

Article

Analog Circuit Design of a Synapse with Supervised Learning Capabilities for Spiking Neural Networks

Alejandro Juarez-Lora ^{1*} , Victor H. Ponce-Ponce ^{1*} , Humberto Sossa-Azuela ¹ , Osvaldo Espinosa-Sosa ¹  and Elsa Rubio-Espino ¹ 

¹ Instituto Politécnico Nacional, Centro de Investigación en Computación, Av. Juan de Dios Bátiz s/n, Alcaldía GAM, Ciudad de México, 07700, México.

* Correspondence: jjuarezl2020@cic.ipn.mx (A.J.); vponce@cic.ipn.mx (V.P.)

Abstract: In this article, we propose a circuit to imitate the behavior of a Reward-modulated Spike-Timing-Dependent Plasticity synapse (R-STDP). When two neurons in adjacent layers produce spikes, each spike modifies the thickness of the common synapse. As a result, the synapse's ability to conduct impulses is controlled, leading to an unsupervised learning rule. Introducing a reward signal enables reinforcement learning by redirecting the growth and shrinkage of synapses based on feedback from the environment. The proposed synapse manages the convolution of the emitted spike signals to promote either the strengthening or weakening of the signal, which is represented as the resistance value of a memristor device. As memristors have a conductance range that may differ from the available current input range of typical CMOS neuron designs, the synapse circuit can be adjusted to regulate the spike's amplitude current to comply with the neuron. The circuit described in this work allows for the implementation of fully interconnected layers of neuron analog circuits. This is achieved by having each synapse conform to the spike signal, thus removing the burden of providing enough power from the neurons. The synapse circuit was tested using CMOS analog neurons described in the literature. Additionally, the article provides insight into how to properly describe the hysteresis behavior of the memristor in Verilog-A code. The testing and learning capabilities of the synapse circuit are demonstrated in simulation using the SKY-130nm process. The article's main goal is to provide the basic building blocks for Deep Neural Neural Networks relying on spiking neurons and memristors as the basic processing elements to handle spike generation, propagation, and synaptic plasticity.

Keywords: Spiking Neural Networks, Analog Computing, Memristor, Crossbar Arrays, Signal Processing

Citation: Juarez-Lora, A.; Ponce-Ponce, V.H.; Sossa-Azuela H.; Espinosa-Sosa, O.; Rubio-Espino, E. Analog Circuit Design of a Synapse with Supervised Learning Capabilities for Spiking Neural Networks. *Mathematics* **2024**, *1*, 0. <https://doi.org/>

Received:

Revised:

Accepted:

Published:

Copyright: © 2024 by the authors. Submitted to *Mathematics* for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Neural networks are mathematical models that can approximate functions. They work by adjusting the strengths of connections between neurons, called synaptic weights, based on the difference between the actual output and the desired output. This difference is called the error function, and it helps the network learn. There are different learning rules used in different contexts, such as control functions in control theory or policies in machine learning. Reinforcement learning (RL) methodologies have been found to be useful in tasks where scarcely available reward signals are provided, or well, the exact relationship between the system's state vector s_t , the current action, a_t , and the reward signal is not clearly mapped into a function (i.e., a model-free system). Generative Adversarial Networks involve two neural networks that compete with each other for content generation. The goal of the first net is to generate new content (i.e., images, audio) indistinguishable from training data. The second network assesses the effectiveness of the first one by assigning a score to be maximized. DDPG, TD3, and Soft Actor-Critic neural architectures are advanced control algorithms that use two, three, and even four neural networks working together to produce the best results in control tasks. These algorithms are particularly useful in

situations where modeling the system and creating a proper policy is difficult. However, the training process can be computationally expensive, and conventional Von Neumann architectures are not optimal for this task. This is because the storage and processing units are separate from each other, and additional circuitry is required to feed the processor with the necessary data. Spiking Neural Networks (SNN) attempt to replicate the cognitive mechanisms of biological brains by simulating the dynamics of neurons and synapses. This involves encoding and decoding information as spiking activity. Neuromorphic computing aims to create hardware that mimics this neuronal model, in order to achieve energy-efficient hardware with high throughput, embedded learning capabilities, and low energy consumption. The circuit implementation can be in digital or analog domain. Digital neuromorphic computing involves developing digital hardware that can solve the differential equations of SNN as quickly as possible. Examples of this type of hardware include Intel's Loihi and IBM's TrueNorth. This technology has already shown promising results in terms of power efficiency and is a research platform that is compatible with current digital technologies. In order to quantize or binarize a signal, Digital to Analog Converters (DACs) and Analog to Digital Converters (ADCs) are used. However, the use of these converters always results in a quantization error, as larger binary words require larger DACs and ADCs. This implies that a greater number of quantization levels would lead to a larger quantization error.

Besides, storing information in binary form requires hardware for each bit, which results in large circuit implementations. However, working entirely in the analog domain eliminates the quantization problem by treating values as currents in a circuit. This approach allows for the implementation of neurons in analog counterparts, synapses with memristors, and additional circuitry in crossbar arrays. Using Kirchoff's laws, values can be added instantaneously.

The conductance in each memristor enables in-memory computing and suppresses the Von Neumann bottleneck. Using SNN models to assemble RL architectures can be counterproductive when executed on typical CPUs and GPUs. However, the same models can lead to high-performance and low-energy implementations if executed on neuromorphic devices, especially if they are analog. However, as circuit analog design can be a challenging and iterative process, most frameworks/libraries or available tools for SNN are implemented in current technologies. For instance, Nest, SNN Torch, and Nengo are software libraries that deploy SNN quite easily but are executed into current CPUs and GPUs. NengoFPGA is a Nengo extension to compile the network architecture into FPGA devices, which results into a digital neuromorphic hardware implementation. Designing circuit analog for SNN can be a demanding and iterative process. Therefore, most available tools and frameworks for SNN are currently implemented using existing technologies. For example, software libraries like Nest, SNN Torch, and Nengo make it easy to deploy SNN, but they are executed on current CPUs and GPUs. NengoFPGA is an extension of Nengo that compiles the network architecture into FPGA devices, resulting in a digital neuromorphic hardware implementation. Intel's Lava is a compiler used to upload software-modeled SNN into Loihi. Both extensions, referred to as frameworks, result in digital neuromorphic implementations that are more efficient than running on Von Neumann architectures. However, they are still digital. At [1], a population encoding framework for SNN is presented that is purely in the analog domain. The framework uses bandpass filters to evenly distribute input signals into input current for analog neurons. However, storage and learning are not included in this framework. At [2], a Trainable Analog Block (TAB) is proposed that only considers encoding of signals. Information storage is left outside the computing unit as synaptic values are computed in an offline manner and stored as binary words. To the best of our knowledge, no end-to-end analog neuromorphic framework is available that includes encoding, learning, and decoding in purely analog blocks.

This article presents a novel reward signal synapse circuit designed in a 130nm technological node to enable supervised learning into Analog SNN circuits. The proposed structure enables a reward signal to switch between potentiation/depreciation of the

synapse and spike reformation in order to be able to implement into a $n \times m$ fully interconnected layers without having loss of power into the spikes, and also current decoupling, to supply the proper amount of current to the receptor neurons. Section 2 explains the modelling of SNN and the RSTDP learning rule dynamics implementation in the synapse circuit. Section 3 describes the implementation of the memristor model in Verilog-A and the synapse circuit. Section 4 describes the neuron CMOS model used to test the synapse. A 2×1 neuron network structure is tested in simulation, demonstrating adequate learning capabilities. Section 5 consists of discussion, conclusions, and future work.

2. Preliminars

2.1. Spiking Neural Networks

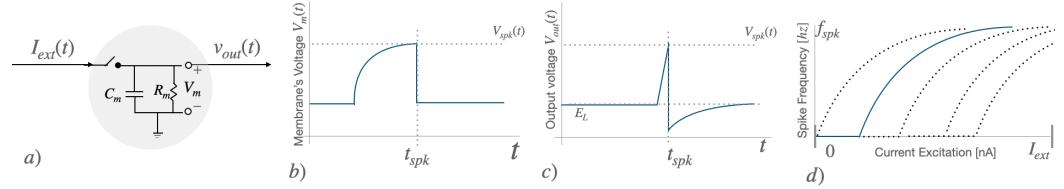


Figure 1. Leaky Integrate and Fire Model. The resulting tunning curve can be modified by changing C_m and R_m values

The behavior of biological brains, including the interactions between synapses and neurons, can be mathematically modeled and recreated using equivalent circuitry. One such example is the biological neuron, which has various models that range from biologically plausible but computationally expensive (such as the Hodgkin and Huxley model) to simplified yet reasonably complex models. The Leaky Integrate and Fire (LIF) neuron model simplifies neuron dynamics by approximating the neuron's membrane as a switched low-pass filter.

$$\tau_m \frac{dv_m(t)}{dt} = E_L - v_m(t) + R_m I_{ext}(t) \quad (1)$$

$$v_{out} = \begin{cases} v_{spk} \cdot \delta(t), & \forall v_m = v_{th} \\ 0, & \forall v_m < v_{th} \end{cases} \quad (2)$$

Here, $v_m(t)$ represents the membrane's voltage, which has certain membrane's resistance R_m and capacitance C_m . The temporal charging constant of the neuron $\tau_m = R_m C_m$ imposes a charging/discharging rate as a function of an input excitation current I_{ext} , starting from a resting potential E_L . When v_m overpasses certain threshold voltage v_{th} , the neuron emits a spike of amplitude v_{spk} , being $\delta(t)$ the Dirac delta function. As described at [3], by solving the differential equation in the time interval it takes to the neuron to $v_m = v_{th}$ and considering the frequency definition, a function which relates I_{ext} with the output spiking frequency can be obtained as:

$$f_{spk}(I_{ext}) = \frac{1}{\tau_m \ln\left(\frac{v_{th}-E_L-R_m I_{ext}}{R_m I_{ext}}\right)} \quad (3)$$

The resulting graph is called *tuning curve* [2] and depicts the sensibility of the neurons against an excitatory signal. By varying C_m , R_m and E_L , different tuning curves, i.e. spike responses can be obtained for neurons in the same layer. (See Fig) This heterogeneity can be used for encoding input signals into spiking activity, by letting that neurons in the same layer have different spike responses for the same input signal.

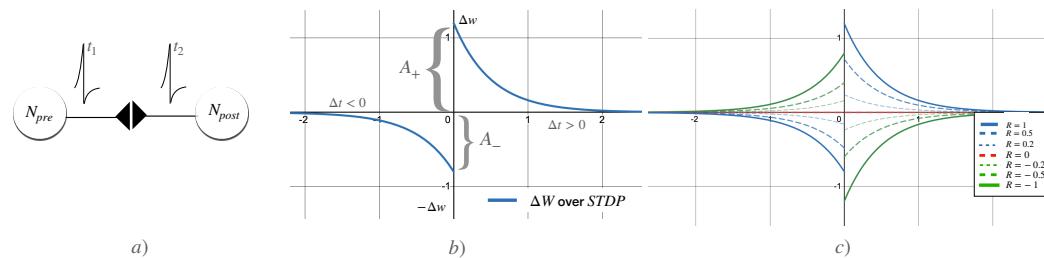


Figure 2. (a) A presynaptic and a postsynaptic neuron emits spikes, producing Hebbian Learning (b) Spike Time Dependant Plasticity (STDP) graph, which models the rate of growth ($\Delta w > 0$) or shrinkage ($\Delta w < 0$) of the synaptic weight. (c) By introducing a reward signal $R \in [-1, 1]$, the same spikes that produce potentiation LTP may produce now depreciation LTD. The response curve is show with different values of R

2.2. R-STDP Learning Rule

Spike-timing-dependent plasticity (STDP) describes *hebbian learning* as neurons that fire together, wire together. Given a couple of neurons interconnected through a synapse, the ability to conduct the spikes is depicted by a synaptic weight $w \in [w_{min}, w_{max}]$. The neuron that emits a spike is here denoted as the pre-synaptic neuron N_i at time t_i , making the synapse increase its conductance value by a certain differential amount Δw . A spike of current resulting from the convolution of the spike through the synapse is produced and fed into a receptor post-synaptic neuron N_j . Each spike contributes to the membrane's voltage of N_j until it emits a spike at time t_j , becoming then the pre-synaptic neuron. Δw is then defined as

$$\Delta w(\Delta t) = \begin{cases} A_+ e^{\frac{-\Delta t}{\tau_+}}, & \forall \Delta t \geq 0 \\ -A_- e^{\frac{\Delta t}{\tau_-}}, & \forall \Delta t < 0 \end{cases} \quad (4)$$

Where $\Delta t = t_i - t_j$. For each spike, the synaptic weight will be modified by a learning rate of A_+ , A_- , multiplied by an exponential decay defined by τ_+ , τ_- , respectively. As $\Delta t \rightarrow 0$, the change in the synaptic weight is bigger. Fig. XXX shows the characteristic graph of STDP, showing that for presynaptic spikes ($\Delta t \leq 0$), the synapse gets *Long Term Potentiation* (LTP), while for post-synaptic spikes (i.e. $\Delta t \geq 0$), the synapse suffers with *Long Term Depression* (LTD). The resulting plasticity rule models how the synaptic weight is modified, taking into consideration only the spiking activity. According to [REF Izhikevich,FLORIAN], a global reward signal, R , is introduced in order to model neuromodulatory signals. Setting $R \in [-1, 1]$, Eq. 4 is changed then to:

$$\Delta w(\Delta t) = \begin{cases} R \times A_+ e^{\frac{-\Delta t}{\tau_+}}, & \forall \Delta t \geq 0 \\ -R \times A_- e^{\frac{\Delta t}{\tau_-}}, & \forall \Delta t < 0 \end{cases} \quad (5)$$

Fig. XXX shows the role of the reward signal $R = -1$, inverting the role of presynaptic and postsynaptic spikes. A presynaptic spike leads to LTD, while a postsynaptic spike leads to LTP. This is the oposite as STDP (i.e. $R = 1$). Finally, please notice when $R = 0$, learning (modification of the synaptic weights) gets deactivated.

3. Materials and Methods

3.1. Memristor Device

A resistive random access memory (RRAM) device consists of a top and bottom metal electrodes (TE and BE, respectively), enclosing a metal-oxide switching layer, forming a metal insulator metal (MIM) structure.

A conductive filament starts to be formed with oxygen vacancies when current flows through the device. The distance from the tip of the filament to the opposite bottom

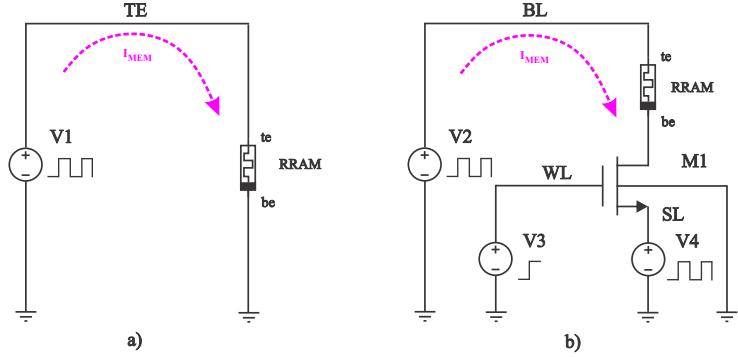


Figure 3. Memristor simulations scenarios (a) Testbench used for the first scenario, a triangular pulse $-2V$ to $2V$ signal is fed. (b) testbench used for the second scenario, where pulses are applied.

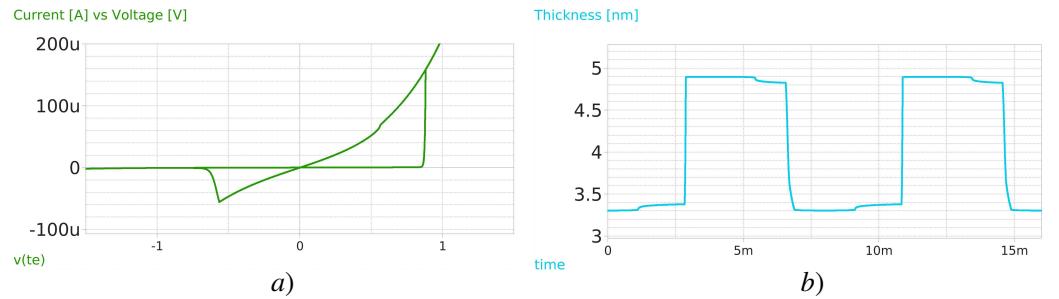


Figure 4. Memristor simulations scenarios for the first testbench (a) Lissajous Curve (I-V) of the memristor, clearly showing hysteresis and the $V_{on} \approx 0.9V$ and $V_{off} \approx -0.6$ (b) Thickness of the filament s , showing the exponential growth/shrinkage once the memristor threshold voltage is overpassed.

electrode is called gap g , which remains under a certain threshold voltage between terminals but can be modified by application of more current. Reverse current increases g while the resistance of the device increases, and vice-versa. Skywater's 130nm fabrication process, see [4], incorporates memristor cells. These cells are produced between the Metal 1 and Metal 2 layers and can be made using materials that exhibit memristive behavior, such as titanium dioxide (TiO_2), hafnium dioxide (HfO_2), or other comparable materials based on transition metal oxides.

The RRAM is able to store bits of information by switching the memristor resistance value R_{mem} between a low resistance state (LRS) and a high resistance state (HRS). However, this work's intention is to use the whole range of resistance available to store the synaptic weights by directly representing w_{min} with HRS, w_{max} with LRS and any continuous value in-between. UC Berkeley's model [5] defines the internal state of the memristor as an extra node in the tip of the formed filament. The memristor dynamics is described by the current between TE and BE electrode $i_{TE,BE}$, the rate of growth of the gap \dot{g} , and the local field enhancement factor:

$$i_{TE,BE}(t) = I_0 \cdot \exp(-g/g_0) \cdot \sinh(v_{TE,BE}/V_0) \quad (6)$$

$$\frac{d}{dt}g(t) = -\nu_0 \cdot \exp(-\frac{E_a}{V_T}) \cdot \sinh(\frac{v_{TE,BE} \cdot \gamma \cdot a_0}{t_{ox} V_T}) \quad (7)$$

$$\gamma = \gamma_0 - \beta \cdot g^3 \quad (8)$$

where $v_{TE,BE}$ is the voltage between TE and BE, t_{ox} is the thickness of the oxide separating TE and BE, a_0 is the atomic distance, and $I_0, V_0, g_0, V_T, \nu_0, \gamma_0, \beta$ are fitting parameters obtained from measurements of the manufactured memristor device [6].

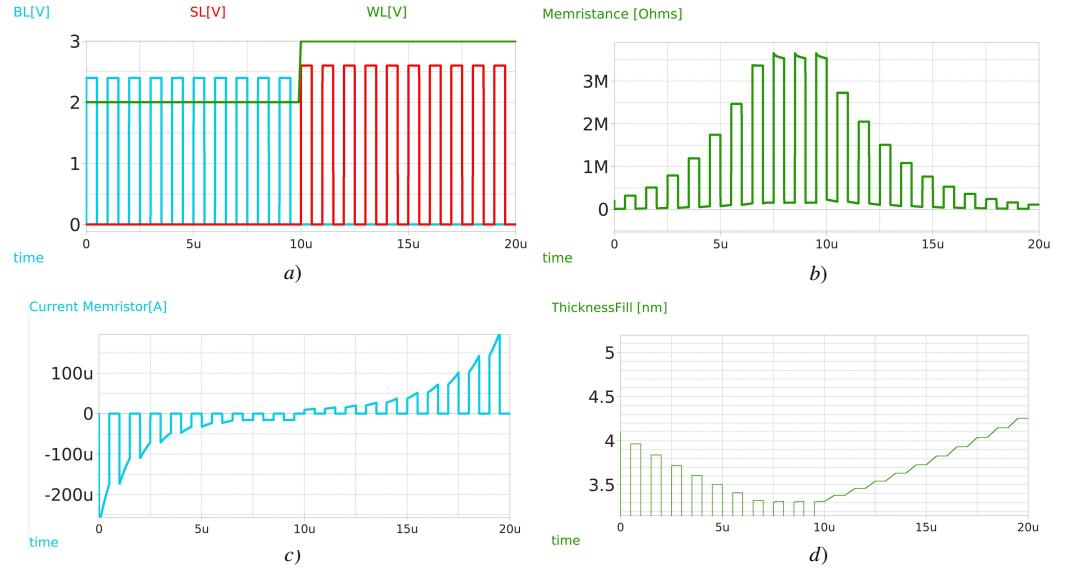


Figure 5. Memristor simulations scenarios for the second testbench (a) Voltage pulses applied in the terminals alongside time $20\mu s$ (b) Evolution of the memristance value, going from $10k\Omega$ to $3.3M\Omega$ (c) Current flowing through the memristor, considering positive current when it flows from TE to BE (a) Evolution of the thickness of the filament s

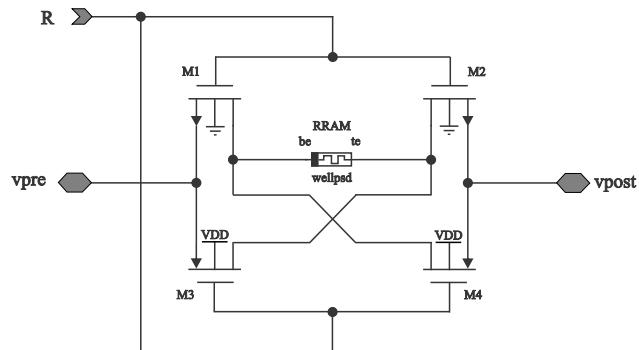


Figure 6. R-STDP hardware implementation testbench. 1M4T structure to handle current direction.

The UC Berkeley's memristor Verilog A implementation is used, but replacing the manufacturing parameters found at [7]. Fig. 3a) shows a memristor test bench where a triangular signal from $-2V$ to $2V$ is applied, resulting in the Lissajous curve (I-V graph) (Fig. 4a), with the typical hysteresis characteristics from memristors, reflecting the $V_{on} \approx 0.9V$ and $V_{off} \approx -0.6$ threshold voltages to increase/decrease the resistance in the device. Fig. 4b) shows the thickness of the filament, which lies between $4.9nm$ to $3.3nm$. On a second testbench depicted in fig. (3b), a 1T1M (one transistor-one memristor) setup where squared pulses ((4a)) are applied first at BE, then at TE to foster the resistance value exploration, reflecting the LRS and HRS values of the device, this is, $R_{mem} \in [10k\Omega, 3.3M\Omega]$ (Fig. 4b). The current flown through the memristor goes from $-200\mu A$ to $100\mu A$, matching the obtained Lissajous curve in the previous testbench (Fig. 4c). The resulting code is available at our Github repository [REF], compiled by the OpenVAF tool [8], and simulation results were obtained using the Ngspice simulation engine.

3.2. R-STDP circuit implementation

Fig. 6 shows a 4T1M configuration to replace the 1T1M set up to manage the current flow of the memristor. Proposed at [9], the structure pretends to invert the current flow according to a reward voltage signal $V_R \in [0, 1.8V]$. When $V_R = 1.8V$, transistors $M1, M2$ are enabled, and $M3, M4$ disabled. If $V_{pre} > V_{post}$, the current then will flow from BE of the

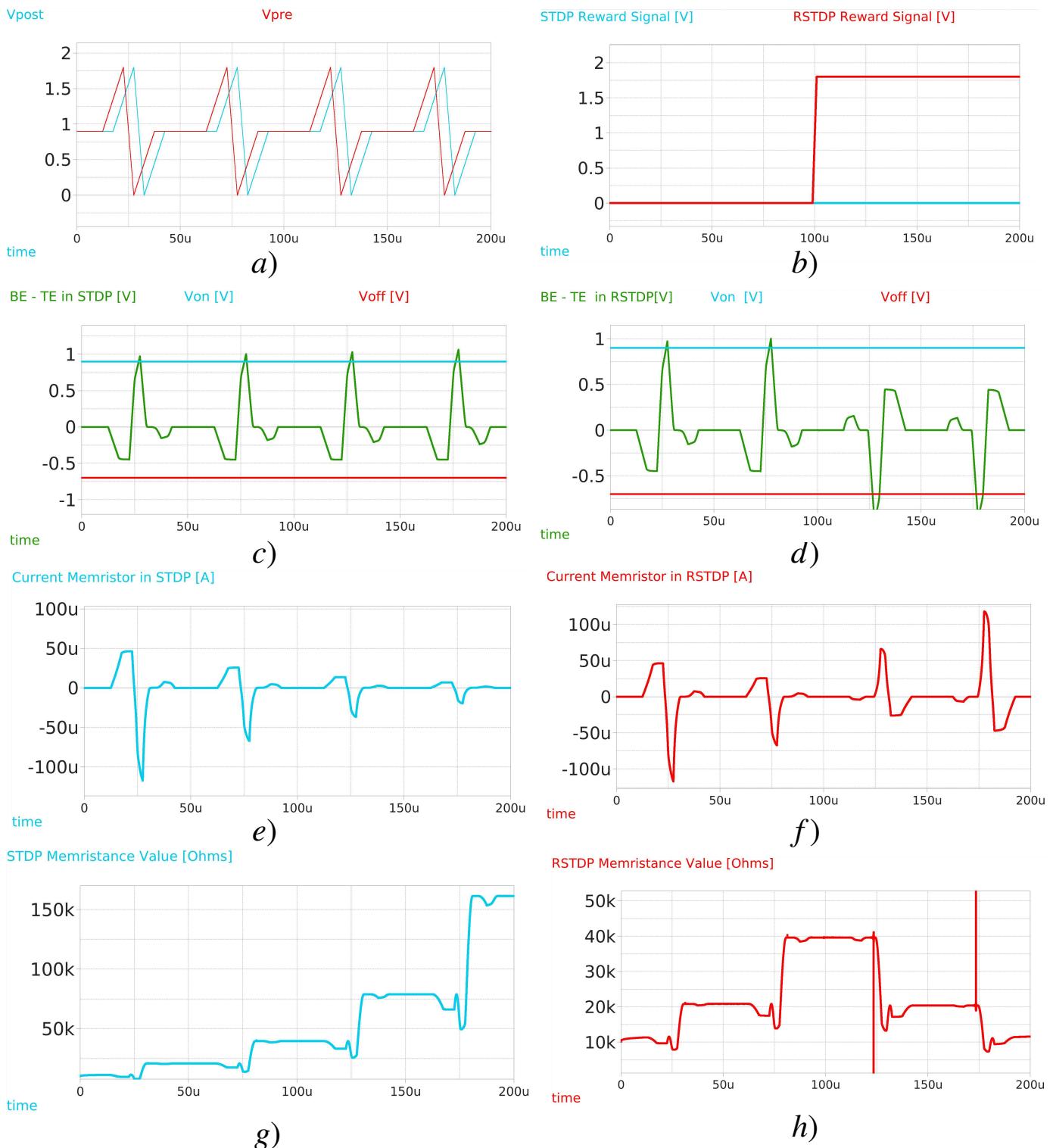


Figure 7. (b) Applied triangular pulses for V_{pre} and V_{post} , varying from $[0V, 1.8V]$. Notice that V_{post} is delayed $5\mu s$ from V_{pre} , making that $\Delta t > 0$. **(c)** Reward signal $R = [0V, 1.8V]$, to deactivate/activate drain the gate voltage in the NMOS and PMOS transistors. STDP scenario, with no reward signal (Blue) and RSTDP scenario, enabling a reward signal at the second half of the simulation **(d)** Voltage difference between TE and BE electrodes of the memristor. Notice $V_{on}(\text{Blue})$ and $V_{off}(\text{Red})$ are overpassed, yielding to a modification in the memristance **(e)** Current flowing through the memristor. **(f)** Obtained memristance values in the STDP scenario. **(g, h, i)** picture the same testbench, but activating the reward signal $R = 1.8V$, showing that current now flows in the opposite direction, yielding a reduction in memristance after $100\mu s$

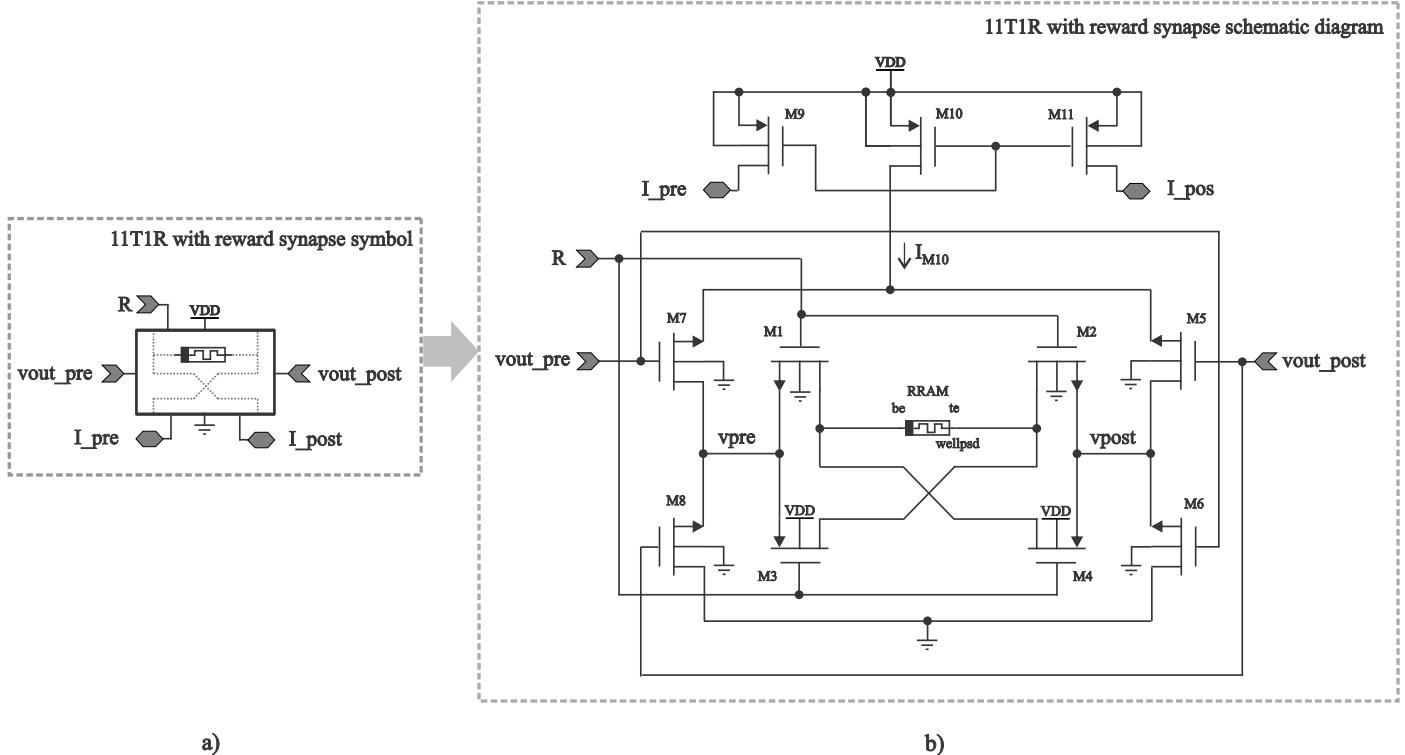


Figure 8. 11T1R Synapse circuit proposal, including the RSTDP subcircuit, the memristor, and current mirrors for the receiving neurons

memristor towards TE. However, when $V_R = 0V$, $M2, M3$ are disabled, and $M1, M4$ are enabled, yielding the current direction from TE to BE. The current direction determines whether the memristor's resistance increases or decreases. On Fig. 7a) testbench signals with triangular shapes are applied with a delay of $5\mu s$. Two scenarios are presented: In the first scenario, $R = 0$ for the entire simulation, while in the second scenario, R flips from $0V$ to $1.8V$. Notice at 7c), 7d) the voltage difference ($V_{mem} = V_{BE} - V_{TE}$) is shown for both scenarios, overpassing the memristor threshold voltage for potentiation. However, when the reward signal is flipped, V_{mem} overpasses the memristor value but in the opposite direction. Notice also at Fig. 7e), 7h) the magnitude of the spike currents the memristor delivers, given by the $M1 - M4$ geometry W/L , which in this case, where selected to provide symmetrical pulses of current. However, these aspect ratios can be selected to foster asymmetric STDP curves.

3.3. Adding Spike Reconfiguration and Current Decoupling to the synapse

Now consider two fully-interconnected neuron layers, with N and M neurons, respectively, therefore needing $N \times M$ synaptic connections. When the $i - th$ neuron of the first layer emits a spike, it should be able to provide enough power for the M post-synaptic neurons. Moreover, when the $j - th$ neuron in the second layer emits a spike, it must provide enough power to the N post-synaptic neurons.

Consider then the schematic at Fig. 8. Notice that the 4T1R R-STDP structure of the previous section is embedded inside this new 11T1R structure, supporting the switch of polarity according to the arrival of spikes. The port labeled as $V_{out,pre}$ activates transistor $M6, M7$, making $V_{pre} > V_{post}$. On the other side, the port labeled as $V_{out,post}$ enables transistors $M5, M8$, setting $V_{post} > V_{pre}$. Then, four scenarios, depicted at Fig. 9) emerge:

1. $V_{pre} > V_{post}$ $R = 1.8V$. When a presynaptic spike arrives and the reward signal is on. This routes the current from BE to TE in the memristor, yielding to LTD;
2. $V_{pre} > V_{post}$ $R = -1.8V$. Due to the reward signal being negative, the same spike train that should produce LTD now produces LTP, as the current flows from TE to BE;

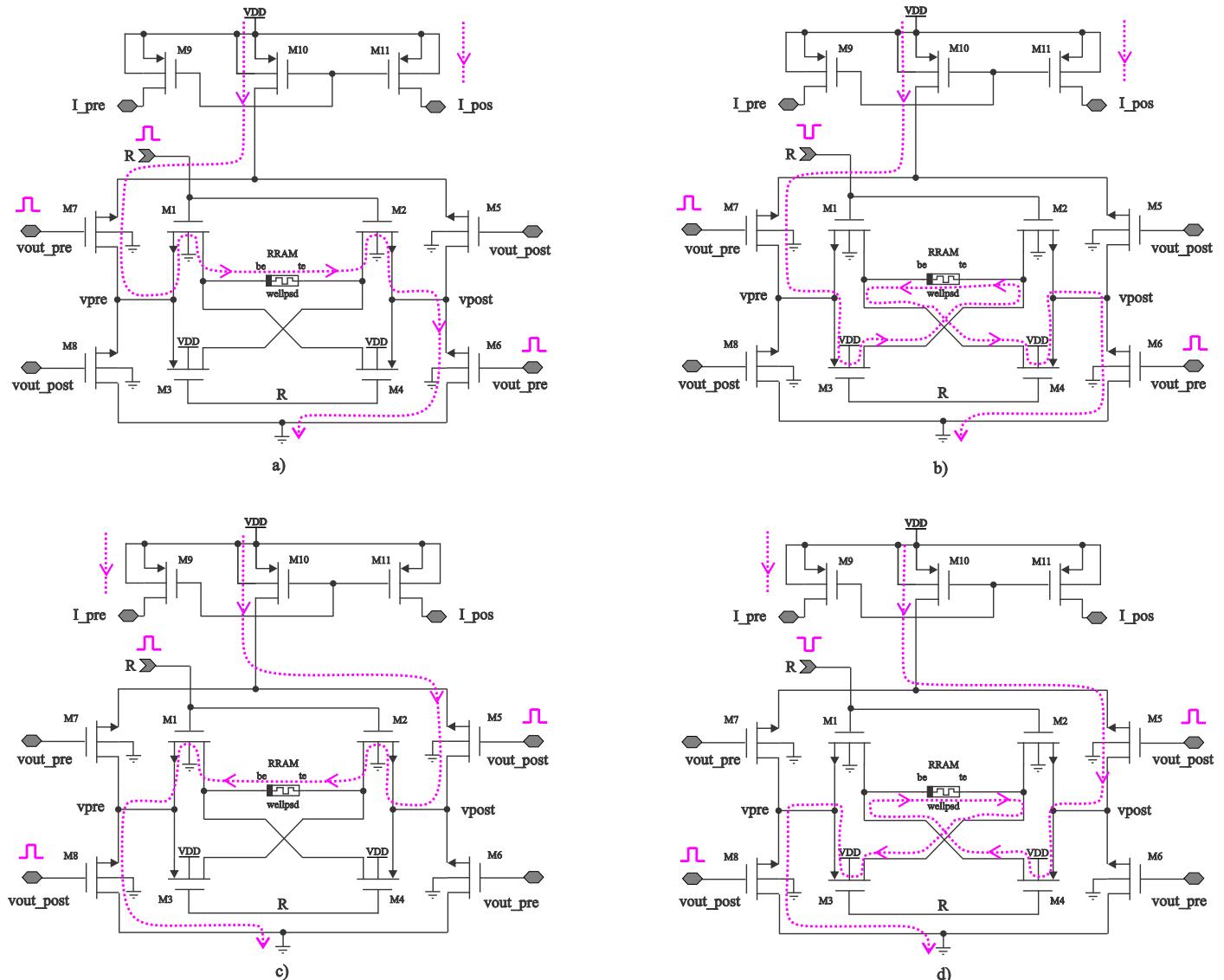


Figure 9. Proposed Synapse Circuit, reflecting the four possible scenarios for the evolution of the synaptic weight: a) $V_{pre} > V_{post}$ $R = 1.8V$ b) $V_{pre} > V_{post}$ $R = -1.8V$ c) $V_{post} > V_{pre}$ $R = 1.8V$ d) $V_{post} > V_{pre}$ $R = -1.8V$

3. $V_{post} > V_{pre}$ $R = 1.8V$. Postsynaptic spikes with reward signal on, the current flows from TE to BE, producing LTP;
4. $V_{post} > V_{pre}$ $R = -1.8V$. Postsynaptic spikes with a reward signal off, the current flows from BE to TE, producing LTD;

As the input spikes are pointing toward gates, no current is provided by the neurons. Instead, each synapse only receives the trigger signal (a spike, with amplitude larger than the threshold value of the transistors), and provides enough current straight from the power source, instead of the v_{out} node of each neuron. Regarding the upper part of the circuit. It's important to note that regardless of whether the spike was presynaptic or postsynaptic, the current that flows through transistor I_{M10} always travels from source to drain. Additionally, this current is the same that flows through the memristor, regardless of its polarity. Transistors $M9, M11$ then serve as current mirrors of I_{M10} . When the postsynaptic neuron fires, current is delivered to the presynaptic neuron by I_{pre} . Then, when a presynaptic spike arrives, I_{post} feeds the post-synaptic neuron. I_{post} and I_{pre} are defined as:

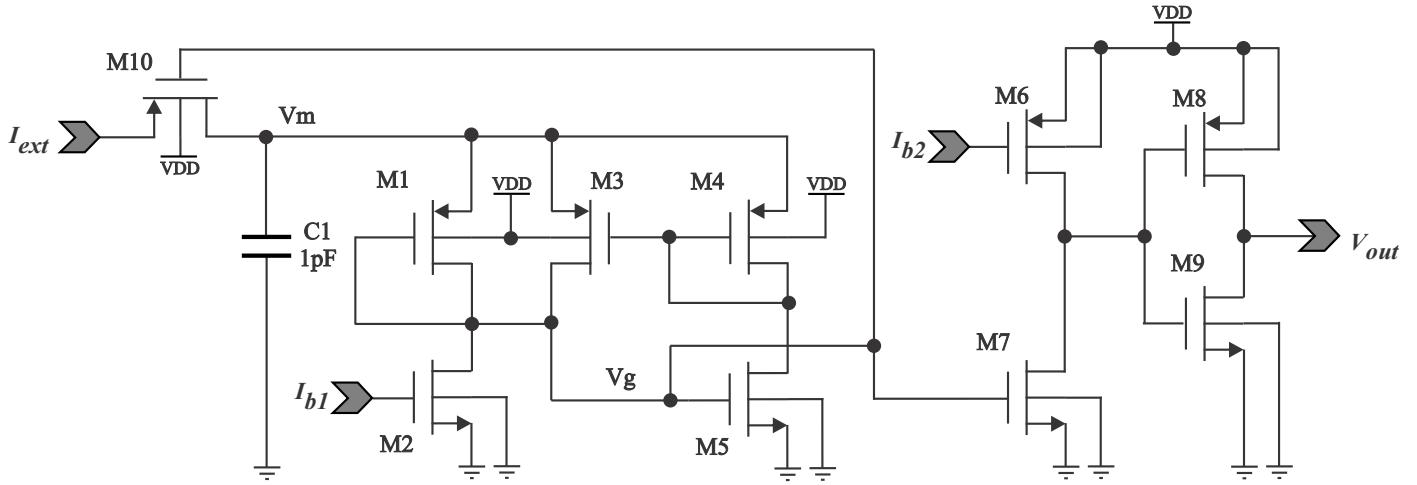


Figure 10. Analog Neuron Circuit.

$$I_{post} = \frac{(W/L)_{11}}{(W/L)_{10}} I_{M10} \quad (9)$$

$$I_{pre} = \frac{(W/L)_9}{(W/L)_{10}} I_{M10} \quad (10)$$

As mentioned in the previous sections, the resistance range of the memristor allows to provide at most $200\mu A$. However the input current range of the neuron may differ input ranges the memristor can provide for the same vdd . Therefore, Eqs. (9, 10) enable regulation of the current contribution for each spike.

3.4. Neuron circuit

Fig. (10) depicts the neuron model used in this work, based on the original design by [10], but with some modifications for the avoidance of the output spike to be fed into the same neuron, as seen as [9]. Transistors $M3, M4, M5$ emulate a thyristor with hysteresis by harnessing the fact that PMOS and NMOS have different threshold voltage values (i.e. $v_{THN} \neq v_{THP}$). The circuit dynamics can be described as follows:

- An external input current excitation I_{ext} arrives through $M10$ (PMOS), enabled at start. $M1$ is set as a diode.
- C_1 charges for each incoming spike, increasing the voltage at node V_m .
- A leaky current I_{leak} is flowing through $M2$ at all times. "If no further incoming electrical impulses are received, then the neuron will lose all of its electrical charge." V_{b1} defines I_{leak} .
- When $V_m \approx 1.5$, $V_g \approx 0.75$, which is the threshold voltage for the $M5$ NMOS device, it enables the C_1 charge to flow through $M4$ and $M5$.
- $M7$ also turns on, enabling current to flow and making voltage at V_1 drops. At the same time, $V_g > 0$, turning off transistor $M10$, disabling current integration for the neuron.
- As \bar{v}_{out} drops, v_{out} rises, as $M8 - M9$ works as an inverter. V_{b2} controls the current of the transistor $M6$, and conforming the width of the spike. The node v_{out} provides the final output spike, which can be fed to subsequent neurons.
- $M10$ acts as a controlled diode, blocking any current from I_{ext} when the neuron is spiking.

Fig. (11) shows the neuron's spiking activity for an input step signal, which rises each $50\mu s$ a step of $50nA$. It can be observed that the frequency increases as more current is added. The amplitude of the output spikes is set as $v_{out} = 1.8V$, but it can be set differently, according to the synapse needs, while the thickness of the spike is approximately $1ms$. The

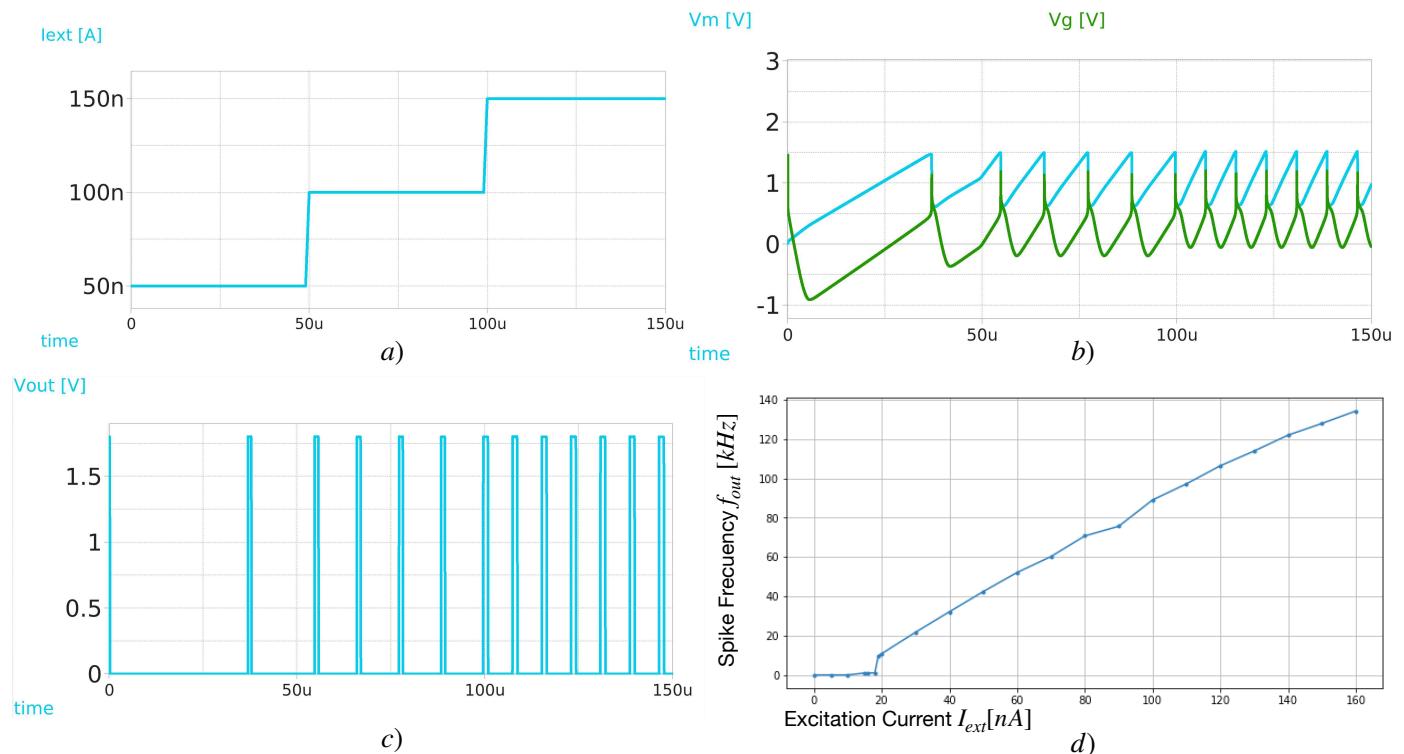


Figure 11. Simulation results for the Analog neuron Circuit(a) Testbench used to supply an step increasing current excitation I_{ext} , stargin from $50nA$, $100nA$ and $150nA$ (b). When $v_m = 1.5V$, it can be seen in c that $v_g \approx 0.75V$, turning on transistor $M5$, which acts as a charge sink for $C_1 = 1pF$ through $M2$. Fig (d) shows the spike frequency for each I_{ext} , resulting into a respectively frequency of $42kHz$, $89kHz$, $128kHz$, respectively. e) Current excitation I_{ext} against spike frequency graph, obtained by sweeping from $0nA$ to $190nA$ and obtaining the corresponding spike frequency

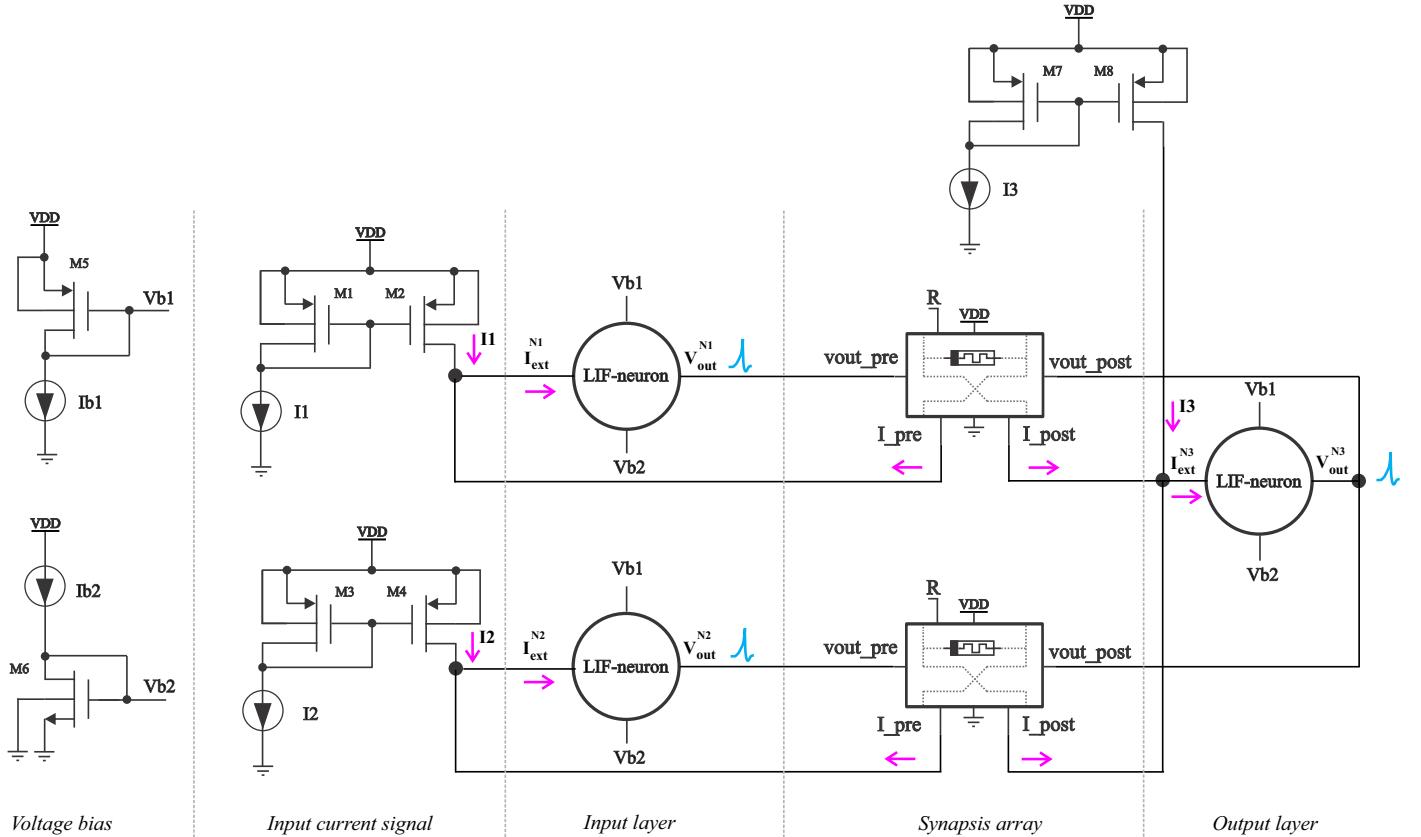


Figure 12. Testbench to test a 2-1 neuron array, with two neurons at the input layer, and one neuron at the input layer.

maximum reached spike frequency is $f_{out} \approx 140\text{kHz}$, for $I_{ext} \approx 160\text{nA}$. The neuron output voltage v_{out} remains on 1.8V for bigger input currents. The final design then needs ten transistors. However, notice $M8 - M9$ can be removed by taking into account V_g as the output voltage node. The geometry of $M5$ can be reshaped to regulate its current, defining then the width of the output spike.

4. Results

4.1. Two-to-One Neuron Simulation

Fig. (12) intends to test all the capabilities of the proposed synapse in a 2-1 network array using then two 11T1R synapses.

During the first half of the simulation, the neurons N1 and N2 in the input layer receive different excitation testing current patterns through current mirrors. This produces output spikes with different rates, as intended. In the second half of the simulation, the neuron N3 in the output layer receives an excitation current, while the current mirrors for N1 and N2 get deactivated. This should result in Long-Term Potentiation (LTP) and Long-Term Depression (LTD). However, the reward signal R is set to switch from 1.8V to -1.8V at each quarter of the simulation, leading to the four cases previously described.

Fig. (13d) shows the evolution of the thickness of the filament s . In the first quarter, the first case occurs, showing how the filament in the memristor decreases at distinct rates. Remember, as the thickness decreases, so does the conductivity. In the second quarter, the same spiking activity led to the opposite effect in the filament, as the reward signal R went from 1.8V to -1.8V , leading the thickness to 4.9nm . In the third quarter, N3 spikes, and N1, N2 ceases to receive excitation current from the current mirrors. With $R = 1.8\text{V}$, the filament should increase in size, however, as its value is already at the maximum,

260
261
262
263
264

265
266

267
268

269

270

271

272

273

274

275

276

277

278

279

280

281

282

283

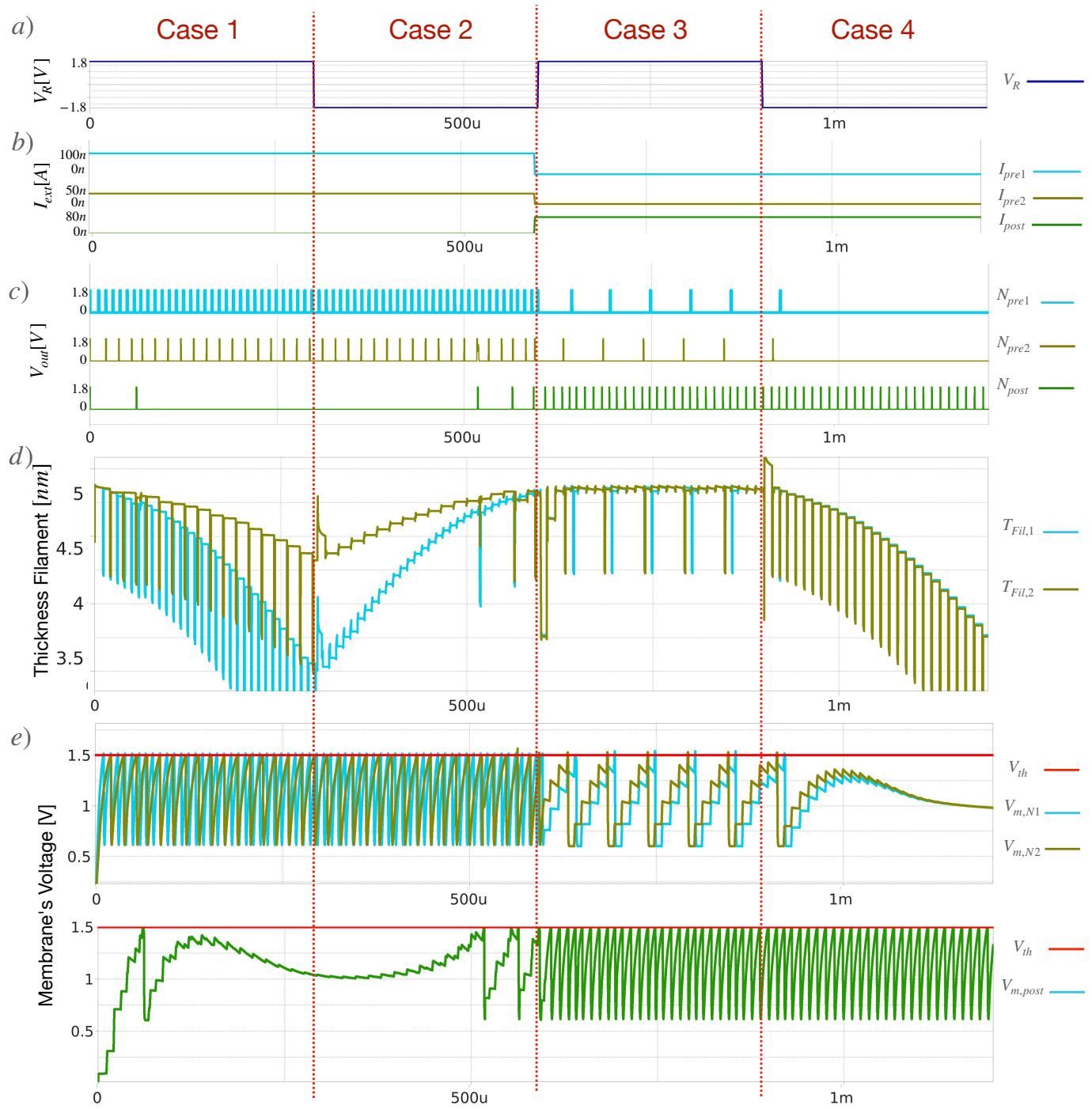


Figure 13. Simulation results for the testbench shown.

This section may be divided by subheadings. It should provide a concise and precise description of the experimental results, their interpretation, and the experimental conclusions that can be drawn.

4.2. Subsection

4.2.1. Subsubsection

Bulleted lists look like this:

- First bullet;
- Second bullet;
- Third bullet.

Numbered lists can be added as follows:

1. First item;
2. Second item;
3. Third item.

The text continues here.

4.3. Figures, Tables and Schemes

All figures and tables should be cited in the main text as Figure 14, Table 1, etc.



Figure 14. This is a figure. Schemes follow the same formatting. If there are multiple panels, they should be listed as: (a) Description of what is contained in the first panel. (b) Description of what is contained in the second panel. Figures should be placed in the main text near to the first time they are cited. A caption on a single line should be centered.

Table 1. This is a table caption. Tables should be placed in the main text near to the first time they are cited.

4T1M structure	11T1M	Neuron	2 × 1 testbench
$(W/L)_1 = 7.5/0.15$	$(W/L)_{1,2} = 7.5/0.15$	$(W/L)_1 = 1/10$	$(W/L)_{1,2} = 2/10$
$(W/L)_2 = 7.5/0.15$	$(W/L)_{3,4} = 15/0.15$	$(W/L)_2 = 1/0.15$	$(W/L)_{3,4} = 2/10$
$(W/L)_3 = 30/0.15$	$(W/L)_{5-8} = 7.5/0.15$	$(W/L)_3 = 1.5/0.15$	$(W/L)_5 = 2/10$
$(W/L)_4 = 30/0.15$	$(W/L)_{9,11} = 1/5$	$(W/L)_4 = 15/0.15$	$(W/L)_6 = 1/0.15$
	$(W/L)_{10} = 2/0.15$	$(W/L)_5 = 1/4$	$(W/L)_{7,8} = 2/10$
		$(W/L)_{6,8} = 2/0.15$	
		$(W/L)_{7,9} = 1/0.15$	

¹ Tables may have a footer.

5. Discussion

Authors should discuss the results and how they can be interpreted from the perspective of previous studies and of the working hypotheses. The findings and their implications should be discussed in the broadest context possible. Future research directions may also be highlighted.

6. Conclusions

This section is not mandatory but can be added to the manuscript if the discussion is unusually long or complex.

Author Contributions: For research articles with several authors, a short paragraph specifying their individual contributions must be provided. The following statements should be used “Conceptualization, X.X. and Y.Y.; methodology, X.X.; software, X.X.; validation, X.X., Y.Y. and Z.Z.; formal analysis, X.X.; investigation, X.X.; resources, X.X.; data curation, X.X.; writing—original draft preparation, X.X.; writing—review and editing, X.X.; visualization, X.X.; supervision, X.X.; project administration, X.X.; funding acquisition, Y.Y. All authors have read and agreed to the published version of the manuscript.”, please turn to the [CRediT taxonomy](#) for the term explanation. Authorship must be limited to those who have contributed substantially to the work reported.

Funding: The authors are thankful for the financial support of the projects to the Secretaría de Investigación y Posgrado del Instituto Politécnico Nacional with grant numbers: 20232264, 20242280, 20231622, 20240956, 20232570 and 20242742, as well as the support from Comisión de Operación y Fomento de Actividades Académicas and Consejo Nacional de Humanidades Ciencia y Tecnología (CONAHCYT).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All of the scripts used in this article are available on the following Github page: <https://github.com/XXXX> (accessed on XXX April 2024).

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

MDPI	Multidisciplinary Digital Publishing Institute
DOAJ	Directory of open access journals
TLA	Three letter acronym
LD	Linear dichroism

Appendix A

Appendix A.1

XXX

Table A1. This is a table caption.

Title 1	Title 2	Title 3
Entry 1	Data	Data
Entry 2	Data	Data

Appendix B

All appendix sections must be cited in the main text. In the appendices, Figures, Tables, etc. should be labeled, starting with “A”—e.g., Figure A1, Figure A2, etc.

References

1. Khan, S.Q.; Ghani, A.; Khurram, M. Population coding for neuromorphic hardware. *Neurocomputing* **2017**, *239*, 153–164. <https://doi.org/https://doi.org/10.1016/j.neucom.2017.02.013>. 337
2. Thakur, C.S.; Hamilton, T.J.; Wang, R.; Tapson, J.; van Schaik, A. A neuromorphic hardware framework based on population coding. In Proceedings of the 2015 International Joint Conference on Neural Networks (IJCNN), 2015, pp. 1–8. <https://doi.org/10.1109/IJCNN.2015.7280591>. 338
3. Juárez-Lora, A.; García-Sebastián, L.M.; Ponce-Ponce, V.H.; Rubio-Espino, E.; Molina-Lozano, H.; Sossa, H. Implementation of Kalman Filtering with Spiking Neural Networks. *Sensors* **2022**, *22*. <https://doi.org/10.3390/s22228845>. 342
4. Hsieh, E.; Zheng, X.; Nelson, M.; Le, B.; Wong, H.S.; Mitra, S.; Wong, S.; Giordano, M.; Hodson, B.; Levy, A.; et al. High-Density Multiple Bits-per-Cell 1T4R RRAM Array with Gradual SET/RESET and its Effectiveness for Deep Learning. In Proceedings of the 2019 IEEE International Electron Devices Meeting (IEDM). IEEE, 2019. <https://doi.org/10.1109/iedm19573.2019.8993514>. 344
5. Wang, T.; Roychowdhury, J. Well-Posed Models of Memristive Devices **2016**. 347
6. Jiang, Z.; Yu, S.; Wu, Y.; Engel, J.H.; Guan, X.; Wong, H.S.P. Verilog-A compact model for oxide-based resistive random access memory (RRAM). In Proceedings of the 2014 International Conference on Simulation of Semiconductor Processes and Devices (SISPAD), 2014, pp. 41–44. <https://doi.org/10.1109/SISPAD.2014.6931558>. 348
7. Alshaya, A.; Han, Q.; Papavassiliou, C. RRAM, Device, Model and Memory. In Proceedings of the 2022 International Conference on Microelectronics (ICM). IEEE, 2022. <https://doi.org/10.1109/icm56065.2022.10005367>. 351
8. Kuthe, P.; Muller, M.; Schroter, M. VerilogAE: An Open Source Verilog-A Compiler for Compact Model Parameter Extraction. *IEEE Journal of the Electron Devices Society* **2020**, *8*, 1416–1423. <https://doi.org/10.1109/jeds.2020.3023165>. 353
9. Tian, M.; Lu, J.; Gao, H.; Wang, H.; Yu, J.; Shi, C. A Lightweight Spiking GAN Model for Memristor-centric Silicon Circuit with On-chip Reinforcement Adversarial Learning. In Proceedings of the 2022 IEEE International Symposium on Circuits and Systems (ISCAS), 2022, pp. 3388–3392. <https://doi.org/10.1109/ISCAS48785.2022.9937639>. 355
10. Stolar, P.; Akita, I.; Schneegans, O.; Hioki, M.; Rozenberg, M.J. A spiking neuron implemented in VLSI. *Journal of Physics Communications* **2022**, *6*, 021001. <https://doi.org/10.1088/2399-6528/ac4e2a>. 358

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.