Computing Nearer to Data

Thomas M. Coughlin, Coughlin Associates
William R. Tonti, IEEE

Nonvolatile Memory Express over Fabrics and Compute Express Link, combined with new memory technologies, are creating computational storage and capabilities near and in memory, driving new computer architectures for use in data centers, at the network edge, and in endpoint devices.

As the feature size of semiconductors approaches 1 nm, both Moore's[1] and Dennard's[2] laws, which have described transistor density and power scaling for many decades, are reaching an endpoint. Tonti[3] described many of the process tweaks and improvements in the work of Moore[1] and Dennard[2] in semiconductor technology, for example, the change of the switch from bipolar to CMOS technology, silicon-on-insulator devices, mobility enhancement using film stress, 3D multigated transistors, high-k dielectrics, and the ever-increasing active and standby chip power.

One outcome is related to semiconductor device lithography: the rate of decline in the minimum feature size, typically coined the *node*, is slowing. This slowing down[1,2] creates new challenges for processing data. At the same time, the demand for data processing for big data applications, such as artificial intelligence (AI) and the growth of data generation, are creating greater demands for processing.

With a conventional von Neumann computing architecture, data processing involves moving large volumes of data, typically in and out of a computing unit. This movement of data generally constrains the system performance and requires ever-increasing power. Constraints on the capability and performance of CPU-based processing have resulted in new approaches for the design of computing systems that may cost-effectively meet the growing demand for processing data in data centers, at the network edge, and in endpoint devices. One of the earlier techniques to manage this complexity is the rise of the multicore CPU (MCPU), clocked at a frequency that is typically lower than that used when one tries to do the same with a single-core CPU (SCPU).

Other new approaches also include the increasing use of domain-specific processors, which are

Digital Object Identifier 10.1109/MC.2022.3171354
Date of current version: 4 July 2022

EDITORS
NORITA AHMAD American University of Sharjah;
nahmad@aus.edu
PREETI CHAUHAN IEEE Reliability Society;
preeti.chauhan@ieee.org

proliferating in many computing scenarios, including, for example, data center systems on chip (SoCs).[4] These new approaches include offload processing from an MCPU to reduce data movement and latency for particular types of data processing. Domain-specific processors are often located close to the memory and/or storage that hold the data they are processing. Newer storage and memory system architectures based upon Nonvolatile Memory express (NVMe) and Compute Express Link (CXL) are helping to bring processing closer to where the data live. In-memory computing using large random-access memory (RAM) shared blocks is also a new technique to maximize MCPU efficiency and minimize power requirements.

COMPUTATIONAL STORAGE

The increasing use of NAND-flash-based solid-state drives (SSDs) has enabled faster storage, especially with SSDs using the NVMe interface running on a peripheral component interconnect express (PCIe) bus. NVMe may soon be a universal storage interface with hard disk drives also being built with a native NVMe interface.[5] NVMe can also be transported over fabrics, enabling NVMe over Fabrics (NVMe-oF). NVMe-oF allows pooling of NVMe storage devices and also supports the use of domain-specific processors near to the NVMe storage devices for various applications, including data reduction (deduplication and compression), data security, and some other types of local data processing. Figure 1 shows a computational storage device with an advanced processor built into the drive [Figure 1(a)] as well as a computational storage array that includes computational storage processors (CSPs) in a network, such as NVMe-oF [Figure 1(b)].

COMPUTING NEAR MEMORY

The CXL interconnect for memory is also built on the PCIe bus and provides a way for a processing device (such as an SCPU, MCPU, or GPU) to access additional shared memory or to include domain-specific processors in a memory pool, close to the data being processed.[7] This allows one to process data faster, using less power than for a CPU or GPU. CXL enables computing near memory as well as memory tiering, including tiering with nonvolatile memories.

Both NVMe-oF and CXL are enabling pooling of storage and memory as well as data center disaggregation and the ability to compose virtual computing systems with shared processing, storage, memory, and network systems. These virtual systems can effectively use domain-specific processing and CPUs to achieve the most efficient and cost-effective solutions to meet the needs of various applications. CXL-based components have been presented, and the first CXL-based computer systems should be introduced by the end of 2022.

IN-MEMORY COMPUTING

In addition to enabling computing near memory and storage, there are efforts to include processing very close to or within a memory device itself. This may be done using conventional digital memory technologies, but it could also involve the use of new approaches, such as analog neuromorphic processing with various memory technologies. Different types of in-memory computing may be better for solving different types of problems. Let's look at various ways to do in-memory computing, assess in-memory computing products, and determine where they may be most useful.

In-memory computing encompasses a great many products and concepts. It is sometimes also referred to as *compute in memory*. Processing

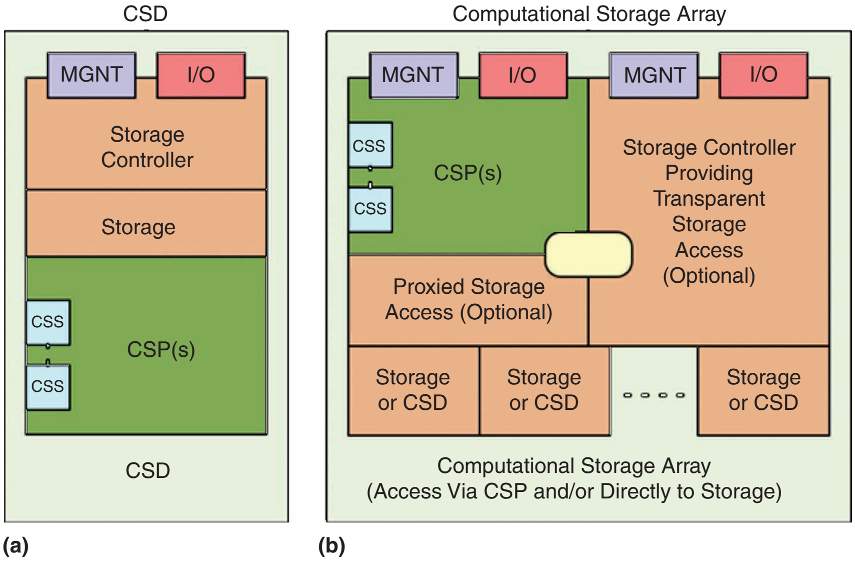
CSD
MGNT
I/O
Storage Controller
Storage
CSS
CSS
CSP(s)
CSD
Computational Storage Array
MGNT
I/O
MGNT
I/O
CSS
CSS
CSP(s)
Storage Controller Providing Transparent Storage Access (Optional)
Proxied Storage Access (Optional)
Storage or CSD
Storage or CSD
Storage or CSD
Computational Storage Array
(Access Via CSP and/or Directly to Storage)
(a)
(b)

FIGURE 1. Examples of computational storage (a) in a drive and (b) in an array. CSD: computational storage drive; CSP: computational storage processor; MGNT: Management; CSS: computational storage service; I/O: input/output. (Source: Storage Networking Industry Association; used with permission.[6])

in memory (PIM) is an older concept that integrates RAM and a processor on a single PIM chip (somewhat similar to putting a processor in a storage device for computational storage, as discussed previously). Both PIM and in-memory computing offload processing from the CPU, reducing energy consumption and processor latency and leaving the CPU to do other tasks.

Putting a processor and memory on a single chip allows faster processing of the data and reduces the movement of data. This approach also reduces the power budget as the largest topology requiring power, the input–output drivers, are no longer required to move data on and off chip. PIM chips can be used to increase relational database processing speeds when the data are loaded directly into the RAM or into a flash memory device with computational capabilities. PIM chips are also used for monitoring and predictive maintenance, financial transactions, and fraud detection. PIM chips are faster but much more expensive than computational storage devices. Figure 2 illustrates the difference of data movement for conventional computation in a processor and for in-memory computing, where the computation is done in the memory itself.

In the future, various methods of stacking and connecting die, known as *3D integration* (*3DI*), will become more common, bringing computation and memory into close proximity.[9] Eventually, 3DI could become the new SoC standard. This would make near and in-memory computing even more common and powerful and result in denser and more powerful electronic packages. 3DI methodology could

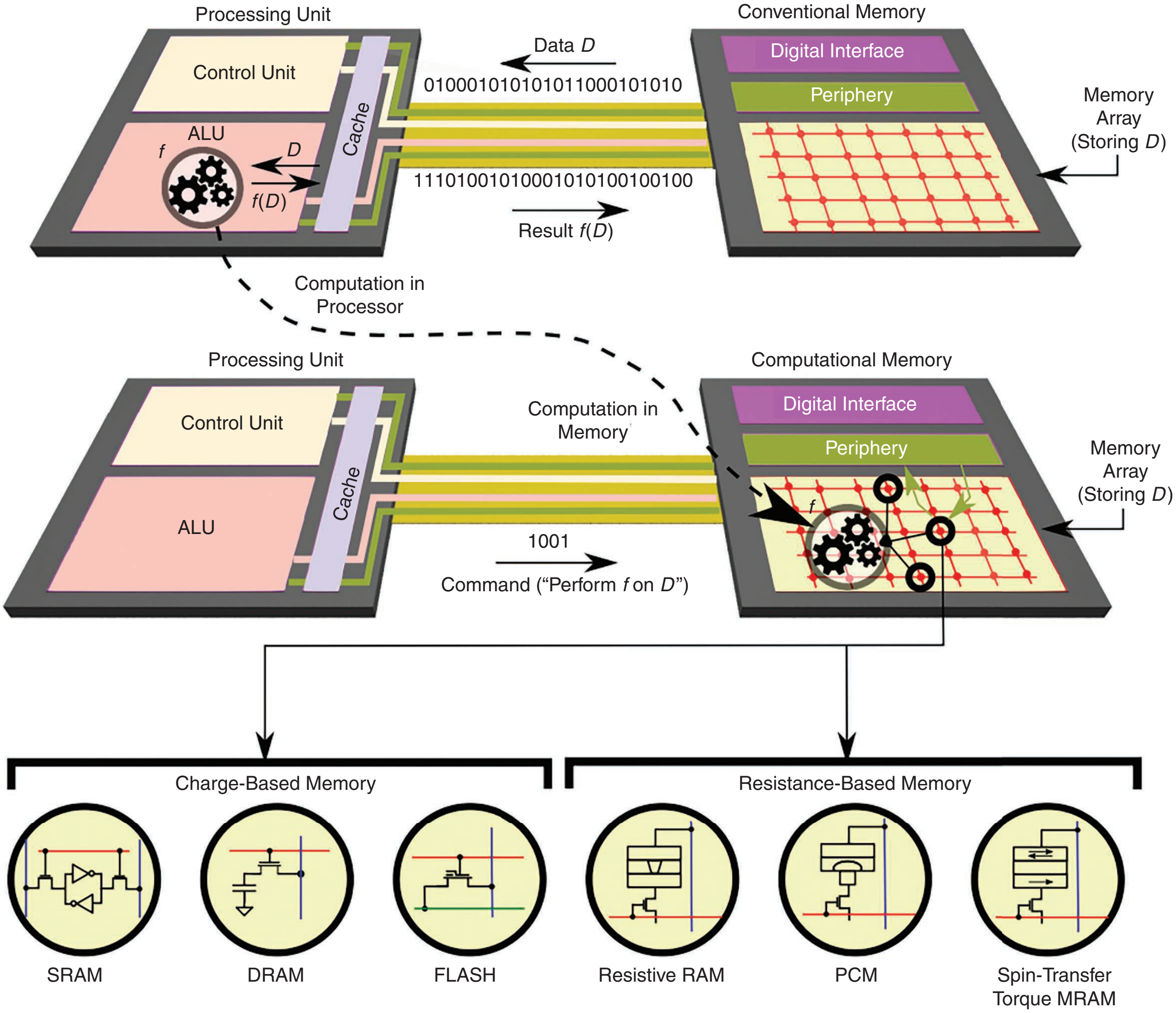

FIGURE 2. Illustrating in-memory computing versus traditional computing. ALU: arithmetic logic unit; SRAM: static RAM; DRAM: dynamic RAM; PCM: phase-change memory; MRAM: magnetoresistive RAM. (Source: IBM; used with permission.[8])

become the new scaling law for denser and faster data processing.

In-memory computing performs certain computation tasks by exploiting the physical attributes of memory devices, which can be charged-based or resistance-based devices, as shown in Figure 2. Charge-based devices include common volatile memory technologies [static RAM (SRAM) and dynamic RAM (DRAM)] as well as flash memory. The resistance-based memories enable interesting computing modes that mimic some of the operations of neurons in living creatures and are often referred to as neural networks. All of the resistive memory devices shown are nonvolatile memories; that is, the data remain on the device even after the power is removed.

PATHS TOWARD IN-MEMORY COMPUTING

Let's look at some examples of in-memory computing, starting with devices using phase-change memory (PCM). PCM has a low-resistance crystalline state and a high-resistance amorphous state. Writing is done by applying a voltage pulse to the crystalline material to make some of it amorphous and thus increase the resistance of the device. Data are read from the memory cell using lower currents that don't write on the cell. The amount of amorphous material written in the cell depends upon how high the voltage pulse is and how long it is applied. This means that writing can create various levels of resistance, depending upon the voltage applied.

In addition, if pulses are applied repeatedly with the same amplitude to a higher resistive cell, the resistance drops with the repeated pulses. So, PCM can store various analog values and also integrate applied pulses. These capabilities allow the creation of a crosspoint array of such memory cells (or synapses) that can perform mathematical functions (computing) in a neuromorphic network, mimicking some of the operations of neurons in a brain.

Training of these cells is an accumulation (or integration) function based upon applied voltages, and inference involves applying lower voltages through the array of memory cells and detecting the current levels at the output nodes of the trained memory array, as illustrated in Figure 3.

A crossbar PCM network configuration, such as the one in Figure 3, can be replicated into "tiles" of such networks. Each of these tiles can be used in a deep neural network (DNN) to store trained weights for a layer of the DNN at the memory cells as conductance values. The tiles perform the matrix–vector multiply operations that correspond to each layer in the DNN. Once trained, lower applied voltages can be used to do inference, looking for matches to the trained weights stored in the memory cells. Companies such as IBM and Intel have made neuromorphic DNN chips to develop this technology.

Forward and backward propagation are possible using such an analog in-memory computer. Accumulation is possible at high precision with weights being updated using accumulative behavior. The same hardware can be used for inference once trained. Figure 4 shows a block diagram of the operation of DNN training using in-memory computing with PCM.

There are challenges to making these devices work well, for example, imprecision arising from factors such as conductance fluctuation and drift. Despite these challenges, PCM array chips have been built with on-chip matrix–vector–multiply operating at more than 1 GHz and with measured energy efficiencies of 10.5 trillion operations per second (TOPS)/W with a performance density of 1.59 TOPS/mm^2.[12] Shrinking these neuromorphic arrays could provide faster devices with energy efficiencies and performance densities of 262 TOPS/W and 655 TOPS/mm^2.[13]

In addition to using PCM for neuromorphic arrays, there have been neuromorphic computing devices designed with resistive memories and magnetorestrictive RAM (MRAM) memories.[14,15] There is also a body of work on spintronic computation using spins rather than electric currents,[16] which can be in close proximity to MRAM memories. Higher speed analog computing, such as image recognition, is also possible using photonic in-memory computing.[17] Neural networks also can be combined with an external memory, which may assist in relearning and adapting to new data.[18] Neural networks made with spiking neurons in spiking neural networks (SNNs) provide even more brain-like computation that provide high levels of asynchronous parallel processing and very high energy efficiencies. SNN chips have been made and used

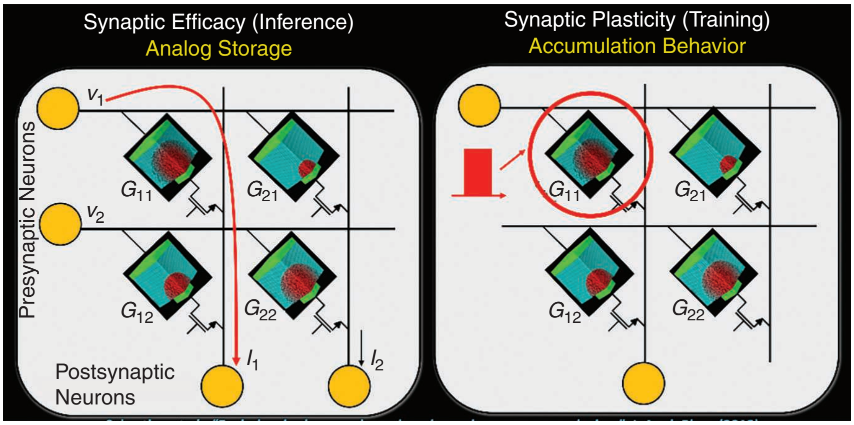

FIGURE 3. Inference and training of phase-change synapses. (Source: IBM; used with permission.[10])

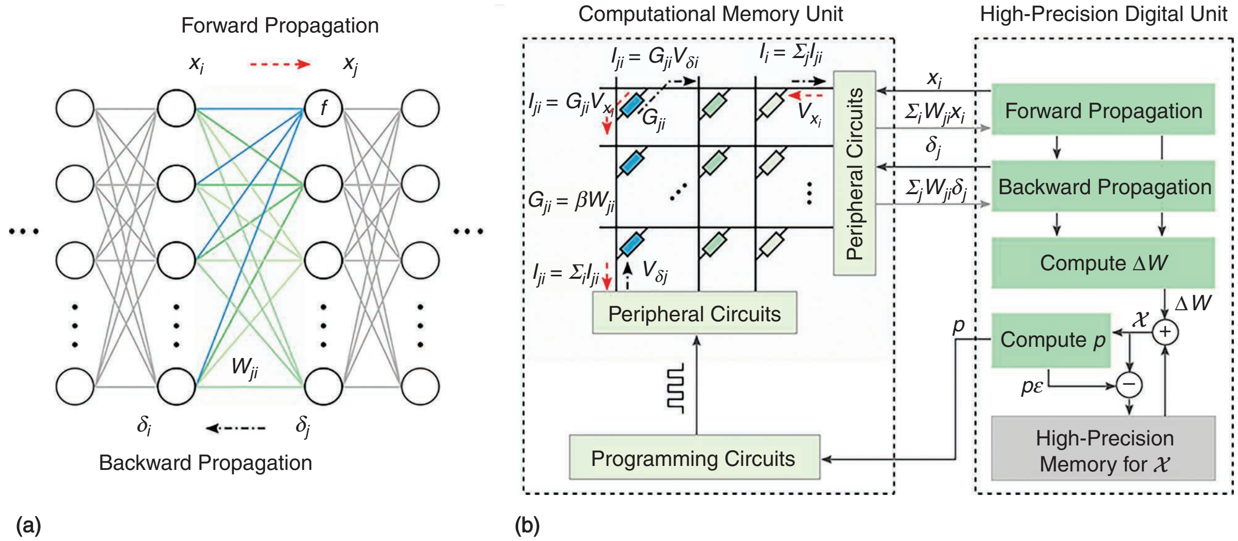

FIGURE 4. DNN training with in-memory computing. (Source: IBM; used with permission.[11])

for DNN acceleration,[19] although SSNs are not yet available in commercial products.

The demand for data processing is increasing to support Internet of things, AI, machine learning, and other big data applications. To keep costs and energy consumption at acceptable levels, computing is evolving from von Neumann architectures that require lots of data movement to and from a CPU to a more distributed computing model, particularly where processing is done much closer to data.

New interface technologies like NVMe are enabling computational computing, where some data processing is done in the storage device or in an NVMe-oF network to offload processing from the CPU. CXL is enabling similar networking of "far" memory devices that may include PIM and heterogeneous memory technologies. NVMe and CXL enable the creation of storage and memory pools. These technologies will transform the design of data centers.

In addition to PIM, in-memory computing solutions are creating even more distributed computing that offloads traditional CPUs. This includes various approaches using memory for processing with neuromorphic memories. These neuromorphic memories can perform analog mathematical functions and will play important roles in data processing in data centers, at the network edge, and in endpoint devices.

3DI enablement will make integration of computing, memory, and storage technologies even more effective and will allow a designer the freedom to choose the best MCPU, memory, and storage technologies for a particular application. ◼

REFERENCES

1. G. Moore, "The future of integrated electronics," Computer History Museum, Mountain View, CA, USA, 1965. [Online]. Available: https://www.computerhistory.org/collections/catalog/102770836
2. R. Dennard, F. H. Gaensslen, H.-N. Yu, V. Leo Rideovt, E. Bassous, and A. R. Leblanc, "Design of ion-implanted MOSFETs with very small physical dimensions," *IEEE J. Solid-State Circuits*, vol. 9, no. 5, pp. 256–268, Oct. 1974, doi: 10.1109/JSSC.1974.1050511.
3. W. Tonti, "MOS technology drivers," *IEEE Trans. Device Mater. Rel.*, vol. 8, no. 2, pp. 406–415, 2008, doi: 10.1109/TDMR.2008.922223.
4. T. Coughlin, "New electronic architectures," *IEEE Consum. Electron. Mag.*, vol. 9, no. 2, pp. 67–69, Mar. 1, 2020, doi: 10.1109/MCE.2019.2954255.
5. T. Coughlin. "NVMe for all data center storage." Forbes.com. https://www.forbes.com/sites/tomcoughlin/2020/05/15/nvme-for-all-data-center-storage/?sh=12bf9a67e05f (Accessed: Apr. 20, 2022).
6. S. Shadley and N. Adams, "What happens when Compute meets Storage?" SNIA, Colorado Springs, CO, USA, 2019. [Online]. Available: https://www.snia.org/sites/default/files/SDC/2019/presentations/Computational/Shadley_Scott_Adams_Nick_What_Happens_when_Compute_meets_Storage_Computational_Storage_TWG.pdf
7. Compute Express Link. [Online]. Available: https://www.computeexpresslink.org (Accessed: Apr. 20, 2022).
8. A. Sebastian, M. Le Gallo, R. Khaddam-Aljameh, and E. Eleftheriou, "Memory devices and applications for in-memory computing," *Nature Nanotechnol.*, vol. 15, no. 7, pp. 529–544, 2020, doi: 10.1038/s41565-020-0655-z.
9. "Heterogeneous integration roadmap," IEEE Electron. Packag. Soc., 2020. [Online]. Available:

https://eps.ieee.org/technology/heterogeneous-integration-roadmap

10. A. Sabastian, M. Le Gallo, G. W. Burr, S. Kim, M. BrightSky, and E. Eleftheriou, "Brain-inspired computing using phase-change memory devices," *J. Appl. Phys.*, vol. 124, p. 111.101, Aug. 2018, doi: 10.1063/1.5042413.
11. S. R. Nandakumar *et al.*, "Mixed-precision deep learning based on computational memory," *Frontier Neurosci.*, vol. 14, p. 406, May 12, 2020. doi: 10.3389/fnins.2020.00406.
12. R. Khaddam-Aljameh *et al.*, "HERMES Core – A 14nm CMOS and PCM-based in-memory compute core using an array of 300ps/LSB linearized CCO-based ADCs and local digital processing," in *Proc. 2021 Symp. VLSI Circuits*, pp. 1–2, doi: 10.23919/VLSICircuits52068.2021.9492362.
13. A. Sabastian, "In-memory computing for deep learning and beyond," Max Planck Soc., Munich, Germany, Jul. 2021. [Online]. Available: https://www.mpi-halle.mpg.de/541049/in-memory-computing-for-deep-learning-and-beyond
14. G. Pedretti and D. Ielmini, "In-memory computing with resistive memory circuits: status and outlook," *Electronics*, vol. 10, no. 9, p. 1063, 2021, doi: 10.3390/electronics10091063.
15. Q. Shao, Z. Wang, and J. J. Yang, "Efficient AI with MRAM," *Nature Electron.*, vol. 5, no. 2, pp. 67–68, 2022, doi: 10.1038/s41928-022-00725-x.
16. K. L. Wang, H. Wu, S. A. Razavi, and Q. Shao, "Spintronic devices for low energy dissipation," presented at the IEEE Int. Electron Devices Meeting (IEDM), 2018, pp. 36.2.1–36.2.4, doi: 10.1109/IEDM.2018.8614671.
17. Y. Shen *et al.*, "Deep learning with coherent nanophotonic circuits," *Nature Photon.*, vol. 11, no. 7, pp. 441–446, 2017, doi: 10.1038/nphoton.2017.93.
18. A. Graves *et al.*, "Hybrid computing using a neural network with dynamic external memory," *Nature*, vol. 538, no. 7626, pp. 471–476, 2016, doi: 10.1038/nature20101.
19. T. Tuma, A. Pantazi, M. Le Gallo, A. Sebastian, and E. Eleftheriou, "Stochastic phase-change neurons," *Nature Nanotechnol.*, vol. 11, no. 8, pp. 693–699, 2016, doi: 10.1038/nnano.2016.70.

THOMAS M. COUGHLIN is president of Coughlin Associates, San Jose, California, 95124 USA. He is a Fellow of IEEE. Contact him at tom@tomcoughlin.com.

WILLIAM R. TONTI is a senior director, managing future technology directions at IEEE. He is a Fellow of IEEE. Contact him at w.r.tonti@ieee.org.

Digital Object Identifier 10.1109/MC.2022.3178651