frontiers in Neuroscience

ORIGINAL RESEARCH
published: 16 July 2021
doi: 10.3389/fnins.2021.674567

Hardware Implementation of Differential Oscillatory Neural Networks Using VO_2-Based Oscillators and Memristor-Bridge Circuits

Jafar Shamsi, María José Avedillo, Bernabé Linares-Barranco and Teresa Serrano-Gotarredona*

Instituto de Microelectrónica de Sevilla (CSIC), Universidad of Sevilla, Seville, Spain

OPEN ACCESS

Edited by:
Georgios Ch. Sirakoulis,
Democritus University of Thrace,
Greece

Reviewed by:
Can Li,
The University of Hong Kong,
Hong Kong
Alon Ascoli,
Technische Universität Dresden,
Germany

***Correspondence:**
Bernabé Linares-Barranco
bernabe@imse-cnm.csic.es

Specialty section:
This article was submitted to Neuromorphic Engineering, a section of the journal Frontiers in Neuroscience

Received: *01 March 2021*
Accepted: *15 June 2021*
Published: *16 July 2021*

Citation:
Shamsi J, Avedillo MJ, Linares-Barranco B and Serrano-Gotarredona T (2021) Hardware Implementation of Differential Oscillatory Neural Networks Using VO_2-Based Oscillators and Memristor-Bridge Circuits. Front. Neurosci. 15:674567. doi: 10.3389/fnins.2021.674567

Oscillatory Neural Networks (ONNs) are currently arousing interest in the research community for their potential to implement very fast, ultra-low-power computing tasks by exploiting specific emerging technologies. From the architectural point of view, ONNs are based on the synchronization of oscillatory neurons in cognitive processing, as occurs in the human brain. As emerging technologies, VO_2 and memristive devices show promising potential for the efficient implementation of ONNs. Abundant literature is now becoming available pertaining to the study and building of ONNs based on VO_2 devices and resistive coupling, such as memristors. One drawback of direct resistive coupling is that physical resistances cannot be negative, but from the architectural and computational perspective this would be a powerful advantage when interconnecting weights in ONNs. Here we solve the problem by proposing a hardware implementation technique based on differential oscillatory neurons for ONNs (DONNs) with VO_2-based oscillators and memristor-bridge circuits. Each differential oscillatory neuron is made of a pair of VO_2 oscillators operating in anti-phase. This way, the neurons provide a pair of differential output signals in opposite phase. The memristor-bridge circuit is used as an adjustable coupling function that is compatible with differential structures and capable of providing both positive and negative weights. By combining differential oscillatory neurons and memristor-bridge circuits, we propose the hardware implementation of a fully connected differential ONN (DONN) and use it as an associative memory. The standard Hebbian rule is used for training, and the weights are then mapped to the memristor-bridge circuit through a proposed mapping rule. The paper also introduces some functional and hardware specifications to evaluate the design. Evaluation is performed by circuit-level electrical simulations and shows that the retrieval accuracy of the proposed design is comparable to that of classic Hopfield Neural Networks.

Keywords: oscillatory neural networks, relaxation oscillators, coupled oscillators, vanadium dioxide, memristor, Hopfield Neural Network, associative memory

INTRODUCTION

Brain efficiency in cognitive processing relies on an architecture made up of distributed processors (neurons) and memories (synapses). Inspired by this brain architecture, novel cognitive processing paradigms are now being developed which go beyond the Von Neumann model. In addition to architecture, the devices employed also play a significant role in power and area efficiency. In this regard, emerging low power and compact devices constitute alternative resources useful for developing efficient cognitive processors.

Synthetic neural networks (NNs) have been under development for more than 50 years. Different types of NNs have been introduced, differentiated by their neuron types and data representation. Examples include classic artificial neural networks (ANNs) (Jain et al., 1996), spiking neural networks (SNNs) (Paugam-Moisy and Bohte, 2012), and oscillatory neural networks (ONNs) (Izhikevich, 1997). In classic ANNs, neurons are the simplest biological computing units that accumulate weighted input values prior to the application of an activation function to obtain the output value. In these neural networks, data representation is based on binary or real numbers. In SNNs, which have more biological features, spiking neurons receive and generate spikes, the spiking time or frequency of which is exploited for processing. In ONNs, an oscillator acts as a neuron, and the oscillator phase is the main characteristic used for processing. Like coupled oscillators, neurons process information by synchronizing. It has been hypothesized that synchronization plays an important role in cognitive processing (Gupta et al., 2016). Similarly to synapses, coupling functions describe the connections between the oscillators (Stankovski et al., 2015). A coupling function determines how one oscillator will affect another.

With regard to hardware, emerging devices such as vanadium dioxide (VO_2) (Velichko et al., 2017) and memristors (Strukov et al., 2008) make it feasible to implement oscillators and coupling functions efficiently. VO_2 devices have been exploited to design compact nano-scale, low power oscillators (Parihar et al., 2015; Shukla et al., 2015; Velichko et al., 2017; Raychowdhury et al., 2019; Corti et al., 2020a). A basic VO_2 oscillator circuit can be made by a series connection of a VO_2 device and a resistor or a CMOS transistor (Corti et al., 2020b). As coupling components, resistors and/or a capacitors are typically used to interconnect the oscillators. Resistively and capacitively coupled oscillators have shown their potential for processing tasks, such as image recognition (Corti et al., 2018) or vertex coloring (Weiher et al., 2021). For the case of a pair of VO_2 oscillators, when a pure resistor is employed for coupling them, increasing the connectivity strength (decreasing the resistance) tends to put the oscillator pair in phase, while for capacitively coupling, increasing the coupling capacitance tends to put the oscillator pair in anti-phase (Parihar et al., 2015). Although parallel resistors and capacitors make it easier to form a different phase difference between a coupled oscillator pair, it is not so easy to adjust the connection strength. Moreover, a capacitor usually consumes a large area of the hardware. For implementing adjustable connections, the memristor is a convenient device (Li et al., 2018; Sung et al., 2018; Camuñas-Mesa et al., 2019), as its non-volatile resistance can be adjusted to the desired value. However, the implementation of negative and zero weights is an issue. One proposed method is to use an extra memristor crossbar array to implement negative weights (Alibart et al., 2013; Molahasani Majdabadi et al., 2020). It is also possible to achieve both in phase and in anti-phase oscillations in a pair of coupled VO_2 oscillators using a single resistor as a coupling component (Corti et al., 2020a). In this method, a specific range of resistances (high resistance) puts the oscillator pair in anti-phase, while low resistances puts it in phase. Therefore, high resistance ranges and low resistance ranges could be equivalent to using negative and positive weights, respectively. However, this has only been studied for small size of ONNs, and scaling up to arbitrary size needs further research.

Here we propose another method, in which oscillatory neurons are made of pairs of VO_2 oscillators, coupled to be in anti-phase. This way, each neuron provides two differential outputs in anti-phase, and these differential oscillatory neurons are interconnected with adjustable positive- or negative-weight memristor-bridge circuits (Adhikari et al., 2012; Shamsi et al., 2017) to implement differential oscillatory neural networks (DONNs). First, a differential oscillatory neuron based on VO_2 devices is proposed to benefit from its differential outputs with anti-phase signals. The oscillatory neurons' differential output allows the memristor-bridge circuit to implement positive, negative, and zero weights. Differential oscillatory neurons are then interconnected through memristor-bridge circuits to form a Hopfield-type architecture (Hopfield, 1982) as an associative memory. The synaptic weights are calculated using the standard Hebbian rule and mapped to the memristors' resistances in the memristor-bridge circuits. Simulation results demonstrate the pattern retrieval capability of the proposed architecture.

To the best of our knowledge, this is the first attempt to introduce and implement differential oscillatory neural networks (DONNs). The main contribution of this paper is the integration of differential oscillatory neurons and memristor-bridge circuits to implement DONNs. Our study also introduces several criteria which can be used to evaluate and compare different implementations of ONNs.

The rest of the paper is organized as follows. Section "Methods" describes the architecture of DONNs, including a brief introduction to ONNs, VO_2 devices, and memristors. Section "Differential oscillatory neural networks (DONNs)" looks at circuit designs for DONNs. A differential oscillator neuron is proposed and the memristor-bridge circuit is introduced as an inter-neuron coupling function. The differential oscillatory neurons and memristor-bridge circuits are then combined to implement fully connected DONNs. Section "Evaluation Method" introduces some design specifications as design criteria for evaluating our design, and Section "Results" provides evaluation and simulation results, followed by some conclusions.

METHODS

Oscillatory Neural Networks

An ONN is a dynamic system that comprises weakly connected oscillatory neurons and is described through

Ordinary Differential Equations (ODEs) (Izhikevich, 1997; Nakao, 2016).

$$\dot{x}_{\mathrm{i}} = f_{\mathrm{i}}(x_{\mathrm{i}}) + \sum_{\mathrm{j}=1}^{n} w_{\mathrm{ij}} g_{\mathrm{ij}}(x_{\mathrm{i}}, x_{\mathrm{j}}), i = 1, 2, ..., n \qquad (1)$$

where n is the number of the oscillatory neurons and x_{i} is the state vector of oscillatory neuron i ($\boldsymbol{x}_{\mathrm{i}} \in \mathbb{R}^m,\ m \geq 2$), which is a function of its phase[1]. Function $f_{\mathrm{i}}(x_{\mathrm{i}})$ describes the dynamic behavior of the oscillatory neuron i and is usually formulated using m-dimensional differential equations. Parameter w_{ij} is the weight between oscillatory neurons i and j. $g_{\mathrm{ij}}(x_{\mathrm{i}},x_{\mathrm{j}})$ is the coupling function that shows the effect of oscillatory neuron j on oscillatory neuron i. The weights, w_{ij}, and the coupling function, $g_{\mathrm{ij}}(x_{\mathrm{i}},x_{\mathrm{j}})$, are crucial factors in oscillatory neural networks' behavior.

From the hardware perspective, oscillator function $f_{\mathrm{i}}(x_{\mathrm{i}})$ can be implemented using both harmonic and relaxation oscillators. In this paper, a relaxation oscillator based on $\mathrm{VO_2}$ devices is used. On the other hand, different coupling functions have been introduced in literature, such as a sinus function in the Kuramoto model (Acebrón et al., 2005) and a phase detector in PLL-based oscillators (Hoppensteadt and Izhikevich, 2000). One of the most straightforward coupling functions in terms of implementation is diffusive coupling, where $g_{\mathrm{ij}}(x_{\mathrm{i}},x_{\mathrm{j}}) = (x_{\mathrm{j}}\text{-}x_{\mathrm{i}})$ (Stankovski et al., 2015). The diffusive coupling function is implemented by connecting the oscillatory neuron outputs via resistors (memristors), creating what is known as resistively coupled oscillatory neurons (Corti et al., 2020a). Moreover, using a mapping method, the weights w_{ij} are mapped to resistance values R_{ij}. Using $\mathrm{VO_2}$-based oscillatory neurons and memristive devices, all elements of an oscillatory neural network can therefore be implemented. The following subsections summarize the main features and models of $\mathrm{VO_2}$ and memristor devices.

Vanadium Dioxide ($\mathrm{VO_2}$) Device Model

A $\mathrm{VO_2}$ device[2] is a two-terminal device based on a phase change material that presents insulator-to-metal (IMT) and metal-to-insulator transitions (MIT) (Velichko et al., 2017). The transition is temperature-driven, caused by the joule heating in the presence of an applied voltage. Increasing the device temperature causes a change from a high resistance (R_H) state to a low resistance (R_L) state and vice versa.

Although the temperature of the device is the main factor that causes the transitions, device behavior is also shaped by the applied voltage. In (Maffezzoni et al., 2015), a SPICE model for $\mathrm{VO_2}$ devices is introduced where the transitions are related to a high voltage V_H and a low voltage threshold V_L. When an increasing applied voltage reaches V_H, the resistance changes from its high resistance value $R_H = 1/G_L$ to its low resistance value $R_L = 1/G_H$. When a decreasing applied voltage drops below V_L, a transition from the low resistance state to the high resistance state occurs. The time constant of the transitions is τ. This model is compatible with fabricated $\mathrm{VO_2}$ devices, making it a reliable model for SPICE simulations. Although it is a compact SPICE model, it makes use of a discontinuous nonlinear function, which may yield convergence problems when simulating large scale circuits. To prevent this problem, we introduce here an equivalent mathematical macro model that uses a continuously differentiable nonlinear function, as follows:

$$\dot{g} = \frac{K}{\tau}\left[-g + \left(\frac{G_H + G_L}{2}\right) + \left(\frac{G_H - G_L}{2}\right) \times S(v, g)\right] \qquad (2)$$

where,

$$S\left(v, g\right) = tanh\left(A.\left(V_H - V_L\right).\left(\frac{v - V_L}{V_H - V_L} - \frac{g - G_L}{G_H - G_L}\right)\right) \qquad (3)$$

Variables g and v are the conductance and voltage of the $\mathrm{VO_2}$ device, respectively. Parameter A is a large constant value, and the other parameters are the main parameters of a $\mathrm{VO_2}$ device. Parameter K is a fitting parameter to match this mathematical model with the original SPICE model. A basic circuit of a $\mathrm{VO_2}$-based oscillator was simulated to compare the proposed mathematical model with the Maffezzoni SPICE model (see **Figure 1A**). The differential equation of the circuit is given by:

$$\begin{aligned} \dot{v} &= \tfrac{1}{C}.\left(-\left(G_s + g\right)v + G_s.Vdd\right) \\ \dot{g} &= G(g, v) \end{aligned} \qquad (4)$$

where G(g,v) is given by the right-hand expression in Eq. (2). **Figure 1B** shows the simulation results of the proposed mathematical model matching very well with Maffezzoni's SPICE model.

Memristor Model

A memristor is a two-terminal resistive device whose resistance is adjustable. It is typically used as an analog memory that can be both non-volatile or volatile (Ohno et al., 2011; van den Hurk et al., 2014; La Barbera et al., 2015; Wang et al., 2017; Ascoli et al., 2021). In addition, the adjustability of non-volatile memristors and their nanoscale size make them attractive candidates to implement massive adjustable synaptic circuits, especially with crossbar structures.

In some models, the memristor behavior is described by using one positive and one negative threshold voltages ($-|V_n|$ and V_p) (Ascoli et al., 2013; Krestinskaya et al., 2020). In memristors with counterclockwise switching, when the applied memristor voltage is larger than the positive threshold voltage V_p, the resistance of the memristor decreases. On the contrary, in memristors with clockwise switching, applying a voltage larger than V_p causes an increase in resistance, while the resistance decreases when the applied voltage is less than $-|V_n|$ (Min and Cho, 2021).With an applied voltage between $-|V_n|$ and V_p, the resistance remains constant.

The operational phases of a memristor are usually known as programming and operating phases (Shamsi et al., 2018). In the programming phase, the amplitude of each positive (negative)

[1] Note that this mathematical phase-based representation method results in an overall dynamical description, which ressembles that of traditional non-oscillatory ANNs.

[2] Although $\mathrm{VO_2}$ devices are negative differential resistance (NDR) devices, they can also be considered as volatile memristors capable of producing oscillatory circuits (Yi et al., 2018), as done throughout the present paper.

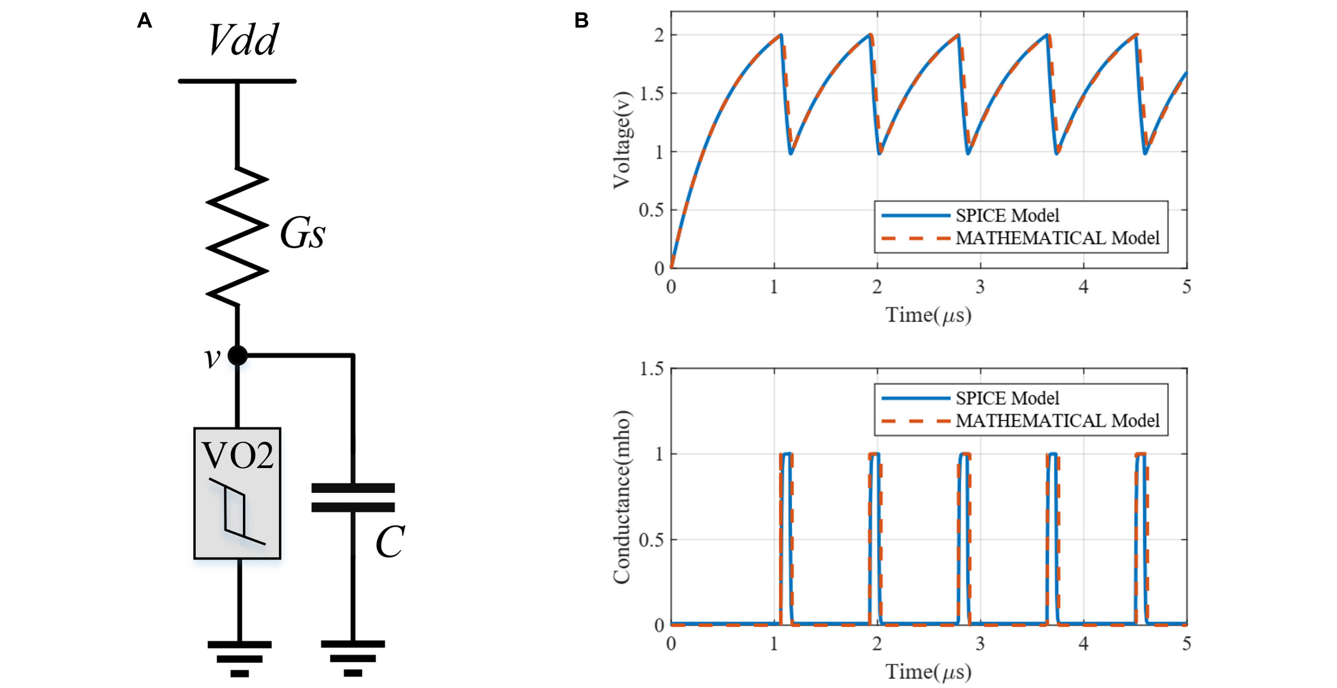

FIGURE 1 | (A) A basic oscillator circuit. **(B)** Simulation of the basic oscillator with Maffezzoni's SPICE model and the proposed mathematical model. The simulation parameters are $V_H = 2$, $V_L = 1$, $R_L = 1$ kΩ, $R_H = 100$ kΩ, $\tau = 30$ ns, and $K = 6$.

voltage pulse, applied across the memristor, is set to some predefined value larger than the modulus of the memristor threshold voltage V_p ($| V_n |$). Depending on the number, polarity, height and width of the pulses, the resistance is adjusted. On the other hand, in the operating phase, the maximum and minimum peaks of the pulses are within the intervals between the upper and lower threshold voltages, so there is no resistance change. This paper focuses on the operating phase of the memristor as a non-volatile synaptic circuit. We thus consider a memristor as a constant resistor, and weights are mapped to the resistance values. Consequently, throughout this paper we consider that memristor terminal voltages never exceed the threshold voltages.

Resistance range and threshold voltages are closely related to the materials used in memristor fabrication (Hadiyawarman et al., 2018), which therefore constitute important factors for memristor-based circuit designs. In this paper, the threshold voltages and the resistance ranges are assumed to meet the needs of our design. Specifically, we assume that maximum voltages across the memristors are never larger than the high threshold voltage V_H of the VO$_2$ devices. Also, some relations for defining a valid range of resistances are introduced in subsection "Circuit Design Method."

Differential Oscillatory Neural Networks (DONNs)

In this study, differential oscillatory neurons and memristor-bridge circuits are combined to implement a fully connected DONN, similar to a classic (non-oscillatory) Hopfield neural network. The proposed neural network circuit, shown in **Figure 2**, is used as an associative memory. The following subsections describe the differential oscillatory neurons and memristor-bridge circuits used as the basic blocks of the DONN.

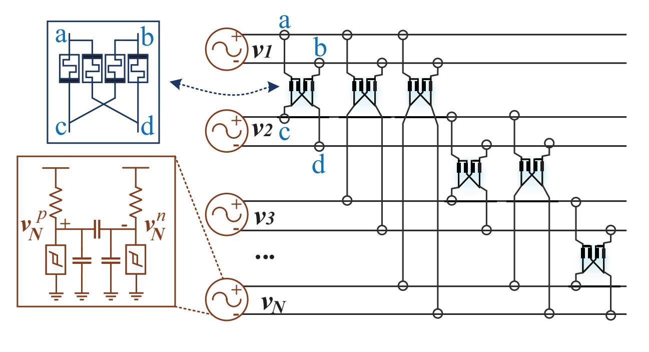

FIGURE 2 | Hardware architecture of the fully connected DONN. Differential neurons are implemented with a pair of VO$_2$ oscillators, oscillating in anti-phase. Synapses are implemented using memristor-bridge circuits.

Differential Oscillatory Neuron Circuit

Figure 3A shows the circuit of the proposed differential oscillatory neuron which generates two anti-phase signals. It comprises two single-ended oscillators connected to each other through a capacitor. The coupling capacitor forces the single-ended oscillators to be in anti-phase, in which they produce differential signals. To get anti-phase waveforms, the coupling capacitor Cc should be much lower than C ($Cc < < C$)

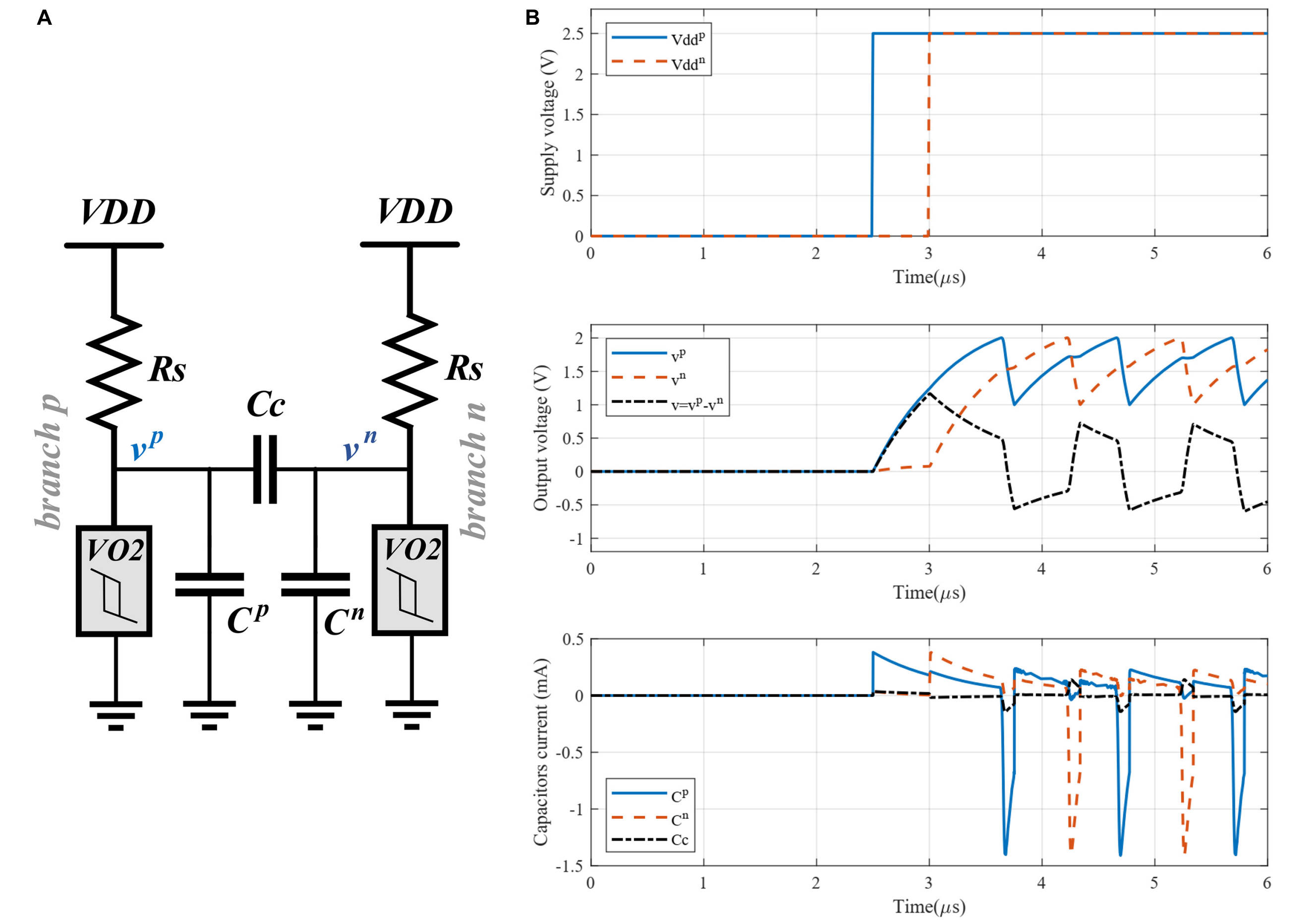

FIGURE 3 | (A) Proposed differential oscillatory neuron ($C^p = C^n = C$). **(B)** Simulation of the differential oscillatory neuron with power-on delay initialization; The positive and negative voltage sources were applied at $t = 2.5$ μs and $t = 3$ μs, respectively. Bottom figure shows the current through the capacitors.

(Parihar et al., 2015). However, when $Cc > > C$ the outputs tend to be in phase. In addition, when $Cc = C$ the output signals do not increase monotonically causing distorted output signals (Parihar et al., 2015). The following equations describe the dynamics of the differential oscillatory neuron.

$$\begin{aligned}
\dot{v}^p &= \frac{1}{C+C_C}\left[-\left(G_s+g^p\right)v^p + C_C.\dot{v}^n + G_s.Vdd\right]\\
\dot{v}^n &= \frac{1}{C+C_C}\left[-\left(G_s+g^n\right)v^n + C_C.\dot{v}^p + G_s.Vdd\right]\\
\dot{g}^p &= G\left(g^p, v^p\right)\\
\dot{g}^n &= G\left(g^n, v^n\right)
\end{aligned} \tag{5}$$

where indexes p and n indicate the corresponding branch p and n, respectively.

The starting point for exploiting a DONN is the initialization phase, in which an input pattern is applied to the DONN. The power-on delay method can be used to initialize the differential oscillatory neuron (Corti et al., 2020b). In this method, the power supplies of the single-ended oscillator branches are applied in a power-on sequence with a specific delay. Given an input value, the power supply is applied to one of the branches first, and then, after a given time, to the second branch. For binary values, the delay time should be half of the oscillators' period. For instance, when the binary value is 1, the power supply is applied to the positive branch first, and then, after one half period, to the negative branch. **Figure 3B** shows the simulation results of a differential oscillatory neuron for which the period is 1 μs and the delay between the applied power supplies is 0.5 μs. The output voltages are anti-phase shown in the middle of **Figure 3B**.

In our design, parameters C, C_C, R_s, and Vdd were calculated using relations that satisfy the operating conditions of the DONN circuit (see subsection "Circuit Design Method"). Some relations are necessary to ensure the VO_2 devices are biased in their negative resistance region to guarantee the respective branches perform as oscillators. Using these relations, parameters R_s and Vdd are calculated. Other relations are used to calculate parameters Cc and C. Details of the operating conditions are provided in subsection "Circuit Design Method."

Synaptic Memristive-Bridge Circuit

As an analogy of the Wheatstone bridge, the memristor-bridge is introduced to implement a synaptic circuit capable of providing

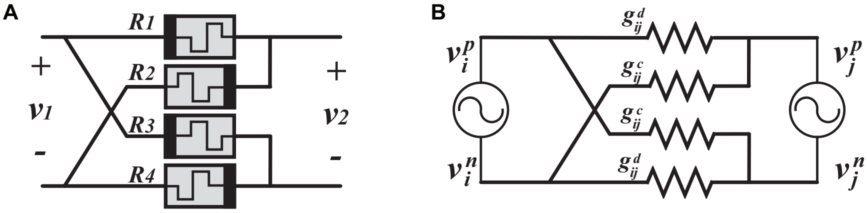

FIGURE 4 | (A) Memristor-bridge circuit. **(B)** Equivalent circuit of two differential oscillatory neurons coupled through one memristor-bridge circuit.

positive, negative, and zero weights (**Figure 4A**; Adhikari et al., 2012). Being a four-terminal circuit makes it appropriate for differential structures. However, it is also used in single-ended neural networks (Jackson et al., 2016; Shamsi et al., 2017). **Figure 4B** shows the equivalent circuit with two differential oscillatory neurons coupled by the memristor-bridge circuit. The following conditions are required to implement a symmetric coupling function.

$$g_{\mathrm{ij}}^{\mathrm{d}} = \frac{1}{R_1} = \frac{1}{R_4} \tag{6}$$

$$g_{\mathrm{ij}}^{\mathrm{c}} = \frac{1}{R_2} = \frac{1}{R_3} \tag{7}$$

where superscript d stands for "direct" path (positive with positive or negative with negative), while superscript c stands for "crossed" path (positive with negative, and vice versa). Conductance $g_{\mathrm{ij}}^{\mathrm{d}}$ is located between the positive branches (negative branches), tending to put positive branches (negative branches) in phase. When the positive branches (negative branches) are in phase, then the two differential oscillatory neurons are considered in phase. On the other hand, conductance $g_{\mathrm{ij}}^{\mathrm{c}}$ is located between a positive and a negative branch, thus tending to put them in phase while the positive branches (negative branches) are anti-phase. When the positive branches (negative branches) are anti-phase, then the two differential oscillators are considered as anti-phase. Considering the values of $g_{\mathrm{ij}}^{\mathrm{d}}$ and $g_{\mathrm{ij}}^{\mathrm{c}}$, differential oscillatory neurons tend to be in phase (anti-phase) when $g_{\mathrm{ij}}^{\mathrm{d}} > g_{\mathrm{ij}}^{\mathrm{c}}$ ($g_{\mathrm{ij}}^{\mathrm{d}} < g_{\mathrm{ij}}^{\mathrm{c}}$). In this regard, it is possible to implement a coupling circuit with a positive (if $g_{\mathrm{ij}}^{\mathrm{d}} > g_{\mathrm{ij}}^{\mathrm{c}}$), negative (if $g_{\mathrm{ij}}^{\mathrm{d}} < g_{\mathrm{ij}}^{\mathrm{c}}$), or zero (if $g_{\mathrm{ij}}^{\mathrm{d}} = g_{\mathrm{ij}}^{\mathrm{c}}$) weight.

It also worth mentioning that the maximum voltage across a memristor is kept less than V_H. Suppose that a memristor is connected to the output nodes of two VO_2 devices from two distinct differential oscillatory neurons. Therefore, the maximum voltage across the memristor will be $|\ V_H - V_L|$. However, during the initialization Vdd is applied with a delay to a branch that may increase the voltage of one terminal of the memristor up to V_H, while the voltage of the other terminal is around zero. Thus, the maximum voltage across a memristor will in general be less than V_H.

The following subsection reviews the standard Hebbian rule used to calculate the synaptic weights. A mapping method is also introduced to map the synaptic weights to the conductance values $g_{\mathrm{ij}}^{\mathrm{d}}$ and $g_{\mathrm{ij}}^{\mathrm{c}}$.

Training and mapping rules

A training rule is used to store patterns in neural networks, adjusting the synaptic weights accordingly. Once the weights are known, we propose a mapping rule to obtain the physical resistances for the memristor-bridge synapses. To store patterns in the DONN, we use the Hebbian rule to calculate the weights.

$$w_{\mathrm{ij}} = \frac{1}{L}\sum_{k=1}^{P} b_{\mathrm{i}}^{k} b_{\mathrm{j}}^{k} \qquad i, j \in \{1,\ 2,\ 3,\ \ldots,\ L\} \tag{8}$$

where P is the number of stored patterns and L is the number of pixels in each pattern (which is equal to the number of neurons in the DONN). Elements b_{i} and b_{j} of all stored patterns are used to calculate the weight w_{ij}.

We propose here the following rules to map the sign and value of the above weights to the memristors' resistances. Weights w_{ij} are mapped to the g_{ij} values using the following relation.

$$Map_{value}: \quad g_{\mathrm{ij}} = \begin{cases} \dfrac{g_0}{1+\beta\times P\times\left|\frac{1}{w_{\mathrm{ij}}}\right|_{norm}} & , w_{\mathrm{ij}} \neq 0 \\ \dfrac{g_0}{1+\beta\times P} & , \ w_{\mathrm{ij}} = 0 \end{cases} \tag{9}$$

where g_0 is the maximum conductance (inverse the minimum resistance) of the memristors. Parameter β is a small positive value (e.g., 0.2) that controls the mapping range for conductance g_{ij}. A larger value for β provides a larger range for conductance g_{ij}. Value $\left|1/w_{\mathrm{ij}}\right|_{norm}$ is the Min-Max normalization of $|1/w_{\mathrm{ij}}|$. In order to obtain the Min-Max normalization, the following relation is used.

$$\left|\frac{1}{w_{\mathrm{ij}}}\right|_{norm} = \frac{\left|\frac{1}{wij}\right| - min}{max - min} \tag{10}$$

where min and max are the minimum and maximum values of $|1/w_{\mathrm{ij}}|$ among all non-zero weights, respectively. The following mapping rule was also used to map the weight signs to resistance values $g_{\mathrm{ij}}^{\mathrm{d}}$ and $g_{\mathrm{ij}}^{\mathrm{c}}$.

$$Map_{sign}: \quad \begin{cases} \alpha g_{\mathrm{ij}}^{\mathrm{d}} = g_{\mathrm{ij}}^{\mathrm{c}} = g_{\mathrm{ij}}, & w_{\mathrm{ij}} < 0 \\ g_{\mathrm{ij}}^{\mathrm{d}} = \alpha g_{\mathrm{ij}}^{\mathrm{c}} = g_{\mathrm{ij}}, & w_{\mathrm{ij}} > 0 \\ \alpha g_{\mathrm{ij}}^{\mathrm{d}} = \alpha g_{\mathrm{ij}}^{\mathrm{c}} = g_{\mathrm{ij}}, & w_{\mathrm{ij}} = 0 \end{cases} \tag{11}$$

where $\alpha > 1$ is a constant value.

The design parameters of the memristor-bridge are w_{ij}, α, β, and g_0. Parameter g_0 will be obtained based on correctly

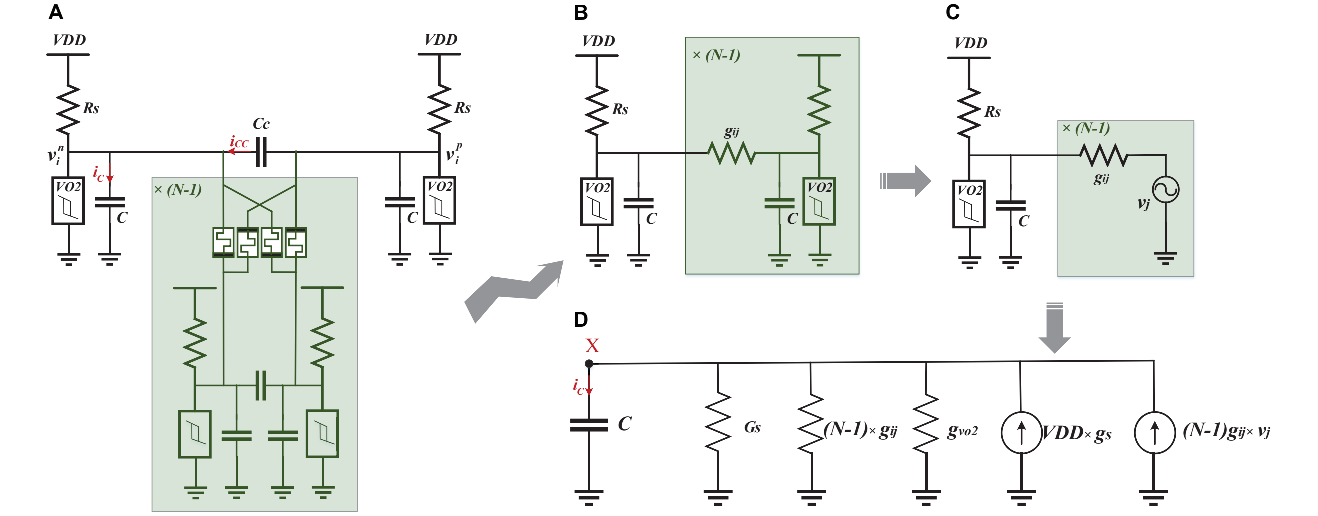

FIGURE 5 | Simplification of the DONN. **(A)** The DONN from the viewpoint of one of the differential oscillatory neurons connected to the other *(N−1)* ones. **(B)** The circuit as simplified by ignoring the current i_{CC}, and assuming that the oscillatory neurons are either in phase or in anti-phase. **(C)** Replacement of the *(N−1)* oscillatory neurons with voltage sources. **(D)** Norton equivalent of the simplified circuit.

functioning hardware, as explained next in subsection "Circuit Design Method." Section "Results" then illustrates how the DONN performance depends on parameters α and β .

Circuit Design Method

As a starting point, design relations are extracted to calculate circuit parameters by analyzing the fully connected DONN circuit (**Figure 2**). **Figure 5A** shows the circuit from the viewpoint of a single differential oscillatory neuron coupled to other *(N−1)* differential oscillatory neurons. The dynamics of the former differential oscillatory neuron is described by:

$$\begin{aligned}\dot{v}_{\mathrm{i}}^{p} &= \tfrac{1}{C+C_C}\left[-\left(G_s+g_{\mathrm{i}}^{p}\right)v_{\mathrm{i}}^{p}+C_C.\dot{v}_{\mathrm{i}}^{n}+G_s.Vdd\right]+\sum_{\mathrm{j}=1}^{N} i_{\mathrm{ij}}^{p}\\ \dot{v}_{\mathrm{i}}^{n} &= \tfrac{1}{C+C_C}\left[-\left(G_s+g_{\mathrm{i}}^{n}\right)v_{\mathrm{i}}^{n}+C_C.\dot{v}_{\mathrm{i}}^{p}+G_s.Vdd\right]+\sum_{\mathrm{j}=1}^{N} i_{\mathrm{ij}}^{n}\\ \dot{g}_{\mathrm{i}}^{p} &= G\left(g_{\mathrm{i}}^{p},v_{\mathrm{i}}^{p}\right)\\ \dot{g}_{\mathrm{i}}^{n} &= G\left(g_{\mathrm{i}}^{n},v_{\mathrm{i}}^{n}\right)\end{aligned} \tag{12}$$

Where i_{ij}^{p} and i_{ij}^{n} are the currents from oscillatory neuron *j* to the positive and negative branch of oscillatory neuron *i*, respectively:

$$\begin{aligned}i_{\mathrm{ij}}^{p} &= g_{\mathrm{ij}}^{\mathrm{d}}(v_{\mathrm{j}}^{p}-v_{\mathrm{i}}^{p})+g_{\mathrm{ij}}^{\mathrm{c}}(v_{\mathrm{j}}^{n}-v_{\mathrm{i}}^{p})\\ i_{\mathrm{ij}}^{n} &= g_{\mathrm{ij}}^{\mathrm{c}}(v_{\mathrm{j}}^{p}-v_{\mathrm{i}}^{n})+g_{\mathrm{ij}}^{\mathrm{d}}(v_{\mathrm{j}}^{n}-v_{\mathrm{i}}^{n})\end{aligned} \tag{13}$$

In order to guarantee that each VO_2 device branch operates as an oscillator, the following condition is imposed:

- Each VO_2 device operates in its negative resistance region: $V_{\mathrm{Max}} > V_{\mathrm{H}}$ and $V_{min} < V_L$, where V_{Max} (V_{min}) is the maximum (minimum) output voltage if IMT (MIT) does not occur in the VO_2 device.

This condition guarantees that the voltage of the oscillators do not reach stable points, resulting in permanent oscillation (Raychowdhury et al., 2019). The DONN circuit is analyzed next to extract relations for V_{Max} and V_{min}. For the analysis, and to simplify the circuit, some assumptions are considered. First, we consider a single point for analysis where the voltage of the positive branch of a differential oscillatory neuron is either at its maximum or minimum value. Regarding the assumption $C >> C_C$, the share of current i_{CC} in the total current i_C is considered negligible. For instance, the current of capacitor C (in positive and negative branches) and C_C are shown at the bottom of **Figure 3B**. When the voltage of the positive branch is at its maximum value, immediately before the jump, the current through capacitors C^p and C_C are 71 and 7.5 μA, respectively. Also, when the voltage of positive branch is at its minimum value, immediately before the jump, the current through capacitors C^p and C_C are 686 and 76 μA, respectively. Thus we omit capacitor *Cc*. It is also assumed that the differential oscillatory neurons with respect to each other are either in phase or in anti-phase. When two differential oscillatory neurons are in phase, the voltage of their positive branches (negative branches) is equal. Therefore, the current through the memristors between the positive branches (negative branches) is zero. Similarly, when in anti-phase, the current through the memristors between a positive and a negative branch is zero, as well. In this regard, the circuit is simplified, as shown in **Figure 5B**. The simplified circuit is a single-ended oscillator connected to the other *(N−1)* single-ended oscillators. We are also able to replace the *(N−1)* single-ended oscillators with *(N−1)* voltage sources, which generate the same signals than the respective single-ended oscillators (**Figure 5C**). The circuit is now simple enough to be analyzed easily. The Norton equivalent circuit is shown in **Figure 5D**. We use next some worst-case operating conditions to extract design relations. In other words, relations are extracted for the most difficult operating conditions to ensure that the calculated parameters are valid for all conditions.

Consequently, parameters G_H, G_L, V_H, and V_L are used in the analysis so that the worst-case conditions are satisfied. In this regard, the maximum (minimum) voltage at node X in **Figure 5D** is obtained for the charging (discharging) period. Then, parameters G_H, G_L, V_H, and V_L are used to find the minimum (maximum) voltage of the charged (discharged) capacitor after the charging (discharging) period, which should be larger (smaller) than V_H (V_L). It worth mentioning that IMT (MIT) in VO$_2$ devices occurs before reaching the maximum (minimum) voltage of the capacitor/VO$_2$ devices, consequently causing permanent oscillation.

$$V_{Max}^{p} = \frac{G_s V_{dd} + g\,(N-1)\,V_L}{\left(G_s + G_L + (N-1)\,g\right)} > V_H \tag{14}$$

$$V_{min}^{p} = \frac{G_s V_{dd} + g\,(N-1)\,V_H}{\left(G_s + G_H + (N-1)\,g\right)} < V_L \tag{15}$$

These relations are used to calculate the values for G_S and g_0. In this regard, we first use relations with $N = 1$ to find a range for Gs and a value in this range is selected for Gs. Afterward, using the same equations with the selected value of Gs and the number of differential oscillatory neurons N, a range for g is obtained. The maximum value of this range is used as g_0. Finally, we substitute all parameters in the equations to verify whether the selected values are valid. Otherwise, the same procedure is repeated with a new value for Gs.

The values for C and C_C are other design parameters that need to be considered. It is worth mentioning that a large coupling capacitance value provides strong coupling. On the other hand, a large value for the Cc/C ratio causes degradation of the waveform shapes, resulting in deviations from ideal anti-phase waveforms for differential oscillatory neurons (Parihar et al., 2015). Strong coupling therefore has to be traded off against ideal anti-phase waveforms. In this paper, the following relation between C and C_C is used.

$$C_c \approx 0.1 \times C \tag{16}$$

Also, the oscillator period relation is used to calculate C and C_C (Corti et al., 2020a).

$$T = C^{*} \times \left[\left(\frac{1}{G_L + G_s} \right) ln \left(\frac{V_{max} - V_L}{V_{max} - V_H} \right) + \left(\frac{1}{G_H + G_s} \right) ln \left(\frac{V_{min} - V_L}{V_{min} - V_H} \right) \right] \tag{17}$$

where $V_{max} = \frac{G_s Vdd}{G_L + G_s}$ and $V_{min} = \frac{G_s Vdd}{G_H + G_s}$. Parameter C^{*} is the capacitance at the output nodes of the differential oscillatory neurons and is approximately considered as $C^{*} \approx C + C_c$. In the relations mentioned earlier, the input parameters were Vdd (voltage supply of the circuit), V_H (high threshold voltage of the VO$_2$ device), V_L (low threshold voltage of the VO$_2$ device), G_H (high conductance value of the VO$_2$ device in the metallic state), G_L (low conductance value of the VO$_2$ device in the insulator state), and N (the number of neurons in the neural network).

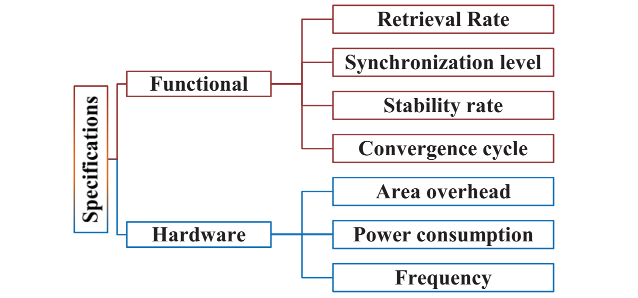

FIGURE 6 | Functional and hardware specifications for evaluating the design.

Evaluation Method

Figure 6 shows the specifications used to evaluate our design. These specifications are classified into two categories: functional and hardware specifications. The functional specifications are defined as follows:

- *Retrieval rate* is defined as a value in the range of [0 1] that shows the ratio of correct retrieved patterns to all applied patterns:

$$RET = \frac{\#\ correct\ retrived\ patterns}{\#\ all\ patterns} \tag{18}$$

- *Synchronization level* is defined to measure how many differential oscillatory neurons (neurons[3]) are synchronized either in phase or in anti-phase. For an input pattern p, the synchronization level is a value in the range of [0 1] that is related to the average deviations between synchronized signals in a period c:

$$SYN_p\,(c) = < \frac{1}{N} \sum_{\mathrm{i}=1}^{N} m(PT_{\mathrm{i}}\,(c) - PT_{ref}\,(c)) > \tag{19}$$

The $SYN_p(c)$ value shows the synchronization level at cycle c, which is related to the time difference between the peak time of signal i at cycle c, $PT_{\mathrm{i}}(c)$, and the peak time of the reference signal $PT_{ref}(c)$ (the signal of positive branches v_{i}^{p} are used for calculations). Function $m = m(\Delta t)$ maps a time difference Δt to a value between 0 to 1 (see **Figure 7**). It is worth mentioning that time difference Δt is equivalent to phase difference. According to function $m()$, synchronization level $SYN_p(c)$ is maximum when the neurons are either in phase ($\Delta t = 0$ or $\Delta t = \pm T$) or in anti-phase ($\Delta t = \pm T/2$). When the neurons are not exactly in phase or in anti-phase, a value less than 1 is assigned depending on the phase difference. For instance, the worst case in terms of synchronization corresponds to the phase difference $T/4$.

- *Stability rate* is defined as a value in the range of [0 1] which is the number of applied patterns resulting in stable outputs divided by all applied patterns:

$$STB = \frac{\#\ patterns\ resulting\ in\ a\ stable\ output}{\#\ all\ patterns} \tag{20}$$

[3] In the following, the word "neuron" is used for "differential oscillatory neurons."

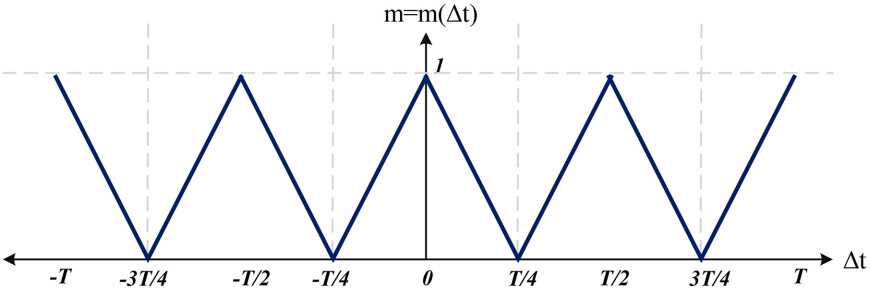

FIGURE 7 | Function *m = m(Δt)* used to map a time difference to a value between 0 and 1.

Output stability for an applied pattern is defined based on the average $SYN_p(c)$ for all signals in the last few cycles. When this average value is larger than 0.9, the output is considered as a stable output.

- *Convergence cycle* is defined as the number of cycles Ψ required to converge to a pattern:

$$CON = \Psi \tag{21}$$

where,

$$SYN_p\,(\Psi) > 0.9\ \times Max_{\forall c>\Psi}\{SYN_p\,(c)\} \tag{22}$$

When the output converges to a pattern, the synchronization level is larger than 90% of the maximum synchronization level in subsequent cycles.

In addition to the functional specifications, some hardware specifications are also considered to evaluate the design, such as power consumption per neuron, frequency, and area overhead.

RESULTS

This section describes the evaluation of the design. A DONN with 15 neurons is first simulated to show how DONNs work[4]. A comprehensive evaluation is then provided to show the performance of DONNs.

Operation of a DONN

A DONN with $N = 15$ neurons (differential oscillatory neurons) was designed to illustrate the concepts underlying the functional specifications and operation of the DONN.

The design relations Eqs. (14)-(17) and the input parameters (the number of neurons *N*, oscillation frequency *f*, supply voltage *Vdd*, and VO_2 parameters) were used to calculate the circuit parameters. First, Eqs. (14)-(15) were used to calculate values for g_0 and *Gs*. The values for *C* and C_C were then computed using relations Eqs. (16)-(17). After that, the synaptic weights were calculated with the standard Hebbian rule to store the three patterns shown in **Figure 8A**. Each pattern has 15 pixels and each pixel value at any time is represented by the phase of one of the 15 neurons (each made of two physical VO2 oscillators in anti-phase) with respect to the reference neuron. The rules in Eqs. (9) and (11) were then used to map the weights to the physical synaptic resistance with α = 1.8 and β = 0.2. A summary of the parameters for this design is shown in **Table 1**.

Figure 8B shows simulation results for an applied input pattern. The voltage of the positive branches can be seen in the top sub-figure. Input patterns are applied by setting a specific initial phase for each neuron. The initial phases of the neurons were set by powering up first the positive branch and then the negative branch of each neuron associated to the white pixels. In contrast, for black pixels, the positive branch was powered up after the negative branch. On the other hand, in order to extract the value of pixel *i* (or neuron *i*) for a specific cycle, the time difference between the voltage peak of the positive branch in neuron *i* with respect to a reference voltage peak (e.g., the voltage of the positive branch in neuron 1) is measured. For each neuron, if the time difference is zero (half of the period), then a white (black) pixel value is assigned. In **Figure 8B** (bottom figure) the 1st cycle pattern (which is not among the stored patterns in **Figure 8A**) is shown. It can be seen that it resembles the input pattern. The same figure also shows the synchronization level during the convergence period. The minimum synchronization level corresponded to a moment in which the deviation between the signals was maximum. The maximum synchronization level occurred when the neurons were either in phase or in anti-phase. The convergence cycle for this pattern was 15 cycles.

Figure 8C shows another simulation of unstable neural network behavior in which the mapping parameter α was a large value (α = 10^6). For this value, g_{ij}^{c} (g_{ij}^{d}) was a negligible value for positive (negative) weights (see the mapping rule). For zero weights, g_{ij}^{c} and g_{ij}^{d} were negligible values in comparison to g_0 and memristors can be considered as open circuits. The simulation results show that the output was unstable for the input pattern (**Figure 8C**). The synchronization level changed periodically, and the evolution of the pattern shows that the retrieved pattern periodically changed from one pattern (pattern 0) to another (pattern 2).

Performance of DONNs

The DONN was evaluated with different numbers of neurons *N* using random orthogonal or semi-orthogonal patterns. The hamming distance between orthogonal patterns is exactly half the length of the patterns (number of neurons), while, for

[4] A presentation with a general overview of DONN is provided in the **Supplementary Material** section.

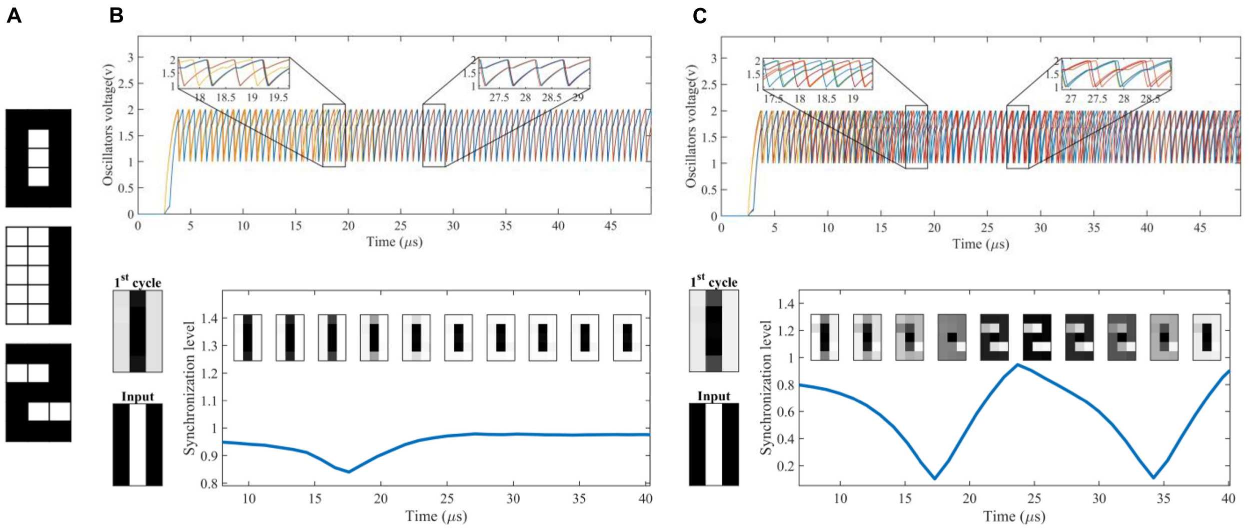

FIGURE 8 | Simulation results of a DONN with 15 neurons. **(A)** Three patterns were stored in the DONN. **(B)** Simulation results for an input pattern with α = 1.8. The top figure shows the voltage of the positive branches of the differential oscillatory neurons. The bottom figure shows the input pattern, the 1st cycle pattern, the synchronization level and the evolution of the retrieved pattern. According to the definition of convergence cycle in Section "Evaluation method," the ONN requires15 cycles to retrieve the stored pattern. **(C)** A simulation result with a large value for mapping parameter α (α = 10^6), showing unstable DONN behavior. Although the output converged to a stored pattern, it periodically changed from one pattern to another.

semi-orthogonal patterns it is close to half of the length of the patterns but different. **Figure 9A** shows three samples for 4×4 randomly generated patterns. The ratio of black and white pixels in each pattern was 50%. Approximately 10% of noise was added to the patterns to generate the test patterns. The number of test patterns was 1.5^N. The applied noise changed the color of a noisy pixel from black to white or from white to black. The values for the frequency, supply voltage, and VO_2 device parameters are taken from **Table 1**. The circuit parameters were calculated in the same way as in the previous case study, and mapping parameter α was selected based on the following Figure-of-Merit for the neural network performance.

$$FoM = < RET > \times < SYN > \times < STB > \\ \times < 1 - |CON|_{norm} > \quad (23)$$

where $|CON|_{norm}$ is Min-Max normalization of convergence cycles CON, which is calculated using a relation similar to Eq. (10). Parameter SYN is the average of $SYN_p(c)$ for all signals of all applied patterns in the last few cycles. **Figure 9B** shows the DONN specifications for different values of α, which were used to calculate the FoM shown in **Figure 9C**. The simulation was carried out for DONNs with different numbers of neurons N, and each point in the figure represents the average of 5 separate simulation results. The number of stored patterns for N = 6, 8 was two, while three patterns were stored for the DONNs with N = 12, 14, 16. The results show that the performance of the neural network was related to the mapping value α. When value α was 1, performance was zero because all weights were zero (considering Eq. 11, $g_{ij}{}^{d} = g_{ij}{}^{c}$ for all signs), and the neurons did not affect each other. In this case, the neuron outputs did not change, the output pattern was the same as the input pattern, and the retrieval rate was consequently zero. By increasing the value of parameter α, positive and negative weights were formed, and performance increased. For larger values, however, the network was unstable for some patterns (**Figure 8C**), and its performance decreased accordingly. With a given performance value, a proper value for parameter α could be selected using **Figure 9C**. The results of the evaluation of the simulations are shown in **Figure 10**. Functional specifications versus the number of neurons are shown in **Figure 10A**. The retrieval rate of the proposed design was compared with the classic HNN, showing comparable accuracy. The stability and synchronization levels show that the output was stable, and that the neurons were synchronized properly. **Figure 10B** represents the convergence cycles, i.e., the average number of cycles required for the

TABLE 1 | Design parameters.

Parameter		Value
Number of neurons	N	15
Frequency	$f = 1/T$	1 MHz
Supply voltage	Vdd	2.5 V
VO_2 parameters	V_H	2 V
	V_L	1 V
	$R_H = 1/G_L$	100 kΩ
	$R_L = 1/G_H$	1 kΩ
Series resistance	$Rs = 1/Gs$	6 kΩ
Minimum memristor resistance	$r_0 = 1/g_0$	221 kΩ
Mapping parameters	α	1.8
	β	0.2
Coupling capacitor	C_C	11 pF
Parallel capacitor	C	108 pF

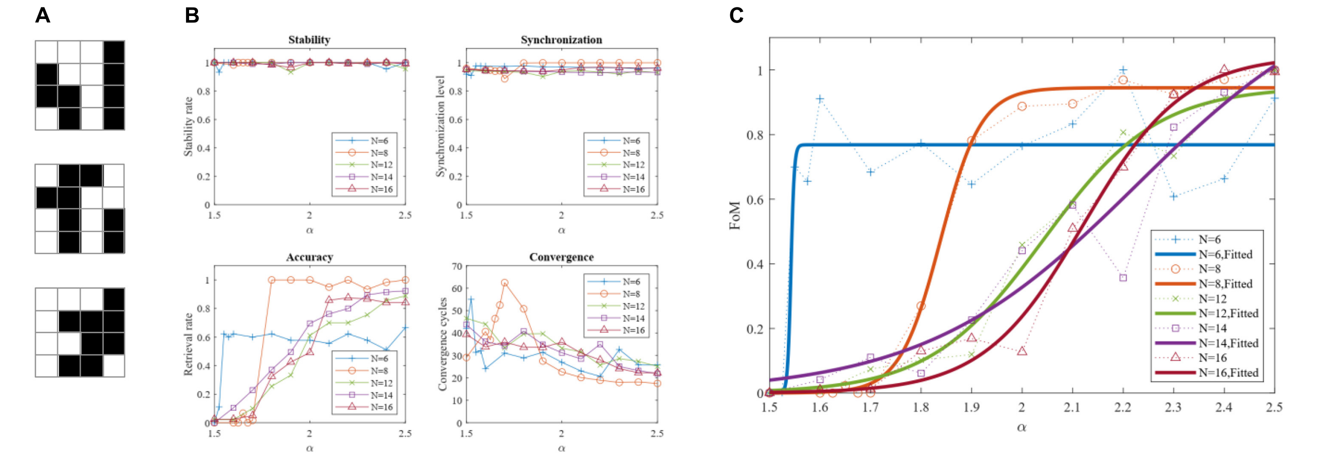

FIGURE 9 | Dependence of the DONN performance on mapping parameter α. **(A)** Three sample patterns generated randomly for a DONN with N = 16 neurons. **(B)** DONN specifications versus parameter α. **(C)** Dependence of FoM on mapping parameter α. The dotted lines are the simulation results for the different number of neurons. The solid fitted curves show how the FoM value changed with the value of parameter α.

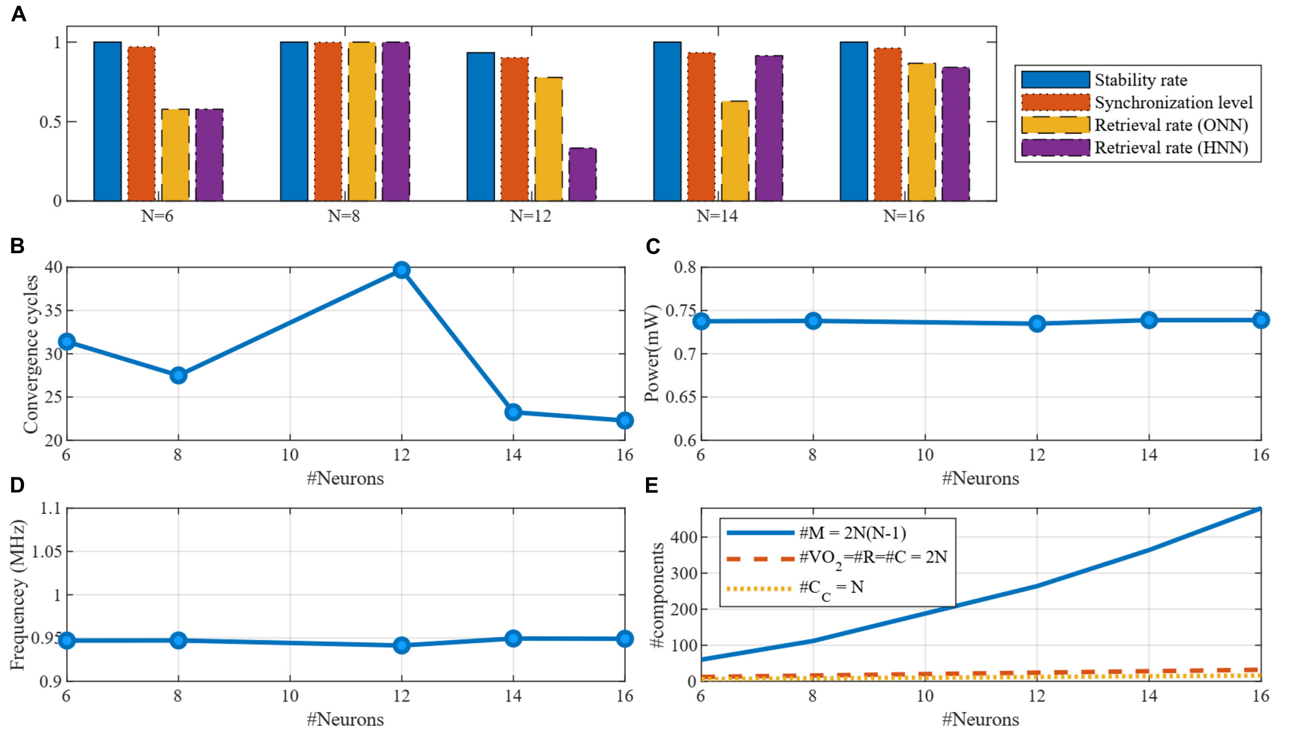

FIGURE 10 | Simulation results of DONNs with different number of neurons. **(A)** Functional specifications of the design including the stability rate, synchronization level, and retrieval rate for both DONN and HNN. **(B)** Average of the convergence cycles for a pattern retrieval. **(C)** Power consumption per differential oscillatory neuron, which was constant with the number of neurons. **(D)** Frequency of the oscillatory neurons. **(E)** Number of components required to implement a fully connected DONN.

retrieval of patterns. **Figure 10C** shows power consumption per neuron, which remains constant with the number of neurons, and was about 735 μW or, equivalently, 0.77 pJ/cycle. The frequency of around 950 kHz shown in **Figure 10D** was around the expected value (1 MHz). The number of components, as an indication of area overhead is given in **Figure 10E**. The number of memristors scales quadratically with the number of neurons.

$$\#M = 2N(N-1) \tag{24}$$

However, the number of the other components scales linearly with the number of neurons.

TABLE 2 | Comparison of memristor-based DONNs with non-oscillatory HNNs.

Metrics	The proposed memristive DONN	Mixed-signal Memristive HNN (Sequential operation) (Cai et al., 2020)	Analog Memristive HNN (Hu et al., 2015)
Frequency	1 MHz (oscillators frequency)	500 MHz (clock for the digital part)	N.A (pure analog HNN)
Convergence time (Number of neurons)	28.6 cycles: 28.6 μs with 1 MHz (N = 6-16)	6.6 μs (N = 60)	35 μs (N = 3)
Power consumption per neuron	735 μW (Vdd = 2.5 v)	182 μW	80000 μW
circuit complexity (N is the number neurons)	#Memristor = 2N(N−1) #Capacitor = 3N #Resistor = 2N #VO2 = 2N	# Memristor = N^2 #Register = N bit #Analog buffer = N #Multiplexer = N bit #Amplifier = 1 #Comparator = 2 #D-FF = 2 #Pass transistor = 2	#Memristor = N(N−1) #Resistor = 5N #Amplifier = N # Analog buffer = N #Inverter = N

Table 2 provides a comparison between the proposed memristive DONN and classic non-oscillatory HNNs. The metrics in the table directly affect the main characteristics of the system which are throughput (convergence time), energy efficiency (power consumption per neuron), and area overhead (hardware complexity). Frequency is applicable for oscillatory neural networks and digital systems, which is 1MHz in the proposed DONN and 500 MHz in the digital hardware introduced in Cai et al. (2020). As a metric for throughput, convergence time is the required time to retrieve a pattern after applying an input. However, this convergence time is highly related to the pattern size (number of neurons in the neural network). The convergence time reported in Cai et al. (2020) is 6.6 μs where the network consists of 60 neurons. In (Hu et al., 2015), it takes 35 μs to retrieve a pattern in an HNN with 3 neurons. In the proposed DONN, the average convergence time is 28.6 cycles, being directly related to the frequency of the neurons. Here, in our simulations, it takes 28.6 μs on average to retrieve a pattern with the frequency being 1 MHz. By increasing the frequency, retrieval time is decreased. For instance, if the frequency would be 500 MHz, the convergence time would be 28.6/500 = 0.027 μs. In the proposed DONN, the power consumption per neuron is 735 μW. One way to improve power consumption is decreasing the voltage of the power supply V_{dd} and V_H, which is possible by reducing the size of the VO_2 devices (Corti et al., 2021). Hardware complexity is another metric to show area overhead, which is provided by counting the number of components. In (Hu et al., 2015), the minimum number of memristors was used, which is half of our design. However, considering other components, DONNs require only basic components (two VO$_2$ devices, two resistors, and three capacitors per neuron) which make them potentially compact blocks. On the other hand, classic non-oscillatory HNNs use larger blocks such as amplifiers (8 transistors) and buffers (4 transistors).

One of the main challenges in using emerging devices is cycle-to-cycle and device-to-device variability. Process variations are caused by the immature fabrication technology of these devices (Niu et al., 2010; Maffezzoni et al., 2015; Chaudhuri and Chakrabarty, 2018; Zhu et al., 2020). Although an accurate control of the fabrication process reduces variability, unavoidable variations degrade the performance. Considering Eq. (1), the behavior of oscillators relies on the coupling function $w_{ij}.g_{ij}(x_i,x_j)$, as well as the oscillator function $f_i(x_i)$. Therefore, the effect of mismatches can be considered for coupling circuits and differential oscillatory neuron circuits, separately. In this regard, we have done preliminary statistical simulations including perturbations in the memristance values and the VO$_2$ device parameters. Simulation results show that the FoM is larger than 90% when the mismatch relative sigma σ of the memristance is less than 20%. However, the operation of the DONN fails when $\sigma \geq 20\%$. On the other hand, mismatches in the threshold voltages of the VO$_2$ devices (V_H *and* V_L) are much more critical. In order to have a proper operation of DONNs, absolute sigma should be in the range of 5-10 mV (relative sigma $\sigma < 0.5\%$). Consequently, device manufacturers should be aware that when VO$_2$ devices are massively fabricated on a chip, mismatches should have such low sigmas for the threshold voltages.

CONCLUSION

A fully connected DONN hardware implementation is proposed comprising VO$_2$-based differential oscillatory neurons and memristor-bridge circuits. From the architectural viewpoint, it is like a counterpart of the traditional non-oscillatory Hopfield neural network, used as an associative memory and, as such, the standard Hebbian rule is used to train it. With regard to hardware, two emerging devices are used for the hardware implementation. VO$_2$ devices are used to implement the differential oscillatory neurons, and a memristor-bridge circuit with adjustable resistance is used to implement the coupling functions. Finally, the design of the hardware neural network is explained and evaluated using criteria that show the proper operation of the resulting DONNs in terms of synchronization, stability, and pattern retrieval.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/**Supplementary Material**, further inquiries can be directed to the corresponding author/s.

AUTHOR CONTRIBUTIONS

BL-B and TS-G conceived the initial circuit, which was further developed by JS and MA. JS performed all simulations and circuit analyses. BL-B, MA, and TS-G supervised the work jointly. JS wrote the initial draft of the manuscript which was supervised by all co-authors. All authors contributed to the article and approved the submitted version.

FUNDING

This work was funded by the EU H2020 grant 871501 "NeurONN," and the Spanish Ministry of Economy and Competitivity grant PID2019-105556GB-C31 (NANOMIND) (with support from the European Regional Development Fund).

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fnins.2021.674567/full#supplementary-material

REFERENCES

Acebrón, J. A., Bonilla, L. L., Pérez Vicente, C. J., Ritort, F., and Spigler, R. (2005). The Kuramoto model: a simple paradigm for synchronization phenomena. *Rev. Mod. Phys.* 77, 137–185. doi: 10.1103/RevModPhys.77.137

Adhikari, S. P., Yang, C., Kim, H., and Chua, L. O. (2012). Memristor bridge synapse-based neural network and its learning. *IEEE Trans. Neural Networks Learn. Syst.* 23, 1426–1435. doi: 10.1109/TNNLS.2012.2204770

Alibart, F., Zamanidoost, E., and Strukov, D. B. (2013). Pattern classification by memristive crossbar circuits using ex situ and in situ training. *Nat. Commun.* 4:2072. doi: 10.1038/ncomms3072

Ascoli, A., Corinto, F., Senger, V., and Tetzlaff, R. (2013). Memristor model comparison. *IEEE Circuits Syst. Mag.* 13, 89–105. doi: 10.1109/MCAS.2013.2256272

Ascoli, A., Demirkol, A. S., Tetzlaff, R., Slesazeck, S., Mikolajick, T., and Chua, L. O. (2021). On local activity and edge of chaos in a namlab memristor. *Front. Neurosci.* 15:651452. doi: 10.3389/fnins.2021.651452

Cai, F., Kumar, S., Van Vaerenbergh, T., Sheng, X., Liu, R., Li, C., et al. (2020). Power-efficient combinatorial optimization using intrinsic noise in memristor Hopfield neural networks. *Nat. Electron.* 3, 409–418. doi: 10.1038/s41928-020-0436-6

Camuñas-Mesa, L., Linares-Barranco, B., and Serrano-Gotarredona, T. (2019). Neuromorphic spiking neural networks and their memristor-cmos hardware implementations. *Materials (Basel)* 12:2745. doi: 10.3390/ma12172745

Chaudhuri, A., and Chakrabarty, K. (2018). "Analysis of process variations, defects, and design-induced coupling in memristors," in *Proceedings of the 2018 IEEE International Test Conference (ITC) (IEEE)*, (Phoenix AZ), 1–10. doi: 10.1109/TEST.2018.8624819

Corti, E., Cornejo Jimenez, J. A., Niang, K. M., Robertson, J., Moselund, K. E., Gotsmann, B., et al. (2021). Coupled VO2 oscillators circuit as analog first layer filter in convolutional neural networks. *Front. Neurosci.* 15:628254. doi: 10.3389/fnins.2021.628254

Corti, E., Gotsmann, B., Moselund, K., Ionescu, A. M., Robertson, J., and Karg, S. (2020a). Scaled resistively-coupled VO2 oscillators for neuromorphic computing. *Solid. State. Electron.* 168, 107729. doi: 10.1016/j.sse.2019.107729

Corti, E., Gotsmann, B., Moselund, K., Stolichnov, I., Ionescu, A., and Karg, S. (2018). "Resistive coupled VO2 oscillators for image recognition," in *Proceedings of the 2018 IEEE International Conference on Rebooting Computing (ICRC) (IEEE)*, (McLean VA), 1–7. doi: 10.1109/ICRC.2018.8638626

Corti, E., Khanna, A., Niang, K., Robertson, J., Moselund, K., Gotsamnn, B., et al. (2020b). Time-delay encoded image recognition in a network of resistively-coupled VO2 on Si oscillators. *IEEE Electron. Device Lett.* 41, 629–632. doi: 10.1109/led.2020.2972006

Gupta, N., Singh, S. S., and Stopfer, M. (2016). Oscillatory integration windows in neurons. *Nat. Commun.* 7:13808. doi: 10.1038/ncomms13808

Hadiyawarman, Budiman, F., Hernowo, D. G. O., Pandey, R. R., and Tanaka, H. (2018). Recent progress on fabrication of memristor and transistor-based neuromorphic devices for high signal processing speed with low power consumption. *Jpn. J. Appl. Phys.* 57:03EA06. doi: 10.7567/JJAP.57.03EA06

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci.U.S.A.* 79, 2554–2558. doi: 10.1073/pnas.79.8.2554

Hoppensteadt, F. C., and Izhikevich, E. M. (2000). Pattern recognition via synchronization in phase-locked loop neural networks. *IEEE Trans. Neural Networks* 11, 734–738. doi: 10.1109/72.846744

Hu, S. G., Liu, Y., Liu, Z., Chen, T. P., Wang, J. J., Yu, Q., et al. (2015). Associative memory realized by a reconfigurable memristive Hopfield neural network. *Nat. Commun.* 6:7522. doi: 10.1038/ncomms8522

Izhikevich, E. M. (1997). "Summary for policymakers," in *Climate Change 2013 - The Physical Science Basis, Intergovernmental Panel on Climate Change*, eds T.F. Stocker, D. Qin, and G.-K Plattner, M. Tignor, S.-K Allen, J. Doschung, et al. (Cambridge: Cambridge University Press), 1–30. doi: 10.1017/CBO9781107415324.004.

Jackson, T. C., Shi, R., Sharma, A. A., Bain, J. A., Weldon, J. A., and Pileggi, L. (2016). Implementing delay insensitive oscillatory neural networks using CMOS and emerging technology. *Analog Integr. Circuits Signal Process.* 89, 619–629. doi: 10.1007/s10470-016-0803-4

Jain, A. K., Mao J., and Mohiuddin, K. M. (1996). Artificial neural networks: a tutorial. *Computer (Long. Beach. Calif).* 29, 31–44. doi: 10.1109/2.485891

Krestinskaya, O., Irmanova, A., and James, A. P. (2020). "Memristors: properties, models, materials," in *Modeling and Optimization in Science and Technologies*, eds A. James (Cham: Springer International Publishing), 13–40. doi: 10.1007/978-3-030-14524-8_2

La Barbera, S., Vuillaume, D., and Alibart, F. (2015). Filamentary switching: synaptic plasticity through device volatility. *ACS Nano* 9, 941–949. doi: 10.1021/nn506735m

Li, Y., Wang, Z., Midya, R., Xia, Q., and Yang, J. J. (2018). Review of memristor devices in neuromorphic computing: materials sciences and device challenges. *J. Phys. D. Appl. Phys.* 51:503002. doi: 10.1088/1361-6463/aade3f

Maffezzoni, P., Daniel, L., Shukla, N., Datta, S., and Raychowdhury, A. (2015). Modeling and simulation of vanadium dioxide relaxation oscillators. *IEEE Trans. Circuits Syst. I Regul. Pap.* 62, 2207–2215. doi: 10.1109/TCSI.2015.2452332

Min, S.-Y., and Cho, W.-J. (2021). Resistive switching characteristic improvement in a single-walled carbon nanotube random network embedded hydrogen silsesquioxane thin films for flexible memristors. *Int. J. Mol. Sci.* 22:3390. doi: 10.3390/ijms22073390

Molahasani Majdabadi, M., Shamsi, J., and Baradaran Shokouhi, S. (2020). Hybrid CMOS/memristor crossbar structure for implementing hopfield neural network. *Analog Integr. Circuits Signal Process.* 107, 249–261. doi: 10.1007/s10470-020-01720-y

Nakao, H. (2016). Phase reduction approach to synchronisation of nonlinear oscillators. *Contemp. Phys.* 57, 188–214. doi: 10.1080/00107514.2015.1094987

Niu, D., Chen, Y., Xu, C., and Xie, Y. (2010). "Impact of process variations on emerging memristor," in *Proceedings of the 47th Design Automation Conference on - DAC '10*, (New York, NY: ACM Press), 877. doi: 10.1145/1837274.1837495

Ohno, T., Hasegawa, T., Tsuruoka, T., Terabe, K., Gimzewski, J. K., and Aono, M. (2011). Short-term plasticity and long-term potentiation mimicked in single inorganic synapses. *Nat. Mater.* 10, 591–595. doi: 10.1038/nmat3054

Parihar, A., Shukla, N., Datta, S., and Raychowdhury, A. (2015). Synchronization of pairwise-coupled, identical, relaxation oscillators based on metal-insulator phase transition devices: A model study. *J. Appl. Phys.* 117:054902. doi: 10.1063/1.4906783

Paugam-Moisy, H., and Bohte, S. (2012). "Computing with spiking neuron networks," in *Handbook of Natural Computing*, (Berlin: Springer), 335–376. doi: 10.1007/978-3-540-92910-9_10

Raychowdhury, A., Parihar, A., Smith, G. H., Narayanan, V., Csaba, G., Jerry, M., et al. (2019). Computing with networks of oscillatory dynamical systems. *Proc. IEEE* 107, 73–89. doi: 10.1109/JPROC.2018.2878854

Shamsi, J., Amirsoleimani, A., Mirzakuchaki, S., and Ahmadi, M. (2017). Modular neuron comprises of memristor-based synapse. *Neural Comput. Appl.* 28, 1–11. doi: 10.1007/s00521-015-2047-0

Shamsi, J., Mohammadi, K., and Shokouhi, S. B. (2018). A hardware architecture for columnar-organized memory based on cmos neuron and memristor crossbar arrays. *IEEE Trans. Very Large Scale Integr. Syst.* 26, 2795–2805. doi: 10.1109/TVLSI.2018.2815025

Shukla, N., Parihar, A., Freeman, E., Paik, H., Stone, G., Narayanan, V., et al. (2015). Synchronized charge oscillations in correlated electron systems. *Sci. Rep.* 4:4964. doi: 10.1038/srep04964

Stankovski, T., Ticcinelli, V., McClintock, P. V. E., and Stefanovska, A. (2015). Coupling functions in networks of oscillators. *New J. Phys.* 17:035002. doi: 10.1088/1367-2630/17/3/035002

Strukov, D. B., Snider, G. S., Stewart, D. R., and Williams, R. S. (2008). The missing memristor found. *Nature* 453, 80–83. doi: 10.1038/nature06932

Sung, C., Hwang, H., and Yoo, I. K. (2018). Perspective: a review on memristive hardware for neuromorphic computation. *J. Appl. Phys.* 124:151903. doi: 10.1063/1.5037835

van den Hurk, J., Linn, E., Zhang, H., Waser, R., and Valov, I. (2014). Volatile resistance states in electrochemical metallization cells enabling non-destructive readout of complementary resistive switches. *Nanotechnology* 25:425202. doi: 10.1088/0957-4484/25/42/425202

Velichko, A., Belyaev, M., Putrolaynen, V., Pergament, A., and Perminov, V. (2017). Switching dynamics of single and coupled VO 2 -based oscillators as elements of neural networks. *Int. J. Mod. Phys. B* 31:1650261. doi: 10.1142/S0217979216502611

Wang, Z., Joshi, S., Savel'ev, S. E., Jiang, H., Midya, R., Lin, P., et al. (2017). Memristors with diffusive dynamics as synaptic emulators for neuromorphic computing. *Nat. Mater.* 16, 101–108. doi: 10.1038/nmat4756

Weiher, M., Herzig, M., Tetzlaff, R., Ascoli, A., Mikolajick, T., and Slesazeck, S. (2021). Improved Vertex Coloring With NbO? Memristor-Based Oscillatory Networks. *IEEE Trans. Circuits Syst. I Regul. Pap.* 68, 2082–2095. doi: 10.1109/TCSI.2021.3061973

Yi, W., Tsang, K. K., Lam, S. K., Bai, X., Crowell, J. A., and Flores, E. A. (2018). Biological plausibility and stochasticity in scalable VO2 active memristor neurons. *Nat. Commun.* 9:4661. doi: 10.1038/s41467-018-07052-w

Zhu, Y., Zhang, G. L., Wang, T., Li, B., Shi, Y., Ho, T.-Y., et al. (2020). "Statistical Training for neuromorphic computing using memristor-based crossbars considering process variations and noise," in *Proceedings of the 2020 Design, Automation & Test in Europe Conference & Exhibition (DATE) (IEEE)*, (Grenoble), 1590–1593. doi: 10.23919/DATE48585.2020.9116244

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.