Rafał DŁUGOSZ[1,2,3], Tomasz TALAŚKA[2], Ryszard WOJTYNA[2]

Swiss Federal Institute of Technology in Lausanne (EPFL) Neuchatel, Switzerland (1)
University of Technology and Life Sciences, Faculty of Telecommunication and Electrical Engineering, Bydgoszcz, Poland (2)
Poznań University of Technology, Department of Computer Engineering, Poland (3)

An influence of current-leakage in analog memory on training Kohonen neural network implemented on silicon

Abstract. *The paper presents how the current leakage encountered in capacitive analog memories affects the learning process of hardware implemented Kohonen neural networks (KNN). MOS transistor leakage currents, which strongly depend on temperature, increase the network quantization error. This effect can be minimized in several ways discussed in the paper. One of them relies on increasing holding time of the memory. The presented results include simulations in Matlab and HSpice environments, as well as measurements of a prototyped KNN realized in a 0.18μm CMOS process.*

Streszczenie. *W pracy pokazano jak prąd upływu występujący w analogowych komórkach pamięci wpływa na proces uczenia w sprzętowych realizacjach sieci neuronowych Kohonena (KNN). Prądy upływu w tranzystorze MOS, które mocno zależą od temperatury, zwiększają błąd kwantyzacji sieci. Efekt ten może być minimalizowany na różne sposoby, omówione w pracy. Jeden z nich polega na wydłużeniu czasu przechowywania informacji w komórkach pamięci. Przedstawione wyniki zawierają symulacje w środowiskach Matlab i Hspice, a także badania laboratoryjne prototypu sieci KNN, wykonanego w technologii CMOS 0.18μm. (**Wpływ prądu upływu w analogowych komórkach pamięci na proces uczenia w sprzętowych realizacjach sieci neuronowej Kohonena**).*

Keywords: analog neural networks, Kohonen networks, memory leakage effect, hardware signal processing
Słowa kluczowe: analogowe sieci neuronowe, sieci Kohonena, upływność informacji, sprzętowe przetwarzanie sygnałów

Introduction

One of the most critical problems in hardware implemented neural networks is how to precisely store information about neuron weights. The way of storing the information depends on type of signals representing the neuron weights as well as on how the network is realized. In this paper, we focus on networks proposed by Kohonen in [1] that are trained in an unsupervised manner. Apart from the Kohonen networks, the obtained results can also be useful in analyzing, on transistor level, neural networks of other types.

Several IC realizations of the Kohonen networks, called the Self-Organized-Feature-Maps (SOFM) are reported in literature [2, 3, 4]. These implementations involve various techniques, i.e. digital, analog or mixed analog-digital.

An example of a fully digital network has been described in [2]. In this network, neuron weights are stored using a digital memory that is robust against external interferences and other negative effects like noise, leakage effect, charge injection effect, applied CMOS process, voltage and temperature (PVT) variations. Networks that use digital memories can function a long time without the necessity of refreshing the information stored in the memory. However, a drawback of such digital implementations is a large chip area occupation, which is a problem in case of big networks. In the network of [2], the number of transistors in each neuron is as great as 10000.

For this reason, one looks for analog ways of storing the information within a chip. The problem with analog memories is that they are short-term ones. One of possible solutions of this problem are hybrid realizations, where the analog and digital techniques complement each other. An example of such an approach is described in [3]. Some network elements in that approach, like Euclidean distance calculation (EDC) and winner takes all (WTA) blocks, are realized by means of analog circuits. Others, like for instance the circuit weight adaptations and memory storing the weight information are implemented using digital technique. The digital memory is based on counters that can calculate in both directions, depending on whether an input data, x, is greater or smaller than a neuron weight w. The adaptation mechanism in this solution is different from a "typical" Kohonen algorithm. In [3], learning rate is kept fixed and its value results directly from the assumed resolution of the counter (5 bits in this case). Furthermore, the adaptation process does not depend on exact values of the x and w signals but only on values taken by the sign(x,w) function, updating the counter value by ± 1. This solution requires using a digital-to-analog converter (DAC) to transform digital weights to analog signals, needed in the analog part of the network.

Another solution has been described in [4]. In this approach, all main operations, including adaptation and data storing, are performed using analog current-mode circuits. To overcome the leakage problem, some method of refreshing the analog memory is required. A disadvantage of the refreshing concept realized in [4] is that it is carried out sequentially, using a single successive approximation analog-to-digital converter (SAR ADC). As a result, period of the overall refreshment cycle depends on the number of memory cells used in the network. An advantage of this approach, in comparison with the previous solutions, is a much lower chip area occupation.

Authors of this paper have designed an experimental Kohonen network, in which most operations are performed in the analog way. In our realization, no refreshing is needed, resulting in a much lower circuit complexity than in the solutions described above. The proposed network along with experimental results illustrating the learning process effects has been described in details in [5, 6]. Here, we present results concerning an influence of the leakage effect on the weight adaptation process outcome. As will be shown, for sufficiently large sampling frequencies (f_S), the leakage effect does not disrupt the adaptation significantly, though for smaller values of f_S, a new memory solution with longer storage time is required.

Weight adaptation mechanism used in the prototyped self-organized Kohonen neural network

Our neural network operates in 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 current domain. The current mode has been chosen, as it allows for very easy implementation of the summation, as particular signals may be simply added at nodes. This helps avoiding the use of operational amplifiers, which usually dissipate much larger power. This approach significantly simplifies the structure of particular building blocks and of the overall network.

Our network comprises several important components. EDC blocks, described in [5], determine values of the currents, that are linearly proportional to squares of Euclidean distances between a vector of the input learning signals and weight vectors of particular neurons. A binary-tree based WTA block identifies, in the following step, the winning neuron, i.e. the neuron with the smallest distance, searching for the minimum current.

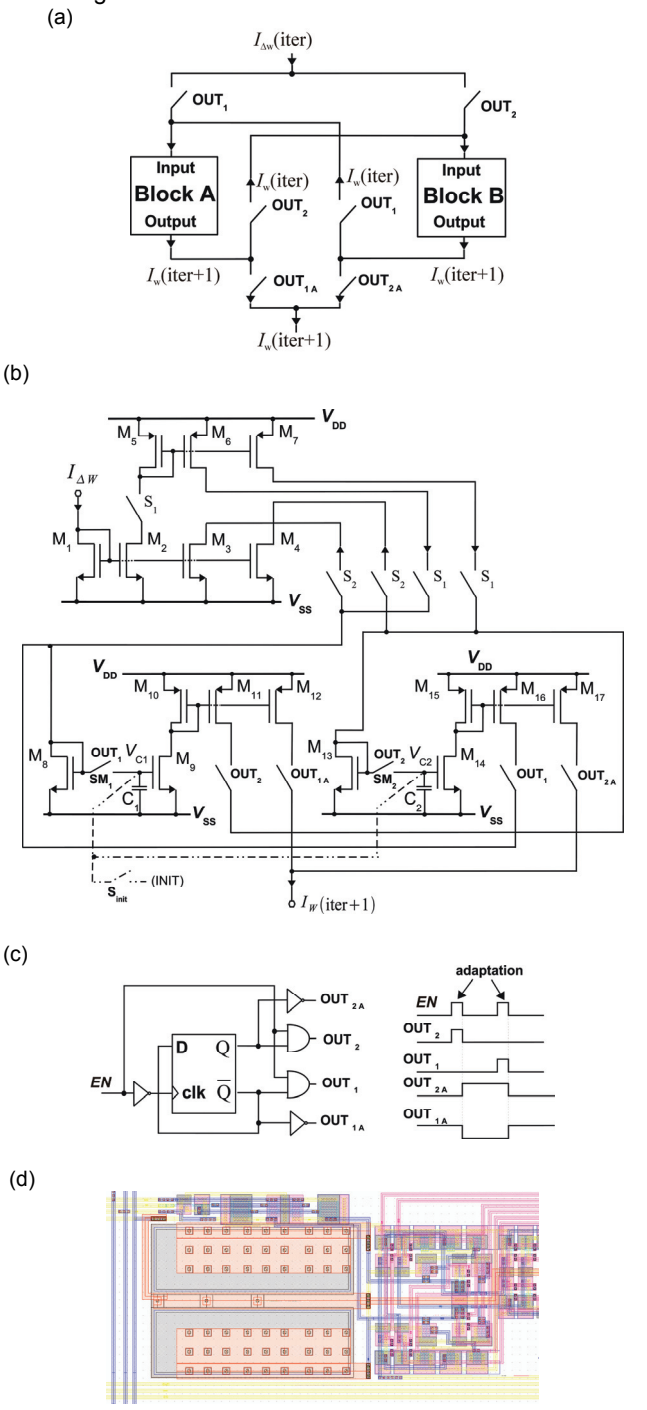


Fig. 1. Electrical scheme of the AWC block used in our analog Kohonen neural network: (a) general block diagram, (b) diagram of the analog part (c) digital control block (d) layout of a single AWC block implemented in CMOS 0.18 µm process (20 x 60µm).

The adaptation weight change mechanism (AWC) proposed in [7], shown in Fig. 1, varies the winning neuron's weights, according to the equation:

$$W_i(iter+1) = W_i(iter) + \eta \cdot [X(iter) - W_i(iter)], \quad (1)$$

where η is a learning rate coefficient.

The number of the AWC blocks equals the number of all weights in the network. A single AWC block consists of two current-mode sample-and-hold (S&H) memory cells, as shown in Fig. 1 (a) and (b), and a specialized control circuit, shown in Fig. 1 (c). Both memory cells operate alternately. If a given neuron becomes a winner, then a current that represents an update of a given weight, $I_{\Delta W}$, is added to the previous value of this weight stored, for example, in the cell A. The result of this adding operation is then stored in the cell B and the output of this cell becomes the output of the AWC block. If this neuron becomes the winner one more time, the role of both cells is turned away and a new weight update, $I_{\Delta W}$, is being added to the previous weight stored in the cell B, while the updated result is stored in the cell A, which now becomes an output cell of the AWC block.

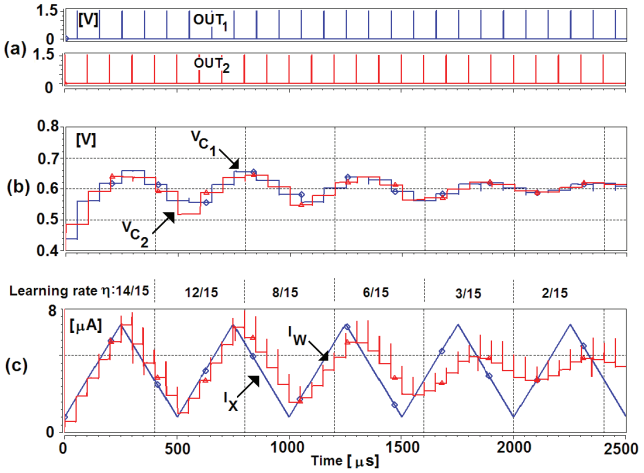


Fig. 2. Postlayout simulations of a single AWC block: a) the output signals of the control circuit, b) voltage V_C on the capacitors in the AWC circuit, c) a testing input current, I_X, and the corresponding output current I_W (the weight) for several values of the learning rate.

Operation of the AWC block is illustrated in Fig. 2 for an triangle input signal, oscillating between 1 and 7 µA. The presented results are postlayout transistor-level simulations in HSPICE environment. Top diagram illustrates the control OUT$_1$ and OUT$_2$ signals that turn away the role of both memory cells thus enabling their alternate operation, as described above. This is shown in the middle diagram, which illustrates the voltages across the capacitors C_1 and C_2 in both cells.

The proposed AWC mechanism is programmable, which means that the learning rate can quickly be reprogrammed using four external bits. The influence of this parameter on the adaptation process is visible in the (b) and (c) diagrams of Fig. 2.

It is worth noticing that for small values of the learning rate, voltages stored on the capacitors C_1 and C_2 vary only moderately, even for relatively large oscillations of the input current. This is an important feature of the proposed adaptation mechanism, since it allows compensating the charge injection effect in a given operation point of M_8, M_9, M_{13}, M_{14} transistors. This compensation is performed by the use of so called dummy switches (not shown in Fig. 1).

In voltage-mode analog memory cells, for instance, such compensation is much more difficult, because the voltage stored on capacitors in such cells can have any value in the range between the V_{DD} and V_{SS} supply voltages, while the compensation with dummy switches is not linear.

Influence of leakage effect on adaptation process in our analog Kohonen neural network

There exists a potential problem with current leakage in the proposed AWC mechanism, but if the weights are updated sufficiently often, it not necessarily has to be severe. This effect distorts the information about weight values, introducing an adaptation error, *e*, to the training process, which for small signals can be expressed as:

$$e \cong K \cdot \frac{n}{f_s \cdot C} \tag{2}$$

Level of this error depends on several factors, such as the capacitance of the memory holding *C* capacitor, its sampling frequency, f_S, and the number, *n*, of neurons used in the networks. *K* is a real-valued coefficient dependent on the leakage current level and temperature.

One of possibilities of minimization the leakage problem is increasing the memory recording (sampling) frequency. This shortens a time interval between two adjacent adaptations of a given neuron, thus minimizing the loss of information between those two events. The adaptation process is not disrupted then. Unfortunately, this increases the number of iterations in a given time period, which raises the amount of power consumed by the network, which is the problem in case of applications that do not require very high sampling frequencies. Fortunately, in many applications, e.g. in telecommunication, the sampling frequency is relatively high and the memory short storing time is not critical in this case.

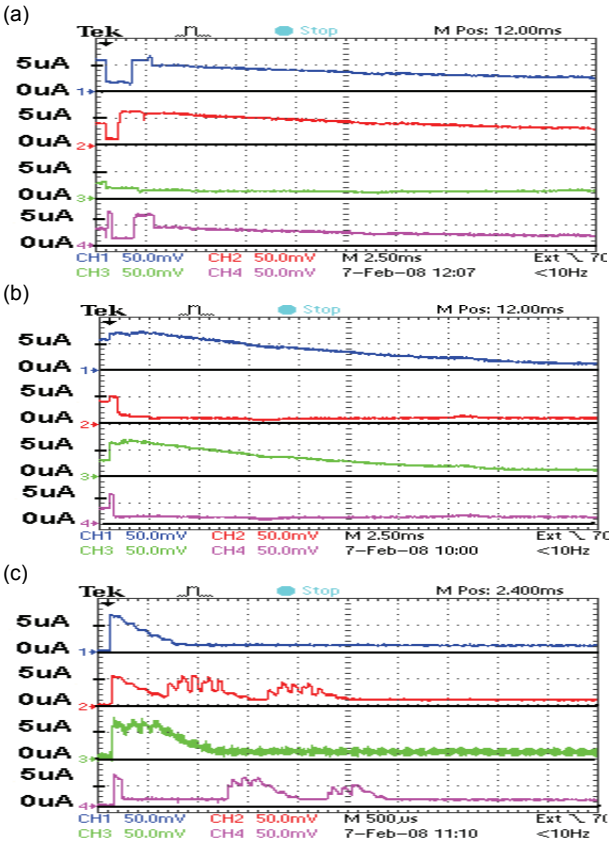


Fig. 3. Leakage effect observed in analog memory cells for: (a) 0°C (b) 20°C (c) 80°C – the experimental measurement results.

Another way of overcoming the short storage time problem relies on increasing the *C* capacitance. Such an approach is attractive in case of networks with low clock frequencies, e.g. in biomedical applications, where sampling frequencies at the level of 1 kHz are sufficient, like in ECG and EMG signal analysis. Using capacitors with larger capacitances increases a charging time constant of the memory cell and time required for updating the weights. This also enlarges the chip sizes, which in small networks is not a great disadvantage.

Considering the *e* error in the context of the entire network, we must be aware that the larger is the number of neurons in the network, the stronger is the *e* error influence on the network training results. This is because particular neurons are adapted less frequently. In case of large networks that operate with smaller sampling frequencies, new circuits are required along with the circuits converting the weights into digital signals. We plan to solve this problem by using a great number of small and low power ADC circuit working in parallel [8]. In this way each neuron or a small group of neurons can have their own ADC. Compared to the network described in [4], this solution will not reduce the sampling frequency.

One of parameters that have a strong influence on the leakage effect is temperature. In general, the larger temperature, the stronger is the leakage effect and its impact on learning results. This is illustrated in Fig. 3 for several temperature values. In the temperature range up to 35°C, our network operates properly for medium frequencies of about 20-100 kHz. For temperatures below 0°C, storage time increases significantly, even by one order of magnitude. For temperatures above 80-90°C, a loss of information about the weight values appears fast. Under such conditions, the network with a small number of neurons still operates properly, when sampling frequencies are in the range of 2-5 MHz.

In this respect, WTA (Winner Takes All) and WTM (Winner Takes Most) learning algorithms differ a lot. In the second case, not only the winning neuron but also the neurons that belong to the winner neighborhood are adapted. As a consequence, in the WTM approach, for a constant number of neurons, particular neurons are adapted more frequently, which better compensates the leakage effect. The WTM networks require a neighborhood mechanism that has been recently proposed in [9].

Network simulations using system level model

To evaluate the leakage effect more precisely, a large number of simulations have been performed on a system level, for different number of neurons and different adaptation process settings. Example results are presented in Fig. 4 for the configuration cases collected in Table 1.

Table 1. Case study for the results shown in Fig. 4.

Variants	B	C	D	E
Conscience mechanism	NOT	NOT	YES	YES
The leakage effect function	YES	NOT	NOT	YES

An example input data set is shown in Fig. 4 (A). It consists of 160 learning patterns clustered in 10 different places of the input data space. The network includes 16 neurons. In some of the demonstrated cases, a conscience mechanism was used [6]. This mechanism eliminates so-called dead neurons, which take part in the competition, thus consuming power, but never win and therefore do not become representatives of any data class. Panels B-E of Fig. 4 illustrate initial and final placements of neurons in a data space. In cases B and E, the impact of the leakage effect on training results is visible. Some neurons lose a part or even the whole information held in the memory.

In the network with the conscience mechanism being switched off (B), dead neurons are affected stronger than the others. These neurons never win the competition and thus their weights are not compensated. The best situation

takes place when the conscience mechanism activates all neurons, which is shown in picture D.

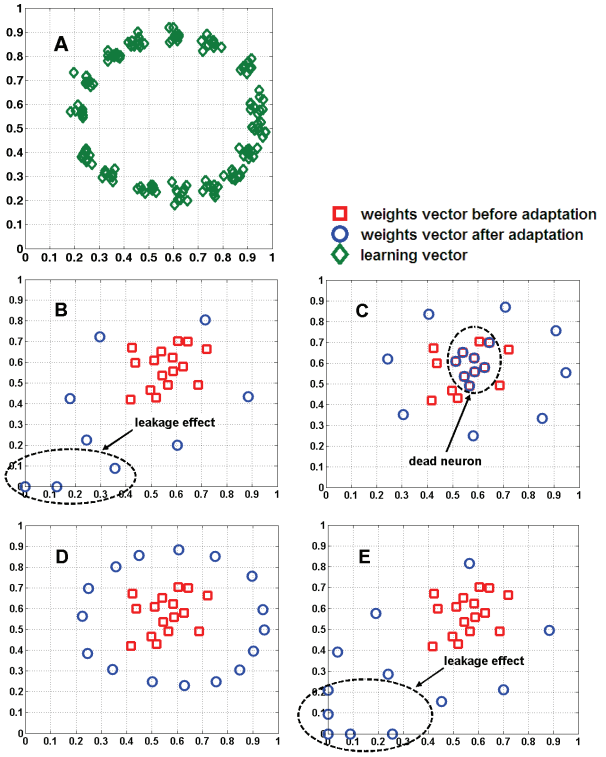


Fig. 4. Simulations of the learning process in the software model of the Kohonen network with 16 neurons, for different configurations described in Table 1: (A) an example training vector.

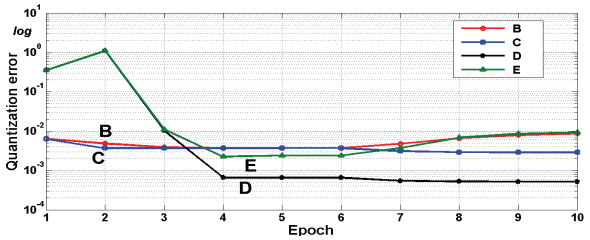


Fig. 5. Quantization error for particular configurations simulated in the software model and shown in Fig. 4.

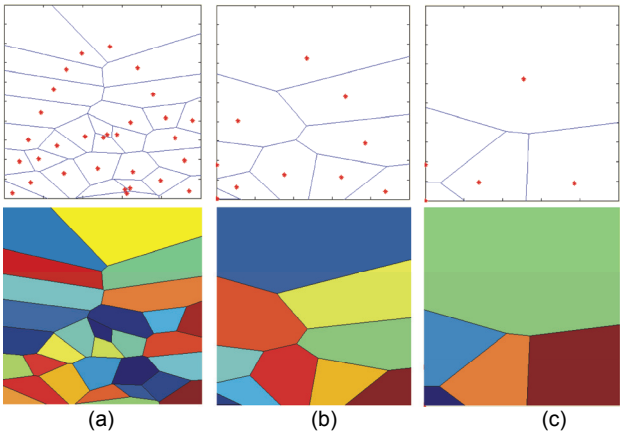


Fig. 6. Voronoi's diagram illustrating the influence of the leakage effect on placement of domination centers of particular neurons for an example case of 30 neurons in the network and different values of the sampling frequency: (a) f_S=2.5MHz (b) f_S=30kHz (c) f_S=2kHz.

One of the commonly used measures of the quality of the learning process is the, so called, quantization error or the adaptation error defined as:

$$Q_E = \frac{1}{z} \cdot \sum_{iter=1}^{z} \left\| X(iter) - W_j(iter) \right\|^2 \tag{3}$$

where Z is the number of iterations in each epoch, while the j index indicates the winning neuron in a given iteration.

The obtained results are shown in Fig. 5. The leakage effect has a strong influence on the quantization error. This is well visible when learning rate becomes small. For cases B and E, it can be observed starting from the epoch 6, where the learning rate becomes 0.1.

The results shown in Figure 6 are for a network with 30 neurons and input data set different than in Figure 4 (A), although particular configuration cases are similar and also are given in Table 1. The visible loss of information has the influence on locations of domination centers of particular neurons, as shown on the Voronoi's diagram in this figure. In the case shown in Fig. 6 (a), all neurons are active. Sampling frequency is here high enough to neglect the influence of the leakage effect. Decreasing the sampling frequency we increase the unwanted influence of leakage effect and reduce the number of classes that can be properly recognized by the network. The worst situation is shown in Fig. 6 (c). In this case, only a small fraction of neurons takes part in the learning process. Shortly after the initialization process, the other neurons become inactive and their weight information is changed in a continuous way due to the leakage effect. For sampling frequencies below 1 kHz, the network does not learn properly.

Influence of temperature on the learning process

To better illustrate the influence of temperature on the learning process, due to the leakage effect, we performed a number of postlayout Hspice simulations for various values of input signals and different temperatures. The results are shown in Fig. 7.

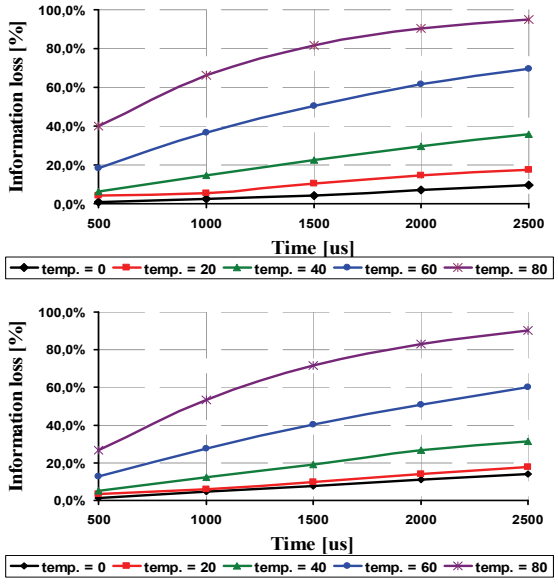


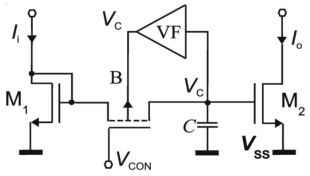
Fig. 7. HSPICE postlayout simulation results illustrating the loss of information held in the memory for different values of input signals and different temperatures. The upper diagram is for the weight currents on the level of 1.25 μA while the lower for currents equal to of 4.25 μA. The diagrams present only a testing phase i.e. the period, in which the adaptation process is turned off.

The obtained results are slightly different (worse) than the measurement ones. The measured loss of information is approximately one half of that observed in simulations. Good learning usually takes place for the error e being in the range between 1 and 3 %. Taking the results shown in

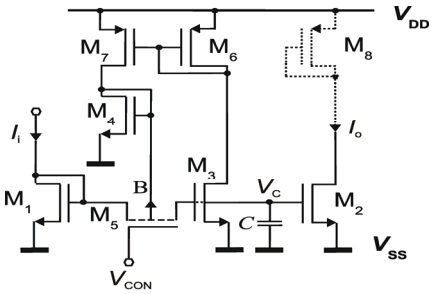
diagrams of Fig. 7, we can calculate a minimum sampling frequency for each case. If, for example, temperature is 40°C. i.e. approximately equal to the body temperature, a loss of information after 1500 μs equals 20%. In other words, a mean loss of 2% is observed over a time period of 150 μs. For an example network with 20 neurons, this leads to the sampling period of about 7.5 μs, which means that the sampling frequency must be at least 135 kHz in this case. Taking into account the measurement results, one can assume a real value of the sampling frequency to be about 70 kHz.

Analog memory with increased holding period

One of possible solutions of the leakage problems is optimizing the memory by increasing its storage time. Such an improved memory cell, shown in Fig. 8, has been proposed and described in details in [10].



(a)



(b)

Fig. 8. The proposed memory with increased storage time: (a) Idea of reducing the main leakage current, (b) detailed diagram

Scheme (a) in Fig. 8 illustrates the idea of increasing the memory storage time, while preserving its short recording one. This is reached due to applying a positive feedback around the main memory switch (M_5) that switches between the memory recording and holding phases. Applying the feedback ensures a zero voltage biasing a *p-n* junction existing in MOS transistors between the channel ends (drain and source) and the transistor well. As a consequence of the zero-voltage biasing, no current flows out to the transistor substrate (leakage current). A necessary condition to obtain a significant extension of the information holding time (by several orders of magnitude compared to the case without any feedback) is that gain of the applied voltage follower, VF, must be equal to one as close as possible.

In the circuit of Fig. 8b, the VF follower is realized by means of only 4 transistors (M_3, M_4, M_6 and M_7). Hence, the whole memory cell is simple, consumes small amounts of both power and chip area and is characterized by a short recording time. Another advantages of the circuit, such as low supply voltage and relatively high recording precision result from its current-mode operation.

Conclusions

The paper discusses a problem of how leakage effect influences weight adaptation process of an experimental analog Kohonen network, prototyped in a 0.18μm CMOS process. The adaptation mechanism used in this network does not require any refreshing mechanism, which allows us to simplify the overall structure of the network. The considered leakage effect causes a short storage time of memories used in the prototyped network. This implies that our network is not suitable for low frequency operations (below 100kHz). In the present form, it is proper for moderate and high sampling frequencies and is attractive in many telecommunication applications [11,12,13].

The authors are currently working on eliminating the leakage effect and make the network to be suitable for a wider range of sampling frequencies.

REFERENCES

[1] Kohonen T., Self-Organizing Maps, *Springer Verlag*, Berlin, 2001

[2] Rajah A., Hani M. K., ASIC design of a Kohonen Neural Network microchip, *IEEE International Conference on Semiconductor Electronics (ICSE)*, 2004, 148-158

[3] Wu Chung-Yu, Kuo Wen-Kai, A new analog implementation of the Kohonen Neural Network, *International Symposium on VLSI Technology, Systems, and Applications (VTSA)*, 1993, 262-266

[4] Macq D., Verleysen M., Jespers P., Legat J. D., Analog implementation of a Kohonen map with on-chip learning, *IEEE Transactions on Neural Networks*, 1993, Vol. 4, Issue 3, 456-461

[5] Długosz R., Talaśka T., Dalecki J., Wojtyna R., Experimental Kohonen Neural Network Implemented in CMOS 0.18 μm Technology, *International Conference Mixed Design of Integrated Circuits and Systems (MIXDES)*, 2008, Poland, 243-248

[6] Długosz R., Talaśka T., Pedrycz W., Wojtyna R., Realization of a Conscience Mechanism in CMOS Implementation of Winner Takes All Neural Networks, accepted for publication in *IEEE Transactions on Neural Networks*, 2009

[7] Talaśka T., Długosz R., Pedrycz W., Adaptive Weight Change Mechanism for Kohonens's Neural Network Implemented in CMOS 0.18 μm Technology, *European Symposium on Artificial Neural Networks (ESANN)*, 2007, Belgium, 151-156

[8] Długosz R., Gaudet V., Iniewski K., Asynchronous Clock Generator for Flexible Ultra Low Power Successive Approximation Analog-to-Digital Converters, *Canadian Conference on Electrical and Computer Engineering (CCECE)*, 2007, Canada

[9] Kolasa M., Długosz R., "Parallel asynchronous neighborhood mechanism for WTM Kohonen network implemented in CMOS technology", *European Symposium on Artificial Neural Networks (ESANN),* 2008, Belgium, 331-336

[10] Wojtyna R., Current-mode analog memory with extended storage time for hardware-implemented neural networks, *Elektronika*, Nr 3/2009, 34-38

[11] Collett M., Pedrycz W., Application of neural networks for routing in telecommunications networks, *IEEE Global Telecommunications Conference*, USA, 1993, Vol. 2. 1001-1006

[12] Soo-Chang Pei, You-Shen Lo, Color image compression and limited display using self-organization Kohonen map, *IEEE Transactions on Circuits and Systems for Video Technology*, 1998, Vol. 8, No. 2, 191-205

[13] Amerijckx C., Verleysen M., Thissen P., Legat J., Image Compression by Self-Organized Kohonen Map, *IEEE Transactions on Neural Networks*, 1998, Vol. 9, No. 3, 503-507

Authors: *dr inż. Rafał Długosz, dr inż. Tomasz Talaśka, dr hab. inż Ryszard Wojtyna (prof. UTP), Faculty of Telecommunication and Electrical Engineering, University of Technology and Life Sciences, ul. Kaliskiego 7, 85-796 Bydgoszcz, E-mail: rafal.dlugosz@epfl.ch, talaska@utp.edu.pl, woj@utp.edu.pl*

dr inż. Rafał Długosz, Institute of Microtechnology, Swiss Federal Institute of Technology in Lausanne (EPFL), A.L.Breguet 2, CH-2000, Neuchatel, Switzerland and Poznań University of Technology, Department of Computer Engineering, ul. Piotrowo 3A, 60-965, Poznań Poland

The correspondence address is: e-mail: talaska@utp.edu.pl