Accepted as Keynote, DOE/AFOSR Energy Consequences of Information Workshop, February 2026.
https://www.energyconsequences.com/home
Theme(s): Neuromorphic Computing, Co-Design Tools for Emerging Technologies

A Full-Stack Neuromorphic System Implementation for Development and Benchmarking

Congyang Li[1], Kameron Gano[1], and Rajit Manohar[1,*]
[1]Department of Electrical and Computer Engineering, Yale University, New Haven, CT, USA
[*]**Corresponding Author:** rajit.manohar@yale.edu

Abstract

Neuromorphic computing leverages the co-design of algorithms and hardware to perform massively parallel and event-driven computation, demonstrating promising efficiency across a wide range of problem domains [1]. Developing new algorithms that can exploit the inherent parallelism of a neuromorphic hardware substrate is challenging, and the field is still exploring the range of potential hardware computational primitives. This exploration requires (i) the ability to rapidly modify the neuromorphic hardware, and obtain accurate estimates of energy, delay, and area; (ii) the ability to design new algorithms that can exploit new hardware features; and (iii) a parameterized compiler that can map algorithms to the hardware substrate. Today, even just benchmarking algorithm performance remains a significant challenge, and the lack of standardization continues to hinder progress in the field [2]. To tackle these issues, we are developing a full-stack neuromorphic system implementation.

The centerpiece of this system is the NeuroScale architecture [3], a parameterized hardware substrate developed using the ACT (Asynchronous Circuit/Compiler Tools) hardware description language [4, 5]. NeuroScale provides a baseline implementation that corresponds to the current state-of-the-art in large scale digital neuromorphic systems. NeuroScale is roughly comparable to the Intel Loihi architecture [6], but addresses a fundamental scalability limitation present in existing systems. Unlike IBM TrueNorth [7] and Intel Loihi/Loihi2 which rely on global synchronization protocols that inherently limit system scalability, NeuroScale employs a distributed synchronization protocol that scales efficiently with system size. Figure 1 demonstrates the software-hardware one-to-one equivalence with a clustered spiking neural network.

NeuroScale can be modified by changing its source in ACT, enabling comprehensive experimentation with different hardware concepts. The ACT simulator supports digital, analog, as well as novel device modeling through integration with Sandia's Xyce parallel circuit simulator [10]. Detailed hardware simulation, although slow, can provide accurate cost/benefit analysis of novel hardware constructs. For fast experimental validation, FPGA prototypes are generated with ACT2FPGA [9], a tool that translates ACT designs into synthesizable Verilog that is portable across FPGA vendors while preserving functional equivalence.

On the algorithms front, spiking neural networks (SNNs) are developed using Sandia's Fugu [8] framework, a high-level programming framework designed for developing neuromorphic algorithms. We are developing a library of Fugu bricks corresponding to a range of neuromorphic algorithms, including hand-crafted algorithms from the neuroscience literature. In addition, we are developing automated converters from more mainstream PyTorch-based machine learning generated SNNs into Fugu bricks so that we can import models from popular SNN training frameworks. The results in Figure 1 use Fugu's built-in simulator as the software reference model to ensure that the Neuroscale hardware simulation as well as the FPGA prototype produce results that precisely match Fugu's view of the SNN.

On the compiler front, we are developing customized software tools that can map Fugu networks to the NeuroScale architecture. Many of the core algorithms required for this purpose are also used by the electronic design automation community for chip implementation. We are leveraging our significant prior investment in

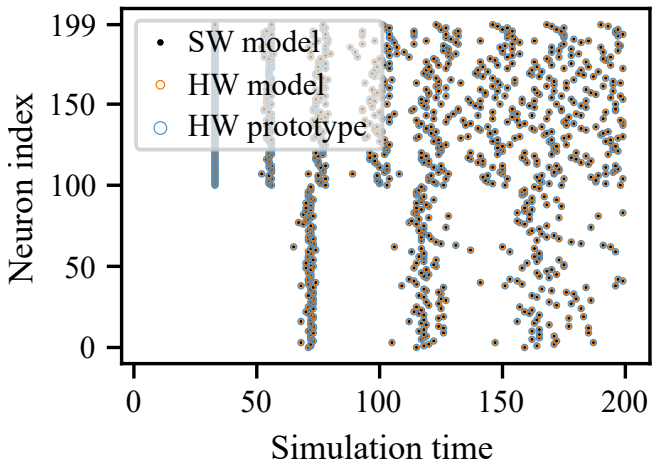

Figure 1: Identical spiking dynamics from the software model (black dots), hardware simulation (orange circles), and FPGA prototype (blue circles) running a clustered spiking neural network with 200 neurons (cluster 0: neurons 0–99; cluster 1: neurons 100–199). Cluster 0 has a bias of +1, and cluster 1 has a bias of +3, with a uniform threshold of +100. The network features an intra-cluster connection probability of 0.3, an inter-cluster connection probability of 0.1, and uniform synaptic weights of 1. The network is mapped onto 4 NeuroScale cores. Matching spike patterns over 200 simulation time steps demonstrate the architecture's deterministic execution. The software model runs on the Fugu SNN simulator [8], and the FPGA prototype is generated via ACT2FPGA [9].

developing open-source software for chip design [4] by customizing our existing tools to tailor them for the neuromorphic context.

References

[1] M. Davies, A. Wild, G. Orchard, Y. Sandamirskaya, G. A. F. Guerra, P. Joshi, P. Plank, and S. R. Risbud, "Advancing neuromorphic computing with Loihi: A survey of results and outlook," *Proceedings of the IEEE*, vol. 109, no. 5, pp. 911–934, 2021.

[2] J. Yik, K. Van den Berghe, D. den Blanken, Y. Bouhadjar, M. Fabre, P. Hueber, W. Ke, M. A. Khoei, D. Kleyko, N. Pacik-Nelson, *et al.*, "The neurobench framework for benchmarking neuromorphic computing algorithms and systems," *Nature Communications*, vol. 16, no. 1, p. 1545, 2025.

[3] C. Li, N. Imam, and R. Manohar, "A deterministic neuromorphic architecture with scalable time synchronization," *Nature Communications*, vol. 16, no. 1, p. 10329, 2025.

[4] S. Ataei, W. Hua, Y. Yang, R. Manohar, Y.-S. Lu, J. He, S. Maleki, and K. Pingali, "An open-source EDA flow for asynchronous logic," *IEEE Design & Test*, vol. 38, no. 2, pp. 27–37, 2021.

[5] R. Manohar, "ACT framework." GitHub `https://github.com/asyncvlsi/act/`, 2025.

[6] M. Davies, N. Srinivasa, T.-H. Lin, G. Chinya, Y. Cao, S. H. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain, *et al.*, "Loihi: A neuromorphic manycore processor with on-chip learning," *IEEE Micro*, vol. 38, no. 1, pp. 82–99, 2018.

[7] P. A. Merolla, J. V. Arthur, R. Alvarez-Icaza, A. S. Cassidy, J. Sawada, F. Akopyan, B. L. Jackson, N. Imam, C. Guo, Y. Nakamura, *et al.*, "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science*, vol. 345, no. 6197, pp. 668–673, 2014.

[8] J. B. Aimone, W. Severa, and C. M. Vineyard, "Composing neural algorithms with Fugu," in *Proceedings of the International Conference on Neuromorphic Systems*, pp. 1–8, 2019.

[9] R. Dashkin and R. Manohar, "Mixed-level emulation of asynchronous circuits on synchronous FPGAs," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2024.

[10] E. R. Keiter, H. K. Thornquist, R. J. Hoekstra, T. V. Russo, R. L. Schiek, and E. L. Rankin, "Parallel transistor-level circuit simulation," in *Simulation and Verification of Electronic and Biological Systems*, pp. 1–21, Springer, 2011.