

Memristor-Based Cellular Automata for Natural Language Processing

Ioannis K. Chatzipaschalis^{*†}, Theodoros Panagiotis Chatzinikolaou*, Iosif-Angelos Fyrigos*, Andrew Adamatzky[‡], Antonio Rubio[†], and Georgios Ch. Sirakoulis*

* Department of Electrical and Computer Engineering, Democritus University of Thrace, Xanthi, Greece

[†] Department of Electronic Engineering, Universitat Politècnica de Catalunya, Barcelona, Spain

[‡] Department of Computer Science and Creative Technologies, University of the West of England, Bristol, UK

Abstract—Towards achieving efficient computational approaches, the perspective of Cellular Automata (CAs) appears to be highly potent, owing to its performance hidden in the parallel computing capabilities inherent in its local units, known as cells. Through their local interconnections and adherence to fixed or dynamic rules, these cells demonstrate exceptional capability in executing intricate physical events and addressing emergent problems that pose significant challenges for conventional computing systems. Capitalizing on the non-volatility and information processing capabilities offered by memristors, an oscillatory memristive CA circuit is proposed aiming to advance the field of neuromorphic computing and artificial intelligence applications, as well as to point out more effective and reliable computational models by bridging the gap between biological and circuit-based networks. In this manner, CAs are introduced as nanoelectronic memristor-based circuits to become a good candidate for utilizing Natural Language Processing (NLP) tasks like the forming of sentences.

Keywords—Memristor, 1T1R, Memristor-based Oscillator (MBO), Cellular Automata (CAs), Natural Language Processing (NLP), Language Generation

I. INTRODUCTION

Artificial Intelligence (AI) has been used extensively to improve the human life in important sectors such as in healthcare, but also for simplifying their everyday tasks [1]. More specifically, chatbots (computer programs designed to simulate human-like conversations through text or speech interactions), are able to interact with people using Natural Language Processing (NLP), a sub-field of AI aiming to the interaction between computers and humans. Its challenge is the interpretation and generation of human language in a way that is meaningful for humans, so a conversation can be initiated. The use of NLP is promising enough to help people make decisions faster through recommendation tasks, generate text or even solve a query regarding a problem they may face with a product or service [2]. In this way, it can save them time and money from their personal or work life. Language is a complex system with intricate patterns and dynamics. Because of that, for the implementation of NLP, researchers have came up with Machine Learning algorithms [3] or even Spiking Neural Network architectures [4] as conventional solutions. However,

This work has been partially supported by the framework of the FUNGATERIA project, which has received funding from the European Union's HORIZON-EIC-2021-PATHFINDER CHALLENGES programme under grant agreement No. 101071145, and by the project PID2022-141391OB-C22, Ministry of Science, Spain.

it is still worthy to realise NLP with an unconventional perspective such as the theory of Cellular Automata (CAs).

A cellular automaton (CA) can be considered as a computational model encapsulating a number of cells, arranged in one or multiple dimensions, each of which can have various states. The behavior of the entire CA is determined by a global rule aiming to dictate how the states of the cells evolve over time. The main characteristics of such a system include homogeneity, spatial representation, and local interactions [5]. In particular, homogeneity is attributed to the fact that all the CA cells are considered identical following the same evolving rule(s), while spatial representation of the cells relies on the exact definition of CA, and regarding the local interactions, each CA cell changes its state from a time step to another taking into account the states of its neighbors as well its previous state, introducing memory to the system. CAs have been used from pattern recognition tasks to urban growth and modeling of ecological problems [6]. Combining CAs with NLP could assist on further exploring linguistic phenomena based on language interrelations. CAs can generate language-like patterns or sequences due to their self organization and emergent characteristics.

CAs have been also proven ideal for the conceptualization of the non von Neumann modern computing architectures that aim to combine memory and processing within the same computing structure when implemented in hardware [7]. On the other hand, future and emergent nanoelectronic structures are promising to boost even-more the efficient design of CAs and their cells in hardware, and, more important, can pave the way for their successful fabrication introducing advantageous, in terms of area and performance, in-memory computing architectures. Among other candidate devices, memristor nanodevices (short for memory resistors) [8], are highly promising for the design of innovative in-memory and neuromorphic computing architectures [9]. Itoh and Chua introduced the tentative use of memristors in CAs configurations for the first time [10]. Following this first theoretical attempt to combine CAs with memristor devices, other memristor-based CA circuits have emerged for the execution of computing tasks, such as shortest route finding methods and sorting systems [11]. Memristor-based CAs has also been suggested for image processing in medical applications [12], to a unique strategy for edge detection by including the adaption of CA's neighborhood [13], and a novel memristive CA approach

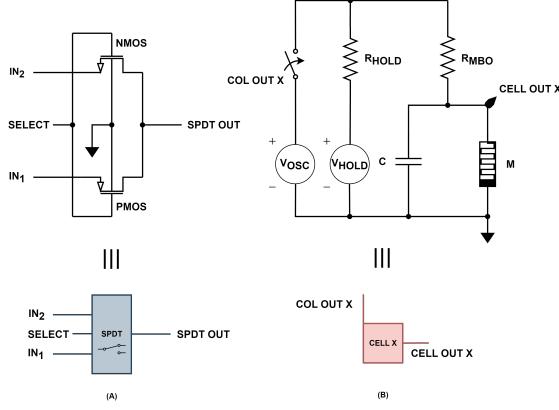


Fig. 1. (A) The SPDT circuit consisted by one NMOS and one PMOS transistor and its symbol. (B) Circuit-level representation of a CA cell with its main component to be an MBO and its symbol.

has been also introduced on the literature utilizing waves propagation as its main computing feature [14].

Because of the aforementioned capabilities, this work is focused on the design of a memristive circuit that can replicate with efficiency the main characteristics of a CA as reported previously and utilize the NLP perspective in order to form simple sentences. In Section II, the memristor model used along with the cell's circuit equivalent and the whole CA-enabled NLP circuit are presented. In the following Section III, the results of the full circuit simulations are reported for the case of sentence forming and, finally, Section IV concludes the manuscript also providing insights for future work.

II. THE CA MAPPING IN HARDWARE

CA cells can change their state due to the Memristive-based Oscillator (MBO) they incorporate that is responsible for generating oscillations, each of which defines an evolution time step. In terms of cell's connecting neighbors, memristor also plays a critical role as it is mandatory for enabling the concept of training as NLP applications demand. For the realistic integration of the memristor into the simulated circuits a compact memristor model [15] that accurately captures the behavior of unipolar Conductive-Bridging RAM (CBRAM) devices has been utilized. The model incorporates the effects of drift, diffusion, and thermo-diffusion, which are essential for simulating the switching dynamics of the memristor and it has been fitted to fabricated devices that exhibit unipolar behavior. Such devices are ideal for CAs NLP applications as they consume negligible power and have large switching speeds for faster language processing.

A. The CA cell circuit equivalent

The equivalent circuit of a CA cell is spotted in Fig. 1(B), which contains a switch connected to its neighbors, two resistors R_{HOLD} and R_{MBO} , two voltage sources V_{HOLD} and V_{OSC} , a capacitor C and a memristor M forming a MBO able to generate neuron-like oscillations. Its function is characterized by two states, namely resting and computing.

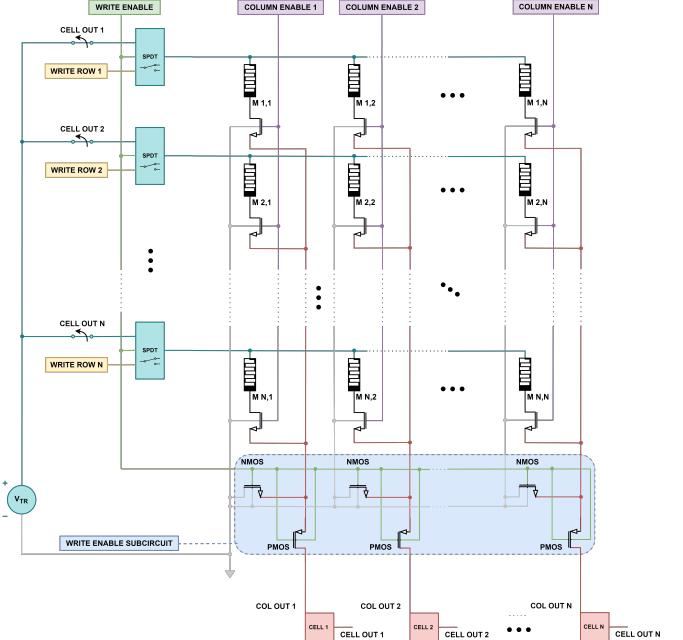


Fig. 2. The full CANLP designed circuit consisted from the crossbar, SPDT modules, the cells and the required switches.

Finally, the cell's output is considered as the MBO output on the CELL OUT X node.

Focusing on the functionality of the CA cell, this is continuously powered by the V_{HOLD} DC voltage to keep itself in the resting state taking advantage of the equilibrium point (EP), which is formed between the resistor R_{MBO} and the memristor M . More specifically, the EP can be observed on CELL OUT X, where its voltage is not sufficient to change the memristor state and the MBO cannot perform any oscillation. In order cell's rule to enable the oscillation, the necessary COL OUT X voltage is provided on the respective switch and surpass its threshold THR so as to trigger the second branch of the V_{OSC} DC voltage. This changes the voltage on the CELL OUT X enabling the memristor state switch and the oscillation of the MBO. The resistor R_{HOLD} has been properly selected to switch between the two voltage sources with the least voltage deviations due to the Millman's theorem. Lastly, it is mentioned that the resistor R_{MBO} and the capacitor C affect the output oscillation form and timing.

B. The CA Natural Language Processing (CANLP) circuit

After the design of the core part of the system, which is the CA cell, it should be incorporated to a circuit that encapsulates the NLP perspective. The CA that is utilized is a one-dimensional (1-D) CA with N cells, each of which reproduces a specific word. However, because of the learning nature of this NLP application presented in this paper, the whole circuit consists of a crossbar with 1T1R components, the CA cells, a train-enable subcircuit, SPDTs, and regular switches as presented in Fig. 2.

Regarding CA's characteristics, the neighborhood of a cell includes all the other cells as the CA is a fully connected 1-D

	I	HAVE	A	MONKEY	MY	IS	LOVELY	HIGH	SMALL	IT	CAN	JUMP	QUICKLY	LEARN
I	0	1	0	0	0	0	0	0	0	0	0	0	0	0
HAVE	0	0	1	0	0	0	0	0	0	0	0	0	0	0
A	0	0	0	1	0	0	0	0	0	0	0	0	0	0
MONKEY	0	0	0	0	0	1	0	0	0	0	0	0	0	0
MY	0	0	0	0	1	0	0	0	0	0	0	0	0	0
IS	0	0	0	0	0	0	1	0	0	0	0	0	0	0
LOVELY	0	0	0	0	0	0	0	0	0	0	0	0	0	0
HIGH	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SMALL	0	0	0	0	0	0	0	0	0	0	0	0	0	0
IT	0	0	0	0	0	0	0	0	0	0	1	0	0	0
CAN	0	0	0	0	0	0	0	0	0	0	0	1	0	1
JUMP	0	0	0	0	0	0	1	0	0	0	0	0	0	0
QUICKLY	0	0	0	0	0	0	0	0	0	0	0	0	0	0
LEARN	0	0	0	0	0	0	0	0	0	0	0	1	0	0

Fig. 3. The Reference Matrix (RM) with 0 indicating the corresponding memristors to be written in OFF state while with 1 to the ON state, respectively.

grid with trainable neighboring connections. As for the cells' states they are two as referred to the previous subsection. In the resting state, the cell waits for the signal of the previous word so that can be triggered to the next state, computing. The latter corresponds to the increase of the voltage in order to produce an oscillation as Fig. 4 depicts. In this time domain simulation figure, colored areas represent the computing states of three (3) different CA cells. During this phase, the output signal passes through the crossbar and the next word's computation is performed. After that, next word is triggered and the cycle continues. The rule responsible for the activation of the cells is described by the following Eq. 1:

$$CELL_j = \begin{cases} \text{computing state}, & \sum_{i=1}^N \frac{SPDT\ OUT_i}{M_{i,j}} < THR \\ \text{resting state}, & \text{otherwise} \end{cases} \quad (1)$$

where N is the number of the CA cells (and of the total words as well), and i is the row and j the column number in the crossbar, accordingly.

Concerning the crossbar configuration, its role is to store the word sequence of a specific train set of sentences so as to properly activate the respective word in each timestep. It has been programmed during the training phase column-by-column using the ex-situ method with external binary weights as pulses of 0.5V amplitude and 4ms duration of a Reference Matrix (RM) presented in Fig. 3. This RM is externally trained utilizing the sentences of Tab. I in such a way to represent the word connections; so that the CA's evolution to be able to reveal a word sequence forming one or multiple sentences. In this way the memristors of the crossbar can be written either in their ON or OFF state regarding the sentence(s) that is (are) going to be formed. The dictionary used, a small one as proof of concept and for readability reasons, consists of $N = 14$ words, so the RM will have a size of 14×14 . Each row represents the previous word and each column its following suitable word(s) for forming a sentence that has to train the connections of the cells in order to produce the right sequence. The rows of each sentence last word should be all filled by zeros as they do not trigger any other word afterwards, while the columns of the first words should be zeros as they are not previously connected to any word. After the training phase, the circuit shall produce the corresponding sentence(s) with triggering its (their) first word.

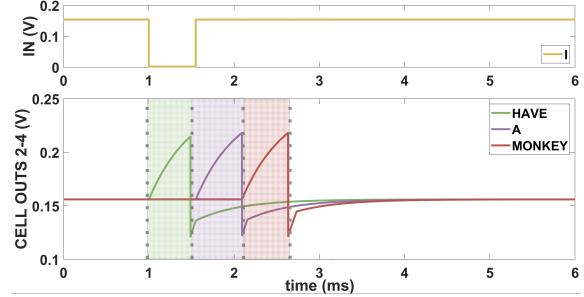


Fig. 4. Time domain simulations of the core CA circuits needed to form the sentence No. 1 of Tab. I. The activation of the corresponding cells happens after triggering the first letter "I". In each colored area the corresponding cell is in the computing state for one time step (0.55ms), and in white all are in the resting state.

The train-enable subcircuit (Fig. 2; light blue area) is consisted by two transistors a NMOS and a PMOS and it has dual role. Firstly, it grounds the crossbar during the train phase, so that memristors to be written successfully as their bottom electrode will be in zero voltage. Secondly, it isolates the CA cells while the crossbar is being written to prevent their false activation. The module that allows the training is the SPDT switches. More specifically, they take the WRITE ENABLE voltage as a select signal and they also have two different inputs, one used for writing the crossbar with ex-situ data (training phase), and the other for the evolution phase as presented in Fig. 1(A). To elaborate further, when the first input is activated, the memristors of the crossbar (which are also the connections of the CA cells) are trained following the RM topology. Otherwise, when the second input is activated, the evolution phase is enabled and each row acquires a DC voltage V_{TR} . This voltage is subject to be dropped in specific rows to trigger the corresponding cells to their computing state and continuing the sentence forming. Every row is equipped with a normally closed switch behind the SPDT module that is responsible for generating the voltage drop happens every time the cell corresponding to the previous word is in the computing state. Finally, it is worth mentioning that all the transistors used are modeled with the generic PTM model, and appear gate length equal to 180nm technology node.

III. SIMULATION RESULTS

In order to simulate the corresponding CANLP circuit presented earlier, the Cadence Virtuoso Design Suite has been utilized. A set of 5 different sentences, which include words only from a standard dictionary, is considered and the expected output is presented to determine its proper functionality. As reported in the previous Section II, the total number of words used in the simulation is 14 and they are appeared in the RM of Fig. 3. The ideal sentences that the circuit should form are mentioned in Table I.

The aforementioned sentences are separated into three distinct test cases deriving from their first word (i.e. "I", "MY", and "IT"). More specifically, in the first test case the circuit is requested to produce only the first sentence in contrast to the second and third test cases that it is going to produce pairs of

TABLE I
THE AVAILABLE SENTENCES BEING FORMED FROM THE CANLP CIRCUIT.

No.	Content of the sentence
1	I HAVE A MONKEY
2	MY MONKEY IS LOVELY
3	MY MONKEY IS SMALL
4	IT CAN JUMP HIGH
5	IT CAN LEARN QUICKLY

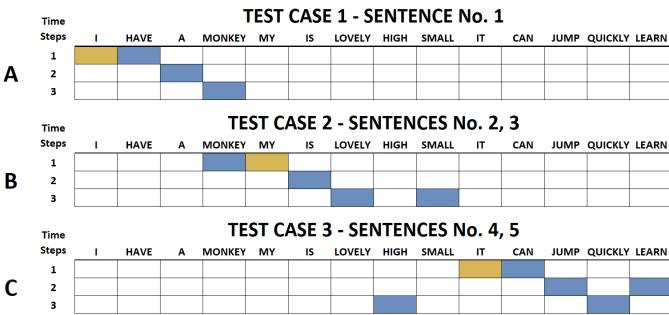


Fig. 5. Time domain evolution of the whole CA in steps (0.55ms each) forming the sentences presented in Tab. I. There are three test cases each of which is presented separately. In (A) sentence 1 is being formed. In (B) sentences 2 and 3 are being formed. Last but not least, in (C) sentences 4 and 5 are being formed.

sentences (No. 2-3 and 4-5, accordingly). The scope of these test cases is to initially test the system in a simple task and then to more complex ones as it can be in position to generate possible sentence evolution at the same time. The difference of the two (2) more difficult test cases is that the initial common words in the extracted sentences are reduced from three to two making the sentence evolution richer and more difficult as more possible words can be predicted.

The evolution of the sentences is time depended and is based on the cell's function as it has been described in previous Section II. First, in Fig. 4 the evolution of the first sentence is presented. Here, in the beginning of the simulation the word “I” is externally triggered and the CA is activating the proper words regarding the RM of the crossbar, in specific started with the word “HAVE”, continued with “A”, and concluded with “MONKEY” forming correctly the sentence (No. 1) “I HAVE A MONKEY”. All the other cells of the CA are resting. Note that the computing state of each cell lasts equal time (0.55ms) promoting the discrete time that characterizes a CA.

Following the previous example, a more holistic approach based on time domain simulation results is presented showing the evolution of the CA to the three test cases that have been simulated. To elaborate, in Fig. 5, in the vertical axis, the time steps of evolution are depicted (each time step lasts for 0.55ms) and in horizontal axis the 14 cells/words of the dictionary are depicted. With white is the initial state of the cells meaning the resting state. Yellow represents the initially triggered word, while blue the state of the cell, that has been activated (computing state). The presented simulation results prove that the circuit successfully formed the corresponding sentences by evolving the CA cells based on the provided sentences in Table I.

IV. CONCLUSIONS AND FUTURE WORK

In this work, a novel approach for incorporating CA into NLP tasks was introduced. By integrating memristors into a memristive CA circuit, the gap between biological and circuit-based networks is bridged while new possibilities in linguistic phenomena and language generation are explored. The circuit design utilized a compact memristor model that accurately captured the behavior of unipolar CBRAM devices and fitted to fabricated devices. The simulation results demonstrated the circuit's ability to generate language-like patterns and form sentences in different test cases, thus enabling the possibility for a fabricated CANLP memristor based chip.

Future work includes utilizing a larger train dictionary with more complex sentences and enhancing the circuit with a second crossbar layer in order to access previous CA states and improve the word's selection accuracy.

REFERENCES

- [1] F. Jiang, Y. Jiang, H. Zhi, Y. Dong, H. Li, S. Ma, Y. Wang, Q. Dong, H. Shen, and Y. Wang, “Artificial intelligence in healthcare: past, present and future,” *Stroke and vascular neurology*, vol. 2, no. 4, 2017.
- [2] R. Dale, “NLP commercialisation in the last 25 years,” *Natural Language Engineering*, vol. 25, no. 3, pp. 419–426, 2019.
- [3] D. W. Otter, J. R. Medina, and J. K. Kalita, “A survey of the usages of deep learning for natural language processing,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 2, pp. 604–624, 2021.
- [4] Y. Lan, X. Wang, and Y. Wang, “Spatio-temporal sequential memory model with mini-column neural network,” *Frontiers in Neuroscience*, vol. 15, p. 650430, 2021.
- [5] J. von Neumann, *Theory of Self-Reproducing Automata*. Champaign, IL: University of Illinois Press, 1966.
- [6] A. Adamatzky, *Cellular Automata: A Volume in the Encyclopedia of Complexity and Systems Science, Second Edition*, 1st ed. Springer Publishing Company, Incorporated, 2018.
- [7] G. Sirakoulis, *Cellular Automata Hardware Implementation: A Volume in the Encyclopedia of Complexity and Systems Science, Second Edition*, 11 2018, pp. 555–582.
- [8] E. Tsipas, T. P. Chatzinikolaou, K.-A. Tsakalos, K. Rallis, R.-E. Karanagi, I.-A. Fyrigos, S. Kitsios, P. Bousoulas, D. Tsoukalas, and G. C. Sirakoulis, “Unconventional memristive nanodevices,” *IEEE Nanotechnology Magazine*, vol. 16, no. 6, pp. 34–45, 2022.
- [9] G. Indiveri and S.-C. Liu, “Memory and information processing in neuromorphic systems,” *Proceedings of the IEEE*, vol. 103, no. 8, pp. 1379–1397, 2015.
- [10] M. Itoh and L. Chua, “Memristor cellular automata and memristor discrete-time cellular neural networks,” *Memristor Networks*, pp. 649–713, 2014.
- [11] I. Vourkas, D. Stathis, and G. C. Sirakoulis, “Memristor-based parallel sorting approach using one-dimensional cellular automata,” *Electronics Letters IET*, vol. 50, no. 24, pp. 1819–1821, 2014.
- [12] J. Secco, M. Farina, D. Demarchi, F. Corinto, and M. Gilli, “Memristor cellular automata for image pattern recognition and clinical applications,” in *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2016, pp. 1378–1381.
- [13] R.-E. Karanagi, I.-A. Fyrigos, K.-A. Tsakalos, V. Ntinas, M.-A. Tsompanas, and G. C. Sirakoulis, “Memristive learning Cellular Automata for edge detection,” *Chaos, Solitons & Fractals*, vol. 145, p. 110700, 2021.
- [14] T. P. Chatzinikolaou, I.-A. Fyrigos, V. Ntinas, S. Kitsios, P. Bousoulas, M.-A. Tsompanas, D. Tsoukalas, A. Adamatzky, and G. C. Sirakoulis, “Wave cellular automata for computing applications,” in *2022 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2022, pp. 3463–3467.
- [15] I.-A. Fyrigos, T. P. Chatzinikolaou, V. Ntinas, S. Kitsios, P. Bousoulas, M.-A. Tsompanas, D. Tsoukalas, A. Adamatzky, A. Rubio, and G. C. Sirakoulis, “Compact thermo-diffusion based physical memristor model,” in *2022 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2022, pp. 2237–2241.