Article

A Switched Capacitor Memristor Emulator using Stochastic Computing

Carola de Benito [1,2], Oscar Camps [1], Mohamad Moner Al Chawa [3], Stavros G. Stavrinides [4] and Rodrigo Picos [1,2,*]

1 Industrial Engineering and Construction Department, Balearic Islands University (Spain); {carol.debenito, oscar.camps, rodrigo.picos}@uib.es
2 Health Institute of the Balearic Islands (Spain)
3 Technical University of Dresden (Germany); mohamad_moner.al_chawa@tu-dresden.de
4 International Hellenic University (Greece); s.stavrinides@ihu.edu.gr
* Correspondence: rodrigo.picos@uib.es
† This paper is an extended version of our paper published in de Benito, C.; Camps, O.; Al Chawa, M. M.; Stavrinides, S.G.; Picos, R. A Stochastic Switched Capacitor Memristor Emulator. 2021 10th International Conference on Modern Circuits and Systems Technologies (MOCAST). IEEE, 2021.

Abstract: Due to the increased use of memristors, and its many applications, the use of emulators has grown in parallel to avoid some of the difficulties presented by real devices such as variability and reliability. In this paper, we present a memristive emulator designed using a Switched Capacitor (SC), this is, an analog component/ block and a control part or block implemented using stochastic computing (SCo) and therefore fully digital. Our design is thus a mixed signal circuit. Memristor equations are implemented using stochastic computing to generate the control signals necessary to work with the controllable resistor implemented as switched capacitor.

Keywords: memristor; emulator; analog design; switched capacitor; stochastic computing; mixed signal

1. Introduction

Leon Chua defined the memristor theoretically in 1971 [1]. The term "memristor" is constructed from the words memory and resistor and completes the relationships provided by the capacitor, inductor and resistor between current, voltage, flux and charge. Leon Chua introduces therefore this fourth passive component to complete the set, proposing that the memristor is defined by a nonlinear relation between charge and flux. However, it was not until 2008 that it could be implemented [2]. Since then, its use has been increasing and its fields of application have also increased. It is a promising but very recent device, this implying that there are many studies that must be carried out to understand well the operating mechanisms and develop new technologies to avoid some of the problems presented, such as the variability and life time. It is because of the problems that the real device presents that the development of memristive emulators is booming.

The emulators reproduce the operating characteristics of the memristor by eliminating the aforementioned problems and allow for the development of more complex and reliable systems [3]. The memristor behavior which is imitated can be ideal memristor or actual device, depending on the implementation. If we are focused on their field of application, emulators have different characteristics, although there are two main lines of study: analog emulators and digital emulators.

Many works develop fully analog emulators, for example, in [4] a memristive system is implemented and results very easy to fabricate it in academic laboratories through classical electrical components from circuit theory, in [5], the emulator is implemented with transistors, resistors and diodes and it operates in passive mode. Other examples like [6] and [7] use amplifiers in their models. In general, analog systems need more power consumption, the volatility of the system is worse than in the digital case, but they present a good implementation of the variable resistance with which the emulator memristance is described.

Citation: de Benito, Carola; Camps, Oscar; Al Chawa, Mohamad Moner; Stavrinides, Stavros G.; Picos, Rodrigo; A Switched Capacitor Memristor Emulator using Stochastic Computing. *Preprints* **2022**, *1*, 0. https://doi.org/

Received:
Accepted:
Published:

 doi:10.20944/preprints202112.0100.v1

On the other hand, digital memristors systems emulators can be implemented in in FPGAs (or ASICS)[8–10]. Their main advantages are that they present short simulation times and better control of the emulator, however, they present a problem when implementing the variable resistance. In digital emulators it's much easier to define the model, but precision is lost (limited number of bits) and usually needs more computational power than the analog equivalent.

For many years, the scientific community has been a great interest to develop different computer architectures. It seeks, among other things to change the structure of serial calculation and perform operations in parallel. A high degree of parallelism allows for faster execution using less complex elements or using approximations even if precision is lost. Due to the large number of data and operations that must be carried out by current computer systems, the well-known Von Neumann architecture [11] is not a good election to use due to a high consumption of time of computation and energy.

One of the alternatives, like mentioned above, is to use non-deterministic computing methods, Stochastic Computing (SCo) is one of them. The use of random variables is it main characteristic. In 1956, Von Neumann introduced probabilistic logic [12], his work was based on the work of R.S. Pierce in 1952 at the California Institute of Technology [13]. The advances in the field of electronics and computing have allowed to implement it during the sixties [14,15]. Nowadays, this approach has been used in many fields like image processing [16,17], data compression[18], mathematical calculations [19,20], control [21], image processing [22], or even A/D conversion[23], among others.

In this paradigm, the representation of data is performed in a probabilistic way using boolean quantities that are switching in random way during a time. The numbers are codified in a vector of random binary numbers, whose average value will be correlated to the number represented [24]. These vectors are referred to as stochastic logic number (SLN). This representation makes possible to reduce the area occupied since complex functions occupying a large space such as multiplication can be reduced to a single logic gate with great savings in terms of power and area[25]. To create the SLN, a Random Number Generator (RNG) is needed and for the designers it is a challenge to use the lowest number of these to not increase the area of the chip. The number of RNGs is related to the fact that the operations in SCo are different depending on the encoding of the number and if the signals are correlated or not (statistical dependent or not). To guarantee uncorrelated SLNs, different RNGs must be employed to generate each stochastic signals.

In this work, we design, simulate and implement a mixed-signal memristor emulator, improving the version presented in [26]. Specifically, in this paper we improve the theoretical discussion, including the description of the stochastic blocks, and we also present some experimental results. The proposed emulator consists of two blocks, taking advantage of the best features of each design part. In the analog block a Switched Capacitor is used to implement a variable resistor, and in the digital one, that is the control block, we use stochastic computation. The simulation is done with Matlab to implement the functionality of both the analog block, similar to that used in [27], and of the control block. For the experimental implementation a quadruple analog switch HCF4066FE and a DE0-Nano FPGA have been used.

This paper is organized as follows: the next section describes the generalities of memristors, memristor emulators and stochastic logic operations; in the third section the model is developed and simulated; the fourth section deals with the experimental implementation; and, finally, the last section discusses the work.

2. Theoretical Background

2.1. Memristor Mathematical Description

A memristor is a two terminal device whose resistance (conductance) can change its value when a voltage or current signal is applied. In addition, the value of the resistance (conductance) of the device also depends on its past history and is named memristance (M) (memconductance (G)). The concept of the memristor was extended by Chua in 1976

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 December 2021 doi:10.20944/preprints202112.0100.v1

to memristive systems to explain the behavior of observed systems [28], for instance, in nature. Nowadays the classification of memristors includes ideal, generic and extended memristor [29].

The most general class is the extended memristor, and include the others. The dynamic of this class is described using internal variables that determine the internal state of the memristor, these variables, can be for example, temperature or geometrical parameters depending on the system. The memristor can be voltage or current controlled depending of the input source. On the other hand, in [30] Corinto et al. proposed a mathematical description in the charge flux domain instead voltage and current domain. We use for our emulator the equations describing a voltage-controlled extended memristor in the charge flux domain, these are:

$$i = G(\phi, v, \mathbf{x}) \cdot v \tag{1}$$

$$\frac{d\mathbf{x}}{dt} = \mathbf{g}(\phi, v, \mathbf{x}) \tag{2}$$

$$\frac{d\phi}{dt} = v \tag{3}$$

The memconductance (G), which can be nonlinear, is the inverse of the memristance of the device, v is the applied voltage, i is the current, ϕ is the voltage first momentum or flux and $\mathbf{x}$ represents the extra state variables.

Finally, it is also important to mention that the memristors present some characteristic fingerprints distinguishing those of others dynamic systems [30,31]:

1. As Leon Chua noted in [32]: "If it's NOT pinched, it's NOT a memristor". The *i-v* curve obtained when a periodic signal with zero DC component (voltage or current) is applied to the memristor shows a pinched (at the (v=0,i=0) point) hysteresis loop.
2. The area of the hysteresis loop should tend to zero for higher frequencies, as noted in [30]. The behavior at low frequency depends on the specifics of the memristor, and it may even exist a frequency where the loop area is maximum [3].

On the other hand, the emulator function must be to mimic the memristor behavior, this is to show it fingerprints. The emulator can be implemented in analog, digital or mixed. It is crucial that the circuit implements, among others, the internal state variables, (vector $\mathbf{x}$) in Eq. (1) and (2). These internal variables must be included as electrical variables in the emulator and cannot interact with the outside and they are the used, along with the voltage and the flux, to determine the actual value of the equivalent memconductance (G) or memristance (M).

2.2. Stochastic Logic Operations

Four ways are used to encode numbers in SCo: Unsigned Classical Stochastic Encoding (UCSE), Signed Classical Stochastic Encoding (SCSE), Unsigned Extended Stochastic Encoding (UESE) and Signed Extended Stochastic Encoding (SESE) [33,34]. Depending of the encoding used, the functions will need to be implemented using different simple logical blocks. For instance, when using UCSE, an AND gate is used to implement the product of two inputs, while a XNOR gate performs that operation when using the SCSE encoding, as shown in Figure 1. Moreover, the same logical gate can perform different operations depending on whether the random signals generated are correlated or not. As an example, with USCE encoding, a two inputs AND gate is used to implement a multiplication for uncorrelated inputs, while it provides the minimum value of the two inputs it they are correlated. In the present work, we will be using SCSE and, thus, our numbers will lay in the real [-1..1] domain. Thus, multiplication requires the use of an XNOR gate.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 December 2021 doi:10.20944/preprints202112.0100.v1

Figure 1. Basic implementation scheme of a SC multiplier in the [0..1] range (AND gate, left) and in the [-1..1] range (XNOR gate, right).

Performing an addition is slightly more complex, because numbers with a probability higher than one cannot be represented, and we may need to add 1+1=2. Thus, it is better to implement an alternate form as $(x+y)/2$, which at most will output a value of 1. This operation is usually implemented using a multiplexer, as shown in Figure 2 (a), where the p(0.5) means a signal with a probability of 0.5 to be '1' or '0'. This signal can be generated using one of the bits from the RNG, thus needing no additional circuitry. The same gate is used for the [0..1] and the [-1..1] domains.

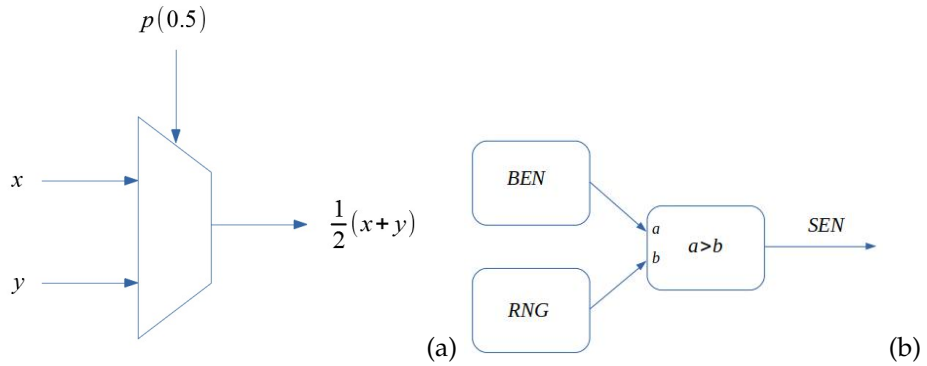


Figure 2. Basic implementation schemes (a) of a SC adder using a multiplexer and (b) a Stochastic Number Generator (SGN) which converts a Binary Encoded Number (BEN) to a Stochastic Encoded Number (SEN), using a Random Number Generator (RNG).

Other more complex operations (division[21], square roots[21], etc...) may also be found in the literature, but are not presented here for the sake of clarity. Finally, the conversion of a number encoded as a classical number can be translated to a stochastic representation using the schema presented in Fig. 2.

3. Memristor emulator design

3.1. Theoretical Design

As mentioned above, our system has been implemented using an analog block and a digital one. Figure 3 shows the analog block, which is a typical switched capacitor module (SC). The equivalent resistance R_{eq} is described by Eq. (4). In this case, both control external signals S_1 and S_2 are equal, with a lag of 180 deg. [35].

$$R_{eq} = \frac{1}{f_S C}\frac{1+e^{\left(\frac{D\,T}{\tau}\right)}}{1-e^{\left(\frac{D\,T}{\tau}\right)}} \tag{4}$$

T and f_s are the period and the frequency of the controlling signal S_1 and S_2, D is the duty cycle with values between 0 and 1, C is the capacitance value and τ is the time constant, this is, $R_{tot}C$ where C is the capacitor and R_{tot} is the total resistance of the circuit taking into account the parasitic ones.

For our design, in the charge flux domain, it is necessary to calculate the flux from the voltage of the terminals of the SC as a first step. Once this is done, then the relations between flux and charge are used to obtain the duty cycle (D) that varies the equivalent resistance of the SC. The digital block is the responsible for all these steps.

For this purpose, a series of approximations shall be done to Eq. (4). The conductance (G) ($G = 1/R_{eq}$) can be rewritten as:

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 December 2021 doi:10.20944/preprints202112.0100.v1

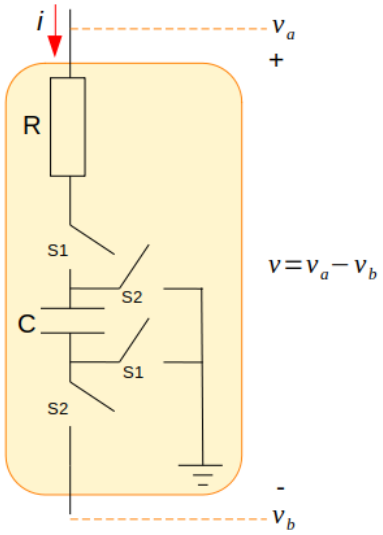


Figure 3. Switched capacitor (SC) circuit schematic.

$$G = f_S C \frac{e^{(-\frac{x}{2})} - e^{(\frac{x}{2})}}{e^{(-\frac{x}{2})} + e^{(\frac{x}{2})}} = f_S \cdot C \cdot tanh(\frac{x}{2}) \tag{5}$$

where x=DT/τ.

A first order Maclaurin expansion of $tanh(x/2)$ can be used, taking into account that the control signals used must be faster than the time the system takes to decay, so the final equation for the conductance G is:

$$G = f_S C \frac{DT}{2\tau} \tag{6}$$

It is important to notice that this last equation implies that conductance is linearly dependent on the duty cycle D.

To calculate the flux, the digital block converts previously each voltage terminal of the SC (v_a and v_b in Fig.3), to non correlated random values. Then, the corresponding value $v_a - v_b$ is accumulated into a counter, which acts as the integrator. Notice that since we are using stochastic computing this up/down counter needs to count only one up ($v_a > v_b$), one down ($v_a < v_b$), or remain the same ($v_a = v_b$). To implement the memristor device, it is necessary to use an equation to describe the relation between flux and charge. In this work the simplest relation is used:

$$Q = M\phi^2 \tag{7}$$

where M is a constant. This equation does not include any internal variables. Applying the firth derivative of the equation, the conductance is:

$$i = 2M\phi \frac{d\phi}{dt} = 2M\phi v \implies G = 2M\phi \tag{8}$$

Matching the Eq. (6) and the Eq. (8) the relation between the duty cycle and flux is:

$$D = \frac{4M\tau}{f_S CT}\phi = K\phi \tag{9}$$

K is therefore a constant value.

To control the analog block, it is the SC, the duty cycle (D) must be used. The duty cycle is calculated by the digital block from ϕ according to Eq. 9 as a stochastic value. To use it the average value of D is calculated to determine R_{eq} with Eq. 4.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 December 2021 doi:10.20944/preprints202112.0100.v1

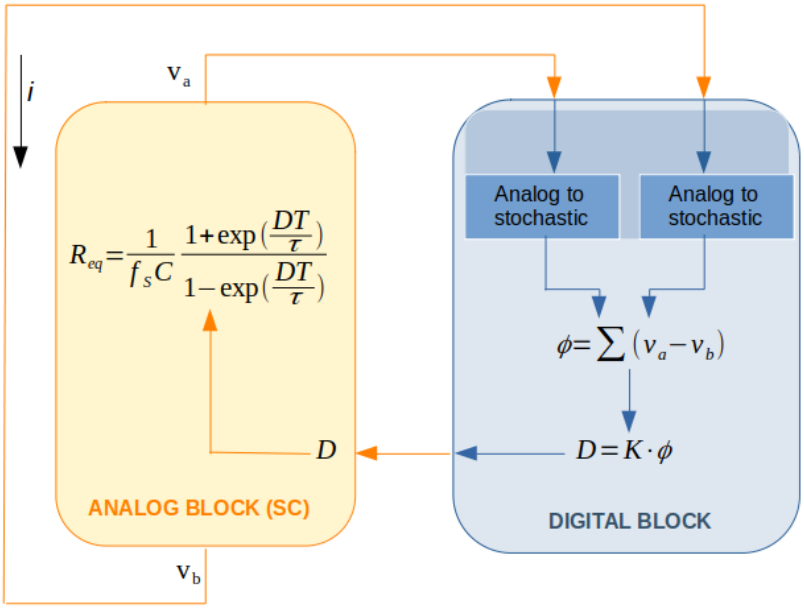


Figure 4. Swiched Capacitor Memristor Emulator (SCME) block diagram.

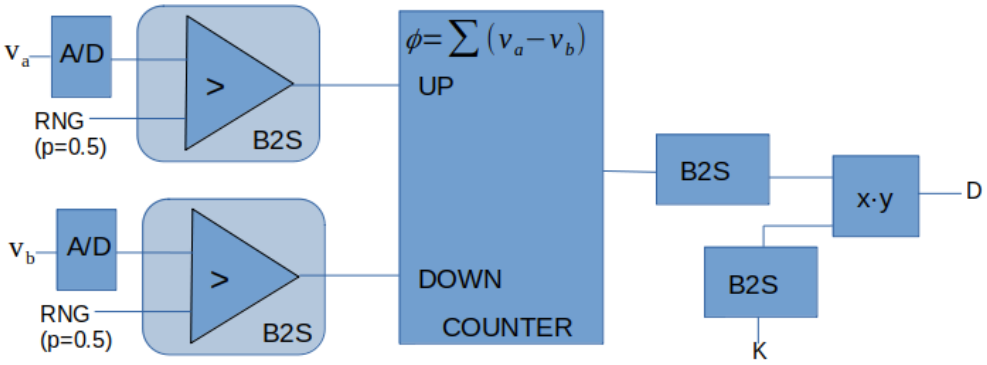


Figure 5. Control Block implementation using Stochastic Computing.

The emulator block design scheme including the two parts of the design, analog and digital, is shown in Fig. 4. The part corresponding to the digital block implemented in Stochastic Computing is shown as a circuit in Fig. 5.

3.2. Simulation Results

In order to be considered as a memristor, the emulator must present two characteristic fingerprints [3,31,32]: (1) a pinched loop, (2) whose area changes with frequency.

Figure 6 presents the $i - v$ of the emulator under inputs of different frequency, in arbitrary units. It is apparent from this figure that the curves are pinched at the origin and that the loop area changes with frequency. Thus, we can consider that the two fingerprints are present.

Because of the way it is constructed, the emulator reaches a saturation for the conductance. This is due to the maximum value of $D = 1$, and can be clearly seen at low frequencies, where the maximum value of flux is reached faster. This may also be seen in Fig. 7, where the behavior of the Q versus ϕ near the origin is quadratic, as can be expected from (7), but it is also seen that its behaviour changes to lineal after a maximum value for $D = 1$ is reached.

Finally, the current signal for different frequencies is shown in Fig. 8. As can be seen

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 December 2021 doi:10.20944/preprints202112.0100.v1

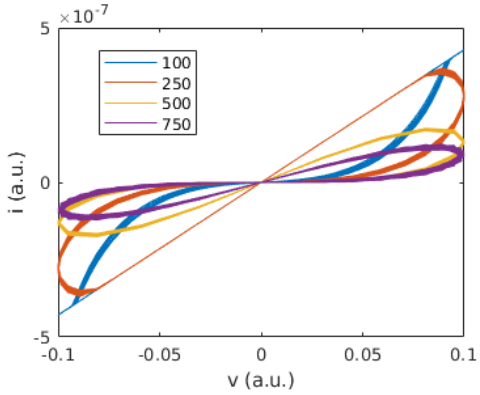


Figure 6. $i - v$ characteristic curve of the memristor implemented using Fig. 4. The different frequencies (in arbitrary units) are shown in different colors.

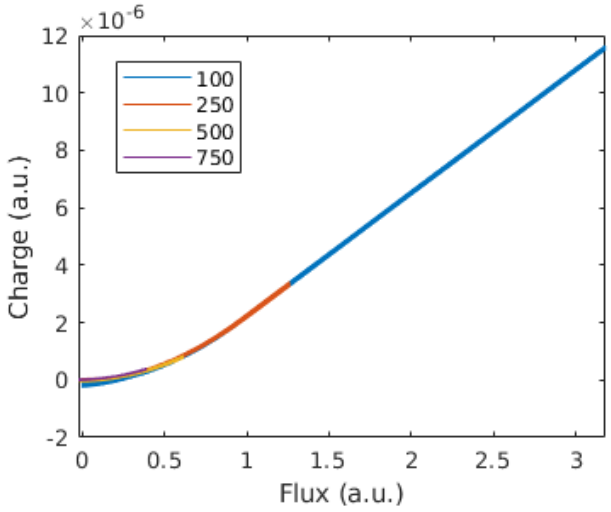


Figure 7. $Q - \phi$ characteristic of the memristor implemented using Fig. 4. The different frequencies (in arbitrary units) are shown in different colors.

there, the maximum conductance (related to the maximum value of the current) is lower for higher values of frequency, as expected.

4. Experimental Implementation

4.1. Experimental Setup

In order to test the proposed circuit, we have implemented using a setup similar to that of [27]. We have used a quadruple analog switch HCF4066FE, driven by a DE0-Nano FPGA. The analog switch has a working voltage between -0.5 V to 22 V, and can switch at a maximum frequency of 25 kHz, when the power supply is 3.3V. The FPGA generated the control signals S_1 and S_2, using two of its 3.3V digital output pins. Additionally, we have also used a 1kΩ shunt resistor, along with a 15μF capacitor. The implemented circuit is shown in Fig. 9. The conversion from analog to stochastic was performed by first converting from analog to digital using two of the on-board available A/D and then converting this digital value into stochastic, as described above.

An AFG320 arbitrary signal generator was used to generate the input signal, while two oscilloscopes where used to monitor the full system. An oscilloscope was monitoring the control signals of the HCF4066FE, while the other oscilloscope was used to monitor the voltage through the shunt resistance of 1kΩ to obtain the current, and also the input voltage.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 December 2021 doi:10.20944/preprints202112.0100.v1

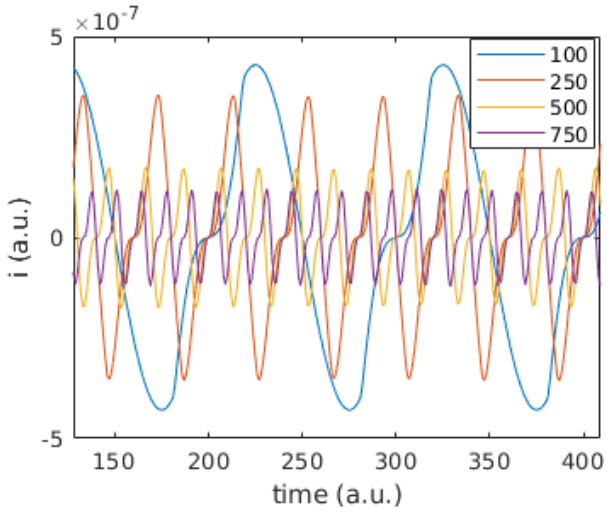


Figure 8. Current signal for different frequencies. The different frequencies (in arbitrary units) are shown in different colors.

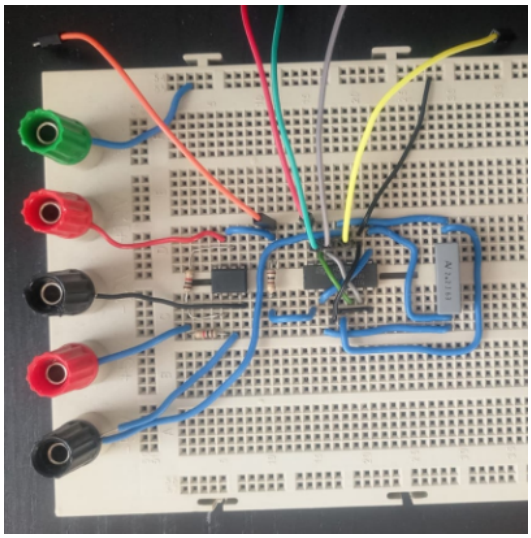


Figure 9. Physical implementation of the circuit on a prototyping board.

4.2. Experimental Results

The system has been tested using different input frequencies: 100, 200 and 400 Hz. The internal behaviour of the circuit is depicted in Fig. 10, that shows the waveform corresponding to the most significant bit of the counter and the control signal S_1 in one of these cases. It has to be noted that S_2 was generated to be exactly S_1 negated and is not shown.

The temporal behavior of the current in these three cases is shown in Fig. 11. The currents are clearly non linear because of memory: if they were nonlinear due to other effects, then they would be symmetrical, which they are not. In addition, they are showing a dependence on the frequency, as expected for a memristor.

The I-V loops are depicted in Fig 12, and they show the experimental fingerprint of a memristor: the pinched loop [32]. That means that the device has a resistive behavior (it's pinched, which means no current when no voltage is applied), and that this resistance has a memory effect (there is a loop, which means that there are two possible values of the resistance and, hence, the current, for each voltage input value).

It has to be noted that the area of the loop changes with frequency, with the higher area corresponding to the higher frequency. This behaviour is caused by the saturation of the internal counter that corresponds to the flux integral (eq. 3 and Fig. 5), which leads to a

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 December 2021 doi:10.20944/preprints202112.0100.v1

Figure 10. Stochastic signals generated by the control circuit. The upper signal is the most significant bit of the counter, while the lower waveform is the S_1 control signal.

linear behaviour once the maximum value is reached.

5. Discussion

In this work we have presented a flux controlled memristor emulator based on the extended memristor equations. We use a switched capacitor as the controlled resistor, this being the analog part of the design. The digital control is implemented using stochastic computing and its purpose is to calculate the flux and generate the control signal for the switched capacitor block. That is, we generate a stochastic signal whose average value is the duty cycle. This signal has been generated by using a simple nonlinear relation between flux and charge, that leads to a linear relation between the duty cycle and the flux. The emulator is thus a mixed emulator using the best properties of analog and digital emulators, showing excellent capabilities and a very easy implementation.

Author Contributions: Conceptualization, C.B. and R.P.; methodology, C.B. and R.P.; software, C.B., S.G.S. and R.P.; validation, C.B., M.A. and R.P.; formal analysis, C.B. and R.P.; investigation, C.B., O.C., M.A. and R.P.; resources, C.B. and R.P.; data curation, C.B. and R.P.; writing—original draft preparation, C.B., O.C., M.A., S.S. and R.P.; writing—review and editing, C.B., O.C., M.A., S.G.S. and R.P.; visualization, C.B., O.C., M.A. and R.P.; supervision, C.B., S.G.S and R.P.; project administration, C.B. and R.P.; funding acquisition, C.B. and R.P. All authors have read and agreed to the published version of the manuscript.

Funding: Some of the authors wish to acknowledge support from DPI2017-86610-P, TEC2017-84877-R projects, awarded by the MICINN and also with partial support by the FEDER program.

Institutional Review Board Statement: Not applicable.

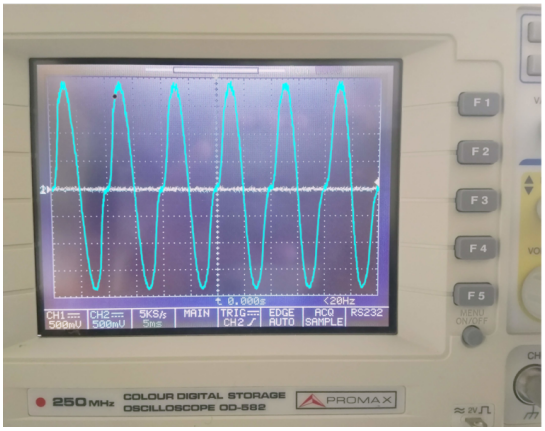
Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

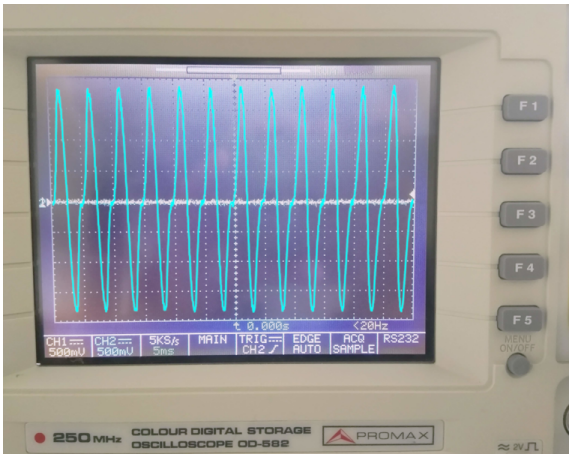
References

1. Chua, L. Memristor-the missing circuit element. *IEEE Transactions on circuit theory* **1971**, *18*, 507–519.
2. Strukov, D.B.; Snider, G.S.; Stewart, D.R.; Williams, R.S. The missing memristor found. *nature* **2008**, *453*, 80–83.
3. Stavrinides, S.G.; Picos, R.; Corinto, F.; Al Chawa, M.M.; de Benito, C. Implementing memristor emulators in hardware. In *Mem-elements for Neuromorphic Circuits with Artificial Intelligence Applications*; Academic Press, 2021; pp. 17–40.
4. Ascoli, A.; Corinto, F.; Tetzlaff, R. A class of versatile circuits, made up of standard electrical components, are memristors. *International Journal of Circuit Theory and Applications* **2016**, *44*, 127–146.

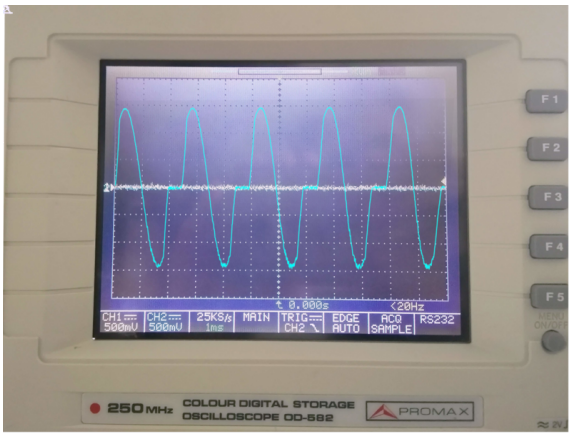
Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 December 2021 doi:10.20944/preprints202112.0100.v1



(a) 100Hz.



(b) 200Hz.

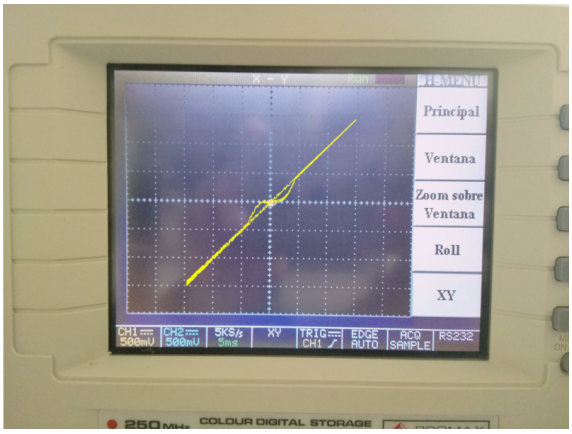


(c) 400Hz.

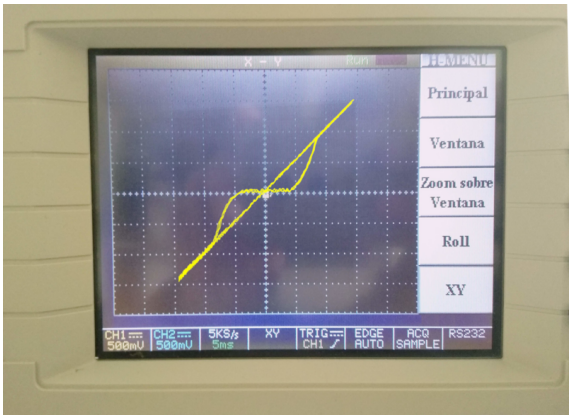
Figure 11. Measured current signals at different frequencies.

5. Kalomiros, J.; Stavrinides, S.G.; Corinto, F. A two-transistor non-ideal memristor emulator. Modern Circuits and Systems Technologies (MOCAST), 2016 5th International Conference on. IEEE, 2016, pp. 1–4.
6. Kim, H.; Sah, M.P.; Yang, C.; Cho, S.; Chua, L.O. Memristor emulator for memristor circuit applications. *IEEE Transactions on Circuits and Systems I: Regular Papers* **2012**, *59*, 2422–2431.
7. Li, Z.; Zeng, Y.; Ma, M. A novel floating memristor emulator with minimal components. *Active and Passive Electronic Components* **2017**, *2017*.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 December 2021 doi:10.20944/preprints202112.0100.v1



(a) 100Hz.



(b) 200Hz.

(c) 400Hz.

Figure 12. Measured I-V signals at different frequencies.

8. Vourkas, I.; Abusleme, A.; Ntinas, V.; Sirakoulis, G.C.; Rubio, A. A Digital Memristor Emulator for FPGA-Based Artificial Neural Networks. Verification and Security Workshop (IVSW), IEEE International. IEEE, 2016, pp. 1–4.
9. Ranjan, R.; Ponce, P.M.; Kankuppe, A.; John, B.; Saleh, L.A.; Schroeder, D.; Krautschneider, W.H. Programmable memristor emulator asic for biologically inspired memristive learning. Telecommunications and Signal Processing (TSP), 2016 39th International Conference on. IEEE, 2016, pp. 261–264.
10. Kolka, Z.; Vavra, J.; Biolkova, V.; Ascoli, A.; Tetzlaff, R.; Biolek, D. Programmable Emulator of Genuinely Floating Memristive Switching Devices. 2019 26th IEEE International Conference on Electronics, Circuits and Systems (ICECS). IEEE, 2019, pp. 217–220.
11. Von Neumann, J. First draft of a report on the EDVAC. *IEEE Annals of the History of Computing* **1993**, *15*, 27–75. doi:10.1109/85.238389.
12. Von Neumann, J. Probabilistic logics and the synthesis of reliable organisms from unreliable components. *Automata studies* **1956**, *34*, 43–98.
13. Von Neumann, J.; Pierce, R.S. *Lectures on probabilistic logics and the synthesis of reliable organisms from unreliable components*; California institute of technology, 1952.
14. Gaines, B.R. Stochastic computing. Proceedings of the April 18-20, 1967, spring joint computer conference, 1967, pp. 149–156.
15. Poppelbaum, W.; Afuso, C.; Esch, J. Stochastic computing elements and systems. Proceedings of the November 14-16, 1967, fall joint computer conference, 1967, pp. 635–644.
16. Fick, D.; Kim, G.; Wang, A.; Blaauw, D.; Sylvester, D. Mixed-signal stochastic computation demonstrated in an image sensor with integrated 2D edge detection and noise filtering. Proceedings of the IEEE 2014 Custom Integrated Circuits Conference. IEEE, 2014, pp. 1–4.
17. Camps, O.; Stavrinides, S.G.; Picos, R. Efficient Implementation of Memristor Cellular Nonlinear Networks using Stochastic Computing. 2020 European Conference on Circuit Theory and Design (ECCTD). IEEE, 2020, pp. 1–4.
18. Wang, R.; Han, J.; Cockburn, B.; Elliott, D. Stochastic circuit design and performance evaluation of vector quantization. 2015 IEEE 26th International Conference on Application-specific Systems, Architectures and Processors (ASAP). IEEE, 2015, pp. 111–115.
19. Yuan, B.; Wang, Y.; Wang, Z. Area-efficient scaling-free DFT/FFT design using stochastic computing. *IEEE Transactions on Circuits and Systems II: Express Briefs* **2016**, *63*, 1131–1135.
20. Camps, O.; Stavrinides, S.G.; Picos, R. Stochastic Computing Implementation of Chaotic Systems. *Mathematics* **2021**, *9*, 375.
21. Toral, S.; Quero, J.; Franquelo, L.G. Digital stochastic realization of complex analog controllers. *IEEE Transactions on Industrial Electronics* **2002**, *49*, 1101–1109.
22. Camps, O.; al Chawa, M.M.; Stavrinides, S.G.; Picos, R. Stochastic Computing Emulation of Memristor Cellular Nonlinear Networks. *(Preprints)* **2021**.
23. Toral, S.; Quero, J.; Ortega, J.; Franquelo, L. Stochastic A/D sigma-delta converter on FPGA. 42nd Midwest Symposium on Circuits and Systems (Cat. No. 99CH36356). IEEE, 1999, Vol. 1, pp. 35–38.
24. Toral, S.; Quero, J.; Franquelo, L. Stochastic pulse coded arithmetic. 2000 IEEE International Symposium on Circuits and Systems (ISCAS). IEEE, 2000, Vol. 1, pp. 599–602.
25. Moons, B.; Verhelst, M. Energy-efficiency and accuracy of stochastic computing circuits in emerging technologies. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* **2014**, *4*, 475–486.
26. de Benito, C.; Camps, O.; Al Chawa, M.; Stavrinides, S.; Picos, R. A Stochastic Switched Capacitor Memristor Emulator. 2021 10th International Conference on Modern Circuits and Systems Technologies (MOCAST). IEEE, 2021, pp. 1–4.
27. Svetoslavov, G.; Camps, O.; Stavrinides, S.G.; Picos, R. A Switched Capacitor Memristive Emulator. *IEEE Transactions on Circuits and Systems II: Express Briefs* **2020**.
28. Chua, L.O.; Kang, S.M. Memristive devices and systems. *Proceedings of the IEEE* **1976**, *64*, 209–223.
29. Leon, C. Everything you wish to know about memristors but are afraid to ask. *Radioengineering* **2015**, *24*, 319.
30. Corinto, F.; Civalleri, P.P.; Chua, L.O. A theoretical approach to memristor devices. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* **2015**, *5*, 123–132.
31. Biolek, D.; Biolek, Z.; Biolková, V.; Kolka, Z. Some fingerprints of ideal memristors. Circuits and Systems (ISCAS), 2013 IEEE International Symposium on. IEEE, 2013, pp. 201–204.
32. Chua, L. If it's pinched it's a memristor. *Semiconductor Science and Technology* **2014**, *29*, 104001.

33. Gaines, B.R. R68-18 random pulse machines. *IEEE Transactions on Computers* **1968**, *100*, 410–410.
34. Gaines, B.R. Stochastic computing systems. In *Advances in information systems science*; Springer, 1969; pp. 37–172.
35. Kimball, J.W.; Krein, P.T.; Cahill, K.R. Modeling of capacitor impedance in switching converters. *IEEE Power Electronics Letters* **2005**, *3*, 136–140.

 doi:10.20944/preprints202112.0100.v1