New Full Adders Using Multi-Layer Perceptron Network

Leila Dehbozorgi, Reza Akbari-Hasanjani, Reza Sabbaghi-Nadooshan*

Department of Electrical and Electronics, Islamic Azad University, Central Tehran Branch, Tehran, Iran, r_sabbaghi@iauctb.ac.ir

Abstract

How to reconfigure a logic gate for a variety of functions is an interesting topic. In this paper, a different method of designing logic gates are proposed. Initially, due to the training ability of the multilayer perceptron neural network, it was used to create a new type of logic and full adder gates. In this method, the perceptron network was trained and then tested. This network was 100% accurate to determine outputs based on inputs. The results of comparison showed that the multilayer perceptron network had higher velocity and less delay in most cases, and used a smaller number of neurons, which will reduce the loss of power. Meanwhile, implementation of these gates will require less space through the multi-layer perceptron network.

Keywords: Multilayer perceptron network, XOR, Full adder, Logic gate, Threshold.
Article history: Received 04-Mar-2020; Revised 05-May-2020; Accepted 22-Jun-2020.

1. Introduction

An artificial neural network is a parallel structure that can solve many complex problems, such as modeling, optimization and classification [1]. Some of the important applications of the neural network are recognition of pattern, diagnosis and identification of human speech, conversion of black and white images to color images, prediction of earthquakes, etc. [2-4].

Often, simulations of various artificial neural networks have been performed using software simulators and high-level programming language, but, due to the parallel structure of artificial neural networks, their hardware implementation is faster and more efficient with the help of FPGA and CPLD than their software simulations [1, 5]. In general, multilayer perceptron artificial neural networks usually consist of an input layer and an intermediate layer (hidden) and an output layer with a proportional number of neurons linked by weights [6-7].

S. Yellamraju et al. have proposed a CMOS technique for designing and implementation of an ecologically inspired neuron that has accepted several synaptic inputs. This circuit accepts synapses as inputs and produces an output waveform according to the activation level. Then, the realization of various logical gates has been investigated through this combination [8].

In [9], the logic gates of the AND and Exclusive-OR (XOR) bases were implemented using the perceptron artificial neural network and specific threshold values.

In another method of designing the gates, the astrocytes function is inspired by biological systems of nerves [10]. The use of such systems has been investigated for the production of logical operators. The resulting systems are simple and homogeneous meaning that only one type of neuron has been used with a spiking rule. With the help of these neurotic logic gates, a much more complex Boolean circuits with Cascade connections can be made and can be used to implement limited computing devices.

In [11], a pulsed neuron system was provided for the design of gates and logic circuits, in which, for synchronization between blocks, a series of concurrent blocks are used.

By using pulsed neural systems, we can use other methods to construct new gates by dividing neuron weights and germination [12]. With this method, complex calculations can be investigated.

A relatively vast research has been done to improve the algorithm of pulsed neural systems. In [13], synapses and their rules have been revised and

reconfigured. For example, the base systems that were constructed by 39 neurons in previous methods were designed by 30 neurons in [13].

In this research, firstly, according to the accuracy tables, the operation of the logic gate was trained to the multi-layer perceptron network, and, then, the 3-bit and 8-bit full adder circuits were designed by a multi-layer perceptron network and was generalized for N-bit adder.

The article is divided as follows: second part is the method used in this study. In Section 3, types of gates and their designed structure are introduced with the help of multi-layer perceptron network and three-bit full adder circuit. The fourth part examines the advantages and disadvantages of this plan and compares them with similar works. The fifth part examines the results of comparing this plan and its advantages.

2. Research method

A) MLP neural network

An artificial neural network is composed of many artificial neurons that are linked together according to specific network architecture [14]. The objective of the neural network was to transform the inputs into meaningful outputs (Fig. 1).

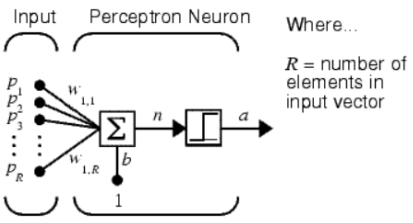

Fig. 1. Simple multilayer perceptron neural network [15].

Multilayer perceptron (MLP) network consists of an input layer, one or more hidden layers and one output layer. Each layer consists of multiple neurons [16-18]. An artificial neuron is the smallest unit that constitutes the artificial neural network (Fig. 2).

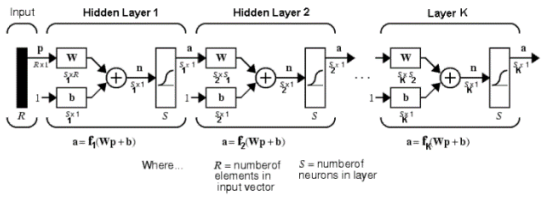

Fig. 2. K layers multilayer perceptron neural network.

B) Design and simulation of logic gates and full adder using multilayer perceptron network

Logic gates and full adder circuits were designed with multi-layered perceptron network. In order to improve the performance and reduce the calculation for each gate, a certain number of neurons (N1, N2, N3, ...) were used in each layer. Then, using the accuracy chart of each gate and the full adder circuit, the training and testing matrix was formed and the multi-layer perceptron network was trained and tested.

3. 3. Measurement, observation and calculation

In this study, with the use of the MATLAB (2016) software, the multi-layer perceptron network was trained and tested for design of the logic gate and the full adder circuits.

A) AND Gate

The AND gate represents the result of multiplying two input bit in the output (Formula (1)). In the design of this gate, the network was trained and finally, it was tested with a precision of 100%. The number of neurons in the first, second layers is 2, 1 respectively (Figure 3).

$$Y = AB \tag{1}$$

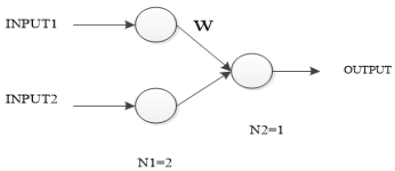

Fig. 3. The perceptron neural network for the implementation of the AND gate with (tan-sigmoid) function

The weights matrix for fully connected neural network is as below:

$$w = \begin{bmatrix} 0.2931 \\ 0.2931 \end{bmatrix} \tag{2}$$

B) OR Gate

The OR gate represents the result of summing two input bit in the output (Formula (3)). In the design of this gate, the network was trained and finally, it was tested with a precision of 100%. The number of neurons in the first, second layers is 2, 1 respectively.

$$Y = A + B \tag{1}$$

The weights matrix for fully connected neural network is as below:

$$w = \begin{bmatrix} 0.7881 \\ 0.7881 \end{bmatrix} \tag{2}$$

C) XOR gate

The XOR gate function is shown in formula 5. In the design of this gate, the network was trained and finally, it was tested with a precision of 100%.

The number of neurons in the first, second, and third layers is 2, 2, 1, respectively.

$$Y = A \oplus \mathrm{B} = A'B + B'A \tag{3}$$

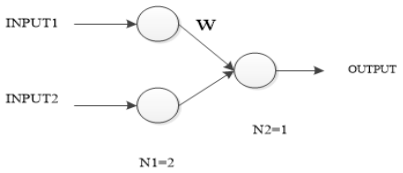

Fig. 4. The perceptron neural network for the implementation of the OR gate with (tan-sigmoid) activation function

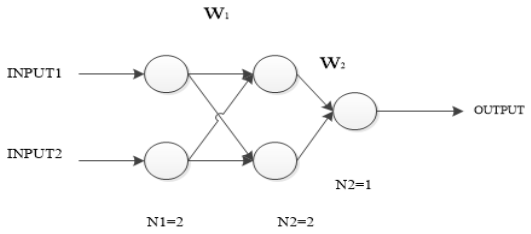

Fig. 5. The perceptron neural network for the implementation of the XOR gate with (tan-sigmoid) activation function

The weights matrix between the input layer and the hidden (middle) layer is W1 and the weights matrix between the hidden layer and the output layer, is W2.

$$W1 = \begin{bmatrix} 0.6550 & 2.4736 \\ 0.6541 & 2.4658 \end{bmatrix} \tag{4}$$

$$\mathrm{W2} = [2.3921\ 1.6857] \tag{5}$$

D) NOT gate

Since the gate of NOT is obtained from XOR of each data with a number, the trained perceptron network for the XOR gate was used in the design of this gate, so that, one of the inputs was always considered one.

E) Three-bit full adder

Generally, two blocks (two half adder) are used to design full adders, but here, with the help of multi-layer perceptron network training, a 3-bit full adder was designed directly. The entire three-bit adder received 3-bit input (usually in the upper classes of one of the digits, is the Carry in the previous floor) and then were added together and the sum (S: SUM) and the (C: CARRY) are calculated. With the change of the first, second, and the third bits (assuming the variation of the Carry), the sum and carry are represented by the help of multilayer neural network. These changes are considered in ten steps for these three bits (Fig. 6).

$$WS = A \oplus \mathrm{B} \oplus \mathrm{C} \tag{6}$$

$$\mathrm{C} = \mathrm{AB} + \mathrm{AC} + \mathrm{BC} \tag{7}$$

In designing a 3-bit full adder, the multi-layer perceptron network was once trained and tested to calculate the sum (SUM) and again to calculate the carry (CARRY). Each network was tested with a precision of 100%.

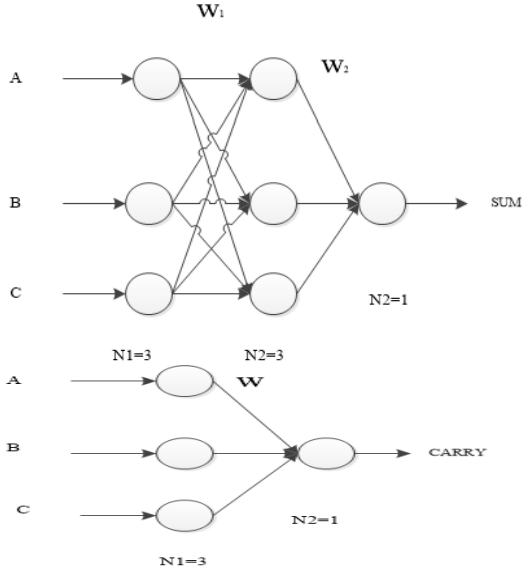

Fig. 6. a 3-bit full adder. A: The first bit and B: The second bit and C: The bit of the previous Carry

The weights matrix for the 3-bit full adder with fully connected neural network is as below:

$$SUM = \left\{ \begin{matrix} W1 = \begin{bmatrix} 3.8370 & 1.3049 & -0.3424 \\ 3.8381 & 1.2968 & -0.3359 \\ 3.8298 & 1.2858 & -0.3258 \end{bmatrix} \\ W2 = \begin{bmatrix} 3.8282 \\ -6.0522 \\ -4.5134 \end{bmatrix} \end{matrix} \right\} \tag{1}$$

$$CARRY = \left\{ W = \begin{bmatrix} 0.3530 \\ 0.3542 \\ 0.3553 \end{bmatrix} \right\} \tag{2}$$

F) Eight-bit full adder

To complete all eight-bit full adders, eight blocks of three-bit full adders were used. CARRY of the first layer output enters the CARRY of the second layer input and the CARRY of output of the second layer enters the CARRY of input of the third layer. These steps are repeated for each of the eight layers. This method can be used to generalize a three-bit full adder to an eight-bit and more (N-bit) full adder (Fig. 8). To sum 8-bits to the input, at first, two 8-bit matrixes (including binary codes) are simultaneously received from the program input. Then, the program begins to sum each 3-bit in each class, respectively, and transfers the CARRY of each class to the next class. In Fig. 9, the sum of the inputs in each five-minute interval has made the

corresponding changes to the output. The simulation results with MATLAB software are shown in Fig. 9.

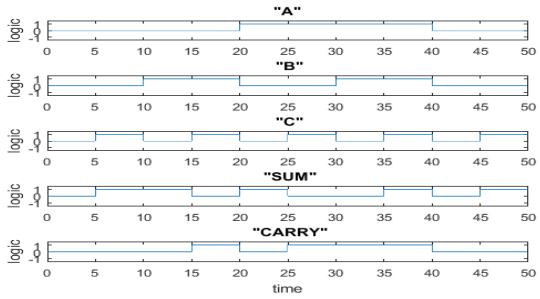

Fig. 7. A three-bit full adder display with a variable binary input in the 10-step change.

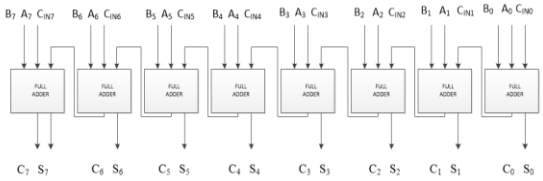

Fig. 8. All eight-bit full adders. A: The first bit and B: The second bit and C: The CARRY bit (The CARRY of the first layer is transmitted to the CARRY of the next layer).

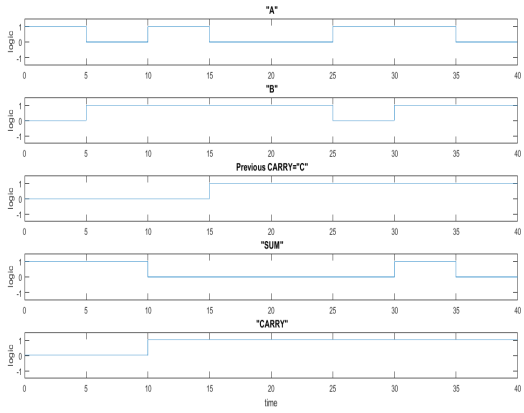

Fig. 9. 8-bit full adder display with variable binary input in 8 steps of input bit changes.

G) *N-bit full adder*

The 8-bit full adder program is written in such a way that by adding the dimensions of the input matrix, the corresponding output is calculated. To design an N-bit full adder, simply, put the number of inputs in the program as N, and after executing the program, the outputs will be stored in two matrices of the SUM and the CARRY.

A 16-bit full adder is considered, the simulation results with MATLAB software are shown in Fig. 10. In this graph, for more clarity, the input matrix bits and, consequently, the total sum (SUM) and the CARRY variable are shown every 5 seconds. This full adder also operates on the basis of a generalized three-bit full adder, except that in the program loop, the CARRY is transferred from the previous step.

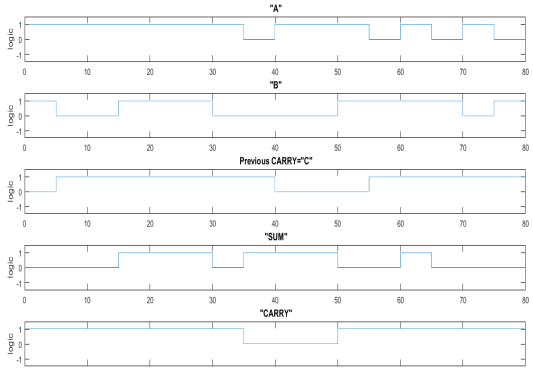

Fig. 10. Chart of 16-bit full adder input changes generalized from the 8-bit full adder.

In the following, a 32-bit matrix is applied to 8-bit full adder MATLAB code, and the adder easily calculates and displays the output matrices (Fig. 11).

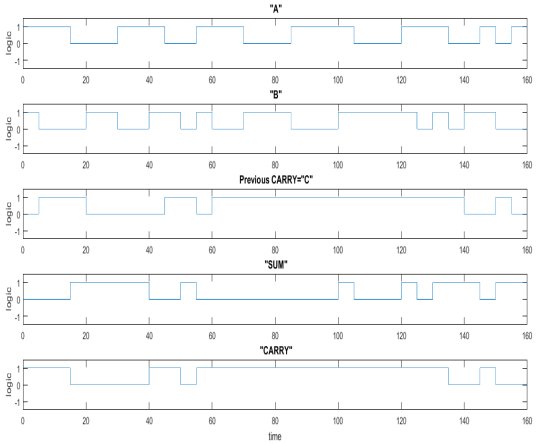

Fig. 11. 32-bit full adder input chart generalized from the 8-bit full adder.

4. Results and discussion

Logic gates can be designed in a variety of ways. Multi-layer perceptron network is a new and proposed method of designing logic gates and full adder circuit. In this paper, their designs have been investigated. In [16], another type of design has been studied. In Table 1, the proposed method of the perceptron network is compared with the method presented in [16]. As you can see, the number of neurons used in the proposed gates are less than the paper [16]. In [16], if we want to use different logical circuits, we need to apply synchronization blocks. However, there are no synchronization blocks in the design of different circuits using the proposed gates in this article. Therefore, the integrity of the circuit and their hardware implementation is possible at a smaller level.

Table.1.
Comparison of proposed Logic Gate Design and full adder characteristics with Paper [16]

	The total number of neurons	*The total number of layers*	*Weights matrix*
Proposed AND gate Designed with MLP	3	2	$W=\begin{bmatrix}0.2931\\0.2931\end{bmatrix}$
AND gate [16]	5	3	$W1=\begin{bmatrix}2&2\\2&2\end{bmatrix}$ $W2=\begin{bmatrix}2\\2\end{bmatrix}$
Proposed OR gate - Designed with MLP	3	2	$W=\begin{bmatrix}0.7881\\0.7881\end{bmatrix}$
OR gate[16]	5	3	$W1=\begin{bmatrix}4&4\\4&4\end{bmatrix}$ $W2=\begin{bmatrix}-2\\2\end{bmatrix}$
Proposed XOR gate Designed with MLP	5	3	$W1=\begin{bmatrix}0.6550&2.4736\\0.6541&2.4658\end{bmatrix}$ $W2=[2.3921\ 1.6857]$
XOR[16]	5	3	$W1=\begin{bmatrix}1&1\\1&1\end{bmatrix}$ $W2=\begin{bmatrix}1\\-1\end{bmatrix}$
Proposed full-adder Designed with MLP	$\begin{cases}SUM=7\\CARRY=4\end{cases}$	$\begin{cases}SUM=3\\CARRY=2\end{cases}$	$\begin{cases}SUM=\begin{cases}W1=\begin{bmatrix}3.8370&1.3049&-0.3424\\3.8381&1.2968&-0.3359\\3.8298&1.2858&-0.3258\end{bmatrix}\\W2=\begin{bmatrix}3.8282\\-6.0522\\-4.5134\end{bmatrix}\end{cases}\\CARRY=\begin{cases}W=\begin{bmatrix}0.3530\\0.3542\\0.3553\end{bmatrix}\end{cases}\end{cases}$
full-adder [16]	$\begin{cases}SUM=10\\CARRY=20\end{cases}$	$\begin{cases}SUM=6\\CARRY=12\end{cases}$	$\begin{cases}SUM=\begin{cases}W1=\begin{Bmatrix}\begin{bmatrix}2&2\\2&2\end{bmatrix}\\\begin{bmatrix}2\\-2\end{bmatrix}\end{Bmatrix}&W2=\begin{Bmatrix}\begin{bmatrix}2&2\\2&2\end{bmatrix}\\\begin{bmatrix}2\\-2\end{bmatrix}\end{Bmatrix}\end{cases}\\CARRY=\begin{cases}W3=\begin{Bmatrix}\begin{bmatrix}2&2\\2&2\end{bmatrix}\\\begin{bmatrix}2\\2\end{bmatrix}\end{Bmatrix}&W4=\begin{Bmatrix}\begin{bmatrix}2&2\\2&2\end{bmatrix}\\\begin{bmatrix}2\\-2\end{bmatrix}\end{Bmatrix}&W5=\begin{Bmatrix}\begin{bmatrix}2&2\\2&2\end{bmatrix}\\\begin{bmatrix}2\\2\end{bmatrix}\end{Bmatrix}\\&W6=\begin{Bmatrix}\begin{bmatrix}4&4\\4&4\end{bmatrix}\\\begin{bmatrix}4\\4\end{bmatrix}\end{Bmatrix}\end{cases}\end{cases}$

One method of implementing a full adder is by using logic gates and Boolean algebra. It should be noted that in this design, for a 3-bit, there is a need for three layers and five logic gates, and if we want to design an 8, 16, 32 or N-bit full adder, we are faced with a large volume of logic gates and complex circuits.

In [10], another type of design has been studied using a pulsed astrocyte network. With the help of the above formula (8), (9) and base gates in [10] the full adder circuit was designed. Fig. 12 compares the specifications of proposed full adder and papers in [10, 1999]. The proposed full adder designed with a multi-layer perceptron network used a very limited number of neurons to train and test the perceptron network, and then the generalized results of this network were used for N-bit using MATLAB programming software. It is noticeable that the hardware complexity of this method is much less than other methods and will occupy less orbital level.

The flexibility of this full adder for its generalization in calculating any dimension of the input matrix is another benefit of this N-bit full adder. Using fewer layers, fewer gates and weights matrix will reduce the power consumption of the circuit.

By comparing the logic gate and full adder circuit design with multi-layer perceptron network and other designs, we concluded that this method was prioritized in terms of the number of neurons and the level of implementation, and the speed of detection of output.

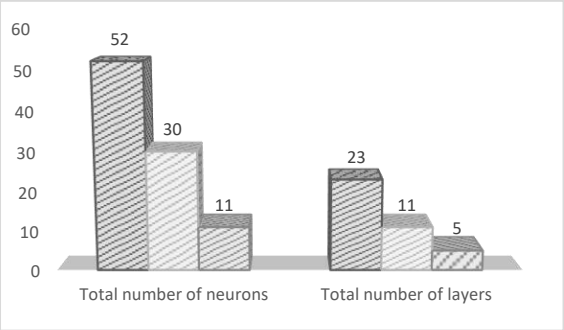

Fig. 12. Comparison of full adder characteristics with different methods

5. Conclusion

In this paper, the logic gates and N-bit full adder circuits were designed with the help of multi-layer perceptron network. Logical gates designed in this way are hardware-less complex and less space-consuming. Then, the 3-bit full adder circuit was designed with the multi-layer perceptron network. In this design, no logic gate was used, and therefore, the full adder speed increased, and the 3-bit full adder was generalized to the 8-bit. The programming of this full adder is such that any number of bits in the input matrix that is applied on it, the result (the sum and the CARRY) will be computed in the output. This method is prioritized in terms of the number of neurons and the level of implementation, and the speed of the detection of output compared to the other design. It also occupies less hardware space and is less complicated. It is suggested that in the future, this method is used in the design of memory, subtraction, and so on.

References

[1] R. Rezvani, M. Katiraee, A. H. Jamalian, SH. Mehrabi and A. Vezvae, "A New Method for Hardware Design of Multi-Layer Perceptron Neural Networks with Online Training", Conference Paper ·, DOI: 10.1109/ICCI-CC.2012.6311205, August 2012.

[2] G. M. N.Thilakarathna and Sri LankaW. K. I. L. "English Character Recognition of an Image and Voicing System", Proceedings of APIIT Business, Law& Technology Conference, ISBN978-955-7678-02-3, 2017.

[3] S. Miyata , "Automatic Recognition of Speed Limits on Speed-Limit Signs by Using Machine Learning", Journal of Imaging, 2017.

[4] L. Debozorgi, "Case Study of Seismic Signals of GHIR Station before Earthquake", Bulletin of earthquake Science and Engineering, 2017.

[5] A.Chvála , L. Nagy , J. Marek , J. Priesol ,D. Donoval , A. Šatka , " Neural Network for Circuit Models of Monolithic InAlN/GaN NAND and NOR Logic Gates", 14th International Conference on Design & Technology of Integrated Systems In Nanoscale Era (DTIS), 2019

[6] V. Kumar Ojha, A. Abraham and V.Sn´aˇsel1 1 Dept, "Metaheuristic Design of Feedforward Neural Networks: A Review of Two Decades of Research", Computer Science, Engineering Applications of Artificial Intelligence, 97–116, 2017.

[7] Z. Ali, I. Hussain, M. Faisal, H. Mamona Nazir, T. Hussain, M. Yousaf Shad, A. Mohamd Shoukry and S. Hussain Gani, "Forecasting Drought Using Multilayer Perceptron Artificial Neural Network Model" , Information Technology,2017.

[8] S.Yellamraju, S. Kumari, S. Girolkar, S. Chourasia, A. D. Tete, "Design of Various Logic Gates in Neural Networks", Annual IEEE India Conference, 2013.

[9] Ele, Sylvester I. and Adesola, W. A, "Artificial Neuron Network Implementation of Boolean Logic Gates by Perceptron and Threshold Element as Neuron Output Function", International Journal of Science and Research (IJSR), 2015.

[10] T. Song, P. Zheng, M.L. D. Wong and X. Wang, "Design of logic gates using spiking neural P systems with homogeneous neurons and astrocytes-like control", Information Sciences, Vol. 372, 380–391, Elsevier, 2016.

[11] S.Patel, J. Sahaya Rani Alex and N. Venkatesan, " Low-Power Multi-Layer Perceptron Neural Network Architecture for Speech Recognition", Indian Journal of Science and Technology, Vol 8(20), 2015.

[12] L.pan, G.paun and M.J.perez-Jimenez, "Spiking Neural P Systems with Neuron Division and Budding", 2016.

[13] T. Song, L. Pan and G. Paun, "Spiking neural P systems with rules on synapses", Theoretical Computer Science, vol. 529, 82–95, 2014.

[14] A. Fayyazi , S.Shababi , P.Nuzzo , S. Nazarian ; M. Pedram , "Deep Learning-Based Circuit Recognition Using Sparse Mapping and Level-Dependent Decaying Sum Circuit Representations", Design, Automation & Test in Europe Conference & Exhibition (DATE), 2019

[15] M. Natsui , T. Chiba , T. Hanyu , " MTJ-Based Nonvolatile Ternary Logic Gate for Quantized Convolutional Neural Networks", IEEE SOI-3D-Subthreshold Microelectronics Technology Unified Conference (S3S), 2018.

[16] N. Farha1, A. Louisa Paul, N. Kousar, R. Divakaran, " Design and Implementation of Logic Gates and Adder Circuits on FPGA Using ANN", International Journal for Research in Applied Science & Engineering Technology (IJRASET), 2016.

[17] A. Abroudi, F. Farokhi and F. Zahedi, "Instance selection for training of intelligent networks based on fast condensed nearest neighbor rule," 2012 IEEE Conference, pp. 1-4, 2012.

[18] L. Dehbozorgi, "Investigate two groups of features for short-term Iran earthquake prediction using seismic signals with neuro-fuzzy classifier and Multilayer Perceptron neural network," The first national conference on metaheuristic algorithms and its Applications in Science and Engineering, 2014.