Spiking Neural Network Based on Cusp Catastrophe Theory

Damian Huderek, Szymon Szczęsny [*], Raul Rato [†]

Abstract. This paper addresses the problem of effective processing using third generation neural networks. The article features two new models of spiking neurons based on the cusp catastrophe theory. The effectiveness of the models is demonstrated with an example of a network composed of three neurons solving the problem of linear inseparability of the XOR function. The proposed solutions are dedicated to hardware implementation using the Edge computing strategy. The paper presents simulation results and outlines further research direction in the field of practical applications and implementations using nanometer CMOS technologies and the current processing mode.

Keywords: spiking neuron, cusp catastrophe, decision support system, XOR problem

1. Introduction

Issues concerning artificial intelligence and machine learning are among the most heavily exploited areas of research in the discipline of computer science. Commercial applications of artificial neural networks mainly focus on the so-called second generation networks based on the classic McCulloch-Pitts model and using continuous activation functions. Due to network perception and effectiveness of data analysis, high hopes are currently associated with third generation neural networks (the so-called spiking neurons), which are the closest model for real animal neurons. IT corporations also research neurons of this type [2, 3]. Unfortunately, a practical application of spiking networks is difficult due to the expensive network training process, the non-trivial way of coding information and the still lower efficiency compared to the second generation neural networks. Data processing, classification of patterns and image analysis using

---

[*]Institute of Computing Science, Poznań University of Technology, Poznań,{damian.huderek, szymon.szczesny}@put.poznan.pl

[†]Universidade Nova de Lisboa, 2829-516 Caparica, Portugal, r.rato@fct.unl.pt

a spiking neurons network requires computation with greater complexity and greater use of resources than implementations based on classical perceptrons, WTA (Winner Takes All) mechanisms or recurrent and convolutional networks.

It may be assumed that the key neural network effectivenes measure is the NU (Number of Units) parameter defining the number of neurons necessary to perform a specific task and the NC (Number of Channels) defining the number of required synapses, and thus the number of multiplication operations necessary for implementation, which, in case of hardware implementation, are particularly expensive. In case of implementations based on, FPGA (Field Programmable Gate Array) for example the possibility of parallelizing calculations is mainly associated with the required number of multiplying modules. In addition, important parameters include the power consumption of the neuromorphic system, and, in case of a third generation network, also the frequency of impulses generated by neurons and the energy per impulse [15]. The physical implementation parameters are particularly important for a number of techniques included in the so-called edge computing, i.e. , processing close to the data source before transferring the data to a cloud [5, 11]. Ensuring high efficiency of data processing in a neural network requires, on one hand, using techniques which minimize energy consumption, and on the other, using techniques which make it easier to reduce the complexity of the neuromorphic processor by reducing the number of connections between neurons or the number of neurons. The authors of this paper present an approach which makes it possible to reduce the size of the network and thus facilitates its training.

In this article, the authors demonstrate an approach based on the cusp catastrophe theory [1] in order to quickly model the phenomenon of nerve impulse propagation. The result of considering the problem of modeling strongly non-linear reactions, which occur in animal nervous systems, is a new model of a fast neuron, as well as its dual model. This rather unusual approach proves to be perfect for solving non-linear problems, as demonstrated in the article with an example of the XOR problem, i.e. the linear inseparability of the space of results. The architecture of the network solving the problem was compared with other implementations based on classic models. The authors put a lot of effort to ensure that solutions described in the paper could be effectively implemented also in the hardware layer. Therefore models presented in the paper are suitable for effective implementation as CMOS circuits implemented in modern nanometer technologies. This is an important advantage which makes it possible to design and use a hardware implementation of a neural network instead of a software simulation on a standard desktop processor. Neurons can be placed in silicon in the form of an IPcore and on a common substrate with a digital processor.

This paper will also include a brief discussion on the adaptation of networks based on the presented models of neurons to the task of solving selected multi-criteria problems, in particular selection problems and the problem of multi-criteria classification. The concept described in this paper is also an introduction to broader topic researched by the authors in the area of fuzzy reasoning, which will be a published in future articles. This paper presents a concept of a new neuron model and a new model of a third generation network.

The paper is organized as follows. Section 2 explains the cusp catastrophe model of

a single neuron. Section 3 describes two spiking neuron models used in this study and a recommended synapse model. Section 4 contains the implementation of a network solving the XOR problem and a comparison of the demonstrated approach to other approaches published in recent years. Status and future directions are presented in Section 5. The last section contains the research summary.

2. Cusp catastrophe model

The concept of an efficient spiking network is based on the spatial model of the cusp catastrophe [1]. The idea of using the catastrophe theory originates in the need to model psychological behaviors. The model using the cusp catastrophe theory behaves in one way when triggering reactions and differently when silencing them. In addition, such a reaction can be strongly non-linear. By controlling the behavior of the model, one can solve multidimensional problems, for example, multi-criteria problems, in which the criteria can be fuzzy.
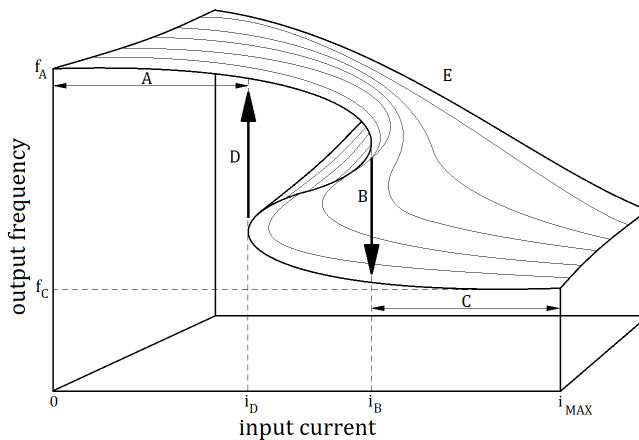
Figure 1. Cusp catastrophe diagram

Figure 1 is a graphical representation of the cusp catastrophe model. The basic element of the model is the existence of two states marked as A and C. The transition between states takes the form of a hysteresis defined by two points i_B and i_D. Passing both points takes place with the achievement of the appropriate state (B, D). This shows the non-linearity of the model – states A and C are separated and the transition between them takes the form of a jump. Regardless of the direction of the jump, it is caused by a sufficiently strong input signal. There is a threshold, exceeding which causes a sudden change, which, due to the hysteresis, is also difficult to invert. This model is perfectly suitable for describing the phenomenon of nerve impulse propagation in a neural network. An example could be a strongly non-linear response to pain registered by a nervous system. Even a small stimulus registered by

a single nociceptor induces the propagation of a nerve impulse to the brain, which analyzes it and generates a reaction based on the impulse. The result of stimulation several neurons registering the impression of pain is a reaction of the whole organism. Silencing such a reaction requires achieving a state, the activation of which takes place with stimuli of a power much lower than the level of activation of the current state. Markings of the axes on the diagram in Figure 1 are adequate to input functions and responses of the neuron, the characteristics of which are based on the presented curve. States of the neuron are coded in two frequency levels, while the thresholds for reaching a given state are determined with the level of current entering the soma of the neuron.

The mathematical description of the cusp catastrophe model is based on the following differential equations [1]:

$$\frac{dz}{dt} = \frac{dV(z; \alpha, \beta)}{dz} \tag{1}$$

and

$$dV(z; \alpha, \beta) = \alpha z + \frac{1}{2}\beta z^2 - \frac{1}{4}z^4 \tag{2}$$

The model consists of three components: two control factors (in this case α and β) and one output variable (z). The second equation is a potential function V. In this equation, the asymmetry or normal control factor is presented as α, and β is the bifurcation or splitting control factor. In a three-dimensional response surface, both α and β are connected to determine the outcome variable z. The equation has a reference to the diagram of the cusp catastrophe theory presented in Figure 1. For the described research, variable z is frequency and variable α is current I.

Section 3 will present two models of neurons developed by the authors based on the cusp catastrophe model [13]. The rationale behind using such models of neurons will be presented in Section 4 with an example of an effective solution to a non-linear problem.

3. Spiking neuron models description

The neuron model presented in this article was created based on the spiking neuron model described in paper [10], instead of the traditional Hodgkin-Huxley model [8]. The Izhikevich model offers better performance in hardware implementations and large neural networks.

Equation 3 describes the model presented by Izhikevich. In this equation dimensionless variables are presented as v and u. Dimensionless parameters are presented as a, b, c, and d. The membrane potential of the neuron is represented by v and the membrane recovery variable is represented by u. This accounts for the activation of K^+ ionic currents and inactivation of NA^+ ionic currents, and it provides negative feedback to v. According to equation 3, after the spike reaches its apex ($+30mV$), the membrane voltage and the recovery variable are reset. Synaptic currents or injected DC-currents are presented with variable I.

$$\frac{dv}{dt} = 0.04v^2 + 5v + 140 - u + I,$$

$$\frac{du}{dt} = a(bv - u) \tag{3}$$

$$if \ v \geq 30mV, \ then \ \begin{cases} v \leftarrow c, \\ u \leftarrow u + d. \end{cases}$$

Figure 2 presents two models used in this study. Model 2(a) previously introduced in [13] consists of two blocks. The first block is an element responsible for modeling the hysteresis. Depending on current I at the input of the model, it is possible to achieve two levels of hysteresis. Depending on the specified level, a specific value of current is passed onto the second block labelled H-H. This block is a classic model of a neuron described with equations 3 and functionally corresponds to the soma model described by the Hodgkin-Huxley or Izhikevich model. Depending on the current, two frequencies of voltage peaks are generated at the output of the neuron.
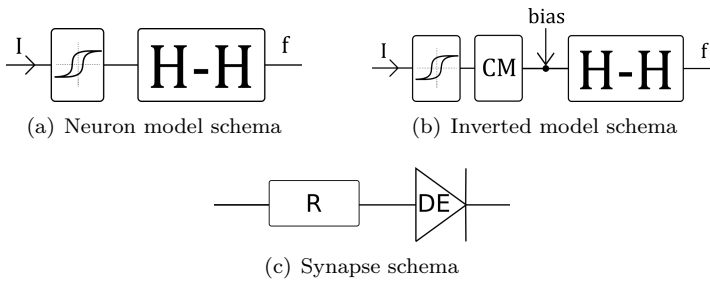

(a) Neuron model schema   (b) Inverted model schema

(c) Synapse schema

Figure 2. Models used in the research

The second model is presented in Figure 2(b). This model is complementary to the first presented model and its response is inverted. It consists of a hysteresis block, a Current Mirror (CM) and a H-H block generating a sequence of peaks. The current mirror was added to invert the direction of the current entering the neuron. In addition, in the model, between the current mirror and the response generator block, the authors added a bias current. Values of levels and transition conditions between the states of the inverted neuron are defined with equation 4 and correspond to the conditions shown in Figure 1. If the input current of the model exceeds threshold i_B, the neuron is in state C and the output features a low frequency of peaks. If the current falls below threshold i_B but not below threshold i_D, the neuron is still in state C. The frequency change at the output occurs when the input current drops below threshold i_D, which means a change of state C to state A. At the output of the neuron the frequency of peaks increases. An increase of the input current above threshold i_D does not change the state until threshold i_D is exceeded. Changes of states are strongly non-linear and sudden. The operation of the non-inverted neuron is similar but the states and transitions diagram is a symmetric reflection of the diagram in

Fig. 1 including a relation between peak frequency at output $f_C > f_A$.

$$f(t) \; = \; \begin{cases} f_C, & I > i_B, \\ f_C, & I > i_D \; \wedge \; f(t-1) = f_C, \\ f_A, & I < i_D, \\ f_A, & I < i_B \; \wedge \; f(t-1) = f_A. \end{cases} \tag{4}$$

A simple circuit shown in Figure 2(c) was proposed as a model of a synapse connecting neurons. It is a serial connection of resistance R modeling the weight of the synapse, and the ideal diode DE. In the ideal model the problem of diode non-linearity was omitted, therefore the current entering neurons of the subsequent layer is linearly dependent on the actual difference in the membrane potential of neurons of the subsequent and previous layer. The presence of the diode improves the training process of the network and is justified in the hardware implementation of the programmable synapse, as further described in Section 5.


(a) Neuron model simulation   (b) Inverted neuron simulation
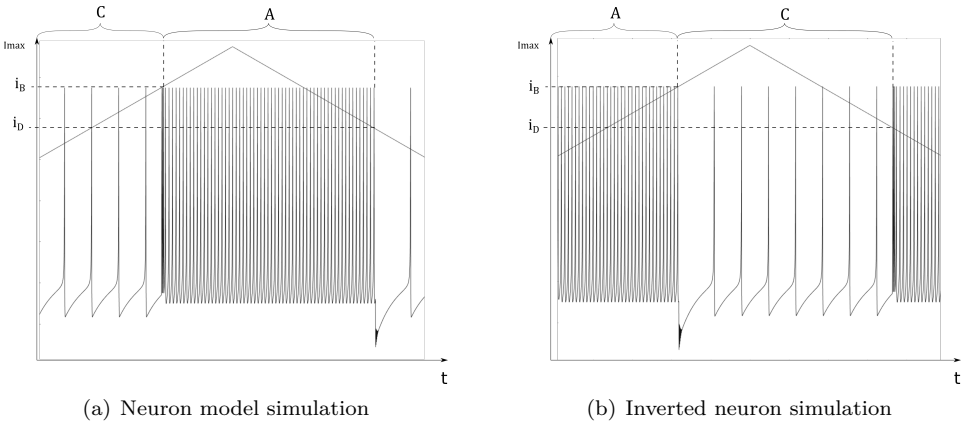
Figure 3. Used neuron models simulation

Figure 3 presents the neuron response plot to a specific input current. It shows single simulations of the original neuron and the inverted neuron. Triangular current in the range of 0 to 40pA is applied to the input of each neuron. The response consists of two frequencies of voltage peaks at the output of the neuron. In both simulations, states A and C, for which we obtain specific peak frequencies, as well as thresholds for transitions between states (i_D, i_B), were marked in the diagram. For the original neuron, a simulation of which is presented in Figure 3(a), it is shown that when the current increases and exceeds the value of 35 pA (threshold i_B), the peak frequency increases and remains high until the input current value is decreased to below 15 pA (current i_D). Figure 3(b) presents the simulation for the inverted neuron. The comparison of the simulations of the original neuron and the inverted neuron proves the complementarity of models of these neurons. As the input current increases above threshold i_B, the frequency of peaks at the output of the neuron decreases.

Due to the application of the previously mentioned cusp catastrophe theory to the spiking neuron model, the processing is done using a frequency of spikes at the output of the neuron, but information is passed with a single spike appearing against a specific frequency. Depending on the number of spikes at the input of the neuron one can end up with two frequencies of spikes at the output of the neuron.

4. Solving the XOR problem

Using the previously described models, the authors successfully implemented a network solving the problem of linear inseparability of the (XOR) space. The network consists of two neurons in the hidden layer and one neuron in the output layer. Each neuron in the hidden layer realizes a division of space into two parts, which are summed up by the neuron in the output layer. The first one neuron in hidden layer implements the OR function and the second one implements the NAND function. The neuron in the output layer implement the AND function of signals of neurons of the hidden layer.
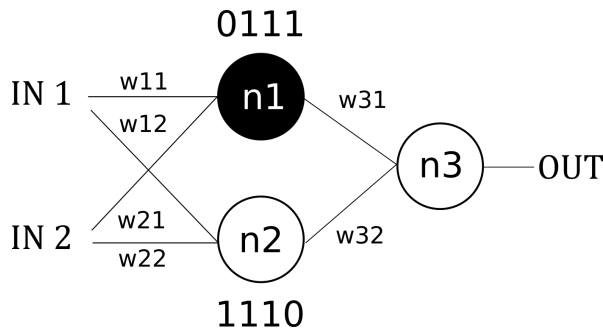
Figure 4. Network diagram,
∘ – non-inverted neuron, ● – inverted neuron

The solution made it possible to implement a network solving the XOR problem using 3 neurons. Most of the papers presented in literature use a network of 6 neurons for this purpose. This is usually associated with increasing the size of the hidden layer, e.g. [7], using 5 neurons in the hidden layer and one at the output. A similar implementation is the network described in [14] containing 4 neurons in the hidden layer and two in the output layer – one for each of the coded logic states. An interesting implementation is [4], the authors of which did solve the XOR problem using 3 neurons, but the solution is inefficient, because it requires using 24 weights. It is worth noting that the solution proposed by the authors of this research is characterized by a small dispersion of weights at the level of 1.99, and weight values are as follows: [w11 w12 w21 w22; w31 w32] = [2.1 0.4 0.47 0.44; 0.11 0.11]. The network was trained using a simulated annealing algorithm. For comparison, in [4]

dispersion equals 97.954. This parameter is a particularly important criterion during hardware implementation. According to the author's of this research best knowledge, the proposed solution based on two complementary neuron models makes it possible to manage network architecture in the most efficient way and to solve the classic XOR problem using the smallest number of neurons and the smallest number of weights. A comparison of various approaches found by the authors in literature is presented in table 1.

Table 1. Comparison of XOR problem solutions,
NU – neurons, NC – synapses

Implementation	NU	NC	Training cycles	Weight dispersion
This article	3	6	100	1.99
[4], 2018	3	24	1000	97.954
[7], 2017	6	n/a	n/a	n/a
[14], 2017	6	16	n/a	n/a

The authors performed a confusion simulation of the network response for a series of 12 possible state changes of input signals. In all cases the network re-created the XOR function of input signals. The response of the network shown in Figure 4 is presented in Figure 5.
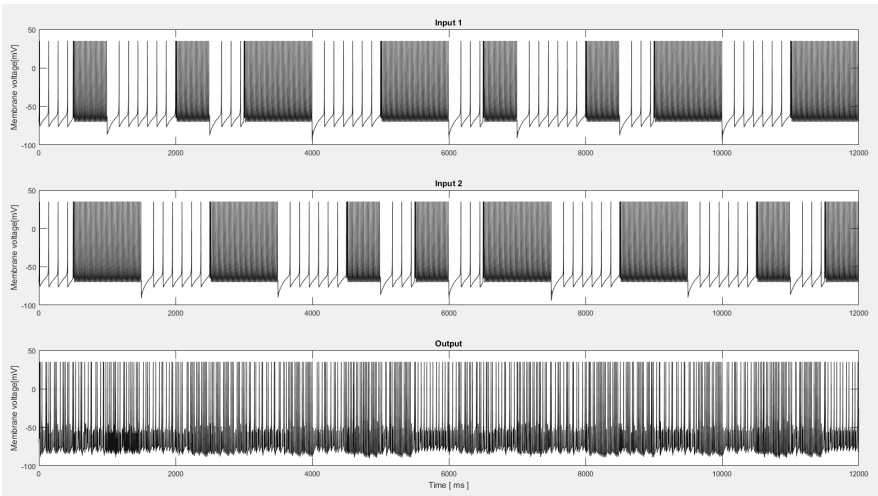
Figure 5. Network simulation for the XOR problem

5. Discussion and Summary

The presented models of neurons demonstrate a large potential thanks to the efficiency of data processing. In the previous section it was proved, for example, that complementarity makes it possible to solve simple problems using a small number of neurons. This is an introduction to a wider subject area, which the authors plan to present in future articles. At present, work is being carried out concerning the implementation of complex classifiers based on predefined blocks of complementary spiking neurons. Initial results show that it is possible to ignore the classic problem of iterative training of a network using the process of routing neurons in blocks. Such an approach speeds up the process of implementing the classifier and provides more opportunities to accurately estimate the required resources (the number of necessary spiking neurons) to implement a network to a specific problem.

The described models have an additional parameter which makes it possible to modulate the curve expressing the frequency dependence of the neuron response signal on the current entering the neuron. These parameters are thresholds i_B, i_D for activating the input current of Figure 1. Various scenarios for selecting these values and modulating the characteristics of the catastrophic curve are possible. Case $|i_B - i_D| \rightarrow 0$ characterizes a neuron of high sensitivity and case $|i_B - i_D| \rightarrow I_{MAX}$ corresponds to a neuron with high selectivity. The possibility of modifying threshold i_B is particularly intresting. The greater the value, the stronger must be the stimulus at the input of the neuron, so that it activates the *jump* to state C and transfers the received signal to subsequent layers of the network. This provides the possibility of implementing in complex networks – whole complex areas consisting of neurons with a common threshold, which are only activated with massive, strongly non-linear reactions of the entire network. This approach makes it possible to analyze network data taking into account additional criteria which determine the shape of the catastrophic curve. This mechanism can be used to solve a selected class of multi-criteria problems. Possible areas of application include, for example, 1. analysis of stock indices obtained using technical analysis, also taking into account additional stock market moods, 2. solving decision problems and making more restrictive or more empathetic decisions depending on the currently activated network areas. A network architecture based on specific nerve centers in which neurons have common bias-dependent characteristics, resembles a brain model and makes it possible to model psychological behaviors. Research also made it possible to unlock trans-Boolean computing functions obeying an L2 distance metric impossible to perform by standard classic Boolean logic.

Finally, let's briefly discuss the hardware implementation. As mentioned in Section 1, the key aspect of most systems is power consumption and strategies for minimizing it. At the same time, the implementation of spiking networks is an area of interest of many IT corporations, as well as a subject of numerous scientific research dedicated to the problem of training [9, 20] or hardware implementation [6, 19]. Models of neurons shown in Figure 2 were developed for hardware implementation using modern nanometer CMOS technologies. It should be added that technologies lower than 90 nm enforce the use of current mode for data processing, which was justified in [16]. In current mode additive operations are carried out in nodes in accordance

with the Kirchhoff's current law. According to this rule, in the model shown in Figure 2(b), the bias current is added to the current entering the neuron. Multiplication operations, in turn, are performed using the so-called current mirrors. The advantage of mirrors is the possibility of constructing programmable multiplication circuits working in current mode [17]. A specific feature of mirrors is the ability to only analyze currents supplied to the mirror input (regardless of the current direction, i.e. its sign). A mirror constitutes a barrier for currents supplied at the output side. This programmable multiplication circuit also functions as a diode. The model of the synapse presented in Figure 2(c) can be implemented using such a mirror, making it possible to implement the weight mechanism in the synapse, as well as the diode element. The authors have so far realized an implementation of the soma in 65 nm technology for the model described in this article, obtaining power consumption at the level of 60 pW, which is a competitive solution compared to others published in recent years [12, 15, 18]. This implementation will be presented in the next article.

The article is an introduction to the research started by the authors in the field of practical applications of models of spiking neurons. The research is focused on processing close to the data source, hence the work presents several solutions to improve hardware implementation. The basic benefit of using neuron models based on catastrophe theory is processing efficiency, as demonstrated with the solution of the non-linear problem, competitive to others published over the last two years. The article also outlines the further direction of research, and will be successively published in subsequent papers.

References

[1] Chen D. D., Chen X. J., Zhang K., *An Exploratory Statistical Cusp Catastrophe Model*, IEEE International Conference on Data Science and Advanced Analytics, pp. 100-109, 2016.

[2] Cheng H.-P. et al., *Understanding the design of IBM neurosynaptic system and its tradeoffs: A user perspective, Design, Automation & Test in Europe Conference & Exhibition*, IEEE, 2017.

[3] Davis M. et al, Loihi, *Neuromorphic Manycore Processor with On-Chip Learning*, IEEE Micro, vol. 38, Issue 1, pp. 82-99, 2018.

[4] Enríquez-Gaytán J., Gómez-Castañeda F., Moreno-Cadenas J.A., Flores-Nava L.M., *Experimental Spiking Neural Network: Solving the XOR Paradigm with Metaheuristics*, 15th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE) Mexico City, Mexico, 2018.

[5] Esposito C., Castiglione A., Pop F., Choo K. R., *Challenges of Connecting Edge and Cloud Computing: A Security and Forensic Perspective*, IEEE Cloud Computing, vol. 4, no. 2, pp. 13-17, March-April 2017.

[6] Frenkel Ch. et al., *A 0.086-mm2 12.7-pJ/SOP 64k-Synapse 256-Neuron Online-Learning Digital Spiking Neuromorphic Processor in 28 nm CMOS*, IEEE Transactions on Biomedical Circuits and Systems, 2018.

[7] Luo Q. Fu, Y., Liu J., Qiu J. Bi, S., Cao Y., Ding X., *Improving Learning Algorithm Performance for Spiking Neural Networks*, 17th IEEE International Conference on Communication Technology, 2017.

[8] Hodgkin A. L., Huxley A. F., *A quantitative description of membrane current and its application to conduction and excitation in nerve*, The Journal of Physiology, vol. 117, pp. 500-544, 1952.

[9] Chang R. Hu, S., Wang H., Huang J. He, Q., *Efficient Multispike Learning for Spiking Neural Networks Using Probability-Modulated Timing Method*, IEEE Transactions on Neural Networks and Learning Systems, 2018.

[10] Izhikevich E. M., *Simple Model of Spiking Neurons*, IEEE Transactions on Neural Networks, vol. 14, no. 6, pp. 1569-1572, 2003.

[11] Jain R., Tata S., *Cloud to Edge: Distributed Deployment of Process-Aware IoT Applications*, 2017 IEEE International Conference on Edge Computing (EDGE), Honolulu, HI, 2017, pp. 182-189.

[12] Joubert A., Belhadj B., Temam O., Héliot R., *Hardware spiking neurons design: analog or digital?*, The 2012 International Joint Conference on Neural Networks.

[13] Rato R., Coito F., *Spike HW Computing*, Champalimaud NeuroScience Symposium, 2013, DOI: 10.13140/2.1.3802.2723.

[14] Reljan-Delaney M., Wall J., *Solving the Linearly Inseparable XOR Problem with Spiking Neural Networks*, Computing Conference 2017, 2017.

[15] Sourikopoulos I., Hedayat S., Loyez Ch., Danneville1 F., Hoel V., Mercier E., Cappy A., *A 4-fJ/Spike Artificial Neuron in 65 nm CMOS Technology*, Frontiers in Neuroscience, 2017.

[16] Szczęsny S., *Current-Mode FPAA with CMRR Elimination and Low Sensitivity to Mismatch*, Circuits, Systems and Signal Processing, vol. 36, Issue 7, pp. 2672-2696, 2017.

[17] Szczęsny S., *High Speed and Low Sensitive Current-Mode CMOS Perceptron*, Microelectronic Engineering, vol. 165, pp. 41-51, 2016.

[18] Wu X., Saxena V., Zhu K., Balagopal S., *A CMOS Spiking Neuron for Brain-Inspired Neural Networks with Resistive Synapses and In-Situ Learning*, IEEE Transatcions on Circuits and Systems II: Express Briefs, vol. 62, Issue 11, 2015.

[19] Zapata M., Balaji U. K., Madrenas J., *PSoC-Based Real-Time Data Acquisition for a Scalable Spiking Neural Network Hardware Architecture*, IEEE Third Ecuador Technical Chapters Meeting, pp. 1-6, 2018.

[20] Zhang M., Qu H., Belatreche A., Chen Y., Yi Z., *A Highly Effective and Robust Membrane Potential-Driven Supervised Learning Method for Spiking Neurons*, IEEE Transactions on Neural Networks and Learning Systems, vol. 30, Issue 1, pp. 123-137, 2019.