Appl. Math. and Comp. Sci., 1994, vol.4, No.3, 483–498

APPLICATION OF SWITCHED-CAPACITOR CIRCUITS TO REALIZATION OF NEURON-LIKE BINARY HAMMING CLASSIFIER

ZDZISŁAW KORZEC*, TOMASZ KACPRZAK*

In the paper, based on the literature and our own research, a concept of application of switched-capacitor circuits for designing neural networks with constant weights has been presented. The solution is attractive, especially in view of designing simple classifiers and pattern recognition systems implemented in a semiconductor very large scale technology (VLSI).

1. Introduction

One of the important areas of technology where artificial neural networks have been successfully applied is the classification and pattern recognition (Lippman, 1987). The main task of the classification is producing an output signal — sometimes called "decision" — telling us which of M exemplar patterns stored in the system is most similar to the unknown, usually distorted and noise-corrupted input, consisting of N elements (sometimes called features). The classical classifier is composed of two distinctive stages of information processing (Robinson *et al.*, 1992). The inputs of the first classifier (see Fig. 1a) are symbols representing N elements of an unknown pattern (input). These symbols enter the system sequentially. Next they are encoded and the so- called simililarity coefficients to each pattern class are calculated. These coefficients, also called matching scores, are recognized as a measure of similarity between the input pattern and each of the exemplar patterns. The similarity coefficients are encoded and transferred sequentially to the next stage where the coefficient of the largest value is chosen. The classifier has only one output signal whose value corresponds best to the input pattern at the given time moment. The classical systems were usually not equipped with adaptive ability, i.e. the system parameters were constant during operation.

The classifier based on the neural network paradigm has its main part similar to the classical classifier, but its operation is quite different (see Fig. 1b). In the neural classifier N elements of the input pattern are introduced to the first stage in parallel, i.e. the whole set at the same time. Also, all the input and output signals are processed in a parallel mode. The second stage has a separate output for each class. The parameters of the system (synaptic weights) may be constant, established previously while designing the system. However, the system can modify the weights using the output signals corresponding to the actual input. In neural classifiers no assumptions (or at least very weak assumptions) on the probability density of patterns (classes) and inputs are required.

* Technical University of Łódź, Institute of Electronics, ul. Stefanowskiego 18/22,
90-924 Łódź, Poland

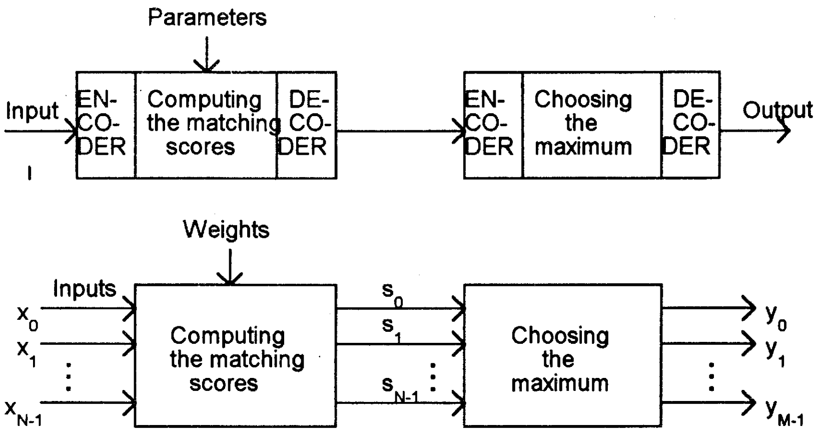

Fig. 1. The classical classifier (a), and the classifier based on neural network paradigm (b).

Neural classifiers can operate in a continous-time mode or binary mode, according to the representation of signals. Also, according to the training method, one can distinguish supervised or unsupervised learning procedures. The main positive aspect of neural network paradigm applied to classification is the ability of parallel processing of information, which is especially important in the case of real-time operation in such technology area like telecommunication, medical and/or technical diagnosis, automatic control, robotics and many others. The neural network paradigm and its application belong to the most important and promissing areas of high-technology solutions for the foreseeable future.

In this paper, based on the achievements discussed in the literature and on our own experience, we present a simple concept of switched-capacitor neural binary classifier circuits for realization in the semiconductor very large scale integration technology. The paper has the following structure. In Section 2 we present an overall concept of a binary Hamming neural network for classification of distorted input codes (input patterns). In Section 3, based on (Cilingiroglu, 1991), we summarize the architecture of capacitance synaptic matrix as a fundamental subsystem for the binary classifiers. A simple Hamming classifier constructed with small-scale integrated circuits, treated as a workhorse in our analysis, is presented in Section 4. Experimental results are summarized in Section 5. In Conclusions we discuss both the advantages and the disadvantages of the concept presented here and we give some recommendations for future work.

2. The Neural Hamming Binnary Classifier

The Hamming neural network,which has the simplest possible architecture from among well known ones (Lippman, 1987), is very well fitted for neural classification. This network makes performing effectively even a very complicated classification possible, using inputs of different code lengths. The classifier converges to the stable points previously stored and can be easily realized in VLSI technology.

The up-to-date VLSI realizations of neural networks can be divided into two groups. The first group comprises the circuits which are based on the charge technique, while the other — those based on current-mode technique. Typical examples of the first group are charge-coupled devices and switched capacitor circuits. As regards the switched capacitor circuit approach, one can point out multi-phase controlled blocks cooperating with microprocessors and digital memory. This technique enables us to mix analog and digital approach in one chip. As a circuit of the current-mode technique operational transconductance amplifiers and voltage-to-current blocks are usually used. These circuits are used mainly in the output part of the classifier.

In this paper we present an example of application of the switched-capacitor approach to realization of the Hamming neural network. The Hamming network (Lippmann, 1987; Robinson *et al.*, 1992) is composed of two subnetworks, as shown in Fig. 2. The exemplar patterns are encoded in the interconnection weights between N inputs and M intermediate nodes at the top of the lower subnet. For the input pattern presented by the vector $x(x_1, x_2, ..., x_N)$ the matching scores are calculated and these values are passed to the upper subnet called "MAXNET". The "MAXNET" selects the exemplar pattern with the maximum matching scores. The matching score (or similarity score) is defined as the number of matching elements between the input code and each of the exemplar patterns (Robinson *et al.*, 1992):

$$S_j(x) = N - \sum_{i=0}^{N-1} |x_i - w_{ij}| = N - h(x\,w), \qquad 0 \le j \le M \tag{1}$$

where x_i is the element of the input code (input vector), w_{ij} – the element of the j-th exemplar vector, and $h(x, w)$ is the Hamming distance for the corresponding exemplar pattern. The Hamming distance, by definition, is a number of non-matching elements. The maximal value of S is $S_{\max} = M$ for a perfect match and the minimal value $S_{\min} = 0$ is when x and w are completely different. In the case of the binary Hamming classifier which operates with $x_j \in \{0, 1\}$ and $w_{ij} \in \{0, 1\}$ the Hamming distance can be expressed by the digital exclusive-or operator:

$$h = \sum_{i=0}^{N-1} x_i \otimes w_{ij}, \qquad 0 \le j \le M \tag{2}$$

and matching scores are

$$S_j = N - \sum_{i=0}^{N-1} x_i \otimes w_{ij}, \qquad 0 \le j \le M \tag{3}$$

One can find various software and hardware implementations of the Hamming network. We shall concentrate mainly on the realization of the switched-capacitor technique.

3. Switched-Capacitor Neural

The neural network architecture can be implemented as an integrated circuit by using analog, digital or mixed analog-digital structures (Korzec, 1994). The advantages of

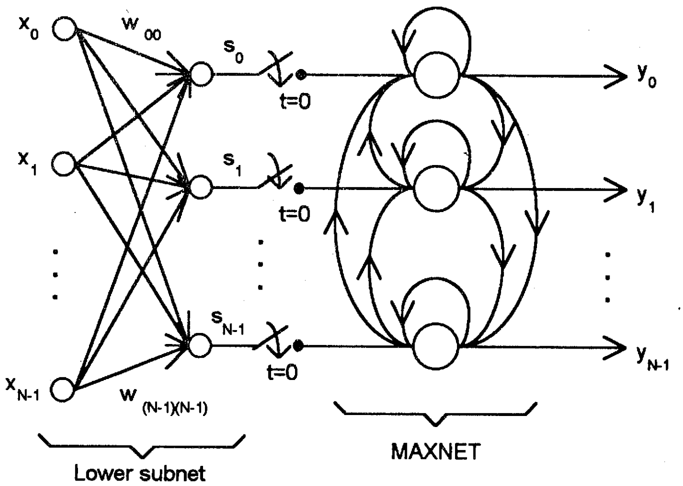

Fig. 2. The Hamming neural network.

an analog implementation are simple basic blocks and communications (interconnections). This in turn results in a small area and thus larger networks on a single chip. However, the storage of analog weight values is difficult and also the noise immunity is worse than in digital circuits. The digital structures are more straightforward to design, their testability is better than in the analog structures and storage of the weight values is easy in the digital form. The interfaces to the digital coprocessors are also simpler. The mixed analog/digital, or switched-capacitor/digital implementations can join together the positive properties of both techniques. The weight storage and input /output interfaces can be digital, but the neuron and synapse structure can be analog or sampled analog circuit (SC circuits).

The discrete-time mode functioning of one neuron cell of an exemplary network shown in Fig. 3 is given in the form:

$$y(nT) = f_p \left[\sum_{j=0}^{N-1} w_j x_j (nT - T) - \Phi \right] \tag{4}$$

where x_j represents the input signal, f_p is non-linear transfer function (e.g. signum or sigmoid), w_j is a synaptic weight, y is the output signal, and Φ is the threshold of neuron excitation. The simplest possible circuit implementation in the switched-capacitor (SC) technique is shown in Fig. 4. This circuit represents an SC realization of summing operation amplifier with the equivalent SC resistors

$$R_{eq} = \frac{1}{f_c C} \tag{5}$$

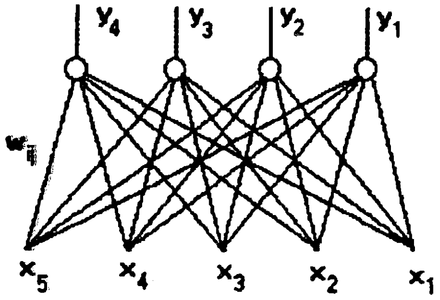

Fig. 3. Basic neural.

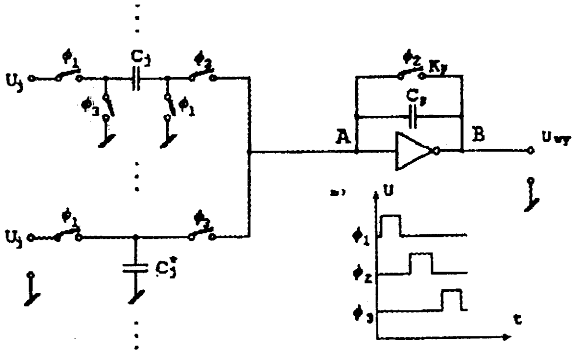

Fig. 4. Switched capacitor implementation of the basic neural network.

The excitatory synaptic weigths are described by

$$w_j = \frac{c_j}{c_F} \tag{6}$$

while the inhibitory synapses

$$w_j = \frac{-C_j^*}{C_F} \tag{7}$$

The inhibitory inputs which are characterized by a negative phase of the signal are usually realized with the equivalent SC circuits presented in Fig. 5.

In this technique it is rather difficult to realize fully connected artificial neural networks of a large size. A much more convenient approach is to use the circuit given in Fig. 6, which is a synaptic capacitor array (Cilingiroglu, 91). Each synaptic connection is represented here by two capacitors C_{ij} and C_{ij}^*.

The bottom plates correspond to the columns which are switched during the successive clock phases to the input voltages u_j or to the reference voltages U_{R1} and U_{R2}. The upper plates of these capacitors correspond to the rows and are connected to the input of a two-stage comparator. The capacitor C_p represents stray capacitance of the rows. Most of the SC network analysis methods involve nodal analysis or modified

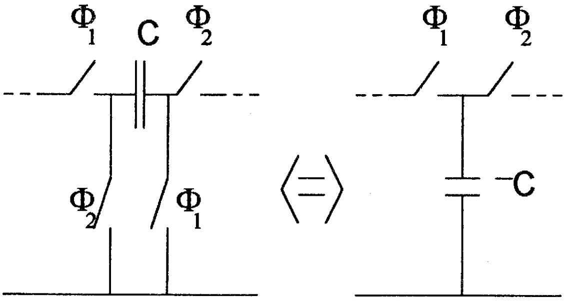

Fig. 5. The non-inverting and inverting parallel SC resistor equivalent circuits.

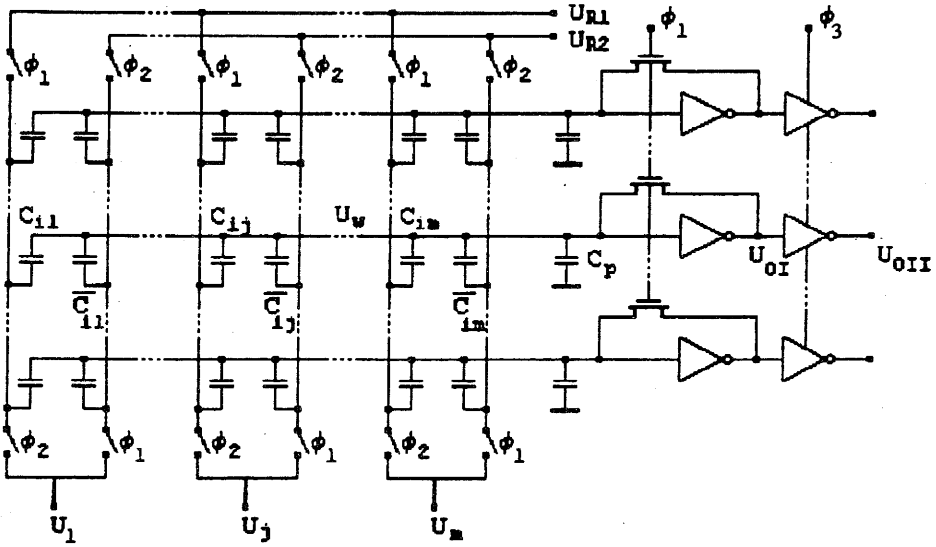

Fig. 6. Synaptic capacitor array for neural-like network realization (Cilingiroglu, 91).

nodal analysis based on the nodal charge equation (Korzec and Ciota, 1987). For one node and for each clock phase, we can write the nodal charge equation in the form

$$q_L(t') = q_M(t) + q_C(t), \qquad t' > t \tag{8}$$

where $q_L(t')$ is the charge left at one particular node at equilibrium, $q_M(t)$ is the charge from the previous phase period, referred to as memory charge, and $q_C(t)$ is the charge injected, designated as the contribution charge.

The circuit in Fig. 6 is controlled by a three-phase clock. During the first phase the charge collected in the raws can be expressed by (Cilingirolu, 1993; Białczak, 1993)

$$Q_i(\Phi_1) = U_T C_P + \sum_{j=1}^{m} \overline{C}_{ij}(U_T - U_j) + \sum_{j=1}^{m} C_{ij}(U_T - U_{R1}) \tag{9}$$

The feedback loop of the first stage inverter is opened at the end of the phase Φ_1. Each row is in its high impedance state holding the collected charge. During the second phase the voltage U_{R2} is connected to the columns possessing capacitors C_{ij} while the input signals are connected to the columns possessing $\overline{C}_{ij}$. Due to the capacitive feedback between columns and rows the voltage U_w changes from U_T to $U_T + U_w$. During the second stage the charge in the rows is

$$Q_i(\Phi_2) = (U_T + U_w)C_P + \sum_{j=1}^{m} \overline{C}_{ij}(U_T + U_w - U_{R2}) + \sum_{j=1}^{m} C_{ij}(U_T + U_w - U_j) \quad (10)$$

Equating formulae (9) and (10) yields

$$U_w = \frac{\sum_{j=1}^{m}(C_{ij} - \overline{C}_{ij})U_j - \left[U_{R1}\sum_{j=1}^{m} C_{ij} - U_{R2}\sum_{j=1}^{m}\overline{C}_{ij}\right]}{C_P + \sum_{j=1}^{m}(C_{ij} + \overline{C}_{ij})} \quad (11)$$

The output voltage U_{0I} of the first stage of the comparator, which is dependent on U_w, is inverted and gained in the second stage of the inverter playing the role of the comparator with enabling input. During the third phase the signal is written at the output and is left there up to the third phase. The output voltage U_{0II} referenced to the power supply U_{DD} can be described by the formula

$$\frac{U_{0II}}{U_{DD}} = f_p\left\{\sum_{j=1}^{m}(C_{ij} - \overline{C}_{ij})U_j - \left[U_{R1}\sum_{j=1}^{m} C_{ij} - U_{R2}\sum_{j=1}^{m}\overline{C}_{ij}\right]\right\} \quad (12)$$

where f_p is an approximated function of the comparator (similar to the unit step function).

4. Design Considerations

When we compare eqn. (12) with eqn. (4), which describes the functional operation of the artificial neuron, we get the following relationships

$$y_i = \frac{U_{0II}}{U_{DD}} \quad (13)$$

$$x_j = \frac{U_j}{U_{DD}} \quad (14)$$

$$w_{ij} = \frac{C_{ij} - \overline{C}_{ij}}{K} \quad (15)$$

$$\Phi_i = \frac{U_{R1}}{U_{DD}}\sum_{j=1}^{m}\frac{C_{ij}}{K} - \frac{U_{R2}}{U_{DD}}\sum_{j=1}^{m}\frac{\overline{C}_{ij}}{K} \quad (16)$$

where K is a positive constant used for scaling the synaptic capacitances with respect to the weigths.

Since all the voltages, excluding only the voltage U_w, are referenced to the ground having the lowest potential in the circuit, the values of excitations x_i and y_i will have positive values in the range from 0 to 1. Synaptic weights are represented by the difference between capacitances C_{ij} and $\overline{C}_{ij}$, and therefore their values can be positive and negative. Using the capacitors C_{ij} for excitatory inputs $w_{ij} > 0$ or the capacitors $\overline{C}_{ij}$ for inhibitory inputs $w_{ij} < 0$, one can reduce by two the number of the capacitors in the circuit. However, during the VLSI fabrication of the capacitor matrix still some stray capacitance $C_{\min}$ is present as a result of the overlapping effect of rows and columns. Taking this capacitance into account one can assume for excitatory inputs $\overline{C}_{ij} = C_{\min}$

$$C_{ij} = Kw_{ij(+)} + C_{\min} \tag{17}$$

where $w_{ij(+)} > 0$, and for the inhibitory inputs $C_{ij} = C_{\min}$

$$\overline{C}_{ij} = -Kw_{ij(-)} + C_{\min} \tag{18}$$

where $w_{ij(-)} < 0$.

Given $C_{\min}$ and K one can determine the synaptic capacitances for the respective weights w_{ij}. The minimal value of the capacitance is easily determined when the area of the overlapping of rows with columns is known. Scale factor K depends on such parameters like values of the weights and the number of synapses of each neuron.

It is seen from formula (16) that the neural network described here is "self-thresholded". For given synaptic capacitances the values of these "own" thresholds are

$$\Phi_i = \frac{U_{R1}}{U_{DD}} \sum_{j=1}^{m} w_{ij(+)} + \frac{U_{R2}}{U_{DD}} \sum_{j=1}^{m} w_{ij(-)} + m\frac{C_{\min}}{K}\,\frac{U_{R1} - U_{R2}}{U_{DD}} \tag{19}$$

This equation has two independent parameters: U_{R1} and U_{R2}. Changing their values one can establish the value of the threshold for all the neurons. The "self-thresholding" feature has this dark side that thresholds cannot be set up individually for each neuron. To avoid this drawback, a good way is to eliminate the common threshold by nullifying reference voltages and to introduce the separate input (two more columns) with unit excitatory value and with the weights of value Φ_i for each neuron. In practice, at least one layer with "self-thresholding" is used while the weights in the other layers are established individually.

The reference voltages U_{R1} and U_{R2} in eqn. (19) can be used for changing the value of the threshold. Treating these voltages as additional signals:

$$s_1 = \frac{U_{R1}}{U_{DD}} \tag{20}$$

$$s_2 = \frac{U_{R1}}{U_{DD}} \tag{21}$$

eqn. (19) can be written in the form:

$$\Phi = s_1 l + s_2(l - m) + A(s_1 - s_2) \tag{22}$$

where m is the number of input words, l is the number of "ones" in the input code, and $A = mC_{\min}/k$. The difference between the thresholds for different values of l, namely $l = l_a$ and $l = l_b$, yields:

$$\Phi(l_b) - \Phi(l_a) = (s_1 - s_2)(l_b - l_a) \tag{23}$$

Equation (22) can be transformed to the form

$$s_2 = \frac{A+l}{A+m-l}s_2 - \frac{\Phi}{A+m-l} \tag{24}$$

If one assumes that the values of l and Φ are treated as parameters, then one can obtain a set of straight lines which divide the plane $s_2 - s_1$ into two parts: for the points lying below the line the excitation of the network is transferred, i.e. the input is recognized as a member of the class suitable for the given neuron, while for the points lying above the line the excitation disappears which means that the neuron does not recognize the input. For different values of the threshold and for constant number of "ones" present in the pattern code these lines are parallel while for different numbers of "ones" and constant threshold these lines intersect at one point. One can determine the coordinates s_1^* and s_2^* of this intersection assuming that $s_1 + s_2 = 1$:

$$s_1^* = \frac{A+m-h}{2A+m} \tag{25}$$

$$s_2^* = \frac{A+h}{2A+m} \tag{26}$$

Taking into account that the value of the threshold Φ can be withim the range from the lower value Φ_d to the upper one Φ_g for which the network recognizes input codes and that the patterns codes can have different numbers of "ones", one can determine the area of correct operation of the classifier (see Fig. 7).

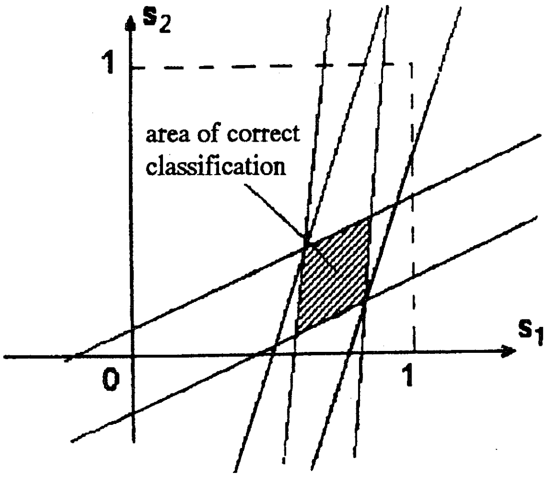

Fig. 7 Graphical determination of the correct operation area of the classifier.

The neural network with capacitance synaptic matrix (array) makes variations of synaptic weights possible through using different matrices. In a general case, the topology of the network is determined by connections existing outside the capacitance matrix. However, if one realizes the connections of each neuron with the others and with itself, then the topology of the network can be varied by exchanging the capacitance matrix only (non-existing connections have zero weight values).

5. Practical Considerations

In order to face the main problems which accompany the practical realization of neural networks in the SC technique, a binary Hamming classifier with discrete small-scale integration digital elements has been designed and tested. The classifier has four neurons with five common inputs (see Fig. 3). This classifier performs calculation of the Hamming distance between the input signal and the previously stored patterns. As a result of this calculation, the class which has a minimal Hamming distance is the winner. The network realized here is a variant of the Hamming neural networks where only these input codes are classified for which the Hamming distance is not greater than some value h. This enables us to classify the codes with errors present on h positions or not to classify them if this distance is greater than h for all the patterns. For the network of Fig. 3, five-bit input signal makes it possible to distinguish four separate classes of codes with the minimal Hamming distance equal to three. This enables us to classify inputs with the error present at one position. An output signal found at one output means that the respective input signal has been classified as a member of the set corresponding to this neuron. Each neuron has its pattern class stored in its synaptic weights, namely weight $w_{ij} > 0$ for $x_j(i) = 1$ and $w_{ij} < 0$ for $x_j(i) = 0$, where i is the number of patterns and j is the number of positions in the input code. This network makes the classification of the input codes with no more than one error. Changing the threshold one can establish the number of errors recognized by the given neuron. The idea of this classification is shown in Fig. 8.

The neural network is controlled by the three-phase clock presented in Fig. 9. This clock is composed of an oscillator, rewritting subcircuit, counter and logic. The main part of the neural network is also realized with discrete elements. It is composed of a capacitance matrix, switches, two-stage comparators, potentiometers and as block used for signalling the decision.

The set of switches PA...PE is used for determination of the value of input signal (binary code). Connecting the power supply to the column of capacitors by means of the given switch coresponds to "logic one" while connecting to ground coresponds to "logic zero". The group of analog switches (unit 4066) is used in the comparator circuit and for switching the capacitors. The capacitance matrix is built of the discrete capacitors of values $C_{\max} = 75\,\mathrm{pF}$ and $C_{\min} = 10\,\mathrm{pF}$. Capacitors $C_{ij} = C_{\max}$ and $\overline{C}_{ij} = C_{\min}$ make the synaptic values $w_{ij} > 0$ in the case of excitatory signals while the capacitors $C_{ij} = C_{\min}$ and $\overline{C}_{ij} = C_{\max}$ make the synaptic values $w_{ij} < 0$ in the case of inhibitory signals.

The state of the first stage of comparator is balanced during the first phase with the value of logic threshold voltage U_T. The second stage of the comparator built of the flip-flop D440449 is controlled by the signal of the third phase. This signal rewrites

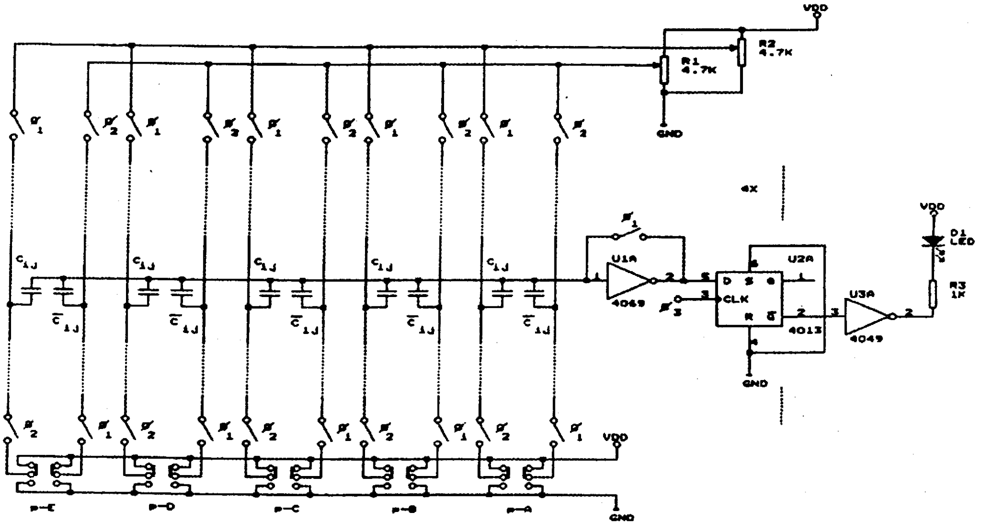

Fig. 8. Hamming classifier realization using discrete small-scale integration elements.

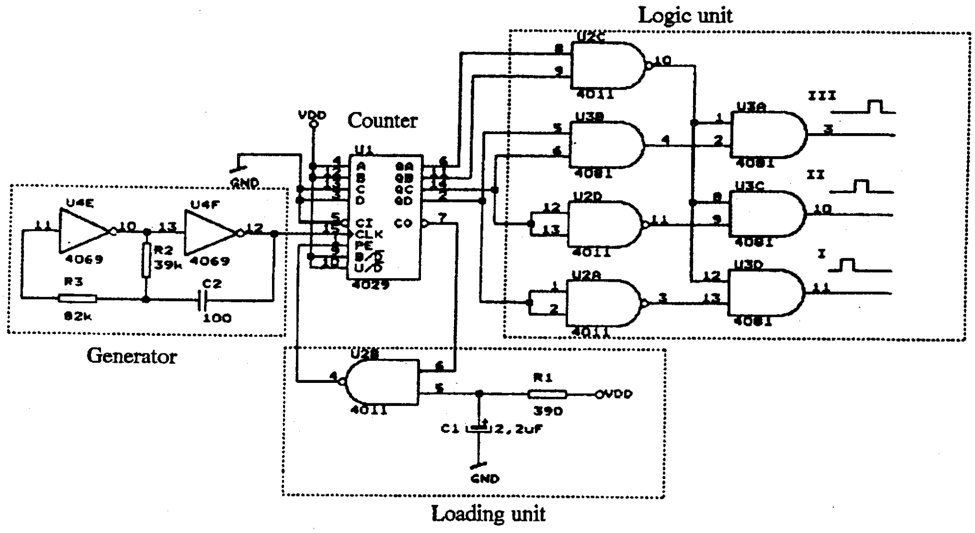

Fig. 9. Three-phase clocking circuitery.

inverted results at the output. The output block has light diodes with resistors and power inverters of type 4049. The resistors limit the currents flowing through diodes. The power inverter works as a buffer with high fan-out factor. The high voltage level at the input turns the light diode on, which signals the excitation present at the output of the respective neuron.

Given the capacitor values of the matrix one can determine the design constants K and A from the formulas

$$K\frac{C_{\max} - C_{\min}}{w_{ij}} = 65\,\text{pF} \qquad A = m\frac{C_{\min}}{K} \approx 0.77$$

Using the values K and A one can calculate the slope of the set of straight lines described by the formula (24):

$$\frac{A+l}{A+m-l} = \begin{cases} 0.73 & \text{for } l=2 \\ 1.36 & \text{for } l=3 \end{cases}$$

The operating point for correct work of the neural network must be chosen from the central part, i.e. $h = 1.5$ (see eqns. (25) and (26)). Hence,

$$s^*_{1\text{sr}} = \frac{A+m-h}{1a+m} \approx 0.65, \qquad s^*_{2\text{sr}} = \frac{A+h}{1a+m} \approx 0.35$$

In order to verify whether the network can recognize input codes it is necessary to show that the minimal required change of charge ΔQ in the circuit for different input codes is greater than the change of the circuit error charge ΔQ_B. The change ΔQ can be calculated from the formula (see also eqn. (11))

$$\Delta Q = \frac{K|w_{ij}|_{\max} U_{DD}}{k_w k_r} \tag{27}$$

If we set $k_w = k_r = 1$, then we obtain

$$\Delta Q = (C_{\max} - C_{\min})U_{DD} \tag{28}$$

For values of capacitances $C_{\max} = 75\,\text{pF}$ and $C_{\min} = 0.13C_{\max} = 10\,\text{pF}$ we have

$$\Delta Q = 0.87 C_{\max} U_{DD}$$

The biggest dominant error for the circuit described above results from the imperfection of capacitors. During the second phase the voltage of rows of value u_w changes. This voltage multiplied by the capacitance of the rows gives the change of the charge Q_d, which is the difference between the charges present in the rows when the voltage $u_w = 0$ is applied (during the second phase). From formula (11) we obtain

$$Q_d = \sum_{j=1}^{5}(C_{ij} - \overline{C}_{ij})U_j - \left(U_{R1}\sum_{j=1}^{5} C_{ij} - U_{R2}\sum_{j=1}^{5}\overline{C}_{ij} \right) \tag{29}$$

Taking into account the worst case for which at all inputs we have the power supply $U_j = U_{DD}$ and $U_{R1} = U_{R2} = 0.5U_{DD}$ one can calculate the change in charge caused by the tolerance of capacitors, namely

$$\Delta Q_b = U_{DD}\frac{\sigma(C)}{C}\sqrt{\sum_{j=1}^{5}(C_{ij}-\overline{C}_{ij})} = U_{DD}\frac{\sigma(C)}{C}\sqrt{m(C_{\max}^2-C_{\min}^2)}$$

$$\approx U_{DD}\frac{\sigma(C)}{C}\sqrt{m}C_{\max} \tag{30}$$

In the system realized here the capacitors with 5% tolerance were used, hence

$$\Delta Q_b = 0.11 C_{\max} U_{DD} \tag{31}$$

From the above calculations one can conclude that the smallest change of charge required in the circuit is several times greater than the value ΔQ_b, therefore the network should properly classify the input codes.

Another crucial parameter for proper functioning of the network is the time required for recharging the capacitors. During each phase of the clock the respective capacitors should be recharged completely. The elements which limit the current in the circuits are potentiometers P1 and P2. They can be regarded as voltage sources with varied output resistance. Each potentiometer is connected to the half of the total capacitance of the matrix during the first phase as well as during the second phase of the clock. For the worst case, i.e. when the values of the potentiometers are in the mid of their ranges, the time constant of charging the capacitor is given by

$$\tau = 0.5 R_p\, 0.5 \sum_{j=1}^{5}(C_{ij}-\overline{C}_{ij}) \approx 2\,\mu\text{s} \tag{32}$$

As the time duration of one phase is $T_F = 28\,\mu$s, therefore with time constant $\tau = 2\,\mu$s, the capacitors will be completely recharged. However, when we increase the total capacitance of the matrix (by an increase of the number of inputs) the voltage sources with reduced output resistance and/or lowering the clock frequency should be used.

6. Experimental Results

In order to test the classifier, the area of correct performance on the plane $s_1 - s_2$ was first determined based on measurements. Next, the network was tested under different values of the reference voltages. During the test the input words with h errors were presented at the input and the reference voltage at the fired output has been noted. This test provides graphical representation of the set of lines which determine the limits of excitations.

Figure 10 shows two exemplary families of these characteristics – one for the neuron with the pattern 01011 (for $l = 3$), the other one for the neuron with the pattern 00110 (for $l = 2$) and different number of errors (h is varied from 0 to 5). Each line divides the plane $U_{R1} - U_{R2}$ into two areas: the lower area transferring the excitation, and the upper one where the input signals are blocked, i.e. not transferred to the output. The lines of this two families are parallel and their slopes are 0.71 for $l = 2$ and 1.34 for $l = 3$, respectively. These results are very close to these calculated from the theoretical formulae (0.73 and 1.36, respectively).

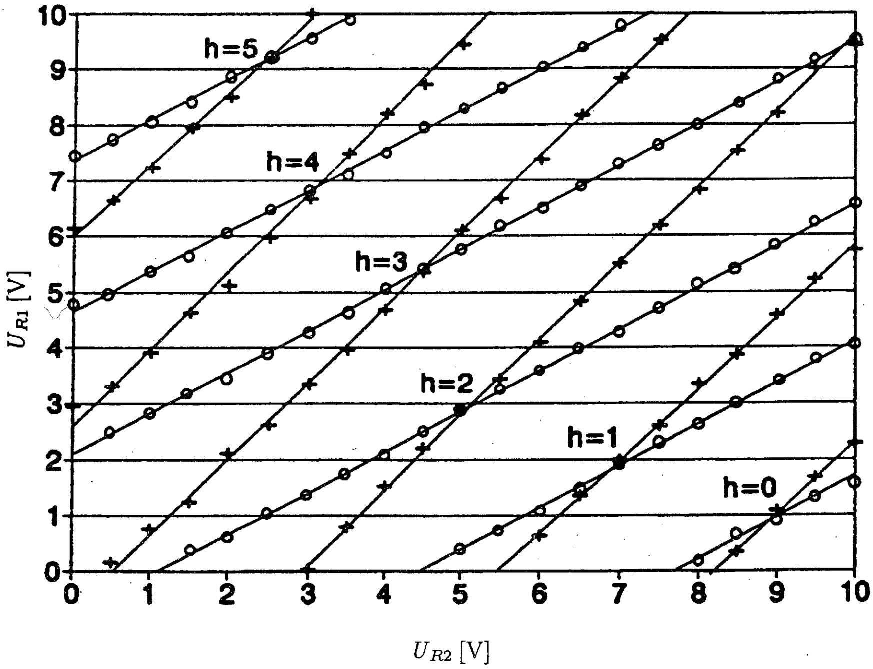

Fig. 10. Graphical representation of the set of lines which determine the limits of excitations.

The network realized should recognize codes with at least one erroneous position. In order to face the problem of sensitivity of the network to the tolerance of the circuit parameters, measurements of input codes with one and two errors under the greatest displacement of voltages have been taken. The results are given in Fig. 11, where the margins resulting from the circuit tolerance can be observed. In the case when the margins overlap one another, the area of the network correct performance vanishes. Accordingly, the neural network cannot recognize the input codes. If the reference voltages U_{R1} and U_{R2} take values from the dashed area, the code is recognized with at least one error. Decreasing the threshold of each neuron one gets the network which recognizes input codes with more than one error. If the threshold is lowered to the level the network is able to recognize codes with more than two errors (when five-bit input is assumed), then some input codes can be properly classified by more than two neurons simultaneously.

For the area of the correct performance shown in Fig. 11, one can determine its centre point P_{sr}. The coordinates of this point are $U_{R1} = 6.9\,\text{V}$ and $U_{R2} = 3.1\,\text{V}$, which agree with the values calculated theoretically, namely: $s_1^* U_{DD} = 6.5\,\text{V}$ and $s_2^* U_{DD} = 3.5\,\text{V}$, respectively. The response of the neural network for this centre point has been tested. The results obtained are shown in Fig. 12, as three-dimensional excitations transferred to the outputs. Codes properly recognized are denoted in black colour, while the white colour represents codes with one error only. Eight codes have not been classified to any class because their distance from the patterns is greater

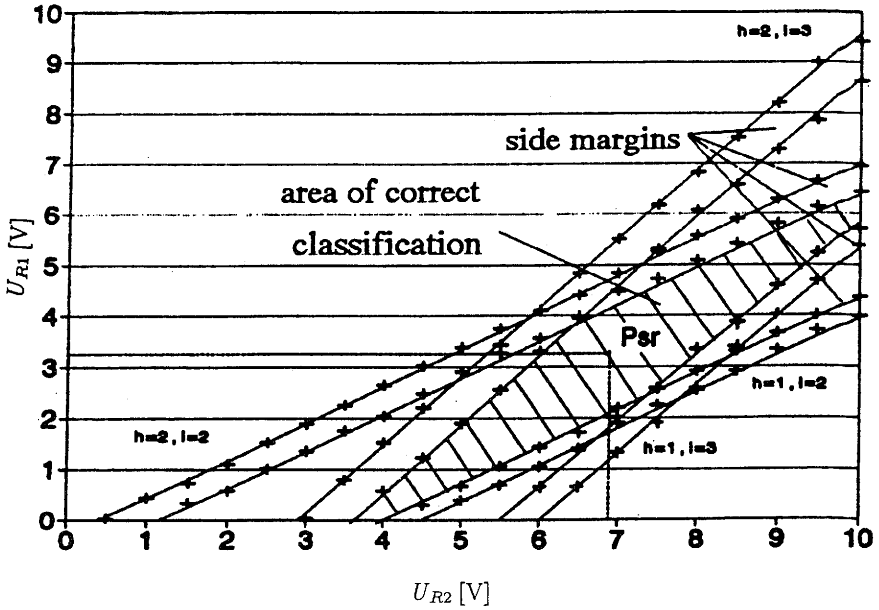

Fig. 11. Area of correct performance with the central point Psr.

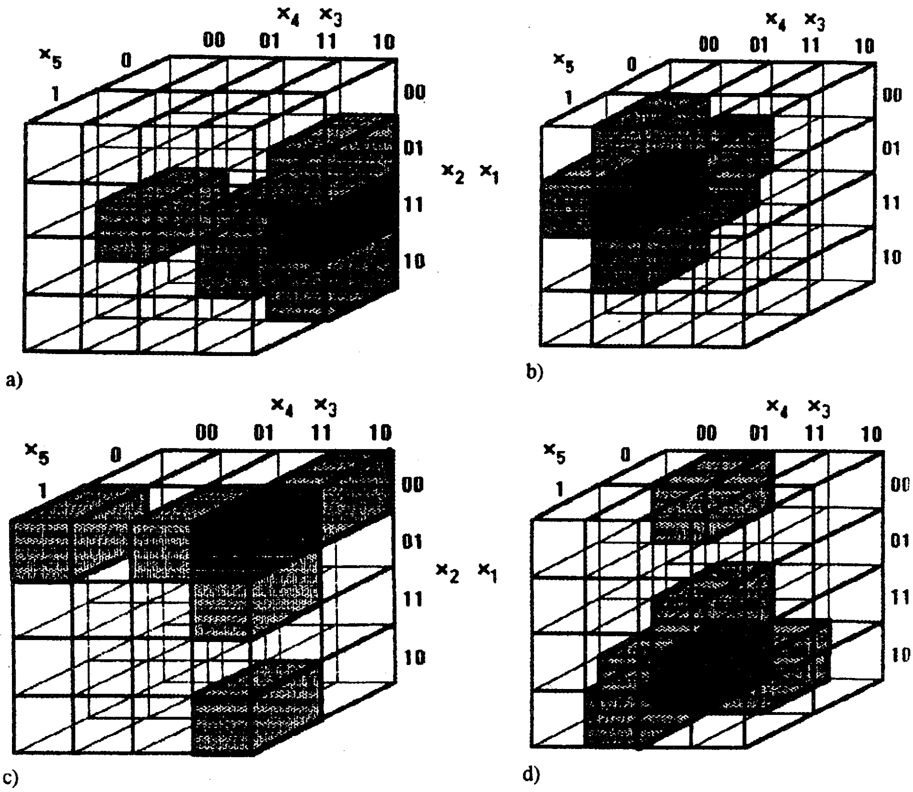

Fig. 12. Three-dimensional representation of the classifier responses for the pattern codes: a) 01011 b) 10101 c) 11000 d) 00110

than one. Each neuron does classify code stored in its weights and five codes with one error, i.e. it recognizes totally eight codes. Four neurons recognize 24 codes, which gives with 8 codes not classified to any class, the total set of 32 combinations of five-bit inputs.

7. Conclusions

Switched-capacitor circuits, beeing mixed analog/digital circuits, found many interesting implementations among which neuron-like systems are one of the most interesting. The SC technique based on the modern CMOS technology meets very well the requirements of the VLSI implementation, giving the possibility to realize neuron-like systems with millions of neurons placed on one chip. The Hamming classifier is one of the interesting examples of the SC technique application in neuron systems. The designing methodology presented in this paper and the measured results form the basis for further developement of the Hamming classifiers. Future work should be concentrated on the VLSI implementation with possible on-chip learning.

References

Białczak A. (1993): *Realization of Neural Cells in the SC Technique.* — M.Sc. Thesis, Technical University of Łódź.

Choi J. and Sheu B.J. (1993): *A high-precision VLSI winner-takes-all circuit for self-organizing neural networks.* — IEEE J. of SSC, v.28, No.5, pp.576–584.

Cilingiroglu U. (1991): *A purely capacitive synaptic matrix for fixed-weight neural networks.* — IEEE Trans. CAS, v.38, No.2, pp.210–217;.

Horio Y. *et al.* (19): *Speech recognition network with SC neuron-like components.* — Proc. ISCAS'88, pp.495–498.

Korzec Z. (1994): *Mixed switched-capacitor/digital networks.* — Conf. *Education of CAD of Modern VLSI Circuits*, Dębe, (Poland).

Korzec Z. and Ciota Z. (1987): *Switched-Capacitor Circuits.* — Warszawa: WNT, (in Polish).

Lippman R.P. (1987): *An Introduction to Computing with Neural Nets.* — IEEE ASSR Magazine, pp.4–22.

Robinson M.E., Yoneda H. and Sanchez-Sinencio E. (1992): *A Modular CMOS Design of a Hamming network.* — IEEE Trans. Neural Networks, v.3, No.3, pp.444-555.

Rodriguez-Vazquez A. *et al.* (1990): *Non-linear SC "neural" networks for optimization problems.* — IEEE Trans. CAS, v.37, pp.384-397.