CERTIFIED NONLINEAR COMPUTING FOR SCALED FP8 DATAPATHS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large Language Models (LLMs) demand substantial computation for inference, motivating the adoption of efficient FP8 matrix engines. However, nonlinear operations such as SwiGLU, RMSNorm, and softmax-based attention still commonly rely on wider intermediate arithmetic before FP8 quantization. Premature rounding of these intermediates can alter the final output codes after downstream composition and shared-scale selection. In this paper, we propose a sufficient-state framework for efficient nonlinear computation in scaled-FP8 pipelines. The framework constructs intermediate states that preserve the information needed to determine final element codes and shared scale bytes. We instantiate this principle through directed states for SwiGLU, exact boundary predicates over whole-row statistics for RMSNorm, and interval aggregates with adaptive refinement for softmax attention. These constructions resolve final quantization decisions through consumer composition and shared reductions. For SwiGLU, mathematical verification and exhaustive finite-state replay establish exact outputs under the declared numerical contract; for attention, interval bounds certify output cells when they are uniquely determined. Based on this framework, we design and implement complete hardware endpoints for SwiGLU, RMSNorm, and softmax attention. Hardware experiments in Nangate45 demonstrate that the SwiGLU semantic ROM reduces area by 54–61% and gate energy by 41–53% relative to polynomial controls. Under matched complete-row configurations, RMSNorm reduces area by 4.1% and gate energy by 23.0%, while attention achieves reductions of 58.8% and 79.6%, respectively, relative to their corresponding controls. Joint replacement experiments across three language models on GSM8K, HumanEval, and GPQA-Diamond demonstrate task performance comparable to native execution.

1 INTRODUCTION

Large Language Models (LLMs) increasingly use FP8 matrix engines to reduce inference cost. Nonlinear operations such as SwiGLU, RMSNorm, and softmax-based attention still commonly use wider intermediate arithmetic before FP8 quantization. Reducing this arithmetic and storage overhead requires compact representations that preserve the information needed by the final quantizer.

Prior work reduces nonlinear cost through approximation. Softermax combines base replacement, low-precision arithmetic, and online normalization (Stevens et al., 2021). I-BERT approximates GELU, softmax, and layer normalization with integer arithmetic (Kim et al., 2021). NN-LUT converts neural approximators into lookup tables (Yu et al., 2022), while NLI optimizes non-uniform interpolation (Yu et al., 2026). For correct rounding, RLIBM-ALL uses rounding intervals and round-to-odd to support multiple precisions and rounding modes with one polynomial approximation (Lim & Nagarakatte, 2021).

Scaled FP8 requires preserving output decisions through consumer composition and shared-scale selection. Correct unary rounding alone does not guarantee correct quantization after multiplication. The shared scale may also remain unknown when an intermediate is stored. Kernel fusion reduces data movement and conversions (Zhang et al., 2026; NVIDIA, 2026a), and Transformer Engine supports MXFP8 activation outputs (NVIDIA, 2026b). These pipelines still require an intermediate representation that preserves the final element codes and scale bytes.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

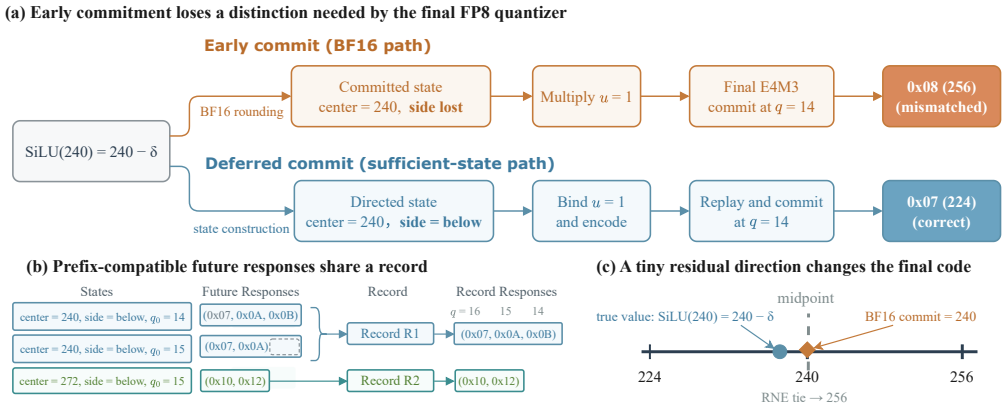

Figure 1: Why commitment should be deferred. (a) Early BF16 commitment loses the side of the $\mathrm{SiLU}(240)$ midpoint, changing the final E4M3 decision; retaining the direction recovers the mathematical result. (b) Prefix-compatible future responses can share a record. (c) The retained direction resolves the final ties-to-even decision. Appendix A gives an exact legal witness.

What must an intermediate state retain to recover the final scaled-FP8 outputs? We call a state *sufficient* if it preserves every distinction that can change an element code or scale byte under any legal remaining context. In SwiGLU, multiplier binding changes the local scale proposal and future scale queries. RMSNorm and attention additionally require shared row statistics. Our goal is to construct such states with efficient generation, storage, and decoding.

Figure 1 shows the lost distinction. $\mathrm{SiLU}(240)$ is strictly below 240, but BF16 rounds it to 240. At output scale 2^{14}, adjacent E4M3 reconstructions are 224 and 256. Their midpoint is 240. The mathematical value maps to `0x07`; the rounded value maps to `0x08` under ties-to-even. Retaining the midpoint side recovers the correct decision. Appendix A constructs a legal block selecting this scale.

A separate activation replay quantizes SiLU outputs to group-32 scaled FP8 before multiplier binding. Relative to the mathematical reference, this changes 26.78% of final element codes and 3.54% of block scales. This intervention tests early FP8 commitment, distinct from the BF16 witness in Figure 1. Section 5 gives the full ablation.

We instantiate the framework for three operators. SwiGLU uses a directed unary state and encodes the bound product before scale selection. Decoding after scale reduction recovers exact codes and scale bytes, established by mathematical verification and exhaustive finite-state replay under the declared contract. RMSNorm uses whole-row statistics and exact boundary predicates, avoiding a rounded reciprocal square root. Softmax attention uses interval aggregates to certify uniquely determined output cells and refines ambiguous cases. Each construction retains decision-relevant information until the remaining context permits output commitment.

We implement complete hardware endpoints, including reduction and storage, and compare each against controls with matched input/output boundaries. SwiGLU's coefficient and semantic-ROM producers share a certified interface. In Nangate45, the ROM reduces area by 54–61% and gate energy by 41–53% relative to polynomial controls. Under the measured complete-row configurations, RMSNorm reduces area by 4.1% and gate energy by 23.0% against its matched control. Attention achieves corresponding reductions of 58.8% and 79.6%.

Across DeepSeek-V4-Flash-0731, Qwen3.5-9B, and Qwen3.8-27B, we evaluate joint operator replacement using complete test sets from GSM8K, HumanEval, and GPQA-Diamond. With the same task-specific scorers for native and replacement responses, the results show comparable performance across mathematical reasoning, code generation, and scientific questions.

Our main contributions are:

- **Sufficient-state framework.** We construct states for final scaled-FP8 decisions: certified compact records for SwiGLU, exact boundary predicates for RMSNorm, and interval certification with adaptive refinement for attention. These constructions preserve output decisions through consumer composition and shared-context resolution.
- **Complete hardware endpoints.** All three implementations reduce area and gate energy against matched controls. SwiGLU producer and transport comparisons isolate implementation costs under a shared numerical contract.
- **Joint model evaluation.** Full-test evaluations across three language models and three benchmarks show that joint replacement maintains task performance comparable to native execution.

2 Background and Problem Definition

Operator outputs. We consider three nonlinear computations followed by scaled-FP8 quantization:

$$\begin{aligned} \text{SwiGLU:} \quad & y_i = \mathrm{SiLU}(g_i)u_i, \\ \text{RMSNorm:} \quad & y_i = \frac{x_i w_i}{\sqrt{T}}, \qquad T = \frac{1}{N}\sum_j x_j^2 + \epsilon, \\ \text{Attention:} \quad & \mathbf{y} = \mathrm{Softmax}(\mathbf{z})V. \end{aligned}$$

The target is the quantized output after multiplier composition in SwiGLU, whole-row normalization in RMSNorm, or value aggregation in attention.

Scaled-FP8 representation. A group shares a scale 2^q, and code c_i reconstructs $\hat{y}_i = 2^q \mathrm{decode}_F(c_i)$ in element format F. The observable includes the element codes and shared scale byte. Our SwiGLU endpoint uses E4M3FN elements (E4M3 below), groups of 32, and E8M0 scales, following the MX representation (Open Compute Project, 2023). Scale selection, rounding, and exceptional-value behavior are specified separately by the endpoint contract.

Known and unresolved context. Consumer operands, such as SwiGLU multipliers, RMSNorm weights, and attention values, may be bound before shared scales or statistics are resolved. Binding conditions the retained representation on known consumer operands, independent of their physical arrival order. The retained state must recover the output for every legal completion of the remaining context (Section 3.1).

SwiGLU numerical contract. For the SwiGLU endpoint, the no-overrange policy assigns each finite nonzero target a local scale proposal:

$$\widetilde{q}_0(y) = \left\lceil \log_2 \frac{|y|}{448} \right\rceil, \qquad q_0(y) = \mathrm{clamp}_{[-127,127]}(\widetilde{q}_0(y)). \tag{1}$$

Here, 448 is the maximum finite E4M3 magnitude; proposals precede exponent clipping, and excess magnitude at $q = 127$ saturates.

The endpoint forms proposals from

$$\mathcal{V} = \{i : \ g_i, u_i \text{ are finite and nonzero}, \ g_i > -208\}.$$

Inputs with $g_i \le -208$ are certified to produce zero element codes at every legal output scale. The contract excludes these tails from scale reduction. This exclusion specifies scale-byte behavior as well as element codes.

The shared exponent and active element codes are

$$q = \begin{cases} \max_{i\in\mathcal{V}} q_0(y_i), & \mathcal{V} \neq \varnothing, \\ 0, & \mathcal{V} = \varnothing, \end{cases} \qquad c_i = \mathrm{RNE}_F(y_i 2^{-q}) \quad (i \in \mathcal{V}). \tag{2}$$

The stored scale byte is $q+127$. Active lanes contribute proposals even if their final codes underflow to zero. Table 1 specifies ties-to-even rounding, saturation, signed zeros, and NaN handling.

Table 1: Numerical contract for the SwiGLU endpoint. MX specifies the representation; the additional scale-selection, rounding, and exceptional-value policies define the endpoint behavior.

Contract item	SwiGLU rule
Inputs	Independent gate and up E4M3 elements with E8M0 exponents in $[-127, 127]$; group size 32.
Scale	Eq. 2; stored scale byte $q+127$. The no-overrange proposal is formed before exponent clipping; excess magnitude at $q = 127$ saturates.
Zero / tail	Zero operand or $g \leq -208$: signed-zero code and no proposal; sign is the XOR of the operand signs.
Active underflow	Active finite lanes retain their proposals even when the final element codes underflow to zero.
NaN	A NaN in either operand takes priority, produces canonical `0x7f`, and contributes no finite proposal.
Empty valid set	$q = 0$, with scale byte `0x7f`.
Minimum scale	A nonempty $\mathcal{V}$ may select $q = -127$, with scale byte `0x00`, even if all final element codes underflow to zero.
Element rounding	RNE with ties-to-even, signed zero, and finite saturation at magnitude 448.
Observable	All 32 element bytes and the shared scale byte.

Related approaches. Function approximation and unary correct rounding provide tools for state generation, as illustrated by NLI and RLIBM-ALL (Yu et al., 2026; Lim & Nagarakatte, 2021). FlashAttention uses tiled online aggregation to reduce attention IO (Dao et al., 2022). Our target is to certify the element codes and scale bytes after composition and shared-context resolution. For reductions, this requires connecting aggregate states to final quantization boundaries.

3 Sufficient-State Constructions for Final Quantization

3.1 State Sufficiency

Let S be the reachable intermediate states and $C(s)$ the legal remaining contexts for state s. The required observable is $O(s, c)$. An encoding E is sufficient if a decoder D satisfies

$$D(E(s), c) = O(s, c), \qquad s \in S,\ c \in C(s). \tag{3}$$

The interface specifies whether the observable includes element codes, the shared scale byte, or both. Construction and encoding determine the cost of preserving it.

Binding operands changes the remaining context. For SwiGLU, binding u produces a local proposal $q_0(s)$; the stored product must answer queries $C(s) = \{q_0(s), \ldots, 127\}$. RMSNorm and attention additionally retain shared reduction information.

3.2 SwiGLU: Directed States and Compact Records

Figure 2 shows the SwiGLU path: decode inputs, generate a directed unary token, and bind the multiplier. Binding produces a compact record and local scale proposal; records are buffered during proposal reduction. Replay then emits the element codes and shared scale byte.

Canonical inputs. Different E4M3/E8M0 pairs can represent the same gate value. We decode each finite nonzero gate exactly as

$$|g| = M_g 2^{L_{\text{in}}-3}, \qquad M_g \in \{8, \ldots, 15\},$$

with separate sign. Subnormal renormalization maps aliases to the same semantic input; zero, tails, and NaNs follow the contract.

Directed unary state. For $h = |\operatorname{SiLU}(g)|$, the G11 token retains

$$v = A2^{L-11}, \qquad 2048 \leq A \leq 4095, \qquad d = \mathbf{1}[h > v].$$

G11 denotes eleven fractional bits; a rounded significand of 4096 is renormalized. Including sign and kind, the token occupies 26 bits.

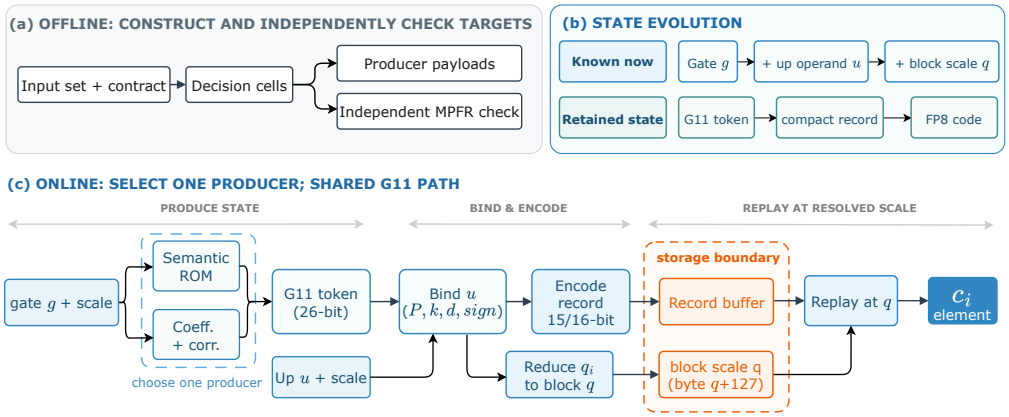

Figure 2: Certified SwiGLU construction and shared dataflow. (a) Offline target construction and independent MPFR certification. (b) Evolution of retained state as downstream context becomes available. (c) Shared online path from either G11 producer through binding, encoding, scale reduction, and replay.

Token generation. A semantic ROM and corrected coefficient producer implement $T_{\mathrm{ROM}}(g) = T_{\mathrm{coeff}}(g) = T^*(g)$ over the finite semantic gate domain, verified by exhaustive mapping. Both share downstream binding, encoding, and replay. Appendix C gives the generation and correction rules.

Multiplier binding. Decode the up operand as $|u| = M_u 2^{L_u - 3}$, with $M_u \in \{8, \ldots, 15\}$. Binding forms

$$P = AM_u, \qquad k = L + L_u - 14, \qquad v_{\mathrm{product}} = P2^k. \tag{4}$$

The 12×4-bit product is exact, with sign given by operand-sign XOR. The state (P, k, d, sign) yields the record and certified local proposal before block-scale reduction.

Compact storage. The endpoint stores the bound product in compact16. It forms

$$\widetilde{v} = (4P + 2d - 1)2^{k-2}, \tag{5}$$

then truncates and jams discarded nonzero bits into the retained least significant bit. This encodes the directional distinction without a separate direction field. Certified-cell checks and exhaustive record replay establish that this surrogate encoding preserves output decisions. Directed15 provides an alternative with an explicit direction bit. Appendix C specifies both formats.

Scale reduction and replay. Proposal reduction follows Eq. 2. Replay decodes each record at scale q, preserving midpoint direction, and emits scale byte $q + 127$. Underflow, saturation, signed zeros, and NaNs follow Table 1.

Information width and encoding cost. With a common context set, N distinct response classes require $\lceil \log_2 N \rceil$ bits when encoding cost is unrestricted. SwiGLU's state-dependent suffix queries instead permit prefix-compatible responses to share labels. Appendix B proves that the minimum label count equals the number of maximal response strings: 8,383, requiring 14 bits. The decoder receives the label and query scale under the interface defined in Appendix B. Directed15 and compact16 use wider records to support practical arithmetic encoding and decoding.

Certification. Center and direction suffice on the certified reachable domain. Correctness is established in three steps. First, output-code and proposal boundaries define constraints on $h = |\,\mathrm{SiLU}(g)|$:

$$h \in \mathcal{I}_{g,M_u} = \bigcap_{\gamma \in \Gamma(g,M_u)} \mathcal{I}_\gamma. \tag{6}$$

Outward-rounded MPFR comparisons (Fousse et al., 2007) and analytic tail bounds verify these constraints, including boundary ownership under Table 1. Second, exhaustive mapping checks verify that each producer emits the certified token. Third, record replay verifies that encoding and decoding preserve every certified suffix response. These checks connect the mathematical output to the implemented representation.

Proposition 1 (Certified SwiGLU composition)**.** *For either verified producer p, every active finite (g, u) in the domain of Table 1, and every legal suffix scale q,*

$$D(\mathrm{Encode}(\mathrm{Bind}(T_p(g), u)), q) = \mathrm{RNE}_F\big(\mathrm{SiLU}(g)u2^{-q}\big)\,.$$

Exact proposal reduction preserves the shared scale byte. Zero, tail, and NaN branches follow Table 1.

Appendix C gives the proof and executable checks.

3.3 RMSNorm: Exact Boundary Decisions

RMSNorm decisions depend on the input–weight products and the whole-row statistic T defined in Section 2. The hardware retains a compact representation sufficient for these decisions. For $T > 0$ and a positive quantization boundary b,

$$|y_i| < b \quad \Longleftrightarrow \quad (x_i w_i)^2 < b^2 T.$$

Clearing denominators under the declared input and constant representations gives an exact integer comparison, avoiding a rounded reciprocal square root.

Finite-precision candidates locate scale thresholds and element-code boundaries. Exact predicates on the original values and T certify these decisions, retaining equality for ties-to-even.

3.4 Attention: Interval Certification and Refinement

Attention certifies the composed output $\mathrm{Softmax}(z)V$. For processed scores and values, define

$$\ell = \sum_j e^{z_j - m}, \qquad \mathbf{o} = \sum_j e^{z_j - m} V_j, \qquad \mathbf{y} = \mathbf{o}/\ell,$$

where m is the running maximum. The evaluator retains m and outward interval bounds for the denominator and signed numerator.

When a new tile changes the maximum from m to m', the previous aggregates are rescaled by $e^{m-m'}$ before accumulation. Outward bounds cover exponential evaluation, rescaling, signed multiplication, accumulation, and division.

A group is committed when its intervals uniquely determine its shared scale and every element code. Ambiguous groups trigger higher-precision refinement; at the precision limit, the evaluator returns unresolved. Appendix D specifies the exceptional cases and refinement rules.

4 Hardware Realization

4.1 Endpoint Boundaries

The three endpoints emit group-32 scaled-FP8 outputs. SwiGLU accepts scaled-FP8 gate/up operands; RMSNorm accepts BF16 inputs and FP32 weights; attention accepts FP32 scores and BF16 values. Each endpoint and its control share the same input/output boundary, including internal state, arithmetic, and output commitment.

4.2 SwiGLU Datapath

In Figure 2, G11 production and multiplier binding yield compact16 records and proposals. Proposal reduction overlaps record generation; replay reads the records and separate scale byte when the shared scale is ready. G11 fields and full product centers are temporary. Two records fit in a 32-bit word, and the block initiation interval is 32 cycles.

Controls use Q30 Exp/Newton arithmetic or cubic evaluation with 16/32/64 segments and 24/24/32-bit coefficients. Each stores a 24-bit scaled-BF16 record: a BF16 value and local scale byte. Replay quantizes it at the reduced block scale. The complete comparison includes evaluation, binding, storage, reduction, and replay, capturing both approximation and intermediate rounding.

4.3 RMSNorm Datapath

RMSNorm accumulates the row sum of squares while encoding input–weight pairs into 40-bit records. Each regular record stores a sign, a nine-bit exponent sum, and seven- and 23-bit BF16/FP32 fractions with implicit leading bits. Readout multiplication reconstructs the numerator, replacing separate operand storage while preserving the exact boundary decisions.

Separate tags distinguish certified tiny nonzero products, which retain minimum-scale proposals, from mathematical zero. Original input squares still enter the row statistic. Exact boundary tests then select group scales and codes. Power-of-two factoring and shared encoding/read selection reduce logic cost. The wide control includes the same whole-row boundary and operand-storage scope.

4.4 Attention Datapath

Q40 first scans scores for the row maximum, then computes exponential bounds and aggregates the denominator and weighted numerators, reading values once. Channels store 64-bit signed centers. Shared scaling and outward error bounds cover exponential approximation, truncation, and rescaling. Registers separate center reconstruction from wide interval arithmetic and certification.

Fixed-Q40 commitment requires a unique scale and all element codes; otherwise it reports unresolved without an internal precision retry. Q48 reads scores and values once. Both endpoints include aggregation and commitment, excluding external score/value storage energy. Adaptive refinement belongs to the software evaluator (Section 3.4).

5 Evaluation

5.1 Experimental Setup

We evaluate correctness, hardware cost, and model performance. Hardware comparisons report routed standard-cell area and energy estimates from the Nangate45 OpenROAD flow at the typical corner (The OpenROAD Project), using a 400 MHz target, standard-cell storage, and matched workloads and output boundaries. Routed zero-delay activity covers reset, fill, computation, and drain; energy includes internal, switching, and leakage components. The separate ASAP7 transport comparison fixes the storage macro.

SwiGLU uses a 32-cycle block initiation interval and 4,096 activity lanes. RMSNorm measures one complete 2,560-element row; attention measures one 64-token row with 256 output channels, excluding external score/value storage energy. Model evaluations separately test joint replacement during prefill and decode on GSM8K, HumanEval, and GPQA-Diamond.

5.2 Correctness and Certification

Table 2 summarizes the three operators' evidence. SwiGLU mathematical checks cover 26,320 gate–multiplier groups, 52,640 boundaries, and 1,026 tails with zero failures; exhaustive mapping and replay connect these decisions to storage. RMSNorm combines exact predicates, width analysis, and row replay. Attention certifies groups whose intervals uniquely determine the scale and all codes.

Fixed-Q40 hardware reports unresolved for ambiguous outputs; software supports adaptive refinement. Table 2 states hardware replay coverage. Appendices C and D give detailed certificates and arithmetic specifications.

Table 2: Correctness evidence for the three operators. Mathematical guarantees and implementation replay cover different obligations. All listed captured-input replays match element codes and scale bytes.

Operator	Decision guarantee	Implementation evidence
SwiGLU	Finite-domain mathematical cells and scale-query responses	4,318 semantic gate inputs; 622,160 record states; 131,072 captured lanes
RMSNorm	Exact integer predicates with bounded arithmetic widths	1,024 captured rows (2,621,440 elements); separate 4,096/8,192-element tests
Attention	Unique output cells certified by outward intervals	44 captured rows, six long-row cases, two extreme cases, and one model diagnostic row; zero unresolved outputs on these tests

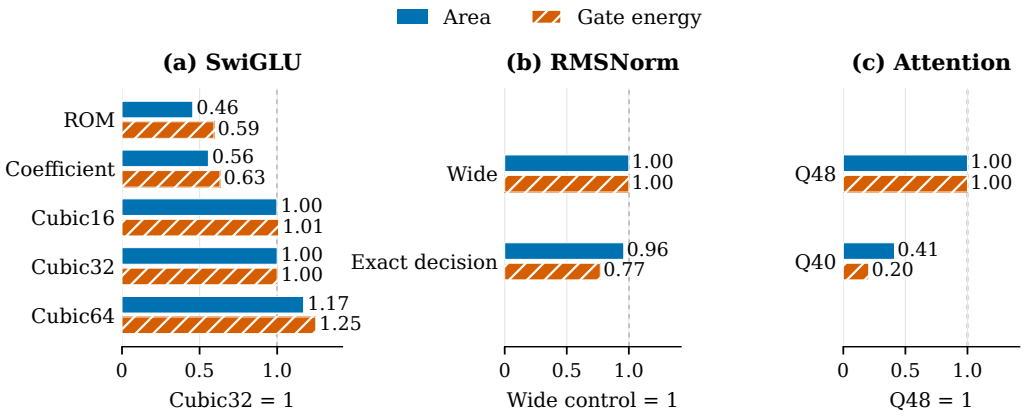

Figure 3: Complete-endpoint area and gate energy, normalized to Cubic32 (SwiGLU), the wide control (RMSNorm), and Q48 (attention); lower is better. Workloads and storage boundaries follow Section 5.1. Tables 5 and 3 give absolute values, including the Q30 control.

5.3 Hardware Efficiency

All endpoints in Figure 3 meet 400 MHz setup and hold constraints. SwiGLU's standard-cell ROM reduces area by 54–61% and energy by 41–53% against polynomial controls. Against the coefficient producer with shared G11 and compact16 interfaces, the reductions are 18.09% and 6.17%. Lower ROM energy also holds on two additional activity streams.

RMSNorm reduces area, gate energy, and row cycles by 4.1%, 23.0%, and 20.4%; attention reduces them by 58.8%, 79.6%, and 37.4%, respectively (Table 3). Both attention endpoints produce identical output bytes on the measured row.

5.4 Mechanism and Representation Ablations

Producer and precision. ROM versus coefficient isolates producer cost under a shared state interface. The G8–G11 sweep includes lower-precision consumer corrections: G11 costs 19,932.4 μm^2 and 30.864 pJ/lane, versus 35,843.2–35,886.9 μm^2 and 58.608–58.705 pJ/lane for G8–G10. Extra correction context and logic outweigh the narrower unary state (Appendix E.3).

Transport. With producer, proposals, macro, clock, and block interval fixed, compact16 reduces ASAP7 area from 1,357.46 to 1,257.00 μm^2 and energy from 5.457 to 5.295 pJ/element versus directed15. Both use one macro at 3.028 pJ/element; encoding, decoding, and interconnect account for the saving (Table 7).

Direction and commitment. Forcing G11 direction to zero or one produces 135 or 191 mismatches among 131,072 captured lanes (0.103% or 0.146%). Unary group-32 MX commitment

Table 3: Matched complete-row endpoints at a 2.5 ns Nangate45 target. Energy is routed zero-delay gate energy per output element on one complete row: 2,560 elements for RMSNorm and 64 tokens with 256 output channels for attention. Attention energy excludes external score and KV storage.

Operator	Endpoint	Area (μm^2)	Energy (pJ/output)	Cycles/row
RMSNorm	**Exact decision**	**1,094,570**	**15,906**	**28,563**
RMSNorm	Wide control	1,141,090	20,658	35,869
Attention	**Q40**	**466,386**	**165,392**	**110,855**
Attention	Q48 control	1,131,310	809,536	177,189

Table 4: Full-set task scores (%). GSM8K: numeric extraction on 1,319 questions; HumanEval: pass@1 on 164 problems; GPQA-Diamond: generated-answer accuracy on 198 questions. Shaded bold rows identify Replacement. A dash denotes an unreported score.

Model	Method	GSM8K	HumanEval	GPQA-Diamond
DeepSeek-V4-Flash-0731	Native	95.68	92.68	89.89
	Replacement	**95.68**	**91.46**	**90.90**
Qwen3.5-9B	Native	93.03	87.80	81.43
	Replacement	**93.63**	**89.63**	**80.30**
Qwen3.8-27B	Native	95.91	96.95	90.91
	Replacement	**95.98**	**97.56**	**91.43**

before multiplication produces 35,096 code mismatches (26.78%) and 145 scale mismatches across 4,096 groups. These interventions respectively change direction information and the unary quantization interface (Appendix E.1).

5.5 Joint Replacement in Language Models

We evaluate DeepSeek-V4-Flash-0731, Qwen3.5-9B, and Qwen3.8-27B on complete task sets: GSM8K (1,319 questions) (Cobbe et al., 2021), HumanEval (164 problems) (Chen et al., 2021), and GPQA-Diamond (198 questions) (Rein et al., 2023). Native retains the original operators; Replacement jointly replaces SwiGLU, supported RMSNorm sites, and softmax-based attention during prefill and decode. Linear-attention branches remain native. Each configuration generates its own responses, scored with the same task-specific rule within each comparison.

Table 4 shows comparable task performance under joint replacement. GSM8K accuracy is unchanged for DeepSeek and increases by eight and one correct answers for the two Qwen models. HumanEval changes by -2, $+3$, and $+1$ passing programs, respectively. These independently generated task outcomes measure empirical model quality; numerical correctness is established by the operator-level evidence above. Appendix F gives the task protocols, matched quantization-interface configurations, and paired-comparison procedures.

6 Conclusion

We presented a sufficient-state framework for nonlinear computation in scaled-FP8 pipelines. It preserves the information needed for final quantization through consumer composition and shared-scale selection. Directed states for SwiGLU, exact boundary predicates for RMSNorm, and interval certification for attention instantiate this principle. Matched hardware comparisons show lower area and gate energy for all three endpoints relative to their respective controls. Joint replacement across three language models yields task performance comparable to native execution on the evaluated benchmarks. Together, these results support a design principle: certify the state required by the final quantizer, then optimize its generation and transport. The lowest-cost implementation depends on the complete datapath, rather than state width alone.

AI USE STATEMENT

Generative AI tools assisted with problem framing, proof and experiment-design suggestions, manuscript drafting, figure code, and experiment scripts. The authors checked the mathematical arguments, ran and audited the reported experiments, and take responsibility for all claims and the final manuscript.

ETHICS STATEMENT

This work does not involve human subjects, private data, or a new dataset release. Model measurements use publicly named checkpoints and the GSM8K, HumanEval, and GPQA-Diamond protocols specified in the appendix. We are not aware of additional Code of Ethics issues beyond the usual dual-use considerations of efficient inference hardware.

REPRODUCIBILITY STATEMENT

Table 1 fixes the numerical interface. Appendices B and C give the suffix-label proof, canonical input enumeration, token and record formats, online algorithm, and independent directed-bound verification procedure. Appendix D specifies the reduction predicates and interval rules. Appendix E states the implementation constraints and activity protocols; Appendix F specifies operator coverage, generation, scoring, and comparison protocols. These descriptions distinguish mathematical certification, implementation replay, physical measurement, and empirical model accuracy. The supplementary numerical artifact provides pinned input hashes and an executable entry point for token mapping, suffix-label counting, and independent mathematical-cell checks, with separate RTL replay instructions.

REFERENCES

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021. URL https://arxiv.org/abs/2107.03374.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. In *Advances in Neural Information Processing Systems*, 2022. URL https://arxiv.org/abs/2205.14135.

Laurent Fousse, Guillaume Hanrot, Vincent Lefèvre, Patrick Pélissier, and Paul Zimmermann. Mpfr: A multiple-precision binary floating-point library with correct rounding. *ACM Transactions on Mathematical Software*, 33(2), 2007. URL https://doi.org/10.1145/1236463.1236468.

Sehoon Kim, Amir Gholami, Zhewei Yao, Michael W. Mahoney, and Kurt Keutzer. I-BERT: Integer-only BERT quantization. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 5506–5518. PMLR, 2021. URL https://proceedings.mlr.press/v139/kim21d.html.

Jay P. Lim and Santosh Nagarakatte. Rlibm-all: A novel polynomial approximation method to produce correctly rounded results for multiple representations and rounding modes, 2021. URL https://arxiv.org/abs/2108.06756.

NVIDIA. Boosting moe training throughput with advanced fusion kernels, 2026a. URL https://developer.nvidia.com/blog/?p=118581.

NVIDIA. Transformer engine: Activation c api, 2026b. URL https://nvidia.github.io/TransformerEngine/api/c/activation.html. Accessed September 21, 2026.

Open Compute Project. Ocp microscaling formats (mx) specification, version 1.0, 2023. URL https://www.opencompute.org/documents/ocp-microscaling-formats-mx-v1-0-spec-final-pdf.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof Q&A benchmark. *arXiv preprint arXiv:2311.12022*, 2023. URL https://arxiv.org/abs/2311.12022.

Jacob R. Stevens, Rangharajan Venkatesan, Steve Dai, Brucek Khailany, and Anand Raghunathan. Softermax: Hardware/software co-design of an efficient softmax for transformers. In *2021 58th ACM/IEEE Design Automation Conference (DAC)*, pp. 469–474, 2021. URL https://arxiv.org/abs/2103.09301.

The OpenROAD Project. Openroad flow scripts. URL https://openroad-flow-scripts.readthedocs.io/en/latest/. Nangate45 platform documentation, accessed September 21, 2026.

Jiangyong Yu, Xiaomeng Han, Xing Hu, Chen Xu, Zhe Jiang, and Dawei Yang. Nli: Non-uniform linear interpolation approximation of nonlinear operations for efficient llms inference. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2602.02988.

Joonsang Yu, Junki Park, Seongmin Park, Minsoo Kim, Sihwa Lee, Dong Hyun Lee, and Jungwook Choi. NN-LUT: Neural approximation of non-linear operations for efficient transformer inference. In *Proceedings of the 59th ACM/IEEE Design Automation Conference*, pp. 577–582, 2022. doi: 10.1145/3489517.3530505. URL https://arxiv.org/abs/2112.02191.

Zixi Zhang, Zhiwen Mo, Yiren Zhao, and Robert Mullins. Deep kernel fusion for transformers, 2026. URL https://arxiv.org/abs/2602.11808.

A Exact midpoint witness

Let $h = \mathrm{SiLU}(240)$. Since

$$e^{240} > 1 + 240 + \frac{240^2}{2} > 479,$$

we have

$$239.5 < h = \frac{240}{1 + e^{-240}} < 240.$$

BF16 has unit spacing in this range, so round-to-nearest-even gives

$$\mathrm{RNE}_{\mathrm{BF16}}(h) = 240.$$

Thus BF16 commitment removes the strict relation $h < 240$.

With $u = 1$ and shared output scale 2^{14}, E4M3 codes `0x07` and `0x08` reconstruct to

$$(7/512)2^{14} = 224, \qquad (1/64)2^{14} = 256,$$

whose midpoint is 240. Since $h < 240$, the exact product rounds to `0x07`. In contrast, BF16 commitment gives exactly 240, which is the midpoint and therefore rounds by ties-to-even to `0x08`. Hence the retained relation $h < 240$ is sufficient to recover the correct final code.

It remains to show that $q = 14$ is reachable under the declared block contract. Choose shared input scales $(q_g, q_u) = (13, 0)$ and two nonzero lanes $(240, 1)$ and $(G, 2)$, where

$$G = 448 \cdot 2^{13},$$

with all remaining lanes zero. For the peer lane,

$$G < 2\,\mathrm{SiLU}(G) < 2G,$$

and therefore

$$2^{13} < \frac{2\,\mathrm{SiLU}(G)}{448} < 2^{14}.$$

Its no-overrange proposal is thus 14, so the block selects $q = 14$. Therefore the mismatch above occurs in a legal block context.

B MINIMUM LABELS FOR SUFFIX RESPONSES

Suffix-query interface. Fix a finite state set and $Q = 127$. For each state s, the legal queries are $q \in \{q_0(s), \dots, Q\}$. The encoder emits a fixed-width label before q is known, and the decoder receives only this label and q. Define the reverse response string

$$r_s = (O(s, Q), O(s, Q-1), \dots, O(s, q_0(s))).$$

Under this interface, two states can share a label exactly when one response string is a prefix of the other. The abstract query domain includes every scale in the suffix, irrespective of joint realizability by a common peer configuration. Labels encode element-code responses; proposal reduction and block-scale generation are separate. With decoder side information, the minimum is computed within each side-information partition.

Proposition 2 (Suffix minimum)**.** *Let L be the number of distinct response strings with no strict extension in the declared response set. Then the minimum number of labels is L, and the minimum fixed width is $\lceil \log_2 L \rceil$.*

Proof. Distinct maximal response strings cannot share a label. Since neither is a prefix of the other, they disagree at some query legal for both states, and a deterministic decoder must distinguish them. Hence at least L labels are required.

Conversely, assign one label to each maximal response string. Every response string in the finite set has a maximal extension, possibly itself, and is encoded by the label of any such extension. At a legal query q, the decoder returns entry $Q - q$ of that maximal string. This entry lies within the original response string and therefore equals its required response. Thus L labels suffice. □

Independent combinatorial check. A separate checker constructs a trie directly from the serialized response strings, without importing the producer, quantizer, or suffix-generation implementation. It counts maximal strings and explicitly verifies that every serialized response is a prefix of its assigned maximal string.

Certificate quantity	Count
Positive-magnitude $(q_0, \text{response})$ records	4,206
Distinct maximal positive-magnitude strings	4,191
Records requiring no additional label	15
Maximal strings including signs and canonical NaN	8,383
Serialized records without a covering maximal string	0

Signed extension applies the sign bit to negative responses, including zero, while canonical NaN contributes a distinct `0x7f` response. The combined trie contains 8,383 maximal strings, so

$$2^{13} = 8192 < 8383 \le 16384 = 2^{14}.$$

Hence the declared suffix-query interface requires 14 label bits. The trie counts the required labels; mathematical certification establishes input-domain coverage and correctness of the response bytes.

C EXECUTABLE SWIGLU CONSTRUCTION

C.1 CONSTRUCTION AND TRANSPORT

Domain coverage. Each operand has 2,158 positive nonzero semantic magnitudes, 4,316 signed nonzero values, and two signed zeros. Sign symmetry reduces the up operand to its positive magnitudes. Of the gate values, 1,026 negative inputs at or below -208 are handled by the separately certified zero-tail branch. The remaining domain therefore contains

$$3290 \times 2158 = 7,099,820$$

active gate–positive-up pairs.

Each pair is mapped to the generator key $(A_g, d_g, M_u, \rho, q_0)$, where $\rho = L_g + L_u - q_0$. Canonicalization first merges raw aliases of the same semantic input, and the generator key then merges identical future decision problems.

Stage	Count	Check or invariant
Finite semantic gate inputs (including signed zeros)	4,318	Canonical input identity
Active gate / positive-up pairs	7,099,820	Complete domain enumeration
Generator states	622,160	Identical decision problem
Positive $(q_0, \text{response})$ records	4,206	Identical-response deduplication

The mathematical certificate validates the decision cells represented by these generator states; Appendix B separately proves the minimum suffix-label count.

G11 token fields. For an active finite gate, L is the normalized exponent of $h = |\operatorname{SiLU}(g)|$. The normalized magnitude is rounded to eleven fractional bits to obtain the twelve-bit significand A. A rounded value of 4096 is renormalized to $A = 2048$ with L incremented. The 26-bit RTL token contains a two-bit kind, one sign bit, a signed ten-bit exponent L, the twelve-bit A, and one direction bit $d = \mathbf{1}[h > A2^{L-11}]$. Zero and NaN use separate kind tags.

Candidate and correction generation. The coefficient producer canonicalizes the gate into sign, exponent, and significand. A Q6-addressed piecewise-linear central path with analytic tail branches produces a candidate G11 token $T_0(x)$. Let $T^*(x)$ denote the certified target. Exhaustive enumeration identifies

$$\mathcal{E} = \{x : T_0(x) \neq T^*(x)\}.$$

The corrected tuple is stored by key (s, L_{in}, M). Of the 269 corrections, 222 restore the direction, 42 replace the token tuple, and five recover an observable tail. After correction,

$$T_{\text{coeff}}(x) = T^*(x)$$

for every finite semantic gate input, matching the semantic-ROM target map.

Online algorithm.

1. Decode both elements and scales exactly. NaNs take priority; zero operands and the certified negative tail produce signed zero without a proposal.
2. Canonicalize the active gate and evaluate either producer to obtain the G11 token (s_g, L, A, d).
3. Decode $|u| = M_u 2^{L_u - 3}$ and compute
$$P = AM_u, \qquad k = L + L_u - 14, \qquad s = s_g \operatorname{xor} s_u.$$
Determine the smallest integer r such that the certified product magnitude does not exceed $448 \cdot 2^r$, and set the local proposal by clipping r to $[-127, 127]$.
4. Encode the directed product as directed15 or compact16, buffer the record, and reduce valid proposals to their maximum. If no valid proposal exists, set $q = 0$.
5. Replay every record at the resolved q and emit the element bytes and scale byte $q + 127$.

Compact16, bit for bit. The record fields are $s[15] \mid (e + 256)[14:6] \mid f[5:0]$, with an implicit leading significand bit. For an active magnitude, define

$$n = 4P + (2d - 1), \qquad j = k - 2, \qquad e = \lfloor \log_2 n \rfloor + j.$$

If $e < -138$, emit signed zero; if $e > 136$, emit signed 2^{136}. Otherwise, retain the seven leading bits of n and set the least significant retained bit if any discarded bit is nonzero. If n has fewer than seven bits, left-shift it to seven bits. Store the lower six retained bits as f. Zero uses a zero magnitude field and NaN uses `0x7fff`.

A regular record decodes to

$$(64 + f)2^{e-6},$$

followed by dyadic E4M3 RNE at the query scale; the sign is retained through rounded zero. Exhaustive replay over all 622,160 reachable generator states verifies preservation of the certified responses.

Directed15. The record fields are $s[14] \mid d[13] \mid a[12:9] \mid (e+136)[8:0]$, with reserved exponent patterns for signed zero and canonical NaN. At the local proposal q_0, the certified E4M3 response determines

$$(a, e) = \begin{cases} (M, q_0 - 9), & E = 0, \\ (8 + M, q_0 + E - 10), & E > 0. \end{cases}$$

The encoder stores this dyadic center and direction d. At query scale q, the decoder aligns $a2^{e-q}$ and performs E4M3 RNE, using d only when the center is a rounding midpoint. Zero, NaN, underflow, and saturation follow the numerical contract.

Semantic-ROM realization. The semantic ROM stores one 26-bit target word for each of the 4,318 finite semantic gate inputs, with fields kind[25:24], sign[23], signed L[22:13], A[12:1], and d[0]. An independent word-level check verifies bitwise equality with the certified target map and unique coverage of all 64,770 finite raw aliases. RTL replay, including NaNs and out-of-domain scale-byte patterns, covers all 65,536 unary code–scale byte pairs. The ROM is synthesized as a standard-cell constant array, not an SRAM macro.

C.2 Certification

Independent mathematical check. For each (g, M_u), the verifier pulls every reachable response and proposal condition back to a rational interval on $h = |\operatorname{SiLU}(g)|$. Let $a = |g|$. Since

$$h = \frac{a}{1 + e^{-g}}, \qquad h > \beta \iff e^{-g} < \frac{a}{\beta} - 1 \quad (\beta > 0),$$

interval endpoints reduce to directed comparisons on the exponential term. MPFR 4.1.0 evaluates these comparisons with outward rounding, beginning at 256 bits and doubling precision when necessary. The check covers 52,640 extremal boundaries in 26,320 gate–multiplier groups with zero failures; the maximum required precision is 417 bits. The 1,026 excluded negative tails are certified separately by

$$|\operatorname{SiLU}(g)| < \frac{2^{-137}}{448 \cdot 2^{127}},$$

which keeps even the largest legal up operand below the smallest-scale zero midpoint.

Verification obligations. The verification chain assigns a separate check to each proof obligation:

Obligation	Evidence
Math-to-cell	Outward-MPFR verification of 52,640 extremal boundaries in 26,320 groups and 1,026 zero tails; zero failures.
Producer mapping	Coefficient and ROM producers agree with the certified G11 target map.
Record replay	Exhaustive encoding/decoding replay over 622,160 generator states; the independent suffix trie yields 8,383 labels.
RTL execution	All 65,536 unary code–scale patterns plus complete-endpoint replay on 131,072 real-activation lanes and their block scales.

These checks respectively connect the mathematical function, certified G11 interface, stored records, and executed RTL to the final element codes and scale bytes.

Proof of Proposition 1. Producer equality gives $T_p(g) = T^*(g)$ and therefore the same downstream bound-product state. The independent mathematical check places every active product in the certified response and proposal cells for each legal suffix query, and exhaustive record replay preserves those responses through encoding and decoding. The certified tail, zero, and NaN branches follow Table 1. Finally, reducing the exact valid proposals, or selecting $q = 0$ for an empty valid set, reproduces the declared block scale. Hence both the element codes and scale byte satisfy Proposition 1.

D Reduction Arithmetic and Implementation Details

D.1 RMSNorm: Exact Predicates and Stored State

For $T = N^{-1}\sum_i x_i^2 + \epsilon > 0$, each positive output boundary b is tested through

$$\frac{|x_i w_i|}{\sqrt{T}} < b \quad \Longleftrightarrow \quad (x_i w_i)^2 < b^2 T. \tag{7}$$

Equality is retained for ties-to-even. The hardware accepts finite BF16 inputs and finite FP32 effective weights, with $\epsilon = \mathrm{FP32}(10^{-6}) = 8796093/2^{43}$. Clearing dyadic denominators gives integer comparisons. The arithmetic supports rows whose lengths are multiples of 32, up to 8,192 elements; the measured physical instance holds 2,560 elements.

Width bounds. An independent enumeration checks the integer coordinate $X = x2^{133}$ for all 65,280 finite BF16 patterns. Algebraic bounds on the row statistic, denominator, product coefficient, and boundary comparison validate the 535-, 579-, 78-, and 1,408-bit containers. Factoring powers of two from E4M3 boundaries reduces the implemented alignment logic while retaining equality and discarded nonzero bits.

Compact product record. The 40-bit record stores a sign, a nine-bit normalized exponent sum, a seven-bit BF16 fraction, and a 23-bit FP32 fraction. Implicit leading bits are restored on read. For normalized significands $M_x < 2^8$ and $M_w < 2^{24}$, a regular record represents $M_x M_w 2^{e-282}$ exactly, with $0 \le e \le 506$.

For $e < 0$, the nonzero product magnitude is below 2^{-251}. Since $\epsilon > 2^{-20}$, the normalized output is below 2^{-241}, hence below the smallest legal zero midpoint 2^{-137}. Tag 507 retains this active tail with proposal -127; tag 511 denotes mathematical zero with no proposal. Both preserve the sign. Original input squares remain in the row statistic. This distinction preserves scale bytes for all-tail and all-zero blocks.

Implementation replay. The measured RTL matches independent mathematical output codes and scales on 1,024 captured rows (2,621,440 elements). Separate 4,096- and 8,192-element replays validate the longer-row arithmetic. The 40-bit record and shared read-selection logic are included in the complete measured endpoint.

D.2 Attention: Interval Aggregation and Commitment

For scores z_j and values V_j, the software evaluator maintains a maximum m, denominator $\ell = \sum_j \exp(z_j - m)$, and signed numerator $o = \sum_j \exp(z_j - m)V_j$. A change from m to m' rescales the aggregates by $\exp(m - m')$. Outward bounds cover exponentiation, rescaling, signed products, accumulation, and final division. Inputs at this boundary are FP32 scores and BF16 values; group-32 output quantization is over channels.

A group is committed when its bounds determine a unique shared scale and every E4M3 code. The software evaluator refines ambiguous intervals and reports unresolved at its configured precision limit. The fixed-Q40 hardware follows a two-score-pass schedule: first find the row maximum, then accumulate the denominator and weighted values. Values are read once. It retains signed 64-bit channel centers and a shared error bound, reconstructs outward intervals for commitment, and reports unresolved without an internal precision retry.

The matched Q48 control reads each score and value once. On the measured 64-token row, Q40/Q48 read 128/64 scores and 16,384/16,384 value elements. Both emit the same 256 output codes and scale bytes. Additional Q40 replay covers 44 captured rows, six long-row cases, two extreme-BF16 cases, and one Qwen3.5-9B diagnostic row. A separate 65-row synthetic differential suite, with lengths from 1 to 128 and 256 channels, checks 16,640 codes and 520 scale bytes. These tests have zero unresolved outputs and zero differences from their independent references.

Table 5: Complete SwiGLU endpoints. Area and energy use matched implementation constraints. Code mismatches are measured over 131,072 real-activation lanes against the mathematical reference; all endpoints have zero scale-byte errors. Energy uses a common 4,096-lane activity prefix.

Complete endpoint	Area (μm^2)	pJ/element	Slack (ns)	Code mismatches
Semantic ROM + compact16	**16,325.8**	**28.958**	**0.521**	**0**
Coefficient G11 + compact16	**19,932.4**	**30.864**	**0.288**	**0**
Q30 Exp/Newton + scaled BF16	123,749.0	164.641	0.079	3,991
Cubic, 16 segments, 24-bit	35,727.5	49.268	0.068	9,229
Cubic, 32 segments, 24-bit	35,795.6	48.843	0.046	5,349
Cubic, 64 segments, 32-bit	41,895.5	60.991	0.036	4,043

E Evaluation protocols and hardware accounting

All endpoints use the same implementation constraints and input activity stream. Area, setup/hold timing, and internal, switching, and leakage power are extracted from the routed design. The 64-segment polynomial control uses additional pipeline stages to meet 400 MHz; endpoint outputs are compared after latency alignment. Numerical mismatch counts are evaluated separately against the mathematical reference.

On an additional pre-generated synthetic boundary/toggling stream, the coefficient and ROM endpoints consume 32.519 and 30.711 pJ/element, respectively, again favoring the ROM. Across this stream and the two Qwen activity traces, the ROM has lower total energy than the coefficient producer.

RMSNorm endpoint. Both complete endpoints include operand storage, whole-row computation, scale selection, and output replay. On the same captured 2,560-element row, the optimized exact-decision RTL and conventional wide control meet the 2.5 ns target after routing. Optimized RTL replay matches the independent mathematical-reference codes and scale bytes on 1,024 captured rows. Separate 4,096- and 8,192-element numerical replays pass; the area and energy in Table 3 refer to the measured 2,560-element physical instance and one activity row.

Attention endpoint. The complete Q40 and Q48 endpoints share a 256-channel output boundary and meet the 2.5 ns target after routing. Their routed gate replays on the same 64-token row produce identical 256 element codes and scale bytes, with no unresolved output. Q40 takes 110,855 cycles and reads 128 scores and 16,384 V elements; Q48 takes 177,189 cycles and reads 64 scores and the same 16,384 V elements. Q40 also matches the independent output reference on 44 captured rows, six long-row cases, two extreme-BF16 cases, and one Qwen3.5-9B diagnostic row, with no unresolved output. These measurements use the fixed-Q40 hardware instance described in Appendix D.

E.1 Mechanism ablation protocol

Both ablations use the same 4,096 captured Qwen3.5-4B input groups (131,072 lanes) and the same mathematical block-output reference.

For the forced-direction ablations, the G11 direction bit is overwritten uniformly with zero or one before binding, proposal generation, compact16 encoding, scale reduction, and replay. All proposals are recomputed from the modified state. The unaltered path has zero code and scale mismatches; forced-zero and forced-one produce 135 and 191 element-code mismatches, respectively, with zero block-scale mismatches over 4,096 groups.

For the early-commit ablation, the certified unary output is first quantized to group-32 E4M3/E8M0 using its own no-overrange scale, then dequantized and multiplied by the original up operand before the final group-32 quantization. This replaces the mathematical SiLU value at the multiplier input with its committed MX value. The ablation produces 35,096 element-code mismatches and 145 block-scale mismatches over the same 131,072 lanes and 4,096 groups.

Table 6: Complete precision/correction implementations in Nangate45. Energy uses the same 4,096-lane routed zero-delay activity. G8–G10 include consumer correction storage and logic.

Unary precision	Area (μm^2)	pJ/lane	Setup slack (ns)
G8 + correction	35,886.9	58.608	0.316
G9 + correction	35,847.5	58.654	0.259
G10 + correction	35,843.2	58.705	0.338
G11	**19,932.4**	**30.864**	**0.288**

Table 7: Same-producer transport comparison. Macro energy is 3.028 pJ/element in both cases. Source-RTL activity is annotated on routed netlists for the same 5,216-element stream (163 blocks), unlike the routed-netlist activity used in Figure 3. Reported maximum frequency is a timing result, not measured throughput at a different clock.

ASAP7 TC endpoint	Area (μm^2)	pJ/element	Non-macro pJ	$F_{\max}$ (MHz)
G11 + directed15	1357.46	5.457	2.429	517.085
G11 + compact16	**1257.00**	**5.295**	**2.268**	**558.963**

E.2 Scaled-BF16 control transport

The polynomial and Q30 controls use a 24-bit lane record consisting of an 8-bit local scale byte $q_i + 127$ and a 16-bit BF16 value. For the approximate product $\widehat{y}_i$, the local proposal q_i is computed before BF16 rounding, and the stored value is

$$\mathrm{RNE}_{\mathrm{BF16}}\left(\widehat{y}_i 2^{-q_i}\right).$$

After block reduction determines q, replay reconstructs the stored value with 2^{q_i} and rounds it to E4M3 at scale q.

For the Q30 control, the decoded gate and up significands are combined with the scaled sigmoid result into a 40-bit product with a separate binary exponent. The no-overrange proposal is computed from this unrounded product and clipped to $[-127, 127]$ before its significand is rounded to BF16 with ties-to-even. Cubic controls use their polynomial evaluator followed by the same scaled-BF16 transport. Thus these controls include evaluator approximation, post-binding BF16 commitment, and final E4M3 rounding.

E.3 Precision and Correction Cost

Each G8–G11 producer uses independently generated directed targets and its own correction table. G8–G10 additionally carry context for a sparse post-scale consumer correction, implemented with a 64×27-bit metadata FIFO and correction logic. All four complete RTL endpoints match the same mathematical-reference byte stream on 131,072 captured lanes; their final routed netlists match the common 4,096-lane activity prefix and meet 400 MHz setup and hold constraints.

Among the evaluated G8–G11 implementations with their corresponding correction mechanisms, G11 avoids post-scale correction context and achieves the lowest complete cost.

E.4 Same-Producer Transport Measurement

E.5 Physical-seed protocol

The coefficient endpoint and Q30 wide control were routed with placement and routing seeds 11, 23, and 37, while keeping RTL, the 400 MHz target, and the 4,096-element activity stream fixed. All six routed designs meet setup and hold constraints and match their finite-arithmetic models in gate-level replay.

Coefficient energy ranges from 30.693 to 30.753 pJ/element, while the Q30 control ranges from 164.707 to 165.482 pJ/element. These ranges quantify physical-flow sensitivity under the fixed activity stream.

F Model Evaluation Protocols and Controls

F.1 Operator Coverage and Task Protocols

Native retains the original model operators. Replacement applies the three operator families during prompt processing and autoregressive decoding. DeepSeek replacement covers shared and routed SwiGLU, supported mHC/qkv/qhead/final normalization sites, and sparse-prefill, sliding-window-decode, and compressed-decode attention paths. The Qwen replacements cover SwiGLU, residual RMSNorm, and softmax-based full attention while retaining linear-attention branches. For Qwen3.5-9B, the selected sites comprise 32 SwiGLU, 65 residual RMSNorm, and eight full-attention sites.

Within each task comparison, the policies use frozen questions, reference answers, prompts, and the same scorer. Each policy generates its own response for every question. These comparisons evaluate task quality under the deployed operator replacements.

GSM8K. All 1,319 test questions use frozen eight-shot prompts and a 512-token completion limit. Numeric extraction selects the last numeric match, removes comma separators and dollar signs, removes one terminal period, and compares decimal-normalized values. Missing or invalid answers are incorrect, and every question remains in the denominator. The Qwen configurations use BF16 weights, greedy decoding, an eager attention backend, and KV cache, with TF32 matrix multiplication disabled.

HumanEval. All 164 problems use zero demonstrations and one generated Python solution, with a 2,048-token limit and greedy decoding. The EvalScope-aligned prompt requests the full implementation; the first code fence is extracted when present. Generated programs are evaluated with the problem tests in an isolated Python 3.11 container. Scores are single-completion pass@1 from this local sandbox. The Qwen runs disable chat-template thinking to produce code directly.

GPQA-Diamond. The Qwen comparisons use all 198 questions, the same frozen reasoning prompts, greedy generation, and a 40,960-token completion limit. The generated answer is extracted with the same parser for Native, Matched-native, and Replacement. Accuracy is computed over the complete question set.

F.2 Matched Quantization-Interface Controls

Matched-native retains framework arithmetic at the selected operator sites and applies the replacement quantization interfaces. For SwiGLU, gate and up inputs are quantized and dequantized before framework SiLU and multiplication, followed by output quantization and dequantization. RMSNorm and full attention retain framework computation followed by the matching output interface.

The control uses the same selected sites, frozen task inputs, generation settings, and scorer as Replacement within each comparison. Native versus Matched-native examines the added quantization interfaces; Matched-native versus Replacement examines the change in evaluator under those interfaces.

F.3 Paired Comparison and Fixed-Token Protocols

Task responses are paired by question or problem identifier. The common task scorer assigns each pair to one of four categories: both correct, Native only, Replacement only, or neither correct. All examples remain in the paired accounting.

The separate DeepSeek fixed-token check uses eight fixed prompt tokens and two forced decode tokens for the replacement and reference evaluators. It captures selected module inputs, outputs, and logits during prefill and decode, then compares corresponding tensor leaves bitwise. The two tensor-parallel workers are paired by ascending worker process ID, assuming this order matches tensor-parallel rank. The comparison covers captured forward tensors before sampling.