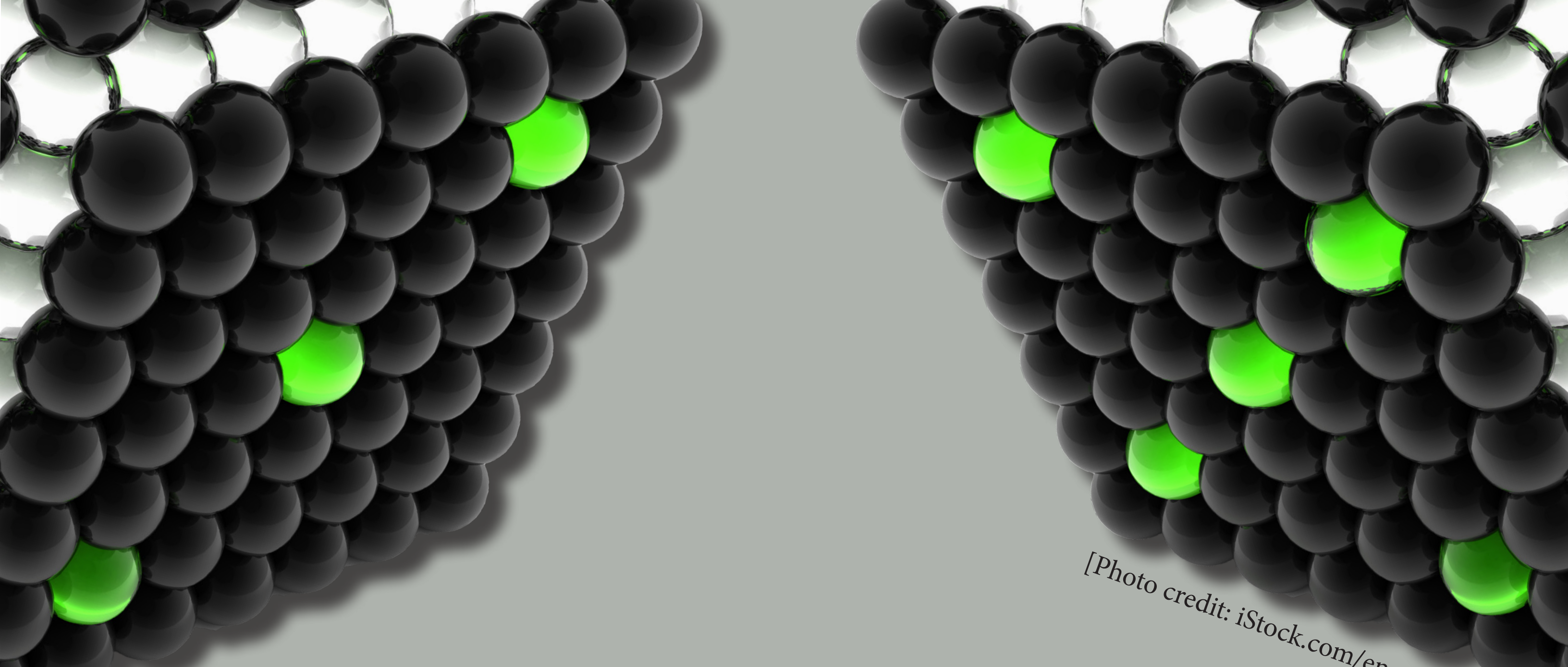


[Photo credit: iStock.com/enot-poloskun]

The Ising Machine—A Probabilistic Processing-in-Memory Computer

Lauren Huckaba

High-performance computing (HPC) systems are growing increasingly complex. With this, the error rate of computation is growing and faults are becoming harder to diagnose and correct. Traditionally, the field of resilience is dedicated to developing methods to keep applications running to a correct solution in spite of errors, but the more complex the computer, the more costly these methods become. Rather than expend energy combating these faults, one possibility is to accept these errors and allow nondeterminism in our computations in exchange for greater energy efficiency.

Further, computer applications must process volumes of data so large that the energy and performance costs of moving this data from memory to the central processing unit (CPU) dominates the total cost of computation. Processing in memory (PIM) is a novel, non-von Neumann model of computation that saves energy by doing computation and storing data in the same place [1].

In this article, we describe a probabilistic PIM computer, made entirely of existing electronic components, based on the Ising model. We discuss how we can use an Ising model in inverse ways to solve two types of important problems.

The Ising model and the Ising problem

An Ising model is a mathematical model originally formulated to describe ferromagnetism in statistical mechanics. It consists of a lattice of spins in one of two states (see figure 1).

There is a measure of the surrounding magnetic field corresponding to each spin and a measure that denotes the interaction between each pair of spins. We call these measurements "weights." We can write down an expression for the total energy of the system in terms of the spin states and weights. The expression for total energy is known as the Hamiltonian (see figure 2).

In the physical world, an Ising system has the important property that once configured with a set of weights, the spins try to settle to a configuration that yields the lowest total energy. It achieves this state with some probability and this is where the nondeterminism comes in.

The goal of the Ising Problem is to fix a set of weights and find the configuration of spins that minimizes the Hamiltonian. The Ising Problem is a combinatorial optimization problem in which the set of local minima grows exponentially as a function of the number of spins, making it NP-hard. Many other combinatorial optimization problems, such as the traveling salesman problem and the MAX-CUT problem, can be mapped to the Ising model. Hence, the ability to efficiently solve the Ising Problem can potentially lead to solutions to a large class of other combinatorial optimization problems [2].

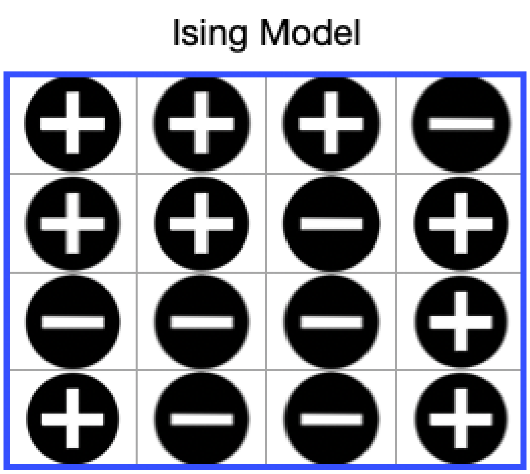


FIGURE 1. An Ising model is a lattice of spins, some positive, or taking the value 1, and some negative, or taking the value -1.

$$H(s) = \sum_{i=1}^{n} h_i s_i + \sum_{i,j=1}^{n} J_{ij} s_i s_j$$

FIGURE 2. In the equation above, $H(s)$ represents the Hamiltonian or total energy of the Ising system when in configuration s. Here, $s = (s_1,\ldots,s_n)$ is the configuration of spins, h_i is the measure of the magnetic field surrounding spin s_i, and J_{ij} is the measure of the interaction between s_i and s_j. We refer to the h_i's and J_{ij}'s as "weights" in this article.

Using an Ising model to perform arithmetic—the inverse Ising problem

The Ising problem above consists of fixing weights and determining the appropriate configuration of spins. Alternately, we can solve the inverse Ising problem by fixing a configuration of spins and finding the weights that minimize the Hamiltonian. In doing this, we can use an Ising model to do arithmetic.

We can fix a set of spins that corresponds to a correct arithmetic equation. Then we solve an optimization problem where we determine the weights that minimize the Hamiltonian. We also add constraint inequalities to our optimization problem to ensure that configurations corresponding to incorrect answers do not give a lower total energy. Even for small problems, say 3-bit multiplication, the number of constraint inequalities is quite large. For this reason, we do not give a concrete example in this article.

The weights found by solving the optimization problem can be used to tune an Ising machine or simulator built to solve the Ising problem. As described in the previous section, the machine will then try to settle to a configuration that gives the lowest total energy. In this case, that configuration is the one that corresponds to the correct answer to our multiplication problem. Because of the way we set our constraints above, getting a correct answer is more likely than getting an incorrect answer. In the following subsection, we model this process mathematically.

At first glance, it may seem as though the inverse Ising problem is easier to solve than the Ising problem since Hamiltonian is clearly quadratic in the spin variables, but linear in the h's and J's. However, after setting up even a small inverse Ising problem, it becomes clear that the number of constraint equations grows exponentially in the number of spins. As such, it quickly becomes difficult to multiply numbers of more than a just a few bits in this way, and alas, the inverse Ising problem is NP-hard as well.

The Resilience and Probabilistic Computing team at the Laboratory for Physical Sciences (LPS) has spent the last few years working on the inverse Ising problem. We have a solution technique that involves an internally developed two-stage algorithm that first searches for a set of feasible parameters and then solves a system of constraints derived from the feasible parameters. Both stages have exponential complexity, but our team improved the solution time of a 3-bit multiplier from 120 days to under 10 minutes for a system of 32,000 constraints by reducing the problem to polynomial complexity. We also successfully solved the system of 267,000,000 constraints for the 4-bit multiplier. The total solve time was 27 days and used 5.5 terabytes of shared memory.

A mathematics illustration

In the absence of an actual Ising machine, we can compute a probability. The probability that the system settles to a certain configuration for a given set of weights is called the Boltzmann probability. It depends on the total energy of the system when in this configuration, as well as the noise present in the system (see figure 3).

Figure 4 (on the following page) shows probabilities of solutions to 2-bit multiplication problems. After solving the inverse Ising problem, we calculated the Boltzmann probability for each configuration, both those corresponding to incorrect solutions and the configuration corresponding to the correct solution, and added some noise into our computation. This illustrates the results of simulating the nondeterminism present in an actual Ising machine. Properly setting our constraints described above gives us control over our nondeterministic computation so that there is hope of obtaining a correct answer. In this model, there is no need for error correction, which allows for a more energy-efficient computation than in a traditional digital machine.

$$\mathbf{Prob}_\beta(s) = \frac{e^{-\beta H(s)}}{\sum_\sigma e^{-\beta H(\sigma)}}$$

FIGURE 3. The equation above represents the probability that an Ising system settles to a configuration s. β represents the noise present. More specifically, $\beta = 1/(k_B T)$, where T is the temperature of the system in kelvin and k_B is the Boltzmann constant. For the purpose of this article, we can think of β as simply the "noise term."

Hardware

While the Ising model dates back to the 1920s, it was re-popularized much later by D-Wave Systems in an attempt to simulate quantum mechanical phenomena to speed up computation, including computation to solve the aforementioned combinatorial optimization problems. Recently, alternative classical methods to solve the Ising problem have emerged using optoelectronic parametric oscillators, memristor cross-bar arrays, electronic oscillators, and GPU-based algorithms [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12].

An analysis of an optoelectronic coupled oscillator system revealed the potential for a significant speedup over digital computing algorithms when the number of oscillators (nodes) is large enough [13]. Scaling up the optoelectronic oscillator Ising machine, however, remains challenging due in part to its high complexity and costly setup [5, 6, 13].

However, an all-electronic oscillator concept initially proposed by Wang and Roychowdhury introduces the idea of creating a similar system using readily available electronic components interconnected in a parallel fashion and is particularly well suited for chip-scale integration and scaling using present day technologies [2, 9].

Sync Computing and MIT Lincoln Laboratory (MIT-LL) built on this initial work and demonstrated a 4-node, fully-connected, differential LC (inductor-capacitor) oscillator-based analog circuit with standard electronic components which accurately maps to the Ising model. To the best of our knowledge, this is the first demonstration of an all-electronic oscillator-based Ising machine with multi-bit weights [2]. In [2], Chou et. al detail a statistical analysis that provides insight into the viability of these systems as computing platforms when scaled to larger node

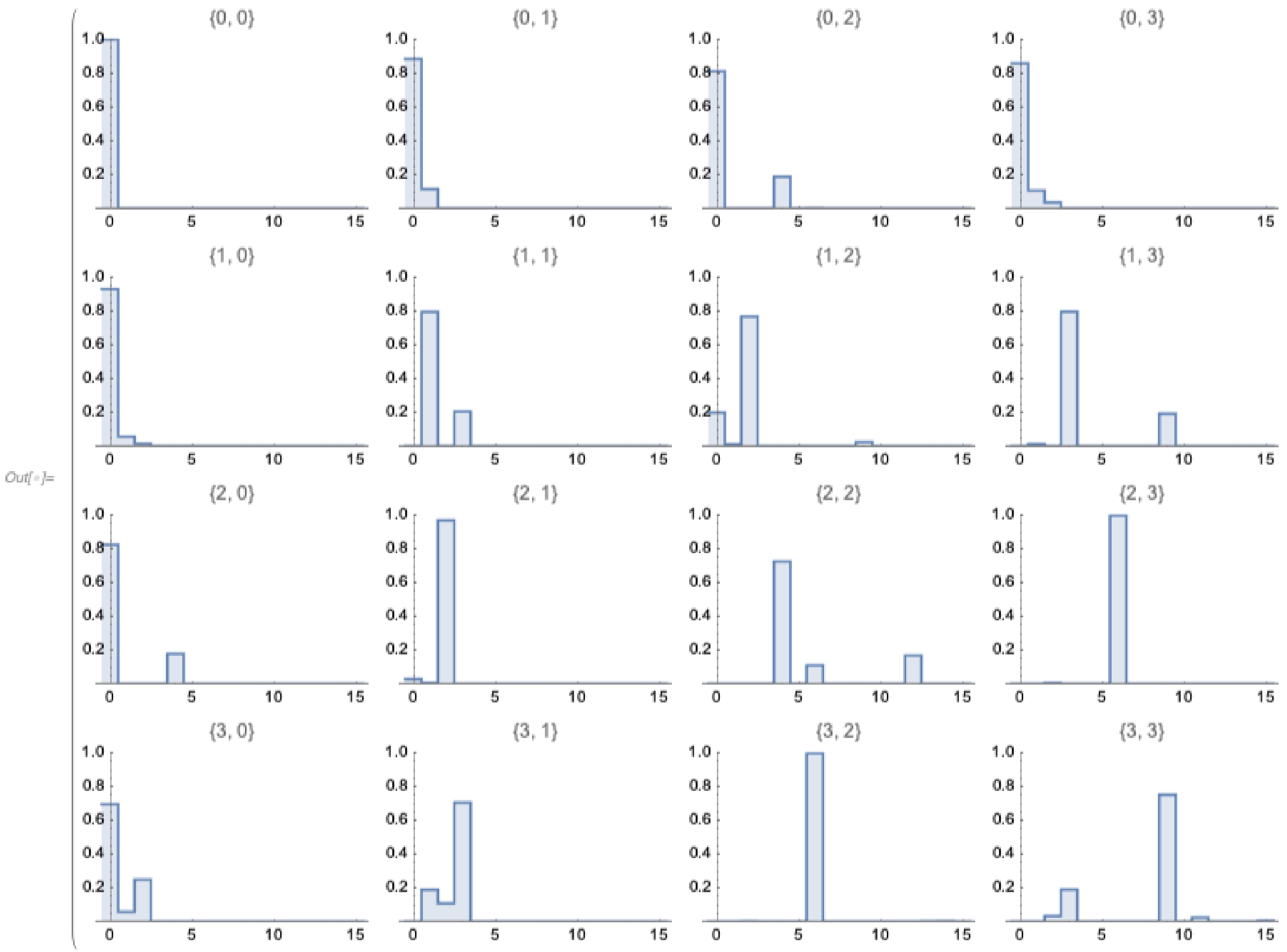


FIGURE 4. This illustrates the nondeterminism inherent in an Ising machine. The horizontal axis is possible answers and the vertical axis is the probability of a getting a certain answer. These are plots of the probabilities of getting different solutions to 2-bit multiplication problems. For example, the upper right corner represents 0 x 3 as denoted by the heading {0, 3}. We see here that we get the answer 0 most of the time, but that the answer is wrong sometimes. If we were to increase the noise term, *β*, for this calculation, we would get a wrong answer even more often.

counts. Figure 5 shows a circuit diagram of the LC oscillator circuit that employs a differential injection-locked frequency divider, the oscillators arranged in a cross-bar array, and the full breadboard system [2, 15]. Currently, Sync Computing is building a 16-node system. Figure 6 shows a photo of the printed circuit board.

Simulation—coupled oscillator system

A well-known benchmark optimization problem is the MAX-CUT problem from graph theory. Following an example in [2], we discuss a small MAX-CUT problem below using a simulation by MIT-LL of the coupled oscillator system.

Given an undirected graph, the MAX-CUT problem consists of finding a partition of that graph into two sets so that the number of edges between the two sets is as large as it can be. It has been shown previously that these graphs can be represented by a network of coupled nonlinear oscillators whose phase dynamics are described by the Kuramoto model, and that this model maps directly to the Ising Hamiltonian if the phases of these oscillators take values of either 0 or 180 degrees. As such, the Kuramoto model is the basis for the MIT-LL simulation [2, 9, 14].

An example of a 4-node (4-spin) system is shown in figure 7a (on page 24). This can also be thought of as a graph with four vertices such that every vertex is

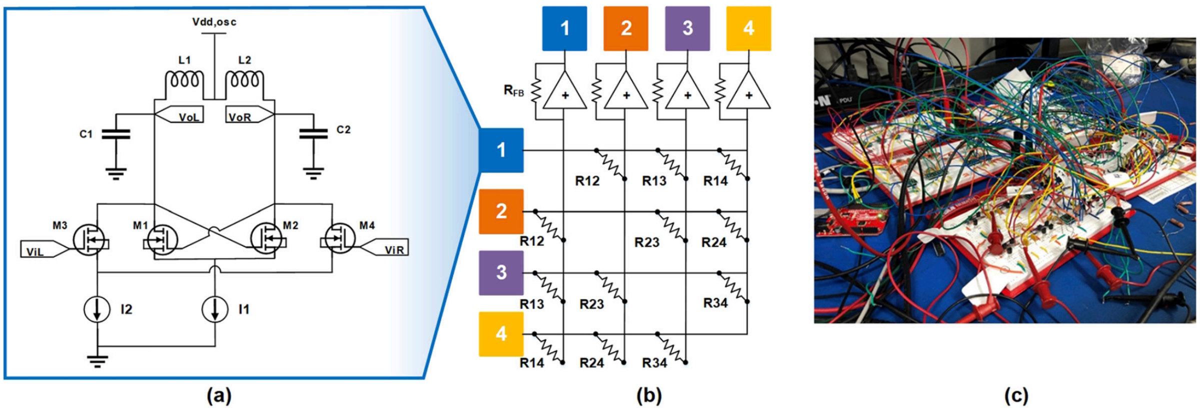


FIGURE 5. (a) This circuit diagram depicts the LC (inductor-capacitor) oscillator circuit. (b) In this diagram, the oscillators are arranged in a cross-bar array. (c) This photo shows the full breadboard system [2]. [Photo credit: MIT Lincoln Laboratory.]

connected to every other vertex. If we let J = 1 for all connections and h = 0, then the Ising problem, in this case, has six solutions shown in figure 7b. One of these solutions is shown in terms of phase in figure 7c. Here, the four spins were intentionally configured to an incorrect solution state and they settled at one of the six correct solution states as expected. Additionally, the system settles to the ground state within three oscillation cycles in this example. Figure 7d shows the results of running the simulation 1,000 times with random initial configurations. We see that the system settles to a correct solution state fairly uniformly [2].

The simulator and the inverse Ising problem

We (i.e., the LPS Resilience and Probabilistic Computing team) used the MIT-LL simulator to validate the results we obtained from solving the inverse Ising problem. We obtained weights from solving the inverse Ising problem, and we used those weights to tune the simulator. The results of using the simulator look similar to figure 4 where we computed Boltzmann probabilities and plotted the results. We found that using the simulator to validate the weights

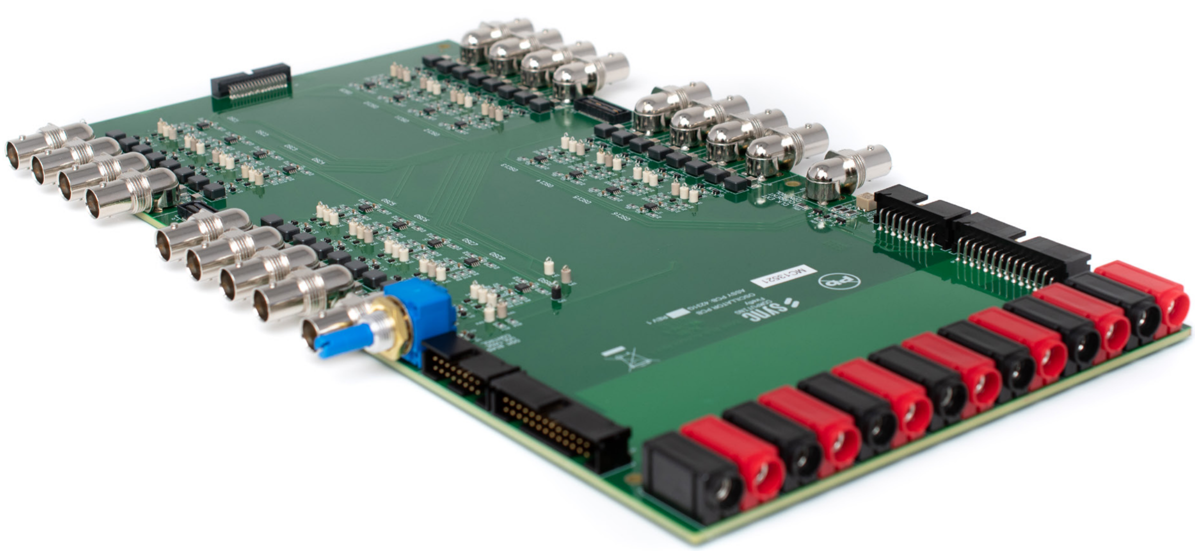


FIGURE 6. Photo of the printed circuit board of the 16-node system. [Photo credit: Sync Computing.]

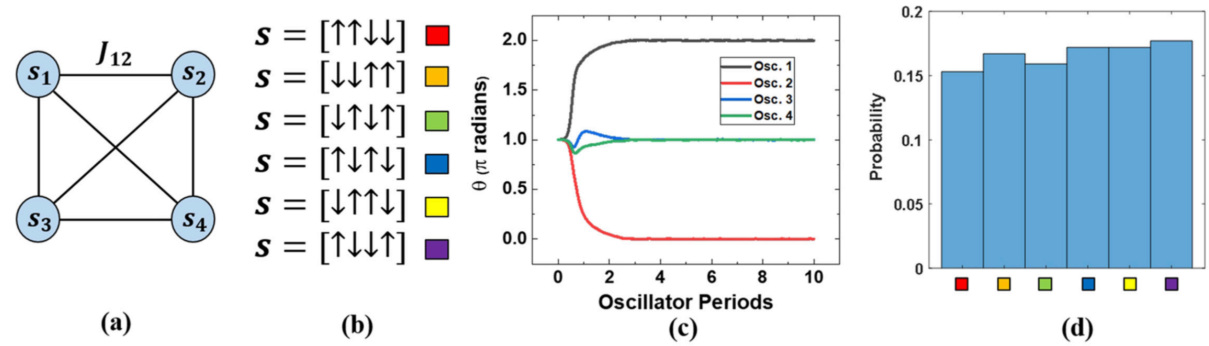


FIGURE 7. (a) This diagram depicts a fully connected 4-node system, for which (b) is the solution set where an up arrow is a positive spin and a down arrow is a negative spin. The graph in (c) shows the first solution in terms of phases, and (d) is a histogram of the results of running the simulation 1,000 times [2]. [Photo credit: MIT Lincoln Laboratory.]

was a step up from computing the Boltzmann probabilities. Our next step is to validate our results using hardware in place of the simulation.

Looking ahead

While quantum computers, like D-Wave, have the potential to solve these NP-hard combinatorial optimization problems, scaling up the number of quantum bits in these systems remains a great challenge. On the other hand, it is possible to build a probabilistic computer out of standard electronic components, as demonstrated by Sync Computing [2]. This allows for faster and more cost-effective scaling. It is reasonable to believe that this all-electronic Ising machine is scalable from four nodes to hundreds of nodes within just a few years. A limiting factor of such a machine is indeed physical space. For the more we scale up, the more oscillator circuits we must add. This does suggest that in order to build an Ising machine for practical use, we will want to explore additional technologies. However, an Ising machine with hundreds of nodes is enough to validate nontrivial results we obtain mathematically that are too large to validate in simulation. While thousands of nodes are necessary for practical use, this system is a step in the right direction and shows promise for a future that includes probabilistic computers.

References

[1] Ghose S, Boroumand A, Kim JS, Gomez-Luna J, Mutlu O. "Processing-in-memory: A workload-driven perspective." *IBM Journal of Research and Development.* 2019;63(6). Available at: https://doi.org/10.1147/JRD.2019.2934048.

[2] Chou J, Bramhavar S, Ghosh S, Herzog W. "Analog coupled oscillator based weighted Ising machine." *Scientific Reports.* 2019;9(14786). Available at: https://doi.org/10.1038/s41598-019-49699-5.

[3] Guo X, Merrikh-Bayat F, Gao L, Hoskins BD, Alibart F, Linares-Barranco B, Theogarajan L, Teuscher C, Strukov DB. "Modeling and experimental demonstration of a Hopfield network analog-to-digital converter with hybrid CMOS/memristor circuits." *Frontiers in Neuroscience.* 2015;9. Available at: https://doi.org/10.3389/fnins.2015.00488.

[4] Inagaki T, Haribara Y, Igarashi K, Sonobe T, Tamate S, Honjo T, Marandi A, Mcmahon PL, Umeki T, Enbutsu K, Tadanaga O, Takenouchi H, Aihara K, Kawarabayashi K, Inoue K, Utsunomiya S, Takesue H. "A coherent Ising machine for 2,000-node optimization problems." *Science.* 2016;354(6312):603–606. Available at: https://doi.org/10.1126/science.aah4243.

[5] Mcmahon PL, Marandi A, Haribara Y, Hamerly R, Langrock C, Tamate S, Inagaki T, Takesue H, Utsunomiya S, Yamamoto Y. "A fully programmable 100-spin coherent Ising machine with all-to-all connections." *Science.* 2016;354(6312):614–617. Available at: https://doi.org/10.1126/science.aah5178.

[6] Hamerly R, Inagaki T, Mcmahon PL, Venturelli D, Marandi A, Onodera T, Ng E, Langrock C, Inaba K, Honjo T, Enbutsu K, Umeki T, Kasahara R, Utsunomiya S, Kako S, Kawarabyashi K, Byer RL, Fejer MM, Mabuchi H, Englund D, Rieffel E, Takesue H, Yamamoto Y. "Experimental investigation of performance differences between coherent Ising machines and a quantum annealer." *Science Advances.* 2019;5(5). Available at: https://doi.org/10.1126/sciadv.aau0823.

[7] Shin JH, Jeong YJ, Zidan MA, Wang Q, Lu WD. "Hardware acceleration of simulated annealing of spin glass by RRAM crossbar array." In: *2018 IEEE International Electron Devices Meeting (IEDM);* 2018 Dec 1–5; San Francisco, CA. Available at: https://doi.org/10.1109/IEDM.2018.8614698.

[8] Cai F, Kumar S, Vaerenbergh TV, Liu R, Li C, Yu S, Xia Q, Yang JJ, Beausoleil R, Lu W, Strachan JP. "Harnessing intrinsic noise in memristor Hopfield neural networks for combinatorial optimization." 2019. Cornell University Library. Available at: https://arxiv.org/abs/1903.11194.

[9] Wang T, Roychowdhury J. "OIM: Oscillator-based Ising machines for solving combinatorial optimisation problems." 2019. Cornell University Library. Available at: https://arxiv.org/abs/1903.07163.

[10] Parihar A, Shukla N, Jerry M, Datta S, Raychowdhury A. "Computing with dynamical systems based on insulator-metal-transition oscillators." *Nanophotonics.* 2017;6:601–611. Available at: https://doi.org/10.1515/nanoph-2016-0144.

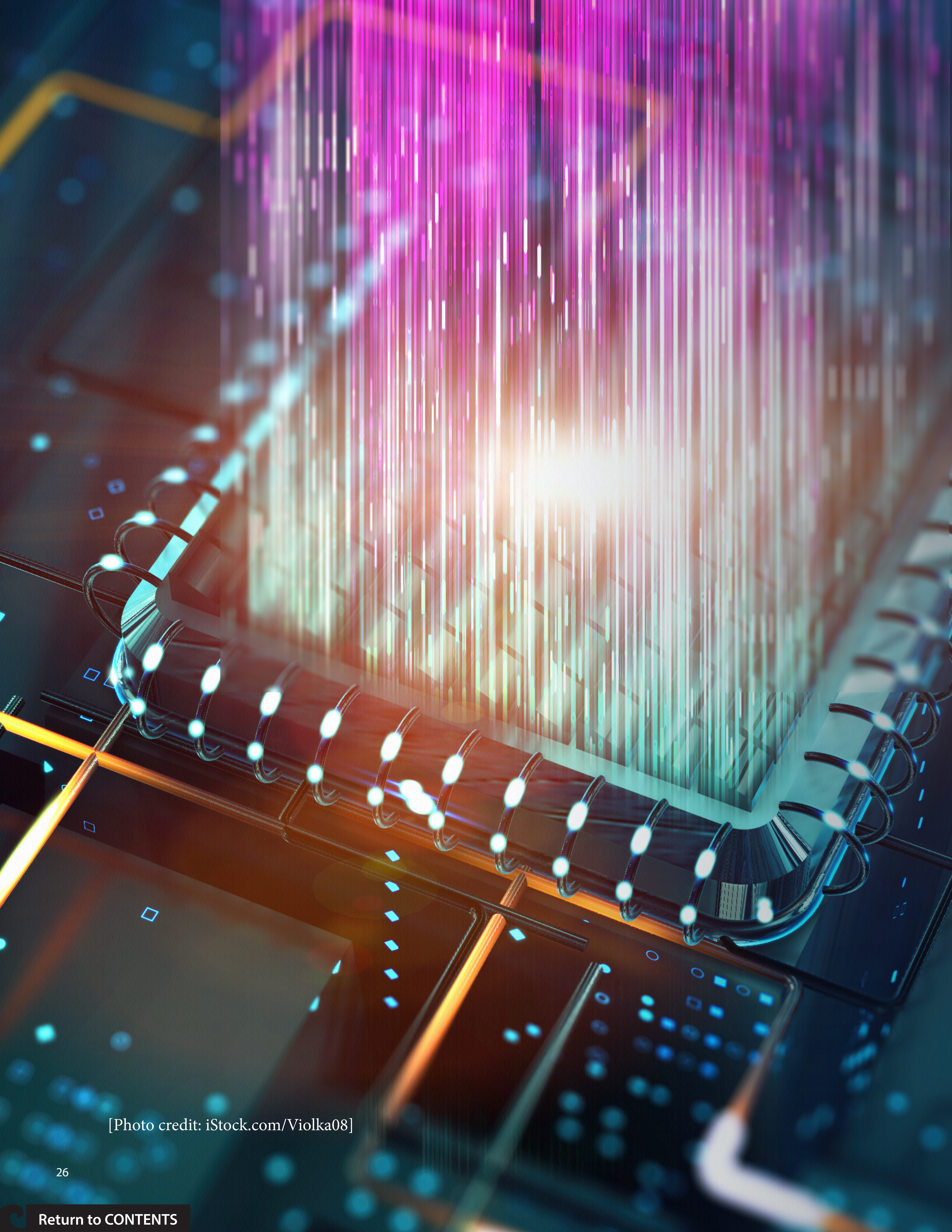
[11] King AD, Bernoudy W, King J, Berkley AJ, Lanting T. "Emulating the coherent Ising machine with a mean-field algorithm." 2018. Cornell University Library. Available at: https://arxiv.org/abs/1806.08422.

[12] Tiunov ES, Ulanov AE, Lvovsky AI. "Annealing by simulating the coherent Ising machine." *Optics Express.* 2019;27(7):10288–10295. Available at: https://doi.org/10.1364/OE.27.010288.

[13] Haribara Y, Utsunomiya S, Yamamoto Y. "A coherent Ising machine for MAX-CUT problems: Performance evaluation against semidefinite programming and simulated annealing." In: Yamamoto Y, Semba K, editors. *Principles and Methods of Quantum Information Technologies.* Japan: Springer; 2016. Pp. 251–262. Available at: https://doi.org/10.1007/978-4-431-55756-2_12.

[14] Acebrón JA, Bonilla LL, PérezVicente CJ, Ritort F, Spigler R. "The Kuramoto model: A simple paradigm for synchronization phenomena." *Reviews of Modern Physics.* 2005;77(1):137–185. Available at: https://doi.org/10.1103/RevModPhys.77.137.

[15] Rategh HR, Lee TH. "Superharmonic injection-locked frequency dividers." *IEEE Journal of Solid-State Circuits.* 1999;34(6):813–821. Available at: https://doi.org/10.1109/4.766815.



[Photo credit: iStock.com/Violka08]

Return to CONTENTS