

# Analog-Counter-Based Conscience Mechanism in Kohonen's Neural Network Implemented in CMOS 0.18 $\mu\text{m}$ Technology

Tomasz Talaśka<sup>1</sup>, Ryszard Wojtyna<sup>1</sup>, Rafał Długosz<sup>2,3,4,\*</sup>, Krzysztof Iniewski<sup>2</sup>, Witold Pedrycz<sup>2</sup>

<sup>1</sup> University of Technology and Agriculture, Faculty of Telecomm. and Electrical Engineering, Kaliskiego 7, 85-791, Bydgoszcz, Poland, talaska@atr.bydgoszcz.pl, woj@atr.bydgoszcz.pl

<sup>2</sup> University of Alberta, Department of Electrical and Computer Engineering, W2-079 ECERF Building, Edmonton, T6G 2V4, Canada, rdlugosz@ualberta.ca, iniewski@ece.ualberta.ca, pedrycz@ece.ualberta.ca

<sup>3</sup> University of Neuchâtel, Institute of Microtechnology, Rue A.-L. Breguet 2, CH-2000, Neuchâtel, Switzerland

<sup>4</sup> Poznań University of Technology, Department of Computing Science and Management, Piotrowo 3A, 60-965 Poznań, Poland  
\* fellow of the Foundation for Polish Science

**Abstract**—In this study, we present a hardware implementation of the conscience mechanism in Kohonen self-organizing maps. The proposed realization of the conscience mechanism is important to the functioning of the neural network as it eliminates so-called dead (inactive) neurons. As a result the network learning, the level quantization error can be reduced. The conscience mechanism and the Winner Take All (WTA) block have been implemented in 0.18  $\mu\text{m}$  CMOS process. The implementation of the conscience mechanism itself occupies 1200  $\mu\text{m}^2$  and its maximum power consumption is 9.5  $\mu\text{W}$ . The WTA block together with the conscience mechanism occupies 0.024  $\text{mm}^2$  and dissipates 55  $\mu\text{W}$ .

## 1. INTRODUCTION

Continuous progress in microelectronics and Integrated Circuit (IC) manufacturing allows for implementation of large neural networks due to ever increasing packing density. Large neural networks are required to model biological systems and can be considered in numerous areas applications [1, 3]. One of the key challenges of the effective neural network implementation concerns a low power dissipation of the underlying hardware.

In this work we are present an IC implementation of the Kohonen's neural network (self-organizing map) and discuss several enhancements to its learning mechanism. As a result the proposed technique is faster when compared to supervised training and does not require high precision (as being required in the case of gradient-based learning).

It is instructive to start with a couple of introductory comments about the paradigm of self-organized learning as being supported by the Kohonen's maps. Typically the map is organized in a form of a grid of  $p \times p$  processing units (neurons) endowed with a vector of connections. The number of inputs (dimension of input space) is equal to "n". The underlying principle is that of a topological preservation of a data structure in a highly dimensional space when being mapped onto a low, two- or three-dimensional space. As usual, let us assume that at the beginning of learning all weights (connections) are initialized to some random values. This competitive learning process using winner-take-all (WTA) block relies on presenting the neural network with learning vectors in order to make the weight vectors resemble to presented examples. As a result of this process only one neuron, the closest to the learning vector (where the distance is

expressed in same metric space) wins the competition [5]. Next this winning neuron adjust its connections in the following way [5]:

$$w(k+1) = w(k) + \eta \cdot (x - w(k)) \quad (1a)$$

Other neurons remain unchanged:

$$w(k+1) = w(k) \quad (1b)$$

The purpose of these changes is to make the weight vector  $\mathbf{w}$  resemble the learning vector  $\mathbf{x}$ . However, all other neurons do not change their weights until they win at some point in time.

In the above formula  $w(k)$  are given values of neuron weights,  $w(k+1)$  are new values of neurons weights, calculated in the Kohonen's algorithm and  $\eta$  is so called learning coefficient. Initial value of this coefficient must be in the range (0, 1) and its value vanishes to 0 with progress of the learning process [5].

When many similar learning vectors are applied to the input of the neural network, one neuron wins all the time and its weights values will converge to the average values of the learning vectors presented previously to the network. In the case when learning vectors are different enough, in each learning cycle other neuron wins the competition. In result we obtain self-organization process of the neural network. These types of networks are frequently called SOMs (Self-Organizing Maps) [5].

An important problem in Kohonen's networks concerns with so called dead neurons. Those are the neurons that never win a competition and as a result their weights never change. Dead neurons decrease efficiency of neural network's training process, because smaller number of neurons take part in the competition [4].

In this work, we are presenting CMOS 0.18 $\mu\text{m}$  implementation of the conscience mechanism, which due to improvement of the learning algorithm, allows for reduction of the so-called dead neurons. The presented algorithm is one of the many variants that are currently under consideration by authors. Recently we have shown the implementation of the conscience mechanism using digital counters [7]. The counters implemented using digital techniques are stable, but have limitation: they exhibit fixed counting limits. The solution presented here proposes a novel concept of analog counters, that enable tuning of the counting range in a large range and offer

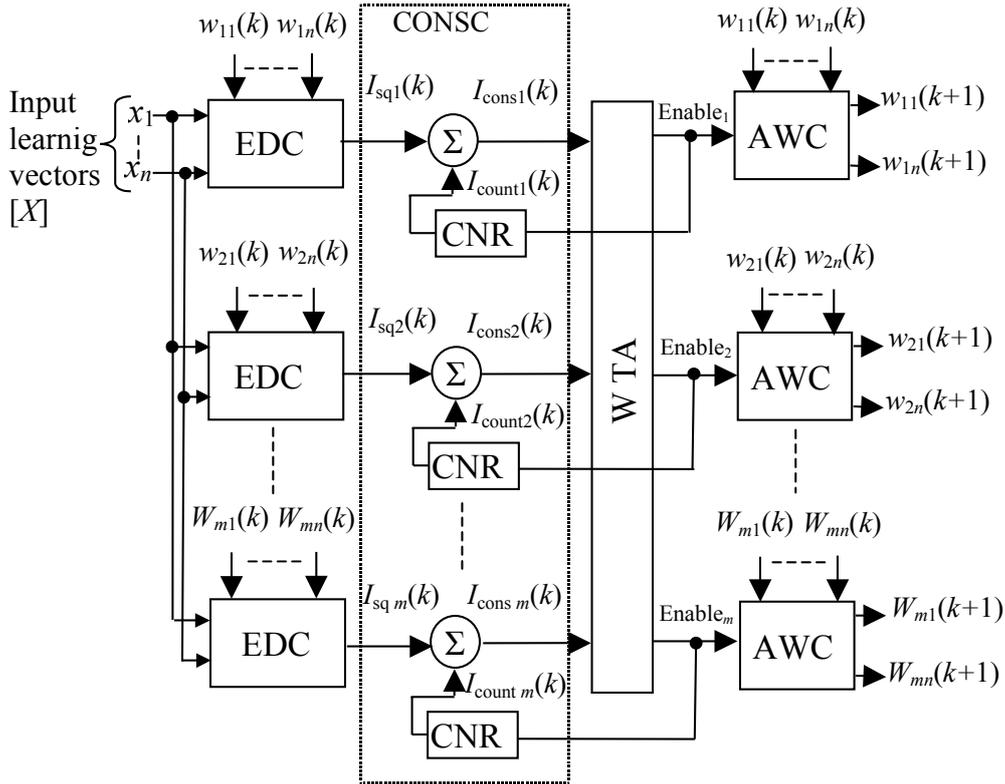


Fig. 1 Diagram of the proposed learning block of Kohonen's neural network for  $m$  neurons and  $n$  inputs: EDC - Euclidian's distance calculation blocks, CONSC - conscience mechanism blocks, CNR - analog counter, AWC - adaptation (Kohonen's algorithm) weight correction block

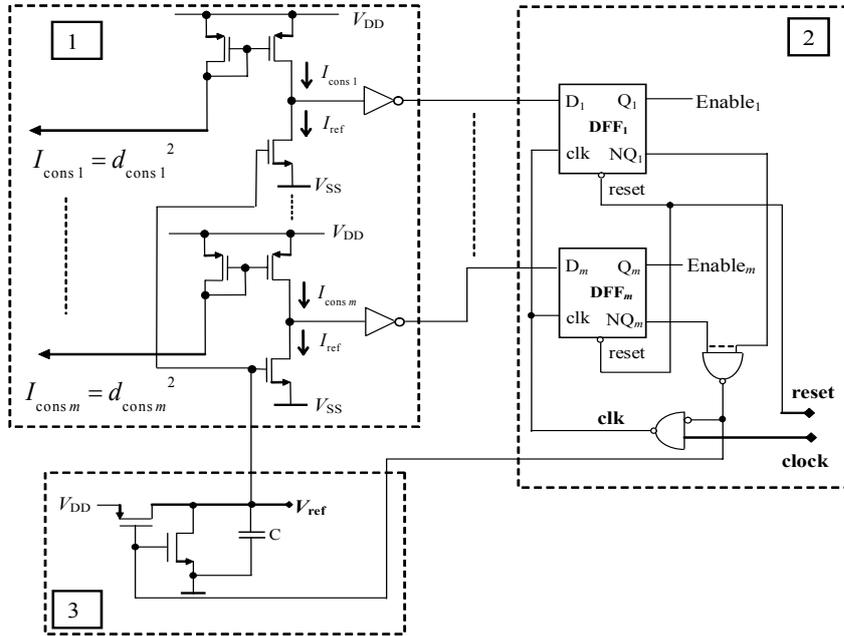


Fig. 2 Winner Takes Block (1) Current comparators (2) Digital control block (3) Source of reference signal [7]

much lower power dissipation levels than their digital counterparts.

The paper is organized as follows. In Section 2 we discuss an essence of the learning process and the corresponding ASIC architecture that implements WTA block using Kohonen's rule.

In Section 3, given are details of the implementation of the conscience mechanism. A collaboration between the conscience mechanism and the WTA block is discussed.

Section 4 presents a CMOS layout implementation and post-layout simulation results. Finally, conclusions and future work are converted in section 5.

## 2. ASIC IMPLEMENTATION OF KOHONEN'S NETWORK

Model responsible for neural network training process is shown in Figure 1. EDC sub-circuits represent blocks responsible for calculating a degree of similarity between training and weight vectors for each neuron. In order to



In above equations index  $j$  determines the input node and the weight of a given neuron, index  $i$  is the number of the following neuron ( $x_j$  is the following input while  $w_j$  is the following weight),  $n$  describes number of weights, which is equal to number of inputs of the neural network.

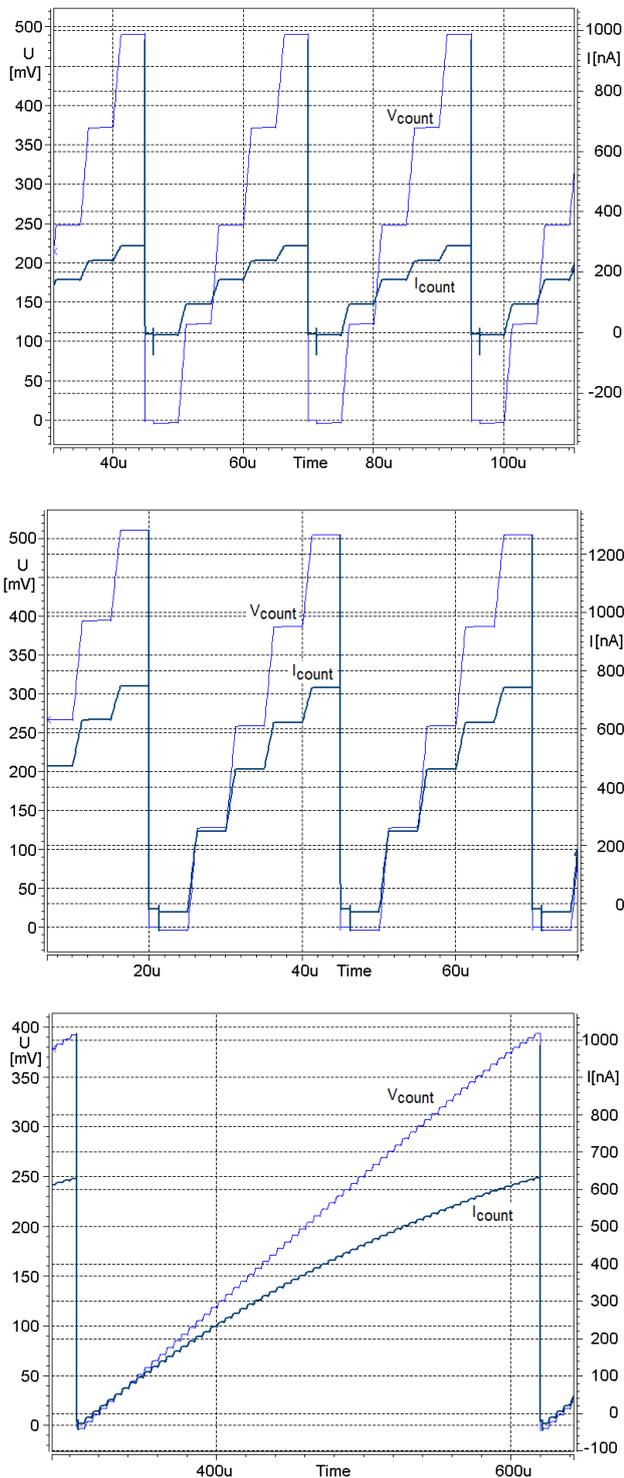


Fig. 7 Post-layout simulations of the conscience mechanism (analog counter and U-I converter) for various values of the control voltages  $V_{\text{contr}}$  (Fig. 4) and  $V_{\text{con}}$  (Fig. 5)

Parameter  $z$  is used to control of the influence of the conscience mechanism the so-called multiplier that can be used for additional operation. This parameter, in hardware

implementation represented by DC voltage, enables for example, the conscience mechanism can be turned-off.

The counter of the neuron wins can be implemented as analog or digital block. Because digital counters become power and area inefficient for large neural networks [7], we have decided to implement the counter in analog domain, which offers a far higher flexibility solution.

A basic idea of the proposed solution for the analog counter is shown in Figure 4. The counter operation is based in the following principle. Successive negated input pulses arriving from the WTA block open the current source MP1 that charges capacitor C2. When the voltage stored on C2 becomes larger than the threshold voltage the voltage at the NOT1 output changes from  $V_{\text{DD}}$  to  $V_{\text{SS}}$ . As a result NOT2 output changes to  $V_{\text{DD}}$  when the switch K1 is closed. When the CLK becomes  $V_{\text{SS}}$  and CLK negated becomes  $V_{\text{DD}}$  the reset signal goes high. This in turn causes turning on the process of discharging C2 through MN2, which implies zeroing of the counter. The switch K1 allows for synchronization of the RESET signal with CLK in order for the discharging impulse to last long enough. The voltage stored on C2 (proportional to number of wins) is converted to current that is added to the output current of the squaring circuits.

In the analog counter circuit shown in Figure 4 the counter input signal CLK is connected to the output of the WTA circuit (signals Enable in Figs 1 and 2). Using the control voltage  $V_{\text{contr}}$  the resolution of the counter can be controlled, as illustrated later in Figure 6. The output of the analog counter  $V_{\text{count}}$  has to be converted to current  $I_{\text{count}}$ . This can be realized in the U-I converter proposed by authors [6], which is a differential transconductor shown in Figure 5.

The following equation represents the relationship between the output current and input voltages:

$$I_{\text{count}} \cong K(V_{\text{G2}} - V_{\text{G1}})(V_{\text{CON}} - V_{\text{SS}} - 2V_{\text{th}}) \quad (5)$$

In the above equation  $K$  is a constant dependent on transistor sizing,  $V_{\text{th}}$  is a transistor threshold voltage.  $V_{\text{CON}}$  voltage is used to control multiplier gain,  $V_{\text{G1}}$  and  $V_{\text{G2}}$  are voltages at the gates of M1 and M2 dependent on the voltages  $V_{\text{in1}}$  and  $V_{\text{in2}}$  (output  $V_{\text{count}}$  from the counter).

Figures 6 and 7 represent post-layout simulation results for the conscience circuit for few values of  $V_{\text{contr}}$  and  $V_{\text{con}}$ . It is apparent that the proposed circuit is very flexible. Figure 6 shows the analog counter output for various counting ranges (e.g. for 3 and for about 400). Figure 7 represents the conscience circuit output current  $I_{\text{count}}$  increasing Euclidian's distance combined with increased number of wins represented as  $V_{\text{count}}$  voltage in the C2 capacitor.

#### 4 KOHONEN'S TRAINING NETWORK IMPLEMENTATION

Layout of the entire Kohonen's block is presented schematically in Figure 1, implemented in CMOS 0.18  $\mu\text{m}$  technology is shown Figure 8.

Experimental results shown in the subsequent part of this section relate to collaboration of the conscience mechanism block with blocks responsible for calculating Euclidian's distance for the neurons and WTA block.

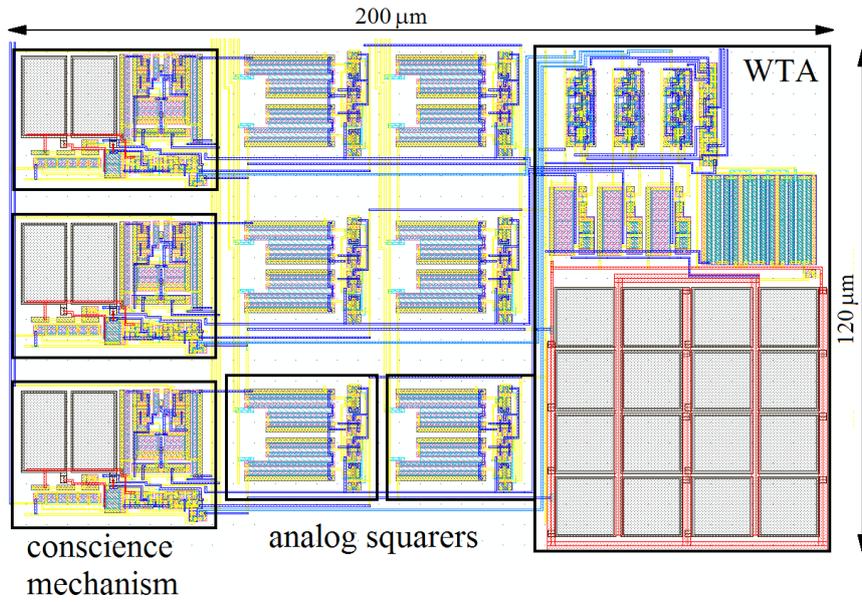


Fig. 8 Layout of the WTA block implemented in 0.18  $\mu\text{m}$  process. Block 1 calculates Euclidian's distance, block 2 implements conscience mechanism, and block 3 provides detection mechanism of the winning neuron

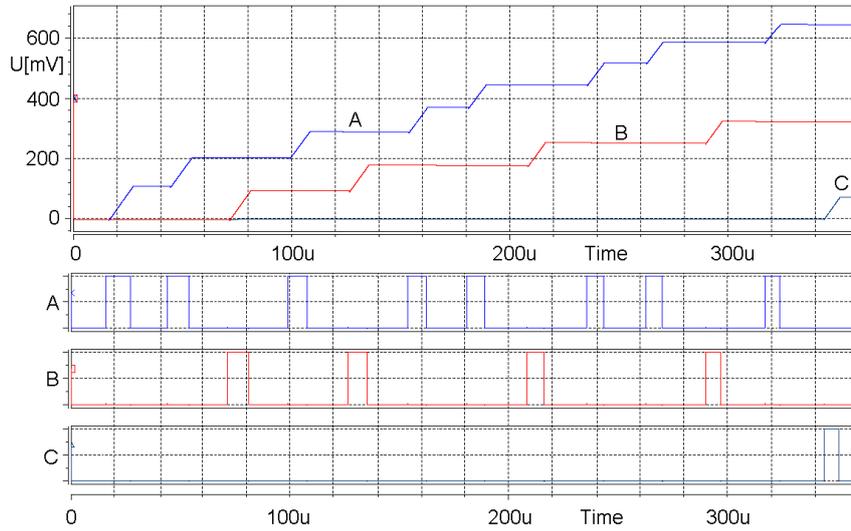


Fig. 9 WTA signals: (up) signals in analog counters in conscience mechanism for neurons A, B, C (bottom) WTA block outputs - signals "Enable" in Fig 1.

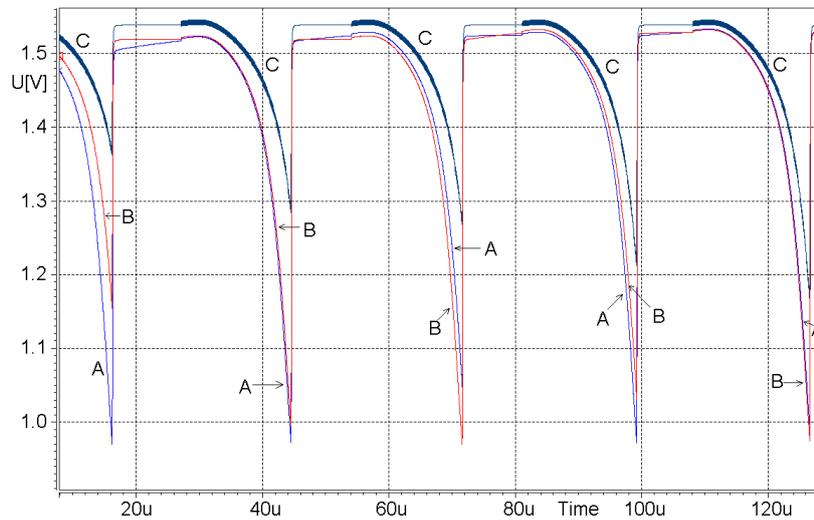


Fig. 10 Signals in the winning neuron detection circuit (Fig 2) that illustrates competition between neurons A, B, C - NOT gates (block 1) input voltages

To verify Kohonen's algorithm and the proposed implementation numerous test case simulations have been undertaken. In one experiment input and weight vectors have been fixed and assigned as follows:  $x_1 = 1 \text{ V}$ ,  $x_2 = 1 \text{ V}$ ,  $w_{11} = 1.2 \text{ V}$ ,  $w_{12} = 0.9 \text{ V}$ ,  $w_{21} = 1.3 \text{ V}$ ,  $w_{22} = 0.8 \text{ V}$ ,  $w_{31} = 1.3 \text{ V}$ ,  $w_{32} = 0.3 \text{ V}$ .

The results of these simulations are shown in Figures 9 and 10. After applying the training vector  $X$  the competition is won by neuron A (Figure 9, bottom-A).  $k$  presentations of the same training vector  $X$  decrease winning probability of the neuron A as a result of actions of the implemented conscience mechanism. Consequently, at some point the neuron B can win (Figure 9, bottom-B). After several presentations of the training vector  $X$  winning probability of neurons A and B is decreased to the level, which enables neuron C also to win (Figure 9, bottom-C) although its real Euclidean's distance between its weight vector and training vector is significantly larger than neurons A and B.

Figure 10 illustrates operation of the WTA block presented in Figure 2 and, in fact, competition between neurons due to operation of the conscience mechanism. WTA's input currents are in fact conscience mechanism's output currents  $I_{\text{cons}}$ . Neuron whose weights  $W$  are the closest to the input signals  $X$  generates the smallest current  $I_{\text{sq}}$ . The reference current (dependent on reference voltage  $V_{\text{ref}}$ ) from block number 3 (Fig 2) is increasing and compared to all  $I_{\text{cons}}$  simultaneously.

Output of the comparator corresponding to the winning neuron will fall down as first, what is detected by digital control circuit and interpreted as a win of this neuron.

It is important to point out that if the conscience mechanism would not be implemented, the neuron A would be always winning when presented with the give example training vector  $X$ . As a result neurons B and C would be dead in this case. In the Figure 11 we see that although neurons A and B win alternately because their weights are the closets to the training vector  $X$ , the time distance between neuron C and neurons A, B decreases after each presentation of the training vector  $X$  as a result of actions of the implemented conscience mechanism. In In Figure 10 we see that after many presentations of the training vector  $X$ , time distance between switching of the outputs of the comparators associated with particular neurons diminishes.

This is the result of the constant training vector  $X$  in this experiment. Different values stored in a given time instant in particular counters compensate the real Euclidean's distances between vectors. These real distances are constant due to the constant input values  $X$  and  $W$ , but in the real network training vector  $X$  will not be constant. Constant value of  $X$  is assumed here only to better illustrate operation of the proposed WTA block with conscience mechanism.

## 5. CONCLUSIONS

The implemented conscience mechanism in the Winner Take All (WTA) block represents a next step towards a complete IC implementation of the Kohonen's neural network. The proposed circuit of the conscience mechanism, implemented in  $0.18 \mu\text{m}$  technology using analog counters, occupies  $1200 \mu\text{m}^2$  ( $40 \mu\text{m} \times 30 \mu\text{m}$ ) and dissipates  $9.5 \mu\text{W}$ .

The analog counter has certain leaking properties wording due its implementation as analog voltage stored on a capacitor. Charge leaking process in the analog counter causes conscience mechanism to become weaker as the timing distance between subsequent winners increases. As a result, this parasitic effect may not necessarily be viewed as a disadvantage from the training process point of view. The penalized neurons slowly recover full rights to competition.

The entire Kohonen's training block occupies  $0.024 \text{ mm}^2$  ( $120 \mu\text{m}$  by  $200 \mu\text{m}$ ) and in the case of 3 output neurons and two-dimensional input vector  $X$  dissipates  $55 \mu\text{W}$ . The chip is currently in fabrication at Taiwan Semiconductor Corporation (TSMC). The power dissipation is small enough that even in very large neural networks (in which one may require several hundreds WTA blocks), the total power consumption would be manageable.

The final step required for full implementation of the training process shown in Figure 1 is adaptation weight correction block (AWC). Such a mechanism is currently under implementation and will be reported elsewhere.

## REFERENCES

- [1] Cauwenberghs G., Bayoumi M.: *Learning on silicon, adaptive VLSI neural systems*, Kluwer Academic Publishers, 1999.
- [2] Deboeck G., Kohonen T.: *Visual explorations in finance with self-organizing maps*, Springer-Verlag, 1998.
- [3] Maass W., Bishop C.: *Pulsed neural networks*, Massachusetts Institute of Technology, The MIT Press, 1999.
- [4] Zurada J.: *Introduction to artificial neural systems*, West Publishing Company, USA, 1992.
- [5] Kohonen T.: *Self-organizing maps*, Springer Verlag, Berlin 2001
- [6] Wojtyna R., Talaska T.: Improved Power-Saving Synapse for Hardware Implemented ANN's, Int. Conf. on Signals and Electronic Systems ICSES'04, Poznań, Poland 2004, pp. 27-30.
- [7] Talaška T., Wojtyna R., Długosz R., Iniewski K.: Implementation of the conscience mechanism for Kohonen's neural network in CMOS  $0.18 \mu\text{m}$  technology, *International Conference Mixed Design of Integrated Circuits and Systems*, Gdynia, Poland 2006, pp. 319-315.
- [8] Wawryn K., Strzeszewski B.: Low power VLSI neuron cells for artificial neural networks, in Proc. ISCAS, vol.3. Atlanta 1996, pp. 372-375.

---

This work was supported partly by the KBN grant No. 1580/T11/2005/29.