Approximate and Stochastic Ising Machines

Tingting Zhang, *Member, IEEE,* Siting Liu, *Member, IEEE,* Honglan Jiang, *Member, IEEE,* Warren J. Gross, *Fellow, IEEE,* Fabrizio Lombardi, *Life Fellow, IEEE,* and Jie Han, *Senior Member, IEEE*

***Abstract*—The Ising model is useful in searching for (sub)-optimal solutions of combinatorial optimization problems (COPs). CMOS implementations of Ising model-based solvers, commonly referred to as Ising machines, provide reliable and accurate solutions with flexible and dense connectivities. However, they incur a significant hardware overhead. Approximate computing, as a low-power technique, offers a way to reduce hardware complexity, while stochastic computing is efficient in simulating the dynamics of the Ising model. The approximations introduced by these techniques may be beneficial in helping the system escape from local minima. In this article, we discuss the potential of using approximate and stochastic computing to improve the performance of Ising machines.**

***Index Terms*—Ising model, Ising machine, approximate computing, stochastic computing, annealing, p-bit.**

I. Introduction

With the rapid advancement of technology, the growing demands for data-intensive and energy-efficient computing pose challenges on conventional von Neumann architectures [1]. Combinatorial optimization problems (COPs) are often classified as non-deterministic polynomial-time (NP)-hard, such as the traffic flow management in smart city [2] and route planning in autonomous systems [3]. To solve these problems, the computational time and hardware of a conventional computer scale exponentially with the size of the problem. This limitation has motivated studies on building new computing architecture for improving performance in solving COPs.

The Ising model mathematically describes the energy of a system constructed with magnetic spins. Given the interactions between spins and the bias on a spin, the states of the spins determine the system energy, which evolves toward the ground state. This convergence closely mirrors the process of minimizing the objective function in COPs. Therefore, the solution search for a COP can be implemented as the exploration of spin states with the Ising model. An Ising model-based solver leads to the development of an Ising machine or Ising computer as a domain-specific architecture.

As shown in Fig. 1, solving a COP using an Ising machine follows three key steps [10]: (1) Problem Formulation: This step transforms a given COP into the Ising model using second-order polynomial representations; (2) Embedding Process: This step obtains the coupling coefficients and bias terms, which are compatible with the topology and precision constraints in the Ising machine; (3) Solution Search: This step explores the spin states by decreasing the energy. Table I summarizes the characteristics of emerging and conventional Ising machines, categorized by their underlying technologies. Compared to other types of Ising machines, the maturity of CMOS digital technology and the stability of its circuits provide high reliability and the potential for dense connectivity. High reliability ensures high solution quality, while dense connectivity reduces the cost of mapping the COP onto the topology of the Ising machine in Step (2). Due to intrinsic nonlinearity, the Ising machines based on some emerging low-power technologies, such as analog and memristive devices, face a major challenge in the linear modulation of the coupling between spins. CMOS digital Ising machines offer a better controlled and deterministic platform. In digital circuits, coupling values are encoded with fixed-point or floating-point representations. Although quantization and resource constraints inevitably introduce non-idealities especially when spin connectivity is dense, these effects can be mitigated through appropriate bit-width selection and pipelining.

Hereafter, this article focuses on CMOS digital Ising machines, which utilize different algorithms [11] to emulate the behavior of an Ising model by leveraging principles from statistical mechanics [12] or oscillator dynamics [13]. Despite these advantages, CMOS digital implementations face significant challenges related to high hardware overhead, primarily due to inherent technology limitations such as lim-

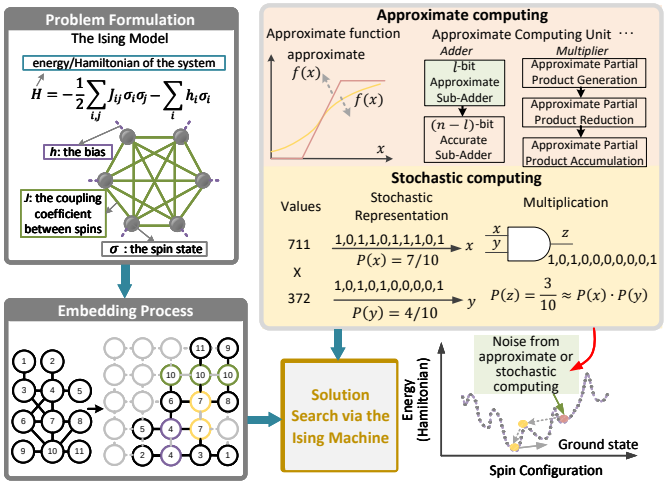


Fig. 1. Solving a combinatorial optimization problem using an Ising machine.

T. Zhang and W. J. Gross are with the Department of Electrical and Computer Engineering, McGill University, Montreal, H3A 0E9, Canada (e-mail: ttzhang@ualberta.ca, warren.gross@mcgill.ca).

S. Liu is with the School of Information Science and Technology, ShanghaiTech University, Shanghai, 201210, China (e-mail: liust@shanghaitech.edu.cn).

H. Jiang is with the Department of Micro-Nano Electronics, Shanghai Jiao Tong University, Shanghai, 200240, China (e-mail: honglan@sjtu.edu.cn).

F. Lombardi is with the Department of Electrical and Computer Engineering, Northeastern University, Boston MA 02115, USA (e-mail: lombardi@ece.neu.edu).

J. Han is with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, T6G 1H9, Canada (e-mail: jhan8@ualberta.ca).

TABLE I
CLASSIFICATION OF ISING MACHINES

Technology	**Quantum** [4]	**Optics** [5]	**Spintronics** [6]	**Memristor** [7]	**CMOS Analog** [8]	**CMOS Digital** [9]
Spin Type	Qubit	DOPO pulse	Magnetization states	Resistive/ conductive states	Voltages or currents	Digital bits
Coupling Type	Qubit flux	Optical interference or electronic feedback	Exchange or dipolar interactions	Conductive/capacitive/ resistive coupling	Capacitive/ resistive coupling	Memory+Logic
Diff. Den. Connec.	Moderate	High	High	Moderate	Moderate to high	Low
Diff. Coupl. Pre.	High	High	High	Moderate to high	Moderate	Low
Power	Low	Low	Low to moderate	Low to moderate	Moderate	High
Area	Small	Moderate	Small to moderate	Small to moderate	Moderate	Large
Energy	Low	Low	Low to moderate	Low to moderate	Moderate	High
Latency	Low	Low to moderate	Moderate	Moderate to high	Moderate	High
Accuracy	Moderate to low	Moderate	Moderate	Moderate	Moderate	High
Reliability	Low to moderate	Moderate	Moderate	Moderate	Moderate	High

Diff. Den. Connec.: The difficulty of achieving dense connectivity; Diff. Coupl. Pre.: The difficulty of extending coupling precision; DOPO: Degenerate optical parametric oscillator.

ited interconnects, process variability, thermal constraints, and leakage currents. Therefore, enhancing the hardware efficiency of CMOS digital implementations is crucial, particularly for dense spin connectivity. Ising machines are susceptible to becoming trapped in local minima; introducing controlled fluctuations can mitigate this issue by enabling the system to escape from suboptimal states. Hence, there has been growing interest in integrating CMOS digital Ising machines with emerging approaches such as approximate and stochastic computing to reduce hardware overhead and enhance the ability to escape from local minima, as shown in Fig. 1.

Approximate computing trades off accuracy for hardware efficiency gains, with techniques spanning over various levels [14]. For example, a complex function can be approximated by a simpler counterpart, and the arithmetic units can be implemented using lightweight circuits, such as OR gates for generating the sum in an addition. Stochastic computing is a computational framework that encodes numerical values as probabilities using stochastic bitstreams, in which the ratio of the number of '1's to the total number of bits can represent the encoded value [15]. Hence, arithmetic operations are carried out on these bitstreams using simplistic arithmetic circuits. Both approximate and stochastic computing introduce errors or fluctuations in signal values, which in turn generate noise that may facilitate the system in escaping from local minima. This article discusses how to efficiently leverage these techniques in various CMOS digital Ising machine architectures.

The remainder of this article is organized as follows. Section II provides an overview of the Ising model and various types of Ising machines. In Sections III and IV, hardware-efficient strategies leveraging approximate and stochastic computing for the Ising machine are discussed, respectively. Finally, Section V concludes the article and discusses prospects.

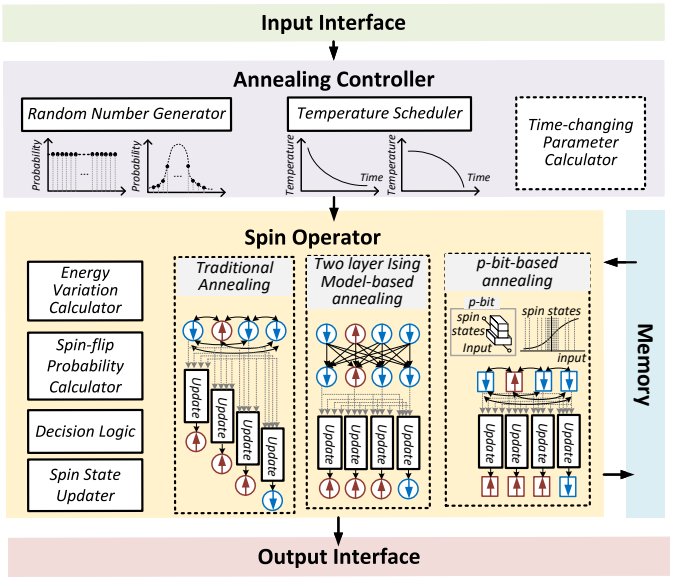
II. REVIEW

Ising machines differ from one another in their underlying architectures, which is shaped by the specific physical behavior they aim to emulate. Current CMOS digital Ising machines are broadly classified into two categories.

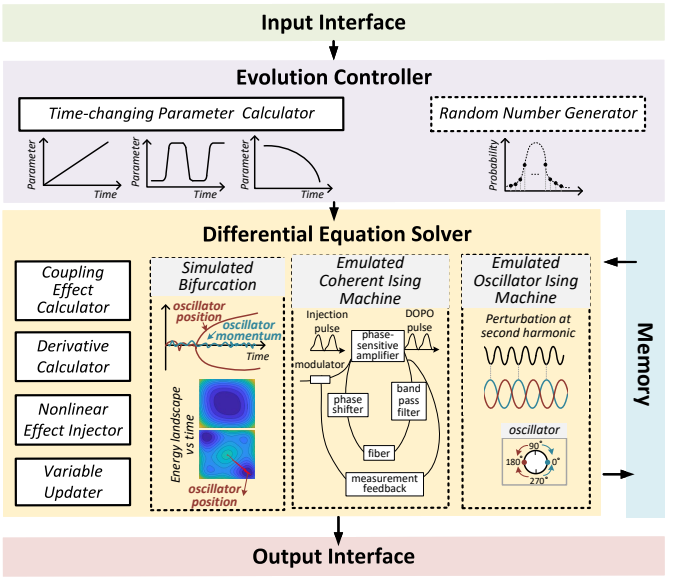
A. Annealing Ising Machines

Annealing Ising machines emulate the thermal annealing process in metallurgy, employing importance sampling-based Monte Carlo simulations to explore the energy landscape [12]. A conventional annealing Ising machine sequentially updates interconnected spins in a stochastic manner [16]. To enhance energy convergence, various methodologies have been developed. In an approach using the parallel-trial update in digital annealing, the feasibility of spin flips is assessed independently and in parallel, followed by the random selection and update of a flippable spin [17], [18]. Stochastic cellular automata annealing [9] and momentum annealing [19] utilize a two-layer structure. Spin replication without inter-spin or inter-replica interactions allows the simultaneous evaluation of all spin states. Another strategy referred to as parallel tempering employs multiple replicas of the system at different temperatures [20]. Simulated quantum annealing emulates the quantum fluctuations in annealing [21], [22]. Using probabilistic bits (denoted as p-bits) as spin states, p-bit based probabilistic computing [23], [24] shares a similar mechanism with annealing. It has driven the investigation on p-bit-based annealing Ising machines [25] that realize the parallel update of spins.

Fig. 2(a) shows a generic diagram for the annealing Ising machine. The annealing controller consists of several key components: a random number generator for determining the spin-flip probability, a temperature scheduler to regulate system temperature, and a time-changing parameter calculator for dynamically adjusting the annealing conditions. The spin operator evaluates whether a spin should flip at each simulation step. The energy variation calculator assesses the impact of a spin flip on the system's energy. This variation is then used by the spin-flip probability calculator to determine the transition likelihoods. The decision hardware ultimately examines whether spin states can be flipped, and the spin state updater finalizes the updated spin states at each simulation step. Note that the time-changing parameter calculator is only required in certain variants of annealing Ising machines, such as those for computing dynamic offsets [18] or modeling the self-interaction between duplicated spins [9]. For a p-bit annealing Ising machine, the spin operator is simplified to a weighted-



(a) Annealing Ising Machines.



(b) Dynamic Ising Machines.

Fig. 2. Different types of digital Ising machines.

sum calculator, an activation function logic module primarily implementing the tanh function, and a comparator.

B. Dynamic Ising Machines

A dynamic Ising machine digitally emulates the behavior of various oscillator networks, such as the Kerr-nonlinear parametric oscillators via the simulated bifurcation (SB) algorithm [26]–[30], the degenerate optical parametric oscillators (DOPOs) through emulated coherent Ising machines (ECIMs) [31], and the electronic nonlinear oscillators using emulated oscillator Ising machines (EOIMs) [32], [33]. Different from an annealing Ising machine, it is interpreted as numerical solvers for ordinary differential equations governing the oscillator dynamics. The discrete spin state (σ_i) is represented by a continuous spin variable (x_i) associated with the oscillator's position or phase. At the end of the evolution of spin variables, their signs determine the final spin states.

As shown in Fig. 2(b), a dynamic Ising machine consists primarily of an evolution controller and a differential equation solver. The evolution controller computes time-dependent parameters (for SB, ECIMs, and EOIMs) and generates random numbers (for ECIMs and EOIMs). Within the differential equation solver, the coupling effect calculator computes $\sum_j J_{ij}x_j + h_i$ for SB and ECIM, and $\sum_j J_{ij}c(x_i - x_j) + h_i$ for EOIM (where $c(\cdot)$ is a nonlinear activation function). The derivative calculator calculates the derivative information based on the time-dependent parameters and the coupling effect. The nonlinear effect injector introduces inelastic barriers for the oscillator position evolution in SB, and Gaussian noise from the random number generator for ECIMs and EOIMs. Finally, the variable updater changes the spin variable values accordingly.

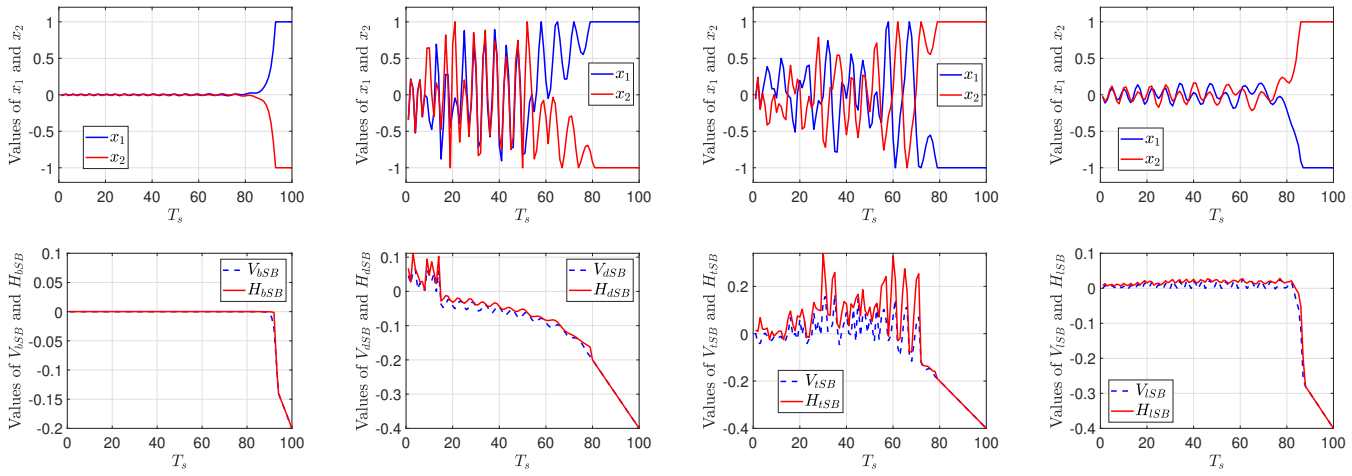
III. Approximate Ising Machines

The use of Ising machines in optimization is inherently approximate for several reasons: (1) COPs with constraints are typically formulated by embedding constraints into the objective function when mapped to the Ising model, which may increase the risk of converging to suboptimal solutions. (2) The energy landscape of many real-world COPs is highly complex, with numerous local minima. The difficulty of reaching the ground state increases with problem size. (3) Hardware implementations introduce approximations due to the finite precision in hardware and limited spin connectivity, often necessitating approximate embeddings of COPs into the Ising machine. Therefore, Ising machines can be regarded as error-tolerant systems, making them well-suited for approximate and stochastic computing. This section discusses multi-level strategies aimed at optimizing the hardware design for digital Ising machines, spanning from data representation to the circuit level.

A. Data Representation-level Approximation

Quantization is often employed when designing computing architectures. By discretizing values into a finite set of quantized levels, arithmetic operations are greatly simplified, resulting in reduced hardware. Uniform quantization is widely used in building CMOS digital Ising machines, in which the quantization levels are determined through experimental investigations or by analyzing the data distribution. In particular, quantization provides even greater benefits in dynamic Ising machines, where continuous spin variables are used to represent spin states.

Given the SB Ising machine as an example, the position values are frequently updated through multiply-and-accumulate (MAC) operations, which dominate the computing hardware. Binary, ternary, and logarithmic quantization strategies have been investigated for hardware savings in MAC by quantizing the position value x_j to $q(x_j)$ for computing $\sum_j J_{ij}x_j$ as $\sum_j J_{ij}q(x_j)$ [27], [28]. Fig. 3 shows the performance of using different quantization strategies for the MAC operations in SB Ising machines, compared to using full precision ones,



(a) Full precision. (b) Binary quantization. (c) Ternary quantization. (d) Logarithmic quantization.

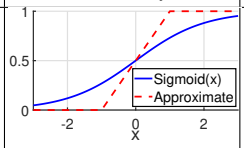
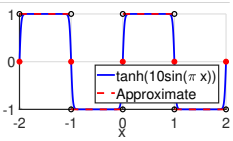
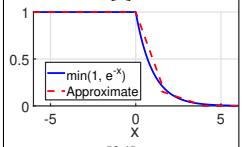
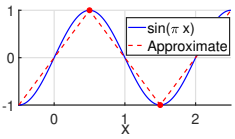
Fig. 3. Performance of quantization strategies for SB Ising machines in terms of position values (x_1 and x_2) versus the time step (T_s) as well as the potential energy (V) and Hamiltonian (H) versus the time step (T_s) on an example two-spin Ising problem with coupling coefficients $J_{12} = J_{21} = -1$. (a) ballistic SB (bSB) uses full precision x_j [27]; (b) dSB uses binary quantization, which discretizes x_j to $+1$ when x_j is positive or to -1 when x_j is negative [27]; (c) tSB uses ternary quantization, which discretizes x_j to 0 when the absolute value of x_j is no larger than a threshold, which linearly increases from 0 to 1, or otherwise to ± 1 like binary quantization [28]; and (d) lSB uses logarithm quantization with nine quantization levels, which rounds up x_j to quantized values logarithmically spaced within $[-1, 1]$ [28].

in terms of evolutions of position values on an example two-spin Ising problem. Due to the use of binarization or ternarization, the position values significantly fluctuate early, as shown in Figs. 3(b) and (c). As shown in Fig. 3(d), logarithmic quantization results in smaller changes in the position values with time. Fig. 3 also gives the potential energy and Hamiltonian versus the time step. For the ternary SB (tSB) and discrete SB (dSB) systems, the Hamiltonian and potential energy fluctuate significantly at the beginning and then gradually decrease with the time step. The logarithmically quantized SB (lSB) system shows a small fluctuation at the start and then quickly reaches the lowest Hamiltonian value. Compared with using full precision, as shown in Fig. 3(a), the use of quantized position values also introduces inherent noise. This noise may act as a form of stochastic perturbation, aiding the system in further exploring the solution space and avoiding local minima. However, it may also increase the search time, because additional iterations are required to thoroughly explore more local minima. For example, when applying ternarization in an SB Ising machine, it enhances the performance when the number of time steps is $10k$, which achieves a high probability of obtaining 99.9% of the best-known value for a 2000-spin max-cut problem [28].

B. Function-Level Approximation

Function-level approximation in Ising machines offers a compelling approach to mitigate the significant hardware overhead associated with implementing computationally intensive functions. Critical components, such as the spin-flip probability function in annealing Ising machines [9], [16], [35], [36], and the second harmonic injection locking (SHIL) function for computing the coupling effect in EOIM [32], [33], involve complex mathematical operations, including the computation of exponential or nonlinear functions. These operations impose substantial demands on hardware. To address these challenges, function-level approximation seeks to replace precise but resource-intensive implementations with simplified or approximate models that capture the essential behavior of the original functions.

TABLE II
FUNCTION-LEVEL APPROXIMATION IN CMOS DIGITAL ISING MACHINES

Ising Machine Type	**Annealing Ising Machine**	**Emulated Oscillator Ising Machine**
Functions	Spin (Non-)Flip Probability	Coupling Effect
Accurate vs Approximate Function	 [9]	 [33]
	 [34]	 [33]

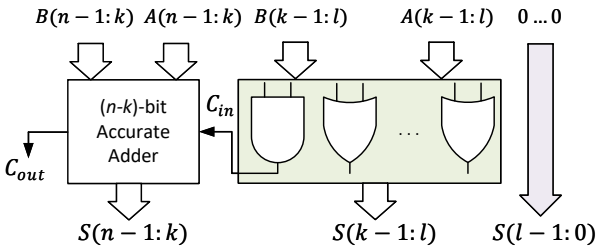
In most annealing Ising machines, the spin flip probability function follows the Boltzmann distribution, which involves computationally expensive exponential functions. In some variants of annealing Ising machines, the spin non-flip probability must be calculated instead, requiring the sigmoid function to be computed. These nonlinear functions are critical for simulating the probabilistic nature of spin dynamics during the annealing process. In contrast, the coupling effects are implemented in EOIMs through more complex mathematical expressions, often involving nonlinear functions such as the sine (sin) and hyperbolic tangent (tanh). These functions

are used to model the dynamic behavior of oscillator-based systems.

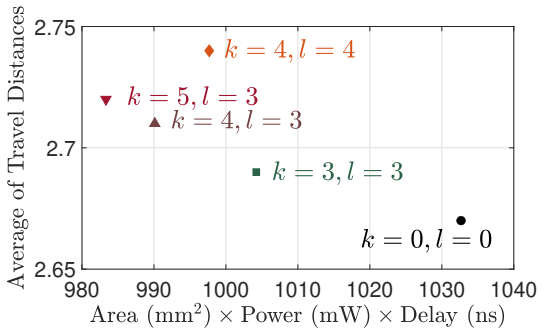
As summarized in Table II, the computational challenges posed by these nonlinear functions, particularly in hardware implementations [9], [33], [34], have led to the exploration of function-level linear approximations. A common approach is to simplify these nonlinear functions into piecewise linear ones, so the original functions are replaced with segmented linear representations. Piecewise linear approximation significantly reduces the complexity of complex functions, making it easier to implement in hardware while achieving an acceptable accuracy. In [36], a hybrid approach combining Taylor series expansion and LUT-based approximation was employed to enhance the accuracy of sigmoid function approximation. It was reported in [33] that an accuracy of 99.1% can be achieved when solving max-cut problems.

C. Circuit-Level Approximation

Circuit-level approximation focuses on optimizing the physical hardware implementation of computational processes in Ising machines. It directly targets the resource-intensive nature of hardware. Some published works are aimed at saving the hardware of random number generators (RNGs), such as introducing randomness by the variability in the minimum operating voltage of static random access memory (SRAM) [37]. Approximation strategies have also been investigated for the hardware-consuming energy variation calculation. To extend the numerical range when solving large-scale complex problems, half-precision floating-point coefficients are utilized in [38], [39]. However, this approach incurs high hardware overhead due to floating-point arithmetic operations. To mitigate this issue, low-cost floating-point logarithmic multipliers are employed in the SB machine [39], which, interestingly, shows improved accuracy for some datasets of max-cut problems. Furthermore, an approximate adder, known as the lower-part-OR and truncated adder (LOTA), is utilized in [40] for the mantissa addition in an annealing-based Ising machine. In this design, the k least significant bits of the mantissa adder are approximated by truncating the least significant l bits and applying OR gates to the remaining $(k-l)$ bits. Fig. 4 shows the LOTA and the performance of solving an example 8-city traveling salesman problem (TSP) using a 64-spin annealing Ising machine with LOTA. A TSP finds the shortest route that visits each city exactly once. Compared to its accurate counterpart, the use of LOTA ($k=4, l=3$) results in a 1.4% degradation of solution quality but saves 4.1% hardware.



(a) A lower-part-OR and truncated adder [40].



(b) Average of travel distance versus area×power×delay [40].

Fig. 4. Using approximate adders in an annealing Ising machine. Hardware simulation results are obtained by using the Synopsys Design Compiler. A CMOS 28 nm technology is applied with a supply voltage of 1.0 V, a temperature of 25°C, and a clock frequency of 200 MHz. Some data in the figure are reported in [40].

IV. Stochastic Ising Machines

The use of stochastic computing necessitates architectural changes. As shown in Fig. 5, a stochastic computing system typically consists of three components: stochastic number generators (SNGs), stochastic computing elements, and a signal reconstruction unit. In classical stochastic computing, a bitstream with 50% being '1's, generated by using an RNG and a comparator, can encode a value of 0.5. This probabilistic representation enables arithmetic and logical operations to be performed with remarkable simplicity using standard logic gates. For examples, when using unipolar encoding, multiplication can be implemented using a single AND gate, while addition can be realized using multiplexers; an integrator can be built by using a counter and a SNG. This section discusses the probabilistic nature and hardware efficiency of novel stochastic computing paradigms in the construction of the fundamental components of an Ising machine.

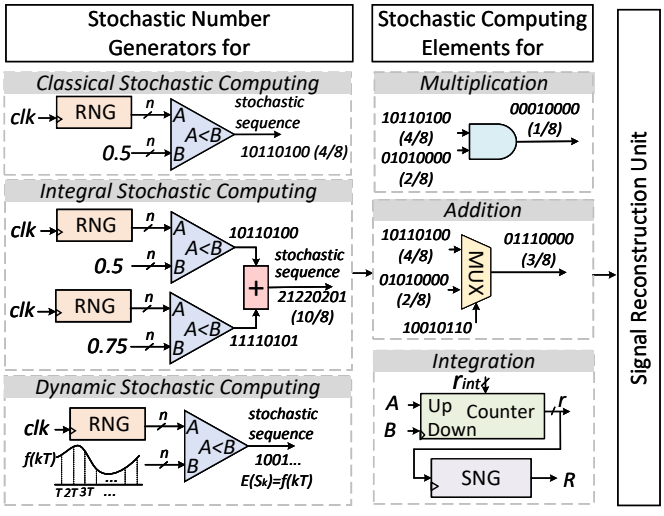
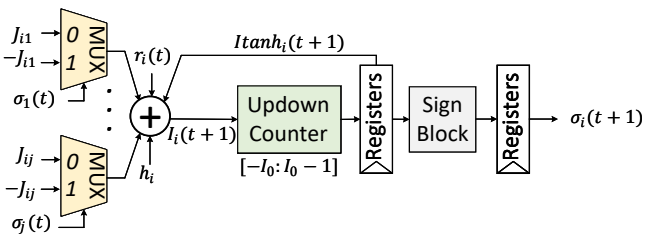


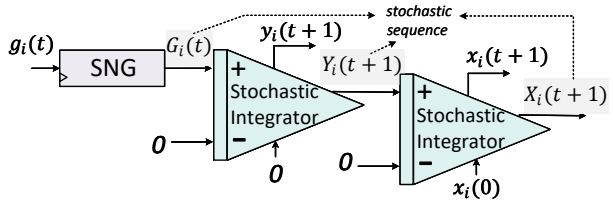
Fig. 5. Stochastic computing components.

A. Integral Stochastic Computing (ISC)-based Ising Machines

For a p-bit-based annealing Ising machine, p-bits can be implemented as bitstreams, in which each bit is a sample from



(a) Stochastic p-bit [41].



(b) Stochastic SB cell [42].

Fig. 6. Stochastic computing cells for CMOS digital Ising machines.

a Bernoulli sequence with a probability for the spin's state. The p-bit (spin) operator first evaluates $\sum_j J_{ij}\sigma_j + h_i$. The result is then scaled by a time-decreasing parameter (acting as the temperature), I_0, for the activation function logic, which implements the hyperbolic tangent ($\tanh$) function. The updated spin states are determined by comparing the activation output to a random number. To represent values with a larger range, ISC [43] is utilized to compute the p-bit function [41]. In ISC, a stochastic stream is obtained by accumulating the conventional stochastic signals, which is a sequence of integer numbers. A real number larger than 1 is first represented as a summation of multiple numbers in the interval $[0, 1]$. As shown in Fig. 5, a real number 1.25 can be implemented as the addition of the stochastic bitstreams for 0.5 and 0.75. In this way, the stochastic multiplication and addition can be implemented by using a binary multiplier and adder.

As shown in Fig. 6(a), the p-bit based on ISC is constructed by using multiplexors (MUXs) for multiplication of $\sigma_i \cdot J_{ij}$, an adder for addition for computing $\sum_j J_{ij}\sigma_j + h_i$, a counter for implementing the $\tanh(\sum_j J_{ij}\sigma_j + h_i)$ function, a sign block for identifying the spin states and some registers for storing the values. In particular, the $\tanh$ function is approximated by using a saturated up-down counter. The output $I_i(t+1)$ is truncated with a range of $2I_0$ states to obtain $Itanh_i(t+1)$, ideally equivalent to $\tanh(I_0 \cdot (\sum_j J_{ij}\sigma_j + h_i))$, where I_0 is a parameter that decreases with time. It has been reported in [41] that the ISC-based Ising machine shows faster convergence compared to the conventional annealing method, achieving an annealing time of 1.00 ms for an example max-cut problem.

B. Dynamic Stochastic Computing (DSC)-based Ising Machines

A dynamic Ising machine inherently functions as a differential equation solver. For example, the SB Ising machine is essentially to solve a pair of differential equations related to oscillator positions and momenta by using Euler integration. Unlike conventional stochastic computing, DSC encodes changing signals by stochastic sequences. The probability of each bit being '1' is equivalent to the corresponding sample value from the digital signal [44]. Taking the stochastic integrator as shown in Fig. 5, as the basic building element, DSC provides an efficient means of implementing accumulation-based iterative computation. The efficiency of DSC in performing integration has motivated research into its application for implementing the integration in dynamic Ising machines.

Fig. 6(b) shows the circuit diagram of a stochastic SB cell, which updates the momentum and position values by taking the gradient information as the input [42]. For the ith spin, after converting the gradient information in the tth time step ($g_i(t)$) to its dynamic stochastic sequence ($G_i(t)$), a stochastic integrator is used to obtain the updated momentum value ($y_i(t+1)$) and a dynamic stochastic sequence ($Y_i(t+1)$). Then, the second stochastic integrator updates the position value ($x_i(t+1)$) encoded in a dynamic stochastic sequence ($X_i(t+1)$). The stochastic SB cell can be used as the basic building block in an SB Ising machine. To further improve the power efficiency, one of the two stochastic integrators in Fig. 6(b) can be replaced by a binary Euler integrator. Quantization in the MAC operation within SB not only introduces beneficial noise, but it also reduces hardware complexity, as discussed in Section III-A. Random ternary quantization is considered for the stochastic integrator-based SB Ising machine, in which the stochastic sequence for position values from the stochastic SB cell is used as the randomly ternarized position values for MAC computation for hardware savings. Figure 7 further illustrates the dynamics of a two-spin SB system with random ternary quantization. Compared to the ternary quantization results shown in Fig. 3(c), random ternary quantization exhibits lower fluctuations in the initial phase, indicating a higher likelihood of becoming trapped in local minima. Consequently, it becomes challenging to achieve further improvements in solution quality over extended search time, compared to the original ternary quantization. Nevertheless, employing one or two stochastic integrators to construct a stochastic SB cell for a 2000-spin system results in at least a 44% reduction in power consumption and a 1.19× speedup, while achieving higher solution quality over extended searches, compared to conventional SB machines [42]. Utilizing a stochastic integrator and a binary Euler integrator reduce power consumption by 12% but it requires 1% larger area than a design using two stochastic integrators [42]. Note that while decorrelators and correlators play crucial roles in conventional stochastic arithmetic circuits, particularly in multi-level architectures like cascaded multiplication circuits, the processing element (i.e., the stochastic integrator) performs integration through temporal averaging rather than relying on correlation/decorrelation manipulation between bitstreams. This architectural distinction eliminates the need for explicit correlators or decorrelators in the implementation. Furthermore, the implementation intentionally leverage fully correlated sequences generated by a shared RNG across processing elements. This design choice not only maintains computational fidelity but also substantially reduces hardware overhead by eliminating redundant RNG circuits.

TABLE III
APPROXIMATE AND STOCHASTIC ISING MACHINES

Ising Machines			Approximate / Stochastic Computing	Platform / Technology	# of Spins	Topology	Coupling Precision	Frequency	Power per Spin	Area per Spin	Problems
Annealing	SA	[16]	Linear Approximation	CMOS 40 nm	$2 \times 30k$	King's graph[a]	3 bit	100 MHz	-	788 μm^2	MCP
		[36]	Taylor-LUT Hybrid Approximation	Virtex Ultrascale+	$11k$	Modified Complete[b]	8 bit	125 MHz	1.8 mW	-	TSP
	PA	[38]	Approximate Adder/ Linear Approximation	CMOS 28 nm	64	Complete	16 bit[c]	200 MHz	0.6 mW	0.09 mm^2	TSP
		[9]	Linear Approximation	CMOS 65 nm	512	Complete	5 bit	320 MHz	1.2 mW	0.01 mm^2	MCP
	PbA	[41]	Integral Stochastic Computing	Xilinx Kintex-7	800	2D lattice[d]	4 bit	100 MHz	2.6 mW	-	MCP
Dynamic	SB	[42]	Quantization/Dynamic Stochastic Computing	CMOS 40 nm	$2k$	Complete	2 bit	250 MHz	0.64 mW	0.06 mm^2	MCP
		[45]	Quantization	Virtex UltraScale+	$2k$	Complete	8 bit	200 MHz	1.9 mW	-	MCP
	EOIM	[32]	Linear Approximation	TSMC 65 nm	33	Complete	-	120 MHz	9.1 mW	0.09 mm^2	MIMOD

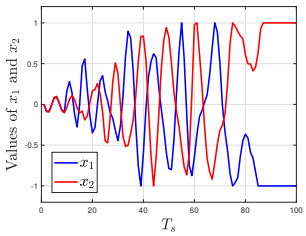
a: Each spin interacts with 8 spins; b: Modified layer-by-layer fully-connected topology; c: Floating-point; d: Each spin interacts with 4 spins; SA: Simulated annealing; PA: Parallel annealing based on the two-layer Ising model; PbA: p-bit-based annealing; SB: Simulated bifurcation; EOIM: Emulated oscillator Ising machine; #: The number of; MCP: Max-cut problems; TSP: Travelling salesman problems; MIMOD: Multi-input multi-output detection.

V. CONCLUSION AND PROSPECTS

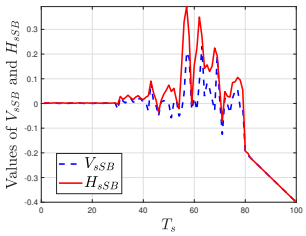
In this article, we discuss the role of approximate and stochastic computing in the design of digital Ising machines. The simplicity of stochastic logic operations and hardware-friendly approximation techniques enable the development of compact hardware architectures.

Table III shows the main features of the state-of-the-art approximate and stochastic Ising machines. Approximate computing techniques have been applied at various design levels to achieve hardware savings. A key challenge is to efficiently integrate approximation strategies across these various levels by taking into account their interdependencies to improve the overall efficiency. Classical stochastic computing often relies on random number generation and long bit-streams, leading to increased energy consumption and latency, which in turn limit scalability. Innovative stochastic Ising machines using integral and dynamic stochastic computing mitigate the long bit-stream issue. However, the need for random number generation is not eliminated.

Approximate and stochastic computing inherently introduce errors, which can impact the convergence and solution quality of Ising machines. In some cases, the introduced noise can enhance the system's ability to escape from local minima, such as using quantization in the simulated bifurcation Ising machine and approximate functions in the emulated oscillator Ising machine. However, it may cause a loss in solution quality, such as the annealing Ising machine using approximate adders. This indicates that different types of Ising machines exhibit varying degrees of error resilience. The effectiveness of approximate and stochastic computing in Ising machines depends on how well these techniques align with the mechanisms of Ising machines. Since the energy landscape varies across different optimization problems, the effects of approximate and stochastic computing strategies differ significantly. Therefore, developing application-driven approximate and stochastic Ising machines is crucial to enhance their practicality for industry.



(a) Position values (x_1 and x_2) versus the time step (T_s).



(b) The potential energy (V_{sSB}) and Hamiltonian (H_{sSB}) versus the time step (T_s).

Fig. 7. Performance analysis of the SB Ising machine using stochastic SB cells and random quantization on an example two-spin Ising problem with coupling coefficients $J_{12} = J_{21} = -1$.

ACKNOWLEDGMENT

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada under Grant RES0048688, Grant RES0051374 and Grant RES0054326, Alberta Innovates Grant RESS0053965, and in part by the National Science Fund of China under Grant 62204155.

REFERENCES

[1] T. N. Theis and H.-S. P. Wong, "The end of moore's law: A new beginning for information technology," *Computing in science & engineering*, vol. 19, no. 2, pp. 41–50, 2017.

[2] B. Sridhar, S. R. Grabbe, and A. Mukherjee, "Modeling and optimization in traffic flow management," *Proceedings of the IEEE*, vol. 96, no. 12, pp. 2060–2080, 2008.

[3] A.-E. Taha and N. AbuAli, "Route planning considerations for autonomous vehicles," *IEEE Communications Magazine*, vol. 56, no. 10, pp. 78–84, 2018.

[4] A. D. King, J. Raymond, T. Lanting, R. Harris, A. Zucca *et al.*, "Quantum critical dynamics in a 5,000-qubit programmable spin glass," *Nature*, vol. 617, pp. 61–66, 2023.

[5] T. Honjo, T. Sonobe, K. Inaba, T. Inagaki, T. Ikuta *et al.*, "100,000-spin coherent Ising machine," *Sci. Adv.*, vol. 7, no. 40, p. eabh0952, 2021.

[6] A. Grimaldi, L. Sánchez-Tejerina, N. A. Aadit, S. Chiappini, M. Carpentieri *et al.*, "Spintronics-compatible approach to solving maximum-satisfiability problems with probabilistic computing, invertible logic, and parallel tempering," *Phys. Rev. Appl.*, vol. 17, no. 2, p. 024052, 2022.

[7] X. Chen, D. Yang, G. Hwang, Y. Dong, B. Cui, D. Wang, H. Chen, N. Lin, W. Zhang, H. Li *et al.*, "Oscillatory neural network-based Ising machine using 2D memristors," *ACS nano*, vol. 18, no. 16, pp. 10 758–10 767, 2024.

[8] J. Vaidya, R. Surya Kanthi, and N. Shukla, "Creating electronic oscillator-based Ising machines without external injection locking," *Sci. Rep.*, vol. 12, no. 1, p. 981, 2022.

[9] K. Yamamoto, K. Kawamura, K. Ando, N. Mertig, T. Takemoto *et al.*, "STATICA: A 512-spin 0.25 m-weight annealing processor with an all-spin-updates-at-once architecture for combinatorial optimization with complete spin–spin interactions," *IEEE JSSC*, vol. 56, no. 1, pp. 165–178, 2020.

[10] T. Zhang, Q. Tao, B. Liu, A. Grimaldi, E. Raimondo, M. Jiménez, M. J. Avedillo, J. Nunez, B. Linares-Barranco, T. Serrano-Gotarredona *et al.*, "A review of Ising machines implemented in conventional and emerging technologies," *IEEE Transactions on Nanotechnology*, vol. 23, pp. 704 – 717, 2024.

[11] T. Zhang, Q. Tao, B. Liu, and J. Han, "A review of simulation algorithms of classical Ising machines for combinatorial optimization," in *ISCAS*. IEEE, 2012, pp. 1–5.

[12] S. Kirkpatrick, "Optimization by simulated annealing: Quantitative studies," *J. Stat. Phys.*, vol. 34, pp. 975–986, 1984.

[13] H. Goto, "Bifurcation-based adiabatic quantum computation with a nonlinear oscillator network," *Sci. Rep.*, vol. 6, no. 1, p. 21686, 2016.

[14] H. Jiang, F. J. H. Santiago, H. Mo, L. Liu, and J. Han, "Approximate arithmetic circuits: A survey, characterization, and recent applications," *Proceedings of the IEEE*, vol. 108, no. 12, pp. 2108–2135, 2020.

[15] B. R. Gaines, "Stochastic computing systems," *Adv. Inf. Syst. Sci.: Volume 2*, pp. 37–172, 1969.

[16] T. Takemoto, M. Hayashi, C. Yoshimura, and M. Yamaoka, "A 2×30k-spin multi-chip scalable CMOS annealing processor based on a processing-in-memory approach for solving large-scale combinatorial optimization problems," *IEEE JSSC*, vol. 55, no. 1, pp. 145–156, 2019.

[17] S. Tsukamoto, M. Takatsu, S. Matsubara, and H. Tamura, "An accelerator architecture for combinatorial optimization problems," *Fujitsu Sci. Tech. J*, vol. 53, no. 5, pp. 8–13, 2017.

[18] S. Matsubara, M. Takatsu, T. Miyazawa, T. Shibasaki, Y. Watanabe *et al.*, "Digital annealer for high-speed solving of combinatorial optimization problems and its applications," in *ASP-DAC*. IEEE, 2020, pp. 667–672.

[19] T. Okuyama, T. Sonobe, K.-i. Kawarabayashi, and M. Yamaoka, "Binary optimization by momentum annealing," *Phys. Rev. E*, vol. 100, no. 1, p. 012111, 2019.

[20] M. Aramon, G. Rosenberg, E. Valiante, T. Miyazawa, H. Tamura, and H. G. Katzgraber, "Physics-inspired optimization for quadratic unconstrained problems using a digital annealer," *Front. Phys.*, vol. 7, p. 48, 2019.

[21] T. Okuyama, M. Hayashi, and M. Yamaoka, "An Ising computer based on simulated quantum annealing by path integral Monte Carlo method," in *ICRC*. IEEE, 2017, pp. 1–6.

[22] A. Grimaldi, K. Selcuk, N. A. Aadit, K. Kobayashi, Q. Cao *et al.*, "Experimental evaluation of simulated quantum annealing with MTJ-augmented p-bits," in *IEDM*. IEEE, 2022, pp. 22.4.1–22.4.4.

[23] J. Kaiser and S. Datta, "Probabilistic computing with p-bits," *Applied Physics Letters*, vol. 119, no. 15, 2021.

[24] N. A. Aadit, A. Grimaldi, M. Carpentieri, L. Theogarajan, J. M. Martinis, G. Finocchio, and K. Y. Camsari, "Massively parallel probabilistic computing with sparse Ising machines," *Nature Electronics*, vol. 5, no. 7, pp. 460–468, 2022.

[25] N. A. Aadit, M. Mohseni, and K. Y. Camsari, "Accelerating adaptive parallel tempering with FPGA-based *p*-bits," in *2023 IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits)*. IEEE, 2023, pp. 1–2.

[26] H. Goto, K. Tatsumura, and A. R. Dixon, "Combinatorial optimization by simulating adiabatic bifurcations in nonlinear Hamiltonian systems," *Sci. Adv.*, vol. 5, no. 4, p. eaav2372, 2019.

[27] H. Goto, K. Endo, M. Suzuki, Y. Sakai, T. Kanao *et al.*, "High-performance combinatorial optimization based on classical mechanics," *Sci. Adv.*, vol. 7, no. 6, p. eabe7953, 2021.

[28] T. Zhang and J. Han, "Quantized simulated bifurcation for the Ising model," in *NANO*. IEEE, 2023, pp. 715–720.

[29] T. Zhang, Q. Tao, and J. Han, "Solving traveling salesman problems using Ising models with simulated bifurcation," in *ISOCC*. IEEE, 2021, pp. 288–289.

[30] T. Zhang and J. Han, "Efficient traveling salesman problem solvers using the Ising model with simulated bifurcation," in *DATE*. IEEE, 2022, pp. 548–551.

[31] E. S. Tiunov, A. E. Ulanov, and A. Lvovsky, "Annealing by simulating the coherent Ising machine," *Opt. Express*, vol. 27, no. 7, pp. 10 288–10 295, 2019.

[32] S. Sreedhara, J. Roychowdhury, J. Wabnig, and P. Srinath, "Digital emulation of oscillator Ising machines," in *DATE*. IEEE, 2023, pp. 1–2.

[33] B. Liu, T. Zhang, X. Gao, and J. Han, "An efficient simulated oscillator-based Ising machine on FPGAs," in *NANO Conf.* IEEE, 2024, pp. 469–474.

[34] Y. Zhang, X. Wang, D. Jiang, Z. Huang, G. Fan, and E. Yao, "A parallel tempering processing architecture with multi-spin update for fully-connected Ising models," in *2024 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2024, pp. 1–6.

[35] A. Lu, J. Hur, Y.-C. Luo, H. Li, D. E. Nikonov, I. Young, Y.-K. Choi, and S. Yu, "Scalable in-memory clustered annealer with temporal noise of finfet for the travelling salesman problem," in *2022 International Electron Devices Meeting (IEDM)*. IEEE, 2022, pp. 22–5.

[36] Z. Huang, Y. Zhang, X. Wang, D. Jiang, and E. Yao, "DCAP: A scalable decoupled-clustering annealing processor for large-scale traveling salesman problems," *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2024.

[37] 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 JSSC*, vol. 51, no. 1, pp. 303–309, 2016.

[38] Q. Tao, T. Zhang, and J. Han, "An approximate parallel annealing Ising machine for solving traveling salesman problems," *IEEE ESL*, vol. 15, no. 4, pp. 226–229, 2023.

[39] T. Zhang, "Design and applications of simulated bifurcation Ising machines," Ph.D. dissertation, University of Alberta, 2024.

[40] Q. Tao, T. Zhang, and J. Han, "Approximate parallel annealing Ising machines (APAIMs): Controller and arithmetic design," in *CNNA*. IEEE, 2023, pp. 1–5.

[41] D. Shin, N. Onizawa, W. J. Gross, and T. Hanyu, "Memory-efficient FPGA implementation of stochastic simulated annealing," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 13, no. 1, pp. 108–118, 2023.

[42] T. Zhang, H. Zhang, Z. Yu, S. Liu, and J. Han, "A high-performance stochastic simulated bifurcation Ising machine," in *DAC*. ACM, 2024.

[43] A. Ardakani, F. Leduc-Primeau, N. Onizawa, T. Hanyu, and W. J. Gross, "VLSI implementation of deep neural network using integral stochastic computing," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 10, pp. 2688–2699, 2017.

[44] S. Liu, W. J. Gross, and J. Han, "Introduction to dynamic stochastic computing," *IEEE Circuits Syst. Mag.*, vol. 20, no. 3, pp. 19–33, 2020.

[45] T. Zhang and J. Han, "Qsbms: Lightweight quantized simulated bifurcation Ising machines," *IEEE Transactions on Nanotechnology*, pp. 1–12, 2025.