The log-arithmetic gates: weight sharing buys compute, not storage

W. Schulz

Aug 26, 2026

Abstract

Eight prior-art reviews and six pre-registered experiments on log-domain neurons and the hardware proposed for them. None of the 16 ideas was new; LNS loses at inference precisions and analog log-domain error compounds with fan-in. The output is six design rules — chiefly, weight sharing pays only when the freed compute is spent.

Over 25 and 26 August 2026 a run of conversation transcripts proposed, one after another, a new kind of neuron and the hardware to go with it. It began with a logarithmic neuron, y = exp(w$^\top$ ln x + b) = e^b $\prod$ x$_i^{w}_i$}, and a gate mixing that multiplicative path with an ordinary additive one, to be run on logarithmic-number-system or analog log-domain circuits. It went on to networks used as neurons, modular "super-neurons" fired sparsely on a photonic coprocessor, parallel analog width in place of depth, multiplexed photonics, light recirculated through one circuit, a glass library of optical operators, and a volume of glass to store them in.

The project gated all of it — eight gates in two days — and ran six pre-registered experiments alongside. The research directory's own one-line verdict:

> Not novel anywhere (0 of 16 threads), sound where it matters most, and attached to three hardware arguments that are each quantitatively wrong.

That tally was written after the first three gates; none of the later five recorded a novel thread either. The useful output was not an architecture. It was a set of decision rules, a hardware screen that became the project's Substrate Gate, and one synthesis: weight sharing buys compute; it does not save storage.

The eight gates

Gate	Proposal	Owners found	Outcome
1	Log neuron; gated additive/multiplicative hybrid; LNS hardware	Product units [1]; NALU [2]; the Domain Mixed Unit [3]; LNS from Mitchell [4] and Swartzlander & Alexopoulos [5]; LNS-Madam [6]; Cobb–Douglas [7]	0 of 6 novel. Algebra correct. The log-space branch is the one the field tried and left [8].

Gate	Proposal	Owners found	Outcome
2	Networks as neurons; hypernetwork weights; learned reusable primitives	Network In Network [9]; HyperNetworks [10]; DreamCoder [11]	0 of 7 novel. Missing variable: bytes moved per useful function.
3	Sparse modular super-neurons on photonics	Modular-architecture collapse [12]	Sparsity and weight residency are the same prize: a 333× saving on a GPU is a 333× penalty on a resident optical mesh.
4	Trade depth for parallel analog width; learn the topology	DARTS [13]; depth-separation theorems [14][15]	0 novel. Width-for-depth is provably exponentially expensive in general; conversions per MAC do not improve with richer units.
5	Space × wavelength × mode × polarization multiplexing	Two 2026 photonic papers, both checked and real [16][17]	Multiplexing multiplies throughput, not efficiency. A fixed-gate analog sum of 100 linear experts collapses to one matrix (residual 4.64×10^{-14}).
6	Optical recirculation as physical weight sharing	Recirculating optical loops (an active field)	Pass budget of order tens, set by precision, not loss.
7	A glass library supplying its own nonlinearity	Diffractive networks [18]	An energy edge at 8 bits paid back in bandwidth; withdrawn at 4 bits.
8	Does a 3-D volume buy capacity?	volume storage, after van Heerden [19]; the M/# law [20]	Geometry says 10^3–10^4×; dynamic range says about 40 operators. The smaller binds.

Two gates were amended in place on 26 August. Gate 7's surviving "~3.6× energy edge in a narrow resonator-Q window" was priced against an 8-bit optical-electrical-optical round trip; at the 4-bit precision the experiments later measured, the conversion it competes with is about 5.4× cheaper and the edge is gone. Gate 8 had claimed that its ~40-operator bound and an experiment converged independently; the amendment notes both derive from the same 4-bit floor, so it is "one assumption propagated, not two independent measurements." What survives is "tens, not thousands", which is robust to a bit either way.

The three hardware arguments, computed

Logarithmic arithmetic inverts with precision. LNS turns a multiply into an add and an add into a log-sum-exp with a table lookup. A multiply-accumulate is both. Priced on Horowitz's 45 nm per-operation energies [21], with the lookup costed optimistically at a tenth of a small SRAM read, the project's calculation has LNS winning only at fp32 (0.70×) and losing 1.13× at fp16 and 2.57× at int8 — the precisions inference runs at. And decode is memory-bound: on the project's calibrated H100 constants, moving one bf16 weight costs 40 pJ against 1.42 pJ for the two FLOPs it enables, so making arithmetic entirely free saves 3.4%. The industry did adopt the log idea, in the place it pays: a floating-point exponent is $\log_2|x|$, and the microscaling formats share one power-of-two scale across a block [22][23]. LNS-Madam, the strongest LNS-for-deep-learning result, multiplies in the log domain and converts to integer to accumulate [6].

Analog log-domain has the wrong exchange rate. Translinear circuits really do give log and exp from transistor physics [24]. But in log space an additive error becomes a multiplicative one, and the unit's product over its fan-in compounds it. At 5.7-bit ENOB — the ceiling of the fabricated analog chip in the project's reference set — a log-domain unit with fan-in 4 carries 13× the RMS error of a linear-domain unit at equal ENOB, rising to 74× at fan-in 64. (The first version of this computation overstated the penalty by about $\sqrt{12}$ in the exponent; it was corrected in the file, with the conclusion unchanged.)

Conditional computation and photonic residency are the same prize. Firing 3 of 1,000 modules is a 333× saving on a GPU, where weights are moved. On a weight-resident optical mesh the movement was already amortised to nearly zero, so sparsity saves nothing, utilisation falls to 0.3%, and hold power per useful operation rises 333×. *Sparsity is a discount on weights you move and a penalty on weights you hold.*

The pre-registered experiments, and their misses

exp01 (90 runs, 10 seeds, parameters matched within 7% at about 1,200) pitted the transcript's gated log unit against an MLP and a linear-space bilinear unit. Median normalized extrapolation error, in this project's run:

Target	MLP	Log unit	Linear-space product
$x_1^2 x_2^3/\sqrt{x_3}$	$1.059 \cdot 0\%$ success	$0.00194 \cdot 100\%$	$0.0440 \cdot 0\%$
$(x_1+x_2)^2 x_3/\sqrt{x_4}$	$0.5082 \cdot 0\%$	$0.00598 \cdot 70\%$	$0.0791 \cdot 10\%$
$x_1 x_2$, inputs crossing zero	$0.1141 \cdot 0\%$	$0.1843 \cdot 0\%$	$0.0425 \cdot 0\%$

The log unit beat the MLP by 545× on the pure power law. P1 and P2 hit; P3 missed both thresholds though the direction held — the log unit was the worst arm on signed inputs. P4, the registered crux, missed cleanly. It predicted the log unit would fail to train even where it could represent the target, replicating Madsen & Johansen [8]; it trained at 100% and 70%. By the project's own kill condition, that vindicated the proposal in its home regime — scoped by an easier protocol than the NALM benchmark's [25]. The results file then attributed the miss to the transcript's *untied* weights and closed that residue by reading the Domain Mixed Unit as untied too, with a caveat that the reading came from automated PDF extraction. Checked for this chapter, it does not hold: the DMU shares one operand-selection matrix across both paths and separates only the gate [3]. The DMU still owns the gated log/linear mix, so novelty is unchanged; but whether untying explains P4's miss is open, not closed.

exp02 contaminated the power-law target with a non-closed-form component and predicted the advantage would vanish by a quarter of its variance. P5 hit; P6 (monotone decay) missed, a five-seed median excursion; P7 missed in the opposite direction — the log unit was still 3.8× better at a measured alien variance share of 0.68. That made two deflationary predictions in a row that failed, which the files log as a bias to watch.

exp03 asked, at fixed bytes, fixed compute (six layer applications) and fixed width, whether to spend memory on distinct layers or on precision. Four of four predictions hit. Both budgets selected 4 bits; six distinct 4-bit blocks came within 0.0165 nats of the 16-bit reference at a quarter of the block memory; six 2-bit blocks were the worst arm overall, worse than a single 12-bit block. exp04 gave each layer a learned combination of a shared basis — a superset of hard assignment — and it trained 5.1–5.4σ worse at k = 3 and 6 (P12 missed; P13 hit for a reason the file records as wrong).

Three registered follow-ups have raw outputs in the directory but no written scoring; they are scored here, by arithmetic on the per-seed results, against their registered thresholds. exp04b re-ran the basis arms with one-hot initialisation: 1.7133 against hard 1.7347 at k = 3, and 1.6333 against 1.6512 at k = 6 — within 2σ, so P19 hits and exp04's 5σ deficit was an initialisation artifact, not a property of the parameterisation. exp03b replicated exp03's budget ordering on out-of-domain Gutenberg prose (P20 and P21 hit; 4-bit cost 0.015 nats). exp05, a saturation curve over the number of distinct blocks at twelve applications, missed P15 narrowly and missed P16 for fixed assignment — k = 3 captured 57% of the k = 1→12 gain against a registered 60% — while P17 and P18 hit: a learned router never beat fixed assignment, and at k = 12 it used only 9, 8 and 7 of the 12 blocks. exp06, a signed-log pathway inside a transformer, is registered with no outputs.

Six rules, and the Substrate Gate

The strand closed with six rules, each tied to a measurement: (1) quantize until precision binds — about 4 bits, with 2 bits a cliff; (2) preserve distinct transformations while you can; (3) use sharing only when you can spend the compute it frees; (4) don't infer optimisation from expressivity; (5) don't multiply gains that consume the same resource; (6) test native arithmetic structure before inventing hardware. Rule 4's best exhibit was exp04, and the rule recorded exp04b as a pending qualifier; on the outputs above, that exhibit does not stand unqualified. The rule's other support — two class-inclusion predictions that missed — remains.

Rule 3 is the synthesis. Recurrent depth beat a matched feedforward network in an earlier experiment in this repository at 0.13× the parameters; exp03 found one shared block the worst use of bytes at fixed compute; ALBERT's all-shared configuration loses 1.5 points of average score against its unshared counterpart at the same depth [26]. One statement covers all three: sharing converts storage into the *option* to apply the same weights more times, and pays only if that compute is spent. Named prior systems built on exactly that — the Universal Transformer and its mixture-of-experts successors [27][28][29].

`docs/SUBSTRATE-GATE.md` distils the hardware half into four divisions, any of which ends an investigation: does the advantage shrink as precision drops; is compute even 5% of the energy; does the proposal's win double-count with the substrate's; and what is the boundary tax in conversions and chained precision? Its meta-check is the strand's real lesson: the hardware checks all fired correctly while both early machine-learning deflations missed. *A substrate can be dead while the algorithm running on it is alive.* A 26 August annotation adds that the conversion tax is itself precision-dependent — at 4 bits a conversion costs about 0.7 int8 MACs, not 111 — so the check must be run at the precision the workload needs.

Sources

1. R. Durbin and D. E. Rumelhart, “Product Units: A Computationally Powerful and Biologically Plausible Extension to Backpropagation Networks,” *Neural Computation* 1(1):133–142, 1989. DOI: 10.1162/neco.1989.1.1.133
2. A. Trask, F. Hill, S. Reed, J. Rae, C. Dyer and P. Blunsom, “Neural Arithmetic Logic Units,” 2018. arXiv:1808.00508
3. P. Curry, “The Domain Mixed Unit: A New Neural Arithmetic Layer,” 2025. arXiv:2509.08180
4. J. N. Mitchell, “Computer Multiplication and Division Using Binary Logarithms,” *IRE Transactions on Electronic Computers* EC-11(4):512–517, 1962. https://www.semanticscholar.org/paper/f3ec25a90ac8d09
5. E. E. Swartzlander Jr. and A. G. Alexopoulos, “The Sign/Logarithm Number System,” *IEEE Transactions on Computers* C-24(12):1238–1242, 1975.
6. J. Zhao, S. Dai, R. Venkatesan, B. Zimmer, M. Ali, M.-Y. Liu, B. Khailany, W. J. Dally and A. Anandkumar, “LNS-Madam: Low-Precision Training in Logarithmic Number System Using Multiplicative Weight Update,” *IEEE Transactions on Computers* 71(12), 2022. arXiv:2106.13914
7. C. W. Cobb and P. H. Douglas, “A Theory of Production,” *American Economic Review* 18(1), Supplement, 139–165, 1928.
8. A. Madsen and A. R. Johansen, “Neural Arithmetic Units,” ICLR 2020. arXiv:2001.05016
9. M. Lin, Q. Chen and S. Yan, “Network In Network,” ICLR 2014. arXiv:1312.4400
10. D. Ha, A. M. Dai and Q. V. Le, “HyperNetworks,” ICLR 2017. arXiv:1609.09106
11. K. Ellis, C. Wong, M. Nye, M. Sablé-Meyer, L. Morales, L. Hewitt, L. Cary, A. Solar-Lezama and J. B. Tenenbaum, “DreamCoder: Bootstrapping Inductive Program Synthesis with Wake-Sleep Library Learning,” PLDI 2021. DOI: 10.1145/3453483.3454080
12. S. Mittal, Y. Bengio and G. Lajoie, “Is a Modular Architecture Enough?,” NeurIPS 2022. arXiv:2206.02713
13. H. Liu, K. Simonyan and Y. Yang, “DARTS: Differentiable Architecture Search,” ICLR 2019. arXiv:1806.09055
14. M. Telgarsky, “Benefits of depth in neural networks,” COLT 2016. arXiv:1602.04485
15. R. Eldan and O. Shamir, “The Power of Depth for Feedforward Neural Networks,” COLT 2016. arXiv:1512.03965
16. W. Liu, Z. Zhang, H. Chen et al., “Photonic Mixture-of-Experts for scalable multi-task on-chip optical neural networks,” *Nature Communications* 17, 7145, 2026. DOI: 10.1038/s41467-026-73983-4
17. Q. Yao, Y. Shi, D. Dai et al., “192-dimensional wavelength-/mode-/polarization-multiplexed photonic architecture for highly parallel optical computing,” *National Science Review* 13(19), nwag350, 2026. DOI: 10.1093/nsr/nwag350
18. X. Lin, Y. Rivenson, N. T. Yardimci, M. Veli, Y. Luo, M. Jarrahi and A. Ozcan, “All-optical machine learning using diffractive deep neural networks,” *Science*, 2018. DOI: 10.1126/science.aat8084
19. P. J. van Heerden, “Theory of Optical Information Storage in Solids,” *Applied Optics* 2(4):393–400, 1963. DOI: 10.1364/AO.2.000393
20. F. H. Mok, G. W. Burr and D. Psaltis, “System metric for holographic memory systems,” *Optics Letters* 21(12):896–898, 1996. https://authors.library.caltech.edu/records/rx8az-emr54
21. M. Horowitz, “Computing’s Energy Problem (and what we can do about it),” ISSCC 2014 Digest of Technical Papers, pp. 10–14, 2014.
22. Open Compute Project, *OCP Microscaling Formats (MX) Specification*, Version 1.0, 2023. https://www.opencompute.org/documents/ocp-microscaling-formats-mx-v1-0-spec-final-pdf
23. E. Alvarez et al., “Introducing NVFP4 for Efficient and Accurate Low-Precision Inference,” NVIDIA Technical Blog, 2025. https://developer.nvidia.com/blog/introducing-nvfp4-for-efficient-and-accurate-low-precision-inference

24. B. Gilbert, "Translinear Circuits: A Proposed Classification," *Electronics Letters* 11(1):14, 1975. https://theiet.org/media/11288/translinear-circuits-a-proposed-classification.pdf
25. B. Mistry, K. Farrahi and J. Hare, "A Primer for Neural Arithmetic Logic Modules," *Journal of Machine Learning Research* 23(185):1–58, 2022. https://jmlr.org/papers/v23/21-0211.html
26. Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma and R. Soricut, "ALBERT: A Lite BERT for Self-supervised Learning of Language Representations," ICLR 2020. arXiv:1909.11942
27. M. Dehghani, S. Gouws, O. Vinyals, J. Uszkoreit and Ł. Kaiser, "Universal Transformers," ICLR 2019. arXiv:1807.03819
28. S. Tan, Y. Shen, Z. Chen, A. Courville and C. Gan, "Sparse Universal Transformer," EMNLP 2023. DOI: 10.18653/v1/2023.emnlp-main.12
29. R. Csordás, K. Irie, J. Schmidhuber, C. Potts and C. D. Manning, "MoEUT: Mixture-of-Experts Universal Transformers," NeurIPS 2024. arXiv:2405.16039

Provenance

Drawn from `log-arithmetic/README.md`, `GATE.md`, `GATE2.md`, `GATE3.md`, `GATE4.md`, `GATE5-photonic-parallelism.md`, `GATE6-optical-recirculation.md`, `GATE7-glass-library-nonlinearity.md`, `GATE8-volumetric.md`, `ENERGY.md`, `PREREGISTRATION.md`, `PREREG-exp03.md` through `PREREG-exp06.md`, `RESULTS-exp01.md` through `RESULTS-exp04.md`, `DECISION-log-arithmetic-rules.md`, `INSIGHT-sharing-buys-compute.md`, `NOTE-shared-vs-distinct-critique.md`, `energy_output.txt`, `photonic_output.txt`, `exp03_scored.txt`, `exp04_scored.txt`, `exp03b_results.jsonl`, `exp04_results.jsonl` and `exp05_results.jsonl` (all under `log-arithmetic/`), and `docs/SUBSTRATE-GATE.md`, in the Quantum-Intelligence research repository.

Published by NeuroGenesys (AW3 Technology, Inc.). Canonical version: https://neurogenesys.si/research/log-arithmetic-gates