Journal of ELECTRICAL ENGINEERING, VOL. 61, NO. 4, 2010, 222–228

NEW DIGITAL ARCHITECTURE OF CNN FOR PATTERN RECOGNITION

Emil Raschman — Roman Záluský — Daniela Ďuračková *

The paper deals with the design of a new digital CNN (Cellular Neural Network) architecture for pattern recognition. The main parameters of the new design were the area consumption of the chip and the speed of calculation in one iteration. The CNN was designed as a digital synchronous circuit. The largest area of the chip belongs to the multiplication unit. In the new architecture we replaced the parallel multiplication unit by a simple AND gate performing serial multiplication. The natural property of this method of multiplication is rounding. We verified some basic properties of the proposed CNN such as edge detection, filling of the edges and noise removing. At the end we compared the designed network with other two CNNs. The new architecture allows to save till 86 % gates in comparison with CNN with parallel multipliers.

K e y w o r d s: CNN, digital circuit, area consumption, pattern recognition

1 INTRODUCTION

The need to process ever larger amounts of information requires a constantly increasingly computational complexity of circuits performing processing. This advance in computational complexity can only be achieved by increasing the working frequency of the circuits which perform information processing or by using parallel processing. For real-time processing an enormous computing power is required that would cope with processing of vast amounts of incoming information constantly. One of the solutions of real-time applications uses neural networks. The neural networks achieve an enormous computing power basically thanks to parallel processing performed by a large number of simultaneously working elementary processors, the so-called cells. The next advantage of some neural networks is their ability to learn due to which we need not develop complex software for processing the data that occupy a lot of place in the memory. The system for data processing performs simple learning with a special training aggregate.

For image and pattern processing most frequently we use the so-called cellular neural networks marked as CNN. This type of networks contains a lot of cells (computing elements) that are interconnected analogically as the neurons in the human brain. An analogue architecture of this network was proposed by L.O. Chua a L. Yang [1, 2]. CNN are used for image processing and pattern recognition.

A CNN can be realized by means of analog or digital circuits. Various hardware implementations are presented in the literature [9–12]. The advantage of analog implementation is a smaller chip area and the advantage of digital implementation is easy modification for weight coefficients in the process of learning. In our work we focus on a digital neural network. Its advantage is a relatively easy implementation in the FPGA circuits. The basic properties of this type of neural network are the small chip area consumption, high speed and low power consumption.

The present VLSI circuits and technology allow to manufacture chips with a very high density of integration, which makes possible hardware implementation with a high number of neurons (neural cells) on the chip. Nowadays the biggest problem in hardware realization of neural networks is the chip area consumption. The cells should occupy the smallest area on the chip. In digital design, the largest area is occupied by the multiplier and the adder, hence we need to propose an alternative way of multiplication and addition in order to lower the chip area consumption.

We designed a new architecture of the digital neural network with multiplication signals decomposed in time by means of an AND gate. The hardware multiplication was replaced by the AND gate and a counter. The counter summarizes the output of the AND gate and transforms this output to a 5 bit number. The circuits also contain a Converter block which performs distribution of the signal in time before multiplication by the AND gate. The natural property of our multiplication is rounding. The design of the new architecture brought saving of 86.8 % of gates in comparison with a standard CNN with parallel signal multipliers.

2 THE BASIC PRINCIPLE OF THE CNN

Cellular neural networks are best for applications focused on pattern recognition. The advantage of the CNN is its simple architecture. The basic principles of the CNN are explained in [3–6]. The cellular neural network is usually a non-linear cellular network. This is a group of spatially distributed cells, where every cell is a neighbour to

* Slovak University of Technology, Department of Microelectronics, Ilkovičova 3, 812 19 Bratislava, Slovakia, emil.raschman@stuba.sk

DOI: 10.2478/v10187-010-0031-6, ISSN 1335-3632

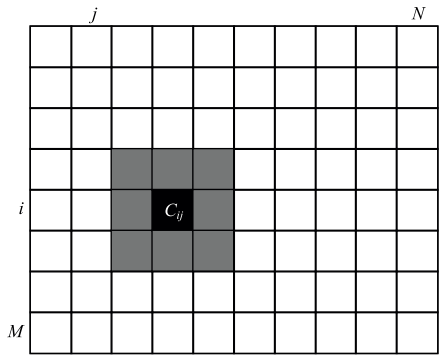

Fig. 1. CNN with size $M \times N$ and marked cell C_{ij} and its r-neighbourhood $= 1$

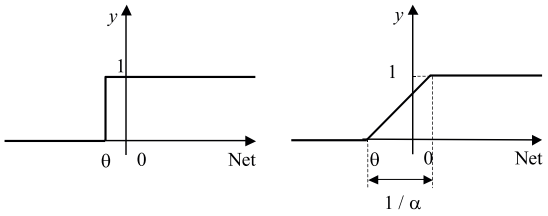

Fig. 3. Transfer functions (θ define boundary, when neuron stake is active)

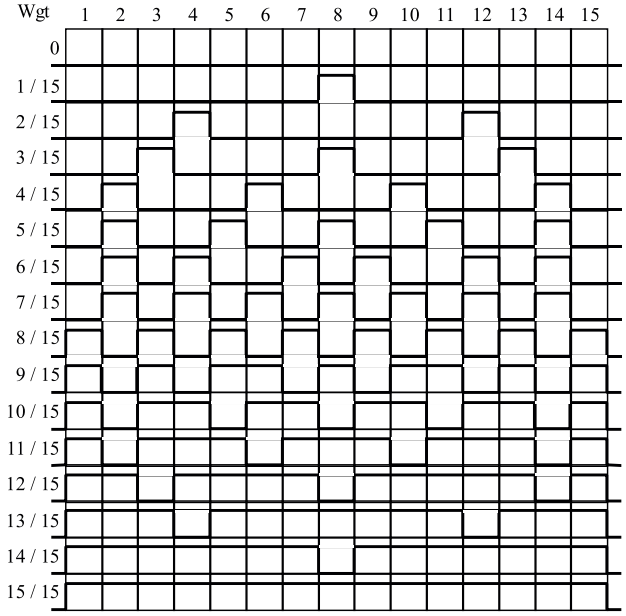

Fig. 4. Weights in the proposed 15 s system of the timing

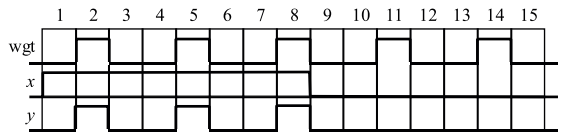

Fig. 5. An example of the evaluation for weight $wgt = 6/15$ and input $x = 11/15$

itself and is locally connected with the neighbouring cells in some extent of action referred to as r-neighbourhood (Fig. 1).

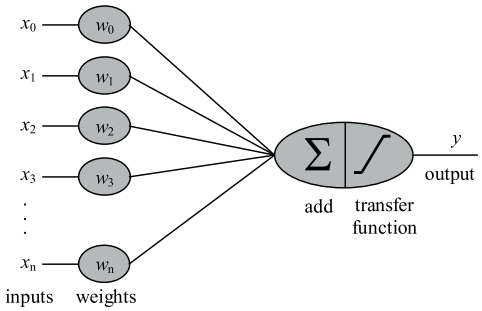

Fig. 2. Graphical representation of the CNN cell

The control input of the CNN is a weight matrix, where each coefficient represents some weight (importance) of the corresponding input. Then each input is multiplied by a certain weight constant. By summarizing this conjunction we get the function ***Net***.

$$\boldsymbol{Net} = w_1 s_1 + w_2 s_2 + \cdots + w_m s_m \,. \tag{1}$$

In this equation, coefficients w represent the weights and coefficients s represent the incoming signals from the surrounding cells. The output of cell y we get from a non-linear transform of ***Net***:

$$Y = f(\boldsymbol{Net}) \,. \tag{2}$$

Function $f(\,)$ is called the transfer or activation function. This function designates the output state of the cell. The block diagram of the cell of the CNN is in Fig. 2.

There exist several transfer functions [7, 8] as for example a sigmoid function, hard-limiter or threshold logic utilized in a few applications. Our CNN has a hard-limiter and a threshold logic function (Fig. 3). Parameter θ represents the boundary, when the neuron is in an active mode. For our network, parameter θ for the hard-limiter function is zero and for the threshold logic parameter θ is from the range $\langle -1, 1 \rangle$, and $\alpha = 1$. The choice of the transfer function depends on the application of the neural network.

By a proper choice of the weight matrix we can achieve that the CNN is able, for example, to remove noise. By choosing the matrix we set input conditions for input data processing by CNN.

3 THE NEW DESIGN OF DIGITAL CNN

The basic principle of the CNN is very similar to that of a biological neuron. The network consists of a number of basic computing elements, the so-called cells. The incoming inputs of the cells are multiplied by corresponding weight coefficients, then the results of multiplication are summarized and in the end the sum is conversed through the transfer function. Since all cells realize data processing by parallel calculation, the computing power of the

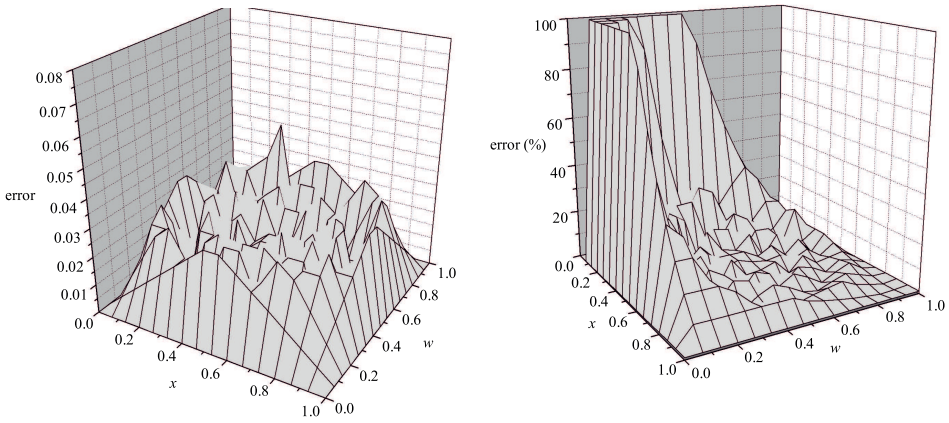

Fig. 6. Decomposition errors depending on the factor conjunction

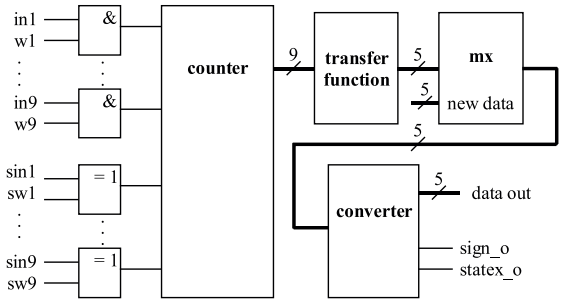

Fig. 7. Block diagram cell of the CNN

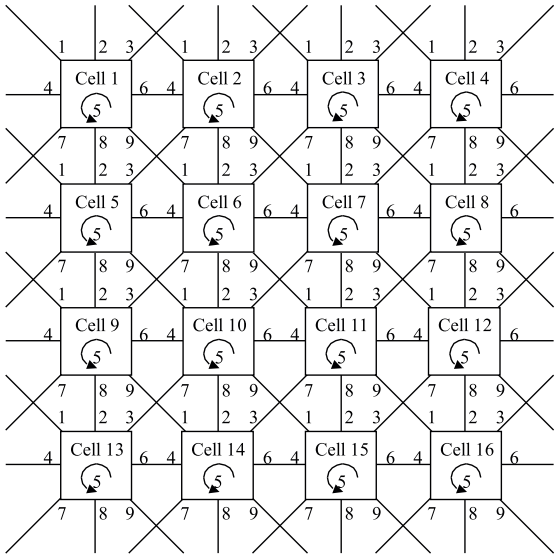

Fig. 8. Connections between the cells of CNN

CNN is directly proportional to the number of cells. The more cells contained in the network, the more information achieve synchronized processing. Therefore in the design of the CNN our effort is focused on minimizing the cell size, thereby providing a maximum number of cells on the chip. The size of the chip is one of the biggest problems in the design of CNNs. The largest area is reserved for the multiplexer, so we looked for alternative multiplications.

3.1 Multiplication of the signal using the AND gate

As an alternative possibility of multiplication we designed a circuit that multiplies the input values and weight coefficients by means of AND gate. The method of multiplication is based on the fact that by multiplication the input value must be converted to the time interval from the beginning of the time-window and the weight value has to be specially picked so that multiplication takes place when the input and weight signals pass the gate. We proposed a special coding for the weights.

We used a special system of 15 parts, *ie* one cycle of multiplication is divided into 15 parts (time-window). In such a time period it is possible to code 16 various values of weight or inputs. Decomposition signals for corresponding weight values are displayed in Fig. 4.

As an example we have an input value of $x = 8/15$ that we multiply by a weight of $w = 5/15$ according to Fig. 5. The input values must be converted to the time interval from the beginning of time axis, its length corresponding to the input size. The result at the end of the time window will be $y = 3/15$. The real value of the multiplication $xw = 0.18$ and our result of multiplication, as shown in Fig. 5, is $3/15 = 0.2$.

The natural property of the proposed method of multiplication is rounding. For verifying the effect of rounding on the result of the CNN we created a simulator as macro in Visual Basic for Application in Microsoft Excel. We used the simulator to recognize that we can neglect this rounding. For example the effect of multiple rounding upon intermediate result causes that the final result of network is delayed by about one iteration than without rounding.

The value of the error of computing depends on the factors of conjunction. Decomposition error is shown in Fig. 6. Picture a) is the decomposition absolute error. Its maximum value is 0.031. Decomposition of the absolute error is relatively constantly within the full range. Picture b) is the absolute error in percentage. In the figure we can see the effect of the rounding, where for very small numbers that were rounding to zero the absolute error is

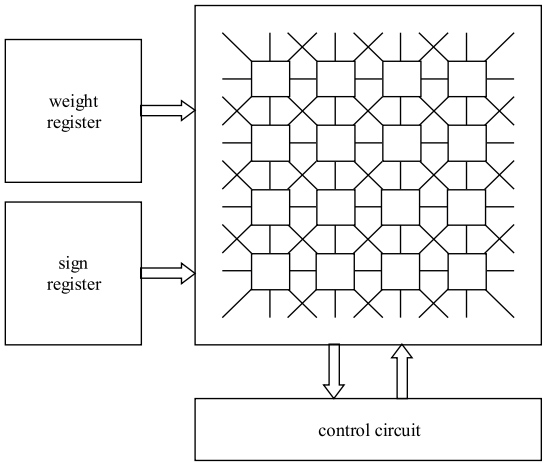

Fig. 9. Block diagram of CNN

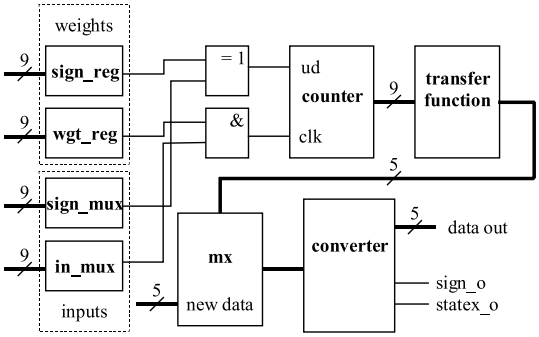

Fig. 10. Block diagram of one cell of the CNN presented at [13]

100 %. Nevertheless, these number are so small that we can ignore them. A proper selection of the weight coefficients allows to shift the calculation in the areas with minimum errors because for the weight matrix the ratio of coefficients is more important than their values.

3.2 The cell of proposed CNN

The proposed circuit is a digital synchronous circuit. The designed circuit has been realized by means of the descriptive language VHDL in Xilinx. The cell contains several sub-circuits. The block diagram of the cell is in Fig. 7. On the inputs of the cell there are 9 weight and 9 input signals and their corresponding signs. Eight input signals are connected to the outputs of the neighboring cells, and one input, the fifth in sequence, is connected to its own output of the cell because in the CNN theory the cell is a neighbour to itself. Then the inputs are multiplied with weights in the logical AND gate and the sign of inputs are compared with the logical XOR gate. The weight must be specially distributed in time, so that by passing through the AND gate the signals are multiplied. After multiplication by the AND gate the results are counting in the *counter* and consequentially converted by the *transfer function*. Our CNN has two transfer functions: hard-limiter and threshold logic function (Fig. 3). The choice of the transfer function depends on the application of the neural network. For edge detection, filling of the edges the hard-limiter function is better, and for noise removing the threshold logic function is better.

The converted signal is coming through the multiplexer *mx* to the block *converter*. The multiplexer *mx* allows entering new input values to the network. Block *converter* has two functions: it contains a register, where the result is saved, from where it can be read (*data out*), and a circuit converting result to the time interval corresponding with size of results (*statex_o, sign_o*), which is fed to the surrounding cells.

3.3 A novel architecture of the CNN

The CNN consists of a field of cells, each cell being coupled with all the nearest neighbours, thus the output of the cell is the input to all surrounding cells. These coupled cells in the CNN are displayed in Fig. 8. In this picture we can see that every cell is on the fifth input coupled with its own output because in the CNN theory every cell is a neighbour to itself too. The boundary cells contain inputs which are not coupled with the surrounding cells because there do not exist neighbouring cells. These inputs are connected to log. "0".

The general design of the CNN is in Fig. 9. The complete network contains a network of coupled cells, weight and sign registers and a control circuit. The weights and their signs from registers are fed to each cell. The *control circuit* performs synchronization of the whole circuit.

4 THE COMPARISION OF OUR CNN WITH OTHERS

The CNNs new architectures were compared with other digital CNNs. For comparison we selected the following networks:

- Our previous CNN [13].
- "Standard" digital CNN with parallel signed multipliers.

All networks compared were designed for processing 5-bit data, where 4 bits represented the values and highest bit represented sign of values.

1 Our previous CNN

This network was previous to our new CNN architecture. The network was proposed as a synchronous digital circuit utilizing multiplication of signals by decomposition in time by means of a logical AND gate. The main difference against the new architecture is a low speed and circuit complexity. The block diagram of one cell of this network is shown in Fig. 10. Every cell contains a weight register which consists of *weg_reg* storing the value of the weight and *sign_reg* containing information about the sign of the weight. On the input there are two multiplexers that are multiplexing the inputs (incoming from the surrounding cells) and their sign. The output of the weight

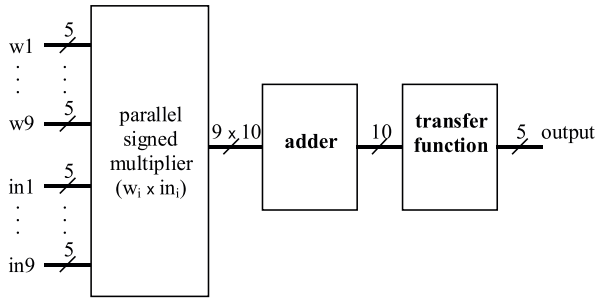

Fig. 11. Block diagram of one cell of "standard" CNN containing parallel signed multipliers

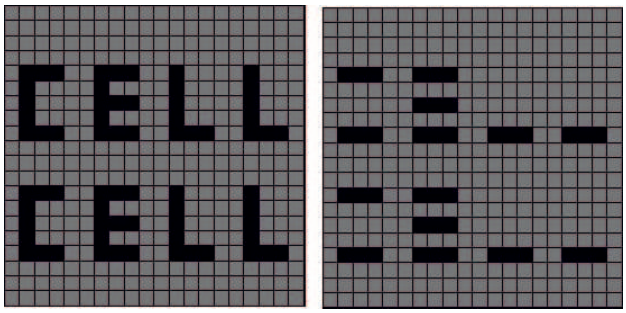

Fig. 12. Detection of horizontal edges

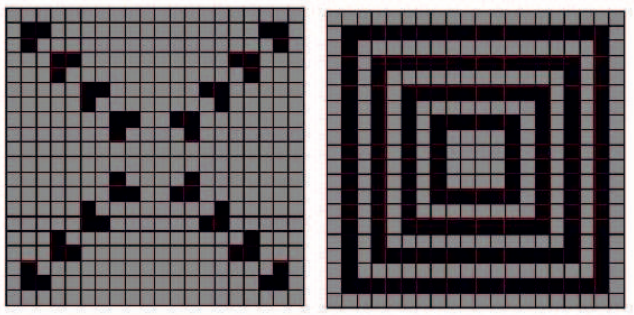

Fig. 13. Completion of missing edges

register is multiplication with the output of the multiplexer *in_mux* in a logical AND gate and their signs are compared in a logical XOR gate. The results of multiplication are added in the block *counter*. The output of the *counter* is converted by the threshold logic function in block *transfer function* and thereafter converted to the time interval corresponding to the values of the output in the block *converter*.

Multiplication of inputs with weights is realized sequentially. In the first step the inputs of all cells are multiplied with the first weight and the result is saved in the *counter*. The next step inputs are multiplied with the second weight and the result is saved in the *counter*, *etc.* Calculation of one iteration step of the network takes 9×15, thus 135 clock cycles.

2 "Standard" digital CNN with parallel signed multipliers

This network was designed by means of standard parallel signed multipliers. The block diagram of one cell of this network is in Fig. 11. Block *parallel signed multiplier* contains 9 signed multipliers which simultaneously multiply all weights with appropriate inputs. This block contains 9 outputs corresponding to 9 results of multiplication. These results are summarized in the *adder* and subsequently converted by the *transfer function* realized by the threshold logic or hard limiter function.

5 RESULTS

For verification of the properties and behaviour of the proposed neural network we need a sufficient number of cells. The proposed network is of fully cascade type, *ie* we can create the network with an optional number of cells. In our experiments we used application for 400 cells. Then the input of the neural network is an input matrix of 20×20 in size and the weight matrix has a size of 3×3. During testing the networks we focused mainly on the properties of the network such as the edge detection, fill-up of missing edges and noise removing. The results of simulation are in Figs. 12–14. During these simulations we applied a hard-limiter transfer function. For better interpretation we displayed the input to output matrix also in a graphical form. Each cell can have 31 different values, which are represented by 31 shades of gray. First we verified the network facilities by detection of horizontal

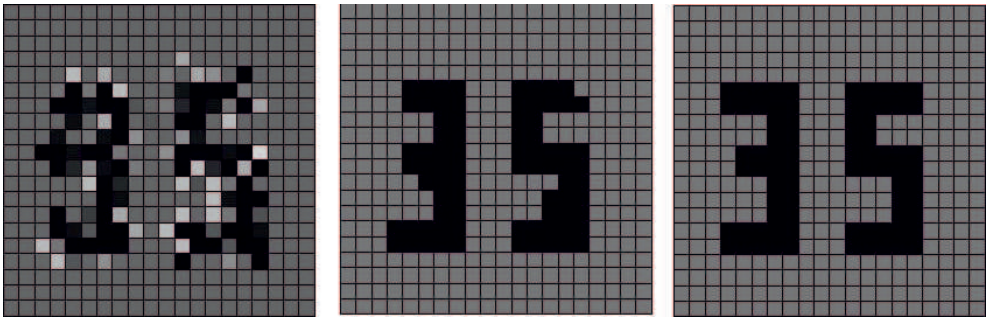

Fig. 14. Noise removal, a) Input, b) Output after the use of an inappropriate weight matrix, c) Perfect noise removal

Table 1. Comparison of parameters for CNN cell

Cell of the CNN	Parameters					
	Speed of one iteration	Bells	Flip Flops	Total	Max. frequency	Time of calculation of one iteration
CNN with 5bit signed parallel multipliers	4 CLK cycles	1207	208	1415	86 MHz	46.5 ns
Our previous CNN	135 CLK cycles	272	189	461	330 MHz	409 ns
New design of the CNN	15 CLK cycles	155	32	187	369 MHz	40.65 ns

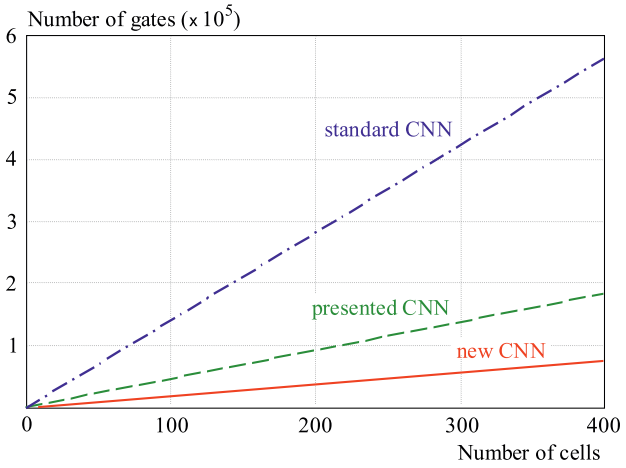

Fig. 15. Comparison of the CNNs

edges. As an input we used a pattern with two words "CELL" containing 14 horizontal edges. In Fig. 12 one can see that the required output is achieved already after the second iteration. Further we verified the network facilities for filling-up the missing edges (Fig. 13). As inputs to the network we used the corners. For connecting these corners to the network we needed 7 iterations. Finally we tested the network facilities as the noise removing from picture (Fig. 14). The input was the number "35" with noise of approx. 30 %. In this figure, *output_1* is the result of incorrectly choosing the weight matrix and *output_2* is the result after properly choosing the weight matrix. In this case, noise was 100 % filtered.

From the presented examples it is possible to conclude that the network satisfies the set requirements. The speed and correctness of the output depends on properly choosing the transfer function and weight matrix. The weight matrix was defined intuitively — "trial-and-error". In the future we would like to set the weight matrix by learning the network.

The main aim of our work was to propose a new architecture of the CNN with an alternative way of multiplication that allows to reduce the chip area. The main parameters compared were the speed and area consumption (number of gates) of the network. In Table 1 there is a comparison the designed network with other neural networks. Our new designed network we compared with network, which are presented in [13] and with standard network containing 5-bits parallel signed multipliers.

Designed circuit was realized by means of description language VHDL in development environment Xilinx. After synthesis of CNN for FPGA "Xilinx Virtex 5 XC5VFX30T" we were given results, which are shown in the "Table 1". The main request of our design has been to propose CNN, which could occupy the minimum of the chip area. Parameters representing the size of one cell are divided in the three columns *Bels* (Lut, mux, xor, inv, ...), *Flip Flops* (flip flops) and *Total* (total number of gates). The network with parallel multipliers has the biggest area consumption. One cell in the standard network with parallel multipliers is created by 1415 of gates. Our previous design contains 461 of gates, which is 3 times less than in the standard network. The new architecture networks has the smallest area consumption. The cells of our new design network have only 187 gates, which is 7.5 times less than in the standard network.

The second parameter was the speed of the circuit. The standard network with parallel multipliers needed the smallest number of clock signals to calculate one iteration (4 clock cycles), though its maximal frequency is only 86 MHz. The time of calculation of one iteration is 46.5 ns. The quickest was the network with the new architecture, which needed 15 clock cycles but its maximal frequency is 369 MHz, thus the time for one iteration is 40.65 ns. The slowest was our previous model, which needed 409 ns for one iteration.

The cell of the new network has 7.5 times less gates than the standard network with parallel multipliers (which is reduction by 86.8 % of gates) and its speed is little higher (calculation of one iteration is about 5.85 ns shorter). Due to parallel calculations of all cells the speed of the network with an optional number of cells is given as the speed of one cell calculation. Duration of one iteration is 15 clock cycles. This speed of network is sufficient. In Fig. 15 there is a dependence of the consumption of gates versus the number of cells. With an increasing number of cells, also the difference between the number of gates of each CNN increases.

6 CONCLUSION

We designed a new architecture of the digital neural network, with multiplication signals decomposed in time by means of an AND gate. The natural property of our multiplication is rounding. The values of the input and output from the cell can be obtained by values from -1 to 1 with a step of $1/15$, which presented 31 shade of gray. We verified some basic properties of the proposed CNN such as edge detection, filling of the edges and noise removing.

In comparison with the standard CNN with parallel multipliers, our designed network is 4 times slower but allows to save up to 86 % of gates and thereby allows to create a network with identical parameters with a fundamentally large number of cells. In Fig. 15 we see that our new CNN needs a lower number of gates for a given amount of cells than other compared networks.

In the future we would like to create a network implementation in to FPGA chip and connect it with a PC, which requires creating a communication and user interface and defining an input weight matrix based on learning networks by means of an incoming learning set of data.

Acknowledgment

This contribution was supported by the Ministry of Education of the Slovak Republic under grant VEGA No 1/0693/08 and conducted in the Centre of Excellence CENAMOST (Slovak Research and Development Agency Contract No. VVCE-0049-07).

References

[1] CHUA, L. O.—YANG, L.: Cellular Neural Network: Theory, IEEE Trans. Circuits Systems **35** (Oct 1988), 1257–1272.

[2] CHUA, L. O.—YANG, L.: Cellular Neural Network: Aplications, IEEE Trans. Circuits Systems **35** (Oct 1988), 1273–1290.

[3] HNGGI, M—MOSCHYTZ, G. S.: Cellular Neural Network: Analysis, Design and Optimalization, Kluwer Academic Publisher, Boston, 2000.

[4] SORDO, M.: Introduction to Neural Networks in Healthcare, Open Clinical Knowledge Management for Medical Care, Oct 2002.

[5] LARSEN, J.: Introduction to Artifical Neural Network, 1 st Edition, Section for Digital Signal Processing Department of Mathematical Modeling Technical University of Denmark, Nov 1999.

[6] SEUNG, S.: Introduction to Neural Networks: Lecture 1, The Massachusetts Institute of Technology — The Seung Lab, Sep 2002.

[7] KVASNIČKA, V.—BEŇUŠKOVÁ, Ľ—POSPÍCHAL, J.—FARKAŠ, I.—TIŇO, P.—KRÁĽ, A.: Introduction to Theory of Neural Network, IRIS, 1997. (in Slovak)

[8] JEON, J.: Fuzzy and Neural Network Models for Analyses of Piles, A dissertation submitted to the Graduate Faculty of North Carolina State University, Raleigh, North Carolina, Aug 2007.

[9] RESTREPO, H. F.—HOFFMAN, R.—PEREZ-URIBE, A.—TEUSCHER, C.—SANCHEZ, E.: A Networked FPGA-Based Hardware Implementation of a Neural Network Application, In (K.L. Pocek and J.M. Arnold Proceedings of the IEEE Symposium on Field Programmable Custom Computing Machines, FCCM'00, eds.), pp. 337–338, Napa, California, USA, Apr 17–19, 2000. IEEE Computer Society, Los Alamitos, CA.

[10] MUTHURAMALINGAM, A.—HIMAVATHI, S.—SRINIVASAN, E.: Neural Network Implementation Using FPGA: Issues and Application, International Journal of Information Technology **4** No. 2 (2008).

[11] BOUBAKER, M.—KHALIFA, K. B.—GIRAU, B.—DOGUI, M.—BEDOUI, M. H.: On-Line Arithmetic Based Reprogrammable Hardware Implementation of LVQ Neural Network for Alertness Classification, IJCSNS International Journal of Computer Science and Network Security **8** No. 3 (Mar 2008).

[12] LARKIN, D.—KINANE, A.—MURESAN, V.—O'CONNOR, N.: An Efficient Hardware Architecture for a Neural Network Activation Function Generator, Lecture Notes in Computer Science, 2006.

[13] RASCHMAN, E.—ĎURAČKOVÁ, D.: The Novel Digital Design of a Cell for Cellular Neural Network, Proceedings of the Electronic Devices and Systems, Brno, Sep 2008, pp. 322–325.

Received 4 December 2009

Emil Raschman received the MS degree in Electronics from Slovak University of Technology in Bratislava, Slovakia, in 2007. Since September 2007 he has been PhD student at Microelectronics Department of Slovak University of Technology. He is the author or co-author of more than 10 papers presented at international conferences. His main research interests are Digital design and neural networks.

Roman Záluský received the MS degree in Electronics from Slovak University of Technology in Bratislava, Slovakia, in 2008. Since September 2008 he has been PhD student at Microelectronics Department of Slovak University of Technology. He is the author or co-author of more than 10 papers presented at international conferences. His main research interests are Digital design and neural networks.

Daniela Ďuračková (Prof, Ing, CSc), received her MSc and CSc (PhD) from the Faculty of Electrical Engineering and Information Technology, Slovak University of Technology in 1974 and 1981 respectively. Since 1991 she has been an associate Professor at the Microelectronics Department of the Faculty of Electrical Engineering and Information Technology, Slovak University of Technology in Bratislava. The main field of her research and teaching activities has moved from semiconductor devices towards the design of analog and digital ASICs and neural network implementation on-chip.