Analog Integrated Circuits and Signal Processing 1, 339-351 (1991)

Synthesis of Artificial Neural Networks by Transconductors Only

MEHMET ALI TAN

Bilkent University, Department of Electrical and Electronics Engineering, Bilkent 06533, Ankara, Turkey

Received January 25, 1991; Revised May 17, 1991.

Abstract. Hardware implementation of artificial neural networks has been attracting great attention recently. In this work, the analog VLSI implementation of artificial neural networks by using only transconductors is presented. The signal flow graph approach is used in synthesis. The neural flow graph is defined. Synthesis of various neural network configurations by means of neural flow graph is described. The approach presented in this work is technology independent. This approach can be applied to new neural network topologies to be proposed or used with transconductors designed in future technologies.

1. Introduction

Recently, there has been a great motivation among the researchers from a wide variety of fields to implement neural computers in various forms: optical [1], digital [2] and analog integrated circuits [3]. These activities are stimulated obviously by the attractive prospects of the neural networks in overcoming the inadequacies of traditional digital computers on sophisticated tasks involving intelligent actions such as pattern matching, perception, recognition, and image processing applications. In addition to these capabilities, the artificial neural networks are healthy, fault-tolerant, and need no programming.

Many applications of neural computers need on-site real-time operation [4]-[6]. This requires implementation by fast, small hardware. One of the most appropriate choices is the analog integrated form. By means of the advances in integrated circuit technology, the analog integrated circuit implementation of artificial neural networks seems to be feasible nowadays [7], [3]. Furthermore, the excellent properties of integrated circuits, such as good matching of like components, make this approach even more advantageous. An extensive work has been recently presented regarding the analog implementation of neural networks [7], [3], [8], [5], [9]-[17].

This work proposes the realization of artificial neural networks using only transconductors. Transconductors can be chosen as basic building blocks for analog integrated circuits. They are simple and readily available in various technologies [18]-[21]. They can be simpler than op amp because the transconductor is a subcircuit in an op amp.

One of the major advantages of the approach presented in this work is that it is technology-independent. Therefore, the transconductance elements to be proposed in any future technology can be used by this approach. Another advantage is that the design of any artificial neural network is reduced to the design of a simple transconductor. The simpler the transconductor used the simpler the overall neural computer to be obtained. The transconductors may be chosen tunable or fixed. Tunable transconductance elements makes the neural computer more flexible and versatile. If the transconductors are chosen fixed, the parameter assignment may be achieved by a computer simulation.

In Section 2, the transconductor is reviewed. The use of transconductor as building block in artificial neural networks is presented in Section 3. Section 4 discusses the implementation of various neural network configurations. The CMOS implementation and its SPICE3d2 simulation results of a Hopfield-type network realization of a 4-bit analog-to-digital converter are presented in Section 5.

2. The Transconductor

The circuit-theoretical model of a transconductor is a voltage controlled current source with a possible output conductance and input capacitance as shown in figure 1. It can be easily verified that a transconductor with the negative transconductance (or the output current inverted) can be implemented by the triple transconductors shown in figure 2b.

Several transconductor elements have been proposed for various technologies [18]-[21]. Beyond these

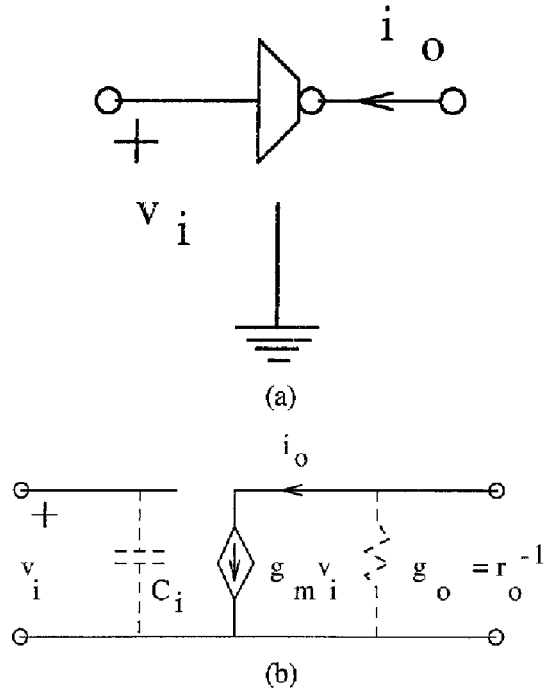


Fig. 1. (a) Symbol of transconductor; (b) its circuit-theoretic model.

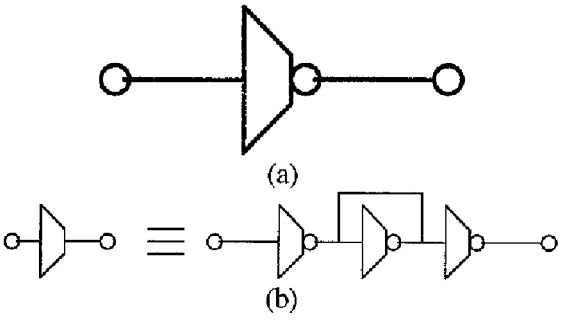


Fig. 2. (a) Positive and (b) negative transconductors.

transconductors, a CMOS logic inverter and an operational transconductance amplifier [22] may be used as transconductance elements. Obviously, the performance and properties of these alternatives, such as offset current, linearity, power consumption, and number of transistors, affect the choice of transconductors.

The parasitic components can be thought as drawbacks at first sight. However, the parasitic capacitance constitutes the dynamical nature of the neural network to be implemented, and the output conductance and offset current can be easily absorbed by an appropriate arrangement as discussed in Section 3.

As an example the transconductance element proposed by Park and Schaumann [19] is shown in figure 3. The output current i_o of this transconductor is

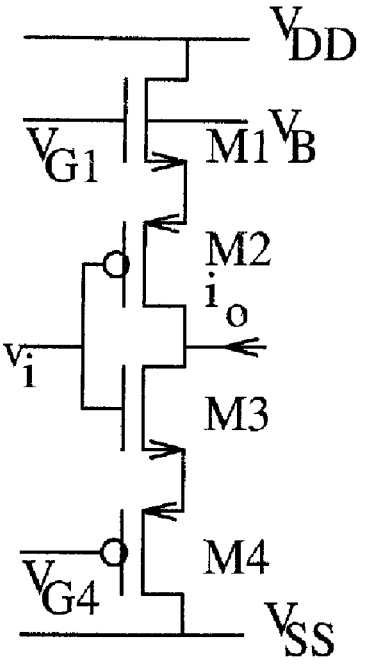


Fig. 3. A CMOS transconductance element proposed by Park and Schaumann [19].

$$i_o = 2k_{\text{eff}}[V_{G1} - V_{G4} - \sum_{i=1}^{4} |V_{Ti}|]\, v_i \quad (1)$$

where

$$k_{\text{eff}} \triangleq \frac{k_n k_p}{(\sqrt{k_n} + \sqrt{k_p})^2} \quad (2)$$

$$k_{n,p} \triangleq \frac{1}{2}\,[\mu_{\text{eff}} C_{\text{ox}} \frac{W}{L}]_{n,p} \quad (3)$$

where V_{Ti}'s are the threshold voltages of corresponding transistors and k_n, k_p, $\mu_{n,\text{eff}}$, $\mu_{p,\text{eff}}$, C_{ox}, W, and L have their usual meanings. This transconductor is quite linear, tunable in a wide range, and the output offset current can be zeroed. It can be easily shown that all transistors are on as long as

$$V_{G1} - V_i \geq V_{T1} + |V_{T2}| \quad (4)$$

$$V_i - V_{G4} \geq V_{T3} + |V_{T2}| \quad (5)$$

And $M1$ and $M4$ are always in saturation as long as they are on, and $M2$ is in saturation if and only if

$$V_o \leq V_i + |V_{T2}| \quad (6)$$

and $M3$ is in saturation if and only if

$$V_o \geq V_i - V_{T3} \tag{7}$$

where V_o is the output voltage. When it is loaded by another identical transconductor whose input and output are connected, the input-output characteristic becomes as shown in figure 4. The transfer characteristic of this unity-gain voltage buffer is obtained by a SPICE simulation with realistic MOS transistors and MOS models and with the transconductor shown in figure 3.

3. Artificial Neuron by Transconductors

An artificial neuron can be considered as multiple-input single-output signal processing element as shown in figure 5 [23]. The input signal is provided either from the input of the entire network or from the output of another neuron. The function of a single neuron with n inputs can be expressed as where y is the output signal, x_i is the ith input signal, w_i is the ith synapse weight, and θ is the threshold and can be considered the weight of a synapse connecting a constant input with a value of -1, as shown also in figure 5, and $f(\cdot)$ is a monotone function, which may be sigmoidal or step type [23] or linear in transition region [15].

The dynamical behavior of a neuron can be described as

$$\frac{1}{\tau}\frac{du}{dt} = -u + \sum_{i=1}^{n} w_i x_i - \theta \tag{9}$$

$$y = f(u) \tag{10}$$

where u is the state variable and τ is the time constant. It should be obvious that the steady-state solution of equation (9), if it exists, yields equation 8.

The operation performed by a neuron, as described above, can be approximately realized by a number of transconductance elements as shown in figure 6. If the nonlinearity is a step or sigmoid function, then v_{yh} is taken as the output. If the nonlinearity has a graded

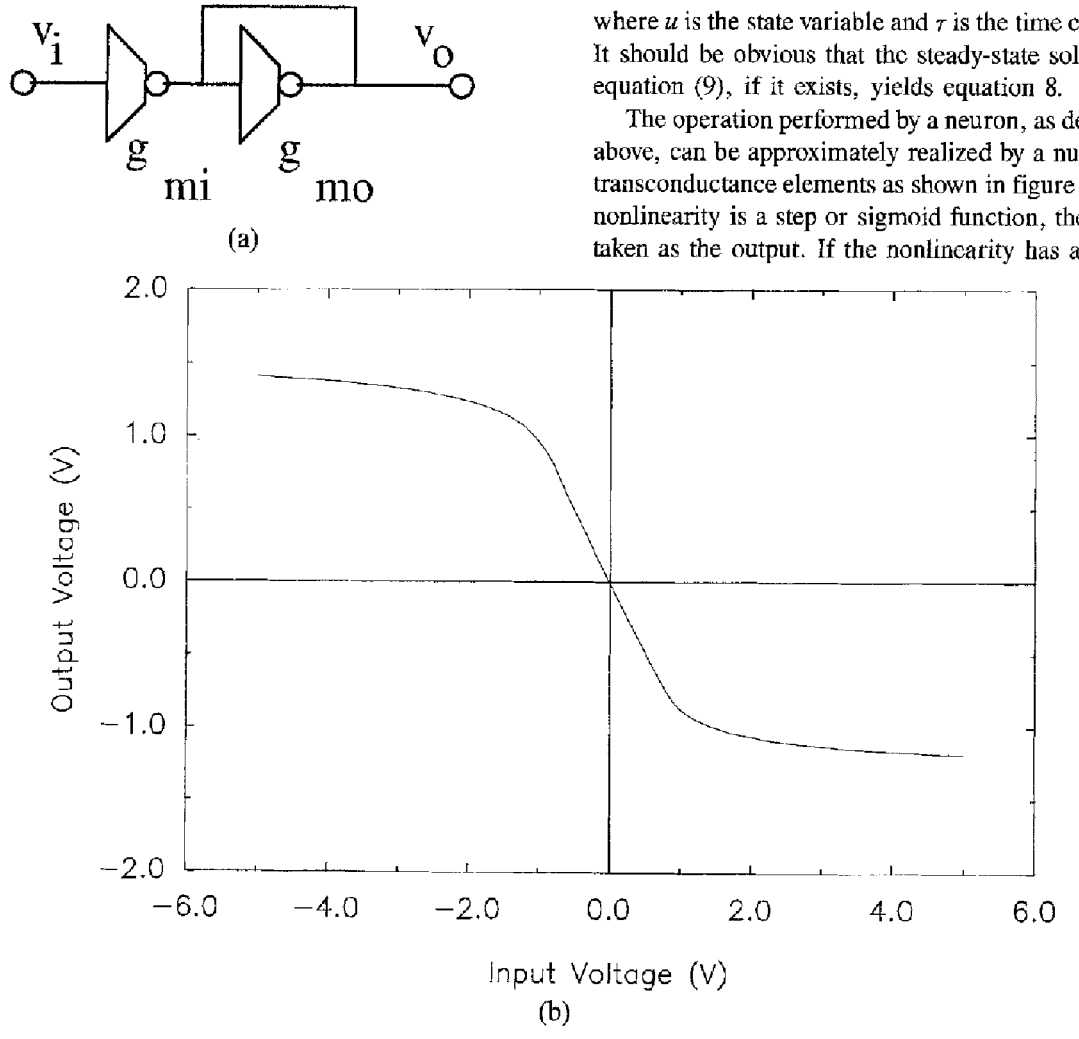


Fig. 4. A voltage buffer with transconductors, and (b) its input-output characteristic.

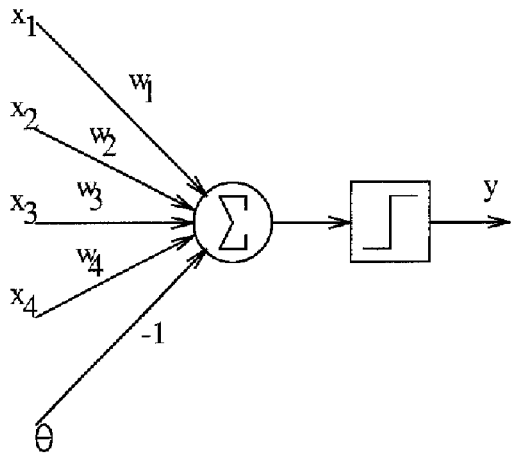


Fig. 5. Functional diagram of a single artificial neuron.

linear transition region, then v_y is taken as the output as shown in the figure.

By writing the node equation for the node at the output of input transconductors, one can easily show that

$$\frac{C}{G}\frac{dv_u}{dt} = -v_u - \frac{g_{p1}}{G}v_{x1} - \frac{g_{n2}}{G}v_{x2} - \frac{g_{p3}}{G}v_{x3} - \frac{g_{n4}}{G}v_{x4} - \frac{g_\theta}{G}v_\theta \quad (11)$$

$$v_o = v_{yh} = f_{vh}(-v_u) \quad (12)$$

or

$$v_o = v_y = f_v(-v_u) \quad (13)$$

where

$$G \triangleq g_l + g_{o\theta} + g_{ol} + \sum_{i=1}^{4} g_{oi} \quad (14)$$

where g_{ok} is the output conductance of the kth input transconductor, g_{ol} and $g_{o\theta}$ are the output conductances of the load transconductor g_l and the threshold transconductor g_θ, and C is the stray capacitance and determines the time constant of a single neuron. Note that the output conductances are absorbed by the quantity G, and the deviation in the weight values can be easily compensated by varying g_l The function $f_{vh}(\cdot)$ is the input-output voltage characteristic of a transconductor loaded with its natural output conductance, and the function $f_v(\cdot)$ is the transfer characteristic shown in figure 4. The limiting voltages can be adjusted by V_{G1} and V_{G4} according to the inequalities (4) and (5). The resemblance between equation (11) and equation (9) is obvious. The synaptic weights of the transconductor neuron are the transconductor ratios. It can

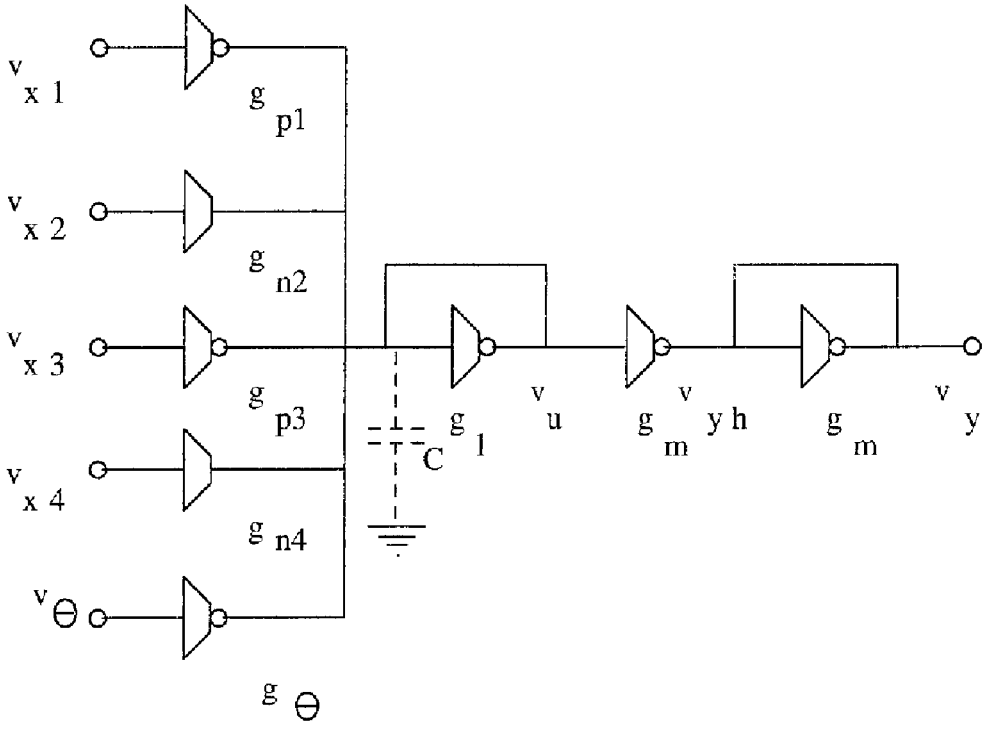


Fig. 6. Synthesis of a single neuron by transconductors.

be easily proved that the limiting range of the buffer can be set by the controlling voltages V_{G1} and V_{G4} for a given set of V_{Ti}'s according to the inequalities given in equations (4) and (5) as well.

In this paper, a new kind of signal flow graph is defined, called a *neural flow graph* (NFG), in order to facilitate the graph representation of the equations, models, and synthesis of the neural circuits. The neural flow graph is defined similarly to the signal flow graph except that the output node is depicted as a box to represent the nonlinearity. The NFG of a neuron is shown in figure 7.

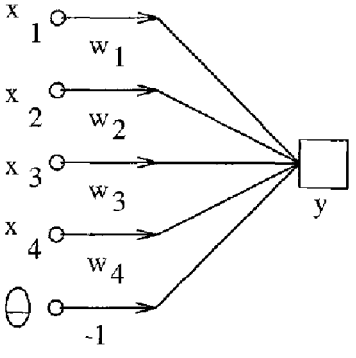


Fig. 7. Neural flow graph of a neuron.

4. Synthesis of Artificial Neural Nets

The synthesis of some of the most prominent neural networks proposed is discussed. The signal flow graph approach is employed to achieve this task.

4.1. A Single-Layer Perceptron

For instance, consider the single-layer perceptron of Minsky et al. with three inputs and three outputs and given with the neural flow graph shown in figure 8. By substituting the transconductance realization of a single neuron in figure 8, one can easily obtain the perceptron realized by transconductors as shown in figure 9.

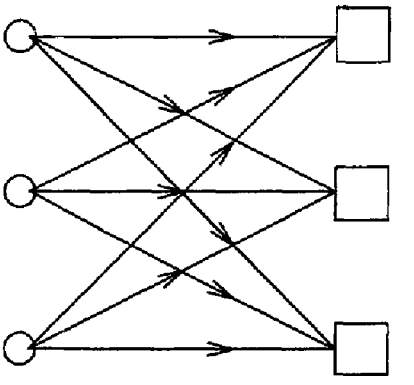


Fig. 8. Neural flow graph of a single-layer perceptron.

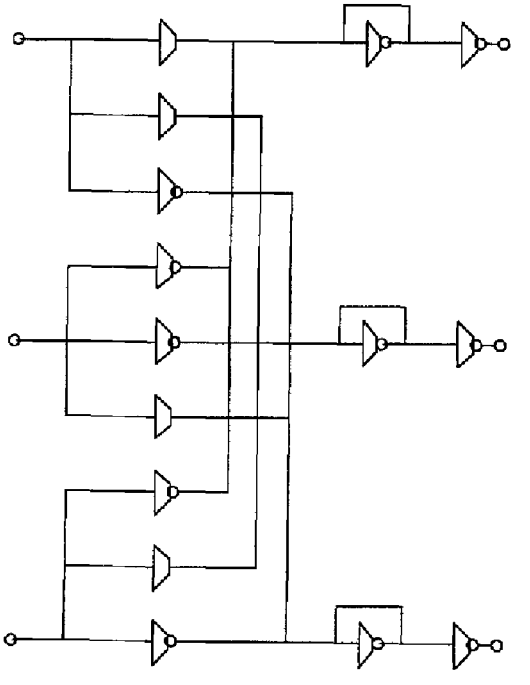


Fig. 9. Transconductor realization of the perceptron in figure 8.

4.2. Hopfield Network

The Hopfield network [24] can be redrawn in a general form [25], [26] with the neural flow graph shown in figure 10. The transconductor realization of this network is shown in figure 11.

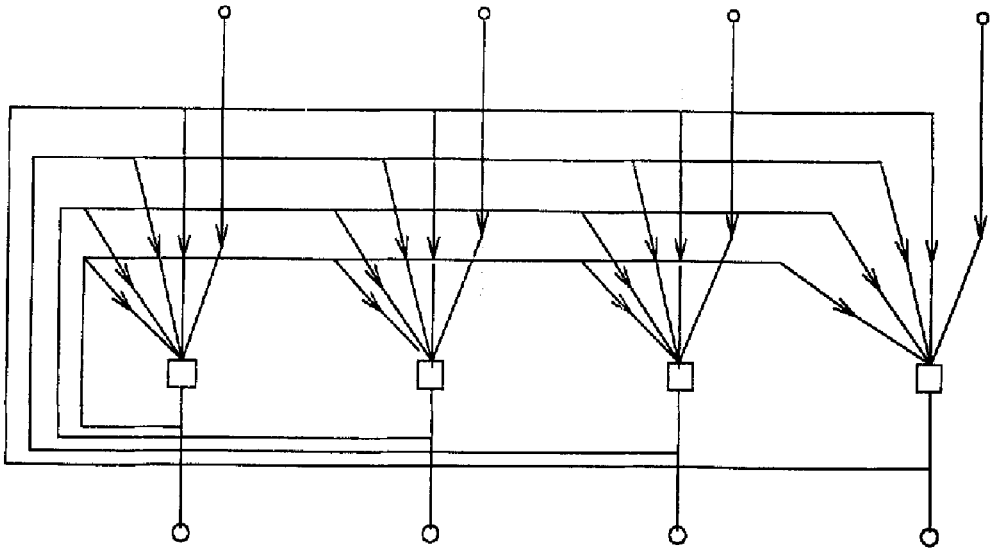


Fig. 10. Neural flow graph of Hopfield network.

4.3. Cellular Neural Network

A cellular neural network (CCN) cell can be described by the following equations:

$$C\frac{dv_{xij}(t)}{dt} = -\frac{1}{R_x}v_{xij} + \sum_{C(k,l)\in N_r(i,j)} A(i, j; k, l)v_{ykl}(t) + \sum_{C(k,l)\in N_r(i,j)} B(i, j; k, l)v_{ukl} + I \quad (15)$$

$$v_{yij}(t) = \frac{1}{2}(|v_{xij}(t) + 1| - |v_{xij}(t) - 1|) \quad (16)$$

where $v_{xij}(t)$, $v_{yij}(t)$, and v_{ukl} are the state variable and output variable of cell $C(i, j)$, and the klth input voltage to the network, respective; $N_r(i, j)$ is the r-neighborhood of cell $C(i, j)$; $A(i, j; k, l)$ and $B(i, j; k, l)$ are the synaptic weights from the output of the klth cell output and from the klth input voltage to the ijth cell; and I is the bias current (i.e., threshold) to the described neural cell.

Note that equations (15) and (16) are in the same form as equation (9). The neural graph of the CNN [15] can be drawn as in figure 12. The self-loops of the cells are not depicted, and two opposite direction branches are summarized in one, as shown in figure 13.

Again, using the neural flow graph of a transconductor neuron, one can easily synthesize the transconductor realization of a cellular neural network as shown in figure 14. Note that the output voltage buffer shown in figure 6 is omitted here because all the cells are identical; therefore the limiting ranges are identical for all cells. This exclusion of the buffer corresponds to the voltage scaling of all signal or the impedance of all weights.

4.4. Partial Differential Equation Solving Linear Neural Network

A linear partial differential equation can be approximated by a linear differential difference equation. For instance, consider

$$\frac{\partial^2 u(x, y, t)}{\partial x^2} + \frac{\partial^2 u(x, y, t)}{\partial y^2} = \frac{1}{\kappa}\frac{du(x, y, t)}{dt} \quad (17)$$

The left side of equation (18) can be approximated by

$$\frac{1}{4}[u_{ij-1}(t) + u_{ij+1}(t) + u_{i-1j}(t) + u_{i+1j}(t)] - u_{ij} \quad \text{for all } i, j \quad (18)$$

where $u_{ij}(t)$ is the ijth function approximating $u(x, y, t)$ at (ih_x, ih_y, t) and h_x and h_y are the space intervals in the x and y directions. Consequently, equation (17) is approximated by

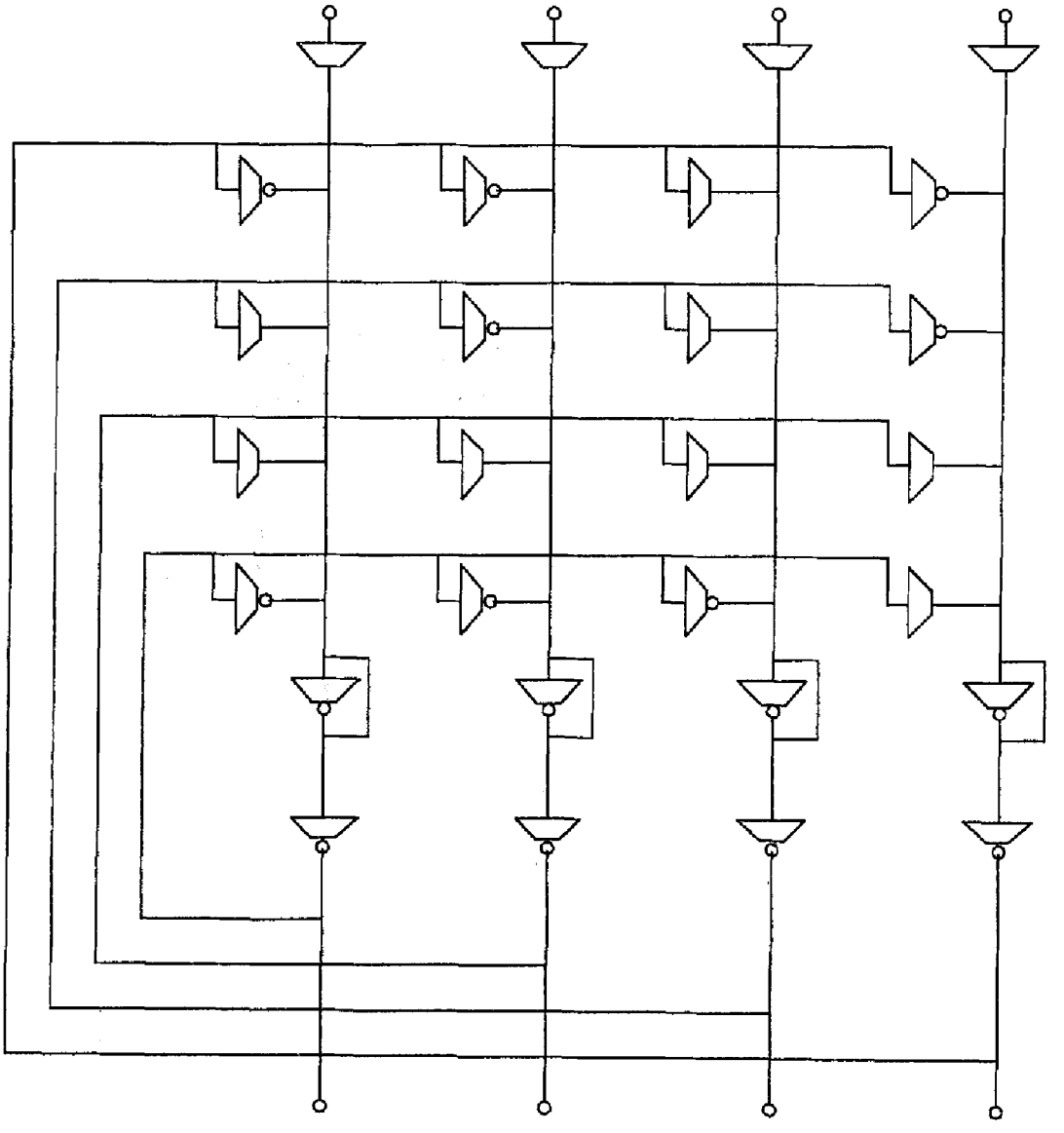


Fig. 11. Transconductor realization of the Hopfield network.

$$\frac{1}{\kappa}\frac{du_{ij}(t)}{dt} = \frac{1}{4}\left[u_{ij-1}(t) + u_{ij+1}(t) + u_{i-1j}(t) + u_{i+1j}(t)\right] - u_{ij} \tag{19}$$

Equation (19) is a special case of the state equation given in equation (9) such that the outputs are linear functions of the corresponding state variables, and therefore can be implemented by neural networks except for the nonlinearity. It is impossible to get rid of the nonlinearity of transconductors, although the linearity range can be kept wide by using a wider rail-to-rail power supply voltage. Therefore, similar to the approach for the previously discussed networks, the neural network for solving a partial differential equation is of the same form as in figure 14.

5. Example: Hopfield Net as a 4-Bit A/D Converter

A 4-bit analog-to-digital converter numerically approximates the number represented by the digital V_i variables to the input voltage x as

$$x \approx \frac{V_\Delta}{V_m}\left(\sum_{i=0}^{3} V_i 2^{(i-1)} - \frac{15}{2}\right) \tag{20}$$

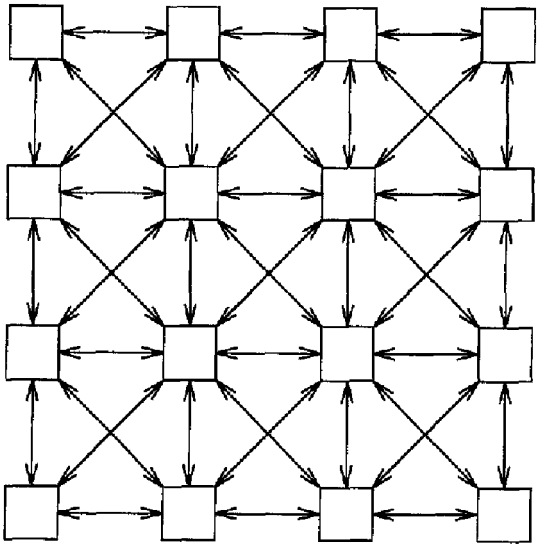


Fig. 12. Neural flow graph of cellular neural network cell.

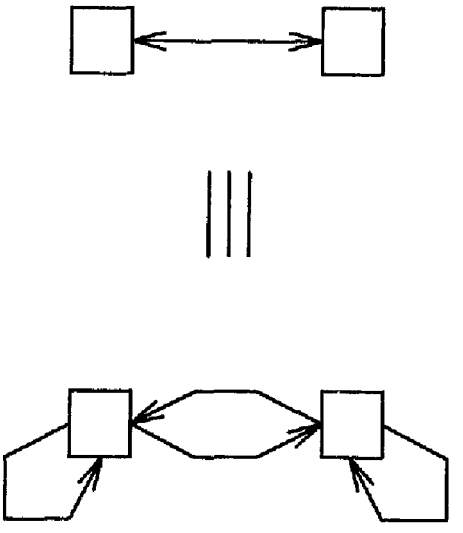


Fig. 13. Omitted branches in figure 12.

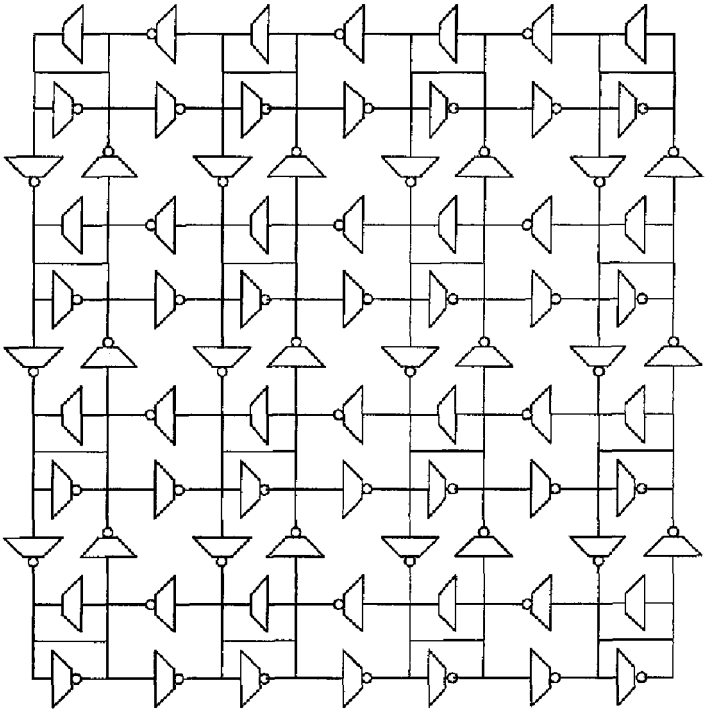


Fig. 14. Transconductor realization of the cellular neural network cell.

where V_Δ is the quantization increment and V_m is the voltage amplitude of digital V_i variables; i.e., V_i assumes $-V_m$ for logic 0 and $+V_m$ for logic 1.

Obviously, the A/D conversion problem is equivalent to the minimization of the energy or cost function

$$E = \frac{1}{2}\left(x - \frac{V_\Delta}{V_m}\sum_{i=0}^{3} V_i 2^{(i-1)} - V_\Delta\frac{15}{2}\right)^2 - \frac{1}{2}\frac{V_\Delta^2}{V_m^2}\sum_{i=0}^{3} 2^{(2i-2)}(V_i - V_m)(V_i + V_m) \quad (21)$$

The first term of equation (21) realizes the approximation expressed in equation (20). The second term makes the diagonal synaptic weights T_{ii}'s zero and forces the solution to the corners of the hypercube defined by the digital V_i variables. Equating the energy function in equation (21) to the energy function of a 4 × 4 Hopfield network as prsented in [26],

$$E = \frac{1}{2}\sum_{i=0}^{3}\sum_{j=0i\neq j}^{3} T_{ij}V_iV_j - \sum_{i=0}^{3} I_iV_i + \sum_{i}^{3} U_iV_i \quad (22)$$

yields

$$T_{ij} = -\frac{V_\Delta}{V_m}2^{i+j-2}, \qquad i = 0, \ldots, 3 \quad j = 0, \ldots, 3 \quad (23)$$

$$I_i = \left(-V_\Delta\frac{15}{2} + x\right)2^{i-1}, \qquad i = 0, \ldots, 3 \quad (24)$$

$$U_i = 0, \qquad i = 0, \ldots, 3 \quad (25)$$

where T_{ij} is the synaptic weight from the ith neuron output to the jth neuron input, V_i is the output variable of the ith neuron, I is the input bias, and U_i is the threshold voltage for the ith neuron.

CMOS transconductors of the type shown in figure 3 [19] are used for this implementation. In order to minimize the area consumption, the unit transconductance is designed with PMOS transistors where W = 7.5 μm, L = 5 μm and NMOS transistors where W = 5 μm, L = 7.5 μm, which yields the transconductance g_m = 27.29 μmhos with $V_{G1} = -V_{G4} = +4.5$ V, $V_{DD} = -V_{SS} = +5$ V. The load transconductors are chosen appropriately such that the area consumption is optimized. The relative values of the transconductors to the unit transconductances are given in figure 15. Note that the ratio of the synaptic transconductors to the load transconductors are chosen according to equations (23), (24), and (25). The effect of the output conductances have been neglected. The output nonlinearity of the neurons is implemented by simple CMOS switch pair as shown in figure 16. Accordingly, V_m = 10 mV. Because the input voltage range is chosen as 0 to 150 mV, V_Δ is chosen to be 10 mV.

The SPICE3 simulations of the network for a triangular input and a sine inut are shown in figures 17 and 18, respectively. The output of the A/D converter is converted by an ideal D/A converter for comparison. The results of the separate implementations using CMOS transconductors and ideal transconductors are shown together. The nonideality of the converter characteristics comes from the known nature of the Hopfield network as having been pointed out in [26] and can be corrected by the hardware annealing technique proposed by Lee and Sheu [27].

6. Conclusion

Transconductor realizations of the most prominent neural networks were discussed. It was shown that neural networks can be synthesized by using only transconductors. This approach facilitates fast design and layout of integrated electronic neural computers. The design of an analog neural computer is then largely reduced to the design of a generic transconductor. The approach applies to any integrated circuit technology. A brief review of integrated neural network topologies was given. In conclusion, basically three types of integrated circuit neuronlike elements are needed: a step-type (i.e., for the outputs of the perceptron and the Hopfield network), graded with a linear transition region for the cellular neural networks, and completely linear for neural networks that solve partial differential equations for vision and for computing motion. All these properties are possessed by transconductors. The only difference between these cases lies in the type of nonlinearity, the value, and the linear transition range of the buffering transconductors. For both negative and positive synaptic weights, positive and negative transconductances are available. The number of transconductors can be minimized by appropriately choosing the sign of the transconductor acting as a load resistor. Also, in technologies where simple operational transconductance amplifiers are available, the transconductors can be replaced by OTAs and the appropriate input of the OTA (i.e., inverting or noninverting) is used according to the sign of the synaptic connections.

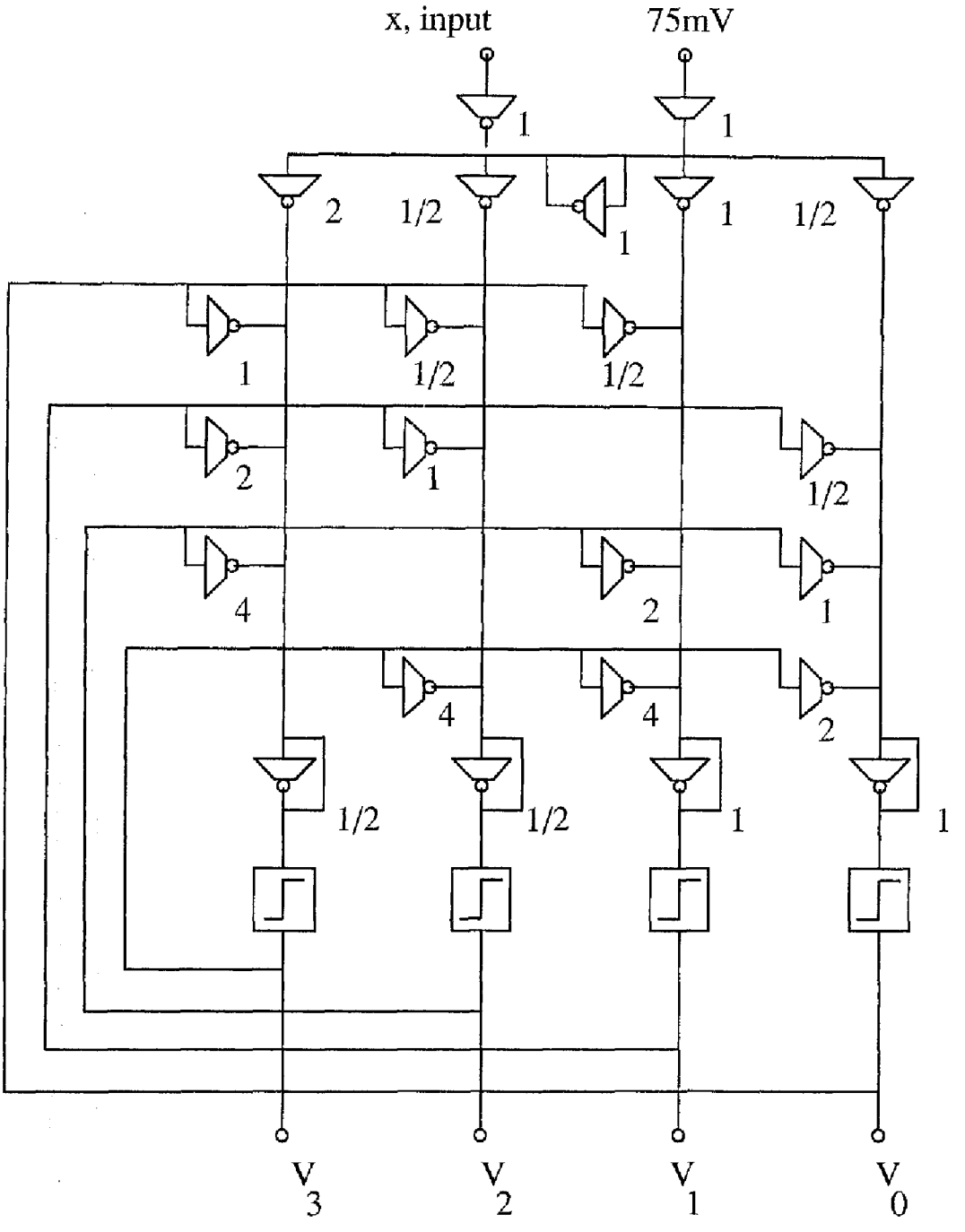


Fig. 15. Example: Hopfield network as a 4-bit analog-to-digital converter.

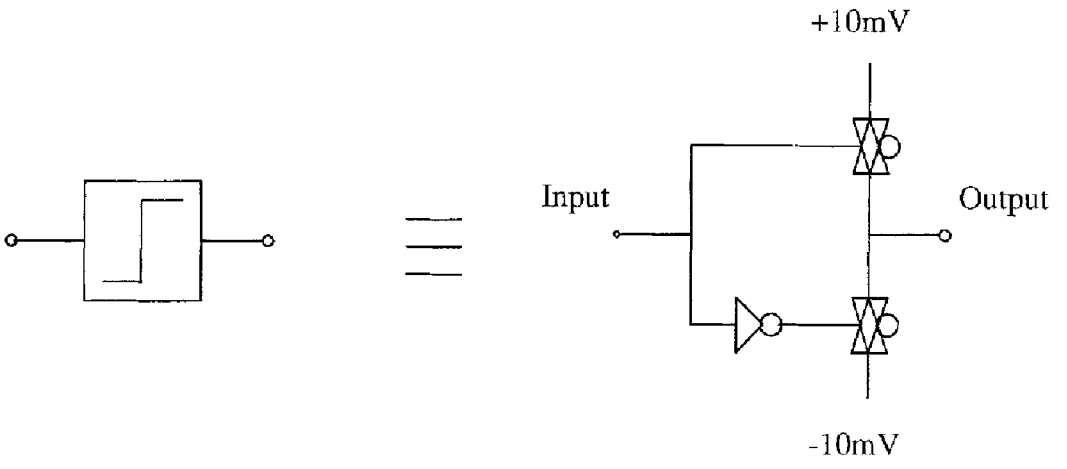


Fig. 16. CMOS transmission gate circuit realizing the output nonlinearity of the neuron.

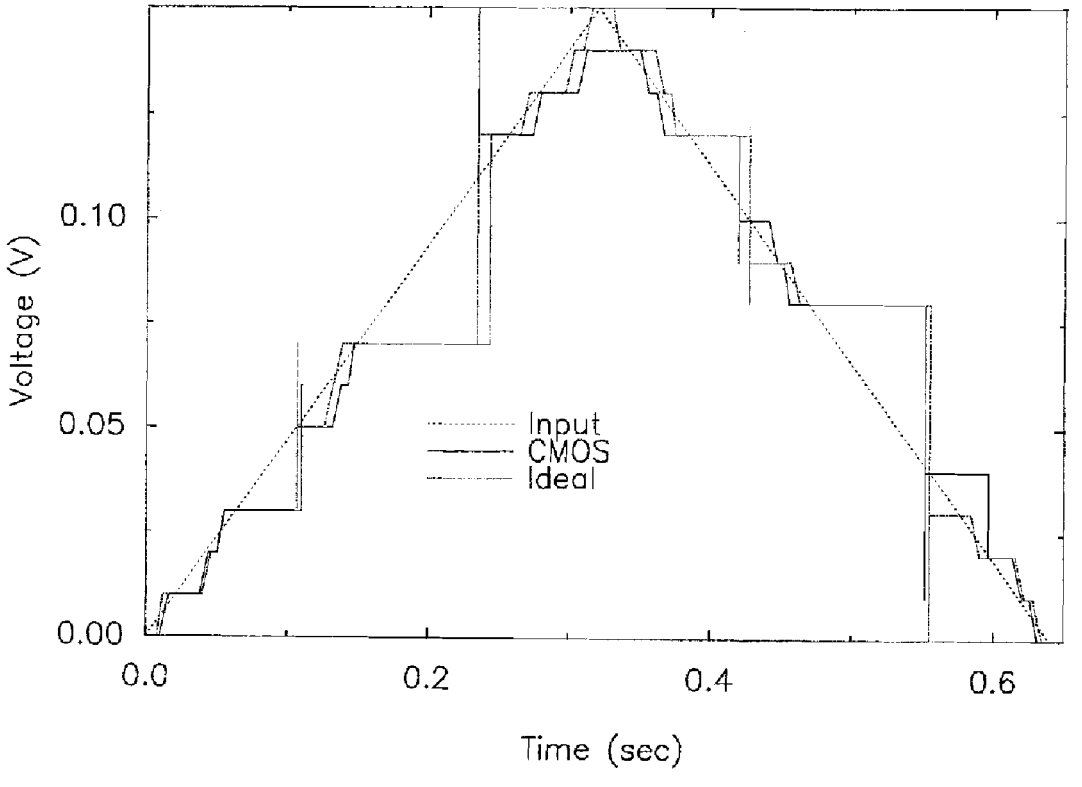


Fig. 17. A/D converter characteristic for a triangular wave input.

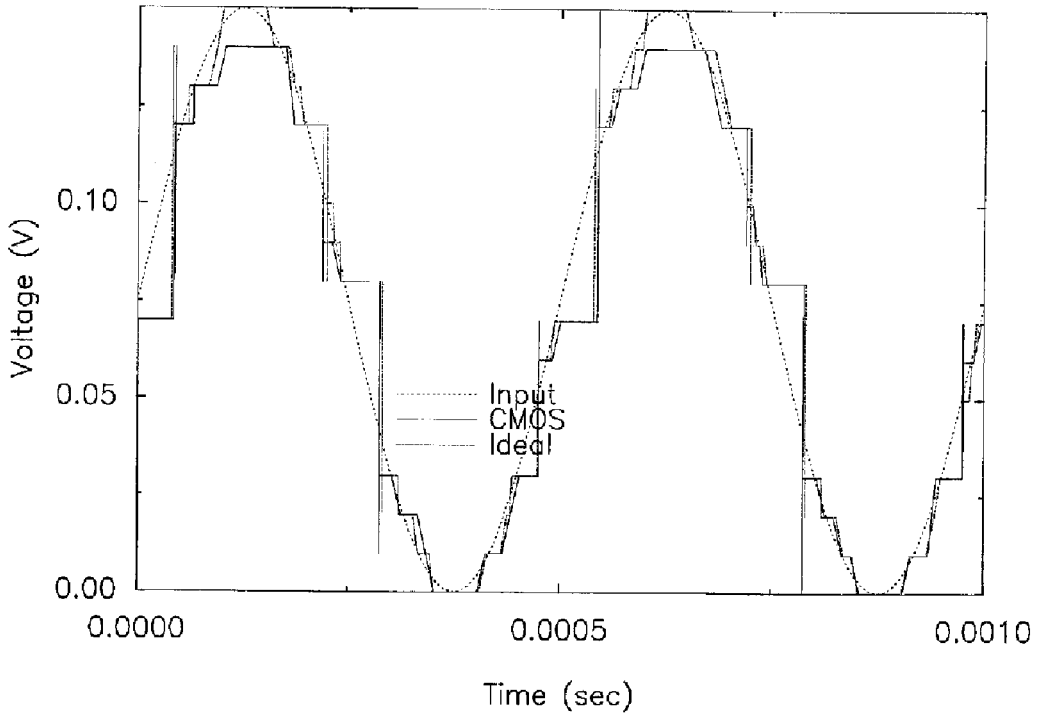
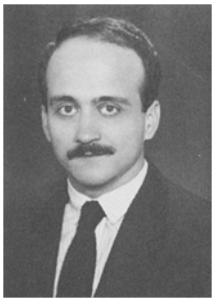


Fig. 18. A/D converter characteristic for a sine wave input.

References

1. N.H. Farhat, D. Psaltis, A Prata, and E. Perek, "Optical implementation of the Hopfield model," *Appl. Optics*, vol. 24, pp. 1469–1475, 1985.
2. H.K. Kwan, "Systolic architectures for Hopfield network, BAM and multi-layer feed-forward network," in *Proc. Int. Symp. Circ. Systems*, 1989, pp. 709–793.
3. C. Mead and M. Ismail, eds., *Analog VLSI Implementation of Neural Systems*, Kluwer: Norwell, MA, 1990.
4. C. Koch, J. Marroquin, and A. Yuille, "Analog 'neuronal' networks in early vision," *Proc. Int. Acad. Sci. USA*, vol. 83, pp. 4263–4267, 1986.
5. J. Hutchinson, C. Koch, J. Luo, and C. Mead, "Computing motion using analog and binary resistive networks," *IEEE Comput.* vol. 21, pp. 52–63, 1988.
6. K.R. Krieg, L.O. Chua, and L. Yang, "Analog signal processing using cellular neural networks," in *Proc. Int. Symp. Circ. Systems*, 1990, pp. 958–961.
7. C. Mead, *Analog VLSI and Neural Systems*, Addison-Wesley: Reading, MA, 1989.
8. Y. Tsividis and D. Anastassiou, "Switched-capacitor neural networks," *Electron. Lett.* vol. 23, pp. 958–959, 1987.
9. P. Mueller et al., "Design and fabrication of VLSI components for a general purpose analog neural computer," in *Analog VLSI Implementation of Neural Systems* (eds. Mead and Ismail) Kluwer: Norwell, MA, 1989, pp. 135–169.
10. H.P. Graf, L.D. Jackel, and W.E. Hubbard, "VLSI implementation of a neural network model," *IEEE Comput.* vol. 21, pp. 41–49, 1989.
11. Y. Tsividis and S. Satyanarayana, "Analogue circuits for variable-synapse electronic neural networks," *Electron. Lett.* vol. 23, pp. 1313–1314, 1987.
12. S. Satyanarayana and Y. Tsividis, "Analogue neural networks with distributed neurons," *Electron. Lett.* vol. 25, pp. 302–303, 1989.
13. J.J. Paulos and P.W. Hollis, "Neural networks using analog multipliers," in *Proc. Int. Symp. Circ. Systems*, 1988, pp. 499–502.
14. D.D. Caviglia, G.M. Bisio, P. Daglio, and M. Valle, "CMOS circuit design of programmable neural net classifier of 'exclusive' classes," *Electron. Lett.* vol. 25, pp. 1074–1076, 3 1989.
15. L.O. Chua and L. Yang, "Cell neural networks: Theory," *IEEE Trans. Circ. Systems* vol. CAS-35, pp. 1257–1274, 1988.
16. L.O. Chua and L. Yang, "Cellular neural networks: Applications," *IEEE Trans. Circ. System* vol. CAS-35, pp. 1275–1290, 1988.
17. L. Yang, L.O. Chua, and K.R. Krieg, "VLSI implementation of cellular neural networks," in *Proc. Int. Symp. Circ. Systems*, 1990, pp. 2425–2427.
18. A. Nedungadi and T.R. Viswanathan, "Design of linear CMOS transconductance elements," *IEEE Trans. Circ. Systems* vol. CAS-33, 891–894, 1985, pp. 1132–1138, 1986.
19. C.S. Park and R. Schaumann, "A high-frequency CMOS linear transconductance element," *IEEE Trans. Circ. Systems* vol. CAS-33, pp. 1132–1138, 1986.
20. E. Seevinck and R.F. Wassenaar, "A versatile CMOS linear transconductor/square-law function circuit," *IEEE J. Solid-State Circ.* vol. SC-22, pp. 366–377, 1987.
21. D.G. Haigh and C. Toumazou, "High-frequency gallium arsenide linearised transconductor for communications," *Electron. Lett.* vol. 26, pp. 497–498, 1990.

22. R.D. Reed and R.L. Geiger, "A multiple-input OTA circuit for neural networks," *IEEE Trans. Circ. Systems* vol. CAS-36, pp. 767–770, 1990.
23. R. Beale, *Neural Computing: An Introduction*, Adam Hilger, New York, 1990.
24. J.J. Hopfield, "Neural networks and phsycial systems with emergent collective computational abilities," *Proc. Nat. Acad. Sci., USA*, vol. 79, pp. 2554–2558, 1982.
25. J.J. Hopfield, "Neurons with graded response have collective computational properties like those of two-state neurons," *Proc. Nat. Acad. Sci. USA*, vol. 81, pp. 3088–3092, 1984.
26. D.W. Tank and J.J. Hopfield, "Simple "neural" optimization networks: An a/d converter, signal decision circuit, and a linear programming circuit," *IEEE Trans. Circ. Systems* vol. 33, pp. 533–541, 1986.
27. B.W. Lee and B.J. Sheu, "Hardware annealing in electronic neural networks," *IEEE Trans. Circ. Systems* vol. 38, pp. 134–137, 1991.



Mehmet Ali Tan was born in 1959 in Adana, Turkey. He received his B.S. and M.S. degrees from Istanbul Teknik Üniversitesi, Istanbul, Turkey, in 1980 and 1982, respectively and his Ph.D. from the University of Minnesota, Minneapolis, MN, in 1988. In 1988, he joined the Department of Electrical and Electronic Engineering at Bilkent University, Ankara, Turkey, where he is currently Assistant Professor. His active research interests include analog integrated circuits and signal processing, electronic implementation of artificial neural networks, and computer-aided design of electronic circuits.