2018 15th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE) Mexico City, Mexico.
September 5-7, 2018

Memristive Optimizer for the Assignment Task

G.M. Tornez-Xavier[1], L.M. Flores-Nava[1], F. Gómez-Castañeda[1], J.A. Moreno-Cadenas[1]
[1]Department of Electrical Engineering, CINVESTAV-IPN, Mexico City, Mexico
phone (52) 55 5747 3800 Ext. 6261
e-mail: {gtornez, lmflores, fgomez, jmoreno}@cinvestav.mx

Abstract — **This work shows an analog CMOS matrix compatible with memristors, where they work as dynamic resistors. They are also programmed to initial state values according to one assignment optimization task, taken here as an application vehicle. The whole matrix accomplishes a parallel competitive computation, where a winner-take-all mechanism is inherent by architecture. The expected complexity due to area in silicon of this memristive circuit and its connectivity are moderate, as is evident from observing its electrical diagram. Also, its performance is acceptable as proven via SPICE simulations, using 0.5-micron CMOS process. Lower processing time might be possible with latest CMOS technologies.**

Keywords — Optimizer, memristor, TEAM, Verilog-A, CMOS, competitive system.

I. INTRODUCTION

At present, the design of hardware that should realize efficient and specific analog functions for CMOS integrated circuits is a research issue. This happens in new parallel processing circuits in the category of neuromorphic systems (NMSs), where analog values in the components of vectors or matrices are common. Likewise, the expected results in the processes in NMSs should be optimal. Some functions in early artificial neural systems (ANSs) namely, recognition, function approximation and classification are supported by self-organization and competitive mechanisms. From an engineering point of view, there is an optimizer, which is embedded in both NMSs and ANSs. Type-examples of these optimizers are the back-propagation error algorithm for layered neural networks and the winner-take-all computation. An intelligent and near-optimal performance can already be reproduced with the above optimizers in hardware for time-series patterns. However, taking into account NMSs with temporal spiking neural networks (SNNs), which are the third generation of the artificial neural network models, requires new hardware design methods. This trend is motivated in part for the possibility of including nano-technological memory resistors or memristors in CMOS, which promise the reproduction in hardware of learning algorithms in SNNs. These temporal-learning algorithms would become new optimizers suitable for real-time processing of intelligent brain functions.

In this context, we present an optimizer in hardware that computes in parallel the solution of the assignment task, receiving analog-entries matrices and giving binary-entries matrices. The assignment task refers to a linear programming paradigm that finds the best configuration of a set of "processors" to a set of "figures of merit", maximizing the performance of the system. This task can be solved by the Hungarian algorithm [1].

This paper is organized as follows. Section II, reviews the assignment task. Section III introduces the memristor and the way its internal state is modeled by the ThrEshold Adaptive Memristor (TEAM) model. Section IV presents the electrical CMOS-memristor architecture that solves the assignment problem. The electrical simulations via SPICE of the competitive CMOS-memristor analog circuit are shown in Section V. This paper ends with concluding remarks, in Section VI.

II. THE ASSIGNMENT TASK

The following statement explains in brief the objective of this task.

The assignment task solves the optimal association of "processors" with selected "figures of merit" (FoMs). One processor has the capability to realize a set of different tasks, we may say T={T_1, T_2, … , T_N}, where: N is the number of tasks to realize. Likewise, this processor can realize the tasks in T with the corresponding FoM ={F_1, F_2, …, F_N}, where each F_i is a real number in the interval [0, 1]. The system in consideration has N processors, where each processor has its own FoM_i where: i=1, 2, … , N. The assignment task is defined by the singular permutation matrix $\boldsymbol{P}$ that maximizes the performance of the system. This matrix indicates which processor does which task. The matrix $\boldsymbol{F}$ specifies the Figures of Merit of the processors, whose entries are: $F_{i,\,j}$, where the first index i=1, 2, …, N counts the Task and the second index j=1, 2, … , N counts the Processor.

Four Figure of Merit matrices, $\boldsymbol{F_1}$, $\boldsymbol{F_2}$, $\boldsymbol{F_3}$, and $\boldsymbol{F_4}$ are presented below along with the respective optimal matrices: $\boldsymbol{P_1}$, $\boldsymbol{P_2}$, $\boldsymbol{P_3}$, and $\boldsymbol{P_4}$. Theses optimization cases are used to show the performance of the memristive optimizer in this paper.

$$\boldsymbol{F_1} = \begin{bmatrix} 0.25 & 0.56 & 0.82 & 0.29 \\ 0.50 & 0.62 & 0.90 & 0.48 \\ 0.70 & 0.48 & 0.20 & 0.50 \\ 0.63 & 0.24 & 0.06 & 0.37 \end{bmatrix} \quad \boldsymbol{P_1} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$\boldsymbol{F_2} = \begin{bmatrix} 0.50 & 0.41 & 0.70 & 0.39 \\ 0.40 & 0.08 & 0.55 & 0.35 \\ 0.35 & 0.50 & 0.40 & 0.80 \\ 0.20 & 0.27 & 0.76 & 0.61 \end{bmatrix} \quad \boldsymbol{P_2} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

978-1-5386-7033-0/18/$31.00 ©2018 IEEE

$$\boldsymbol{F}_3 = \begin{bmatrix} 0.55 & 0.37 & 0.15 & 0.80 \\ 0.85 & 0.45 & 0.26 & 0.47 \\ 0.67 & 0.75 & 0.53 & 0.07 \\ 0.81 & 0.52 & 0.79 & 0.77 \end{bmatrix} \quad \boldsymbol{P}_3 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\boldsymbol{F}_4 = \begin{bmatrix} 0.55 & 0.80 & 0.15 & 0.80 \\ 0.85 & 0.45 & 0.85 & 0.47 \\ 0.67 & 0.75 & 0.53 & 0.75 \\ 0.81 & 0.52 & 0.79 & 0.77 \end{bmatrix} \quad \boldsymbol{P}_4 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

III. THE MEMRISTOR

The memristor is a non-linear resistor with memory that was theoretically predicted by L.O. Chua [2] and recognized as a nanostructure by HP Labs in the first decade of 2000´s [3]. At present, there are different research topics involving this electrical component in engineering, physical, computing and biological disciplines. Modeling the memristor for designing electrical systems should take into account the way its state variable x changes. Following the presentation given in [4], we count the models that observe: 1) linear ion drift, 2) a window function, 3) nonlinear ion drift, and 4) a Simmons tunnel barrier. In [4], it is remarked that a good computing model of the memristor fulfills both accuracy and numerical efficiency for a broad set of electrical parameters associated to its technology. Therefore, the authors in [4] present the TEAM model for the titanium oxide memristor approaching the Simmons tunnel barrier, whose original expressions are complex. Fig. 1a) shows the state variable in TEAM, which is complementary to the common definition namely, $x = D- w$; D is the width of the used nanometric material and w is the width of the region the oxygen vacancies fill by displacement. Fig. 1b) also presents its electrical symbol. In TEAM, the dynamic change of x or dx/dt is given in 3 intervals of current defined as:

$$0< i_{off} < i; \quad i_{on} < i < i_{off}; \quad i < i_{on} <0.$$

Whose value is modeled with Eqs. (1), (2) and (3).

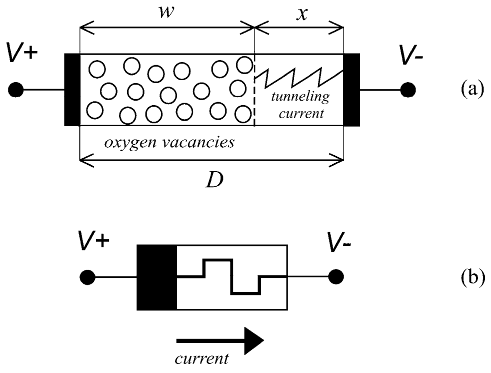

Fig. 1 a) Graphic definition of state variable, b) Symbol

$$\frac{dx}{dt} = K_{off}\left(\frac{i(t)}{i_{off}} - 1\right)^{\alpha_{off}} f_{off}(x) \qquad 0 < i_{off} < i \qquad (1)$$

$$\frac{dx}{dt} = K_{on}\left(\frac{i(t)}{i_{on}} - 1\right)^{\alpha_{on}} f_{on}(x) \qquad i < i_{on} < 0 \qquad (2)$$

$$\frac{dx}{dt} = 0 \qquad i_{on} < i < i_{off} \qquad (3)$$

Where: $f_{on}(x)$ and $f_{off}(x)$, are window functions that are evaluated with Eqs. (4) and (5).

$$f_{off}(x) = exp\left[-exp\left(\frac{x - a_{off}}{w_c}\right)\right] \qquad (4)$$

$$f_{on}(x) = exp\left[-exp\left(-\frac{x - a_{on}}{w_c}\right)\right] \qquad (5)$$

Finally, the current-voltage relation in the TEAM model can be approached with Eq. (6).

$$v(t) = R_{on}\, exp\left(\frac{\lambda}{x_{off} - x_{on}}(x - x_{on})\right) i(t) \qquad (6)$$

Where: λ, is a fitting parameter given by Eq. (7).

$$\lambda = ln\left(\frac{R_{off}}{R_{on}}\right) \qquad (7)$$

We simulated the memristor using the Verilog-A code for the TEAM model reported in [5] with technological and electrical parameters oriented to fast digital systems as given in Table 1. Fig. 2 illustrates the current-voltage characteristic for f = 2, 3 and 4 MHz, taken from (6). Fig. 3 shows the applied voltage, the current, the state variable and the memristance as a function of time, in the interval 0-1.0 µs. Fig. 3 (top) considers 3 sinusoidal signals for 2, 3, and 4 MHz with 1 V of amplitude. This type of titanium oxide memristor was used for the memristive Neuron, described in the next Section.

Table 1 TEAM electrical parameters used in simulations.

Parameter	Value	Comment
dt	1×10^{-9} s	Differential of time
R_{on}	100×10^{3} Ω	Memristance at $x = 0$ m
R_{off}	2×10^{6} Ω	Memristance at $x = 3\text{x}10^{-9}$ m
i_{off}	10×10^{-12} A	Threshold current
i_{on}	-10×10^{-12} A	Threshold current
w_c	107×10^{-12} m	Fitting parameter
a_{on}	0 m	Fitting parameter
a_{off}	3×10^{-9} m	Fitting parameter
k_{on}	-5×10^{-8} m/s	Fitting parameter
k_{off}	5×10^{-8} m/s	Fitting parameter
α_{on}	1	Fitting parameter
α_{off}	1	Fitting parameter
x_{on}	0 m	Lower limit of x
x_{off}	3×10^{-9} m	Upper limit of x

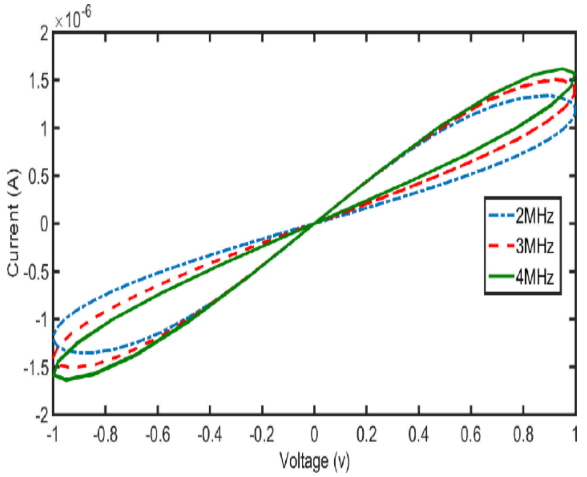

Fig. 2 I-V characteristics in frequency

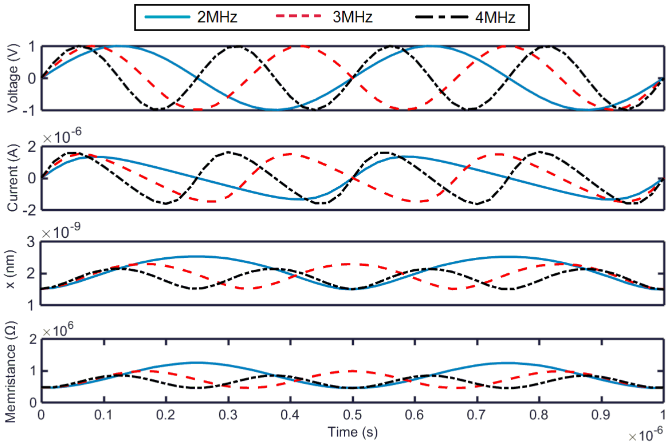

Fig. 3 (from top to bottom) Applied voltage, current, state variable and memristance

IV. MEMRISTIVE NEURON

The memristive CMOS circuit, which should be replicated to build a matrix of dimension *NxN* corresponding to the matrices $\boldsymbol{F_1}$, $\boldsymbol{F_2}$, $\boldsymbol{F_3}$, and $\boldsymbol{F_4}$, is presented in Fig. 4. This circuit is called Neuron N(i, j), where: i=1, 2, … , N and j=1, 2, … , N. The electrical matrix behaves as a winner-take-all or competitive system. In order to transfer the values of the entries of the matrix $\boldsymbol{F}$, the state variable of the memristors of N(i, j) are initially programmed equally to an analog value $\theta_{i,\,j}$ and proportional to a linear function G of the entry $F_{i,j}$ namely, $\theta_{i,\,j}= D\ G(F_{i,j})$, where: D is the width of the memristor. The Neurons settle into a stable matrix $\boldsymbol{P}$, whose entries are given by the set $\{Vo_{i,j}\}$, which are the output voltages of the CMOS inverters (M1 and M2). Fig. 5 presents the interconnectivity of the Neurons for a *NxN* array with lateral connections, without self-connection.

In general, the number of memristors for a *NxN* array is $N_{mr} = 2(N)^2(N\text{-}1)$; with $N = 4$, $N_{mr} = 96$. In fact, the Neuron N(i, j) follows the circuit configuration of the NOR digital gate analyzed in [6], where fast logic gates are designed with memristors. Therefore, we observe that the electrical matrix with memristive Neurons is equivalent to a NOR-gate array, which will be stable and settle into a permutation matrix for all possible initial values of the state variable in the memristors. This is true for architecture principle.

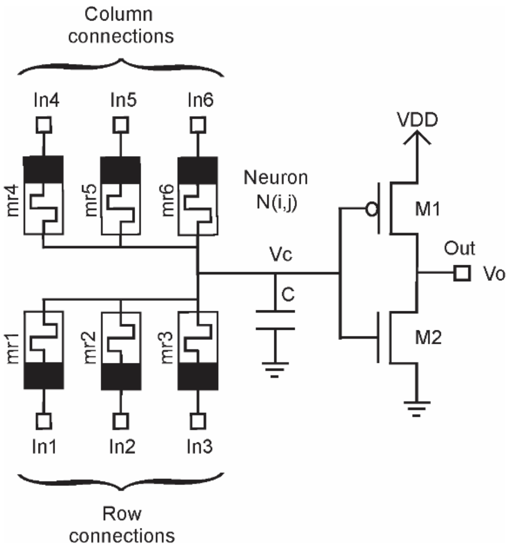

Fig. 4 Memristive Neuron

V. ELECTRICAL SIMULATIONS

Using the TEAM model, which was described above, and with the programming lines given in [5] for optimal running time, electrical simulations of the *4x4* Neuron array were realized using ADMS platform of Mentor Graphics.

In Fig. 4, the CMOS inverter was simulated using 0.5-micron, n-well, standard technology. Both C and the input capacitance of M1 and M2, which integrate current from the memristors and establish V_C, have the values 1 pF and 0.868 pF, respectively. The ratio W/L of M1 and M2 are 360/0.6 and 120/0.6, where W and L are in µm and VDD = 5V.

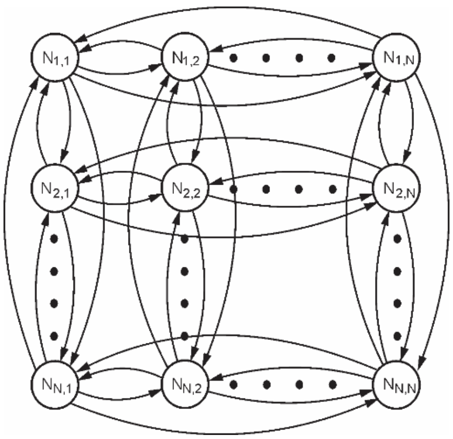

Fig. 5 Interconnectivity of the Neurons array

The six memristors (mr1 to mr6) in Fig. 4 are "programmed" equally as follows. Having the request matrix $\boldsymbol{F}$ and taking the entry $F_{i,\,j}$ then, the initial value $\theta_{i,j}$ of the state variable of the memristors in N(i, j) is given by Eq. (8) with i=1..,4 and j=1..,4. Moreover, the set $\{\theta_{i,j}\}$ contains the entries of the initial state matrix $\boldsymbol{\theta}$.

$$\theta_{i,j} = D\left(0.5 + \beta\left(F_{i,j} - 0.5\right)\right) \qquad (8)$$

The graph of Eq. 8 is shown in Fig. 6, where the values of β were found empirically in the interval: $0.20 \leq \beta \leq 0.80$ and $D = 3\text{x}10^{-9}$ m.

Making β=0.5, the values for $\theta_{i,j}$ are in the range [0.25 to 0.75] for the matrices $\boldsymbol{\theta_1}$, $\boldsymbol{\theta_2}$, $\boldsymbol{\theta_3}$, and $\boldsymbol{\theta_4}$, which correspond to matrices: $\boldsymbol{F_1}$, $\boldsymbol{F_2}$, $\boldsymbol{F_3}$, and $\boldsymbol{F_4}$.

$$\boldsymbol{\theta_1} = (3x10^{-9})\begin{bmatrix} 0.375 & 0.530 & 0.660 & 0.395 \\ 0.500 & 0.560 & 0.700 & 0.490 \\ 0.600 & 0.490 & 0.350 & 0.500 \\ 0.565 & 0.370 & 0.280 & 0.435 \end{bmatrix}$$

$$\boldsymbol{\theta_2} = (3x10^{-9})\begin{bmatrix} 0.500 & 0.455 & 0.600 & 0.445 \\ 0.450 & 0.290 & 0.525 & 0.425 \\ 0.425 & 0.500 & 0.450 & 0.650 \\ 0.350 & 0.385 & 0.630 & 0.555 \end{bmatrix}$$

$$\boldsymbol{\theta_3} = (3x10^{-9})\begin{bmatrix} 0.525 & 0.435 & 0.325 & 0.650 \\ 0.675 & 0.475 & 0.380 & 0.485 \\ 0.585 & 0.625 & 0.515 & 0.285 \\ 0.655 & 0.510 & 0.645 & 0.635 \end{bmatrix}$$

$$\boldsymbol{\theta_4} = (3x10^{-9})\begin{bmatrix} 0.525 & 0.650 & 0.325 & 0.650 \\ 0.675 & 0.475 & 0.675 & 0.485 \\ 0.585 & 0.625 & 0.515 & 0.625 \\ 0.655 & 0.510 & 0.645 & 0.635 \end{bmatrix}$$

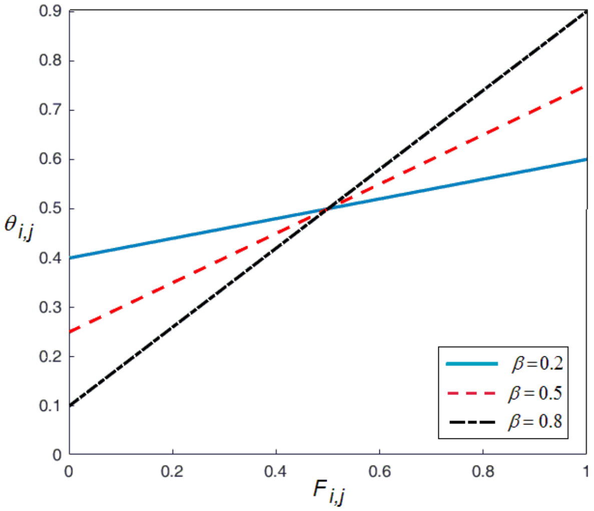

Fig. 6 Initial value $\theta_{i\,j}$ of the state variable

Setting V_C = 0V as the initial condition of the simulations, we obtain the expected matrices $\boldsymbol{P_1}$, $\boldsymbol{P_2}$, $\boldsymbol{P_3}$, and $\boldsymbol{P_4}$ at the end of the time course namely, at $1\text{x}10^{-6}$ s. Fig. 7 shows the simulations for computing $\boldsymbol{P_1}$, $\boldsymbol{P_2}$, $\boldsymbol{P_3}$, and $\boldsymbol{P_4}$ using the initial values in matrices $\boldsymbol{\theta_1}$, $\boldsymbol{\theta_2}$, $\boldsymbol{\theta_3}$, and $\boldsymbol{\theta_4}$. The traces with the expected 5V-value in the permanent regime are recognized as "Winners". Additionally, Figs. 8a) and 8c) show the final voltages of the "Winner" and the "Loser", respectively. Also, Figs. 8b) and 8d) show the memristance of the "Winner" and the "Loser", respectively. In the "Winners" all the memristors get R_{OFF} and, in the "Losers" only two memristors remain at R_{ON}. There is a transient regime inside the interval: $0 < t < 0.2\text{x}10^{-6}$s, where the initial memristance charges C and itself decreases up to R_{ON} due to the difference potential. In that moment the voltage at C reaches a value proportional to $\theta_{i,j}$ and with this value the competition starts.

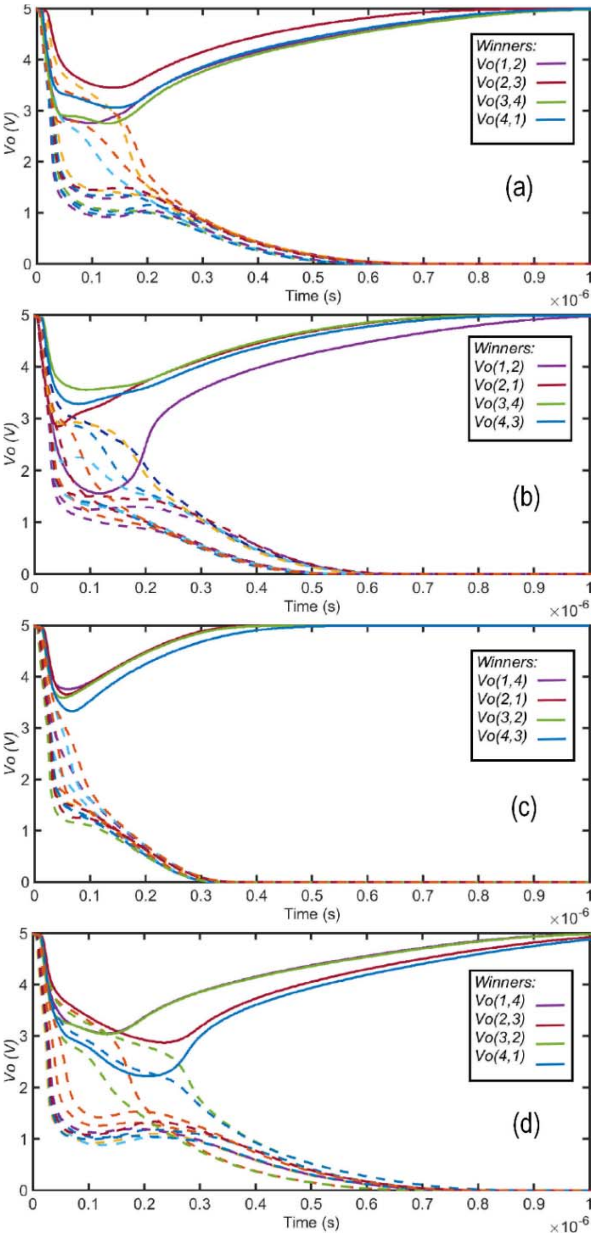

Fig. 7 SPICE simulation for computing a) $\boldsymbol{P_1}$, b) $\boldsymbol{P_2}$, c) $\boldsymbol{P_3}$, and d) $\boldsymbol{P_4}$

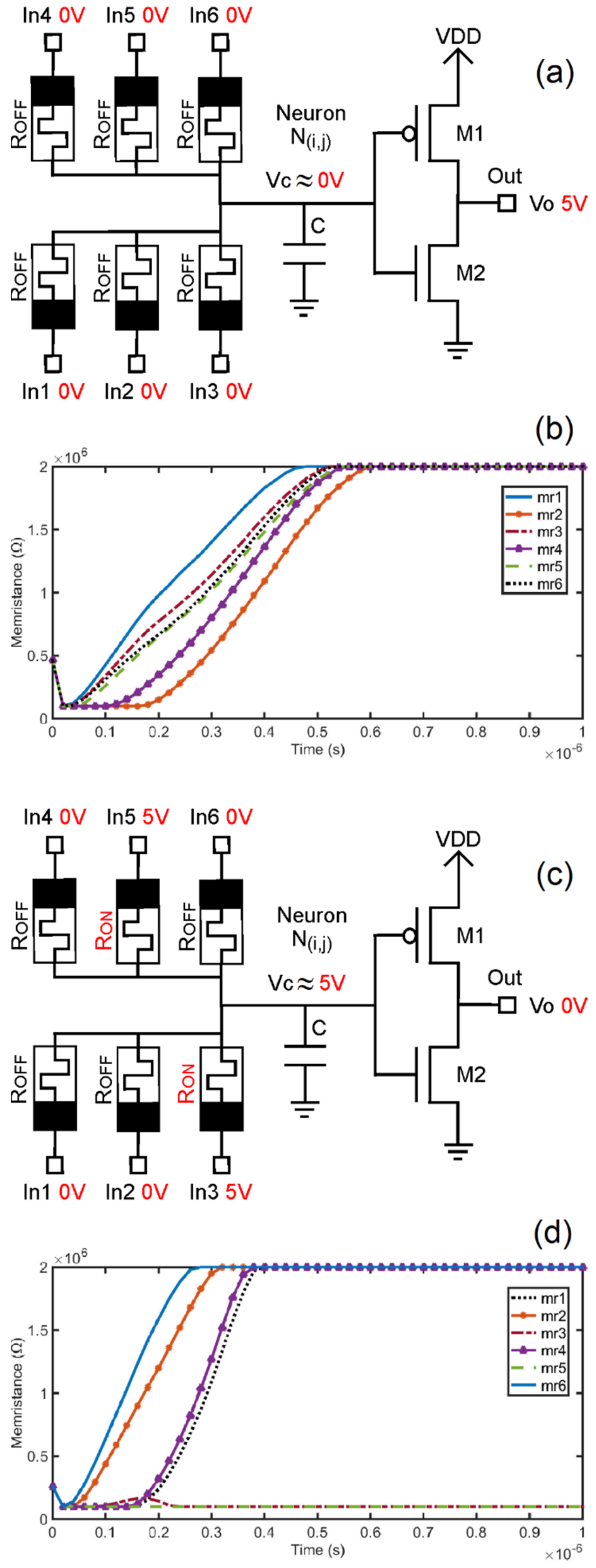

Fig. 8 Schematic diagrams a) and c) for “Winner” and “Loser”, respectively and, electrical simulations b) and d) for “Winner” and “Loser”, respectively.

VI. CONCLUDING REMARKS

This memristive analog circuit, whose alternative configuration can be seen as an array of six-input NOR gates behaves as a competitive system. The memristors change their memristance from established initial states in favor of the extreme values of either R_{ON} or R_{OFF} and solving in parallel the four assignment tasks. In other words, it is solved a combinatorial optimization problem. In fact, the nonlinear differential equations of this circuit encode in their permanent regime the solution. The analytical task associated to these equations is left out of this paper. Considering the end of the transient regime, we observe: 1) for the “Winner” Neuron, all of its memristors get R_{Off} and, 2) for the “Loser” Neuron, two of its memristors get R_{On} and the rest R_{OFF}. We have used 4 numerical matrices $\boldsymbol{F_1}$, $\boldsymbol{F_2}$, $\boldsymbol{F_3}$, and $\boldsymbol{F_4}$ as an experimental vehicle to demonstrate via electrical simulations in SPICE that the proposed memristive analog circuit is stable and finds the respective solution matrices $\boldsymbol{P_1}$, $\boldsymbol{P_2}$, $\boldsymbol{P_3}$, and $\boldsymbol{P_4}$. Following conservative design rules for the geometric sizes in M1 and M2 in the CMOS inverters, these transistors can provide the transient currents through the memristors capable to charge dynamically the input capacitive node of their neighbor connected CMOS inverters.

This memristive analog optimizer has demonstrated via electrical simulations its correct operation with processing time lower than 1.0 µs. Finding other circuit configurations memristor/CMOS for reducing the number of memristive elements is a pendent research issue.

REFERENCES

[1] H. A. Taha, Operations research: an introduction. Pearson Prentice Hall, USA 2007.

[2] L. O. Chua “Memristor: the missing circuit element,” IEEE Trans Circuit Theory, vol. 18(5), pp. 507–519, September 1971.

[3] D. B. Strukov, G. S. Snider, D. R. Stewart, R. S. Williams “The missing memristor found,” Nature, vol. 453(7191), pp. 80-83, May 2008.

[4] S. Kvatinsky, E. G. Friedman, A. Kolodny, U. C. Weiser “TEAM: ThrEshold adaptive memristor model,” IEEE Trans. Circuits Syst, vol. 60(1), pp. 211-221, January 2013.

[5] S. Kvatinsky, K. Talisveyberg, D. Fliter, A. Kolodny, U. C. Weiser, E. G. Friedman “Models of memristors for SPICE simulations”, IEEEI 2012 27th Convention of Eliat, Israel. doi: 10.1109/EEEI.2012.6377081.

[6] S. Kvatinsky, N. Wald, G. Satat, A. Kolodny, U. C. Weiser, E. G. Friedman “MRL-memristor ratioed logic”, CNNA 2012 13th international workshop. Turin, Italy. doi: 10.1109/CNNA.2012.6331426 2013.