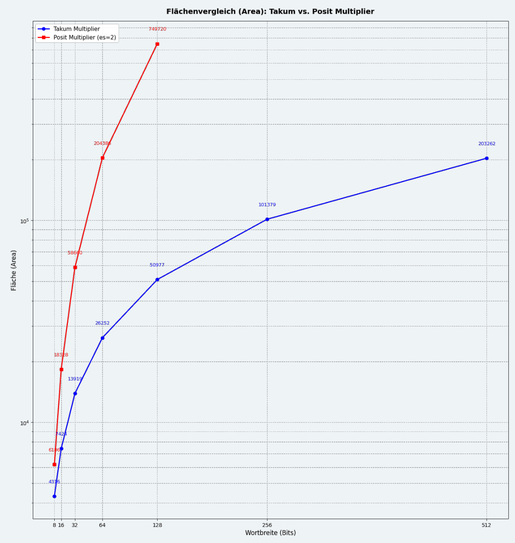


Figure: Area difference in nanometers

References

[GY] J. L. Gustafson and I. Yonemoto. “Beating Floating Point at its Own Game: Posit Arithmetic”. en. In: ().

[Hun24] L. Hunhold. “Beating Posits at Their Own Game: Takum Arithmetic”. In: vol. 14666. arXiv:2404.18603 [math.NA]. 2024, pp. 1–51. DOI: 10.1007/978-3-031-72709-2_1.

{felix.blenk, thomas.schloegl}@fau.de