IEEE Journal on Exploratory Solid-State Computational Devices and Circuits

Received XX Month, XXXX; revised XX Month, XXXX; accepted XX Month, XXXX; Date of publication XX Month, XXXX; date of current version XX Month, XXXX.

Digital Object Identifier 10.1109/JXCDC.2023.3238030

A CMOS Probabilistic Computing Chip with Hardware-Aware Learning

Jinesh Jhonsa, William Whitehead, Yihao Wu, David McCarthy, Shuvro Chowdhury, Kerem Y. Camsari, and Luke Theogarajan*

[1]Department of Electrical and Computer Engineering, UCSB, Santa Barbara, CA 93106 USA

CORRESPONDING AUTHOR: Luke Theogarajan (lusthe@ucsb.edu).

ABSTRACT This work demonstrates a compact probabilistic computing system based on a physics-inspired p-bit architecture with 440 interacting spins configured in a Chimera graph and occupying 0.44 mm² of silicon area. Area efficiency is achieved through a current-mode neuron update circuit and a mixed-signal design approach that integrates pitch-matched standard-cell analog blocks with digital logic and a shared power supply network. Device mismatch and process variations arising from this tightly coupled mixed-signal implementation are addressed using a hardware-aware contrastive divergence training framework. By incorporating non-ideal circuit behavior directly into the training loop, the system achieves reliable stochastic dynamics without requiring per-device calibration. Measurement results validate robust probabilistic operation across all spins. The chip is evaluated on probabilistic logic functions, including logic gates and full adders, and on combinatorial optimization problems such as Max-Cut. These results demonstrate that learning-enabled compensation of hardware non-idealities enables scalable probabilistic computing architectures. The proposed system demonstrates effective cross-layer co-design across circuit, architectural, and algorithmic levels, supporting emerging non-von Neumann computing paradigms.

INDEX TERMS Ising, p-bit, hardware-aware learning, mixed-signal

I. INTRODUCTION

PRobabilistic bits (p-bits) have emerged as a hardware-friendly abstraction for solving optimization problems, probabilistic inference, machine learning, and quantum-inspired computing [1]–[3]. A p-bit is a stochastic binary unit whose output fluctuates between two states with a tunable probability, enabling direct hardware sampling from Boltzmann-like distributions. Networks of interacting p-bits can naturally emulate Ising and Potts models, making them suitable for solving combinatorial optimization problems such as Max-Cut, Boolean satisfiability, and constraint satisfaction, as well as for implementing probabilistic logic and inference engines [4]–[8].

Ideally, probabilistic computation can be realized using intrinsically stochastic physical devices such as magnetic tunnel junctions and single-photon avalanche diodes [9], [10]. These devices naturally produce stochastic behavior arising from device-level noise and thermal fluctuations. While these approaches offer compact stochastic behavior, large-scale integration remains challenging due to fabrication complexity, device variability, and limited compatibility with standard CMOS design flows. Consequently, several recent efforts have explored CMOS-based annealing and Ising machines using digital, mixed-signal, and in-memory architectures [11]–[16] . Examples include latch-based continuous-time Ising machines [11], [12], SRAM-based and in-memory annealers [13], [14], and fully connected CMOS annealing processors such as STATICA [15]. Many prior hardware implementations of Ising and spin-based systems employ deterministic spin update dynamics, requiring stochastic behavior to be emulated using externally generated randomness and off-chip annealing control [26]. Such approaches rely on host-driven random number generation and temperature scheduling, resulting in increased area, power consumption, and system-level complexity

In contrast, p-bit–based probabilistic computing realizes stochastic spin dynamics directly in hardware through on-chip pseudo-random number generation and temperature control, enabling self-contained probabilistic sampling without dependence on externally supplied randomness or off-chip annealing schedules [25].The proposed architecture further supports single-cycle p-bit updates, allowing each probabilistic neuron to complete its update within a single clock period, thereby improving throughput and scalability. However, large-scale CMOS implementations of multi-bit p-bits remain constrained by the area overhead of the neuron

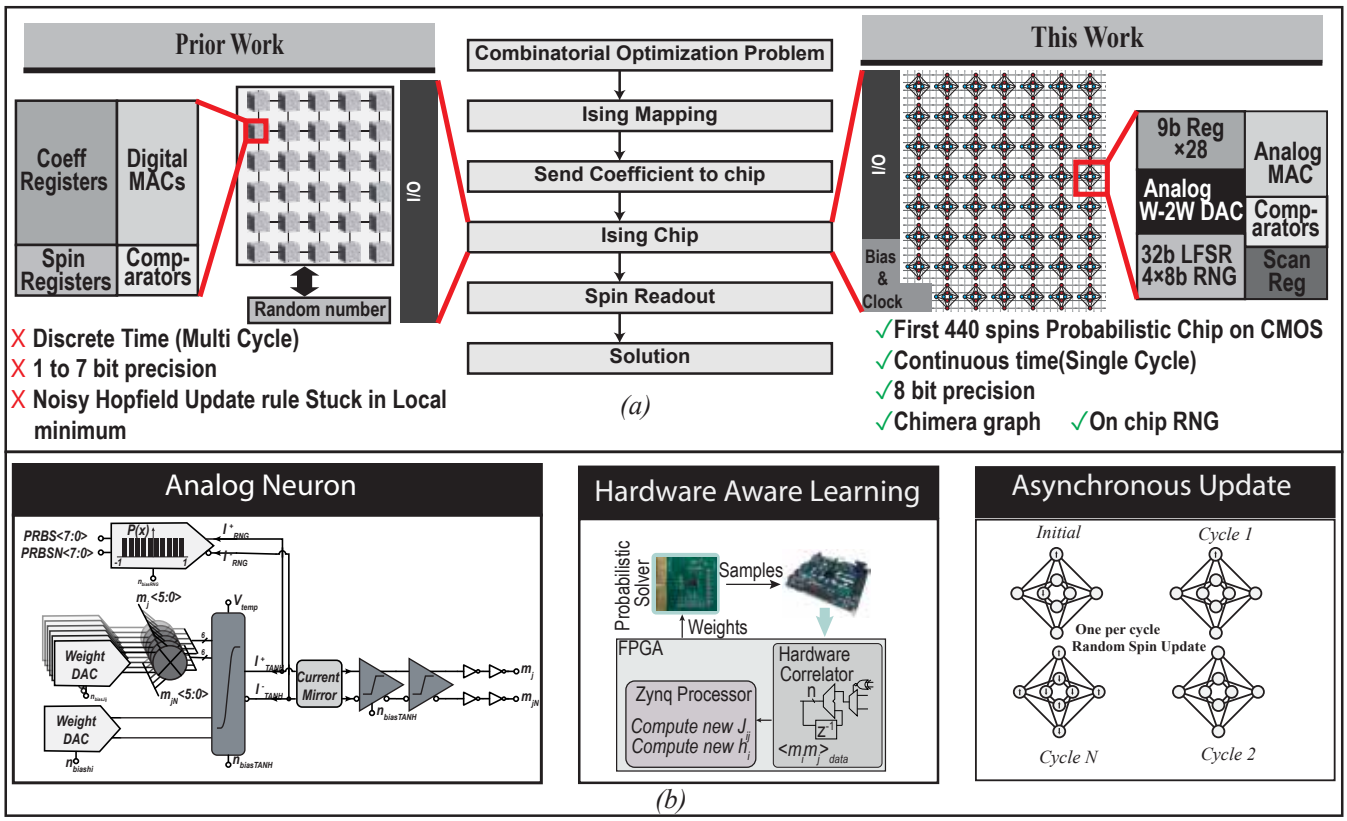

FIGURE 1. **a) Ising workflow for combinatorial optimization: prior deterministic versus proposed probabilistic b) Key features**

update circuitry, particularly the multiply–accumulate and nonlinear activation operations required to implement the probabilistic update function with sufficient precision. This challenge becomes increasingly severe for densely connected graphs and higher-precision p-bits, where the cost of implementing accurate activation functions dominates the overall system area.

In this work, we present a fully CMOS-integrated probabilistic computing chip that addresses these limitations through a mixed-signal, current-mode neuron update architecture. The key innovations in this work are: (1) quasi-asynchronous stochastic updates using decorrelated pseudo-random clocks, avoiding global lock-step behavior; (2) an area-efficient mixed-signal architecture compatible with standard digital place-and-route (PnR) flows; (3) hardware-aware learning that compensates for device mismatch and analog non-idealities. Specifically, area efficiency was improved by implementing analog blocks using standard-cell methodologies that are pitch-matched to digital logic and by sharing power delivery between analog and digital domains. Quasi-asynchronous operation was achieved using LFSR-based randomized clocking, reducing synchronization overhead and improving sampling diversity. Although this aggressive integration introduces process-variation-induced mismatches, these effects are mitigated using a hardware-aware contrastive divergence training approach, consistent with recent algorithm–hardware co-design frameworks for probabilistic computing [9], [17]. The fabricated chip integrates 440 p-bits arranged in a Chimera topology within an active area of 0.44 mm^2 and is experimentally validated on probabilistic logic circuits and optimization benchmarks such as Max-Cut.

From a system-level perspective, solving a combinatorial optimization or probabilistic inference problem using the proposed architecture follows a structured workflow.As shown in Fig1, First, the target problem is mapped onto an Ising or Potts energy function by encoding interaction coefficients (J_{ij}) and local biases (h_i) that define the problem Hamiltonian, see equation 1. These parameters are programmed onto the chip through on-chip coefficient registers and bias control circuitry. During operation, each p-bit updates its state stochastically based on the weighted interactions with neighboring p-bits and its local bias, implemented using a current-mode mixed-signal neuron update circuit. Randomized clocking enables quasi-asynchronous updates, allowing the system to naturally explore the underlying energy landscape. As the network evolves, low-energy configurations are sampled with higher probability, and the resulting p-bit states are read out to obtain candidate solutions. For learning-enabled tasks, measured spin correlations are fed back to an external controller to update interaction weights using a hardware-aware contrastive divergence algorithm, enabling ~~in-situ~~ adaptation and compensation of hardware non-idealities. This closed-loop workflow enables efficient

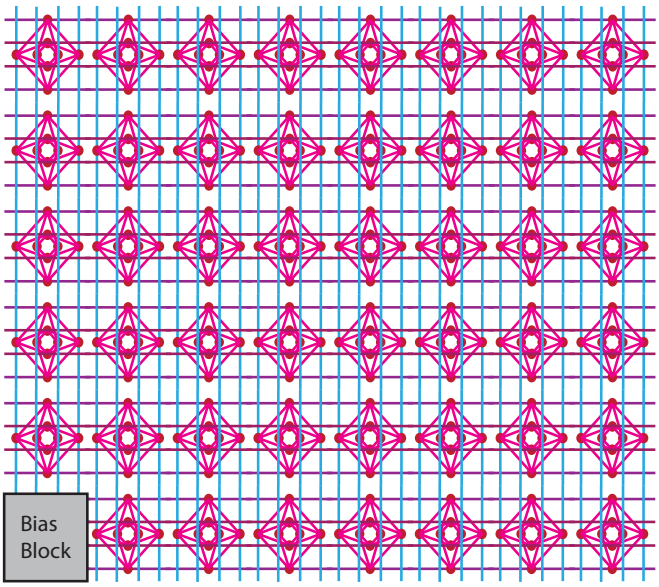

FIGURE 2. Chimera network with 440 p-bits arranged in 55 unit cells, each forming a 4×4 bipartite Restricted Boltzmann Machine (RBM).

probabilistic computation and optimization using a fully CMOS-integrated p-bit system

II. Architecture

Previously implemented digital Ising machines utilize a noisy Hopfield network to perform the computation, where the acceptance probability of a proposal (the local Hamiltonian) is a step function. While simpler to implement, it does not approach the Boltzmann equilibrium probability distribution. The p-bit utilizes the Barker/Glauber acceptance criteria requiring a tanh. Each P-bit computes two fundamental equations, shown below:

$$I_i = \sum_{j \neq i} J_{ij} m_j + h_i m_i \tag{1}$$

$$m_i = \mathrm{sgn}(\tanh(\beta I_i) + \mathrm{Rand}(-1, 1)) \tag{2}$$

Equation 1 represents the change in energy associated with flipping the ith spin. Equation 2 implements a stochastic spin update, where the probability of a spin flip is determined by a sigmoid function of the local energy change. In hardware, this is realized by comparing $\tanh(\Delta \mathrm{E_i})$ to a uniformly distributed random number, resulting in probabilistic acceptance of energetically favorable and unfavorable updates. We utilize a hardware friendly Chimera graph topology pioneered by D-Wave rather than the interconnect intensive all-to-all topology. The Chimera topology enables embedding many graphs in contrast to the commonly used King's graph. Each unit cell of the Chimera graph is a 4:4 Restricted Boltzmann Machine (RBM). Restricted Boltzmann Machines are bipartite graphs, where intra-layer connections are absent as shown in Fig. 2. Each RBM cell also receives inputs from neighbors. Our design philosophy was driven by two

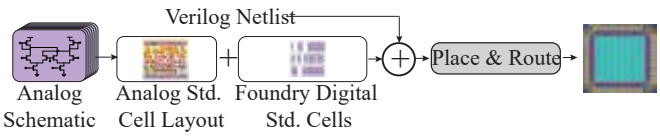

FIGURE 3. Automation layout methodology

major considerations, 1) area and 2) automation. The first constraint was met by utilizing current mode implementation of eqns. 1 and 2. The second constraint was met by adopting a standard cell design for all analog blocks, pitch matched to the digital blocks. The analog blocks are then treated like digital blocks by the automated place and route vastly simplifying the layout as shown in Fig. 3. We have recently published this approach of using digital PnR tools for mixed signal design [24]. The major difference in this design is the shared power supply between the analog and digital block for area efficiency. One downside to the approach is the analog circuits are subject to mismatch and power-supply noise. While this design methodology may not be suitable for general mixed-signal design, it lends itself to the probabilistic computing environment.The 440 spins are arranged in a 7×8 array of Chimera unit cells, with one cell substituted by bias circuits and SPI interfaces for loading weights and reading spin values. Digital weights (8 bits) were converted to analog bias values using a MOS transistor-based R-2R digital to analog converter (DAC) as shown in Fig. 4c. The MOS R-2R DAC was chosen due to its high area -efficiency and low complexity. However, it should be noted that the choice of using a low supply voltage (1V) and lack of any circuit techniques to improve output resistance will lead to some mismatch issues.

A. System Overview

Since the topology is an undirected graph, coupling strengths $\mathrm{J_{ij}} = \mathrm{J_{ji}}$, to save area the current was converted into a bias voltage and distributed to the respective nodes. As setting the weight to zero might not necessarily remove a connection due to mismatch, an enable bit was added as a precaution. The enable bit forces the bias voltage (and hence the current in the current DAC) to zero, see figure 4. It should be noted both differential biases are set to zero. The scale for coupling weights, bias weight, random number and tangent hyperbolic are independently set using external resistors as shown in Fig. 4a(Global bias inputs). Implementing equation 1 requires the multiplication of the weights by the spin value. Since the weight is represented by a current, a current mode Gilbert multiplier can be used (Fig. 4e) resulting in a differential representation. The differential format enables the straightforward representation of bipolar weights. Summation results directly from current summation by connecting the output nodes together. Each node has 6 current inputs summed on the current input to implement equation (2), the bias current branch ($\mathrm{h_i}$) utilizes the same current DAC circuit as the coupling term. For the p-bit

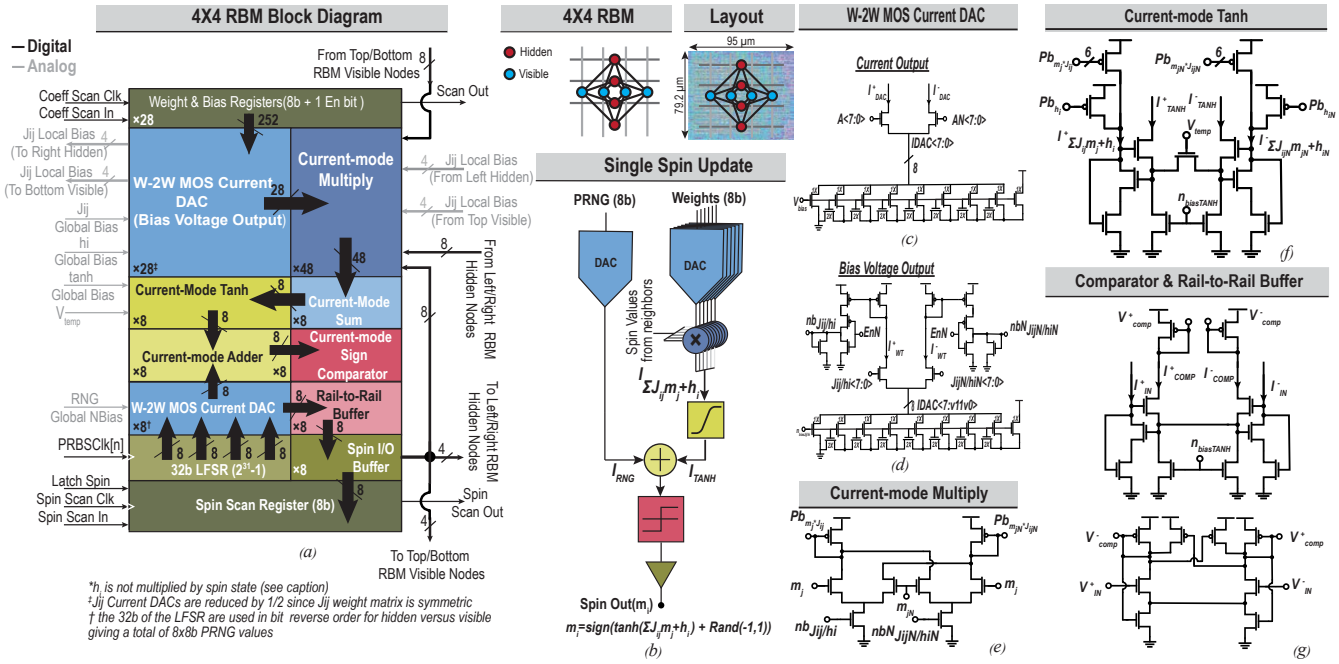

FIGURE 4. **(a) Detailed block diagram of a single 4×4 RBM unit cell in the Chimera network. Since h_i is a bias term it is always present irrespective of the spin state and therefore not multiplied by the spin state. To accommodate cell reuse the spin value, when the weight cell is used as the bias, is tied high. (b) Single p-bit spin update operation implementing (1). (c) W–2W MOS current digital-to-analog converter (DAC). (d) Bias voltage generation circuit. (e) Current-mode multiplier used for weight–spin multiplication. (f) Current-mode tanh circuit with V_{temp}-controlled temperature for simulated annealing. (g) Comparator and rail-to-rail output buffer.**

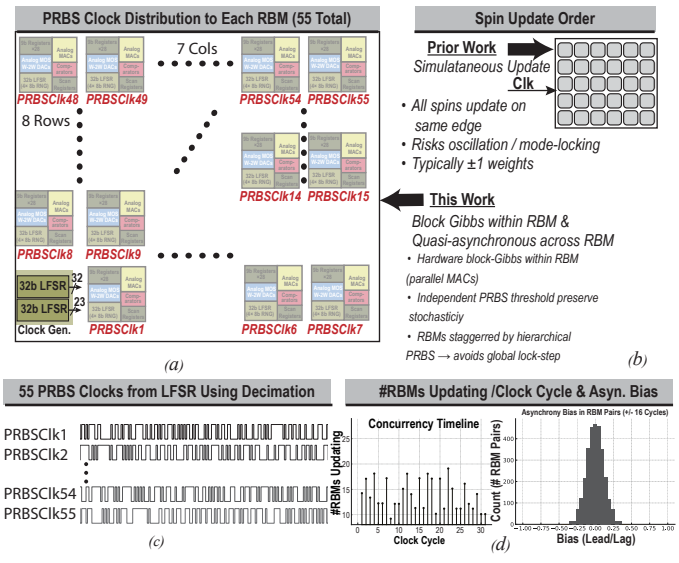

FIGURE 5. **Random clocking and quasi-asynchronous updates. (a) 55 decimated PRBS clocks distributed across RBMs. (b) Update order contrasting simultaneous versus block-Gibbs asynchronous updates. (c) Decimated LFSR clocks with normal and reversed bit ordering. (d) RBM concurrency and asynchrony bias.**

representation the summed current needs to serve as the input to a tangent hyperbolic (tanh) function. We adopted an improved fully differential winner take-all circuit as shown in Fig. 4f [19], to both pin the voltage enabling better current matching and perform a tanh transformation of the summed current. Each branch implements a sigmoid function of the differential currents and a subtraction then yields the required tanh function. The differential pairs in the tanh circuit were degenerated using long-channel NMOS devices with the gate voltage providing a means of scaling the computational temperature using a control voltage (V_{temp}) enabling on-chip annealing. Before performing the subtraction, a differential random current with a uniform distribution was summed to the output node, generated by a DAC circuit identical to the one utilized for the weight/bias, see Fig. 4c. The summed current from the tanh circuit and the random number generator was fed into a winner take-all based comparator and then further amplified using a self-biased fully differential comparator (Fig. 4g)

The value of the RNG DACs are set by a pseudorandom number generated by a decimated Linear Feedback Shift Registers (LFSR) [20], which generate a new pseudorandom value in every bit position each clock cycle. Bitstreams from two LFSRs clocked at 200 MHz generated 64 unique random clocks of which 55 were used to drive a 32 bit LFSR in each unit cell of the Chimera graph as shown in Fig. 5a. While the LFSRs acting as clock-generators are clocked by a common master clock, the use of independent pseudo-random sequences as clocks results in effectively aperiodic update events across the array. As a result, different RBM clusters receive uncorrelated update triggers, preventing synchronous global updates and approximating asynchronous stochastic dynamics. Since each 32-bit LFSR yields only 4 unique 8-bit random numbers, the vertical nodes of the chimera

graph received a normal bit sequence while the horizontal nodes received a reversed bit sequence as shown in Fig. 5c. While there is a possibility the reverse sequence and the original sequence could have some correlation, there was no noticeable degradation in chip performance using this method. This was verified by calculating the correlation of number of RBM pairs with +/- 16 cycles and is shown in Fig. 5d. The number of RBMs updating together is a maximum of 16 RBMs across the entire chip at any given clock cycle, this leads to a quasi-asynchronous block-Gibbs update as shown in Fig. 5b. Additionally, since the 4x4 RBM is a bipartite connection (no within layer connections) the visible and hidden layer are allowed to update simultaneously without causing issues.

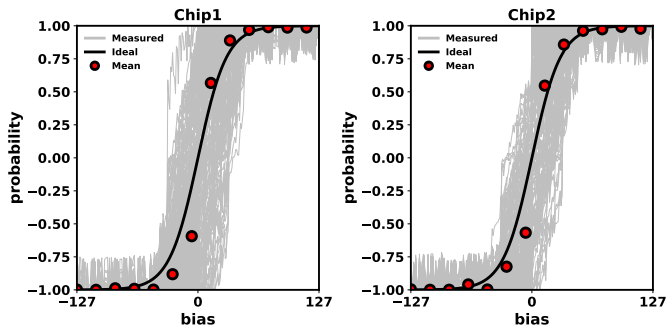

FIGURE 6. Sweeping 440 spins from -127 to 127 bias only connected spin.Measured variability across two different chips

III. Results

A central question in building a mixed-signal p-bit chip is the impact of device mismatch, leading to variations. To characterize device variability, a bias sweep was performed on all 440 p-bits by varying the input bias h_i from -127 to $+127$ while disabling the interconnection weights J_{ij} as shown in Fig. 6. The average spin value of each p-bit is expected to follow a hyperbolic tangent (tanh) characteristic as the bias is swept. However, due to device mismatch, each p-bit exhibits a distinct offset and slope, highlighting the inherent variability present in the chip.

The data shown in Fig. 6 indicates that device mismatch is the primary source of variation. Averaging across all 440 p-bits (spatial average across the chip) at each bias point yields the expected tanh response (with a residual offset). The measured offset variation across spins has a standard deviation of approximately 8% and 4% of full-scale for chip 1 and chip 2 respectively , indicating that static mismatch dominates. Supply-induced noise is largely mitigated by the differential design, where it appears as common-mode variation limited by the power supply rejection ratio.

To compensate for this variability, we employed a hardware-aware learning approach based on contrastive divergence to individually learn the effective offset and slope of each p-bit [9]. In the proposed p-bit network, correlation serves as the central statistical quantity for encoding logic constraints and probabilistic behavior, rather than explicitly enforcing deterministic truth-table outputs.

Algorithm 1: Correlation-Based Probabilistic Learning

Input: initial state configuration $(\mathbf{m}_{\text{init}})$
coupling matrix $(\mathbf{J})$, bias vector $(\mathbf{h})$
ideal correlations $(\mathbf{W}_{\text{ideal}})$, ideal biases $(\mathbf{B}_{\text{ideal}})$
number of samples per iteration S
maximum iterations MaxIter
learning rate ϵ, inverse temperature β
Output: learned coupling matrix $(\mathbf{J})$, bias vector $(\mathbf{h})$

$\mathbf{m} \leftarrow \mathbf{m}_{\text{init}}$;
for *iter* $= 1$ **to** *MaxIter* **do**
  Initialize empty set $\mathcal{M}_{\text{samples}}$;
  for *sample* $= 1$ **to** S **do**
    for $i = 1$ **to** N **do**
      `// Local field calculation`
      $y_i = \sum_j J_{ij} m_j + h_i$;
      `// Stochastic neuron update`
      $z_i = \tanh(\beta y_i) + \mathcal{U}(-1, 1)$;
      $m_i = \text{sign}(z_i)$;
    Store current $\mathbf{m}$ in $\mathcal{M}_{\text{samples}}$;
  `// Convert spin states`
  Convert $\mathcal{M}_{\text{samples}}$ from $\{-1, +1\}$ to $\{0, 1\}$;
  `// Model statistics`
  Compute $\mathbf{B}_{\text{model}}[i] = \sum_k \mathcal{M}_{\text{samples}}[k][i]$;
  Compute $\mathbf{W}_{\text{model}}[i][j] = \sum_k \neg(\mathcal{M}_{\text{samples}}[k][i] \oplus \mathcal{M}_{\text{samples}}[k][j])$;
  `// Normalization`
  $\mathbf{B}_{\text{model}} \leftarrow (\mathbf{B}_{\text{model}} - S/2)/(S/2)$;
  $\mathbf{W}_{\text{model}} \leftarrow (\mathbf{W}_{\text{model}} - S/2)/(S/2)$;
  `// Parameter update`
  $\mathbf{J} \leftarrow \mathbf{J} + \epsilon(\mathbf{W}_{\text{ideal}} - \mathbf{W}_{\text{model}})$;
  $\mathbf{h} \leftarrow \mathbf{h} + \epsilon(\mathbf{B}_{\text{ideal}} - \mathbf{B}_{\text{model}})$;

TABLE 1. Ideal truth table for an AND gate with inputs A and B, where C_1 and C_2 are copy-gate outputs equal to the AND result, each occurring with probability 1/4 = 0.25.

A	B	C_1	C_2	P_{ideal}
0	0	0	0	0.25
0	1	0	0	0.25
1	0	0	0	0.25
1	1	1	1	0.25

The learning process begins by computing the ideal correlations corresponding to the target logic function. For the AND gate example, the desired behavior is fully specified by the ideal node biases and pairwise correlations derived from the Boltzmann distribution of the truth table. The ideal

bias and pairwise correlations are computed as

$$\mathrm{corr}_{AB} = \frac{1}{N}\sum_{t=1}^{N}\left(2\,\mathrm{XNOR}\big(A(t),B(t)\big)-1\right), \quad (3)$$

$$\mathrm{bias}_{A} = \frac{1}{N}\sum_{t=1}^{N}\left(2A(t)-1\right). \quad (4)$$

As illustrated in Fig. 7(a), these ideal correlations define the target operating point of the network and are embedded into a slightly larger Chimera-compatible graph using minor embedding. The learning algorithm (Algorithm 1) then iteratively updates the bias and coupling parameters. During each iteration, stochastic spin updates generate samples from the current model distribution, from which the measured correlations are estimated. The parameters are updated proportionally to the difference between the measured and ideal correlations, driving the system toward convergence.

In each training iteration, as depicted in Algorithm 1, 10,000 samples are collected from the chip and the difference between the measured and ideal correlations is computed using an FPGA. The updated biases and weights are then written back to the chip. While the I/O overhead dominates training time (100MHz, 16316 cycles per update), this cost is incurred only once per problem instantiation. After training, inference and sampling proceed entirely on-chip without further communication overhead. The convergence of the learned parameters for the AND gate is shown in Figs. 7(b) and 7(c). Fig. 7(b) shows the learning dynamics, where the state probabilities converge toward the ideal Boltzmann distribution with a 0.25 probability peak. Fig. 7(c) illustrates the evolution of bias and weight correlations as a function of iteration count. Initially, the correlations deviate significantly from their ideal values due to random initialization and mismatch. As learning progresses, both bias and pairwise weight correlations converge toward their target values, reshaping the energy landscape to favor valid logical states while suppressing invalid ones.

The effectiveness of the hardware-aware learning framework is further quantified using the Kullback–Leibler (KL) divergence between the ideal probability distribution and the distribution produced by the fabricated p-bit chip, defined as

$$\mathrm{KL}\left(P_{\mathrm{ideal}} \parallel P_{\mathrm{exp}}\right) = \sum_{\mathbf{m}} P_{\mathrm{ideal}}(\mathbf{m}) \log \frac{P_{\mathrm{ideal}}(\mathbf{m})}{P_{\mathrm{exp}}(\mathbf{m})}. \quad (5)$$

where $P_{\mathrm{exp}}(\mathbf{m})$ is obtained by time-averaging sampled hardware configurations, and $P_{\mathrm{ideal}}(\mathbf{m}) \propto \exp(-\beta E(\mathbf{m}))$ denotes the target Boltzmann distribution.

As shown in Fig 7d During early learning iterations, mismatch-induced bias offsets and weak effective couplings result in a broad sampled distribution and a large KL divergence. As learning proceeds, adaptive updates of the coupling matrix and bias vector progressively align the measured correlations with the target statistics, leading to a reduction and eventual saturation of the KL divergence. It can also be see the KL divergence mimics the correlation dynamics shown in in Fig. 7(c). The convergence behavior is largely governed by the correlation between the copy nodes c_1 and c_2. These nodes enforce structural correctness of the AND-gate embedding by ensuring that both outputs represent identical logical values. As a result, the $\mathrm{c}_1 - \mathrm{c}_2$ correlation strongly influences the consistency of the embedded logical constraints, and its convergence dominates the overall KL divergence reduction.

TABLE 2. Ideal correlation table for a full adder derived from the ideal Boltzmann distribution, showing correlations between the inputs (A, B, C_{in}) and outputs (Sum S and Carry-out C_{out}), where all eight valid input–output states occur with equal probability $1/8$.

A	B	C_{in}	S	C_{out}	P_{ideal}
0	0	0	0	0	0.125
0	0	1	1	0	0.125
0	1	0	1	0	0.125
0	1	1	0	1	0.125
1	0	0	1	0	0.125
1	0	1	0	1	0.125
1	1	0	0	1	0.125
1	1	1	1	1	0.125

Overall, correlation-based learning provides a compact and hardware-friendly mechanism for programming probabilistic logic into large-scale p-bit networks. By enforcing agreement between ideal and measured correlations, the network naturally converges to the correct Boltzmann distribution, enabling reliable realization of logic gates and more complex probabilistic inference tasks.Future work may explore problem-aware initialization of the coupling matrix and bias vector. Such initialization strategies can place the network closer to the optimal energy landscape at the start of learning, potentially reducing the number of iterations required for convergence.

The probability distribution of a full adder was also implemented as shown in Fig. 8 to demonstrate the chip's ability to perform complex computations. As seen from the figure the network visits only the expected 8 low energy states out of 32 available states.

We also implemented the Max-Cut problem using all 440 p-bits, as illustrated in Fig. 9. The ground state of the system was configured to represent the word "IEEE." In the Max Cut formulation, interaction weights are either +1 or -1, which we mapped to +127 and -127, respectively, on the hardware. Since the p-bits are updated asynchronously, the system was able to reach the ground state within a single FPGA clock iteration. This is an "easy" max-cut problem, though it is often used as a benchmark in recently reported CMOS Ising machines. For example, a digital Ising machine with 8b coefficients converged to a similar "easy" instance after 120 cycles [21]. In our case, the 440-spin instance corresponding to the "IEEE" pattern converged to the ground state in a single iteration in $> 99\%$ of trials (N=20 runs). Because the coupling matrix is effectively

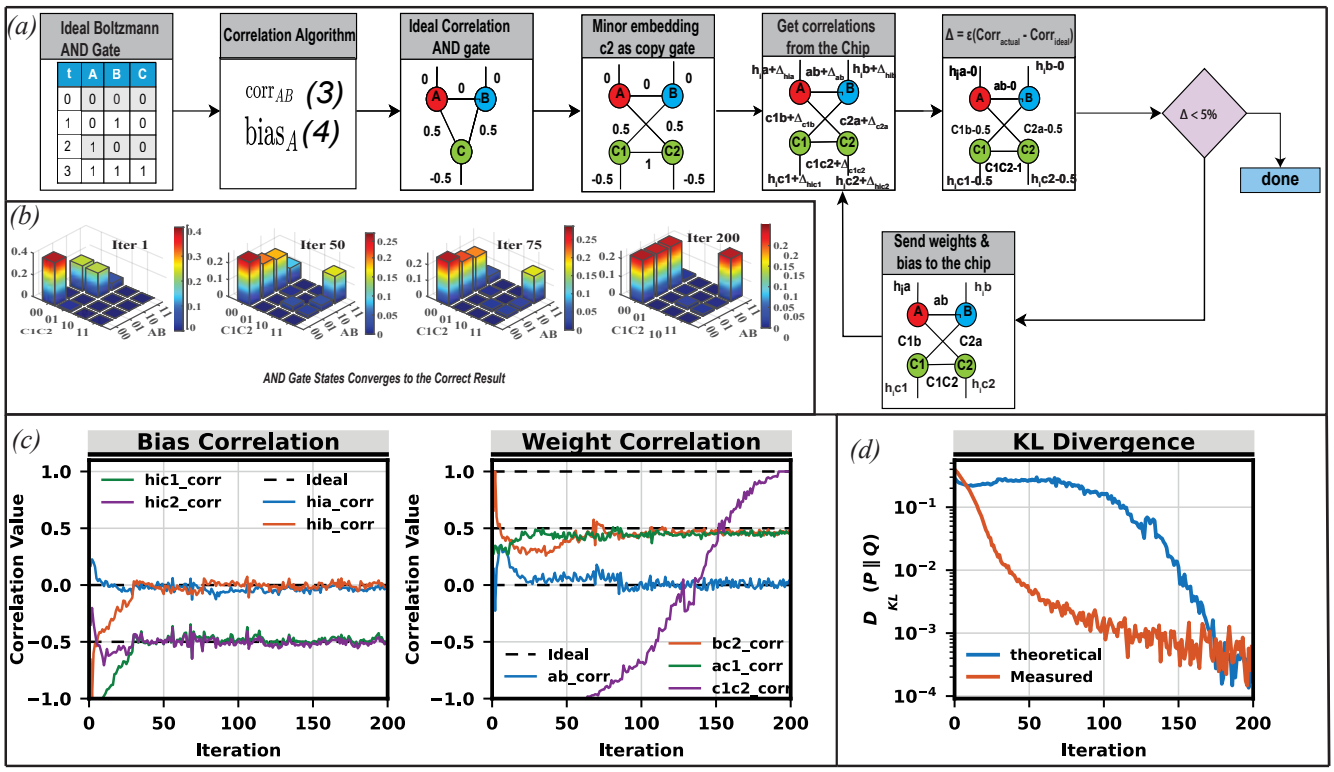

FIGURE 7. **(a) Learning flow showing minor embedding of the AND gate into the Chimera network and iterative correlation-based updates of biases and weights using hardware measurements. (b) Evolution of AND-gate output-state probabilities over learning iterations, converging to the ideal Boltzmann distribution. (c) Convergence of measured bias and weight correlations toward their ideal target values, compensating device-level variability. (d) Kullback–Leibler (KL) divergence between measured and ideal distributions as a function of iterations, demonstrating convergence.**

TABLE 3. **Comparison of CMOS-based Ising and probabilistic computing architectures**

Metric	ROSC Ising [16]	CMOS Latch Ising [11]	SRAM Ising [13]	Digital MAC Ising [14]	p-Circuits [27]	This Work (P-bit)
Technology	65 nm	65 nm	65 nm	65 nm	65 nm	65 nm
Core Architecture	Ring Oscillator	CMOS Latch	SRAM Cell	Digital MAC	p-gate	P-bit
Coupling Coeff.	1 b	Ternary $(-1, 0, 1)$	Ternary $(-1, 0, 1)$	7 b	2 b	8 b
Spin Variable	ROSC Phase	Latch Voltage	Latch Voltage	Binary State	Binary State	Binary State
Graph	Hexagonal	Lattice	e-Chimera	m-Zephyr	Lattice	Chimera
# Neighbors	6	4	11	24	7	8
Spin Operation Cycle	No	No	One-shot	Multi-cycle	Multi-cycle	Multi-cycle
Spin Update	Simulated Bifurcation	Latch Equalization	SRAM Equalization	Block Gibbs Update	Sequential Gibbs Update	Block Gibbs Sampling
Supply Voltage	1 V	0.7–1.0 V	0.8–1.4 V	0.8–1.2 V	0.5 V	1 V
Number of Spins	560	1440	1536	240	72(1440 virtual)	440
Core Area	0.53 mm^2	0.44 mm^2	0.16 mm^2	–	0.95 mm^2	0.44 mm^2
TTS	1–10 μs	<20 ns	<100 ns	<180 ns	40 ms	50 ns estimated
Power / Spin	–	–	NA	151 mW	5 μW	340 μW

rank-1, the energy landscape has a single dominant basin, allowing rapid convergence driven by analog dynamics rather than stochastic exploration. The system is an un-clocked analog dynamical system where intrinsic device noise aids convergence. Therefore, the random number generator is not the primary bottleneck. Based on measured analog settling times and simulation, we estimate a time-to-solution (TTS) of approximately 50 ns (Table 3). However, this value is specific to low-rank instances, and TTS is strongly dependent on the structure and rank of the coupling matrix. All 440-spins were then utilized in a simulated annealing experiment of a Sherrington-Kirkpatrick spin-glass. Fig. 10 shows the energy of the system decreasing as the annealing proceeds. On-chip annealing temperature was controlled by V_{temp} as previously described in section II. Results from varying V_{temp} and its effect on the tanh slope is shown in Fig. 10. For demonstrating ground state approach using on chip annealing, we assigned random weights ranging from +127 to -127 across all 440 p-bits drawn from a Gaussian distribution resulting in a complex energy landscape. As shown in Fig. 10, the system's energy decreases as the temperature is gradually lowered from hot to cold, demonstrating successful

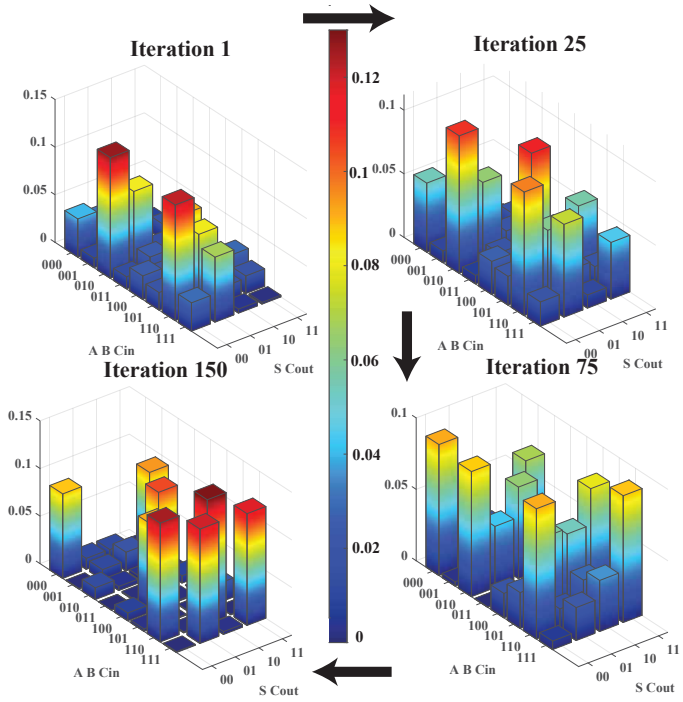

FIGURE 8. The learning of a harder distribution (Full Adder) despite variability

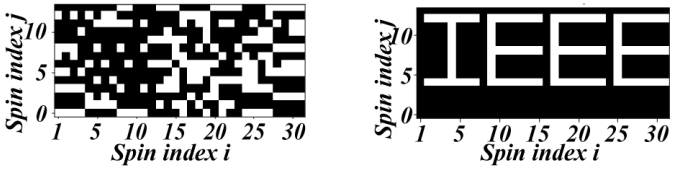

FIGURE 9. One-shot update of Max-Cut problem

convergence through annealing. The comparison table of our work compared to recently published Ising machines is shown in comparison table. Our chip achieves low-power consumption of 320μW per p-bit with stochastic neurons implementing a Boltzmann machine using a 200 MHz clock for the LFSRs. The closest comparison is the recently published all-digital p-bit chip [27]. which consumes 5μW/pbit with 2b weights at 0.5V and 10 MHz. Scaled to 1V and

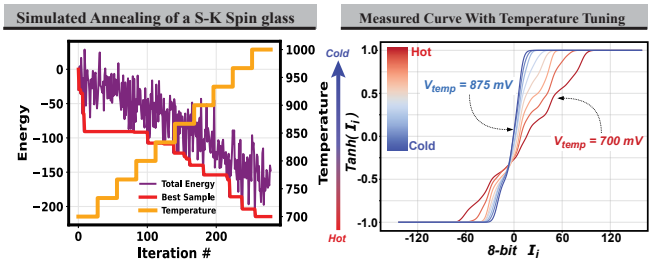

FIGURE 10. (a)On-chip simulated annealing of a 440-spin Sherrington–Kirkpatrick spin glass. System energy versus iteration during annealing with full spin-glass interactions enabled. (b) Independently measured tanh activation curves. Sweeping a spin bias from -127 to $+127$, with bias only applied and different V_{temp} to a spin.

CMOS NODE	Spin Array	Spin Size	Core Size	Supply	TTS	Power
65nm	440	76.03 μm^2	0.44mm^2	1.0V	<50ns	150mW

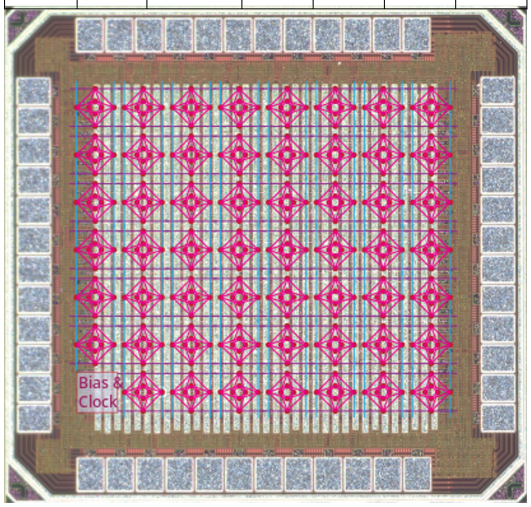

FIGURE 11. Chip Microphotograph with the Chimera Graph overlaid

200 MHz, this corresponds to approximately 400μW/pbit (with reduced precision). While the per-spin power in our implementation is higher than some low-frequency digital designs, power must be interpreted in conjunction with convergence time. Due to continuous-time analog dynamics, our system achieves significantly faster convergence, resulting in competitive energy-to-solution. Direct comparison with digital Ising machines is further complicated by differences in precision, architecture, and problem classes. However, two commensurate digital implementations with 8b coefficients [14], [21] consume 5.625 mW/spin (243nJ/180 ns) with a 166.67 MHz clock and 1.4μW/spin with a relatively long execution time (62.5 μs) for an "easy" max-cut problem running at 64MHz. The chip micrograph is presented in Fig 11.

IV. Conclusion

We demonstrated an all-CMOS p-computer implementing 440 p-bits with 8-bit weight precision, performing asynchronous Gibbs sampling—all within a compact 0.44mm^2 area. We successfully demonstrated the hardware-aware learning algorithm using both an AND gate and a full adder, achieving significant area reduction compared to traditional implementations. Additionally, we showed that our chip can efficiently solve combinatorial optimization problems, with time-to-solution reaching as low as 50 ns for simpler cases such as Max-Cut. While hardware aware training can partially correct the variation on-chip, we recommend more matching using current-mode bias distribution rather than a single voltage-mode bias for the entire chip. Another mitigation strategy is to incorporate a set of calibration bias weights to compensate for static offset. A split-capacitive

DAC can be an attractive approach, since capacitor matching is generally superior and can be compact in modern CMOS [22], [23].

Smaller CMOS nodes generally do not have the headroom to allow output impedance boosting and could lead to issues in current summation. In our design, we partially mitigate this by summing in to a low impedance diode-connected device keeping the drain-source voltage across all devices relatively equal. For larger number of summing nodes, a virtual ground node imposed by a high gain amplifier in feedback might be more appropriate. One could go even further by forcing the virtual ground value to be the voltage of the current mirroring device. These limitations highlight the need for hybrid approaches combining analog efficiency with digital calibration and control.

V. ACKNOWLEDGMENT

K.Y.C. and S.C. have been supported by an ONR-MURI grant N000142312708. JJ, DM, WW and LT would like to acknowledge support from the DARPA PIPES DARPA PIPES program under contract HR0011-19-C-0083. The views, opinions and/or findings expressed are those of the author and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government. Released under Distribution Statement "A" (Approved for Public Release, Distribution Unlimited).

REFERENCES

[1] K. Y. Camsari, R. Faria, B. M. Sutton, and S. Datta, "Stochastic p-bits for invertible logic," *Physical Review X*, vol. 7, no. 3, p. 031014, Jul. 2017.

[2] K. Y. Camsari, S. Chowdhury, and S. Datta, "Probabilistic computing with p-bits," *Nature Electronics*, vol. 2, pp. 587–593, Dec. 2019.

[3] J. Kaiser and S. Datta, "Probabilistic computing with p-bits," *Appl. Phys. Lett.*, vol. 119, no. 15, p. 150503, Oct. 2021, doi: 10.1063/5.0067927. ' *IEEE Journal on Exploratory Solid-State Computational Devices and Circuits*, vol. 3, no. 2, pp. 73–82, Dec. 2017.

[4] S. Chowdhury, A. Grimaldi, N. A. Aadit, S. Niazi, M. Mohseni, S. Kanai, H. Ohno, S. Fukami, L. Theogarajan, G. Finocchio, S. Datta, and K. Y. Camsari, "A full-stack view of probabilistic computing with p-bits: Devices, architectures, and algorithms," *IEEE J. Explor. Solid-State Comput. Devices Circuits*, vol. 9, no. 1, pp. 1–11, 2023, doi: 10.1109/JXCDC.2023.3256981.

[5] Y. Zhou, X. Hao, Q. Cai, L. Liao, and Z. Chen, "A reconfigurable Potts machine with successive boundary approximation annealing for solving combinatorial optimization problems," in *Proc. IEEE Custom Integr. Circuits Conf. (CICC)*, 2025, pp. 1–3, doi: 10.1109/CICC63670.2025.10982933.

[6] C. Shim, J. Bae, and B. Kim, "30.3 VIP-Sat: A Boolean satisfiability solver featuring 5×12 variable in-memory processing elements with 98% solvability for 50-variables 218-clauses 3-SAT problems," in *Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC)*, vol. 67, 2024, pp. 486–488, doi: 10.1109/ISSCC49657.2024.10454397.

[7] K. Y. Camsari, B. M. Sutton, and S. Datta, "p-bits for probabilistic spin logic," *Appl. Phys. Rev.*, vol. 6, no. 1, p. 011305, Mar. 2019.

[8] W. Whitehead, W. Oh, and L. Theogarajan, "CMOS single-photon avalanche diode circuits for probabilistic computing," *IEEE Journal on Exploratory Solid-State Computational Devices and Circuits*, vol. 10, pp. 49–57, 2024, doi: 10.1109/JXCDC.2024.3452030.

[9] J. Kaiser, W. A. Borders, K. Y. Camsari, S. Fukami, H. Ohno, and S. Datta, "Hardware-aware in situ learning based on stochastic magnetic tunnel junctions," *Physical Review Applied*, vol. 17, no. 1, p. 014016, Jan. 2022.

[10] W. Whitehead, Z. Nelson, K. Y. Camsari, and L. Theogarajan, "CMOS-compatible Ising and Potts annealing using single-photon avalanche diodes," *Nature Electronics*, vol. 6, no. 12, pp. 1009–1019, 2023.

[11] J. Bae, W. Oh, J. Koo, C. Yu, and B. Kim, "CTLE-Ising: A continuous-time latch-based Ising machine featuring one-shot fully parallel spin updates and equalization of spin states," *IEEE J. Solid-State Circuits*, vol. 59, no. 1, pp. 173–183, Jan. 2024.

[12] J. Bae *et al.*, "A 1440-spin continuous-time latch-based Ising machine," in *IEEE International Solid-State Circuits Conference (ISSCC) Dig. Tech. Papers*, Feb. 2024.

[13] J. Bae, C. Shim, and B. Kim, "15.6 e-Chimera: A scalable SRAM-based Ising macro with enhanced-Chimera topology for solving combinatorial optimization problems within memory," in *Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC)*, San Francisco, CA, USA, Feb. 2024, pp. 286–288.

[14] Y. Wu, J. Bae, C. Shim, and B. Kim, "m-Zephyr: A Digital In-Memory Ising Chip with 240 Spins Featuring Enhanced Connectivity Based on a Modified 3D Zephyr Topology," in *Proc. IEEE Symposium on VLSI Technology and Circuits*, pp. 1–3, 2025.

[15] K. Yamamoto *et al.*, "STATICA: A 512-spin 0.25M-weight annealing processor with an all-spin-updates-at-once architecture," *IEEE Journal of Solid-State Circuits*, vol. 56, no. 1, pp. 165–178, Jan. 2021.

[16] I. Ahmed *et al.*, "A probabilistic self-annealing compute fabric based on 560 coupled ring oscillators," in *IEEE Symposium on VLSI Circuits*, Jun. 2020.

[17] A. S. Abdelrahman, S. Chowdhury, F. Morone, and K. Y. Camsari, "Generalized probabilistic approximate optimization algorithm," *Nature Communications*, 2025.

[18] J. Koo and L. Theogarajan, "A 10 MHz to 3.2 GHz differential current-starved inverter-based self-biased adaptive bandwidth PLL in 65-nm CMOS," in *Proc. IEEE Int. Midwest Symp. Circuits and Systems (MWSCAS)*, Springfield, MA, USA, 2024, pp. 1244–1247.

[19] J. Lazzaro, S. Ryckebusch, M. A. Mahowald, and C. A. Mead, "Winner-Take-All Networks of O(N) Complexity," in *Advances in Neural Information Processing Systems*, vol. 1, D. Touretzky, Ed. San Francisco, CA, USA: Morgan-Kaufmann, 1988.

[20] E. Laskin and S. P. Voinigescu, "A 60 mW per lane, 4×23-Gb/s 2^7–1 PRBS generator," *IEEE Journal of Solid-State Circuits*, vol. 41, no. 10, pp. 2198–2208, Oct. 2006, doi: 10.1109/JSSC.2006.878112.

[21] 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," *IEEE Journal of Solid-State Circuits*, vol. 59, no. 8, pp. 2659–2670, Aug. 2024, doi: 10.1109/JSSC.2024.3352907.

[22] J. Koo and J.-Y. Sim,and L. Theogarajan,"A Parasitic and Mismatch Tolerant Fully Common-Centroided and Shielded Split-CDAC With Identical Unit Capacitors for SAR-ADC", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 34, no. 1, 2026, pp: 144 - 152, doi: 10.1109/TVLSI.2025.3619937

[23] Y.-H. Tsai and S.-I. Liu, "A 0.0067-mm 2 12-bit 20-MS/s SAR ADC using digital place-and-route tools in 40-nm CMOS", *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 30, no. 7, pp. 905–914, Jul. 2022 doi: 10.1109/TVLSI.2022.3170325

[24] W. Oh and L. Theogarajan, "AMS Layout Automation Using Circuit-Level Analog Standard Cells and Self-Biasing Techniques in Digital PnR Tools," in *Proc. 63rd ACM/IEEE Design Automation Conf. (DAC)*, 2026, (accepted).

[25] W. A. Borders, A. Z. Pervaiz, S. Fukami, K. Y. Camsari, H. Ohno, and S. Datta, "Integer factorization using stochastic magnetic tunnel junctions," *Nature*, vol. 573, no. 7774, pp. 390–393, Sep. 2019, doi: 10.1038/s41586-019-1557-9.

[26] M. Yamaoka, C. Yoshimura, M. Hayashi, T. Okuyama, H. Aoki, and H. Mizuno, "A 20k-Spin Ising Chip to Solve Combinatorial Optimization Problems With CMOS Annealing," *IEEE Journal of Solid-State Circuits*, vol. 51, no. 1, pp. 303–309, Jan. 2016.

[27] M.-C. Li *et al.*, "p-Circuits: Neither Digital Nor Analog," in *2025 IEEE International Solid-State Circuits Conference (ISSCC)*, San Francisco, CA, USA, 2025, pp. 1–3, doi:10.1109/ISSCC49661.2025.10904553.