Stochastic *p*-Bits for Invertible Logic

Kerem Yunus Camsari,[*] Rafatul Faria, Brian M. Sutton, and Supriyo Datta[†]

School of Electrical and Computer Engineering, Purdue University, West Lafayette, Indiana 47907, USA

(Received 16 November 2016; revised manuscript received 27 April 2017; published 20 July 2017)

Conventional semiconductor-based logic and nanomagnet-based memory devices are built out of stable, deterministic units such as standard metal-oxide semiconductor transistors, or nanomagnets with energy barriers in excess of ≈40–60 kT. In this paper, we show that unstable, stochastic units, which we call "*p*-bits," can be interconnected to create robust correlations that implement precise Boolean functions with impressive accuracy, comparable to standard digital circuits. At the same time, they are *invertible*, a unique property that is absent in standard digital circuits. When operated in the direct mode, the input is clamped, and the network provides the correct output. In the inverted mode, the output is clamped, and the network fluctuates among all possible inputs that are consistent with that output. First, we present a detailed implementation of an invertible gate to bring out the key role of a single three-terminal transistorlike building block to enable the construction of correlated *p*-bit networks. The results for this specific, CMOS-assisted nanomagnet-based hardware implementation agree well with those from a universal model for *p*-bits, showing that *p*-bits need not be magnet based: any three-terminal tunable random bit generator should be suitable. We present a general algorithm for designing a Boltzmann machine (BM) with a symmetric connection matrix $[J]$ ($J_{ij} = J_{ji}$) that implements a given truth table with *p*-bits. The $[J]$ matrices are relatively sparse with a few unique weights for convenient hardware implementation. We then show how BM full adders can be interconnected in a partially directed manner ($J_{ij} \neq J_{ji}$) to implement large logic operations such as 32-bit binary addition. Hundreds of stochastic *p*-bits get precisely correlated such that the correct answer out of 2^{33} ($\approx 8 \times 10^9$) possibilities can be extracted by looking at the statistical mode or majority vote of a number of time samples. With perfect directivity ($J_{ji} = 0$) a small number of samples is enough, while for less directed connections more samples are needed, but even in the former case logical invertibility is largely preserved. This combination of digital accuracy and logical invertibility is enabled by the hybrid design that uses bidirectional BM units to construct circuits with partially directed interunit connections. We establish this key result with extensive examples including a 4-bit multiplier which in inverted mode functions as a factorizer.

DOI: 10.1103/PhysRevX.7.031014 Subject Areas: Electronics, Magnetism, Spintronics

I. INTRODUCTION

Conventional semiconductor-based logic and nanomagnet-based memory devices are built out of stable, deterministic units such as standard metal-oxide semiconductor (MOS) transistors, or nanomagnets with energy barriers in excess of ≈40–60 kT. The objective of this paper is to introduce the concept of what we call "*p*-bits" representing unstable, stochastic units which can be interconnected to create robust correlations that implement precise Boolean functions with impressive accuracy comparable to standard digital circuits. At the same time, this "probabilistic spin logic" (PSL) is *invertible*, a unique property that is absent in standard digital circuits. When operated in the direct mode, the input is clamped, and the network provides the correct output. In the inverted mode, the output is clamped, and the network fluctuates among all possible inputs that are consistent with that output.

Any random signal generator whose randomness can be tuned with a third terminal should be a suitable building block for PSL. The icon in Fig. 1(b) represents our generic building block whose input I_i controls the output m_i according to the equation [Fig. 1(a)]

$$m_i(t) = \mathrm{sgn}\{\mathrm{rand}(-1,1) + \tanh[I_i(t)]\}, \qquad (1)$$

where $\mathrm{rand}(-1,+1)$ represents a random number uniformly distributed between -1 and $+1$. It is assumed to change every τ seconds, which represents the retention time of individual *p*-bits. We normalize the time axis to τ so that t is dimensionless and progresses in steps $(0,1,2,\ldots)$. At each time step, if the input is zero, the output takes on a value of -1 or $+1$ with equal probability, as shown in the middle panel of Fig. 1(d). A negative input I_i makes

[*] kcamsari@purdue.edu
[†] datta@purdue.edu

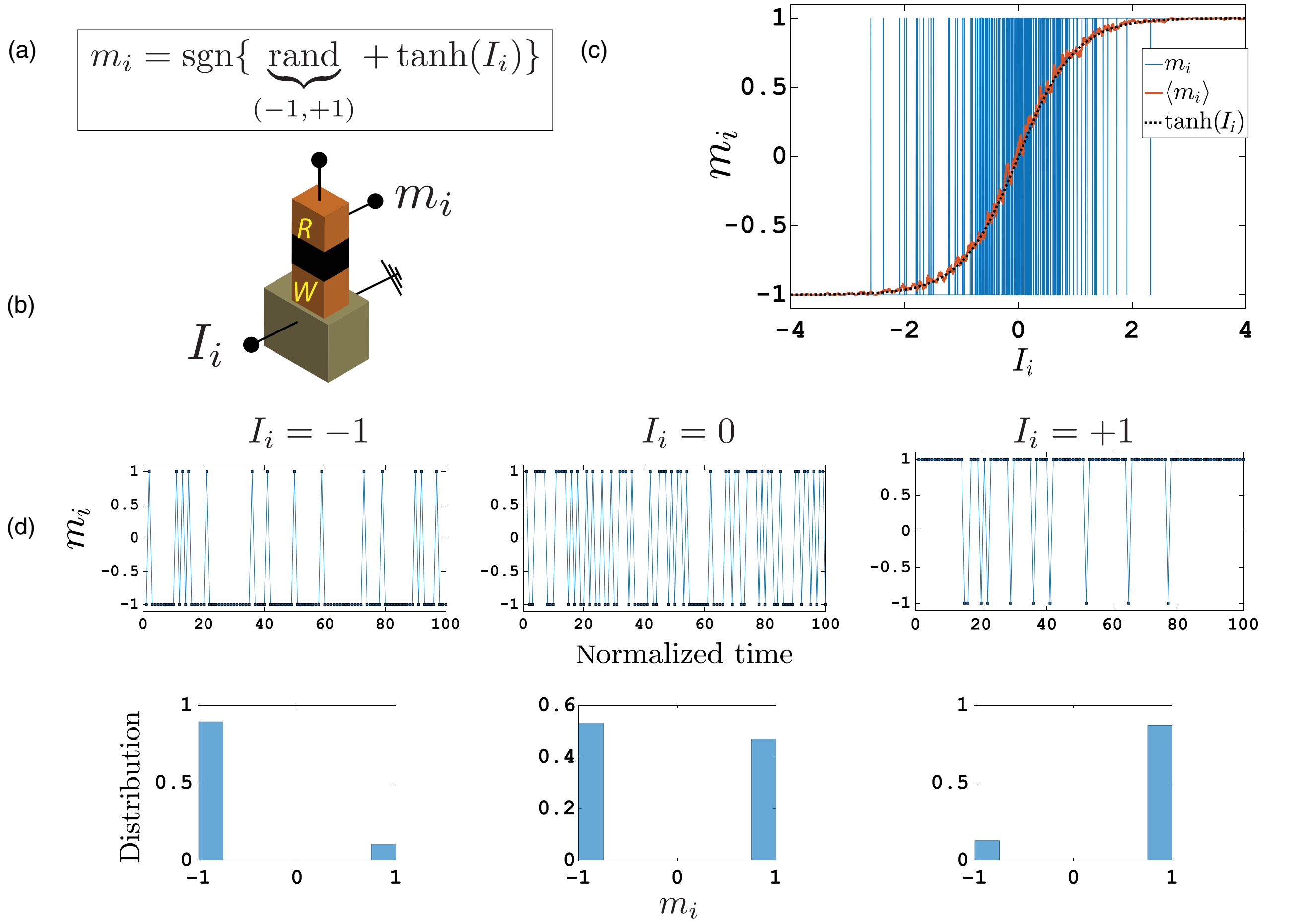


FIG. 1. Generic building block for PSL. (a) Generic model for PSL described by Eq. (1) with distinct READ and WRITE units represented by the R and W icon shown in (b). Useful functionalities are obtained by interconnecting R and W units according to Eq. (2), $I_i = I_0(h_i + \sum J_{ij} m_j)$, with appropriately designed $\{h\}$ and $[J]$. (c) The blue trace shows the "magnetization" (m_i) obtained from Eq. (1) as the current (I_i) is ramped. The red trace shows the sigmoid response obtained from a RC circuit which provides a moving average of the time-dependent "magnetization" that agrees very well with the black curve showing $\tanh(I_i)$. The bias terminal could involve a voltage (V) instead of a current (I), just as the output could involve quantities other than magnetization. (d) The idealized telegraphic behavior of the model is shown at various bias points along with corresponding distributions.

negative values more likely (left-hand panel) while a positive input makes positive values more likely (right-hand panel). Figure 1(c) shows $m_i(t)$ as the input is ramped from negative to positive values. Also shown is the time-averaged value of m_i, which equals $\tanh(I_i)$.

A possible physical implementation of p-bits could use stochastic nanomagnets with low-energy barriers Δ whose retention time [1],

$$\tau = \tau_0 \exp(\Delta/kT),$$

is very small, on the order of τ_0, which is a material-dependent quantity called the attempt time and is experimentally found to be $\approx$10 ps − 1 ns [1] among different magnetic materials. Such stochastic nanomagnets can be pinned to a given direction with spin currents that are at least an order of magnitude less than those needed to switch 40-kT magnets. The sigmoidal tuning curve in Fig. 1(c) describing the time average of a fluctuating signal represents the essence of a p-bit. Purely CMOS implementations of a p-bit are possible [2,3], but the sigmoid seems like a natural feature of nanomagnets driven by spin currents. Indeed, the use of stochastic nanomagnets in the context of random number generators, stochastic oscillators, and autonomous learning [4–6] has been discussed in the literature. But performing "invertible" Boolean logic utilizing large-scale correlations has not been discussed before to our knowledge.

Note that we are using the term *invertibility* in the broader sense of relation inverses and not in the narrower sense of function inverses. For example, AND, when interpreted as a relation, consists of the set $\{\{1,1 \rightarrow 1\}, \{0,0 \rightarrow 0\}, \{1,0 \rightarrow 0\}, \{0,1 \rightarrow 0\}\}$, where each term is of the form $\{A, B \rightarrow \text{AND}(A,B)\}$. The relation inverse of 0 is the set $\{\{0,0\},\{0,1\},\{1,0\}\}$ even though the corresponding functional inverse is not defined. What our scheme provides, probabilistically, is the relation inverse [7,8].

Ensemble average versus time average.—A sigmoidal response was presented in Ref. [9] for the ensemble-averaged magnetization of large barrier magnets biased along a neutral state. This was proposed as a building block for both Ising computers as well as directed belief networks and a recent paper [10] describes a similar approach applied to a graph coloring problem. By contrast, low-barrier nanomagnets provide a sigmoidal response for the time-averaged magnetization, and a suitably engineered network of such nanomagnets could cycle through the 2^N collective states at GHz rates, with an emphasis on the "low-energy states" which can encode the solution to the combinatorial optimization problems, like the traveling salesman problem, as shown in Ref. [11]. Once the time-varying magnetization has been converted into a time-varying voltage through a READ circuit, a simple RC circuit can be used to extract the answer through a moving time average. For example, in Fig. 1(c) the red trace iss obtained from the rapidly varying blue trace using a RC circuit in a SPICE simulation.

The central feature underlying both implementations is the p-bit that acts like a tunable random number generator, providing an intrinsic sigmoidal response for the ensemble-averaged or the time-averaged magnetization as a function of the spin current. It is this response that allows us to correlate the fluctuations of different p-bits in a useful manner by interconnecting them according to

$$I_i(t) = I_0\left(h_i(t) + \sum_j J_{ij} m_j(t)\right), \tag{2}$$

where h_i provides a local bias to magnet i and J_{ij} defines the effect of bit j to bit i, and I_0 sets a global scale for the strength of the interactions like an inverse "pseudotemperature" giving a dimensionless current I_i to each p-bit. The computation of $I_i(t)$ in terms of $m_j(t)$ in Eq. (2) is assumed instantaneous; in hardware implementations there can be interconnect delays that relate $m_j(t)$ to currents at a later time $I_i(t')$.

Equation (1) arises naturally from the physics of low-barrier nanomagnets, as we discuss above. Equation (2) represents the "weight logic" for which there are many candidates such as memristors [12], floating-gate-based devices [13], domain-wall-based devices [14], and standard CMOS [15]. The suitability of these options will depend on the range of J values and the sparsity of the J matrix.

Equations (1)–(2) are essentially the same as the defining equations for Boltzmann machines introduced by Hinton and his collaborators [16], which have had enormous impact in the field of machine learning, but they are usually implemented in software that is run on standard CMOS hardware. The primary contributions of this paper are threefold.

(i) *Hardware implementation.*—It may seem obvious that an unstable magnet could provide a natural hardware for representing a p-bit, but we stress a less obvious point. To the best of our knowledge, simple two-terminal devices are not suitable for constructing large-scale correlated networks of the type envisioned here. Instead, we need three-terminal building blocks with transistorlike gain and input-output isolation, as shown in Fig. 1(b) [9]. To stress this point, we describe a concrete implementation of a Boolean function using detailed nanomagnet and transport simulations that are in good agreement with those obtained by the generic model based on Eq. (1). All other results in this paper are based on Eq. (1) in order to emphasize the generality of the concept of p-bits, which need not necessarily be nanomagnet based [17,18].

(ii) *Boltzmann machines (BM) for invertible Boolean logic [Fig. 2(a)].*—Much of the current emphasis on BMs is on "learning" giving rise to the concept of restricted Boltzmann machines [19]. By contrast, this paper is about Boolean logic, extending an established method for Hopfield networks [20] to provide a mathematical prescription to turn *any* Boolean truth table into a symmetric J matrix [Eq. (2), with $J_{ij} = J_{ji}$], in one shot with no learning being involved. This design principle seems quite robust, functioning satisfactorily even when the J-matrix elements are rounded off, so that the required interconnections are relatively sparse and quantized, which simplifies the hardware implementation. The numerical probabilities agree well with those predicted from the energy functional.

$$E(\{m\}) = -I_0\left(\sum_{i,j} \frac{1}{2}(J_{ij} m_i m_j) + \sum_i h_i m_i\right) \tag{3}$$

using the Boltzmann law:

$$P(\{m\}) = \frac{\exp(-E)}{\sum_{i,j} \exp(-E)}. \tag{4}$$

Most importantly, we show that the resulting Boolean gates are invertible: not only do they provide the correct output for a given input, for a given output they provide the correct input(s). If the given output is consistent with multiple inputs, the system fluctuates among all possible answers. This remarkable property of invertibility is absent in standard digital circuits and could help provide solutions to the Boolean satisfiability problem (Fig. 8) [21].

(iii) *Directed networks of BM [Fig. 2(b)].*—Finally, we show that individual BMs can be connected to perform *precise* arithmetic operations, which are the norm in standard digital logic, but quite surprising for BM, which are more like a collection of interacting particles than like a digital circuit. We show that a 32-bit adder converges to the one correct sum out of $2^{33} \approx 8 \times 10^9$ possibilities when the interaction parameter is suddenly turned up from, say, $I_0 = 0.25$ to $I_0 = 5$. This can be likened to quenching a molten liquid and getting a perfect crystal. What we expect is plenty of defects, distributed differently every time we do the experiment. That is exactly what we get if the individual BM full adders (FA) comprising the 32-bit adder are connected bidirectionally ($J_{ij} = J_{ji}$). But by making the connection between adders directed ($J_{ij} \neq J_{ji}$), we obtain the striking accuracy of digital circuits while largely retaining the invertibility of BM. This is a key result that we establish with extensive examples including a 4-multiplier which in inverted mode functions as a factorizer.

Each of these three contributions is described in detail in the three sections that follow.

II. EXAMPLE HARDWARE IMPLEMENTATION OF PSL

To ensure that individual p-bits can be interconnected to produce robust correlations, it is important to have separate terminals for writing (more correctly biasing) and reading, marked W and R, respectively in Fig. 3(a). With in-plane magnetic anisotropy (IMA) nanomagnets (e.g., circular nanomagnets) this could be accomplished following existing experiments [22,23] using the giant spin Hall effect (GSHE). Recent experiments using a built-in exchange bias [24–27] could make this approach applicable to perpendicular magnetic anisotropy (PMA) as well. Note, however, that these experiments have all been performed with stable free layers, and would have to be carried out with low-barrier magnets in order to establish their suitability for the implementation of p-bits. As the field progresses, one can expect the bias terminal to involve voltage control [28,29] instead of current control, just as the output could involve quantities other than magnetization. We now show a concrete implementation of a Boolean function using minimal CMOS circuitry in conjunction with stochastic nanomagnets through detailed nanomagnet and transport simulations that are in good agreement with those obtained from the generic model based on Eq. (1).

Figure 3(a) shows a possible, CMOS-assisted p-bit that has a separate READ and WRITE path. The device consists of a heavy metal exhibiting GSHE that drives a circular magnet which replaces the usual elliptical magnets in order to provide the stochasticity needed for the magnetization. A small read current, which is assumed to not disturb the magnetization of the free layer in our design, that flows through the fixed layer is used to sense the instantaneous magnetization, which is amplified and isolated by two inverters that act as a buffer. This structure is very similar to the experimentally demonstrated GSHE switching of elliptical magnets that were similarly read-out by a magnetic tunnel junction (MTJ) [22], with the only exception that the elliptical magnets are replaced by circular magnets with an aspect ratio of one. This device could be viewed as replacing the free layers of the GSHE-driven MTJs demonstrated in Ref. [22] with those in the telegraphic regime [23,30–32] .

In the presence of thermal noise the magnetization of such a circular magnet rotates in the plane of the circle without a preferred easy axis that would have arisen due to the shape anisotropy, effectively making its thermal stability $\Delta \approx 0$ kT [33]. This magnetization can be pinned by a spin current that is generated by flowing a charge current through the GSHE layer. The magnetic-field-driven sigmoidal responses of magnetization for such circular magnets have experimentally been demonstrated [34,35], while the spin-current-driven pinning has not been demonstrated to our knowledge. Using validated modules for transport and magnetization dynamics [36] [Fig. 3(b)], we solve the stochastic Landau-Lifshitz-Gilbert (sLLG) equation in the presence of thermal noise and a GSHE current. The following section shows detailed simulation parameters.

Sigmoidal response.—A long-time average ($t = 500$ ns) of the magnetization $\langle m_z \rangle$ as a function of a GSHE-generated spin current is plotted in Fig. 3(e) that displays the desired sigmoidal characteristic for p-bits dictated by Eq. (1). The x axis of Fig. 3(e) is normalized to the geometric gain factor that relates the charge current to the spin current exerted [37,38]:

$$\beta \equiv \frac{I_s}{I_c} = \theta_{\mathrm{SH}} \frac{L_{\mathrm{FM}}}{t} \left[1 - \mathrm{sech}\left(\frac{t}{\lambda}\right)\right], \tag{5}$$

where θ_{SH} is the Hall angle, t is the thickness, and λ is the spin-relaxation length of the heavy metal. The quantity β can be made to be much greater than 1 providing an intrinsic gain [39]; however, for the parameters used in the present examples, β is ≈ 1.5.

Another quantity that is used to normalize the x axis of Fig. 3(e) is the "thermal spin current" that corresponds to the strength of the thermal noise that needs to be overcome for a circular magnet to be pinned in a given direction:

$$I_s^{\mathrm{th}} = \left(\frac{4q}{\hbar}\right) \alpha(kT), \tag{6}$$

where q is electron charge, α is the damping coefficient of the magnet. I_s^{th}, I_s, and I_c all have units of charge current; therefore, we can define the dimensionless interaction parameter I_0 of Eq. (2) as $I_0 \equiv \beta I_c / I_s^{\mathrm{th}} = I_s / I_s^{\mathrm{th}}$.

It can be seen from Fig. 3(e) that when the applied spin current $\beta I_c / I_s^{\mathrm{th}} = I_s / I_s^{\mathrm{th}} \approx 10$, the magnetization of the circular magnet is pinned in the $\pm z$ directions for these particular parameters. For PMA magnets with low barriers ($\Delta \ll kT$), the pinning current is independent of the volume as long as increasing the volume does not invalidate the $\Delta \ll kT$ assumption. This can be analytically shown from a 1D Fokker-Planck equation [40], and we reproduce this behavior directly from sLLG simulations. For the in-plane (circular) magnets we consider here, the pinning current in general has a M_s and volume dependence and the dimensionless pinning current can be larger.

Nevertheless, it is possible to estimate the thermal spin current for typical damping coefficients of $\alpha = 0.01$–0.1, $I_s^{\rm th}$ is ≈ 0.25 to -2.5 μA. Pinning currents for superparamagnets are at least an order of magnitude smaller than the critical switching currents of stable magnets [41]. $I_s^{\rm th}$, defined by Eq. (6), also sets the scale for I_0 defined in Eq. (2), suggesting that a stochastic nanomagnet-based implementation of PSL could be more energy efficient than the standard spin-torque switching of stable magnets that suffer from high current densities.

Need for three-terminal devices with READ-WRITE separation.—Note that a crucial function of the READ circuit and the CMOS transistors in this design is the ability to turn the magnetization into an output voltage that is proportional to m_z, providing gain for fan out and isolation to avoid any read disturb. Indeed, a critical requirement for any other alternative implementations of p-bits is the need for three terminal devices with separate READ and WRITE paths to provide gain and isolation. In this particular design these features come in by directly integrating CMOS transistors, but CMOS-free, all-magnetic designs with these characteristics have been proposed [39,42]. Our purpose is to simply show how a p-bit can be realized by using experimentally demonstrated technology. Alternative designs are beyond the scope of this paper.

READ circuit.—For the output to provide symmetric voltage swings on the GSHE layer, the minus supply V^- needs to be set to $V_{DD}/2$ since $V_{\rm OUT}$ ranges between 0 and V_{DD}. V^+ is set to $V_{DD}/2 + V_R$, where V_R is a small READ voltage that is amplified by the inverters. We assume a simple, bias-independent MTJ model [43]:

$$G_{\rm MTJ} = G_0(1 + P^2 m_z), \tag{7}$$

where P is the interface polarization and G_0 is the average MTJ conductance. Setting the reference resistance [Fig. 3] R_0 equal to G_0^{-1}, the input voltage to the inverters, V_M in Fig. 2(d) becomes

$$V_M = \frac{V_{DD}}{2} + \frac{V_R}{2 + m_z P^2}. \tag{8}$$

In the absence of a bias, $\langle m_z \rangle$ becomes 0 and the middle voltage fluctuates around the mean $\langle V_M \rangle = V_{DD}/2 + V_R/2$. This requires the inverter characteristic to be shifted to this value to produce a telegraphic output that fluctuates between 0 and V_{DD} with equal probability [Fig. 3(f)]. This shift is easily engineered by sizing the p-channel FET and n-channel FET transistors differently: a wider p-channel FET shifts the inverter characteristic towards V_{DD}, as we show in the next section.

Interconnection matrix.—A passive resistor network can be used as a possible interconnection scheme to correlate the p-bits, as shown in Fig. 4. A proper design of the interconnection matrix J that has only a few discrete values ensures a minimal number of different conductances (G_{ij}). In this demonstrated example the AND gate requires only two unique, discrete conductance values.

The spin currents that need to be delivered to each p-bit are on the order of a few μA and can be generated with charge currents that are even smaller, due to the GSHE gain. This means the interconnection resistances R_{ij} could be on the order of 100 kΩ since the voltage drops across these resistances are around $V_{\rm OUT} - V^- \approx \pm 0.5$ V. Since

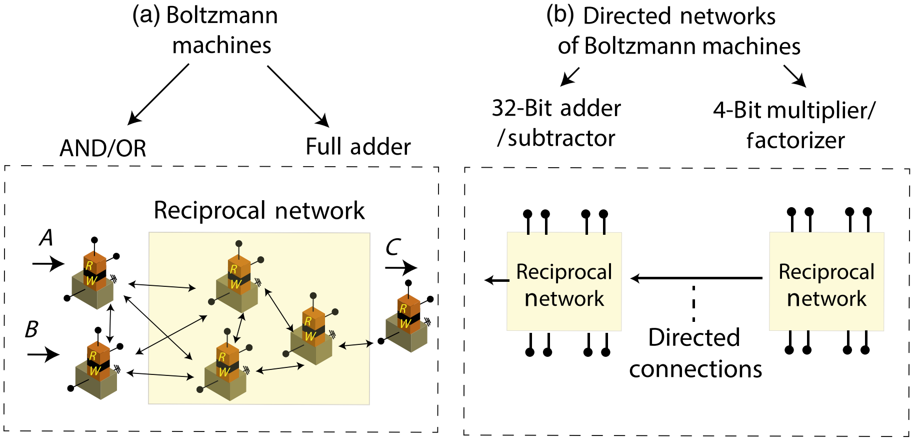


FIG. 2. PSL designs discussed in this paper. (a) Basic Boolean elements (AND and OR, full adder) are implemented as Boltzmann machines based on symmetrically coupled networks with $J_{ij} = J_{ji}$. (b) Complex Boolean functions like a 32-bit ripple carry adder or subtractor and 4-bit multiplier or factorizer are implemented by combining the reciprocal Boltzmann machines in a directed fashion.

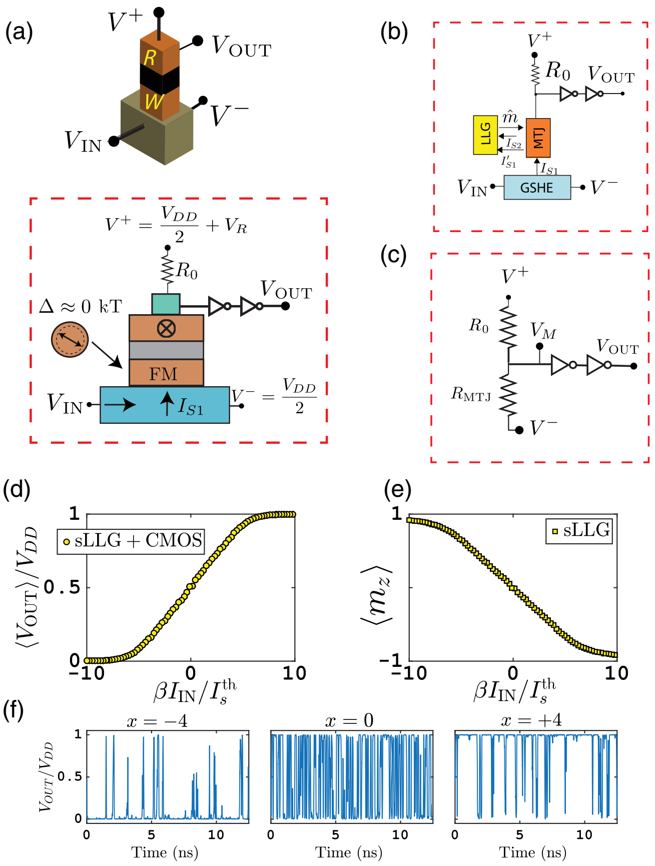


FIG. 3. CMOS-assisted implementation of p-bits. (a) A possible CMOS-assisted implementation of p-bits that have separate READ-WRITE paths. A GSHE layer provides a spin current is able to pin the magnetization of circular ferromagnets (FM) ($\Delta \approx 0$ kT). The change in magnetization is sensed by a MTJ and amplified by two CMOS inverters that act as a buffer, providing the necessary isolation and gain. (b) Self-consistent, modular modeling of transport and magnetization dynamics. See "Assumptions of the model" in the text. (c) Equivalent READ circuit. (d) SPICE-based average output voltage normalized to the $V_{DD} = 0.8$ V of 14-nm Fin Field-Effect Transistor (FinFET) high-performance (HP) inverters [44]. (e) sLLG-based average magnetization of the circular magnet as a function of the spin current (averaged over 500 ns for each bias point with a time step of $\Delta t = 0.05$ ps, 10×10^6 points per marker), normalized to the GSHE gain and the thermal noise strength $I_s^{\rm th}$. (f) The time-dependent output voltage at various bias points.

the GSHE ground $V^- = V_{DD}/2$ simply shifts all the voltages to get symmetric ± swings, we define the voltages $(V'_{\rm OUT})_i = (V_{\rm OUT})_i - V^-$. Then input currents to each p-bit can be expressed [Fig. 4(a)]:

$$(I_{\rm IN})_i = \sum_j G_{ij}(V'_{\rm OUT}) + G_i(V'_{\rm BIAS}) \qquad (9)$$

assuming $\sum_j G_{ij} \ll G_{\rm GSHE}$, since the heavy metal resistances are typically much less than hundreds of kΩ. We verify the validity of Eq. (9) by SPICE simulations, for the parameters chosen for these examples.

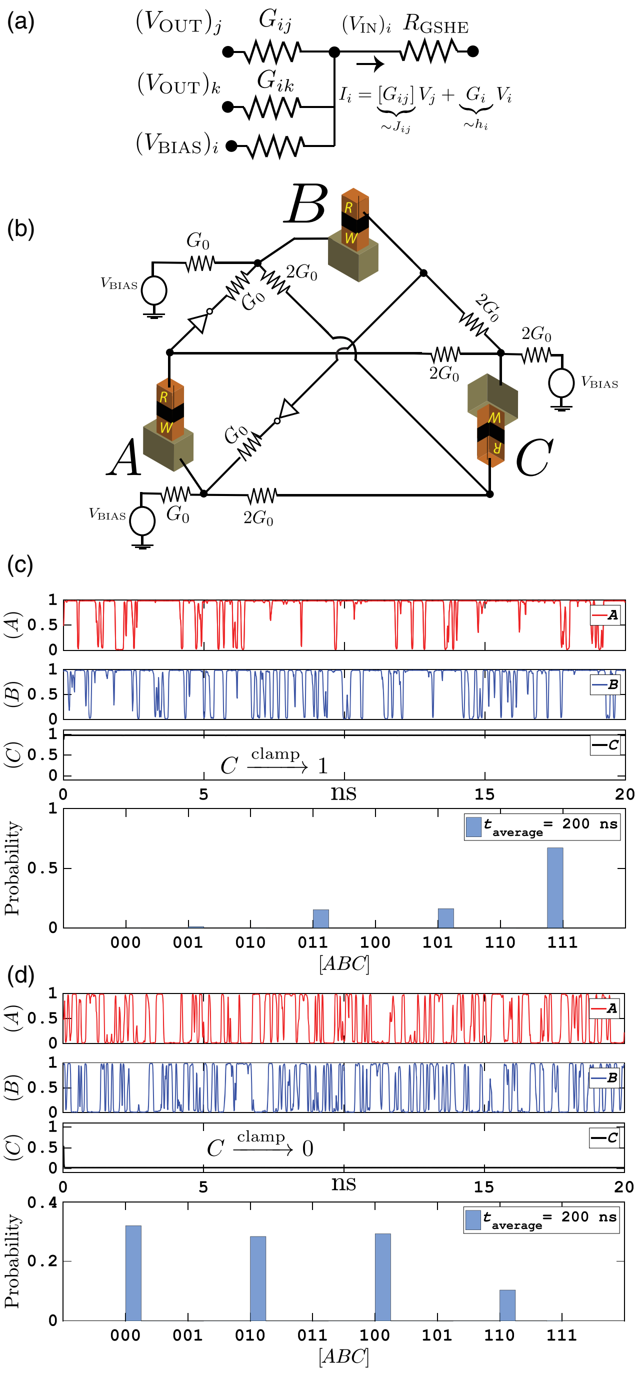


FIG. 4. Invertible AND gate. (a) Passive resistor network that is used to obtain the connection terms J_{ij} to correlate p-bits. The output impedance $R_{ij} = 1/G_{ij}$ is much smaller than the input impedance $R_{\rm GSHE}$, allowing separate voltages to add at the input of the ith p-bit. (b) Explicit implementation of an AND gate based on Eq. (10). (c) When C is clamped to 1, A and B spend most of their time in the (11) state, the only combination consistent with $C = 1$. (d) The invertible operation of the AND gate when the C gate is clamped to a zero, while A and B are left floating. A and B bits fluctuate between three possible combinations consistent with $C = 0$, $(A, B) = (00), (01), (10)$. The time response of A, B, C voltages are normalized by V_{DD}. Histogram is obtained by averaging over 200 ns of thresholded voltages, only the first 20 ns of A, B, C voltages are shown for clarity.

As a result, we observe that Eq. (9) constitutes a hardware mapping for the interconnections of Eq. (2). In this scheme G_{ij} conductances are initially adjusted to obtain a global interaction strength I_0 for a given problem. Alternatively, the interaction strength can be adjusted electrically by varying the supply voltages.

Invertible AND gate.—Figure 4(b) shows an explicit implementation of an invertible AND gate ($A \cap B = C$) corresponding to $[J]$ and $\{h\}$ matrices [45] that have three unique, integer entries:

$$J = \begin{array}{cc} & \begin{array}{ccc} A & \;B & \;C \end{array} \\ \begin{array}{c} A \\ B \\ C \end{array} & \begin{bmatrix} 0 & -1 & +2 \\ -1 & 0 & +2 \\ +2 & +2 & 0 \end{bmatrix} \end{array}, \qquad h^T = [+1 \quad +1 \quad -2]. \tag{10}$$

In Fig. 4(d), we show the *inverse* operation of the AND gate where we clamp the output bit C to a 0 or 1 by the bias voltage attached to its input terminal. The interconnection resistance is chosen to be $R_0 = 125\ \mathrm{k}\Omega$ that roughly provides $\approx \pm 6\ \mu\mathrm{A}$ of charge current to each p-bit, corresponding to an $I_0 \approx 3.5$ for the chosen parameters.

Generating the histogram.—At the end of the simulation ($t = 200$ ns), we threshold the voltage output of A, B, and C by legislating all voltages above $V_{DD}/2 = 0.4$ V to be 1, and below $V_{DD}/2$ to be 0. Then a histogram output for the thresholded word [ABC] is obtained and normalized to unit probability. Clamping the output to 0 and letting A and B float, make A and B fluctuate in a correlated manner and they visit the three possible states (00, 01, 10) with approximately equal probability. Resolving the output 0 to the three possible input combinations is, in a way, "factorizing" the output. Conversely, clamping the output to 1 produces a strong (11) peak in the histogram of [ABC], which is the only consistent input combination for $C = 1$ [Figs. 4(d)].

Assumptions of the model.—We make several simplifying assumptions while modeling the hardware implementation of a p-bit. (1) The READ voltage that is amplified by the inverters produces a small current that passes through the circular magnet and might potentially disturb its current state. We assume that this current [labeled as I_{S2} in Fig. 3(b)] is negligible and does not affect the magnetization of the stochastic magnet. (2) We assume that the spin current generated by the heavy metal is deposited to the free layer with perfect efficiency [$I'_{S1} = I_{S1}$ in Fig. 3(b)]; however, depending on the interface properties this conversion factor can be less than 100%. (3) We also assume that the fixed layer does not produce a notable stray field on the circular magnet. Note that the presence of such a constant field would simply shift the sigmoidal behavior presented in Figs. 3(e)to the right (or left) and could have been offset by a constant bias current. (4) Finally, we neglect the resistance of the GSHE portion in the READ circuit [Fig. 3(c)], assuming the MTJ resistance would be dominant in this path.

A. Detailed simulation parameters

This section shows the details of simulation parameters for the hardware implementation of p-bits that we use for Figs. 3 and 4.

sLLG for stochastic circular magnets.—The magnetization of a circular nanomagnet described as $\hat{m}_i$ is obtained from the stochastic Landau-Lifshitz-Gilbert equation:

$$\begin{aligned}(1+\alpha^2)\frac{d\hat{m}_i}{dt} &= -|\gamma|\hat{m}_i \times \vec{H}_i - \alpha|\gamma|(\hat{m}_i \times \hat{m}_i \times \vec{H}_i) \\ &\quad + \frac{1}{qN_i}(\hat{m}_i \times \vec{I}_{Si} \times \hat{m}_i) + \left(\frac{\alpha}{qN_i}(\hat{m}_i \times \vec{I}_{Si})\right),\end{aligned} \tag{11}$$

where α is the damping coefficient, q is the electron charge, γ is the electron gyromagnetic ratio, I_s is the spin current that is assumed to be uniformly distributed over the total number of spins in the macrospin, $N_i = M_s\mathrm{Vol.}/\mu_B$, μ_B being the Bohr magneton. We assume that the spin current generated from the GSHE layer is polarized in the z direction, such that $\vec{I}_{Si} = I_S\hat{z}$. $\vec{H}_i$ is the effective field of the circular magnet, where the uniaxial anisotropy is assumed to be negligible, but there is still a strong demagnetizing field. The thermal fluctuations also enter through the effective magnetic field: $\vec{H}_i = -4\pi M_s m_x \hat{x} + \vec{H}_{\mathrm{th}}$, x axis being the out-of-plane direction of the magnet, and $\langle|\vec{H}_{\mathrm{th}}|^2\rangle = 2\alpha kT/(|\gamma|M_s\mathrm{Vol.})$ in units [[Oe^2/Hz]] with zero mean, and equal in all three

TABLE I. Parameters used for simulations in Figs. 3 and 4.

Parameters	Value
Saturation magnetization (M_s)	300 emu/cm^3
Magnet diameter (Φ), thickness (t)	15 nm, 0.5 nm
MTJ polarization (P) [Eq. (7)]	0.5
MTJ conductance (G_0) [Eq. (7)]	176 μS
Damping coefficient (α)	0.1
Spin Hall length, width [Eq. (5)]	$L = W = 15$ nm
Hall angle, spin relax. length	$\theta = 0.5$ [46], $\lambda_{sf} = 2.1$ nm [47]
Spin Hall res. (ρ), thickness (t)	200 $\mu\Omega$ cm [48], 3.15 nm
Temperature (T)	300 K
CMOS models	14-nm HP-FinFET [44]
Supply and READ voltage	$V_{DD} = 0.8$ V, $V_R = 0.5$ V
Time step for transient sim. (SPICE)	$\Delta t = 0.05$ ps

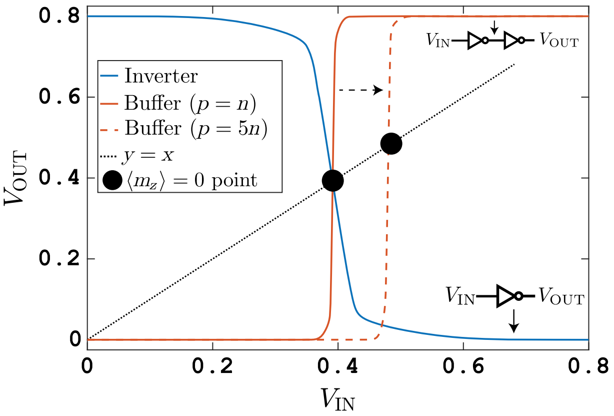


FIG. 5. 14-nm Predictive Technology Model, inverter or buffer. dc response of 14-nm HP FinFETs based on Ref. [44] for an inverter and buffer. Sizing the transistors differently allows the switching point to be shifted.

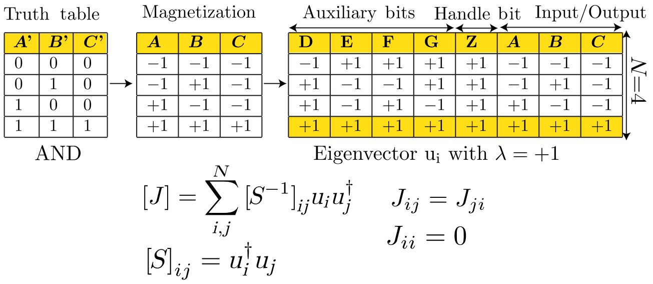
directions. Table I shows the parameters we use in Figs. 3 and 4. We note that this parameter selection is simply one possibility; many other parameters could have been used with no change in the basic conclusions.

Obtaining the sigmoidal response of CMOS+sLLG.—Each data point in the sigmoids shown in Figs. 3 and 4 is obtained by averaging the z component of the magnetization after 500 ns, with a time step of $\Delta t = 0.05$ ps. The CMOS inverter characteristics in conjunction with a spherical representation-based sLLG are obtained using the modular framework developed in Ref. [36] using HSPICE.

14-nm FinFET inverter characteristics.—Figure 5 shows the input and output characteristics of the single and double inverters that are used to amplify the stochastic signal that is generated by the MTJ (Fig. 3). At zero bias from the GSHE, the amplified signal V_M [Eq. (8)] is in the middle of V^+ and V^-, which is $V_{DD}/2 + V_R/2$. The buffer response can be shifted to this value by increasing the size of p-channel FETs, as shown in Fig. 5.

III. INVERTIBLE BOOLEAN LOGIC WITH BOLTZMANN MACHINES

We now present a mathematical prescription that shows how any given truth table can be implemented in terms of Boltzmann machines, in "one shot" with no learning being involved, unlike much of the past work in this area (see, for example, Refs. [49,50]). In Sec. II, we choose a simple $[J]$ and $\{h\}$ matrix to implement an AND gate based on Ref. [45]. In this section, we outline a general approach to show how any truth table can be implemented in terms of such matrices. Our approach, pictorially described in Fig. 6, begins by transforming a given truth table from binary (0,1) to bipolar $(-1, +1)$ variables. The lines of the truth table are then required to be eigenvectors each with eigenvalue $+1$, all other eigenvectors are assumed to have eigenvalues equal to 0. This leads to the following prescription for J as shown in Fig. 6:



A'	B'	C'
0	0	0
0	1	0
1	0	0
1	1	1

A	B	C
−1	−1	−1
−1	+1	−1
+1	−1	−1
+1	+1	+1

D	E	F	G	Z	A	B	C
−1	+1	+1	+1	+1	−1	−1	−1
+1	−1	−1	+1	+1	−1	+1	−1
+1	−1	+1	−1	+1	+1	−1	−1
+1	+1	+1	+1	+1	+1	+1	+1

$$[J] = \sum_{i,j}^{N} [S^{-1}]_{ij} u_i u_j^{\dagger} \qquad J_{ij} = J_{ji}$$

$$[S]_{ij} = u_i^{\dagger} u_j \qquad J_{ii} = 0$$

FIG. 6. Truth table to J matrix. A given truth table is first transformed from binary to bipolar variables by using the transformation $m = 2t - 1$, where m and t represent the magnetization and binary values of the truth table. Additional bits are introduced to each line of the truth table to ensure that the resultant S matrix is invertible. The indices i, j correspond to the number of lines in the truth table. u_i, u_j are column vectors. As an example, we show auxiliary bits that result in an S matrix equal to the identity matrix, since the eigenvectors are orthogonal. The J matrix is then obtained by Eq. (12a), which ensures that the truth table corresponds to the low-energy states of the Boltzmann machines according to Eq. (4). A handle bit of $+1$ is introduced to each line of the truth table, which can be biased to ensure that the complementary truth table does not appear along with the desired one. This bit also allows a truth table to be electrically reconfigured into its complement.

$$[J] = \sum_{i,j} [S^{-1}]_{ij} u_i u_j^{\dagger}, \qquad (12a)$$

$$S_{ij} = u_i^{\dagger} u_j, \qquad (12b)$$

where u_i are the eigenvectors corresponding to lines in the truth table of a Boolean operation and S is a projection matrix that accounts for the nonorthogonality of the vectors defined by different lines of the truth table. Note that the resultant J matrix is always symmetric ($J_{ij} = J_{ji}$) with diagonal terms that are subtracted in our models such that $J_{ii} = 0$. The number of p-bits in the system is made greater than the number of lines in a truth table through the addition of hidden units (Fig. 6) to ensure that the number of conditions we impose is less than the dimension of the space defined by the number of p-bits.

Another important aspect in the construction of $[J]$ is that an eigenvector u_i implies that its complement $-u_i$ is also a valid eigenvector. However, only one of these might belong to a truth table. We introduce a "handle" bit to each u_i that is biased (h_i) to distinguish complementary eigenvectors. These handle bits provide the added benefit of reconfigurability. For example, AND and OR gates have complementary truth tables, and a given gate can be electrically reconfigured as an AND or an OR gate using the handle bit.

J matrices for AND and FA.—We now provide the details of the J matrix for the AND gate, obtained using the prescription shown in Fig. 6 based on Eq. (12a). The eigenvectors of the truth table for the AND in Fig. 6 are placed into a matrix U, such that $U = [\,u_1 \quad u_2 \quad u_3 \quad u_4\,]$, where u_1 is the first row of the matrix shown in Fig. 6, $u_1 = [-1+1+1+1+1-1-1-1]^T$, and so on. In matrix notation, the S matrix can be written as

$$S = U^T U = 8I_{4\times4}. \tag{13}$$

Then the J matrix becomes

$$J = \sum_{ij} \underbrace{[S^{-1}]_{ij}}_{1/8\delta_{ij}} u_i u_j^\dagger = 1/8 \sum_i u_i u_i^\dagger. \tag{14}$$

Removing the diagonal entries by making $J_{ii} = 0$ and multiplying the matrix entries by 2, to obtain simple integers, J_{AND} evaluates to

$$J_{\mathrm{AND}} = \begin{pmatrix} 0 & -1 & 0 & 0 & 1 & 1 & 1 & 0 \\ -1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & -1 & 0 \\ 0 & 1 & 0 & 0 & 1 & -1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & -1 \\ 1 & 0 & 1 & -1 & 0 & 0 & 0 & 1 \\ 1 & 0 & -1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & -1 & 1 & 1 & 0 \end{pmatrix}, \tag{15}$$

with the notation [1–5, auxiliary bit and handle bit; 6,“A”; 7,“B”; 8,“C”]. Following a similar procedure, we use the following 14×14 full adder matrix J_{FA}:

$$J_{\mathrm{FA}} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & -1 & -1 & -1 & -1 & -2 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & -1 & -1 & 2 & -1 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & -1 & -1 & -1 & 2 & 1 & -1 \\ 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & -1 & -2 & 1 & 1 & -1 & 1 \\ 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & -1 & 2 & -1 & -1 & 1 & -1 \\ 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 1 & -2 & -1 & 1 \\ 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & -2 & 1 & -1 & 1 \\ 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 1 & 1 & 2 & 1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & -1 & -2 & 2 & 1 & 1 & 1 & 0 & 0 & -1 & -1 & 1 & 2 \\ -1 & 2 & -1 & 1 & -1 & 1 & -2 & 1 & 0 & -1 & 0 & -1 & 1 & 2 \\ -1 & -1 & 2 & 1 & -1 & -2 & 1 & 1 & 0 & -1 & -1 & 0 & 1 & 2 \\ -2 & 1 & 1 & -1 & 1 & -1 & -1 & 2 & 0 & 1 & 1 & 1 & 0 & -2 \\ -1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & 0 & 2 & 2 & 2 & -2 & 0 \end{pmatrix}, \tag{16}$$

with the notation [1–9, auxiliary bits and handle bit; 10, “C_{in}”; 11, “B”; 12, “A”; 13, “S”; 14, “C_{out}”].

These are the J matrices (AND and FA) that are used for all examples in the paper, except for the AND gate described in Sec. II. Figure 10 shows the “truth table” operation of the full adder where all input or output terminals are “floating” using the J matrix of Eq. (16), showing excellent quantitative agreement with the Boltzmann distribution of Eq. (4) at steady state even for the undesired peaks of the truth table.

Note that this prescription for $[J]$ is similar to the principles developed originally for Hopfield networks [Ref. [51] and Eq. (4.20) in Ref. [20]]. However, other approaches are possible along the lines described in the context of Ising Hamiltonians for quantum computers [45]. We have tried some of these other designs for $[J]$, and many of them lead to results similar to those we present here. For practical implementations, it is important to evaluate different approaches in terms of their demands on the dynamic range and accuracy of the weight logic.

Description of universal model.—Once a J matrix and the h vector are obtained for a given problem, the system is initialized by randomizing all m_i at time $t = t_0$. First, the current (voltage) that a given p-bit (m_i) feels due to the other coupled m_j is obtained from Eq. (2), and the m_i value is updated according to Eq. (1). Next, the procedure is repeated for the remaining p-bits by finding the current they receive due to all other m_i using the *updated* values of

m_i. For this reason, the order of updating is chosen randomly in our models and we find that the order of updating has no effect in our results. However, updating the p-bits in *parallel* leads to incorrect results. These two observations are well known in the context of Hopfield networks and Boltzmann machines [52–54]. This type of serial updating corresponds to the "asynchronous dynamics" [20,55]. We note that the hardware implementation we discuss in this paper naturally leads to an asynchronous updating of p-bits in the absence of a global clock signal. We have set up an online simulator based on this model in [56] so that interested readers can simulate some of the examples discussed in this paper.

Figure 7 shows the time evolution of an AND based on Eq. (15). Initially for $t < t_0$, the interaction strength is zero ($I_0 = 0$), making the pseudotemperature of the system infinite and the network produces uncorrelated noise visiting each state with equal probability. In the second phase ($t > t_0$), the interaction strength is suddenly increased to $I_0 = 2$, effectively "quenching" the network by reducing the temperature. This correlates the system such that only the states corresponding to the truth table of the AND gate are visited, each with equal probability when a long-time average is taken. The average probabilities in each phase quantitatively match the Boltzmann law defined by Eq. (4).

In Fig. 8, we show how a correlated network producing a given truth table can be used to do directed computation analogous to standard CMOS logic. An OR gate is constructed by using the same $[J]$ matrix for an AND gate, but with a negated handle bit. By "clamping" the input bits of an OR gate ($t < t_0$) through their bias terminals h_i to $(A, B) = (+1, +1)$, the system is forced to only one of the peaks of the truth table, effectively making C = 1.

The PSL gates, however, exhibit a remarkable difference with standard logic gates, in that inputs and outputs are on an equal footing. Not only do clamped inputs give the corresponding output, a clamped output gives the corresponding input(s). In the second phase ($t > t_0$), the output of the OR gate is clamped to +1, which produces three possible peaks for the input terminals, corresponding to various possible input combinations that are consistent with the clamped output $(A, B) = (0, 1)$, (1,0), and (1,1). The probabilistic nature of PSL allows it to obtain multiple solutions [Fig. 8(c)]. It also seems to make the results more resilient to *unwanted* noise due to stray fields that are inevitable in physical implementations, as shown in Fig. 9. Here, we simulate an AND gate in the presence of a normally distributed random noise that enters the bias fields of each p-bit and define the computation to be faulty, if the mode (most frequent value) of the output bit is not consistent with the programed input combinations after $T = 100$ time steps. We observe that even large levels of uncontrolled noise produce correct results with high probabilities.

Figure 10 shows the design of a full adder with the 8-line truth table shown. There are three inputs in all, two

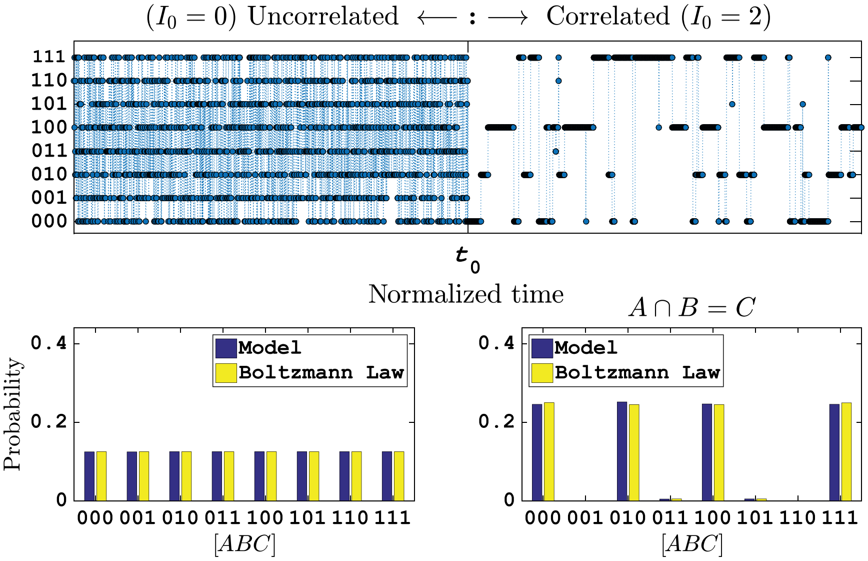


FIG. 7. Correlated p-bits, AND gate. When the interaction strength (I_0) is zero, p-bits produce uncorrelated noise, visiting all possible states with equal probability. In this example, the interaction strength (pseudo inverse temperature) is suddenly increased from 0 to 2 as a step function at $t = t_0$, to effectively "quench" the network. This correlates the p-bits to produce the truth table of an AND gate (AND: $A \cap B = C$). Note that after this quenching, the p-bits visit only the low-energy states corresponding to the truth table of the AND gate, and once the system is in one of the low-energy states, it tends to stay there for a while, until being kicked out by the thermal noise. The time averages of the uncorrelated and the correlated system are well explained by the Boltzmann law stated in Eq. (4). The total simulation uses $T = 4 \times 10^6$ steps to compare the results with the Boltzmann distribution, though only a fraction are shown in the upper panel for clarity.

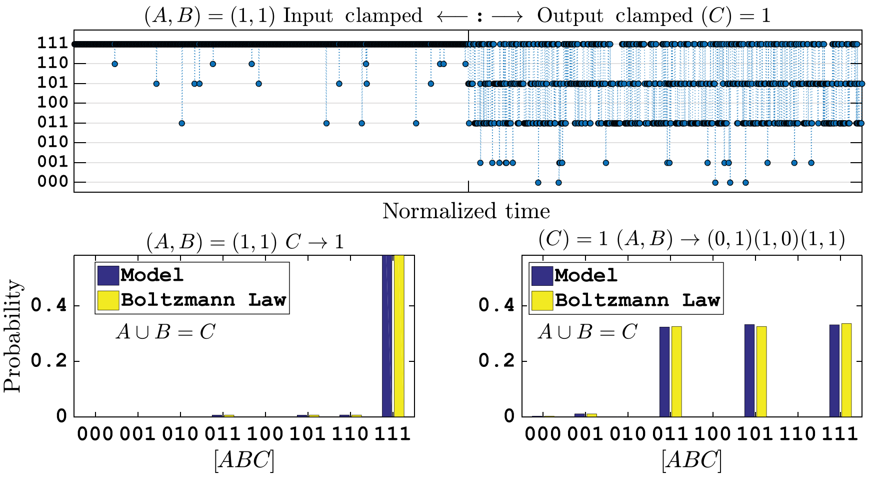


FIG. 8. Implementing a Boolean function and its inverse.: The input or output terminals of an appropriately interconnected network of p-bits can be "clamped" to perform a specific logic operation or its *inverse*. In this example, the input bits (A, B) of an OR gate are clamped to be $+1$, forcing the output bit C to be 1, during the first phase of operation $(t < t_0)$. In the second phase of operation $(t > t_0)$, the output of the OR gate C is clamped to the value $+1$, which is consistent with three different combinations of (A, B). As shown in the time response and the long-time histogram plots, all three possibilities emerge with equal probability, demonstrating the "inverse" OR operation. In each case, the expected probabilities from the Boltzmann law [Eq. (4)] closely match those produced by the generic model, Eqs. (1) and (2), after running the system for 10^6 steps. Only a fraction are shown in the upper panel for clarity.

from the numbers to be added and one carry bit from previous FA. It produces two outputs, one the sum bit and the other a carry bit to be passed on to the next FA. The probabilities of different states are calculated using J_{FA} from Eq. (16), with $I_0 = 0.5$ in the truth table mode, where all inputs and outputs are floating and the states are numbered using the decimal number corresponding to the binary word $[C_i \quad A \quad B \quad S \quad C_o]$. The decimal numbers corresponding to the truth table are shown in the inset, and these match the location of the taller peaks in the histogram. Note that the Boltzmann distribution [Eq. (4)] quantitatively matches the model even for the suppressed peaks. A higher I_0 reduces these suppressed peaks further. The statistics are collected for $T = 10^6$ steps, and each terminal output is then placed in the histogram.

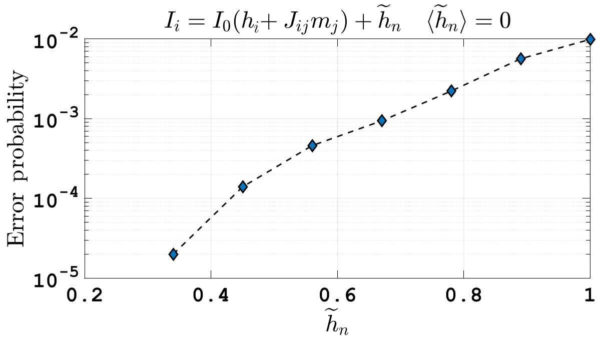
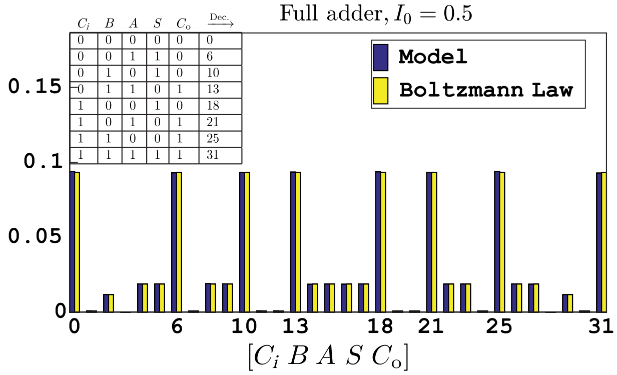


FIG. 9. Noise tolerance of AND. The probability of a wrong output for an (AND) gate [Eq. (15)] operated with clamped inputs is investigated in the presence of a random noise field which enters Eq. (2) as indicated in the figure. The noise is assumed to be uniformly distributed over all p-bits in a given network, and centered around zero with magnitude $\pm\widetilde{h}_n$, where $(I_0 = 2, h_i = \pm 1)$. Each gate is simulated 50 000 times for $T = 100$ time steps to produce an error probability for a given noise value, and the maximum peak produced by the system is assumed to be an output that can be read with certainty. The system shows robust behavior even in the presence of large levels of noise.



C_i	B	A	S	C_o	$\xrightarrow{\text{Dec.}}$
0	0	0	0	0	0
0	0	1	1	0	6
0	1	0	1	0	10
0	1	1	0	1	13
1	0	0	1	0	18
1	0	1	0	1	21
1	1	0	0	1	25
1	1	1	1	1	31

FIG. 10. Full adder. Full adder in the truth table mode, where all inputs and outputs are floating, calculated using J_{FA} from Eq. (16), with $I_0 = 0.5$. The statistics are collected for $T = 10^6$ steps, and each terminal output is then placed in the histogram. The states are numbered using the decimal number corresponding to the binary number $[C_i \quad A \quad B \quad S \quad C_o]$. The decimal numbers corresponding to the truth table are shown in the inset, and these match the location of the taller peaks in the histogram. Note that the Boltzmann distribution [Eq. (4)] quantitatively matches the model even for the suppressed peaks.

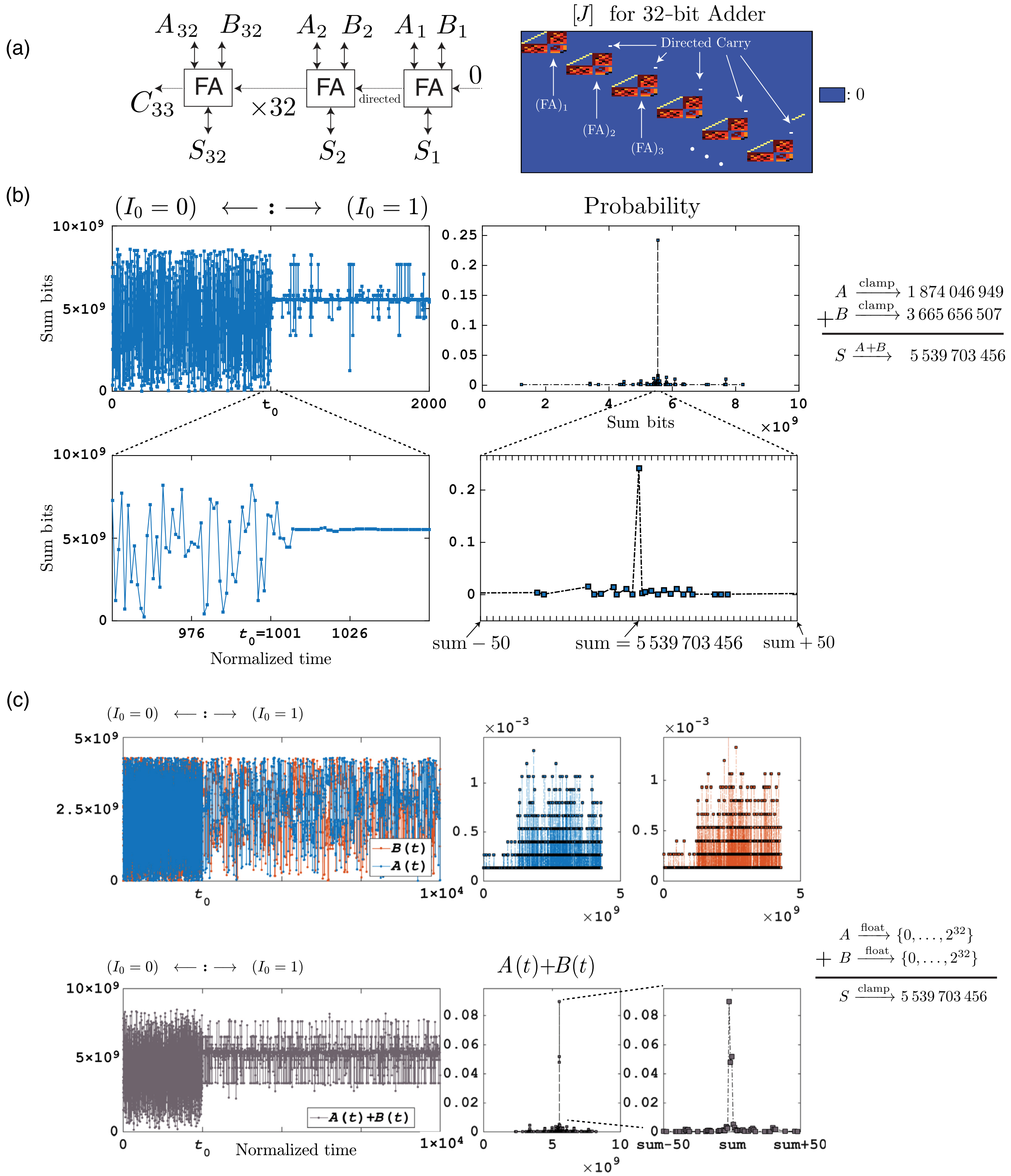


FIG. 11. 32-bit ripple carry adder (RCA). (a) 32-bit ripple carry adder is designed using individual full adder units with the carry bit designed as a *directed* connection from the least significant bit to the most significant bit. The overall J matrix for a 32-bit adder J matrix is shown, and it is quite sparse and quantized. (b) For $t < t_0$, $I_0 = 0$ and the sum fluctuates randomly. At $t = t_0$, I_0 is suddenly increased, and the adder converges on the correct result for two random inputs A and B. The distribution of 1000 data points ($t > t_0$) shows a single peak with 24% probability of time spent in the correct state (not including the uncorrelated time points for $t < t_0$). (c) Even though the connections between the full adder units are directed, the system performs the inverse function as well. When the output (S) is clamped to a fixed number, the inputs (A) and (B) fluctuate in a correlated manner to make $A + B = S$ when $I_0 = 1$. Note the broad distributions of A and B (collected for $t > t_0$) as compared to the extremely sharp distribution of $A + B$.

IV. DIRECTED NETWORKS OF BOLTZMANN MACHINES

When constructing larger circuits composed of individual Boltzmann machines, the reciprocal nature of the Boltzmann machine often interferes with the directed nature of computation that is desired. It seems advisable to use a hybrid approach. For example, in constructing a 32-bit adder we use full adders that are individually BMs with symmetric connections, $J_{ij} = J_{ji}$. But when connecting the carry bit from one FA to the next, the coupling element J_{ij} is nonzero in only one direction from the least significant to the most significant bit. This directed coupling between the components distinguishes PSL from purely reciprocal Boltzmann machines. Indeed, even the full adder could be implemented not as a Boltzmann machine but as a directed network of more basic gates. But then it would lose its invertibility. On the other hand, the directed connection of BM full adders largely preserves the invertibility of the overall system, as we show.

A. 32-bit adder or subtractor

Figure 11 shows the operation of a 32-bit adder that sums two 32-bit numbers A and B to calculate the 33-bit sum S. In the initial phase ($t < t_0$), we have $I_0 = 0$ corresponding to infinite temperature so that the sum bits (S) fluctuate among $2^{33} \approx 8 \times 10^9$ possibilities. With $I_0 = 1$, Fig. 11 shows that the correct answer has a probability of ≈12%, which is much lower than the ≈100% that can be achieved with larger I_0 values (as in Figs. 13(c) with $I_0 = 5$). Nevertheless, the peak is unmistakable, as evident from the expanded scale histogram, and the correct answer is extracted from the majority vote of $T = 100$ samples, as shown in Fig. 13. This ability to extract the correct answer despite large fluctuations is a general property of probabilistic algorithms.

Interestingly, although the overall system includes several unidirectional connections, it seems to be able to perform the inverse function as well. With A and B clamped it calculates $S = A + B$, as noted above. Conversely, with S clamped, the input bits A and B fluctuate in a correlated manner so as to make their sum sharply peaked around S. Figure 11 shows the time evolution of the input bits that have broad distributions spanning a wide range. Initially, when I_0 is small, the sum of A and B also shows a broad distribution, but once I_0 is turned up to 1, the distributions of A and B get strongly correlated making the distribution of $A + B$ sharply peaked around the fixed value of S. It must be noted that the 32-bit adder shown in Fig. 11 is not like standard digital circuits which are not invertible. The demonstration of such an invertible 32-bit adder could be practically significant, since binary addition is noted to be the most fundamental and frequently used operation in digital computing [57].

Delay of ripple carry adder.—Just as in CMOS-based ripple carry adders (RCA), the delay of the p-bit-based RCA is a function of the inputs A and B. In Fig. 12, we have systematically studied the worst-case delay of the p-bit-based RCA as a function of increasing bit size. We selected a "worst-case" combination that results in a carry that needs to be propagated from bit 1 to bit N, which results in a linear increase in the delay, exhibiting $O(n)$ complexity with input size similar to CMOS implementations [58]. When the inputs are random, the delay seems to increase sublinearly. The system is quenched at $t = 0$ for different interaction parameters I_0 and the delay is defined to be the time it takes for the system to settle to the mode of the array for $T = 200$. An error check has been carried out separately to ensure the calculated sum (mode) is always exactly equal to the expected sum. For random inputs the 32-bit adder is close to 20 time steps, in accordance with the example shown in Fig. 11.

Digital accuracy and logical invertibility.—The striking combination of accuracy and invertibility is made possible by our hybrid design, whereby the individual full adders are Boltzmann machines, even though their connection is directed. Our 32-bit adder is more like a collection of interacting particles than like a digital circuit, as evident from Fig. 13(a), which shows a color map of the binary state of each of the 448 p-bits as a function of time with the interaction parameter I_0 suddenly increased from 0.25 to 5 at $t_0 = 50$, thereby quenching a "molten liquid" into a "solid." Nevertheless, it shows the striking accuracy of a digital circuit, with S–A–B exactly equal to zero in each of the 1000 trials, as shown in Fig. 13(b). We do not expect a

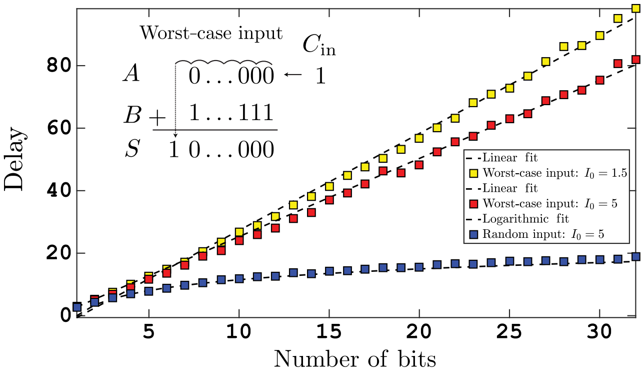


FIG. 12. Ripple carry adder delay. The delay of the RCA as a function of number of bits in the ripple carry adder is shown. The worst-case input combination generates a carry that propagates all the way through bit 1 to bit N, and has a linear dependence on the number of bits, exhibiting $O(n)$ complexity. When the inputs are random, the delay increases logarithmically. The delay is defined to be the time it takes for the network to reach the mode of the array for $T = 200$ after getting quenched at $t = 0$. Each point is an average of 500 trials with random initial conditions for an $I_0 = 1.5$, and the mode of the array is exactly equal to the arithmetic sum of the inputs in each case. The worst-case inputs are $A = 0\ldots000$ and $B = 1\ldots111$ with an input carry (C_{in}) of 1. Results show a weak I_0 dependence.

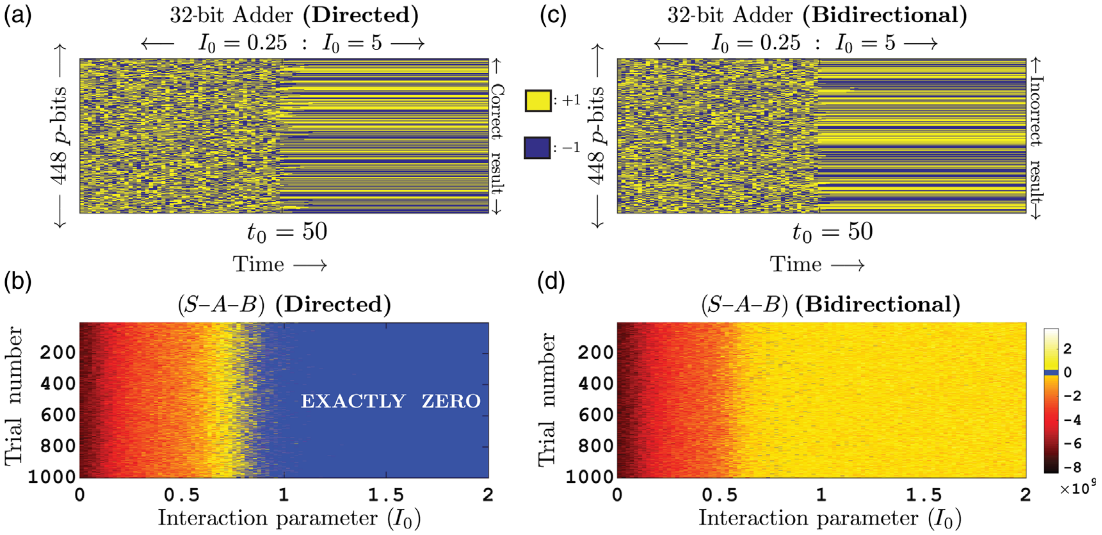


FIG. 13. Accuracy of 32-bit adder, directed versus bidirectional. The results are shown for the adder operating in a subtractor mode, clamping one (random) 32-bit input (A) and a (random) 33-bit output ($C_{\mathrm{out}} + S$), and observing the other 32-bit input B, which should provide the difference S–A. (a) Color map of the binary state of each of the 448 p-bits comprising the directed adder as a function of time with the interaction parameter I_0 suddenly increased from 0.25 to 5 at $t_0 = 50$. For low values of I_0 at $t < 50$, the collection of p-bits is like a molten liquid which is quenched at $t_0 = 50$ into a solid. (b) Surprisingly, this solid corresponds to a "perfect crystal" in each of the 1000 trial experiments, with S–A–B exactly equal to zero (dark blue). (c) Same as (a) but for a bidirectional adder. Here, too, the "liquid" quenches to a solid at $t_0 = 50$, but in this case the resulting "solid" is full of defects (with hardly any zeros), with S–A–$B \neq 0$, yielding a different wrong result for each trial as evident from (d). For (c) and (d) the color bar is modified to have a dark blue color corresponding to exactly zero. S, A, B are taken to be the statistical mode of the 100×1 array obtained at the end of each trial.

molten liquid to be quenched into a "perfect crystal" every time. Instead, we would expect a "solid full of defects" with different nonzero values for S–A–B in each trial. That is exactly what we get if the carry bits are bidirectional, as in a fully BM implementation [Fig. 13(d)].

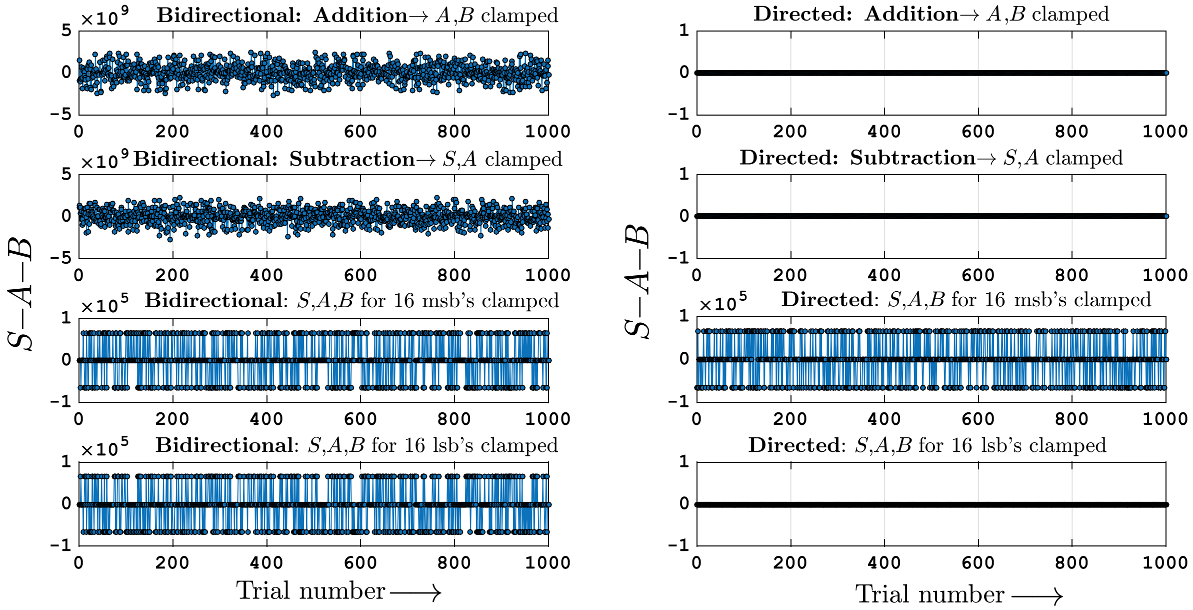
Note, however, that this digital accuracy is achieved while maintaining the property of invertibility that is absent in digital circuits. Figure 13 is not for direct mode operation, but for the adder operating in reverse mode as a subtractor. It might be expected that the directed connection of carry bits from the less significant to the more significant bit could lead to a loss of invertibility. To investigate this point, we show the error S–A–B as a function of trial number (Fig. 14) for four different modes of operation with (i) A and B clamped (addition), (ii) S and A clamped (subtraction), (iii) A, B, and S for the 16 most significant bits (msb) clamped, and (iv) A, B, and S for the 16 least significant bits (lsb) clamped. The fully bidirectional implementation shows very large errors for all modes of operation. The directed implementation, on the other hand, works perfectly for both the adder and the subtractor modes. It also works if we clamp the least significant bits, but not if we clamp the most significant bits. This seems reasonable since we expect to be able to control a flow by making changes upstream (lsb) but not downstream (msb).

Partial directivity.—Thus far in our examples we have only considered fully directed ($J_{ij} = 2J_0$, $J_{ji} = 0$) or fully bidirectional ($J_{ij} = J_0$, $J_{ji} = J_0$) carry bits when connecting the individual full adders. In Fig. 15, we systematically analyze the effects of partial directivity in the operation of a 32-bit adder. We observe that the 32-bit adder operates correctly even when there is a large degree of bidirectionality ($J_{ji} = J_{ij} \times 0.75$) provided that the system is allowed to run for a long time, $T = 50000$, in stark contrast to the fully directed case that could resolve the right answer within $T = 100$, shown in Fig. 14(b). Decreasing the time steps systematically increases the error. Increasing the correlation parameter while keeping T constant also seems to adversely affect the bidirectional designs that might be getting the system stuck in local minima.

Directionality and computation time, $2 - p$-bit model.—The qualitative relation between I_0, T, and bidirectionality J_{12}/J_{21} described above is derived from extensive numerical simulations based on Eqs. (1) and (2). However, the broad features can be understood from a model involving just two p-bits, 1 and 2, with

$$h = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \text{and} \quad J = \begin{bmatrix} 0 & J_{12} \\ J_{21} & 0 \end{bmatrix}.$$

It is straightforward to write a master equation describing the time evolution of the probabilities of different configurations:


Bidirectional: Addition→ A,B clamped

Bidirectional: Subtraction→ S,A clamped

Bidirectional: S,A,B for 16 msb's clamped

Bidirectional: S,A,B for 16 lsb's clamped

Directed: Addition→ A,B clamped

Directed: Subtraction→ S,A clamped

Directed: S,A,B for 16 msb's clamped

Directed: S,A,B for 16 lsb's clamped

FIG. 14. Invertibility of 32-bit adder, directed versus bidirectional. An adder that provides the sum S of two 32-bit numbers A and B: $S = A + B$. The left-hand panel shows the adder implemented with bidirectional carry bits, while the right-hand panel shows one with carry bits directed from the least significant to the most significant bit. Four different modes are shown with (i) A and B clamped (addition), (ii) S and A clamped (subtraction), (iii) A, B, and S for the 16 most significant bits (msb) clamped, and (iv) A, B, and S for the 16 least significant bits (lsb) clamped. Note that the bidirectional implementation shows very large errors for all modes of operation. The directed implementation works perfectly for both the adder and the subtractor modes. It also works if we clamp the least significant bits, but not if we clamp the most significant bits. Correlation parameter $I_0 = 1$, $T = 100$ steps for all trials. S, A, B are taken to be the mode (most frequent value) of the 100×1 array obtained at the end of each trial. Clamped inputs are random 32-bit words for each trial, for a total of 1000 trials.

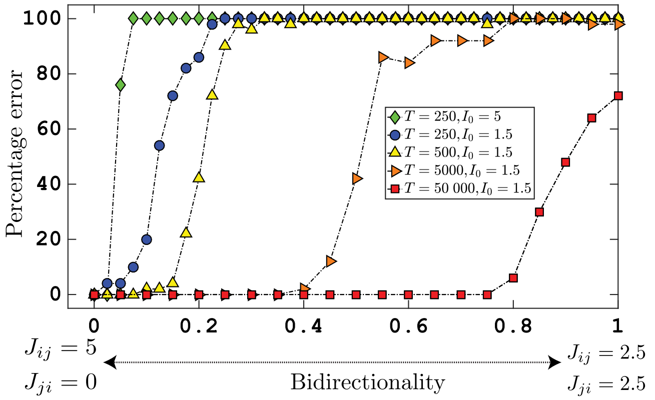


FIG. 15. Error versus bidirectionality. The degree of bidirectionality J_{ji}/J_{ij} of the carry-out (j) to carry-in (i) link between the full adders is systematically varied while keeping the sum $J_{ij} + J_{ji}$ constant. In each case the sum is obtained from the statistical mode (or majority vote) of T time samples over 50 trials. The y axis shows the fraction of trials that yield the wrong result. Note that for large I_0 and small T, error-free operation is obtained only if bidirectionality is close to zero, similar to standard digital circuits. But with $I_0 = 1.5$ and $T = 50000$, error-free operation (at least for 50 trials) is obtained even with ≈75% bidirectionality.

$$\frac{d}{dt}\begin{bmatrix} P_{11} \\ P_{10} \\ P_{01} \\ P_{00} \end{bmatrix} = [W]\begin{bmatrix} P_{11} \\ P_{10} \\ P_{01} \\ P_{00} \end{bmatrix},$$

W being the transition matrix [20], P_{00} representing the probability of both p-bits being -1, P_{11} both being $+1$, and so on. We can write two matrices W_1 and W_2 describing the updating of p-bits 1 and 2, respectively:

$$W_1 = \begin{array}{c} (1,2) \\ (11) \\ (10) \\ (01) \\ (00) \end{array}\begin{array}{c} \begin{array}{cccc} (11) & (10) & (01) & (00) \end{array} \\ \begin{bmatrix} p & 0 & p & 0 \\ 0 & \overline{p} & 0 & \overline{p} \\ \overline{p} & 0 & \overline{p} & 0 \\ 0 & p & 0 & p \end{bmatrix} \end{array},$$

$$W_2 = \begin{array}{c} (1,2) \\ (11) \\ (10) \\ (01) \\ (00) \end{array}\begin{array}{c} \begin{array}{cccc} (11) & (10) & (01) & (00) \end{array} \\ \begin{bmatrix} q & q & 0 & 0 \\ \overline{q} & \overline{q} & 0 & 0 \\ 0 & 0 & \overline{q} & \overline{q} \\ 0 & 0 & q & q \end{bmatrix} \end{array},$$

where $W(i,j)$ represents the probability that state (j) makes a transition to state (i), and $\bar{p} = 1 - p$, $\bar{q} = 1 - q$. p and q are obtained from Eqs. (1) and (2):

$$p = \frac{1}{2}\{1 + \tanh[I_0(J_{12} + h_1)]\} = \frac{1}{2}[1 + \tanh(I_0 J_{12})],$$
$$q = \frac{1}{2}\{1 + \tanh[I_0(J_{21} + h_2)]\} = \frac{1}{2}[1 + \tanh(I_0 J_{21})].$$

The overall transition matrix W is given by $W_2 \times W_1$ or $W_1 \times W_2$ depending on which bit is updated first. Either way, the matrix W has four eigenvalues, $\lambda_1 = 1$, $\lambda_2 = 0$, $\lambda_3 = 0$, and $\lambda_4 = (2p - 1)(2q - 1) = \tanh(I_0 J_{12}) \tanh(I_0 J_{21})$, and the corresponding eigenvectors evolve with time $\sim\lambda^T$.

The components corresponding to $\lambda = 0$ decay instantaneously while the eigenvector corresponding to $\lambda = 1$ is the stationary result representing the correct solution. But for the system to reach this state, we have to wait for the fourth eigenvector corresponding to λ_4 to decay sufficiently. A fully directed network has $J_{21} = 0$, so that $\lambda_4 = 0$ and the system quickly reaches the correct solution. But in a bidirectional network with $J_{12} = J_{21}$, the fourth eigenvalue can be quite close to one, especially for large I_0, and take an exponentially long time to decay, as $\lambda^T = \exp(T \ln \lambda) \approx \exp[-T(1 - \lambda)]$ when λ is close to 1.

This $2 - p$-bit model provides some insight into our general observation that directivity can be used to obtain accurate answers quickly. However, depending on the problem at hand, it may be desirable to retain some degree of bidirectionality, since full directivity does lead to some loss of invertibility, as we see for one set of inputs in Fig. 14. We discuss an example of a partially directed p-bit network in the next section.

B. 4-bit multiplier or factorizer

Figure 16 shows how the invertibility of PSL logic blocks can be used to perform integer factorization using a multiplier in reverse. Normally, the factorization problem requires specific algorithms [59] to be performed in CMOS-like hardware; here, we simply use a digital 4-bit multiplier working in *reverse* to achieve this operation.

Specifically with the output of the multiplier clamped to a given integer from 0 to 15, the input bits float to the correct factors. The interconnection strength I_0 is increased suddenly from 0 to 2 at $t = t_0$ (Fig. 16) and the input bits

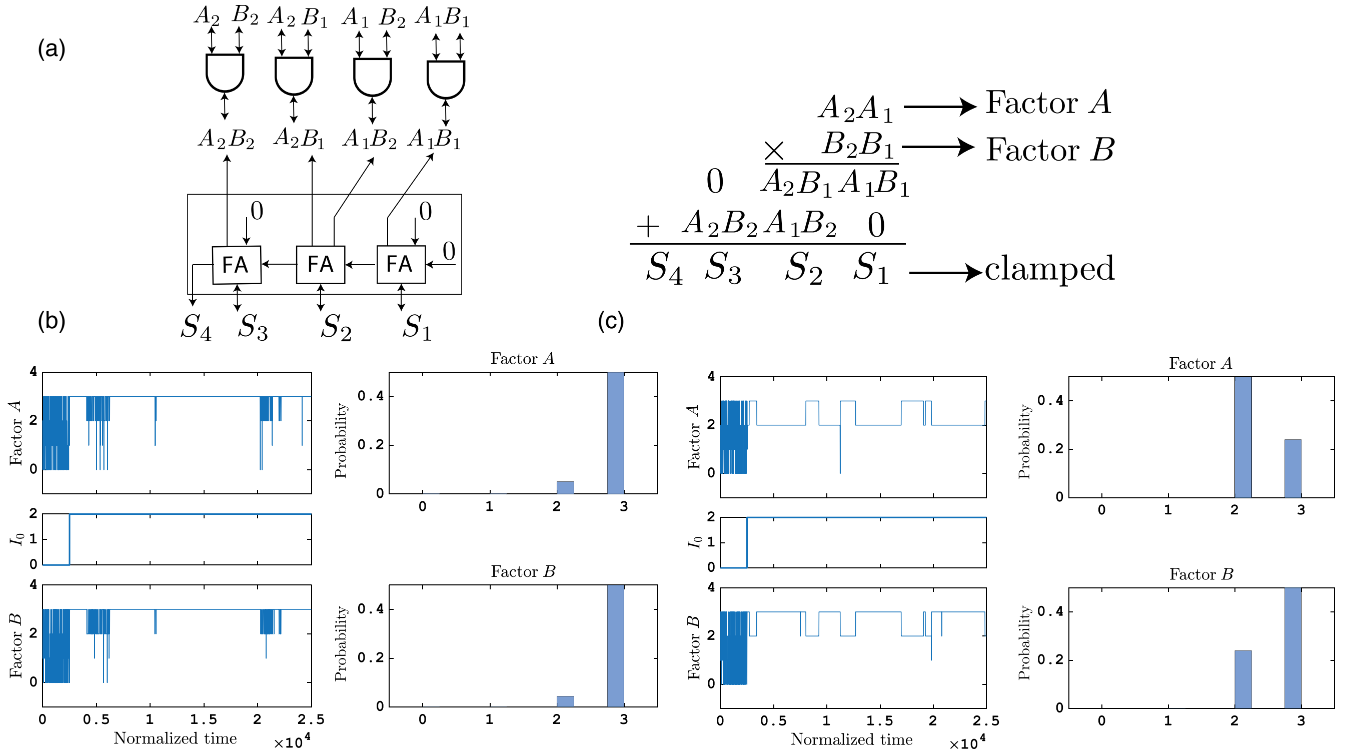


FIG. 16. Factorization through inverse multiplication. The reversibility of PSL allows the operation of integer factorization using a binary multiplication circuit implemented using the principles of digital logic using AND gates and full adders, as shown in (a). The output nodes of a 4-bit multiplier are clamped to a given integer, and the system produces the only consistent factors of the product at the input terminals, probabilistically. The interaction parameter I_0 is suddenly increased to a saturation value of 2, and held constant as shown. (b) The output terminal is clamped to 9 and is factored into 3×3; note that 9×1 is not an achievable solution in this setup since encoding 9 requires 4-bit inputs in binary, whereas inputs are limited to 2-bits. (c) The output terminal is clamped to 6 and after being correlated, the factors cross-oscillate between 2 and 3. In both cases the histogram is obtained by counting outputs after $t > t_{\mathrm{total}}/2 = 1.25 \times 10^4$ time steps to collect statistics after the system is thermalized.

get locked to one of the possible solutions. For example, when the output is set to 9, both inputs float to 3. With the output set to 6, both inputs fluctuate between two values, 2 and 3. Note that factors like $9 = 9 \times 1$ do not show up, since encoding 9 in binary requires 4-bits (1001) and the input terminals only have 2-bits. We check other cases where factorizing 3 shows both 3×1 and 1×3, and factorizing zero shows all possible peaks since there are many solutions such that $0 = 0 \times 1$, 2, 3 and so on.

We also keep the same directed connections between the full adders for the carry bits, making them a directed network of Boltzmann machines, similar to the 32-bit adder. Moreover, we keep a directed connection *from* the full adders *to* the AND gates, as shown in Fig. 16(a) since the information needs to flow from the output to the input in the case of factorization. The input bits that go to multiple AND gates are "tied" to each other with a positive exchange ($J > 0$) value much like 2-spins interacting ferromagnetically; however, in PSL we envision these interactions to be controlled purely electrically. In this example, we observe that the system is sensitive to the relative strengths of couplings within the AND gates and between the AND gates and the full adders, which can also depend on a chosen annealing profile.

The design of factorizers of practical relevance is beyond the scope of this paper. Our main purpose is to establish how the key feature of invertibility of p-bits can be creatively used for different circuits with unique functionalities. The demonstration of 4-bit factorization through reverse multiplication is similar to memcomputing [60] based on deterministic memristors. Note, however, that the building blocks and operating principles of stochastic p-bits and memcomputing [61] are very different and the only similarity we note here is the fact that both approaches treat the input and output terminals on an equal footing.

V. SUMMARY

It is generally believed that (1) probabilistic algorithms can tackle specific problems much more efficiently than classical algorithms [62], and that (2) probabilistic algorithms can run far more efficiently on a probabilistic computer than on a deterministic computer [62,63]. As such, it seems reasonable to expect that probabilistic computers based on robust room-temperature p-bits could provide a practically useful solution to many challenging problems by rapidly sampling the phase space in hardware.

In this paper, we present a framework for using probabilistic units or "p-bits" as a building block for a probabilistic spin logic, which is used to implement precise Boolean logic with an accuracy comparable to standard digital circuits while exhibiting the unique property of invertibility that is unknown in deterministic circuits. Specifically, first, we present an implementation based on stochastic nanomagnets to illustrate the importance of three-terminal building blocks in the construction of large-scale correlated networks of p-bits. We emphasize that this is just one possible implementation that is by no means the only one (Sec. II). Second, we present an algorithm for implementing Boolean gates as BM with relatively sparse and quantized J-matrix elements, benchmark their operation against the Boltzmann law, and establish their capability to perform not just direct functions but also their inverse (Sec. III). Third, we present a 32-bit adder implemented as a hybrid BM that achieves digital accuracy over a broad combination of the interaction parameter I_0, directionality, and the number of samples T. This striking accuracy is reminiscent of digital circuits, but it is achieved while preserving a certain degree of invertibility that is absent in digital circuits. The accuracy is particularly surprising with high degrees of bidirectionality ($J_{12} = 0.75 \times J_{21}$), where the system is picking out the one correct answer out of nearly $2^{33} \approx 8 \times 10^9$ possibilities. This may require a larger number of time samples, but these could be collected rapidly at GHz rates (Sec. IV).

We hope these findings will help emphasize a new direction for the field of spintronic and nanomagnetic logic by shifting the focus from stable high-barrier magnets to stochastic, low-barrier magnets, while inspiring a search for other possible physical implementations of p-bits.

ACKNOWLEDGMENTS

It is a pleasure to acknowledge many helpful discussions with Behtash Behin-Aein (Globalfoundries) and Ernesto E. Marinero (Purdue University). We thank Jaijeet Roychowdhury (UC Berkeley) for suggesting the phrase "invertible." This work was supported in part by C-SPIN, one of six centers of STARnet, a Semiconductor Research Corporation program, sponsored by MARCO and DARPA, in part by the Nanoelectronics Research Initiative through the Institute for Nanoelectronics Discovery and Exploration (INDEX) Center, and in part by the National Science Foundation through the NCN-NEEDS program, Contract No. 1227020-EEC.

[1] L. Lopez-Diaz, L. Torres, and E. Moro, *Transition from Ferromagnetism to Superparamagnetism on the Nanosecond Time Scale*, Phys. Rev. B **65**, 224406 (2002).

[2] K. Palem and A. Lingamneni, *Ten Years of Building Broken Chips: The Physics and Engineering of Inexact Computing*, ACM Trans. Embed. Comput. Syst. **12**, 87 (2013).

[3] S. Cheemalavagu, P. Korkmaz, K. V. Palem, B. E. S. Akgul, and L. N. Chakrapani, in *Proceedings of the International Federation for Information Processing International Conference on Very Large Scale Integration, Perth, Australia* (2005), pp. 535–541.

[4] A. Fukushima, T. Seki, K. Yakushiji, H. Kubota, H. Imamura, S. Yuasa, and K. Ando, *Spin Dice: A Scalable Truly Random Number Generator Based on Spintronics*, Appl. Phys. Express **7**, 083001 (2014).

[5] W. H. Choi, Y. Lv, J. Kim, A. Deshpande, G. Kang, J.-P. Wang, and C. H. Kim, *A Magnetic Tunnel Junction based True Random Number Generator with Conditional Perturb and Real-Time Output Probability Tracking*, in *Proceedings of the 2014 IEEE International Electron Devices Meeting (IEDM), San Francisco* (IEEE, New York, 2014), pp. 12.5.1–12.5.4.

[6] J. Grollier, D. Querlioz, and M. D. Stiles, *Spintronic Nanodevices for Bioinspired Computing*, Proc. IEEE **104**, 2024 (2016).

[7] J. Roychowdhury (private communication).

[8] For an example of the use of, invertible relations, see R. Canetti and M. Varia, in *Proceedings of the Conference on Theory of Cryptography* (Springer, New York, 2009), pp. 73–90.

[9] B. Behin-Aein, V. Diep, and S. Datta, *A Building Block for Hardware Belief Networks*, Sci. Rep. **6**, 29893 (2016).

[10] Y. Shim, A. Jaiswal, and K. Roy, *Ising Computation based Combinatorial Optimization using Spin-Hall Effect (SHE) Induced Stochastic Magnetization Reversal*, J. Appl. Phys. **121**, 193902 (2017).

[11] B. Sutton, K. Y. Camsari, B. Behin-Aein, and S. Datta, *Intrinsic Optimization using Stochastic Nanomagnets*, Sci. Rep. **7**, 44370 (2017).

[12] J. J. Yang, D. B. Strukov, and D. R. Stewart, *Memristive Devices for Computing*, Nat. Nanotechnol. **8**, 13 (2013).

[13] V. Q. Diep, B. Sutton, B. Behin-Aein, and S. Datta, *Spin Switches for Compact Implementation of Neuron and Synapse*, Appl. Phys. Lett. **104**, 222405 (2014).

[14] A. Sengupta, Y. Shim, and K. Roy, *Proposal for an All-Spin Artificial Neural Network: Emulating Neural and Synaptic Functionalities through Domain Wall Motion in Ferromagnets*, IEEE Trans. Biomed. Circuits Syst. **10**, 1152 (2016).

[15] M. Yamaoka, C. Yoshimura, M. Hayashi, T. Okuyama, H. Aoki, and H. Mizuno, *Ising Computer*, Hitachi Review **65**, 157 (2016).

[16] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, *A Learning Algorithm for Boltzmann Machines*, Cogn. Sci. **9**, 147 (1985).

[17] M. Yamaoka, C. Yoshimura, M. Hayashi, T. Okuyama, H. Aoki, and H. Mizuno, in *IEEE International Solid-State Circuits Conference (ISSCC), 2015* Technical Digest (IEEE, New York, 2015), pp. 1–3.

[18] T. Inagaki, K. Inaba, R. Hamerly, K. Inoue, Y. Yamamoto, and H. Takesue, *Large-Scale Ising Spin Network Based on Degenerate Optical Parametric Iscillators*, Nat. Photonics **10**, 415 (2016).

[19] R. Salakhutdinov, A. Mnih, and G. Hinton, in *Proceedings of the 24th International Conference on Machine Learning* (ACM, Corvalis, 2007), pp. 791–798.

[20] D. J. Amit, *Modeling Brain Function: The World of Attractor Neural Networks* (Cambridge University Press, Cambridge, England, 1992).

[21] D. Du, J. Gu, P. M. Pardalos *et al.*, *Proceedings of the DIMACS Workshop on Satisfiability Problem: Theory and Applications1996*, Vol. 35 (American Mathematical Society, Providence, 1997).

[22] L. Liu, C.-F. Pai, Y. Li, H. W. Tseng, D. C. Ralph, and R. A. Buhrman, *Spin-Torque Switching with the Giant Spin Hall Effect of Tantalum*, Science **336**, 555 (2012).

[23] N. Locatelli, A. Mizrahi, A. Accioly, R. Matsumoto, A. Fukushima, H. Kubota, S. Yuasa, V. Cros, L. G. Pereira, D. Querlioz *et al.*, *Noise-Enhanced Synchronization of Stochastic Magnetic Oscillators*, Phys. Rev. Applied **2**, 034009 (2014).

[24] A. van den Brink, G. Vermijs, A. Solignac, J. Koo, J. T. Kohlhepp, H. J. M. Swagten, and B. Koopmans, *Field-Free Magnetization Reversal by Spin-Hall Effect and Exchange Bias*, Nat. Commun. **7**, 10854 (2016).

[25] Y.-C. Lau, D. Betto, K. Rode, J. M. D. Coey, and P. Stamenov, *Spin-Orbit Torque Switching without an External Field Using Interlayer Exchange Coupling*, Nat. Nanotechnol. **11**, 758 (2016).

[26] A. K. Smith, M. Jamali, Z. Zhao, and J.-P. Wang, *External Field Free Spin Hall Effect Device for Perpendicular Magnetization Reversal Using a Composite Structure with Biasing Layer*, arXiv:1603.09624.

[27] S. Fukami, C. Zhang, S. DuttaGupta, A. Kurenkov, and H. Ohno, *Magnetization Switching by Spin-Orbit Torque in an Antiferromagnet-Ferromagnet Bilayer System*, Nat. Mater. **15**, 535 (2016).

[28] J. T. Heron, J. L. Bosse, Q. He, Y. Gao, M. Trassin, L. Ye, J. D. Clarkson, C. Wang, J. Liu, S. Salahuddin *et al.*, *Deterministic Switching of Ferromagnetism at Room Temperature Using an Electric Field*, Nature (London) **516**, 370 (2014).

[29] S. Manipatruni, D. E. Nikonov, and I. A. Young, *Spin-Orbit Logic with Magnetoelectric Nodes: A Scalable Charge Mediated Nonvolatile Spintronic Logic*, arXiv:1512.05428.

[30] Roger H. Koch, G. Grinstein, G. A. Keefe, Yu Lu, P. L. Trouilloud, W. J. Gallagher, and S. S. P. Parkin, *Thermally Assisted Magnetization Reversal in Submicron-Sized Magnetic Thin Films*, Phys. Rev. Lett. **84**, 5419 (2000).

[31] S. Urazhdin, N. O. Birge, W. P. Pratt, Jr., and J. Bass, *Current-Driven Magnetic Excitations in Permalloy-Based Multilayer Nanopillars*, Phys. Rev. Lett. **91**, 146803 (2003).

[32] I. N. Krivorotov, N. C. Emley, A. G. F. Garcia, J. C. Sankey, S. I. Kiselev, D. C. Ralph, and R. A. Buhrman, *Temperature Dependence of Spin-Transfer-Induced Switching of Nanomagnets*, Phys. Rev. Lett. **93**, 166603 (2004).

[33] A. V. Khvalkovskiy, D. Apalkov, S. Watts, R. Chepulskii, R. S. Beach, A. Ong, X. Tang, A. Driskill-Smith, W. H. Butler, P. B. Visscher *et al.*, *Basic Principles of STT-MRAM Cell Operation in Memory Arrays*, J. Phys. D **46**, 074001 (2013).

[34] R. P. Cowburn, *Property Variation with Shape in Magnetic Nanoelements*, J. Phys. D **33**, R1 (2000).

[35] P. Debashis, R. Faria, K. Y. Camsari, J. Appenzeller, S. Datta, and Z. Chen, *Experimental demonstration of nanomagnet networks as hardware for ising computing*, in *Proceedings of the 2016 IEEE International Electron Devices Meeting (IEDM)* (IEEE, New York, 2016), pp. 34.3.1–34.3.4.

[36] K. Y. Camsari, S. Ganguly, and S. Datta, *Modular Approach to Spintronics*, Sci. Rep. **5**, 10571 (2015).

[37] L. Liu, T. Moriyama, D. C. Ralph, and R. A. Buhrman, *Spin-Torque Ferromagnetic Resonance Induced by the Spin Hall Effect*, Phys. Rev. Lett. **106**, 036601 (2011).

[38] S. Hong, S. Sayed, and S. Datta, *Spin Circuit Representation for the Spin Hall Effect*, IEEE Trans. Nanotechnol. **15**, 225 (2016).

[39] S. Datta, S. Salahuddin, and B. Behin-Aein, *Non-Volatile Spin Switch for Boolean and Non-Boolean Logic*, Appl. Phys. Lett. **101**, 252411 (2012).

[40] W. H. Butler, T. Mewes, C. K. A. Mewes, P. B. Visscher, W. H. Rippard, S. E. Russek, and R. Heindl, *Switching Distributions for Perpendicular Spin-Torque Devices within the Macrospin Approximation*, IEEE Trans. Magn. **48**, 4684 (2012).

[41] A. D. Kent and D. C. Worledge, *A New Spin on Magnetic Memories*, Nat. Nanotechnol. **10**, 187 (2015).

[42] D. Morris, D. Bromberg, J.-G. J. Zhu, and L. Pileggi, in *Proceedings of the 49th Annual Design Automation Conference* (ACM, San Fransisco, 2012), pp. 486–491.

[43] D. Datta, B. Behin-Aein, S. Datta, and S. Salahuddin, *Voltage Asymmetry of Spin-Transfer Torques*, IEEE Trans. Nanotechnol. **11**, 261 (2012).

[44] Predictive Technology Model (PTM), http://ptm.asu.edu/.

[45] J. D. Biamonte, *Nonperturbative k-Body to Two-Body Commuting Conversion Hamiltonians and Embedding Problem Instances into Ising Spins*, Phys. Rev. A **77**, 052331 (2008).

[46] K.-U. Demasius, T. Phung, W. Zhang, B. P. Hughes, S.-H. Yang, A. Kellock, W. Han, A. Pushp, and S. S. P. Parkin, *Enhanced Spin-Orbit Torques by Oxygen Incorporation in Tungsten Films*, Nat. Commun. **7**, 10644 (2016).

[47] C.-F. Pai, L. Liu, Y. Li, H. W. Tseng, D. C. Ralph, and R. A. Buhrman, *Spin Transfer Torque Devices Utilizing the Giant Spin Hall Effect of Tungsten*, Appl. Phys. Lett. **101**, 122404 (2012).

[48] Q. Hao and G. Xiao, *Giant Spin Hall Effect and Switching Induced by Spin-Transfer Torque in a* $\mathrm{W/Co_{40}Fe_{40}B_{20}/MgO}$ *Structure with Perpendicular Magnetic Anisotropy*, Phys. Rev. Applied **3**, 034009 (2015).

[49] T. J. Sejnowski, P. K. Kienker, and G. E. Hinton, *Learning Symmetry Groups with Hidden Units: Beyond the Perceptron*, Physica (Amsterdam) **22D**, 260 (1986).

[50] S. Patarnello and P. Carnevali, *Learning Networks of Neurons with Boolean Logic*, Europhys. Lett. **4**, 503 (1987).

[51] L. Personnaz, I. Guyon, and G. Dreyfus, *Collective Computational Properties of Neural Networks: New Learning Mechanisms*, Phys. Rev. A **34**, 4217 (1986).

[52] S. V. B. Aiyer, M. Niranjan, and F. Fallside, *A Theoretical Investigation into the Performance of the Hopfield Model*, IEEE Trans. Neural Networks **1**, 204 (1990).

[53] H. Suzuki, J.-i. Imura, Y. Horio, and K. Aihara, *Chaotic Boltzmann Machines*, Sci. Rep. **3**, 1610 (2013).

[54] G. E. Hinton, *Boltzmann Machine*, Scholarpedia **2**, 1668 (2007), revision no. 91075.

[55] J. J. Hopfield, *Neural Networks and Physical Systems with Emergent Collective Computational Abilities*, Proc. Natl. Acad. Sci. U.S.A. **79**, 2554 (1982).

[56] B. Sutton, K. Y. Camsari, R. Faria, and S. Datta, *Probabilistic Spin Logic Simulator*, 2017, https://nanohub.org/resources/26057

[57] J. Liu, S. Zhou, H. Zhu, and C.-K. Cheng, in *Proceedings of the 2003 IEEE/ACM International Conference on Computer-Aided Design* (IEEE Computer Society, San Jose, 2003), p. 734.

[58] R. Uma, V. Vijayan, M. Mohanapriya, and S. Paul, *Area, Delay and Power Comparison of Adder Topologies*, Int. J. VLSI Design Commun. Syst. **3**, 153 (2012).

[59] D. E. Knuth and L. T. Pardo, *Analysis of a Simple Factorization Algorithm*, Theor. Comput. Sci. **3**, 321 (1976).

[60] F. L. Traversa and M. Di Ventra, *Polynomial-Time Solution of Prime Factorization and NP-Complete Problems with Digital Memcomputing Machines*, Chaos **27**, 023107 (2017).

[61] M. Di Ventra, F. L. Traversa, and I. V. Ovchinnikov, *Topological Field Theory and Computing with Instantons*, arXiv:1609.03230.

[62] A. Ekert and R. Jozsa, *Quantum Computation and Shor's Factoring Algorithm*, Rev. Mod. Phys. **68**, 733 (1996).

[63] R. P. Feynman, *Simulating Physics with Computers*, Int. J. Theor. Phys. **21**, 467 (1982).