Realization of the Conscience Mechanism in CMOS Implementation of Winner-Takes-All Self-Organizing Neural Networks

Rafał Długosz, Tomasz Talaśka, Witold Pedrycz, *Fellow, IEEE*, and Ryszard Wojtyna

***Abstract*—This paper presents a complementary metal–oxide–semiconductor (CMOS) implementation of a conscience mechanism used to improve the effectiveness of learning in the winner-takes-all (WTA) artificial neural networks (ANNs) realized at the transistor level. This mechanism makes it possible to eliminate the effect of the so-called "dead neurons," which do not take part in the learning phase competition. These neurons usually have a detrimental effect on the network performance, increasing the quantization error. The proposed mechanism comes as part of the analog implementation of the WTA neural networks (NNs) designed for applications to ultralow power portable diagnostic devices for on-line analysis of ECG biomedical signals. The study presents Matlab simulations of the network's model, discusses postlayout circuit level simulations and includes results of measurement completed for the physical realization of the circuit.**

***Index Terms*—Analog circuits, complementary metal–oxide–semiconductor (CMOS) implementation, conscience mechanism, low energy consumption, parallel data processing, self-organizing map (SOM), winner-takes-all (WTA) neural networks.**

I. Introduction

HARDWARE implementations of artificial neural networks in an analog or analog-digital application specific integrated circuit (ASIC) form may sometimes lead to much more power-economic and faster computational operations in comparison to what becomes achievable in software implementations [13], [19]. The reduction of power consumption and increase in operation speed by several orders of magnitude can be reached. Thus, achieved significant computing improvements are mostly due to parallel data processing being realized in such networks.

The low power consumption results from the use in some processing tasks analog blocks (modules) that offer a much lower circuit complexity than their digital counterparts.

On the other hand, designing analog or analog-digital networks implemented as ASICs is extremely complex, as it must be usually done in a *full-custom* style. This is one of the main reasons why hardware implementations are still quite rare in comparison with the existing software implementations of neural networks. Another reason is a common opinion that commercial digital processors are fast enough to realize neural networks thus reducing the usefulness of any specialized hardware implementation. Consequently, the research on this topic has been slowed down, resulting in a situation, in which the main hardware implementations of neural networks are those reported many years ago [1]–[9].

While there is some rationale behind the arguments presented above, we are currently witnessing new developments calling for a careful reexamination of the views concerning the usefulness of hardware realizations of neural networks. Indisputably, the recent development of electronic circuits in the very first place targets those applications in which ultralow energy consumption is highly relevant. In wireless sensor networks, for example, substantial effort has been made to develop devices that are self-sufficient with regard to energy supply. This is a class of circuits, in which analog artificial neural networks (ANNs) that are much more power efficient than their software counterparts may find a wide range of applications.

We have been working on the hardware implementation of ANNs in an ASIC form for some time now; cf., [10]–[13], [27], and [28]. Our particular interest is in self-organizing maps (SOMs) being trained using the learning vector quantization (LVQ) rule [14], [29]. We have recently designed an experimental power-economic analog winner-takes-all (WTA) network that can be trained on silicon. The experimental measurements of the fabricated prototype along with comparative simulation studies carried out in software (C++) demonstrate that analog networks of this type are highly efficient. For instance, 100 neurons operating in parallel can be even several dozen times faster than a similar network being implemented using a PC while dissipating only 1% of power required by a typical PC [13]. The intended application of such networks could be in an online analysis of electrocardiography (ECG)

Manuscript received November 18, 2008; revised August 26, 2009, November 16, 2009, and January 15, 2010; accepted March 15, 2010. Date of publication April 22, 2010; date of current version June 03, 2010. This work was supported by M. Curie Outgoing International Fellowship.

R. Długosz is with the Institute of Microtechnology, Swiss Federal Institute of Technology, Lausanne, CH-2000 Neuchâtel, Switzerland (e-mail: rafal.dlugosz@epfl.ch).

T. Talaśka is with the Faculty of Telecommunication and Electrical Engineering, University of Technology and Life Sciences, 85-796 Bydgoszcz, Poland (e-mail: talaska@utp.edu.pl).

W. Pedrycz is with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 2V4, Canada and also with the Systems Research Institute, Polish Academy of Science, Warsaw 01-447, Poland (e-mail: pedrycz@ece.ualberta.ca).

R. Wojtyna is with the Faculty of Telecommunication and Electrical Engineering, University of Technology and Life Sciences, 85-796 Bydgoszcz, Poland and also with the Academy of Information Technology, Institute of Computer Science, Łódź 93-008, Poland (e-mail: woj@utp.edu.pl).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNN.2010.2046497

and electromyography (EMG) biomedical signals in power-efficient portable diagnostic devices forming a part of wireless body sensor networks [20], [21].

One of the problems in the WTA networks are so-called "dead neurons" i.e., the neurons the weights of which do not change during the learning process. The common reason for that could be improperly selected initial values of the weights (connections) [12].

As regards classification problems, dead neurons reduce the number of classes that could be discriminated. Moreover, they increase a mapping (quantization) error of the network. For this reason, reducing or eliminating these neurons becomes an important design objective. One of the efficient methods of eliminating the dead neurons is the so-called conscience mechanism [15]–[18]. Its role is to give all neurons a chance to win the competition and participate in the weight adaptation process.

This study focuses on the conscience mechanism implemented on silicon together with the overall neural network by making use of the ASIC style of design. The proposed realization is based on the utilization of analog counters studied by Talaśka *et al.* [10]. The paper presents a design methodology of the conscience mechanism completed at three different levels. The system level, realized by using the software model of the network, allows for a determination of the optimal parameters of the conscience mechanism for a given number of neurons. At the second level, the postlayout simulations allow for the optimization of the circuit. Here the internal signals, like outputs of the analog counters, are available for monitoring. In the presented implementation of the WTA NN, these signals cannot be directly observed from outside the chip. The measurements completed at the third level are necessary to verify all assumptions made at the previous levels.

Some early hardware developments of Kohonen self-organized feature maps (SOFMs) are reported in [24]–[26]. They are based on different signal processing techniques, i.e., digital, analog, and mixed analog-digital ones.

The fully digital Kohonen network implemented in complementary metal–oxide–semiconductor (CMOS) 0.5-μm process has been presented in [24]. The main problem in this realization was a very large chip area that made the implementation of large networks fairly impractical. Each processing element (PE) that represents a single neuron of the map contains about 10 000 transistors and occupies an area of 4 mm^2. Even if, owing to newer technologies, this circuit is scaled down, the area of a single neuron at the level of mm^2 leads to substantial difficulty in the implementation of more than several dozen of neurons on a single chip. A single prototype chip contains four PEs, while the larger map (KARN processor) is realized by the parallel arrangement of many KARN chips.

Another example of the hardware realization of such a network has been reported in [25]. In this solution, some modules of the overall architecture like the Euclidean distance calculation block (EDC) and the WTA block are implemented as analog components, while the adaptation process is realized by the use of digital technology. The analog memory is implemented using digital counters that can count in both directions depending on whether an input data x is greater than a neuron's weight w. The adaptation mechanism used in this solution is different from the "typical" Kohonen's algorithm. The learning rate η is kept fixed and its value results directly from the assumed resolution of the counter (5 bits in this particular case). In addition, the adaptation process does not depend on exact values of x and w but only on the value of the sign function $\mathrm{sign}(x, w)$, always updating the counter value by ± 1.

An interesting network implemented by using current-mode analog circuits has been reported in [26]. In this approach, the distance between currents representing input data and neuron weights is calculated using Manhattan (Hamming) metric, as this metric is easier to implement in hardware than the Euclidean one. All main operations, including the adaptation, are carried out in an analog way. In comparison to the solutions described above, this approach is much more chip area efficient, even though this network has been implemented in the older CMOS 2-μm technology. This demonstrates that the analog technique is much more suitable in the implementation of neural networks than its digital counterpart. One of the disadvantages in this analog solution is a complex refreshing system of the analog memory. The memory cells are refreshed sequentially using a single current-mode successive approximation analog-to-digital converter (SAR ADC). As a result, the period of the overall refreshment cycle depends on the number of memory cells. An additional disadvantage is that the refreshing mechanism always rounds weights stored in the analog memory to the nearest discrete values.

The number of levels of the reference signal depends on the resolution of the ADC, while the accuracy of the learning process additionally depends on the linearity of the input–output characteristics of this ADC.

The WTA network reported in this paper uses both the analog current-mode technique and parallel data processing. As a result, the network occupies an area that is two orders of magnitude smaller than the one being used by its digital counterpart. The adaptation process is completed at the analog side. In comparison to analog and mixed implementations described in this section, the proposed network dissipates only a small fraction of the power dissipated by these networks while achieving a much higher data rate.

It is important to stress that in our network, trained in accordance with the LVQ rule, we do not use the neighborhood mechanism, as it is presented in the solutions described in [24]–[26], but considering the criterion of the hardware complexity alone, the comparison of this nature still remains justified, since the conscience mechanism requires an additional hardware. The learning rule used in our WTA network may be considered as a special case of the algorithms used in the winner-takes-most (WTM) SOM [31], for the neighborhood radius equal to zero. The use of the conscience mechanism allowed us to avoid implementing the neighborhood mechanism on this prototype chip, although we are currently working on such mechanism [32], [33] that is to be included in the next prototype chip.

This paper is organized as follows. An overview of existing conscience algorithms is presented in Section II. Section III treats on the proposed conscience mechanism realized by means of analog circuits. A brief overview of other components of the prototype network is also presented in this section. A verification of the conscience mechanism including software simula-

tions, postlayout circuit level simulations, and experimental results of the designed neural network in CMOS technology are presented in Section IV. Conclusions are covered in Section V.

II. Conscience Mechanism in Neural Networks—A Review

As already mentioned, one of the important problems of the WTA network are dead neurons. This problem can be significantly reduced or even completely eliminated by applying a conscience mechanism. In the literature, one can find several realizations of this mechanism [16]–[18]. One of the existing implementations is the algorithm described by Ahalt *et al.* [16] that takes advantage of frequency sensitive competitive learning (FSCL). In this approach, each neuron includes an individual counter that counts the number of times a given neuron wins during the overall training process. As a result, the value of the Euclidean distance used here as the measure of dissimilarity between the input data and the weights of the winning neuron is increased. This is done by multiplying an original distance by a value that is proportional to the number of times this neuron has been declared as a winner. This modification decreases the winning likelihood of this neuron in the next iterations, thus allowing other neurons in the network to compete and win. The modified distance can be described as follows:

$$\|W_i - X\|^{2^*} = \|W_i - X\|^2 \cdot (1 + K \cdot L_{\text{count } i}) \quad (1)$$

where W_i is a weight vector of the winning neuron, X is a learning vector, the term $\|W_i - X\|^2$ is a square of the Euclidean distance that serves as a measure of resemblance between the W_i and the X, $\|W_i - X\|^{2*}$ is an artificially increased distance between X and W, $L_{\text{count } i}$ is the number of wins of a winning ith neuron, while K is the gain factor that controls strength of the conscience mechanism. The weights' adaptation process of the winning neuron in the kth presentation of an input pattern is realized according to

$$W_i(k+1) = W_i(k) + \eta \cdot [X(k) - W_i(k)] \cdot z_i. \quad (2)$$

Here, z_i relates to the ith winning neuron and is expressed in the form

$$z_i = \begin{cases} 1, \text{ if } \|W_i - X\|^{2^*} \le \|W_j - X\|^{2^*}, & \forall_{j \ne i} \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

Another solution for the realization of the conscience mechanism has been proposed by De Sieno [18]. In the proposed distance modification, there is a variable p associated with each neuron in the network, which is a measure of the number of the wins being calculated according to

$$p_i^{\text{new}} = p_i^{\text{old}} + B \cdot \left(y_i - p_i^{\text{old}}\right) \quad (4)$$

where B is a constant selected experimentally to be in the range $0 < B \ll 1$, y_i is the output of the ith neuron, which is equal to 1 for a winning neuron and 0 otherwise. The weights of the winning neuron are in this case expressed by

$$W_i(k+1) = W_i(k) + \eta \cdot [X(k) - W_i(k)] \cdot z_i \quad (5)$$

where η is a learning rate, while in this case z_i is calculated according to

$$z_i = \begin{cases} 1, \text{ if } \|W_i - X\|^2 - b_i \le \|W_j - X\|^2 - b_j, & \forall_{j \ne i} \\ 0, & \text{otherwise.} \end{cases} \quad (6)$$

The term b_i is expressed in the form

$$b_i = C \cdot \left(\frac{1}{N} - p_i\right). \quad (7)$$

C is a constant value. This value, reported in [18], is equal to 10, while N is the number of neurons in the network (the number of outputs). For $C = 0$, this algorithm realizes the classic learning rule used in the WTA network.

It is worth noting that at the beginning of the learning process all variables p are set to zero, which means that all variables b are equal to some positive values that depend upon the number of neurons in the network. The values of p increase nonlinearly with the number of wins of the particular neurons. As shown in (4), they never exceed a value of 1, approaching this value asymptotically. Since the Euclidean distance (or its squared value) can vary in-between 0 and a certain positive value that depends on the input data range, therefore the terms on both the left-hand side and the right-hand side of inequality (6) can be either positive or negative.

The said nonlinearity of the variable p makes De Sieno's algorithm better protected against the overflow as both sides of (6) are bound to some, usually small values. On the other hand, this algorithm is computationally more complex than the FSCL method. Additionally, in the FSCL algorithm, in contrast to De Sieno's approach, the modified distance between training and weight vectors is always positive. These conclusions are important from the hardware implementation point of view as it will be described in Section III.

III. Implementation of the Proposed Conscience Mechanism

The learning process in the WTA NN starts with generating random weights for all neurons (network initialization) [12], [22], [23]. At the next phase, the training process starts. At this phase, the network is seeking for the winning neuron, whose weight vector is the closest to the given learning vector. As a measure of resemblance between the vectors, squared Euclidean distance has been used. This allows for omitting the square root operation, which contributes to the simplification of the circuit. The winning ith neuron adapts its weight vector in the following way:

$$W_i(k+1) = W_i(k) + \eta \cdot [X(k) - W_i(k)]. \quad (8)$$

The weights associated with other neurons remain unchanged

$$W_{j \ne i}(k+1) = W_{j \ne i}(k). \quad (9)$$

The initial value of η has been selected experimentally. The value of this parameter decreases during the learning process.

A block diagram of the designed network is shown in Fig. 1. It includes four key modules, which have been fabricated in analog ASIC, as described in [10]–[13] and [27]. The Euclidean

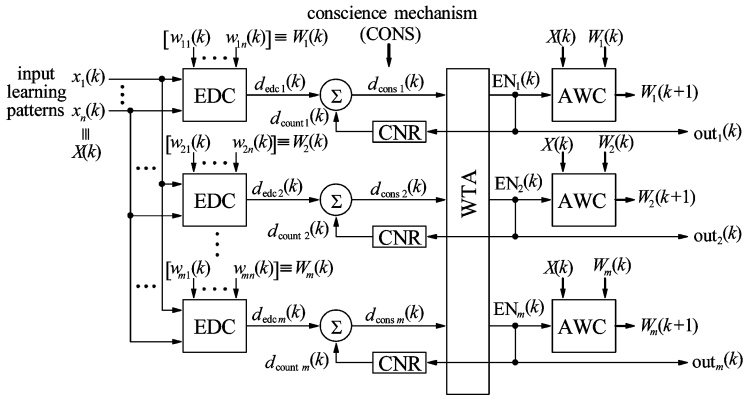

Fig. 1. Block diagram of the WTA neural network implemented by the authors. Signals d_{edc}, d_{cons}, and d_{count} are a part of the hardware realization; they are represented by currents I_{edc}, I_{cons}, and I_{count}.

distance calculation (EDC) blocks determine the value of the square of the Euclidean distances for the particular neurons. In the designed network, the EDC block consists of three identical channels, each containing the comparator, the subtracting circuit (SUB), and the squarer. The comparator compares the input signals x with the signal representing the neuron weight w, and generates two complementary logical signals that control the SUB block as well as the adaptive weight change (AWC) mechanism. The SUB block calculates the absolute value of the difference between the signals x and w. Using the absolute value allows us to use a simple one-quadrant squarer that simplifies the overall EDC block. The binary-tree WTA block [34] identifies the winning neuron, searching for the smallest current I_{cons}. These currents are direct measures of the Euclidean distance.

The AWC mechanism updates the winning neuron's weights. Each of the AWC blocks presented in Fig. 1 consists of two current-mode sample and hold (S&H) memory cells, not shown for simplicity, that work alternately, and an individual controlling circuit. New data that occur at the AWC input—the term "$\eta(x-w)$" in (8)—are summed with the previous value coming from one of these cells and afterwards the result is stored in the second cell. A potential problem with current leakage in this memory occurs, but if the weights are updated, as long as training continues, this leakage is compensated. Nevertheless, the leakage significantly modifies the weights of dead neurons, which are not updated. This has been modeled in our network model as described later in the text, and shown in Fig. 6(c).

Once the adaptation has been completed, an external clock switches over the control circuit, which makes the new weights' values at the neuron's output ready to be used at the next training cycle.

All these blocks operate in the current mode, which means that both the input training signals and the neuron weights are represented by currents and all calculations in the network are performed in the current domain. The current mode has been chosen as it allows for an easy implementation of the summation, as particular signals may be simply added at junctions. This helps avoiding the use of operational amplifiers, which usually dissipate much larger power [19]. This approach significantly simplifies the structure of the building blocks described above as well as the overall network. More details on these functional blocks are reported in [11]–[13] and [27]

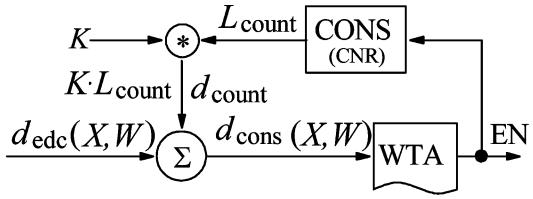

Fig. 2. Block diagram of the proposed conscience mechanism realized with the use of analog counters.

In this study, we concentrate on the conscience mechanism (CONS), which has been also implemented as the part of the network. The general idea of this mechanism is visualized in Fig. 2. This approach takes into consideration the advantage of the current-mode circuits mentioned above.

In this case, the distance between the weight and the training vectors is made higher by adding a signal that is proportional to the number of the wins

$$d_{\mathrm{cons}}(X,W) = d_{\mathrm{edc}}(X,W) + L_{\mathrm{count}} \cdot K \qquad (10)$$

where distance $d_{\mathrm{edc}}(X,W)$, shown in Fig. 2 and described by the term $||W_i - X||^2$, is represented by the current I_{edc}, and $K \cdot L_{\mathrm{count}}$ is a component that is proportional to the number of wins L_{count} of a given neuron. The K coefficient is the conscience gain factor, which allows us to control and optimize the learning process by adjusting the strength of the conscience mechanism.

The proposed conscience mechanism is shown in Fig. 3[10]. It consists of three main components. One of them is the analog counter (CNR), shown in Fig. 3(a). This block is controlled by the ENt impulses that come from the temperature compensation block (CTEMP), shown in Fig. 3(b). The CTEMP block is, in turn, controlled by the input pulses (EN) arriving from the WTA block. The ENt impulses for short periods of time open the current source MP2 in the counter, which in consecutive iterations charges the capacitor C_2. After a series of zero voltage states of the ENt signal, i.e., series of MP2 drain-current pulses that charge the C_2 capacitor, the analog counter reaches an overflow state and may be reset. This is realized by switching on the MN2 transistor controlled by the AND-gate output voltage. The point is that the gate signal must be delayed in relation with the moment when the counter content crossed the overflow level, which is a necessary condition to avoid instability of the counter. This

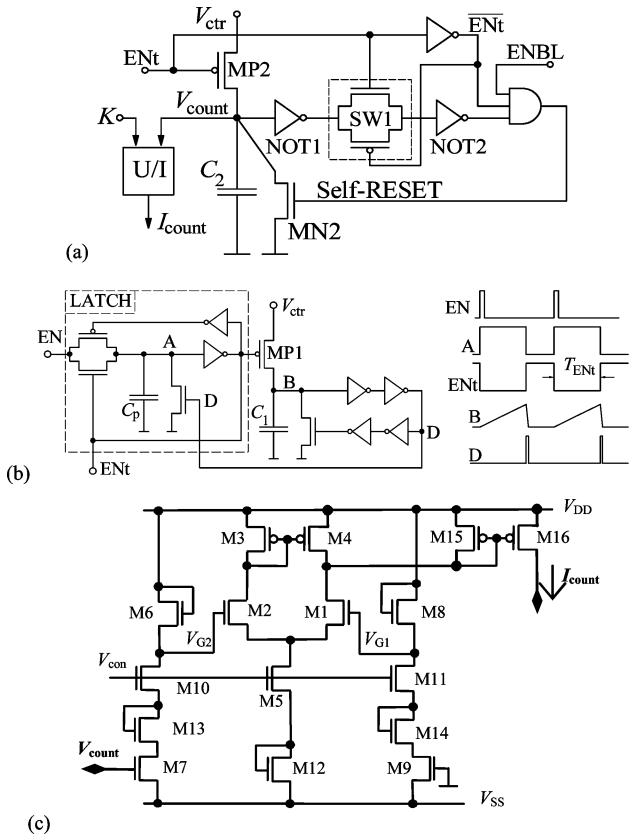

Fig. 3. Main components of the proposed conscience mechanism: (a) an analog counter (CNR), (b) a temperature compensation circuit (CTEMP), and (c) a differential U-I converter.

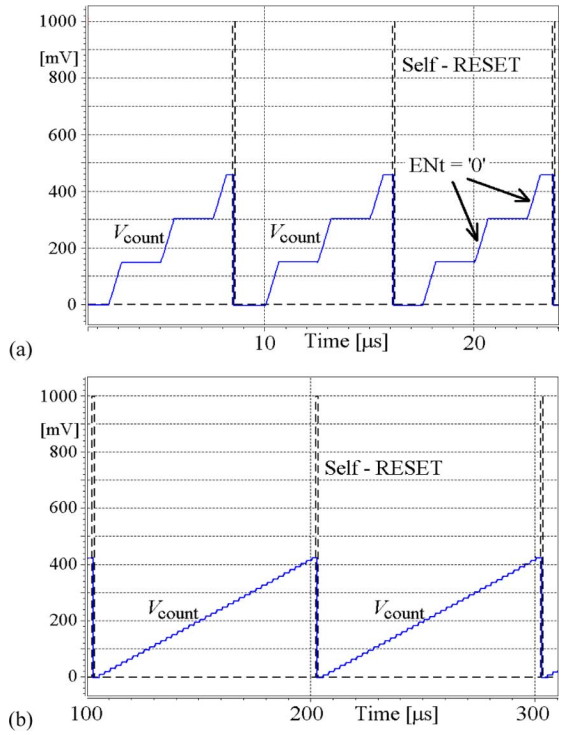

Fig. 4. Illustration of the range flexibility of the analog counter used in the conscience mechanism. Voltage $V_{\rm ctr}$ is regarded as a parameter: (a) counter modulo 3, (b) counter modulo 40.

is ensured by means of the NOT1 and NOT2 inverters and the SW1 switch. The reset function may be activated only in case when the ENBL control signal is equal to logical "1."

The presented analog counter provides a flexible solution, since its modulo can be easily controlled in a wide range, using only a single voltage $V_{\rm ctr}$. Simulation results for the selected values of $V_{\rm ctr}$ voltage are shown in Fig. 4, where they concern modulo 3 and 40.

It is worth mentioning, that we program modulo in such a way to make the $V_{\rm count}$ only rising up and to avoid resetting when the CONS mechanism is active. To avoid unexpected reset, in this phase, the ENBL signal is additionally equal to logical "0."

One of the problems associated with the analog counter presented here is that the charge amount that flows to the capacitor C_2 strongly depends on temperature due to variations of channel resistance of the MP2 PMOS transistor. This means in practice that operation of the conscience mechanism and the learning process become dependent on temperature of the environment. We have solved this problem by applying the CTEMP block, which controls width of the ENt impulses, depending on the variations of the temperature. The circuit realizes a similar principle as the one used in analog counters. The channel resistance of both the MP1 and MP2 transistors is the same over a wide range of temperatures. When, for instance, the channel resistance of the transistors decreases, due to temperature variations, the charging process of the capacitor C_1 becomes faster, and C_1 is reset after a shorter period of time. This in turn decreases the width of the ENt impulses and stabilizes the charge amount that flows to the capacitor C_2.

The $V_{\rm count}$ voltage is stored on the C_2 capacitor and can be treated as the measure of the number of wins related to a given neuron. This voltage is then converted into current $I_{\rm count}$ using a differential U-I converter shown in Fig. 3(c). This current is directly added to the EDC output current $I_{\rm edc}$, according to

$$I_{\text{cons}\,i}(k) = \overbrace{\left[A \cdot \sum_{j=1}^{n} (x_j - w_{ij})^2\right]}^{I_{\text{edc}\,i}(k)} + I_{\text{count}\,i}\,(k). \qquad (11)$$

The converter consists of a differential transconductor (transistors M1, M2, M3, M4), a tail current (M5, M12) biasing the differential pair M1–M2, two voltage attenuators, and an output current mirror (CM) M15–M16. This mirror serves as a current amplifier, which adjusts the U-I converter output current $I_{\rm count}$ to the level of currents delivered by the EDC blocks (see Fig. 1), which represent the Euclidean distance $d_{\rm edc}(X, W)$, shown in Fig. 2. One of the attenuators (M6, M7, M10, M13) delivers a suitable voltage $V_{\rm G2}$ to gate of M2 and the other attenuator (M8, M9, M11, M14) delivers the voltage $V_{\rm G1}$ to gate of M1. For a small-signal operation of the transconductor, $I_{\rm count}$ is linearly related to $V_{\rm G2}$, according to [28]

$$I_{\rm count} \cong (V_{\rm G2} - V_{\rm G1})\sqrt{2B_{\rm R}I_{\rm B}}. \qquad (12)$$

$I_{\rm B}$ is a drain current of M5 (tail current of the M1–M2 pair) and $B_{\rm R}$ is a real-valued coefficient dependent on the aspect ratios of M1 and M2. As can be seen from (12), the gain of the transconductor can be controlled by the current $I_{\rm B}$, which is dependent on the control voltage $V_{\rm con}$.

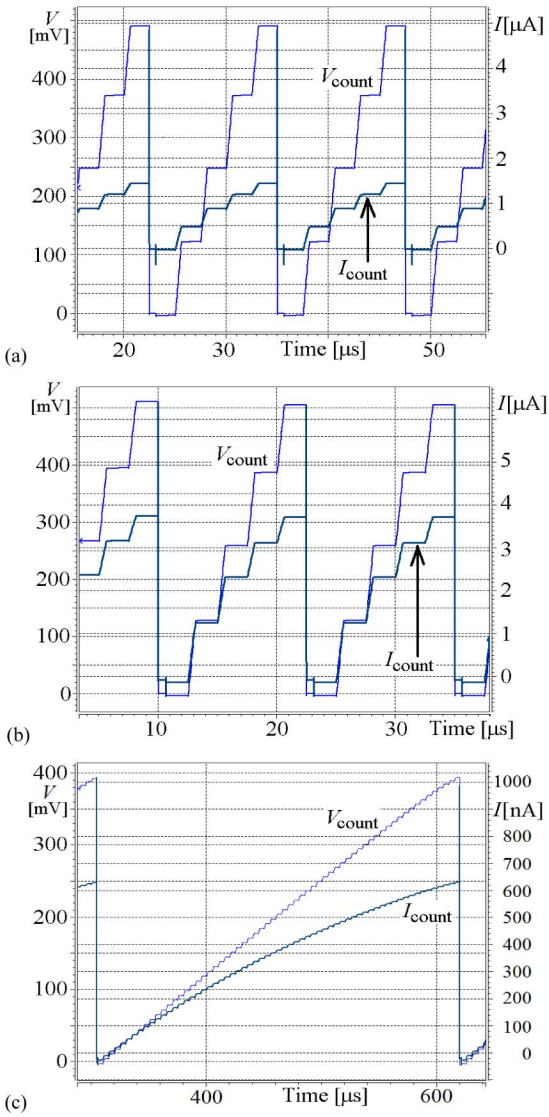

Fig. 5. Influence of the U-I circuit gain controlled by the $V_{\rm CON}$ voltage on the current $I_{\rm count}$ at the CONS mechanism output for: (a) $V_{\rm CON} = 0.9$ V, (b) $V_{\rm CON} = 1.3$ V, and (c) another modulo and $V_{\rm CON} = 0.7$ V.

Finally, as shown in Fig. 5, the $V_{\rm con}$ voltage controls the gain of the overall U-I converter. The presented waveforms come from HSPICE postlayout simulations.

The role of the left-hand side attenuator is to enable a wide variation range of the input voltage $V_{\rm count}$ for small variations of $V_{\rm G2}$, i.e., to ensure a small signal operation of the transconductor. The attenuator can work linearly with a small negative gain coefficient, if the circuit is supplied by a sufficiently high $V_{\rm DD} - V_{\rm SS}$ voltage difference. This, unfortunately, is in conflict with the required low-power consumption of the converter. In our case, a low supply voltage option has been applied to minimize power consumption, which results in a nonlinear relation between $V_{\rm G2}$ and $V_{\rm count}$. This nonlinearity is the main reason for nonlinear character of the converter transfer function visible in Fig. 5(c).

The ASIC design affects the properties and effectiveness of the obtained conscience mechanism realized on silicon. Of some importance also is the realization mode (analog, digital, or mixed) as well as the operation mode of the applied signal processing circuits. The last issue is essential in the case of analog circuits. As mentioned above, in current-mode circuits, it is extremely easy to implement summation operations as addition can be carried out at a single node. Multiplication operations performed in this mode are significantly more difficult to implement than the addition. That is the reason why we have decided to utilize De Sieno's approach as a basis for this hardware realization. On the other hand, in De Sieno's algorithm, the variables b may be either positive or negative. We have modified this algorithm to avoid this rather inconvenient situation. In the current-mode circuits, it is much easier to work with positive signals only, as this simplifies the entire circuit and minimizes both the power dissipation and the chip area. In this sense, the proposed method is also similar to the FSCL method, mentioned in Section II, in which the term $K \cdot L_{\rm count}$ standing in formula (1) is nonnegative. An overflow problem encountered in the FSCL approach is not critical to the implementation of the analog counter and the U-I converter due to a natural hardware saturation of the counter signals, as seen, for example, in Fig. 5(c).

In other words, the counter is unable to work with $V_{\rm count}$ higher than $V_{\rm ctr}$. Moreover, the direct current (dc) characteristics of the U-I converter are nonlinear and its output current $I_{\rm count}$ saturates with an increase of $V_{\rm count}$. In this way, we naturally approach the functionality of De Sieno's algorithm without having to implement in hardware formulas (4) and (7). Furthermore, in the software implementation of De Sieno's algorithm, the relationship (4) is determined for all neurons. In the case of neurons that did not win the competition, it introduces a kind of "amnesty" mechanism as variables p_j decrease for all losing neurons. In the hardware implementation, such an amnesty mechanism, depending on transistor sizes, results from the leakage effect occurring in the analog memory cells used in the counter. This occurs despite the fact that this leakage is small enough to keep the circuit functional when clock frequency is sufficiently high (0.5–2 MHz).

To summarize, the described features of the analog counter allow for a very easy hardware realization of the proposed conscience mechanism, which can be effectively adjusted to the working conditions and used in networks with a different number of neurons. The proposed circuit is characterized by low power dissipation and low chip area. When the conscience mechanism is active, the power consumed by a single CONS blocks is 70 μW, while the chip area of this block is equal to 1400 μm^2. When the neural network works with 1-MHz clock frequency, the CONS mechanism is active during the period of 1400 μs, consuming at that time an average energy of 28–30 nJ per neuron.

IV. Verification of the Proposed Conscience Mechanism

A. System Level Simulations

Initially, the performance of the proposed conscience mechanism has been verified for different values of the parameters of a basic model of the network. Next, to make the obtained results more real, additional parameters of the model, resulting

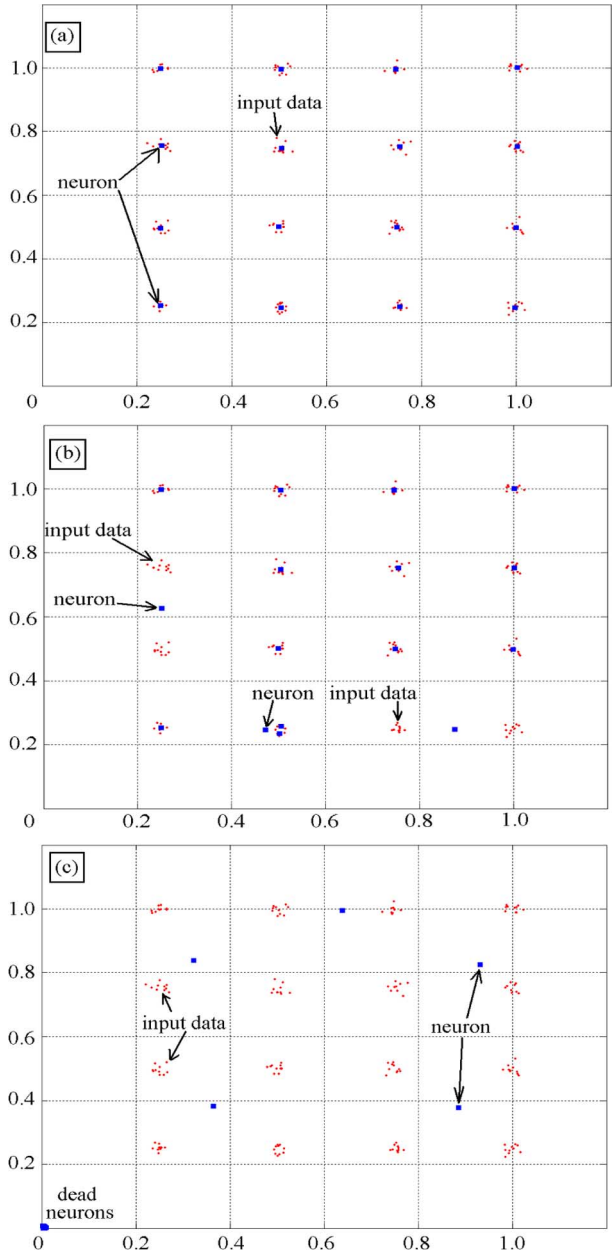

Fig. 6. Simulation results (Matlab) showing the WTA NN learning for different values of the K coefficient, i.e.: (a) for $K = 1$ (optimal case), (b) for $K = 0.05$, and (c) for $K = 0$, i.e., without the conscience mechanism. The input signals are normalized to [1].

from circuit layout and physical phenomena, like charge injection effect of the used switches and memory leakage currents, have been added to the basic model [11].

In the simulations, the results of which are presented in Fig. 6, the initialization of the weights of the neurons has been performed before starting the learning process. As indicated, these weights are assigned small random values. In the ideal case, the initial values should make the neurons "cover" an entire input data space. The small initial values have been selected since we wanted to be sure that the obtained results are mostly the result of the use of the conscience mechanism itself.

The term responsible for modeling the leakage effect causes that after several hundred presentations of the training pattern, the information about dead neurons has been lost, thus zeroing the weights of these neurons and in this way placing them close to the origin of the data space.

To illustrate the performance of the neural network for different settings of the conscience mechanism, an example test with the training data containing 160 2-D learning vectors representing 16 different classes has been illustrated in Fig. 6. The network was composed of 16 neurons, so that the potential number of the network output (classes) was also equal to 16. We started with the initial value of the learning coefficient η equal to 0.9. After a few epochs, this value was reduced to 0.05. The total number of all epochs was equal to 100. Fig. 6 illustrates the influence of the strength of the conscience mechanism (the K coefficient) on the learning process.

An adequate value of this parameter must be determined experimentally for a given number of neurons in the network, although the experiments have shown that there are some values of this parameter that may be considered to be quite acceptable in a certain range of the network size, i.e., the number of neurons. Fig. 6(a) and (b) confirms that all neurons participate in the learning process due to the conscience mechanism applied there. However, only in the case shown in Fig. 6(a), the obtained results are correct and all 16 neurons have become representatives of the 16 regions of the input data space. In the case shown in Fig. 6(b), the too weak conscience mechanism becomes responsible for the observed learning disruptions. The learning process without this mechanism is demonstrated in Fig. 6(c). As can be seen, only five neurons actively participated in the learning process, winning in particular iterations. The others became dead neurons.

Similar experiments have been completed for different sizes of the network, different distributions of the training data, and different values of the K parameter.

To summarize, in Fig. 7, we show the values of the normalized quantization error, which is regarded as a measure of the quality of the training process. As expected, when the conscience mechanism is turned off (as K approaches 0), then the quantization error becomes the largest. Increasing the values of K results in the error decrease and finally network reaches the smallest value of this error, for which all neurons are placed in proper regions of the input data space. Further increase of this parameter shows no impact on the learning process. The value of K equal to 1 may be considered as the optimal one, as in that case the number of improperly placed neurons tends to 0.

As already mentioned, instead of the Euclidean distance we have decided to use its square as a measure of dissimilarity between two vectors, which simplifies the resulting hardware realization. To highlight the characteristic features of our approach, consider the network input signals and values of neuron weights to be normalized so that they are located in the $[0, 1]$ range. Let us denote by $d_{\max}$ the square of the largest possible distance between any two points in an n-dimensional data space. The value of $d_{\max}$, i.e., the square of the diagonal of this 0–1 hypercube, equals n. For example, in 2-D data space, $d_{\max}$ equals 2. If in such a space, the value of the K parameter equals 1, then the conscience mechanism block of the winning neuron always increases by $0.5 \cdot d_{\max}$ ($0.33 \cdot d_{\max}$ for $n = 3$) a distance d_{edc} between the input vector X and the weight vector W of the neuron. Under conditions like these, the conscience mechanism

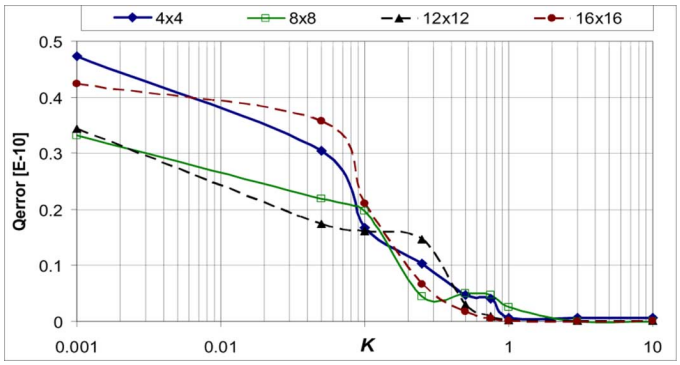

Fig. 7. Normalized quantization error as a function of the K coefficient and different numbers of neurons in the network.

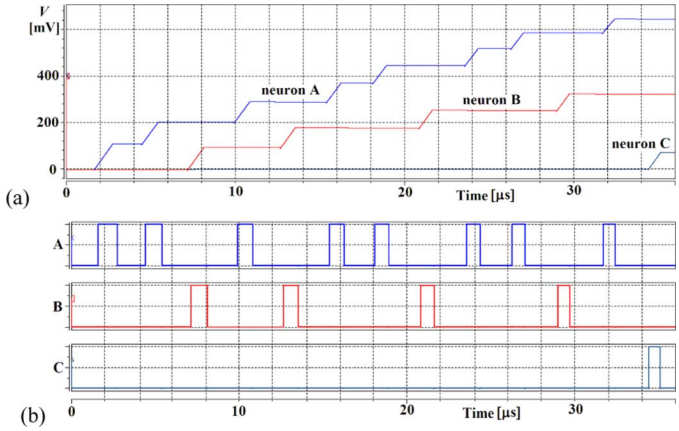

Fig. 8. WTA signals: (a) analog counter signals used in the conscience mechanism for the A, B, and C neurons, (b) EN ("enable") signals at the WTA outputs (shown in Fig. 1).

can be considered to be "strong" and, in most cases, it allows for a proper placement of all neurons in the data space. On the other hand, when the conscience mechanism is "weak," some neurons remain dead or are not properly placed after the training has been completed. Example results illustrating this situation are shown in Fig. 6(b) where $K = 0.05$. Here, about 25% of all neurons are badly placed. For $K = 0.05$, the conscience mechanism increases the distance d_{edc} related to the winning neuron, by a factor of $0.025 \cdot d_{\max}$. Nevertheless, these results are still much better than those obtained when the conscience mechanism is turned off, as shown in Fig. 6(c), where around 70% of neurons are dead.

The simulation results presented in Figs. 6 and 7 are compared with the measurements obtained in our prototype network. In the presented experiments, the input signals of the WTA block are in the range from 3 to 8 μA, which means that their amplitudes are equal to 5 μA. The low value of 3 μA is the bias current, necessary to properly set up the working points of transistors included in some other blocks of the network. To minimize the power consumption, we have assumed the input signals to be below 20 μA.

One of the problems in current-mode circuits is the influence of technological mismatches between transistor parameters on a gain error of CM. Since in our network current mirrors are basic elements, this occurrence affects precision of the overall network significantly and needs to be minimized. One of the possible solutions of this problem is to increase transistor sizes, as this improves the matching between transistors [30]. Unfortunately, this solution is insufficient in the case of small signals. Increasing transistor sizes, for constant values of CM currents, we decrease the value of the gate-to-source voltage V_{GS} of these transistors. This increases the gain error [30] and makes the matching improvement weaker. A careful analysis of this problem revealed that from the precision point of view, transistors with sizes (W/L) of ca. 3/1–5/1 μm are the most suitable for the assumed values of currents. Such transistors are unable to conduct considerably larger currents, which imposes a limitation on the strength of the conscience mechanism. For example, setting the value of the K parameter to be equal to 1 causes saturation of the currents (being positioned close to the upper level) after a few wins of each neuron. For this reason, in the experiments performed at the circuit level, we have chosen smaller values of K, i.e., those varying in between 0.05 and 0.1, that are somewhat suboptimal as far as precision is concerned. This effect will be discussed later in the paper.

The results shown in Fig. 7 are concerned with the software model of the network. In this case, we do not observe the limitations that occur in hardware implementation, as described above. For this reason, the presented results for $K > 1$ differ significantly between both implementations. Here, the hardware realization of the network does not work properly.

B. Circuit Level Postlayout Simulations

At this stage of the chip designing, electrical parameters of all network components have been carefully verified and modified to minimize, as far as possible, the negative influence of parasitic capacitances and leakage currents on the network properties and to minimize both its power consumption and chip area occupation. Illustrative results of the HSPICE simulation carried out at the transistor level are presented in Fig. 8.

In the performed test, the results of which are presented in Fig. 8, a network with two inputs and three neurons has been used. Both the training vector X and the neuron weight vectors W, represented by currents expressed in μA, are 2-D with the following entries: $X = [1, 1]$, $W_{\mathrm{A}} = [1.2, 0.9]$, $W_{\mathrm{B}} = [1.3, 0.8]$, and $W_{\mathrm{C}} = [1.3, 0.3]$.

The calculated "real" distances between particular W vectors and the training X vector were constant during the entire test.

For these parameters, when the conscience mechanism was turned off, only neuron A won the competition, as the distance between its weight vector W and X was always the smallest. For the conscience mechanism being turned on, with all counters being earlier zeroed, at the beginning of the competition neuron A was the winner and the value stored in the win counter associated with this neuron was increased, enlarging the distance. After several presentations of the same training vector X, the resulting modified distance of this neuron to X vector became large enough to enable neuron B to win. Next, neurons A and B won in an alternate fashion. Finally, after the distances of neurons A and B to X vector became large enough, neuron C also won.

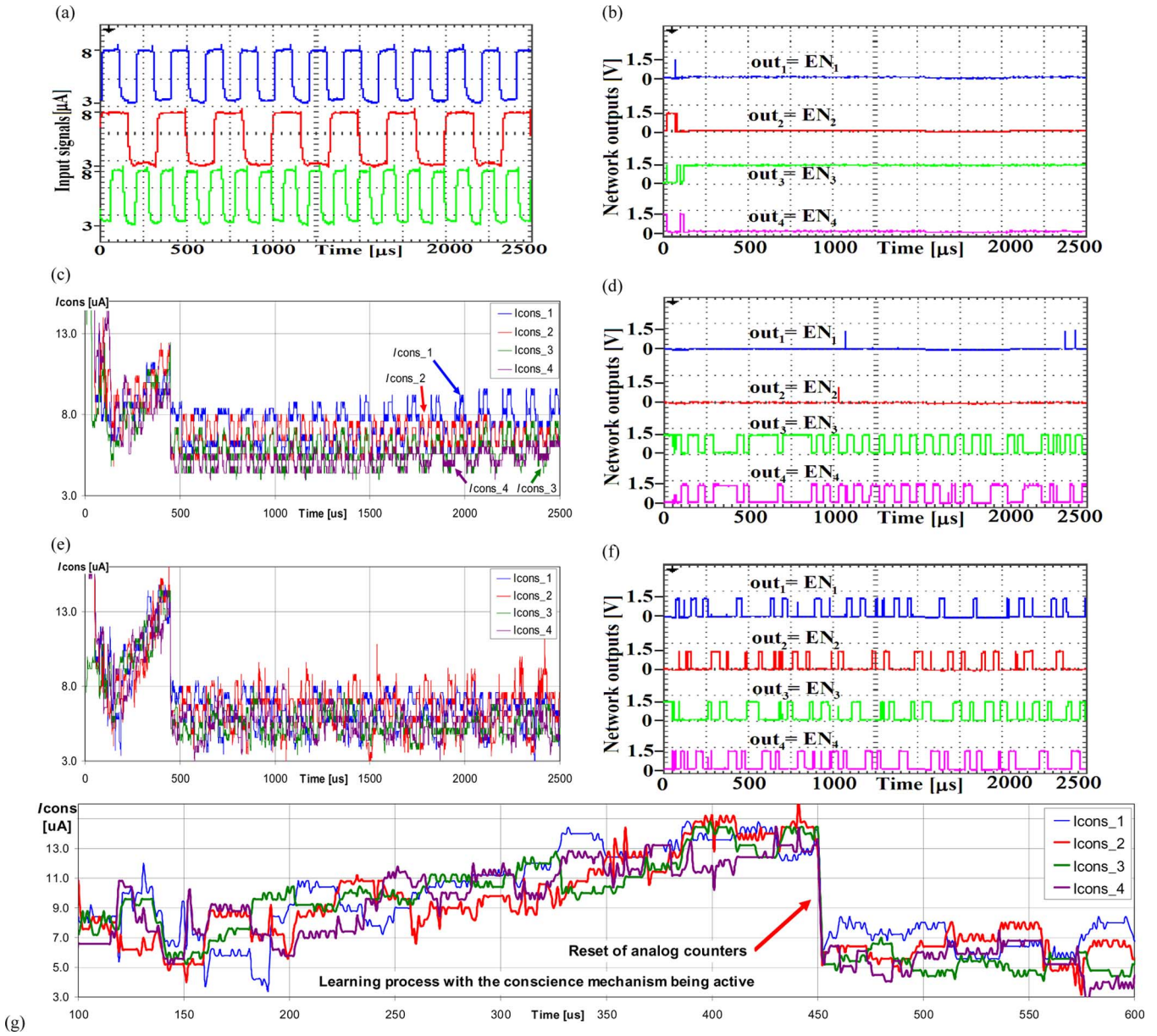

Fig. 9. Influence of the conscience mechanism on the learning process for different values of the K parameter: (a) training input signals x, (c) and (e) the EDC output currents I_{edc} for different setting of the conscience mechanism (K parameter), (d) and (f) the network outputs for cases (c) and (e), respectively; (b) results for the conscience mechanism being inactive; (g) the WTA block input in the early phase of the learning process—the plot has been zoomed out from the plot (e) for better illustration.

The results in Fig. 8 concern optimal settings of the conscience mechanism. When the values of K were too small, then neuron A won more frequently, excluding other neurons from the competition for a long time.

On the other hand, the values of K larger than the optimal setup imply greater currents at the inputs of the WTA circuit which unnecessarily increases the power dissipation of the circuit, while the system performance is left unchanged. This observation is in agreement with the results for the former system level simulations presented in Fig. 7.

C. Measurement Results

Some experimental results of the WTA NN implemented on a chip in the CMOS 0.18-μm process and learned on silicon are presented in Fig. 9. The prototype network consists of four neurons, each having three inputs ($n = 3$). Measurement results that illustrate an overall training process have been reported in [13]. Here, we focus on the results that concern an influence of the conscience mechanism on the learning process. The input currents (x_1, x_2, and x_3) are pulse signals ranging from 1 to 3 μA, with the frequencies of 5, 3, and 7 kHz, respectively, while the network clock frequency is equal to 1 MHz. The first stage of this process is an initialization that takes 150 μs [12]. In the following 300 μs, the learning process is aided by the CONS mechanism and the EDC output currents I_{edc} are increased by adding a current component coming from this mechanism. After 450 μs, the CONS mechanism is turned off that zeroes the analog counters and decreases the WTA input currents I_{cons} as shown in Fig. 9(c), (e), and (g).

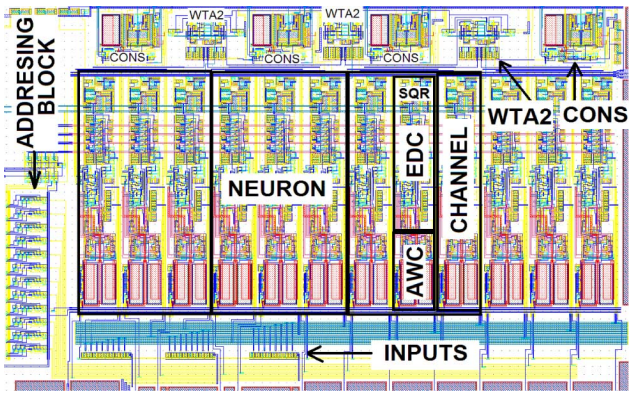

Fig. 10. Layout of the experimental current-mode analog network designed in CMOS 0.018-μm technology. The circuit sizes are 320×200 μm^2.

The input periodical signals shown in Fig. 9(a) have been selected in such a way that we obtain a representative set of the input vectors X. Fig. 9(c) and (e) illustrates the resulting waveforms of the EDC output currents of particular neurons, increased by currents supplied by the conscience mechanism block, according to (11), for two values of the dc voltage $V_{\rm con}$ that controls the K parameter of the CONS mechanism. Fig. 9(d) and (f) illustrates the network output signals in cases (c) and (e), respectively. The results in Fig. 9(c) and (d) concern a very weak conscience mechanism. The amplitude of the $I_{\rm edc}$ currents, i.e., the equivalent of $d_{\max}$ parameter, equals 5 μA. The CONS mechanism block, during its activity period, increases these currents by about 4 μA, i.e., 80% of $d_{\max}$. This means that an effective value of the K parameter is equal to 0.032. In this case, two neurons remain dead and therefore this situation is not optimal at all. The results in Fig. 9(e) and (f) deal with a stronger CONS mechanism, which enlarges the $I_{\rm edc}$ currents by ca. 7.5 μA, i.e., by 150% of $d_{\max}$. The effective value of the K parameter used here is equal to 0.061. This value allows for activating all neurons in the network and therefore this case can be considered as quasi-optimal. The results shown in Fig. 9(b) refer to a situation where the CONS mechanism is not active. As a result, three neurons remained dead. The measurement results are in good agreement with those obtained in simulation studies.

It is both interesting and useful to compare the performance of analog networks with their digital counterparts as presented, for example, in [19]. Analog networks are substantially more difficult in realization and the undertaking of this effort makes sense if such networks are able to offer much better performance. We compared the performance (expressed in terms of power dissipation and data rate) of our network implemented in CMOS technology and a comparable network with the same number of inputs and outputs implemented in software (C++). The prototype analog network, at the clock frequency of 1 MHz, was able to operate as fast as a counterpart network working on a typical PC with the clock frequency of 2 GHz. These results have been thoroughly verified for a wide range of the neural network parameters. The calculation time in software networks increases approximately linearly with the number of channels that equals $n \cdot m$. In the hardware implemented WTA NN, computing time is independent of the number of channels, which becomes an important advantage of such implementation.

It is also important to stress that the proposed network consumes the power of 1 mW when the CONS mechanism is active and only 700 μW without this mechanism. This is only a small fraction of the power consumed by a PC running the same learning task. Since the presented network is the first prototype, being designed to verify the concept of the particular components, there is still plenty of room for further optimization with regard to power dissipation.

The internal structure (layout) of the network is shown in Fig. 10. The circuit realized in the CMOS 0.18-μm process occupies an area of 0.07 mm^2.

The obtained results allow for making prediction of the chip area in case of the larger number of neurons. A single neuron with n weights consists of n identical channels that in current mode technique are connected by simply shorting their outputs together. In the prototype chip, we have included 12 identical channels grouped into four neurons. In case of two neurons with six inputs or six neurons with two inputs the number of channels will remain the same and the overall chip area will not significantly change. The number of neurons has the influence on the area of the WTA as well as the CONS blocks, which increases linearly with the number of neurons in the network.

This demonstrates that the realization of larger networks with hundreds of neurons is feasible and could incur low cost.

V. Conclusion

This paper presents both the idea and chip implementations of a conscience mechanism used to improve learning of WTA networks, i.e., to eliminate dead neurons.

The general idea of the proposed mechanism is based on the use of analog counters. The associated advantage is that the counting range (modulo) of analog counters can be easily controlled in a wide range by means of an external voltage. The paper offers one of the first attempts at a comprehensive design and implementation involving not only simulations of the software model but also the use of measurement coming from the real hardware. The results confirm that the conscience mechanism has a very positive impact on the effectiveness of learning of the WTA neural network. The prototyped small network shows that the implementing of a large WTA NN in analog hardware (or Kohonen NN when the neighborhood mechanism is additionally used) capable of power efficient learning on silicon on a large scale is a realistic endeavor.

References

[1] G. Cauwenberghs, M. A. Bayoumi, and E. Sanchez-Sinencio, *Learning on Silicon: Adaptive VLSI Neural Systems*. Norwell, MA: Kluwer, 1999.

[2] S. Fakhraie and K. C. Smith, *VLSI-Compatible Implementations for Artificial Neural Networks*. Norwell, MA: Kluwer, 1997.

[3] B. Linares-Barranco, E. Sanchez-Sinencio, A. Rodriguez-Vazquez, and J. L. Huertas, "A CMOS analog adaptive BAM with on-chip learning and weight refreshing," *IEEE Trans. Neural Netw.*, vol. 4, no. 3, pp. 445–455, May 1993.

[4] J. Choi, S. H. Bang, and B. J. Sheu, "A programmable analog VLSI neural network processor for communication receivers," *IEEE Trans. Neural Netw.*, vol. 4, no. 3, pp. 484–495, May 1993.

[5] H. Hikawa, "Frequency-based multilayer neural network with on-chip learning," *IEEE Trans. Neural Netw.*, vol. 10, no. 3, pp. 545–553, May 1999.

[6] R. Coggins, M. Jabri, B. Flower, and S. Pickard, "A hybrid analog and digital VLSI neural network for intracardiac morphology classification," *IEEE J. Solid-State Circuits*, vol. 30, no. 5, pp. 542–550, May 1995.

[7] C. Mead, *Analog VLSI and Neural Systems*. Reading, MA: Addison-Wesley, 1989.
[8] M. Holler, S. Tam, H. Castro, and R. Benson, "An electrically trainable artificial neural network (ETANN) with 10240 'Floating gate' synapses," in *Proc. Int. Joint Conf. Neural Netw.*, Washington, DC, Jun. 1998, pp. 191–196.
[9] H. C. Card, D. K. McNeill, and C. R. Schneider, "Analog VLSI circuits for competitive learning networks," *Analog Integr. Circuits Signal Process.*, vol. 15, no. 3, pp. 291–314, Mar. 1998.
[10] T. Talaśka, R. Długosz, W. Wojtyna, K. Iniewski, and W. Pedrycz, "Analog-counter-based conscience mechanism in Kohonen's neural network implemented in CMOS 0.18 μm technology," in *Proc. IEEE Workshop Signal Process. Syst. Design Implement.*, Banff, AB, Canada, Oct. 2006, pp. 416–421.
[11] T. Talaśka, R. Długosz, and W. Pedrycz, "Adaptive weight change mechanism for Kohonens's neural network implemented in CMOS 0.18 μm technology," in *Proc. Eur. Symp. Artif. Neural Netw.*, Bruges, Belgium, Apr. 2007, pp. 151–156.
[12] T. Talaśka and R. Długosz, "Initialization mechanism in Kohonen neural network implemented in CMOS technology," in *Proc. Eur. Symp. Artif. Neural Netw.*, Bruges, Belgium, Apr. 2008, pp. 337–342.
[13] T. Talaśka, R. Długosz, J. Dalecki, and R. Wojtyna, "Experimental Kohonen neural network implemented in CMOS 0.18 μm technology," in *Proc. Int. Conf. Mixed Design Integr. Circuits Syst.*, Poznań, Poland, Jun. 2008, pp. 243–248.
[14] T. Kohonen, *Self-Organizing Maps*, 3rd ed. Berlin, Germany: Springer-Verlag, 2001.
[15] R. Rizzo and A. Chella, "A comparison between habituation and conscience mechanism in self-organizing maps," *IEEE Trans. Neural Netw.*, vol. 17, no. 3, pp. 807–810, May 2006.
[16] S. C. Ahalt, A. K. Krishnamurthy, P. Chen, and D. E. Melton, "Competitive learning algorithms for vector quantization," *Neural Netw.*, vol. 3, pp. 277–290, 1990.
[17] D. E. Rumelhart and D. Zipser, "Feature discovery by competitive learning," *Cogn. Sci.*, vol. 9, no. 1, pp. 75–112, Jan.-Mar. 1985.
[18] D. DeSieno, "Adding a conscience to competitive learning," in *Proc. IEEE Conf. Neural Netw.*, San Diego, CA, Jul. 1988, vol. 1, pp. 117–124.
[19] L. Gatet, H. Tap-Béteille, and F. Bony, "Comparison between analog and digital neural network implementations for range-finding applications," *IEEE Trans. Neural Netw.*, vol. 20, no. 3, pp. 460–470, Mar. 2009.
[20] W. Zhang, S. Jing, L. Gui, and Y. Zhang, "Physiological data acquisition system for education assessment using wireless sensor network," in *Proc. Int. Conf. Technol. Appl.*, Shenzhen, China, May 2008, pp. 471–473.
[21] S.-L. Chen, H.-Y. Lee, Y.-W. Chu, C.-A. Chen, C.-C. Lin, and C.-H. Luo, "A variable control system for wireless body sensor network," in *Proc. IEEE Int. Symp. Circuits Syst.*, Seattle, WA, May 2008, pp. 2034–2037.
[22] G. Thimm and E. Fiesler, "Neural network initialization," in *From Neural to Artificial Neural Computation*. Berlin, Germany: Springer-Verlag, 1995, vol. 930, pp. 535–542, DOI: 10.1007/3-540-59497-3_220, ISSN: 0302-9743, ISBN: 978-3-540-59497-0.
[23] Y. Chen and F. Bastani, "ANN with two-dendrite neurons and its weight initialization," in *Proc. Int. Joint Conf. Neural Netw.*, Baltimore, MD, Jun. 1992, pp. 139–146.
[24] A. Rajah and M. H. Khalil, "ASIC design of a Kohonen neural network microchip," in *Proc. IEEE Int. Conf. Semicond. Electron.*, Kuala Lumpur, Malaysia, Dec. 2004, pp. 148–151.
[25] C.-Y. Wu and W.-K. Kuo, "A new analog implementation of the Kohonen neural network," in *Proc. Int. Symp. VLSI Technol. Systems Appl.*, Taipei, Taiwan, May 1993, pp. 262–266.
[26] D. Macq, M. Verleysen, and P. J.-D. Legat, "Analog implementation of a Kohonen map with on-chip learning," *IEEE Trans. Neural Netw.*, vol. 4, no. 3, pp. 456–461, May 1993.
[27] T. Talaśka and R. Długosz, "Current mode Euclidean distance calculation circuit for Kohonen's neural network implemented in CMOS 0.18 μm technology," in *Proc. IEEE Can. Conf. Electr. Comput. Eng.*, Vancouver, BC, Canada, Apr. 2007, pp. 437–440.
[28] R. Wojtyna and T. Talaśka, "Improved power-saving synapse for adaptive neuroprocessing on silicon," in *Proc. IEEE Int. Conf. Signals Electron. Syst.*, Poznań, Poland, Sep. 2004, pp. 27–30.
[29] P. Somervuo and T. Kohonen, "Self-organizing maps and learning vector quantization for feature sequences," *Neural Process. Lett.*, vol. 10, no. 2, pp. 151–159, Oct. 1999.
[30] J. A. Croon, M. Rosmeulen, S. Decoutere, W. Sansen, and H. E. Maes, "An easy-to-use mismatch model for the MOS transistor," *Int. J. Solid-State Circuits*, vol. 37, no. 8, pp. 1056–1064, Aug. 2002.
[31] C.-H. Chang, P. Xu, R. Xiao, and T. Srikanthan, "New adaptive color quantization method based on self-organizing maps," *IEEE Trans. Neural Netw.*, vol. 16, pp. 237–249, Jan. 2005.
[32] R. Długosz and M. Kolasa, "CMOS programmable asynchronous neighborhood mechanism for WTM Kohonen neural network," in *Proc. Int. Conf. Mixed Design Integr. Circuits Syst.*, Poznań, Poland, Jun. 2008, pp. 197–201.
[33] M. Kolasa and R. Długosz, "Parallel asynchronous neighborhood mechanism for WTM Kohonen network implemented in CMOS technology," in *Proc. Eur. Symp. Artif. Neural Netw.*, Bruges, Belgium, Apr. 2008, pp. 331–336.
[34] R. Długosz and T. Talaśka, "Low power current-mode binary-tree asynchronous Min/Max circuit," *Microelectron. J.*, vol. 41, no. 1, pp. 64–73, Jan. 2010.

Rafał Długosz received the M.Sc. degree in automatic and robotic and the Ph.D. degree in telecommunication (with honors) from Poznań University of Technology (PUT), Poznań, Poland, in 1996 and 2004, respectively.

Since 2008, he has been a Scientist at the Swiss Federal Institute of Technology in Lausanne (EPFL), Switzerland, Institute of Microtechnology IMT. From 1996 to 2001, he was with the Institute of Electronics and Telecommunication, PUT. Since 2001, he has been with the Department of Computer Science and Management, PUT. He is the fellow of several scientific fellowships from Foundation for Polish Science in Poland for young scientists and young doctors and from European Union funds (Marie Curie Outgoing International Fellowship). During these fellowships, between 2005 and 2008, he was with the Department of Electrical and Computer Engineering, University of Alberta, Canada. He has published over 90 research papers and book chapters. His main research areas are ultralow power reconfigurable analog and mixed analog-digital integrated circuits such as analog finite impulse response filters, analog-to-digital converters (ADCs), and artificial neural networks.

Tomasz Talaśka received the M.Sc. degree and the Ph.D. degree in telecommunication from the Institute of Telecommunication, University of Technology and Live Sciences, Bydgoszcz, Poland, in 2002 and 2009, respectively.

Currently, he is an Assistant Professor at the University of Technology and Life Sciences, Bydgoszcz, Poland. His research areas are ultralow power analog and analog-digital integrated circuits and hardware implementation of the artificial neural networks especially self-organized neural networks. He is the fellow of two research grants granted by European Union and by Polish Government. He is coauthor of more than 20 research papers.

Witold Pedrycz (M'88–SM'90–F'99) received the M.Sc., Ph.D., and D.Sci. degrees from the Silesian University of Technology, Gliwice, Poland, in 1977, 1980, and 1984, respectively.

He is a Professor and Canada Research Chair (CRC) in Computational Intelligence at the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada. He is also with the Polish Academy of Sciences, Systems Research Institute, Warsaw, Poland. His research interests encompass computational intelligence, fuzzy modeling, knowledge discovery and data mining, fuzzy control including fuzzy controllers, pattern recognition, knowledge-based neural networks, granular and relational computing, and software engineering. He has published numerous papers in these areas. He is also an author of 12 research monographs.

Dr. Pedrycz has been a member of numerous program committees of IEEE conferences in the area of fuzzy sets and neurocomputing. He serves as the Editor-in-Chief of the IEEE TRANSACTIONS ON SYSTEMS MAN AND CYBERNETICS—PART A: SYSTEMS AND HUMANS and an Associate Editor of the IEEE TRANSACTIONS ON FUZZY SYSTEMS. He is also the Editor-in-Chief of *Information Sciences*. He is a recipient of the prestigious Norbert Wiener award from the IEEE Society of Systems, Man, and Cybernetics and an IEEE Canada Silver Medal in Computer Engineering.

Ryszard Wojtyna was born in Bydgoszcz, Poland, in 1950. He received the M.Sc. degree in electronics from Technical University of Gdańsk, Gdańsk, Poland, in 1974 and the Ph.D. degree from the Technical University of Poznań, Poznań, Poland, in 1982.

In 1974, he joined the Institute of Telecommunication, University of Technology and Live Sciences, Bydgoszcz, Poland, and in 1997, the Academy of Information Technology, Institute of Computer Science, Łódź, Poland. Currently, he is a Professor at both universities. His research interests are in the field of analog integrated circuits and signal processing, computer science, hardware neuroprocessing, and power economic electronics.

Dr. Wojtyna is a member of the Polish Electrical Engineers Association (SEP).