nature electronics

Supplementary information

https://doi.org/10.1038/s41928-026-01700-6

An Ising machine for combinatorial optimization based on sub-nanosecond CMOS-integrated magnetoresistive random-access memory

In the format provided by the authors and unedited

CONTENTS:

Supplementary Note 1 Probabilistic Characteristics and Calibration of VCMA-MTJ Devices

The switching probability of VCMA-MTJs as a function of pulse width is illustrated in Supplementary Figure 1a. In Supplementary Figure 1b, these curves are fitted to a sigmoid-like function:

$$P_{sw} = \frac{1}{1 + e^{-\alpha(W_v - W_0)}}$$

where α and W_0 represent fitting coefficients. The update mechanism of the Ising machine is described as follows:

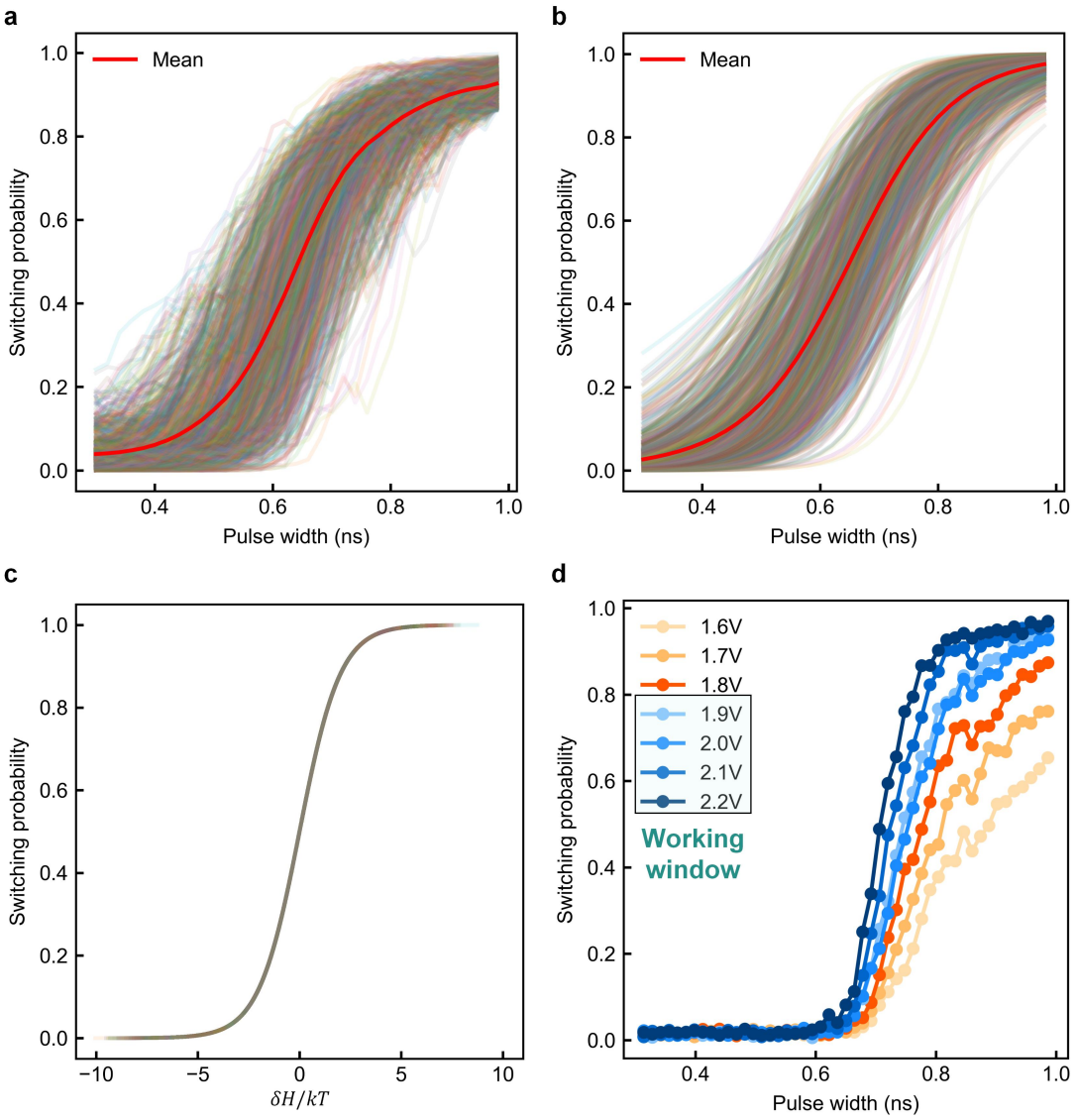
$$P_{flip} = \frac{1}{1 + e^{-(\delta H/T)}}$$

By enforcing the equation $\delta H/T = \alpha(W_v - W_0)$ to be satisfied, a standard sigmoid curve can be effectively realized using any of these devices (Supplementary Figure 1c).

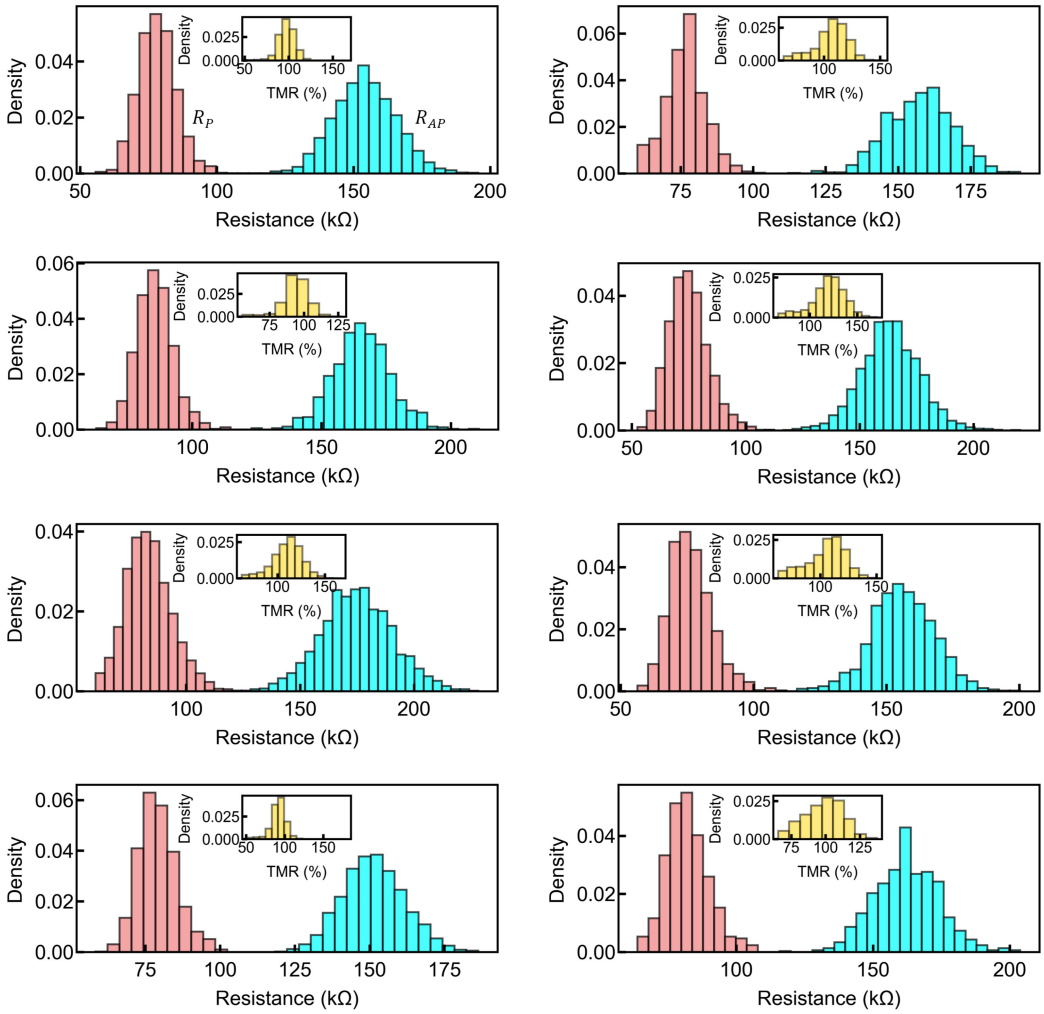
The switching probability under varying voltage pulses across 1000 VCMA-MTJs (ranging from 1.6 to 2.2 V) is presented in Supplementary Figure 1, confirming consistent sigmoidal behavior with moderate variations in slope and offset. Lower voltage values result in a reduced peak switching probability (PSP). We define the working window of operational voltage as the range required to achieve a PSP exceeding 90%. We found that approximately 20% of the devices exhibit a PSP below 90%, randomly distributed across the array. We adopt a device variation calibration methodology following established approaches in the literature[1,2]. The calibration consists of two steps: (i) Offset calibration: The pulse-width configuration is adjusted for each device to center the switching probability at 50%, effectively removing the offset of the sigmoid curve. This ensures that all devices operate around their balanced switching point; (ii) Slope calibration: Variations in the slope are absorbed into the effective temperature parameter of the system, corresponding to a rescaling rather than an error. During Ising computation, the ideal update signal derived from the energy difference is transformed to the device-specific pulse width using the calibration parameters (see Supplementary Figure 1c). As shown in Supplementary Figure 1d, a minimum voltage of approximately 1.9 V is necessary, while 2.2 V is sufficient to produce a well-defined sigmoid curve.

Beyond voltage dependence, we systematically evaluate both static and dynamic

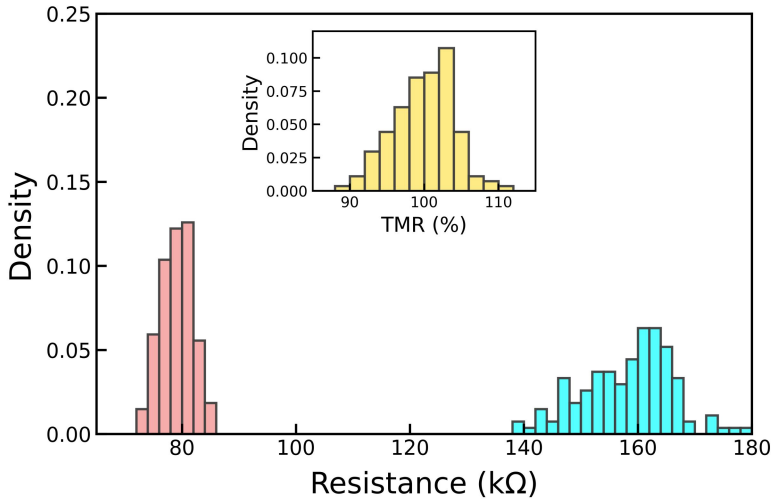
device variations, and quantify their impact on probabilistic switching behavior. First, device-to-device (D2D) variability is characterized through large-scale resistance statistics across multiple dies (see Supplementary Figures 2 and 3). Second, directional symmetry is examined by comparing AP-to-P and P-to-AP switching probabilities across multiple devices (Supplementary Figures 4 and 5), showing minimal P-AP asymmetry. Third, temporal stability is validated through 1000-cycle repeated measurements on individual devices (Supplementary Figure 6), demonstrating negligible cycle-to-cycle (C2C) variations.



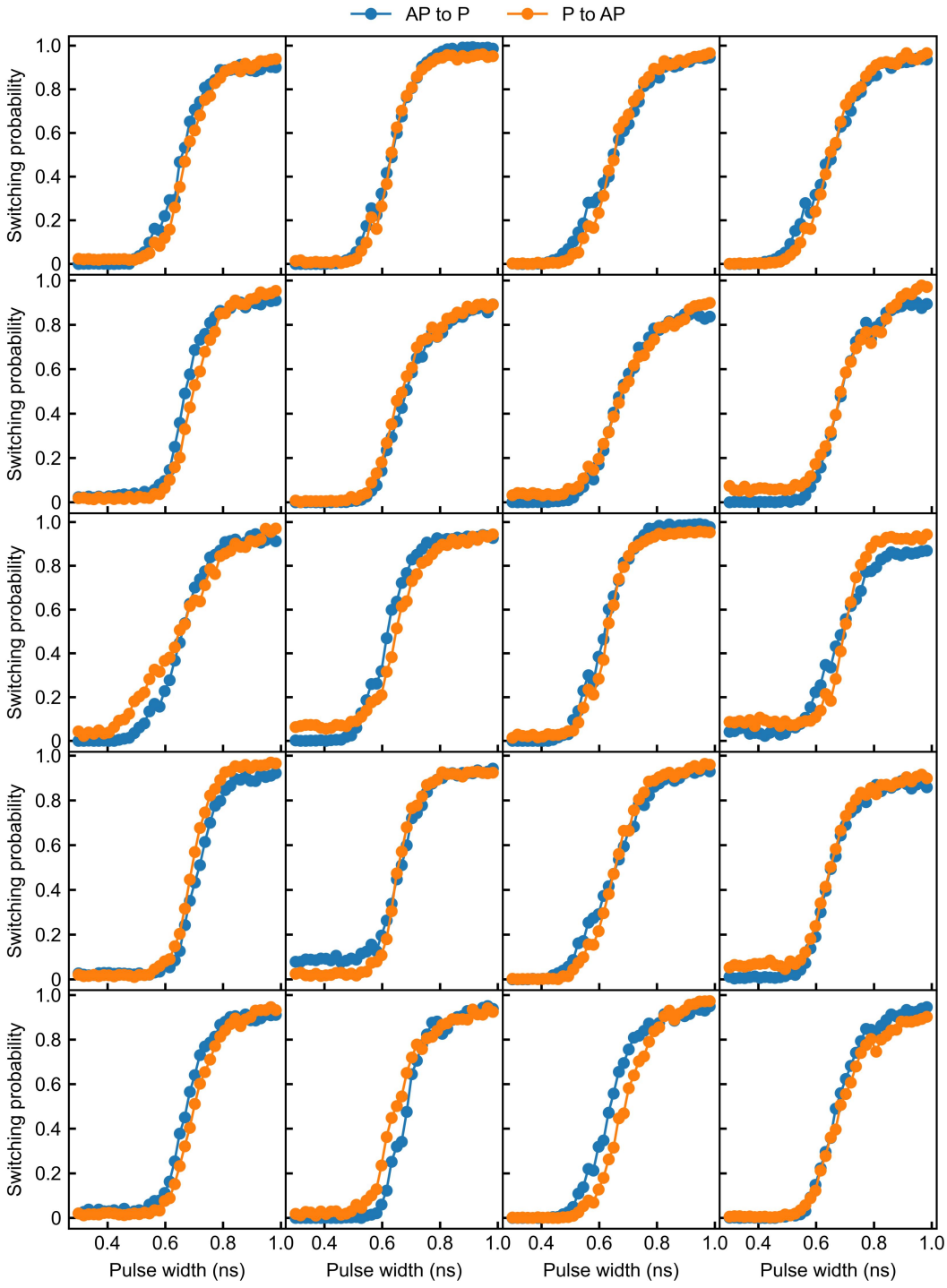
Supplementary Figure 1. Switching probability distribution across 1000 VCMA-MTJs and their calibration. a, Switching probability as a function of sub-nanosecond pulse width. **b,** Sigmoidal fitting results for the switching probability curves. **c,** Realization of a standard sigmoid function using the derived fitting coefficients. **d**, Switching probability under varying voltage pulses (1.6~2.2 V), highlighting the working window of 1.9~2.2 V.



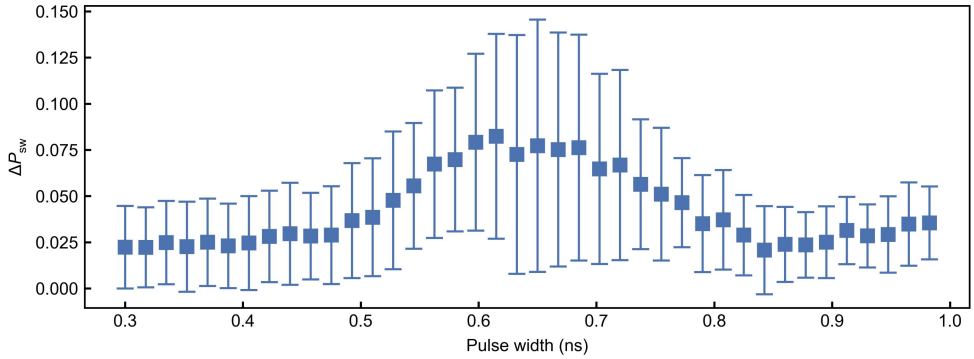
Supplementary Figure 2. Resistance distributions of VCMA-MTJ devices across 8 dies. Each panel shows the R_P and R_{AP} distributions for a single die, with the corresponding TMR shown as an inset.



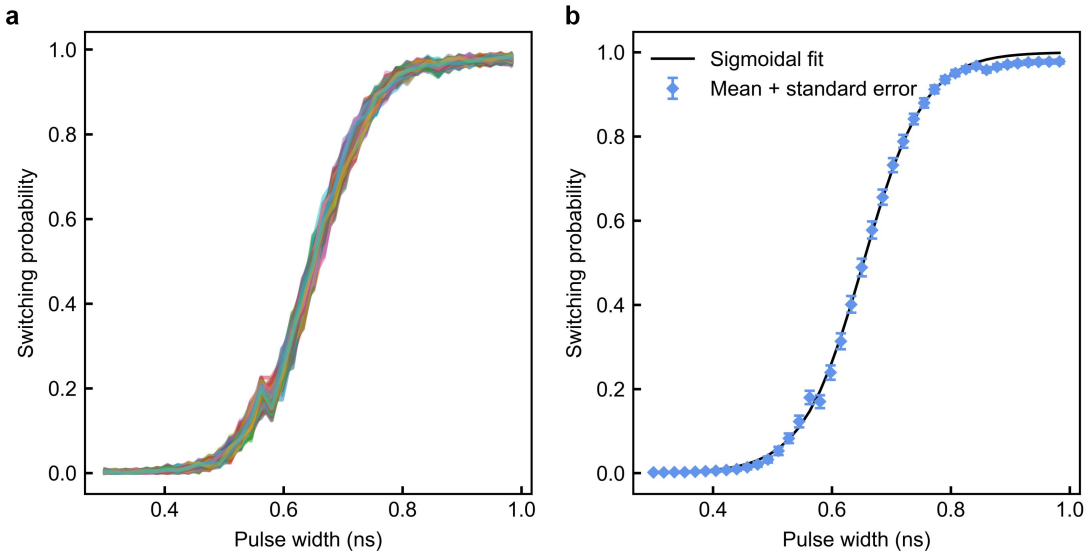
Supplementary Figure 3. Resistance distribution across 135 dies. Aggregated mean R_P and R_{AP} distributions with the corresponding TMR (inset) confirm good die-to-die consistency.



Supplementary Figure 4. AP-to-P and P-to-AP switching probability curves as a function of pulse width for 20 devices.



Supplementary Figure 5. Switching probability difference ΔP_{SW} between P-to-AP and AP-to-P transitions as function of versus pulse width, measured across 20 devices. Error bars represent the standard deviation.

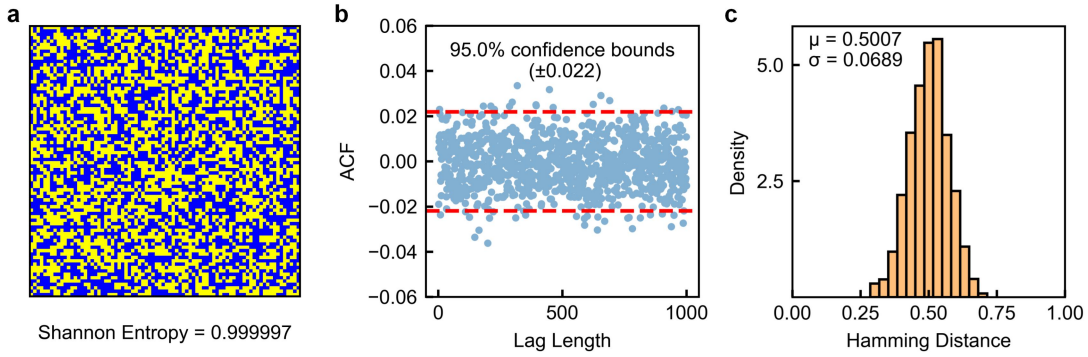


Supplementary Figure 6. C2C stability characterization of the VCMA-MTJ device. a, Overlaid switching probability curves from 1,000 repeated full-sweep measurements, showing tightly clustered traces with minimal variation. **b,** Mean switching probability with standard error bars and sigmoidal fit.

Supplementary Note 2 Quantitative Assessment of True Randomness Generated by VCMA-MTJ

The intrinsic stochasticity of VCMA-MTJs enables the generation of true randomness. To assess the randomness quality of VCMA-MTJs, which is critical for the operation of the Ising machine, we applied consecutive write pulses to the MTJs. The pulse width was set to correspond to a 50% switching probability. After each write pulse, the MTJ state was read out, producing random bit streams (RBSs). To rigorously eliminate any residual static bias and support robust RBS, we implemented an XOR-based generation scheme.

A representative RBS with a length of 6400 is shown in Supplementary Figure 7a, exhibiting a Shannon Entropy of 0.999997. Additionally, we performed an autocorrelation function (ACF) test and a Hamming distance test (Supplementary Figure 7b, c) to further evaluate the randomness. Moreover, the results of the NIST SP800-22 randomness tests, presented in Supplementary Table 1, consistently validate the high randomness quality of the VCMA-MTJ.



Supplementary Figure 7. Randomness quality tests of RBSs generated by VCMA-MTJ. a, Visualization of a RBS with a length of 6400, reshaped into an 80×80 matrix. Blue squares represent 0, and yellow squares represent 1. The Shannon Entropy of the RBS is 0.999997. **b,** ACF test result of a RBS at a 95% confidence level (±0.022). **c,** Inter-sample Hamming distance between multiple RBSs, with a mean value of 0.5.

Test	P-value	Result
Frequency	0.68	Pass
Block Frequency	0.36	Pass
Runs	0.02	Pass
Longest Runs of Ones	0.73	Pass
DFT	0.94	Pass
Non-overlapping Template Matching	0.99	Pass
Serial	0.18	Pass
Approximate Entropy	0.21	Pass
Cumulative Sums	0.76	Pass
Random Excursion	0.25	Pass
Random Excursions Variant	0.31	Pass

Supplementary Table 1. Results of NIST SP800-22 randomness test using RBSs generated by VCMA-MTJ.

Supplementary Note 3 On-Chip Pulse Modulation of VCMA-MTJ as Ising Spin

This supplementary note explains the on-chip pulse width modulation mechanism implemented in the VC-MRAM chip. As shown in Supplementary Figure 8, the pulse width modulation is achieved through an internal programmable delay circuit. When a write operation is initiated, a trigger signal is generated and used to control both the top and bottom electrodes of the VCMA-MTJ. One of these control signals passes through the programmable delay circuit, which determines the duration of the voltage pulse applied across the MTJ.

The timing diagram on the right side of Supplementary Figure 8 illustrates the applied voltage waveforms. The voltage at the top electrode (V_{top}) and bottom electrode (V_{bottom}) define the effective voltage across the MTJ (V_{MTJ}), which governs the switching behavior. By adjusting the delay, the pulse width can be modulated to control the switching probability of the MTJ.

The tunability of write pulse width is essential for on-chip VCMA-MTJ-based Ising computing. Supplementary Figure 9 presents a representative example, illustrating the simulated pulse-width generation with a target duration of 0.3 ns. Furthermore, to provide a comprehensive verification of architecture scalability, we systematically characterized the impact of various hardware non-idealities through post-layout Cadence simulations, including delay-line resolution, array skew, timing jitter, and PVT variations. The results are shown in Supplementary Figure 10.

(1) Delay-Line Resolution

To provide sufficient dynamic range while meeting the stringent timing accuracy required by the steep sigmoid transition, our pulse-generation scheme employs a hybrid coarse-fine delay mechanism:

- Fine resolution control ($\frac{1}{n*RC}$):

A digitally controlled RC network is used to modulate the effective pull-down resistance, enabling fine-grained control of the pulse width. With an 8-bit control, the effective delay resolution is approximately ~1 ps per step.

- Coarse resolution control ($n * t_d$):

A delay-chain-based structure provides coarse delay tuning, with a unit delay of approximately ~8 ps per stage, allowing robust and scalable timing generation.

The overall pulse width is the sum of these two components, enabling both fine resolution (~1 ps) and sufficient dynamic range.

(2) Skew/mismatch across rows/columns

To mitigate pulse-width variation and timing degradation across large arrays, our proposed architecture adopts a localized pulse-generation strategy. Specifically, pulse generators are distributed at the bank level, rather than being globally shared. Local buffering stages are inserted within each bank, ensuring that the generated pulses maintain their edge integrity and amplitude when driving each sub-array and each cell. As the array scales, the system is partitioned into multiple banks, each with its own dedicated delay-line circuitry. This ensures that the critical timing path remains physically short and well-controlled, avoiding long global routing that would otherwise introduce skew, jitter accumulation, and interconnect-induced distortion. Thus, this design ensures that (i) pulse-width variation does not scale with global array size; (ii) timing skew between rows/columns is confined within each subarray; and (iii) pulse shape degradation due to interconnect parasitics is effectively suppressed.

To evaluate pulse integrity across the array, we performed a post-layout critical path analysis using PEX-extracted parasitics. The interconnect and device parasitics along the worst-case path were modeled as distributed π-type RC networks, capturing both resistive and capacitive loading effects. We analyzed the pulse-width variation from the nearest cell to the driver (e.g., cell (0,0)) to the furthest cell in the array under the minimum pulse-width condition. At the minimal pulse configuration of 0.3 ns, the pulse received at the nearest cell is 308 ps; and the maximum pulse variation across the array is less than 7.5 ps, indicating that pulse distortion due to interconnect parasitics is well-controlled.

In addition, the proposed architecture employs an interleaved access scheme, in which adjacent cells are not simultaneously activated, which is common in all memory designs to reduce coupling and avoid simultaneous defect fails. In this design, unselected

columns are held at VSS, which effectively suppresses dynamic coupling between neighboring interconnects.

(3) RMS and Peak-to-Peak Jitter

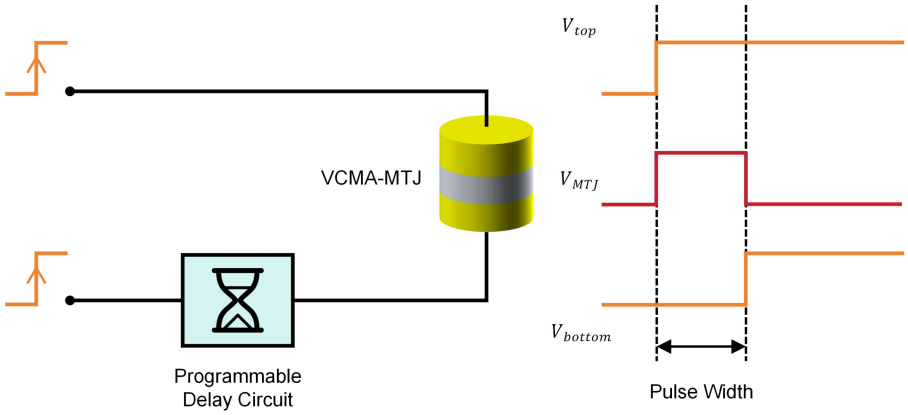
We further characterized the timing uncertainty of the proposed pulse-generation scheme through transient simulations with the minimum delay step configuration. From this, we can infer that the Peak-to-peak jitter is ~1.38 ps and RMS jitter is ~0.146 ps. These results indicate that the timing noise is well-controlled and remains significantly smaller than the intrinsic transition region of the switching probability curve.

(4) PVT variations

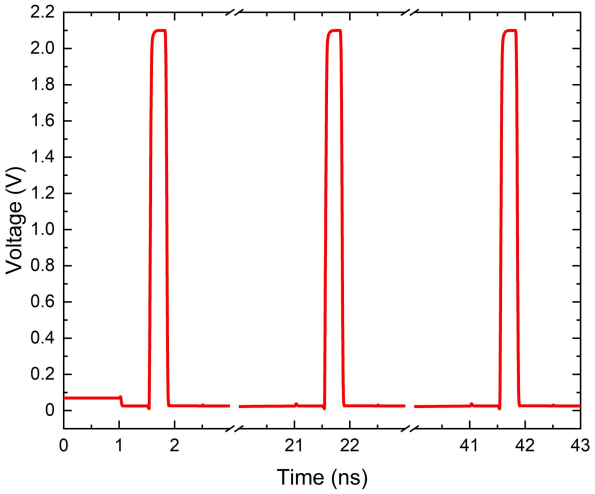
To ensure robust operation under PVT variations, we employ the following methods.

- The delay-line circuitry is powered through a dedicated on-chip low-dropout regulator (LDO), isolating it from global supply fluctuations. This ensures that the delay elements operate under a stable supply voltage, significantly reducing sensitivity to dynamic IR drop and supply noise.
- Process and temperature variations affect both device characteristics and delay elements, potentially shifting the effective pulse width and the resulting switching probability. To address this, the system employs a calibration-based compensation scheme. As discussed earlier, the delay-line configuration is digitally programmable, enabling adjustment of pulse width after fabrication. Calibration is performed based on the observed switching probability of the local array, effectively compensating for process-induced variations. Temperature-induced drift is compensated by re-tuning the delay-line configuration, maintaining the desired mapping between pulse width and switching probability. Since the combined coarse–fine delay architecture ($n * t_d + \frac{1}{n*RC}$) provides both sufficient dynamic range and ps-level resolution, the full range of PVT-induced variation can be covered by calibration and the required timing precision can be maintained.

As a result, the pulse conditions delivered to the device are effectively compensated for PVT variations, ensuring that the correct switching probability is reliably generated across all operating conditions.

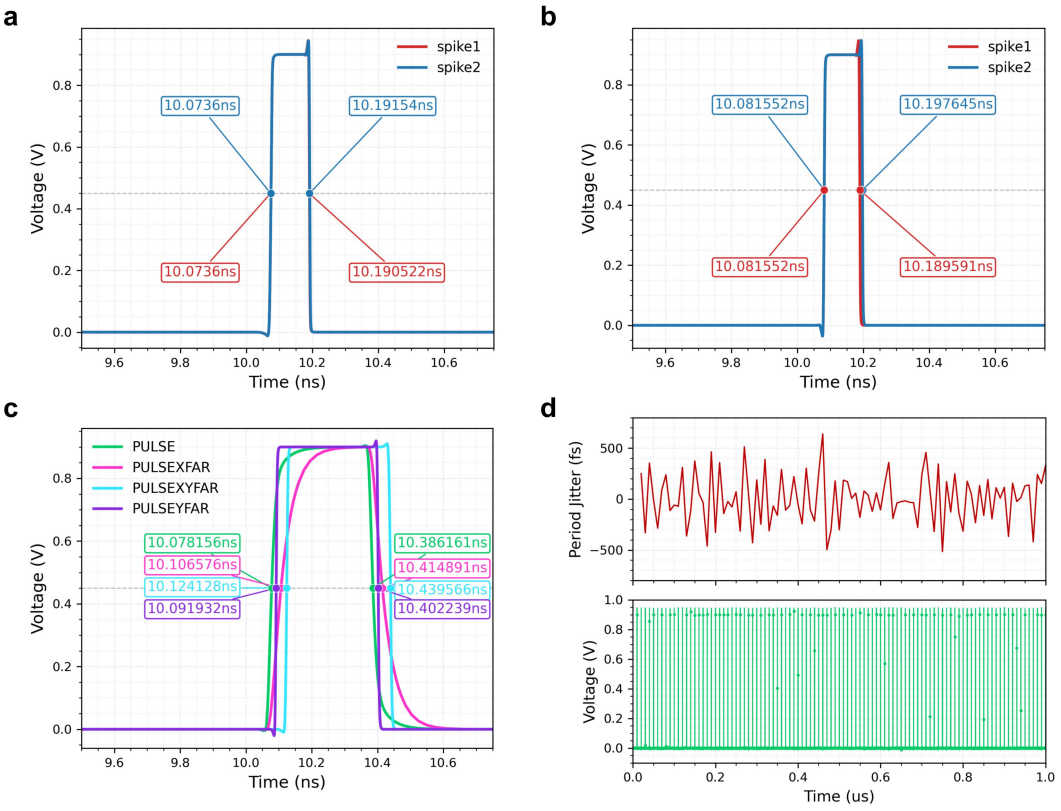


Supplementary Figure 8. On-chip pulse width modulation mechanism for VCMA-MTJ switching control. A programmable delay circuit adjusts the relative timing of the voltage signals applied to the top and bottom electrodes of the VCMA-MTJ, thereby modulating the pulse width of V_{MTJ}.



.

Supplementary Figure 9. Sub-nanosecond writing of VCMA-MTJ for ultra-fast Ising spin update. Circuit simulation of the write pulses exhibits a width of less than 1 ns (W_v ~0.3 ns in the figure as an example), enabling sub-nanosecond probabilistic spin updates. The periodic pulses (V_{MTJ} ~2.1 V) demonstrate the feasibility of fast and energy-efficient MTJ operation in the Ising machine.



Supplementary Figure 10. Cadence simulations showing the generated pulse resolution. a, the minimum pulse resolution for the $\frac{1}{n*RC}$ is ~1 ps. **b,** the minimum resolution for the $n * t_d$ is: ~8 ps. **c,** Pulse integrity evaluation across the array. **d,** Peak-to-peak jitter is ~1.38 ps and RMS jitter is ~0.146 ps.

Supplementary Note 4 Ising Mapping of Global Routing Problem

The global routing problem can be modeled as a rectilinear Steiner minimum tree (RSMT) problem. The total Hamiltonian for the RSMT problem is stated as:

$$H_{RSMT}(x) = H_o + \lambda_1 H_{c1} + \lambda_2 H_{c2} + \lambda_3 H_{c3} + \lambda_4 H_{c4}$$

where H_o represents the optimization objective, specifically the total cost of all edges in the Steiner tree:

$$H_o = \sum_{\substack{(u,v)\in E,\\ v\neq v_0}} c_{u,v} x^e_{u,v}$$

H_{c1}~H_{c4} denote four essential constraints for Steiner tree formation, with corresponding penalty coefficients λ_1~λ_4 that must be carefully calibrated for optimal performance. These constraints serve distinct purposes. H_{c1} ensures that each terminal vertex (except the root) has precisely one incoming arc:

$$H_{c1} = \sum_{v\in U\setminus\{v_0\}} (1 - \sum_{(u,v)\in E} x^e_{u,v})^2$$

H_{c2} enforces acyclicity, a fundamental requirement for tree structures:

$$H_{c2} = \sum_{\substack{1\le u<v<w\le|V|,\\ u,v,w\neq v_0}} (x^o_{u,w} + x^o_{u,v}x^o_{v,w} - x^o_{u,v}x^o_{u,w} - x^o_{u,w}x^o_{v,w}) + \sum_{\substack{(u,v)\in E,\\ u<v,\\ u,v\neq v_0}} (x^e_{u,v}(1 - x^o_{u,v}) + x^e_{v,u}x^o_{u,v})$$

H_{c3} penalizes configurations where Steiner vertices either have multiple incoming arcs or possess outgoing arcs without any incoming ones:

$$H_{c3} = |V| \sum_{v\in W} (\sum_{(u,v),(w,v)\in E} x^e_{u,v}x^e_{w,v}) + \sum_{v\in W} (1 - \sum_{(u,v)\in E} x^e_{u,v})(\sum_{(u,v)\in E} x^e_{v,u})$$

H_{c4} addresses the rectilinearity constraint specific to routing scenarios. When two vertices u and v cannot be connected by a straight segment (denote as $r_{u,v} = 1$, otherwise $r_{u,v} = 0$), a positive penalty is imposed:

$$H_{c4} = \sum_{\substack{(u,v)\in E,\\ v\neq v_0}} r_{u,v} x^e_{u,v}$$

The spin variables s in the Ising model are derived through a composite transformation of x^e and x^o:

$$s = \begin{pmatrix} s^e \\ s^o \end{pmatrix} = 2 * \begin{pmatrix} x^e \\ x^o \end{pmatrix} - 1$$

where s^e and s^o are Ising spins for edge and order variables, respectively.

For convenience, denote $M = |U|, N = |V|$. All vertices are numbered sequentially from 0 to $N-1$, with vertex 0 assumed to be the root. The Ising Hamiltonian can be written as:

$$H_{Ising} = -\sum_{i,j,m,n} J_{i,j,m,n} s_{i,j} s_{m,n} - \sum_{i,j} h_{i,j} s_{i,j}$$

The 'interactions' and 'external field' brought by the objective term and all constraint terms in H_{RSMT} are denoted as $\boldsymbol{J^0}, \boldsymbol{h^0}, \boldsymbol{J^1}, \boldsymbol{h^1}, \ldots \boldsymbol{J^4}, \boldsymbol{h^4}$, respectively. These terms are then subjected to variable substitution in sequence:

(1)

$$H_o = \sum_{1 \le j < N} \sum_{0 \le i < N, i \ne j} c_{i,j} \cdot \frac{s_{i,j} + 1}{2}$$

$$J^0_{i,j,m,n} = 0$$

$$h^0_{i,j} = -\frac{c_{i,j}}{2}, \quad 0 \le i < N, 1 \le j < N, i \ne j$$

(2)

$$H_{c1} = \sum_{1 \le j < M} (1 - \sum_{0 \le i < N, i \ne j} \frac{s_{i,j} + 1}{2})^2$$

$$= \frac{1}{4} \sum_{1 \le j < M} (N - 3 + \sum_{0 \le i < N, i \ne j} s_{i,j})^2$$

$$= \frac{1}{4} \sum_{1 \le j < M} [(\sum_{0 \le i < N, i \ne j} s_{i,j})^2 + 2(N-3) \sum_{0 \le i < N, i \ne j} s_{i,j}]$$

$$= \frac{1}{2} \sum_{1 \le j < M} \sum_{\substack{0 \le i < m < N, \\ i \ne j, m \ne j}} s_{i,j} s_{m,j} + \frac{N-3}{2} \sum_{1 \le j < M} \sum_{0 \le i < N, i \ne j} s_{i,j}$$

$$J^1_{i,j,m,n} = -\frac{1}{2}\delta(j,n), \quad 0 \le i < m < N, 1 \le j, n < M, i \ne j, m \ne n$$

$$h^1_{i,j} = -\frac{N-3}{2}, \quad 0 \le i < N, 1 \le j < M, i \ne j$$

(3)

$$H_{c2,1} =$$

$$\sum_{1\le i<j<k<N}\left(\frac{s_{N+i,k}+1}{2}+\frac{s_{N+i,j}+1}{2}\cdot\frac{s_{N+j,k}+1}{2}-\frac{s_{N+i,j}+1}{2}\cdot\frac{s_{N+i,k}+1}{2}-\frac{s_{N+i,k}+1}{2}\cdot\frac{s_{N+j,k}+1}{2}\right)$$

$$=\frac{1}{4}\sum_{1\le i<j<k<N} s_{N+i,j}s_{N+j,k}-s_{N+i,j}s_{N+i,k}-s_{N+i,k}s_{N+j,k}$$

$$J^{2,1}_{i,j,m,n}=-\frac{1}{4}[\delta(j+N,m)\chi(i-N,j)\chi(j,n)-\delta(i,m)\chi(i-N,j)\chi(j,n)$$

$$-\delta(j,n)\chi(i,m)\chi(m-N,n)],\ \ N+1\le i,m<2N,1\le j,n<N$$

$$h^{2,1}_{i,j}=0$$

(4)

$$H_{c2,2}=\sum_{1\le i<j<N}\frac{s_{i,j}+1}{2}\cdot\frac{1-s_{N+i,j}}{2}+\frac{s_{j,i}+1}{2}\cdot\frac{s_{N+i,j}+1}{2}$$

$$=\frac{1}{4}\sum_{1\le i<j<N} s_{i,j}+s_{j,i}-s_{i,j}s_{N+i,j}+s_{j,i}s_{N+i,j}$$

$$J^{2,2}_{i,j,m,n}=$$

$$\frac{1}{4}[\delta(m,N+i)\delta(n,j)\chi(i,j)-\delta(m,N+j)\delta(n,i)\chi(j,i)],\ \ 1\le i,j,n<N,N+1\le m<2N$$

$$h^{2,2}_{i,j}=-\frac{1}{4},\ \ 1\le i\ne j<N$$

(5)

$$H_{c3,1}=N\sum_{M\le j<N}\sum_{\substack{0\le i<m<N,\\ i\ne j,m\ne j}}\frac{s_{i,j}+1}{2}\cdot\frac{s_{m,j}+1}{2}$$

$$=\frac{N}{4}\sum_{M\le j<N}\sum_{\substack{0\le i<m<N,\\ i\ne j,m\ne j}} s_{i,j}s_{m,j}+s_{i,j}+s_{m,j}$$

$$=\frac{N}{4}\sum_{M\le j<N}\sum_{\substack{0\le i<m<N,\\ i\ne j,m\ne j}} s_{i,j}s_{m,j}+\frac{N(N-2)}{4}\sum_{M\le j<N}\sum_{\substack{0\le i<N,\\ i\ne j}} s_{i,j}$$

$$J^{3,1}_{i,j,m,n}=-\frac{N}{4}\delta(j,n),\ \ 0\le i<m<N,M\le j,n<N,i\ne j,m\ne n$$

$$h^{3,1}_{i,j}=-\frac{N(N-2)}{4},\ \ 0\le i<N,M\le j<N,i\ne j$$

(6)

$$H_{c3,2}=\sum_{M\le j<N}\left(1-\sum_{0\le i<N,i\ne j}\frac{s_{i,j}+1}{2}\right)\left(\sum_{1\le i<N,i\ne j}\frac{s_{j,i}+1}{2}\right)$$

$$= \frac{1}{4} \sum_{M \le j < N} (3 - N - \sum_{0 \le i < N, i \ne j} s_{i,j})(\sum_{1 \le i < N, i \ne j} s_{j,i} + N - 2)$$

$$= - \frac{N-3}{4} \sum_{M \le j < N} \sum_{1 \le i < N, i \ne j} s_{j,i} - \frac{N-2}{4} \sum_{M \le j < N} \sum_{0 \le i < N, i \ne j} s_{i,j} - \frac{1}{4} \sum_{M \le j < N} \sum_{0 \le i < N, i \ne j} \sum_{1 \le k < N, k \ne j} s_{i,j} s_{j,k}$$

$$J^{3,2}_{i,j,m,n} = \frac{1}{4} [\delta(j,m)\chi(M-1,j) + \delta(i,n)\chi(M-1,i)],\ 0 \le i < m < N, 1 \le j, n < M, i \ne j, m \ne n$$

$$h^{3,2}_{i,j} = \frac{N-2}{4} \chi(M-1,j) + \frac{N-3}{4} \chi(M-1,i),\ \ 0 \le i < N, 1 \le j < N, i \ne j$$

(7)

$$H_{c4} = \sum_{1 \le j < N} \sum_{0 \le i < N, i \ne j} r_{i,j} \cdot \frac{s_{i,j} + 1}{2}$$

$$J^{4}_{i,j,m,n} = 0$$

$$h^{4}_{i,j} = - \frac{r_{i,j}}{2},\ \ 0 \le i < N, 1 \le j < N, i \ne j$$

In the derivation above, constant terms are omitted from the expressions, as the annealing process in the Ising machine only focuses on the relative changes in the Hamiltonian and not its absolute value. The definitions of the two functions $\delta(i,j)$ and $\chi(i,j)$ used are as follows.

$$\delta(i,j) = \begin{cases} 1, i = j \\ 0, i \ne j \end{cases}$$

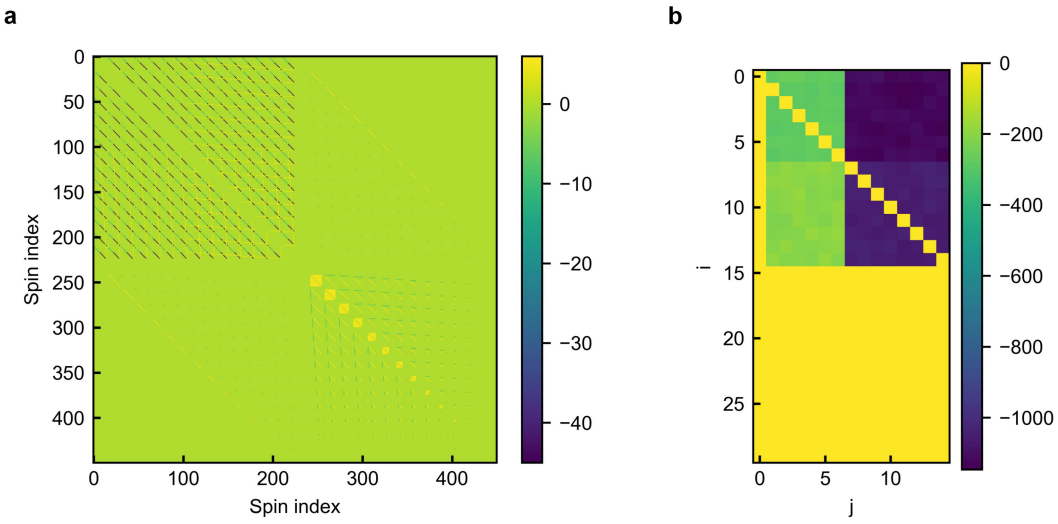
$$\chi(i,j) = \begin{cases} 1, i < j \\ 0, i \ge j \end{cases}$$

By summing all components, the total $\boldsymbol{J}$ and $\boldsymbol{h}$ can be obtained:

$$\boldsymbol{J} = \boldsymbol{J^0} + \lambda_1 \boldsymbol{J^1} + \lambda_2 (\boldsymbol{J^{2,1}} + \boldsymbol{J^{2,2}}) + \lambda_3 (\boldsymbol{J^{3,1}} + \boldsymbol{J^{3,2}}) + \lambda_4 \boldsymbol{J^4}$$

$$\boldsymbol{h} = \boldsymbol{h^0} + \lambda_1 \boldsymbol{h^1} + \lambda_2 (\boldsymbol{h^{2,1}} + \boldsymbol{h^{2,2}}) + \lambda_3 (\boldsymbol{h^{3,1}} + \boldsymbol{h^{3,2}}) + \lambda_4 \boldsymbol{h^4}$$

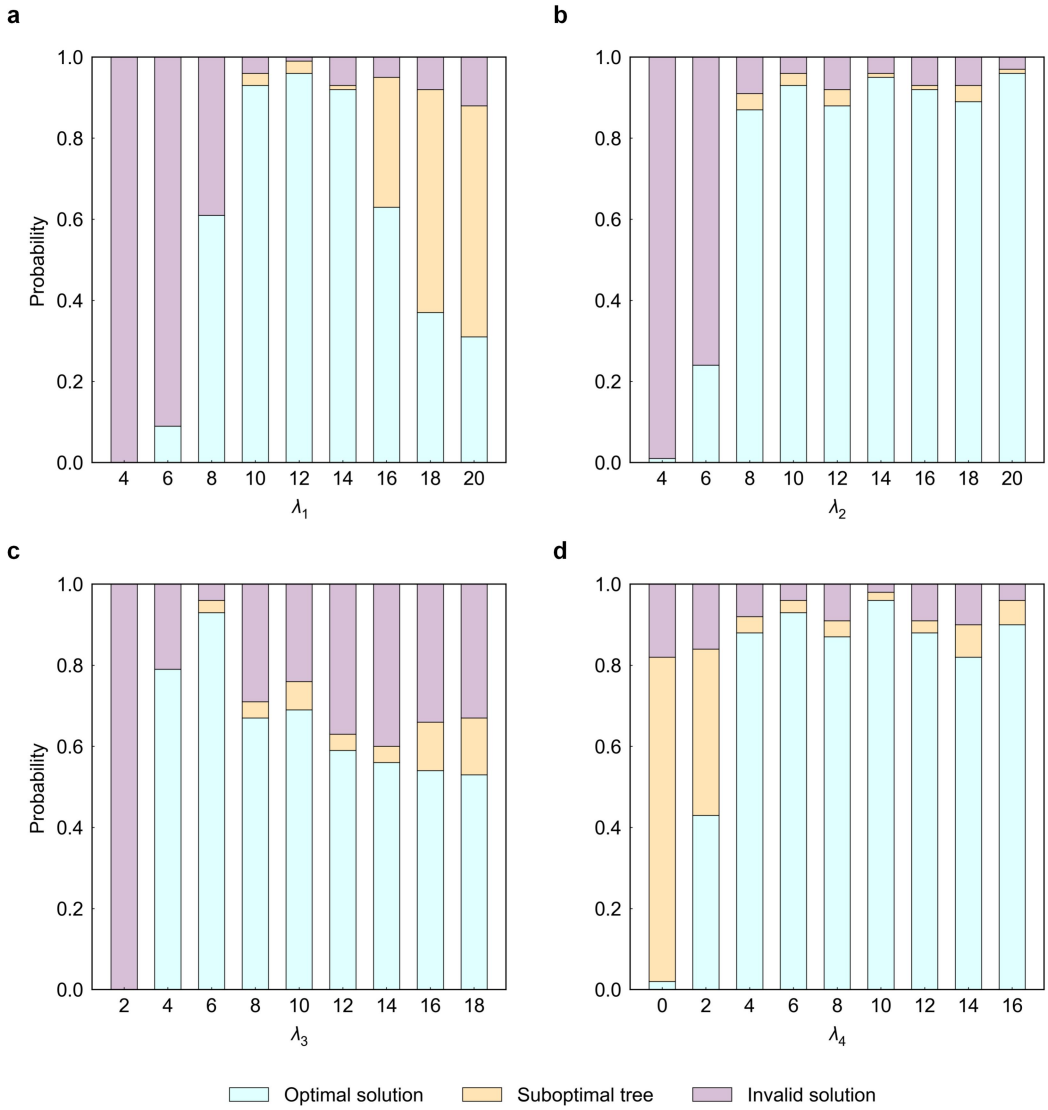
The Ising mapping results corresponding to the global routing problem in Fig. 4a (main text) is demonstrated in Supplementary Figure 11.



Supplementary Figure 11. Ising mapping results of a global routing problem. a, Heatmap of the coupling matrix $\boldsymbol{J}$, visualized in a 2D representation after dimensionality reduction from the original 4D space for enhanced interpretability. **b,** Heatmap of the external field $\boldsymbol{h}$, illustrating the local bias terms in the Ising model.

Supplementary Note 5 Impact of Penalty Coefficients on Global Routing Solution Probability

The selection of appropriate penalty coefficients is crucial for solving COPs using an Ising machine. In the context of the global routing problem, insufficient penalty values fail to enforce the necessary constraints for valid Steiner tree construction, while excessively large values hinder the optimization of the total cost. Supplementary Figure 12 (the same as the Extended Data Fig. 3) illustrates how the solution probability varies with different configurations of the four penalty coefficients ($\lambda_1 \sim \lambda_4$).



Supplementary Figure 12. Solution probability of global routing problem varying with penalty coefficient settings. The initial configuration is $\lambda_1 = \lambda_2 = 10$ and $\lambda_3 = \lambda_4 = 6$. In panels **a, b, c,** and **d,** only one of these coefficients is varied while the others remain fixed.

For λ_1, which controls the constraint on the number of incoming arcs at terminal vertices, a value that is too small results in invalid solutions due to the failure to form a proper tree structure (Supplementary Figure 12a). Conversely, an excessively large λ_1 creates a conflict between minimizing the total cost and satisfying the constraint, limiting the exploration of better solutions.

λ_2, associated with enforcing acyclicity, exhibits a similar behavior to λ_1: a value that is too small leads to invalid solutions by failing to guarantee the tree structure (Supplementary Figure 12b). However, unlike λ_1, increasing λ_2 does not reduce the solution probability, as minimizing the total cost and satisfying the acyclicity constraint are not inherently conflicting objectives.

λ_3 penalizes unreasonable arcs at Steiner vertices. As with λ_1 and λ_2, a value that is too small results in invalid solutions (Supplementary Figure 12c). Additionally, an excessively large λ_3 diminishes the relative importance of λ_1 and λ_2, increasing the likelihood of failing to form a valid tree.

Finally, λ_4 enforces the rectilinearity constraint specific to routing scenarios. Supplementary Figure 12d highlights the importance of λ_4: a value that is too small leads to suboptimal tree solutions. Increasing λ_4 does not reduce the solution probability, as the rectilinearity constraint neither conflicts with minimizing the total cost nor with satisfying the other constraints.

Supplementary Note 6 Ising Mapping of Layer Assignment Problem

When both reducing wire density and minimizing the usage of vias are considered, the layer assignment problem becomes a multi-objective optimization problem. The total Hamiltonian for the DVLA problem can be expressed as

$$H_{DVLA} = \lambda_D H_D + \lambda_V H_V$$

where H_D and H_V represent the density-driven and via-aware contributions to the overall objective, respectively. The coefficients λ_D and λ_V denote the relative importance of each component in the optimization process.

Let Ising spin variables $s = [s^h\ s^v]$ directly encode the layer assignment for each segment. The density-driven term H_D is expressed as[3]:

$$H_D = \frac{1}{2}\sum_{0\le i<j<N_h} w_{i,j}^h(s_i^h s_j^h - 1) + \frac{1}{2}\sum_{0\le m<n<N_v} w_{m,n}^v(s_m^v s_n^v - 1)$$

where N_h and N_v denote the numbers of horizontal and vertical segments, respectively, and w^h and w^v denote local density matrices of horizontal and vertical segments, respectively. In the case of H_V, vias are used when a horizontal segment and a vertical one need to be connected, or when a segment must reach a pin (assuming that all pins are located on L_0). The via-aware term H_V can be expressed as:

$$H_V = \sum_{\substack{0\le i<N_h,\\ 0\le m<N_v}} c_{i,m}\left[3\left(\frac{s_m^v+1}{2} - \frac{s_i^h+1}{2}\cdot\frac{s_m^v+1}{2}\right) + \left(\frac{s_i^h+1}{2}\right) + \left(\frac{-s_i^h+1}{2}\cdot\frac{-s_m^v+1}{2}\right)\right] + \sum_{0\le i<N_h} p_i^h(s_i^h+2) + \sum_{0\le m<N_v} p_m^v(s_m^v+3)$$

where $c_{i,m}$ indicates whether the i-th horizontal segment and the m-th vertical segment are connected, and p_i^h/p_m^v denote the flag indicating if a segment must reach a pin. The first term of H_V addresses all scenarios where a horizontal segment and a vertical segment require connection. Three vias are necessary only when the horizontal segment is on layer L_1 and the vertical segment is on layer L_4; in all other cases, only one via is required. The second and third terms of H_V cover situations where a segment needs to connect to a pin. For horizontal segments, the number of vias needed is $s_i^h + 2$; while for vertical segments, the number of vias needed is $s_m^v + 3$.

The Ising Hamiltonian can be written as

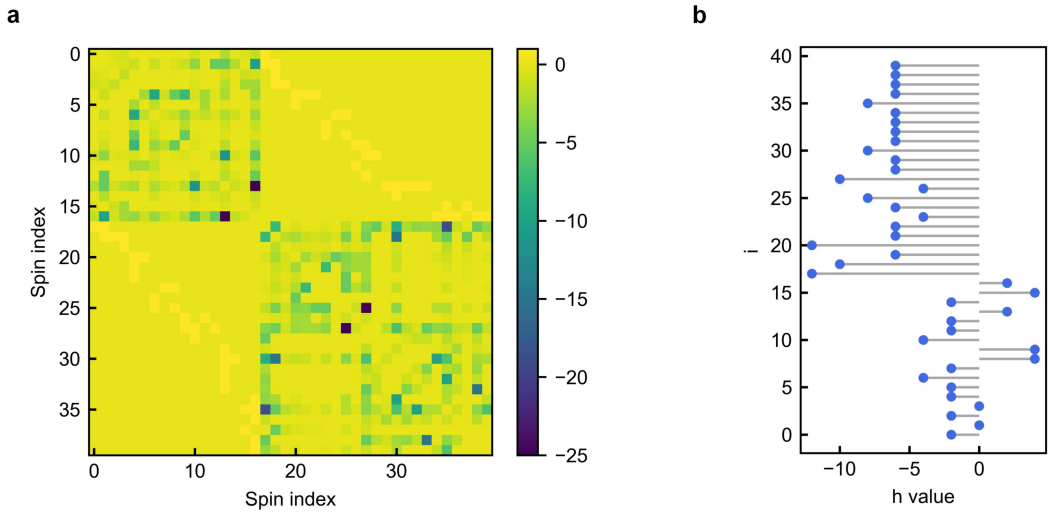
$$H_{Ising} = -\sum_{i,j} J_{i,j} s_i s_j - \sum_i h_i s_i$$

By comparing the Ising Hamiltonian with H_{DVLA}, the expressions for $\boldsymbol{J}$ and $\boldsymbol{h}$ can be derived as follows:

$$J_{i,j} = \begin{cases} -\frac{\lambda_D}{2} w^h_{i,j}, 0 \le i < j < N_h \\ -\frac{\lambda_D}{2} w^v_{i-N_h, j-N_h}, N_h \le i < j < N_h + N_v \\ \frac{\lambda_V}{2} c_{i,j-N_h}, 0 \le i < N_h \; and \; N_h \le j < N_h + N_v \\ 0, others \end{cases}$$

$$h_i = \begin{cases} \lambda_V \left(-p^h_i + \frac{1}{2} \sum_{N_h \le j < N_h + N_v} c_{i,j-N_h} \right), 0 \le i < N_h \\ \lambda_V \left(-p^v_i - \frac{1}{2} \sum_{0 \le j < N_h} c_{j,i-N_h} \right), N_h \le i < N_h + N_v \\ 0, others \end{cases}$$

The matrix $\boldsymbol{J}$ obtained above is an upper triangular matrix, and it needs to be symmetrized to obtain the final $\boldsymbol{J}$. The Ising mapping results corresponding to the DVLA problem in Fig. 5a (main text) is demonstrated in Supplementary Figure 13.



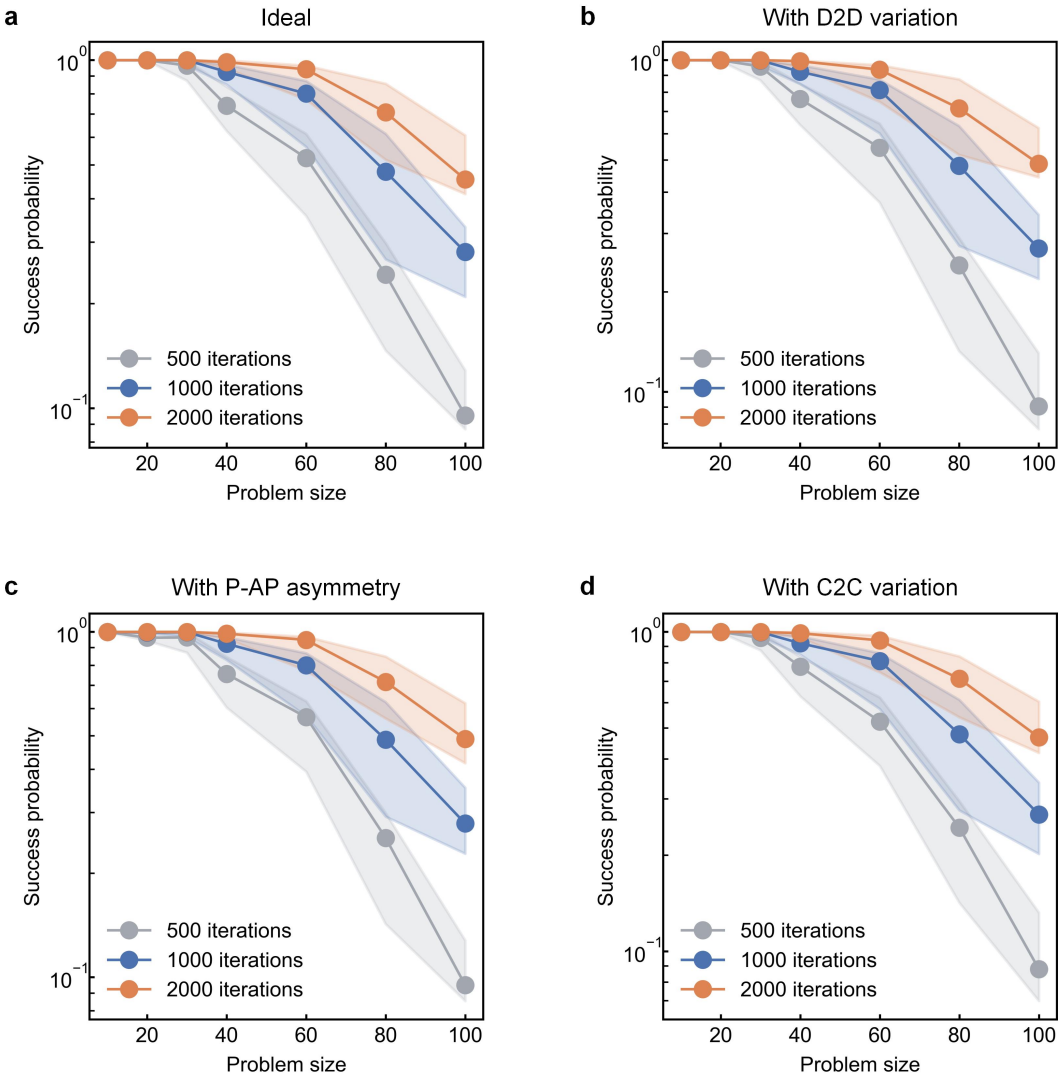
Supplementary Figure 13. Ising mapping results of a DVLA problem. a, Heatmap of the coupling matrix $\boldsymbol{J}$. **b,** Stem plot of the external field $\boldsymbol{h}$.

Supplementary Note 7 Estimation of the Impact of Device Variation on the Solution Quality

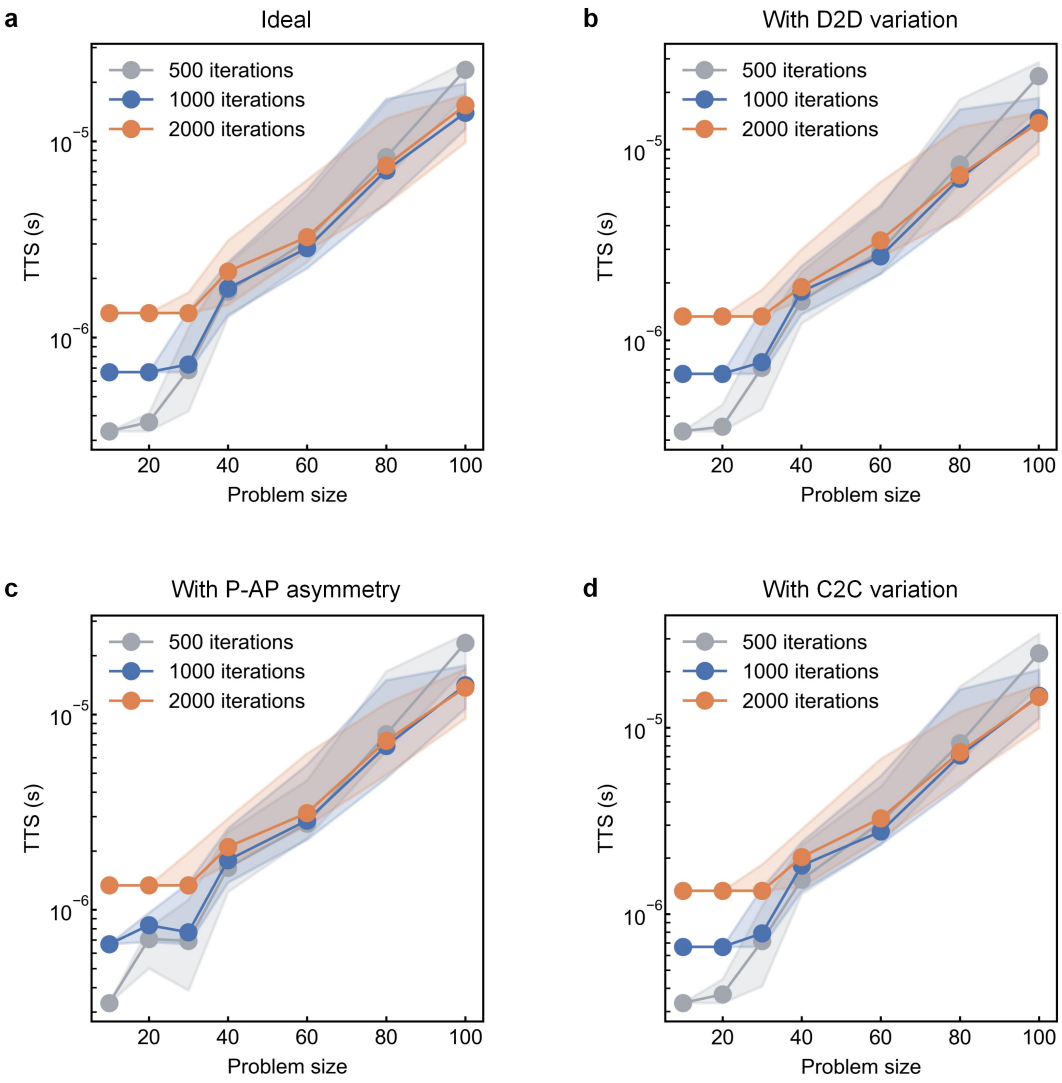
We systematically evaluate both static and dynamic device variations, and quantify their impact on probabilistic switching behavior in Supplementary Note 1. In this section, we extend this analysis to assess the impact of these device-level non-idealities on the system-level performance of COP performance across different problem sizes, including D2D variation, P-AP asymmetry, and C2C variation. For rigorous and consistent comparison with prior work, we normalize the performances to a common benchmark built well by previous works[4–7]: solving dense Max-cut problems with 50% connectivity. As shown in Supplementary Figure 14, we observe that the success probability of the VSIM scales as e^{-N} with problem size N.

To evaluate the computational speed of the VSIM, we employ the time to solution (TTS) metric. TTS is defined as the computation time for finding the optimal solution to a COP (in this case, Max-cut problems) with 99% probability[8], which can be formulated as $T_{com}log(1-0.99)/log(1-P_s)$, where T_{com} is the computation time with a success probability of P_s.

Crucially, under the reasonable assumption that FPGA operations (i.e., peripheral auxiliary modules) can ultimately be optimized through advanced ASIC implementations, the TTS estimations in the Supplementary Figures 15 are based on the projected VSIM (The projected T_{com} is detailed in Supplementary Note 10). The corresponding TTS results (see Supplementary Figures 15) confirm that the device variations introduce only minor perturbations and do not degrade the computational performance of the VSIM.



Supplementary Figure 14. VSIM success probability versus problem size under (a) ideal, (b) D2D variation, (c) P-AP asymmetry, and (d) C2C variation conditions. Results are shown for 500, 1000, and 2000 iterations on Max-cut problems. Ten instances are tested at each problem size. Shaded regions indicate the interquartile range.



Supplementary Figure 15. TTS versus problem size under (a) ideal, (b) D2D variation, (c) P-AP asymmetry, and (d) C2C variation conditions. Results are shown for 500, 1000, and 2000 iterations on Max-cut problems. Ten instances are tested at each problem size. Shaded regions indicate the interquartile range.

Supplementary Note 8 Speed, Area and Energy Estimation of Ising Spin Implemented by FPGA

As shown in Supplementary Figure 16, we utilize a 16-bit LFSR, an 5-bit-input/16-bit-output sigmoid look-up table and a 16-bit digital comparator to implement a Ising spin in the FPGA. To ensure a fair comparison, a 5-bit input sigmoid look-up table is used, considering the MTJ switching probability curve is fitted with around 30 data points.

The FPGA employed in this work is the programmable logic portion of the Zynq-7000, fabricated using a 28 nm process. The implementation results in terms of timing, resource utilization, and power consumption are presented in Supplementary Table 2. The speed, area, and energy consumption of the FPGA-based Ising spin are estimated as follows:

(1) Speed: The critical path delay is calculated as 5.683 ns × 2 ≈ 11 ns.

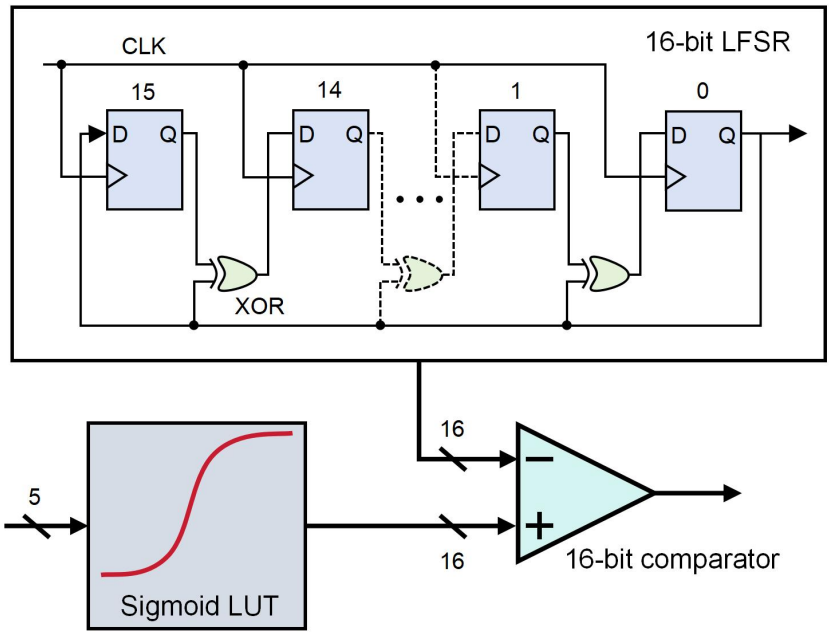
(2) Area:

- The BRAM is used to store discrete values on the sigmoid curve. In this experiment, an 5-bit input and 16-bit output sigmoid look-up table is employed, assuming each storage unit is a 6-transistor SRAM.
- The flip-flops (FFs) are assumed to be edge-triggered D flip-flops, each composed of 16 transistors.
- The look-up tables (LUTs) are assumed to be 4-input and 8-bit wide, with each bit implemented as a 6-transistor SRAM.

Therefore, the total transistor count is calculated as:

$$2^5 \times 16 \times 6 + 20 \times 16 + 78 \times 4 \times 8 \times 6 = 18368$$

(3) Energy consumption: The energy consumption is estimated as 11 ns × 540 μW = 5.94 pJ.



Supplementary Figure 16. FPGA implementation of Ising spin. A 16-bit LFSR generates random numbers, which are compared with the output of a sigmoid LUT to emulate the sigmoidal probabilistic behavior of Ising spin.

Timing		Utilization			Power
Estimated clock	Max clock cycles	BRAM	FF	LUT	
5.683 ns	2	1	20	78	540 μW

Supplementary Table 2. Timing, utilization, and power reports of Ising spin implemented by FPGA.

Supplementary Note 9 Speed, Area and Energy Estimation of Ising Spin Implemented by VCMA-MTJ

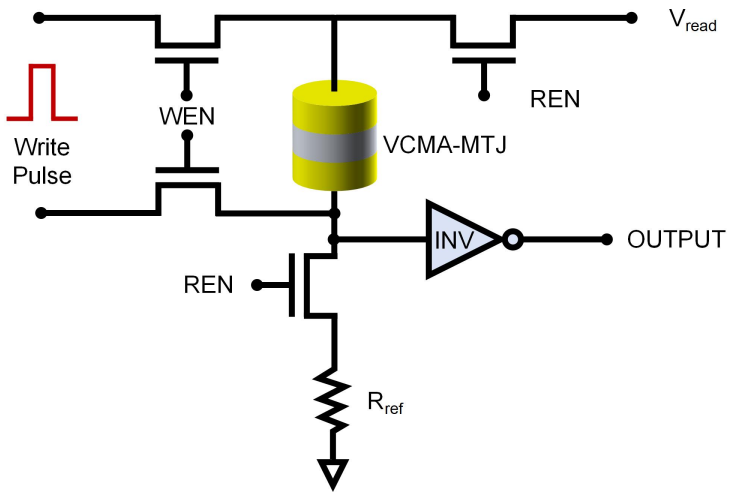
The VCMA-MTJ-based Ising spin, illustrated in Supplementary Figure 17, comprises a VCMA-MTJ, four transistors for R/W control, a reference resistor, and an inverter (INV).

Its performance metrics are estimated as follows:

(1) Speed: Leveraging the sub-nanosecond probabilistic switching characteristic of the VCMA-MTJ, the delay is maximally evaluated to be as the 1 ns.

(2) Area: The transistor count includes 4 from the R/W control and 2 from the INV, resulting in a total of 6 transistors.

(3) Energy consumption: Given the resistance range of VCMA-MTJs is 80–150 kΩ, a median value of 100 kΩ is used for energy estimation. With a write pulse configuration of 2.0 V and a maximum duration of 1 ns, the energy consumption is calculated as: $(2.0\ \text{V})^2$ / 100 kΩ × 1 ns = 40 fJ.



Supplementary Figure 17. VCMA-MTJ-based implementation of Ising spin. By leveraging the intrinsic stochasticity of the VCMA-MTJ, a compact Ising spin is realized. The ultra-fast writing capability of VCMA-MTJ ensures high efficiency in Ising computing.

Supplementary Note 10 Latency and Power Analysis of VSIM Implementation

The VSIM hardware system consists primarily of a VC-MRAM chip, which provides the probabilistic Ising spins, and an FPGA, which handles spin coupling and other auxiliary operations.

The decomposition of one iteration time is listed in Supplementary Table 3, neglecting the PCB I/O latency. Ising spin flipping is implemented in-situ through the VCMA-MTJ with a voltage-controlled probability, which consumes less than 1 ns. For Ising machine demonstrations, where only binary spin states need to be distinguished rather than precise resistance values, off-chip high-precision ADCs are not required for the readout. Instead, the MTJ states are directly read on-chip with digital signal outputs, consuming approximately 4 ns per read operation. Ising spin coupling is implemented digitally in the programmable logic fabric of the Zynq-7000 SoC on a Xilinx PYNQ-Z2 board operating at 100 MHz, with a power consumption of 40 mW verified by Vivado analysis. The multiply-and-accumulate (MAC) operation consumes ~20 ns, which is accelerated by a flip-driven incremental method[9]. For $\delta H = -2s_i(\sum_j J_{i,j}s_j + h_i)$, its initial value is precalculated and stored in registers. When a spin j is flipped, the change $-4J_{i,j}s_i s_j$ is added to the δH. Other FPGA-based functions, including annealing schedule and δH-to-W_v conversion, require about 20 ns.

To evaluate the potential of a fully integrated VSIM System-on-Chip, we analyze performance metrics based on experimentally grounded projections, where the latency currently dominated by programmable logic can be substantially mitigated through a dedicated ASIC implementation (see Supplementary Table 4). In embedded VC-MRAM chips, a fundamental trade-off exists between array efficiency and read latency. By optimizing the bit-line length through a mini-array configuration, substantial acceleration in readout speed is achievable[10]. Specifically, in-situ spin flipping can be reduced to less than 0.8 ns, comprising a 0.5 ns voltage-controlled MTJ write pulse[11,12] and a 0.3 ns high-speed readout (Cadence simulation results are detailed in Supplementary Figure 18). For the subsequent Ising spin coupling operation, prior research on Ising machines[13–15]

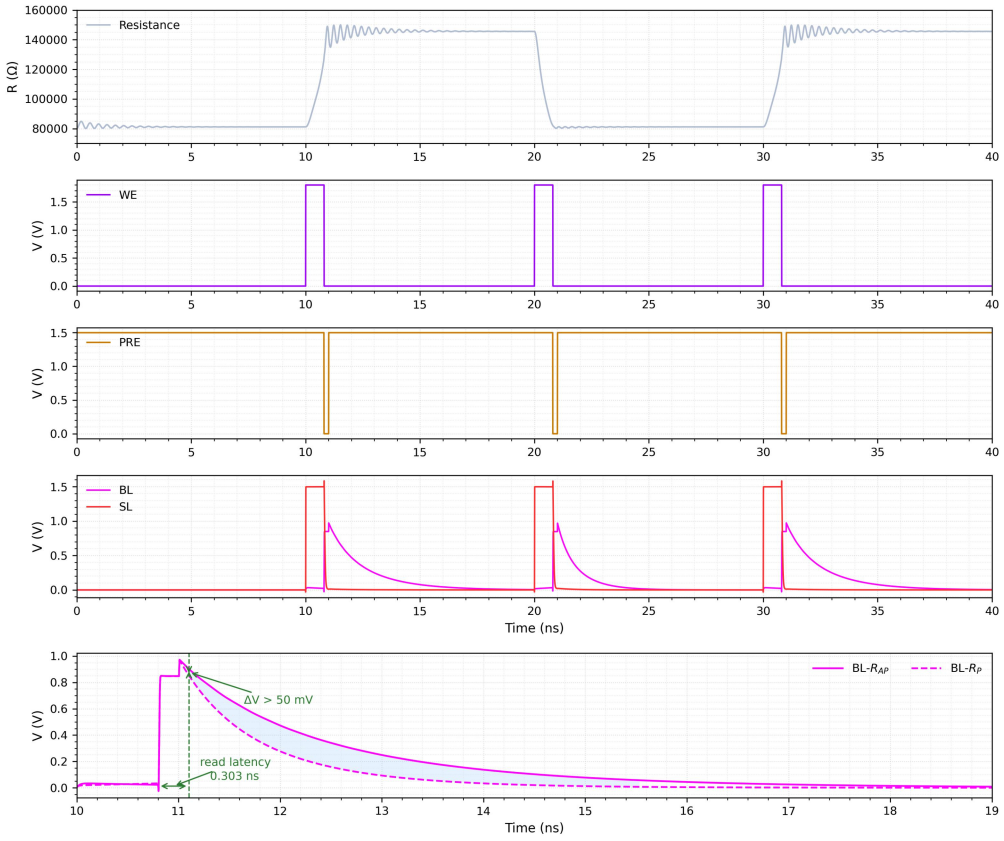
has indicated that total computation latency can be reduced to 0.2 ns per iteration and the power consumption of the state-of-the-art chips can reach approximately 1.0 mW.

Component	VC-MRAM chip		FPGA	
Function	Spin flipping		Spin coupling	Others*
Operation	MTJ write	MTJ read	MAC	
Delay	< 1ns	4 ns	20 ns	20 ns

Supplementary Table 3. Main delay decomposition of VSIM. *Other operations contain annealing schedule (temperature control) and δH-to-W_v conversion.

Component	Embedded VC-MRAM		ASIC of Spin coupling
Operation	MTJ write	MTJ read	MAC + Others*
Delay	< 0.5 ns	0.3 ns	0.2 ns

Supplementary Table 4. Main delay decomposition of the projected VSIM. *Other operations contain annealing schedule (temperature control) and δH-to-W_v conversion.



Supplementary Figure 18. Cadence circuit simulation of the read/write operation and optimized read latency evaluation.

Supplementary Note 11 Performance Comparison with State-of-the-Art Ising Machines

In Table 1 of the main text, we present a fair cross-platform comparison with prior works by carefully accounting for their reported performance and realistic device-level physical parameters for each representative Ising-machine approach, including both experimental data and projected estimates.

` At the Ising spin level, each platform is determined based on the following considerations:

- **GPU:** The Noisy Mean-Field Annealing (NMFA) algorithm[6] shares a deep algorithmic correspondence with quantum annealing (QA): both methods relax discrete combinatorial optimization into dynamical evolution over a continuous state space (quantum superposition amplitudes in QA; mean-field spin expectations in NMFA), governed by a time-decaying global control parameter (transverse field strength in QA; temperature in NMFA). This makes NMFA the most appropriate classical digital counterpart for cross-platform comparison with QPU and physics-based Ising machines. The speed and energy consumption per individual Ising spin update are not available (N/A) for GPU.
- **QPU**: One of the latest QPUs for QA is the D-wave Advantage^{TM} processor[16]. It contains 5627 qubits and its annealing time is estimated as 7 ns[17] when used for quantum simulation.
- **CIM:** A cavity round-trip time of 5 μs in the experiment[18] can be considered as the time consumption per spin update.
- **Mem-HNN:** The Ising machine demonstration is conducted using a memristor array experimentally[4], it is difficult to disentangle and attribute parameters to an individual Ising spin.
- **PTNO:** The 8-node proof-of-concept system requires 3.7 ms for 250 cycles[5]. Because the oscillator Ising machine can update spins in parallel, the time per spin update is estimated as 3.7 ms / 250 = 14.8 μs. Because the power consumption of each PTNO is 20 μW, the energy consumption per spin update is estimated as 14.8

μs * 20 μW = 296 pJ.

- **SMTJ:** The retention time of the SMTJ is evaluated as 100 μs[2] experimentally, which can be regarded as the spin update time. And the voltage and current to write the SMTJ are 0.15 V and 10 μA, respectively. Thus, the energy consumption per spin update is estimated as 100 μs * 0.15 V * 10 μA = 150 pJ. Recently, the computation time of the SMTJ device has been shortened to 2.7 ns[19], but its operation requires assistance from 70 mT external field. Using the projection in the reference[2], a relative ideal energy consumption estimation of the SMTJ is 1.35 pJ.
- **V-MTJ**: Voltage pulses with a duration of 10 ns are applied to V-MTJs to generate 50%/50% random bits[20]. Sixteen such random bits are combined to function as a single probabilistic bit (Ising spin), resulting in an estimated single-spin update time of 160 ns. Experimental measurements indicate that voltage pulses of 1.8 V are used to drive the V-MTJs during random bit generation. The average device resistance is measured to be 110 kΩ. The energy consumption per single spin update can be calculated as $(1.8\ V)^2$ / 110 kΩ × 160 ns = 4.7 pJ. In the projection with parallel operation and ASIC integration[20], the time and energy consumption can be optimized to 10 ns and 430 fJ, respectively.
- **VSIM:** By harnessing the intrinsic stochasticity of VCMA-MTJ, the VSIM achieves single Ising spin update in just 1 ns. And as demonstrated in Supplementary Note 9, the VCMA-MTJ-based Ising spin consumes 40 fJ per update.

At the system level, the TTS metric is evaluated on a standardized benchmark: 60-node dense Max-cut problems with 50% connectivity. Here, TTS is defined as the computation time for finding the optimal solution to a COP (in this case, Max-cut problems) with 99% probability[8], which can be formulated as $T_{com}log(1-0.99)/log(1-P_s)$, where T_{com} is the computation time with a success probability of P_s. Based on the TTS, we further derive the energy efficiency in terms of solutions per second per watt, which is defined as $1/(TTS * Power)$. Here, the power value is taken from those reported in the literature.

The detailed methodology and considerations are as follows:

■ **GPU:** As presented in the reference[6], the TTS of GPU (NVIDIA GeForce GTX 1080 Ti GPU) is measured as 10 μs. However, this result is derived by amortizing the total wall-clock time over a large batch of parallel samples. Specifically, the implementation executes a batch of N_s (10^4 in the literature) independent annealing runs concurrently on the GPU, and measures the total wall-clock time T_{batch} required to generate all samples. The reported computation time is then calculated as $T_{com} = T_{batch}/N_s$ and the appropriate TTS of GPU is 10 μs × 10^4 = 100 ms. Considering the massive parallelism of GPU, its effective power is estimated as 250 W / 10^4 = 25 mW.

■ **QPU**: There have been no demonstrations of 50%-connectivity Max-cut problems by the D-wave Advantage™ processor (although it has shown capability on G-set dataset[21], it relies on quantum-classical hybrid computing[22]). Therefore, we use the performance metrics of the D-wave 2000Q processor. It is reported that its TTS on a 55-node 50%-connectivity Max-cut problem is 10^4 s, and the TTS will be substantially larger when the problem size scales to 60 nodes.

■ **CIM:** The CIM's TTS for the 60-node benchmark is reported as 600 μs[6]. The actual system-level power consumption of CIM is not reported.

■ **Mem-HNN:** The Mem-HNN has shown capability of solving dense Max-cut problems experimentally. Its TTS is estimated through simulation with a 16-nm CMOS technology. In the simulation, it takes 2 ns for each computation cycle, and it adopts a hybrid update mechanism that updates 10 nodes simultaneously. Based on this setup, the Mem-HNN's TTS for the 60-node benchmark is simulated as 6.6 μs[4].

■ **PTNO:** The simulation results of PTNO-based Ising machine show that it solves the 60-node benchmark using 100 cycles with approximately 25% success probability[5]. Thus, its TTS is calculated as 23.7 ms.

■ **VSIM:** As shown in Supplementary Note 10, our experimentally demonstrated version of VSIM consumes 45 ns per iteration and the corresponding power is 40 mW. Based on these parameters, the TTS is scaled to 130.5 μs (1000 iterations, $P_s \approx 80\%$, $T_{com} = 45\ \mu s$) and the solutions per second per watt is calculated as 1.92×10^5. The projected VSIM's TTS on the 60-node dense Max-cut problems is estimated as 2.9 μs (assuming it takes 1 ns per iteration and 1000 iterations, $P_s \approx 80\%$, $T_{com} =$

$1\,\mu s$) . The projected power is 1 mW, so the solutions per second per watt is calculated as 3.45×10^{8}.

■ **SMTJ, V-MTJ**: Although SMTJ- and V-MTJ-based Ising machines[2,20] have no reported results on the dense Max-cut benchmark, they belong to the same class of spintronic probabilistic Ising machines. From a performance evaluation perspective, while the per-iteration time varies across device platforms, the iterations-to-solution and the corresponding success probability can be treated as platform-independent, reflecting the shared algorithmic behavior of probabilistic Ising machines operating under simulated annealing. Accordingly, the TTS of these platforms is estimated based on the proportional relationship between their per iteration computation time and that of VSIM, assuming that all latency components except the device write time are identical.

Supplementary Note 12 Operational Lifetime Analysis

To assess the practical operational lifetime of the VSIM chip, we provide a detailed analysis based on the actual system parameters and device endurance characteristics.

The experimentally measured write endurance of our VCMA-MTJ devices is 10^{13} cycles. We note that this value reflects the duration of our testing window rather than an observed failure point. Based on the absence of significant degradation during testing, along with theoretical considerations and SOT-MRAM endurance benchmarks, the intrinsic device endurance is anticipated to be considerably higher, likely exceeding 10^{14} cycles[23].

- System specifications:

Time for one iteration(includes time of the FPGA and VC-MRAM): 45 ns

MTJs used per real-world problem (Problem size): 1000

Total MTJs available on chip: 96k

Endurance per MTJ: 10^{14} cycles

- Lifetime calculation:

Operational time (continuous operation): $10^{14} * 45\ ns = 52\ days$

Total operational lifetime: $96000/1000 * 52\ days \approx 13.7\ years$

In practice, non-uniform access patterns may arise when specific MTJs are addressed more frequently than others due to problem-specific write distributions. In such scenarios, wear-leveling techniques can be applied at the system level to redistribute write operations across the available MTJ array, preventing localized endurance exhaustion. For example, Park et al.[24] demonstrated swap-based wear-leveling for non-volatile memory arrays and the peak write count can be reduced by up to 39.3% compared to unmanaged non-uniform access. This technique compresses the effective hot-spot write concentration factor to approximately a factor of 1.52, yielding a projected lifetime of 9.0 years.

References

1. Raimondo, E. *et al.* High-Performance and Reliable Probabilistic Ising Machine Based on Simulated Quantum Annealing. *Phys. Rev. X* **15**, 041001 (2025).
2. Si, J. *et al.* Energy-efficient superparamagnetic Ising machine and its application to traveling salesman problems. *Nat. Commun.* **15**, 3457 (2024).
3. Jun-Dong Cho, Raje, S. & Sarrafzadeh, M. Fast approximation algorithms on maxcut, k-coloring, and k-color ordering for VLSI applications. *IEEE Trans. Comput.* **47**, 1253–1266 (1998).
4. Cai, F. *et al.* Power-efficient combinatorial optimization using intrinsic noise in memristor Hopfield neural networks. *Nat. Electron.* **3**, 409–418 (2020).
5. Dutta, S. *et al.* An Ising Hamiltonian solver based on coupled stochastic phase-transition nano-oscillators. *Nat. Electron.* **4**, 502–512 (2021).
6. King, A. D., Bernoudy, W., King, J., Berkley, A. J. & Lanting, T. Emulating the coherent Ising machine with a mean-field algorithm. Preprint at https://doi.org/10.48550/arXiv.1806.08422 (2018).
7. Hamerly, R. *et al.* Experimental investigation of performance differences between coherent Ising machines and a quantum annealer. *Sci. Adv.* **5**, eaau0823 (2019).
8. Goto, H. *et al.* High-performance combinatorial optimization based on classical mechanics. *Sci. Adv.* **7**, eabe7953 (2021).
9. Yang, S., Bao, Y., Humianto, E., Prabhakar, A. & Yang, H. A parallel magnetic tunnel junction-based probabilistic Ising processor for efficient quadratic optimization. *Nat. Commun.* https://doi.org/10.1038/s41467-026-71128-1 (2026) doi:10.1038/s41467-026-71128-1.
10. Lee, A. *et al.* A Dual-Data Line Read Scheme for High-Speed Low-Energy Resistive Nonvolatile Memories. *IEEE Trans. Very Large Scale Integr. VLSI Syst.* **26**, 272–279 (2018).
11. Kaiser, M. A.-A. *et al.* Voltage controlled magnetic tunnel junction based ADC less processing-in-pixel for extreme edge intelligence. *Npj Unconv. Comput.* **2**, 14 (2025).
12. Wu, Y. C. *et al.* Deterministic and Field-Free Voltage-Controlled MRAM for High Performance and Low Power Applications. in *2020 IEEE Symposium on VLSI*

Technology 1–2 (IEEE, Honolulu, HI, USA, 2020). doi:10.1109/VLSITechnology18217.2020.9265057.

13. Duan, X. *et al.* Ultrafast Dual-Mode Ising Machine with Real Time Sensing-Storage-Computation Paradigm. in *2025 IEEE International Electron Devices Meeting (IEDM)* 1–4 (IEEE, San Francisco, CA, USA, 2025). doi:10.1109/IEDM50572.2025.11353718.
14. Wu, Y., Bae, J., Shin, S. & Kim, B. 10.8 COBI: A Degree-of-56 Column-Bipartite Densely Connected Digital Ising Chip with 8b Spin Coefficients. in *2026 IEEE International Solid-State Circuits Conference (ISSCC)* 182–184 (IEEE, San Francisco, CA, USA, 2026). doi:10.1109/ISSCC49663.2026.11409141.
15. Su, Y., Kim, T. T.-H. & Kim, B. FlexSpin: A Scalable CMOS Ising Machine with 256 Flexible Spin Processing Elements for Solving Complex Combinatorial Optimization Problems. in *2022 IEEE International Solid- State Circuits Conference (ISSCC)* 1–3 (IEEE, San Francisco, CA, USA, 2022). doi:10.1109/ISSCC42614.2022.9731680.
16. McGeoch, C. & Farré, P. Advantage Processor Overview. (2022).
17. King, A. D. *et al.* Beyond-classical computation in quantum simulation. *Science* **388**, 199–204 (2025).
18. Inagaki, T. *et al.* A coherent Ising machine for 2000-node optimization problems. *Science* **354**, 603–606 (2016).
19. Soumah, L. *et al.* Entropy-assisted nanosecond stochastic operation in perpendicular superparamagnetic tunnel junctions. *Phys. Rev. Appl.* **24**, L011002 (2025).
20. Duffee, C. *et al.* An integrated-circuit-based probabilistic computer that uses voltage-controlled magnetic tunnel junctions as its entropy source. *Nat. Electron.* **8**, 784–793 (2025).
21. Ye, Y. Gset collection of graphs for Max-Cut benchmarking. https://web.stanford.edu/~yyye/yyye/Gset/.
22. Vodeb, J., Eržen, V., Hrga, T. & Povh, J. Accuracy and Performance Evaluation of Quantum, Classical and Hybrid Solvers for the Max-Cut Problem. Preprint at https://doi.org/10.48550/arXiv.2412.07460 (2024).
23. Nguyen, V. D., Rao, S., Wostyn, K. & Couet, S. Recent progress in spin-orbit torque

magnetic random-access memory. *Npj Spintron.* **2**, 48 (2024).

24. Park, Y. M., Yeom, J., Kim, D. & Chung, E.-Y. Unified Wear-Leveling Technique for NVM-Based Buffer of SSD. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **42**, 5161–5165 (2023).