

Ultrametric p-Adic Arithmetic as Energy-Proportional Precision Computing: An Energy Model for Digit-Terminable Multiply-Accumulate

Ultrametric p-Adic Arithmetic as Energy-Proportional Precision Computing: An Energy Model for Digit-Terminable Multiply-Accumulate

Abstract

Fixed-width Archimedean floating-point arithmetic pays a constant energy cost per multiply-accumulate (MAC) regardless of the precision actually required by the computation. We conjecture and formalize an alternative: p-adic (ultrametric) number representations, whose digits are hierarchically ordered by the p-adic valuation, permit computation to terminate after k digits once the required precision is met. We derive an energy model for digit-serial p-adic MAC in which energy scales as $O(k^2)$ in the number of retained digits, and compare it against reported CMOS floating-point MAC energies of 3.7 pJ at 45 nm and 0.9 pJ at 7 nm. Under a stated bit-cell energy assumption of $e_d = 0.01$ pJ per digit-product at 45 nm, we compute a break-even precision of $k^* = 19$ binary p-adic digits at 45 nm and $k^* = 9$ at 7 nm. For a quantization tolerance of $\varepsilon = 10^{-2}$, requiring $k = 7$ digits, the p-adic MAC costs 0.497 pJ versus 3.7 pJ, an 86.57% saving at 45 nm and 44.78% at 7 nm. A labeled projection for a heterogeneous 10^9 -MAC inference workload gives a 58.0% total-energy saving. We analyze conditions under which ultrametric architectures beat Archimedean ones in joules per compute, discuss off-chip communication as a labeled projection, and state the assumptions, failure modes, and falsifiers of the model.

1. Introduction

Energy is now the binding constraint on large-scale computation: data movement, not arithmetic, dominates the joule budget of modern accelerators, and every operation pays the full width of its datapath whether or not the application needs that width. Floating-point MAC units are built to a worst-case precision (e.g., 24-bit mantissas in IEEE binary32), and their energy per operation is essentially independent of the numerical difficulty of the particular operands. This is an Archimedean assumption baked into hardware: the absolute value underlying IEEE arithmetic is the usual real absolute value, in which digits carry global position weights and no digit can be meaningfully final until carries from all less significant positions have resolved.

The p-adic numbers $\mathbb{Q}_p$ offer a different topology. The p-adic absolute value $|\cdot|_p$ is ultrametric — it satisfies $|x + y|_p \leq \max(|x|_p, |y|_p)$ — and p-adic digits are ordered by increasing significance in a way that makes truncation semantically clean: a p-adic integer known to k digits is correct to within p^{-k} in $|\cdot|_p$, and addition proceeds digit-by-digit from the least significant residue upward with carries moving only toward higher digits. A computation that needs only k digits of accuracy can therefore stop after k digit-slices, and the energy it consumed is a function of k , not of a fixed machine word length.

This paper takes seriously the conjecture that this structure enables energy-proportional precision computing: hardware that performs p-adic MAC can stop after k digits, so its energy scales with the precision actually demanded, roughly as $O(k^2)$ for the convolution structure of multiplication, rather than as a constant full-width cost.

Conjecture 1 (Energy-proportional precision). There exist p-adic arithmetic architectures whose energy per MAC operation scales polynomially in the number of computed digits k , whereas an Archimedean floating-point MAC costs a fixed E_f independent of k ; consequently, for workloads whose required precision is heterogeneous and bounded, p-adic arithmetic achieves lower joules per compute.

Our contributions are: (i) a formal energy model $E_p(k)$ for digit-serial p-adic MAC, with all coefficients stated as explicit assumptions; (ii) a computed break-even analysis against reported floating-point MAC energies of 3.7 pJ at 45 nm and 0.9 pJ at 7 nm; (iii) a precision-demand analysis mapping application error tolerances ε to required digit counts $k(\varepsilon)$ and hence to energy; and (iv) a discussion of ultrametric tree-structured data movement and off-chip communication, which we treat as a labeled projection rather than a computed result. We establish sufficient conditions under which p-adic architectures beat Archimedean ones in joules per compute, and we are explicit about where the argument is conjectural.

2. Background and Related Work

The p-adic literature relevant here spans pure number theory, mathematical physics, and data modeling; we review the provided corpus in its given numbering.

[1] On the p-adic Beilinson conjecture for number fields. This work formulates a p-adic analogue of Borel's theorem, relating regulators of higher K -groups of number fields to special values of zeta-functions via syntomic regulators and p-adic L -functions, and conjectures the precise relation between the p-adic and classical situations. Its relevance to us is structural: it exemplifies how p-adic and Archimedean invariants (regulators, special values) come in parallel pairs — the same duality our energy model exploits when it pairs a p-adic digit-serial cost with a floating-point fixed cost.

[2] From Data to the p-Adic or Ultrametric Model. This paper models anomaly and change in data by embedding the data in an ultrametric space, using Correspondence Analysis to pass from a Euclidean information space to an induced ultrametric, with particular interest in sequential data. It demonstrates that ultrametric structure is practically extractable from real data, supporting our premise that hierarchical, tree-like organization — the precondition for digit-terminable computation — is natural for workloads, not merely a number-theoretic curiosity.

[3] Application of p-adic analysis methods in describing Markov processes on ultrametric spaces. The authors reduce the study of stationary Markov processes on ultrametric spaces isometrically embeddable in $\mathbb{Q}_p$ to processes on $\mathbb{Q}_p$ itself, thereby importing the machinery of p-adic mathematical physics. For us this is a template for hierarchical state evolution: a computation whose state space is ultrametric can be processed level-by-level, exactly the digit-serial discipline our MAC model assumes.

[4] On generalized Iwasawa main conjectures and p-adic Stark conjectures for Artin motives. This work introduces families of p-adic Stark regulators and an Iwasawa–Greenberg main conjecture for p -stabilized Artin representations, strengthening conjectures of Perrin-Riou and Benois. It illustrates the depth of p-adic analytic machinery available for controlling convergence and precision in p-adic settings — the mathematical counterpart of the precision-certification question (how many digits suffice?) that our energy model turns into a hardware parameter.

[5] On Phase Transitions for P-Adic Potts Model with Competing Interactions on a Cayley Tree. Reducing the description of p-adic Gibbs measures to a recursive equation, this paper proves a phase transition occurs if and only if $p = 3$ on a Cayley tree of order two. The Cayley tree is precisely the ultrametric hierarchy; the paper shows that recursive, level-by-level processes on such trees admit sharp, exactly solvable criteria — analogous to the sharp break-even criterion $k \leq k^*$ we derive for energy.

[6] p-adic Equiangular Lines and p-adic van Lint-Seidel Relative Bound. This paper introduces p-adic equiangular lines in $\mathbb{Q}_p^d$ and derives the first fundamental relation between common angle, dimension, and line count, an inequality the authors call the p-adic van Lint–Seidel relative bound. Finite frames and signal representations over $\mathbb{Q}_p$ suggest that transform-style workloads (the consumers of MAC arrays) can be formulated intrinsically in p-adic inner-product spaces, which is the representational prerequisite for our architecture proposal.

[7] The p-adic analytic subgroup theorem revisited. Revisiting the p-adic analogue of Wüstholz's analytic subgroup theorem, a cornerstone of transcendence theory, this paper systematizes how p-adic analytic hypotheses force algebraic conclusions. It supplies the rigor culture — exact, valuation-based control of analytic objects — against which our engineering-level precision bounds ($|e|_p \leq p^{-k}$) should be read as deliberately coarse but honest first approximations.

[8] A p-Adic Model of DNA Sequence and Genetic Code. Using basic properties of p-adic numbers, this work builds an ultrametric p-adic information space whose elements are nucleotides, codons, and genes, showing a 5-adic model is appropriate for DNA sequence, combined with a 2-adic distance. It is the clearest precedent in the corpus for the claim that discrete, hierarchical, symbolic-combinatorial data — the kind that dominates AI workloads — maps naturally onto small-prime p-adic digit hierarchies.

[9] QNFO: Ultrametric Engine. This protocol formalizes 20 principles for knowledge-graph navigation, paper discovery, and research-question generation under non-Archimedean distance constraints, demonstrating ultrametric organization as an operational compute discipline rather than only a mathematical structure.

[10] QNFO: ultrametric-paradigm and **[11] QNFO: FACTORING, Adelic Complexity, and the Silent-Radix Principle.** These documents develop the ultrametric paradigm and an adelic/silent-radix viewpoint in which computation over mixed places (Archimedean and non-Archimedean) is organized by radix hierarchy; they motivate, though do not quantify, the energy claims

formalized here.

[12] QNFO: The Consilience Framework. This synthesis connects valuation theory to a foundational hierarchy and autonomous research workflows, providing the cross-domain framing in which a valuation-theoretic number system is treated as an engineering substrate.

What is missing from all of these strands, and what this paper supplies, is an explicit joules-per-operation model connecting ultrametric digit structure to CMOS switching energy.

3. Methods

3.1 p-adic representation and digit-serial computation

Fix a prime p . Every $x \in \mathbb{Z}_p$ (the ring of p-adic integers) has a canonical expansion

$$x = \sum_{i=0}^{\infty} a_i p^i, \quad a_i \in \{0, 1, \dots, p-1\},$$

and the p-adic absolute value satisfies $|x|_p = p^{-v_p(x)}$, where $v_p(x)$ is the p-adic valuation. The ultrametric inequality implies that arithmetic on residues modulo p^k is self-contained:

$$(x \bmod p^k) \circ (y \bmod p^k) \equiv (x \circ y) \bmod p^k, \quad \circ \in \{+, \times\},$$

and truncation error is bounded by p^{-k} :

$$\left| x - \sum_{i=0}^{k-1} a_i p^i \right|_p \leq p^{-k}.$$

Multiplication of two k -digit truncations is a digit convolution: the product digit at position j depends on digit pairs (a_i, b_{j-i}) , so a full k -digit product requires k^2 digit-digit products, each over the finite alphabet of size p .

3.2 Energy model

We define the model inputs, each stated explicitly:

- e_d : energy of one digit-digit product-and-accumulate (a $\lceil \log_2 p \rceil \times \lceil \log_2 p \rceil$ -bit multiply plus small add) in a digit-serial p-adic array. **Stated assumption:** $e_d = 0.01$ pJ at 45 nm, i.e., 1.0×10^{-2} pJ, consistent in order of magnitude with bit-level CMOS multiply energies at that node; this is a modeling assumption, not a measurement.
- e_a : energy of one digit-position accumulation pass, **stated assumption** $e_a = 0.001$ pJ at 45 nm.
- $E_f(45) = 3.7$ pJ: energy of one full-width floating-point MAC at 45 nm (given input).
- $E_f(7) = 0.9$ pJ: energy of one full-width floating-point MAC at 7 nm (given input).
- $p = 2$ throughout the numeric analysis, so one p-adic digit carries $\log_2 p = 1$ bit of precision.

The energy of a k -digit p-adic MAC is then

$$E_p(k) = k^2 e_d + k e_a.$$

This is the $O(k^2)$ scaling of the conjecture: the quadratic term is the convolution structure of multiplication, the linear term the digit-serial accumulation. Early termination means the hardware computes only the first k digits, where k is set by the application's precision demand.

3.3 Precision demand from error tolerance

If an application tolerates relative error ε , truncation to k digits suffices when $p^{-k} \leq \varepsilon$, i.e.,

$$k(\varepsilon) = \left\lceil \frac{\ln(1/\varepsilon)}{\ln p} \right\rceil.$$

3.4 Break-even condition

The p-adic MAC beats the floating-point MAC when $E_p(k) < E_f$. With $e_a \ll e_d$ we use the leading term for the break-even digit count:

$$k^* = \left\lfloor \sqrt{\frac{E_f}{e_d}} \right\rfloor.$$

3.5 Workload-level energy

For a workload of N MACs whose required precisions $k_1, \dots, k_N$ vary, total p-adic energy is $\sum_{n=1}^N (k_n^2 e_d + k_n e_a)$ versus $N \cdot E_f$ for fixed-width floating point. We evaluate a heterogeneous workload as a labeled projection in Section 4.

4. Analysis

Every number below is derived from the inputs stated in Section 3.2: $E_f(45) = 3.7$ pJ, $E_f(7) = 0.9$ pJ, $e_d = 0.01$ pJ, $e_a = 0.001$ pJ, $p = 2$.

A. Break-even at 45 nm. With $E_f(45) = 3.7$ pJ and $e_d = 0.01$ pJ:

$$k^*(45) = \left\lfloor \sqrt{\frac{3.7}{0.01}} \right\rfloor = \lfloor \sqrt{370} \rfloor.$$

Since $19^2 = 361$ and $20^2 = 400$, we get $\sqrt{370} \approx 19.24$, so

$$k^*(45) = 19.$$

A p-adic MAC of up to 19 binary p-adic digits costs less than the 3.7 pJ floating-point MAC at 45 nm. Check at $k = 19$: $E_p(19) = 19^2 \times 0.01 + 19 \times 0.001 = 3.61 + 0.019 = 3.629$ pJ < 3.7 pJ. At $k = 20$: $E_p(20) = 4.00 + 0.020 = 4.020$ pJ > 3.7 pJ. The break-even is confirmed.

B. Break-even at 7 nm. With $E_f(7) = 0.9$ pJ:

$$k^*(7) = \left\lfloor \sqrt{\frac{0.9}{0.01}} \right\rfloor = \lfloor \sqrt{90} \rfloor.$$

Since $9^2 = 81$ and $10^2 = 100$, $\sqrt{90} \approx 9.49$, so

$$k^*(7) = 9.$$

Check: $E_p(9) = 81 \times 0.01 + 9 \times 0.001 = 0.81 + 0.009 = 0.819$ pJ < 0.9 pJ; $E_p(10) = 1.00 + 0.010 = 1.010$ pJ > 0.9 pJ.

C. Worked example at $\varepsilon = 10^{-2}$, 45 nm. From Section 3.3, for $p = 2$ and $\varepsilon = 10^{-2}$:

$$k(10^{-2}) = \left\lceil \frac{\ln(100)}{\ln 2} \right\rceil = \left\lceil \frac{4.60517}{0.693147} \right\rceil = \lceil 6.64386 \rceil = 7.$$

Then

$$E_p(7) = 7^2 \times 0.01 + 7 \times 0.001 = 49 \times 0.01 + 0.007 = 0.49 + 0.007 = 0.497 \text{ pJ}.$$

The saving per MAC against the 45 nm floating-point MAC is

$$\Delta E = 3.7 - 0.497 = 3.203 \text{ pJ},$$

a fractional saving of $3.203/3.7 = 0.8657$, i.e., 86.57%, rounded in the Abstract as

$$\frac{3.203}{3.7} = 0.8656756756756757 \approx 86.57\%.$$

At 7 nm the same $k = 7$ MAC costs 0.497 pJ versus 0.9 pJ, a saving of $0.9 - 0.497 = 0.403$ pJ, i.e., $0.403/0.9 = 0.4477777777777778 \approx 44.78\%$.

D. Energy per bit of precision. With $b = k \log_2 p = k$ bits for $p = 2$, the energy per delivered bit is

$$\frac{E_p(k)}{b} = \frac{k^2 e_d + k e_a}{k} = k e_d + e_a,$$

which is linear in k : $0.01k + 0.001$ pJ/bit. At $k = 7$: 0.071 pJ/bit. The floating-point MAC delivers 24 mantissa bits (binary32) for 3.7 pJ at 45 nm, i.e., $3.7/24 = 0.15416666666666667$ pJ/bit. The p-adic digit-serial unit at $k = 7$ delivers precision at $0.071/0.1542 = 0.4604$, i.e., about 46.0% of the floating-point per-bit cost.

E. Workload projection (labeled projection). Consider a neural-network inference layer with $N = 10^9$ MACs and a heterogeneous precision profile in which a fraction $f = 0.8$ of MACs need only $k = 7$ digits and a fraction $1 - f = 0.2$ need full $k = 24$ digits (comparable to binary32 mantissas). **Assumptions:** the e_d, e_a values of Section 3.2 hold; precision demands are as stated; no overhead for precision scheduling. Then

$$E_{\text{total}} = N [f E_p(7) + (1 - f) E_p(24)].$$

$$E_p(24) = 576 \times 0.01 + 24 \times 0.001 = 5.76 + 0.024 = 5.784 \text{ pJ. So}$$

$$E_{\text{total}} = 10^9 \times [0.8 \times 0.497 + 0.2 \times 5.784] = 10^9 \times [0.3976 + 1.1568] = 1.5544 \times 10^9 \text{ pJ} = 1.5544 \text{ mJ.}$$

The floating-point baseline is $10^9 \times 3.7 \text{ pJ} = 3.7 \text{ mJ}$. The projected saving is $(3.7 - 1.5544)/3.7 = 2.1456/3.7 = 0.580$, i.e., 58.0%, with the caveat that $E_p(24) > E_f(45)$: at 45 nm, full binary32-equivalent precision is more expensive in this p-adic model, so the saving comes entirely from the early-terminating majority. Uncertainty (computed sensitivity, same projection assumptions): at $e_d = 0.003$ pJ, $E_p(7) = 49 \times 0.003 + 0.007 = 0.154$ pJ and $E_p(24) = 576 \times 0.003 + 0.024 = 1.752$ pJ, so the per-MAC cost is $0.8 \times 0.154 + 0.2 \times 1.752 = 0.1232 + 0.3504 = 0.4736$ pJ and the saving is $(3.7 - 0.4736)/3.7 = 3.2264/3.7 = 0.8720$, i.e., 87.2%. At $e_d = 0.03$ pJ, $E_p(7) = 49 \times 0.03 + 0.007 = 1.477$ pJ and $E_p(24) = 576 \times 0.03 + 0.024 = 17.304$ pJ, so the per-MAC cost is $0.8 \times 1.477 + 0.2 \times 17.304 = 1.1816 + 3.4608 = 4.6424$ pJ, exceeding $E_f(45) = 3.7$ pJ; the saving is $(3.7 - 4.6424)/3.7 = -0.9424/3.7 = -0.2547$, i.e., a 25.5% loss. The projected saving therefore ranges over -25.5% to 87.2% across $e_d \in [0.003, 0.03]$ pJ; we report the point value only under the stated $e_d = 0.01$ pJ.

F. Off-chip communication (labeled projection). Off-chip DRAM traffic dominates accelerator energy, with per-bit transport energies commonly one to two orders of magnitude above on-chip arithmetic energies. Because p-adic digit hierarchies are trees (digits at level i depend only on levels $\leq i$), a precision-adaptive memory system could stream only the leading k digits of weights and activations, cutting off-chip bit-traffic proportionally to k rather than to full word width. As one illustrative projection with fully stated assumptions: if a workload moves $B = 100 \times 10^9$ bytes off-chip at 0.2 nJ/byte, flat communication costs $0.2 \times 10^{11} = 2 \times 10^{10}$ nJ = 20 J; if tree-structured aggregation halves the moved volume (factor $f = 0.5$, assumed), the cost is 10 J. We flag both the byte count, the per-byte energy, and the halving factor as assumptions, not measurements; no hardware prototype or trace supports them.

5. Results

All results are computed in Section 4 from the stated inputs ($E_f(45) = 3.7$ pJ, $E_f(7) = 0.9$ pJ, $e_d = 0.01$ pJ, $e_a = 0.001$ pJ, $p = 2$), except item 6, which is a labeled projection.

1. **Break-even precision at 45 nm:** $k^*(45) = 19$ binary p-adic digits; verified by $E_p(19) = 3.629$ pJ < 3.7 pJ and $E_p(20) = 4.020$ pJ > 3.7 pJ.
2. **Break-even precision at 7 nm:** $k^*(7) = 9$ digits; verified by $E_p(9) = 0.819$ pJ < 0.9 pJ and $E_p(10) = 1.010$ pJ > 0.9 pJ.

3. **Energy at $\varepsilon = 10^{-2}$ tolerance ($k = 7$):** $E_p(7) = 0.497$ pJ, saving 3.203 pJ (86.57%) versus 45 nm floating point and 0.403 pJ (44.78%) versus 7 nm floating point.
4. **Energy per precision bit:** $ke_d + e_a = 0.071$ pJ/bit at $k = 7$, versus 0.15416666666666667 pJ/bit for binary32 at 45 nm — a 46.0% lower per-bit cost (ratio $0.071/0.15416666666666667 = 0.4606$).
5. **Projected workload saving (projection, assumptions in Section 4.E):** for $N = 10^9$ MACs with 80% at $k = 7$ and 20% at $k = 24$, p-adic total 1.5544 mJ versus floating-point 3.7 mJ, a 58.0% saving at 45 nm; computed sensitivity to $e_d \in [0.003, 0.03]$ pJ spans -25.5% (a loss) to 87.2%, as derived in Section 4.E.
6. **Off-chip traffic reduction (projection):** under the stated assumptions of Section 4.F, flat communication 20 J versus tree-structured 10 J; hypothesis-level, no measured support.

The sufficient condition for p-adic advantage is: the application's precision demand $k(\varepsilon)$ satisfies $k(\varepsilon) \leq k^* = \lfloor \sqrt{E_f/e_d} \rfloor$, and the fraction of MACs meeting this condition, weighted by $E_f - E_p(k)$, is positive.

6. Discussion

Limitations of the energy model. The coefficient $e_d = 0.01$ pJ is an assumption, not a measurement or a synthesis-tool extraction. Real digit-serial arrays pay costs our model omits: digit scheduling and control, precision-adaptive sequencing logic, p-adic-to-Archimedean conversion at I/O boundaries, and the non-ideal scaling of small-integer multipliers at advanced nodes. The $O(k^2)$ convolution count is the algorithmic minimum; a real array may pipeline digits and approach $O(k)$ latency with $O(k)$ energy for streaming MAC if digit-products are reused across accumulation steps, which would only strengthen our conclusions — but a naive design could also pay $O(k^2)$ overheads we have not modeled. The comparison against 3.7 pJ and 0.9 pJ floating-point MAC energies inherits whatever methodology produced those figures; node-to-node comparison at fixed e_d is inconsistent, since e_d should also shrink at 7 nm, which would lower $k^*(7)$ below our computed 9 if e_d scaled by the same factor as E_f (e.g., $e_d \rightarrow 0.01 \times 0.9/3.7 = 0.00243$ pJ gives $k^*(7) = \lfloor \sqrt{0.9/0.00243} \rfloor = \lfloor \sqrt{370.4} \rfloor = 19$, restoring parity in break-even precision while lowering absolute energies).

Failure modes. (i) Workloads requiring uniformly high precision ($k \geq k^*$) gain nothing; the model's advantage is entirely an early-termination effect. (ii) The p-adic absolute value is not the Archimedean one: error tolerances stated in real-relative terms ($|e|/|x| \leq \varepsilon$ over $\mathbb{R}$) do not translate directly to $|\cdot|_p$ bounds without an embedding or scaling convention; our $p^{-k} \leq \varepsilon$ step assumes the quantity of interest has $|x|_p \leq 1$, which fails for quantities with negative p-adic valuation, where more digits are needed. (iii) Numerical algorithms with cancellation or division behave very differently ultrametrically; neural-network MAC is favorable precisely because it is division-free and cancellation-tolerant in the ultrametric sense. (iv) Off-chip memory systems are not precision-adaptive today; realizing digit-terminable streaming is a systems problem at least as hard as the arithmetic one.

What would falsify the claims. A silicon or RTL-level demonstration showing that a digit-serial p-adic MAC's measured energy per operation does not fall below the fixed-width floating-point MAC for any $k \leq k^*$ would falsify the core conjecture under our assumptions. Alternatively, showing that realistic AI workloads rarely tolerate $k < k^*$ (i.e., that precision demands are uniformly at or above binary32 level) would make the savings vanish even if the arithmetic model holds. A third falsifier: demonstrating that precision-scheduling overhead (control, conversion, irregular memory access) exceeds the computed per-MAC savings of 3.203 pJ at 45 nm.

Against ourselves. The strongest counterargument is that mixed-precision floating-point and block-floating-point formats already deliver energy-proportional precision with mature toolchains, and the p-adic proposal must beat not fixed-width binary32 but these adaptive baselines — a comparison we have not made quantitatively. Also, the ultrametric topology means p-adic "closeness" is divisibility-closeness, which matches some workloads (as [2] and [8] argue for data and genetic code) but is alien to others; the applicability boundary is an empirical question this paper does not settle. Finally, the corpus documents [9]–[12] motivate the paradigm but provide no quantitative energy data; our model stands or falls on the stated assumptions, not on prior art.

Open questions. What is the measured e_d for a $p = 2$ digit cell at 7 nm? Can ultrametric tree-structured addressing reduce cache-tile and DRAM row-activation energy in practice? Do p-adic frames in the sense of [6] admit fast transforms with sub- k^* precision? Under what conditions does the adelic viewpoint of [11] — using both places jointly — yield further savings?

7. Conclusion

We formalized the conjecture that p-adic, ultrametric arithmetic enables energy-proportional precision computing. Under explicit assumptions ($e_d = 0.01$ pJ, $e_a = 0.001$ pJ at 45 nm, $p = 2$), a digit-serial p-adic MAC with energy $E_p(k) = k^2 e_d + k e_a$ beats a 3.7 pJ floating-point MAC for up to $k^* = 19$ digits at 45 nm and a 0.9 pJ MAC for up to $k^* = 9$ digits at 7 nm; at a 10^{-2} error tolerance ($k = 7$), the computed saving is 86.57% at 45 nm and 44.78% at 7 nm, and a labeled projection for a heterogeneous 10^9 -MAC workload gives a 58.0% total-energy saving. The sufficient condition for p-adic advantage is an application precision demand below the break-even digit count. The result is a modeling contribution whose assumptions — especially e_d and the realizability of digit-terminable memory streaming — define the empirical agenda: measured digit-cell energies, precision profiles of real workloads, and prototype digit-serial arrays. If those confirm the model, ultrametric arithmetic offers a principled route to joules-per-compute scaling with precision, complementing the Archimedean fixed-width paradigm rather than replacing it.

References

- [1] On the p-adic Beilinson conjecture for number fields. arXiv:0707.3682v2. <https://arxiv.org/abs/0707.3682v2> [2] From Data to the p-Adic or Ultrametric Model. arXiv:0809.0492v1. <https://arxiv.org/abs/0809.0492v1> [3] Application of p -adic analysis methods in describing Markov processes on ultrametric spaces isometrically embeddable into $\mathbb{Q}_p$. arXiv:1504.03629v1. <https://arxiv.org/abs/1504.03629v1> [4] On generalized Iwasawa main conjectures and p -adic Stark conjectures for Artin motives. arXiv:2103.06864v4. <https://arxiv.org/abs/2103.06864v4> [5] On Phase Transitions for P -Adic Potts Model with Competing Interactions on a Cayley Tree. arXiv:math-ph/0512018v2. <https://arxiv.org/abs/math-ph/0512018v2> [6] p-adic Equiangular Lines and p-adic van Lint-Seidel Relative Bound. arXiv:2408.00810v3. <https://arxiv.org/abs/2408.00810v3> [7] The p -adic analytic subgroup theorem revisited. arXiv:1502.00768v1. <https://arxiv.org/abs/1502.00768v1> [8] A p-Adic Model of DNA Sequence and Genetic Code. arXiv:q-bio/0607018v1. <https://arxiv.org/abs/q-bio/0607018v1> [9] <https://papers.qnfo.org/papers/zbw-majorana-tqc-p6-ultrametric-engine/>. QNFO: Ultrametric Engine: Deploying a 20-Principle p-Adic Discovery Worker. [10] <https://papers.qnfo.org/papers/ultrametric-intelligence/>. QNFO: ultrametric-paradigm. [11] QNFO: FACTORING, Adelic Complexity, and the Silent-Radix Principle [12] <https://papers.qnfo.org/papers/consilience-framework/>. QNFO: The Consilience Framework: From Valuation Theory to the Void — A Cross-Domain Synthesis.