



**HAL**  
open science

## Sequential spin-logic algorithmics for Ising machines

Frédéric Lechenault, Laura Michel, Marco Rogier

► **To cite this version:**

Frédéric Lechenault, Laura Michel, Marco Rogier. Sequential spin-logic algorithmics for Ising machines. 2024. <hal-04801182>

**HAL Id: hal-04801182**

**<https://hal.science/hal-04801182v1>**

Preprint submitted on 25 Nov 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

# Sequential spin-logic algorithmics for Ising machines

Laura Michel<sup>1</sup> , Marco Rogier<sup>1</sup>, and Frédéric Lechenault<sup>1</sup> 

<sup>1</sup>*Laboratoire de Physique de l'École Normale Supérieure,  
ENS, PSL Research University, CNRS, Sorbonne University,  
Université Paris Diderot, Sorbonne Paris Cité, 75005 Paris, France*

(Dated: August 15, 2024)

Ising machines are considered as promising hardware solvers. The idea is to map the solution of a problem to the ground state of an Ising spin glass and find this minimum by performing annealing in an Ising machine that emulates its spin physics. However, there is no general way to formulate arbitrary algorithms within this framework. Here, we present a method to systematically map sequential algorithms to an Ising problem, and illustrate this with  $n$ -bit binary number adders. We show that the resulting Ising systems can be scalable, opening a way for efficient Ising machines for sequential algorithms.

*Introduction.*—Ising systems have been widely considered for computational purposes [1–3]. The main idea is to map the solution of a problem onto the ground state of an Ising spin glass model. Since the system can only relax towards its ground state, it naturally converges to the problem’s solution, regardless of its complexity. Hence, in theory, such an approach allows to solve problems more efficiently than traditional algorithms. However, finding the ground state of an Ising spin glass is itself an NP-hard problem [4]. To address this, a common strategy is to use a physical analog device that mimics the physics of a spin glass model to find its ground state efficiently (or an approximation of it). These analog devices, known as Ising machines, thus constitute promising hardware solvers [5]. In practice, they can be built using a wide range of technologies including optics [6–8], memristors [9], electronics [10–12], and quantum bits [13]. Although still new, Ising machines have been shown to solve hard problems efficiently such as the MaxCut problem [6, 8, 14, 15], 3SAT problems [16] advertisement allocation optimization [17], Diophantine problems [18], or chemical pathway-finding problems [19]. However, to our understanding, there is no general way to explicitly formulate arbitrary algorithms within this framework. Indeed, the mapping from an original problem to an Ising problem is made on a case-by-case basis and can be subtle [20]. In contrast, implementing algorithms via Boolean circuits is straightforward, and many problems can be represented with such a circuit. Hence, an encouraging path is to develop a method to directly embed Boolean circuits into the ground state of an Ising system, thereby offering the possibility to systematically map a wide range of problems to an Ising problem, which could then be solved by an Ising machine.

Such a mapping has been successfully achieved for probabilistic computing approaches, where solvers for binary arithmetic and SAT problems have been implemented using Boolean circuits of stochastic spins [16, 21–23]. However, for lattices of deterministic spins, this mapping has proved less successful. Indeed, previous works have demonstrated the capacity of these lattices

to simulate deterministic logic gates [24] and Boolean circuits [25, 26] in their ground state. However, in [25] no strategy for embedding arbitrary circuits has been proposed. Conversely, while in [26] the authors proposed a systematic embedding strategy, the spin-logic schemes require an exponential hierarchy of couplings to enforce causal information flow, making the practical realization of such systems unfeasible.

This letter addresses this issue. In particular, we present a method to systematically map sequential algorithms to an Ising problem with deterministic spins. Furthermore, we restrict the Ising problem to a quadratic one, connecting only physically adjacent spins, to facilitate its future implementation in an Ising machine [17, 27]. We mainly focus on binary addition algorithms, which are realized by crafting an Ising lattice to simulate their associated Boolean circuits. We ensure the algorithm’s sequential execution by introducing spin coupling amplitudes that decrease with each addition step. Moreover, we show that, at least for binary addition, subtraction, and 2-bit by 2-bit multiplication, the resulting Ising systems can be scalable, with couplings that decay linearly with the system’s size. This makes the practical realization of such solvers possible, opening a path towards the development of efficient Ising machines for sequential algorithms.

*Model.*—We consider an Ising model with external fields composed of  $N$  discrete spins such that the global configuration can be written  $\sigma = \{\pm 1\}^N$ . We place ourselves at zero temperature to ensure that the system is always put in its ground state configuration. Two nearest neighbour spins  $\sigma_i, \sigma_j$  interact via a coupling constant  $J_{i,j}$  and each spin  $\sigma_i$  is subjected to an external magnetic field  $h_i$ . The Hamiltonian of the system for a configuration  $\sigma$  then takes the following form

$$H(\sigma) = - \sum_{\langle i,j \rangle} J_{i,j} \sigma_i \sigma_j - \sum_i h_i \sigma_i, \quad (1)$$

where  $\langle i, j \rangle$  denotes that the sum is run over nearest neighbour pairs of spins  $i$  and  $j$ .

For simplicity, we note the interaction term with the

external magnetic field,  $h_i\sigma_i$ , as an interaction term between a pair of spins  $\sigma_i$  and  $\sigma_j$ . The magnetic term is then re-written  $J_{i,j}\sigma_i\sigma_j$ ,  $\sigma_j$  being a fictive spin with fixed value  $+1$ .

$$H(\sigma) = - \sum_{\langle i,j \rangle} J_{i,j}\sigma_i\sigma_j. \quad (2)$$

In the following, the spin state  $-1$  is chosen to encode the bit 0 and the state  $+1$  the bit 1. Some spins of the model are chosen as inputs and are placed in a specific state by adjusting their associated external fields. As the input spins interact with their nearest neighbours, themselves interacting with theirs, an avalanche process may be activated leading to several spins switching states. Our aim is to make this avalanche process correspond with a binary addition algorithm.

*Embedding logic gates.*—A binary adder can be simulated via a Boolean circuit whose elementary building blocks are logic gates. We show that we can implement the OR and AND gates within an Ising system with three spins and one fictive spin (see Figures 1 a) and b)).

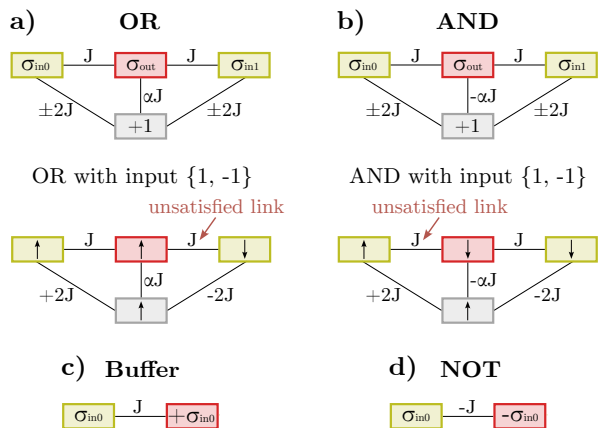


FIG. 1. a) OR and b) AND gate models and operational examples with input  $\{1, -1\}$ . Both gates present one unsatisfied link (an interaction for which the energy is not minimized) for this input. c) Buffer and d) NOT operators. On all graphs, the green color represents the input spins, the red one the output spins and the gray one the fictive spin associated with the applied magnetic fields. The external field factor  $\alpha$  and the coupling constant  $J$  are defined positive.

Note that, throughout this article, the negative Hamiltonian of a system is considered. For instance, in Fig. 1 a), the negative Hamiltonian of the OR gate,  $-H^{OR}$ , is depicted instead of  $H^{OR}$ . In this representation, a ferromagnetic interaction is characterized by a positive coupling constant, while an anti-ferromagnetic interaction is characterized by a negative coupling. Additionally, a positive external field encourages a spin to flip to  $+1$ , whereas a negative field prompts a spin to switch to  $-1$ .

The input spins,  $\sigma_{in0}$  and  $\sigma_{in1}$ , are subjected to external magnetic fields of value  $\pm 2J$ , where  $J$  is the gates coupling constant and is defined positive. The signs of

these external fields are set free and depend on the inputs the user wants to write. For instance, applying a field of  $+2J$  to the input spin  $\sigma_{in0}$  forces the spin to be in  $+1$ . The output spin,  $\sigma_{out}$ , interacts ferromagnetically with both input spins and is subjected to an external magnetic field of value  $\alpha J$  (OR) or  $-\alpha J$  (AND). In the following, we call  $\alpha$  the external field factor. This factor can *a priori* take any value within the interval  $]0, 2[$  to ensure the functionality of the gates (see Supplementary Materials).

To minimize the energy of the gate, the output spin is placed in the configuration that minimizes its number of unsatisfied links (interactions involving reversed spin states in the case of a ferromagnetic interaction or similar spin states in the case of an anti-ferromagnetic interaction). Moreover, this configuration is unique as the output spin interacts with an odd number of spins. The sign of the external field applied on the output spin is then chosen such that it favours the outcome  $+1$  (OR) or  $-1$  (AND), giving the expected behavior for both gates (see Table I). All other logic gates can be constructed from

$\sigma_{in0}$	$\sigma_{in1}$	OR	AND
-1	-1	-1	-1
-1	1	1	-1
1	-1	1	-1
1	1	1	1

TABLE I. Expected output of the OR and AND gates with input  $\{\sigma_{in0}, \sigma_{in1}\}$ .

the OR and AND gates, along with the NOT operator (see Fig. 1 d) and Supplementary Materials).

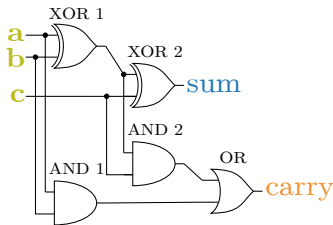
*Assembling two gates.*—These gates can then be integrated into circuits to simulate a particular algorithm. In the case of binary addition, the AND, XOR, and OR gates are typically used to build full adders (see Fig. 2 a)), which are fundamental components that add three digits (two input bits and a carry bit). By connecting multiple full adders in series and propagating the carry from one to the next, large number additions can be performed.

A full adder is composed of several dependent gates where the output of a gate is the input for another gate. We assemble two dependent gates by introducing a ferromagnetic interaction between the output spin of a first gate,  $\sigma_1$ , and the input spin of a second gate,  $\sigma_2$  (see Fig. 1 c)). This ensures that  $\sigma_2$  adopts the state of  $\sigma_1$ , thereby propagating the outcome of the first gate to the second one. However, the total minimal energy of the resulting system might not be the sum of the minimal energies of the two independent gates. It might be energetically beneficial for the whole system to switch the output spin of the first gate,  $\sigma_1$ , in order to switch  $\sigma_2$  and minimize the energy of the second gate. In this case, the working of the first gate is perturbed, damaging the operation of the whole adder.

Please note that the ferromagnetic interaction between two linked gates is chosen sufficiently strong such that the

input spin of the second gate ( $\sigma_2$ ) can not flip without flipping the output spin of the first gate ( $\sigma_1$ ). Hence, the scenario where  $\sigma_2$  flips alone will not be considered.

a) Full adder circuit -  $a+b+c$



b) Adder with two full adders -  $a_2a_1+b_2b_1+c_1$

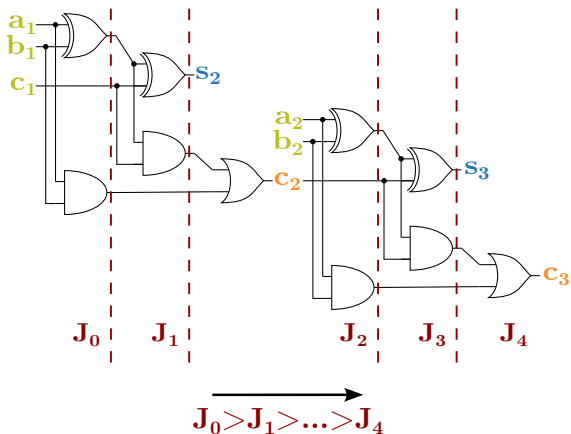


FIG. 2. a) Full adder Boolean circuit using the OR, AND, and XOR gates. b) Adder composed of two full adders. The adder performs the addition  $a_2a_1 + b_2b_1 + c_1$  and outputs the result  $c_3s_3s_2$ .

*Ensuring the sequentiality of the addition algorithm—*

To avoid the reversal of a previous gate spin and therefore guarantee the sequential execution of the addition algorithm, we vary the amplitude of the coupling constants  $J$  for each step of the addition process as shown in Fig. 2 b). All gates associated with step  $k$  share a common coupling amplitude  $J_k$ , whose value differs from the coupling amplitude values of the gates in the previous step  $J_{k-1}$  or the subsequent step  $J_{k+1}$ . In particular, we know that the amplitude of the couplings must decrease with the addition step in order to prioritize the minimization of energy of the first steps gates, ensuring a causal flow of information. The relationship between the coupling amplitudes of step  $k$  and  $k+1$ , respectively  $J_k$  and  $J_{k+1}$ , then takes the following form

$$J_k > \gamma J_{k+1}, \quad (3)$$

where  $\gamma$  is a constant superior or equal to one.

*Scaling relation.*—Interestingly, beyond ensuring sequentiality, this operation considerably simplifies the ground state computation as the global minimal energy of the system can be found by minimizing the energy of each gate independently. However, the sequence  $(J_k)_{k \in \mathbf{N}}$

decays exponentially if  $\gamma > 1$ , making an experimental realization of the system unfeasible. Fortunately, each gate within a full adder has at least one free parameter: its external field factor(s). By adjusting these parameters, it may be possible to ensure that the algorithm executes sequentially while keeping the decay of the sequence  $(J_k)_{k \in \mathbf{N}}$  reasonable. Ideally, this decay would be linear, with  $\gamma = 1$  or  $\gamma \rightarrow 1^+$ .

A full adder is composed of two AND gates (AND 1 and AND 2) and one OR gate (see Fig. 2 a)). The external field factor of these gates is chosen identical and is noted  $\alpha$ . Additionally, a full adder includes two XOR gates (XOR 1 and XOR 2), themselves composed of two OR gates and one AND gate. Each XOR is then associated with three external field factors. By choice, each XOR has its own set of external field factors, which we note  $\{\alpha_1^{(1)}, \alpha_2^{(1)}, \alpha_3^{(1)}\}$  for XOR 1 and  $\{\alpha_1^{(2)}, \alpha_2^{(2)}, \alpha_3^{(2)}\}$  for XOR 2. In the following, our objective is to find a set of parameters  $\{\alpha, \alpha_1^{(1)}, \alpha_2^{(1)}, \alpha_3^{(1)}, \alpha_1^{(2)}, \alpha_2^{(2)}, \alpha_3^{(2)}\}$  such that  $\gamma = 1$  or  $\gamma \rightarrow 1^+$ .

Consider two dependent gates linked via a ferromagnetic interaction between the output spin of the first gate,  $\sigma_1$ , and the input spin of the second gate,  $\sigma_2$ . Imagine that the gates are placed in a spin configuration such that the second gate would minimize its energy by flipping  $\sigma_2$ , hence  $\sigma_1$ . If the flips occur, as the spin  $\sigma_1$  initially chose the unique state that minimizes the number of unsatisfied links in its interacting group, its switch leads to the addition of unsatisfied links in the group, increasing the energy of the first gate. In this case, the minimization of energy of the second gate enters into conflict with the minimization of energy of the first gate and two balancing energy terms are to be considered: the loss of energy of the second gate and the gain of energy of the first one. In general, imposing that the gain of energy due to the addition of unsatisfied links in an interacting group is superior than the absolute loss of energy of the following gates is sufficient to prevent the reversal of an output spin. By imposing this condition for all possible reversal scenarios in a system composed of two full adders, one finds a set of inequalities between the coupling amplitudes and can express the coefficient  $\gamma$  as a function of the parameters  $\{\alpha, \alpha_1^{(1)}, \alpha_2^{(1)}, \alpha_3^{(1)}, \alpha_1^{(2)}, \alpha_2^{(2)}, \alpha_3^{(2)}\}$  (see Supplementary Materials). By optimizing these parameters with the objective that  $\gamma \rightarrow 1^+$ , we find the following scaling relation between the coupling amplitudes.

$$\begin{cases} J_k > (1 + \epsilon)J_{k+1} & \text{if } k \text{ is even,} \\ J_k > J_{k+1} & \text{if } k \text{ is odd,} \end{cases} \quad (4)$$

with  $\epsilon \rightarrow 0^+$  and for the set of parameters  $\{\alpha = 1, \alpha_1^{(1)} = 1, \alpha_2^{(1)} = 1, \alpha_3^{(1)} = 1, \alpha_1^{(2)} = \frac{3}{2} - \epsilon, \alpha_2^{(2)} = \frac{3}{2} - \epsilon, \alpha_3^{(2)} = \frac{1}{2}\}$ .

Consequently, the sequence  $(J_k)_{k \in \mathbf{N}}$  tends towards the following arithmetic progression

$$J_k = J_{k+1} + \lambda, \quad (5)$$

with  $\lambda$  the common difference that can be chosen arbitrarily small.

While the above argument focused on a system of two full adders, it is generalizable to systems comprising  $n$  full adders. The presented framework then offers a way to systematically map sequential algorithms to an Ising problem by embedding their associated Boolean circuits into an Ising system. Furthermore, in the case of binary addition, the resulting adders are scalable, making the practical realization of such solvers feasible. We expect this scalability to extend beyond binary addition to a wide range of algorithms. For instance, scalable binary subtractors and 2-bit by 2-bit multipliers can be constructed without difficulty (see Supplementary Materials).

*Testing a 5-bit binary number adder*—We now show that an adder integrated into an Ising system and satisfying relation (4) accurately executes 5-bit binary number additions. As stated before, the ground state of the adder can be found by minimizing the energy of each gate independently. However, to prove that our theory is correct, we also computed the ground state configuration using a simulated annealing algorithm. This allows a comparison between the theoretical ground state configuration and the configuration found by the algorithm, therefore assessing our theory.

In order for the algorithm to converge, we used  $\epsilon = 0.7$  and  $\lambda = 1$ . Note that we also tested all the possible spin configurations for a full adder with  $\epsilon = 10^{-12}$  and  $\lambda = 10^{-12}$ , for all additions. We found that the minimum energy state always corresponds to the spin configuration predicted by our theory, meaning the configuration that encodes the correct result of the addition (both intermediary and final results).

Fig. 3 shows the minimum energy achieved by the adder at each iteration of the simulated annealing algorithm when executing the additions  $10110+00111$  ( $22+7$ ) and  $11101+10011$  ( $29+19$ ). In both cases, this quantity decreases until it reaches the ground state energy. Once the adder reaches its ground state, the result of the binary additions,  $011101$  ( $= 29$ ) and  $110000$  ( $= 48$ ), can be read onto the system. Beyond the output spins, all other spins in the system align with the states predicted by our theory. This test then seems to validate the accuracy of our theoretical framework.

*Conclusion and outlook*—To summarize, we presented a method to systematically translate sequential algorithms into the minimisation of an Ising Hamiltonian. Using chosen external fields and couplings, we could build functional deterministic logic gates within the ground state of an Ising system. These logic gates could then be integrated into larger circuits, such as  $n$ -bit binary number adder circuits. We guaranteed the sequential execution of the addition algorithm by introducing different step-dependent coupling amplitudes which ensured local causality. Furthermore, the sequence of the coupling am-

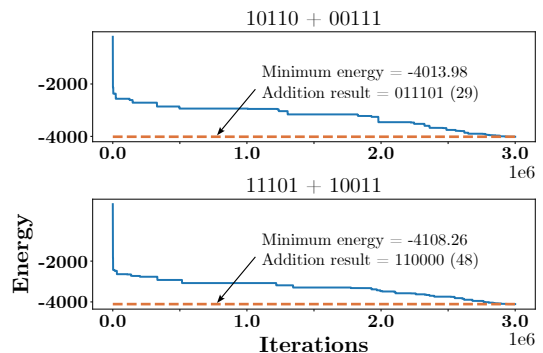


FIG. 3. Minimum energy achieved by the 5-bit binary adder with input  $10110+00111$  or  $11101+10011$  at each iteration of the simulated annealing algorithm. The probability of a spin flip leading to an energy change  $\Delta E$  was given by  $\exp\left\{\frac{-\Delta E}{T(t)}\right\}$  where  $T(k)$  is the temperature at iteration  $k$ . We performed linear annealing with  $T(k+1) = T(k) - \frac{T_0 - T_1}{N}$ ,  $T_0$  being the initial temperature,  $T_1$  the final temperature, and  $N$  the maximum number of iterations. We used  $T_1 = 300$ ,  $T_0 = 10^{-2}$  and  $N = 3 \cdot 10^6$ .

plitudes was shown to tend towards an arithmetic progression with a common difference that can be arbitrarily small, making the resulting adders scalable. We are confident that implementing other sequential algorithms using the presented formalism is feasible without the need for additional elements. If the resulting Ising systems are also scalable, they can actually be built, potentially leading to Ising machines that would run those algorithms more efficiently than traditional hardware solvers. While mapping an Ising problem to an Ising machine can itself be a difficult optimization problem [28–30], we believe that our restriction to quadratic Ising problems with only adjacent spin connections facilitates such mappings. The framework thus provides a pathway for realizing efficient sequential algorithm solvers by exploiting the natural relaxation of Ising systems.

- 
- [1] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, *Science* **220**, 671 (1983).
  - [2] A. Das and B. K. Chakrabarti, *Rev. Mod. Phys.* **80**, 1061 (2008).
  - [3] B. Sutton, K. Y. Camsari, B. Behin-Aein, and S. Datta, *Scientific Reports* **7**, 44370 (2017).
  - [4] F. Barahona, *Journal of Physics A: Mathematical and General* **15**, 3241 (1982).
  - [5] N. Mohseni, P. L. McMahon, and T. Byrnes, *Nature Reviews Physics* **4**, 363 (2022).
  - [6] T. Inagaki, Y. Haribara, K. Igarashi, T. Sonobe, S. Tamate, T. Honjo, *et al.*, *Science* **354**, 603 (2016).
  - [7] D. Pierangeli, G. Marcucci, and C. Conti, *Phys. Rev. Lett.* **122**, 213902 (2019).
  - [8] T. Honjo, T. Sonobe, K. Inaba, T. Inagaki, T. Ikuta, Y. Yamada, *et al.*, *Science Advances* **7**, eabh0952 (2021).

- [9] F. Cai, S. Kumar, T. Van Vaerenbergh, X. Sheng, R. Liu, C. Li, *et al.*, *Nature Electronics* **3**, 409 (2020).
- [10] N. Shukla, A. Parihar, E. Freeman, H. Paik, G. Stone, V. Narayanan, *et al.*, *Scientific Reports* **4**, 4964 (2014).
- [11] P. A. Merolla, J. V. Arthur, R. Alvarez-Icaza, A. S. Cassidy, J. Sawada, F. Akopyan, *et al.*, *Science* **345**, 668 (2014).
- [12] M. Yamaoka, C. Yoshimura, M. Hayashi, T. Okuyama, H. Aoki, and H. Mizuno, *IEEE Journal of Solid-State Circuits* **51**, 303 (2016).
- [13] M. W. Johnson, M. H. Amin, S. Gildert, T. Lanting, F. Hamze, N. Dickson, *et al.*, *Nature* **473**, 194 (2011).
- [14] I. Ahmed, P.-W. Chiu, and C. H. Kim, in *2020 IEEE Symposium on VLSI Circuits* (2020) pp. 1–2.
- [15] I. Ahmed, P.-W. Chiu, W. Moy, and C. H. Kim, *IEEE Journal of Solid-State Circuits* **56**, 2870 (2021).
- [16] N. A. Aadit, A. Grimaldi, M. Carpentieri, L. Theogarajan, J. M. Martinis, G. Finocchio, and K. Y. Camsari, *Nature Electronics* **5**, 460 (2022).
- [17] K. Tanahashi, S. Takayanagi, T. Motohashi, and S. Tanaka, *Journal of the Physical Society of Japan* **88**, 061010 (2019).
- [18] S. A. Abel and L. A. Nutricati, *Fortschritte der Physik* **70**, 2200114 (2022).
- [19] Y. Mizuno and T. Komatsuzaki, *Phys. Rev. Res.* **6**, 013115 (2024).
- [20] A. Lucas, *Front. Phys.* **2**, 5 (2014).
- [21] K. Y. Camsari, R. Faria, B. M. Sutton, and S. Datta, *Phys. Rev. X* **7**, 031014 (2017).
- [22] A. Z. Pervaiz, L. A. Ghantasala, K. Y. Camsari, and S. Datta, *Scientific Reports* **7**, 10994 (2017).
- [23] N. Onizawa, K. Nishino, S. C. Smithson, B. H. Meyer, W. J. Gross, H. Yamagata, *et al.*, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **40**, 655 (2021).
- [24] M. Gu and A. Perales, *Phys. Rev. E* **86**, 011116 (2012).
- [25] J. D. Whitfield, M. Faccin, and J. D. Biamonte, *Europhysics Letters* **99**, 57004 (2012).
- [26] F. Caravelli and C. Nisoli, *New Journal of Physics* **22**, 103052 (2020).
- [27] S. Tanaka, Y. Matsuda, and N. Togawa, in *2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC)* (2020) pp. 659–666.
- [28] J. Cai, W. G. Macready, and A. Roy, arXiv preprint arXiv:1406.2741 (2014).
- [29] N. Dattani and N. Chancellor, arXiv preprint arXiv:1901.07676 (2019).
- [30] T. Shirai, S. Tanaka, and N. Togawa, *IEEE Access* **8**, 210490 (2020).