Friedrich-Alexander-Universität
Technische Fakultät

FAU

EESP Workshop at ISC 2026

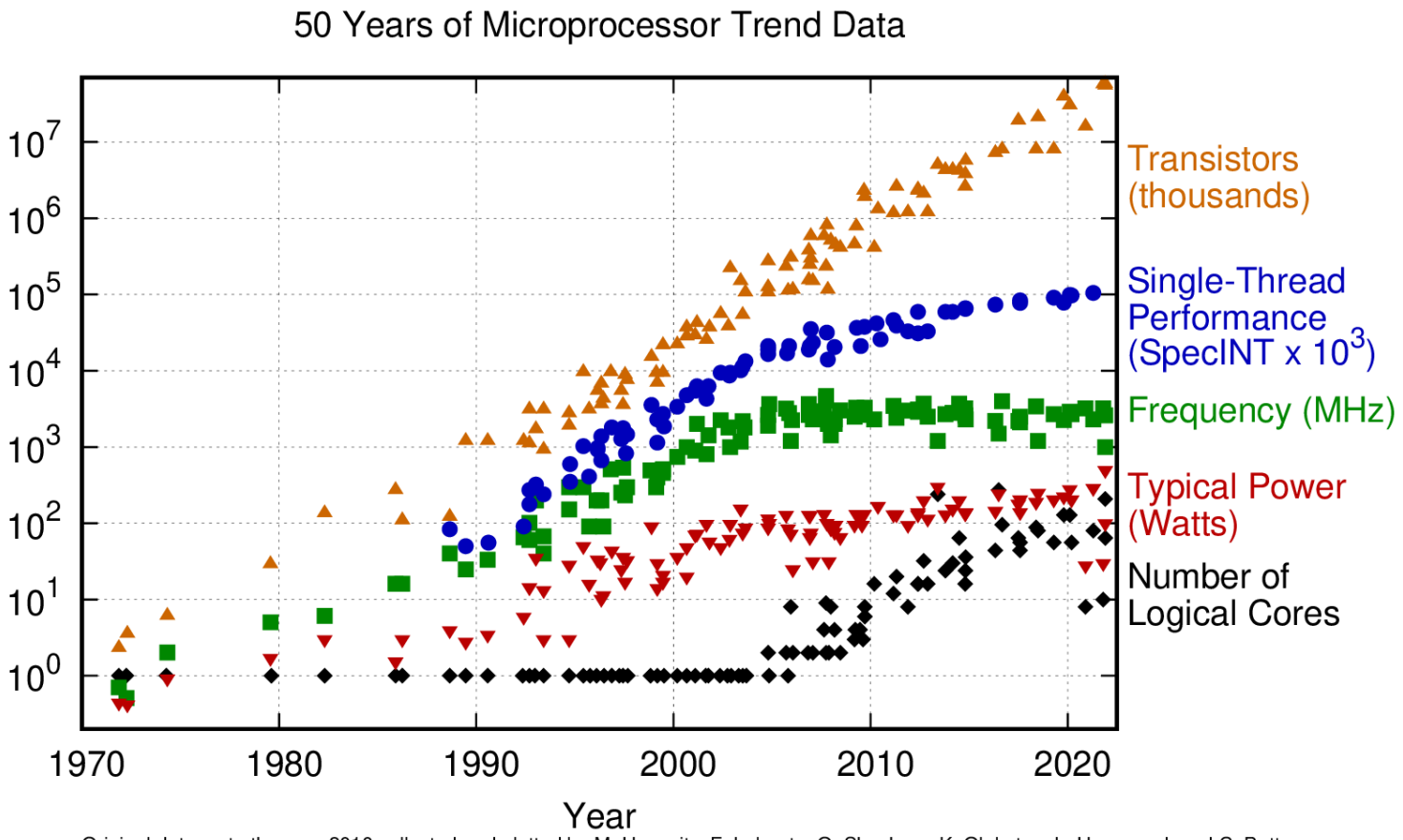
Energy Consumption Comparison of Conjugate Gradient Method using Posit Arithmetic and IEEE Floating Point.

Thomas Schlögl[1] **Tuan Huy Do**[1] **Dietmar Fey**[1]

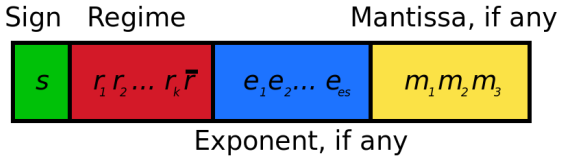
[1]Friedrich-Alexander-Universität Erlangen-Nürnberg, Department Computer Science

June 26, 2026

Motivation
The "Power Wall"



Primer
What are Posits?

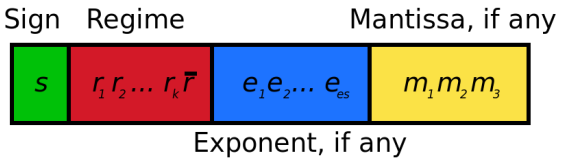


Posit Encoding

$$p = (-1)^{\text{sign bit}} \times \text{useed}^k \times 2^{\text{unbiased exponent}} \times (1.M) \text{ with useed} = 2^{2^{es}}$$

$$k = \begin{cases} r-1, & \text{when regime starts with '1' bit} \\ -r, & \text{else} \end{cases}$$

Primer
What are Posits?



Posit Encoding

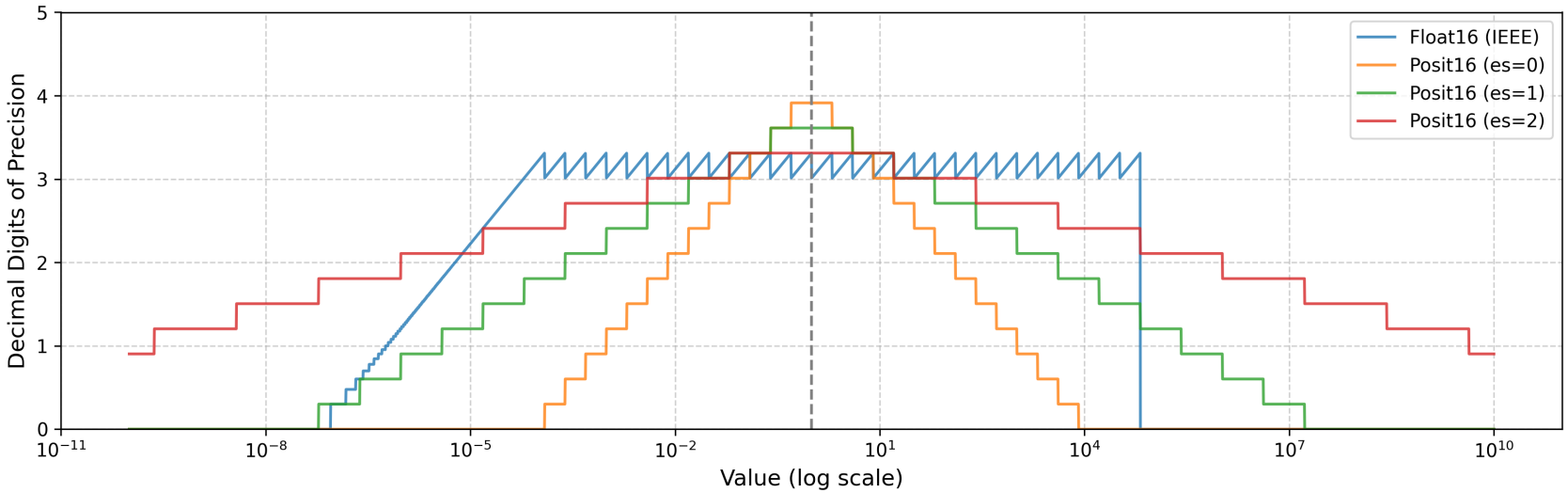
$$p = (-1)^{\text{sign bit}} \times \text{useed}^k \times 2^{\text{unbiased exponent}} \times (1.M) \text{ with useed} = 2^{2^{es}}$$

$$k = \begin{cases} r-1, & \text{when regime starts with '1' bit} \\ -r, & \text{else} \end{cases}$$

Posit<8,1> Examples

$$\text{zero} = 00000000_2$$
$$\text{NAR} = 10000000_2$$

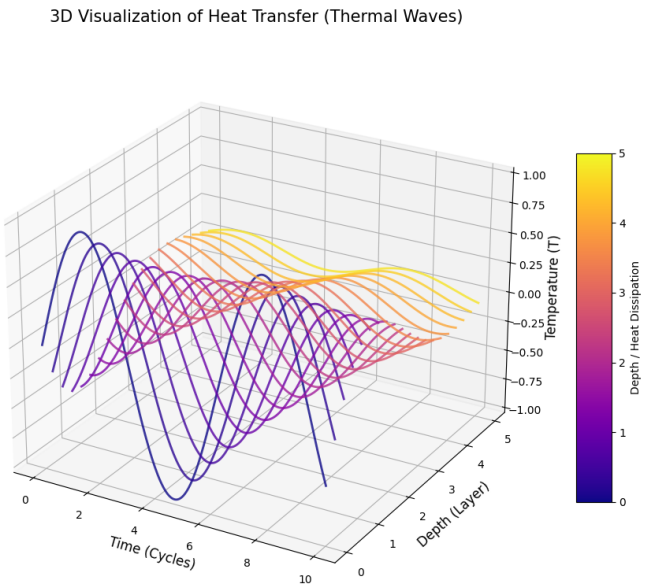
$$2.5 = 01010100_2$$
$$0.0625 = 00010000_2$$
$$-124.0 = 10000110_2$$



The Case Study

Heat Transfer discretized by Finite Differences

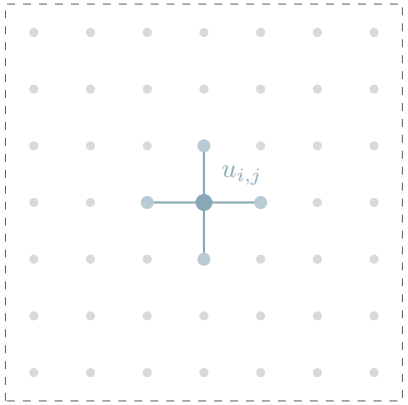
1. The Physics



Poisson Equation: $-\Delta u = f$

Homogeneous Dirichlet boundary conditions on a unit square.

2. Discretization

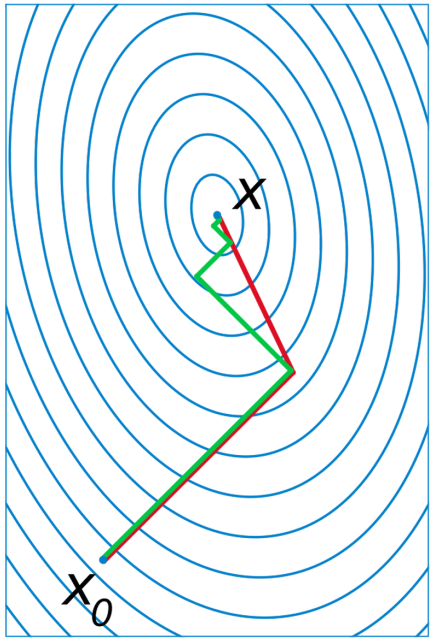


$\Omega = (0,1)^2$

$$L_h u = f$$

Matrix L_h is sparse ($5 \times N$ non-zeros), symmetric, and positive definite.

3. The Solver



Conjugate Gradient

- Iterative Krylov subspace method.
- Dominant ops: **Dot Products** and **SpMV**.

Methodology

Hardware Implementation of Adders and Multipliers

Minimalist IEEE FPU

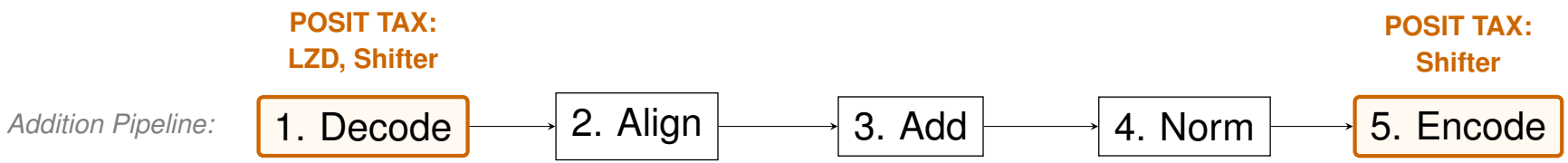
- No subnormal support
- Only one rounding mode (Round-to-Nearest-Even)

vs.

Posit Units (based on Pacogen)

- no special optimizations

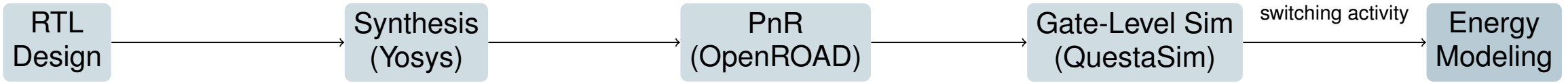
Worst Case Scenario: Advantage given to Float.



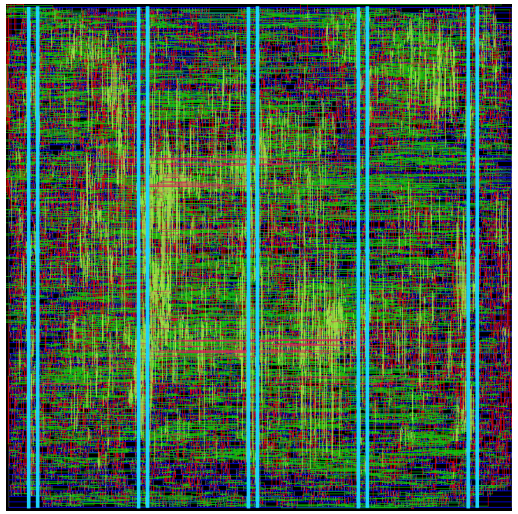
ASIC Evaluation Flow

Post-layout methodology

- **Toolchain:** IHP 130nm PDK, Yosys, OpenROAD (Open-source flow for physical implementation)
- **Goal:** Evaluate Post-Layout Energy/Op (includes *glitches* & *parasitics*)
- **Method:** Linear regression ($y = mx + c$) to isolate energy per operation

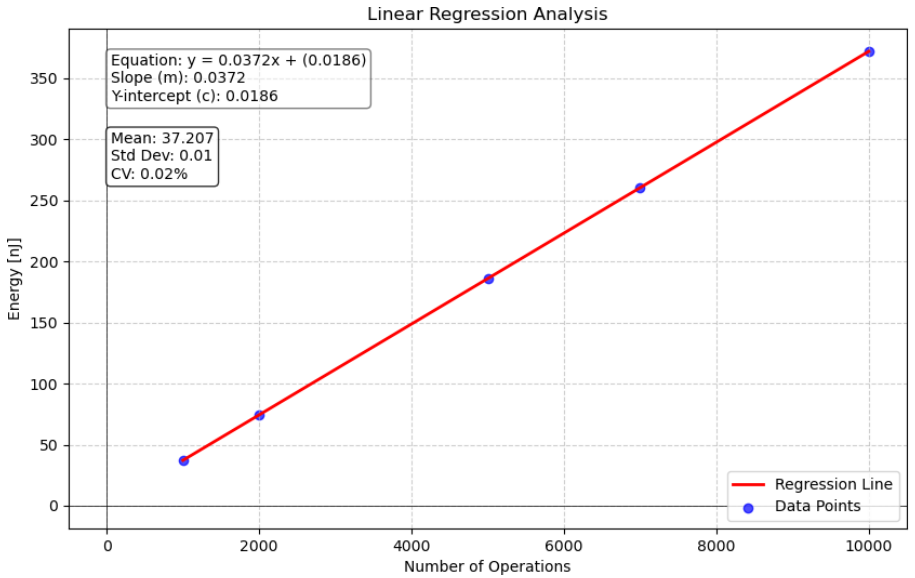


Physical Implementation



Macrocell layout in OpenROAD

Energy Modeling



Results 1
Hardware Resource Overhead

Unit	Metric	IEEE 754	Posit$\langle N, 2\rangle$	Posit$\langle N, 3\rangle$
32-bit Adder	Gate Equiv. (GE)	5,358	7,619 **(+42%)**	7,429 **(+39%)**
	Max Delay (ns)	5.14	6.68 (+29%)	6.41 (+24%)
	ADP	27,540	50,894 (+84%)	47,619 (+72%)
64-bit Adder	Gate Equiv. (GE)	10,392	15,594 **(+50%)**	15,325 **(+47%)**
	Max Delay (ns)	5.56	9.67 (+73%)	8.91 (+60%)
	ADP	57,779	150,793 (+160%)	136,545 (+136%)
32-bit Mult	Gate Equiv. (GE)	9,642	12,816 **(+33%)**	12,442 **(+29%)**
	Max Delay (ns)	6.09	7.38 (+21%)	7.03 (+15%)
	ADP	58,719	94,582 (+61%)	87,467 (+48%)
64-bit Mult	Gate Equiv. (GE)	36,601	47,182 **(+29%)**	46,710 **(+28%)**
	Max Delay (ns)	10.48	10.96 (+4%)	11.77 (+12%)
	ADP	383,579	517,114 (+34%)	549,776 (+43%)

GE: technology independent size comparison (NAND2 gate equivalents).

ADP: Area-Delay Product. Percentages relative to IEEE 754 baseline.

Key Takeaway: All Posit units show overhead (+30 to +50% area), primarily due to complex regime decoding (the "Posit Tax").

Results 3
Energy per Operation

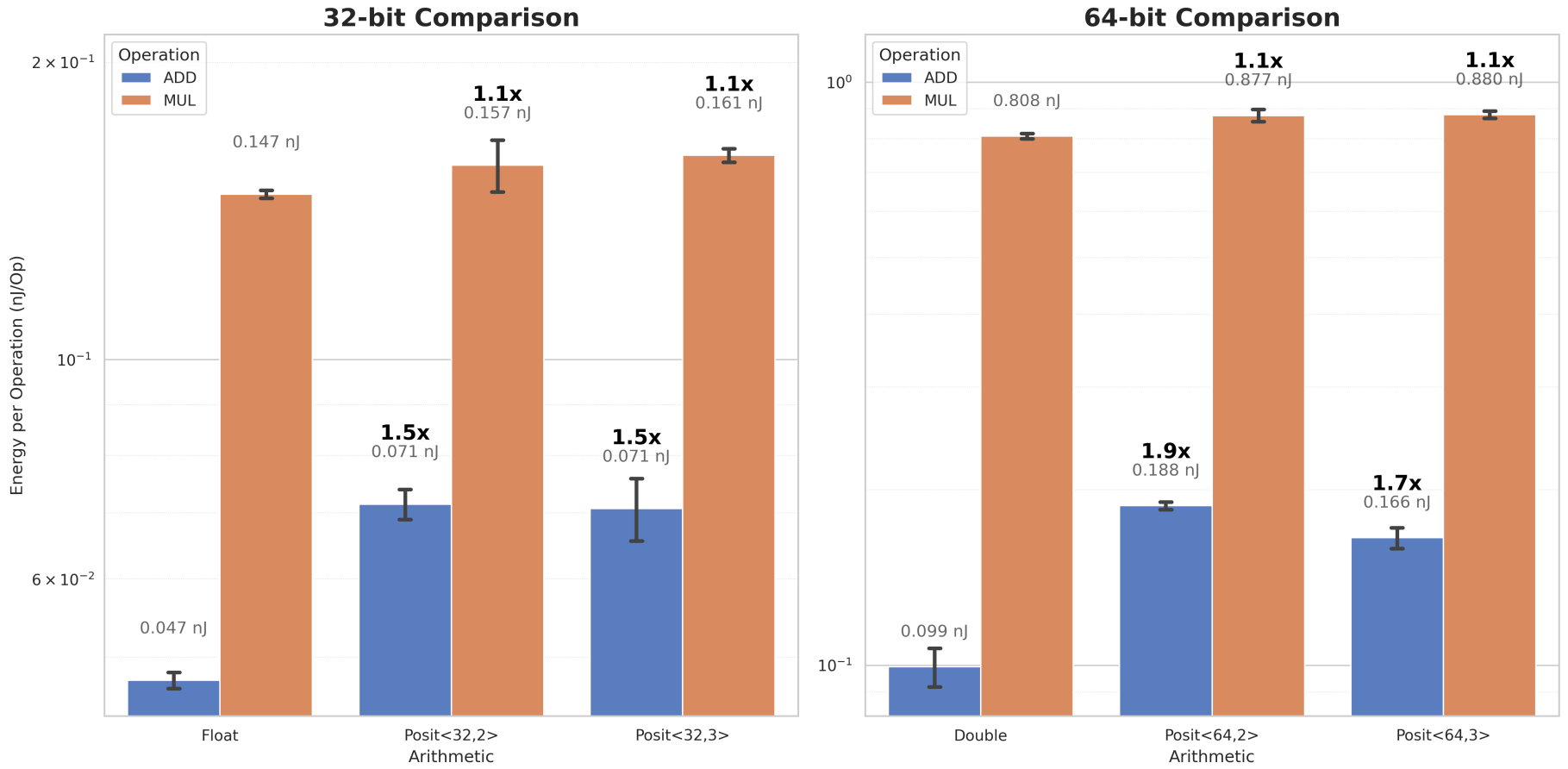
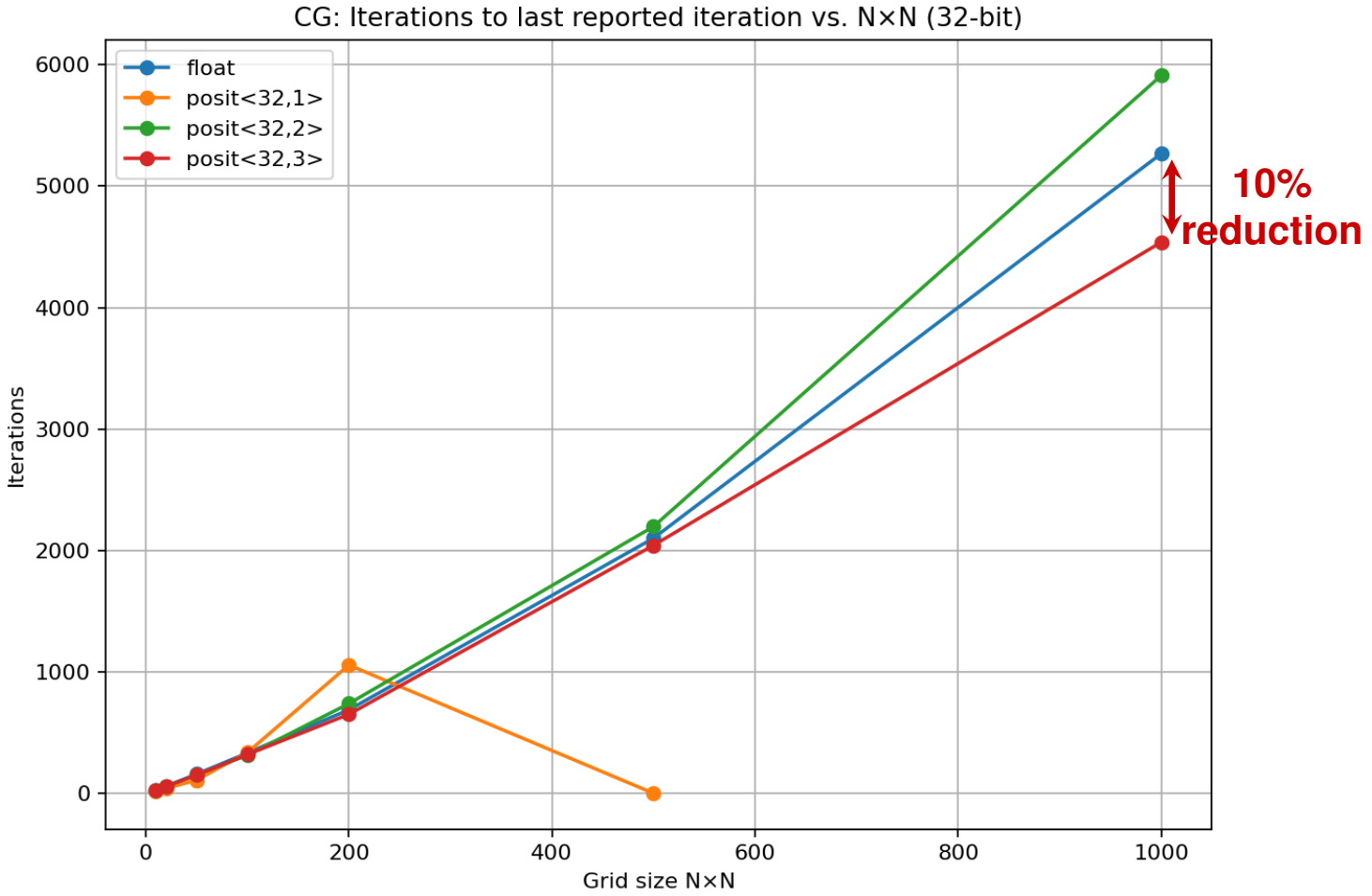
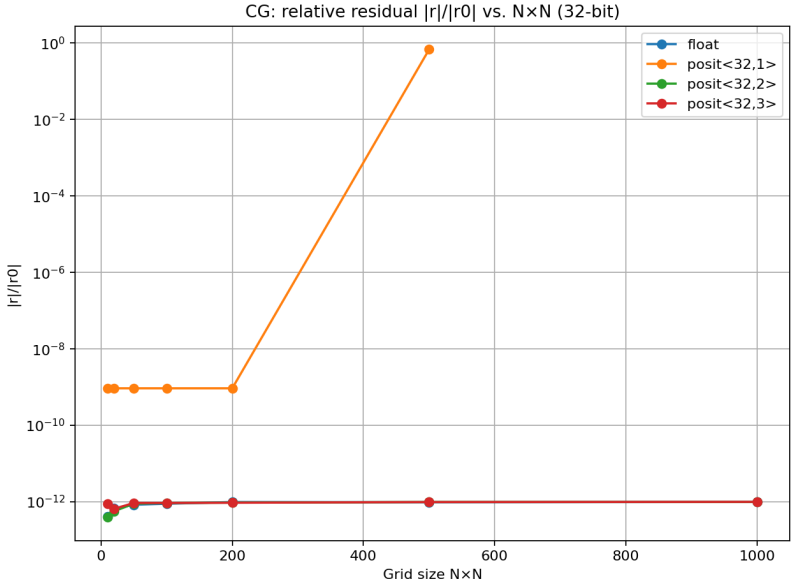


Figure: Energy per operation comparison for additions and multiplications using floating-point and Posit arithmetic. The evaluation was done using values extracted during the calculation of the CG algorithm.

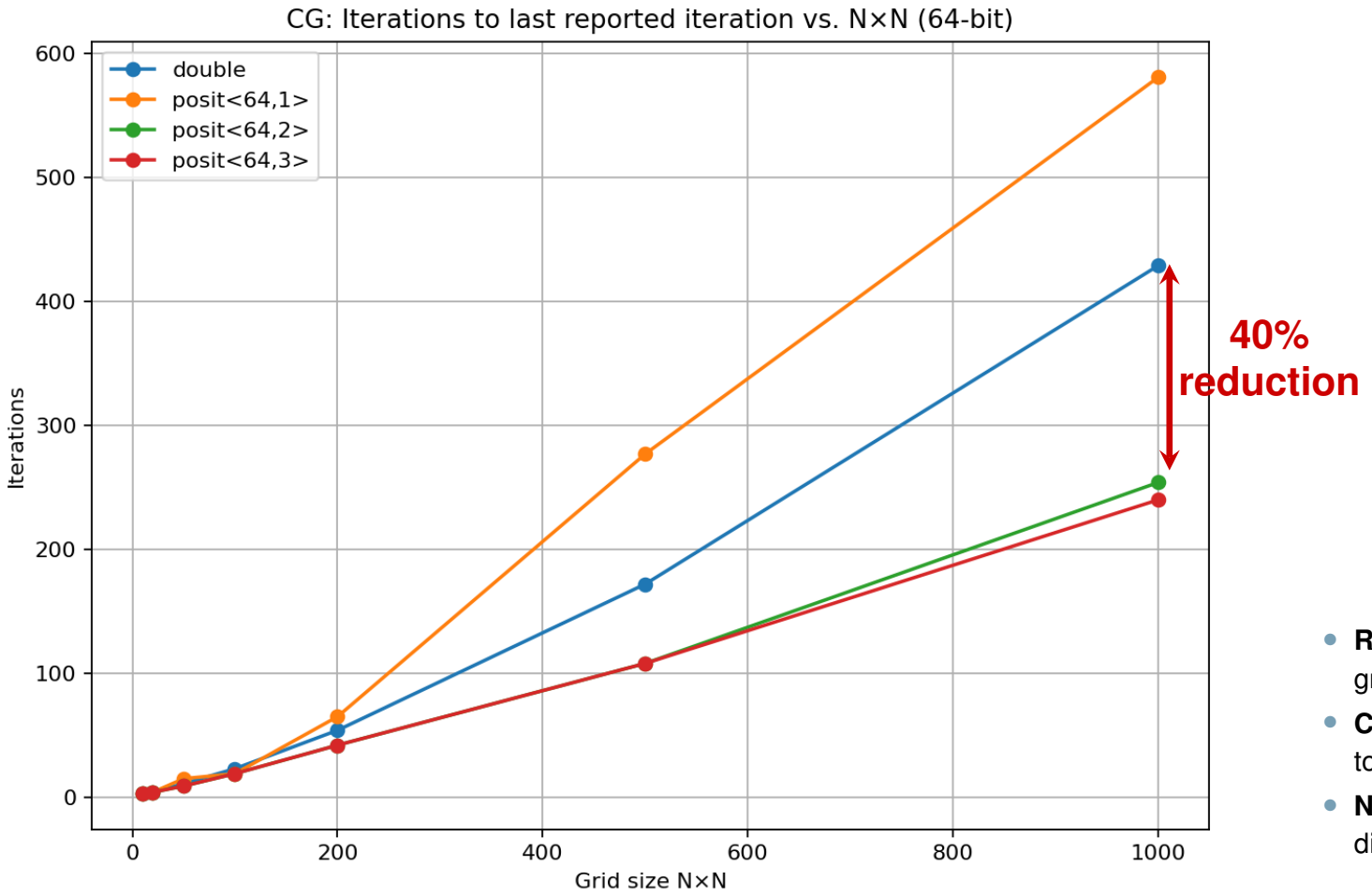
Results 2

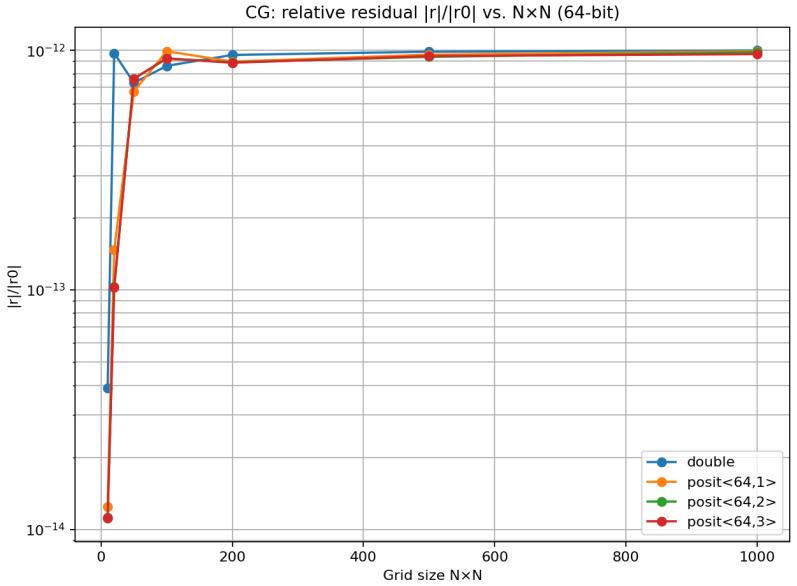
Algorithmic Convergence (32-bit)





Results 2
Algorithmic Convergence (64-bit)





- **Reduced Iterations:** 64-bit Posit achieves a **40% reduction** for large grids ($N = 1000$).
- **Consistent Accuracy:** All formats reliably reach the prescribed 10^{-12} tolerance.
- **Numerical Stability:** Tapered precision allows more consistent search directions.

Total Energy Consumption

Putting it all together

The Equation:

$$E_{total} = Iterations \times E_{iter} \tag{1}$$

Where E_{iter} is determined by the operation counts:

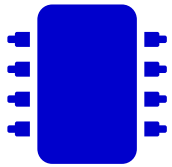
$$E_{iter} = N_{mul} \times E_{mul} + N_{add} \times E_{add} \tag{2}$$

Results relative to IEEE floating point:

32-bit Posits: **10% more total energy.**

64-bit Posits: **10-20% net energy saving.**

Conclusion (64-bit Posit)



up to +50%

Hardware Overhead

−40%

Iteration reduction

−20%

Total Energy in Arithmetic

Future: Investigate effects of Quire.

Thank you for your attention!

Questions?

Contact: *thomas.schloegl@fau.de*