Data Movement-Aware, Ping-Pong Ising Machine Supporting Full Connectivity and Variable Bitwidths

Chieh-Pu Lo, Sirish Oruganti, Yipeng Wang, Mengtian Yang, Shanshan Xie, Jaydeep P. Kulkarni
Chandra Department of Electrical and Computer Engineering, University of Texas at Austin, Austin, TX
kcplo@utexas.edu, jaydeep@austin.utexas.edu

2024 IEEE European Solid-State Electronics Research Conference (ESSERC) | 979-8-3503-8813-8/24/$31.00 ©2024 IEEE | DOI: 10.1109/ESSERC62670.2024.10719421

***Abstract*—Ising computation is an emerging paradigm for efficiently solving the time-consuming Combinatorial Optimization problems (COP). In particular, Compute-In-Memory (CIM) based Ising machines are promising for Hamiltonian computations capturing the spin state dynamics using a dense memory array. However, the advancement of CIM-based Ising machines for accurately solving complex COPs is limited by the lack of full spin-connectivity, small bitwidths of the spin interaction coefficients (J), data movement energy costs within the CIM, and the area/energy overheads of peripheral CIM analog circuits.**

In this work, we present a data movement-aware, CIM-based Ising machine for efficiently solving COPs. The unique design contributions are: (i) "Ping-Pong" transpose array architecture restricting single-bit spin data movement within the memory array while maintaining interaction coefficients (J) stationary, (ii) Fully connected spins and multi-bit J for >329× faster solution time, (iii) Configurable J bit-widths and spin connectivity to support wide variety of complex COPs. (iv) Bitcell-Reference (BR) based capacitor-bank-less ADC for sensing, occupying 1.81× less area than the baseline SAR ADC. The silicon prototype in 65nm CMOS demonstrates >4.7× better power efficiency, and >9.8× better area efficiency compared to prior works.

I. Introduction

One of the popular approaches for efficiently solving non-deterministic polynomial-time hard (NP-hard) combinatorial optimization problems (COPs) is to utilize naturally occurring phenomena to achieve the optimal solution by mapping a given COP onto an Ising model. By finding the minimum energy of the Hamiltonian function H (Ising model equation in Fig. 1), the model computes the lowest energy point as the optimal COP solution. In recent years, CMOS-based Ising machines have emerged as an energy-efficient and low-cost approach compared to quantum or optical approaches. CMOS Ising machines use either coupled oscillators, or map the hamiltonian into a Compute-in-Memory (CIM) representation. Oscillator based machines are faster in solving the COP, but consume significantly higher energy due to frequent switching activity, making CIM based approaches more desirable. However, previously reported CIM Ising machines suffer from the following bottlenecks: (i) They are mostly limited either in their graph connectivity (Lattice graph, King's graph, or arbitrary connectivity) incurring longer solution time [1]–[6] or require a large footprint to process a fully connected graph [7], [8] as shown in Fig. 1. (ii) Intra/Inter-memory data movement cost increases by frequently accessing J coefficients from the dedicated SRAMs. Such data movement cost is exacerbated in proportion to Spin-count×Connectivity×Bit Precision of J [2], [5], [7], [8]. (iii) Multi-bit CIM computing requires analog-to-digital converters (ADC) with capacitors and/or additional reference voltage generators, causing considerable area overhead [9].

In this work, we mitigate the above critical bottlenecks by devising a fully connected, data-movement-aware CIM-Ising machine with reduced solution time and power consumption. We propose a Ping-Pong Ising machine featuring unique design techniques: (i) Ping-Pong transpose memory architecture with reduced data movement for spin update and next iteration activation (ii) Full spin connectivity and multi-bit J precision through analog Hamiltonian computation (iii) Reconfigurable and scalable Ising macro to support wide COP complexity (iv) Compact bitcell-reference SAR ADC, eliminating the need for large capacitor banks in SARs.

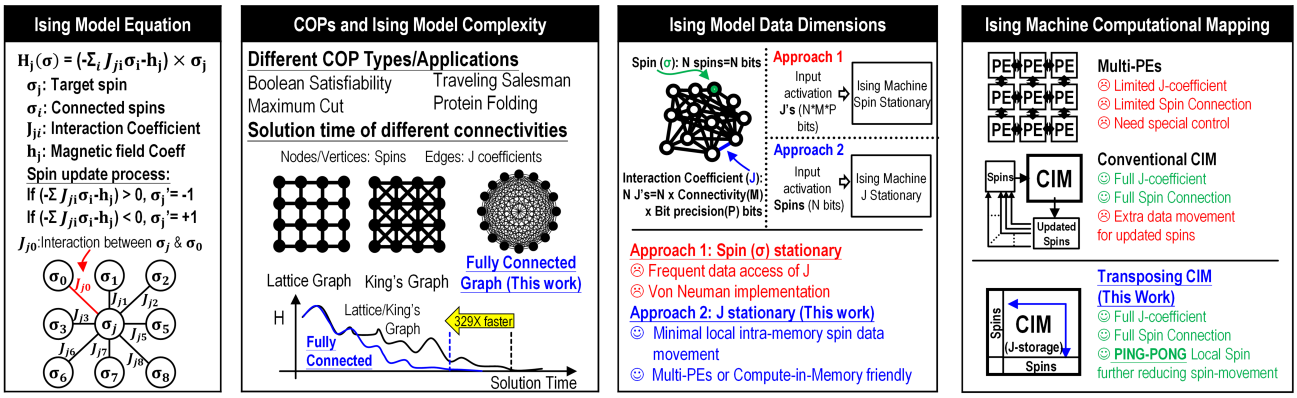


Fig. 1: Challenges of Ising machine hardware. Proposed fully connected, J-stationary and "Ping-Pong" computation within the memory.

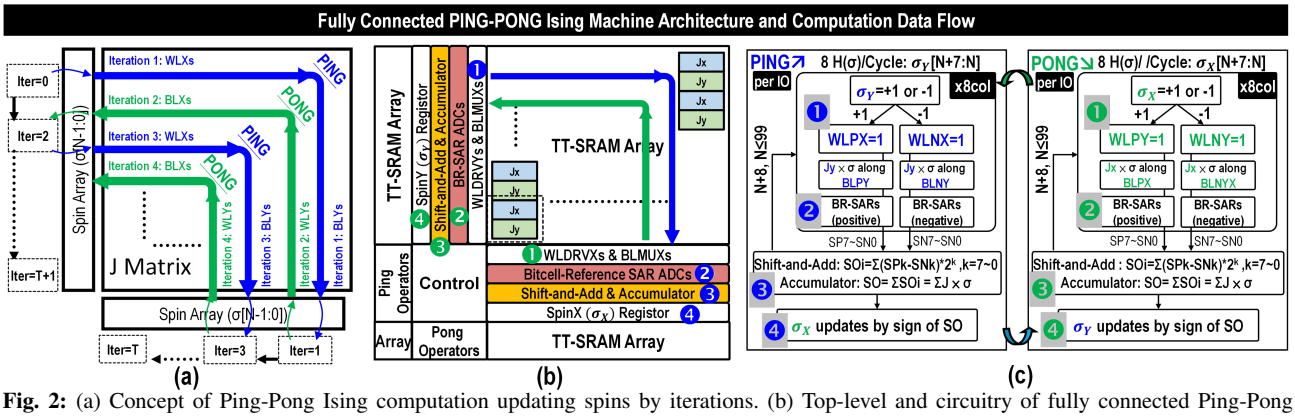


Fig. 2: (a) Concept of Ping-Pong Ising computation updating spins by iterations. (b) Top-level and circuitry of fully connected Ping-Pong Ising macro. (c) Dataflow chart of overall Ping-Pong Ising computation.

II. Proposed Ping-Pong Architecture

In contrast to CIM for ML applications, where layers feed into one another, and the dataflow is *linear*, J Coefficients in an Ising model are fixed for a given COP, and the spin-update dataflow in CIM-based Ising model is *iterative*. A fully connected Ising model (graph) is one in which all spins interact with one another, i.e. the J matrix has coefficients for all combinations of spin pairs. J-matrix storage within the CIM engine, with entries representing spin-to-spin connectivity in either direction, is adopted to eliminate large multi-bit precision J coefficient data access/movement.

The transposing Ping-Pong architecture has two orthogonally arranged computation paths which eliminate the requirement for spin data movement between iterations (Fig. 2(a)). In the first iteration, spins (σ) are broadcast across the J-matrix in X-to-Y direction through horizontal wordline activations, and Hamiltonian ($H_\sigma = \Sigma\sigma \times J$) is computed along vertical bitlines. The spin updates from this iteration are broadcast across the J-matrix in Y-to-X direction through vertical wordline activation and horizontal bitline accumulation in the next iteration to compute the H. The two-phase transposing Ping-Pong CIM computes the H_σ entirely within the memory array for all spin iterations. The PING phase computes σ_y with J_y to obtain an updated σ_x, as shown in Fig. 2(b,c). The σ_x then starts the PONG phase in the orthogonal direction, to obtain updated σy. This spin-update process continues across X/Y orientations in alternate iterations until the lowest energy state is reached

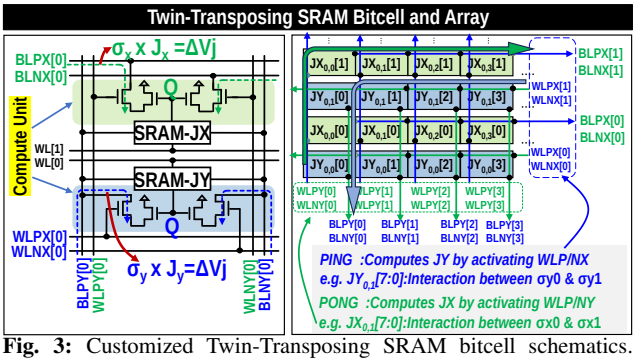


Fig. 3: Customized Twin-Transposing SRAM bitcell schematics. Activation of WLs & BLs for Ping-Pong Hamiltonian computation

(H_{min}). Thus, the spin-update data movement is localized within the memory array, resembling a Ping-Pong style spin dataflow as illustrated in Fig. 2. The Ping-Pong Ising macro is a butterfly-organized array composed of four Twin-Transposing SRAM arrays, Word-line drivers, Bit-line multiplexers, Bitcells-Reference SAR ADCs, Shift-and-Add Accumulator, and SPIN registers as shown in Fig. 2(b).

A. TT-SRAM for Ping-Pong

Most conventional CIM Ising approaches have a unidirectional computational path (horizontal wordline activations and accumulation on vertical bitlines) such that the updated spins require extra/external data buses to reach the CIM's input interface (wordlines) for the next iteration. In contrast, the Ping-Pong architecture enables two data movement directions within the same memory array. To facilitate this, a Twin-Transposing (TT) SRAM bitcell is proposed.

The TT-SRAM (Fig. 3) comprises two SRAM cells storing J_x and J_y to support PING and PONG phases of computation. For spins implemented with values of ±1, this architecture separates the paths for signed multiplication for positive and negative products on BLP(X/Y) and BLN(X/Y), which are later added with appropriate signs. For the Ising computation H(σ_k) = -$\Sigma J_{k,i} \times \sigma_i$ with 100 σ 's, the 8-bit J_k[i=0:99] are mapped onto bitcells of 8 consecutive columns, and the value σ_i is applied as the wordline activation on WLP/WLN lines. N out of eight wordlines are activated per cycle, which discharge BLs by N×ΔVj (a single TT-SRAM cell discharges by ΔVj), computing partial sums of eight spin-J products. Thus, we perform one iteration of a 100-spin COP in 13 clock cycles.

B. Bitcell-Reference SAR (BR-SAR) ADC

Conventional analog CIM techniques use Flash or SAR ADCs requiring multiple reference generators or large capacitor banks, which can add significant area overhead specific to computation when used for a wide column of bits. With a multi-bit precision J and multiple spin·J partial product accumulation, an ADC is required to resolve the data into a 4-bit partial sum-of-products. To avoid ADC area overhead, while maintaining ADC precision, a Bitcell-Reference SAR (BR-SAR) is proposed, shown in Fig. 4.

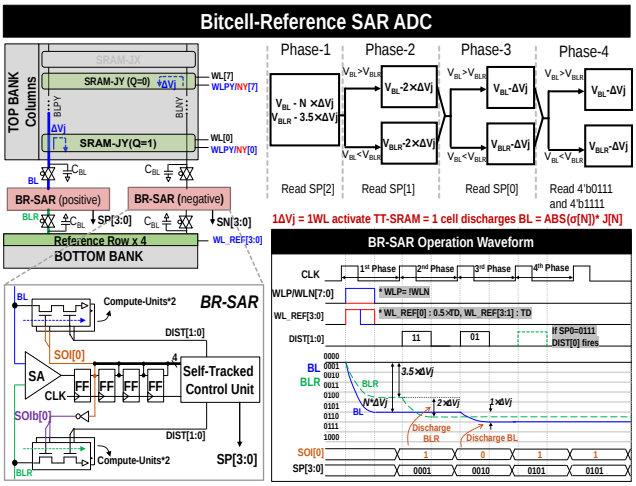


Fig. 4: Circuitry and waveform of Bitcell-Reference (BR) SAR. BR-SAR eliminates conventional ADC coupling-cap and read out 4-bits

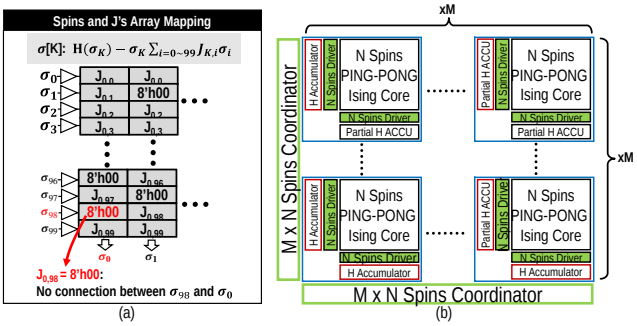


Fig. 5: (a) Programming J's to reconfigure connectivity between spins. (b) Expand Ising network M times by M^2 Ising cores.

Since data is stored in multiple banks owing to the butterfly SRAM structure, ADC reference words can be stored in the opposite bank and used to provide a reference to the BR-SAR. Four reference wordlines are first activated to generate a midpoint reference voltage for [MSB-1] resolution, by discharging the reference bitline (BLR) by 3.5ΔVj. In the conversion phase, a first comparison is made between the bitline (BLP/BLN) and BLR, the result of which discharges either BLP/BLN or BLR by 2ΔVj, to the midpoint reference voltage for [MSB-2] resolution. Based on the comparison, once again the lines are discharged by ΔVj to the midpoint voltage for [LSB] resolution. The [MSB] is resolved in the end by a fourth discharge by ΔVj, which resolves whether the input was 4'b0111 or 4'b1000. This way, a 4-bit (9 levels) ADC resolution is achieved, without the need for additional circuits to generate references or perform successive approximation using capacitors. Following the shift-and-add accumulation of partial products for all 13 read cycles, the new spin value is computed based on the sign of the iteration's Hamiltonian energy. This process, completely contained in the Ping-Pong memory array, is repeated for multiple iterations until the lowest energy solution is obtained, solving the COP.

C. Reconfigurability and Scalability of Ising Network

The Ping-Pong Ising machine can adapt to various graph structures such as Lattice-graph, King's-graph, or arbitrary

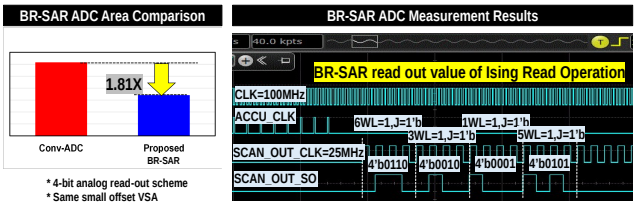


Fig. 6: Captured measured waveform and comparison of BR-SAR.

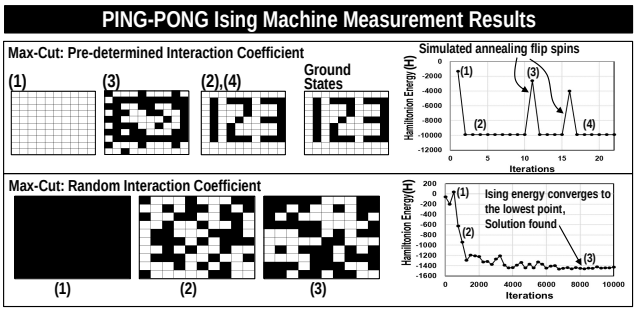


Fig. 7: Max-Cut problem checkerboard map and Ising Energy results. (a) Solving pre-determined interaction coefficient (J) Max-Cut problem (b) Solving randomized J Max-Cut problem

graph by programming the corresponding 8bits $J_{k,i}$ to zero as non-connection (Fig. 5(a)). This enables the machine to be reconfigured to support from 4 to (N-1) spin connections, alongwith J's precision ranging from 2 to 8 bits in the Ising network for different size/complexity of COPs. To solve large COPs, Ping-Pong Ising array can scale up the Ising model size (i.e. spin count) from N spins to M × N spins by organizing M^2 Ping-Pong Ising macros along with the spin coordinator and partial Hamiltonian accumulator (ACCU) logic (Fig. 5(b)).

III. Measurements and Results

The proposed Ping-Pong Ising memory macro has been implemented in 65nm CMOS. BR-SAR is validated for multi-bit read-out correctness operating at 100MHz, as shown in Fig. 6. By eliminating the capacitor bank, BR-SAR achieves 1.81X smaller area compared to conventional SAR ADCs. To test the capability of solving COPs, the Ping-Pong Ising machine was mapped to a pre-determined max-cut problem as shown in Fig. 7, which was solved in <3 iterations, owing to the fully connected graph. It also solved a randomized max-cut problem successfully to get the lowest Hamiltonian energy.

To demonstrate the efficacy of fully connected Ising machines with multi-bit J precision, a given COP problem (Max-cut in fully-connected graph) is mapped into Ising machines with lattice graph, King's graph, and fully-connected graph, with varying bit-precisions. Figure 8(a) illustrates that employing a fully connected Ising machine can solve COP 329×∼1256× faster compared to a non-fully connected Ising machine. Additionally, the bit-precision of J in the Ising network can moderately impact result accuracy. Fig. 8(b) shows that using a higher precision of J results in a better solution quality (higher cut-value in the given Max-cut problem).

To fairly compare different Ising machines, power and area efficiency comparison are normalized factoring in (1) connectivity ratio (connection per spin / total spin counts), (2)

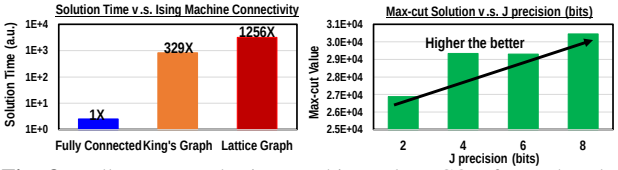


Fig. 8: Fully connected Ising machine solves COP faster than less connected Ising machine. Precision of J affects the Max-cut value.

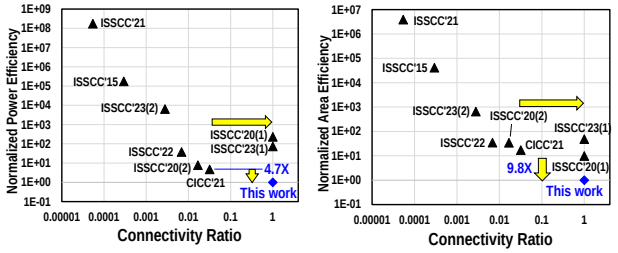
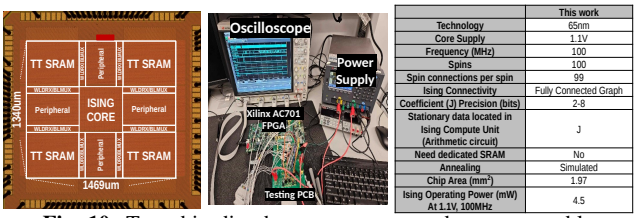


Fig. 9: PING-PONG Ising machine has 4.7X+ better normalized power efficiency and 9.8X+ normalized power efficiency

J-coefficient precision, and (3) technology node (feature size^2). Ping-Pong Ising machine consumes $>72.1\times$ and $>4.7\times$ less power than prior fully connected Ising machines and less connected Ising machines. Further, Ping-Pong Ising machine has $>9.8\times$ and $>17\times$ better area efficiency than prior fully-connected Ising machines and sparsely-connected Ising machines. Table I shows the proposed Ping-Pong Ising machine's improved power and area efficiency metrics. Fig. 10 shows the die and test setup photos, and chip summary.

IV. Conclusion

This work introduces a novel data movement-aware, CIM-based Ising machine designed to solve COPs. A "Ping-Pong" transpose memory array reduces spin data movement. Fully connected spins with multi-bit J's enhance solution time and accuracy. Configurable connectivity and J bit-widths accommodate a broad spectrum of COPs. A Bitcell-Reference based SAR ADC consumes $1.81\times$ lesser area compared to conventional SAR sensing. The 65nm CMOS silicon prototype demonstrates significant improvements, solving COPs $329\times$ faster, with over $4.7\times$ better power efficiency and more than $9.8\times$ better area efficiency relative to existing solutions.



	This work
Technology	65nm
Core Supply	1.1V
Frequency (MHz)	100
Spins	100
Spin connections per spin	99
Ising Connectivity	Fully Connected Graph
Coefficient (J) Precision (bits)	2-8
Stationary data located in Ising Compute Unit (Arithmetic circuit)	J
Need dedicated SRAM	No
Annealing	Simulated
Chip Area (mm^2)	1.97
Ising Operating Power (mW) At 1.1V, 100MHz	4.5

Fig. 10: Test-chip die photo, test setup, and summary table

Acknowledgments

This research is supported by the NSF CAREER award. Authors thank the TSMC University Shuttle program for chip prototype fabrication.

References

[1] Y. Su, H. Kim, and B. Kim, "31.2 cim-spin: A 0.5-to-1.2v scalable annealing processor using digital compute-in-memory spin operators and register-based spins for combinatorial optimization problems," in *2020 IEEE ISSCC*, pp. 480–482.

[2] T. Takemoto *et al.*, "4.6 A 144Kb Annealing System Composed of 9×16Kb Annealing Processor Chips with Scalable Chip-to-Chip Connections for Large-Scale Combinatorial Optimization Problems," in *2021 IEEE ISSCC*, vol. 64, pp. 64–66.

[3] Y. Su, J. Mu, H. Kim, and B. Kim, "A 252 Spins Scalable CMOS Ising Chip Featuring Sparse and Reconfigurable Spin Interconnects for Combinatorial Optimization Problems," in *2021 IEEE CICC*, pp. 1–2.

[4] Y. Su, T.-H. Kim, and B. Kim, "FlexSpin: A Scalable CMOS Ising Machine with 256 Flexible Spin Processing Elements for Solving Complex Combinatorial Optimization Problems," in *2022 IEEE ISSCC*, vol. 65, pp. 1–3.

[5] 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, 2016.

[6] J. Bae, W. Oh, J. Koo, and B. Kim, "CTLE-Ising:A 1440-Spin Continuous-Time Latch-Based isling Machine with One-Shot Fully-Parallel Spin Updates Featuring Equalization of Spin States," in *2023 IEEE ISSCC*, pp. 142–144.

[7] K. Yamamoto *et al.*, "7.3 STATICA: A 512-Spin 0.25M-Weight Full-Digital Annealing Processor with a Near-Memory All-Spin-Updates-at-Once Architecture for Combinatorial Optimization with Complete Spin-Spin Interactions," in *2020 IEEE ISSCC*, pp. 138–140.

[8] K. Kawamura *et al.*, "Amorphica: 4-Replica 512 Fully Connected Spin 336MHz Metamorphic Annealer with Programmable Optimization Strategy and Compressed-Spin-Transfer Multi-Chip Extension," in *2023 IEEE ISSCC*, pp. 42–44.

[9] S. Spetalnick *et al.*, "A 2.38 MCells/mm2 9.81 -350 TOPS/W RRAM Compute-in-Memory Macro in 40nm CMOS with Hybrid Offset/IOFF Cancellation and ICELL RBLSL Drop Mitigation," in *2023 IEEE Symp.VLSI*, pp. 1–2.

TABLE I: Comparison table with prior Ising machine works

	This work	ISSCC'20(1)	ISSCC'23(1)	ISSCC'20(2)	ISSCC'21	CICC'21	ISSCC'22	ISSCC'15	ISSCC'23(2)
Technology	65nm	65nm	40nm	65nm	40nm	65nm	65nm	65nm	65nm
Core Supply	1.1V	1.1V	0.8V-1.1V	0.5V-1.2V	1.1V	0.6V-1.2V	1V-1.2V	1.1V	0.75V-1.05V
Frequency (MHz)	100	320	120-336	0.5-200	100	0.5-200	64	100	N/A
Spin Count	100 (*4)	512	2048	480	147456	252	1024	20480	1440
Spin connections per spin	99	511	2047	8	8	8	7	6	4
Ising Connectivity	Fully Connected Graph	Fully Connected Graph	Fully Connected Graph	King's Graph	King's Graph	King's Graph	King's Graph	Lattice Graph	Lattice Graph
Coefficient (J) Precision (bits)	2-8	5	8	4	5	4	8	2	1
Stationary operand during Ising Compute	J	Spins	Spins	J	Spins	J	J	Spins	J
Need dedicated SRAM for J	No	Yes	Yes	No	Yes	No	No	Yes	No
Annealing	Simulated	SCA	DA/RPA/SA/SCA	Simulated	Metropolis	Simulated	Simulated	Simulated	Simulated
Chip Area (mm2)	1.97	12.00	36.00	0.56	97.31	0.54	0.47	5.93	0.45
Connectivity Ratio *1	1.0	1.0	1.0	1.67E-02	5.43E-05	3.19E-02	6.84E-03	2.93E-04	2.78E-03
Normalized Area Efficiency *2	1.0	9.8	48.3	34.2	3.85E+06	17.1	34.6	41114.9	652.2
Normalized Power Efficiency *3	1.0	230.8	72.8	7.8	1.73E+08	4.7	37.9	175543.0	6395.6

*1: Connectivity ratio: Connection per spin/ (Spin Count – 1)
*2: Ising core area / (Connectivity Ratio x Coefficient Precision x feature size^2)
*3: Ising operation power / (Connectivity Ratio x Coefficient Precision x feature size^2)
*4: Spin counts limited to 100 by the limited test-chip size