Hopfield vs Ising: A Comparison on the SoC FPAA

Pranav O. Mathews, *Graduate Student Member, IEEE*, and Jennifer O. Hasler, *Senior Member, IEEE*

***Abstract*— Physical computing techniques can efficiently solve combinatorial optimization problems and could outperform conventional digital techniques. This paper presents the first direct comparison of two physical quadratic optimization solvers: analog Hopfield and Ising networks. Both networks are built in a large scale Field Programmable Analog Array (FPAA) and two NP-hard problems (max-cut and associative memory) are solved over various initial conditions and graphs. The convergence time, energy, power, and peripheral circuitry of the two networks is then compared where it is found that the Hopfield network outperforms the Ising networks in these test cases.**

***Index Terms*— Hopfield, Ising, FPAA.**

I. Energy Surface Minimization

COMBINATORIAL problems, such as the NP-hard class, have been shown to be intractable on digital machines [1]. Exact algorithms that run in exponential time can be built but practically an approximation algorithm that computes a good, but typically sub-optimal, solution in polynomial time is used [2]. Simulated annealing is an example of a common approximation technique, continuously minimizing a cost function until an optimum is found [3].

Physical computing techniques have solved NP-hard problems more efficiently than digital algorithms and have the potential to solve problems faster as well ([4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15]). Problems can be mapped to a Quadratic Unconstrained Binary Optimization (QUBO) problem and then minimized through physical dynamics; examples include quantum annealing and Ising networks. Much like simulated annealing a cost function (or energy) is minimized until a minimum is reached, but through continuous physical dynamics instead of discrete algorithmic steps [16].

Analog circuits have been used to efficiently implement Ising networks ([4], [6], [8], [12], [14], [15], [17]). These systems have dynamics that minimize the function

$$\mathbf{H}(\mathbf{s}) = -\mathbf{s}^T\mathbf{J}\mathbf{s} - \mathbf{h}^T\mathbf{s}, \tag{1}$$

where $\mathbf{s}$ is a vector describing the spin of each state (± 1 for a binary network), $\mathbf{J}$ is a weight matrix describing the connections between oscillators, and $\mathbf{h}$ is a constant bias. Problems are mapped to the network by choosing $\mathbf{W}$ to create an energy function that when minimized gives the optimum solution; mappings have been shown for each of Karp's 12 NP-Hard problems [18].

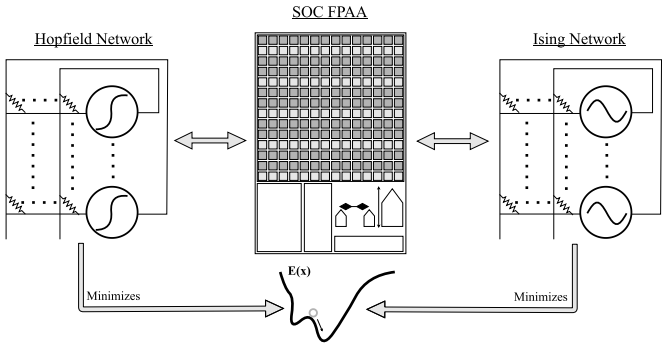


Fig. 1. Both Hopfield and Ising networks dynamically minimize energy surfaces, allowing for good or optimal solutions to otherwise computationally difficult problems. The question of which implementation has an advantage is tested on the SoC FPAA to give a level playing field and fair comparison.

Hopfield networks ([19], [20], [21]) are another structure that minimizes an almost identical energy function as Ising networks,

$$\mathbf{E}(\mathbf{x}) = -\frac{1}{2}\mathbf{x}^T\mathbf{W}\mathbf{x} - \theta^T\mathbf{x}, \tag{2}$$

and is sometimes also referred to as an "Ising machine." We will define Hopfield networks as having a state variable $\mathbf{x}$ that describes the output of a perceptron with a sigmoidal activation function between 0 and 1 in contrast to the Ising oscillatory state. These networks have a long history of analog circuit implementations ([5], [22], [23], [24]).

The similarities between the energy surfaces of Hopfield and Ising networks extends to their physical implementation. Both networks are fully recurrent structures, requiring every node to be connected to every other node in a network. Problem scaling is also the same, with most problems that have N parameters typically requiring $O(N)$ Hopfield/Ising nodes. A few exceptions such as the Travelling Salesman Problem (TSP) require $O(N^2)$ nodes and no NP-hard problem requires more than $O(N^3)$ nodes [18].

The biggest difference between Hopfield and Ising networks lies in their state variable: spins versus sigmoids. Artificial Ising spins are typically created with analog oscillators, while Hopfield's sigmoidal output can be approximated with a sharp tanh. This small difference results in massive practical changes in the peripheral circuitry and electrical signals that each network uses.

Manuscript received 18 March 2024; revised 3 June 2024; accepted 4 June 2024. This article was recommended by Associate Editor J. Goes. *(Corresponding author: Pranav O. Mathews.)*

The authors are with Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: pmathews30@gatech.edu; jennifer.hasler@ece.gatech.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCSI.2024.3411407.

Digital Object Identifier 10.1109/TCSI.2024.3411407

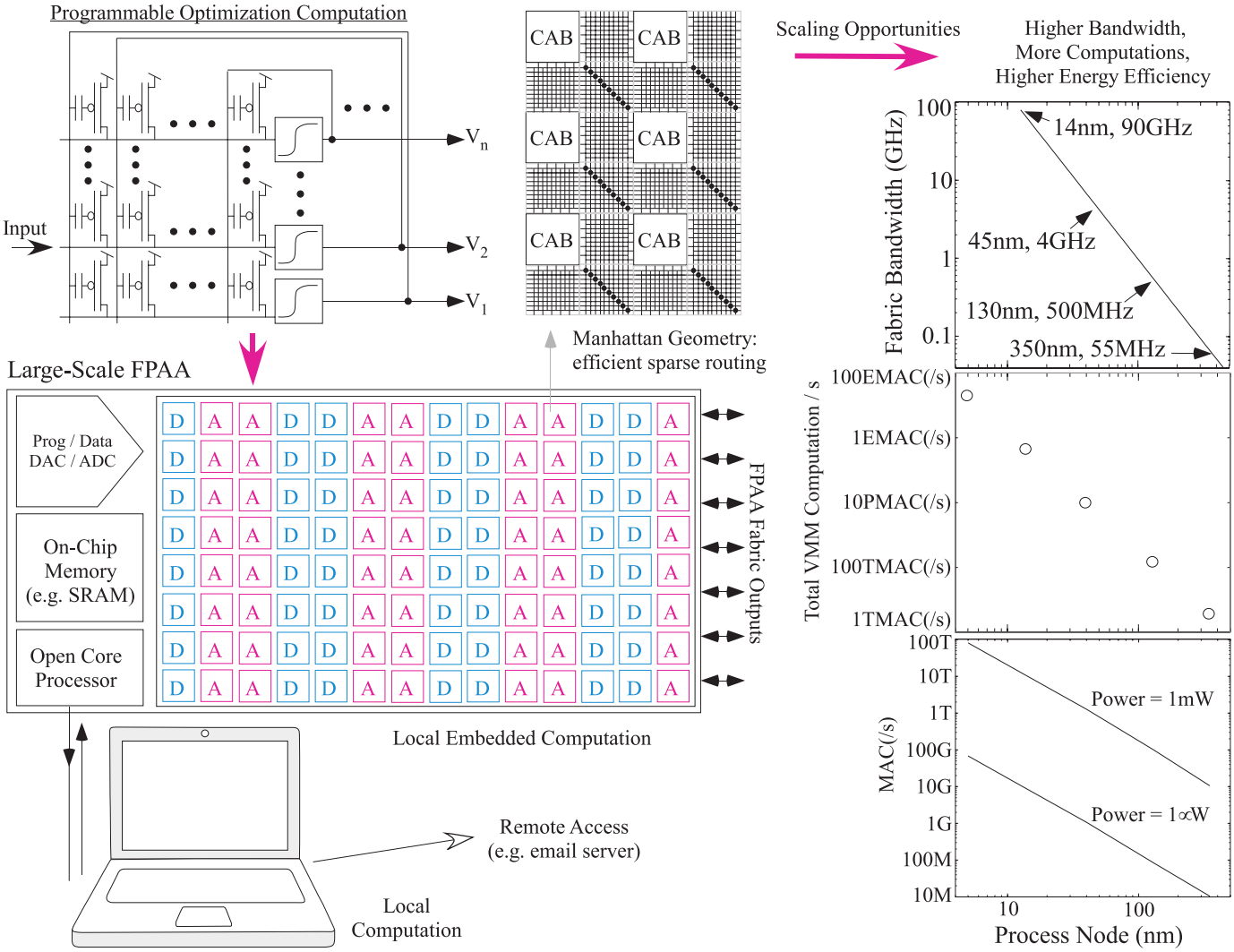


Fig. 2. The SoC FPAA enables users to harness the power of analog computing while offering flexibility in circuit design. External devices interface to an on-chip open-source MSP 430 processor directly through a computer or remotely through an email-based remote system [26]. The controlling system specifies what circuits to program and route together in the FPAAs Manhattan architecture. Each analog block has multiple analog elements that can be wired together and programmed through FG devices. Through this flexibility both Hopfield and Ising networks can be programmed and tested on the exact same chip.

When designing an analog optimization network for a specific application, which choice would be best? This work begins to answer this question, being the first direct comparison of analog Hopfield and Ising networks in the same technology (Fig. 1). Section II presents the comparison platform, the SoC Field Programmable Analog Array (FPAA), [25], and the analog circuits that were used to create both Hopfield and Ising networks; the Hopfield network extends [22] and the Ising network is the first demonstrated implementation on the FPAA. Section IV shows two NP-hard case studies and their mapping to Hopfield/Ising networks, describing the resulting networks structure and showing sample temporal outputs. The performance of both networks on these case studies are evaluated in Section IV, contrasting solution time and energy. Finally the comparison results are discussed along with concluding thoughts (Section V).

II. Implementation on the Soc FPAA

A fair comparison between networks requires equitable devices and conditions of operation; comparing different technologies such as a 1 GHz nanoscillator with a 1 μm CMOS oscillator is hardly fair. To facilitate a just comparison a reconfigurable analog computing platform (SoC FPAA) that can implement both Hopfield and Ising networks on the same 350nm CMOS substrate is used [25]. The architecture of the SoC FPAA is discussed (Section II-A) along with the in-routing source coupled Vector-Matrix-Multiplier (VMM) that facilitates inter-node connections in both the Hopfield and Ising networks (Section II-B). Both implementations of the Hopfield (Section II-C) and Ising (Section II-D) networks are explained, followed by a brief discussion of device mismatch and calibration (Section II-E).

A. The Soc FPAA

The SoC FPAA is a mixed-signal, analog computing platform which includes analog elements and signals integrated with potential logic and mixed-signal enabled routing (Fig. 2). A Manhattan routing scheme connects Computational Analog Blocks (CABs) together; each CAB has different programmable analog components such as Transconductance Amplifiers's (TAs), nFETs, pFETs, and current mirrors that

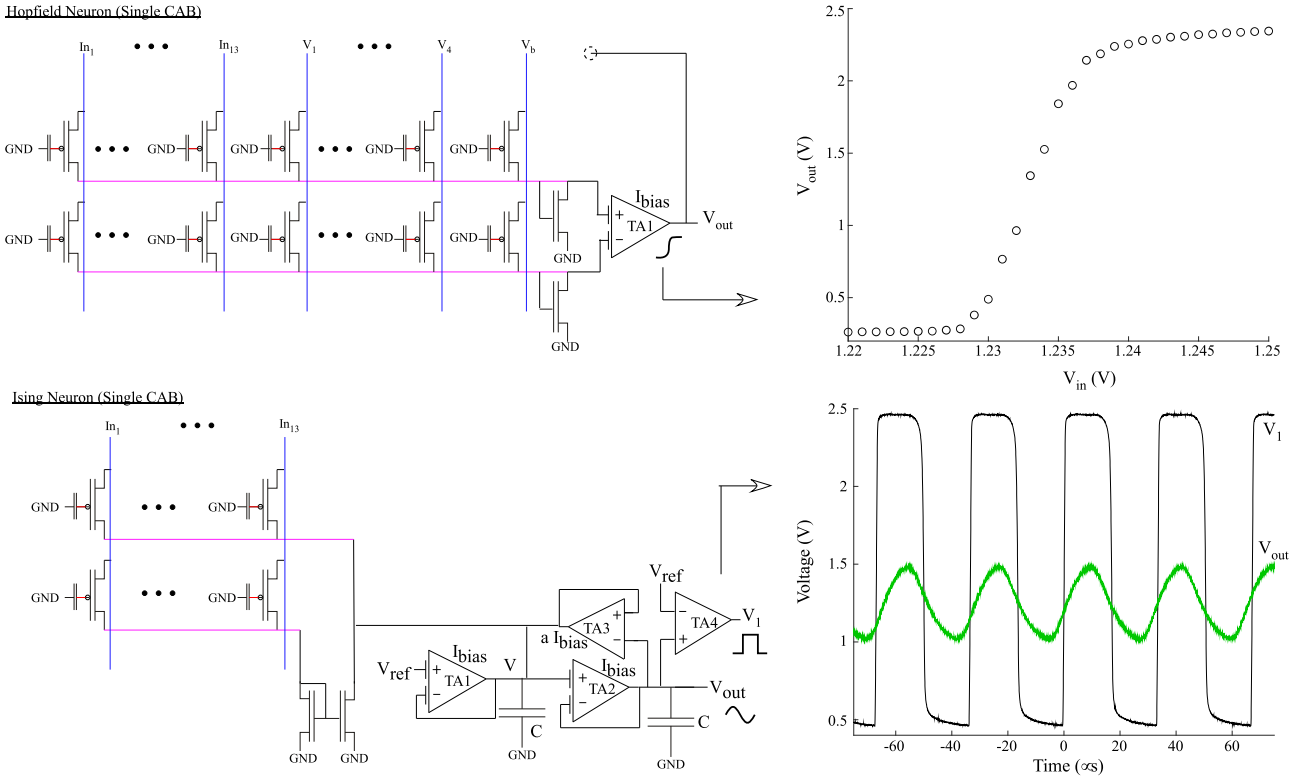


Fig. 3. Circuit diagrams of a single neuron in the Hopfield and Ising networks along with their corresponding outputs. Both neurons fit in one CAB of the FPAA, but the Ising neuron takes up the entire CAB while four Hopfield neurons can fit inside one CAB. The Hopfield network uses a single and two VMM rows. The Ising network uses a MeadSOS oscillator made of three TAs, two rows of a VMM with an Nmirror, and an external buffer to rail the signal. The Manhattan routing scheme allows for recurrent connections both inside and between CABs.

can be routed internally and between CABs. Floating-Gate (FG) circuits are the main programmable element on the FPAA, with 14-bit accuracy ([27]) in standard CMOS, as well as configurability through long-term retention of FG charge (0-100 μV over 10 years ([28])). The routing fabric itself is made of FG devices, which allows for both local memory and efficient computation. There has been a long history of analog computing on the FPAA [25], [26], [29].

The flexibility of the SoC FPAA makes it an ideal platform to compare Hopfield and Ising networks on. Different analog circuits can be compiled on the exact same chip and tested, allowing for a fair comparison. Although the implementations of either network may not be the fastest or most efficient, the point is comparison, and both networks will deal with identical parasitics and non-idealities. The next few subsections will detail the circuit implementations of both Hopfield and Ising networks on the FPAA.

B. Source Coupled VMM

Both Hopfield and Ising networks have weighted interconnects, **W**, multiplied by a state vector, **x** or **s**. A Vector Matrix Multiply (VMM) can compute this product, and on the FPAA the routing fabric doubles as a VMM because the switch elements are also precisely programmable FG devices. Input voltages come to the source of an FG pFET and weight the resulting current,

$$I = I_{th} e^{(\kappa(V_{dd}-V_{fg}-V_{T0})-(V_{dd}-V_s)+\sigma(V_{dd}-V_d))/U_T}, \quad (3)$$

assuming a saturated transistor ($V_{ds} > 4U_T$). The pFET has a threshold current (I_{th}) and associated threshold voltage (V_{T0}), κ is the capacitive coupling ration for the transistor between the gate and surface potential, and U_T is the thermal voltage or kT/q, roughly 25 mV at room temperature. The FG pFET has capacitive coupling onto the floating node

$$V_{fg} = (C_1/C_T)V_g + (C_{ov}/C_T)V_d + V_Q, \quad (4)$$

where the coupling gate input capacitor is C_1 and the coupling parasitic capacitance from V_d is C_{ov}. The total capacitance at the FG node is C_T, and V_Q is the programmed charge stored at the FG node. In the FPAA V_g is fixed to a voltage near Gnd in operational, or run, mode.

Assuming a change at that source ΔV_s and a corresponding change in drain voltage ΔV_d around a nominal bias current I_{s0} the pFET VMM device can be modeled as

$$\begin{aligned} V_s &= V_{s0} + \Delta V_s \\ V_d &= V_{d0} + \Delta V_d \\ I &= I_{s0} W e^{(\Delta V_s - \sigma \Delta V_d)/U_T}, \end{aligned} \quad (5)$$

where W is a positive weight that can be programmed by changing the V_Q on the FG node. When multiple weighted FG pFETs are connected to the same drainline, KCL adds the weighted currents up to create a VMM.

C. Hopfield Networks on the FPAA

The weighted connections into a Hopfield neuron result in an output along a sharp sigmoid

$$S_i = \begin{cases} 1, & \sum_j O_{ji} \geq T \\ 0, & \sum_j O_{ji} < T, \end{cases} \quad (6)$$

modeled on the FPAA with a TA that outputs a sigmoidal voltage according to the difference in voltage on its positive and negative terminals. This allows for signed weights, where positive and negative weights go to the corresponding VMM row for the neuron. The VMM connects to the TA through a transistor to GND, which converts the weighted current on each row into a voltage input for the Hopfield neuron to compare (Fig. 3).

Four Hopfield neurons can fit in one CAB of the FPAA. Each neuron needs two rows of a VMM and one TA. The largest bus into a CAB is 13 lines, reserving one line for an external input leaves 12 external recurrent connections. This added to the four local recurrent connections inside a CAB means the FPAA can create an all-to-all Hopfield network of 16 neurons.

D. Ising Networks on the FPAA

The oscillator for the Ising network on the FPAA is generated by a Hopf bifurcation circuit: the MeadSOS ([27]), which has known oscillatory dynamics created from the non-linear feedback element

$$\frac{d^2y_{out}}{dt^2} + \frac{dy_{out}}{dt}\left(2 - b + \frac{b}{3}\left(\frac{dy_{out}}{dt}\right)^2\right) + y_{out} = 0, \quad (7)$$

where b is the bifurcation parameter. By putting an oscillatory current into the feedback node, coupling between oscillators can occur (Fig. 4). This coupling signal can be weighted by a VMM, allowing for Ising networks to be formed out of coupled MeadSOS oscillators. A signed weight can be achieved by using two rows of a VMM similar to the Hopfield network where one row adds current into the feedback node and the other goes into an nFET mirror which takes current away from the node (Fig. 3c). Current going into the feedback node corresponds to a positive weight (matched phases) while current leaving the node corresponds to a negative weight (opposite phases).

One Ising neuron can fit inside a CAB on the FPAA. Each neuron needs two rows of a VMM, and the MeadSOS oscillator requires two FG TAs and one TA. The last TA is used as a buffer to fix the output capacitance of the MeadSOS and to change the 600 mV sinusoidal signal into a rail-to-rail output that can drive a source coupled VMM. The reference voltages can be generated inside a CAB by FG pFETs attached to GND and Vdd, so all 12 inputs (plus 1 external) can be used as recurrent connections to make the largest all-to-all Ising network of 13 neurons.

E. Nonidealities Between Devices on the FPAA

Process variation is inherent to any analog circuit. Devices don't exactly match, leading to differences between TAs, FG pFETs, and Nmirrors. In both the Hopfield and Ising networks mismatch causes non-ideal deviations that need to be accounted for.

The source coupled VMM pFETs are affected by mismatch and the indirect programming scheme of the FPAA. Measured pFETs during programming are not the same device used in the VMM, and mismatch between these transistors leads to

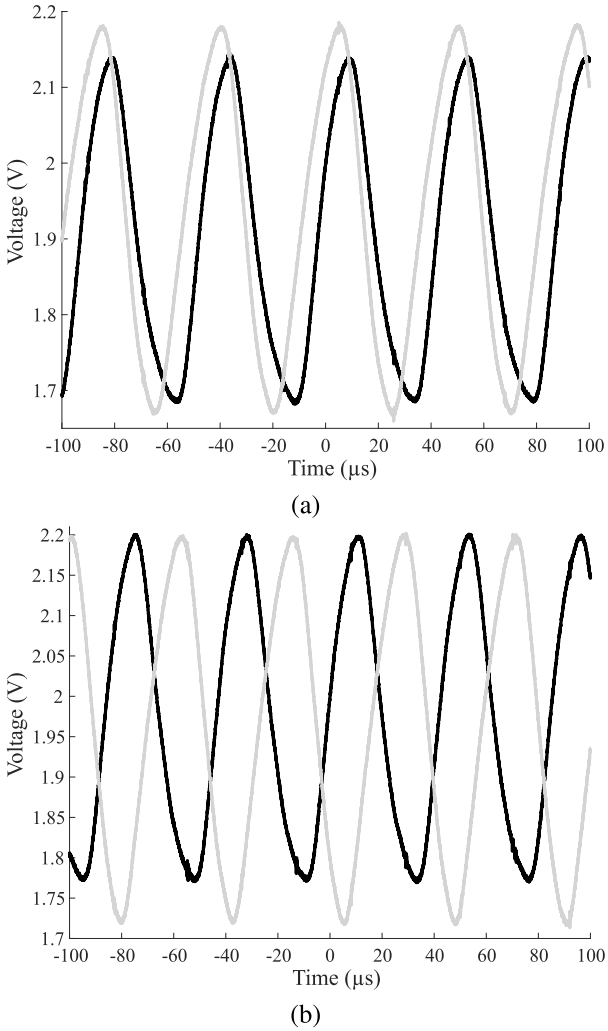


Fig. 4. Coupled MeadSOS oscillators that form the Ising network. a) Oscillators lined up in phase by coupling through a VMM and Nmirror, taking current away from the feedback node. b) Oscillators opposite in phase by coupling through a VMM, adding current onto the feedback node.

deviation in programmed weights between neurons. This can be accounted for with a calibration algorithm [30], which reduced the voltage mismatch of the VMM+I2V system's standard deviation from 26.5 mV to 1.6 mV over 816 pFETs.

Similarly, the indirect programming scheme affects the programmed offset charge on the input of the FG TAs and the bias current on all TAs. Mismatch in the current mirror structure and input differential pair of the nine transistor TA circuit also affects behavior. These heavily affect the oscillatory behavior of the MeadSOS oscillator, and differences were manually calibrated out in an iterative manner according to expected dynamics [27].

Other effects due to noise and temperature variation were not explicitly calibrated out; however, because both networks use primarily subthreshold analog circuits, they are both affected keeping the comparison even. Thermal variation will affect the U_T term for both VMMs (3), changing the common mode of the weights but preserving their differences. The TA is also affected by noise, which can impact the comparison operation for the Hopfield network and cause phase jitter in the Ising networks.

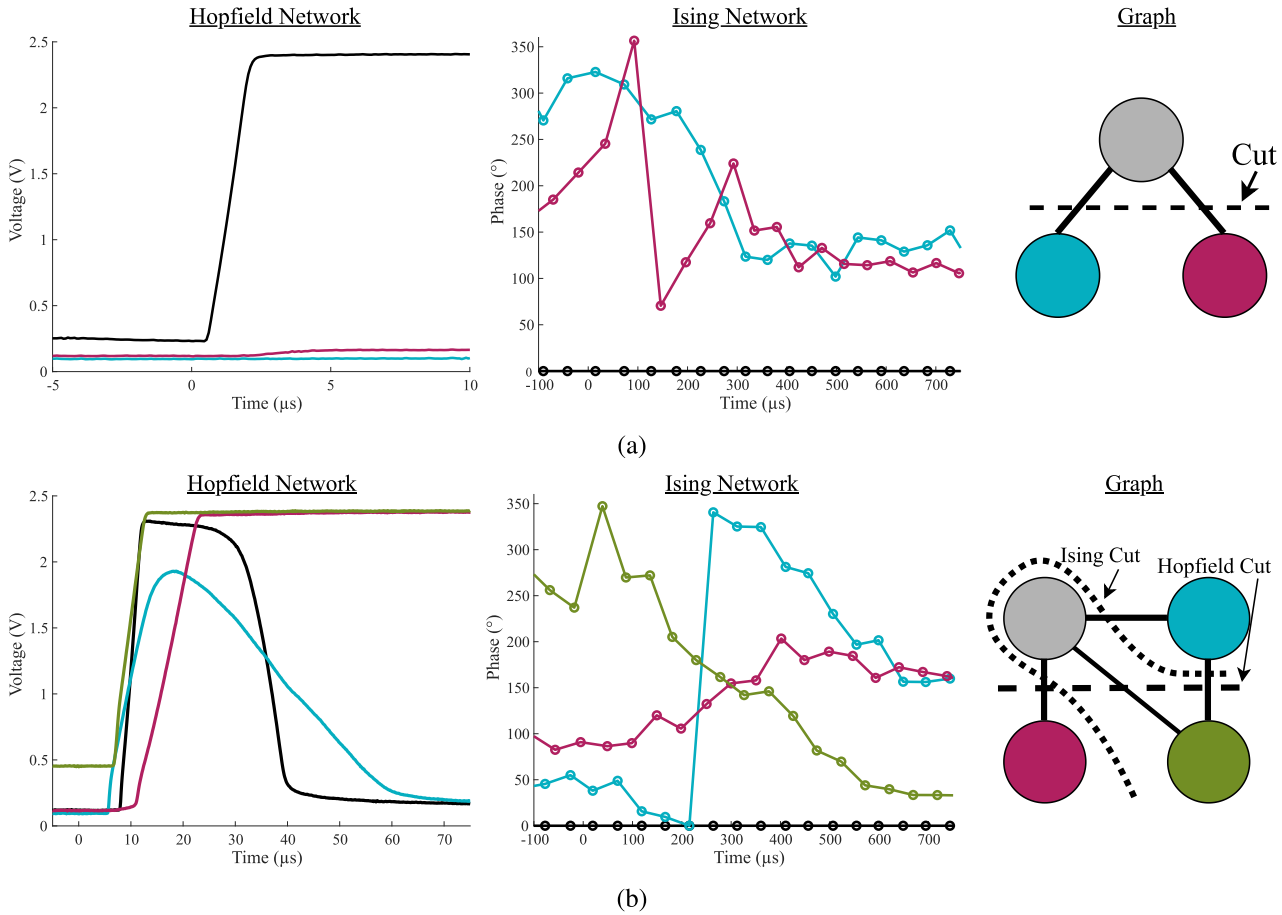


Fig. 5. a) Three neuron max-cut problem converging to the optimal solution from an unstable state on both a Hopfield and Ising network built on the FPAA. Both networks converge to the same energy minimum b) Four neuron max-cut problem converging to two different, but optimal, solutions from an unstable state for both a Hopfield and Ising network built in the FPAA.

III. Solving Energy Minimization Problems

Both Hopfield and Ising networks minimize quadratic energy surfaces as described in (2) and (1). These energy expressions can be written in summation form as well

$$E(\mathbf{x}) = -\frac{1}{2}\sum_{i}^{N}\sum_{j}^{N} W_{ij}x_i x_j - \sum_{k}^{N} I_k x_i, \tag{8}$$

for Hopfield ([20]) and

$$H(\mathbf{s}) = -\sum_{i}^{N}\sum_{j}^{N} J_{ij}s_i s_j - \sum_{k}^{N} h_k s_i, \tag{9}$$

for Ising [18]. If the ground state of these energy functions describes the solution to a problem, Hopfield and Ising networks can find good and even optimal solutions to otherwise computationally difficult problems. The NP-hard class of problems are one example of such problems, and quadratic energy surfaces that describe these problems have been formulated [18]. We will examine two optimization problems: max-cut (Section IV-A) and associative memory (Section IV-B), solving both with Hopfield and Ising networks on the FPAA.

A. Case Study: Max-Cut

The max-cut problem is an NP-hard problem that asks for a partition of a given graph into two sets such that the number of edges between the partitions is maximum [1]. For the Ising network, with spins of -1 and 1, an energy function with a ground state that corresponds to the optimal solution for the max-cut problem has been shown

$$H = H_A + H_B = A\left(\sum_{i=1}^{N} s_i\right)^2 + B\sum_{(uv)\in E}\frac{1 - s_u s_v}{2} \tag{10}$$

where H_A constrains the partition split to be even, and H_B constrains the cutsize [18]. This energy function is for both the min-cut and max-cut problems, to solve max-cut B needs to be a negative number. Intuitively we can see that with a negative B, more cuts lower the energy so that the maximum amount of cuts would result in the lowest energy. Each vertex of the problem graph maps to one neuron of the Ising network, and the relative phase between neurons determines what partition they end up in. We can match this energy function with the standard format for an Ising network, assuming $A = 0$, where the weight matrix and input biases are defined as

$$W_{ij} = \begin{cases} B * w_{ij} & i, j \in E \\ 0 & \text{otherwise,} \end{cases} \tag{11}$$

$$h_k = 0, \tag{12}$$

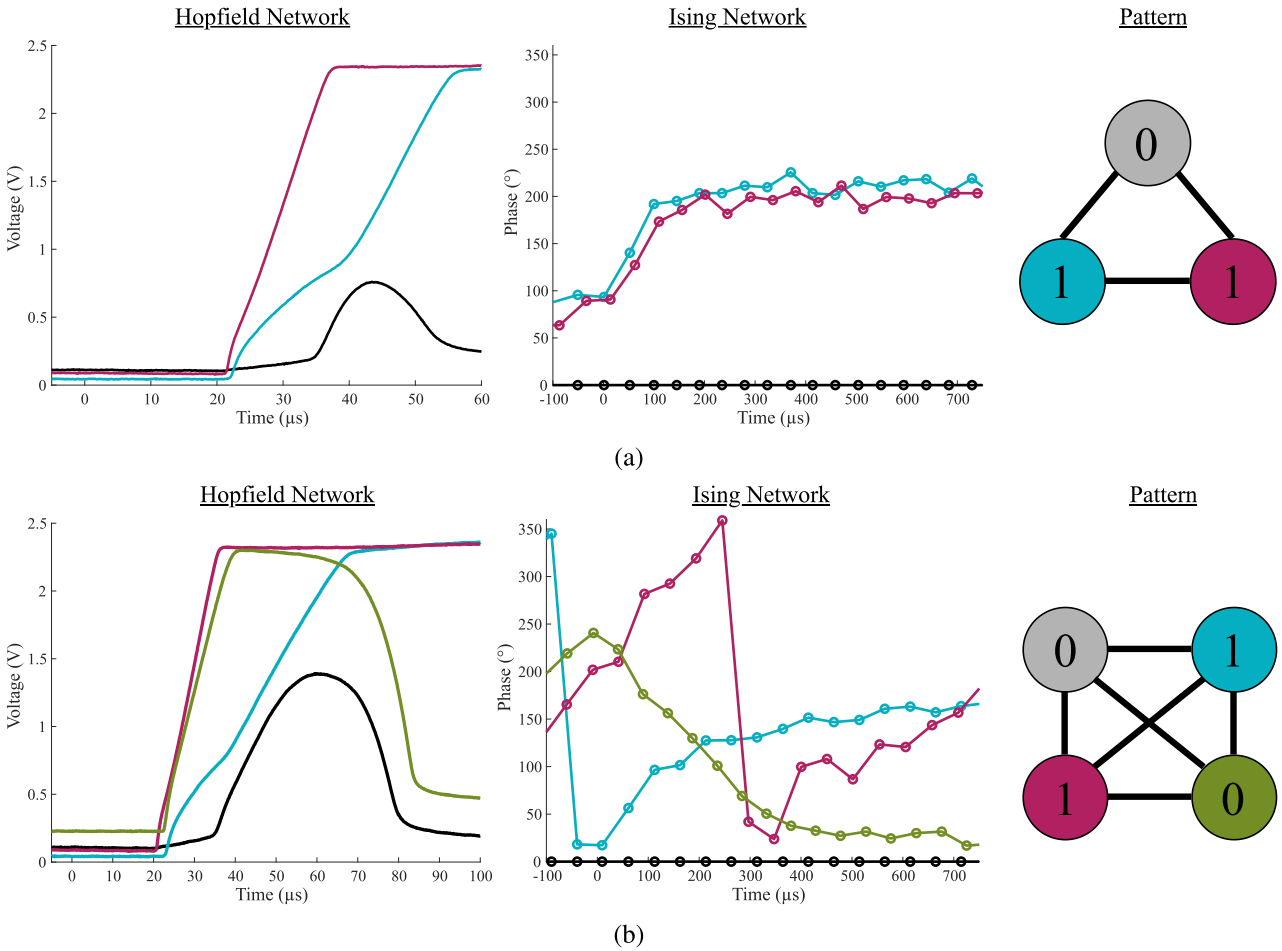


Fig. 6. a) Both analog Hopfield and Ising networks restoring the three bit stored pattern from a distorted initial state on the FPAA b) Both Hopfield and Ising networks restoring the four bit stored pattern from a distorted initial state on the FPAA.

where w_{ij} is the edge weight between vertices i and j.

The Hopfield network has output states of 0 and 1, so the energy function for an Ising network cannot exactly be used. However, a substitution can be made

$$s_i = 2x_i - 1 \tag{13}$$

where x_i is 0 or 1. Substituting into (10), simplifying the expression and ignoring the partition size constraint ($A = 0$) gets us to the Hopfield energy expression (2) where the weight matrix and input biases are defined as

$$W_{ij} = \begin{cases} B * w_{ij} & i, j \in E \\ 0 & \text{otherwise,} \end{cases} \tag{14}$$

$$I_k = -\frac{1}{2}\sum_j W_{kj}, \tag{15}$$

where w_{ij} is the same as in the Ising case.

Three and four neuron max-cut problems were programmed onto an Ising network built on the FPAA and allowed to converge to a final solution (Fig. 5). The constant B was chosen to be -1, corresponding to Max-Cut. In both cases the network converges to an optimal solution with a clear separation between the final states of each neuron in the network, both in phase and voltage.

B. Case Study: Associative Memory

The associative memory (AM) problem was the first problem to be solved on the Hopfield network, introduced during its debut [19]. Patterns are stored by creating an energy minimum at the network state corresponding to the desired pattern, and they are recalled through the energy minimization properties of the network. The weight matrix for a given pattern V maps onto the Hopfield energy function

$$W_{ij} = \sum (2V_i - 1)(2V_j - 1) \tag{16}$$

$$I_k = 0, \tag{17}$$

where each neuron represents one bit in the stored pattern. The Ising network formulation can be derived by performing the same substitution as before in (13)

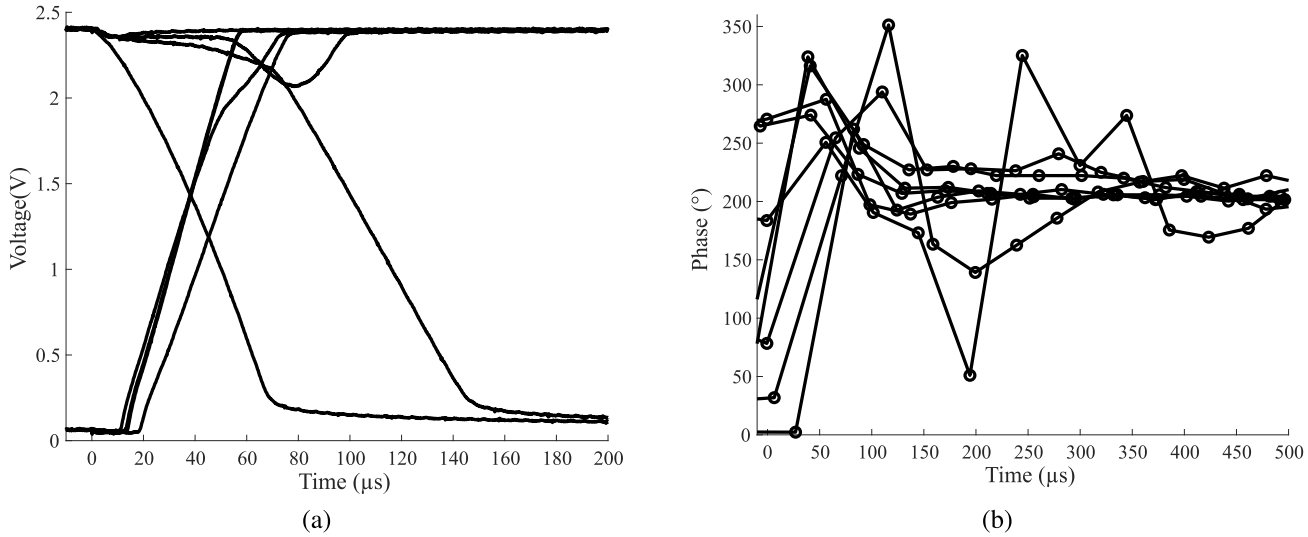
$$J_{ij} = \sum s_i s_j \tag{18}$$

$$h_k = 0, \tag{19}$$

where in both cases the weight matrix is

$$W_{ij} = J_{ij} = \begin{cases} 1 & V_i = V_j \\ -1 & V_i \neq V_j, \end{cases} \tag{20}$$

resulting in a full all-to-all network.



(a) (b)

Fig. 7. a) Single neuron of the Hopfield network converging to its final state over eight different initial conditions. An initial condition is forced through an external static input, and then relaxed to allow convergence. b) Single neuron of the Ising network converging to its final state over eight different initial conditions. An initial condition is set by uncoupling the neurons from each other. The network is then instantly coupled together to allow for convergence.

Three and four bit AM problems were programmed onto Hopfield and Ising networks built on the FPAA and allowed to settle (Fig. 6). Both networks converge back to the stored pattern from their initial condition, correctly retrieving the stored memory from a distorted version. The separation in final state is clear in both phase and voltage.

IV. Comparison Between Hopfield and Ising

While Hopfield and Ising networks solve mathematically identical equations the method in which states are encoded differs: static output level for Hopfield and relative phase for Ising. These separate encodings result in tangible differences between the two networks in metrics such as power & energy (Section IV-A), convergence time (Section IV-B), and necessary peripheral circuitry (Section IV-C) even when both correctly solve identical problems. To test this the two case studies (max-cut and associative memory) are examined on a three and four neuron network, covering unique cases for each problem. Networks were forced to an unstable initial condition, eight per problem, and then allowed to converge.

The initial condition was set on a Hopfield network through the activation of external inputs that forced a non-ideal state. Because of the high gain sigmoidal nonlinearity most states of the Hopfield network are binary, so the initial conditions also forced a binary unstable starting point. The external input was then released, allowing the network to converge back to a stable energy minimum (Figure 7a).

A random initial condition was set on the Ising network by switching V_{ref} to Gnd, uncoupling the network. The neuron phases were allowed to drift apart, and then V_{ref} was set back to the midpoint of the oscillations, ($\approx$ 1.3 V) which coupled the oscillators together and caused convergence to a final state. An external signal was applied at twice the frequency of the oscillators to force convergence to two different states. Because neuron 1 is always used as a reference, it always converges in zero time and consumes zero energy (Figure 7b).

A. Convergence Time

Both networks converged to the optimal solution from every trial in both case studies, regardless of initial condition. The time taken to converge to this optimal final solution depends on the type of network, the initial conditions, and the programmed problem. Over the eight initial conditions for each problem, the Hopfield network consistently converged faster than the Ising network (Fig. 8), averaging around 100 µs overall while the Ising network averaged 425 µs.

Ising network readout depends on phase, so at minimum one oscillation cycle needs to pass to measure convergence. The average frequency of the MeadSOS oscillators used was 25 kHz, which means at least 40 µs would need to pass become confirming convergence. In this time the Hopfield network could have already finished converging, as demonstrated in the three neuron problems. A faster oscillation period could speed up Ising convergence time, but at the cost of energy.

Hopfield network readout depends on a binary voltage level and would be limited by the clock speed of the digital periphery. System convergence is limited by the delay in between neurons, a combination of parasitic capacitance, transistor switching speeds, and activation function speeds. The overall convergence time did take longer than one Ising oscillation period in the larger 4 node network, but the Hopfield network was still faster to converge over time.

Larger Hopfield and Ising networks have shown a logarithmic trend in convergence time versus network size, suggesting both architectures scale well to large problems [5], [8]. Because both networks are mathematically similar, the similarity in trend makes sense. However, the results of this work suggest that the absolute values of convergence time and energy favor the Hopfield network despite both networks scaling similarly.

Hopfield and Ising networks can get stuck in local minimum, increasing convergence time when trying to find optimal solutions. Annealing solutions can be used on both networks

		Hopfield Convergence Time (µs)					Ising Convergence Time (µs)				
Size	Problem	N1	N2	N3	N4	Total Avg	N1	N2	N3	N4	Total Avg
3	MaxCut 2E	33.68	63.62	47.84	–	48.38	0	485.44	710.24	–	398.56
	MaxCut 3E	63.75	92.80	44.83	–	67.13	0	940.16	675.56	–	538.57
	AM 001	41.80	69.36	55.65	–	55.60	0	817.36	469.28	–	428.88
	AM 111	16.41	19.03	18.76	–	18.07	0	391.56	288.44	–	226.67
4	MaxCut 3E	146.10	145.54	133.51	120.76	136.48	0	269.92	332.92	715.6	329.61
	MaxCut 4E	92.37	96.57	92.88	107.02	97.21	0	339.44	826.24	578.48	436.04
	MaxCut 5E	203.90	120.53	76.18	176.28	144.22	0	250.84	208.36	212.64	167.96
	MaxCut 6E	71.22	102.70	56.76	96.68	81.84	0	617.88	392.96	788.48	449.83
	AM 0001	108.88	86.97	115.36	147.45	114.66	0	168.68	222.96	358.28	187.48
	AM 1100	166.60	191.04	157.77	142.79	164.55	0	410.16	244.12	191.12	211.35
	AM 1111	116.89	68.32	80.11	60.80	81.53	0	172.32	46.96	215.28	108.64
	Averages	96.51	96.04	79.97	121.68	91.79	0	443.16	401.64	437.13	316.69

Fig. 8. Average convergence time in three and four neuron Hopfield and Ising networks solving the max-cut and associative memory problems. Each network is started from 8 different initial conditions, and the convergence of each neuron is measured individually. The binary max-cut problem is labeled by the number of edges, or E. The associative memory (AM) problem is labeled by the binary pattern stored. On average the Hopfield network converges faster than the Ising network.

to favor better solutions, and typically take the form of analog noise which can be injected into either architecture [4], [5]. Because the core network behavior is unchanged through annealers, we would still expect Hopfield networks to outperform their oscillatory Ising counterparts.

Advanced technology nodes and devices can be used to accelerate both Hopfield and Ising networks, and we would expect any hardware performance gains from a smaller process node to help both architectures. A higher gain sigmoid, faster and smaller transistors, and less parasitic capacitances are some examples of optimizations that would improve the time constant of the Hopfield network. While it is unclear how well the time constants of the Hopfield network scale on very large problems, here it is demonstrably faster than the equivalent Ising network built using the same devices on the same analog computing platform.

B. Power and Energy

The energy taken by the networks over their convergence window is intrinsically tied to the convergence time. Energy calculations agree with this (Fig. 9), with the Ising network requiring more energy than the Hopfield network. Other work also follows this trend [4], [5]. The energy consumed by the buffers in both networks was omitted as they consume a similar amount of energy and are not intrinsic parts of the network structure. In addition, the energy provided by external inputs, such as the subharmonic injection locking (SHIL) signal for the Ising network, was omitted.

Energy usage of the Ising network is dominated by the oscillator. Two of the TAs used in the MeadSOS oscillator had bias currents of 200 nA, while the other had a bias current of 80 nA. The weights were programmed to 10 nA for the three neuron network and 20 nA for the four neuron network; the four neuron network would not converge with the lower weights. While a faster oscillator would speed up the convergence time, the energy usage is tied to the number of oscillatory cycles and would remain the dominant factor.

In contrast the Hopfield network's energy usage is dominated by the weights, which were programmed to 10 nA for all problems. The lower bound of the Hopfield weights is limited by the resolution of the TA and current to voltage transformer; the TA needs to be able to compare the voltage difference that results from 1 additional weight. If we look at the average power consumed by the networks while solving the test cases the Hopfield network takes around 0.02 µW and the Ising network takes 1 µW.

C. Peripherals and General Usage

The analog nature of both networks is essential to their efficiency; however, end-to-end systems will have some peripheral digital and analog interfacing. The Hopfield network state is essentially a digital signal, and any digital circuit could read in the network state. An Ising network state is in phase, so a phase detection circuit needs to be employed, raising the complexity of a full system implementation.

External inputs also need to follow the state variables of each network. Many NP-hard problems that can be mapped and solved on Hopfield and Ising networks require an external input to set a threshold, [18], meaning circuits that generate the input are necessary. The Hopfield network simply requires a digital signal in. An Ising network requires oscillatory signals, additionally adding to the circuitry needed. The Ising network additionally always requires one external signal for the SHIL, which is twice the frequency of the network oscillators. These external signals further the complexity of full system implementations.

Mismatch between neurons affects both Hopfield and Ising networks. Each network required an initial calibration to program out the process variation inherent to the analog circuits. After this calibration some additional tuning was necessary for each problem. The Hopfield network required minimal additional tuning, only needing to adjust the strength of an external bias current for the max-cut problem. The Ising network required a larger amount of problem-specific tuning because the feedback node DC level is affected by

		Hopfield Convergence Energy (pJ)					Ising Convergence Energy (pJ)				
Size	Problem	N1	N2	N3	N4	Total Avg	N1	N2	N3	N4	Total Avg
3	MaxCut 2E	0.15	0.06	0.06	–	0.09	0	472.09	690.74	–	387.61
	MaxCut 3E	0.09	0.10	0.12	–	0.10	0	926.22	665.59	–	530.60
	AM 001	0.24	0.15	0.15	–	0.18	0	805.18	462.28	–	422.48
	AM 111	0.01	0.06	0.14	–	0.07	0	385.71	284.15	–	223.29
4	MaxCut 3E	3.95	0.41	0.65	1.08	1.52	0	272.88	328.04	705.01	326.48
	MaxCut 4E	0.22	0.40	0.86	0.46	0.49	0	342.92	834.72	584.25	440.47
	MaxCut 5E	1.98	1.67	2.41	1.29	1.84	0	253.44	210.54	220.13	171.03
	MaxCut 6E	0.30	0.46	0.02	0.62	0.35	0	639.86	406.92	816.37	465.79
	AM 0001	6.62	5.95	7.57	0.03	5.04	0	174.83	231.09	370.89	194.20
	AM 0011	3.43	1.76	6.76	6.11	4.52	0	424.89	252.91	197.79	218.90
	AM 1111	6.89	3.94	4.23	3.51	4.64	0	178.52	48.69	223.18	112.59
	Averages	2.17	1.36	2.09	1.87	1.71	0	443.32	401.42	445.37	317.59

Fig. 9. Averaged, approximate energy consumption of a three and four neuron Hopfield and Ising network solving the max-cut and associative memory problems. Each neuron was started from 8 different initial conditions and allowed to converge. The energy consumption of the buffers on both networks was excluded, as the numbers would be very similar and make a direct comparison less clear. On average the Hopfield network consumes less energy than the Ising network.

external currents such as the coupling signal. This resulted in the oscillation frequency, DC level, and amplitude of the oscillators all being slightly different for every problem. These differences could be adjusted for by tuning V_{ref} and the frequency of the SHIL but combined to require more work than the Hopfield network.

V. Summary and Discussion

The first direct comparison of Hopfield and Ising networks is presented, using the 350nm circuits in the SoC FPAA as an equitable comparison platform. Four Hopfield neurons and one Ising oscillator fit into an FPAA CAB, and process variation was accounted for in both networks using the programmable FG circuitry. Three and four node networks were tested on the max-cut and associative memory problems. For each problem the networks were set to eight different initial conditions and then allowed to converge to a final state. The convergence time, energy, and power of the convergence was compared, along with the necessary peripherals for each network. On average the Hopfield network outperformed the Ising network on all metrics, converging faster with lower average power usage. The Hopfield network was also easier to use, requiring less complex external signals and readout circuits. These results suggest that Hopfield networks may be a superior choice to Ising networks because of their higher efficiency and easier practical interfacing.

The scaling of this comparison result with larger networks is expected to yield similar results. Because the core features, oscillations versus static voltage levels, remains the same no matter the implementation it is still expected that Ising networks would consume more energy than Hopfield networks provided that the convergence time does not drastically differ. In addition, the interfacing difficulties of Ising networks such as SHIL and phase readout still contribute more overhead than a Hopfield network would use. Most other scaling challenges, such as memory overhead, larger control logic, and physical area are challenges that are shared by both networks and thus wash out in a comparison; however, new advances in many-phase Ising networks, where each spin has more than two states, would change the Ising scaling and thus would need to be considered in future comparisons [31].

Both Hopfield and Ising networks face challenges in effectively reaching globally optimal solutions. New algorithmic ways of using hardware solvers [24] have emerged that attempt to tackle these issues. Hybrid networks, which use a combination of analog and digital solvers [32], are another emerging technique that uses an analog solver as an initial condition to a digital machine. While these new solutions offer general architectural improvements, the results of this work suggest that Hopfield networks are the optimal choice for the analog portion of a combinatorial optimization solver.

References

[1] R. Karp, "Reducibility among combinatorial problems," in *Complexity of Computer Computations*. Boston, MA, USA: Springer, Jan. 1972.

[2] V. Paschos, "An overview on polynomial approximation of NP-hard problems," *Yugoslav J. Operations Res.*, vol. 19, no. 1, pp. 3–40, 2009.

[3] R. Howell, "Simulated annealing on NP-complete problems," in *Proc. ACMS Conf.*, Jun. 1989, pp. 1–20.

[4] S. Dutta et al., "An Ising Hamiltonian solver based on coupled stochastic phase-transition nano-oscillators," *Nature Electron.*, vol. 4, no. 7, pp. 502–512, Jul. 2021.

[5] F. Cai et al., "Power-efficient combinatorial optimization using intrinsic noise in memristor Hopfield neural networks," *Nature Electron.*, vol. 3, no. 7, pp. 409–418, Jul. 2020.

[6] 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," *Commun. Phys.*, vol. 5, no. 1, p. 333, Dec. 2022.

[7] N. Mohseni, P. L. Mcmahon, and T. Byrnes, "Ising machines as hardware solvers of combinatorial optimization problems," *Nature Rev. Phys.*, vol. 4, no. 6, pp. 363–379, May 2022.

[8] T. Wang and J. Roychowdhury, "OIM: Oscillator-based Ising machines for solving combinatorial optimisation problems," in *Unconventional Computation and Natural Computation: 18th International Conference, UCNC 2019, Tokyo, Japan, June 3–7, 2019, Proceedings 18*. Springer, 2019, pp. 232–256.

[9] R. L. Wang, Z. Tang, and Q. P. Cao, "A parallel algorithm for maximum cut porblem using gradient ascent learning of Hopfield neural networks," *IEEJ Trans. Electron., Inf. Syst.*, vol. 122, no. 11, pp. 1986–1994, 2002.

[10] J.-R. Jiang and C.-W. Chu, "Solving NP-hard problems with quantum annealing," in *Proc. IEEE 4th Eurasia Conf. IoT, Commun. Eng. (ECICE)*, Oct. 2022, pp. 406–411.

[11] S. Yarkoni, E. Raponi, T. Bäck, and S. Schmitt, "Quantum annealing for industry applications: Introduction and review," *Rep. Prog. Phys.*, vol. 85, no. 10, Sep. 2022, Art. no. 104001.

[12] J. Chou, S. Bramhavar, S. Ghosh, and W. Herzog, "Analog coupled oscillator based weighted Ising machine," *Sci. Rep.*, vol. 9, no. 1, p. 14786, Oct. 2019.
[13] P. O. Mathews et al., "High-speed CMOS-free purely spintronic asynchronous recurrent neural network," *APL Mach. Learn.*, vol. 1, no. 1, Mar. 2023, Art. no. 016107.
[14] 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 Electron.*, vol. 5, no. 5, pp. 310–317, May 2022.
[15] 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 Electron.*, vol. 6, no. 10, pp. 771–778, Aug. 2023.
[16] S. K. Vadlamani, T. P. Xiao, and E. Yablonovitch, "Physics successfully implements Lagrange multiplier optimization," *Proc. Nat. Acad. Sci. USA*, vol. 117, no. 43, pp. 26639–26650, Oct. 2020.
[17] R. Hamerly et al., "Experimental investigation of performance differences between coherent Ising machines and a quantum annealer," *Sci. Adv.*, vol. 5, no. 5, May 2019, Art. no. eaau0823.
[18] A. Lucas, "Ising formulations of many NP problems," *Frontiers Phys.*, vol. 2, p. 74887, Feb. 2014.
[19] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proc. Nat. Acad. Sci. USA*, vol. 79, no. 8, pp. 2554–2558, Apr. 1982.
[20] J. J. Hopfield and D. W. Tank, "'Neural' computation of decisions in optimization problems," *Biol. Cybern.*, vol. 52, no. 3, pp. 141–152, Jul. 1985.
[21] J. J. Hopfield, "Neurons with graded response have collective computational properties like those of two-state neurons," *Proc. Nat. Acad. Sci. USA*, vol. 81, no. 10, pp. 3088–3092, May 1984.
[22] P. O. Mathews and J. O. Hasler, "Physical computing for Hopfield networks on a reconfigurable analog IC," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2023, pp. 1–5.
[23] (2006). *Hopfield Networks Class Notes*. [Online]. Available: https://www.dna.caltech.edu/courses/cns187/problemsets/Hopfield_intro.pdf
[24] A. Lu et al., "Scalable in-memory clustered annealer with temporal noise of charge trap transistor for large scale travelling salesman problems," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 13, no. 1, pp. 422–435, Mar. 2023.
[25] J. Hasler, "Large-scale field-programmable analog arrays," *Proc. IEEE*, vol. 108, no. 8, pp. 1283–1302, Aug. 2020.
[26] J. Hasler and A. Natarajan, "An open-source toolset for FPAA design," in *Proc. Workshop Open-Source EDA Technol. (WOSET)*, 2020, pp. 1–4.
[27] J. Hasler, "A programmable on-chip Hopf bifurcation circuit," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 69, no. 12, pp. 4958–4968, Dec. 2022.
[28] V. Srinivasan, G. J. Serrano, J. Gray, and P. Hasler, "A precision CMOS amplifier using floating-gate transistors for offset cancellation," *IEEE J. Solid-State Circuits*, vol. 42, no. 2, pp. 280–291, Feb. 2007.
[29] S. Kim, S. Shah, R. Wunderlich, and J. Hasler, "CAD synthesis tools for floating-gate SoC FPAAs," *Design Autom. Embedded Syst.*, vol. 25, no. 3, pp. 161–176, Sep. 2021.
[30] S. Kim, S. Shah, and J. Hasler, "Calibration of floating-gate SoC FPAA system," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 9, pp. 2649–2657, Sep. 2017.
[31] C. Bybee, D. Kleyko, D. E. Nikonov, A. Khosrowshahi, B. A. Olshausen, and F. T. Sommer, "Efficient optimization with higher-order Ising machines," *Nature Commun.*, vol. 14, no. 1, p. 6033, Sep. 2023.
[32] A. Mallick, M. K. Bashar, D. S. Truesdell, B. H. Calhoun, and N. Shukla, "Overcoming the accuracy vs. performance trade-off in oscillator Ising machines," in *IEDM Tech. Dig.*, Dec. 2021, pp. 40.

Pranav O. Mathews (Graduate Student Member, IEEE) received the B.S. degree from The University of Texas at Dallas in 2021 and the M.S. degree from Georgia Institute of Technology in 2023, both in electrical engineering, where he is currently pursuing the Ph.D. degree with the School of Electrical and Computer Engineering, where he works on floating-gate, analog, and neuromorphic systems.

Jennifer O. Hasler (Senior Member, IEEE) received the B.S.E. and M.S. degrees in electrical engineering from Arizona State University in 1991, the Ph.D. degree in computation and neural systems from California Institute of Technology in 1997, and the Master of Divinity degree from Emory University in 2020. She is a Regents Professor with the School of Electrical and Computer Engineering, Georgia Institute of Technology. She received the NSF CAREER Award in 2001, the ONR YIP Award in 2002, the Paul Raphorst Best Paper Award at IEEE Electron Devices Society in 1997, the Best Paper Award at SCI 2001, the Best Paper at CICC 2005, the Best Sensor Track Paper at ISCAS 2005, the Best Paper Award at Ultrasound Symposium in 2006, the Best Demonstration Paper Award at ISCAS 2010, and the Best Paper Award at ICRC 2023.