DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof. Reference herein to any social initiative (including but not limited to Diversity, Equity, and Inclusion (DEI); Community Benefits Plans (CBP); Justice 40; etc.) is made by the Author independent of any current requirement by the United States Government and does not constitute or imply endorsement, recommendation, or support by the United States Government or any agency thereof.

2024 Analog Computing for Science Workshop

Position Papers

September 11-13, 2024
Bethesda, MD

Co-Chairs
David Soloveichik, University of Texas at Austin
Antonino Tumeo, Pacific Northwest National Laboratory

Organizing Committee
Sara Achour, Stanford University
Natalia Berloff, Cambridge University
Shantanu Chakrabartty, Washington University in St. Louis
Suma George Cardwell, Sandia National Laboratories
Jennifer Hasler, Georgia Institute of Technology
Siddharth Joshi, University of Notre Dame
Jaijeet Roychowdhury, University of California, Berkeley

U.S. DEPARTMENT *of* ENERGY | Office of Science

Disclaimer

U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research Program
Points of Contact:

Hal Finkel, hal.finkel@science.doe.gov
Marco Fornari, marco.fornari@science.doe.gov
Margaret Lentz, margaret.lentz@science.doe.gov
Kalyan Perumalla, kalyan.perumalla@science.doe.gov
Robinson Pino, robinson.pino@science.doe.gov
David Rabson, david.rabson@science.doe.gov
Bill Spotz, william.spotz@science.doe.gov

https://doi.org/10.2172/2506701

Contents

Srimani et al., Next-generation Probabilistic Computing Hardware with 3D MOSAICs, Illusion Scale-up, and Co-design

Talin et al., Device challenges to practical analog computing systems

Tossoun et al., Heterogeneous Computing with Analog Accelerators for Future AI Supercomputers

Valiante et al., Native-Domain Analog Computing for Combinatorial Optimization

Winstead et al., Probability as a Signal: "Bayesian Circuits" as a Computing Foundation

Wright et al., Physical Foundation Models: How to build practical 10^{18}-parameter artificial-intelligence processors

Part 3: Pre-Workshop Report

Part 1: Call for Position Papers

DOE/ASCR Workshop on Analog Computing for Science

Important Dates

- July 22, 2024: Deadline for position paper submission
- August 2, 2024: Notification of position paper acceptance
- September 11-13, 2024: Workshop
- WORKSHOP URL: https://www.orau.gov/2024AnalogComputingWorkshop

Motivation

On behalf of the Advanced Scientific Computing Research (ASCR) program in the U.S. Department of Energy (DOE) Office of Science, we are organizing a workshop on analog computing for science.

Analog computing capabilities have existed since the dawn of science, but modern techniques potentially allow for the construction, verification, and characterization of complicated analog-computing systems in a wide variety of contexts, from high-performance computational accelerators to nanorobotics and synthetic biology. In short, advances in analog computing can enable the creation of physical systems with complex behaviors that meet sophisticated requirements. Meeting our nation's needs, from needs in computing and modeling, to needs for advanced materials and energy technologies, continues to motivate pursuing novel kinds of complex systems, and thus analog-computing techniques, in all of these spaces.

The purpose of this workshop is to identify the priority research directions in analog computing for science. Participants will consider the status, recent trends, and challenges facing DOE's science and technology missions to which analog computing advances might be relevant. The workshop participants will then examine the opportunities, barriers, and potential for high scientific impact through fundamental advances in the underlying mathematical, statistical, and computational research foundations in addition to fundamental advances in computer science. The grand challenges and resulting priority research directions should span a variety of approaches; cover computer science, architectural advances, and infrastructure for analog-computing systems of all kinds; and cover different classes of techniques for analog computing: programming, implementation, verification, fabrication and characterization.

The workshop will be structured around a set of breakout sessions, with every attendee getting the opportunity to participate actively in the discussions. Afterward, workshop attendees—from DOE, industry, and academia—will produce a report that summarizes the findings made during the workshop.

Invitation

We invite community input in the form of two-page position papers that identify key challenges and opportunities in the area of analog computing for science. In addition to providing an avenue for identifying workshop participants, these position papers will be used to shape the workshop agenda, identify panelists, and contribute to the workshop report. Position papers should not present the

authors' current or planned research, contain material that should not be disclosed to the public, nor should they recommend specific solutions or discuss narrowly focused research topics. Rather, they should aim to improve the community's shared understanding of the problem space, identify challenging research directions, and help to stimulate discussion.

One author of each selected submission will be invited to participate in the workshop. By submitting a position paper, authors consent to have their position paper published publicly. Authors are not required to have a history of funding by ASCR.

Submission Guidelines

Position Paper Structure and Format

Position papers should follow the following format:

- Title
- Authors (with affiliations and email addresses)
- Topic: provide a short phrase capturing the topic(s), for example:
    - mathematical foundations
    - analog algorithms and programming
    - novel materials and devices
    - design and verification software
    - circuits and systems
    - error correction methods for analog computation
    - hybrid analog and digital systems
    - probabilistic computing
    - analog optimization (e.g. Ising machines)
    - in-memory analog computing
    - analog brain-inspired computing
    - chemical/biochemical computation
    - analog computation with photonics
    - distributed analog computation
- Challenge: Analyze and identify unique opportunities for analog computation in our era, dominated by digital technologies, and the future, considering comparative advantages, niche applications, and hybrid approaches.
- Opportunity: Describe how the identified opportunities may be pursued, whether it is through new tools and techniques, new technologies, or new groups collaborating on analog computing.
- Timeliness or maturity: Why now? What breakthrough or change makes progress possible now where it wasn't possible before? What will be the impact of success?
- References
- Each position paper must be no more than two pages, in single column format using 10pt or larger font, including figures and references. The paper may include any number of authors, but contact information for a single author who can represent the position paper at the workshop must be provided with the submission. There is no limit to the number of position papers that an individual or group can submit. Authors are strongly encouraged to follow the structure previously outlined. Papers should be submitted in PDF format using the designated page on the workshop website.

Notional Questions

Position papers should present a view on the future of analog computation, possibly taking inspiration from some of the following:

- For which applications does analog computation demonstrate superiority, and for what metrics?
- How does analog computation's energy efficiency compare to digital computation in various applications?
- What new materials, devices, systems, design software, etc, are needed to enable the future analog computation applications?
- What can be learned from biology's reliance on a mixture of analog and digital computation and applied to science and engineering problems? Examples of biological computing hardware include regulatory reaction networks and neural tissue.
- Is there a cross-cutting mathematical framework for analog computation?
- How do we scale analog computing systems to solve large-scale problems?
- What are the limits of analog computing and how do we approach this limit using practical devices, circuits, and systems?
- How do we program analog computing systems? Do we need new programming models and compilers? What does the software stack look like?
- What benchmarks and standards are necessary to evaluate and compare the performance of analog computing systems? How can we establish a common framework for assessing the capabilities and limitations of different analog computing approaches?
- How can hybrid systems that combine analog and digital computing be designed to exploit the strengths of both approaches? What are the challenges in developing efficient interfaces between analog and digital components? What opportunities or challenges does analog computing offer for integration with sensing devices?
- How do we design extremely heterogeneous systems for large-scale and edge systems?
- How do we address noise, variability, and robustness issues in analog computation? How do we leverage these non-idealities?
- What role can interdisciplinary collaboration play in advancing analog computation? How can fields as diverse as physics, materials science, biology, neuroscience, and computer science contribute to the development of analog computing?
- What programs and curricula must be developed to train the scientific workforce in analog computation?

Selection

Submissions will be reviewed by the workshop's organizing committee using criteria of overall quality, relevance, likelihood of stimulating constructive discussion, and ability to contribute to an informative workshop report. Unique positions that are well presented and emphasize potentially transformative research directions will be given preference.

Workshop Organizers

Co-Chairs

- David Soloveichik, The University of Texas at Austin
- Antonino Tumeo, Pacific Northwest National Laboratory

Organizing Committee

- Sara Achour, Stanford University
- Natalia Berloff, University of Cambridge
- Suma George Cardwell, Sandia National Laboratories
- Shantanu Chakrabartty, Washington University in St. Louis
- Siddharth Joshi, University of Notre Dame
- Jennifer Hasler, Georgia Institute of Technology
- Jaijeet Roychowdhury, University of California, Berkeley

For meeting technical questions, please contact: Todd Munson, Todd.Munson@science.doe.gov

Part 2: Position Papers

R&D Infrastructure for Analog Computing

Sapan Agarwal[1] (sagarwa@sandia.gov), Matt Marinella[2], Patrick Xiao[1], Chris Bennett[1], Nad Gilbert[1], Ben Feinberg[1]
[1]Sandia National Laboratories [2]Arizona State University

Topic: Co-design tools for analog systems

Background: Analog computational systems provide three fundamental advantages over digital systems:

1) Analog performs computation at the memory elements: The best digital ASICs and accelerators achieve around 0.5 pJ/operation when the hardware matches an application so that almost all data is kept locally near a processing unit, and any long-range data-movement will be limited enough to avoid dominating the power. A programmable resistor or analog memory overcomes this by performing both multiplication using Ohm's law and addition using Kirchhoff's current law. This allows for matrix vector multiplication (MVM) to be directly performed in an analog memory array or crossbar.
2) Analog devices can replace complex digital circuits: a single memory element to replace a digital multiplier that would require thousands of transistors. A massive and expensive pseudo random number generator can be replaced with a stochastic analog device.
3) Analog provides fundamental scaling advantages over digital by optimally using precision [1]: Analog resistive memory crossbars can perform a parallel read or a vector-matrix multiplication as well as a parallel write or a rank-1 update with high computational efficiency. For an N × N crossbar, these two kernels can be O(N) more energy efficient than a conventional digital memory-based architecture. If the read operation is noise limited, the energy to read a column can be independent of the crossbar size (O(1)). To avoid roundoff errors, digital systems need to maintain a high intermediate precision throughout a computation, even if the result is quantized down to a lower precision. Analog systems allow for energy to be paid for only the output precision needed and not for intermediate calculations.

Challenge - *The Need for a Comprehensive Co-Design Framework:*

To take advantage of the potential for analog computing novel algorithms will need to be designed that can execute on hybrid analog and digital systems. Consequently, we must simultaneously co-design algorithms and processing within the limitations of analog devices and SWaP (size, weight, and power) constraints. Converting from digital to analog and back is restricted to low precision, as the energy and time costs are exponential with the number of bits. Analog devices have inherent analog noise, device to device variability, radiation susceptibility, and other non-idealities that impact the accuracy [2], requiring a careful co-design of the devices, circuits, architectures and algorithms to ensure their robustness. These are key challenges as models are often not experimentally verified. It is frequently cost prohibitive to fabricate a test chip based on novel devices that are not typically available in foundries. Given this, it is unclear how these novel technologies will impact real algorithms when real device behavior and architectural designs are considered.

This is especially challenging as we cannot rely on the same abstractions that powered digital systems for the last 50 years. In digital computing, a transistor is abstracted to an ideal 1 or 0, logic gates are abstracted to CPU instructions, and instructions are abstracted to high level programming code. In analog systems, the properties of the underlying devices directly impact the application. This means that those properties need to be properly abstracted, modeled and passed up to application and system developers. Optimally exploiting analog computing means leaning into and exploiting the fundamental the limits of noise and precision and not just trying to mitigate the noise.

To realize the full potential of analog neuromorphic computing, we need a universal, consistent experimental and modeling methodology which can benchmark the energy efficiency and algorithmic performance of emerging analog architectures. We need to develop a fundamental understanding and a predictive multi-scale model of noise and non-idealities on analog devices. This means being able to understand and mitigate the impact of noise on analog accelerators. The mitigations include device level mitigation such as developing new memories that are more resilient and higher precision, circuit techniques such as calibration rows and periodic refreshes, and algorithmic techniques such as retraining neural networks to be aware of and to compensate for errors.

Opportunity: To advance the development of analog computing systems three related tools are needed:

1) An experimental platform that allows 1000's of devices to be characterized and allows for small array demonstrations. This platform must: support a range of device types (2-4 terminals), programming voltages, read & programming currents; separate device fabrication from control circuitry fabrication; and provide an easy programming interface that has been abstracted to a higher-level language like python.
    - Without the statistics that come from 100's to 1000's of devices it will be impossible to move towards real systems and beyond single hero device demonstrations. However, the infrastructure required to do this is prohibitive for most academic groups.
2) Accuracy modeling tools that can take the data from the previous platform and project the impact on a range of applications. Such a tool needs account for all relevant physical effects including read noise, write errors, parasitic resistances, ADC variability and such. It must also support the easy addition of new error models and new device types will have new types of error that must be modeled. From the application side, the tool should be plug-in compatible with standard programming tools like keras, pytorch and numpy (for non-neural network applications).
3) System architecture tools are needed to model a full hybrid analog and digital system, track data movement, and project energy efficiency. Discrete event simulations are needed to accurately model how data flows through a system and count the computations needed. Compilers need to be built on top of the simulation framework to allow for applications to be mapped to potential architectures.

There is a unique opportunity for the Department of Energy to lead the development of these tools and to ensure that they are made broadly available to the R&D community.

Timeliness or maturity:
There are a number of groups that have started building examples and tools to do this. For example, a full software infrastructure has been built for Field Programmable Analog Arrays (FPAA) that allow for a graphical user environment, place and route, and abstractions for higher level design[3]. This extremely powerful ecosystem is specifically built around the floating gate FPAA hardware.

A more flexible device metrology platform is being developed by both NIST [4] and Sandia/ASU [5]. Several open-source modeling tools have been developed to model the accuracy of analog accelerators including AiHWkit[6], neurosim[7] and CrossSim[8]. Sandia has begun building a discrete event framework for modeling hybrid analog/digital accelerators based on the structural simulation toolkit [9], a DOE supported discrete event simulator originally designed for modeling high performance computing (HPC) systems. To date, a RISC-V processor with an analog co-processor has been modeled and a compiler for it has been developed.

Going forward it is going to be critical to ensure that all of the tools at different levels of design are broadly available, linked, and support a wide range of devices and applications. There is critical need and opportunity for the Department of Energy to support and encourage the development of this design ecosystem.

Acknowledgement: Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

References:
[1] S. Agarwal *et al.*, "Energy Scaling Advantages of Resistive Memory Crossbar Based Computation and its Application to Sparse Coding," *Frontiers in Neuroscience,* vol. 9, p. 484, 2016, Art no. 484
[2] S. Agarwal *et al.*, "Resistive Memory Device Requirements for a Neural Algorithm Accelerator," in *International Joint Conference on Neural Networks*, Vancouver, CA, 7/2016 2016, pp. 929-938.
[3] J. Hasler, "Large-scale field-programmable analog arrays," *Proceedings of the IEEE,* vol. 108, no. 8, pp. 1283-1302, 2019.
[4] M. Dowell, "Advancing measurement science for microelectronics: CHIPS R&D metrology program," in *Metrology, Inspection, and Process Control XXXVIII*, 2024, vol. 12955: SPIE, p. 129550I.
[5] D. Wilson *et al.*, "A Discovery Platform to Characterize Emerging Nonvolatile Memories for Computing," in *2024 IEEE 42nd VLSI Test Symposium (VTS)*, 22-24 April 2024 2024, pp. 1-5
[6] https://github.com/IBM/aihwkit
[7] https://github.com/neurosim
[8] S. Agarwal *et al.* "CrossSim." http://cross-sim.sandia.gov
[9] A. F. Rodrigues *et al.*, "The structural simulation toolkit," *SIGMETRICS Performance Evaluation Review,* vol. 38, no. 4, pp. 37-42, 2011.

Nanoscale Sampling for Robust and Energy-Efficient Edge-Based Uncertainty Quantification

Christopher Bennett[1](cbennet@sandia.gov), Patrick Xiao[1], Ravi Patel[1], Sangheon Oh[1], Robin Jacobs-Gedrim[1], Will Wahby[1], Ben Feinberg[1], Todd Monson[1], Bert Debusschere[1], A. Alec Talin[1], Jean Anne Incorvia[2], and Sapan Agarwal[1]
[1]Sandia National Laboratories, [2]UT Austin

Topics: Probabilistic Computing, Future Analog Applications, Analog in-memory computing

Challenge: *Demonstrating that machine learning systems can be trusted via uncertainty quantification (UQ), and proving that hardware implementations of UQ systems are energy-efficient enough for edge applications.*

Machine-learning (ML) models are already wide-spread in industry and defense applications, and becoming more so every month; however, their predictions are difficult or impossible to interpret due to a general interpretability crisis (*i.e.,* the neural network is often a "black box"). Among others, out of distribution (OoD) extrapolation remains a severe issue, as deep models will confidently generate wrong outputs even when inputs are OoD with respect to the training set. When applying ML to science applications like particle detection it's critical to be able to quantify the uncertainty and place error bounds on whether a detection is valid. In safety-critical ML applications, for instance autonomous vehicles carrying humans or valuable cargo, the long tail outcomes are unacceptable. For instance, knowing whether a foggy environment is OoD with respect to a perception model in an autonomous vehicle, could have life or death consequences.

Fortunately, Bayesian Neural Networks (BNNs), which approximate the probability distribution of their parameters based on the training set, produce both an output and its associated confidence interval (CI). In addition, BNNs can estimate both aleatoric, or data-derived uncertainty, and epistemic, or world-model-derived, uncertainty [1]. Despite this, Bayesian neural networks are rarely used due to their immense computational cost. A network must be sampled multiple times and random values need to be sampled for every weight. The workhorse of machine learning, GPUs are not optimized for repeated random sampling and cannot run Bayesian networks at scale.

Opportunity: *Analog computing will enable real-time uncertainty quantification through the use of analog hardware-accelerated Bayesian networks.*

Analog accelerators are particularly well suited to accelerate BNNs trained with variational inference (VI) [2]. In a VI-BNN, neural network parameters are presented by a trainable mean (*mu*) and standard deviation (*sigma*), as in Fig.1(a). In digital there are two limiting computational challenges. First, each weight must be sampled from a stochastic distribution and second, stochastic weight matrices need to be multiplied by activations. Analog is uniquely suited to accelerate this. It's been well established that given a matrix of pre-programmed conductances, the combination of Ohm's law and Kirchhoff's law yield's huge energy benefits. Specifically for an N × N crossbar, analog vector-matrix multiplies are O(N) more energy efficient than a conventional digital memory-based architecture due to a high degree of parallelism [3]. Recently, it's been shown that the same concept can be applied to VI-BNNs[4], as shown in Fig.1(c). Critically, a *single stochastic memory device* now encodes the standard deviation noise added to each weight to generate the stochastic weight. Depending on the quality of the gaussian from this nanodevice, and the size of the crossbar array in the AIMC Bayesian system, a high quality digital pseudo-random number (p-RNG) generator circuit may require between 10-1000 as much energy to harvest the same physical scalar [4]. Notably, this is assuming an optimized circuit for p-RNG; the advantage can scale to thousands of times better, for various transistor nodes or CPU/GPU architectures.

Despite these initially promising results, three critical challenges remain:

1) *Algorithm/Application*: Our early work showed that VI-BNN performs on smaller ML tasks such as CIFAR100 in a way that increases energy efficiency per probabilistic multiply-and-accumulate (MAC) by 10-100x as compared to digital multiply-and-accumulates (depending on size of the AIMC array). However, beating digital isn't enough to guarantee edge deployment, if the number of sampling operations to support complex tasks such as ImageNet or sensor fusion increases super-linearly.
2) *Device Engineering:* It has been hypothesized that an in-plane magnetic tunnel junction (iMTJ) can contain an internal state variable that modifies the sigma of the device in a non-volatile way[4]. However, to date, simulations and experiments only prove volatile modifications of that internal state variable (*i.e*., via the voltage-controlled magnetic anisotropy effect [5]). Recently, modifications of noise in an electrochemical random access memory (ECRAM) device meeting software accuracy BNNS were made [6], but the results require temperature

and/or programming conditions that would both be hard to scale in realistic systems (Fig 1(d)). In addition, programmable ReRAM noise has recently been experimentally realized on Sandia's Discovery Platform [7] which is near-ideal immediately after programming, but the internal state variable quickly relaxes (Fig. 1(c)).

3) *Scale/Energy Efficiency*: Hardware-ready VI-BNN only works if the AIMC modules combining mean and sigma values is large enough, as otherwise the fixed analog-to-digital (ADC) converter circuits begin to cut into the energy-efficiency of the system [4]. This poses a high bar for system architecture.

Given our clear-eyed view of the topic's many pitfalls, several promising routes of research are available:

1) *Evaluation of ideal devices for stochastic synaptic sampling:* As above, the device engineering aspect is difficult, but critically, a co-design approach can be used to accurately uncover which parameters are important and which are not. Using CrossSim's [8] built in ability to simulate VI-BNNs, we can conduct deep studies on the effect of synaptic bit-reduction, controllable and uncontrollable noise factors, and device-to-device variability. These can inform projected paths for realized nanosynaptic samplers.
2) *Invention of non-volatile stochastic nanosynapses*: While device requirements are high, they are solvable given fundamental device designs. Promising routes include exploration of modification of ECRAM device, i.e. selective modification of the reservoir based on programming conditions or thermal modification internal to the device, or modification of existing MTJ devices, such as via magneto-ionic modification of the junction.
3) *System design for reliable AI*: Routes such as ADC quantization or other network simplifications, have been scarcely explored so far to yield more compact hardware UQ kernels. Using CrossSim and architecture simulations, there is an opportunity to optimize at the system-level.

Timeliness or maturity:

The timeliness of the proposal is high; conventional ML systems, up to an including recent large-language-models (LLMs), continue to gain popularity, but their ultimate safety impacts are under-addressed. Meanwhile, the maturity of the current approaches is verging onto full demonstrations (e.g., Technology Readiness Level (TRL) 2 [4],[6], with TRL3 soon achievable as a lab demonstration). This presents a unique opportunity for the DOE ASCR program, which can use the establishment of newer large-scale array and chip prototypes to execute not only standard ML kernels, but novel hardware-friendly UQ approaches. Overall, this presents a unique opportunity for standardization of interpretable ML hardware using emerging analog memory approaches.

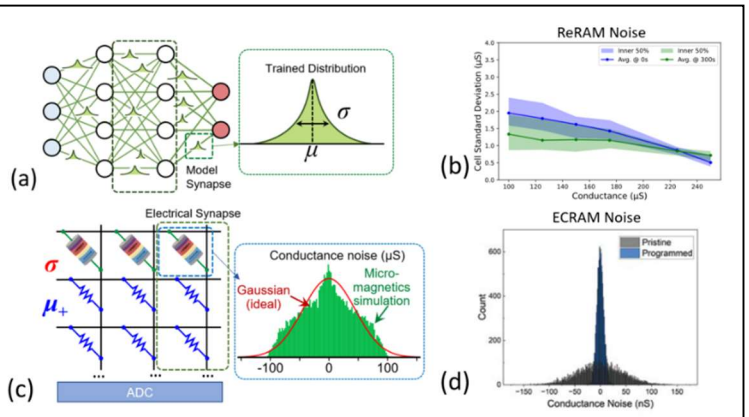


Fig 1. (a) A Bayesian Neural Network; each synapse is a trained/learned distribution (described by a mean and sigma); adapted from [4]. (b) Demonstration of modifiable noise with ReRAM devices, with as-programmed (blue) and relaxed values. (c) Representation of compact Bayesian inference in the AIMC format, with low-noise mean devices interleaved with programmed stochastic synapse, here represented as magnetic tunnel junctions (MTJs); adapted from [4] (d) Demonstration of modifiable noise with ECRAM devices, from [6].

References:

[1] Hüllermeier, Eyke, and Willem Waegeman. "Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods." *Machine learning* 110.3 (2021): 457-506.

[2] Blundell, Charles, et al. "Weight uncertainty in neural network." *International conference on machine learning*. PMLR, 2015.

[3] S. Agarwal *et al.*, "Energy Scaling Advantages of Resistive Memory Crossbar Based Computation and its Application to Sparse Coding," *Frontiers in Neuroscience,* vol. 9, p. 484, 2016, Art no. 484

[4] Liu, Samuel, et al. "Bayesian neural networks using magnetic tunnel junction-based probabilistic in-memory computing." *Frontiers in Nanotechnology* 4 (2022): 1021943.

[5] Bennett, Christopher H., et al. *Probabilistic Nanomagnetic Memories for Uncertain and Robust Machine Learning*. No. SAND2022-13142. Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), 2022

[6] Oh, S., et al. "Bayesian Neural Network Implemented by Dynamically Programmable Noise in Vanadium Oxide." *2023 International Electron Devices Meeting (IEDM)*. IEEE, 2023

[7] D. Wilson *et al.*, "A Discovery Platform to Characterize Emerging Nonvolatile Memories for Computing," in *2024 IEEE 42nd VLSI Test Symposium (VTS)*, 22-24 April 2024 2024, pp. 1-5

[8] S. Agarwal *et al.* "CrossSim." http://cross-sim.sandia.gov (accessed)

Compilation Infrastructure for Chemical Reaction Networks

Nicolas Bohm Agostini[1, *], Antonino Tumeo[1], Connah G. M. Johnson[1], William Cannon[1]

[1]Physical and Computational Sciences Division, Pacific Northwest National Laboratory

*Corresponding author: Nicolas.Agostini@pnnl.gov

Topic: Compilers for Chemical Reaction Networks

Introduction

Applications from Department of Energy (DOE) process an ever increasing amount of data, requiring the fastest and most efficient computational system possible to enable scientific discovery at massive scales [3]. Chemical Reaction Networks (CRNs), a type of analog computing based on mass-action kinetics that can be modeled by Ordinary Differential Equations (ODEs), have been shown to be Turing Complete and capable of performing computations through the interconnected structure of coupled chemical interactions [4,7]. As Moore's law is reaching an end, understanding their applicability and developing methods to enable their adoption in extreme-scale science is critical for realizing the full potential in scientific computing. However, several challenges remain in using CRNs as a general-purpose computing platform, such as decoupling reactions from chemical implementations and providing higher-level programming abstractions. To address these gaps, we highlight the need for a Compilation Framework for Computing with CRNs that aims to enable DOE domain scientists and academic/industrial collaborators to implement complex mathematical equations on a computational physical system.

Challenges

To use CRNs as computing machines, it is necessary to translate computations into sets of chemical reactions, control the reactions, and measure the outputs. Existing approaches, such as CRN++ [9], provide a language and compiler for programming deterministic hypothetical chemical kinetics but do not describe how these hypothetical reactions would be implemented with biochemical compounds or consider practical efficiency, scalability, and uncertainty in the results. Additionally, communication with CRN systems presents another significant challenge as information is transmitted via chemical species interacting in a soluble well-mixed environment, making efficient data encoding and decoding using chemical species and interactions an open problem that needs addressing before practical CRN machines can be built.

Opportunity.

To overcome these challenges and unlock the full potential of CRNs in computation, we highlight the need for compilation frameworks tailored for CRNs. This framework should include end-to-end toolchains similar to those that made neuromorphic systems popular [5,6], which are capable of mapping high-level applications into this new computing paradigm. As such, there is an opportunity to leverage a co-design approach enabled by an expressive Domain Specific Language (DSL), and a state of the art (SOTA) compilation flow. By doing so, we can establish a robust ODE-to-CRN flow instead of the traditional inverse flow of CRN simulation using ODEs.

SOTA compilation frameworks, such as Multi-Level Intermediate Representation (MLIR) [8], have been paving the path for the renascence in compilers for heterogeneous computing. Leveraging recent experience using MLIR in flows for high-level synthesis of digital circuits [1] and in enabling custom accelerators [2], we highlight the opportunity to design expressive abstractions capturing physically realistic chemical and biological processes as operators in this modern compiler infrastructure, allowing conversion from higher-level mathematical representations, such as ODEs, to novel abstractions for CRNs. This will enable researchers to map high-level applications to the computational capacity of chemical/biological operations more effectively while facilitating the study of tradeoffs (execution speed, energy efficiency) and scalability (both in terms of complexity of mapped operations and system size).

Timeliness

The growing interest in unconventional computing paradigms based on analog computations such as neuromorphic systems highlights the need to explore alternative approaches beyond traditional digital architectures. As Moore's Law continues to slow down, researchers are increasingly turning towards analog and hybrid computing solutions that offer potential advantages in terms of energy efficiency, parallelism, and scalability. This shift creates a unique opportunity for analog CRN-based model of computation to gain traction within the broader scientific community and beyond, paving the way for new discoveries and technological breakthroughs across various domains. Furthermore, recent advancements in our understanding of CRNs have provided valuable insights into their computational capabilities, making this an opportune time to invest resources into developing novel tools, techniques, and applications that can fully harness the potential of these systems. By doing so, we may unlock a new era of computing characterized by unprecedented levels of energy efficiency, parallelism, and adaptability, ultimately leading to significant advancements in fields as diverse as artificial intelligence, drug discovery, and climate modeling. In conclusion, new compilers offers a unique opportunity to explore the potential of CRNs in computation by addressing key challenges related to programming abstractions, communication schemes, and developing innovative algorithms tailored specifically for CRN-based computing systems. By leveraging interdisciplinary collaboration and investing in cutting-edge tools and techniques, we can build robust end-to-end toolchains that enable researchers to design and implement practical CRN machines capable of tackling some of today's most pressing computational challenges.

References

[1] Nicolas Bohm Agostini, Serena Curzel, Jeff Jun Zhang, Ankur Limaye, Cheng Tan, Vinay Amatya, Marco Minutoli, Vito Giovanni Castellana, Joseph Manzano, David Brooks, Antonino Tumeo . 2022. Bridging Python to Silicon: The SODA Toolchain. IEEE Micro 42, 5 (2022), 78–88.

[2] Nicolas Bohm Agostini, Jude Haris, Perry Gibson, Malith Jayaweera, Norm Rubin, Antonino Tumeo, José L. Abellán, José Cano, and David Kaeli. 2024. AXI4MLIR: User-Driven Automatic Host Code Generation for Custom AXI-Based Accelerators. In 2024 IEEE/ACM International Symposium on Code Generation and Optimization (CGO'24). IEEE, Edinburgh, United Kingdom, 143–157.

[3] James Ahrens, Amber Boehnlein, Rich Carlson, Joshua Elliot, Kjiersten Fagnan, Nicola Ferrier, Ian Foster, Lee Gimpel, John Shalf, and Dan Ratner. 2022. Envisioning Science in 2050. (6 2022). https://doi.org/10.2172/1871683
[4] Olivier Bournez, Daniel S. Graça, and Amaury Pouly. 2017. Polynomial Time Corresponds to Solutions of Polynomial Ordinary Differential Equations of Polynomial Length. J. ACM 64, 6, Article 38 (oct 2017), 76 pages.

[5] Serena Curzel, Nicolas Bohm Agostini, Shihao Song, Ismet Dagli, Ankur Limaye, Cheng Tan, Marco Minutoli, Vito Giovanni Castellana, Vinay Amatya, Joseph Manzano, Anup Das, Fabrizio Ferrandi, and Antonino Tumeo. 2021. Automated Generation of Integrated Digital and Spiking Neuromorphic Machine Learning Accelerators. In 2021 IEEE/ACM Interna- tional Conference On Computer Aided Design (ICCAD). IEEE, 1–7.

[6] Jason K Eshraghian, Max Ward, Emre Neftci, Xinxin Wang, Gregor Lenz, Girish Dwivedi, Mohammed Bennamoun, Doo Seok Jeong, and Wei D Lu. 2023. Training spiking neural networks using lessons from deep learning. Proc. IEEE 111, 9 (2023), 1016–1054.

[7] François Fages, Guillaume Le Guludec, Olivier Bournez, and Amaury Pouly. 2017. Strong Turing Completeness of Continuous Chemical Reaction Networks and Compilation of Mixed Analog-Digital Programs. In Computational Methods in Systems Biology, Jérôme Feret and Heinz Koeppl (Eds.). Springer International Publishing, Cham, 108–127.

[8] Chris Lattner, Mehdi Amini, Uday Bondhugula, Albert Cohen, Andy Davis, Jacques Pienaar, River Riddle, Tatiana Shpeisman, Nicolas Vasilache, and Oleksandr Zinenko. 2020. MLIR: A Compiler Infrastructure for the End of Moore's Law. arXiv:2002.11054 [cs.PL]

[9] M. Vasić, Soloveichik D., and S. Khurshid. 2020. CRN++: Molecular programming language. Natural Computing. Springer, 19, 19 (2020), 391–407.

Analog in-memory computing for high-throughput scientific experiments and nuclear medicine

Luca Buonanno*, Giacomo Pedretti*, Thomas Van Vaerenbergh*, Yanir London*, Bassem Tossoun*
Stanley Cheung*, Wolfger Peelaers*, Jim Ignowski*, Raymond Beausoleil*, Paolo Faraboschi*
*Hewlett Packard Labs, USA and EU
Email: luca.buonanno@hpe.com

Topic: In-memory computing for high-throughput applications.

Challenge: Identifying the trajectory, or *track*, of a charged particle across multiple detector layers is a crucial component of present and future physics experiments supported by the Department of Energy (DOE) complex [1]. Accepting tracks within a low-latency decision window has become essential to maintain physics acceptances in the face of ever-increasing luminosity. A common approach is to use FPGAs in an iterative process to create *tracklets*, a solution that has super-linear algorithmic complexity. Another solution is to use content-addressable memories (CAMs) to recognize individual sensor addresses on different detector layers and then find tracks from those matched addresses [2]. Based on the reconstructed tracks and other coarse information, the low-level trigger system decides, using locally deployed AI models, on retaining or discarding the event [3]. While the processing of these experimental data is today already a challenge, the data throughput of a pivotal facility such as the Linac Coherent Light Source at SLAC National Accelerator Laboratory is expected to show a 2×10^2-fold increment of the generated data compared to 2020 levels, producing more than 1 TB per second by 2029 [4].

Data processing at high rate and low latency also enables the in-vivo range verification in hadron therapy [5], [6], where the emission points distribution of $10^8\sim10^9$ particles has to be identified in $\leq$1 ms to adjust the hadron beam energy and irradiated spot accordingly. An array of monolithic scintillators coupled to gamma cameras based on the Compton effect or collimators can convert each gamma photon into an image of data collected by the sensors. Regression or classification models are then used to infer the emission point. On the other hand, at such detection rates, it is unfeasible to leverage lightweight AI models at the edge, e.g. small neural networks (NNs) or XGBoost, for consolidating the acquired data. The inference time for AI model in high-end FPGAs for a few tens of 4-bit features is $\geq$ 100 ns for a single decision tree and $\geq$ 1 μs for a single dot product.

Implementing AI pipelines in a massively parallel fashion close to or inside the detectors enables inference on the fly, and therefore to either store only adequately pre-processed data, or control with low-latency a system that requires monitoring a high throughput of data.

Opportunity: In-memory computing (IMC, [7]) is gaining momentum for performing high-throughput machine learning inference with low energy consumption. In particular, thanks to emerging non-volatile and analog programmable memory technologies such as resistive switching memories, it is possible to build a compact and efficient non-volatile analog computing primitive known as Dot Product Engine (DPE), which has been shown able to accelerate multiple ML workloads such as NN inference and PCA [8]–[11]. Hewlett Packard Labs recently invented a CAM capable of storing and searching multi-bit states [12] as complementary computing primitives to DPEs, and invented multiple photonic integrated circuits (PICs) that are functionally a DPE and a ternary CAM, capable of processing input words at the data rate of a photonic link [13], [14].

IMC-based hardware shows multiple orders of magnitude improved performance than solutions based on off-the-shelf GPUs and FPGAs. As an example, an IMC accelerator for tree-based machine learning based on CAMs achieves $\times10^2$ higher throughput and $\times10^4$ lower latency when compared to a GPU, and $\times10^2$ higher throughput and $\times10^3$ lower latency when compared FPGA [15]; the 20 W power envelope allows its integration in hardware closely coupled to the sensors, and therefore the embedding of the required AI models. Furthermore, as an example of how to use these building blocks for accelerating larger models, in [16] the authors discuss the combined use of DPEs and multi-bit CAMs for accelerating transformer models.

The availability of optical ternary CAMs [14] enables the acceleration of tree-based ML in PICs, leveraging an architecture similar to the one described in [15]. Aggregating data coming from spatially distant detectors into a single PIC and accelerating ML inference with an O-CAM has the potential to operate at signal data rate, eliminating electro-optic overhead and resolving speed and latency bottlenecks that exist in current CMOS-based content-addressable memories and FPGA emulations.

The edge processing of high-throughput analog data can be supported by analog IMC having a broad impact on the capabilities of scientific and nuclear medical instruments. In addition, developing high-count rate and low-power radiation monitors is relevant to many homeland security applications, such as the embedding of gamma cameras on unmanned aerial vehicles. Finally, the high costs and long-term planning involved with managing large physics experiments and hadron therapy clinics make them a unique platform for developing revolutionary technologies.

Maturity: Multiple IMC solutions have been proposed and tested in the last decade, but there is a gap that has to be filled between the available prototypes and the advanced IMC hardware that could support the applications here mentioned. Examples of analog IMC prototypes supporting DPE functionalities are the resistive-RAM hardware presented in [9], the phase-change memory-based hardware presented in [17], and the experimental characterization of a PIC-DPE is presented in [18]. The ReRAM-based implementation of a multi-bit CAM is presented in [19], where the CMOS substrate includes all the sensing and routing peripherals, but the ReRAM [20] were in that case integrated into the *back-end of the line* (BEOL); for analog IMC, the most profound challenge remains the scarce availability of foundries that can integrate non-volatile memories close to the transistors. While custom BEOL processing allows for building reliable and durable devices, with tens of tunable states per device, their area footprint forces the use of thick-oxide transistors, limiting integration capabilities and achievable performances. Similarly, there remain several challenges preventing optical NNs from achieving competitive performance when compared to digital NNs, such as the lack of technologies that can monolithically integrate all the necessary components, and the energy dissipation dictated by the transmission of the optical weights, which scales with the size of the ONNs. While silicon photonics are easy to manufacture, they are poor light emitters, and while compound semiconductors enable efficient emitters, they are difficult to scale for complex integrated circuits. Furthermore, to build truly scalable, robust, and energy-efficient PICs for IMC, heterogeneous integration of active optoelectronic devices is essential. Founding synergies between US-based large foundries and projects led by the National Laboratories will lead to the innovation we need to achieve the high-throughput capabilities that the applications here mentioned require, and to the technological development needed to transform the IMC prototypes into hardware ready for mass production.

REFERENCES

[1] J. Ang *et al.*, "Reimagining codesign for advanced scientific computing: Report for the ascr workshop on reimagining codesign," *USDOE Office of Science (SC) Technical Report*, 2022.

[2] J. R. Hoff *et al.*, "Vipraml1cms: A 2-tier 3d architecture for pattern recognition for track finding," *2016 IEEE NSS-MIC*, pp. 1–6, 2016.

[3] D. J. Kösters *et al.*, "Benchmarking energy consumption and latency for neuromorphic computing in condensed matter and particle physics," *APL Machine Learning*, vol. 1, no. 1, 2023.

[4] J. Thayer *et al.*, "Massive scale data analytics at lcls-ii," in *EPJ Web of Conferences*, vol. 295. EDP Sciences, 2024, p. 13002.

[5] S. Rossi, "Hadron therapy achievements and challenges: The cnao experience," *Physics*, vol. 4, no. 1, pp. 229–257, 2022.

[6] E. S. Diffenderfer *et al.*, "Design, implementation, and in vivo validation of a novel proton flash radiation therapy system," *International Journal of Radiation Oncology* Biology* Physics*, vol. 106, no. 2, pp. 440–448, 2020.

[7] D. Ielmini and H.-S. P. Wong, "In-memory computing with resistive switching devices," *Nature Electronics*, vol. 1, no. 6, pp. 333–343, Jun. 2018. [Online]. Available: http://www.nature.com/articles/s41928-018-0092-2

[8] M. Hu *et al.*, "Dot-product engine for neuromorphic computing: Programming 1t1m crossbar to accelerate matrix-vector multiplication," *Proceedings of the 53rd annual design automation conference*, pp. 1–6, 2016.

[9] W. Wan *et al.*, "A compute-in-memory chip based on resistive random-access memory," *Nature*, vol. 608, no. 7923, pp. 504–512, 2022.

[10] M. Le Gallo *et al.*, "A 64-core mixed-signal in-memory compute chip based on phase-change memory for deep neural network inference," *Nature Electronics*, vol. 6, no. 9, pp. 680–693, 2023.

[11] J.-M. Hung *et al.*, "A four-megabit compute-in-memory macro with eight-bit precision based on cmos and resistive random-access memory for ai edge devices," *Nature Electronics*, vol. 4, no. 12, pp. 921–930, 2021.

[12] C. Li *et al.*, "Analog content-addressable memories with memristors," *Nature Communications*, vol. 11, no. 1, p. 1638, Apr. 2020. [Online]. Available: https://www.nature.com/articles/s41467-020-15254-4

[13] X. Xiao *et al.*, "Wavelength-parallel photonic tensor core based on multi-fsr microring resonator crossbar array," in *Optical Fiber Communication Conference*. Optica Publishing Group, 2023, pp. W3G–4.

[14] Y. London *et al.*, "Multiplexing in photonics as a resource for optical ternary content-addressable memory functionality," *Nanophotonics*, vol. 12, no. 22, pp. 4137–4155, 2023.

[15] G. Pedretti, J. Moon *et al.*, "X-time: An in-memory engine for accelerating machine learning on tabular data with cams," *arXiv preprint arXiv:2304.01285*, 2023.

[16] L. Zhao *et al.*, "Race-it: A reconfigurable analog cam-crossbar engine for in-memory transformer acceleration," *arXiv preprint arXiv:2312.06532*, 2023.

[17] S. Ambrogio *et al.*, "An analog-ai chip for energy-efficient speech recognition and transcription," *Nature*, vol. 620, no. 7975, 2023.

[18] W. Zhou *et al.*, "In-memory photonic dot-product engine with electrically programmable weight banks," *Nature Communications*, 2023.

[19] A. Natarajan *et al.*, "Design space exploration of analog cam for tree-based models," *2023 IEEE 66th International MWSCAS*, 2023.

[20] X. Sheng *et al.*, "Low-Conductance and Multilevel CMOS-Integrated Nanoscale Oxide Memristors," *Advanced Electronic Materials*, vol. 5, no. 9, p. 1800876, Sep. 2019. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/aelm.201800876

Probabilistic Computing with Probabilistic Bits: From Physics to Systems

Kerem Çamsarı[1], Luke S. Theogarajan[1], Masoud Mohseni[2]
[1]UC Santa Barbara, camsari@ucsb.edu
[2] LSIP, Hewlett Packard Labs, Milpitas, CA, US

Executive Summary

With the slowing down of Moore's Law, computing has become increasingly domain-specific. Here, we draw attention to probabilistic computing with probabilistic bits (*p-bits*) as an abstraction between bits and qubits. The target application space of p-bits has a strong overlap with that of quantum computing, with significant potential impact on nearly all of computing, ranging from Machine Learning (ML) and Artificial Intelligence (AI) to combinatorial optimization and quantum simulation. The key point we would like to stress is this: the eventual success of this research program critically depends on a coherent inter-disciplinary effort from all parts of the stack, from material science and technology to devices and circuits and to algorithms and systems research. Using a ***Physics-to-Systems*** co-design approach, novel algorithms with enhanced scaling behavior on dedicated probabilistic computers could open up new applications and business models that have not been historically accessible to any other computing paradigm, classical or quantum.

Challenge: A perfect storm in computing. The slowing down of transistor scaling coincided with a revolution in ML and AI, with increasingly unsustainable computational demands. We are in an era of electronics where progress must come from connecting the physics of materials and devices all the way up to systems and architectures. To this end, we believe that a new kind of computer, designed from natively probabilistic building blocks can help solve an enormous variety of computational problems. The probabilistic computers we envision take a great deal of inspiration from Physics, where nature employs microscopic stochastic local dynamics, quasi-sparse graph connectivity, strong heterogeneity, asynchronous updates (without global clocks) in a massively parallel manner leading to many-body emergent phenomena at mesoscopic scales that are computationally rich and hard to simulate with conventional computing paradigms (FIG. 2). Local (as opposed to all-to-all) connectivity is increasingly important in the post-Dennard era, as we are facing an interconnect wall and wire-limited design rules.

Opportunity: Richard Feynman is credited with starting the field of quantum computing with his famous 1981 talk 'Simulating Physics with Computers [1]. What is less well-known is, before getting to quantum computers, Feynman imagines a probabilistic computer with the same idea: "The other way to simulate a probabilistic Nature, ... is by a computer C, which itself is probabilistic, ... in which the output is not a unique function of the input." Inspired by Feynman's vision and to better understand the precise relationship between probabilistic and quantum worlds, a small but steadily growing community has been working on full-stack probabilistic computing research (see, for example, [2–4] and references therein).To build probabilistic computers, one needs probabilistic bits, which are conceptually between bits and qubits (FIG. 1). Unlike bits that are 0 or 1, p-bits fluctuate between 0 and 1, naturally representing probabilities. Just as 'bits' specialize in Graphics and Tensor Processing Units (GPU/TPU) to accelerate ML problems and qubits specialize to many-body quantum physics simulations, many-body interacting p-bits could be engineered to act as native processors for a wide range of probabilistic applications including Monte Carlo algorithms, Bayesian learning and inference, training and inference in energy-based generative models, combinatorial optimization, modeling complex systems and accelerating a subset of quantum simulations.

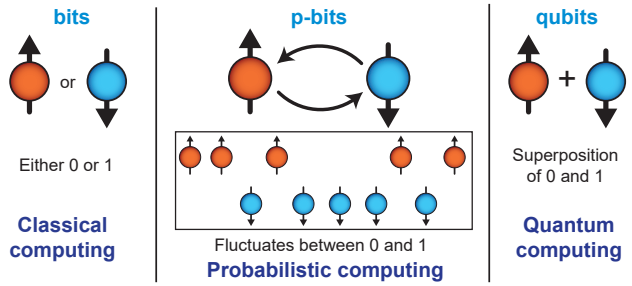


FIG. 1: **bits, p-bits, and qubits**: p-bits are conceptually in between bits and qubits. The application space of p-bits are probabilistic, physics-inspired algorithms pervading all of science and engineering.

Potential of p-bits. Extreme hardware requirements have largely prevented implementation of fault-tolerant quantum computers, even though worldwide efforts are underway. A large part of the motivation to build quantum computers comes from the celebrated Shor's and Grover's algorithms and other quantum simulation algorithms where quantum interference (in a 2^N dimensional Hilbert space) can be creatively engineered to amplify the magnitude of the wavefunction on a non-trivial solution. However, prefactor considerations [5] suggest that in the case of meager speedup (especially from Grover), quantum computers may not outperform digital computers in practice for the foreseeable future. Interestingly, probabilistic bits *also* operate in a 2^N dimensional space, where N p-bits represent a sample from a $2^N \times 1$ dimensional probability density function. One could argue that significant scaling advantages for probabilistic algorithms cannot survive as long term targets since such breakthroughs are always adopted as part of ''conventional" computing. However this often comes at a significant cost in physical resources, latency, and power consumption, which makes such remedies unsustainable in the

long run. Intriguingly, the potential probabilistic analogs of quantum variational algorithms such as Quantum Approximate Optimization Algorithm (QAOA) have been proposed [6] and the true potential of these algorithms remains largely unknown. Powerful ML methods combined with human ingenuity may usher in breakthrough algorithms with monumental implications for computing may be around the corner (see some of the emerging attempts on non-equilibrium Monte Carlo algorithms [7] beyond traditional equilibrium heuristics with annealing and tempering schedules). The true power of probabilistic computers versus quantum computers remains unknown, as codified by the unresolved $\mathrm{BPP} \stackrel{?}{=} \mathrm{BQP}$ question in theoretical computer science.

Timeliness: Why now? The power of randomness in computational algorithms has long been recognized by computer scientists. Monte Carlo algorithms (popularized by Ulam and von Neumann and then put on a firm footing by Metropolis and Teller) have been recognized as one of the top 10 algorithms of the 20th century by the ACM. So why have we not capitalized on probabilistic algorithms for the most critical AI workloads? This stems from having to emulate probabilistic behavior on deterministic Turing machines, which can be very expensive: low-level hardware analysis indicates that a high-quality tunable random number generator requires up to 15,000 transistors *per node* [8]. In addition, Markov Chain Monte Carlo algorithms are fundamentally serial: as the "chain" in the name implies, the next step in the computation critically depends on the previous one, preventing easy parallelization. In general, probabilistic algorithms and randomness are often desired yet remain computationally expensive [9].

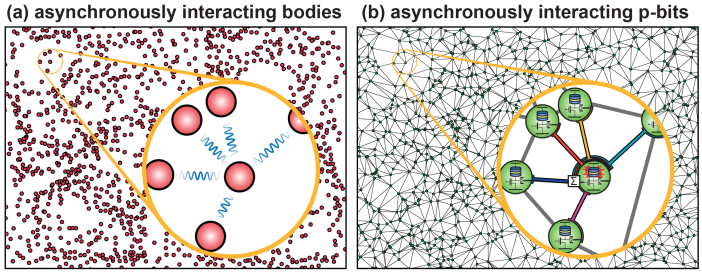


FIG. 2: **Physical analogy** between asynchronously interacting (a) bodies and (b) p-bits: both systems are asynchronous, locally interacting with sparse connectivity and massively parallel.

Synergy between the digital and the analog. The intrinsic noise found in analog nanodevices can lead to compact implementations of p-bits with far better energy-efficiency and scalability. Not all analog sources of noise are appropriate, however. The source of noise in analog devices must be large enough (compared to $k_B T/q$) so that it can be harnessed by surrounding circuitry without requiring power-hungry amplifiers and other peripherals. In our view, some of the most promising analog noise sources have been demonstrated in natural 'noise amplifiers' such as stochastic magnetic tunnel junctions [10] and single photon avalanche diodes [11], though other options may exist. In particular, projections indicate that 1 to 10 million bit p-computers, using magnetic tunnel junctions (MTJ) scaled up to such densities by the magnetic memory industry, could provide up to 100,000X speedup in performance on probabilistic applications, even *without considering* algorithmic benefits enabled by such dedicated p-computers. We believe a large opportunity is left at the table without a thorough investigation of probabilistic computation from ***Physics-to-Systems***.

Outlook We believe physics-inspired probabilistic computers realized with analog stochastic building blocks hold great potential for a wide range of applications from optimization, AI and quantum simulation. In addition to hardware, an important piece of the puzzle is the development of new, beyond equilibrium Monte Carlo algorithms [6, 7]. For near-term success, we believe in practical combinations of analog-digital hybrid computers, using analog computation *when necessary* to replace key functionalities difficult to achieve with digital computers. This approach leverages the strengths of each domain using the best possible combination of analog and digital hardware. In addition, new algorithms research on such machines may lead to qualitatively different behavior compared to the "business as usual" in the traditional deep learning revolution.

[1] R. P. Feynman, "Simulating physics with computers," *International journal of theoretical physics*, vol. 21, no. 6-7, pp. 467–488, 1982, [PDF].

[2] N. A. Aadit, A. Grimaldi, M. Carpentieri, L. **Theogarajan**, J. M. Martinis, G. Finocchio, and K. Y. **Camsari**, "Massively Parallel Probabilistic Computing with Sparse Ising Machines," *Nature Electronics*, vol. 5, no. 7, pp. 460–468, 2022, [PDF].

[3] S. Chowdhury, **M. Mohseni**, K. Y. **Camsari** *et al.*, "A Full-stack View of Probabilistic Computing with p-bits: Devices, Architectures and Algorithms," *IEEE Journal on Exploratory Solid-State Computational Devices and Circuits*, 2023, [PDF].

[4] S. Niazi, S. Chowdhury, N. A. Aadit, **M. Mohseni**, Y. Qin, and K. Y. **Camsari**, "Training deep boltzmann networks with sparse ising machines," *Nature Electronics*, pp. 1–10, 2024, [PDF].

[5] R. Babbush, J. R. McClean, M. Newman, C. Gidney, S. Boixo, and H. Neven, "Focus beyond quadratic speedups for error-corrected quantum advantage," *PRX quantum*, vol. 2, no. 1, p. 010103, 2021, [PDF].

[6] G. Weitz, L. Pira, C. Ferrie, and J. Combes, "Sub-universal variational circuits for combinatorial optimization problems," *arXiv preprint arXiv:2308.14981*, 2023, [PDF].

[7] M. **M. Mohseni**, D. Eppens, J. Strumpfer, R. Marino, V. Denchev, A. K. Ho, S. V. Isakov, S. Boixo, F. Ricci-Tersenghi, and H. Neven, "Nonequilibrium monte carlo for unfreezing variables in hard combinatorial optimization," *arXiv preprint arXiv:2111.13628*, 2021, [PDF].

[8] N. S. Singh, K. Kobayashi, Q. Cao, S. Kanai, H. Ohno, S. Fukami, and **Camsari** K. Y., "Cmos plus stochastic nanomagnets enabling heterogeneous computers for probabilistic inference and learning," *Nature Communications*, vol. 15, no. 1, p. 2685, 2024, [PDF].

[9] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks," *Advances in neural information processing systems*, vol. 29, 2016, [PDF].

[10] W. A. Borders, K. Y. **Camsari** *et al.*, "Integer Factorization Using Stochastic Magnetic Tunnel Junctions," *Nature*, 2019, [PDF].

[11] W. Whitehead, Z. Nelson, K. Y. **Camsari**, and L. **Theogarajan**, "Cmos-compatible ising and potts annealing using single-photon avalanche diodes," *Nature Electronics*, vol. 6, no. 12, pp. 1009–1019, 2023, [PDF].

The Prospects for Biological Computers

William Cannon[1], Connah G. M. Johnson[1] , Nicolas Bohm Agostini[1], Chris Oehmen [2] and Antonino Tumeo[1]

[1]Physical and Computational Sciences Directorate, [2]Earth and Biological Sciences Directorate, Pacific Northwest National Laboratory

Topic: Analog Biological and Biochemical Computation

Relevance: This paper concerns the use of biochemical reaction and signaling networks, and cells, for computing in analogy to any process that involves an optimization, regression, or time entrainment, including human (brain) computing and electronic computing. The quantities involved in the computations may be continuous real-valued, discrete, or a mixture.

Challenge – *Applications and Opportunities for Analog Computation*. Biological processes excel at solving optimization and regression challenges and challenges involving frequency/temporal encoding. Biochemical reaction networks, much like chemical reaction networks (CRNs), can be constructed to solve ordinary differential equations using a regression approach [1]. Trivially, if a biological CRN faithfully represents a set of complex ODEs, the cell can solve the problem. More generally, what is required is that the biological CRN be mapped or controlled to represent the set of complex ODEs. Any problem that can be compiled by a sensing/input mechanism that transforms the problem into a chemical signal can potentially be solved by biocomputation. While optimization, regression and temporal encoding fall into this domain, biocomputation will likely not make impacts in areas such as sending emails, reading documents, or browsing the internet. In other words, scientific computing may be the best domain for biocomputation.

Towards this goal of control, biological processes embedded in devices are currently being developed in which the platforms aim to solve computational optimization challenges. Current devices range from catalytic enzymes embedded in porous media to brain organoids attached to microelectronic arrays, for which a start-up company deploying this technology is already in place (finalspark.com). The company provides continuous remote access to the brain organoid platform, a python API, real-time neural input stimulation and output measurement and input/output data storage. Other promising new technologies include the 3D printing of redox-inducing microbes onto microelectronic arrays (MEAs), which may offer an additional level of control over the structure of biological reaction networks involved in biocomputation. The latter approach is attractive because it may combine the use of MEAs with synthetic biology to achieve cellular supremacy in computing [2]. Even more control is possible using structured arrays of immobilized enzymes laid out in microfluidic platforms [3]. The technology for immobilizing enzymes in microporous media results in long lasting catalysts (enzymes) [4], opening the door to the use of the thousands of small molecule chemical reactions from metabolism in a controlled manner.

Importantly, developing biological computers may require rethinking what it means to be a computer, in that while computing, biological processes also are likely to be performing beneficial services and tasks.

Opportunities – *Motivation*. Biological and biochemical computing technologies have the potential for enormous energy and power advantages over current digital computers. While recent feats of exascale computation and large language models are impressive, the power consumed in such feats is equally impressive – and alarming. Training one large ML model can be equivalent to driving 242,231 miles in an average automobile [5], and a typical supercomputer used for the training can easily consume 1-10 megawatts. For comparison, the human brain operates on ~20 watts. Due to the approximately 86 billion neurons with over 10^{15} connections, the storage capacity of the brain is ~2.5 PB [6]. Even more impressive, a microbial cell operates on ~2.4 x 10^{-16} watts (unpublished data) and produces as much power/weight as hydrogen fuel cells, which are the most powerful fuel cells per weight known.

Consequently, biocomputers may have a unique advantage for learning. A key insight into any kind of learning is that rapid learning requires reducing the costs to adapt the relevant network to learn the solution to the computational challenge. These learning networks are either chemical signaling networks or chemical reaction networks (CRNs).

How these chemical networks are exploited for computing may differ depending on the cell type. In the case of brain organoids, the CRN is the synaptic network of forebrain organoids. As mentioned above, one company has already made available for researchers forebrain organoids interfaced to electrical circuits that stimulate neurons due to programmed spiking [7]. In the case of printed microbial biofilms, the internal chemical reaction networks of microbes can be coupled across cells in analogy to deep neural networks. Each microbe having an internal computing layer coupled to other microbes through metabolic exchange reactions.

How can fast and efficient learning be realized? Laboratory-directed adaptation is a concept analogous to laboratory-directed evolution, but instead is adaptation of cellular circuits without genome modification and on a shorter timescale. Still, adaptation is only as fast as growth. In neurons, the relevant growth is known as neuroplasticity. Neuroplasticity involves creating new neurons, new connections or synapses, or modifying existing synaptic connections. Neural growth takes the longest time, while modifying existing connections takes the least time. The key to quick learning is fast and energy efficient adaption or growth.

In this regard, bacteria are among the fastest growing biological tissues. Research to develop and implement microbial neurons could result in designed microbial communities that can quickly learn to carry out highly energy efficient computations. Furthermore, microbial communities can now be fabricated using 3D printing [8] and lithography [9]. This adds another layer of control beyond what is capable with organoids.

Timeliness - *Why the time is right*. Currently several technologies are maturing and converging to make the prospect of biological computing greater than ever. Microelectrode arrays (MEAs) have been developed that organoids, culture tissues and biofilms can be grown on that provide an electrical input/output interface to biological processes. APIs have been developed for a few of these MEAs and have even allowed researchers 24/7 access to MEA platforms with organoids.

Likewise, the development of 3D printing of biofilms on anodes and cathodes have paralleled organoid development [8], which will allow researchers to control the redox processes that control bacterial metabolism at will. At the same time, synthetic biology is maturing and promises to allow the design of cellular circuits at will. Indeed, the molecular components of cells are being compiled into building blocks that mimic electrical components [10].

The engineering of cells however requires a physics or engineering based approach to biology. In this regard, recent developments in the statistical physics of chemical reaction and signaling networks have allowed researchers to predict the operating principles of cells [11], predict regulation [12] and to even predict rate parameters for the cellular circuits involved [11].

Taken together, these developments indicate that we are at the beginning of a new phase in biocomputing.

References

[1] M. G. Baltussen *et al.*, "Chemical Reservoir Computation in a Self-Organizing Reaction Network," *Nature,* vol. 631, no. 8021, pp. 549-555, Jul 2024, doi: 10.1038/s41586-024-07567-x.

[2] L. Grozinger *et al.*, "Pathways to Cellular Supremacy in Biocomputing," *Nat Commun,* vol. 10, no. 1, p. 5250, Nov 20 2019, doi: 10.1038/s41467-019-13232-z.

[3] P. D. Patil *et al.*, "Microfluidic Based Continuous Enzyme Immobilization: A Comprehensive Review," (in English), *Int J Biol Macromol,* vol. 253, Dec 31 2023, doi: 10.1016/j.ijbiomac.2023.127358.

[4] H. Wu *et al.*, "Enhanced Stability of Catalase Covalently Immobilized on Functionalized Titania Submicrospheres," *Mater Sci Eng C Mater Biol Appl,* vol. 33, no. 3, pp. 1438-45, Apr 1 2013, doi: 10.1016/j.msec.2012.12.048.

[5] C.-J. Wu *et al.*, "Sustainable Ai: Environmental Implications, Challenges and Opportunities," *ArXiv,* vol. abs/2111.00364, 2021.

[6] P. Reber, "What Is the Memory Capacity of the Human Brain?," *Scientific American Mind,* vol. 21, no. 2, p. 70, May 2010, doi: 10.1038/scientificamericanmind0510-70.

[7] F. D. Jordan *et al.*, "Open and Remotely Accessible Neuroplatform for Research in Wetware Computing," (in English), *Frontiers in Artificial Intelligence,* Technology and Code vol. 7, 2024-May-02 2024, doi: 10.3389/frai.2024.1376042.

[8] E. Lazarus *et al.*, "Three Dimensional Printed Biofilms: Fabrication, Design and Future Biomedical and Environmental Applications," *Microbial biotechnology,* vol. 17, no. 1, p. e14360, Jan 2024, doi: 10.1111/1751-7915.14360.

[9] X. Jin *et al.*, "Biofilm Lithography Enables High-Resolution Cell Patterning Via Optogenetic Adhesin Expression," *Proc Natl Acad Sci U S A,* vol. 115, no. 14, pp. 3698-3703, Apr 3 2018, doi: 10.1073/pnas.1720676115.

[10] J. T. Atkinson *et al.*, "Living Electronics: A Catalogue of Engineered Living Electronic Components," *Microbial biotechnology,* vol. 16, no. 3, pp. 507-533, Mar 2023, doi: 10.1111/1751-7915.14171.

[11] W. R. Cannon *et al.*, "Probabilistic and Maximum Entropy Modeling of Chemical Reaction Systems: Characteristics and Comparisons to Mass Action Kinetic Models," *J Chem Phys,* vol. 160, no. 21, Jun 7 2024, doi: 10.1063/5.0180417.

[12] S. Britton *et al.*, "Enzyme Activities Predicted by Metabolite Concentrations and Solvent Capacity in the Cell," *J R Soc Interface,* vol. 17, no. 171, p. 20200656, Oct 2020, doi: 10.1098/rsif.2020.0656.

Analog In-Memory Training for Physical Neural Networks

Douglas Durian, Univ. Pennsylvania - Physics & Astronomy, djdurian@physics.upenn.edu
Andrea Liu, Univ. Pennsylvania - Physics & Astronomy, ajliu@physics.upenn.edu
Marc Miskin, Univ. Pennsylvania - Electrical & Systems Engineering, mmiskin@seas.upenn.edu

Topics:

- physical neural networks
- in-memory analog computing
- in-memory analog training
- integration of training and inference

Challenge: Analog physical neural networks (PNNs) that can perform in-memory compute, such as memristor crossbar arrays and photonic networks, have long been recognized as faster and far more energy efficient per unit computation than digital artificial neural networks (ANNs). However, the ANNs of today vastly outperform PNNs in terms of complex functionality thanks to the power of GPUs to perform backpropagation for the supervised training of very large networks. One difficulty with scaling PNNs up to similar size is that they are trained *ex situ* using backprop on a digital computer simulation model of the physical system, and the learning parameters are fed into the PNN after the training iterations have converged. In this approach there is an inevitable reality gap between the computer model and the actual PNN, errors from which compound with system size. Additional errors can accrue because, for example, individual memristors cannot be set very precisely to the desired conductances. Furthermore, physical systems can degrade with time and necessitate periodic retraining. Thus, an important grand challenge is to develop new training methods tailored specially to PNNs that will accelerate their scalability and functionality, much as GPUs and backprop have done for ANNs. Just as the advantages of PNNs for inference originate from analog in-memory compute, advantages should be sought from the development of analog in-memory training tailored to PNNs [1,2].

Opportunity: The envisioned analog computing systems are not just physical networks with inputs/outputs and adjustable parameters, but physical systems where the inference and training components are fully and inseparably integrated in individual repeat units – just as for neurons in the brain or for the spiking neurons in neuromorphic computing chips. The key feature to emulate is that neurons, real and microfabricated, learn by *local rules*. In other words, neurons update without global knowledge of the parameter values of other neurons. Thus the idea is to supplement PNNs such as memristive arrays with special-purpose circuitry that performs local learning. The development of appropriate local rules and ways to integrate them into physical systems will open a host of opportunities for advances in analog computing. By construction, this sidesteps the reality gap between PNNs and simulation models as well as the need to externally feed each parameter -usually imperfectly- from a computer into the physical network. This should enable the training of larger-scale PNNs, and harden them against both manufacturing defects as well as degradation over time. Analog in-memory training should also enable adaptability to changing conditions for edge computing in the field since external memory and digital processing r are no longer needed for training updates. And, being analog, it will of course enable greater energy efficiencies for training as well as inference in comparison with ANNs.

This is no pipedream: Small laboratory networks with in-memory analog training for in-memory analog computation have been successfully demonstrated (next section and Refs. [3–6]). While these use a particular *supervised* local learning rule [7], further opportunities would arise by developing *unsupervised* local learning rules for integration into PNNs. In all cases, specific research is needed to address the comparative merits of different rules for different PNNs with regards to scalability, noise, optimal network

size and architecture for given tasks, the range of possible tasks, the need for nonlinearity (*c.f.* the activation function in ANNs), and more.

Timeliness: In-memory analog training is on an accelerating upward trajectory. An important impetus was provided by the advent of "Equilibrium Propagation" in 2017 [8,9] and "Coupled Learning" in 2021 [7]. In these closely-related supervised learning schemes, local update rules are derived from gradient descent on the contrast of behavior for two equilibrium states: under ordinary "free" operation in which inputs are applied and outputs are left free, and under "clamped" conditions where labels are imposed as inputs applied to the output nodes. Both have been simulated at fairly large scale [10], and appear capable of universal approximation [11]. Notably, several small scale implementations of Coupled Learning have been demonstrated in the lab using a twin network method, where two identical networks are run under the contrastive conditions [3–6]. However, none of these approaches have yet been physically implemented at scale. In addition, several other local learning rules have also been proposed and simulated [12–15]. Clearly the time is ripe for rapid advances in both theory and experiment for analog in-memory learning for PNNs. Certainly this will lead to niche analog computing applications where low power and adaptability are required, such as sensors and cameras deployed in the field. It remains to be seen the scale to which such approaches can compete with ANNs for AI, but successes will bring enormous advantages in speed, energy, robustness, and adaptability.

References:

[1] M. Stern and A. Murugan, *Learning Without Neurons in Physical Systems*, Annu. Rev. Condens. Matter Phys. **14**, 417 (2023).

[2] A. Momeni et al., *Training of Physical Neural Networks*, arXiv:2406.03372.

[3] S. Dillavou, M. Stern, A. J. Liu, and D. J. Durian, *Demonstration of Decentralized Physics-Driven Learning*, Phys. Rev. Appl. **18**, 014040 (2022).

[4] J. F. Wycoff, S. Dillavou, M. Stern, A. J. Liu, and D. J. Durian, *Desynchronous Learning in a Physics-Driven Learning Network*, J. Chem. Phys. **156**, 144903 (2022).

[5] M. Stern, S. Dillavou, D. Jayaraman, D. J. Durian, and A. J. Liu, *Training Self-Learning Circuits for Power-Efficient Solutions*, APL Mach. Learn. **2**, 016114 (2024).

[6] S. Dillavou, B. D. Beyer, M. Stern, A. J. Liu, M. Z. Miskin, and D. J. Durian, *Machine Learning without a Processor: Emergent Learning in a Nonlinear Analog Network*, Proc. Natl. Acad. Sci. **121**, e2319718121 (2024).

[7] M. Stern, D. Hexner, J. W. Rocks, and A. J. Liu, *Supervised Learning in Physical Networks: From Machine Learning to Learning Machines*, Phys. Rev. X **11**, 021045 (2021).

[8] B. Scellier and Y. Bengio, *Equilibrium Propagation: Bridging the Gap between Energy-Based Models and Backpropagation*, Front. Comput. Neurosci. **11**, 24 (2017).

[9] J. Kendall, R. Pantone, K. Manickavasagam, Y. Bengio, and B. Scellier, *Training End-to-End Analog Neural Networks with Equilibrium Propagation*, arXiv:2006.01981.

[10] B. Scellier, M. Ernoult, J. Kendall, and S. Kumar, *Energy-Based Learning Algorithms for Analog Computing: A Comparative Study*, (2023).

[11] B. Scellier and S. Mishra, *A Universal Approximation Theorem for Nonlinear Resistive Networks*, arXiv:2312.15063.

[12] S. Yi, J. D. Kendall, R. S. Williams, and S. Kumar, *Activity-Difference Training of Deep Neural Networks Using Memristor Crossbars*, Nat. Electron. (2022).

[13] V. R. Anisetti, B. Scellier, and J. M. Schwarz, *Learning by Non-Interfering Feedback Chemical Signaling in Physical Networks*, Phys. Rev. Res. **5**, 023024 (2023).

[14] M. J. Falk, J. Wu, A. Matthews, V. Sachdeva, N. Pashine, M. L. Gardel, S. R. Nagel, and A. Murugan, *Learning to Learn by Using Nonequilibrium Training Protocols for Adaptable Materials*, Proc. Natl. Acad. Sci. **120**, e2219558120 (2023).

[15] V. R. Anisetti, A. Kandala, B. Scellier, and J. M. Schwarz, *Frequency Propagation: Multimechanism Learning in Nonlinear Physical Networks*, Neural Comput. **36**, 596 (2024).

Synthetic Neurocomputers: Advancing Scientific Research through Computing with Living Neurons

Austin R. Ellis-Mohr, Mattia Gazzola, Lav R. Varshney
University of Illinois Urbana-Champaign; {austine4, mgazzola, varshney}@illinois.edu

Topics: Analog Brain-Inspired Computing; Chemical/Biochemical Computation

1. Introduction

Synthetic neurcomputers leverage modern engineering progress and evolutionary design through an *in vitro* neural computing system. This unique interface between biological and silicon computing opens the door to novel scientific advancements.

Analog computing, tracing back to Vannevar Bush's differential analyzer, has historically relied on linear computing components. However, the introduction of nonlinear dynamics—which is naturally implemented in neurobiology with remarkable energy efficiency—can significantly enhance computational capabilities in support of scientific discoveries and solving complex problems in dynamical systems. Additionally, biological neural networks offer collocated processing and memory, parallel processing capabilities, and robustness to damage and noise, making them an untapped resource for advanced information processing. They can not only mitigate noise concerns, a burden of analog computing, but also leverage it to their advantage [1], enabling them to perform complex computations under noisy conditions with high accuracy.

Neurocomputers are designed using living neurons that can be cultured *in vitro* through various methods, including dissociated primary neuron cultures [2–4], organotypic brain slices [4], stem-cell derived neurons [3–5], and brain organoids [5,6]. Neural networks can be stimulated chemically, optogenetically, electrophysiologically, or mechanically, and their activity measured using electrophysiology or microscopy. These cultures can be engineered into structured 2D [2–4] and 3D networks with diverse topologies (e.g., rods, cubes, toroids, spheroids) leveraging modern biofabrication techniques [7]. These systems can include various neuronal cell types and glial cells, such as astrocytes, at different concentrations and locations.

Computationally, synthetic neurocomputers have evolved from performing simple logical operations [2] to playing pong [3], performing speech recognition, and predicting nonlinear equations [6], albeit with limited performance. Yet, even at their nascent design stage, neurocomputers have been shown to outperform conventional state-of-the-art (SOTA) AI in terms of sample efficiency [8]. This is particularly notable given that the brain is predicted to be over five orders of magnitude more power efficient than conventional AI hardware running a comparable neural network [6]. These systems are now versatile, open-source [4], and remotely accessible [5], but there is considerable room for growth.

2. Potential Contributions

Expanding on these foundational developments and the versatile applications of synthetic neurocomputers, a myriad of potential contributions can further revolutionize scientific research and computational technologies.

These systems can be modularized and scaled into functional building blocks to improve engineering design. Techniques exist to selectively route information from different units using macroscopic or microscopic structures, in addition to dynamically altering activity patterns through neurons' inherent plasticity, either through selective stimulation or innate reorganization [9]. Individual components should be defined and characterized to allow for more complex system design. For example, structural mapping using connectomics [10] and microscopy [3,11] can provide better analysis methods. With efforts in scaling and modularization [4,5] across hardware, software, and wetware (i.e., biological components), the barrier to entry will continue to decrease and more researchers will be able to remotely or directly access these systems.

It is important to establish benchmarks for neurocomputers. These standards can include cognitive tests such as the Go/No-Go and N-back tasks, as well as standard computational benchmarks like MNIST for computer vision and GLUE for natural language processing. Additionally, tests akin to SPEC benchmarks could be developed for evaluating general-purpose computing performance.

Integrating neurocomputers into existing brain-on-a-chip research [12] is crucial and has already begun [3,13,14], as both fields can benefit significantly from this consolidation. In the future, adjusting various conditions while performing learning and computational tasks can break new ground in understanding and addressing cognitive disease states or enhancements. Conversely, the customization of culturing (e.g., cell concentrations) and interfacing techniques (e.g., flexible 3D micro-electrode arrays), combined with improved architectural principles (e.g., multi-scale designs) and advanced analysis techniques (e.g., information flow),

promises to unlock new computational capabilities. This further allows for unparalleled control and precision in addressing neuroscientific questions. These synergistic advancements will mutually enhance both fields, driving progress in understanding and utilizing neurocomputers.

Computation can be performed through reservoir computing[6] or other innovative paradigms. Higher-order programming considerations, such as languages (e.g., assembly calculus[15]) and compilers, should be developed and tested, while efforts to understand theoretical constraints, guarantees, and limitations are undertaken concurrently. These insights can also be applied to other neural architectures and computational technologies. Neurocomputers may even provide a pathway to artificial general intelligence (AGI).

Concerted efforts should be made to advance culturing and biofabrication techniques for ease, longevity, and capacity; to engineer more affordable, high-density, and flexible stimulation and recording[16] techniques in a low-power, online setting; and to broadly explore architectural and algorithmic principles.

3. Strategic Directions

Beyond any individual contribution, it is crucial to strategically guide the field's development to ensure broad impact and sustainable growth.

Interdisciplinary collaboration—focusing on symbiotic research and development—should be central, as neurocomputers, like conventional computers, require the wide-ranging expertise of individuals (e.g., neuroscientists, computer scientists, materials scientists) while providing cross-domain benefits. Emphasis should be placed upon developing general open-source techniques and broader remote-access pathways, coupled with improved user interfaces, to expand the pool of researchers advancing the field. Outreach and educational opportunities should be implemented to engage and inform a wider audience and future workforce.

Maturation of synthetic neurocomputers will occur as they continue to be scaled through modular engineering principles. System longevity, reliability, and self-sustainability should remain major foci to expand potential use cases. Rigorous testing of theories, models, and technologies is needed to establish the validity and reliability of neurocomputers. Open data and resource sharing should be prioritized to facilitate this process. As the field matures, interdisciplinary work should continue but also transform to allow for specialization within sub-domains to accelerate future progress.

Given the unique considerations of working with living biological materials, a strong focus on bioethics and social reception should be maintained to ensure safe and responsible development.

In the long term, broader commercialization of more developed and sustainable systems will help reduce costs and advance the field. This should include identifying potential markets and applications for neurocomputers (e.g., neuroscience research, brain machine interfaces, scientific computing) under regulatory compliance and limitations. Moreover, forming strategic partnerships within industry and governmental agencies will aid in accelerating the development of neurocomputers.

4. Conclusion

Key challenges and opportunities in the field of synthetic neurocomputing have been highlighted, with a particular focus on their application in scientific research. By leveraging the complex dynamical processes inherent in biological neurons, synthetic neurocomputers have the potential to tackle a wide range of scientific questions that require sophisticated information processing. Additionally, these systems can provide valuable insights and inspiration for developing future computational technologies.

The promise of synthetic neurocomputers in advancing scientific research is immense, offering new tools and methodologies to further our understanding of the biological, computational, and general sciences.

References

[1] A. A. Faisal, L. P. Selen, and D. M. Wolpert, Nature reviews neuroscience **9**, 292 (2008).
[2] O. Feinerman *et al.*, Nat. Phys. **4**, 967 (2008).
[3] B. J. Kagan *et al.*, Neuron **110**, 3952 (2022).
[4] X. Zhang *et al.*, Advanced Science **11**, 2306826 (2024).
[5] F. D. Jordan *et al.*, Frontiers in Artificial Intelligence **7**, 1376042 (2024).
[6] H. Cai *et al.*, Nature Electronics **6**, 1032 (2023).
[7] G. J. Pagan-Diaz *et al.*, Proc. Natl. Acad. Sci. U.S.A. **116**, 25932 (2019).
[8] F. Habibollahi *et al.*, in *Deep Reinforcement Learning Workshop NeurIPS 2022* (2022).
[9] A. R. Ellis-Mohr and L. R. Varshney, in *Proc. IEEE Int. Symp. Inf. Theory Workshops* (2024).
[10] A. K. Fletcher, S. Rangan, L. R. Varshney, and A. Bhargava, Advances in neural information processing systems **24** (2011).
[11] J. W. Lichtman and J.-A. Conchello, Nature Methods **2**, 910 (2005).
[12] B. Servais *et al.*, Nature Reviews Bioengineering , 1 (2024).
[13] F. Habibollahi *et al.*, Neural Comput. **14**, 5287 (2023).
[14] K.-Y. Huang *et al.*, Proceedings of the National Academy of Sciences **121**, e2313590121 (2024).
[15] C. H. Papadimitriou *et al.*, Proc. Natl. Acad. Sci. U.S.A. **117**, 14464 (2020).
[16] B. D. He *et al.*, Journal of neurophysiology **114**, 746 (2015).

Accuracy Benchmarking for Analog Computing Systems

Ben Feinberg[1] (bfeinbe@sandia.gov), T. Patrick Xiao[1], Christopher Bennett[1], Sapan Agarwal[1]
[1]Sandia National Laboratories

Topic: Benchmarking, in-memory analog computing

Analog accelerators have been widely proposed as a potential solution to continue scaling performance and energy-efficiency in a post-Dennard scaling world. Analog matrix vector multiplication (MVM) has in particular been suggested due to the wide use of MVMs and the natural synergy with recent research interest and development of dense CMOS-compatible, multi-bit memory devices. By taking advantage of basic circuit properties these systems can both reduce costly data movement by computing directly on data stored in a memory array and can approximate many operations with single circuit components. Based on these properties, systems using analog MVM and similar processing-using-memory (PUM) or *in situ* computing have the potential for multiple-order-of-magnitude improvements over conventional digital systems [1].

Challenge: *Demonstrating that analog systems can achieve sufficient accuracy for target applications.*

During this recent resurgence in interest around analog computing, the primary challenge from earlier incarnations remains—accuracy. Recent work has focused heavily on neural networks which are known to be tolerant of imprecision, but multiple studies have demonstrated that large scale neural network inference tasks analog MVM accelerators struggle to reach critical accuracy[1] targets for these problems [2,3]. Moreover, evaluating the accuracy of applications run on analog systems is complicated due to the interplay of numerous factors across the stack from circuits and devices to algorithms. We identify four key challenges that must be addressed to demonstrate that analog hardware can achieve sufficient accuracy:

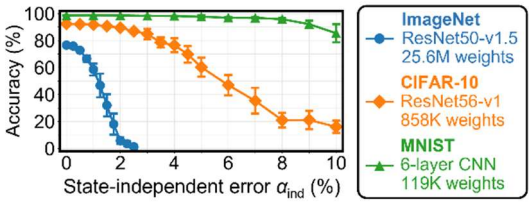


Figure 1-Figure 1: Comparison of accuracy sensitivity to cell errors of different neural network tasks. Reproduced from [2].

1) Problem size and complexity: many works on analog accelerators focus on benchmarking on smaller application problems which can have significantly less sensitivity to analog imprecision. Figure 1 shows how neural network accuracy on an image classification benchmark is affected by different levels of cell errors on three different pairs of dataset and neural networks. Notably, MNIST hand-written digits classification shows almost no accuracy loss from errors until significant (>10%) cell errors, whereas the more complex tasks show substantial accuracy loss at far less accuracy. This means, analysis that focuses on MNIST or even CIFAR-10 can be fundamentally misleading about the efficacity or necessity of various optimizations. This focus on smaller applications is understandable given the challenges of building highly scaled accelerators, but simulation studies can augment smaller-scale characterization to help perform large-scale analysis.
2) Unconsidered accuracy tradeoffs for baseline systems: in computer systems we often use the concept of a Pareto frontier to help analyze performance vs energy tradeoffs (*e.g.*, a slower system can often be lower power and vice versa). When analyzing analog accelerators, papers frequently make claims about finding a better performance for a given power budget (or vice versa) while acknowledging that the analog system cannot hit the exact same accuracy as the digital baselines. When considering systems which cannot provide the same level of accuracy, we are effectively adding a third dimension to the Pareto frontier which must be considered in both the proposed and baseline systems. For instance, in a neural network task if an analog system provides 3% lower accuracy than the comparison system, the relevant baseline is a system that also provides 3% lower accuracy through algorithmic simplifications (*e.g.,* quantization).
3) Unconsidered variables in accuracy assessments: assessing the sources of error in analog accelerators can be a challenge, especially when analyzing large scale applications. Instead, many works use first principles-based arguments about why certain sources of error can be ignored. Assessments to skip certain sources of error must be carefully evaluated as if incorrect can lead to misleading conclusions and irrelevant optimizations.

[1] Here, accuracy is a general term describing the quality of the output of a system as compared to digital. Although this term is primarily used in the context of neural network classification, it also refers to other metrics of solution quality such as signal to noise ratio (SNR) for signal processing applications.

4) Operation-level vs end-to-end analysis: due to the complexity of analyzing the myriad sources of error, many proposals attempt to achieve the same results on a per-analog-operation basis as digital systems. These systems often over-design various aspects in an attempt to achieve exact bit-accuracy compared to digital. However, in many cases this attempt to match digital is based on the aforementioned first-principles arguments which can mislead, resulting in systems which are neither efficient nor as accurate as proposed.

Opportunity: *To show the benefits of analog systems, we need to develop norms for analog accelerator papers and proposals:*

1) Accuracy analysis should be the first consideration. Since accuracy remains a primary question for analog accelerators, it needs to be demonstrated in papers. This analysis should be inclusive of all relevant non-idealities with the assessment of non-ideality relevance based on simulation data rather than first-principles arguments. Where possible this should also use per-component models rather than simple lumped error models to ensure that the interplay between sources is correctly accounted for.
2) Proposals should define a concrete accuracy target for accelerators based on what is achievable in digital. These questions of accuracy of performance and efficiency are not unique to analog systems and many application domains have developed field and/or industry norms for "sufficient" accuracy. For instance, in neural network inference, the MLPerf Inference benchmark defines a 1% relative accuracy metric for a valid submission [4]. Where possible, researchers on analog systems should use existing field norms to define accuracy targets.
3) Proposals should focus evaluation on problem scales and complexities matching the target application domain. For instance, if a proposal is motivated by the challenge of autonomous vehicles, the benchmark applications should either come from this domain, or match the field norms of other researchers working on autonomous vehicles applications (e.g., similar datasets and accuracy targets). Analysis of smaller scale applications is often still interesting, but it cannot replace relevant problem scale analysis. For many works this will require a blended approach of characterization for small scale applications and large-scale simulations based on characterization data, but as discussed below, this workflow is well-supported by existing tools.

Timeliness or maturity

Over the past several years we have seen multiple large-scale array and chip prototypes. With larger mixed-signal prototypes, effective benchmarking against digital systems becomes the primary concern. Thankfully, greater tool maturity from Sandia's CrossSim [5] and IBM's AI Hardware Kit [6] enables the high-fidelity analysis described above. Both tools provide interfaces for user-developed models of analog non-idealities and sufficient performance to run large scale applications within a reasonable amount of time.

By establishing a set of norms for accuracy analysis on analog systems, ASCR can help push the field of analog accelerators toward directions which are more likely to yield productive results. Even for work not funded directly, these norms can help focus reviewers on the important problem of accuracy, at least until the ability of analog systems to hit relevant thresholds on problems of interest has been sufficiently well established.

References:

[1] M. J. Marinella *et al.*, "Multiscale Co-Design Analysis of Energy, Latency, Area, and Accuracy of a ReRAM Analog Neural Training Accelerator," in *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 8, no. 1, pp. 86-101, March 2018, doi: 10.1109/JETCAS.2018.2796379.
[2] T. P. Xiao *et al.*, "On the Accuracy of Analog Neural Network Inference Accelerators," in *IEEE Circuits and Systems Magazine*, vol. 22, no. 4, pp. 26-48, Fourth quarter 2022, doi: 10.1109/MCAS.2022.3214409.
[3] Ambrogio, S *et al.* An analog-AI chip for energy-efficient speech recognition and transcription. *Nature* 620, 768–775 (2023). https://doi.org/10.1038/s41586-023-06337-5
[4] V. J. Reddi *et al.*, "MLPerf Inference Benchmark," *2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA)*, Valencia, Spain, 2020, pp. 446-459, doi: 10.1109/ISCA45697.2020.00045.
[5] T. P. Xiao *et al.*, "CrossSim: accuracy simulation of analog in-memory computing," [Online]. Available: https://github.com/sandialabs/cross-sim
[6] M. J. Rasch *et al.*, "A Flexible and Fast PyTorch Toolkit for Simulating Training and Inference on Analog Crossbar Arrays," *2021 IEEE 3rd International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, Washington DC, DC, USA, 2021, pp. 1-4, doi: 10.1109/AICAS51828.2021.9458494.

Truly Eliminating Data Movement With Hybrid Processing Using Memory

Saugata Ghose (*Univ. of Illinois Urbana-Champaign*, ghose@illinois.edu)
Ryan Wong (*Univ. of Illinois Urbana-Champaign & Sandia National Labs*, ryanw13@illinois.edu)
Ben Feinberg (*Sandia National Labs*, bfeinbe@sandia.gov)

Topics: in-memory analog computing; hybrid analog and digital systems

1. Challenge: The Limited Computational Capabilities of Analog Processing Using Memory

Processing-using-memory (PUM; a.k.a. *in-memory compute*/IMC or *compute-in-memory*/CIM) makes use of intrinsic properties and operational principles of memory cells within a memory by *inducing interactions between cells* such that they can perform computation. PUM systems minimize costly data movement energy even compared to processing-near-memory approaches by minimizing data movement within a memory array, i.e., between the memory cell and the read out circuitry. These advantages can potentially provide multiple-order-of-magnitude improvements in energy efficiency over conventional digital accelerators on specific application kernels [7].

Analog PUM architectures have become popular in both academia (e.g., [1, 3, 9, 10]) and industry (e.g., [2, 6, 8, 14]) due to their ability to perform bulk parallel multiplication. Fig. 1 shows an example of how an analog PUM system can be used to map *matrix–vector multiplication* (MVM). Fig. 1 (left) shows the mathematical operation of an MVM, while Fig. 1 (right) shows the circuit mapping. To implement MVM, the conductances of each memory cell in the array are programmed proportionally to the values of the matrix, and a voltage proportional to the input vector is applied. This allows a single memory cell to represent a multi-bit value, operating the cell in analog mode. Through Ohm's Law and Kirchhoff's Current Law, the resulting current at the bottom of each column will be proportional to the result of the MVM operation. Importantly, this simple example elides several complex details such as data conversion and data representation, but all analog PUM MVM implementations share the same core idea [13].

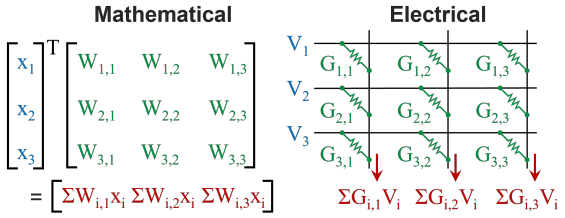


Figure 1: Example of an analog PUM system for MVM.

While analog PUM accelerators have garnered significant excitement for its ability to accelerate neural network (NN) inference, it faces two obstacles that significantly limit its potential in larger end-to-end applications. First, most applications require more than just multiplication. For an MVM, a dot product must sum up multiple partial products (i.e., *multiply–accumulate*, MAC). As analog PUM can only perform the accumulation when operands are narrow, wider-width MVM (e.g., 8-bit operands) requires *each* partial product to be converted from the current domain into the binary voltage domain using **costly analog–digital converters** (ADCs), after which CMOS-compatible digital shift and add circuits, or a host CPU, complete the computation. If the result of the MVM is needed as an input for a subsequent MVM, this also needs a reverse (digital–analog) conversion. Any computation beyond a MAC operation **requires the host CPU to intervene**, incurring CPU–memory data movement that generates high latencies and energy usage. Second, analog PUM is prone to a **high error rate**. Analog PUM's multi-bit operations are difficult to perform reliably in many memory technologies. For example, most resistive memories have a highly non-linear relation between current and the programmed resistive state, which requires significant precision to discern between adjacent bit value representations. Stochastic error sources for analog PUM include effects such as programming error (e.g., imprecision in the programmed state), cycle-to-cycle read noise, and state drift [14].

2. Opportunity: Hybrid Memory Arrays Capable of Both Analog and Digital PUM

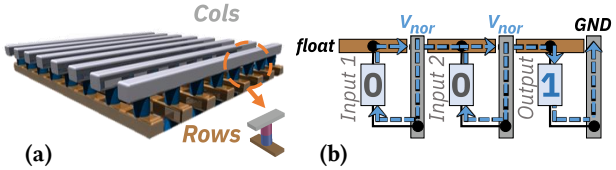


Figure 2: (a) ReRAM array: selectors in purple, resistive switches in blue; (b) NOR operation: current in blue.

An alternative approach to analog PUM is *digital PUM*, which can perform bulk bitwise instructions using the same memory array used for analog PUM, but with a careful selection of different voltages and currents that are applied to the column lines [4, 5, 12]. In the digital case, each resistive memory cell holds only a single-bit value, which significantly increases its reliability compared to analog PUM (i.e., it reduces a major source of stochasticity). Figure 2 shows how a NOR operation can be performed across three cells in the same row of a memory crossbar (i.e., array), by applying V_{nor} as the column selection voltage to the two inputs, and *GND* as the column selection voltage to the output cell, while keeping the row selection voltage floating. As illustrated by Figure 2b, floating the row line allows the input cells to set the row line voltage, which is then used to program the output cell. This behavior can be extended to entire columns of input cells, by floating *all* row selection voltages in an array, which enables bitwise NOR across any two columns of data in the array. Architectures built upon digital PUM (e.g., [11]) can perform some scalar operations in addition to bulk parallel operations.

We argue that the best of both worlds can be achieved with a *hybrid PUM* system. The key idea behind hybrid PUM is that

both analog PUM and digital PUM make use of the same underlying memory arrays and devices. The difference between the two is the peripheral circuitry used to assert voltages on memory array rows/columns, which in turn dictate the types of operations the array can perform. Even for MVM, a hybrid PUM architecture can eliminate reliance on external logic: the analog PUM arrays perform bit-sliced multiplies where each column of the memory generates a partial product, and digital PUM arrays perform the shift and add operations on these partial products. There are two general approaches to building a hybrid PUM architecture: (1) a *static hybrid PUM* architecture, where arrays are fixed at design time to be only analog PUM or only digital PUM; or (2) a *dynamic hybrid PUM* architecture, where each array contains peripheral circuits to perform both analog *and* digital PUM, and runtime software determines when an array should operate in analog PUM or digital PUM mode.

3. Bringing Single-Chip Computation to Fruition

A hybrid PUM system has the potential to avoid any need for integration with a host CPU, allowing for computers built from a single hybrid PUM memory chip that handles both data storage *and* all computation. However, there are three key challenges that must be addressed before such single-chip systems can become a reality:

1. ***The High Cost of ADCs.*** The basic hybrid PUM approach above does not solve the ADC issue, as operands sent from an analog PUM array to a digital PUM array still need to be converted from a multi-bit analog current on a single wire to a multi-wire digital current. While such conversions are not required for select data representations (e.g., binary neural networks), a robust single-chip solution must identify efficient ways to convert the data without the need for full ADCs. One option could be to use simple amplifiers that output a digital bit value 1 when the input analog current exceeds a threshold, though this will require either digital operations that complete the conversion or alternate digital number representations.
2. ***Efficient Runtime Support for Both PUM Modes.*** A hardware–software solution is required to truly manage a dynamic hybrid PUM system. On the hardware side, there is a need to innovate on low-overhead hybrid peripheral circuits, as peripheral circuits for single-mode PUM often dominate the total chip area. On the runtime software side, several options exist for deciding when to use analog PUM or digital PUM for each array within a chip (as well as potentially coordinating decisions across multiple chips). A simple mechanism can rely on an expert programmer to manually manage data mapping and mode decisions, but such a direction is likely to harm long-term viability. Realistically, there is a need for an intelligent runtime, which can use high-level properties such as target accuracy, the types of operations that the program wants to perform in its current phase of execution, and energy estimators to predict an optimal partitioning of arrays into analog mode and digital mode. Such a runtime can reconfigure the partitioning whenever an application enters a new phase.
3. ***Scalable Control Flow and Scalar Execution.*** While data-driven control flow is omnipresent in most real applications, most works on PUM avoid addressing efficient control flow handling. Instead, they depend on the host CPU to execute such decisions. Unfortunately, given that control flow can make up a fifth of all instructions in a program on average, this offloading can erode many of the potential benefits of PUM. Focused work on efficient front ends for PUM can ultimately enable efficient control flow handling, as well as support for variable amounts of parallelism. With such front ends, PUM systems can eliminate the need for any offloading to a host CPU.

4. References

[1] A. Ankit *et al.*, "PUMA: A Programmable Ultra-Efficient Memristor-Based Accelerator for Machine Learning Inference," in *ASPLOS*, 2019.
[2] C. H. Bennett *et al.*, "Device-Aware Inference Operations in SONOS Non-Volatile Memory Arrays," in *IRPS*, 2020.
[3] T. Chou *et al.*, "CASCADE: Connecting RRAMs to Extend Analog Dataflow in an End-to-End In-Memory Processing Paradigm," in *MICRO*, 2019.
[4] S. Gupta, M. Imani, and T. Rosing, "FELIX: Fast and Energy-Efficient Logic in Memory," in *ICCAD*, 2018.
[5] S. Kvatinsky *et al.*, "MAGIC: Memristor-Aided Logic," *TCAS II*, Sep. 2014.
[6] M. Le Gallo *et al.*, "A 64-Core Mixed-Signal In-Memory Compute Chip Based on Phase-Change Memory for Deep Neural Network Inference," *Nature Electronics*, Aug. 2023.
[7] M. J. Marinella *et al.*, "Multiscale Co-Design Analysis of Energy, Latency, Area, and Accuracy of a ReRAM Analog Neural Training Accelerator," *JETCAS*, 2018.
[8] Mythic, Inc., "M1076 Analog Matrix Processor," https://mythic.ai/products/m1076-analog-matrix-processor/.
[9] A. Shafiee *et al.*, "ISAAC: A Convolutional Neural Network Accelerator With In-Situ Analog Arithmetic in Crossbars," in *ISCA*, 2016.
[10] L. Song *et al.*, "PipeLayer: A Pipelined ReRAM-Based Accelerator for Deep Learning," in *HPCA*, 2017.
[11] M. S. Q. Truong *et al.*, "RACER: Bit-Pipelined Processing Using Resistive Memory," in *MICRO*, 2021.
[12] M. S. Q. Truong *et al.*, "Adapting the RACER Architecture to Integrate Improved In-ReRAM Logic Primitives," *JETCAS*, 2022.
[13] T. P. Xiao *et al.*, "Analog Architectures for Neural Network Acceleration Based on Non-Volatile Memory," *APR*, 2020.
[14] T. P. Xiao *et al.*, "On the Accuracy of Analog Neural Network Inference Accelerators," *IEEE CAS Magazine*, 2022.

A Position Paper on: **Biological Computing for Science and Engineering**

Peter Graf (NREL)(peter.graf@nrel.gov), Yakoub Nemouchi (NREL) Topic: Biological Computing

"Computation" is a relatively new concept. The success of silicon transistor-based implementation has, reasonably, obscured other approaches. But (among other things) their excessive energy use motivates renewed interest in more exotic ideas, in particular analog computing. It is our position that among the various analog computing modalities, there should be an increased research focus on *Biological Computing* (BC). BC employs biochemical mechanisms such as DNA and metabolic networks to perform computations. There is evidence that, using gene editing (e.g., CRISPR) and other advanced bioengineering technology, BC could be implemented in living cells [1]. We believe BC will be an increasingly relevant form of "unconventional" computing, and that the following three research **challenges** and corresponding **opportunities** are important areas to address and develop:

1. *Biological computing for biological applications*. Nature has designed "circuits" in cells that are up to 4-5 orders of magnitude more efficient than human designed electronic circuits [1]. And synthetic biology, increasingly, can implement circuits that *we* design (Figure 1). However, nature imposes many challenging constraints on BC (cell stays alive, chemical composition is fluctuating and stochastic, etc.) that suggest that *specialized applications* are the best first use case for BC. One class of such opportunities is application to biology itself. For example, we can envision a bioreactor (say, for producing biofuel, or biological ammonia) that contains "programmed" cells that are able to monitor and control the process *in situ*. Such applications are already in the domain of synthetic biology, but we envision developing the *computational* aspects more thoroughly, comparable to the way, for example, the increasingly sophisticated and powerful electronics of modern vehicles is far beyond the imagination of simple relays and dials of earlier generations.

2. *Biological computing for specialized non-biological applications*. Arguably the most well-developed approach to BC is through so-called *chemical reaction networks* (CRNs) [2]. CRNs can implement both discrete, exact algorithms (i.e., traditional programs) and act as analog computers for the wide variety of models reducible to systems of differential equations. In fact, CRNs have been shown to be Turing complete [3]; but there is large gap between Turing-completeness and demonstrable utility. CRNs are at a stage where basic building blocks of computation are being established, but there are no established "killer applications". Like other alternative computing paradigms (e.g., quantum computing and neuromorphic computing), BCs will inherit characteristics of their biological substrate that may suggest certain applications. For example, physical CRNs will be inherently stochastic (relying on well-mixed chemical baths), motivating study of *stochastic* CRN, where it has been proven that "when answers must be guaranteed to be correct, computational power is limited, but when an arbitrarily small error probability can be tolerated, the computational power is dramatically increased" [4]. Examples of such problems abound in DOE research; in particular, large scale stochastic optimization occur frequently (e.g., in power systems planning and operations) and are currently intractable and only solved approximately. We suggest an opportunity for research to match the properties of DOE-relevant computational problems to the characteristics of the proposed BC machines.

3. *Assured and reliable computational capability transfer from traditional computers to biological computers.* We would like BCs to be "programmable" in a sense that is similar enough to traditional programming for adoption by application engineers. For this, we need something like a "C-to-CRN" compiler. Languages such as CRN++ [5] and nascent compilers exist [6], but to date there is little work in formal verification of the compilation pipelines for BCs, which we contend is critical, as we are mapping between wildly different domains. Can this be accomplished in a reliable manner? This situation presents an opportunity for research. One option (Figure 2) is a stepwise one to formally verify compilations of discrete programs (e.g. programs written in the C language) down to differential equations. Two major parts can be distinguished; during the first part we translate the discrete program to an abstract program (e.g., relational predicate) using formally verified abstraction techniques. For the second part we could use differential dynamic logic [7] and differential refinement logic [8] to translate the abstract program to differential equations through an intermediate hybrid program that combines discrete and continuous computations. All these steps are done while provably preserving important properties of the computational results.

We believe that there are complementary reasons why research into BCs is especially **timely**. First (and this is a motivation for all unconventional computing), the energy use of traditional silicon-based machines is rapidly becoming unsustainable; we must do something to "offload" a good deal of computation to other modalities.

Second, the rapid development of the bioengineering capabilities underlying BC suggests there will soon be an explosion in our ability to build operational and practical BCs. Both quantum computing (QC) and neuromorphic computing (NC) are illustrative comparisons. The recent rapid grown of QC and to a lesser extent NC are motivated by a recognition of and algorithms for "possibly-beyond-classical" capabilities converging with hardware advancements enabling their physical implementation. BC is at a much lower TRL level presently, but we can already hypothesize a similar trend.

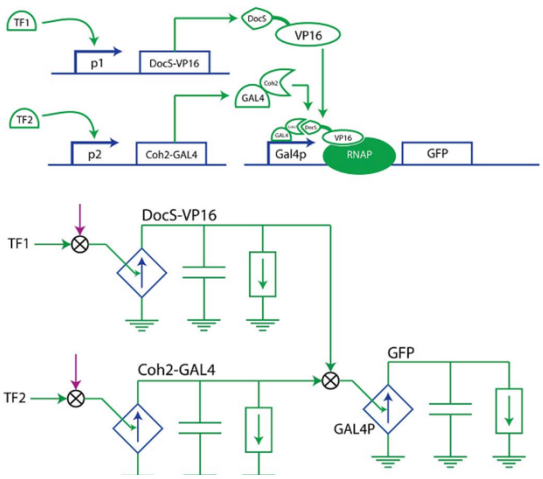


Figure 1From [1]. A multiplier circuit that can be implemented within a living cell. Top is the biochemistry corresponding to the analog electrical circuit on bottom.

References:

[1] Teo, Jonathan JY, Sung Sik Woo, and Rahul Sarpeshkar. "Synthetic biology: A unifying view and review using analog circuits." *IEEE transactions on biomedical circuits and systems* 9, no. 4 (2015): 453-474.

[2] Brijder, Robert. "Computing with chemical reaction networks: a tutorial." *Natural Computing* 18 (2019): 119-137.

[3] Fages, François et al, "Strong Turing completeness of continuous chemical reaction networks and compilation of mixed analog-digital programs." In *Computational Methods in Systems Biology: 15th International Conference, CMSB 2017.*

[4] Cook, Matthew, David Soloveichik, Erik Winfree, and Jehoshua Bruck. "Programmability of chemical reaction networks." In *Algorithmic bioprocesses*, pp. 543-584. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009.

[5] Vasić, Marko, David Soloveichik, and Sarfraz Khurshid. "CRN++: Molecular programming language." *Natural Computing* 19 (2020): 391-407.

[6] Hemery, Mathieu et al, "Compiling elementary mathematical functions into finite chemical reaction networks via a polynomialization algorithm for ODEs." In *Computational Methods in Systems Biology: 19th International Conference, CMSB 2021*.

[7] Platzer, André. "A complete uniform substitution calculus for differential dynamic logic." *Journal of Automated Reasoning* 59, no. 2 (2017): 219-265.

[8] Loos, Sarah M., and André Platzer. "Differential refinement logic." In *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science*, pp. 505-514. 2016.

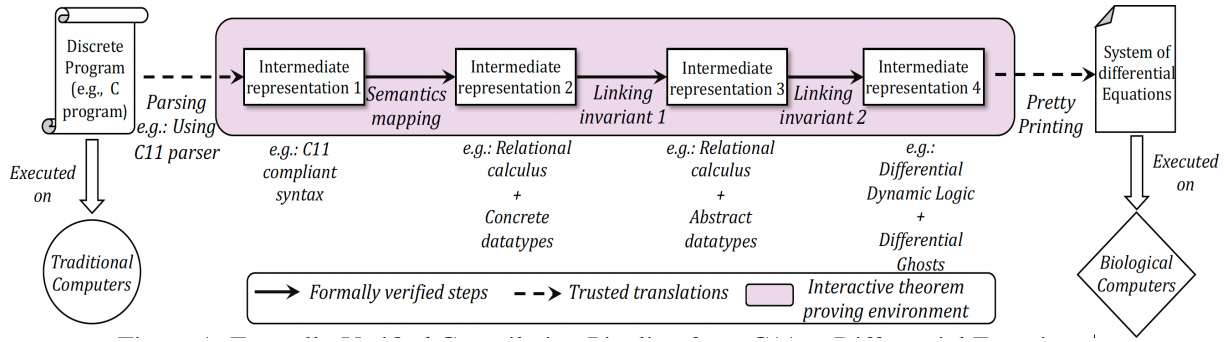


Figure 2 Formally verified compilation pipeline from C11 to differential equations

Leveraging Flow-Based In-Memory Computing Paradigms for Advancing Analog Stochastic Computing in Scientific Applications

Authors: Sumit Kumar Jha, Florida International University
Alvaro Velasquez, University of Colorado Boulder

Topic: Analog Stochastic Crossbar Computing, Mathematical Foundations, and Design Automation.

Challenges

The rapid growth of data-intensive applications has exposed the limitations of traditional digital architectures, primarily due to the significant data movement between memory and processing units and the need to perform repeated digital switching to perform computations. In-memory computing (IMC) paradigms, particularly path-based and flow-based computing, offer a promising solution to this bottleneck by integrating memory and computation, and reducing the number of compute-time switching of devices for computation.

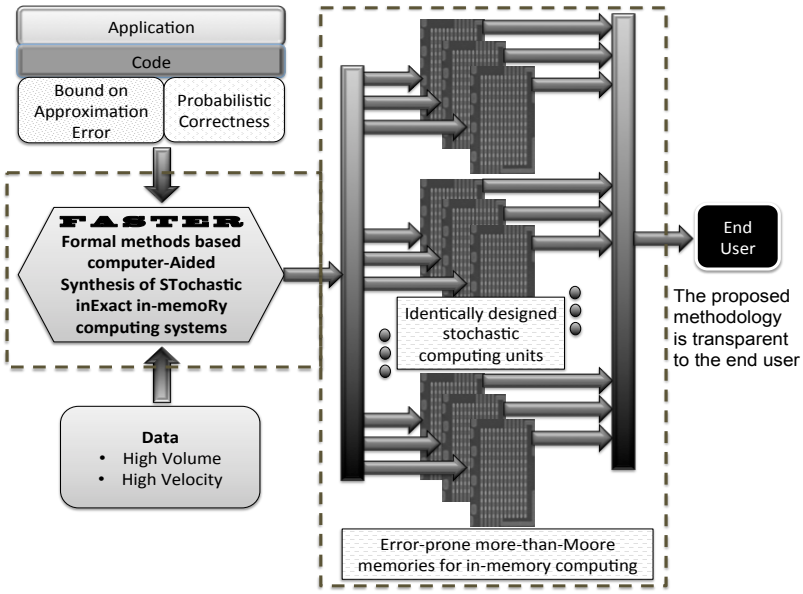


Figure 1: Stochastic analog computing using nanoscale 2-D and 3-D crossbars can lead to the design of more energy-efficient computation systems.

Stochastic analog crossbar computing leverages the inherent stochasticity of analog memristor devices and the exponential number of sneak paths in nanoscale memristor crossbars to perform computations efficiently. This method exploits the probabilistic behavior of nanoscale devices to perform approximate computing, which may be more energy-efficient and suitable for applications where exact results are not critical. However, existing methods face challenges such as high area overhead and susceptibility to analog errors, which hinder their practical implementation and scalability in scientific applications.

Opportunity

Flow-based computing uses programmable 2-D and 3-D arrays of nonlinear devices to store data and then employ an exponential number of sneak paths to put together the stored information and perform a desired computation. Recent advancements in path-based and flow-based in-memory computing paradigms present unique opportunities to enhance analog computing systems. Path-based computing leverages READ operations to evaluate functions of data stored in nanoscale devices using nanoscale crossbars, offering low power consumption and computational delay. Flow-based computing minimizes computational delays by utilizing efficient crossbar designs for computations. By extending these paradigms to stochastic analog computing that manipulates analog data and probabilistic switching of these devices, we can develop systems that exploit the strengths of each approach, thereby addressing their limitations and enhancing the overall performance of analog crossbar computing systems.

Stochastic analog computing uses the inherent noise and variability in nanoscale devices to perform probabilistic computations. This approach can lead to significant improvements in energy efficiency, as the devices do not need to switch states fully and can use lower voltages. Hybrid systems, where analog computing is used for coarse solutions and digital computing for refinement, can leverage the benefits of both paradigms, providing energy-efficient and high-performance solutions for scientific applications.

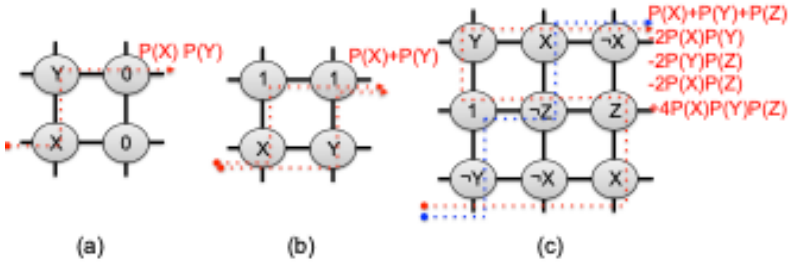


Figure 2: Example of nonlinear computation implemented using stochastic computing on a crossbar using sneak paths.

Our focus is on developing EDA tools that create flow-based stochastic analog crossbar computing systems. The software needs to design the analog crossbar by compiling high-level specifications, and verification tools need to check against potential drifts and associated errors in crossbar circuit design and operation. Formal verification tools, such as decision procedures and multi-terminal decision diagrams, can be used to establish the correctness of the design, explore the design space, and provide assurance arguments at runtime.

Flow-based computing replicates the idea of information flow through neurons by utilizing the flow of current through numerous sneak paths in nanoscale crossbars. By focusing on the analog and stochastic aspects of device dynamics, such as those of memristors, flow-based computing offers new opportunities for efficient and resilient computing that has the potential to replicate neuronal pathways.

Mathematical frameworks have been established for modeling current flow through crossbar devices using closed-form equations, which can be manipulated both symbolically and numerically. In the analog and stochastic domain, multi-terminal binary decision diagrams and Bayesian hypothesis testing serve as tools for reasoning with analog crossbars. Additional work is needed both in the design space and in the simulation and modeling space to fully document the computational capabilities of crossbars using analog stochastic computing.

Timeliness or Maturity

The urgency to overcome the digital bottleneck in data-intensive applications and the availability of advanced nonlinear analog nanoscale crossbar designs make this an opportune time to explore and develop hybrid in-memory computing systems. Our proposed methodologies, such as the XORG framework for resilient data layout organization, have demonstrated significant improvements in energy efficiency, computational latency, and robustness against analog errors. These pave the way for practical implementations of analog stochastic in-memory computing systems in scientific applications, providing a timely impetus for further research and development.

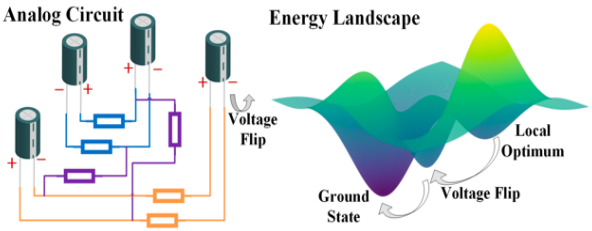
References

1. S. Thijssen, M. Rashed, SK Jha, and R. Ewetz, "Synthesis of Compact Flow-based Computing Circuits from Boolean Expressions." In 61st ACM Design Automation Conference (DAC), 2024.
2. M. Rashed, S. Thijssen, D. Simon, SK Jha, and R. Ewetz, "Execution Sequence Optimization for Processing In-Memory using Parallel Data Preparation." In 61st ACM Design Automation Conference (DAC), 2024.
3. S. Thijssen, SK Jha, and R. Ewetz, "PATH: Evaluation of Boolean Logic using Path-based In-Memory Computing." In 59th Design Automation Conference (DAC), 2022.
4. M. Rashed, Amro Awad, SK Jha, and R. Ewetz, "Towards Resilient Analog In-Memory Deep Learning via Data Layout Re-Organization." In 59th Design Automation Conference (DAC), 2022.
5. S. Thijssen, M. Rashed, SK Jha, and R. Ewetz, "UpTime: Towards Flow-based In-Memory Computing with High Fault-Tolerance." In 60th Design Automation Conference (DAC), 2023.

Analog Computing through Dynamic Systems for Science

Ang Li, Pacific Northwest National Laboratory, ang.li@pnnl.gov
Michael Huang, University of Rochester, michael.huang@rochester.edu
Tony Geng, University of Rochester, tgeng@ur.rochester.edu
Topic: Analog Optimization

Challenges: The exponential scaling of scientific simulation and machine learning necessitates unprecedented increases in computational power for the DOE. This soaring computational demand, combined with the deceleration of Moore's Law, leads to surging energy costs. While improving the computation efficiency of traditional digital computing remains critical, exploring novel analog computing paradigms could offer fundamental and transformative solutions.



Recently, the concept of "using nature as a computer" has gained significant traction, underscored by federal initiatives such as DARPA's Nature As Computer program and several seminal studies. These studies showed that the intrinsic power of Ising Machine-based dynamical systems (DS) – which naturally converge to equilibrium at the lowest energy states (e.g., chemical reactions, water crystallization) – can be harnessed for extraordinarily efficient computing by embodying an analog dynamical system as a semiconductor chip. Since the system remains closed and self-driving, theoretically no extra external energy cost is required throughout the dynamic process. Meanwhile, as the dynamic process is achieved through current flow among the capacitors, the solution is obtained at the "speed of nature" [2-4].

DS can potentially solve certain challenging optimization and learning problems seen in DOE applications. These problems are typically NP-Hard [1] and can take hours to days on traditional CPU/GPU processors. With the CMOS-compatible analog DS chip, the running time for these problems, when properly formulated, can be reduced to milliseconds or even microseconds, achieving performance of approximately 3.9 TFLOPs/ms and energy efficiency of about 15.4 MFLOPs/nJ compared to equivalent digital operations for the same functions. This represents a three orders of magnitude performance gain and five orders of magnitude energy savings [2-3].

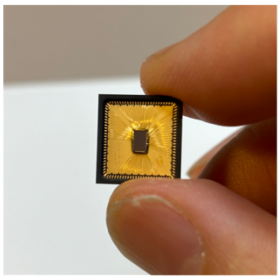
Despite its great potential, present Ising-machine-based DS systems, such as the seminal BRIM approach, face three fundamental challenges when applied to DOE applications:

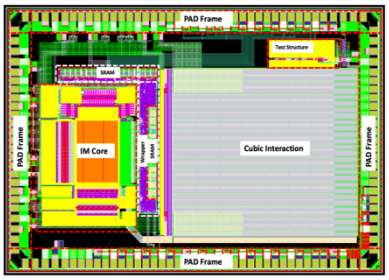
- Traditionally, the Ising machine can only handle binary optimization problems, known as Quadratic Unconstrained Binary Optimization (QUBO). However, most DOE scientific and learning applications are non-binary and do not contain any QUBO subfunctions. The inability to handle continuous problems remains a significant hurdle.
- Existing Ising machines can only handle linear optimization problems, whereas many DOE scientific and learning applications are non-linear. Formulating the non-linear problems through linearization or equipping the Ising machine with non-linear capability remains an issue.

- The rapid evolving & annealing process of Ising machines demands full connectivity among the spins, leading to fundamental constraints on scalability. Effectively scaling the Ising machines is crucial to determining whether the DS solution can be deployed for practical DOE domain applications.

Opportunities: New opportunities arise regarding each of the identified challenges.

- Recent CMOS-compatible DS machines embed the target problem settings as the initial capacitor charge and resistance of the resistors, driving the dynamic process through the natural flow of current in the capacitor/resistor network. Unlike conventional Ising machine implementations, this dynamic process is fundamentally continuous [4]. New Hamiltonian formulations and architecture & analog circuit designs are required to unleash this capability and extend the solving capability from QUBO to general optimization problems.
- New techniques for linearizing DS for DOE non-linear algorithms and applications are needed to handle the non-linear hurdle. Additionally, new architecture & circuit designs might be required through a codesign approach to enable DS with the ability to handle non-linear processes, similar to the activation functions (e.g., ReLU) of neural networks.
- Novel architecture and interconnect device designs are required to scale the DS machines to large-scale and possibly across multiple DS chips [3]. Additionally, software techniques are needed to explore problem feature space, such as sparsity and clustering patterns, to exploit locality and avoid excessive remote communications among Ising units or chips.





Maturity: The great opportunities for DS-based computing is triggered by recent breakthroughs demonstrating the Ising machine via CMOS-based semiconductor technology [2]. The right figure shows the Ising analog chip and its layout designed at the University of Rochester. With the potential to fabricate the DS chip through a mature semiconductor process and its fundamental capability in handling continuous variables, it presents outstanding opportunities to accelerate DOE scientific optimization and learning applications. Significant investment now is needed to tackle the remaining challenges .

References:

[1] A. Lucas. Ising formulations of many NP problems. Frontiers in physics, 2:5, 2014.

[2] R. Afoakwa, Y. Zhang, U.K.R. Vengalam, Z. Ignjatovic, and M. Huang. Brim: Bistable resistively-coupled Ising machine. In IEEE International Symposium on High-Performance Computer Architecture (HPCA), 2021.

[3] R. Song, C. Wu, C. Liu, A. Li, M. Huang, and T. Geng. DS-GL: Advancing graph learning via harnessing the power of nature within scalable hardware dynamical systems. the 51th IEEE/ACM International Symposium on Computer Architecture (ISCA), 2024.

[4] [4] C. Wu, R. Song, C. Liu, Y. Yang, A. Li, M. Huang, and T. Geng. Extending power of nature from binary to real-valued graph learning in real world. The Twelfth International Conference on Learning Representations (ICLR), 2024.

Efficient hybrid analog computing for numerical algorithms

Rui Peng Li* Yuanzhe Xi† Vassilis Kalantzis‡ Mark S. Squillante‡
Chai Wah Wu‡

Topic: Analog algorithms and programming

Challenge: Harnessing the full potential of analog devices requires simultaneous efforts on several different fronts, from algorithmic analysis to simulations and practical implementations. Due to the non-deterministic behavior of noise, it is imperative to design and analyze new algorithms that are intrinsically resilient to hardware imperfections, yet are able to leverage the unique speedups attainable by an analog device. *This is quite challenging, since in digital computing noise is typically attributed to the data collection in hand rather than the computing device.* The ultimate goal towards a general-purpose realization of analog computing thus passes through the design and development of new algorithms that are robust to computational noise.

Opportunities: Analog and hybrid numerical algorithms

1. Preconditioning of sparse linear systems: The solution of sparse systems of linear algebraic equations is one of the most common numerical problems encountered in computational mathematics [4]. For large-scale linear systems, iterative solvers are a popular choice, especially when combined with preconditioners. The preconditioner must be chosen so that, ideally, it reduces the condition number of the iteration matrix and is also relatively inexpensive, both in terms of computational cost and latency, to apply. Unfortunately, it is hard to find a preconditioner that fulfills both objectives for general sparse problems. That is, constructing accurate preconditioners that are fully parallelizable and have low power consumption can be challenging on digital computers. Moreover, applying preconditioners can be a daunting task due to the potential high nonzero density of the preconditioner. In this context, hybrid algorithms where analog hardware is responsible for the application of the preconditioner while digital hardware is responsible for constructing the preconditioner and all remaining computational tasks, is an attractive choice that combines the best of two worlds [2, 3]. Nonetheless, due to the large level of noise introduced by analog hardware, a very accurate preconditioner will not pay off beyond a certain critical point. As a result, it is imperative to exploit a systematic strategy to regularize the construction of the preconditioner so that the approximation accuracy is not more accurate than what the analog co-processor can offer.

2. Quasi-Newton optimization: Numerical optimization is a field of paramount importance in science and engineering. A prerequisite towards the broader utilization of analog devices in optimization algorithms beyond (stochastic) gradient descent and related variants is the study and analysis of analog noise and overall non-deterministic behavior on the qualitative performance of optimization algorithms. In particular, the applicability of analog devices in optimization requires the fault-tolerance analysis of optimization algorithms as well as the impact that analog non-idealities have on their convergence rate. In principle, the convergence

*Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, P. O. Box 808, L-561, Livermore, CA 94551 (li50@llnl.gov). This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344
†Department of Mathematics, Emory University, Atlanta, GA 30322 (yxi26@emory.edu)
‡IBM Research, Thomas J. Watson Research Center, Yorktown Heights, NY 10598 (vkal@ibm.com)

rate of an analog-based, hybrid optimization algorithm might converge at a slower rate but the execution of each iteration is much cheaper due to offloading matrix-based operations to analog hardware.

One specific category of optimization algorithms where analog hardware has the potential to be highly beneficial is quasi-Newton optimization algorithms where the approximate Hessian resides on analog hardware. A particularly interesting example in this class of algorithms is the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm [1], an iterative method for solving unconstrained nonlinear optimization problems, but the same concepts qualify for other algorithms such as the Davidon–Fletcher–Powell formula and Symmetric Rank 1. The computational complexity of BFGS is quadratic with respect to the number of unknowns, however the operations that are responsible for this complexity are involved only during computations with the approximate Hessian matrix and thus can be offloaded to an analog co-processor. These computational operations are matrix-vector products and rank-2 outer-product updates, which are precisely the linear algebra kernels that analog crossbar arrays can handle efficiently. Using analog hardware, the digital-based computational complexity of BFGS is only linear with respect to the number of unknowns.

Timeliness or maturity: For decades, the need to solve larger computational problems at increasingly faster rates was mainly met by shrinking the size and increasing the number of transistors in complementary metal-oxide-semiconductor (CMOS) integrated circuits. Nonetheless, CMOS-based von Neumann microprocessors are now much closer to their physical limits of scaling and power dissipation, and continuing leveraging them "as-is" becomes increasingly more challenging. In addition to the above constraints, the von Neumann aspect of predominant classical digital hardware, i.e., the physical separation of processing units and primary memory, increases the latency and energy profile of scientific workloads.

In response to the drawbacks mentioned above, recent years have seen arguments favoring the use of analog devices configured as crossbar arrays of non-volatile memories. These devices have demonstrated potential in accelerating many modern scientific computing applications, such as numerical optimization algorithms in deep learning. In these applications, computations are often dominated by matrix-vector multiplications and outer products [5, 6]. By mapping matrices onto arrays of memristive elements, where each cross-point can store information and execute simple operations like multiply-and-add, a very high degree of parallelism with low power consumption can be achieved. Each crossbar performs its respective operations in-memory, which further enhances the efficiency.

References

[1] Roger Fletcher. *Practical methods of optimization*. John Wiley & Sons, 2000.

[2] Vasileios Kalantzis, Anshul Gupta, Lior Horesh, Tomasz Nowicki, Mark S Squillante, Chai Wah Wu, Tayfun Gokmen, and Haim Avron. Solving sparse linear systems with approximate inverse preconditioners on analog devices. In *2021 IEEE High Performance Extreme Computing Conference (HPEC)*, pages 1–7. IEEE, 2021.

[3] Vasileios Kalantzis, Mark S Squillante, Chai Wah Wu, Anshul Gupta, Shashanka Ubaru, Tayfun Gokmen, and Lior Horesh. Solving sparse linear systems via flexible GMRES with in-memory analog preconditioning. In *2023 IEEE High Performance Extreme Computing Conference (HPEC)*, pages 1–7. IEEE, 2023.

[4] Yousef Saad. *Iterative methods for sparse linear systems*. SIAM, 2003.

[5] Abu Sebastian, Manuel Le Gallo, Riduan Khaddam-Aljameh, and Evangelos Eleftheriou. Memory devices and applications for in-memory computing. *Nature nanotechnology*, 15(7):529–544, 2020.

[6] Qiangfei Xia and J Joshua Yang. Memristive crossbar arrays for brain-inspired computing. *Nature materials*, 18(4):309–323, 2019.

Tackling Numerical Modeling Challenges for Large-scale Photonic Neural Networks

Yang Liu
Applied Mathematics and Computational Research Division, Lawrence Berkeley National Laboratory
liuyangzhuan@lbl.gov

Topic: Analog computation with photonics

1 Introduction

The explosive AI development in recent years has led to the demand for increasing the computational capability of information processing systems by at least five orders of magnitudes [1]. Existing electronics-based computing systems, ranging from general-purpose CPU and GPU, to specialized hardware like FPGA and TPU, suffer from high energy consumption and high interconnect latency. Photonic neural networks (PNNs), as one of the emerging analog computing techniques, can process and transmit information at the speed of light with low latency, enjoy low power consumption, and can exploit large parallelism (via wavelengh, polarization, mode, etc.).

PNNs have been shown to be well-suited for machine learning and high-dimensional nonlinear optimization tasks. For machine learning, PNNs can naturally implement multi-layer perception, convolutional neural networks using passive elements like waveguides, modulators, splitters and filters, as well as spiking neural networks and reservoir computing networks using active devices like lasers and amplifiers, in either free-space or integrated circuit fashions [2]. For nonlinear optimization, PNNs can naturally emulate the Hopfield network for Ising machines using devices like degenerate optical parametric oscillators [1]. A few recent papers have demonstrated that PNNs can achieve significantly higher flop performance and orders of magnitudes lower power consumption compared with state-of-the-art GPU or TPU architectures [1, 2]. As such, PNNs represent promising computing methodologies for many DOE applications requiring processing of large-volume and real-time data, such as data compression and trajectory reconstruction in HEP experiments, power system anomaly prediction, optimization for material science, fusion reactor plasma control, and phase retrieval in X-ray scattering experiments.

2 Challenge

Despite these exciting ongoing developments and demonstrations, PNNs face several well-known challenges such as lack of optical logic gates and memory units, and modeling/implementation difficulties for the nonlinearity function. More crucially, the cascadability to deeper networks and the scalability to large number of neurons suffer from intrinsic noise, propagation loss, cross-talk, and fabrication imperfection. New materials, devices and architectures have been proposed to address some of these bottlenecks, such as phase change material (PCM) for the synaptic weight, neuron and memory elements, advances in both O/E/O and all-optical solutions for the nonlinearity implementation, and new reservoir computing architectures, etc. In addition to the hardware development, this position paper highlights another opportunity for addressing the scalability issue - leveraging recent algorithmic advances in applied mathematics: numerical linear algebra, computational eletromagnetics/photonics and uncertainty quantification for modeling and characterising large-scale PNNs.

3 Opportunity

Here we bring up two emerging directions for scaling up PNNs: (1) leveraging new numerical algebra and uncertainty quantification tools for modeling existing PNN architectures, and (2) leveraging new numerical algebra and computational electromagnetic tools for designing new PNN architectures.

3.1 Fast and high-fidelity modeling tools beyond device level

Due to the large electrical size of PNNs, most existing numerical modeling tools such as MEEP can only handle photonic components at the device level [3] via e.g., the finite-difference time-domain algorithm (FDTD). Take the matrix-vector multiplication in each PNN layer for example, one matrix element (i.e., synaptic weight) can be implemented using devices like Mach–Zehnder Interferometers (MZIs), microring resonators, microdisk resonators, or PCM-enhanced waveguides [2, 4]. These devices can easily span 10-100 wavelengths in size, and simulation beyond device levels are typically handled by reduced models or circuit simulations. Another case is the photorefractive material-based weights whose interaction length spans around 300 wavelengths and cross section is proportional to the

number of matrix elements. Due to numerical dispersion of FDTD, accurate first-principle modeling PNN components at the circuit level requires very high spatial resolution and large high-performance computing resources.

In contrast to FDTD, recent advances of fast algorithms in computational electromagnetics have made high-fidelity simulation of PNNs at the circuit level (or even the micro-architecture level) computationally feasible. Examples include windowed Green's function-based frequency domain simulator [5], quantized tensor-train (QTT)-based frequency domain solver [6], QTT-based FDTD solver [7] and butterfly-based tensor and matrix algorithms [8]. These algorithms feature low algorithmic complexity based on fast matrix/tensor decomposition and error controllability, and can handle systems spanning thousands or even ten thousands of wavelengths in size. These forward simulators can be further combined with recently developed uncertain quantification frameworks for characterizing sensitivity and uncertainty due to bending noise, parameter drift, intensity fluctuation, and laser noise, etc.

3.2 Network architectures leveraging mathematical properties of light propagation

In addition to large-scale modeling of existing PNN architectures, one can also leverage physical and mathematical properties of light propagation to design more scalable architectures. One example is the tensorized PNN with QTT representation of the synaptic weights requiring much fewer MZIs, which is able to achieve 100x speedups compared with existing PNNs and 10000x speedups compared with an electronics NN [9]. Another example is the spatial-photonic Ising machine[10] with low-rank and/or FFT-like compression of the coupling matrices of an Ising Hamiltonian leveraging its physical property. This line of research direction provides a promising recipe to construct parameter (i.e., device) efficient PNNs leveraging underlying mathematical structures or physical properties.

4 Timeliness and maturity

The projected computational capacity required by future-generation AI foundational models has posed pressing needs for energy efficient analog computing technologies including photonic computing. On the other hand, the past five years have witnessed tremendous efforts and renewed interests in PNNs leveraging ongoing advances in physics, material science, computer hardware and applied mathematics. Large/deep PNNs consisting of over one million neurons/nodes have been demonstrated only very recently [1], and it's now the right time to foster deeper collaborations between mathematicians, computer scientists and material scientists, and target at realization of large-scale general- and special-purpose PNNs for scientific and industry applications in the next 5-10 years.

References

[1] N. Stroev and N. G. Berloff, "Analog photonics computing for information processing, inference, and optimization," *Advanced Quantum Technologies*, vol. 6, no. 9, p. 2300055, 2023.

[2] L. De Marinis, M. Cococcioni, P. Castoldi, and N. Andriolli, "Photonic neural networks: A survey," *Ieee Access*, vol. 7, pp. 175 827–175 841, 2019.

[3] A. M. Hammond, A. Oskooi, M. Chen, Z. Lin, S. G. Johnson, and S. E. Ralph, "High-performance hybrid time/frequency-domain topology optimization for large-scale photonics inverse design," *Optics Express*, vol. 30, no. 3, pp. 4467–4491, 2022.

[4] B. J. Shastri, A. N. Tait, T. Ferreira de Lima, W. H. Pernice, H. Bhaskaran, C. D. Wright, and P. R. Prucnal, "Photonics for artificial intelligence and neuromorphic computing," *Nature Photonics*, vol. 15, no. 2, pp. 102–114, 2021.

[5] E. Garza, C. Sideris, and O. P. Bruno, "A boundary integral method for 3-d nonuniform dielectric waveguide problems via the windowed green function," *IEEE Transactions on Antennas and Propagation*, vol. 71, no. 4, pp. 3758–3763, 2023.

[6] E. Corona, A. Rahimian, and D. Zorin, "A tensor-train accelerated solver for integral equations in complex geometries," *Journal of Computational Physics*, vol. 334, pp. 145–169, 2017.

[7] E. Ye and N. Loureiro, "Quantized tensor networks for solving the vlasov-maxwell equations," *arXiv preprint arXiv:2311.07756*, 2023.

[8] Y. Liu, "Recent algorithm developments in the butterflypack package: tensor algorithms," in Ann. Rev. Prog. Appl. Computat. Electromagn., 2024.

[9] X. Xiao, M. B. On, T. Van Vaerenbergh, D. Liang, R. G. Beausoleil, and S. Yoo, "Large-scale and energy-efficient tensorized optical neural networks on iii–v-on-silicon moscap platform," *Apl Photonics*, vol. 6, no. 12, 2021.

[10] R. Z. Wang, J. S. Cummins, M. Syed, N. Stroev, G. Pastras, J. Sakellariou, S. Tsintzos, A. Askitopoulos, D. Veraldi, M. C. Strinati *et al.*, "Efficient computation using spatial-photonic ising machines: Utilizing low-rank and circulant matrix constraints," *arXiv preprint arXiv:2406.01400*, 2024.

Predictive Simulations for CMOS and Beyond-CMOS Devices for Energy-Efficient, Analog Computing

Denis Mamaluy and Juan Mendez
Sandia National Laboratories, Albuquerque, New Mexico

Novel computing hardware accelerators based on beyond-von-Neumann architectures, analog and neuromorphic computing schemes on practice can only emerge as faster and more energy-efficient alternatives to the state-of-the-art digital CMOS circuitry used in Central Processing Units (CPUs), Graphics Processing Units (GPUs), and Neural Processor Units (NPUs). While the empirical Moore's law shown in Figure 1 and the exponential device performance scaling continue, albeit with a different exponent, it is essential to assess the actual benefits of beyond-von-Neumann architectures against the state-of-the-art and future CMOS digital circuits. Otherwise, any potential advantage of an alternative computing approach could be surpassed by the exponential performance gains of the digital CMOS circuitry.

Historically, advancements in materials and devices have driven the most significant improvements in energy efficiency and speed for computing systems. Since the power consumption per chip remained roughly constant, the power dissipation per transistor has been exponentially reducing, supported by the continuing Moore's law (Figure 1). In modern CMOS chips the energy dissipation occurs primarily at the device level (70-80%) and interconnects (20-30%) [1]. Therefore, a material-device-circuit level analysis is crucial to assess the real gains in energy efficiency and speed of alternative computing approaches compared to their digital CMOS counterparts.

The continuing reduction of the dimensions of conventional devices to the true nano-scale, as well as recent advances in device structures, such as multi-channel GAAFETs, and novel materials (doped high-k dielectrics, 2D materials, etc.), present significant challenges for traditional device TCAD simulation tools. These tools, which predict the device physics and electrical characteristics by solving the semi-classical Boltzmann transport equation through drift-diffusion or particle-based approaches with some quantum corrections, may not be sufficient. For example, IBM's state-of-the-art "3nm" GAAFET devices [3] have a channel height of just 5nm and a physical gate length of about 12nm (Figure 3). At these dimensions, quantum-mechanical effects dominate the device physics, resulting in new effects on channel conductivity due to significant conduction band quantization [2,4]. Therefore, simulators based on fully quantum, first-principles approaches are needed to predict accurately the physics and electrical characteristics of modern devices and/or their possible alternatives.

Predictive, i.e. certifiably accurate within a certain parameter range, device simulations additionally allow one to switch on/off physical effects of a particular kind, for instance inelastic scattering, and observe the consequences for the device characteristics. Such 'beyond-physical' simulations then can reveal individual contributions of different effects and elucidate the real nano-scale device physics and hence new possibilities for device optimization, which otherwise would be very difficult to extract from experimental data or traditional methods. This analysis becomes particularly fruitful when used for studying effects of defects of diverse types on device characteristics [6].

Further development of predictive device simulation tools that allow for extracting electrical characteristics of both the state-of-the-art transistors (such as shown in Figure 3) and novel beyond-CMOS devices (shown in Figure 4) without their actual fabrication, can significantly aid in developing truly energy-efficient computing systems and potentially enabling a whole-system co-design (Figure 2) for the application-driven optimal efficiency.

References

[1] Ng, Len Luet, Yeap, Kim Ho, Goh, Magdalene Wan Ching and Dakulagi, Veerendra. (2022). Power Consumption in CMOS Circuits. https://doi.org/10.5772/intechopen.105717

[2] Mamaluy, D., Mendez, J.P., Gao, X., Misra, S., Revealing quantum effects in highly conductive δ-layer systems. *Commun. Phys.* **4**, 205 (2021). https://www.nature.com/articles/s42005-021-00705-1

[3] N. Loubet *et al*., "Stacked nanosheet gate-all-around transistor to enable scaling beyond FinFET," *2017 Symposium on VLSI Technology*, Kyoto, Japan, pp. T230-T231, 2017. https://doi.org/10.23919/VLSIT.2017.7998183

[4] Mamaluy, D., Mendez, J.P., Titze, M., Arghavani, R., Predictive quantum simulation and device physics of GAAFETs, *MicDAT-2024*, Ibiza, Spain, September 25-27, 2024.

[5] Ward, D. *et al.* Atomic precision advanced manufacturing for digital electronics. *EDFAAO* **22**, 4 (2020).

[6] Mendez, J.P., Misra, S., Mamaluy, D. Influence of imperfections on tunneling rate in δ-layer junctions, Phys. Rev. Applied 20, 054021 (2023). https://arxiv.org/abs/2209.11343

[7] Mendez, J.P., Mamaluy, D. Uncovering anisotropic effects of electric high-moment dipoles on the tunneling current in δ-layer tunnel junctions. *Sci Rep* **13**, 22591 (2023). https://www.nature.com/articles/s41598-023-49777-9

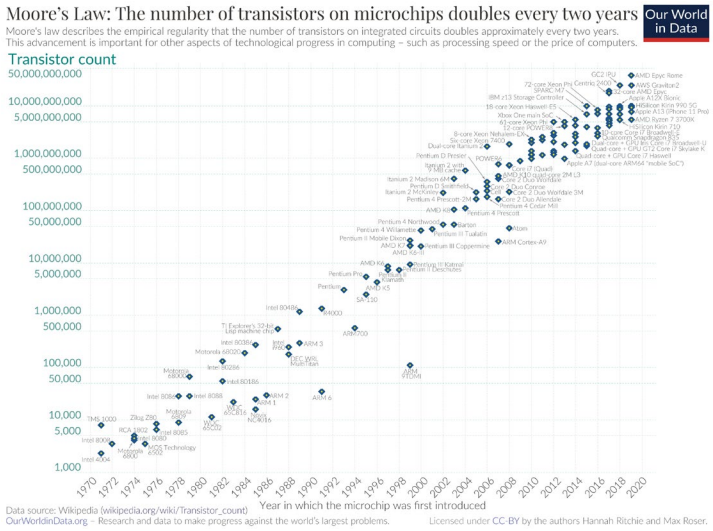


Figure 1. A semi-log plot of transistor counts for microprocessors against dates of introduction. At the same time, the power consumption per chip remains approximately the same, which implies exponential reduction of energy dissipation per transistor.

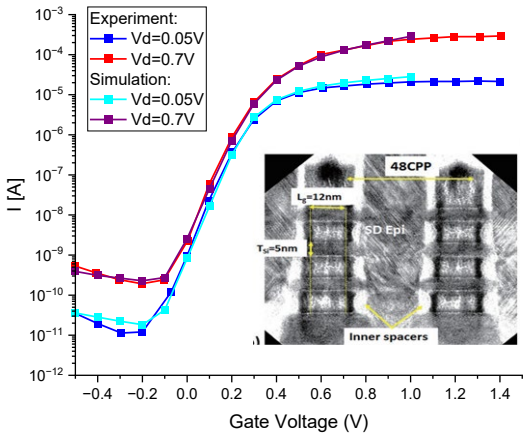


Figure 3. Predictive Simulation of the state-of-the-art GAAFETs devices by IBM [3]: experimental measurements vs simulation data that contain no fitting parameters [4].

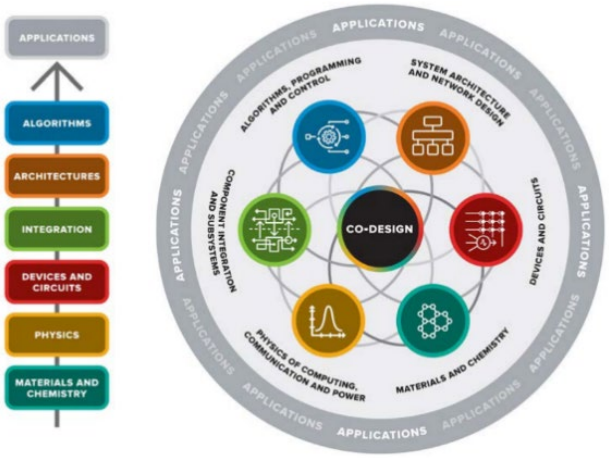


Figure 2. Co-design diagram [*DOE SC BRN for Microelectronics Report (2018)*]. Illustrates the need of a whole system co-design for the application-driven optimal efficiency.

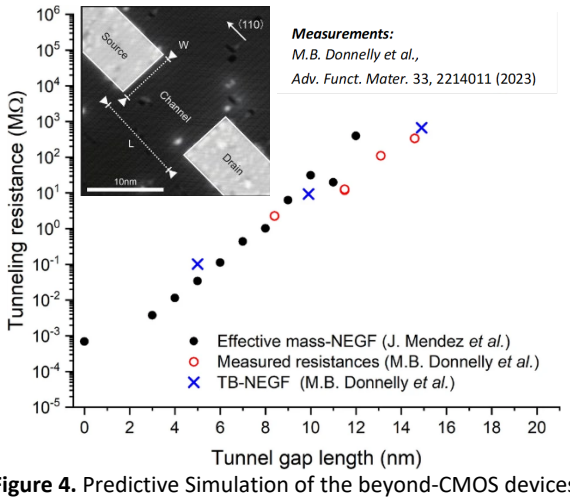


Figure 4. Predictive Simulation of the beyond-CMOS devices based on Atomically Precise Advanced Manufacturing (APAM) technique [5]. Resistances computed without fitting parameters (black circles) [7] and measured experimentally (red circles).

Superconducting Spiking Neural Networks for Cryogenic Near-Sensor Computing

Soumyajit Mandal[1], Grzegorz W. Deptuch[1], Prashansa Mukim[1], Piotr Maj[1]
1: Instrumentation Department, Advanced Technology Research Office, Brookhaven National Laboratory
smandal@bnl.gov

DEADLINE: JULY 22
TOPICS: NEUROMORPHIC COMPUTING, SENSOR INTERFACES, CRYOGENIC DEVICES

1 Motivating Scientific Use Cases

Cryogenic sensors offer unmatched performance for scientific applications. For example, superconducting-nanowire single-photon detectors (SNSPDs) operating at ∼1 K combine single-photon sensitivity with low timing jitter (< 3 ps), high quantum efficiency (> 98%), very low dark count rates ($< 10^{-3}$/sec), broadband responses, fast recovery times (< 10 ns), and monolithic fabrication [3]. Thus, arrays of SNSPDs are widely used in cosmology and dark matter search experiments, for tests of local realism, and for quantum computation and communication [9]. Similarly, arrays of hot-electron bolometers operating at ∼100 mK are widely used for energy-resolved photon detection in astronomy [1]. The development of computing platforms optimized for readout and processing of signals from such arrays is therefore of significant interest for both fundamental science and applications such as quantum computing and simulation.

2 Challenges

Readout of current pulses from arrays of SNSPDs currently relies on using arrays of bulky coaxial cables to connect their outputs to room-temperature electronics, which creates a severe input/output (I/O) bottleneck that prevents scaling to larger arrays (beyond a few kilopixels). Similar bottlenecks arise for other cryogenic sensors, such as kinetic inductance detectors (KID) and transition-edge sensors (TES), and also for arrays of qubits used in quantum computing. Thus, there is a major need for near-sensor signal processing, which includes operations such as multiplexing, coincidence detection, and photon counting, to reduce I/O requirements. While cryogenic CMOS ASICs have been demonstrated to work reliably at ambient temperatures down to ∼50 mK [10, 11], their relatively poor energy efficiency results in a significant thermal load within the cryostat. An alternative approach is based on processing the sensor outputs (e.g., SNSPD voltage pulses) using superconducting electronics [6]. In particular, the development superconducting devices based on Josephson junctions (JJs) enables near-sensor readout using single flux quantum (SFQ) circuits. SFQ circuits store information in the form of magnetic flux quanta and transfer them as quantized voltage pulses, making them of interest as an energy-efficient alternative to transistors for computing in cryogenic environments. However, many years of SFQ-based circuit development have not resulted in widespread adoption of the technology. A fundamental issue is that the fast pulse-based dynamics of SFQ logic is not well-matched to conventional clocked digital processors based on von Neumann architectures [7]. Additionally, the interface between the superconducting sensors and SFQ-based readout circuits remains an I/O bottleneck, with existing solutions relying on coaxial cables or board-level interconnects. The inductance associated with these interfaces also poses severe challenges to signal integrity for high-bandwidth sensors such as SNSPDs. Finally, there is a lack of high-density memory using superconducting devices, so weight storage is a problem.

3 Opportunities

A promising opportunity for overcoming the challenges discussed above is to closely integrate the sensor array with an SFQ-based neuromorphic processor, thus enabling scaling of cryogenic sensors to megapixel-size arrays. Both heterogeneous 3D integration (based on die stacking) and monolithic integration on a single wafer are of interest. These approaches address the issue of sensor-readout bottleneck, but with different trade-offs. Heterogeneous integration simplifies each fabrication process and allows them to be independently optimized, but introduces new assembly challenges. On the other hand, monolithic integration results in more complex fabrication and introduces possible thermal cross-talk between the sensors and readout circuits, but greatly simplifies the assembly process. In both cases, the use of neuromorphic computing has the potential to eliminate the inefficient mapping between SFQ circuits and conventional processor architectures. Instead, asynchronous data streams (e.g., transduced photons) are fed into ae neuromorphic processor in which SFQ pulses act as neural spikes to perform feature extraction, denoising, and other signal processing tasks [4, 12]. The inherent energy-efficiency advantages of SFQ circuits for implementing such

neuromorphic architectures are demonstrated by the rapid growth of interest in SFQ-based spiking neural networks (SNNs) [5]. For example, a recent SFQ-based single-chip SNN containing $\sim 10^5$ JJs achieved an experimental energy efficiency of 32.4 TSOPS/W, which is two orders of magnitude better than state-of-the-art neuromorphic processors in CMOS technology [8]. There are significant research opportunities to build upon this earlier work by developing the fabrication methods, sensor and circuit designs, co-design techniques, and design tools required to closely integrate SFQ-based SNNs with cryogenic sensor arrays.

4 Timeliness and Priority Research Directions

Hardware-aware AI model Design and Training

The first priority research direction we advocate is on deepening our understanding of the unique properties of superconducting devices and how the design and training of AI models can utilize them. For example, mapping SNNs to binary networks is beneficial since it 1) allows neurons to be efficiently implemented using single-bit pulse stream processing, and 2) minimizes on-chip weight storage requirements. Additionally, using asynchronous neural processing elements is desirable for eliminating the common SFQ problem of distributing high-speed clock signals, while hardware reuse (time-division multiplexing) exploits the ultra-high-speed of SFQ circuits to improve compute density. Further research along these lines is required to realize smart cryogenic detectors for applications in high-energy physics, astronomy, and quantum information science.

Automated Optimization and Co-Design of Superconducting Sensors and Circuits

Establishing a quantitative characterization of cryogenic sensors and devices is a necessary step towards realistic simulations. Several simulation tools for SFQ circuits have been developed [2], but have limited capabilities compared to conventional SPICE-based simulators. Validating and expanding these tools to support scripting, additional simulation types, and other types of superconducting circuit families, such as adiabatic quantum-flux-parametrons (AQFP), could be immediate research activities [13]. Another direction is to extend existing work on automated neural network architecture search and various neural network optimization techniques to the superconducting domain. The optimization algorithms, such as Bayesian optimization or reinforcement learning-based optimization, could be reused, but the search space needs to be extended to include superconducting devices and circuits. Such automated optimization also requires the integration of validated SFQ simulation tools into a device-circuit-algorithm co-design flow.

References

[1] Peter K Day et al. "A broadband superconducting detector suitable for use in large arrays". In: *Nature* 425.6960 (2003), pp. 817–821.

[2] Johannes Arnoldus Delport et al. "JoSIM—superconductor SPICE simulator". In: *IEEE Transactions on Applied Superconductivity* 29.5 (2019), pp. 1–5.

[3] Iman Esmaeil Zadeh et al. "Superconducting nanowire single-photon detectors: A perspective on evolution, state-of-the-art, future developments, and applications". In: *Applied Physics Letters* 118.19 (2021).

[4] Md Mazharul Islam et al. "A review of cryogenic neuromorphic hardware". In: *Journal of Applied Physics* 133.7 (2023).

[5] Mustafa Altay Karamuftuoglu et al. "Unsupervised SFQ-Based Spiking Neural Network". In: *IEEE Transactions on Applied Superconductivity* 34.3 (2024), pp. 1–8.

[6] Saeed Khan et al. "Monolithic integration of superconducting-nanowire single-photon detectors with Josephson junctions for scalable single-photon sensing". In: *Superconductor Science and Technology* (2024).

[7] Konstantin K Likharev and Vasilii K Semenov. "RSFQ logic/memory family: A new Josephson-junction technology for sub-terahertz-clock-frequency digital systems". In: *IEEE Transactions on Applied Superconductivity* 1.1 (1991), pp. 3–28.

[8] Zeshi Liu et al. "SUSHI: Ultra-High-Speed and Ultra-Low-Power Neuromorphic Chip Using Superconducting Single-Flux-Quantum Circuits". In: *Proc. IEEE/ACM Intl. Symp. on Microarchitecture*. 2023, pp. 614–627.

[9] Bakhrom G Oripov et al. "A superconducting nanowire single-photon camera with 400,000 pixels". In: *Nature* 622.7984 (2023), pp. 730–734.

[10] SJ Pauka et al. "A cryogenic CMOS chip for generating control signals for multiple qubits". In: *Nature Electronics* 4.1 (2021), pp. 64–70.

[11] R Saligram, A Raychowdhury, and Suman Datta. "The future is frozen: cryogenic CMOS for high-performance computing". In: *Chip* 3.1 (2024), p. 100082.

[12] Michael Schneider et al. "SuperMind: A survey of the potential of superconducting electronics for neuromorphic computing". In: *Superconductor Science and Technology* 35.5 (2022), p. 053001.

[13] Ramy N Tadros et al. "SystemVerilog modeling of SFQ and AQFP circuits". In: *IEEE Transactions on Applied Superconductivity* 30.2 (2019), pp. 1–13.

Hybrid Analog Computing for Intelligent Microscopic Robots.

Marc Miskin, Department of Electrical and Systems Engineering,
University of Pennsylvania, mmiskin@seas.upenn.edu

Topics:

- **Circuits and Systems**
- **Hybrid Analog and Digital Systems**
- **Novel Materials and Devices**

Challenge:

Robots too small to see by eye are emerging as a powerful way to shape and control the microworld, with proposed uses ranging from drug delivery to manufacturing to biomedical research[1], [2], [3], [4]. Yet as they move into real-world applications, microrobots will increasingly face complex and dynamic environments, demanding that they sense, adapt, and use on-board information processing to overcome uncertainty[4]. For sub-mm robots, this level of intelligence has not yet been achieved. Instead, today's microrobots are either externally controlled by bulky laboratory equipment[5], [6], and thus incapable of sensing and responding, or can only change between a limited number of hardcoded states on command[7]. Conventional wisdom holds that sub-mm robots are too small to implement learning algorithms or fully autonomous behaviors. After all, intelligent behaviors are energy and memory intensive, while microrobots are resource starved. In just a few hundred microns, how can one fit power, memory, a microprocessor, and sensors? Even if one could, would it have enough computing power to be useful?

Opportunity:

We believe the idea that microrobots are 'too small to be smart' is increasingly out of date. Instead, key innovations in hybrid design of low-power analog and digital circuits and specialized analog machine learning algorithms suggest that the time is ripe to build tiny machines that can sense, compute, and even learn. If successful, the capacity to process information or adapt will help enable robust and sophisticated microrobots that operate in realistic environments with minimal human oversight.

Intelligent microscopic robots would represent a dramatic improvement over today's state of the art. Current designs either run fixed control laws that use prespecified notions of the environment (limiting adaptability) or use try to leverage external computing resources to implement learned behaviors (limiting autonomy). Both approaches struggle with noisy sensor data and latency, even in controlled laboratory settings, because sensing and computing systems are entirely off-robot. Furthermore, moving computation to the robot would lower the barrier to entry in microrobotics by dramatically reducing laboratory overhead. In principle, an intelligent microrobot could be given digital instructions and left to carry out its task autonomously, paving the way for widespread use.

Timeliness:

Two convergent trends point to the feasibility of microscale robots with onboard computation. The first is a sustained reduction in computing size ("Bell's law"), which heralds the emergence of sub-mm computing systems in this decade[8]. Indeed, several sub-mm computing systems have appeared in the literature incorporating the core capabilities of an autonomous microrobot, most notably on-board sensing, power management, memory, and processing capabilities[9], [10], [11]. While in the past, Bell's law was driven through scaling of semiconductor components, these recent reductions in size have been realized by developing and optimizing mixed analog/digital architectures. Such designs, which leverage the low-power, small size advantages of analog circuitry in conjunction with the robust, controlled nature of digital electronics, are extremely well suited to meeting the constraints of microrobotics. Further research should specifically address how to make the most of the time/space/energy tradeoff associated to computing and sensing[12], maximizing robotic function in small form-factors.

The second major trend favoring smart microrobots is the emergence of new classes of machine learning algorithms and hardware tailored to resource starved, distributed systems. Largely a consequence of edge and IoT applications, these approaches compensate for low computational power by distributing learning across agents or leveraging analog device structures for energy savings in hardware. Examples of the former include federated learning[13], and distributed reinforcement learning[14], while examples of the later including emerging memristor arrays[15], [16] or analog-based neural computing[17], [18], [19]. Microrobots are well positioned to adopt these strategies in the real world: large numbers of agents are common in microrobotics as devices can be mass

manufactured[6], [7], communication to either central servers or between agents is easier to execute that crunching numbers on each agent, robots are already built using back-end semiconductor processing (paving the way for novel material integration), and imperfections in analog hardware can broadly be tolerated provided they do not impact robot operation (i.e., unlike a hardware accelerator, the parameters of the learning system itself are unimportant, only the resulting behavior matters). Future work should develop an ecosystem where algorithms are co-developed with hardware for robotic capabilities at the microscale. Success in this space could yield impact across robotics by developing optimized analog chipsets for robotics tasks (e.g., SLAM on a chip[20]), with microrobotics serving as a testbed.

References:

[1] H. Ceylan, J. Giltinan, K. Kozielski, and M. Sitti, "Mobile microrobots for bioengineering applications," *Lab on a Chip*, vol. 17, no. 10, Art. no. 10, 2017.

[2] S. Palagi and P. Fischer, "Bioinspired microrobots," *Nature Reviews Materials*, vol. 3, no. 6, Art. no. 6, 2018.

[3] A. T. Liu *et al*., "Colloidal robotics," *Nature materials*, vol. 22, no. 12, pp. 1453–1462, 2023.

[4] T.-Y. Huang, H. Gu, and B. J. Nelson, "Increasingly intelligent micromachines," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, no. 1, pp. 279–310, 2022.

[5] W. S. Trimmer, "Microrobots and micromechanical systems," *Sensors and actuators*, vol. 19, no. 3, Art. no. 3, 1989.

[6] M. Z. Miskin *et al*., "Electronically integrated, mass-manufactured, microscopic robots," *Nature*, vol. 584, no. 7822, Art. no. 7822, 2020.

[7] M. F. Reynolds *et al*., "Microscopic robots with onboard digital control," *Science Robotics*, vol. 7, no. 70, p. eabq2296, 2022, doi: 10.1126/scirobotics.abq2296.

[8] Y. Lee, D. Sylvester, and D. Blaauw, "Circuits for ultra-low power millimeter-scale sensor nodes," in *2012 Conference Record of the Forty Sixth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, IEEE, 2012, pp. 752–756.

[9] L. Xu *et al*., "A 210x340x50 micrometer Integrated CMOS System for Micro-Robots with Energy Harvesting, Sensing, Processing, Communication and Actuation," in *2022 IEEE International Solid- State Circuits Conference (ISSCC)*, 2022, pp. 1–3. doi: 10.1109/ISSCC42614.2022.9731743.

[10] Y. Lee *et al*., "A modular 1 mm^3 die-stacked sensing platform with low power I^2C inter-die communication and multi-modal energy harvesting," *IEEE Journal of Solid-State Circuits*, vol. 48, no. 1, Art. no. 1, 2013.

[11] I. Lee *et al*., "mSAIL: milligram-scale multi-modal sensor platform for monarch butterfly migration tracking," in *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, 2021, pp. 517–530.

[12] R. Sarpeshkar, "Analog versus digital: extrapolating from electronics to neurobiology," *Neural computation*, vol. 10, no. 7, pp. 1601–1638, 1998.

[13] L. Li, Y. Fan, M. Tse, and K.-Y. Lin, "A review of applications in federated learning," *Computers & Industrial Engineering*, vol. 149, p. 106854, 2020.

[14] X. Qiu, W. Zhang, W. Chen, and Z. Zheng, "Distributed and collective deep reinforcement learning for computation offloading: A practical perspective," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 5, pp. 1085–1101, 2020.

[15] H. Kim, M. Mahmoodi, H. Nili, and D. B. Strukov, "4K-memristor analog-grade passive crossbar circuit," *Nature communications*, vol. 12, no. 1, p. 5198, 2021.

[16] M. Rao *et al*., "Thousands of conductance levels in memristors integrated on CMOS," *Nature*, vol. 615, no. 7954, pp. 823–829, 2023.

[17] S. Dillavou, B. D. Beyer, M. Stern, A. J. Liu, M. Z. Miskin, and D. J. Durian, "Machine learning without a processor: Emergent learning in a nonlinear analog network," *Proceedings of the National Academy of Sciences*, vol. 121, no. 28, p. e2319718121, 2024.

[18] B. Scellier and Y. Bengio, "Equilibrium propagation: Bridging the gap between energy-based models and backpropagation," *Frontiers in computational neuroscience*, vol. 11, p. 24, 2017.

[19] T. P. Xiao, C. H. Bennett, B. Feinberg, S. Agarwal, and M. J. Marinella, "Analog architectures for neural network acceleration based on non-volatile memory," *Applied Physics Reviews*, vol. 7, no. 3, 2020.

[20] M. R. Gkeka *et al*., "Reconfigurable system-on-Chip architectures for robust visual SLAM on humanoid robots," *ACM Transactions on Embedded Computing Systems*, vol. 22, no. 2, pp. 1–29, 2023.

Title: Where is the "big win" in analog computing hiding?

Authors: Shashank Misra (presenting author; smisra@sandia.gov), Christopher R. Allemang (crallem@sandia.gov), Andrew D. Kent (andy.kent@nyu.edu), J. Darby Smith (jsmit16@sandia.gov), and Tzu-Ming Lu (tlu@sandia.gov)

Topics: novel materials and devices, circuits and systems, error correction methods for analog computation, hybrid analog and digital systems, probabilistic computing

Challenge: Amdahl's law applies to analog computing. To overcome the cost of adoption, analog computing systems likely need to enable a 10-100x improvement in performance or energy efficiency in a class of applications, or to make an intractable problem solvable through improved scaling. The evidence for this comes from outside of analog computing - a business truism about digital architectures (transition from scaling processor frequency to core number to moving to accelerators) for the former, and quantum computing for the latter. *Most arguments advocating for analog computing compare the computational accuracy and energy cost of one or a small system of analog devices compared to a software implementation of the mathematics performed by the analog devices in some representative miniature application (miniapp).* This both overestimates the computational capability of the analog system – miniapps tend to place greater computational intensity in the aspect that is being accelerated – and it underestimates the cost of the analog system – estimates often ignore the transistor overhead of analog infrastructure and need to also do digital logic. *This strategy is fundamentally flawed because it violates Amdahl's law – the limitation to improvements produced by analog computation come from the parts that remain unaccelerated.* This makes the hype of analog computing, centered around orders of magnitude of efficiency improvements, ring hollow.

We illustrate these points with assessments from a DOE-SC Microelectronics Codesign Call project called COINFLIPS, which sought to research the viability of probabilistic computing. We used hardware-generated random bitstreams from beyond CMOS devices in a Monte Carlo-based event generator for high energy particle collisions. The devices were chosen to be simple, producing a random low or high output given an input pulse and thermal fluctuations, to promote fine-grained integration with digital logic and avoid a von Neumann bottleneck [1,2]. We found that the microelectronic devices used to generate the bitstreams consumed far less power than the simple circuit elements required to turn those bitstreams into uniform random numbers, that is, a comparator, a logical exclusive-or for error correction, and a shift register [3]. *At the device level, the energy consumption was determined entirely by the transistors required to make use of the analog device; this is an analog device version of Amdahl's law.* Still, the resulting circuit should function as a hardware true random number generator that is several orders of magnitude more efficient than the software-based pseudo-random number generator used by the event generator. Unfortunately, in profiling the full calculation, we find that only up to 25-50% of the CPU time is spent on random number generation, with the remainder being integer, logic, and floating-point operations [4]. Codesign work between our devices and algorithm teams produced a new scheme that replaced performing calculations on uniform random samples with more complex sampling, effectively replacing digital logic with more complex stochastic circuits. *A factor of 10x improvement was forecasted by adopting both the changed hardware and changed algorithm, with virtually no improvement from making only one of the two changes. This results from the cost of an analog-accelerated calculation being determined by the remaining digital operations – i.e. Amdahl's law.* We summarize our ideas in the figure below. **We conclude that for analog computing to 'make it,' Amdahl's law requires eliminating transistor/digital logic usage in all the repetitive parts of a system/computation.**

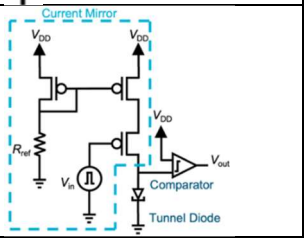
Opportunity: Analog systems need to become breathtakingly more complex, while target applications need to become much simpler. Perhaps based on the success of digital architectures, analog computing researchers have focused on simple analog devices which can address complex applications through a combination of scaling up and integration with digital computation. Breaking apart problems into simple component operations which are strung together is not only instinctive because it has been so productive in information science, but it is convenient because the components which cannot be made analog, from lack of resources, can instead be done digitally rather simply. Taking the COINFLIPS example, the circuit cost can be driven down if we had the resources to discover a way to take an analog exclusive-or and construct an analog shift register. More generally, the opportunity is in systems which combine many different modes of control of some physical phenomenon through easy integration of their required materials, and where the local control mechanism is derived from analog concepts like feedback/equilibrium and not digital concepts like programming. *Fortunately, physical systems that are this complex*

have generally not been considered for analog computing before, and they present a plausible path to a solely analog computation that shuts digital out of the repetitive parts of the calculation.

The main reason to be skeptical of such complicated analog systems is that their computational utility is unknown. The underlying physics is often not well-understood, and mathematical insights must be developed simultaneously with algorithms to isolate which phenomena are computationally useful. That makes requiring discoveries to be contextualized as a big step towards a lofty goal (say, neuronal behavior in a device and machine learning-based image recognition) quite risky. At the same time, the temptation for the application to be either saccharine (e.g., the system simulates another physical system efficiently but with uncontrolled approximations) or vapid (e.g. the system tunably implements a specific piece of mathematics, like a filter) must be avoided for the outcome to be worthwhile. *Thus, there is an under-appreciated opportunity for applications which are shallower in complexity but still wide in impact.* This also enables examining a wider range of candidate problems, accelerating the identification of which have repetitive kernels that map well to analog computation. The analog computer needs to be efficient enough to aggressively drive down the cost of the recurring part of the calculation, while digital computation is only engaged in the beginning and end to produce a more accurate answer, e.g., through how the problem is split up or averaging.

Timeliness: Mitigation for Quantum Winter. To be blunt, analog computing as envisioned above partly mitigates the risk of the commercial failure of quantum computing in the medium term. Quantum computing has no widely agreed-upon path to a commercial application, despite enormous private sector investment. Meanwhile, there are significant overlaps in the underlying skills needed for working in both areas. Work in complex analog systems can spawn commercially viable off-ramps for a quantum workforce if there is a contraction.

Figures

Level	Energy cost	Description	Notes
Device	< 1 pJ	Single magnetic tunnel junction/ tunnel diode	
Circuit	~ 1 nJ	Produces random bitstream	

Number of simple digital operations required to draw a 32 bit sample.

Distribution	Non-uniform	Non-uniform	Non-uniform	Non-uniform
RNG	PRNG	TRNG	PRNG	TRNG
Approach	Rejection sampling	Rejection sampling	Sampling tree	Sampling tree
Operations	242	202	154	24

References

[1] Laura Rehm *et al.*, “Stochastic Magnetic Actuated Random Transducer Devices Based on Perpendicular Magnetic Tunnel Junctions”, Phys. Rev. Appl. 19, 024035 (2023).

[2] James B. Aimone and Shashank Misra, “Will Stochastic Devices Play Nice With Others in Neuromorphic Hardware: There’s More to a Probabilistic System Than Noisy Devices”, IEEE EDM 1, 50-56 (2023).

[3] Ankit Shukla *et al.*, “A True Random Number Generator for Probabilistic Computing using Stochastic Magnetic Actuated Random Transducer Devices”, 224th International Symposium on Quality Electronic Design (ISQED) 2023, 1-10 (2023).

[4] Shashank Misra *et al.*, “Probabilistic Neural Computing with Stochastic Devices”, Advanced Materials 35, 2204569 (2023).

Analog/Hybrid Co-Design Flow Methodology

Benjamin Parpillon[1,2], Amit Trivedi[2], Haitong Li[3], Jennifer Hasler[4], Farah Fahim[1]
bparpill@fnal.gov, amitrt@uic.edu, haitongli@purdue.edu, jennifer.hasler@ece.gatech.edu, farah@fnal.gov
[1]Fermi National Accelerator Laboratory, [2]University of Illinois Chicago, [3] Purdue University, [4]Georgia Institute of Technology

1 Topic

Our position paper discusses hybrid analog and digital systems design methodology.

2 Challenge

The rapid growth of analog sensor data has outpaced intelligent processing capabilities in many domains, including scientific experiments, causing an *analog data deluge* that obscures valuable information. At the LHC, Petabytes per second of data are generated thus requiring very high-speed offline filtering to avoid data pileups. A 2021 report [1] depicted in Fig.1, illustrates the total amount of data produced by LHC collisions in one year exceeded the total size of files ever stored on Amazon cloud storage services by approximately two order of magnitude. More than 90% of collision data is generated by a single detector system the **silicon pixel detectors**.

The data bottleneck is by far the greatest challenge faced by the HEP community. Currently, more than 99.995% of collision data is filtered out during offline data analysis looking only for rare interactions hoping to discover new Physics. The Level 1 trigger is mostly responsible for large amount of data to be rejected on-sensor reducing the information transfer from PBps to TBps. The innermost layers of the CMS/ ATLAS detectors currently do not contribute to the Level 1 trigger.

3 Opportunity

The data bottleneck challenge creates a unique opportunity for the HEP community to develop new computing co-design paradigm and methodology.

Current digital systems are unable to handle data rates needed at the LHC due to conversion overheads, limited parallelism, and resource intensiveness [2]. *On-sensor analog deep learning* can eliminate data conversion and storage overheads by filtering non-essential signals at the acquisition stage. Similarly, at the

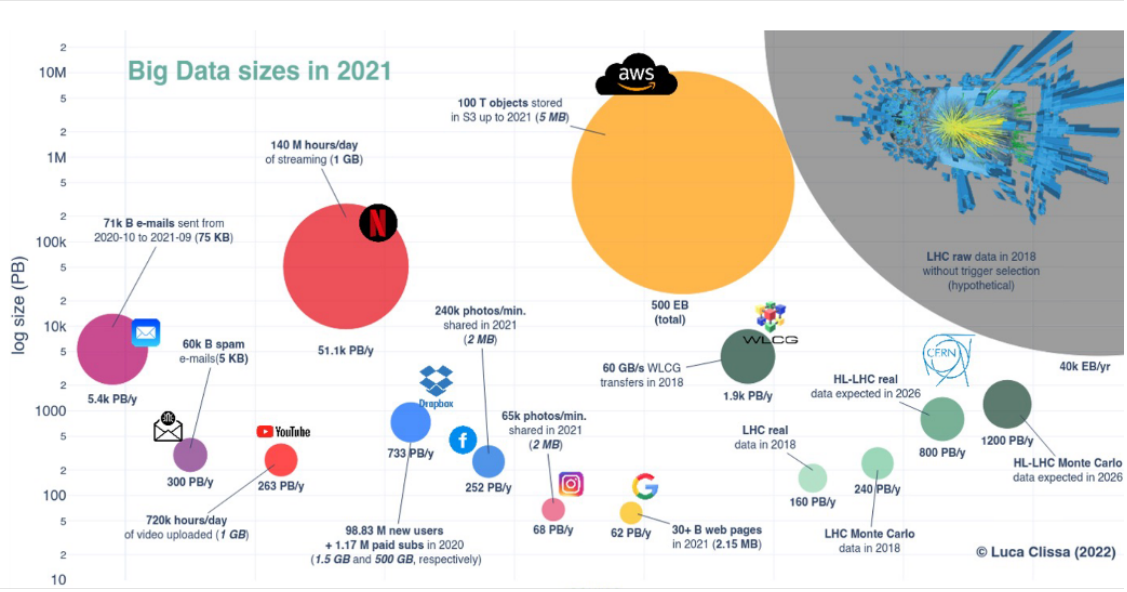


Figure 1: **Big Data in HEP:** Orders of magnitude involved in different data sources for several big data players. The area of each bubble represents the amount of data streamed, hosted or generated. The accompanying text annotations emphasize the crucial factors considered in the estimation process. Average per-unit sizes are indicated in parentheses, where italic denotes measures derived from reasonable assumptions due to the absence of available references

signal processing stage, analog deep learning can leverage *non-von Neumann architectures* to minimize data movements and employ *physics-based computing*, such as using Kirchoff's law for summation by representing operands in the charge or current domain, to maximize energy efficiency [3]. Thereby processing signals closer to the source in the analog domain offers significant benefits in performance, speed, area, and power, and can enable novel signal processing paradigms such as asynchronous real-time waveform analysis and direct signal-to-inference capabilities.

The flow need to address three sections illustrated in Fig. 2 :

- A first **pre-silicon** stage algorithmic flow to convert bit representations of software learning models to analog representations (such as charge, time, or current) suited for silicon placement. The algorithm also needs to solve optimal mapping of deep learning layers to a system of analog crossbars while maximizing throughput by concurrent processing, minimizing data transmission lengths and rates, maintaining load balancing, minimizing model load cycles from off-chip memories, *etc.*
- A second **on-silicon** stage that leverages novel mixed-signal processing architectures as well CMOS based and CMOS+X (FeFET, Floating Gates, PRAM, ReRAM, MRAM...) devices to fully exploit analog processing capabilities while addressing challenges such as design complexity and process variability.
- Finally, a third **post-silicon** stage is needed to address the need for continual monitoring and correction of the computing substrate against chip-to-chip, across-chip, and time-varying degradation such as process variability, aging, and temperature-induced variations.

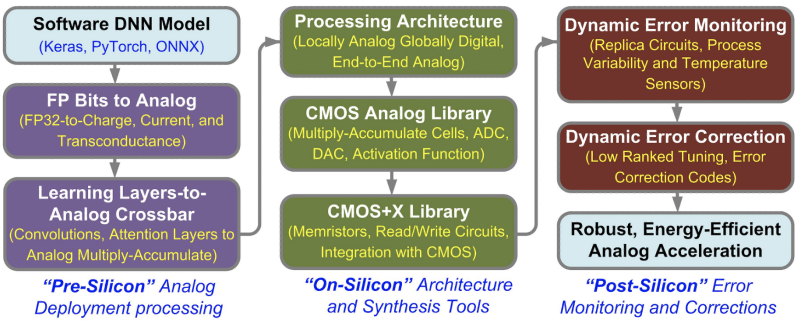


Figure 2: Co-designed analog acceleration flow consisting of pre-, on-, and post-silicon stages for reliable software-to-analog conversion, scalable synthesis and design, and runtime error monitoring and corrections.

4 Timeline

The AI/ML community for electronics design of HEP applications has now reached critical mass. The demand and need for increasingly more complex AI/ML models require new computing paradigm and design methodologies that supports analog/hybrid implementations. Recently, we introduced the support of Siemens Catapult HLS [4] as a backend of `hls4ml` to target specifically the ASIC flow. The significance for the industry of our framework has brought us to collaborate with Siemens EDA to release *Catapult AI NN* [5].

References

[1] Luca Clissa, Mario Lassnig, Lorenzo Rinaldi. How big is Big Data? A comprehensive survey of data production, storage, and streaming in science and industry. *Frontiers Big Data*.

[2] Igor L Markov. Limits on fundamental limits to computation. *Nature*, 512(7513):147–154, 2014.

[3] Jennifer Hasler. Opportunities in physical computing driven by analog realization. In *2016 IEEE international conference on rebooting computing (ICRC)*, pages 1–8. IEEE, 2016.

[4] Siemens. Catapult HLS. `https://eda.sw.siemens.com/en-US/ic/ic-design/high-level-synthesis-and-verification-platform`.

[5] Siemens Digital Industries Software. Catapult ai nn simplifies development of ai accelerators, May 2024.

Molecular Programming for Biological Circuit Design

Niles A. Pierce,[1,2,*] Mark E. Fornace[3], Richard M. Murray,[1,2] Lulu Qian,[1] Erik Winfree[1,2]
[1]*Division of Biology & Biological Engineering, California Institute of Technology*
[2]*Division of Engineering & Applied Science, California Institute of Technology*
[3]*Lawrence Berkeley National Laboratory*
niles@caltech.edu, mefornace@lbl.gov, murray@cds.caltech.edu, luluqian@caltech.edu, winfree@caltech.edu

Topic: biological circuit design

Challenge: Biological circuits encoded in the genome of each organism direct development, maintain integrity in the face of attacks, operate intricate molecular machinery, control responses to environmental stimuli, and sometimes malfunction to cause disease. The emerging discipline of molecular programming is jointly inspired by these remarkable programmable molecular circuits and devices that orchestrate life and by the transformative impact of computer science on technology and society. Molecular programming researchers seek to develop the principles and practice for a new engineering discipline that will enable the function of molecules to be programmed with the ease and rigor with which computers are programmed today, while achieving the sophistication, complexity, and robustness evident in the programmable DNA, RNA, and protein machinery of biology. However, while there has been foundational progress in the emerging field of molecular programming over the last two decades, our ability to program molecular circuits *in vitro* and *in vivo* remains modest compared to the virtuosity exhibited in biology. *Example challenges:* 1) computational design of programmable regulators for scalable synthetic biology, 2) interfacing sensitive and robust synthetic programmable regulators with endogenous inputs and outputs, 2) simulating and designing the kinetic molecular interactions of programmable regulators operating within dynamic circuits, 3) engineering synthetic biological circuits that are energy efficient, 4) engineering feedback between biological circuits and mechanical force.

Opportunities: Over the coming decades, the emerging fields of molecular programming, dynamic nucleic acid nanotechnology, and synthetic biology are poised to generate transformative programmable molecular and cellular technologies addressing challenges to science and society ranging from environmental monitoring and biosphere engineering to diagnosis and treatment, and from renewable energy to sustainable manufacturing. *Example opportunities:* 1) research tools and therapeutics that leverage programmable cell-selective regulation, 2) sensitive, programmable, instrument-free at-home diagnostics, 3) multi-species biological circuits for environmental monitoring, 4) ecological engineering to enhance carbon sequestration, 5) synthetic development.

To take advantage of these opportunities, effective analog computation must be scalable, robust, and efficient. These requirements call for interdisciplinary research incorporating novel methodologies from computer science, bioengineering, and mathematics. Key approaches for biological circuit design will include: 1) thermodynamic analysis via dynamic programming algorithms, 2) computational acceleration via ensemble decomposition and other methods, 3) efficient and robust Markov chain Monte Carlo simulation of molecular circuit kinetics including automated coarse-graining, 4) chemical reaction network simulation and design, and 5) high-quality and user-friendly software.

Timeliness: Over the last decade, there has been significant progress demonstrating proofs-of-principle engineering programmable regulators and circuits that operation *in vitro* and *in vivo*. Significant progress has also been made in developing algorithms to support systematic engineering of these programmable molecular devices and systems (e.g., NUPACK; www.nupack.org). A broad and enduring multi-decade effort is required to progress the state-of-the-art to the point where robust and versatile technology platforms are in place across a range of applications for the benefit of science and society. The timeliness of

these efforts is reflected in the number of companies (from pre-seed startups to multi-national corporations) that are building new teams to work on development of programmable molecular technologies. Current efforts and capabilities represent the tip of the iceberg, yet already reflect a markedly different outlook from 10 years ago, and are unrecognizable from 20 years ago.

References

Algorithms for Programmable Regulator and Circuit Engineering

Wolfe, B. R.; Porubsky, N. J.; Zadeh, J. N.; Dirks, R. M.; Pierce, N. A. Constrained Multistate Sequence Design for Nucleic Acid Reaction Pathway Engineering. *Journal of the American Chemical Society* **2017**, *139*, 3134–3144.

Fornace, M. E.; Porubsky, N. J.; Pierce, N. A. A Unified Dynamic Programming Framework for the Analysis of Interacting Nucleic Acid Strands: Enhanced Models, Scalability, and Speed. *ACS Synth. Biol.* **2020**, *9* (10), 2665–2678.

Badelt, S.; Grun, C.; Sarma, K. V.; Wolfe, B.; Shin, S. W.; Winfree, E. A Domain-Level DNA Strand Displacement Reaction Enumerator Allowing Arbitrary Non-Pseudoknotted Secondary Structures. *Journal of The Royal Society Interface* **2020**, *17* (167), 20190866.

Fornace, M. E.; Huang, J.; Newman, C. T.; Porubsky, N. J.; Pierce, M. B.; Pierce, N. A. NUPACK: Analysis and Design of Nucleic Acid Structures, Devices, and Systems. *chemrxiv-2022-xv98l*, **2022**.

Jones, T. S.; Oliveira, S. M. D.; Myers, C. J.; Voigt, C. A.; Densmore, D. Genetic Circuit Design Automation with Cello 2.0. *Nature Protocols* **2022**, *17* (4), 1097–1113.

Synthetic Programmable Regulators and Biological Circuits

Green, A. A.; Silver, P. A.; Collins, J. J.; Yin, P. Toehold Switches: De-Novo-Designed Regulators of Gene Expression. *Cell* **2014**, *159* (4), 925–939.

Chappell, J.; Takahashi, M. K.; Lucks, J. B. Creating Small Transcription Activating RNAs. *Nat Chem Biol* **2015**, *11* (3), 214–220.

Srinivas, N.; Parkin, J.; Seelig, G.; Winfree, E.; Soloveichik, D. Enzyme-Free Nucleic Acid Dynamical Systems. *Science* **2017**, *358* (6369), eaal2052.

Thubagere, A. J.; Li, W.; Johnson, R. F.; Chen, Z.; Doroudi, S.; Lee, Y. L.; Izatt, G.; Wittman, S.; Srinivas, N.; Woods, D.; Winfree, E.; Qian, L. A Cargo-Sorting DNA Robot. *Science* **2017**, *357* (6356), eaan6558.

Li, J.; Green, A. A.; Yan, H.; Fan, C. Engineering Nucleic Acid Structures for Programmable Molecular Circuitry and Intracellular Biocomputation. *Nat. Chem.* **2017**, *9*, 1056.

Siu, K.-H.; Chen, W. Riboregulated Toehold-Gated gRNA for Programmable CRISPR–Cas9 Function. *Nat. Chem. Biol.* **2019**, *15*, 217–220.

Hanewich-Hollatz, M. H.; Chen, Z.; Hochrein, L. M.; Huang, J.; Pierce, N. A. Conditional Guide RNAs: Programmable Conditional Regulation of CRISPR/Cas Function in Bacterial and Mammalian Cells via Dynamic RNA Nanotechnology. *ACS Cent. Sci.* **2019**, *5* (7), 1241–1249.

Oesinghaus, L.; Simmel, F. C. Switching the Activity of Cas12a Using Guide RNA Strand Displacement Circuits. *Nature Communications* **2019**, *10* (1), 1–11.

Hong, F.; Ma, D.; Wu, K.; Mina, L. A.; Luiten, R. C.; Liu, Y.; Yan, H.; Green, A. A. Precise and Programmable Detection of Mutations Using Ultraspecific Riboregulators. *Cell* **2020**, *180* (5), 1018-1032.e16.

Hochrein, L. M.; Li, H.; Pierce, N. A. High-Performance Allosteric Conditional Guide RNAs for Mammalian Cell-Selective Regulation of CRISPR/Cas. *ACS Synthetic Biology* **2021**, *10* (5), 964–971.

Jiang, K.; Koob, J.; Chen, X. D.; Krajeski, R. N.; Zhang, Y.; Volf, V.; Zhou, W.; Sgrizzi, S. R.; Villiger, L.; Gootenberg, J. S.; Chen, F.; Abudayyeh, O. O. Programmable Eukaryotic Protein Synthesis with RNA Sensors by Harnessing ADAR. *Nature Biotechnology* **2023**, *41* (5), 698–707.

Noise-Resilient Analog Computing through AI-Circuit-Material Co-Design

Yihui Ren (yren@bnl.gov)[1], Soumyajit Mandal[2], Chang-Yong Nam[3], Shubha Kharel[1], Shinjae Yoo[1]
1: AI Department, Computational Science Initiative, Brookhaven National Laboratory
2: Instrumentation Department, Advanced Technology Research Office, Brookhaven National Laboratory
3: Center for Functional Nanomaterials, Brookhaven National Laboratory

Topics: In-memory analog computing, AI-Circuit-Material Co-Design

1 Motivating Scientific Use Cases

As contemporary deep learning-based AI models have demonstrated their efficacy, integrating them within scientific workflows and deploying them at the edge have gained strong interest in many scientific applications. In particle physics experiments, the highest data rates occur at the outputs of tracking detectors. For example, the time-projection chamber in the SPHENIX experiment at the Relativistic Heavy-Ion Collider (RHIC) produces raw data at a rate of 1Tb/s [7], while the future ePIC detector in the Electron-Ion Collider (EIC) will feature a 16-billion channel tracking detector [12]. In future high-energy physics experiments such as the High-Luminosity Large Hadron Collider (HL-LHC), data rates can reach 1 Pb/s on average, while the silicon tracker for the Future Circular Collider for hadrons (FCC-hh) will have ∼20 billion channels and a zero-suppressed data rate of 10 Pb/s [1]. Additionally, future beamlines at the National Synchrotron Light Source (NSLS-II) will generate time-resolved high-resolution 2D X-ray scattering data at 100Gb/s. Traditional ways to process such high-bandwidth data streams are based on triggering systems that down-select data using heuristic methods. Alternatively, AI-based real-time processing methods have shown promising results, such that integration of AI technologies near the detector has been identified as a priority research direction by the HEP community [6]. Most recent efforts on such near-detector AI have focused on optimizing conventional artificial neural networks (ANNs) for edge deployment [9] using either complementary-metal-oxide-semiconductor (CMOS) application-specific integrated circuits (ASICs) or field-programmable gate array (FPGA) [4]. However, few have considered the added power and die area consumption of such "smart detectors". Analog in-memory computing (AIMC) provides a promising alternative for near-detector AI inference due to three major reasons: 1) lower power consumption; 2) ability to use radiation-hard materials; and 3) avoiding the need for energy-intensive digitization of raw sensor data.

2 Challenges

AIMC has been rapidly developing as a research topic. However, its adoption in real applications has been lacking due to many challenges. Firstly, the materials used for implementing AIMC are diverse and rapidly evolving. Based on their working principles [8], we can roughly divide them into resistive random-access memory (RRAM), phase-change memory (PCM), magnetoresistive random-access memory (MRAM), and optical interferometers. Each type has its own advantages and disadvantages. Secondly, these materials and devices usually exhibit intrinsic stochasticity, mismatch, and limited bandwidth. The circuitry designed around these devices can introduce additional noise, further degrading computational accuracy. As a result, AIMC has limited dynamic range (DR), such that non-linear functions such as exponentiation do not work well and have to be dealt with using digital circuits. Additionally, the AI model weights are programmed as conductances, whose values fluctuate and drift over time. Thirdly, the energy efficiency of AIMC systems is severely degraded by the analog-to-digital and digital-to-analog converters (ADCs and DACs) required to interface them with digital processors and/or memory. Finally, realization of a practical analog computing solution requires a close collaboration between material scientists, circuit designers, AI experts, and domain scientists. The design space is vast, spanning material choices, crossbar array dimensions, device design, data converters, circuit architecture, numeric representations, AI algorithms, and noise-aware training.

3 Opportunities

As microelectronics and AI have been identified as topics of national strategic importance [3, 5], long-term large research collaborations should be fostered to support energy-efficient computing for scientific applications, which would be most benefited by AIMC if associated challenges could be addressed. With respect to the challenges mentioned above, three corresponding research opportunities and suitable for pursuit by such collaborations. Firstly, we need to establish a quantifiable and, most desirably, unified representation of different materials and their characteristics such as stochasticity and non-ideality. These quantifiable

characteristics of materials and devices will facilitate a simulation framework for evaluating feasibility of new materials and devices for a given AI model architecture and specific scientific applications. Secondly, to tackle the limitation of these quantifiable characteristics of materials and devices, a hardware-aware and noise-resilient AI algorithm design and training framework will be needed. Thirdly, a comprehensive and all-inclusive co-design and automated optimization framework will be needed to help algorithm designers, circuit engineers, and material scientists in navigating the complex design landscape.

4 Timeliness and Priority Research Directions

Hardware-aware AI model Design and Noise-Resilient Training

The first priority research direction we advocate is on deepening the understanding of stochasticity of analog computing materials and devices and how the training of AI models can mitigate or make use of them. Stochasticity has played a crucial role in AI, ranging from the foundational stochastic gradient descent (SGD) algorithm that shuffles training data into random mini-batches, to dropout layers and reduced precision arithmetic. Such perturbation mimics a noisy vector-matrix multiplication (VMM) unit. Similar ideas of injecting noise into neural network training [10] have been used for regularizing the process of ANN training. Existing work on AIMC [11, 2] that integrates noise during AI model training has shown promising results. However, a deeper understanding of the impact of choosing various noise characteristics, AI model architectures, crossbar array dimensions, and fundamental materials characteristics is required.

Establishing Co-Design Space and Automated Design Optimization

Establishing a quantitative characterization of analog computing materials and devices is a necessary step towards a realistic simulation. Several AIMC simulation tools have been developed. Validating and expanding them for other types of materials and circuits, such as spiking neural networks (SNNs), could be immediate research activities. Another direction is to extend the line of work on automated neural network architecture search and various neural network optimization techniques. The optimization algorithms, such as Bayesian optimization or reinforcement learning-based optimization, could be reused, but the search space needs to be extended to include materials, devices, and circuits. Such automated optimization also requires the integration of validated AIMC simulation tools into the device-circuit-algorithm co-design flow.

References

[1] Asmâa Abada et al. “FCC-hh: The hadron collider”. In: *The European Physical Journal Special Topics* 228.4 (2019), pp. 755–1107.

[2] Stefano Ambrogio et al. “An analog-AI chip for energy-efficient speech recognition and transcription”. In: *Nature* 620.7975 (2023), pp. 768–775.

[3] *Chips and Science Act*. https://en.wikipedia.org/wiki/CHIPS_and_Science_Act. Accessed: 2024-07-15.

[4] Giuseppe Di Guglielmo et al. “A reconfigurable neural network ASIC for detector front-end data compression at the HL-LHC”. In: *IEEE Transactions on Nuclear Science* 68.8 (2021), pp. 2179–2186.

[5] *Executive Order on Artificial Intelligence*. https://en.wikipedia.org/wiki/Executive_Order_14110. Accessed: 2024-07-15.

[6] HEP. “DOE Basic Research Needs Study on High Energy Physics Detector Research and Development”. In: *DOE Basic Research Needs* (2020).

[7] Yi Huang et al. “Efficient data compression for 3d sparse tpc via bicephalous convolutional autoencoder”. In: *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE. 2021, pp. 1094–1099.

[8] Daniele Ielmini and H-S Philip Wong. “In-memory computing with resistive switching devices”. In: *Nature Electronics* 1.6 (2018), pp. 333–343.

[9] S Miryala et al. “Waveform processing using neural network algorithms on the front-end electronics”. In: *Journal of Instrumentation* 17.01 (2022), p. C01039.

[10] Nitish Srivastava et al. “Dropout: a simple way to prevent neural networks from overfitting”. In: *The Journal of Machine Learning Research* 15.1 (2014), pp. 1929–1958.

[11] Weier Wan et al. “A compute-in-memory chip based on resistive random-access memory”. In: *Nature* 608.7923 (2022), pp. 504–512.

[12] Ferdinand Willeke and J Beebe-Wang. *Electron ion collider conceptual design report 2021*. Tech. rep. Brookhaven National Lab.(BNL), Upton, NY (United States); Thomas Jefferson ..., 2021.

Analog Computation for Near-Sensor Processing

Sahil Shah, Timothy K. Horiuchi, Pamela Abshire
Department of Electrical and Computer Engineering
University of Maryland, College Park, MD 20742
Email: sshah389@umd.edu, timmer@umd.edu, pabshire@umd.edu

Topics: Analog Algorithms and Programming, Error Correction Methods for Analog Computation, Hybrid Analog and Digital Systems, In-memory Analog Computing, Analog Brain-inspired Computing, Distributed Analog Computation

*We argue that: i) analog computing is well-suited to energy- and resource-constrained applications in **near-sensor processing**, and ii) the current **challenges** in analog computing implementations are **adaptation** techniques for large, hierarchical analog computing architectures and judicious **mapping** of computation and optimization problems onto analog and hybrid computing hardware architectures.*

Unlike digital computation, analog computation is intimately tied to its physical implementation. As a consequence, physical noise (i.e., Nyquist-Johnson noise, 1/f noise, etc.) and fixed-pattern noise (i.e., manufacturing variability) in the computing devices impose hard limits on the achievable dynamic range and precision. This leads to the conventional understanding that analog computing offers greater efficiency at low resolution, while digital computing incurs a higher initial penalty related to the bit error probability of quantization. Thus analog computing offers greater power efficiency at low signal-to-noise ratio (SNR), while digital computing offers greater efficiency at high SNR [1]– as illustrated in Fig. 1.

Scientific computing has many distinct end-goals: modeling (i.e., exploring the theoretical behavior of a model), using computation to conduct scientific experiments (e.g., control systems), or using computations in the course of collecting data (e.g., robot navigation). At first glance, analog computing appears to be ill-suited to support scientific inquiry, particularly when mathematical simulation requires *high precision* to avoid issues of error build-up over time. Nearly all sensor measurements unavoidably begin with an analog signal. Real-world signals have notoriously high *dynamic range* and care must be taken in choosing signal representations and data acquisition hardware to properly capture the phenomena under study.

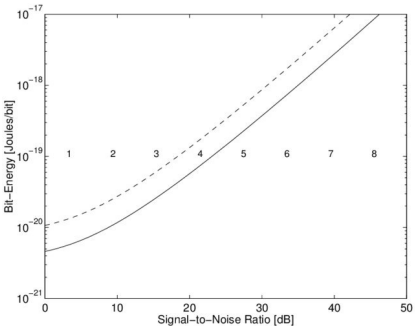


Fig. 1: Energy per bit vs SNR for analog (solid) & digital (123) signals [1]

There are many practical applications where a real-world signal has a large dynamic range, while the application-relevant aspect of the signal is limited to a modest dynamic range amenable to analog representations and computing. For example, an EMG signal often includes very large low-frequency components, but the tiny muscle activation signal is higher in frequency. While expensive high-bit-count analog-to-digital converters (ADCs) and digital filters can be used, a simple *analog* filter could be used to first isolate the smaller, relevant signal in conjunction with modest-bit-count ADCs. With advanced analog filtering techniques and integrated sensor signal processing, signals and signal features can be isolated prior to use, eliminating large uninformative signal excursions and enabling lower-precision operation. When the relevant signal varies over a modest dynamic range, adaptation and calibration techniques may be implemented in analog computing to map the relevant signals into the accessible dynamic range of analog signals. In many systems the choice of modest vs high dynamic range has the main consequence of requiring more power for higher DR signal representations, implying resource tradeoffs; in specific systems, proper operation requires the dynamic range of a sensor or physical process to be centered around a desired operating point. This can be done using adaptive analog techniques; examples include dark current compensation for ultra low light optical sensors [2] and offset cancellation to eliminate circuit bias in true random number generation [3]. Similarly, if the computation allows, analog *signal compression* circuits can be used with high-dynamic range signals to preserve information, as is often seen in biological signal processing. Signal compression is often an important computational step, providing the basis for comparisons between signals (e.g., the difference between log-compressed channels is related to their ratio).

Currently, on-chip analog storage technologies such as Floating-Gate Transistors, ReRAM [4], and Ferroelectric Field Effect Transistors (FeFETs) are not widely used commercially. However, they hold potential for various applications. These technologies can be employed to calibrate mismatched devices [2], [3], [5]–[7] or as analog memory for computational tasks, like neural networks [4], histogram equalization [8] and motion feature estimation [9]. Utilizing these storage methods can significantly enhance precision by reducing fixed-pattern noise. Although analog memory systems can be highly compact and integrated closely with the circuits they support [8], the circuitry required for modifying the memory might demand considerable space. Nonetheless, advancements in non-volatile technologies like FeFETsand ReRAMs, which are compatible with standard CMOS processes, now enable programming within the limits of standard power supplies and reduce the size of the peripherals.

There are many practical situations where scientific inquiry requires the *large-scale collection* of sensor data where devices must be inexpensive and low-power to provide long-term observation with low maintenance issues. In cases where the pure sensor signal is not used, rather a sensor product is locally computed, it may be the case that only low-precision computations are needed. Examples include: traffic speed monitoring, human face detection and counting, audio voice prompts, etc. In this context, analog computation (i.e., analog circuits) can play a valuable role in implementing low-power, adaptive, sensor signal processing.

Finally, there are several types of parallel and recurrent computations such as: *optimization/relaxation* algorithms and convolution, that are very computationally-intensive and can be slow and/or energy consuming in CPU-based implementations, but IF an analog implementation (usually a massively parallel analog system; e.g., resistive networks) can be formulated to perform this computation, it can be faster and lower-power, albeit, a dedicated computational engine with limited reconfigurability. Examples include: a silicon retina chip [10], [11], a visual horizon detection chip [12], a gradient visual motion chip [13], tomography, Ising machines, and reservoir computing.

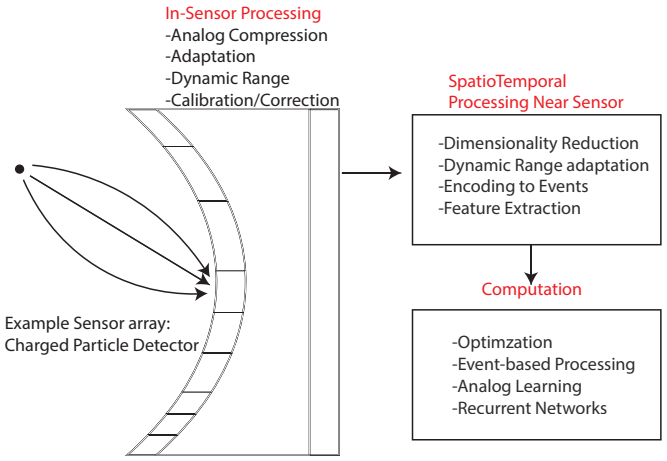


Fig. 2: Sensor array used for detecting charged particle (e.g Large Hadron Collider) generate significant amount of data per second. Analog near sensor processing at various stages can efficiently enable processing the sensor data.

Analog computing is a critical enabler for emerging hybrid computing approaches. Spike-based, biologically-inspired, neuromorphic systems occupy a very interesting position in this space, utilizing both *analog and digital* representations to capture some of the benefits of each approach, most notably the ability to leverage well-established digital hardware and interfaces in many instances. Relatively less-well explored, efficient spike (or "event-based") sensor and signal representations can also lead to dramatically different algorithms that are beginning to show promising results and more insight into biological computations where these spike-based representations are ubiquitous [14].

Opportunity: Achieving the potential of analog computing demands a new generation of researchers to develop simulation/verification tools and design techniques, utilizing commercial and emerging technologies to devise new analog computing architectures and solutions.

Timeliness: Although significant progress and promise was demonstrated in the past, funding for analog computing research has been severely restricted in recent decades. New technologies and applications have emerged (e.g., new nonvolatile memories, memristors, internet of things, artificial intelligence at the edge), and there is a huge opportunity to realize extreme power and size scaling by leveraging analog computing in these systems.

REFERENCES

[1] P. Furth and A. Andreou, "Bit-energy comparison of discrete and continuous signal representations at the circuit level." in *Proceedings of the Fourth Workshop on Physics and Computation*, 1996.

[2] D. Sander and P. Abshire, "Mismatch reduction for dark current suppression," in *SENSORS, 2010 IEEE*, 2010, pp. 1696–1700.

[3] P. Xu, Y. Wong, T. Horiuchi, and P. Abshire, "Compact floating-gate true random number generator," *Electronics Letters*, vol. 42, pp. 1346–1347(1), November 2006.

[4] C. Brando, M. Park, S. N. Chowdhury, M. Chen, K. Lee, and S. Shah, "Modeling and Analysis of Analog Non-Volatile Devices for Compute-In-Memory Applications," Apr. 2023, arXiv:2305.00618 [cs].

[5] Y. Wong, M. Cohen, and P. Abshire, "Differential hot electron injection in an adaptive floating gate comparator," *Analog Integr Circ Sig Process*, vol. 47, pp. 169–181, 2006.

[6] Y. L. Wong, M. H. Cohen, and P. A. Abshire, "A 1.2-GHz comparator with adaptable offset in 0.35-μm CMOS," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 55, no. 9, pp. 2584–2594, 2008.

[7] C. Sonnadara and S. Shah, "On-Chip Adaptation for Reducing Mismatch in Analog Non-Volatile Device Based Neural Networks," in *2024 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2024, pp. 1–5, iSSN: 2158-1525.

[8] Y. L. Wong, M. H. Cohen, and P. A. Abshire, "A 750-MHz 6-b adaptive floating-gate quantizer in 0.35-μm CMOS," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 56, no. 7, pp. 1301–1312, 2009.

[9] P. Xu, J. Humbert, and P. Abshire, "Analog VLSI implementation of wide-field integration methods," *J Intell Robot Syst*, vol. 64, pp. 465–487, 2011.

[10] M. A. Mahowald, "Silicon retina with adaptive photoreceptors," in *Visual Information Processing: From Neurons to Chips*. SPIE, Jul 1991, p. 52–58.

[11] K. A. Boahen and A. Andreou, "A contrast sensitive silicon retina with reciprocal synapses," in *NIPS*, vol. 4, 1991.

[12] T. K. Horiuchi, "A low-power visual-horizon estimation chip," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 56, no. 8, pp. 1566–1575, 2009.

[13] J. Tanner and C. Mead, "An integrated analog optical motion sensor," in *VLSI signal processing, II*. IEEE Press, 1986, p. 59–76.

[14] T. K. Horiuchi, "A spike-latency model for sonar-based navigation in obstacle fields," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 56, no. 11, pp. 2393–2401, 2009.

Next-generation Probabilistic Computing Hardware with 3D MOSAICs, Illusion Scale-up, and Co-design

Tathagata Srimani[1,†], Robert Radway[2], Masoud Mohseni[3], Kerem Çamsarı[4], Subhasish Mitra[5]
[1]Carnegie Mellon University, [2]University of Pennsylvania, [3]Hewlett Packard Labs, [4]UC Santa Barbara,
[5]Stanford University, e-mail: tsrimani@andrew.cmu.edu

Topic (Probabilistic Computing): The vast majority of 21st century AI workloads are based on gradient-based deterministic algorithms such as backpropagation. One of the key reasons for the dominance of deterministic ML algorithms is the emergence of powerful hardware accelerators (GPU and TPU) that have enabled the wide-scale adoption and implementation of these algorithms. Meanwhile, discrete and probabilistic Monte Carlo algorithms have long been recognized as one of the most successful algorithms in all of computing with a wide range of applications. Specifically, Markov Chain Monte Carlo (MCMC) algorithm families have emerged as the most widely used and effective method for discrete combinatorial optimization and probabilistic sampling problems. We adopt a hardware-centric perspective on probabilistic computing, outlining the challenges and potential future directions to advance this field. We identify two critical research areas: 3D integration using MOSAICs (Monolithic/Stacked/Assembled ICs) and the concept of *Illusion*, a hardware-agnostic distributed computing framework designed to scale probabilistic accelerators.

Challenges: Despite their significance in ML and AI, MCMC algorithms have yet to be accelerated with domain-specific hardware and are still primarily run on conventional CPUs or GPUs, severely limiting their widespread adoption. A fundamental challenge in accelerating MCMC algorithms is their inherently *serial* nature which obstructs parallelism. Another practical challenge is the need to generate a massive amount of uncorrelated random numbers (RNG), requiring trillions (10^{12}) of them within a few seconds. Even in simplified models like 2D checkerboards implemented on GPUs, sampling throughputs have saturated at around 10 billion (10^{10}) RNGs per second per chip, consuming 10 to 100W of power. Exacerbating this problem, today's silicon systems face fundamental limitations: the energy and speed benefits of smaller feature sizes have dramatically slowed over the past decade (*miniaturization wall*), and computing performance and energy efficiency are now dominated by data-movement (*energy/latency*) overheads rather than actual computation (*memory wall*).

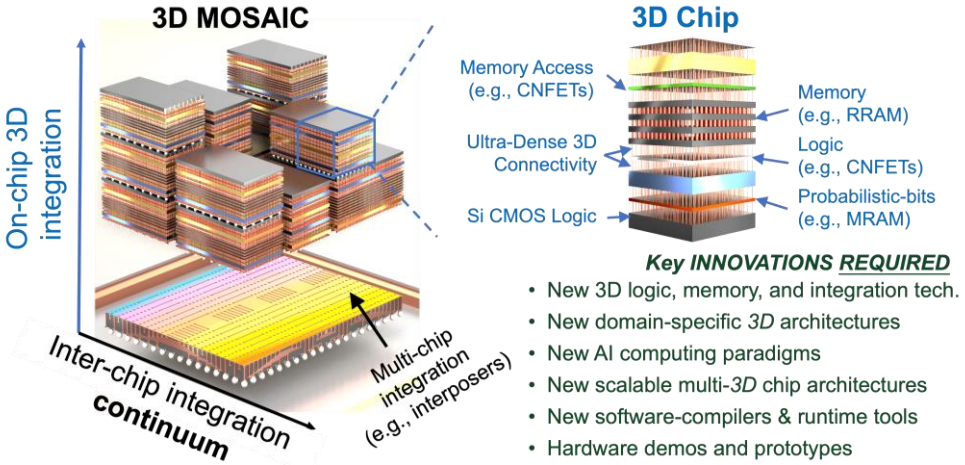


Fig. 1: Domain-Specific 3D MOSAICs (Monolithic/Stacked/Assembled ICs) targeting 100-1,000× system-level energy-delay benefits in scientific computing algorithms

Opportunities: To overcome these challenges and advance AI-hardware scaling for scientific computing, we need a synergistic combination of new device and integration technologies, circuits, domain-specific

architectures, AI-optimized algorithms, compilers, and runtime software support. To advance probabilistic computing hardware, we see two complementary opportunities: one with 3D integration of CMOS + X systems [1], and the other with *Illusion* where a large problem graph can be broken into pieces with minimal communication between interconnected chips [2].

3D MOSAICS: We assert that ultra-dense 3D integration of multiple layers of compute and memory access logic will be essential for achieving 100-1,000× improvements in system-level Energy-Delay-Product for scientific computing algorithms (e.g., the powerful Markov Chain Monte Carlo algorithms for combinatorial optimization and sampling) through technology-architecture-software co-exploration on 3D MOSAICs (Monolithic/Stacked/Assembled ICs, Fig. 1). Each 3D chip leverages ultra-dense integration of logic and memory layers to vastly reduce the memory-to-compute data movement overheads, resulting in substantial system-level energy and throughput benefits. Multiple 3D chips can be integrated using a combination of chip stacking, interposer, and wafer-level assembly techniques. In 3D MOSAICs, benefits can be expected at various levels of the stack:

1. **Device and Integration Level**: New materials enhance functionality (e.g., carbon nanotube FETs, RRAM/MRAM integrated on top of silicon CMOS).
2. **Circuit and Architecture Level**: 3D integration can enable new communication capabilities across chips, leading to large-scale probabilistic computation with millions of probabilistic bits.
3. **System and Software Level**: New software compilation and runtime technologies can complement domain-specific multi-chip architectures and enable efficient software execution on 3D MOSAICs.

Illusion* for probabilistic computing*: Beyond 3D MOSAICs, another effective method in scaling up computation is Illusion. The concept of Illusion was initially demonstrated in DNN inference to address limited on-chip memory, which necessitated frequent and costly off-chip memory accesses. By networking multiple chips, each with a minimal amount of local memory and rapid wakeup/shutdown techniques, Illusion achieved energy and execution times close to an ideal single-chip solution without off-chip memory.

We believe that this concept can be adapted to probabilistic computing [3] in a *hardware-agnostic* manner. This would involve breaking a large problem graph into smaller pieces through graph partitioning – potentially using weighted min-cut algorithms – and distributing the graph across interconnected chips. Thanks to the forgiving nature of probabilistic algorithms in sampling and optimization, the sampling throughput and the accuracy of an *ideal (*hypothetical) probabilistic computer that can house the entire graph can closely be approximated in synchronous and asynchronous architectures. Importantly, the concept of Illusion is *agnostic* to the choice of a probabilistic accelerator whether it is based on MRAM-based probabilistic bits, coupled oscillators, or other Ising machines.

Outlook: The main challenges in probabilistic computing with domain-specific hardware are related to scaling up the nodes, increasing the number of parameters and their interaction (2nd order, higher order, etc.) in a graph, and maintaining a very large throughput. We believe that MOSAIC3D and Illusion approaches are two complementary methods that enable the large-scale deployment of probabilistic computers, with a co-design approach across the stack.

References:

[1] Srimani, T., A. Bechdolt, S. Choi, C. Gilardi, A. Kasperovich, S. Li, Q. Lin et al. "N3XT 3D Technology Foundations and Their Lab-to-Fab: Omni 3D Logic, Logic+ Memory Ultra-Dense 3D, 3D Thermal Scaffolding." In *2023 International Electron Devices Meeting (IEDM)*, pp. 1-4. IEEE, 2023.

[2] Radway, R. M., K. Sethi, W-C. Chen, Jimin Kwon, S. Liu, T. F. Wu, E. Beigne, M. M. Shulaker, H-SP Wong, and S. Mitra. "The future of hardware technologies for computing: N3XT 3D MOSAIC, illusion scaleup, co-design." In *2021 IEEE International Electron Devices Meeting (IEDM)*, pp. 25-4. IEEE, 2021.

[3] Camsari, Kerem Y. "Probabilistic Computing with p-Bits: Optimization, Machine Learning and Quantum Simulation." In *2024 IEEE International Magnetic Conference-Short papers (INTERMAG Short papers)*, pp. 1-2. IEEE, 2024.

Device challenges to practical analog computing systems

A. A. Talin[1], S. Agarwal[1], S. Kumar[1], E. J. Fuller[1], S. Oh[1], W. Wahby[2], P. Xiao[2], C. Bennett[2], R. Jacobs-Gedrim[2], M. J. Marinella[3], Y. Li[4]

1-Sandia National Laboratories, Livermore, CA, USA
2-Sandia National Laboratories, Albuquerque, NM, USA
Arizona State University, Tempe, AZ, USA
University of Michigan, Ann Arbor, MI, USA

Topic: Improving analog devices

Background In the last 60 years, electronics have relied on deterministic, binary thermodynamic equilibrium states accessed by electrons to store and process information. The end of physical scaling of this approach motivates the need for post-CMOS or post-digital analog approaches, which offer improved functional density and energy efficiency instead of merely physical scalability[1]. Compared to digital approaches, analog encoding and processing of information are fundamentally different in two ways. First, instead of relying on binary states separated by at least 10kT of energy to store and process information, analog electronics employ quasi-continuous states defined by non-equilibrium thermodynamics and kinetics and separated by much smaller energy barriers[2]. Second, instead of using only electron motion to encode information, analog electronics can use electrical, thermal and electrochemical gradients in various heterogeneously integrated materials to move electrons, ions, and domains[3]. Understanding the scientific basis of these complex, frequently coupled mechanisms is difficult, resulting in few reliable physics-based models that can be used by circuit and chip designers. Furthermore, while these aspects of emerging analog devices may be advantageous in some ways, they also present increased sensitivity to variability, noise, and poorly controlled kinetic processes[4]. Current device and circuit design models, which guide manufacturing of digital CMOS in terms of dimensions, interfaces and dopants, are overly simplistic and provide limited guidance for exploiting the features or mitigating the issues of analog electronics. As such, despite decades of research and promising laboratory-scale performance, scientific knowledge gaps in the features of analog electronics identified above have led to their consistent failures to meet the stringent requirements needed for their commercialization[1].

The Challenge. We posit that the failing of analog electronics to challenge digital alternatives even in applications that do not require high bit precision is due to the lack of device concepts with well-defined fabrication processes, predictable and controllable characteristics including noise and drift, and scalable compact models for the design of arbitrary circuits and networks. Additionally, we believe that relying only on isolated device data to characterize emerging analog memory can be highly misleading.

The Opportunity: There are two key opportunities in both improving the analog devices and in improving the way we characterize and understand them.

1) Most current analog devices encode information in 2 dimensional interfaces of 1-dimensional filaments. Filamentary devices such as ReRAM are inherently susceptible to the placement of a single ion and are therefore limited in the accuracy that they can achieve. Similarly, two dimensional devices such as SONOS are more stable, but still have a limited number of electrons storing the state when scaled.

We propose that this challenge can be addressed by designing analog devices, such as synaptic transistors and ECRAM that encode information in three dimensional volumes,[5] rather than 2-dimensions channels or 1-dimensional filaments. Devices should also combine thermodynamic and kinetic mechanisms to stabilize a high density of analog states and rely on switching mechanisms that avoid chaotic and unpredictable behaviors during programming. Using ions rather than electrons to store a state also

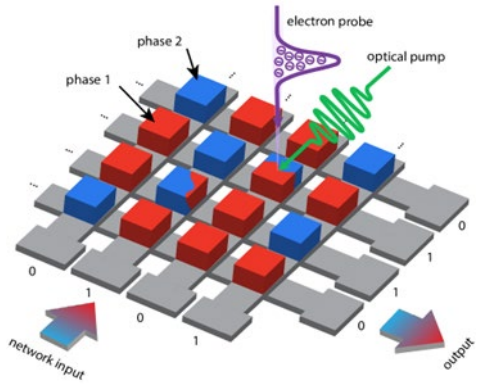


Fig 1: Materials need to be probed in-operando while being operated in small networks

results in far better retention, lower state drift over time, and potentially improved resilience in harsh conditions such as high radiation and temperature.

2) To develop a better fundamental understanding of how new devices work and assess their potential for technological impact, we need to go beyond probing single isolated devices. Rather, testing should include small networks to reveal how various device 'imperfections' such as drift, noise, and variability affect the power, latency and accuracy of well-established computing tasks. Additionally, even small networks can be used to investigate device-to-device coupling via electrical, thermal or other mechanisms, and to explore novel computing primitives based on ensemble effects. Furthermore, the electrical tests should be combined with other characterization methods that can inform on the fundamental physical and chemical mechanisms that underpin the computational primitives as illustrated in Fig 1. This approach, in conjunction with physics and machine learning, can be used to develop compact models and software tools similar to those widely used in Si CMOS, such as process design kit (PDK) and technology computer aided design (TCAD). Additionally, the large amount of high quality data correlating electrical test data with physical/chemical characteristics can substantially accelerate discovery of new materials and device concepts for analog computing.

Timeliness and maturity: A new class of volumetric devices called electrochemical ram (ECRAM) has recently been developed. These devices operate by storing the state using ions in the volume of the channel.

Acknowledgement: Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

REFERENCES:
[1] *SRC Decadal Plan for Semiconductors* (SRC 2021), https://www.src.org/about/decadal-plan/.
[2] S. Kumar et al., Nature Review Materials, 7, 575 (2022)
[3] K. Berggren et al., Nanotechnology 32, 012002 (2020)
[4] N. Semenova et al., Chaos 29, 103128 (2019).
[5] A. Talin et al., Adv. Mat. 2204771 (2022).

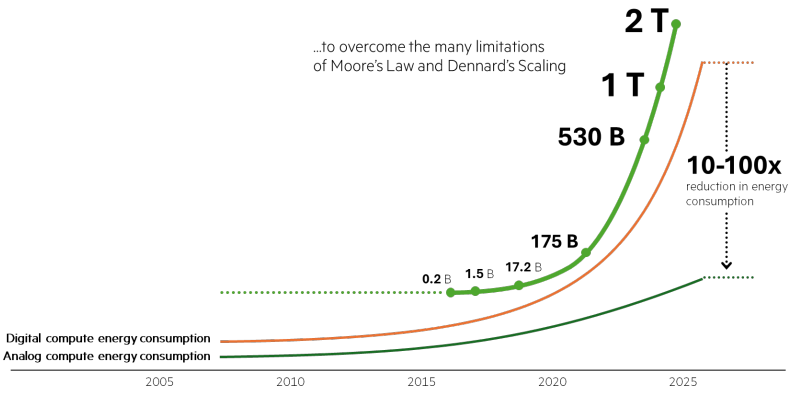
Heterogeneous Computing with Analog Accelerators for Future AI Supercomputers

Bassem Tossoun*, Giacomo Pedretti*, Paolo Faraboschi, Cullen Bash, Masoud Mohseni, Kirk Bresniker, Dejan Milojicic. Jim Ignowski, Ray Beausoleil, Andrew Wheeler
Hewlett Packard Labs, Hewlett Packard Enterprise, 820 N. McCarthy Blvd., Milpitas, 95831 CA, USA
bassem.tossoun@hpe.com, giacomo.pedretti@hpe.com

Abstract: Today, we are faced with a unique era in which Big Data, Artificial Intelligence (AI), and High-Performance Computing (HPC) provide a once in a generation opportunity to profoundly accelerate the way we advance science. Furthermore, in order to scale HPC systems to scale to keep up with the rising demands on computational power and efficiency, heterogeneous computing architectures including a variety of specialized analog accelerators based on emerging technology platforms such as CMOS, ReRAM, and photonics are essential. Each of these unique analog accelerators can be used to process specific AI workloads and resources can be efficiently used under a stack that supports this diversification across hardware and software.

Challenge: On Monday, May 21, 2023, at the ISC High Performance Computing Conference in Hamburg, Germany, HPE announced Aurora genAI, a series of science-focused generative AI models that will operate on the HPE Aurora supercomputer, capable of computing 2+ exaflops [1]. Aurora genAI is aimed at developing a suite of generative AI models for the scientific research community. These models, with up to 1 trillion parameters, 5.7 times more power than GPT-4, will be used in various scientific disciplines, ranging from cancer and disease research to the design of molecules and materials. The models will be trained on a vast array of data, including general text, code, scientific texts, and structured scientific data across fields such as chemistry, biology, material science, physics, and medicine. Generative models tailored for science and trained on extensive scientific data have the potential to accelerate discoveries in healthcare, material science, and many other research areas. This project could significantly impact the scientific community, providing insights that could directly benefit humanity.

The HPE Aurora features 21,248 CPUs, 63,744 GPUs, and 1,024 DAOS nodes, all connected via HPE Slingshot high-performance Ethernet interconnects. Testing has also shown near-linear scaling up to hundreds of nodes, though scalability testing for Aurora, which contains 10,624 nodes, is still ongoing. As these models scale in the number of parameters, the energy and financial costs increase at an alarming rate. For example, OpenAI's GPT-3 GenAI model contained about 340 million parameters and required 5 exaflops of computation for an entire day, and GPT-4 increased to about 1.8 trillion parameters only five years later. Moreover, OpenAI trained GPT-4 for nearly $100 million and today it costs about $500 million per year to run inferences on an LLM at 100 billion inferences per day [2]. In 5 years, it will cost about $160 billion to train an LLM, which equates to the amount of energy the United States consumes in one entire year.



Current digital computing systems were not primarily designed to run massively parallel algorithms such as deep learning neural networks. In between each multiply-accumulate operation, data must be transferred between memory and the processor. Moreover, it is well known that data movement is the primary source of energy consumption within digital HPC systems executing AI workloads.

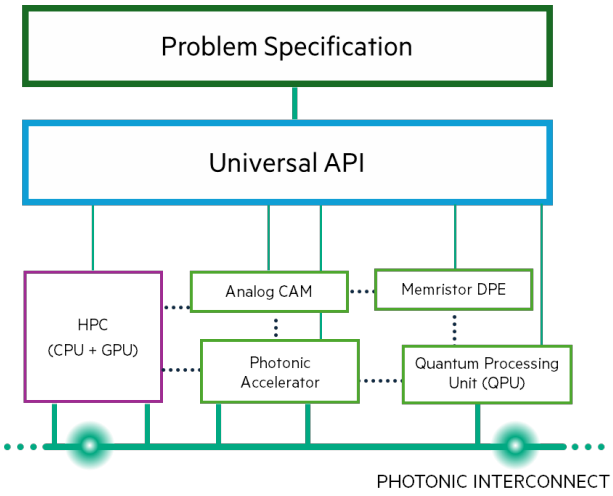
Opportunity: With that said, at Hewlett Packard Labs, researchers have been developing variations of analog computing technologies, which aim to address the underlying challenges of executing AI workloads with 10-100x lower energy consumption. One key advancement we intend to make is to establish a versatile and powerful heterogeneous computing system that includes neuromorphic photonic integrated circuits, state-of-the-art ReRAM crossbar-based dot product engines (DPEs), and content-addressable memories (CAM) to produce versatile and high-throughput AI HPC [3, 4, 5, 6]. For example, image processing, LLMs, GenAI, autonomous driving, natural language processing, and NP-hard optimization problems may all require different accelerators that are specialized in processing for each particular workload.

Within integrated photonic neural networks, arbitrary matrix multiplication can be performed without consuming any power. It should be noted that it is necessary to consume power at the inputs and outputs of the network to encode the input vectors (this includes light generation) and decode the output vectors. However, multiplying an N x N matrix by an N x 1

vector requires only O{N} encode/decode operations, while requiring O{N^2} scalar multiplications and additions. Digital electronics require the dissipation of energy to perform scalar multiplications and additions. From the dramatically lower scaling of energy-consuming operations required in optics, it should be clear that there is some value α for which when $N > \alpha$, optics will always be more efficient. Furthermore, with silicon photonics, on-chip training is enabled given the availability of reconfigurable photonic integrated circuits through low-power, high-speed optical modulators.

However, some data (such as parameters of a pre-trained model) are static and change infrequently throughout the model's life. In the case of deep neural networks, these parameters are inferred several times during inference. A similar workload is executed during the solution of optimization problems through stochastic local search algorithms, such as simulated annealing on unconstrained binary optimization problems. In these cases, a non-volatile memory able to store efficiently the model and perform inference without any data movement is preferable. In-memory computing with ReRAM has been proven a strong candidate for accelerating static dot products with a fraction of the energy consumption required by GPUs.

Some of the challenges associated with building such a system are building the appropriate compilers to work with data living in different domains between each accelerator. Software practitioners, such as AI/ML engineers/researchers, don't want to change the way they operate high-level tools (e.g. PyTorch/Tensorflow). However, the operation of photonic and in-memory computing hardware can be abstracted out as dynamic or static matrix operations, respectively. Intermediate representations can be used as a starting point for developing compilers for emerging hardware, which can leverage pre-existing and open-source instruction sets (such as RISC-V) to reduce the effort of building the entire toolchain.



Another key challenge is a path to scale and volume for both technologies. HPE Labs' Dot-Product Engine (DPE) technology and the Programmable Ultra-efficient Memristor-based Accelerator (PUMA) deep neural network (DNN) inference architecture are highly energy-efficient approaches to DNN inference, but the on-die weight storage approach, which is central to these architectures, limits the DNN size if the hardware is restricted to just one silicon device. Heterogeneous integration of chiplets enables multi-chip architectures that scale the size of the DNN being implemented.

Furthermore, silicon photonic integrated circuits can be purposed as an interposer and an optical reconfigurable network-on-chip which routes and transfers data from one ASIC to another for high computing throughput and versatility in model architecture. This can also provide an interface with which to convert data from the electrical to the optical domain and keep as much of the data movement in the optical domain since silicon photonics can provide a high-bandwidth, energy-efficient fabric for interconnects. One practical use case of this is to use the HPE Slingshot interface to interconnect different accelerators within an HPC system.

At the current state, there are basic foundry offerings of both silicon photonics and ReRAM-based memory. However, a PDK and specific process for neuromorphic computing devices and circuits need to be developed. We need to accelerate the path from research proof-of-concept prototypes to manufacturing foundries that can build these analog accelerators at higher volume and scale.

References

[1] https://aragonresearch.com/aurora-genai-to-enter-generative-ai-race/
[2] https://fortune.com/2024/04/04/ai-training-costs-how-much-is-too-much-openai-gpt-anthropic-microsoft/
[3] Liang, Di, et al., "An energy-efficient and bandwidth-scalable DWDM heterogeneous silicon photonics integration platform." IEEE Journal of Selected Topics in Quantum Electronics 28.6 (2022): 1-19.
[4] Li C., et al., "CMOS-integrated nanoscale memristive crossbars for CNN and optimization acceleration," 2020 IEEE International Memory Workshop (IMW), Dresden, Germany, 2020, pp. 1-4, doi: 10.1109/IMW48823.2020.9108112.[7] Li, C., Graves, C.E., Sheng, X. et al. Analog content-addressable memories with memristors. Nat Commun 11, 1638 (2020).
[5] Graves, C. E., Li, C., Sheng, X., Miller, D., Ignowski, J., Kiyama, L., Strachan 2003437, J. P., In-Memory Computing with Memristor Content Addressable Memories for Pattern Matching. Adv. Mater. 2020, 32, 2003437. https://doi.org/10.1002/adma.202003437
[6] Pedretti, G., Graves, C.E., Serebryakov, S. et al. Tree-based machine learning performed in-memory with memristive analog CAM. Nat Commun 12, 5806 (2021). https://doi.org/10.1038/s41467-021-25873-0

Native-Domain Analog Computing for Combinatorial Optimization

E. Valiante[1], I. Rozada[1], M. Noori[1], X. Zhang[1], C. W. Yang[1], F. Böhm[2], M. Mohseni[2], G. Pedretti[2], T. Van Vaerenbergh[2], R. Beausoleil[2]

[1] 1QB Information Technologies (1QBit), P.O. Box 16012, 1221 Lynn Valley Road, North Vancouver, BC, V7J 1A1, Canada
[2] Hewlett Packard Labs, Hewlett Packard Enterprise, 820 N. McCarthy Blvd., Milpitas, 95831 CA, USA

elisabetta.valiante@1qbit.com, ignacio.rozada@1qbit.com

Topic: *analog algorithms and programming, probabilistic computing, analog optimization, discrete optimization*

Challenge

Analog technologies can be used to build fast and energy-efficient accelerators. Many applications have been explored in the field of machine learning [1], but this technology can provide advantages in optimization and sampling for a diverse range of areas, including planning, decision making, operations research, and computational science [2]. Ising machines have been developed as an example of analog optimization solvers, but their restriction to quadratic unconstrained binary optimization (QUBO) models is a strong limitation on their performance. The most-challenging, and practically useful, optimization problems are typically characterized by dense connectivity, non-quadratic interactions, intricate constraints, and strong heterogeneity; converting these features into a QUBO formulation can result in large overheads, often making the problem even more difficult to solve.

Problems that are particularly adversely affected by the conversion to the QUBO formulation are NP-hard combinatorial problems that can be defined as permutation problems, such as the vehicle routing problem (VRP) and the quadratic assignment problem (QAP). The QUBO formulation for these problems implies introducing many infeasible configurations in the search space, as shown in Fig. 1. Other classes of problems adversely affected by the QUBO formulation are mixed-integer and constraint-satisfaction problems, both of which may include continuous variables and constraints.

Analog computation has the potential to solve optimization problems in their native formulation. This is already possible in the case of Boolean satisfiability (SAT) and satisfiability modulo theory (SMT) problems, for which current analog hardware can natively implement high-order clauses and continuous variables, without applying discretization and reductions required by Ising solvers [3, 4]. Solving permutation problems with a similar accelerator would require the addition of digital computing devices, which poses strong limitations on the time and energy-efficiency advantages that might be reached by the analog hardware alone.

Opportunity

Our experience shows that the research and testing of algorithms suitable for analog hardware cannot proceed efficiently without access to the most-advanced accelerators. At the same time, the development of new analog hardware greatly benefits algorithms and benchmarking technique expertise to guide the research and evaluate the performance in solving practical industrial problems. Algorithm and hardware development might lead to optimal but incompatible solutions, if done separately. The synergy of expertise in both domains can instead generate the ideal combined product.

Optimization is therefore also an ideal use-case to drive the evolution of heterogeneous computing systems, as their scale and complexity require multiple analog accelerators that are tightly integrated with digital processing systems. To fully leverage the advantage of analog accelerators, a thorough system-level design of this heterogeneous computing architecture will be key to avoiding system-level bottlenecks. While there is a variety of design tools available on an individual accelerator level, there exists a unique opportunity to create new system-level design tools that will enable the efficient design and realistic evaluation of novel accelerators within a larger HPC infrastructure. This will be key to taking advantage of emerging HPC technologies, such as Compute Express Link (CXL), that could facilitate high-speed interconnects between accelerators [5].

Moreover, translating and adapting different optimization algorithms to analog hardware remains a challenge. We identify a need to standardize and optimize this workflow, such that using the hardware becomes possible without in-depth knowledge of its operating principles. By leveraging emerging compiler frameworks for heterogenous computing systems [6], there is an opportunity to considerably reduce development overhead, while also enhancing ease of use, both of which are critical to ensuring the adoption of analog computing hardware.

The advancement of integrated heterogeneous infrastructures and compiler frameworks will allow the analog hardware implementation of the most-advanced solvers, such as conflict-driven clause learning (CDCL) algorithms, branch-and-bound (BB) algorithms, and SMT algorithms. In general, optimization solutions are adapted to the discretization established by digital hardware. As with analog computation, this imposition will not be in place anymore—new formulations and algorithms might be efficiently implemented, changing the current paradigm for many optimization problems.

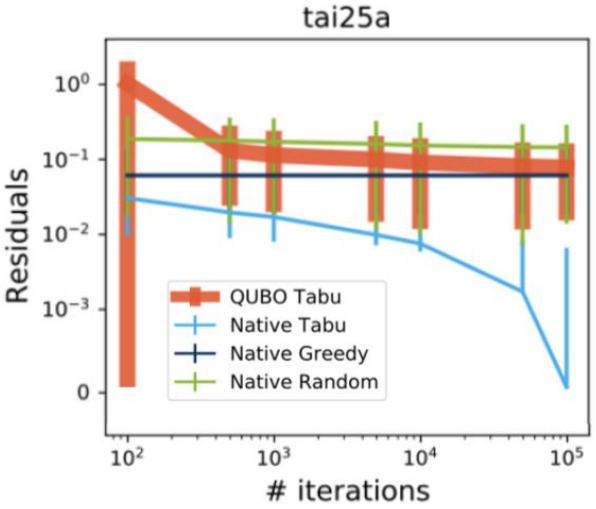
Timeline

There is currently an ever-increasing demand for optimization solvers: in addition to traditional domains such as logistics, supply chain management, and resource allocation, optimization is becoming increasingly popular in finance, telecommunications, healthcare, energy, and environmental management. More recently, ongoing work uses solvers (such as SAT solvers) alongside large language models (LLM) for better reasoning [7]. Quantum technologies are still too early in their development to provide an actual impact on industrial problems, and even the most-advanced quantum solutions are not yet sufficiently mature.

Optimization problem formulations and solving algorithms were developed in a specific way to match the advantages and limitations of digital hardware. Today, with the development of new analog technologies, we are in a unique position to shape formulations, algorithms, and hardware simultaneously, to find solutions that might not be considered optimal in each of the three fields if considered individually, but will provide the best performance when combined, improving the solving time and the energy efficiency by orders of magnitude.

References

[1] Bavikadi at al., 2020, https://doi.org/10.1145/3386263.3407649
[2] Mannocci et al., 2023, https://doi.org/10.1063/5.0136403
[3] Pedretti et al., 2023, https://doi.org/10.1109/IEDM45741.2023.10413853
[4] Bhattacharya, et al., 2024, https://doi.org/10.48550/arXiv.2401.16204
[5] Van Doren, 2019, https://doi.org/10.1109/HOTI.2019.00017
[6] Lattner et al., 2021, https://doi.org/10.1109/CGO51591.2021.9370308
[7] Ye at al., 2023, https://arxiv.org/pdf/2305.09656



QAPLIB Instance Name	Number of Facilities and Locations	Number of Binary Variables	Solution Space: Native	Solution Space: QUBO/Ising
chr12a	12	144	~10^{8}	~10^{43}
tai25a	25	625	~10^{25}	~10^{188}
esc32a	32	1024	~10^{35}	~10^{308}
tho40	40	1600	~10^{47}	~10^{481}
wil50	50	2500	~10^{64}	~10^{753}

Figure 1: (left) Example of residuals as a function of the number of iterations for the QAPLIB instance tai25a with 25 facilities and locations, corresponding to 625 binary variables in QUBO formulation; four heuristics were tested: a tabu implementation on the QUBO formulation, a tabu implementation and a greedy search on the native formulation, and a baseline of generated random feasible solutions. (right) Number of variables and size of the solution space in the native and QUBO formulations for several QAPLIB instances (https://qaplib.mgi.polymtl.ca).

Probability as a Signal: "Bayesian Circuits" as a Computing Foundation

Authors: Chris Winstead, Lukas Buecherl, Zhen Zhang
4120 Old Main Hill, Utah State University, Logan, UT 84322-4120, *chris.winstead@usu.edu*
Topic: Mathematical foundations for analog computation

Challenge: The digital world's reliance on binary signalling has delivered immense gains for technology and society. The power of digital systems is owed to the simplicity and mathematical power of the binary information model. Many alternative computing systems were conceived over the decades – analog, probabilistic, multiple-valued, biochemical, quantum annealing, and others – but none has yet emerged as a viable competitor to modern digital systems. Analog technologies tend to lack a unifying information paradigm; there is no analog equivalent to the powerful high/low binary abstraction. Although there are analog systems that achieve excellent characteristics in niche applications, there has not been a clear winning theory that cuts across those niches.

In most modern computing systems, the essential function is to sense a collection of states and signals, and to actuate one or more events in response. A computer is fundamentally a device for making inferences and decisions. The binary abstraction is quite intuitive for making logical inferences, derived from classical logic theory. In the 21st century, the Bayesian paradigm rose to importance in cognitive science and inference theory, providing well-defined methods to utilize the continuous space of information that falls between zero and one. Many analog technologies have been explored in connection with Bayesian methods. We argue that Bayesian circuits can serve as a unifying paradigm for accelerating progress in analog computation.

In the Bayesian realm, logic gates become probability gates. Given some physical signals A and B, a simple probability gate implements a conditional probability transformation $\Pr(C \mid A, B)$. A key feature of probability signals at gates is that they include conventional logic under the special case where all probabilities are at the logic "rails". This detail make it easy to interface between Bayesian circuits and conventional digital systems, since a probability can be simplified at any point to a binary decision.

Opportunity: Bayesian approaches (and related probability-themed methods) have been around for decades, and are well positioned for a renewal. As with traditional digital circuits, Bayesian approaches can be implemented via innumerable circuit and signal combinations. What's important is the information paradigm, since it facilitates a unified design approach for complex and scalable systems in many application areas. This creates myriad opportunities for collaboration and cross-pollination between specialties.

As a key example of cross-cutting application, the Muller C-Element[4] is a classic logic gate originally developed for asynchronous logic circuits. The C-Element can also perform Bayesian inference[2] and has been adapted for stochastic error correction.[5] Its functions are now known in many technologies ranging from memristor circuits[1] to genetic circuits in synthetic biology.[3] To illustrate the cross-cutting application of Bayesian approaches, the C-Element can be used for "probability restoration" via positive-feedback of probability. This concept can be used to mask transient upsets in triple-modular-redundancy systems (it works for binary, multi-valued logic, and quantized analog signals[6]), and to realize population "quorum" sensor in synthetic biology.

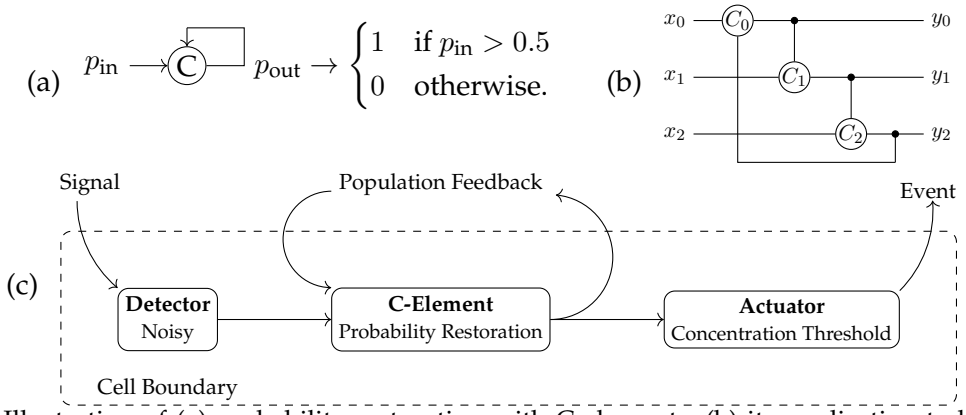


Figure: Illustration of (a) probability restoration with C-elements, (b) its application to binary and non-binary TMR circuits, and (c) its application to a quorum sensor circuit in synthetic biology.

Timeliness: With a renewed national interest in domestic research on electronics and semiconductors, there is an opportunity to build capacity for research on analog information processing, with Bayesian concepts as a unifying framework. Bayesian concepts have been a steady research topic within many disciplines, including analog computation, but a large share of this research is presently conducted outside the United States.

Beyond computation, there is an overlapping national interest in bio-security and biological engineering. The function of biological systems is influenced by chemical noise. While digital abstraction of genetic systems is possible, chemical signals are not naturally amenable to the binary model. Therefore, viewing them as analog systems and using Bayesian approaches promise advancements in the biological design space, as illustrated by the genetic C-element example.

Finally, with the now-ubiquitous label of "artificial intelligence," there is an urgent need to make progress on understanding the fundamentals of machine inference. Contemporary neural-like systems have "black box" nature, making it difficult (or impossible) to guarantee or even analyze their trustability in the wild. With Bayesian networks, the knowledge model is exposed – at least in principle – in the structure of the network or circuit. This opens the possibility of building AI systems that can explain their decisions, a feature that will prove especially valuable when they inevitably make consequential bad decisions.

References

[1] D. Bonnet et al. "Bringing uncertainty quantification to the extreme-edge with memristor-based Bayesian neural networks". In: *Nature Communications* (Nov. 2023).

[2] J. S. Friedman et al. "Bayesian Inference With Muller C-Elements". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* (2016).

[3] N. Nguyen et al. "Design and analysis of a robust genetic Muller C-element". In: *Journal of Theoretical Biology* (2010).

[4] David Muller. "A theory of asynchronous circuits". In: *Proc. Int. Symp. on Theory of Switching, 1959*. Vol. 29. 1959, pp. 204–243.

[5] V. Gaudet and W. Gross, eds. *Stochastic Computing: Techniques and Applications*. Springer, 2019.

[6] Chris Winstead et al. "An Error Correction Method for Binary and Multiple-Valued Logic". In: *Proc. IEEE Intl. Symp. Multiple-Valued Logic*. 2011.

Physical Foundation Models: How to build practical 10^{18}-parameter artificial-intelligence processors

Logan G. Wright (Yale U., logan.wright@yale.edu), Peter L. McMahon (Cornell U., pmcmahon@cornell.edu)

Topic: ultra-large-scale analog-neural-network processors

Summary: ***We present our position that there is a radically different approach to building analog neural-network processors that could enable inference with 10^{18}-parameter models.*** The key assumptions are: (1) building inference processors where the vast majority of model parameters are fixed at the time of fabrication now makes sense in light of the discovery of the phenomenon of *foundation models* [1]; (2) eliminating the conventional hierarchy of abstractions in the design of analog processors, and instead designing the functionality of processors directly at the level of the physical substrate can yield powerful neural networks [2,3]. AI processors that make and exploit both assumptions can enjoy orders-of-magnitude improvements in model-parameter storage density and energy efficiency in inference.

Challenge

Introduction: Making AI systems really big

AI systems will get bigger. Today's models routinely reach 10^{12} parameters, a number that would have seemed farfetched if not comical decades ago. Tomorrow's models will, provided there are economically feasible paths, likely reach scales that we would find comical today. With that in mind, we ask here: How can we possibly construct AI systems that have 10^{18} or even 10^{21} parameters?

The approximate default answer to this question is that AI systems will just need to be digital supercomputers available via the cloud, and applications of AI will always be limited by latency and uptime, which will improve over time (but subject to the limit of speed-of-light delays). This is plausible, and for many applications of AI it may be adequate. But is it the only way? And it is the best way?

Why really big AI systems pose an unprecedented challenge for computer hardware

We think there is an alternative, analog route worth exploring, but to see why consider first the challenges of scale. A 10^{18}-parameter network is large - a *million times* larger than today's neural networks (~10^{12} parameters), which are in turn at least a million times bigger than late-1990s neural networks.

A key observation is that 10^{18} parameters require a lot of memory: 1 exabyte (assuming 8-bit parameters). There is no small system that can store this amount of data. At 1 Tbit/in^2 (roughly the current state-of-the-art) this would require 5000 square meters. We could fold this into 3D, but even if we assume N layers separated by only 5 μm, the volume of such a hard drive would be substantial: V = (N x 5 μm) x (5000 m^2)/N = a cube 30x30x30 cm^3. This is far from cellphone-sized, and is large enough to be impractical for most robots. While improved storage density could help, transferring such a volume of data will still be incredibly time (and energy) consuming. With a 1 Tbit/s bus, it would take 90 days to read in the full exabyte-scale model. While improvements will occur, a reasonable conclusion is that neither traditional inference nor training would be practical on such a device.

There is, however, an alternative. Capitalizing on the emergence of foundation models that exhibit zero-shot transfer learning, we note that *programmability is not strictly necessary for AI inference*. Rather than attempting to construct programmable hardware, what if we designed *single-purpose processors* to execute foundation-model inference at the absolute maximum efficiency, using analog physical systems? For the purposes of this short paper, we'll call these single-purpose analog computers *physical foundation models.*

Opportunity

Physical foundation models

A physical foundation model (PFM) is a physical device, such as an electronic or photonic material or circuit, that is made to perform a single computation (the inference of a very large, pretrained AI model) using the absolute lowest level of hardware physics possible (for example, the nanoscale flow of electrons in nanostructured semiconductors).

What could a physical foundation model look like?

One (hypothetical) example of a physical foundation model is a three-dimensional, nanoelectronic material in which the material properties (e.g., conductivity) in each voxel of the material are the adjustable parameters. In such a material, the 3-dimensional flow of electrons from one side to the other could enact

highly complex transformations of input currents. If properly designed, these analog transformations could realize inference of a corresponding AI model. Such a device's parts do not have simple, 1:1 analogies with the mathematical operations of today's Transformers. Rather, the device would need to be designed using detailed physical simulations that accurately predict the nanoscale electronic flow. Using such a simulation, the material properties of each voxel could be *learned* like neural-network weights [3], iterating until the simulated evolution of electrons through the device matches the desired inference. Alternatively, a kind of *compiler* could efficiently translate pretrained neural-networks specifications into nanostructured material designs that approximate them. Either process would be difficult (at the very least, computationally intensive). Architectural or manufacturing challenges may also require imposing additional structures to ensure the device can be accurately fabricated and/or can approximate the necessary mathematical operations. These are all, to say the least, substantial challenges. But they are also not clearly impossible, and intriguingly so given the potential outcome.

Why PFMs might be a promising route to large-scale AI systems:

- PFMs would dramatically reduce the energy cost and size of executing large-scale AI inference. While the details can be debated, removing the necessity of programmability and compiling computations to the most low-level physical realization will vastly reduce the requirements of hardware and eliminate virtually all inefficiencies from abstraction. Such physical, analog systems may also circumvent fundamental limitations of digital CMOS hardware, like Landauer's limit.
- Single-purpose analog computers are a more centrally controllable way to deploy large-scale AI systems into edge devices. In contrast to Hinton's mortal computation, each PFM can be the same, and can thus be designed to obey certain requirements (e.g., for safety purposes).
- In contrast to cloud-based implementations, per-inference costs could be vastly reduced (no need for communication with a server or to use programmable hardware). Inferences could also be private.
- By enabling foundation-model inference to be performed locally and with low energy cost, PFMs could allow completely new approaches to computer hardware and software.

The challenge of PFMs

As our example highlights, the central challenge to realizing PFMs is one of large-scale physical inverse design: *How can we design (or "compile") large-scale AI models, with many parameters, directly to the lowest level of a scalable hardware's underlying physics (e.g., nanoelectronic flow)?*

This has within it several sub-challenges, e.g.:

- What physical platform is scalable from the point of view of scalable neural-network architectures? This has many sub-issues, including expressive power, designability, and manufacturability.
- How can we physically model hardware accurately enough so that fabricated designs work as designed? And/or how can we compile models that are resilient to fabrication and modelling errors?
- How should we design these systems to interface with traditional digital components efficiently?

The ability to create large-scale, energy-efficient PFMs would have wide-ranging implications for the design of computing systems more generally. The ability to design and manufacture PFMs could transform how we design other physical devices too, opening the possibility for AI-powered physical inverse design of electronic nanosystems for applications far beyond traditional computing, such as microsensors or nanorobots.

Timeliness

Why now? (1) There is enormous and growing demand for efficient AI inference on ever-larger models; (2) Whereas before the advent of foundation models in the past ~4 years [1], it was nonsensical to think of a neural-network processor whose model parameters couldn't be changed, now it is reasonable to consider processors where the vast majority of parameters are fixed at fabrication time. (3) It has become apparent that deep neural-networks don't necessarily have to follow the conventional structure developed in the artificial-neural-network community, and that co-designing the physical substrate and the network may lead to expressive, accurate neural networks that are also extremely well-matched to hardware [2,3].

References

[1] R. Bommasani et al. arXiv:2108.07258 (2021)
[2] J. Laydevant*, L.G. Wright* et al. *Neuron* **112**, 2, 180-183 (2024) doi:10.1016/j.neuron.2023.11.004
[3] L.G. Wright*, T. Onodera* et al. *Nature* **601**, 549-555 (2022) doi:10.1038/s41586-021-04223-6

Part 3: Pre-Workshop Report

1 Introduction

Analog computing, once overshadowed by the digital revolution, is experiencing a resurgence driven by the growing demand for energy-efficient and specialized computational paradigms. As we approach the physical limits of traditional digital computing, researchers are revisiting and reimagining analog approaches to tackle complex computational challenges. This renewed interest spans a wide range of disciplines from exploring the mathematical foundations of analog computation to new ideas in devices and physics.

Analog computation fundamentally differs from digital computation in its representation of data. In analog systems, physical quantities directly correspond to the values they represent. For instance, a tide's height might be represented by an axle's rotation angle or a specific voltage level, creating a direct "analogy" between the physical variable and the represented value. Conversely, digital systems encode values into discrete binary states, requiring implicit computation to establish the connection between the physical representation and the actual value (e.g., interpreting sequences of bits as numbers through positional notation). In modern analog computation, these variables can be represented in a wide variety of ways, including but not limited to voltages, probabilities, chemical concentrations, and light intensities. While digital computing has dominated due to its precision, scalability, and noise resistance, analog computation is experiencing a renaissance. This resurgence is driven by several factors, including potential advantages in energy efficiency, processing speed, and applicability to specific domains such as synthetic biology. However, analog systems still face significant challenges, particularly in terms of accuracy and scalability when compared to their digital counterparts. The ongoing research in analog computing seeks to leverage its unique strengths while addressing these limitations, potentially complementing digital systems in specialized applications.

We highlight several key areas of analog computing below. Physics-based approaches, such as Ising machines, leverage fundamental physical phenomena to solve complex optimization problems. Analog electrical systems revisit and enhance classical analog computing techniques, offering energy-efficient solutions for certain computational tasks. Computational memory paradigms blur the line between processing and storage, potentially revolutionizing data-intensive applications. Hybrid systems combine the strengths of both analog and digital computing, promising to overcome limitations of each approach when used in isolation. Additionally, we explore emerging fields such as chemical and biological computation, photonics-based analog systems, and probabilistic computing.

These diverse approaches to analog computing share common challenges, including issues of scalability, precision, and integration with existing digital infrastructure. By addressing these challenges and leveraging the unique advantages of each approach, analog computing has the potential to complement and enhance our current computational capabilities.

2 Notional Questions

To define the vision for the future of analog computing for science, we have identified several notional questions that will shape the discussions at the workshop. Accepted position papers try to provide answers to some of these questions.

- For which applications does analog computation demonstrate superiority, and for what metrics?
- How does analog computation's energy efficiency compare to digital computation in various applications?
- What new materials, devices, systems, design software, etc, are needed to enable the future analog computation applications?
- What can be learned from biology's reliance on a mixture of analog and digital computation and applied to science and engineering problems? Examples of biological computing hardware include regulatory reaction networks and neural tissue.
- Is there a cross-cutting mathematical framework for analog computation?
- How do we scale analog computing systems to solve large-scale problems?
- What are the limits of analog computing and how do we approach this limit using practical devices, circuits, and systems?
- How do we program analog computing systems? Do we need new programming models and compilers? What

does the software stack look like?

- What benchmarks and standards are necessary to evaluate and compare the performance of analog computing systems? How can we establish a common framework for assessing the capabilities and limitations of different analog computing approaches?
- How can hybrid systems that combine analog and digital computing be designed to exploit the strengths of both approaches? What are the challenges in developing efficient interfaces between analog and digital components? What opportunities or challenges does analog computing offer for integration with sensing devices?
- How do we design extremely heterogeneous systems for large-scale and edge systems?
- How do we address noise, variability, and robustness issues in analog computation? How do we leverage these non-idealities?
- What role can interdisciplinary collaboration play in advancing analog computation? How can fields as diverse as physics, materials science, biology, neuroscience, and computer science contribute to the development of analog computing?
- What programs and curricula must be developed to train the scientific workforce in analog computation?

3 Mathematical Foundations

Introduction

The systematic study of the mathematical foundations of analog computation dates back to at least to mechanical differential analyzers when Shannon famously showed that a small set of basic analog components (adders, multipliers, integrators, constants)—equivalent to polynomial ordinary differential equations (ODEs)—can generate a well-defined, very broad class of functions[1]. There are systematic methods to implement ODEs in mechanics and electronics[2], and more recently for chemical reaction networks (CRNs)[3]. ODEs, and their DAE (Differential-Algebraic Equation) generalizations, are naturally useful for simulations of existing dynamical systems (the emphasis of differential analyzers was on physics simulations; circuit simulators, exemplified by SPICE, operate on large/complex DAEs that model circuits). However, ODEs/DAEs can also execute algorithms more commonly associated with digital electronics, by embedding algorithms in a continuous-time/value framework. Indeed, it has been proven that in principle, any discrete computation can be encoded into polynomial ODEs (i.e., that polynomial ODEs are Turing universal), albeit at the cost of significant system complexity[4]. Real-valued Turing computation, typical of physical computation and ODEs, may have advantages[5]. Recent breakthrough results showed that efficient (polynomial time) discrete computation can be implemented with short "curve length" analog computation[4]. Thus polynomial ODEs appear to be a particularly promising mathematical model for defining and studying broad classes of analog computation. Another example is embedding the (discrete/"digital") Ising problem into a continuous framework, leading to the currently very active field of Ising machines, which solve discrete optimization problems using continuous ODE/DAE embeddings. The mathematical model for oscillator Ising machines (the Kuramoto or generalized Kuramoto models) is not polynomial but involves periodic functions.[6]

Open issues and key research needs

We need the same depth of understanding of the mathematical foundations of analog computation as for Boolean circuits. As one example, Post's lattice[7] exactly enumerates the closed classes of Boolean functions, while no similar comprehensive understanding exists for classes relevant for analog computation, although initial efforts in analog abstraction provide a starting point[8]. Another example concerns the operation of practical Ising machine schemes, whose global minimization properties are not well understood. While Lyapunov functions serve to explain local minimization properties, their ability to move towards global minima, *e.g.*, using parameter cycling, has so far resisted attempts at rigorous analysis. Another direction is to understand the precise nature of the connection between analog operation regimes, which seem crucial for eventually finding global minima, with discrete operation regimes which correspond to the discrete problems being optimized.

Polynomial ODEs are likewise hard to analyze and program. Thus we need to develop models and programming languages at higher levels of abstraction to capture analog computation as few models exist (e.g.[9]). One

promising approach involves designing an energy function and then systematically creating polynomial ODEs to minimize it. For example, there is a line of work showing that computationally complex problems (Boolean satisfiability) can be solved in this manner[10]. However, it remains unclear whether the techniques employed in that work for ensuring that the system does not become trapped in local minima could be generalized to other problems.

Other important questions includes issues of complexity. How do we measure the time-complexity of analog computation for analog algorithms? Are there other relevant notions of complexity such as robustness complexity?

4 Analog Integrated Circuits

Introduction

Analog circuits exist in all consumer electronic devices, and yet, the design and test methodology has changed little over several decades. Most individuals assume Integrated Circuits (IC) are always digital devices, and that all of the real efforts are done by digital devices. Analog ways of doing things are products of a gone-by age, gone the way of an analog clock and related devices. And yet, the energy efficient and area efficient analog opportunities are not simply a small improvement on an existing digital landscape, but rather can be central to the next generations of advanced computing.

Current State of the Art

Analog Electronics are a significant industry for interfacing sensor signals and driving actuator devices throughout the electronics industry. The high percentage of revenue as well as high commercial margins (40-50% or higher) show a healthy analog industry with high-demand for its components. Every sensor requires some amplification, signal conditioning, and data conversion to a digital format, such as an Analog to Digital Converter (ADC). Every actuator has multiple driver circuits as well as data conversion from a digital format, such as a Digital to Analog Converter (DAC). Voltage regulation for analog and digital systems (e.g Low-DropOut regulators (LDO) as well as supplying many digital voltages are ubiquitous. Analog circuits can operate at a wide range of frequencies and ranges. The amazing capabilities analog circuit design is illustrated by the range of of the test and measurement equipment companies (e.g. Tektronics, Keithley) innovations and capabilities. When faced with device mismatch, a wide range of digital devices can be used to counter these issues, including measuring and selecting desired devices during experimental verification.

Analog Integrated Circuits for Scientific Instrumentation

Custom analog integrated circuits can meet stringent specifications required for many scientific experiments where precision or speed might be required while meeting strict power or noise budgets. Such circuits find use in applications ranging from laser expertiments to high-energy physics, where ultra-low-noise ($\leq$ 1000 e- RMS), low-power ($\leq$ 1mW/channel) front-end circuits must be developed. Such analog and mixed-signal ICs have been designed, customized for sensors in ATLAS at CERN and DUNE. Specialized readout ASICs have been developed for neutrino physics experiments such as nEXO, where extreme environments must be handled, such as operation below 100 mK while displaying radiation tolerance, and experiment-specific triggering capabilities. Ultra-high-speed applications such as those involving photon science applications, including LCLS and LCLS-II at SLAC, have developed novel analog circuits such as the ePix Camera System and SLAC Analog Memory. These systems have resulted in high-speed ASICs capable of sub-20 ps timing resolution and GHz-range digital data transmission, addressing experiment needs of high-resolution (14–16b), low-jitter (sub-100 fs) signal acquisition.

Open issues

Analog electronics needs its VLSI revolution similar to digital electronics in its computing scope, in its wide applicability, and in its wide usability. In general, analog IC industry faces the fears of becoming a commodity business of typical component parts. The high margins coupled with a fairly conservative industry results in the commercial concern around flattening new markets, innovation, and revenue. Analog electronics needs to no longer be primarily the domain of analog "artists," while further amplifying up the innovations of those artists. The open issues to reach these opportunities include:

Programmable & Mismatch Insensitive Design: The primary problem in analog IC design is mismatch,

and the great killer in analog IC design is threshold voltage mismatch. Programming of some form is typically the accepted way of handling these concepts; programming techniques need to be generalized for the next generations of analog IC capabilities.

More Analog Designers: A major issue is having enough engineering talent for the entire analog circuit ecosystem that includes system architects, analog IC designers, verification, analog test, and analog system development.

Configurability: Digital design can develop large systems as well as take advantage of economies of scale by having concepts, like microprocessors, FPGAs, and GPUs, that can be reconfigured for a large number of tasks. The corresponding analog research systems do exist (e.g large-scale Field Programmable Analog Arrays (FPAAs), e.g.[11]),), although these concepts need to be widely available and utilized.

Synthesis Tools: Many digital computing systems (microprocessors, FPGAs, and GPUs) require and enable synthesis tools both for developing these large systems as well as to use these systems (e.g.FPGAs). The corresponding analog tools do exist in an early form(e.g.[12]), although these concepts need to be widely available and utilized.

Key research needs

To create the opportunities of an analog VLSI revolution, that includes analog system design at a large scale as well as analog computing concepts (e.g. Sec. 5), we need major efforts along the following directions

CMOS Programmable, Mismatch Insensitive, Analog Design: Although the great killer in analog IC design is threshold voltage mismatch, it can be directly accounted for, as well as programmed away, using FG devices, even over a wide range in temperature[13]. FG programming out and effectively eliminate this mismatch greatly improves the SWaP of these devices, including decreasing load capacitances and resulting bias currents. These techniques become more important to scale analog design techniques to advanced IC nodes, including FinFET nodes. Although such techniques have been experimentally demonstrated[13], significant research opportunities as well as educational opportunities exist to enable these concepts. This approach can optimizeeverything possible in standard CMOS design, as well as pushing some IC fabrication DRC rules that might enable some more favorable all CMOS design. Most design requires testing techniques to measure and set parameters (typically in Non-volitile memory, typically a form of an embedded FG device to adjust for the device mismatch.

Develop the wider Analog ecosystem: More engineers are needed to fill the entire analog ecosystem, as most system development applications are constrained by the number of available analog designers. Most analog IC designers, a subset of the entire design community, mostly have expertise in component level design and maybe a larger component like a DAC or ADC. The number of analog architects is an even more critical need. Even fewer analog IC designers and architects are available to fuel the analog computing discussion (e.g. Sec. 5), although the system focus might encourage the next generation.

Configurable Analog ICs and Systems: FPAA devices could become ubiquitous for analog system development as well as enable experimentally building analog computing opportunities if they were widely available. Research is necessary to both enable the wide vision of FPAA devices[14], as well as research is necessary to enable a wide community to use these concepts for solving their applications in the same way an entire digital FPGA community exists targeting a range of application problems.

Synthesis Tools: Although analog synthesis and related design tools is rather rare, FPAA development gives a window into the opportunites around analog synthesis. The current development of analog synthesis tools (e.g.[12] coupled with recent innovations of programmable analog standard cells (e.g.[15;16]) paints a picture of analog IC design for large systems than can come up to the capabilities of its digital counterpart[9].

The critical use of analog ICs for scientific instrumentation: Although large-scale analog capabilities are important, the continued development of new analog IC development pushing the state of the art in scientific measurement and system instrumentation to match the ever expanding specifications required for analog instrumentation, particularly in filtering and data conversion.

5 Computational Memory

Introduction

Computational memory, also known originally as Compute in Memory (CiM, or alternately in-Memory computing) computing, related to processing-in-memory (PIM), creates energy efficient analog architectures for particular analog computations that nearly eliminates additional memory access and data movement. The original development of CiM[17;18] was developed with the early Floating-Gate (FG) computational crossbars. This architecture approach addresses the energy efficiency wall problem[19], both for analog and digital architectures, eliminating the data transfer cost and bottleneck between the processor and memory. This issue has been further exacerbated by data-intensive tasks such as machine learning and artificial intelligence that require frequent memory accesses. This has led to increasing interest in CiM and dataflow architectures. The neuromorphic field has long advocated for computational memory as a key feature seen in the brain.

Current state of the art digital approaches for CiM include SRAM-based IMC, FG (including Flash), high bandwidth memory (HBM), and dataflow architectures with distributed memory, while analog CiM can be constructed with floating-gate based technologies. There are also novel NanoDevice Technologies such as FeFETs, MTJs, Memristors etc.

Open Issues

Even though the CiM internal computation is energy and area efficient, poor peripheral architecture design can overwhelm the overall system cost. Using moderate- to high- precision edge D/A or A/D conversions are a very costly architecture choice. Architecture development around CiM designs must minimize data movement, including resisting the desire to needlessly shuttling huge amounts of high precision data at high energy cost. Analog architectures must consider the entire system path, including the high relative costs of moving data to memory. The lack of highly skilled analog designers with CiM expertise as well as having domain-specific knowledge slows the development of this technology. Availability of analog synthesis tools could reduce this design stress. Analog architecture and numerics do not have the same tradeoffs as digital architectures and numerics[20], requiring different tradeoffs as well as a lot more design experience for system implementation. Programmability is essential to eliminate analog variability, limiting the computation to thermal noise as well as enabling architectures where noise could be a useful feature.

Key Research Needs

An important future research need is analog co-design approaches tailored for hybrid and heterogeneous Computation-in-Memory (CiM) enabled architectures. This involves integrating analog architecture, numerics, and abstraction to create more efficient and powerful systems[20;21]. Equally important is the development of synthesis and co-design tools that can effectively capture domain knowledge, including better abstraction for analog CiM components. Currently, there is a significant lack of such tools, with only a few examples available[12]. Another critical need is the widespread availability of programmable and reconfigurable analog CiM integrated circuits[11]. These would provide the flexibility and adaptability required for various applications and research purposes. Finally, there is a pressing need for platforms that allow quick prototyping and testing of computational memory ideas. Examples such as the Field Programmable Analog Array (FPAA)[11;14] demonstrate the potential of such platforms in accelerating research and development in this field. Addressing these key research needs will be crucial in realizing the full potential of CiM.

6 Photonics

Introduction

Photonic analog computing has emerged as a promising field at the intersection of optics and computational science. By exploiting the unique properties of light, such as its ability to propagate and interfere in parallel, high speed, and low heat dissipation, photonic systems offer significant advantages for implementing ultra-fast, energy-efficient complex computational tasks[22].

Current State of the Art

Photonic analog computing has made significant strides in recent years, with several key developments:

Optical Neural Networks (ONNs): Implementation of matrix-vector multiplications using photonic integrated circuits[23], along with convolutional neural networks using diffractive optical elements, has been demonstrated[24]. Photonic tensor cores for accelerating deep learning operations have also been developed[25].

Optical Reservoir Computing: Time-delay reservoirs using optical feedback loops and spatiotemporal reservoirs using multimode fibers and spatial light modulators have been realized[26]. **Analog Optical Computing Primitives.** Optical Fourier transforms, frequency filtering, and implementation of differential equation solvers using photonic waveguides have been achieved. Optical implementation of integral transforms and convolution operations has also been demonstrated[27].

Programmable Photonic Processors: Reconfigurable photonic integrated circuits have been developed[28], along with the integration of phase change materials for non-volatile photonic memories[29].

Open Issues

Despite the progress, several challenges remain in realizing the full potential of photonic analog computing:

Scalability, Precision and Noise: There is significant difficulty in scaling up the number of optical components while maintaining coherence and stability. There are also challenges in integrating large-scale photonic systems with electronic interfaces[30]. Precision of optical computations remains limited due to shot noise and other sources of noise, and there is difficulty in achieving high dynamic range in analog optical systems[31].

Nonlinearity: There is limited availability of efficient, low-power optical nonlinear elements. There are also significant challenges in implementing complex activation functions in the optical domain[32].

Programmability and Reconfigurability: There is a significant need for faster and more efficient methods to reconfigure optical systems, and overcoming the limitations in the flexibility of current photonic architectures[28].

Energy Efficiency: We are hindered by the high power consumption of certain optical components (e.g., lasers, modulators), as well as the significant energy overhead in optical-electrical-optical conversions[33].

Hybrid Systems: While photonics provide opportunities for accelerating some heavily used kernels (e.g., matrix multiplication), integration with digital systems and control of the optical components still are challenging.

Key Research Needs

To address these challenges and advance the field of photonic analog computing, several key research directions should be pursued:

Advanced Materials and Devices: The development of novel nonlinear optical materials with improved efficiency and speed is crucial. Research should focus on low-loss, high-bandwidth photonic integrated circuit platforms. Additionally, the exploration of emerging material platforms, such as 2D materials and perovskites, for photonic computing should be prioritized[34;35].

Architectures and Algorithms: It is essential to design photonic computing architectures that are inherently robust to noise and variability. The development of algorithms that can leverage the unique capabilities of photonic systems is also necessary. Furthermore, investigation into hybrid photonic-electronic architectures to combine the strengths of both domains should be conducted[30].

Precision Enhancement Techniques: Research efforts should be directed towards error correction and noise mitigation strategies for analog optical systems. The development of techniques to increase the effective number of bits in photonic computations is also of great importance[31].

Programming and Control: The creation of high-level programming frameworks for photonic analog computers is a key priority. Additionally, the development of efficient methods for training and optimizing photonic neural networks should be pursued[36].

System Integration: Research into the seamless integration of photonic computing elements with electronic systems is crucial. Moreover, the development of standardized interfaces and protocols for photonic computing modules should be a focus area[28].

General Take-aways for Analog Computation

The pursuit of analog computation with optical systems highlights several crucial aspects relevant to the broader field of analog computation. First, the choice of physical platform should carefully consider the specific application and the required computational task[37]. Utilizing the inherent advantages of the chosen platform is crucial for achieving a computational advantage over traditional approaches[38]. Second, the development of task-specific algorithms and architectures that exploit the unique features of the physical system can significantly improve performance and efficiency. Lastly, understanding the limitations of the chosen platform, such as noise and scalability constraints, is crucial for developing robust and reliable analog computing systems.

7 Probabilistic Computing

Introduction

Stochasticity and uncertainty are ubiquitous in the world around us. However, stochasticity in devices is seldom exploited in computation. The random number generation (RNG) capabilities offered by today's computing platforms can be computationally inefficient, especially for larger workloads[39]. Indeed, conventional digital logic computing is deterministic with the goal of removing any variability or non-deterministic behavior and compute with high precision. However, not only is this is not an energy efficient approach, there are many applications where expressing uncertainty in the output can be beneficial to decision-making. As a contrasting example, the brain exploits stochasticity for highly energy efficient computation. Applications of probabilistic computing include modeling complex problems like nuclear and high-energy physics events, complex biological systems, precise climate models, large-scale neuromorphic applications, and artificial intelligence (AI) algorithms.

The current state of the art in random number generation primarily relies on pseudo-random number generators (pRNGs) that employ expensive rejection sampling techniques. These methods use deterministic digital hardware and then expend considerable energy to reintroduce randomness, which is inherently inefficient. In response to these limitations, there has been growing interest in true random number generation (TRNGs). Recent advancements include the development of CMOS-based TRNGs and emerging device TRNGs, such as probabilistic bits (p-bits)[40] and coinflips[39]. These novel approaches leverage the inherent physical properties of devices to generate random numbers more efficiently, potentially offering significant improvements in both computational efficiency and energy consumption for probabilistic computing applications.

Key Research Needs

Developing methods to generate billions of random numbers with a low energy budget is crucial for scaling probabilistic computing applications. To achieve this, there is a particular need to bridge the gap between low-level device engineering and high-level applications. The design of probabilistic circuits requires a multidisciplinary approach, integrating expertise from algorithms, architectures, and devices.

Novel, hardware-aware algorithms are needed for probabilistic computing, capable of generating specific distributions directly from a given set of devices. While the development of stochastic or programmable devices is important, it's not sufficient alone to bring about a paradigm shift. Integration and system-level considerations are equally crucial. Emerging device issues, including device-to-device and cycle-to-cycle variability, as well as endurance, need to be addressed for reliable probabilistic computing systems.

To advance probabilistic computing, several key strategies emerge. Stochasticity should be reframed as a feature rather than a bug, with the physics of devices leveraged for computation. The development of hardware-aware algorithms is crucial, as is the use of AI-guided methods to efficiently narrow the search space for optimal solutions. Possible approaches include those arising from *thermodynamic computing* which include novel algorithms that can leverage these sources of device-stochasticity to explore complex state-spaces and stabilize on the solution[41]. Co-design and co-optimization across the full stack will be essential to realizing the potential of probabilistic computing. Furthermore, programmability and reconfigurability must be incorporated as key features in RNG systems to cater to diverse applications.

8 Physics Inspired Computing and Spin Hamiltonians

Introduction

Physics-based/physics-inspired ($\pi-$) computing represents a paradigm shift in information processing. It leverages physical systems' inherent properties to perform computations, rather than using binary logic gates. This approach encodes problems into spin Hamiltonians (e.g., Ising, XY, Q-clock, k-local models) and uses their natural evolution to achieve efficiency, scalability, and energy consumption gains[22]. The core principle involves mapping complex optimization problems onto universal spin Hamiltonians. The function to be minimized is encoded into coupling strengths between 'spins', allowing efficient navigation of vast solution spaces. This versatile approach can be implemented on various platforms (photonic, electronic, atomic, spintronic), each offering unique advantages. For instance, photonic systems leverage light speed and wave properties, atomic systems exploit quantum coherence and entanglement, while coupled light-matter systems are driven through a symmetry-breaking transition on the changing loss landscape until a mode that minimises losses is selected. Electronic IC (integrated circuit) implementations offer the advantages of extremely small size, very low energy consumption, scalability to large number of spins, and mass producibility at low cost.

Two primary directions have emerged for developing physics-based hardware, each utilizing distinct aspects of physics' role in computational processes. The first approach exploits natural evolution principles of physical systems influenced and driven by external parameters, with the challenge of establishing controllable couplings between 'spins'. Polariton condensates in inorganic[42] and organic-inorganic halide perovskites[43], atoms in QEDs[44], and degenerate laser systems[45] exemplify this, leveraging the natural dynamics of the system for computation while extending the natural XY Hamiltonians to Ising, Clock and k-local spin Hamiltonians. Conversely, the second approach, represented by technologies like analogue interactive machines[46], and spatial photonic machines[47], focuses on establishing couplings through processes like light propagation, optical modulation, and signal detection, thereby managing system dynamics through feedback loops. Electronic implementations typically achieve coupling using programmable resistive connections or variable capacitors, large numbers of which can be integrated on chips with small area.

Current state of the art

The field of physics-based computing is experiencing a period of rapid development, with several promising hardware platforms demonstrating the feasibility of this paradigm. Gain-based computing, a notable example, uses driven-dissipative systems where the gain and loss rates encode the computational problem[48]. By driving these systems through a symmetry-breaking transition, the system naturally selects a mode that minimizes losses, effectively revealing the optimal solution. This approach has shown potential for solving combinatorial optimization problems by mapping them onto various spin Hamiltonians, and is applicable across a range of physical platforms. Another promising direction is the development of spatial photonic Ising machines (SPIMs)[47](SPIMs), which leverage the properties of light to perform computations at the speed of light. SPIMs have demonstrated the ability to efficiently solve Ising problems with specific interaction matrix structures, paving the way for potential applications in machine learning and optimization. Yet another promising direction is Oscillator Ising Machines (OIMs), a primarily electronic scheme based on injection locking that features high-quality optimization performance.[49]

As of 2024, we are on the cusp of expanding physics-based hardware platforms that bring to life the theoretical concepts of unconventional computing. These platforms now enable proof-of-principle experiments, shedding light on their actual performance and operational capabilities. We need to explore and establish the scalability of these systems, a critical factor in assessing their long-term viability and guiding future investments in this technology.

Open issues

Despite the significant progress, several fundamental questions remain regarding the practical applicability and scalability of physics-based computing. Addressing these challenges is crucial for realizing the full potential of this paradigm. Key open issues include:

Scalability: While current experimental platforms demonstrate proof-of-principle, scaling these systems to handle real-world problems remains a significant challenge. Research is needed to understand the limitations imposed by noise, decoherence, variability, propagation delays, and manufacturing constraints.

Problem Suitability: It is essential to identify the specific classes of problems best suited for physics-based computing. Understanding the relationship between problem structure, encoding onto spin Hamiltonians, and the dynamics of the physical platform is crucial for optimizing performance.

Control and Optimization: Developing robust methods for controlling and manipulating physical systems (for example, frequency and coupling weight calibration) to ensure reliable and accurate computation is a critical area of research. This includes understanding the role of quantum fluctuations, topological defects, and reservoir dynamics.

Error Mitigation: Developing strategies to mitigate errors arising from noise, imperfections in the physical system, and the inherent stochasticity of quantum mechanics is crucial for achieving reliable computational results.

Key research needs

Moving forward, research should focus on addressing the open issues outlined above. Specific research directions include:

Comparative Analysis: Conducting comparative studies across diverse experimental platforms is essential. This includes examining polaritons, lasers, cold atoms, electronic ICs, and superconducting circuits to identify universal principles and platform-specific advantages and limitations.

Theoretical Modeling: Developing robust theoretical models is crucial. These models should accurately capture the dynamics of the physical systems and provide insights into performance, limitations, and optimization strategies.

Algorithm Development: Designing algorithms specifically tailored for physics-based computing platforms is necessary. These algorithms should take into account the constraints and capabilities of the underlying physical systems.

Hybrid Architectures: Exploring the potential of hybrid architectures is an important direction. These architectures should combine physics-based computing with traditional digital computing to leverage the strengths of both paradigms.

Despite advancements in the physical realisation of these concepts, critical questions remain about scalability, the influence of phase space structures on system performance, and identifying problems best suited for these unconventional computing architectures. We need to understand the dynamical behaviour of the systems during symmetry-breaking transitions, trajectory optimisation towards global minima, error probabilities, and the potential for dissipation and nonlinearities to rectify these errors, highlighting the pivotal role of theory in addressing these challenges. By comparing various experimental platforms, including polaritons, lasers, cold atoms, and electronic Ising machines, we should emphasise and exploit the universal nature of these research questions, unlock the disruptive potential of gain-based photonic computing, parameter-cycling based electronic Ising schemes, *etc.*.

General takeaways for analog computation

Analog computation, facilitated by physics-based approaches, offers a compelling alternative to digital computing, particularly for specific problem classes. By harnessing the inherent properties of physical systems, analog computation can potentially achieve significant advantages in speed and energy efficiency. However, successful implementation requires careful consideration of inherent limitations, such as precision constraints and the potential for noise and errors. The development of robust control mechanisms, error mitigation strategies, and tailored algorithms is crucial for realizing the full potential of analog computation in the context of physics-based computing.

9 Chemical and Biological Computation

Introduction

State of the art work in synthetic biology can embed a dozen engineered Boolean logic gates in a cell[50], but the space of meaningful computations with so few Boolean gates is limited. On the other hand, a dozen analog components such as integrators, which can be constructed with similar component complexity, can realize much

more complex functionalities[2;51]. Thus analog computation is uniquely compatible with (bio)chemical computing hardware—where parts are "expensive" and many "digits of precision" are not required. We envision applications in environments where electronic microcontrollers cannot go: into a cell[52], or in vitro biochemical systems such as for controlling molecular self-assembly, or DNA data storage[53]. A particularly promising application is in the area of "smart drugs" that can detect specific disease state in vivo and autonomously generate a therapeutic response[54]. For example, such a smart drug could target cancer cells and release localized cell killing therapies specifically in cancer tissues by identifying cancer signatures. Importantly, analog computation has been used to differentially weigh markers according to their importance[55].

Unlike electronic computation, biological chemical computation is distinctly energy efficient. The current cost per logical operation of electronic computers is six orders of magnitude more than for cellular biochemical processes[56], processes which were optimized by evolution to conserve energy. Even if future ultra-low energy computers will not rely on chemical reactions as such, biology provides a concrete proof-of-principle hardware and inspiration.

Chemical and biological substrates also provide unique mechanism to implement decentralized optimization frameworks. Signal transduction in cellular signaling pathways, which consist of protein-based cell surface receptors get activated by various ligands, such as ions, small organic or inorganic molecules, and proteins, binding to the receptor surface in response to some external stimuli (neurotransmitters, hormones, cytokines, and so on), irrespective of the activities at the other receptor sites[57;58]. This leads to conformational changes in the protein (receptor) composition, thus releasing enzymes or secondary messengers into the cell or the substrate and initiating cellular activities which may lead to various microscopic and macroscopic changes in the organism. Examples of such cellular signaling pathways can be found in different biological phenomena, such as quorum sensing[59], gene expression, growth factor signaling, metabolism, and apoptosis. Using the distributed exchange of signals, the ensemble itself evolves to arrive at a system-level decision corresponding to the solution of a global optimization task (each agent being oblivious of the actions of its peers during the process), i.e., ranking the agents based on the amount of the quantity possessed[60].

Instead using chemicals and receptor elements to build analog computing engines in a bottom-up manner, one can use a top-down approach where an entire biological module is embedded a hybrid synthetic and biological computing system[61]. One such emerging biocomputing substrate is using organoids, which are miniature 3D models of developing brain tissue derived from stem cells[62]. These sub-systems can self-organize intricate neural circuits recapitulating aspects of the cell composition, connectivity, and functionality of the human brain[63].

Open issues and key research needs

How can we program desired biochemical analog computation? Some approaches involve converting from analog electronic circuits to networks of coupled chemical reaction[64], or compiling analog chemical computation from discrete algorithms[65]. Compilers were also developed for specific hardware such as DNA strand displacement reactions[66]. However, there is the obvious danger that we are trying to shoehorn ill-fitting paradigms, resulting in systems that are too complex and fragile. On the other hand, there is much work in the area of systems biology which tries to develop the necessary ideas to understand biological regulatory networks without relying on electronic paradigms, yet the understanding of complex information processing remains limited[67].

Thus there is a key research need to develop a chemistry-first approach to complex analog computation. Asking "what can chemistry naturally do" leads to the consideration of such paradigms as networks of weakly interacting RNA strands[68], or the surprisingly tight connection between Boltzmann machines and stochastic chemical reaction networks[69]. Similarly, computation by stoichiometry rather than reaction rates appears well-suited to certain kinds of analog computation[70].

Some other engineering challenges that are the key to scaling biological and chemical substrates as a practical analog computing engine are (a) lack of extensive input/output interfaces for reading the state of the network, (b) difficulty in controlling and modulating the dynamics, and (c) high batch-to-batch variability.

It is important to point out that several areas mentioned above provide opportunities in partnership between DOE and National Institutes of Health (NIH)—particularly when related to human health. As it pertains to elucidating the fundamental design principles of biological systems, the related work is also within the scope of the Biological Systems Science Division (BSSD) of DOE.

10 Co-Design

Introduction

To enable realization and effective use of analog computational models, co-design is needed across the whole stack, from applications to system software, from architectures to devices, and materials. Enabling design, integration, and providing the necessary tools and resource to make an analog computing effective require cross-cutting approaches[71].

Current State of the Art

Current solutions to address analog computing typically look only at one specific element of the stack[21]. For example, research has focused on computational principles with in-memory computing, optical devices, Ising machines, or even micro-fluidic machines. Most of the focus has been architectural, but there's significant interest in understanding device and material properties to understand miniaturization and scalability, reliability, level of control and impact of noise. Separately, research has started looking at how to actually map and compile application kernels on these computational models[12;72;73], and how to design fully hybrid systems. Taken separately, there have been demonstrations that analog computing devices, typically with specifically hand mapped algorithms (the mapping process itself is a research contribution), can provide significant energy efficiency benefits with respect to conventional digital computing devices[74;75]. Most of the research and development infrastructure is, however, composed of separate, ad hoc, tools with several glaring omissions.

Open Issues

There are several open issues to enable co-design across the stack for analog computing. **Materials and devices.** Significant work is ongoing to identify new materials and devices to reduce noise and enable accurate control of the physical phenomena and laws that perform the computation. **Integration in hybrid systems.** Analog computing devices are expected to work as accelerators for certain computational patterns, but will not be able to support entire general purpose computing. Analog computing solutions will need to be integrated with conventional computing elements, and both hardware and software interfaces are still unclear. **Programming.** The type of computations performed by analog devices poses the question on how they can be programmed to map applications kernels. Part of the research focuses in identifying new formulations for algorithms to enable hand mapping on the computational physical system, but there are no general abstractions. It is not clear if we need new languages, as well as how to represent, and eventually combine, the set of operations that an analog computing device can execute. Defining programming models for analog computing devices also requires considering assumptions on accuracy and range of the (potentially continuous) values that are respresentable. **Scaling and evaluation.** Evaluating the effective impact of analog computing on complex applications is extremely complicated for the lack of a comprehensive research infrastructure. How much these solutions can scale is still questionable, and there are few methods to evaluate their impact (simulation, emulation, and prototyping) on representative applications. Conversely, while some computational patterns that analog computing systems can perform well are known, there is no yet a systematic approach that enables quantifying the real impact on full applications, as well as application drivers to drive the design of hybrid digital-analog systems. **System software** There is very limited work on managing heterogeneous architectures that integrate analog accelerators with conventional digital heterogeneous computing platforms. As previously highlighted, this integration is not only hardware, but also require software integration. We need to define the system software (e.g., libraries, application programming interfaces, runtime interfaces) that could allow invoking analog accelerators from conventional general purpose processors, executing the computation, and collecting results.

Key Research Needs

To address these challenges and advance the field of photonic analog computing, several key research directions should be pursued.

Simulation infrastructures: Simulation and modeling is a key element in enabling studies (for mapping applications, integration in systems, understanding scalability, etc). However, integrating typical discrete event simulation used in computing systems with continuous time simulation (focused on analog signals) is challenging. Additionally, simulation infrastructures needs to be multi-scale and multi-fidelity, enabling scalability but also considering physical and device properties that could impact feasibility of the designs.

Design Tools: There is a need for automated electronic design automation (EDA) tools to enable construction and synthesis of the devices. The physical design and layout processes for analog circuits currently remain mainly hand-based with minimal or no automation, except for some recent research examples (e.g. [9;12]) Approaches that could also raise abstraction levels (e.g. [8]) and enable mapping or synthesis of analog operators are needed.

Programming models and system software: There is a need to understand how to systematically program analog computing devices - including opportunities for languages and/or compiler (e.g. [9;12]). System-software is also critical, both to enable execution of applications and integration of analog devices in heterogeneous computing systems.

Benchmarks and proxy applications: Co-design is typically application driven, typically requiring benchmarks for the system optimization. Analog computing system benchmarks are in its early stage (e.g. [9]), where further benchmarks are desired to avoid an ad-hoc analog design approach. The many open questions around neuromorphic principles (e.g., spiking neurons, dendrites, reservoir computing) arise from not understanding the important engineering computations arising from neurobiology.

General Take-aways for Analog Computation

We need to invest on tools and methodologies that could on one side enable design and evaluation of novel analog computing devices, and on the other provide the infrastructure needed to deploy analog computing devices in heterogeneous computing systems. We need flows that could target both analog CMOS and CMOS+X technologies. Simulation, emulation, and prototyping platforms are critical to enable the co-design process, as well as evaluate feasibility and actual impact of analog computing on realistic workloads. Approaches that enable physical and logical (software) integration of analog computing devices in computing systems are critical to enable their practical use.

11 Integration of Sensing and Compute

Introduction

Biological systems display a tight integration between sensing and computation. Indeed, predicting, and making sense of, sensory inputs forms a core tenet of multiple theories of cognition while also explaining experimental observations of neural response and activity. Additionally, embedding computational capabilities within sensors indicates that it could lead to a dramatic reduction in their energy-costs while simultaneously enabling downstream decision making [11;76].

In parallel, a convergence of trends has emerged in computing and sensing. This has been characterized by miniaturization, algorithmic advancements, and reduced manufacturing costs, enabling the large-scale integration of sensors in our environment. This realization of the internet-of-everything has driven progress across fields, including micro- and nano-robotics, industry 4.0 through ubiquitous sensing and computing, applications in the circular economy, smart infrastructure, and personalized healthcare. In scientific applications like climate monitoring, where satellite-based hyperspectral imaging systems must process vast amounts of data under severe power and bandwidth constraints, the energy cost of communication between signal acquisition and processing often renders such systems impractical.

However, realizing the potential of ubiquitous sensing and computing requires addressing challenges for analog computing at different energy dissipation and operational scales. The promise of integrated sensing and computation systems can only be fully realized by overcoming these hurdles, which span from device-level considerations to system-wide architectural challenges.

Current state of the art

Hybrid systems and in-sensor computing have shown promising prototype demonstrations, paving the way for future advancements. Some such examples include: self-powered sensors [77], high-dimensional acquisition [78] and analog processing systems with local pattern matching [79;80]; analog feature extraction systems to relax the requirements for analog-to-digital conversion, processing [81], or communication [82]; stochastic sensors and those with feature-extraction embedded into the transducers [83]; and neuromorphic and event-driven sensors which enforce temporal and spatial sparsity [84]. A key trade-off across these systems has remained balancing energy

efficiency with accuracy.

Open Issues

Despite these advances, several open problems remain unsolved. There is no clear theoretical work that underpins the partitioning of resources across the processing chain. Even though existing in-sensor computing prototypes have been shown to outperform traditional digital computing, the fundamental limitations of such systems are not known. Principled, end-to-end optimization and development of techniques for this are not currently understood. Lower in the abstraction, circuit primitives and architectures for efficient computing and sensing are still an active area of research. Architectures that use data-conversion circuits at the analog-digital interface result in limited system performance[20] (as in Section 5). Balancing energy efficiency with circuit robustness remains an open problem, which is further exacerbated in scaled nodes. In addition to CMOS-based circuits, new avenues for novel circuit topologies open up with emerging devices and materials. As design-automation support for such systems is still in its infancy (e.g.[12]), existing in-sensor computing platforms are hand crafted designs.

Key Research Needs

To address the challenges noted above, several advances are required:

End-to-End Design Optimization: It is necessary to optimize algorithms for area, energy-efficiency, robustness to hardware impairments, and latency, considering the specific characteristics of analog computation. Since active analog circuits introduce noise in the signal chain, there is a need for principled techniques to analyze algorithms using energy and noise budgets across different components of the signal acquisition and computing chain[85].

Algorithmic Advances: There is a need to develop systematic approaches for joint design of hardware and algorithms, carefully studying the trade-offs between front-end complexity and back-end computational requirements. Algorithms that can leverage analog principles and feedback on perceptual inputs must be developed. Techniques to better leverage the output sparsity of event-based sensors are needed to realize their practical use.

Novel Circuits and Systems: New designs are needed to help push the boundaries of energy-efficiency, bandwidth, and linearity. In addition to new primitives, techniques that can enhance robustness and mitigate the impact of process, voltage, and temperature (PVT) on analog circuits are critical, especially for scaled nodes.

CMOS+X for Sensing: Novel devices and materials can open up new modalities for sensing and computing or alternatively deliver unparalleled energy-efficiency and performance[86]. Demonstrating these principles at scale, with comprehensive measurements and simulation is critical to transitioning these technologies from lab to fab.

General Take-aways for Analog Computation

The next-generation of edge-computing systems must be capable of extracting key information from the sensory data-stream, processing it either entirely, or partially in analog. Successful demonstrations of such platforms will require novel algorithms resilient and robust to analog impairments, energy-efficient system design with co-designed hardware and algorithms, and the development of novel integrated sensing and computation circuits. These integrated systems face several challenges, including noise management, precision limitations, and variability in analog components. To overcome these hurdles, research should focus on developing robust analog computing primitives, noise-tolerant algorithms, and adaptive calibration techniques. Additionally, exploring novel materials and devices that offer unique computational properties could lead to breakthroughs in energy efficiency and performance.

12 Conclusions

Analog computing systems, and more broadly, physical computing systems, hold the potential to profoundly transform the future of computing[5]. These approaches promise to achieve the energy efficiency levels needed to overcome the limitations imposed by the end of Dennard scaling and Moore's Law. They offer both the specialization and efficiency required to address the diverse computing demands across various scales—from integration with scientific instruments, to edge computing, to centralized supercomputing systems—essential for the Department of Energy's scientific discovery workflows today and in the future.

However, numerous research challenges remain to be addressed before the effectiveness and applicability of

these new computing paradigms can be fully realized. These challenges span multiple domains, including materials and devices, architectural design, integration with conventional systems, system software and programming, and application development. This workshop aims to bring together experts from these diverse areas, alongside practitioners and stakeholders, to identify the foundational research needs that will make analog computing for science a reality.

References

[1] Claude E Shannon. Mathematical theory of the differential analyzer. *Journal of Mathematics and Physics*, 20(1-4):337–354, 1941.

[2] Bernd Ulmann. *Analog Computing*. De Gruyter Textbook. De Gruyter Oldenbourg, Munich, Germany, 2nd edition, 2022.

[3] David Soloveichik, Georg Seelig, and Erik Winfree. DNA as a universal substrate for chemical kinetics. *Proceedings of the National Academy of Sciences*, 107(12):5393–5398, 2010.

[4] Olivier Bournez, Daniel S Graça, and Amaury Pouly. Polynomial time corresponds to solutions of polynomial ordinary differential equations of polynomial length. *Journal of the ACM (JACM)*, 64(6):1–76, 2017.

[5] Jennifer Hasler and Eric Black. Physical computing: Unifying real number computation to enable energy efficient computing. *Journal of Low Power Electronics and Applications*, 11(2), 2021.

[6] Tianshi Wang and Jaijeet Roychowdhury. OIM: Oscillator-based Ising Machines for Solving Combinatorial Optimisation Problems. LNCS sublibrary: Theoretical computer science and general issues. Springer, June 2019.

[7] Emil L Post. Introduction to a general theory of elementary propositions. *American journal of mathematics*, 43(3):163–185, 1921.

[8] Jennifer Olson Hasler, Aishwarya Natarajan, and Sihwan Kim. Enabling energy-efficient physical computing through analog abstraction and ip reuse. *Journal of Low Power Electronics and Applications*, 2018.

[9] Jennifer Hasler and Cong Hao. Programmable analog system benchmarks leading to efficient analog computation synthesis. *ACM Trans. Reconfigurable Technol. Syst.*, 17(1), jan 2024.

[10] Mária Ercsey-Ravasz and Zoltán Toroczkai. Optimization hardness as transient chaos in an analog approach to constraint satisfaction. *Nature Physics*, 7(12):966–970, 2011.

[11] Suma George, Sihwan Kim, Sahil Shah, Jennifer Hasler, Michelle Collins, Farhan Adil, Richard Wunderlich, Stephen Nease, and Shubha Ramakrishnan. A programmable and configurable mixed-mode FPAA SoC. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 24(6):2253–2261, 2016.

[12] Afolabi Ige, Linhao Yang, Hang Yang, Jennifer Hasler, and Cong Hao. Analog system high-level synthesis for energy-efficient reconfigurable computing. *Journal of Low Power Electronics and Applications*, 13(4), 2023.

[13] Venkatesh Srinivasan, Guillermo J. Serrano, Jordan Gray, and Paul Hasler. A precision cmos amplifier using floating-gate transistors for offset cancellation. *IEEE Journal of Solid-State Circuits*, 42(2):280–291, 2007.

[14] Jennifer Hasler. The rise of SoC FPAA devices. In *2022 IEEE Custom Integrated Circuits Conference (CICC)*, pages 1–8. IEEE, 2022.

[15] Jennifer Hasler, Praveen Raj Ayyappan, Afolabi Ige, and Pranav Mathews. A 130nm cmos programmable analog standard cell library. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 71(6):2497–2510, 2024.

[16] Pranav O. Mathews, Praveen Raj Ayyappan, Afolabi Ige, Swagat Bhattacharyya, Linhao Yang, and Jennifer O. Hasler. A 65 nm cmos analog programmable standard cell library for mixed-signal computing. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, pages 1–11, 2024.

[17] Matt Kucic, Hasler, Jeff Dugger, and David Anderson. Programmable and adaptive analog filters using arrays of floating-gate circuits. In *Proceedings 2001 Conference on Advanced Research in VLSI. ARVLSI 2001*, pages 148–162. IEEE, 2001.

[18] Hasler. Low-power programmable signal processing. In *Fifth International Workshop on System-on-Chip for Real-Time Applications (IWSOC'05)*, pages 413–418. IEEE, 2005.

[19] Bo Marr, Brian Degnan, Hasler, and David Anderson. Scaling energy per operation via an asynchronous pipeline. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 21(1):147–151, 2012.

[20] Jennifer Hasler. Analog abstraction, computation, and numerical analysis. In *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5. IEEE, 2018.

[21] Jennifer Hasler. Opportunities in physical computing driven by analog realization. In *2016 IEEE International Conference on Rebooting Computing (ICRC)*, pages 1–8, 2016.

[22] Nikita Stroev and Natalia G. Berloff. Analog photonics computing for information processing, inference, and optimization. *Adv. Quantum Technol.*, 6(9):2300055, 2023.

[23] Alexander N Tait, Thomas Ferreira De Lima, Ellen Zhou, Allie X Wu, Mitchell A Nahmias, Bhavin J Shastri, and Paul R Prucnal. Neuromorphic photonic networks using silicon photonic weight banks. *Scientific reports*, 7(1):1–10, 2017.

[24] Xing Lin, Yair Rivenson, Nezih T Yardimci, Muhammed Veli, Yi Luo, Mona Jarrahi, and Aydogan Ozcan. All-optical machine learning using diffractive deep neural networks. *Science*, 361(6406):1004–1008, 2018.

[25] Johannes Feldmann, Nathan Youngblood, Maxim Karpov, Helge Gehring, Xuan Li, Maik Stappers, Manuel Le Gallo, Xin Fu, Anton Lukashchuk, Arslan Sajid Raja, et al. Parallel convolutional processing using an integrated photonic tensor core. *Nature*, 589(7840):52–58, 2021.

[26] Gouhei Tanaka, Toshiyuki Yamane, Jean Benoit Héroux, Ryosho Nakane, Naoki Kanazawa, Seiji Takeda, Hidetoshi Numata, Daiju Nakano, and Akira Hirose. Recent advances in physical reservoir computing: A review. *Neural Networks*, 115:100–123, 2019.

[27] Xingyuan Xu, Mengxi Tan, Bill Corcoran, Jiayang Wu, Andreas Boes, Thach G Nguyen, Sai T Chu, Brent E Little, Damien G Hicks, Roberto Morandotti, et al. 11 tops photonic convolutional accelerator for optical neural networks. *Nature*, 589(7840):44–51, 2021.

[28] Wim Bogaerts, Daniel Pérez, José Capmany, David AB Miller, Joyce Poon, Dirk Englund, Francesco Morichetti, and Andrea Melloni. Programmable photonic circuits. *Nature*, 586(7828):207–216, 2020.

[29] Carlos Ríos, Nathan Youngblood, Zengguang Cheng, Manuel Le Gallo, Wolfram HP Pernice, C David Wright, Abu Sebastian, and Harish Bhaskaran. In-memory computing on a photonic platform. *Science advances*, 5(2):eaau5759, 2019.

[30] Bhavin J Shastri et al. Photonics for artificial intelligence and neuromorphic computing. *Nat. Photon.*, 15(2):102–114, 2021.

[31] Ryan Hamerly, Liane Bernstein, Alexander Sludds, Marin Soljačić, and Dirk Englund. Large-scale optical neural networks based on photoelectric multiplication. *Physical Review X*, 9(2):021032, 2019.

[32] Ian AD Williamson, Tyler W Hughes, Momchil Minkov, Ben Bartlett, Sunil Pai, and Shanhui Fan. Reprogrammable electro-optic nonlinear activation functions for optical neural networks. *IEEE Journal of Selected Topics in Quantum Electronics*, 26(1):1–12, 2019.

[33] Kengo Nozaki, Shinji Matsuo, Takuro Fujii, Koji Takeda, Akihiko Shinya, Eiichi Kuramochi, and Masaya Notomi. Femtofarad optoelectronic integration demonstrating energy-saving signal conversion and nonlinear functions. *Nature Photonics*, 13(7):454–459, 2019.

[34] Matthias Wuttig, Harish Bhaskaran, and Thomas Taubner. Phase-change materials for non-volatile photonic applications. *Nature Photonics*, 11(8):465–476, 2017.

[35] Natalia G Berloff, Matteo Silva, Kirill Kalinin, Alexis Askitopoulos, Julian D Töpfer, Pasquale Cilibrizzi, Wolfgang Langbein, and Pavlos G Lagoudakis. Realizing the classical xy hamiltonian in polariton simulators. *Nature materials*, 16(11):1120, 2017.

[36] Yichen Shen, Nicholas C Harris, Scott Skirlo, Mihika Prabhu, Tom Baehr-Jones, Michael Hochberg, Xin Sun, Shijie Zhao, Hugo Larochelle, Dirk Englund, et al. Deep learning with coherent nanophotonic circuits. *Nature Photonics*, 11(7):441–446, 2017.

[37] Gordon Wetzstein, Aydogan Ozcan, Sylvain Gigan, Shanhui Fan, Dirk Englund, Marin Soljačić, Cornelia Denz, David AB Miller, and Demetri Psaltis. Inference in artificial intelligence with deep optics and photonics. *Nature*, 588(7836):39–47, 2020.

[38] Chong Li, Xiang Zhang, Jingwei Li, Tao Fang, and Xiaowen Dong. The challenges of modern computing and new opportunities for optics. *PhotoniX*, 2(1):1–31, 2021.

[39] Shashank Misra, Leslie C. Bland, Suma G. Cardwell, Jean Anne C. Incorvia, Conrad D. James, Andrew D. Kent, Catherine D. Schuman, J. Darby Smith, and James B. Aimone. Probabilistic neural computing with stochastic devices. *Advanced Materials*, 35(37), 11 2022.

[40] Jan Kaiser and Supriyo Datta. Probabilistic computing with p-bits. *Applied Physics Letters*, 119(15):150503, 10 2021.

[41] Thomas M. Conte, Erik P. Debenedictis, Natesh Ganesh, Todd Hylton, John Paul Strachan, R. Stanley Williams, Alexander A. Alemi, Lee Altenberg, Gavin E. Crooks, James P. Crutchfield, Lídia del Rio, Josh Deutsch, Michael Robert DeWeese, Khari Douglas, Massimiliano Esposito, Michael P. Frank, Robert Fry, Peter Harsha, Mark D. Hill, Christopher T. Kello, Jeffrey L. Krichmar, Suhas Kumar, Shih-Chii Liu, Seth

Lloyd, Matteo Marsili, Ilya Nemenman, Alex Nugent, Norman H. Packard, Dana Randall, Peter Sadowski, Narayana P. Santhanam, Robert Shaw, Adam Z. Stieg, Elan Stopnitzky, Christof Teuscher, Chris Watkins, David H. Wolpert, J. Joshua Yang, and Yan M. Yufik. Thermodynamic computing. *ArXiv*, abs/1911.01968, 2019.

[42] Natalia G. Berloff et al. Realizing the classical XY Hamiltonian in polariton simulators. *Nat. Mater.*, 16(11):1120–1126, sep 2017.

[43] Kai Peng, Wei Li, Natalia G Berloff, Xiang Zhang, and Wei Bao. Room temperature polaritonic soft-spin XY hamiltonian in organic–inorganic halide perovskites. *Nanophotonics*, 13(14):2651–2658, 2024.

[44] Brendan P. Marsh, Ronen M. Kroeze, Surya Ganguli, Sarang Gopalakrishnan, Jonathan Keeling, and Benjamin L. Lev. Entanglement and replica symmetry breaking in a driven-dissipative quantum spin glass. *In Press, Phys. Rev. X, arXiv:2307.10176*, 2023.

[45] Micha Nixon, Eitan Ronen, Asher A Friesem, and Nir Davidson. Observing geometric frustration with thousands of coupled lasers. *Physical review letters*, 110(18):184102, 2013.

[46] Kirill Kalinin, George Mourgias-Alexandris, Hitesh Ballani, Natalia G Berloff, James H Clegg, Daniel Cletheroe, Christos Gkantsidis, Istvan Haller, Vassily Lyutsarev, Francesca Parmigiani, Lucinda Pickup, et al. Analog iterative machine (aim): using light to solve quadratic optimization problems with mixed variables. *arXiv preprint arXiv:2304.12594*, 2023.

[47] D Pierangeli, G Marcucci, and C Conti. Large-scale photonic ising machine by spatial light modulation. *Phys. Rev. Lett.*, 122(21):213902, 2019.

[48] Marvin Syed, Kirill Kalinin, and Natalia Berloff. Beyond digital: Harnessing analog hardware for machine learning. In *Machine Learning with New Compute Paradigms*, 2023.

[49] Tianshi Wang, Leon Wu, Parth Nobel, and Jaijeet Roychowdhury. Solving combinatorial optimisation problems using oscillator based Ising machines. *Natural Computing*, pages 1–20, April 2021.

[50] Alexander A Green, Jongmin Kim, Duo Ma, Pamela A Silver, James J Collins, and Peng Yin. Complex cellular logic computation using ribocomputing devices. *Nature*, 548(7665):117–121, 2017.

[51] Sara Achour. *Compilation Techniques for Reconfigurable Analog Devices*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2021.

[52] Yuan-Jyue Chen, Benjamin Groves, Richard A Muscat, and Georg Seelig. DNA nanotechnology from the test tube to the cell. *Nature nanotechnology*, 10(9):748–760, 2015.

[53] Boya Wang, Siyuan Stella Wang, Cameron Chalk, Andrew Ellington, and David Soloveichik. Parallel molecular computation on digital data stored in DNA. *bioRxiv*, pages 2022–08, 2022.

[54] Friedrich C Simmel, Bernard Yurke, and Hari R Singh. Principles and applications of nucleic acid strand displacement reactions. *Chemical reviews*, 119(10):6326–6369, 2019.

[55] Randolph Lopez, Ruofan Wang, and Georg Seelig. A molecular multi-gene classifier for disease diagnostics. *Nature chemistry*, 10(7):746–754, 2018.

[56] Christopher P Kempes, David Wolpert, Zachary Cohen, and Juan Pérez-Mercader. The thermodynamic efficiency of computations made in cells across the range of life. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 375(2109):20160343, 2017.

[57] Andrew Camilli and Bonnie L. Bassler. Bacterial small-molecule signaling pathways. *Science*, 311(5764):1113–1116, 2006.

[58] S. Artavanis-Tsakonas, M. D. Rand, and R. J. Lake. Notch signaling: Cell fate control and signal integration in development. *Science*, 284(5415):770–776, 1999.

[59] J. Garcia-Ojalvo, M. B. Elowitz, and S. H. Strogatz. Modeling a synthetic multicellular clock: Repressilators coupled by quorum sensing. *Proc. Nat. Acad. Sci. USA*, 101(30):10955–10960, 2004.

[60] Oindrila Chatterjee and Shantanu Chakrabartty. Decentralized global optimization based on a growth transform dynamical system model. *IEEE Transactions on Neural Networks and Learning Systems*, 29(12):6052–6061, 2018.

[61] Gupta, P., Chandak, R., Debnath, A., Traner, M., Watson, B. M., Huang, H., Derami, H. G., Baldi, H., Chakrabartty, S., Raman, B., and Singamaneni, S. Augmenting insect olfaction performance through nano-neuromodulation. *Nature Nanotechnology*, 19(5):677–687, 2024.

[62] Tan, H.-Y., Cho, H., and Lee, L. P. Human mini-brain models. *Nature Biomedical Engineering*, 5(1):11–25, 2021.

[63] Smirnova, L., Caffo, B. S., Gracias, D. H., Huang, Q., Morales Pantoja, I. E., Tang, B., Zack, D. J., Berlinicke, C. A., Boyd, J. L., Harris, T. D., Johnson, E. C., Kagan, B. J., Kahn, J., Muotri, A. R., Paulhamus, B. L., Schwamborn, J. C., Plotkin, J., Szalay, A. S., Vogelstein, J. T., Worley, P. F., and Hartung, T. Organoid intelligence (oi): The new frontier in biocomputing and intelligence-in-a-dish. *Frontiers in Science*, 1, 2023.

[64] Luca Cardelli, Mirco Tribastone, and Max Tschaikowski. From electric circuits to chemical networks. *Natural Computing*, 19:237–248, 2020.

[65] Marko Vasić, David Soloveichik, and Sarfraz Khurshid. CRN++: Molecular programming language. *Natural Computing*, 19:391–407, 2020.

[66] Niranjan Srinivas, James Parkin, Georg Seelig, Erik Winfree, and David Soloveichik. Enzyme-free nucleic acid dynamical systems. *Science*, 358(6369):eaal2052, 2017.

[67] Uri Alon. *An introduction to systems biology: design principles of biological circuits*. Chapman and Hall/CRC, 2019.

[68] Maxim P Nikitin. Non-complementary strand commutation as a fundamental alternative for information processing by dna and gene regulation. *Nature Chemistry*, 15(1):70–82, 2023.

[69] William Poole, Andrés Ortiz-Munoz, Abhishek Behera, Nick S Jones, Thomas E Ouldridge, Erik Winfree, and Manoj Gopalkrishnan. Chemical boltzmann machines. In *DNA Computing and Molecular Programming: 23rd International Conference, DNA 23, Austin, TX, USA, September 24–28, 2017, Proceedings 23*, pages 210–231. Springer, 2017.

[70] Marko Vasić, Cameron Chalk, Austin Luchsinger, Sarfraz Khurshid, and David Soloveichik. Programming and training rate-independent chemical reaction networks. *Proceedings of the National Academy of Sciences*, 119(24):e2111552119, 2022.

[71] Catherine Graves. High performance, power efficient hardware accelerators: emerging devices, circuits and architecture co-design. In Francesca Palumbo, Michela Becchi, Martin Schulz, and Kento Sato, editors, *Proceedings of the 16th ACM International Conference on Computing Frontiers, CF 2019, Alghero, Italy, April 30 - May 2, 2019*, page 1. ACM, 2019.

[72] Marko Vasić, David Soloveichik, and Sarfraz Khurshid. Crn++: Molecular programming language. *Natural Computing*, 19(2):391–407, 2020.

[73] Sara Achour. *Compilation Techniques for Reconfigurable Analog Devices*. PhD thesis, Massachusetts Institute of Technology, USA, 2021.

[74] Ruibing Song, Chunshu Wu, Chuan Liu, Ang Li, Michael C. Huang, and Tong Geng. DS-GL: advancing graph learning via harnessing nature's power within scalable dynamical systems. In *51st ACM/IEEE Annual International Symposium on Computer Architecture, ISCA 2024, Buenos Aires, Argentina, June 29 - July 3, 2024*, pages 45–57. IEEE, 2024.

[75] Joao Ambrosi, Aayush Ankit, Rodrigo Antunes, Sai Rahul Chalamalasetti, Soumitra Chatterjee, Izzat El Hajj, Guilherme Fachini, Paolo Faraboschi, Martin Foltin, Sitao Huang, Wen-Mei Hwu, Gustavo Knuppe, Sunil Vishwanathpur Lakshminarasimha, Dejan Milojicic, Mohan Parthasarathy, Filipe Ribeiro, Lucas Rosa, Kaushik Roy, Plinio Silveira, and John Paul Strachan. Hardware-software co-design for an analog-digital accelerator for machine learning. In *2018 IEEE International Conference on Rebooting Computing (ICRC)*, pages 1–13, 2018.

[76] Siddharth Joshi, Chul Kim, Sohmyung Ha, and Gert Cauwenberghs. From algorithms to devices: Enabling machine learning through ultra-low-power vlsi mixed-signal array processing. In *2017 IEEE Custom Integrated Circuits Conference (CICC)*, pages 1–9. IEEE, 2017.

[77] Darshit Mehta, Mustafizur Rahman, Kenji Aono, and Shantanu Chakrabartty. An adaptive synaptic array using fowler–nordheim dynamic analog memory. *Nature communications*, 13(1):1670, 2022.

[78] Matteo Cartiglia, Filippo Costa, Shyam Narayanan, Cat-Vu H Bui, Hasan Ulusan, Nicoletta Risi, Germain Haessig, Andreas Hierlemann, Fernando Cardes, and Giacomo Indiveri. A 4096 channel event-based multielectrode array with asynchronous outputs compatible with neuromorphic processors. *Nature Communications*, 15(1):7163, 2024.

[79] Amin Fazel, Amit Gore, and Shantanu Chakrabartty. Resolution enhancement in $\sigma\delta$ learners for superresolution source separation. *IEEE transactions on signal processing*, 58(3):1193–1204, 2009.

[80] Feichi Zhou and Yang Chai. Near-sensor and in-sensor computing. *Nature Electronics*, 3(11):664–671, 2020.

[81] Xinming Liu, Emre Gönültaş, and Christoph Studer. Analog-to-feature (a2f) conversion for audio-event classification. In *2018 26th European Signal Processing Conference (EUSIPCO)*, pages 2275–2279. IEEE, 2018.

[82] Siddharth Joshi, Chul Kim, Sohmyung Ha, Yu Mike Chi, and Gert Cauwenberghs. 21.7 2pj/mac 14b 8x8 linear transform mixed-signal spatial filter in 65nm cmos with 84db interference suppression. In *2017 IEEE International Solid-State Circuits Conference (ISSCC)*, pages 364–365. IEEE, 2017.

[83] Vincent T Lee, Armin Alaghi, John P Hayes, Visvesh Sathe, and Luis Ceze. Energy-efficient hybrid stochastic-binary neural networks for near-sensor computing. In *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017*, pages 13–18. IEEE, 2017.

[84] Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J Davison, Jörg Conradt, Kostas Daniilidis, et al. Event-based vision: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(1):154–180, 2020.

[85] Zephan M Enciso, Seyed Hadi Mirfarshbafan, Oscar Castañeda, Clemens JS Schaefer, Christoph Studer, and Siddharth Joshi. Analog vs. digital spatial transforms: A throughput, power, and area comparison. In *2020 IEEE 63rd International Midwest Symposium on Circuits and Systems (MWSCAS)*, pages 125–128. IEEE, 2020.

[86] S Dutta, W Chakraborty, J Gomez, K Ni, S Joshi, and S Datta. Energy-efficient edge inference on multi-channel streaming data in 28nm hkmg fefet technology. In *2019 symposium on VLSI technology*, pages T38–T39. IEEE, 2019.