Received 29 February 2020; revised 11 April 2020 and 29 April 2020; accepted 30 April 2020.
Date of publication 6 May 2020; date of current version 6 July 2020.

Digital Object Identifier 10.1109/JXCDC.2020.2992691

A Spiking Recurrent Neural Network With Phase-Change Memory Neurons and Synapses for the Accelerated Solution of Constraint Satisfaction Problems

GIACOMO PEDRETTI[1] (Member, IEEE), PIERGIULIO MANNOCCI[1], SHAHIN HASHEMKHANI[1], VALERIO MILO[1] (Member, IEEE), OCTAVIAN MELNIC[1], ELISABETTA CHICCA[2] (Member, IEEE), and DANIELE IELMINI[1] (Fellow, IEEE)

[1]Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano and IU.NET, 20133 Milan, Italy
[2]Faculty of Technology and the Center of Cognitive Interaction Technology (CITEC), Bielefeld University, 33615 Bielefeld, Germany
CORRESPONDING AUTHOR: D. IELMINI (daniele.ielmini@polimi.it)

This work was supported in part by the European Research Council (ERC) through the European Union's Horizon 2020 Research and Innovation Programme under Grant 842472, in part by the Ministero degli Affari Esteri e della Cooperazione Internazionale under Grant PGR01011, and in part by the German Research Foundation (DFG) through the Cluster of Excellence Cognitive Interaction Technology "CITEC" (EXC 277), Bielefeld University.

ABSTRACT Data-intensive computing applications, such as object recognition, time series prediction, and optimization tasks, are becoming increasingly important in several fields, including smart mobility, health, and industry. Because of the large amount of data involved in the computation, the conventional von Neumann architecture suffers from excessive latency and energy consumption due to the memory bottleneck. A more efficient approach consists of in-memory computing (IMC), where computational operations are directly carried out within the data. IMC can take advantage of the rich physics of memory devices, such as their ability to store analog values to be used in matrix–vector multiplication (MVM) and their stochasticity that is highly valuable in the frame of optimization and constraint satisfaction problems (CSPs). This article presents a stochastic spiking neuron based on a phase-change memory (PCM) device for the solution of CSPs within a Hopfield recurrent neural network (RNN). In the RNN, the PCM cell is used as the integrating element of a stochastic neuron, supporting the solution of a typical CSP, namely a Sudoku puzzle in hardware. Finally, the ability to solve Sudoku puzzles using RNNs with PCM-based neurons is studied for increasing size of Sudoku puzzles by a compact simulation model, thus supporting our PCM-based RNN for data-intensive computing.

INDEX TERMS Phase change memory (PCM), artificial synapses, hopfield neural network, stochastic process, optimization.

I. INTRODUCTION

OPTIMIZATION problems are among the most intensive computing tasks for several application fields, such as industry, finance, and transport. In general, optimization is carried out by several iterations to identify the global minimum of a certain cost function. In each iteration, a conventional digital system must access the memory to fetch input data and upload the temporary output, which is time and energy consuming. To enable a more efficient optimization, a non-von Neumann architecture can be adopted to eliminate the latency and energy spent for shuttling the data between the memory and the central processing unit (CPU) [1]. An example of non-von Neumann computing architecture is the concept of in-memory computing (IMC) where the computation is executed directly within the memory array. For instance, IMC can efficiently accelerate the typical multiply–accumulate (MAC) operation, which is the foundation for modern digital accelerators for artificial intelligence (AI) and optimization [2]. Emerging memory devices, such as phase-change memory (PCM) [3], [4] and resistive random access memory (RRAM) [5], [6], offer scalable, efficient, and CMOS-compatible solutions to store analog information as the conductance value. Several IMC demonstrators have thus been reported for accelerating neural network training [7], [8], inference [9], image processing [10], and the solution of algebraic problems [11]–[14].

In a constraint satisfaction problem (CSP), the objective is to find a set of states satisfying a collection of constraints. Typical CSPs include Max-SAT, Max-Cut, graph coloring, and the Sudoku puzzle [15]. The latter is indicated as an NP-complete problem in its generic form where the time complexity for solving the problems rapidly increases with the size [16]. CSPs can be implemented by the Hopfield recurrent neural networks (RNNs) where the constraints are mapped with synaptic weights, whereas the electrical stimulation allows minimizing the energy cost function to find the solution of the optimization problem [17], [18]. Note that the solution of a CSP in a Hopfield RNN becomes increasingly difficult when the number of the local minima increases because the network state can be trapped within a local minimum [17]. To circumvent this limitation, the stochastic computational annealing is generally adopted, where the external stimulation is suitably mixed with random noise to help the system escape from local minima [19], [20]. Various solutions have been proposed for practical hardware implementation of computational annealing with CMOS circuits [22]–[26], FPGA [27], quantum computing [28], [29], photonic computing [30], and IMC [31]–[33]. The IMC implementation facilitates two key operations in the computational annealing: 1) the matrix–vector multiplication (MVM) among neuron output signals and synaptic weights, which is accelerated in the crosspoint memory array [2] and 2) the random network stimulation, which can take advantage of the stochastic memory behaviors, such as random telegraph noise (RTN) [34] and $1/f$ noise [35]. While the intrinsic memory noise has been shown to be fruitfully adopted as an entropy source for hardware-based true random number generators (TRNGs) [36], [37], the same approach is not easily applicable to computational annealing, especially where a fine control of the annealing temperature is needed for dynamic cooling [38]. In fact, resistive memory devices suffer from resistance broadening [35], namely, the spread of read noise increases with time, which makes the control of stochasticity less controllable. The adoption of a nonphysical, pseudorandom number generator (PRNG), such as the linear-feedback shift register (LFSR), was previously proposed for providing the entropy source in stochastic annealing [40]. However, a physics-based entropy source, such as the PCM can provide true, tunable stochastic input with a higher quality of the random noise [41]. Tunable stochastic properties of the memory device, such as the stochastic switching [2], [39], [41], [43]–[45], may also be explored to solve CSPs with an IMC approach.

In this article, we propose a Hopfield RNN for computational annealing based on stochastic spiking neurons, in analogy with the biological brain [46]. The PCM device acts as the source of noise for generating random spikes [35]. First, we show an experimental demonstration of a PCM-based stochastic neuron with a tunable output frequency of the generated spikes. After characterizing the integrating neuron element, we implement a Hopfield RNN with PCM synapses [48]. The stochastic RNN is demonstrated for the solution of a 2 × 2 Sudoku puzzle in hardware. Finally, the convergence of the solution for various annealing algorithms and puzzle sizes up to 16 × 16 is studied by simulations to allow for the comparison with other types of hardware Sudoku solvers.

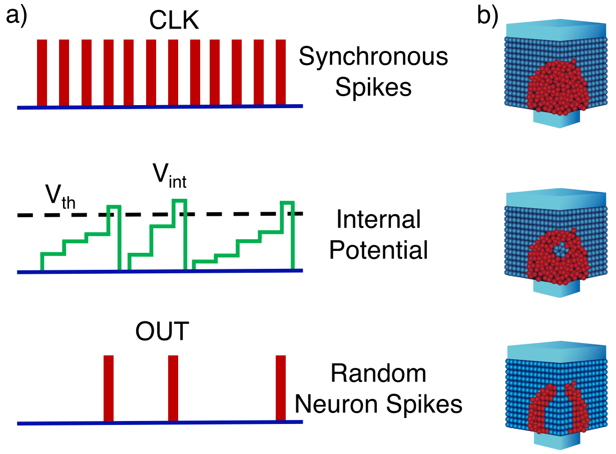

FIGURE 1. (a) Operation principle of the proposed stochastic neuron. A train of synchronous spikes applied to the neuron leads to a stochastic gradual increase of the internal potential V_{int}. When V_{int} reaches a threshold V_{th}, a spike is generated and V_{int} is restored to zero. (b) Sketch of the gradual crystallization process in PCM devices.

II. STOCHASTIC NEURON

A stochastic spiking neuron can act as computational primitive for solving complex CSP problems. Fig. 1(a) shows the operation concept of the proposed stochastic spiking neuron. A deterministic train of spikes of frequency f_{clk} (top) is accumulated by the neuron. The membrane potential V_{int}, representing the input integral, is stored as a suitable state variable of the neuron device, such as the device conductance in the case of the PCM (center). As a threshold potential V_{th} is reached, the neuron releases a spike (bottom), whereas the membrane potential is reset to zero to reinitialize the integration process. Due to the stochastic integration of the memory device [43], where the state variable update is affected by variations, the output spikes are randomly generated in time, thus providing the fundamental basis of the stochastic neuron.

A. STOCHASTIC PCM CRYSTALLIZATION

The neuron integration function was implemented by a PCM device, where each applied pulse causes a partial crystallization in the amorphous volume. Among the two-terminal nonvolatile memories, the PCM is one of the most promising concepts because of many ideal properties, including high switching speed, low current operation, and tunable analog resistance [3], [4]. PCM devices exhibit two resistance states, associated with the crystalline and amorphous phases of the chalcogenide active material, e.g., $Ge_2Sb_2Te_5$, or GST.

To amorphize the GST, a reset voltage pulse is applied above the melting voltage V_m, thus leading to a transition to the liquid phase, followed by rapid freezing into the amorphous phase, corresponding to the high-resistance state (HRS) or reset state. To crystallize the GST, a set voltage pulse is applied usually below V_m and above the threshold voltage V_t for threshold switching [50]. The set pulse causes Joule heating and consequent crystallization within the amorphous volume, thus leading to the low-resistance state (LRS).

The crystallization process can also be executed gradually by applying a train of voltage pulses, each inducing partial crystallization within the amorphous volume. Fig. 1(b) shows a schematic of the gradual crystallization of a PCM device, starting from the HRS (top) corresponding to a complete amorphous phase, to an intermediate state (center) with some material already in the crystalline phase, until the LRS with fully crystalline phase is reached (bottom). Applications of the PCM as analog weight in neural network accelerators have been widely demonstrated [7], [11].

A key issue for analog conductance update is the statistical variability of the PCM device, where the pulse-induced increase in conductance changes from cycle to cycle due to the stochastic nature of the crystallization process. As a result, a PCM can also be used as a stochastic entropy source for a PCM neuron [43]. To study the variation of gradual crystallization dynamics, we characterized a PCM device with a one-transistor-one-resistor (1T1R) structure. Fig. 2(a) shows the measured conductance of a PCM device as a function of the number of pulses of voltage V_A, normalized with the initial conductance G_0. The measurement was repeated 100 times, and each time the device was reinitialized in the HRS by a reset pulse to reach the initial conductance G_0. After an initial incubation phase where the applied pulse causes no change of conductance, G/G_0 steeply increases as a result of the cumulative crystallization within the amorphous volume and eventually saturates to a value G_{sat}/G_0. The onset of crystallization shows the statistical variation in the same device, which can be attributed to the stochastic nucleation and growth processes in the amorphous volume [51]. Fig. 2(b) shows the average conductance change G/G_0 for increasing conductance G_0 of the initial HRS as a function of the number of programming pulses. The initial conductance G_0 impacts on all the parameters of the update characteristics, including the incubation number of pulses, the slope of the G increase, and the G_{sat} value.

To study the stochastic variations of crystallization, Fig. 2(c) and (d) show the mean value $\langle N_C \rangle$ and the deviation $\sigma(N_C)$, respectively, of the number of incoming set pulses N_C to reach a threshold conductance $G_{\mathrm{th}} = G_{\mathrm{sat}}/2$ as a function of the applied V_{set} and the preprogramming V_{reset} pulse to reach the different desired values of G_0. The average number of pulses to crystallization decreases with V_{set} since a higher set voltage induces a more abrupt crystallization. Conversely, the number of pulses to crystallization increases with increasing V_{reset} because of the larger initial amorphous volume that needs to be crystallized. Similarly, the deviation of the number of pulses increases with increasing V_{reset} and decreasing V_{set}, which correctly tracks the behavior of the average number of pulses.

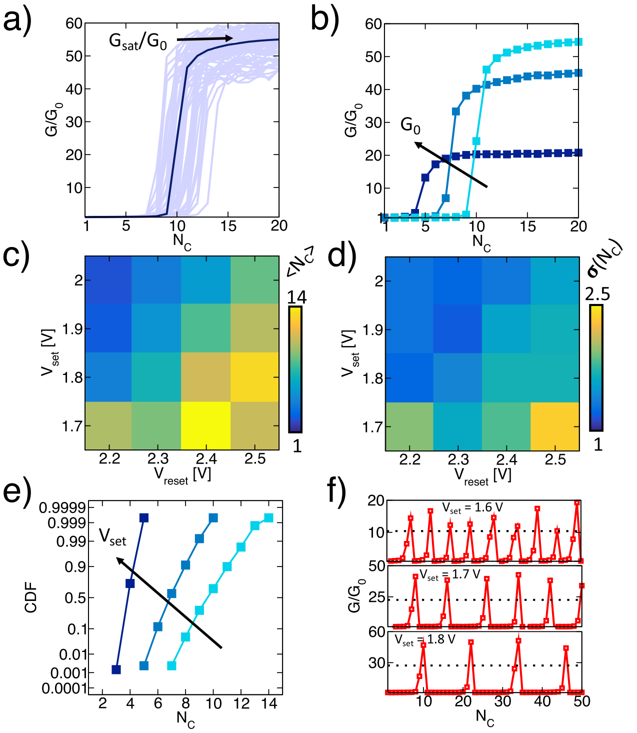

FIGURE 2. (a) Relative PCM conductance G/G_0 as a function of N_C for a given set and reset voltage. Both the individual 100 measurements and their average G/G_0 are shown. (b) Average G/G_0 for increasing G_0, i.e., decreasing initial amorphous volume. As G_0 decreases, the number of pulses in the incubation phase increases. (c) Average number of cycles N_C to reach a given conductance threshold $G_{\mathrm{sat}}/2$ and (d) its standard deviation $\sigma(N_C)$. (e) Cumulative distribution function of N_C to reach the conductance threshold at increasing V_{set}. (f) Measured conductance change G/G_0 as a function of N_C for increasing V_{set} values, namely, 1.6 V (top), 1.7 V (center), and 1.8 V (bottom). The conductance is initialized to G_0 every time the threshold (dashed line) is reached, thus resulting in a stochastic train of spikes.

The statistical analysis is summarized in the cumulative distributions of the number of crystallizing pulses at increasing V_{set} for a fixed $V_{\mathrm{reset}} = 2.4$ V in Fig. 2(e). All distributions show a Gaussian-like behavior. Note that the standard deviation of N_C in Fig. 2(e) controls the statistics of the output spikes of the stochastic neuron in Fig. 1(a) and hence the annealing dynamics in the RNN. Therefore, the ability to tune the average number and spread of N_C in Fig. 2(e) is deeply beneficial for the hardware solution of CSPs. Fig. 2(f) shows the measured G/G_0 as a function of the number of pulses for $V_{\mathrm{set}} = 1.6$ V (top), $V_{\mathrm{set}} = 1.7$ V (center), and $V_{\mathrm{set}} = 1.8$ V (bottom). To reproduce the response of the integrate and fire (I&F) neuron, the PCM device was reset every time G/G_0 exceeded the threshold value, which is indicated as a dashed line. It is possible to observe the variation of the number of pulses between every fire event, which supports the stochastic

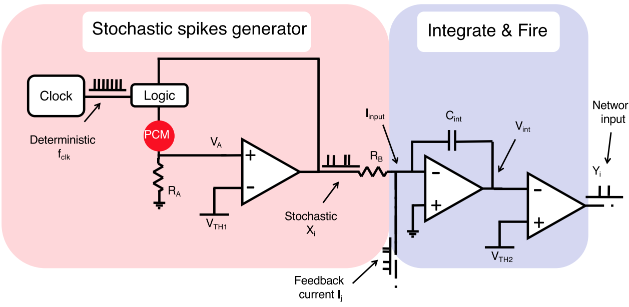

FIGURE 3. Schematic of the neuron circuit with the stochastic spike generator (left) and the I&F unit (right). Input deterministic spikes with clock frequency f_{clk} stimulate the PCM device that is connected in series with a fixed resistance R_A. As the voltage across R_A exceeds the comparator threshold V_{TH1}, a stochastic spike is emitted. The output spikes X_i stimulate a current across resistance R_B, which is summed with the feedback current from the RNN and integrated by the I&F unit. As the internal potential V_{int} exceeds the second comparator threshold V_{TH2}, the I&F unit generates a spike Y_i, which is then propagated within the synaptic network of the RNN.

behavior of the PCM neuron. Note that the PCM device offers the unique physical property of stochastic integration of Fig. 2, which would not be equally feasible in other types of memory device, such as resistive switching memory of magnetic spin-torque memory.

B. STOCHASTIC NEURON CIRCUIT

Fig. 3 shows a schematic of the stochastic neuron circuit that includes two stages: 1) a stochastic spike generator based on a PCM stochastic seed and 2) an I&F output stage. A deterministic train of spikes with clock frequency f_{clk} is applied across the PCM in series with a load resistance R_A, thus acting as a voltage divider. As the PCM conductance increases from the initial value G_0 because of gradual crystallization, the voltage divider output V_A evaluated at the PCM bottom electrode (BE) between every input spike increases. The voltage V_A is compared with the threshold V_{TH1} of a comparator. As V_A reaches V_{TH1}, an output spike is generated and the PCM conductance is restored to G_0 by a feedback signal. The correct voltage applied to the PCM device is synchronized by a control logic circuit. The stochastic voltage spikes X_i of average frequency f_{input} are then converted into a spiking current by the resistor R_B and summed with the feedback column current spikes I_j collected from the RNN. The total current is then integrated on a capacitor C_{int}, thus causing the membrane potential V_{int} to increase spike after spike. As V_{int} reaches the threshold V_{TH2} of the second comparator, a spike is generated and applied to the ith row of the RNN. The neuron response was implemented by a Monte Carlo (MC) model of the stochastic PCM device using the parameters extracted from Fig. 2 and as a stochastic primitive for solving CSP problems. The MC simulations were carried out in a MATLAB environment.

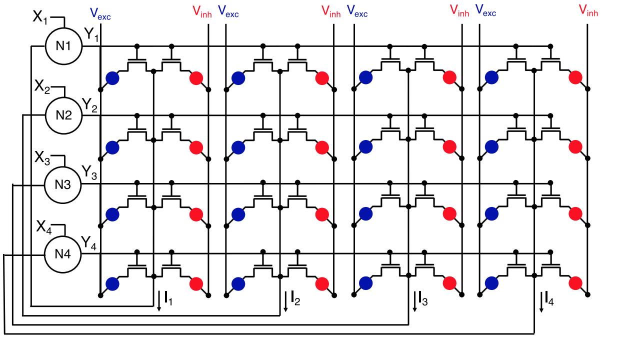

FIGURE 4. Schematic of the Hopfield RNN. Neurons N_i represent the I&F units of the stochastic neuron of Fig. 3. The input X_i is a stochastic signal generated by the stochastic spike generator of Fig. 3. Synapses consist of 1T1R PCM devices organized in excitatory (blue) and inhibitory (red) paths. The neuron output is applied to the gates of the 1T1R in the same row, whereas the synaptic source currents are collected along the same columns and fed back to the neurons. The TEs of the excitatory and inhibitory synapses are biased at V_{exc} and V_{inh}, respectively.

III. HARDWARE RNN

Fig. 4 shows the hardware implementation of a Hopfield RNN with PCM-based synapses and neurons [48], [49]. Each neuron N_i represents the I&F unit of Fig. 3, while the stochastic unit to generate stimulating signal X_i is not shown. The input stimulation to every neuron N_i is thus the stochastic signal X_i generated by the stochastic spike generator block of Fig. 3. Each synaptic unit consists of two PCM devices with 1T1R structure, acting as the excitatory synapse and the inhibitory synapse, respectively. In each synaptic element, the gate terminals of the excitatory and inhibitory synapses are tied together, as well as shared with all other synaptic elements along the same row in the RNN. The source terminals are also connected and shared among all the synaptic elements in the same column of the RNN. Finally, the top electrodes (TEs) of the excitatory and inhibitory synapses are all biased to a positive read voltage V_{exc} and a negative read voltage V_{inh}, respectively, to induce the corresponding column currents. As a result, the overall synaptic weight G_{ij} can be obtained from the difference between the excitatory conductance G_{ij}^{+} and the inhibitory conductance G_{ij}^{-}, according to $G_{ij} = G_{ij}^{+} - G_{ij}^{-}$. The neuron output signal Y_i controls the synaptic gates along the row, whereas the source currents along each column are collected and applied back to the neurons for integration. For instance, N_1 controls all gates in the first row of the RNN, while the synaptic currents in the first column are all collected by Kirchhoff's law, forming the internal signal I_1, which is applied back to N_1. The synaptic current I_j of column j induced by the output spike voltage V_i of neuron N_i is given by $I_j = \sum_i G_{ij} V_i$, thus accelerating the physical MVM within the RNN by IMC. Note that, according to the Hopfield topology of the RNN, synapses in all diagonal positions are omitted to prevent self-excitation/inhibition in any neuron.

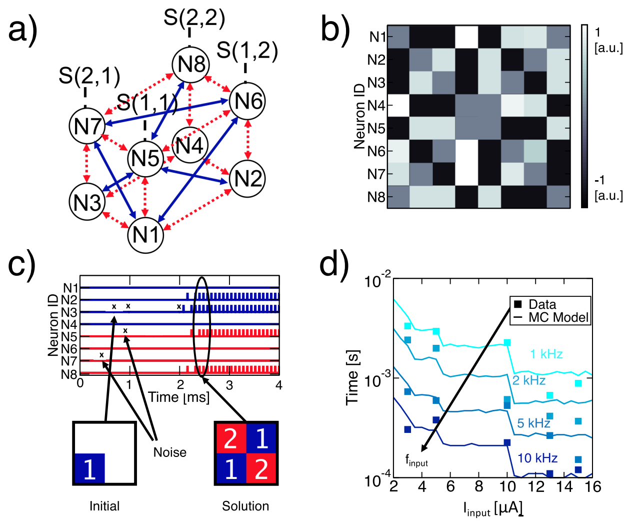

FIGURE 5. **(a) Schematic of the synaptic connectivity to solve a 2 x 2 Sudoku puzzle, with inhibitory connections (red dashed arrows) and excitatory connections (blue solid arrows). A 2 x 2 Sudoku solver needs $N^3 = 8$ neurons and $N^6 - N^3 = 56$ synapses connecting every neuron to each other (without self-connection). (b) Map of synaptic conductance for a 2 x 2 Sudoku programmed in a PCM array. (c) Experimental solution of a 2 x 2 Sudoku, including spikes X_i of the external stimulation, corresponding to the initial condition of number "2" in position (2,1), and spikes Y_i, reflecting the spiking activity of each neuron in the RNN. The solution is achieved after about 2 ms of stimulation. (d) Experiments (squares) and MC simulations (lines) of the solution time for a 2 x 2 Sudoku puzzle as a function of the stimulation amplitude I_{input} and average frequency f_{input}.**

IV. HARDWARE SOLUTION OF A SUDOKU PUZZLE

To implement a certain problem with the RNN, the constraints need to be correctly mapped in the synaptic weights, i.e., the conductance values G_{ij}. Considering a Sudoku puzzle of size N, the constraints can be mapped in N layers of $N \times N$ neurons, where each neuron corresponds to a certain number in a certain position of the puzzle (e.g., number "1" in position $S(1, 1)$). Each layer corresponds to a possible number in the puzzle, e.g., "1" or "2" for the 2 × 2 Sudoku. The neurons can thus be rearranged in an $N \times N \times N$ matrix [23], where the entry corresponding to "1" indicates a firing neuron and the entry corresponding to "0" indicates a silent neuron.

Fig. 5(a) shows the constraints for a 2 × 2 Sudoku problem, represented by two layers of four neurons each. Every neuron represents the neuron circuit of Fig. 3. Solid lines indicate excitatory connections, where a number in a certain position is exciting the same number in a different row/column or the other number in the same row/column. Dashed lines instead indicate inhibitory connections, where a number inhibits the same number in the same row/column or the other number in the same position. In larger Sudoku puzzles, there are also excitatory connections from any neuron to any possible other neuron that does not violate the constraints. For example, in a regular size Sudoku (with $N = 9$), the number "1" will excite any neuron on the same row/column with the numbers "2"–"9."

Fig. 5(b) shows the conductance map for a 2 × 2 Sudoku, which was implemented in two 8 × 8 PCM arrays of Fig. 4: one for excitatory and one for inhibitory synapses. In this RNN, the neuron spikes are applied to its corresponding row, while the currents are collected on the columns and fed back according to the schematic in Fig. 4.

A. EXPERIMENTAL SOLUTION OF A SUDOKU PUZZLE

We carried out experiments for the solution of a 2 × 2 Sudoku puzzle with the stochastic spiking RNN. The 2 × 2 Sudoku solution can only contain numbers 1 and 2, each appearing only once in each row/column, i.e., only solutions (1,2;2,1) and (2,1;1,2) are possible. Fig. 5(c) shows the measured train of spikes X_i and Y_i, namely, the stochastic stimulation and the neuron spiking output, respectively, in Fig. 4. An initial guess $S(2, 1) = 1$ is given as external stimulation, corresponding to neuron N_3 being externally stimulated by a stochastic spiking train X_3 at a relatively high average frequency, whereas all other neurons are only subject to random spikes at lower average frequency. Random spiking in nonstimulated neurons is necessary to prevent trapping in a local minimum of the cost function in the RNN. Note that a stochastic implementation is not necessary to solve a 2 × 2 Sudoku; however, the simplicity of the Sudoku puzzle allows to clearly illustrate the solution algorithm and the spiking signals in Fig. 5(c). The stochastic spikes were generated with the MC model by assuming a stimulation of the PCM with a deterministic train of spikes, where f_{clk} and $f_{clk}/10$ were used to generate the stochastic stimulation of average frequency f_{input} and $f_{input}/10$ and then uploaded on a microcontroller (μC). The latter was programmed to serve as I&F neuron, with the output connected to the gates of 1T1R PCM synapses. The synaptic currents were collected, converted into voltage signals by a transimpedance amplifier (TIA), digitalized with an analog-to-digital converter (ADC) and fed back into the μC. The system was temporized by a clock with frequency $f_{clk} = 10$ kHz, which also limits the maximum f_{input}. The experimental results show that after about 2 ms, neurons N_2, N_3, N_5, and N_8 start to regularly fire at high frequency, whereas all other neurons remain silent. This corresponds to a stable attractor [48] of the RNN and to the minimum of the cost function, thus yielding the hardware solution of the Sudoku puzzle. The configuration of spiking neurons was sustained even after the external stimuli have been removed, thus indicating the stability of the attractor state.

The solution can be easily accelerated by increasing the clock frequency and the stimulating currents. This is shown in Fig. 5(d), which reports the computing time to solve Sudoku as a function of the input current I_{input} of the stimulating stochastic spikes X_i for increasing spiking frequency f_{input}. Data points represent the average over five experiments conducted on our RNN. The solution becomes faster as I_{input} and f_{input} increase since the activated neurons generate ran-

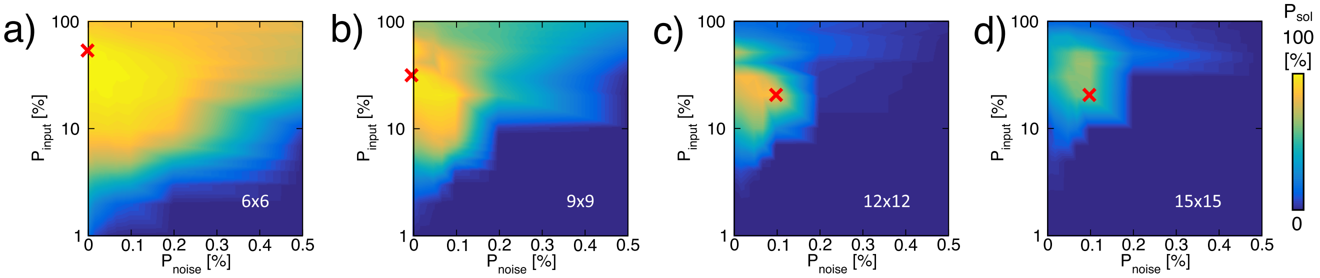

FIGURE 6. Probability P_{sol} to solve Sudoku puzzles for increasing size. (a) 6 x 6, (b) 9 x 9, (c) 12 x 12, and (d) 15 x 15 as a function of the probability P_{input} of generating an input spike and the probability P_{noise} of generating a noise spike. The maximum P_{sol} is marked, indicating that more stochasticity is needed to solve the problem at increasing N.

dom spikes at higher average frequency. It should also be noted that the solution can be further accelerated by optimizing the neuron threshold, which was set to $V_{th} = 1$ V in the experiment of Fig. 5(c), considering an equivalent capacitor of $C_{int} = 100$ pF. Simulation results from an MC model of the network, including PCM variability in neurons and synapses, are also shown in Fig. 5(d). The simulation results clearly show a staircase behavior of the computing time, where the step change corresponds to I_{input} being a submultiple of $I_{th} = V_{th} \cdot C_{int} \cdot f_{clk}$, and steps are in fact clearly visible in correspondence of $I_{input} = 10\ \mu$A or $I_{input} = 5\ \mu$A.

V. TEMPERATURE OPTIMIZATION

With the developed MC model, we simulated various Sudoku problems with increasing size from 4 to 16, to study the RNN performance and the correct tuning of the random spikes, which can be viewed as an equivalent temperature in the simulated annealing process to reach the global minimum of the cost function. To properly solve the puzzles, we assumed an RNN with N^3 neurons and $N^6 - N^3$ PCM synapses encoding all constraints of the Sudoku. We then ran MC simulations to evaluate the success probability P_{sol}, namely the probability of reaching the right solution, for a fixed iteration number and variable input and noise frequency to study the optimal tuning of the stochastic neurons. Fig. 6 shows the calculated P_{sol} for increasing size, namely, 6 × 6, 9 × 9, 12 × 12, and 15 × 15. P_{sol} is shown as a function of the probability P_{input} of generating an input spike, namely, the ratio between the number of stimulating stochastic spikes X_i and the number of deterministic spikes of frequency f_{clk} in Fig. 3, and the probability P_{noise} of generating a noise spike, which is defined similar to P_{input} but referred to random noise spikes. Each simulation was run for 1000 cycles and was repeated 100 times. The position of the maximum P_{sol} moves to lower P_{input} and higher P_{noise} for increasing N, thus indicating an increasing need for stochasticity for increasing Sudoku size. This can be explained by the number of local minima increasing with N, thus resulting in a higher noise contribution, hence temperature, to prevent trapping within a local minimum. On the other hand, an excessive temperature may instead lead to an unstable result, where the RNN can escape also from the global minimum. Note that this method combines the stochastic spike timing and the stochastic PCM conductance variations as entropy sources of the stochastic annealing, whereas previous works only considered stochastic conductance variations [31]–[33]. The simulation results suggest that a tunable source of stochastic spikes is essential for an efficient solution of CSPs in hardware.

VI. EFFICIENCY AND SCALING

To study the impact of Sudoku size on CSP complexity, we ran MC simulations to evaluate the error probability $P_{err} = 1 - P_{sol}$ for increasing N. Fig. 7(a) shows the calculated error probability P_{err} as a function of the number of computing cycles for increasing N between 4 and 16. The error probability increases with N for a given computational cycle. Conversely, the computing speed to solve the Sudoku problem with sufficiently low P_{err} decreases for increasing N. While the system becomes more unreliable for bigger problems, computation can be parallelized on more than one memory array to enhance the probability of reaching a correct solution [33].

The algorithm can also be improved to reduce P_{err} and accelerate the annealing process by properly considering conflicts among constraints at the hardware level. For instance, Fig. 7(b) shows possible conflicts within the first row of a 4 × 4 Sudoku. An initial condition, namely $S(1, 1) = 1$, promotes with excitatory synapses all the other numbers on the same row, namely, $S(1, j) = [2, 3, 4]$, for $j \neq 1$. At the same time, a random noise spike could activate the neuron coding for digit 2 on the same cell, thus conflicting with the initial condition and leading to escape from the correct global minimum [18]. To avoid this, the neurons coding for digits directly conflicting with initial condition should be inhibited from firing. This can be achieved by properly transforming the initial condition such that an excitatory stimulation is provided to neurons coding for givens, whereas an inhibitory stimulation is provided for neurons coding for digits conflicting with the givens following the Sudoku constraints.

To this purpose, we propose a double-layer (DL) network as shown in Fig. 7(c), where a feedforward layer is added as an input layer to the RNN for filtering the input conditions and inhibit wrong stimulations. Noise is only injected in the second layer, while the inhibitions given by the input

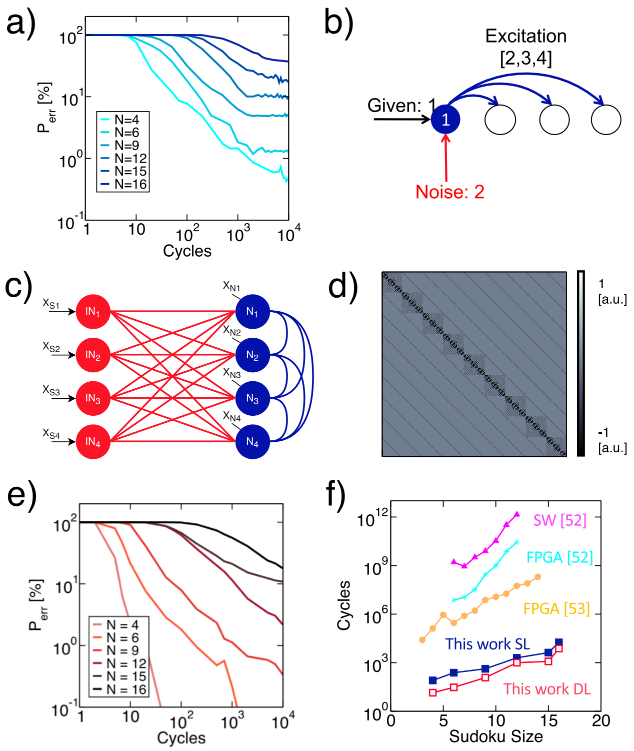

FIGURE 7. (a) Error probability P_{err} as a function of the number of cycles for increasing Sudoku size. (b) Schematic of the possible conflict while solving a Sudoku puzzle, where neurons can be excited and inhibited at the same time by spiking neurons. (c) Schematic of the DL network to prevent conflicting excitation/inhibition and improve the convergence. (d) Matrix of the synaptic weights of the feedforward input layer. (e) P_{err} as a function of the number of cycles for increasing Sudoku size for the DL network. (f) Performance of the Sudoku solver, namely number of iteration cycles for the solution as a function of size, for the SL RNN and the DL RNN, compared with other solvers from the literature.

layer also control the annealing temperature by acting as a cooling effect when the correct solution is reached. In fact, if noise stimulates a neuron N_i that is in contrast with the input condition, the first layer will inject a negative current to N_i to prevent its activation. Fig. 7(d) shows the synaptic weights of the input layer for a 9 × 9 Sudoku indicating all the inhibitions to the recurrent layer.

Fig. 7(e) shows the error probability P_{err} as a function of the number of cycles for increasing size of the Sudoku size by adopting the DL network of Fig. 7(c). Fig. 7(f) summarizes the computing speed, evaluated as the number of cycles to reach $P_{err} = 1\%$, for the single-layer (SL) RNN and the DL network, indicating that the computing speed is clearly improved by the DL network. The performance of the RNN is compared with state-of-the-art systems for solving Sudoku with FPGA implementations [52], [53] and software approaches [52]. The results indicate that the IMC approach allows accelerating the solution of CSP by about four orders of magnitude compared with FPGA and seven orders of magnitude compared with software-based solvers. This is due to the compact implementation of stochastic spikes generator and the low-latency MAC operation of IMC compared with other techniques [54]. Moreover, the novel DL network further improves the performance of the CSP solver and takes full advantage of the compact MAC core. Our implementation shows a reduced number of cycles to solution also compared with state-of-the-art analog neuromorphic processor [26], which takes about 50 cycles to solve a 4 × 4 Sudoku, compared with just 14 cycles of our DL network. These results support the use of the PCM technology for computational annealing techniques.

VII. CONCLUSION

We have developed a brain-inspired spiking RNN for solving CSP problems with stochastic PCM neurons and PCM synapses. First, the stochasticity behavior of the PCM device was experimentally studied during gradual crystallization. Then, the PCM-based stochastic neuron was implemented in an RNN for solving Sudoku puzzles that were experimentally validated on a small scale (2 × 2). An MC model was developed to describe the network stochastic behavior and predict multiple experimental results as a function of stimulating current and frequency. The model was then used to scale the system and study the error probability as a function of the problem size. Finally, a DL spiking neural network was designed to further reduce the error probability. The results demonstrate the superior performance of our system compared with the state-of-the-art implementations, confirming IMC as a promising approach to accelerate the hardware solution of CSPs.

REFERENCES

[1] W. A. Wulf and S. A. McKee, "Hitting the memory wall: Implications of the obvious," *ACM SIGARCH Comput. Archit. News*, vol. 23, no. 1, pp. 20–24, Mar. 1995, doi: 10.1145/216585.216588.

[2] D. Ielmini and H.-S. P. Wong, "In-memory computing with resistive switching devices," *Nature Electron.*, vol. 1, pp. 333–343, Jun. 2018, doi: 10.1038/s41928-018-0092-2.

[3] S. Raoux, W. Wełnic, and D. Ielmini, "Phase change materials and their application to nonvolatile memories," *Chem. Rev.*, vol. 110, no. 1, pp. 240–267, Jan. 2010, doi: 10.1021/cr900040x.

[4] G. W. Burr *et al.*, "Phase change memory technology," *J. Vac. Sci. Technol. B, Nanotechnol. Microelectronics: Mater., Process., Meas., Phenomena*, vol. 28, no. 2, pp. 223–262, Mar. 2010, doi: 10.1116/1.3301579.

[5] H.-S. P. Wong *et al.*, "Metal–oxide RRAM," *Proc. IEEE*, vol. 100, no. 6, pp. 1951–1970, Jun. 2012, doi: 10.1109/JPROC.2012.2190369.

[6] D. Ielmini, "Resistive switching memories based on metal oxides: Mechanisms, reliability and scaling," *Semicond. Sci. Technol.*, vol. 31, no. 6, Jun. 2016, Art. no. 063002, doi: 10.1088/0268-1242/31/6/063002.

[7] S. Ambrogio *et al.*, "Equivalent-accuracy accelerated neural-network training using analogue memory," *Nature*, vol. 558, no. 7708, pp. 60–67, Jun. 2018, doi: 10.1038/s41586-018-0180-5.

[8] C. Li *et al.*, "Efficient and self-adaptive *in-situ* learning in multilayer memristor neural networks," *Nature Commun.*, vol. 9, no. 1, Dec. 2018, doi: 10.1038/s41467-018-04484-2.

[9] V. Milo *et al.*, "Multilevel HfO_2-based RRAM devices for low-power neuromorphic networks," *APL Mater.*, vol. 7, Apr. 2019, Art. no. 081120, doi: 10.1063/1.5108650.

[10] C. Li *et al.*, "Analogue signal and image processing with large memristor crossbars," *Nature Electron.*, vol. 1, no. 1, pp. 52–59, Jan. 2018, doi: 10.1038/s41928-017-0002-z.

[11] M. Le Gallo *et al.*, "Mixed-precision in-memory computing," *Nature Electron.*, vol. 1, no. 4, pp. 246–253, Apr. 2018, doi: 10.1038/s41928-018-0054-8.

[12] M. A. Zidan *et al.*, "A general memristor-based partial differential equation solver," *Nature Electron.*, vol. 1, no. 7, pp. 411–420, Jul. 2018, doi: 10.1038/s41928-018-0100-6.

[13] Z. Sun, G. Pedretti, E. Ambrosi, A. Bricalli, W. Wang, and D. Ielmini, "Solving matrix equations in one step with cross-point resistive arrays," *Proc. Nat. Acad. Sci. USA*, vol. 116, no. 10, pp. 4123–4128, Mar. 2019, doi: 10.1073/pnas.1815682116.

[14] Z. Sun, G. Pedretti, A. Bricalli, and D. Ielmini, "One-step regression and classification with cross-point resistive memory arrays," *Sci. Adv.*, vol. 6, no. 5, Jan. 2020, Art. no. eaay2378, doi: 10.1126/sciadv.aay2378.

[15] M. R. Garey, "A guide to the theory of NP-completeness," in *Computers and Intractability*, New York, NY, USA: W. H. Freeman Company, 1979.

[16] H. Simonis, "Sudoku as a constraint problem," in *Proc. 4th Int. Works. Modelling Reformulating Constraint Satisfaction Problems*, B. Hnich, P. Prosser, and B. Smith, Eds., 2005, pp. 13–27.

[17] J. J. Hopfield, "Searching for memories, sudoku, implicit check bits, and the iterative use of not-always-correct rapid neural computation," *Neural Comput.*, vol. 20, no. 5, pp. 1119–1164, May 2008, doi: 10.1162/neco.2007.09-06-345.

[18] S. Habenschuss, Z. Jonke, and W. Maass, "Stochastic computations in cortical microcircuit models," *PLoS Comput. Biol.*, vol. 9, pp. 1–28, Nov. 2013, doi: 10.1371/journal.pcbi.1003311.

[19] E. H. L. Aarts and J. H. M. Korst, *Simulated Annealing and Boltzmann Machines*. Hoboken, NJ, USA: Wiley, 1988.

[20] V. Pavlovic, D. Schonfeld and G. Friedman, "Enhancement of Hopfield neural networks using stochastic noise processes," in *Proc. Neural Netw. Signal Process. XI: IEEE Signal Process. Soc. Workshop*, 2001, pp. 173–182, doi: 10.1109/NNSP.2001.943122.

[21] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, May 1983, doi: 10.1126/science.220.4598.671.

[22] G. A. Fonseca Guerra and S. B. Furber, "Using stochastic spiking neural networks on SpiNNaker to solve constraint satisfaction problems," *Frontiers Neurosci.*, vol. 11, p. 714, Dec. 2017, doi: 10.3389/fnins.2017.00714.

[23] J. Binas, G. Indiveri, and M. Pfeiffer, "Spiking analog VLSI neuron assemblies as constraint satisfaction problem solvers," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Montreal, QC, Canada, May 2016, pp. 2094–2097, doi: 10.1109/ISCAS.2016.7538992.

[24] H. Mostafa, L. K. Müller, and G. Indiveri, "An event-based architecture for solving constraint satisfaction problems," *Nature Commun.*, vol. 6, no. 1, Dec. 2015, doi: 10.1038/ncomms9941.

[25] T. Takemoto, M. Hayashi, C. Yoshimura, and M. Yamaoka, "2.6 A 2×30 k-spin multichip scalable annealing processor based on a processing-in-memory approach for solving large-scale combinatorial optimization problems," in *IEEE ISSCC Dig. Tech. Papers*, San Francisco, CA, USA, 2019, pp. 52–54, doi: 10.1109/ISSCC.2019.8662517.

[26] D. Liang and G. Indiveri, "A neuromorphic computational primitive for robust context-dependent decision making and context-dependent stochastic computation," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 66, no. 5, pp. 843–847, May 2019, doi: 10.1109/TCSII.2019.2907848.

[27] F. L. Traversa, C. Ramella, F. Bonani, and M. Di Ventra, "Memcomputing NP-complete problems in polynomial time using polynomial resources and collective states," *Sci. Adv.*, vol. 1, no. 6, 2015, Art. no. e1500031, doi: 10.1126/sciadv.1500031.

[28] S. Boixo *et al.*, "Evidence for quantum annealing with more than one hundred qubits," *Nature Phys.*, vol. 10, no. 3, pp. 218–224, Mar. 2014, doi: 10.1038/nphys2900.

[29] V. S. Denchev *et al.*, "What is the computational value of finite-range tunneling?" *Phys. Rev. X*, vol. 6, no. 3, Aug. 2016, doi: 10.1103/PhysRevX.6.031015.

[30] R. Hamerly *et al.*, "Experimental investigation of performance differences between coherent ising machines and a quantum annealer," *Sci. Adv.*, vol. 5, no. 5, May 2019, Art. no. eaau0823, doi: 10.1126/sciadv.aau0823.

[31] J. H. Shin, Y. J. Jeong, M. A. Zidan, Q. Wang, and W. D. Lu, "Hardware acceleration of simulated annealing of spin glass by RRAM crossbar array," in *IEDM Tech. Dig.*, San Francisco, CA, USA, Dec. 2018, pp. 3.3.1–3.3.4, doi: 10.1109/IEDM.2018.8614698.

[32] M. R. Mahmoodi, "An analog neuro-optimizer with adaptable annealing based on 64×64 0T1R crossbar circuit," in *IEDM Tech. Dig.*, San Francisco, CA, USA, Dec. 2019, p. 14, doi: 10.1109/IEDM19573.2019.8993442.

[33] F. Cai *et al.*, "Harnessing intrinsic noise in memristor hopfield neural networks for combinatorial optimization," 2019, *arXiv:1903.11194*. [Online]. Available: http://arxiv.org/abs/1903.11194

[34] S. Ambrogio *et al.*, "Statistical fluctuations in HfO$_x$ resistive-switching memory: Part II—Random telegraph noise," *IEEE Trans. Electron Devices*, vol. 61, no. 8, pp. 2920–2927, Aug. 2014, doi: 10.1109/TED.2014.2330202.

[35] S. Ambrogio, S. Balatti, V. McCaffrey, D. Wang, and D. Ielmini, "Noise-induced resistance broadening in resistive switching memory—Part I: Intrinsic cell behavior," *IEEE Trans. Electron Devices*, vol. 62, no. 11, pp. 3805–3811, Nov. 2015, doi: 10.1109/TED.2015.2475598.

[36] C.-Y. Huang, W. C. Shen, Y.-H. Tseng, Y.-C. King, and C.-J. Lin, "A contact-resistive random-access-memory-based true random number generator," *IEEE Electron Device Lett.*, vol. 33, no. 8, pp. 1108–1110, Aug. 2012, doi: 10.1109/LED.2012.2199734.

[37] Z. Wei *et al.*, "True random number generator using current difference based on a fractional stochastic model in 40-nm embedded ReRAM," in *IEDM Tech. Dig.*, Dec. 2016, pp. 4–8, doi: 10.1109/IEDM.2016.7838349.

[38] Y. Nourani and B. Andresen, "A comparison of simulated annealing cooling strategies," *J. Phys. A: Math. Gen.*, vol. 31, no. 41, pp. 8373–8385, Oct. 1998, doi: 10.1088/0305-4470/31/41/011.

[39] S. Ambrogio *et al.*, "Statistical fluctuations in HfO$_x$ resistive-switching memory: Part I—Set/Reset variability," *IEEE Trans. Electron Devices*, vol. 61, no. 8, pp. 2912-2919, Aug. 2014, doi: 10.1109/TED.2014.2330200.

[40] G. Cauwenberghs, "An analog VLSI recurrent neural network learning a continuous-time trajectory," *IEEE Trans. Neural Netw.* vol. 7, no. 2, pp. 346–361, Mar. 1996, doi: 10.1109/72.485671.

[41] R. Carboni and D. Ielmini, "Stochastic memory devices for security and computing," *Adv. Electron. Mater.*, vol. 5, no. 9, Sep. 2019, Art. no. 1900198, doi: 10.1002/aelm.201900198.

[42] S. Kumar, J. P. Strachan, and R. S. Williams, "Chaotic dynamics in nanoscale NbO$_2$ Mott memristors for analogue computing," *Nature*, vol. 548, pp. 318–321, Aug. 2017, doi: 10.1038/nature23307.

[43] T. Tuma, A. Pantazi, M. Le Gallo, A. Sebastian, and E. Eleftheriou, "Stochastic phase-change neurons," *Nature Nanotechnol.*, vol. 11, no. 8, pp. 693–699, Aug. 2016, doi: 10.1038/nnano.2016.70.

[44] A. Mizrahi *et al.*, "Neural-like computing with populations of superparamagnetic basis functions," *Nature Commun.*, vol. 9, no. 1, p. 1533, Dec. 2018, doi: 10.1038/s41467-018-03963-w.

[45] W. A. Borders, A. Z. Pervaiz, S. Fukami, K. Y. Camsari, H. Ohno, and S. Datta, "Integer factorization using stochastic magnetic tunnel junctions," *Nature*, vol. 573, no. 7774, pp. 390–393, Sep. 2019, doi: 10.1038/s41586-019-1557-9.

[46] W. Maass, "Noise as a resource for computation and learning in networks of spiking neurons," *Proc. IEEE*, vol. 102, no. 5, pp. 860–880, May 2014, doi: 10.1109/JPROC.2014.2310593.

[47] A. Redaelli *et al.*, "Impact of the current density increase on reliability in scaled BJT-selected PCM for high-density applications," in *Proc. IEEE Int. Rel. Phys. Symp.*, Anaheim, CA, USA, May 2010, pp. 615–619, doi: 10.1109/IRPS.2010.5488760.

[48] V. Milo, D. Ielmini, and E. Chicca, "Attractor networks and associative memories with STDP learning in RRAM synapses," in *IEDM Tech. Dig.*, San Francisco, CA, USA, Dec. 2017, pp. 11.2.1–11.2.4, doi: 10.1109/IEDM.2017.8268369.

[49] G. Pedretti, V. Milo, S. Hashemkhani, et.al., "A spiking recurrent neural network with phase change memory synapses for decision making," presented at the ISCAS, Oct. 2020.

[50] D. Ielmini, "Threshold switching mechanism by high-field energy gain in the hopping transport of chalcogenide glasses," *Phys. Rev. B, Condens. Matter*, vol. 78, no. 3, Jul. 2008, Art. no. 035308, doi: 10.1103/PhysRevB.78.035308.

[51] U. Russo, D. Ielmini, A. Redaelli, and A. L. Lacaita, "Intrinsic data retention in nanoscaled phase-change memories—Part I: Monte Carlo model for crystallization and percolation," *IEEE Trans. Electron Devices*, vol. 53, no. 12, pp. 3032–3039, Dec. 2006, doi: 10.1109/TED.2006.885527.

[52] P. Malakonakis, M. Smerdis, E. Sotiriades, and A. Dollas, "An FPGA-based Sudoku solver based on simulated annealing methods," in *Proc. Int. Conf. Field-Program. Technol.*, Sydney, NSW, Australia, Dec. 2009, pp. 522–525, doi: 10.1109/FPT.2009.5377608.

[53] M. Dittrich, T. B. Preusser, and R. G. Spallek, "Solving sudokus through an incidence matrix on an FPGA," in *Proc. Int. Conf. Field-Programmable Technol.*, Dec. 2010, pp. 465–469, doi: 10.1109/FPT.2010.5681460.

[54] X. Peng *et al.*, "Inference engine benchmarking across technological platforms from CMOS to RRAM," in *Proc. Int. Symp. Memory Syst.*, Sep. 2019, pp. 471–479, doi: 10.1145/3357526.3357566.

GIACOMO PEDRETTI (Member, IEEE) received the B.S., M.S., and Ph.D. *(cum laude)* degrees in electronics engineering from the Politecnico di Milano, Milan, Italy, in 2013, 2016, and 2020, respectively.

He is currently a Postdoctoral Research Associate with the Politecnico di Milano. His research interest includes the design of neuromorphic circuits for optimization and analog computing.

PIERGIULIO MANNOCCI received the B.Sc. and M.Sc. degrees in electronics engineering from the Politecnico di Milano, Milan, Italy, in 2016 and 2020 respectively, where he is currently pursuing the Ph.D. degree in information technology.

His main research interest includes modeling and design of analog and neuromorphic accelerators with emerging memory devices for in-memory computing.

SHAHIN HASHEMKHANI received the B.S. degree in electronics from Islamic Azad University Central Tehran Branch, Tehran, Iran, in 2014, and the M.S. degree in electronics from the Politecnico di Milano, Milan, Italy, in 2019, where he is currently pursuing the Ph.D. degree in electronics engineering.

His main research interest includes the design and characterization of neuromorphic networks.

VALERIO MILO (Member, IEEE) received the B.S., M.S., and Ph.D. *(cum laude)* degrees in electronics engineering from the Politecnico di Milano, Milan, Italy, in 2012, 2015, and 2019, respectively.

He is currently a Postdoctoral Researcher with the Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano. His current research interests include design, modeling, and simulation of neuromorphic networks with resistive switching random access memory (RRAM) and phase-change memory (PCM) for neuromorphic computing applications.

OCTAVIAN MELNIC received the B.S. and M.S. degrees in electrical engineering from the Politecnico di Milano, Milan, Italy, in 2015 and 2018, respectively, where he is currently pursuing the Ph.D. degree in electrical engineering.

His current research interest includes the characterization and modeling of phase-change memories.

ELISABETTA CHICCA (Member, IEEE) received the Laurea degree (M.Sc.) in physics from the Università degli Studi di Roma "La Sapienza," Rome, Italy, in 1999, the Ph.D. degree in natural science from the Department of Physics, Swiss Federal Institute of Technology, Zürich, Switzerland, in 2006, and the Ph.D. degree in neuroscience from the Neuroscience Center, Zürich, in 2006.

Since 2017, she has been a Professor with the Cognitive Interaction Technology Center of Excellence (CITEC) and the Faculty of Technology, Bielefeld University, Bielefeld, Germany, where she is leading the Neuromorphic Behaving Systems Research Group. Her research interests include the development of neuromorphic full-custom VLSI models of neural circuits for brain-inspired computation, learning in spiking neural networks, learning in memristive devices and arrays, bioinspired sensing, and motor control.

DANIELE IELMINI (Fellow, IEEE) received the Ph.D. degree from the Politecnico di Milano, Milan, Italy, in 2000.

He is currently a Full Professor with the Dipartimento di Elettronica, Informazione, e Bioingegneria, Politecnico di Milano. He conducts research on emerging nanoelectronics devices, such as phase-change memory (PCM) and resistive switching memory (RRAM), and novel computing circuits with memory devices.

Dr. Ielmini was a recipient of the Intel Outstanding Researcher Award in 2013, the ERC Consolidator Grant in 2014, and the IEEE EDS Rappaport Award in 2015.