An AI-Driven Co-Design Framework for CMOS and Emerging Memory Devices

Ruichen Qi[1], Anhang Li[2], Saptarshi Ghosh[3], Fred Rothganger[4], Suma George Cardwell[4] and Mehdi Saligane[1*]
[1]School of Engineering, Brown University, USA; [2]EECS, University of Michigan, USA;
[3]Institute of Neuroinformatics, University of Zurich and ETH Zurich, Switzerland; [4]Sandia National Laboratories, USA;
[*]mehdi_saligane@brown.edu

Designing bio-inspired hardware that approaches the efficiency of biological neural systems requires joint optimization across algorithms, circuits, devices, and physical implementation under device-physics and layout constraints. This work presents an end-to-end AI-driven *Neurons-to-Silicon (N2S)* framework that integrates task-level modeling, circuit synthesis, and physical-layout generation through a unified closed-loop optimization flow. By supporting CMOS and emerging magnetic-device implementations with physics-aware dynamical analysis, N2S provides a scalable co-design framework for future neuromorphic EDA research.

***Index Terms*—AI-assisted EDA, emerging devices, neuromorphic computing**

I. Introduction and Motivation

NEUROMORPHIC hardware offers a path to ultra-low-power bio-inspired computing by exploiting the temporal dynamics and internal states of Spiking Neural Networks (SNNs) [1]. However, these dynamics often fail to transfer reliably from models to silicon, where parasitics, mismatch, variability, and layout constraints can distort circuit behavior and require manual tuning [2]. This issue is especially critical for magnetic and spintronic neuromorphic systems, where computation is tightly coupled to device-level dynamics [3].

While AI-assisted EDA has advanced VLSI topology and layout optimization, most flows still target static metrics such as delay, power, and area, rather than the continuous-time dynamics required by neuromorphic computing. This creates a model-to-silicon gap, where network- or schematic-level behavior may collapse after layout due to missing links among models, circuits, devices, parasitics, and layout effects.

To address this challenge, we propose N2S, an AI-driven framework that connects Neurons to Algorithms (N2A) modeling [4], Algorithms to Netlists (A2N) synthesis, and Netlists to Silicon (N2S) implementation across CMOS and emerging-device platforms. The N2S framework enables device-constrained modeling, emerging-device-aware circuit design, code-based layout generation, AI-guided optimization, and benchmarking under realistic device-level effects. This work focuses on the N2S engine, with N2A and A2N serving as framework components to be developed in future work.

II. Method

Unlike conventional EDA flows, N2S formulates neuromorphic synthesis as an AI-assisted co-design problem driven by task objectives, model dynamics, device physics, and magnetic-device constraints. As shown in Fig. 1, the framework consists of three conceptual engines: N2A, A2N, and N2S. In this work, N2A and A2N define future interfaces for mapping user goals and optimized network representations into circuit-level netlists, while the current focus is the physical-level N2S engine.

The N2S engine further translates these netlists into physical layouts through an LLM-assisted code-to-layout workbench

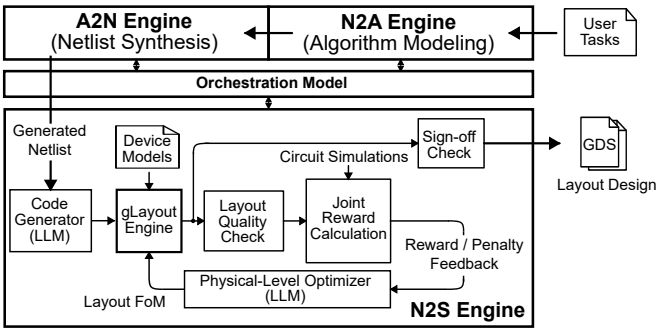


Fig. 1: The N2S Co-Design Framework.

based on gLayout [5], [6]. Instead of one-shot netlist-to-GDS generation, it performs closed-loop physical optimization using the generated netlist, PDK information, and magnetic-device models. Parameterized layouts are generated through code-based placement, routing, and layout construction, and are evaluated by layout-quality checks, circuit simulations, DRC/LVS verification, and PEX-aware sign-off analysis. These evaluations capture physical correctness, parasitic effects, device-geometry constraints, and deviations from the target neuromorphic dynamics.

The resulting metrics are combined into a joint reward calculation, which provides reward/penalty feedback to the reinforcement-learning-driven LLM physical-level optimizer. Within this loop, the LLM revises layout-generation code, adjusts placement and routing, tunes device parameters, and regenerates selected blocks through gLayout. For spintronic and domain-wall devices, the engine also incorporates geometry ranges, terminal configurations, compact-model parameters, and variability-sensitive behavior. This closed-loop RL process enables N2S to optimize circuit implementation, device-level dynamics, and layout quality, producing GDS layouts for sign-off evaluation.

III. Results

We illustrate the current N2S engine through three supporting components: compact-model support for magnetic devices, an MTJ-SNN simulation case study, and AI-driven physical-level optimization.

TABLE I: Compact model for relevant magnetic devices*.

Device class	Ref.
STT-MTJ / STT-MRAM	[7]
VCMA / voltage-controlled MRAM	[8]
SOT-MRAM / SHE / VGSOT-MTJ	[9], [10]
Probabilistic MTJ / p-bit / TRNG	[11]
Racetrack / domain-wall MTJ	[12]

* Github repo: `github.com/ReaLLMASIC/Memory-Survey`

Table I summarizes open-source compact-model support relevant to the N2S physical-design and simulation flow.

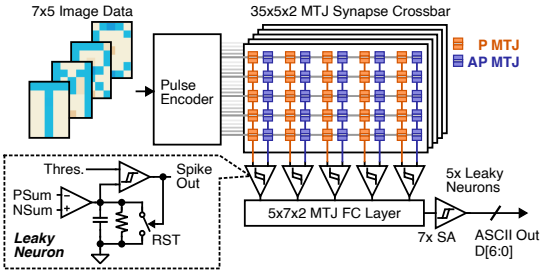


Fig. 2: Circuit of a reference MTJ-SNN image-to-text converter.

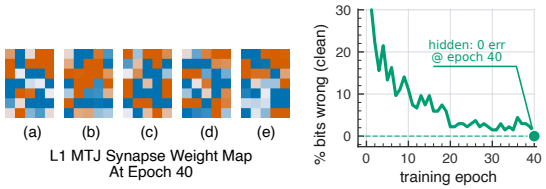


Fig. 3: ngSpice simulation-based MTJ-SNN results.

A. MTJ-SNN Simulation with Open-Source Tools

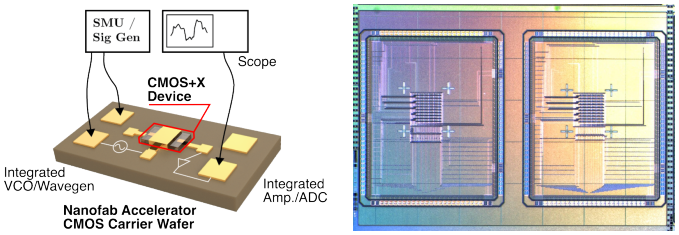
Using the compact-model support above, Fig. 2 shows an MTJ-based spiking neural network case study for image-to-text conversion in an open-source *SPICE* simulator. The circuit maps 7×5 binary image inputs into pulse-encoded signals, processes them through an MTJ crossbar, and uses leaky-neuron circuits with sense amplifiers to generate an ASCII output [13]. The trained weight map and validation loss are shown in Fig. 3. This example demonstrates how N2S can connect device models with neuromorphic behaviors.

B. Testing Carriers Fabricated using Open-Source PDKs

Closing the model-to-silicon gap also requires physical test structures for model calibration and validation. Open-source PDKs provide a practical path for fabricating such carriers and collecting device-level data at scale. As an example, the Nanofab Accelerator Project [14] contains tapeout of memory-device test carriers using the open-source SkyWater 130 nm process node, as shown in Fig. 4.

C. AI-driven N2S Physical-level Optimization

Fig. 5 shows an RL-based sizing optimization example for a two-stage Miller-compensated operational amplifier, demonstrating how the N2S engine can use RL to explore new circuit design spaces. Starting from an initial design, the RL trajectory progressively moves toward higher unity-gain-bandwidth-per-power FoM. Compared with the shotgun-sampled candidates, several RL-optimized designs reach a substantially higher FoM region, indicating that reinforcement learning can efficiently guide circuit design toward high-performance solutions.



(a) Nanofab Concept (b) MRAM Carrier Array

Fig. 4: Test tiles fabricated in Sky130.

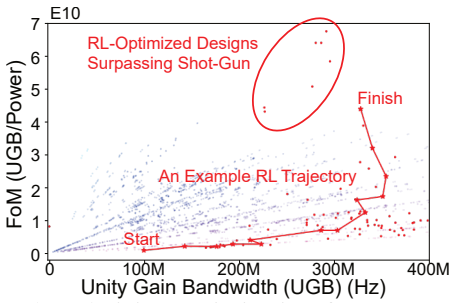


Fig. 5: RL-based sizing optimization for a two-stage Miller-compensated operational amplifier.

IV. Conclusion

N2S frames neuromorphic hardware design as a cross-layer search problem, linking model-level behavior with device- and layout-level feedback to identify physically feasible implementations early. This is especially important for magnetic and emerging-device platforms, where computation depends strongly on device physics, variability, and post-layout effects. N2S supports ultra-low-power neuromorphic systems as a promising alternative for edge-computation applications.

References

[1] G. Indiveri, "Neuromorphic is dead. long live neuromorphic." *Neuron*, 2025.

[2] M. Yan *et al.*, "Emerging opportunities and challenges for the future of reservoir computing," *Nature Communications*, 2024.

[3] J. Grollier *et al.*, "Neuromorphic spintronics," *Nature electronics*, 2020.

[4] F. Rothganger *et al.*, "N2a: a computational tool for modeling from neurons to algorithms," *Frontiers in neural circuits*, 2014.

[5] A. Hammoud *et al.*, "Human language to analog layout using glayout layout automation framework," in *ACM/IEEE MLCAD*, 2024.

[6] A. Hammoud *et al.*, "Openfasoc: An open platform towards analog and mixed-signal automation and acceleration of chip design," in *IEEE ISDCS*. IEEE, 2023.

[7] F. Garcia-Redondo *et al.*, "A compact model for scalable MTJ simulation," *arXiv preprint arXiv:2106.04976*, 2021.

[8] H. Lee *et al.*, "Analysis and compact modeling of magnetic tunnel junctions utilizing voltage-controlled magnetic anisotropy," *IEEE Trans. on Magnetics*, 2018.

[9] M. Kazemi *et al.*, "Compact model for spin–orbit magnetic tunnel junctions," *IEEE Transactions on Electron Devices*, 2016.

[10] K. Zhang *et al.*, "Compact modeling and analysis of voltage-gated spin-orbit torque magnetic tunnel junction," *IEEE Access*, 2020.

[11] K. Y. Camsari, B. M. Sutton, and S. Datta, "p-bits for probabilistic spin logic," *Applied Physics Reviews*, 2019.

[12] A. J. Edwards *et al.*, "Kinematic model of magnetic domain wall motion for fast, high-accuracy simulations," *Physical Review Applied*, 2025.

[13] A. Sengupta *et al.*, "Magnetic tunnel junction mimics stochastic cortical spiking neurons," *Scientific Reports*, 2016.

[14] National Institute of Standards and Technology, "The Nanotechnology Xccelerator," https://www.nist.gov/programs-projects/nanotechnology-xccelerator, 2024, created Mar. 13, 2024; updated Mar. 26, 2025; accessed Jun. 4, 2026.