Architectural Considerations for Scalable Analog Ising Machines

Pranav O. Mathews, Jennifer O. Hasler
School of Electrical and Computer Engineering
Georgia Instititute of Technology
Atlanta, USA
Email: {pmathews30,ph67}@gatech.edu

***Abstract*—Analog circuits can be used to efficiently implement Ising machines, a class of recurrent neural networks that gives good solutions to NP-hard problems. However, designers are faced with numerous choices when deciding how to implement the activation function, weighted connections, and network architecture of an analog Ising machine. This paper explores the tradeoffs between different Ising machine architectural choices (the activation function, weighted connections, and network architecture) and discusses the best combination of choices to solve different NP-hard problems.**

***Index Terms*—Hopfield, Ising, NP-Hard, Compute-in-Memory**

I. Analog Ising Machines

Combinatorial optimization problems, especially those of the NP-complete class, are at the core of many useful problems such as route planning and VLSI placement [1]. Digital computers can effectively verify answers to these problems but reliably finding an optimal solution is not possible. Therefore conventional solvers are forced into approximation algorithms or monte-carlo methods to generate solutions ([2]–[4]).

Analog Ising machines have emerged as a promising alternative method of solving combinatorial problems ([5]–[12]). The dynamics of an Ising machine perform energy minimization, akin to gradient descent, over an energy surface described as

$$H = -\frac{1}{2}\sum_{i}^{N}\sum_{j}^{N} W_{ij}x_i x_j - \sum_{k}^{N} I_k x_i, \tag{1}$$

where W_{ij} describes a weight matrix, x_i represents the state of node i, and I_k is an external bias [13]. This energy surface also describes the NP-hard Quadratic Unconstrained Binary Optimization (QUBO) problem, and because Ising machines can find energy minima of this surface they can also find good solutions to QUBO. Because all NP-complete problems can be reduced to an NP-hard problem, techniques that find good solutions to QUBO can be applied to other useful problems. Therefore, Ising machines can be applied to all NP-complete problems as an efficient analog solver [1].

An Ising machine is physically constructed as an all-to-all graph, where each node is connected to every node in the network. Each node has a state, which is sent over an edge to all other nodes in the network. The edges modify the state with a weight, which can be positive or negative. Each node aggregates the weighted connections from every other node and modifies its state accordingly. This process repeats until the network settles in a stable state, representing an energy minimum or good solution to an NP-complete problem.

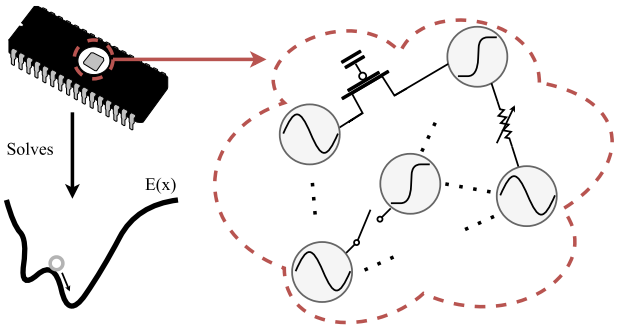


Fig. 1. Chips that contain analog Ising machines can effectively perform gradient descent to solve QUBO. When designing these circuits there are a plethora of options to choose from: what is inside a node, how nodes are connected, and what connects nodes. These architectural decisions affect performance, and understanding the tradeoffs inherent in each choice is crucial to creating an efficient system.

Analog circuits can be used to efficiently construct Ising machines, and there are various ways to create the nodes and edges of the Ising graph (Figure 1). This paper explores tradeoffs in the main architectural elements of an analog Ising machine: the activation function (Section II), the weighted connections (Section III), and the overall network architecture (Section IV). These tradeoffs are then applied to different NP-hard problems (Section V) followed by a final discussion (Section VI).

II. Activation Functions

The activation functions of analog Ising machines are typically interpreted as a binary signal with a lower state of 0 or -1 and an upper state of 1. However, the physical mechanism behind the binary state could be an oscillatory phase (Section II-A) or sigmoidal voltage (Section II-B). Physical differences between the two results in practical effects on scaling Ising machines to large problems.

978-8-3315-4127-9/24/$31.00 @2024 IEEE

		Sigmoid		Oscillator	
Size	Problem	Convergence (µs)	Energy (pJ)	Convergence (µs)	Energy (pJ)
3	MaxCut 2E	48.38	0.09	398.56	387.61
	MaxCut 3E	0.10	0.09	538.57	530.6
	AM 001	55.60	0.18	428.88	422.48
	AM 111	18.07	0.07	226.67	223.29
4	MaxCut 3E	136.48	1.52	329.61	326.48
	MaxCut 4E	97.21	0.49	436.04	440.47
	MaxCut 5E	144.22	1.84	167.96	171.03
	MaxCut 6E	81.84	0.35	449.83	465.79
	AM 0001	114.66	5.04	187.48	194.20
	AM 0011	164.55	4.52	211.35	218.90
	AM 1111	81.53	4.64	108.64	112.59
	Averages	91.79	1.71	316.69	317.59

Fig. 2. Averaged, approximate energy consumption and convergence time of an Ising machine with sigmoidal and oscillatory activation functions solving multiple max-cut and associative memory problems over different initial conditions ([14]). It can be seen that the sigmoidal activation function consistently outperforms the oscillator in energy and convergence time for the two problems. These results suggest that sidmoidal activation functions may be preferred to oscillatory ones for efficient Ising machines.

A. Oscillators

Ising machines with oscillatory activation functions, often called Ising networks, represent a binary state in a phase difference between nodes. A phase difference of 0° corresponds to a logical -1 while 180° represents a logical 1. In order for two oscillators to properly couple into binary phases, an external subharmonic injection locking (SHIL) signal is needed.

Many analog circuits can generate an oscillatory waveform, leading to many Ising machines that use an oscillatory activation function ([7], [10], [15]–[18]). The choice of oscillator has included discrete LC oscillators ([7]), ring oscillators ([18]), and nano-oscillators ([10]).

It was stated that activation functions represent binary states, however it has been shown that Ising machines with higher order coupling can better represent and solve certain NP-hard problems ([19]). Unfortunately, to the authors knowledge there has not been a physical circuit implementation shown that demonstrates such behavior.

B. Sigmoid

Ising machines with sigmoidal activation functions, more commonly referred to as Hopfield networks, represent a binary state in a voltage level. A level of 0 corresponds to a logical 0 while a level of V_{dd} corresponds to a logical 1.

Sigmoidal functions can be approximated by any amplifier. A common choice is an operational amplifier ([13]), but transconductance amplifiers can also be used ([9], [20]). It has been shown that the sigmoid needs to have high gain to properly separate local minima in the energy landscape ([21]).

C. Hopfield Network vs Ising Network

An experimental comparison between sigmoids (Hopfield Network) and oscillations (Ising Network) was performed on a Field Programmable Analog Array (FPAA, [22]), where it was found that the sigmoid outperformed the oscillator in both power and speed while solving the Associative Memory and Max-Cut problems. (Figure 2) ([14]). Intuitively this result makes sense as oscillators constantly consume dynamic switching power, which is less efficient than the sigmoids static voltage levels. These results suggest that for any binary Ising machine implementation it is more efficient to use a sigmoidal activation function over an oscillatory one.

Problem	Nodes	Weights
0-1 Integer Programming	N	Binary
Feedback Node Set	N^2	Binary
Undirected Hamiltonian Cycle	N^2	Precise
Chromatic Number	$n * N$	Binary
Clique Cover	$n * N$	Binary
Job Sequencing	$m * N$	Precise

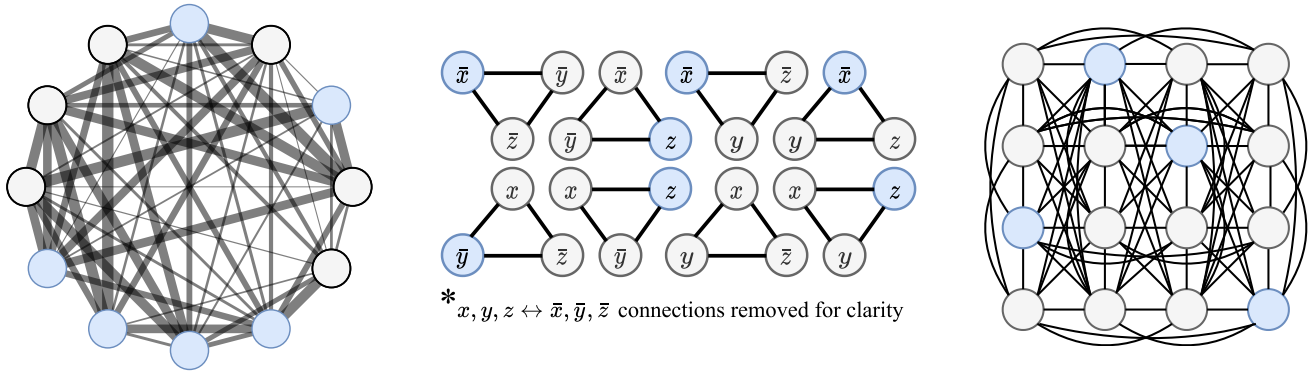
Fig. 3. Scaling for different NP-complete problems. This set of problems can be reduced to another NP-complete problem with linear scaling, making it a good representation of the entire spectrum [23]. The worst scaling problem is the Undirected Hamiltonian Cycle, which can be reduced to the famous TSP.

III. Weighted Connections

Weighted connections can be locally stored with a non-volatile memory element doubling as an analog multiplier, a technique referred to as Compute-In-Memory (CiM) and often used to create the weighted connections and summation necessary between activation functions in Ising machines. However, the weight precision depends on the problem being solved (Figure 3). Some problems such as the Boolean Satisfiability Problem (3SAT) only require binary weights (Section III-A). Others, such as the Travelling Salesman Problem (TSP) require a much higher precision (III-B).

A. Binary Weights

Binary weighted connections can be made with an analog switch. This switch can simply take the form of a transmission gate ([18]). However, local memory must be present to effectively encode the binary weight into the system which takes additional area.



(a) (b) (c)

Fig. 4. Ising machine graphs for different NP-hard problems with example solutions (blue = state 1, gray = state 0). (a) Weighted Max-Cut, a dense graph with high weight precision; the width of the connections indicates the weight. (b) Boolean Satisfiability, a sparse graph with low weight precision; each connection is a weight of -1. (c) Travelling Salesman Problem, a dense graph with medium weight precision.

Any higher precision weight can also function as a binary weight. Therefore, for Ising machines that encode problems of different precision, a more precise weight is ideal to cover all possible scenarios. However, application-specific machines, for example ones that only solve 3SAT, can operate with purely binary connections.

B. High Precision Weights

When used with oscillatory activation functions, transmission gate switches have been combined to form multi-level coupling strengths ([18]). While effective, this does require multiple switches and would not scale to higher bit precision.

Many non-volatile memories (NVMs) have been shown to perform weighed vector-matrix-multiplication in CiM systems. The same technology can be used in Ising machines to connect activation functions together, and has been demonstrated with memristors (RRAM) and Floating-Gates (FG) ([6], [20]). In general, FGs have shown higher bit-precision and lower power than RRAM, suggesting that they are the better choice for high-precision weights in Ising machines ([24]–[26]). Emerging memories like FeFETs and Spintronic devices are also canidates for weights.

IV. Network Architecture

The number of possible connections in an Ising machine grows exponentially with size: each new node adds $N-1$ bi-directional connections. Full usage of every connection requires an all-to-all architecture (IV-A). However some NP-hard problems contain many zero connections (sparse) between nodes and don't require a full all-to-all connection scheme (Figure 4). Networks can take advantage of this sparsity and prune nodes, lessening routing requirements when building the chip (IV-B,C).

A. Sparse Mesh Architectures

Sparse mesh architectures assume a high level of sparsity, which translates to each node having a low degree. By pruning most connections in the network one can make easily tileable nodes that can scale to extremely large networks. Examples include the Lattice (Figure 5a) and King's Graph (Figure 5b) architecture, where each node has a degree of two or three respectively (Figure 5).

Most practical problems are not sparse enough to only have degrees below three. However, this does not mean sparse mesh architectures are not useful. By utilizing mirror nodes, or nodes that simply copy their state, a higher degree can be achieved by trading off neuron counts. This allows for efficient embeddings of sparse problems onto sparse mesh architectures even without the problem node degree being below three [18].

B. Manhattan Architecture

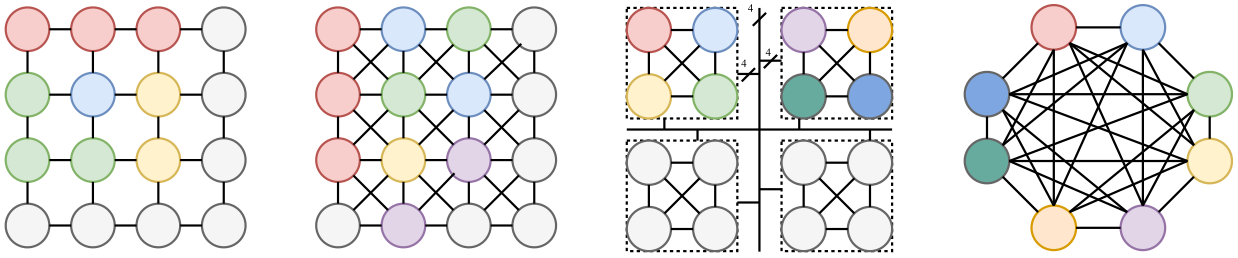
While sparse mesh architectures efficiently encode sparse problems, they exacerbate inefficiencies for problems of medium density or higher because of the overhead required for mirror nodes. Even small pockets of dense connections can blow up the number of nodes required to encode a problem. The Manhattan architecture, common in Field Programmable Gate Array (FPGA) and FPAA toplogies, has been shown to efficiently encode sparse and dense problems ([20], [27]).

Nodes in a Manhattan architecture reside inside Computational Analog Blocks (CABs) (Figure 5c) Each CAB is essentially a small all-to-all network that is then sparsely connected to other CABs by utilizing external routing lines from connection (C) and switch (S) blocks. This connection scheme retains sparsity across the system but alleviates the effects of density through local connections.

C. All-to-All Architecture

The all-to-all architecture can encode up to the maximum density of Ising problem. However, for sparse problems there is a high degree of unused processing elements, which is an inefficient use of area. In addition, the routing complexity of an all-to-all topology is immense, placing practical limits on circuit creation that may not scale to large problems.

Efficient system design, such as a hybrid network, may keep network sizes relatively small and allow for all-to-all architectures to be used for large problems. Breaking a QUBO problem into sub-groups and iteratively solving on



(a) (b) (c) (d)

Fig. 5. Different Ising machine architectures and an example complete graph embedding. (a) Lattice Architecture, effective for extremely sparse problems. (b) Kings Graph Architecture, effective for sparse problems. (c) Manhattan Architecture, made for sparse-medium density problems. (d) Complete architecture, can support problems of any density.

a smaller all-to-all Ising machine has been shown to still increase solution efficiency ([28]).

V. Architectures for NP-Hard Problems

We have discussed different architectural considerations when designing Ising machines. In this section we will apply these principles to the design of Ising machine architectures for different NP-hard problems that have unique scaling requirements (Figure 4), spanning the gambit on possible connectivity/weight density requirements. The Max-Cut problem (Section V-A) is of medium density and ranges from binary weights to high precision, the 3SAT problem (Section V-B) is sparse with binary weights, and the TSP problem is dense with high precision weights (Section V-C).

A. Max-Cut

Max-Cut is an NP-complete problem that asks for a graph to be partitioned into two sets such that the number of edges between the sets is maximum [29]. A variant on this problem weights the edges and asks for the maximum weight between the two sets. This problem can be expressed in QUBO form as

$$H = H_A + H_B = A\left(\sum_{i=1}^{N} s_i\right)^2 + B\sum_{(uv)\in E}\frac{1 - s_u s_v}{2} \quad (2)$$

where H_A constrains the partition split to be even, and H_B constrains the cutsize [1].

If the Max-Cut problem is binary, any connection type can be used. To keep it simple we will choose a switch. Because sigmoids are generally more efficient than oscillations, we will also use a sigmoidal activation function. Finally, a Manhattan architecture will be chosen because of the problem's medium density, taking advantage of the limited sparsity.

A weighted max-cut problem would demand a NVM connection with precision on the scale of the weights. Depending on the problem, either RRAM or FGs could work; however, the higher precision of FGs makes them a more viable solution.

B. 3SAT

3SAT, or boolean satisfiability, is an NP-complete problem that asks if a Boolean expression will be true for at least one combination of its input variables. The 3SAT problem asks this for a boolean expression in conjunctive normal form, where each clause has three variables. This problem can be reduced to the Maximum Independant Set problem expressed in standard QUBO form (Equation 1) where W_{ij} is -1 for vertices that are connected by an edge and 0 otherwise [1].

The resulting weight matrix is sparse and would suggest a sparse mesh such as the Kings Graph or Lattice, and indeed 3SAT has been shown on such architectures ([12]). However, larger problems have proven to be more easily mapped to denser architectures such as Manhattan ([20], [27], [30]) and all-to-all ([28]). Therefore we will choose the Manhattan architecture for this problem and low precision switches for the connections.

Typically sigmoids would be a better choice than oscillations, and that is true for this problem as well. However, if multi-phase Ising networks take practical form ([19]) then they would lend computational advantages to 3SAT and therefore would be the preferred choice.

C. TSP

The Travelling Salesman Problem (TSP) is a famous NP-complete problem that asks for the shortest path to traverse N cities and return to the starting city, visiting each location only once. A solution to this problem looks like an ordered pair of citie and can be represented in a table, therefore necesitating N^2 nodes.

	1	2	3
A	0	0	1
B	1	0	0
C	0	1	0,

(3)

therefore necesitating N^2 nodes. The QUBO formulation of this problem enforces the table representation (only a single 1 in each row and column) and also encodes the distances between each city [21].

TSP is a dense, high precision problem. It also has an $O(N^2)$ scaling, meaning it generally requires a large number of nodes. The combination of density with high neuron count suggests an all-to-all network configuration and the high precision necessitating FG connections. Once again, sigmoids would be preferred.

Unlike the other NP-hard problems discussed, the connections for TSP do not change from problem-to-problem

(assuming same size). Most of the connections enforce the structure of the $N \times N$ structure, and the only connections that are problem-specific are between each column of nodes. Therefore, TSP may most benefit from none of the architectures discussed in this paper and instead could have the most efficiency from a custom architecture that is slightly less dense than all-to-all.

VI. Conclusion

In this paper, we have discussed different architectural considerations for analog Ising machines: the activation function (sigmoids vs oscillations), the connections (switch vs RRAM vs FG), and the architecture (sparse mesh vs manhattan vs all-to-all). It was found that sigmoids are generally superior to oscillations, FG devices cover the widest range of precision, and the Manhattan architecture strikes a good balance between extremely sparse and all-to-all architectures.

From the architectural evaluations and application to NP-hard problems, it would appear that the ideal architecture is a Manhattan architecture with FG connections and a sigmoidal activation function. However, not discussed is specific circuit implementations for the activation function, FG devices, and Manhattan architecture. Variation exists in all of these elements and must be individually optimized to reap the full benefits of the architectural advantage.

Minor embeddings in Manhattan architectures also should be investigated. Mappings of graphs to the sparse mesh architectures has been investigated, but there has been less activity into the Manhattan architecture. Efficient embeddings are essential for practical use.

Ising machines with sigmoidal activation functions, or Hopfield networks, were first used to model firing rates in the brain. Following this history, a spiking, neuromorphic approach may yield further efficiencies when applied to an Ising machine architecture. Further architectural efficiencies can still be made when designing scalable Ising machines.

References

[1] A. Lucas, "Ising formulations of many NP problems," *Frontiers in Physics*, 2014.

[2] K. M. Zick, "Improved Sparse Ising Optimization," Nov. 2023, arXiv:2311.09275 [cond-mat].

[3] V. Paschos, "An overview on polynomial approximation of NP-hard problems," *Yugoslav Journal of Operations Research*, 2009.

[4] R. Howell, "Simulated Annealing on NP-Complete Problems," *ACMS Conference Proceedings 1989*, Jun. 1989.

[5] J. J. Hopfield and D. W. Tank, "Computing with neural circuits: a model," *Science (New York, N.Y.)*, Aug. 1986.

[6] F. Cai, S. Kumar, T. Van Vaerenbergh, X. Sheng, R. Liu, C. Li, Z. Liu, M. Foltin, S. Yu, Q. Xia, J. J. Yang, R. Beausoleil, W. D. Lu, and J. P. Strachan, "Power-efficient combinatorial optimization using intrinsic noise in memristor Hopfield neural networks," *Nature Electronics*, Jul. 2020.

[7] T. Wang and J. Roychowdhury, "OIM: Oscillator-Based Ising Machines for Solving Combinatorial Optimisation Problems," in *Unconventional Computation and Natural Computation*, 2019.

[8] P. O. Mathews, C. B. Duffee, A. Thayil, T. E. Stovall, C. H. Bennett, F. Garcia-Sanchez, M. J. Marinella, J. A. C. Incorvia, N. Hassan, X. Hu, and J. S. Friedman, "High-speed CMOS-free purely spintronic asynchronous recurrent neural network," *APL Machine Learning*, Mar. 2023.

[9] P. O. Mathews and J. O. Hasler, "Physical Computing for Hopfield Networks on a Reconfigurable Analog IC," in *ISCAS*, May 2023.

[10] S. Dutta, A. Khanna, A. S. Assoa, H. Paik, D. G. Schlom, Z. Toroczkai, A. Raychowdhury, and S. Datta, "An Ising Hamiltonian solver based on coupled stochastic phase-transition nano-oscillators," *Nature Electronics*, Jul. 2021.

[11] A. Lu, J. Hur, Y.-C. Luo, H. Li, D. E. Nikonov, I. A. Young, Y.-K. Choi, and S. Yu, "Scalable In-Memory Clustered Annealer With Temporal Noise of Charge Trap Transistor for Large Scale Travelling Salesman Problems," *JETCAS*, 2023.

[12] Y. Su, T. T.-H. Kim, and B. Kim, "FlexSpin: A CMOS Ising Machine With 256 Flexible Spin Processing Elements With 8-b Coefficients for Solving Combinatorial Optimization Problems," *JSSC*, 2024.

[13] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities." *Proceedings of the National Academy of Sciences*, Apr. 1982.

[14] P. O. Mathews and J. O. Hasler, "Hopfield vs Ising: A Comparison on the SoC FPAA," *TCAS I*, 2024.

[15] L. Q. English, A. V. Zampetaki, K. P. Kalinin, N. G. Berloff, and P. G. Kevrekidis, "An Ising machine based on networks of subharmonic electrical resonators," *Communications Physics*, 2022.

[16] J. Chou, S. Bramhavar, S. Ghosh, and W. Herzog, "Analog Coupled Oscillator Based Weighted Ising Machine," *Scientific Reports*, Oct. 2019.

[17] W. Moy, I. Ahmed, P.-w. Chiu, J. Moy, S. S. Sapatnekar, and C. H. Kim, "A 1,968-node coupled ring oscillator circuit for combinatorial optimization problem solving," *Nature Electronics*, 2022.

[18] H. Lo, W. Moy, H. Yu, S. Sapatnekar, and C. H. Kim, "An Ising solver chip based on coupled ring oscillators with a 48-node all-to-all connected array architecture," *Nature Electronics*, 2023.

[19] C. Bybee, D. Kleyko, D. E. Nikonov, A. Khosrowshahi, B. A. Olshausen, and F. T. Sommer, "Efficient optimization with higher-order ising machines," *Nature Communications*, Sep. 2023.

[20] P. O. Mathews and J. O. Hasler, "A Programmable and Reconfigurable CMOS Analog Hopfield Network for NP-hard Problems," *TVLSI*, 2024.

[21] J. J. Hopfield, "Neurons with graded response have collective computational properties like those of two-state neurons." *Proceedings of the National Academy of Sciences*, May 1984.

[22] S. George, S. Kim, S. Shah, J. Hasler, M. Collins, F. Adil, R. Wunderlich, S. Nease, and S. Ramakrishnan, "A Programmable and Configurable Mixed-Mode FPAA SoC," *TVLSI*, Jun. 2016.

[23] J. A. Filar, M. Haythorpe, and R. Taylor, "Linearly-growing reductions of Karp's 21 NP-complete problems," *Numerical Algebra, Control and Optimization*, 2018.

[24] M. Rao, H. Tang, J. Wu, W. Song, M. Zhang, W. Yin, Y. Zhuo, F. Kiani, B. Chen, X. Jiang, H. Liu, H.-Y. Chen, R. Midya, F. Ye, H. Jiang, Z. Wang, M. Wu, M. Hu, H. Wang, Q. Xia, N. Ge, J. Li, and J. J. Yang, "Thousands of conductance levels in memristors integrated on CMOS," *Nature*, 2023.

[25] 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," in *MWSCAS*, Aug. 2023.

[26] S. Kim, J. Hasler, and S. George, "Integrated Floating-Gate Programming Environment for System-Level ICs," *TVLSI*, 2015.

[27] G. H. Hutchinson, E. Sifferman, T. Bhattacharya, D. Kwon, and D. B. Strukov, "FPIA: Field-Programmable Ising Arrays with In-Memory Computing," in *Proceedings of the 29th ACM/IEEE International Symposium on Low Power Electronics and Design*, New York, NY, USA, 2024.

[28] H. Cılasun, Z. Zeng, R. S, A. Kumar, H. Lo, W. Cho, W. Moy, C. H. Kim, U. R. Karpuzcu, and S. S. Sapatnekar, "3SAT on an all-to-all-connected CMOS Ising solver chip," *Scientific Reports*, 2024.

[29] R. Karp, "Reducibility Among Combinatorial Problems," Jan. 1972.

[30] T. Jagielski, R. Manohar, and J. Roychowdhury, "FPIM: Field-Programmable Ising Machines for Solving SAT," Sep. 2023, arXiv:2306.01569 [cs].