THE CHIPS TO SYSTEMS CONFERENCE

SHAPING THE NEXT GENERATION OF ELECTRONICS

JUNE 23-27, 2024

MOSCONE WEST CENTER
SAN FRANCISCO, CA, USA

THE CHIPS TO SYSTEMS CONFERENCE

61

SHAPING THE NEXT GENERATION OF ELECTRONICS

JUNE 23-27, 2024

MOSCONE WEST CENTER
SAN FRANCISCO, CA, USA

上海科技大学
ShanghaiTech University

UNIVERSITY OF ALBERTA

A High-Performance Stochastic Simulated Bifurcation Ising Machine

Tingting Zhang, Hongqiao Zhang, Zhengkun Yu, Siting Liu*, Jie Han

Outline

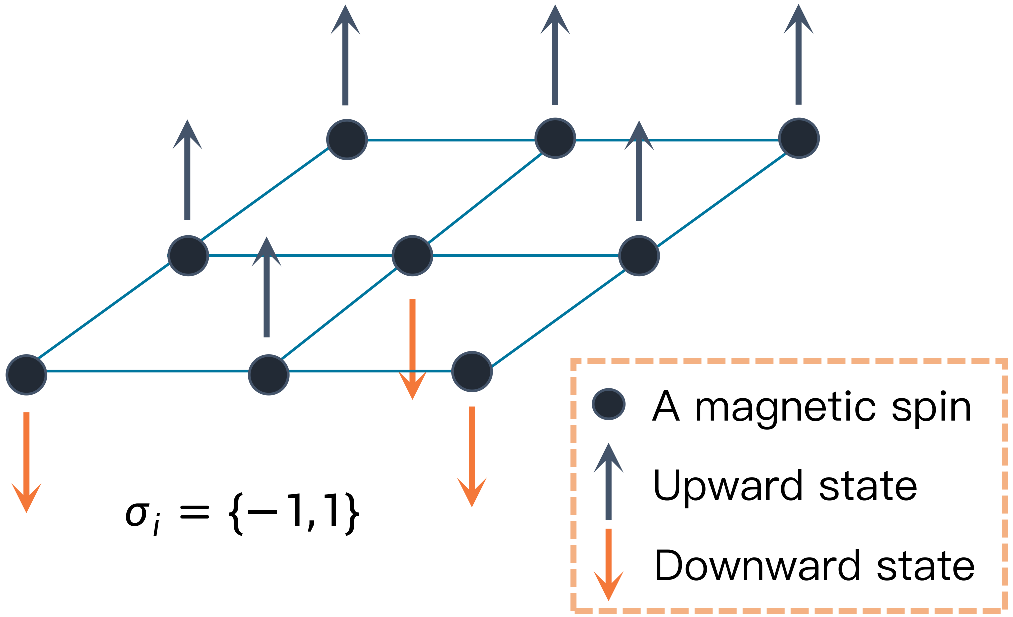
- Background
  - Ising machine
  - (Dynamic) stochastic computing
- Formulation and circuit design
- System design
- Experiments and results
- Conclusion

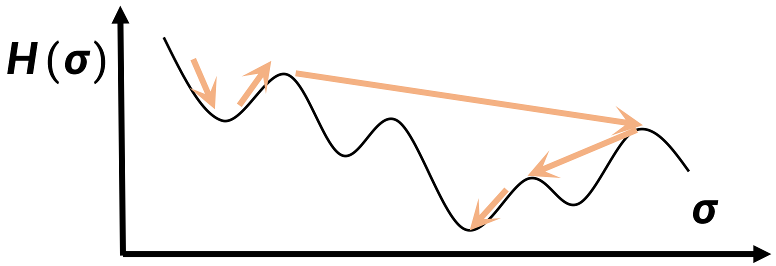
The Ising Model

- Describe ferromagnetic interactions of magnetic spins
- Each spin: either an upward (+1) or downward (−1) state
- Energy of an Ising model (Hamiltonian):

$$H(\sigma) = -\sum_{i,j} J_{ij}\sigma_i\sigma_{\boldsymbol{j}} - \sum_i \mathrm{h}_i\sigma_i$$

- Converge to the lowest energy state
- The Ising machine: Combinatorial optimization with a polynomial time



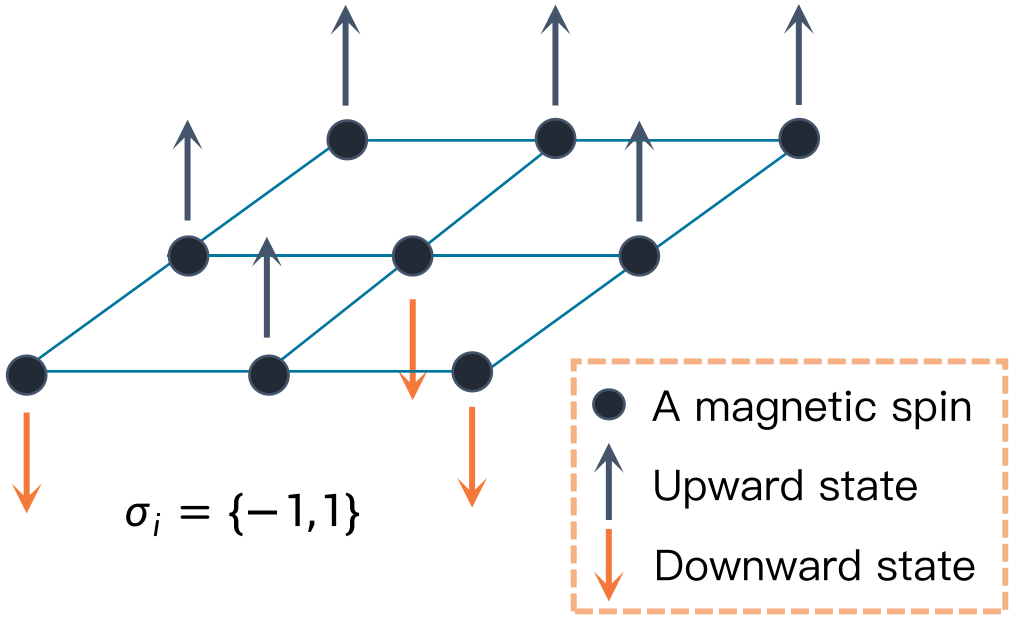


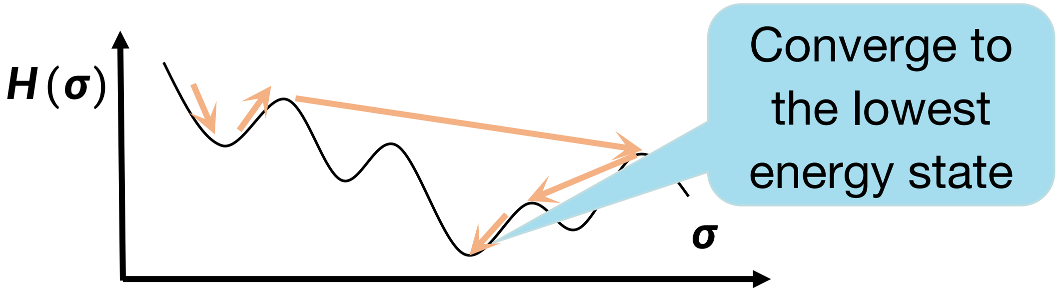
The Ising Model

- Describe ferromagnetic interactions of magnetic spins
- Each spin: either an upward (+1) or downward (−1) state
- Energy of an Ising model (Hamiltonian):

$$H(\sigma) = -\sum_{i,j} J_{ij}\sigma_i\sigma_j - \sum_i h_i\sigma_i$$

- Converge to the lowest energy state
- The Ising machine: Combinatorial optimization with a polynomial time

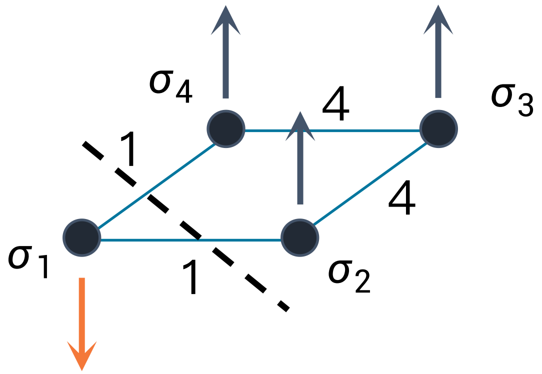




Solving MCPs using the Ising Machine

- The Max-cut problem (MCP): Partition the vertices in a weighted graph to two independent subsets such that the sum of edges between the subsets is maximized.

$$H(\sigma) = -\sum_{i,j} J_{ij}\sigma_i\sigma_{\mathbf{j}} - \sum_i h_i\sigma_i$$



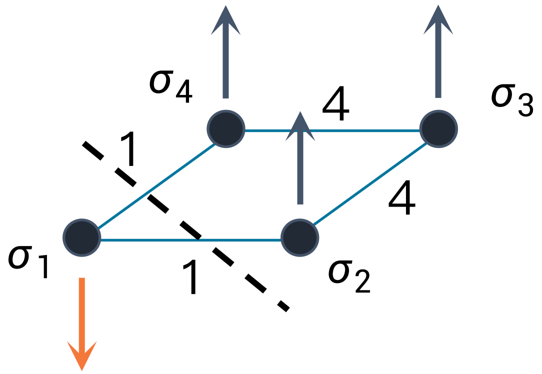
$$cut = 1 + 1 = 2$$

$$H(\sigma) = -(1 \times \sigma_1\sigma_4 + 1 \times \sigma_1\sigma_2 + \dots) = -6$$

Solving MCPs using the Ising Machine

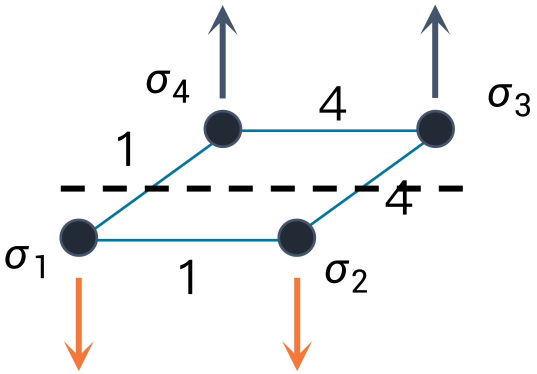
- The Max-cut problem (MCP): Partition the vertices in a weighted graph to two independent subsets such that the sum of edges between the subsets is maximized.

$$H(\sigma) = -\sum_{i,j} J_{ij}\sigma_i\sigma_j - \sum_i h_i\sigma_i$$



$cut = 2$

$H(\sigma) = -6$



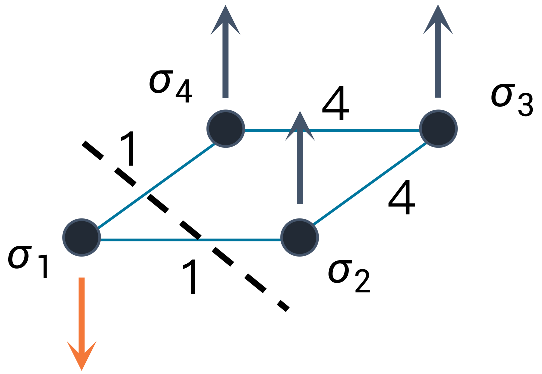
$cut = 1 + 4 = 5$

$H(\sigma) = -(1 \times \sigma_1\sigma_4 + 4 \times \sigma_2\sigma_3 + \dots)$
$= 0$

Solving MCPs using the Ising Machine

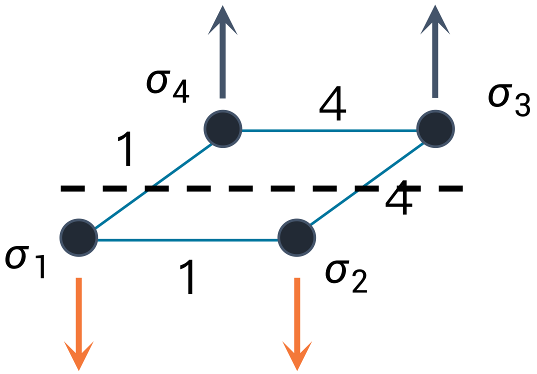
- The Max-cut problem (MCP): Partition the vertices in a weighted graph to two independent subsets such that the sum of edges between the subsets is maximized.

$$H(\sigma) = -\sum_{i,j} J_{ij}\sigma_i\sigma_{\mathbf{j}} - \sum_{i} \mathrm{h}_i\sigma_i$$



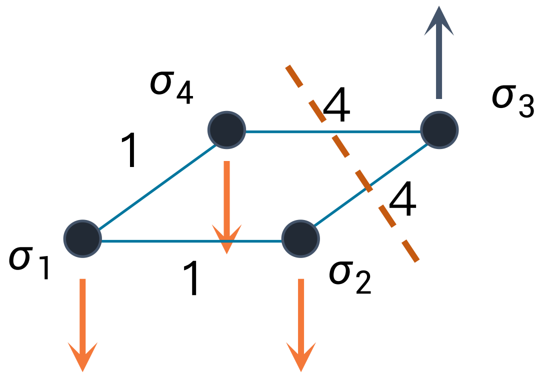
$cut = 2$

$H(\sigma) = -6$



$cut = 5$

$H(\sigma) = 0$

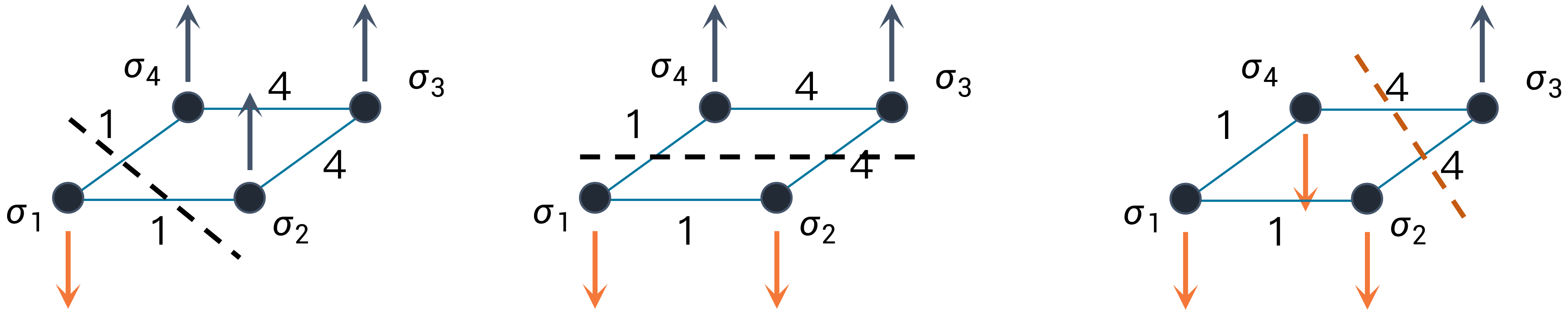


$H(\sigma) = -(4 \times \sigma_3\sigma_4 + \ldots) = 6$

Solving MCPs using the Ising Machine

- The Max-cut problem (MCP): Partition the vertices in a weighted graph to two independent subsets such that the sum of edges between the subsets is maximized.

$$H(\sigma) = -\sum_{i,j} J_{ij}\sigma_i\sigma_j - \sum_i h_i\sigma_i$$



$cut = 2$

$H(\sigma) = -6$

$cut = 5$

$H(\sigma) = 0$

$cut = 6$

Max–cut found!

$H(\sigma) = 6$

Solving MCPs using the Ising Machine

- Good news: simulated bifurcation (SB) realizes parallel update of the spin values, unlike simulated annealing (SA).

Simulated bifurcation (SB)

$$\dot{x}_{i,t} = a_0 y_{i,t},$$

$$\dot{y}_{i,t} = -\{a_0 - a(t)\}x_{i,t} + c_0 J x_{i,t} + \eta(t) h_i$$

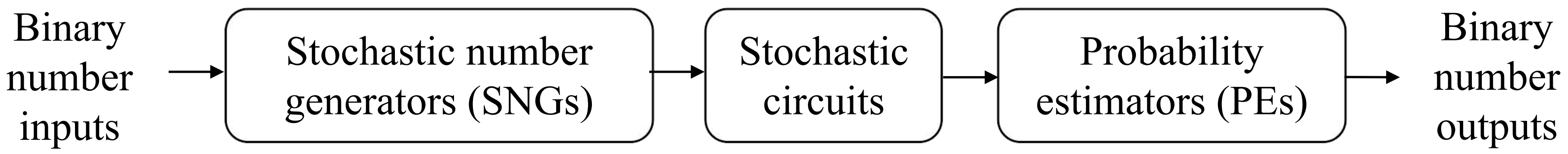
x_i is replaced with its sign and y_i is initialized to 0 if $|x_i| > 1$.

$x_{i,t}$ and $y_{i,t}$ are the position and momentum of oscillator s_i, respectively. $J_{i,j}$ describes the interaction between s_i and s_j. a_0 and c_0 are constants. $a(t)$ is a linear function. $\dot{x}_{i,t}$ and $\dot{y}_{i,t}$ are derivatives of $x_{i,t}$ and $y_{i,t}$, respectively.

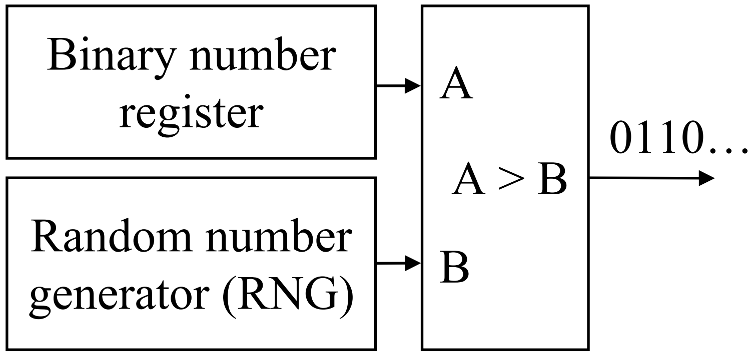
- Bad news: solving differential equations is not easy, expecially when the matrices are large (compute-intensive)

Stochastic Computing (SC)

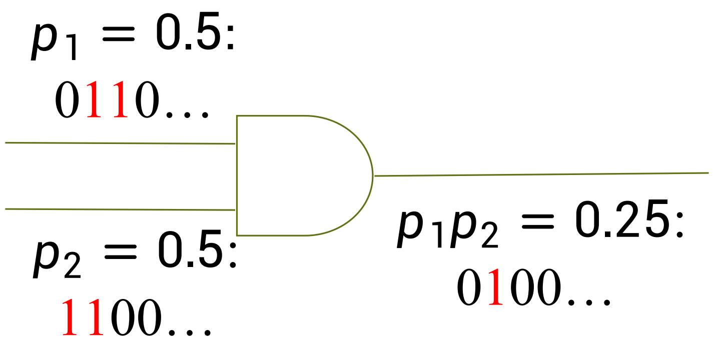
- Good news: In SC, values are represented and processed as random bit streams of 0s and 1s; simple logic gates/counters can perform arithmetics



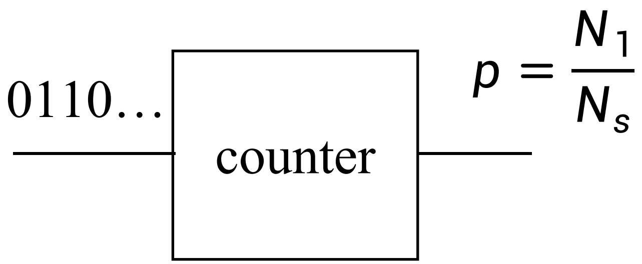
A stochastic computing system.



An SNG.



A unipolar stochastic multiplier.



A probability estimator.

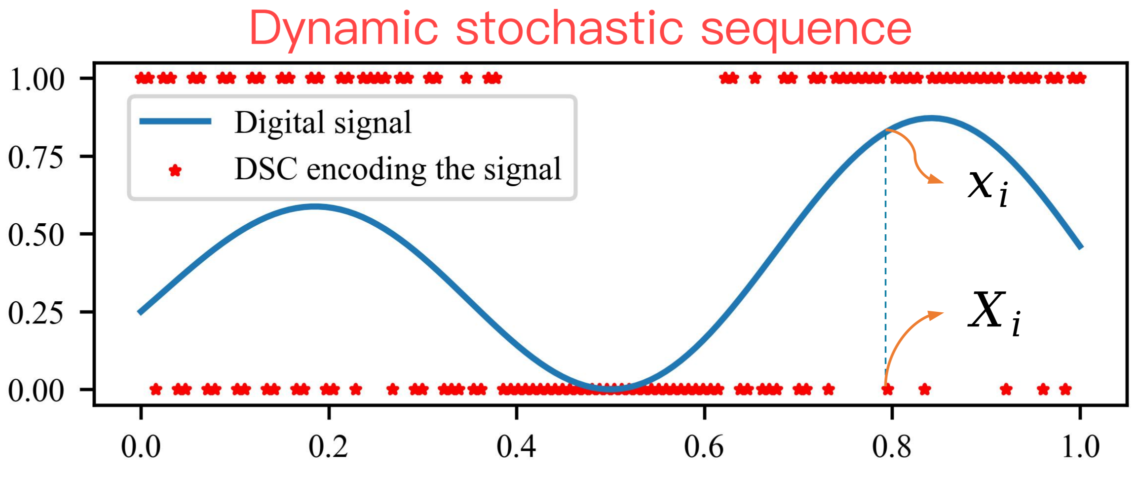
N_1: the number of 1s.

N_s: the number of all bits.

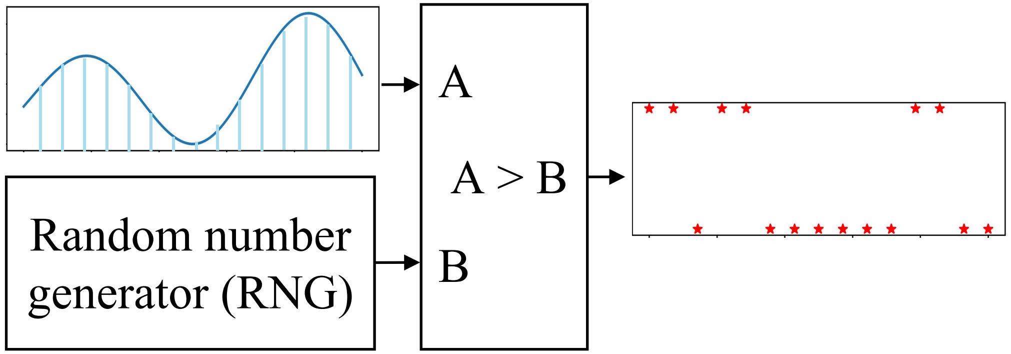
Dynamic Stochastic Computing (DSC)

- Good news: In DSC, signals are sampled as random bit streams of 0s and 1s; each bit encodes a (changing) value or probability of the signal.

Specifically, we use **dynamic stochastic computing (DSC)**



For each sampling point, $\mathbb{E}[X_i] = x_i$



A DSNG.

Dynamic Stochastic Computing (DSC, Cont'd)

- Good news: In DSC, signals are sampled as random bit streams of 0s and 1s; each bit encodes a (changing) value or probability of the signal.

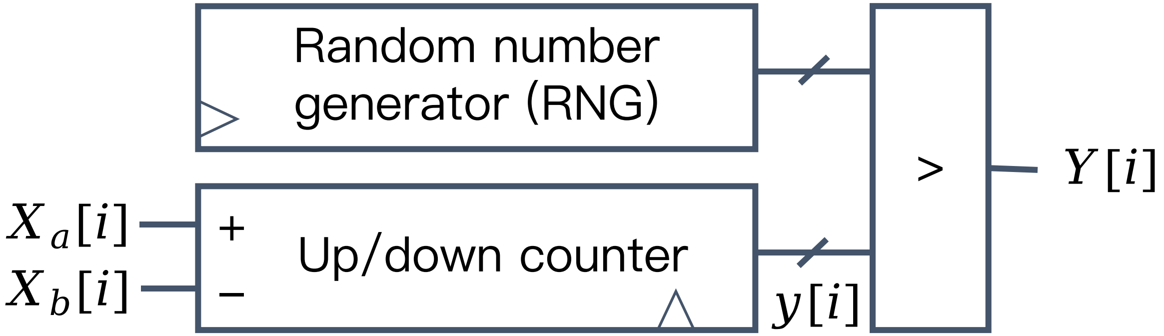
Ordinary differential equation (ODE)

$$\frac{\mathbf{d}y(t)}{\mathbf{d}t} = f(t)$$

Euler method $\hat{y}_i \approx \sum f_i$

Instead $\hat{y}_i \approx \sum F_i$ F_i: DSS encoding $f(t)$

In our previous work, published in DAC'17 [1]



A stochastic integrator.

$$y[i] = y(t)|_{t=hi} \approx \int [x_a(t) - x_a(t)]$$

Outline

- Background
  - Ising machine
  - (Dynamic) stochastic computing
- Formulation and circuit design
- System design
- Experiments and results
- Conclusion

Formulation of SB

1

$$\dot{x_{i,t}} = a_0 y_{i,t} = f(\boldsymbol{y}_t)_i,$$

$$\dot{y_{i,t}} = -\{a_0 - a(t)\}x_{i,t} + c_0 J\boldsymbol{x}_{i,t} + \eta(t)\mathrm{h}_i = g(\boldsymbol{x}_t)_i$$

A linear function

Semi-implicit Euler integration

$$y_{i,t+1} = y_{i,t} + \eta g(\boldsymbol{x_t})_i$$

$$x_{i,t+1} = x_{i,t} + \eta f(\boldsymbol{y_{t+1}})_i$$

$$x_{i,t+1} = x_{i,0} + \eta^2 \sum_{j=0}^{t} \sum_{k=0}^{j} g(x_k)_i$$

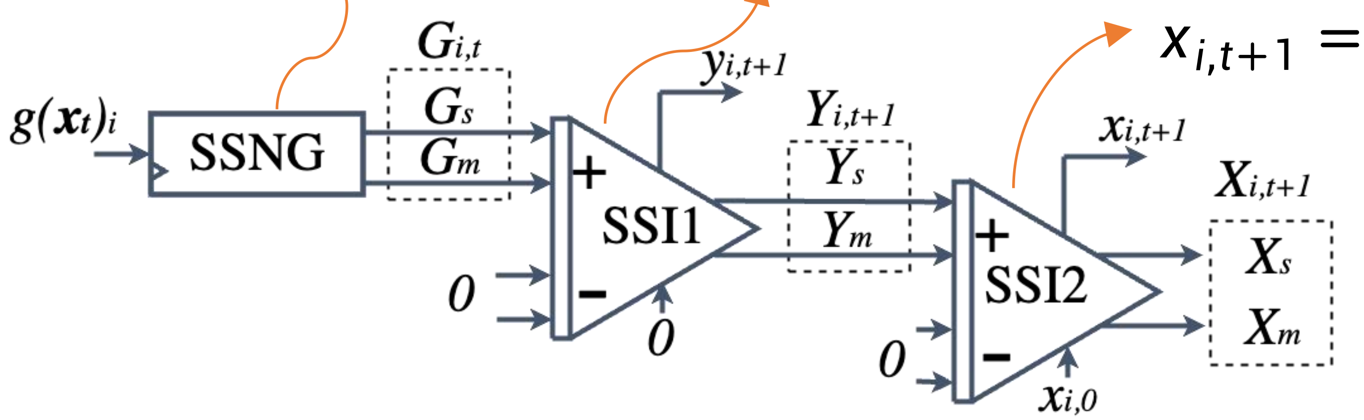
A Stochastic Computing SB Cell

$$x_{i,t+1} = x_{i,0} + \eta^2 \sum_{j=0}^{t} \sum_{k=0}^{j} g(x_k)_i$$

Convert a binary $g(\boldsymbol{x_t})_i$ to a DSS

$$y_{i,t+1} = y_{i,t} + \eta g(\boldsymbol{x_t})_i$$

$$x_{i,t+1} = x_{i,t} + \eta f(\boldsymbol{y_{t+1}})_i$$



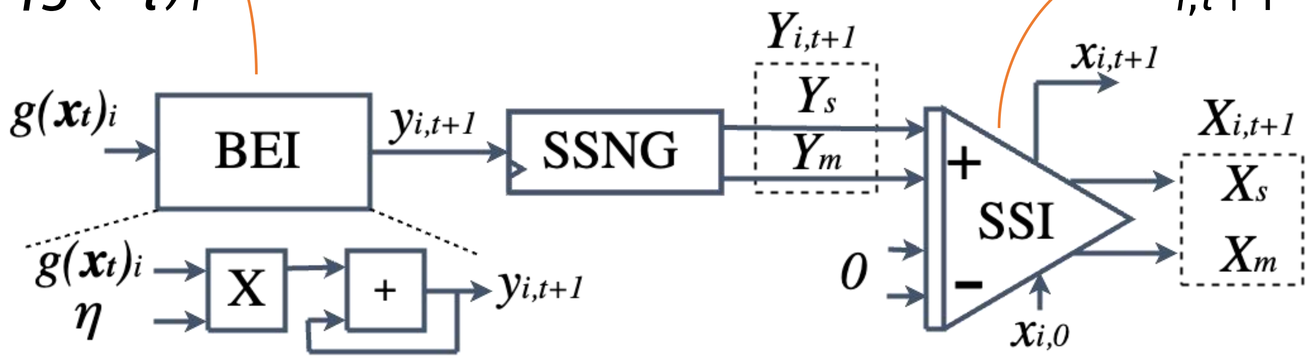
The Stochastic Computing SB Cells (SC-SBCs)
Aimed for higher area efficiency

A Binary–Stochastic Computing SB Cell

$$x_{i,t+1} = x_{i,0} + \eta^2 \sum_{j=0}^{t} \sum_{k=0}^{j} g(x_k)_i$$

$$y_{i,t+1} = y_{i,t} + \eta g(\mathbf{x_t})_i$$

$$x_{i,t+1} = x_{i,t} + \eta f(\mathbf{y_{t+1}})_i$$

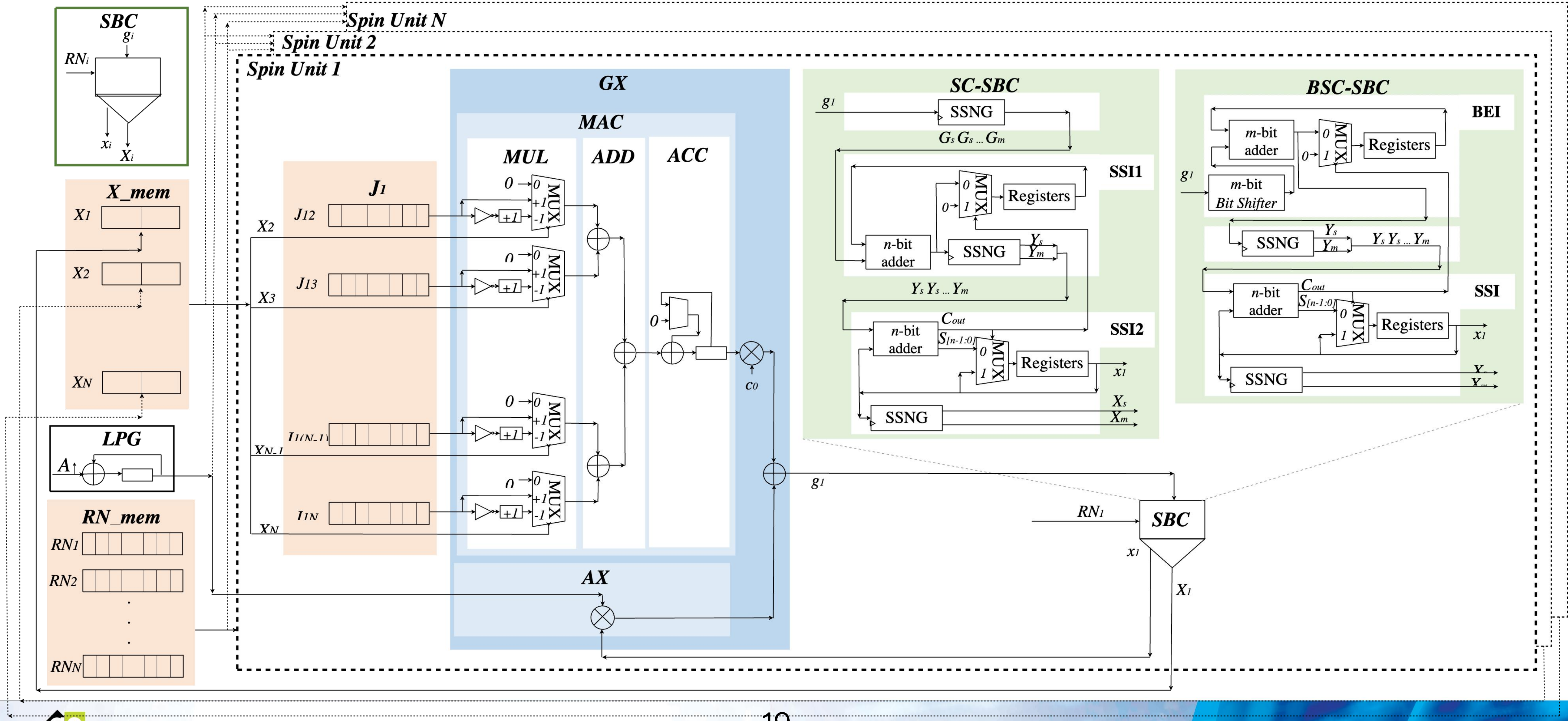


The Binary-Stochastic Computing SB Cells (BSC-SBCs)
Aimed for higher speed

Outline

- Background
  - Ising machine
  - (Dynamic) stochastic computing
- Formulation and circuit design
- System design
- Experiments and results
- Conclusion

The SSBM System Design



THE CHIPS TO SYSTEMS CONFERENCE
SHAPING THE NEXT GENERATION OF ELECTRONICS

Outline

- Background
  - Ising machine
  - (Dynamic) stochastic computing
- Formulation and circuit design
- System design
- Experiments and results
- Conclusion

Applicaiton: Max-Cut Problems (MCPs)

■ Experimental Setup

- Algorithms: bSB, dSB, SC-SBM (η= 0.125,0.25,0.5), BSC-SBM (η= 0.125,0.25,0.5).
- Benchmark: the *K2000* benchmark
- Time steps: $T_s = 1000$, $T_s = 10000$

■ Evaluation:

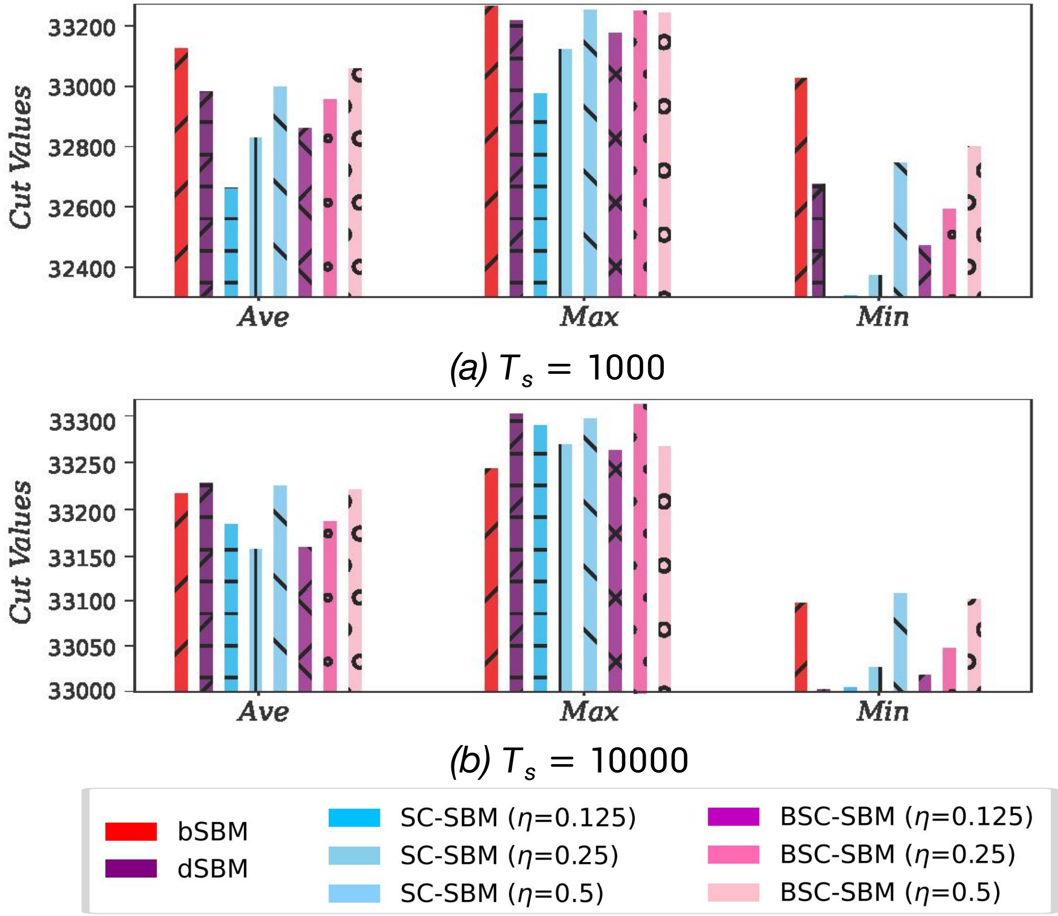
- The statistics of cut values from 100 trails

  ***Ave**: the average of cut values;* ***Max**: the maximum of cut values;* ***Min**: the minimum of cut values.*

  A larger ***Ave**,* ***Max*** *and* ***Min*** *indicate a higher performance, given by a higher likelihood to jump out of the local optima, and thus a higher stability.*
- **Probability-to-target (**P_g**)** and **Step-to-target (**S_g**)**

Performance Evaluation

- The proposed SSBM: the higher *Ave* and *Min* values are obtained with η= 0.5 than with η = 0.125, 0.25.
- Evaluated by *Ave* and *Min*, when η = 0.5, the BSC-SBM performs better than the SC-SBM when T_s = 1000; the SC-SBM performs better than the BSC-SBM when T_s = 10000.
- **It shows the advantages of BSC-SBM in a short search, and SC-SBM in a long search.**



(a) T_s = 1000

(b) T_s = 10000

* bSBM: ballistic simulated bifurcation machine;
dSBM: discrete simulated bifurcation machine.

Performance Evaluation (Cont’d)

- **For $T_s = 1000$**, the SSBMs can achieve a higher $P_{99.5\%}$ value than dSBM. Moreover, the proposed BSC-SBM performs similarly to bSBM.
- For $T_s = 10000$, it is difficult for bSBM to reach $P_{99.8\%}$ of the best-known cut value due to the lack of ability to jump out of the local minima, and a better solution can be obtained by dSBM and SSBMs.
- It shows that SSBMs find a better solution than dSBM in a short search and have a lower probability of being stuck at the local minima than bSBM in a long search.

The Values of Pg and Sg for the Max-cut Problems on *K2000* Benchmark

Vaules of P_g and S_g with T_s		SB Machines			
		bSBM	dSBM	SC-SBM	BSC-SBM
$T_S = 1000$	$P_{99.5\%}$	38%	4%	6%	22%
	$S_{99.5\%}$	7633	112811	74426	18534
$T_S = 10000$	$P_{99.8\%}$	0	6%	4%	2%
	$S_{99.8\%}$	-	744265	1128110	2279481

* *K2000*: 2000 nodes, 1999000 edges, a complete graph, edge weight $w_{ij} \in \{-1, +1\}$, best-known cut value: 33337.

Hardware Efficiency

■ Experimental Setup

- Ising Machines: D-wave[3], JSSC'21[8], JSSC'15[14], ISSCC'21[15], CICC'21[16], JSSC'22[17], vs. SC-SBM, BSC-SBM
- Simulation results for SC-SBM and BSC-SBM are obtained by using the Synopsys Design Compiler. A CMOS 40nm technology is applied with a supply voltage of 1.0 V and a temperature of 25℃.

■ Evaluation

- Computing Method; Technology; # Spin; Topology; # Spin Interactions; Coefficient Bit-Width; Spin Type
- Power per Spin; Area per Spin; Frequency; # Spin Update Cycles
- Normalized Power per Spin, Normalized Area per Spin

Hardware Efficiency (Cont'd)

- The dense connectivity between spins leads to an increase in area and power.
- The spins in SC–SBM and BSC–SBM require 1.5X and 1.3X more normalized power per spin than [8], respectively, due to the 3.9X larger connectivity.
- The proposed SC–SBM and BSC–SBM utilize at least 10.62% smaller normalized area than [8].

	D-wave [3]	JSSC'15 [14]	JSSC'21 [8]	ISSCC'21 [15]	CICC'21 [16]	JSSC'22 [17]	Prop. SC-SBM	Prop. BSC-SBM
Computing Method	Quantum Annealing	CMOS Annealing	SCA Annealing	Metropolis Annealing	Simulated Annealing	Simulated Annealing	Simulated Bifurcation	Simulated Bifurcation
Technology	Superconductor	65nm CMOS	65nm CMOS	65nm CMOS	65nm CMOS	65nm CMOS	40nm CMOS	40nm CMOS
# Spins	$2k$	$20k$	512	$16k$	252	480	$2k$	$2k$
Topology	Chimera	Lattice	Complete	King	King	King	Complete	Complete
# Spin Interactions	5	5	511	8	8	8	1999	1999
Coefficient Bit-Width	N/A	2	5	5	4	4	2	2
Spin Type	Qubit	SRAM	SRAM	Register	Register	Register	Register	Register
Power per Spin	12.2 W	2.83 μW	1.27 mW	N/A	1.33 μW	0.18 μW	0.74 mW	0.64 mW
Area per Spin (Normalized Area)	N/A	289 μm^2 (6.86 ×)	12207 μm^2 (1.13 ×)	552 μm^2 (3.28 ×)	1671 μm^2 (12.41 ×)	832 μm^2 (6.17 ×)	6370 μm^2 (1 ×)	6453 μm^2 (1.01 ×)
Frequency	N/A	100 MHz	320 MHz	100 MHz	64 MHz	200 MHz	250 MHz	250 MHz
# Spin Update Cycles	N/A	N/A	512	22	N/A	1	20	20

Conclusion

- A high-performance fully connected stochastic SB machine (SSBM) is designed for low-cost and accurate combinatorial optimization using the Ising model.
- Based on stochastic computing, two efficient SB cells are further designed by using SSIs to solve pairs of differential equations in SB.
- The 2000-spin fully connected SSBM using the SC-SBC or BSC-SBC as a building block realizes fast energy convergence in a short search and also prevents from being stuck at the local minimum in a long search.
- An improvement of at least 44% in power is achieved with a 1.19X speedup, compared to conventional SB machines.

References

[1] S. Liu and J. Han, "Hardware ODE solvers using stochastic circuits," in DAC, pp. 1–6, 2017.

[3] M. W. Johnson et al., "Quantum annealing with manufactured spins," Nature, vol.473, no. 7346, pp. 194–198, 2011.

[8] K. Yamamoto 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, 2021.

[14] M. Yamaoka et al., "A 20k-spin Ising chip to solve combinatorial optimization problems with CMOS annealing," IEEE JSSC, vol. 51, no. 1, pp. 303–309, 2015.

[15] T. Takemoto et al., "4.6 a 144kb annealing system composed of 9X16kb annealing processor chips with scalable chip-to-chip connections for large-scale combinatorial optimization problems," in ISSCC, vol. 64. IEEE, 2021, pp. 64–66.

[16] Y. Su et al., "A 252 spins scalable CMOS Ising chip featuring sparse and reconfigurable spin interconnects for combinatorial optimization problems," in CICC. IEEE, 2021, pp. 1–2.

[17] Y. Su et al., "A scalable CMOS Ising computer featuring sparse and reconfigurable spin interconnects for solving combinatorial optimization problems," IEEE JSSC, vol. 57, no. 3, pp. 858–868, 2022.

Q&A

Thanks for your attention!