jet INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2010, VOL. 56, NO. 4, PP. 367–373
Manuscript received October 21, 2010; revised November 2010. DOI: 10.2478/v10177-010-0048-6

Programmable, Asynchronous, Triangular Neighborhood Function for Self-Organizing Maps Realized on Transistor Level

Marta Kolasa, Rafał Długosz, and Krzysztof Bieliński

***Abstract*—A new hardware implementation of the triangular neighborhood function (TNF) for ultra-low power, Kohonen self-organizing maps (SOM) realized in the CMOS 0.18μm technology is presented. Simulations carried out by means of the software model of the SOM show that even low signal resolution at the output of the TNF block of 3-6 bits (depending on input data set) does not lead to significant disturbance of the learning process of the neural network. On the other hand, the signal resolution has a dominant influence on the overall circuit complexity i.e. the chip area and the energy consumption. The proposed neighborhood mechanism is very fast. For an example neighborhood range of 15 a delay between the first and the last neighboring neuron does not exceed 20 ns. This in practice means that the adaptation process starts in all neighboring neurons almost at the same time. As a result, data rates of 10–20 MHz are achievable, independently on the number of neurons in the map. The proposed SOM dissipates the power in-between 100 mW and 1 W, depending on the number of neurons in the map. For the comparison, the same network realized on PC achieves in simulations data rates in-between 10 Hz and 1 kHz. Data rate is in this case linearly dependend on the number of neurons.**

***Keywords*—Self-Organizing Maps, parallel signal processing, CMOS realization, low energy consumption, digital circuits.**

I. Introduction

SELF-ORGANIZING MAPS (SOM), broadly described in the literature, are used in various applications. They are powerful tools frequently used in classification of the signals that due to their nature can not be easy described mathematically. The SOMs are powerful tools used, for example, in medical healthcare systems in classification of various biomedical signals [1], [2]. Different network architectures of this type with different learning algorithms have been proposed. One of them is the Kohonen SOM, which is often referred to as the classical approach [3]. This SOM is trained according to the following formula:

$$W_j(l+1) = W_j(l) + \eta(k)G(i,j,R,d)[X(l) - W_j(l)] \quad (1)$$

where $\eta(k)$ is the learning rate in the k^{th} training epoch, W_j are the weight vectors of particular neurons in the map, and X is an input training pattern in an l^{th} cycle. The neurons that belong to the winners neighborhood are trained with different intensities, depending on the neighborhood function (NF) $G()$ of the topological distance, $d_{i,j}$, between the winning neuron i^{th} and a given j^{th} neurons in the map. In the classical approach a simple rectangular neighborhood function (RNF) is used, defined as [3], [4]:

$$G(i,j,R,d) = \begin{cases} 1 & \text{for } d(i,j) \le R \\ 0 & \text{for } d(i,j) > R \end{cases} \quad (2)$$

where R is the neighborhood range, which is decreased after each training epoch.

The common opinion is that better results can be achieved in case of using the Gaussian neighborhood function (GNF) instead of the rectangular one [5]. The Gaussian function is defined as follows:

$$G(i,j,R,d) = \exp\left(-\frac{d^2(i,j)}{2R^2}\right) \quad (3)$$

Various hardware realizations of the Gaussian function have been proposed [6], [7], [8]. The reported solutions usually are based on the analog circuits. On the other hand, in the SOMs with large numbers of neurons, in which the neighborhood mechanism must be distributed over a large chip area, digital solutions seem to be more efficient, as they are robust against the variation of external parameters, such as the supply voltage, the environment temperature, the noise and various technology nonidealities that occur during the chip fabrication process. The main disadvantage of a digital realization of the GNF is relatively large complexity of such circuit. It makes difficult its implementation in large neural networks with the neurons operating in parallel, in which low chip area and low power disspation are required [6].

To solve this situation the authors have recently focused on the triangular neighborhood function (TNF). They proposed an efficient digital realization of this function. Detailed investigations revealed that for a variety of the combinations of different network parameters the TNF offers a very similar performance as in case of using the GNF [9]. The TNF can be expressed as follows:

$$G(i,j,R,d) = \begin{cases} -a(\eta_0)\cdot d(i,j) + c & \text{for } d(i,j) \le R \\ 0 & \text{for } d(i,j) > R \end{cases} \quad (4)$$

where $a()$ is the assumed steepness of this function, η_0 is the winning neurons learning rate, while c is a bias value. All

M. Kolasa is with the Institute of Electrical Engineering, University of Technology and Life Sciences, ul. Kaliskiego 7, 85-796, Bydgoszcz, Poland (e-mail: markol@utp.edu.pl).

R. Długosz is with the Faculty of Telecommunication and Electrical Engineering, University of Technology and Life Sciences, ul. Kaliskiego 7, 85-796, Bydgoszcz, Poland (e-mail: rafal.dlugosz@gmail.com).

K. Bieliński is with White Electronics, Bydgoszcz, Poland (e-mail: krzysztof.bielinski@whiteelectronics.pl).

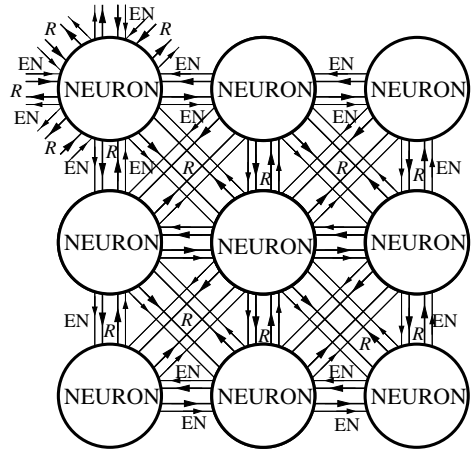

Fig. 1. The proposed realizations of the Kohonen SOM – the neighborhood scheme.

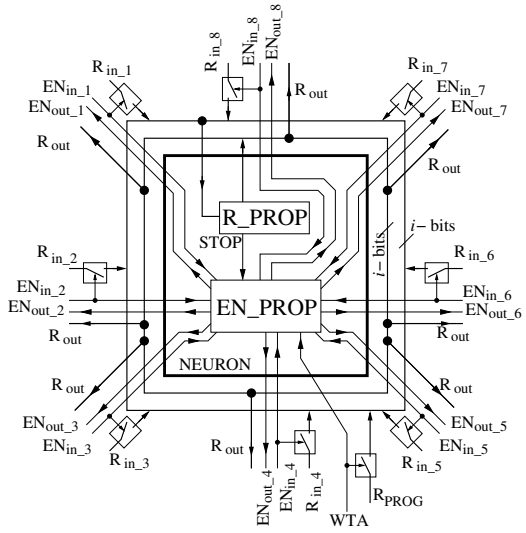

Fig. 2. Simplified model of the neuron used in the proposed hardware implementation of the Kohonen SOM.

these parameters decrease toward zero after each, k^{th}, epoch. Such scheme enables rapid changes of the neuron weights at the beginning of the learning process, and only tuning their values at the end.

II. An Overview of the Proposed Neighborhood Mechanism

The programmable asynchronous neighborhood mechanism proposed by the authors is a very fast and energy efficient solution [10]. A general diagram of the proposed SOM is shown in Figure 1. The simplified structure of a single neuron used in this network is presented in Figure 2. For the simplicity, only these blocks that are directly used by the neighborhood mechanism have been shown in this diagram. Detailed structures of these components are shown in Figure 3.

The proposed neighborhood mechanism works as follows. First the winning neuron receives an initial value of the radius R (the R_{PROG} signal in Figure 2), which is equal to the

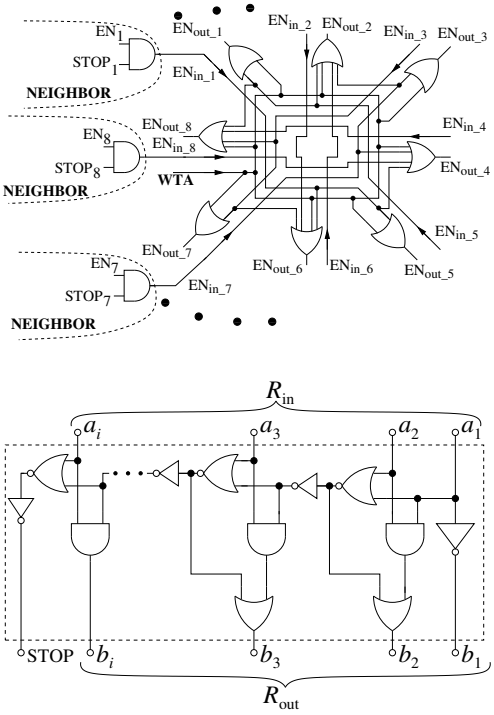

Fig. 3. Diagrams of the blocks used in proposed neighborhood mechanism: (top) the enable signal EN_PROP and (bottom) radius signal R_PROP propagation blocks.

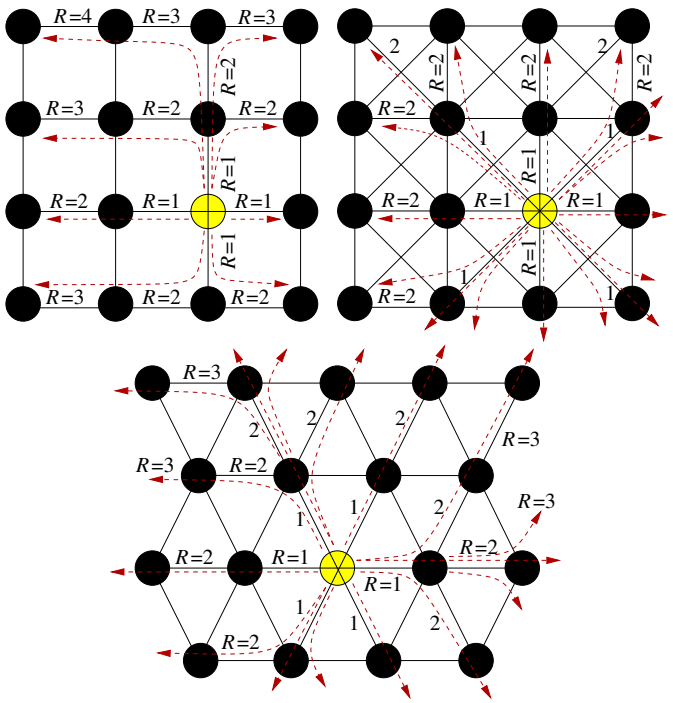

Fig. 4. Basic topologies of the SOM: (top) the rectangular one with four neighbors (Rect4), (middle) the rectangular with 8 neighbors(Rect8), and (bottom) the hexagonal one (Hex).

topological distance d. This externally programmed signal is sent by the winning neuron in all directions, which are allowed in a given topology, as shown in Figure 4. In parallel with this

signal an enable (EN) signal is also sent in all directions. As soon as a given neighboring neuron receives the EN signal, it allows the R_{in} signal that comes from the same direction to enter its internal R_PROP circuit, which decreases it by one and then resends it also in all directions. To prevent collisions, i.e. the situation in which one neuron receives the R signal from more than one direction, the EN_PROP circuit has been designed in such a way to enable the propagation of the EN signal only in selected directions, which are always out of the winning neuron. The proposed scheme resembles a wave that spreads out of the winning neuron. At each following ring of neurons surrounding the winner the value of the R signal is decreased by 1. If the value of this signal reaches the value 0, the further propagation of the EN signal is prohibited. For the R_{PROG} signal set to its maximal value, the neighborhood range covers the entire map. This means that in this case each neuron in the map is allowed to adapt its weights. The EN_PROP and the R_PROP circuits are shown in Figure 3. In such form the neighborhood mechanism can be directly used in the classic SOM with the RNF. In this case the learning rate η, which is equal in all neurons, is externally reprogrammed after each epoch in the same way as the R_{PROG} signal.

The network topology, defined as a grid of neurons, plays an important role in the overall learning process. This parameter determines which neurons belong to the winners neighborhood for a given value of the distance d [3], [4], [5]. The most frequently used topologies are the rectangular one with either four or eight neighbors and the hexagonal grid, as shown in Figure 4. In this paper they are referred to as Rect4, Rect8 and Hex, respectively. The authors' previous investigations shows that particular topologies are suitable for different combinations of remaining network parameters, as well as different sizes of the map (the numbers of neurons). For this reason they proposed the programmable solution, which can operate with all these topologies on a single chip [10].

III. An Influence of Particular Network Parameters on the Training Process

A. An Influence of Type of the Neighborhood Function

The comparative study for all three neighborhood functions described in first section by means of the software model of the SOM has been presented in [9]. This study shows that the triangular neighborhood function is a very good approximation of the Gaussian one. It is worth mentioning that the majority of reported applications of the Kohonen SOM were realized as software systems. In such systems both the TNF and GNF can be very simply realized as single instructions, and the conclusion above is of second importance in this case. For this reason, to our knowledge, the investigations presented in [9] have not been undertaken before. A different situation is if such networks are realized on transistor-level as application specific integrated circuits (ASIC) [11]. Such chips find an application in ultra-low power portable devices, e.g. in Wireless Sensor Networks (WSN) [12], which are more and more frequently used in medical healthcare systems. In this case the conclusion drawn above is very important, as application of the TNF instead of the GNF enables the simplification of the overall circuit structure and significant reduction of both the chip area and the energy consumption.

B. An Influence of the Signal Resolution at the NF Output

The second important aspect is the influence of the signal resolution at the output of the NF block on the overall circuit complexity i.e. the power dissipation and the chip area, as well as on the learning quality of the SOM. Selected results presented in this section show that there exists a trade-off between these two aspects. The number of transistors used in the neighborhood mechanism linearly depends on the signal resolution. On the other hand, decreasing the signal resolution below some critical values disturbs the overall learning process.

To show this problem in numbers the learning process of the SOM has been evaluated by use of the quantization error (Q_{err}), which is a commonly used criterion in such cases. The Q_{err} can be expressed as follows:

$$Q_{\mathrm{err}} = \frac{\sum_{j=1}^{m} \sqrt{\sum_{l=1}^{n} (x_j - w_{j,l})^2}}{m} \tag{5}$$

The m parameter is the number of the learning patterns, X, in the input data set, while n is the number of the network inputs.

In this section are presented selected simulation results by means of the software model of the network for different signal resolutions. The network was trained with data either regularly or randomly distributed in the input data space. The number of training patterns X depends on the map sizes. For example, the map with 16x16 neurons was trained with either 1280 or 2560 training patterns, which are either regularly or randomly distributed in the input data space. The map with 10x10 neurons was trained with either 500 or 1000 learning patterns, respectively.

The results in Figures 5 – 7 are shown versus an initial value, $R_{\max}$, of the neighborhood radius R. The $R_{\max}$ parameter is the radius in the first epoch after starting the learning process. The influence of $R_{\max}$ on the quantization error for the RNF, given by (2), has been studied by the authors in [13]. It has been demonstrated that for different input data sets and different other network parameters varying in the wide range, the optimal values of the $R_{\max}$ are usually very small, even for large SOMs with hundreds neurons. This conclusion is in contrast to a very common opinion that $R_{\max}$ at the beginning of training process should be large enough to cover at least half of the map. In hardware realization this conclusion is important, as low values of $R_{\max}$ allow for further reduction of the circuit complexity [13].

The results in Figures 5 and 6 are shown for two inputs and data regularly distributed in the input data space, for the maps with 16x16 and 10x10 neurons, respectively. Figure 7 shows example results for the map with 16x16 neurons, for three inputs and input data randomly distributed.

On the basis of presented results some conclusions can be drawn. In case when the network is trained with regular data it is possible to point out such values of $R_{\max}$, for which the

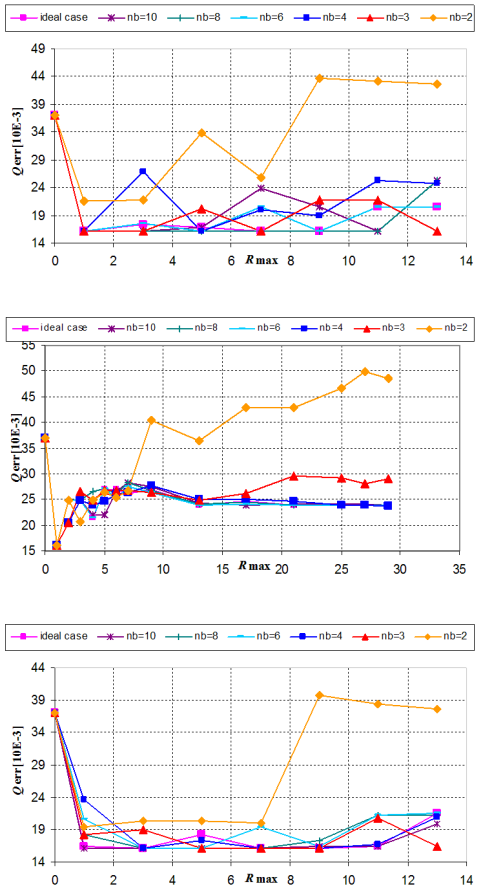

Fig. 5. Quantization error after completing the learning process for different resolutions of the $\eta \cdot G()$ signal, for 16x16 neurons, for: (a) rect8 (b) rect4 (c) hex topology, for two inputs and data regularly distributed in the input data space.

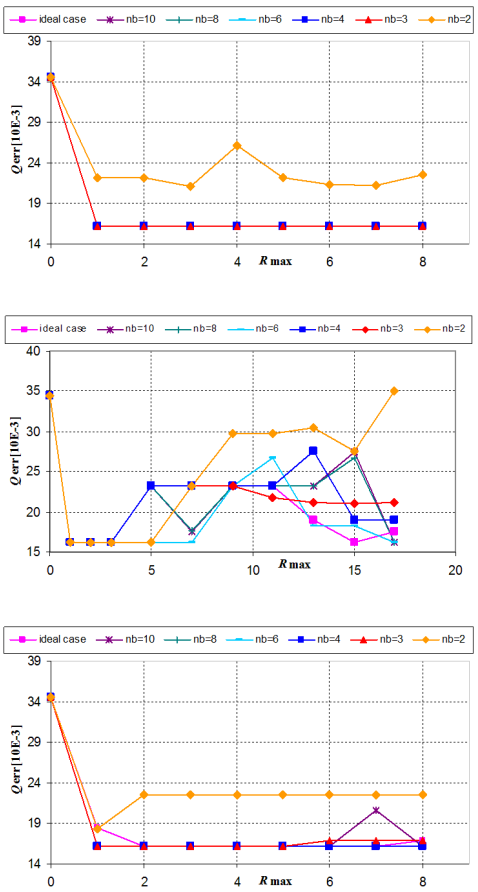

Fig. 6. The same results as in Figure 5 for 10x10 neurons in the map.

map becomes properly organized for all topologies even for the resolution of 3 bits. This case is shown in Figure 8 (a). The Rect8 and the Hex topologies offer better properties in this case, as this optimal arrangement of neurons is visible for wider range of the $R_{\max}$ parameter. For smaller maps, e.g. with 10x10 neurons, the resolution of 3 bits does not disrupt the learning process, while for larger maps this effect starts to be visible, as shown in Figure 8 (b). Nevertheless, even in this case a proper ordering of the map is achievable for selected values of $R_{\max}$. In the optimal case the quantization error equals 16.18e-3. The nonzero value of this parameter in this case results from the arrangement of data in the input data space. In the not optimal case shown, for example, in Figure 8 (b), 29 neurons of 256 are not properly placed, resulting in the Q_{err} enlarged by 25%.

A different situation can be observed in the case shown in Figure 7. In this case the best results have been achieved for the resolution of 6 bits, for the Rect4 topology. Nevertheless, even for 3 bits the Q_{err} exhibits comparable values for selected values of $R_{\max}$. For the Rect8 topology the best results have been achieved for the resolution of 10 bits, although for 3 bits Q_{err} is only 13% larger, which in some cases is acceptable. For the Hex topology the best solution has been achieved for the resolution of 4 bits, although for 3 bits the Q_{err} is only 8% larger.

The number of transistors in case of the resolution of 3-bits, for $R_{\max} = 8$ (coded using 3 bits) equals c. 550 per a single neuron, while for 6 bits equals c. 1200. The ability to reduce the signal resolution for an example map with 16 x 16 neurons leads to reduction of the number of transistors by about 170,000. If such network were realized in the CMOS 0.18μm technology the chip area was reduced by about 2mm^2, while the power dissipation was reduced by 30 %.

IV. Hardware Implementation of the Proposed TNF Block

The neighborhood mechanism as a whole is composed of two different components that should be clearly distinguished. The first one is the topological distance calculation block proposed by the authors in [10], briefly described in previous section. This block is composed of the EN_PROP and the R_PROP circuits described above. As it has been mentioned, at each following ring of neurons surrounding the winner the value of the R signal is decreased by one. Particular R signals can directly be used as the input signals to the second component – the neighborhood function (NF) block. Each neuron contains its own NF block, so that the calculation of the factor $\eta \cdot G()$ in (1) can be carried out in parallel in

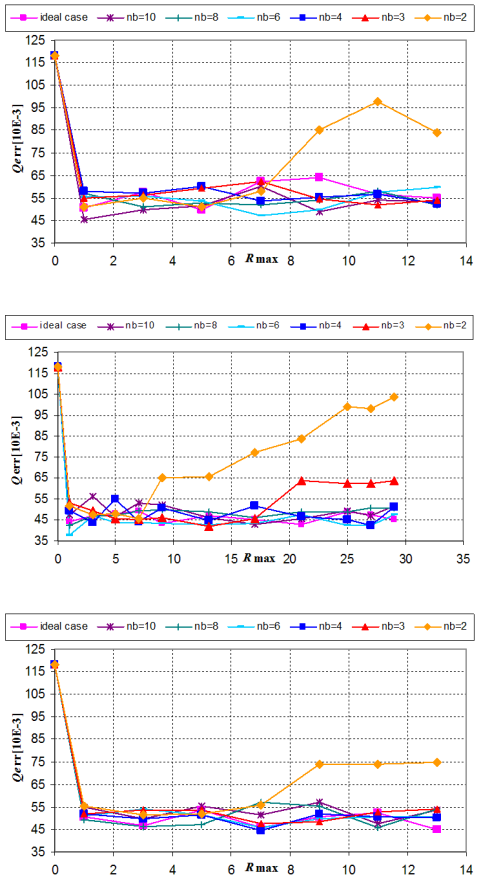

Fig. 7. The same results as in Figure 5 for: (a) rect8 16x16, (b) rect4 16x16 (b) hex 16x16 cases but for three inputs and data randomly spread in the input data space.

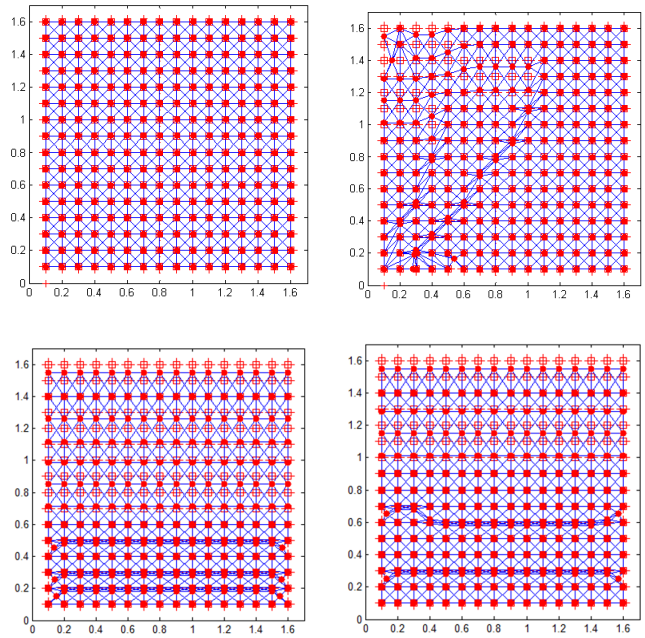

Fig. 8. Final placement of neurons in the map with 16x16 neurons, for the Rect8 topology, for data regularly distributed: (a) the optimal case, (b, c, d) not optimal cases with the quantization error increased by 25%, 42%, and 56% respectively.

all neurons in the map. A majority of the reported solutions concern implementation of the NF as a separate block. In most reported cases the Gaussian function is realized using the analog circuits. Only a few works present implementation of the overall neighborhood mechanism with the distance calculation block. One of such implementations is the analog neighborhood mechanism reported in [14].

In this paper authors focus on the implementation of the digital triangular NF block, which can be used with either a fully digital SOM [10], or with the analog network reported by the authors in [11].

An idea of the proposed TF block implemented as a digital circuit can be described as follows:

$$\eta(k) \cdot G(i,j,R,d) = R \cdot E/D + C \tag{6}$$

The C, D and E variables determine the shape of the triangular function. The $R \cdot E$ multiplication is performed using a typical shift-and-add circuit. If, for example, in such a circuit a binary number $E = 1001$ has to be multiplied by $R = 110$, a series of add operations is performed say, 0·1001+1·10010+1·100100. The summing operation of particular terms is performed by use of the multi-bit adders. The next operation is division by the D variable. In the proposed solution the allowed values of D are limited to the numbers that are the following powers of 2 i.e. 1, 2, 4, 8, . As a result, the division operation is realized very simply by shifting all bits in the $R \cdot E$ product to the right, using the circuit shown in Figure 10.

To illustrate how the proposed TNF circuit operates, several example cases are shown in Figure 9 for selected values of the C, D, E and R parameters. All neurons in the SOM receive the C, D and E variables with equal values, while the R parameter is provided only to the winning neuron, as an R_{PROG} signal. At the following neighborhood stages, i.e. the following rings of neurons, the value of R decreases toward zero, so at a ρ^{th} stage $R_\rho = R - d(i,j)$.

The number of bits in the C, D, E and R variables has direct influence on the circuit complexity, as described above. The number of the multi-bit adders in a single multiplier is linearly proportional to the resolution of E. On the other hand, the resolution of R determines the number of 1-bit full-adders in particular multi-bit adders. In the proposed TNF block a 1-bit adder composed of 26 transistors [15] has been used. Several other solutions with even less numbers of transistors were also considered, but in simulations the solution of [15] was the most efficient, considering the speed and the power dissipation.

Figures 5–7 show that quite good quality of the learning process is possible for $R_{\max} < 9$ i.e. for the resolution of this signal of only 3 bits. In the worst case, shown in Figure 7, the optimal resolution of the $\eta \cdot G()$ factor is 6 bits. As a result, the resolution of E should be 6 as well.

In the proposed circuit all neurons in the map operate in parallel. Simulations carried out in the CMOS 0.18μm

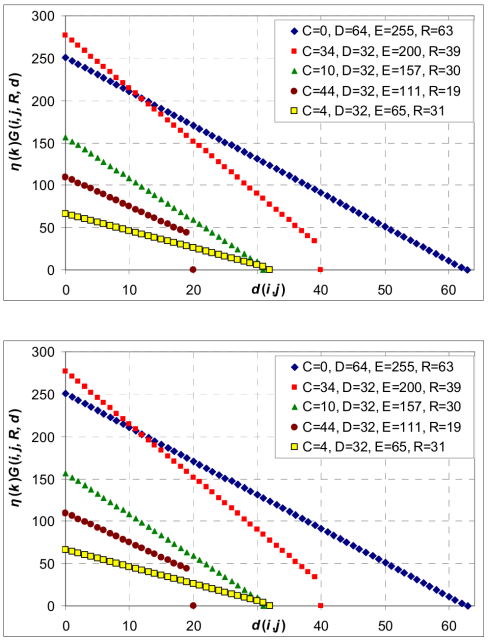

Fig. 9. The influence of the C, D, E and R parameters on the shape as well as the range of the triangular function. This Figure is a good illustration of the flexibility of the proposed solution.

technology show that input data rate can be as high as 10 – 20 MHz, depending on the number of the network inputs,x, and the topology. Each neuron for a single learning pattern X with 3 elements x (n=3) performs about twenty arithmetic operations. As a result, the map with 64 neurons performs 50e9 operations/s, dissipating the power of 50–100 mW. Larger maps with more than 1000 neurons will achieve data rate even as high as 1e12 operations/s in the technology which is not the newest one. A single operation in this case means e.g. an addition, multiplication, detection of the winning neuron, adaptation of a single weight, etc.

Since all neurons in the map are composed of equal blocks, therefore any reduction of the complexity of any block in a single neuron has a great effect on the complexity of the overall neural network. The most complex block in the proposed TNF block is the multi-bit multiplier, which is composed of several multi-bit adders. In this approach the shift-and-add multiplier has been realized as an asynchronous binary-tree structure. At the first layer of the tree particular terms corresponding to the bits of R are summed in the following fashion: 1 with 2, 3 with 4, and so on. All these operations are performed in parallel. Then in the second layer the results of the first layer are summed as follows: The output of the pair 1-2 is added to 3-4, 5-6 to 7-8, and so on. The number of adders at each following layer in the tree is always reduced by half in comparison with the previous layer. In the binary-tree approach for a resolution of the R signal of κ bits a delay that is introduced by the multiplier equals $T_{\mathrm{add}} \log_2 \kappa$, where T_{add} is delay introduced by a single multi-bit adder. This is an important advantage of this approach, as data rate degrades only moderately with increasing the number of bits.

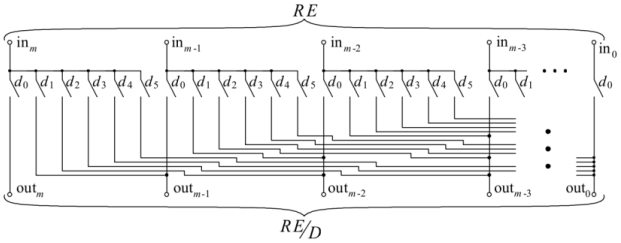

Fig. 10. The structure of the bits-shift block, which shifts the bits of the $R \cdot E$ product to the right, thus dividing the signal by D, which is always a power of 2.

Division operations usually require very complex circuits. In the proposed solution the $R \cdot E$ product is divided by selected values only and therefore this circuit is very simple and fast, as shown in Figure 10. The bits-shift operation is performed by use of a set of switches directly controlled by particular bits of the D variable: d_0, d_1, d_2, ..., d_p. Note that only one bit in this variable is allowed to be equal to 1. As a result, the division is based on the following scheme:

$d_0 = 1$ shifts the bits by 0 bits $\rightarrow$ division by 1
$d_1 = 1$ shifts the bits by 1 bits $\rightarrow$ division by 2
$d_2 = 1$ shifts the bits by 2 bits $\rightarrow$ division by 4
.....
$d_p = 1$ shifts the bits by p bits $\rightarrow$ division by 2^p

One additional circuit has to be used in the bits-shift block, which is not shown in Figure 10. Shifting the bits to the right by p positions makes the terminals that correspond to the p most significant bits floating. These terminals should be connected to ground to avoid the ambiguity. This task is realized by additional switches (one per each terminal), which are controlled by the signals also dependent on particular bits of the D variable. Instead of the switches, realized here as transmission gates, a series of the AND gates can be used, but at the expense of larger number of transistors and a little bit larger power dissipation.

The operation of the realized TNF block is illustrated in Figure 11. In this simulation a series of multiplications and divisions has been performed for R decreasing from 15 down to 0 and E decreasing from 31 down to 0, i.e. for 512 combinations. The products $R{\cdot}E$ are in the second step divided by 32 (i.e. are shifted by 5 bits to the right). Sampling period equals in this case 20 ns but the circuit works properly even for 6 ns. Figure 11 shows also the supply current. The values of the current spikes varies in-between 1 and 25 mA, while the width of these spikes is less than 1 ns. An average energy consumption does not exceed in this way a few pJ per a single operation.

Figure 12 illustrates transistor level simulations of the overall neighborhood mechanism with 8x8 neurons operating in the Rect4 mode. This mode allows for reaching the highest distances, as only the horizontal and the vertical directions are allowed, as shown in Figure 4. Looking from data rate point of view it is the worst case scenario. The top diagram illustrates the enable signals, EN, in the first column of the

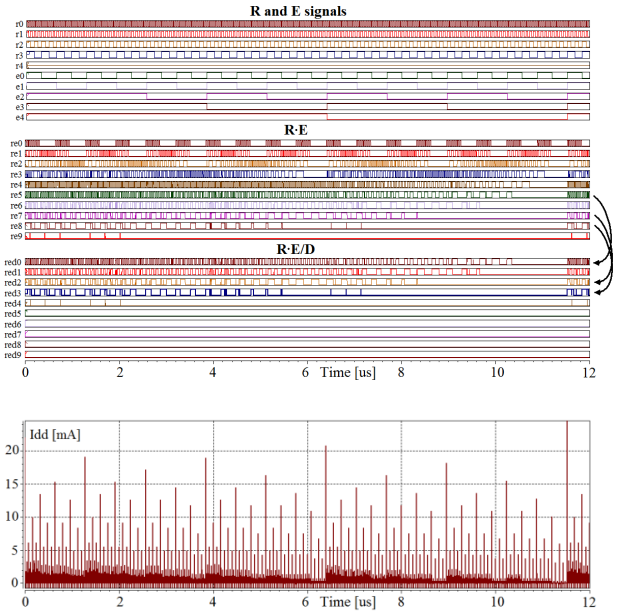

Fig. 11. Transistor level simulations of a separate TNF block. The top plot illustrates different values of the R and E parameters applied to the TNF inputs. The next two plots illustrate particular stages of the TNF block. The bottom plot illustrates the supply current. The results are presented for V_{DD}=1.8V.

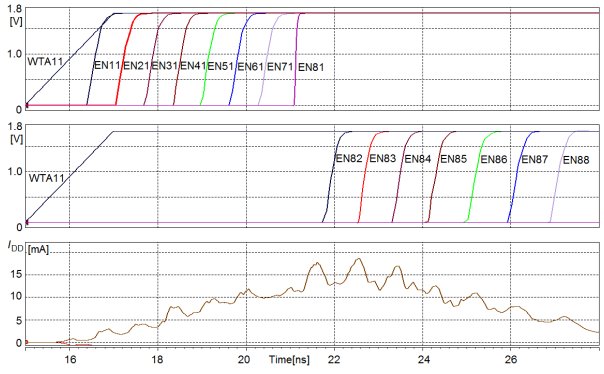

Fig. 12. Transistor level simulations of the overall programmable neighborhood mechanism for 8x8 neurons in the map together with the included TNF blocks. The first and the second plot illustrate the enable, EN, signals at particular stages of the neighborhood. The bottom plot illustrates a total supply current.

map. These signals trigger the adaptation process in particular neurons. Once the EN signal arrives at the bottom row of the map, the propagation immediately starts in this row, as shown in the second plot. A delay between the EN signal at the first (1, 1) and the last (8,8) neurons in this chain equals only 14 ns. Since a delay of a single TF block does not exceed 6 ns, the entire map is ready for the adaptation after no more than 20 ns. For the Rect8 topology this time is even shorter, as the diagonal directions are also permitted in this case. The remaining operations performed by the SOM take 20 – 30 ns depending on the number of the inputs. This means that the achievable data rate is larger than 10 – 20 MHz.

V. Conclusions

A new very fast and power efficient triangular neighborhood function (TNF) for hardware realized Kohonen SOMs has been proposed. The proposed circuit is a digital programmable solution. Using digital technique makes the proposed circuit robust against process, voltage and temperature (PVT) variation, so it is suitable for the applications working under different conditions [12].

The presented simulation results show that even low signal resolution at the output of the TNF block does not disturb the learning process of the SOM. On the other hand, low resolution allows for a significant reduction of the circuit complexity and the power dissipation. In the proposed architecture of the SOM all neurons operate in parallel. As a result, large neural networks can achieve the computational capacity as high as 1e12 operations/s in the CMOS 0.18μm technology.

References

[1] S. Osowski and T. H. Linh, "ECG beat recognition using fuzzy hybrid neural network," *IEEE Transactions on Biomedical Engineering*, vol. 48 (11), November 2001.

[2] A. Wismüller, O. Lange, R. Dersh *et al.*, "Cluster analysis of biomedical image time-series," *International Journal of Computer Vision*, vol. 46 (2), February 2002.

[3] T.Kohonen, *Self-Organizing Maps*. Berlin: Springer, 2001.

[4] P. Boniecki, "The kohonen neural network in classification problems solving in agricultural engineering," *Journal of Research and Applications in Agricultural Engineering*, vol. 50 (1), pp. 37–40, January 2005.

[5] I. Mokriš and R. Forgáč, "Decreasing the feature space dimension by kohonen self-organizing maps," in *Proc.* 2nd *Slovakian – Hungarian Joint Symposium on Applied Machine Intelligence*, Herľany, Slovakia, 2004.

[6] F. Li, C. H. Chang, and L. Siek, "A compact current mode neuron circuit with gaussian taper learning capability," *IEEE International Symposium on Circuits and Systems*, pp. 2129–2132, May 2009.

[7] M. T. Abuelma'ati and A. Shwehneh, "A reconfigurable gaussian/triangular basis function computation circuit," *IEEE International Conference on Computer Systems and Applications*, pp. 232–239, March 2006.

[8] D. S. Masmoudi, A. T. Dieng, and M. Masmoudi, "A subtreshold mode programmable implementation of the gaussian function for rbf neural networks applications," in *Proc. IEEE International Symposium on Intelligent Control*, October 2002, pp. 454–459.

[9] R. Długosz, M. Kolasa, and W. Pedrycz, "Programmable triangular neighborhood functions of kohonen self-organizing maps realized in CMOS technology," in *Proc. European Symposium on Artificial Neural Networks (ESANN09)*, Bruges, Belgium, April 2010, pp. 529–534.

[10] R. Długosz and M. Kolasa, "CMOS, programmable, asynchronous neighborhood mechanism for wtm kohonen neural network," in *Proc. International Conference Mixed Design of Integrated Circuits and Systems (MIXDES08)*, Poland, June 2008, pp. 197–201.

[11] R. Długosz, T. Talaśka, W. Pedrycz, and R. Wojtyna, "Realization of a conscience mechanism in CMOS implementation of winner takes all neural networks," *IEEE Transactions on Neural Networks*, vol. 21 (6), pp. 961–971, June 2010.

[12] P. Dubois, C. Botteron, V. Mitev, C. Menon, P. A. Farine, P. Dainesi, A. Ionescu, and H. Shea, "Ad hoc wireless sensor networks for exploration of solar-system bodies," *Acta Astronautica*, vol. 64 (5-6), pp. 626–643, 2009.

[13] R. Długosz and M. Kolasa, "Optimization of the neighborhood mechanism for hardware implemented kohonen neural networks," in *Proc. European Symposium on Artificial Neural Networks (ESANN09)*, Bruges, Belgium, April 2009, pp. 565–570.

[14] V. Peiris, "Mixed analog-digital VLSI implementation of a kohonen neural network," Ph.D. dissertation, Ph.D thesis, Ecole Polytechnique Fédérale de Lausanne (EPFL), 1994.

[15] C. H. Chang, J. Gu, and M. Zhang, "A review of 0.18m full adder performances for tree structured arithmetic circuits," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 13 (6), pp. 686–695, June 2005.