Open Access

OBM Neurobiology

LIDSEN Publishing Inc.

Original Research

Modeling of Information Processing in Biomorphic Neuroprocessor

Sergey Yu. Udovichenko [*], Alexander D. Pisarev, Alexander N. Busygin, Abdulla H. Ebrahim, Andrey N. Bobylev, Alexey A. Gubin

University of Tyumen, REC "Nanotechnology", Tyumen, Russia; E-Mails: udotgu@mail.ru; spcb.doc@gmail.com; a.n.busygin@utmn.ru; abdulla.ybragim@mail.ru; a.n.bobylev@utmn.ru; a.a.gubin@utmn.ru

[*] **Correspondence:** Sergey Yu. Udovichenko; E-Mail: udotgu@mail.ru

Academic Editor: Raul Valverde

Special Issue: Neuroscience and Information Technology

OBM Neurobiology
2022, volume 6, issue 3
doi:10.21926/obm.neurobiol.2203134

Received: June 16, 2022
Accepted: September 13, 2022
Published: September 23, 2022

Abstract

In the present study, we present the results of the modeling of incoming information processing in a neuroprocessor that implements a biomorphic spiking neural network with numerous neurons and trainable synaptic connections between them. Physico-mathematical models of processes of encoding information into biomorphic pulses and their decoding following a neural block into a binary code were developed as well as models of the process of routing the output pulses of neurons by the logic matrix to the synapses of other neurons and the processes of associative self-learning of the memory matrix as part of the hardware spiking neural network with long-term potentiation and with the spike-timing-dependent plasticity of the memristor. The performance of individual devices of the biomorphic neuroprocessor in processing the incoming information is shown based on developed models using numerical simulation.

Keywords

Biomorphic neuroprocessor; encoding and decoding of information; logic matrix; spikes router; memory matrix; hardware pulse neural network; memristor-diode crossbar

1. Introduction

Spiking neural networks (SNN) have emerged as promising artificial neural network (ANN) types. SNNs are more biologically plausible than point-neuron ANNs, have a simpler structure due to the lesser number of neurons, and are able to implement biosimilar self-learning mechanisms [1]. However, SNNs require higher computing power to calculate the function of each neuron. This issue can be solved by hardware implementation of SNN.

Several studies have reported the hardware implementations of self-learning SNN with discrete memristor synapses [2-6]. However, the proposed discrete circuits for hardware implementation of SNN cannot be used to construct a super-large hardware neural network due to lack of integration of memristors and a significant number of active electronic elements with high power consumption in the circuits.

Currently, existing hardware devices based on memory matrices with an integrated memristor crossbar [7-11] perform highly specialized functions, largely matrix multiplication—a weighted summation of input voltage pulses. Certain hardware [10, 11] contains integrated memristor crossbars; however, its self-learning is delayed due to software calculation of weight adjustment in the peripheral system. The weight update procedure results in a loss of information because the input information is ignored during this process. The self-learning can occur while input pulses are processed in the memristor crossbar, and the activated neuron return pulses to the synapse [12].

Building highly integrated hardware SNN can avoid the above issues. Earlier, our team had presented the biomorphic neuroprocessor concept [13], which is based on a memristor-diode crossbar and implements a hardware SNN with numerous biomorphic neurons [14]. Key nodes of the hardware component of the neuroprocessor, super-large memory [15], and logic [13] matrices are used, which are an array of synapses and specify the weight and route of communication between the neurons, respectively. These matrices must be super-large because each neuron in the network can have numerous synaptic connections. The neuroprocessor is unique as it is built based on the original electrical biomorphic neuron model and is biomorphic in terms of performing the functions of a biomorphic neural network created based on the original software biomorphic neuron model. Furthermore, the neuroprocessor contains input encoding [16] and output decoding [17] devices built using a universal logic matrix based on a composite memristor-diode crossbar. The maximum size of the square logic [13] and memory matrices [15] should be at least 10^6 cells, according to the analysis of input signal attenuation. Thus, it is possible to construct input and output devices and a hardware neural network of the appropriate size using these matrices.

Current memristors are of low stability and reproducibility. Nevertheless, because calculations of hardware biomormphic SNN are distributed over the whole network, it lowers requirements to maintain each memristor's performance stability and reproducibility. A hardware implementation of a network allows the embedding of an electric circuit and algorithm of the astrocyte, which can increase the conductivity of memristors adjacent to the failed memristor [18, 19]. In addition, certain methods can increase fault tolerance by adjusting the input signals [20, 21].

A previous study [15] reported a physical model of the process of signal processing for weighing voltage pulses and summing cell currents by a memory matrix. The physical models consisted of processes of encoding the information into biomorphic pulses and their decoding into a binary code following a neural block [16], routing the neuron output pulses by the logic matrix to the synapses of other neurons [13], and associative self-learning of the memory matrix as a component of a hardware SNN [12].

The purpose of this study is to construct physico-mathematical models based on the above-mentioned physical models of individual neuroprocessor nodes. Numerical simulations based on the developed models can be used to confirm the performance of corresponding nodes.

2. Information Encoding into Biomorphic Pulses Processes Simulation

It is necessary to transform the incoming information from a number set into a pulse sequence because a neuroprocessor is a hardware implementation of a spiking neural network. Biosimilar pulses [14] are generated in the input device of the biomorphic neuroprocessor [16], built based on a universal logic matrix with a memristor-diode crossbar.

2.1 Information Encoding Model

The physico-mathematical operational model of the encoder's electrical circuit, as well as the operational model of the logic matrix (the spikes router), is based on programmable logic circuits that form disjunctive normal forms (DNF). The logic matrix should have the functional completeness of logical operations to implement a set of DNFs from the operations NAND and NOR. A complete logical basis is realized in the logic matrix when logical variables are supplied in the direct and inverse form. AND logic gates (conjunctions) are implemented using the diode-resistive logic with the ability to disconnect any gate inputs by changing the resistance of the memristor in the memristor-diode crossbar. The output inverters restore the logical voltage values. The output conductors of the second matrix are connected to pull-up resistors. In the first matrix, delay lines are connected to pull-up resistors as a voltage source. The matrices are chained in series and create a perfect disjunctive normal form that switches delayed pulses to outputs based on a binary number at the input.

The encoder logic circuit is a perfect disjunctive normal form. The encoding is performed in two operations: first, the input binary number is converted into a positional code, and afterward, according to the position, the signal from the corresponding delay line is transmitted to the output. Thus, the binary address decoder circuit is implemented in the conjunction matrix.

Decoder commutation matrix elements m_{ij} are bound to values of positive integer j in the binary form.

$$j = \sum_{i=0}^{n-1} \alpha_{ij} 2^i \, ; \; m_{ij} = \begin{cases} \alpha_{ij}, i < n \\ \overline{\alpha_{ij}}, n \le i < 2n \end{cases}, j \in [0, 2^n - 1], i \in [0, 2n - 1], \tag{1}$$

n – bit depth of the input number. The conversion of the number x into the value of the pulse delay from the neuron $l(x)$ is determined using the Gaussian function.

$$l(x) = W \cdot \left(1 - \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right)\right). \tag{2}$$

Here W is the temporal width of the encoding window; μ is the eigenvalue of the neuron, the encoding of which will have the minimum delay; σ is the parameter that determines the selectivity of the response of the neuron.

Thus, for *j* line (of *j* neuron) the delay value is:

$$l_j = W \cdot \left(1 - \exp\left(-\frac{1}{2}\left(\frac{j-\mu_j}{\sigma_j}\right)^2\right)\right). \tag{3}$$

The accuracy of the input number representation depends on the number of used bits – n. The number of delay lines determines the transform accuracy. During encoding, the input number is sampled by a pulse from the control circuit. The delay circuit consists of two RC circuits and logic gates. The time constant of the first integrating RC circuit determines the delay value. The delay is a programmable value because a memristor is used in the RC circuit. The second RC circuit is responsible for the width of the output pulse.

2.2 Simulation of the Input Unit of the Neuroprocessor

Numerical simulation of the operation of the encoder was conducted in the specialized program MDC-SPICE (Memristor-Diode Crossbar - Simulation Program with Integrated Circuit Emphasis) using Formulae (1)-(3), as well as Formula (5) for the model of the logic matrix cell operation. The high resistance state of memristors was 100 kΩ, and the low resistance state was 1 kΩ. Pulses formed by the delay line were of 1.5 V magnitude and 1 ms width. The inverter power supply was 1.5 V. Figure 1 shows the result of modeling the encoding of two input numbers n_1 and n_2 by a population of three virtual neurons into the delay and frequency of the pulses simultaneously performed by the electrical circuit [11]. Every 20 ms, the number n_1 was sequentially increased from 1 to 7, whereas n_2, on the contrary, was decreased from 7 to 1.

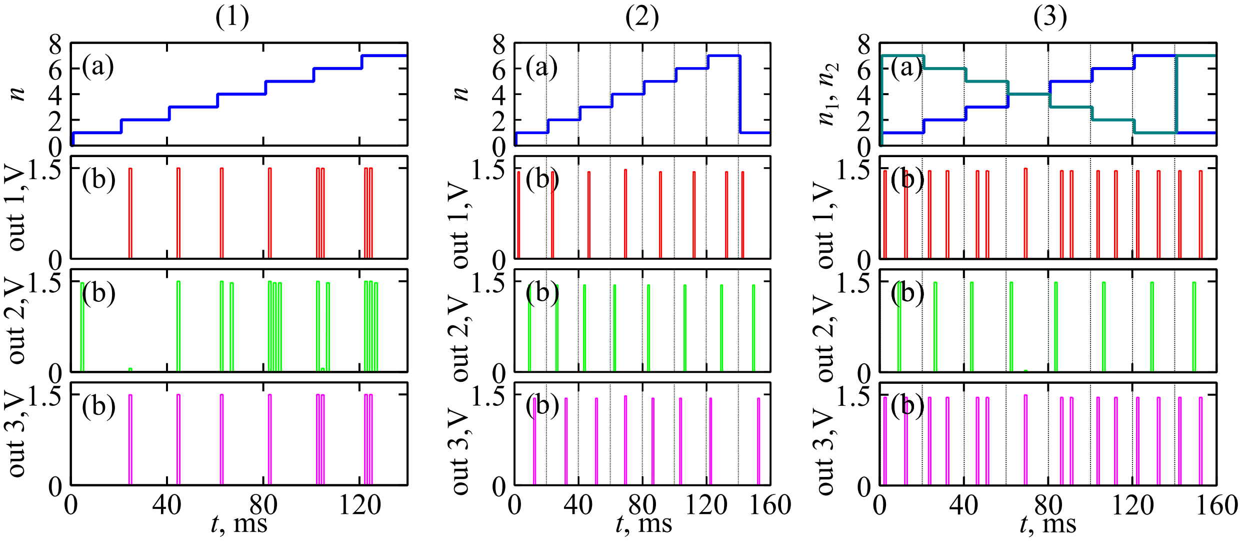

Figure 1 Simulation of encoding modes by a population of three neurons into the frequency (1) and delay (2) of pulses, as well as simultaneously into delays and into the frequency of pulses by a population of neurons: (a) change in input numbers in time; (b) - output pulses.

Simultaneous encoding of the pixel brightness value and its spatial derivative into delay and frequency, respectively, by a population of virtual neurons, allows the transfer of more visual information for the same amount of time than frequency or delay encoding. Although biological neural networks display similar coding, it does not take into account the time derivative [22].

3. Simulation of the Logic Matrix Operation in the Mode of Neurons Unit Output Signals Routing

The physico-mathematical model of the cell operation was created based on simplified electrical models of the memristor and Zener diode to develop and numerically simulate an extra-large logic matrix based on a composite memristor–diode crossbar.

3.1 Logic Matrix Cell Operation Model

The electrical circuit of a logic matrix cell is a combination of a memristor and a selective Zener diode, connected to one of the crossbar conductors. In turn, this conductor is connected to a transistor-based CMOS inverter gate [13].

An idealized current–voltage characteristic was built for the Zener diode: it is a piecewise linear function of three straight lines. The differential electrical resistance of the Zener diode is set large ranging from reversible breakdown voltage to the opening voltage of the p–n junction. In the areas of the plot far from strong non-linearity, the piecewise linear current–voltage characteristic coincides with the current–voltage characteristic in the diode model [23]. The total voltage at the input of the inverter is described by the following formula:

$$V_j = \frac{\frac{V_b}{R_b} + \sum_i \frac{V_i}{R_i + R_d(V_{di})}}{\frac{1}{R_b} + \sum_i \frac{1}{R_i + R_d(V_{di})}};\ V_{di} = V_j - I_i R_i;\ R_d \approx \begin{cases} 0, V_i = 0, \\ \infty, V_i \approx V_b. \end{cases} \tag{4}$$

Here V_b is the bias voltage, R_b is the pull-up resistor, R_d is the Zener diode resistance, R_i is the memristor resistance, and V_i is the input voltage of the cell. Given that the input voltages, within a small error, can only have two values corresponding to logical 0 and 1, it is sufficient to consider the work at the level of logical variables. Every *j*-th crossbar column within a logic inverter performs the NAND function of input bits x_i:

$$y_j = \overline{\bigwedge_{\imath=0}^{n} m_{\imath j} x_\imath}\,. \tag{5}$$

Commutation matrix elements $m_{ij} \in [0,1]$ determine will x_i be used in conjunction or not.

3.2 Operational Model of Logic Matrix in the Mode of Pulse Routing

The neural unit sends pulses to the logic matrix to transmit the information [13]. The shape of pulses is highly distorted due to previous transformations and the magnitude decreases, thereby degrading the entire signal. The best solution, in this case, is using a routing pulse using a digital circuit element that brings pulses into a normalized form by amplifying them and limiting their magnitude. The degradation of the signal can be avoided using a logic matrix due to the presence of inverter gates in it. Inverters amplify the signal, i.e., the output splitting factor of the neuron is large, which is an advantage in the case of extra-large block size.

The logic matrix has two functions: as a router, it directs output pulses between the neurons and protects signals from degradation. In the case of complex routing of pulses, the logic matrix can be made in a 3D topology containing sufficient layers. Inside the layer, signals are routed horizontally through conductive buses; between the layers, the signal is switched in the vertical direction through memristors. The 3D logic matrix solves the problem of transmitting the signals from one layer to another vertically without using the complex technology of 3D assembly of wafers or through-silicon vias.

The physico-mathematical model of the router that directs the output pulses of neurons to the synapses of other neurons is implemented in a logic matrix of two functional layers. Memristors on the primary diagonal of the first layer of the logic matrix are programmed to have low resistance. Therefore, the NOT element is formed by a set of one-input NAND elements in the first layer of the matrix. Low-resistance memristors in the second layer of the logic array connect the required outputs from the first layer to the input bus of the output inverter. Next, the commutation matrix is diagonal: $m_{ij} = \delta_{ij}$, where δ_{ij} is the Kronecker symbol.

$$y_j = \overline{\bigwedge_{\imath=0}^{n} m_{\imath\jmath} x_\imath} = \overline{\bigwedge_{\imath=0}^{n} \delta_{\imath\jmath} x_\imath} = \overline{x_j}. \tag{6}$$

One inverter and memristors connected to it perform conjunction with negation over the signals inverted in the first layer, which is equivalent to a disjunction over them.

$$y_j = \overline{\bigwedge_{\imath=0}^{n} m_{\imath\jmath} \overline{x_\imath}} = \bigvee_{i=0}^{n} m_{ij} x_i \,. \tag{7}$$

Consequently, the logic matrix consisting of two functional strata allows routing signals from any of its inputs to its random output. The signal routes programmed in the logic matrix determine the neural network architecture to be installed into the neuroprocessor.

3.3 Numerical Simulation of Pulse Routing

The performance of a 3D logic matrix as a router was analyzed by SPICE simulation of a fragment consisting of two layers, which, for example, contains two logic cells. The low resistance state of memristors was 10 kΩ and the high resistance state was o 1 MOhn. The inverter power supply was 1.5 V. Figure 2 shows the voltage levels at the inputs and outputs of the matrix obtained during the simulation.

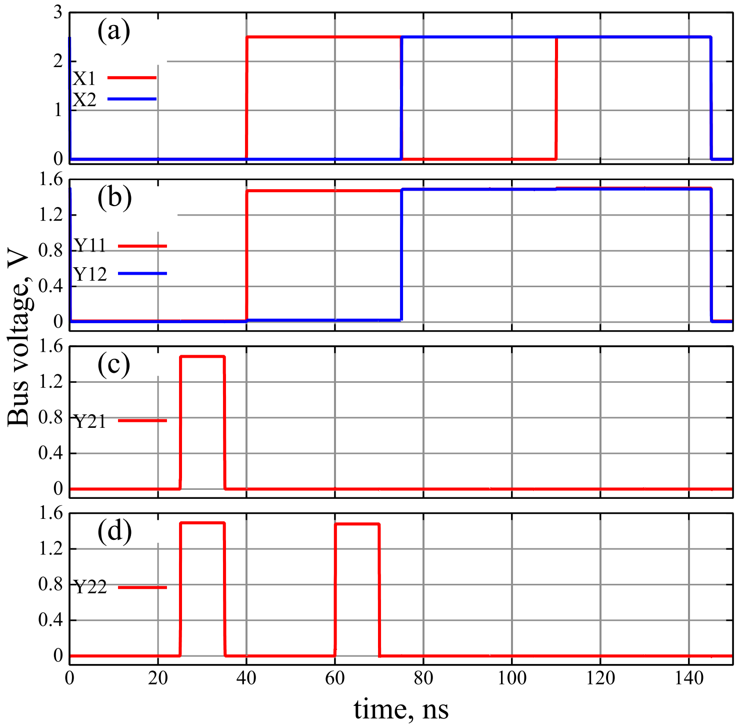

Figure 2 Diagrams of voltage at the inputs and outputs of the layers of the logic matrix: a) input of the upper layer; b) outputs of the upper layer/inputs of the lower layer; c) output of the lower layer; and d) output of the lower layer.

Corresponding commutation matrices are:

$$m_{ij}^{(1)} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix};\ m_{ij}^{(2)} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

Input variables X1 и X2 are provided in the inverted form. Thus, the following logic functions are being implemented using the higher layer:

$$Y11\ = V(x111) = X1\ V\ X2 \text{ и } Y12\ = V(x112) = X2,$$

The logic functions of the lower layer are:

$$Y21\ = NOT\ (Y11) = NOT\ (X1\ V\ X2) \text{ и } Y22\ = NOT\ (Y12) = NOT\ (X2).$$

Figure 2 shows that the output matrix signals obtained by simulation correspond to the programmed logic functions: the signal at Y11 is the disjunction of X1 and X2, and the signal at Y12 is the inversion of X2. Signals Y21 and Y22 are gated by pulses supplied to power the output inverters.

The redirection of signals from any inputs of the logic matrix of two functional layers to its arbitrary output was shown by numerical simulation.

4. Associative Self-Learning of a Memory Matrix as Part of Hardware Spiking Neural Network

Self-learning of synapses in the memory matrix follows Hebb's rule, as in a real synapse: the strength of the connection between simultaneously activated neurons increases. The change in the weight of a synapse depends on the difference between the firing times of the presynaptic and postsynaptic neurons $\Delta t = t_{pre} - t_{post}$.

The original cell electrical circuit requires a specific implementation of self-learning rules. The presynaptic signal is a pair of voltage pulses of the same amplitude but different polarities due to the use of a complementary pair of memristors in the cells in both self-learning models. When triggered, the neuron sends a negative voltage pulse (post) back to the synapse. In this case, a change in the strength of the synapse occurs when pulses from the presynaptic neuron arrive at the synapse. The shape of the presynaptic pulses determines the implemented local learning rule.

4.1 LTP Rule Self-Learning Model

The long-term potentiation (LTP) rule assumes an increase in the weight of the synapse when presynaptic and postsynaptic signals coincide in time. Therefore, the voltage pulse sign of the presynaptic signals is maintained constant in the corresponding model, changing the state of the memristors in the cell, always in the direction of decreasing resistance and increasing the weight of the synapse.

Weight of synapse, that is, memory matrix cell is:

$$w = \frac{1}{R_d}\left(\frac{R_- - R_+}{R_- + R_+}\right),$$

where R_d is the Zener diode resistance, and R_+ и R_- is the resistance of cell memristors.

The implementation of the LTP rule involves the use of voltage pulses, the shape of which is shown in Figure 3a.

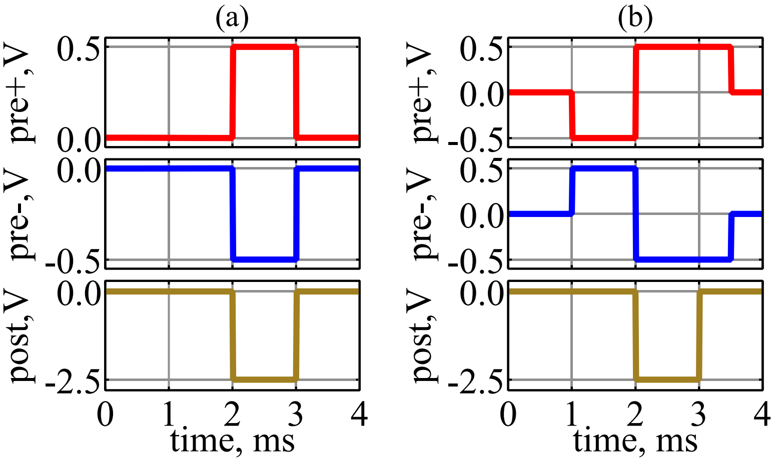

Figure 3 Pre and postsynaptic voltages of the memory matrix cell: (a) LTP and (b) spike-time-dependent plasticity (STDP).

The plasticity function at initial states of memristors from the middle of the possible range $R_+ = R_- = 0{,}5 \cdot (R_{off} + R_{on}) = R_0$ described by the analytical formula obtained using the memristor model:

$$\Delta \boldsymbol{w}(\Delta \boldsymbol{t}) = \left[\frac{\boldsymbol{2R_0R_d}}{\boldsymbol{\beta V_p(\tau - |\Delta t|)}} - \boldsymbol{R_d}\right]^{-1}.$$

The analytical representation of the plasticity function allows to select parameters of the memristor to achieve the optimal learning rate of the neural network, which ensures the convergence of the numerical calculation of the neural network in minimum time.

4.2 STDP Rule Self-Learning Model

The STDP rule works similarly to LTP for Δt < 0. According to this rule, the weight of the synapse should decrease for Δt > 0. In the corresponding model, a decrease is implemented using alternating voltage pulses (Figure 3b). The final direction of change in the state of memristors and the weight of synapses are determined by the prevailing signs of the voltage of presynaptic pulses during the action of the postsynaptic pulse.

The width of the initial phase of the presynaptic pulse and the postsynaptic pulse is τ, and the width of the second phase of the presynaptic pulse is $\tau_1 > \tau$. The complex shape of pulses in the synapse causes the analytical plasticity function to be piecewise and consist of six curves:

1) $\Delta w(\Delta t) = 0$, при $\Delta t > 2\tau$;

2) $\Delta w(\Delta t) = \left[R_d - \frac{2R_0R_d}{\beta V_p(2\tau-\Delta t)}\right]^{-1}$, when $\tau < \Delta t < 2\tau$;

3) $\Delta w(\Delta t) = \left[\frac{2R_0R_d}{\beta V_p\tau} - R_d + 2R_d\frac{\Delta t}{\tau}\right]^{-1}$, when $0 < \Delta t < \tau$;

4) $\Delta w(\Delta t) = \left[\frac{2R_0R_d}{\beta V_p\tau} - R_d\right]^{-1} = const$, when $\tau - \tau_1 < \Delta t < 0$;

5) $\Delta w(\Delta t) = \left[\frac{2R_0R_d}{\beta V_p(\tau_1+\Delta t)} - R_d\right]^{-1}$, when $-\tau_1 < \Delta t < \tau - \tau_1$;

6) $\Delta w(\Delta t) = 0$, when $\Delta t < -\tau_1$.

4.3 Numerical Simulation and Testing of a Hardware Pulsed Perceptron

Hardware perceptron electrical circuit [12] is based on a memory matrix with complementary memristor synapses [15] constructed using a memristor–diode crossbar with four pairs of input conductors and two output buses [24]. Accordingly, the crossbar contains eight cells, which act as synapses of the neural network. The average resistance of memristor–diode half-cell after fabrication was 18.8 MΩ with a standard deviation of 31.6 MΩ.

The neural network is a single-layer perceptron that consists of four virtual input neurons and two hardware output neurons. The output layer of the perceptron consists of two hardware neurons based on operational amplifiers. The electric circuit of a neuron consists of a current–voltage converter, an analog integrator, a comparator, a delay circuit in the form of an integrating RC circuit, and a field–effect transistor. The current–voltage converter is the input of the neuron and maintains a virtual zero potential on the output buses of the crossbar, providing the summation of the output currents of synapses. A voltage proportional to the input synaptic current is supplied to the integrator, simulating charge accumulation on the neuron membrane.

A neural network consisting of four virtual input neurons and two hardware output neurons was trained (Figure 4) to recognize a set of input images with a resolution of 2 × 2 pixels. The input picture pixel brightness values were converted into the average frequency of the uniform input pulse sequence using the PIC family microcontroller. The pulse magnitude was 6 V and the width was 1 ms. The resulting frequency of the pulse was proportional to the brightness value of the pixel.

The simulation was performed in the MDC-SPICE program, in which the averaged experimental current–voltage characteristics of these states were used as table functions instead of constant resistances in the low-conductivity and high-conductivity states of the memristor. The experimental and model learning curves presented in Figure 4 show the proportion of correctly recognized input images from the number of learning cycles (epochs). Each training cycle contained 128 pictures obtained by adding noise to the two original reference pictures.

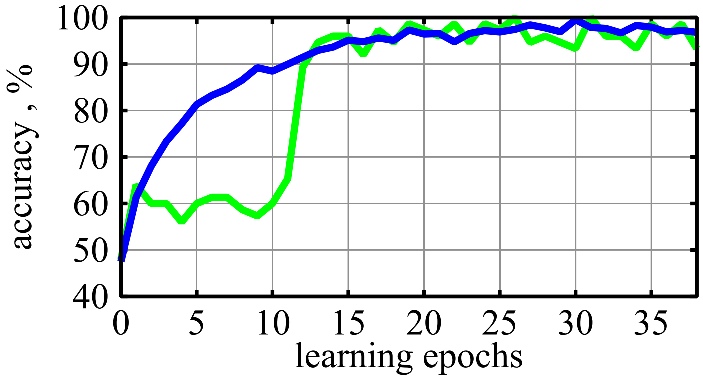

Figure 4 Self-learning curves of the hardware neural network: green color is for experiment and blue is for SPICE simulation.

The model learning curve is smoother than the experimental one, which could be attributed to a change in the conductivity in the memristor model occurring smoothly, without jumps. The difference between the experimental and calculated curves is attributed to the unequal characteristics of individual memristors in the crossbar.

The new association generation occurs during retraining due to the receipt of new input information. It manifests itself as new impulse at the input of the neuron, as shown in Figure 5. Such generation has already been shown on discrete memristors in associative memory circuits [2, 3], reproducing the formation of a conditioned reflex according to Pavlov. An increase in the input voltage of a neuron, proportional to the synaptic current, is due to an increase in the synapse during the generation of a new association.

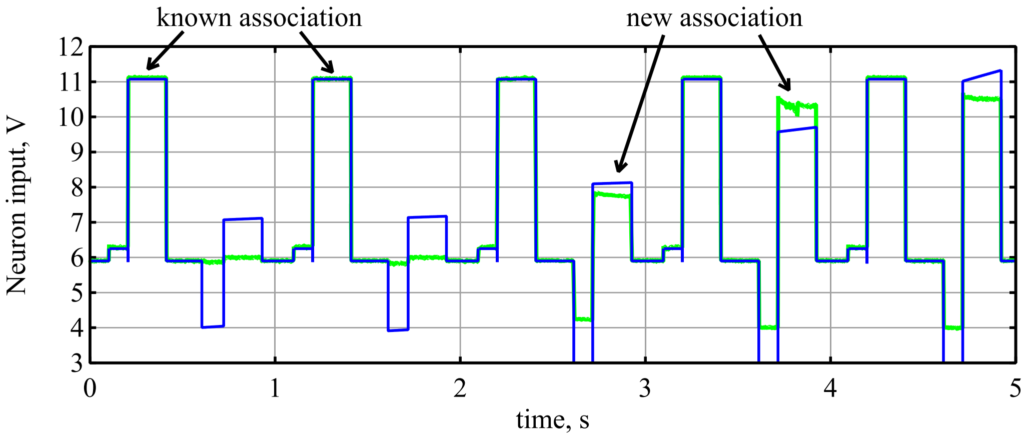

Figure 5 Generation of a new association with the known one: green color is for experiment and blue is for SPICE simulation.

The overlapping of output pulses in time is due to the assignment of the corresponding identical input voltage pulses in the experiment and during the simulation. The difference in the rate of increase in the voltage amplitude of the pulses is attributed to the change in the conductivity of the memristor in the experiment, which has a probabilistic spread.

5. Simulation of Decoding Biomorphic Pulses into a Binary Code

5.1 Pulses Decoding Model

The physico-mathematical model of the output device operation is based on the pulse signals processing in a universal logic matrix [13], one layer of which is a set of logic gates “AND” or “OR” with arbitrarily connected inputs. An information value or its modification was set by routing pulse signals and combining them according to the NAND logic on one line. The logical element “OR” increases the frequency by combining the pulses of the input signal with the pulses of another signal or generator. This is equivalent to the operations of the summation of information values. The information is modified by switching the routing of information pulses between the positions of the lines of the logic matrix. The transformation formula in the model was set by programmable connections of the routing matrix. Within the routing matrix, there must be N·K possible connections between N input and K output lines, which is required to program K routing links. For a one-to-one transformation, the condition N = K must be satisfied for a bijection mapping of the input set of values to the output one, and it can also be N > K in the case of an incomplete transformation.

The commutation matrix of the output device has non-zero elements:

$$m_{ij} = 1, \qquad i \cdot n \leq j < (i+1) \cdot n, \tag{8}$$

where n is the bit depth of the binary number generator, i is the pulse input index, and j is the output bits index.

For states of the output inverters of the logical matrix y_j, the following relation is valid:

$$y_j = \overline{\left(\bigwedge_{i=0}^{l} m_{ij}x_i\right) \wedge g_j}, \tag{9}$$

where x_i is the state of i-th input of logic matrix, l is the full number of inputs, and g_j is the voltage state of the pull-up resistor as determined by the binary number generator.

5.2 Simulation of Operation of the Output Device of the Neuroprocessor

The mathematical model for the electrical circuit is shown on the left, and the results of its numerical simulation are shown on the right in Figure 6.

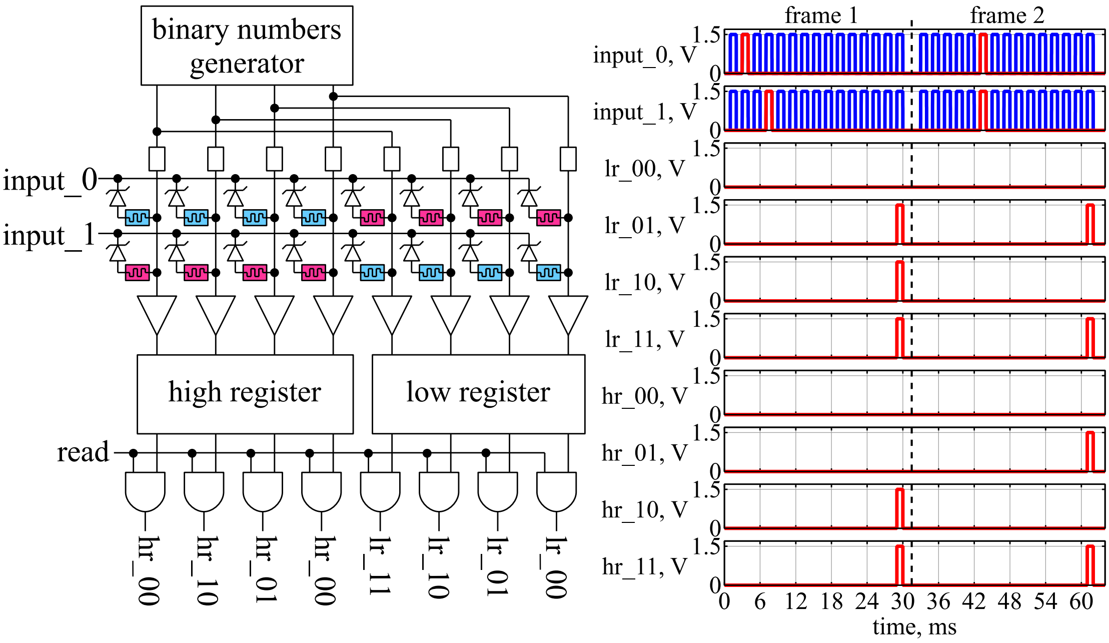

Figure 6 Electric circuit and simulation results of decoding a two-bit pulse signal into a two-bit number using a generator of binary numbers.

Numerical simulation is based on Formulae (5), (8), and (9) in the specialized MDC-SPICE program for the output device operation that decodes a two-bit pulse signal from a population of neurons into a two-digit number using a binary number generator. The high resistance state of memristors was 100 kΩ, and the low resistance state was 1 kΩ. Pulses formed by the generator were of 1.5 V magnitude, and the width was 1 ms. The inverter power supply was 1.5 V. The hexadecimal two-digit number format in the range from 0 × 00 to 0xFF was selected as an example of decoding.

The diagram shown in Figure 6 on the right shows that according to the SPICE simulation and Formula (9), the decoded value corresponds to the given example. The values of the converted number at the input are shown on the diagrams by the signals input_0 and input_1 in the form of delayed red pulses. The delay value of these pulses is determined by the position of the red pulses in relation to the blue clock signal. Plots of blue color are obtained by combining the pulses of the binary number generator. There are 15 pieces in one frame for the hexadecimal format of the information signal.

Pulses are counted from the end of the frame. In the first frame, the position of the input pulse on the input_0 line corresponds to the delay value 14, whereas the position of the input pulse on the input_1 line is 12, implying that in the first frame, the hexadecimal number 0xCE is encoded as delays. In the second frame, the number 0xAA is encoded. The result is output in a parallel binary code using eight output lines. In the first frame, as shown in the SPICE simulation diagram (Figure 3 on the left), on the lr and lh lines, the binary value is b'11001110', corresponding to the hexadecimal number 0xCE. In the second frame, the lines lr and lh show the binary value b'10101010', corresponding to the hexadecimal number 0xAA.

6. Summary

The physico-mathematical model of the cell operation was created based on simplified electrical models of the memristor and Zener diode for numerical simulation of an extra-large logic matrix with a composite memristor-diode crossbar.

The physico-mathematical model of processing signals in an extra-large logic matrix of a biomorphic neuroprocessor during routing the output signals of a neural block was developed. The results of numerical simulation, conducted on the basis of the developed models, of the routing process of output pulses of neurons to the synapses of other neurons in the logic matrix are presented.

The physico-mathematical model of information processing in the encoding device of a biomorphic neuroprocessor based on a logic matrix with a composite memristor–diode crossbar was developed. The input encoder in the mode of encoding a binary number simultaneously into frequency and delay of pulses by a population of three neurons is numerically simulated in the specialized program MDC-SPICE. The last type of information encoding is observed in biological neural networks [3]; however, it is not considered a time derivative.

A physico-mathematical model of the information processing in the logic matrix as a decoding device of a biomorphic neuroprocessor was developed. The results of numerical simulation for decoding pulse-type information signals into a binary data format were obtained. A compact implementation of a neuroprocessor signal decoding circuit is shown, which can be implemented in one layer of a memristor–diode logic matrix. The result of the circuit solution was achieved using logical transformations performed inside the memristor–diode crossbar and a binary number generator installed on the periphery of the memristor logic matrix.

Physico-mathematical models of associative self-learning of a memory matrix were developed, providing a specific implementation of the LTP and STDP self-learning rules associated with the originality of the memristor–diode cell. Using numerical simulation, the ability of a memory matrix with a number of cells 4 × 2 to associative self-learning as a component of hardware spiking neural network was demonstrated. The results of numerical simulation are in good agreement with the experimental data, confirming the correctness of the developed physico-mathematical learning model according to the STDP rule.

The numerical simulation of hardware spiking neural network based on a memristor–diode crossbar demonstrates the generation of a new association.

Thus, the operability of individual neuroprocessor nodes was demonstrated using numerical simulation based on the developed models. The next step will be simulation and benchmarking of the complete circuit of the neuroprocessor, including input and output devices, a router, and a hardware neural network.

Author Contributions

Dr. S.Yu. Udovichenko – project development, writing, analytics, theory; Dr. A.D. Pisarev – modelling, analytics, theory; Dr. A.N. Busygin – modelling, writing, analytics, theory; A.H. Ebrahim – modelling, analytics; Dr. A.N. Bobylev – project development, writing, experimental; A.A. Gubin – analytics, experimental.

Funding

This work was partially supported by the Russian Foundation for Basic Research grant No. 19-37-90030 “Generation of new knowledge in a neural network based on an array of memristor synapses in the memory matrix of a biomorphic neuroprocessor and the principles of increasing the speed and energy efficiency of information processing on a specialized device in comparison with existing computing facilities”.

Competing Interests

The authors have declared that no competing interests exist.

References

1. Khacef L, Abderrahmane N, Miramond B. Confronting machine-learning with neuroscience for neuromorphic architectures design. Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN); 2018 July 8-13; Rio de Janeiro, Brazil. New York: IEEE.
2. Wang Z, Wang X. A novel memristor-based circuit implementation of full-function Pavlov associative memory accorded with biological feature. IEEE Trans Circuits Syst I. 2018; 65: 2210-2220.
3. Minnekhanov AA, Emelyanov AV, Lapkin DA, Nikiruy KE, Shvetsov BS, Nesmelov AA, et al. Parylene based memristive devices with multilevel resistive switching for neuromorphic applications. Sci Rep. 2019; 9: 10800.
4. Demin VA, Nekhaev DV, Surazhevsky IA, Nikiruy KE, Emelyanov AV, Nikolaev SN, et al. Necessary conditions for STDP-based pattern recognition learning in a memristive spiking neural network. Neural Netw. 2021; 134: 64-75.
5. Antonov IN, Belov AI, Mikhaylov AN, Morozov OA, Ovchinnikov PE. Formation of weighting coefficients in an artificial neural network based on the memristive effect in metal–oxide–metal nanostructures. J Commun Technol Electron. 2018; 63: 950-957.
6. Surazhevsky IA, Demin VA, Ilyasov AI, Emelyanov AV, Nikiruy KE, Rylkov VV, et al. Noise-assisted persistence and recovery of memory state in a memristive spiking neuromorphic network. Chaos Solitons Fractals. 2021; 146: 110890.
7. Li C, Belkin D, Li Y, Yan P, Hu M, Ge N, et al. Efficient and self-adaptive in-situ learning in multilayer memristor neural networks. Nat Commun. 2018; 9: 2385.
8. Ielmini D. Brain-inspired computing with resistive switching memory (RRAM): Devices, synapses and neural networks. Microelectron Eng. 2018; 190: 44-53.

9. Yao P, Wu H, Gao B, Tang J, Zhang Q, Zhang W, et al. Fully hardware-implemented memristor convolutional neural network. Nature. 2020; 577: 641-646.
10. Cai F, Correll JM, Lee SH, Lim Y, Bothra V, Zhang Z, et al. A fully integrated reprogrammable memristor–CMOS system for efficient multiply–accumulate operations. Nat Electron. 2019; 2: 290-299.
11. Wang Z, Li C, Song W, Rao M, Belkin D, Li Y, et al. Reinforcement learning with analogue memristor arrays. Nat Electron. 2019; 2: 115-124.
12. Bobylev AN, Busygin AN, Gubin AA, Pisarev AD, Udovichenko SY. Manufacture and testing of a pulsed hardware neural network with memristor synapses for a biomorphic neuroprocessor. Nanobiotechnol Rep. 2021; 16: 761-766.
13. Pisarev AD, Busygin AN, Udovichenko SY, Maevsky OV. A biomorphic neuroprocessor based on a composite memristor-diode crossbar. Microelectron J. 2020; 102: 104827.
14. Filippov VA, Bobylev AN, Busygin AN, Pisarev AD, Udovichenko SY. A biomorphic neuron model and principles of designing a neural network with memristor synapses for a biomorphic neuroprocessor. Neural Comput Appl. 2020; 32: 2471-2485.
15. Pisarev A, Busygin A, Udovichenko S, Maevsky O. 3D memory matrix based on a composite memristor-diode crossbar for a neuromorphic processor. Microelectron Eng. 2018; 198: 1-7.
16. Busygin AN, Ebrahim AH, Pisarev AD, Udovichenko SY. Input device for a biomorphic neuroprocessor based on a memristor-diode crossbar for the pulse coding of information. Nanobiotechnol Rep. 2021; 16: 798-803.
17. Ebrahim AH, Udovichenko SY. Automatic building of electrical circuits of biomorphic neuroprocessor units and visualization of their numerical simulation. In: Science and global challenges of the 21st century - Science and technology. Cham: Springer; 2021. pp. 16-23.
18. Liu J, Harkin J, Maguire LP, McDaid LJ, Wade JJ. SPANNER: A self-repairing spiking neural network hardware architecture. IEEE Trans Neural Netw Learn Syst. 2017; 29: 1287-1300.
19. Liu J, McDaid LJ, Harkin J, Karim S, Johnson AP, Millard AG, et al. Exploring Self-Repair in a Coupled Spiking Astrocyte Neural Network. IEEE Trans Neural Netw Learn Syst. 2018; 30: 865-875.
20. Danilin SN, Shchanikov SA, Panteleev SV. Determining operation tolerances of memristor-based artificial neural networks. Proceeding of 2016 International Conference on Engineering and Telecommunication (EnT); 2016 November 29-30; Moscow, Russia. New York: IEEE.
21. Danilin SN, Shchanikov SA. Neural network control over operation accuracy of memristor-based hardware. Proceeding of 2015 International Conference on Mechanical Engineering, Automation and Control Systems (MEACS); 2015 December 1-4; Tomsk, Russia. New York: IEEE.
22. Gollisch T, Meister M. Rapid neural coding in the retina with relative spike latencies. Science. 2008; 319: 1108-1111.
23. Wong S, Hu CM. Spice macro model for the simulation of zener diode I-V characteristics. IEEE Circuits Devices Mag. 1991; 7: 9-12.
24. Pisarev A, Busygin A, Bobylev A, Gubin A, Udovichenko S. Fabrication technology and electrophysical properties of a composite memristor-diode crossbar used as a basis for hardware implementation of a biomorphic neuroprocessor. Microelectron Eng. 2021; 236: 111471.