ANALOG CMOS IMPLEMENTATION OF CELLULAR NEURAL NETWORKS

A THESIS
SUBMITTED TO THE DEPARTMENT OF ELECTRICAL
AND ELECTRONICS ENGINEERING
AND THE INSTITUTE OF ENGINEERING AND SCIENCES
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

By
Ismet Adil Baboğ
July, 1991

ANALOG CMOS IMPLEMENTATION OF CELLULAR NEURAL NETWORKS

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND

ELECTRONICS ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCES

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

İzzet Adil Baktır

July 1991

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Mehmet Ali Tan(Principal Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Abdullah Atalar

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Enis Çetin

Approved for the Institute of Engineering and Sciences:

Prof. Dr. Mehmet Baray
Director of Institute of Engineering and Sciences

ABSTRACT

ANALOG CMOS IMPLEMENTATION OF CELLULAR NEURAL NETWORKS

İzzet Adil Baktır
M.S. in Electrical and Electronics Engineering
Supervisor: Assoc. Prof. Dr. Mehmet Ali Tan
July 1991

An analog CMOS circuit realization of cellular neural networks with transconductance elements is presented in this thesis. This realization can be easily adapted to various types of applications in image processing by just choosing the appropriate transconductance parameters according to the predetermined coefficients. The noise–reduction and edge detection examples have shown the effectiveness of the designed networks in real time image processing applications. For "fix function" cellular neural network circuits the number of transistors are reduced further by a new multi–input voltage–controlled current source.

Keywords : Cellular Neural Networks, Analog VLSI, CMOS, transconductance.

ÖZET

HÜCRESEL SİNİR AĞLARININ EŞLENİK-METAL-OKSİT-YARIİLETKEN DEVRELERLE GERÇEKLENMESİ

İzzet Adil Baktır
Elektrik ve Elektronik Mühendisliği Bölümü Yüksek Lisans
Tez Yöneticisi: Doç. Dr. Mehmet Ali Tan
Temmuz 1991

Bu çalışmada, yeni bir sınıf doğrusal olmayan bilgi işleme sistemi olan Hücresel Sinir Ağlarının (CNN), Eşlenik-Metal-Oksit-Yarıiletken (CMOS) transkondüktans elemanlarla gerçeklenmesi sunulmaktadır. Bu gerçeklemenin, görüntü işlemedeki değişik kullanım alanlarına uyarlanması, görüntü işleme tekniklerine ve/veya bilgisayar benzetişimlerine göre önceden bulunan katsayılara uygun transkondüktans parametrelerinin seçimiyle sağlanabilir. Gürültü yoketme ve kenar belirleme örnekleri, bu gerçeklemenin gerçek zamanda yapılan görüntü işleme amacıyla kullanılabileceğini göstermektedir. "Sabit fonksiyonlu" hücresel sinir ağlarının gerçeklenmesinde, çok-girişli yeni bir gerilim-kontrollü akım kaynağıyla transistör sayısı azaltılmıştır.

Anahtar kelimeler : Hücresel Sinir Ağları, Analog VLSI, CMOS, transkondüktans.

ACKNOWLEDGMENT

I would like to thank Assoc. Prof. Dr. Mehmet Ali Tan for his supervision, guidance, suggestions and encouragement throughout the development of this thesis.

It is pleasure to express my thanks to all my friends for their valuable discussions and to my family for providing morale support during this study.

Contents

List of Figures

Chapter 1

Introduction

1.1 Artificial Neural Networks

Neurons are living nerve cells and neural networks are networks of these cells. The cerebral cortex of the brain is an example of a *natural* neural network. The average human brain consists of 1.5×10^{10} neurons of various types with each neuron receiving signals through as many as 10^4 synapses. With that kind of complexity, it is no wonder that the human brain is considered to be the most complex piece of biological machinery on the earth.

Today the term *Artificial Neural Network* (ANN) has come to mean any computing architecture that consists of massively parallel interconnections of simple neural processors. Artificial neural networks try to mimic, at least partially, the structure and functions of brains and nervous systems. The motivation comes mainly from the fact that humans are much better at pattern recognition than digital computers. In traditional single processor Von Neumann computers, the speed is limited by the propagation delay of transistors. Artificial neural networks, on the other hand, because of their massively parallel nature, can perform computations at a much higher rate [1, 2]. Furthermore, because of their robust (fault–tolerance) nature, a few degraded or non–functional processing elements will not greatly affect the overall operation of the neural network. The speed and fault–tolerance of ANNs make them attractive for variety of applications, such as pattern recognition, robotic control, and optimization.

The complexity of a neural system does not stem from the complexity of its devices but rather from the multitude of ways in which a large collection of these devices can interact. It is generally assumed that the performance of

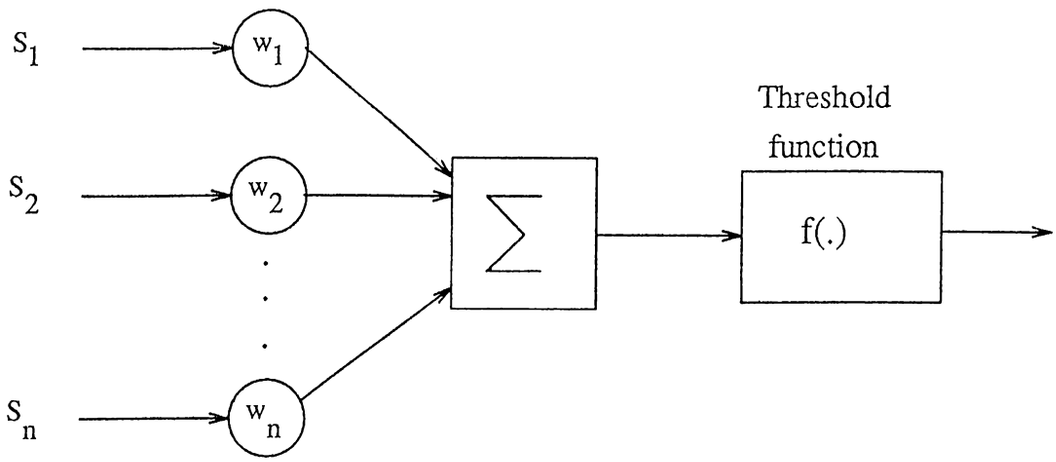


Figure 1.1: Structure of an artificial neuron.

neural systems arises from the collective behavior of many primitive, highly interconnected processing units. Therefore, artificial neural networks can be characterized by their degree of connectivity. Hopfield's networks [3] interconnect fully. A feedback process connects the output of each neuron to the input of each other neuron. In Rumelhart's multilayer Perceptrons [4], the output of each neuron connects to all neurons in the next layer. In locally interconnected networks like Cellular Neural Networks, every neuron is connected to the nearest neurons in a neighborhood. In each of these three models, the functionality of neurons (processing elements) and synapses (interconnection elements) is roughly equal. An artificial neuron performs the weighted summa-

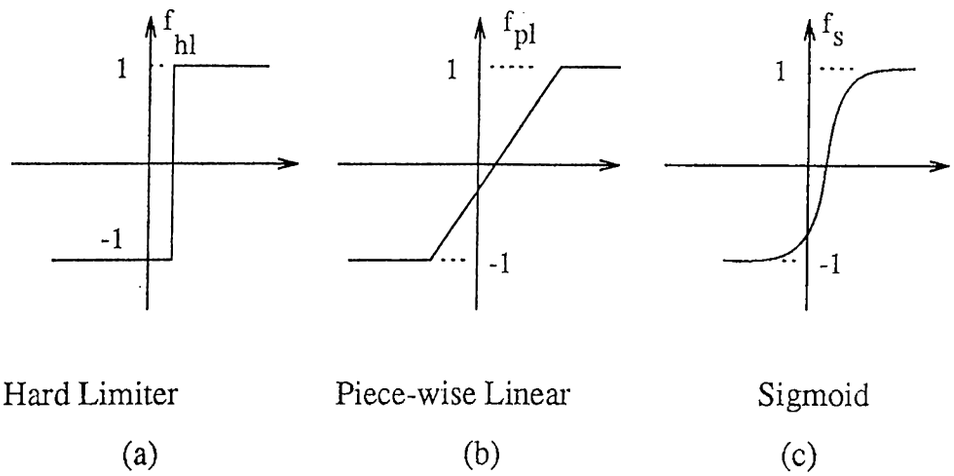


Figure 1.2: Nonlinear neuronal functions : hard limiter (a), piecewise linear (b) and sigmoid (c).

tion of its inputs as shown in Fig.1.1. The result of this summation proceeds through a threshold function. Fig.1.2 shows three common types of nonlinear threshold functions; hard limiter, piecewise linear and sigmoid. The function of a synapse is to perform a simple multiplication between the output value

of the connected neuron and the weight coefficient w_i which may be a positive (called excitatory) or a negative (called inhibitory) real number.

One of the most important aspects of neural networks is their learning capability, whereby synaptic weights between neurons are adaptively changed according to a learning procedure. Networks trained with supervision such as Hopfield and Perceptrons are used as associative memories or as classifiers. These nets provided with side information or labels that specify the correct class for new input patterns during training. ANNs trained without supervision, such as the Kohonen's feature–map forming nets [5] are used as vector quantizers or to form clusters. No information concerning the correct class is provided to these nets during training.

1.2 VLSI Circuits For Neural Networks

Simulations performed on classical computers account for most of the actual research in artificial neural networks in recent years. However speed looses its importance in simulations because they are much slower intrinsically than electronic devices. Since effective simulations of neural networks exceed the limits of conventional machines, researchers have been working on analog and digital VLSI implementations of neural networks which take the advantage of the inherent parallelism to yield fast solutions. Moreover the integrated neural networks are needed for decentralized mobile systems, robotics and automotive applications in the expanding area of microelectronics.

Neural network models have highly parallel, regular and modular architectures based on matrices that make them attractive for VLSI systems [6, 7, 15]. Due to the placing and routing problems in silicon wires occupy the most space on an integrated circuit and high–performance interconnections limit the possible number of integrated processing units. Furthermore, communication delays degrade the performance, becoming progressively more expensive in silicon area and propagation time. Therefore, a successful integration of neural network should exhibit the following architectural properties :

- design simplicity that lead to an architecture based on copies of a few simple cells and simple chips;

- regularity of the communication structure that reduces wiring problems and localized communications;

• expandibility and design scalability that allow many identical units by packing a number of neurons onto a chip and interconnecting many chips for a complete system.

In the implementation of neural networks, the weighted summation of input signals (activation function) incurs the largest computational load. A digital implementation serially calculates this sum and requires a data bus for each processing unit. The bus must have the width proportional to the data format of the weights and inputs. As a consequence, the summation must be synchronous but it gives a high precision and noise immunity. In analog case the weighted sum of the inputs is performed by summing the analog currents or charge packets. A conventional operational amplifier (the simplest circuit is an inverter or an analog comparator) can perform the transfer function. Analog implementation is fast and requires less silicon area than digital implementations.

The second critical task is the implementation of connection element. The design of interconnection elements must balance the cell size and the resolution of the connection weight. The implementation of digital memories are well mastered techniques and storage in analog memories is difficult. Proposals for analog synapses include capacitors, charge–coupled devices (CCDs) and MNOS/CCD (metal nitride oxide silicon) circuits [8].

Learning or self-organization does require incremental adjustment of the weights in small steps. In general, such a connection element requires considerable circuitry, and hence a large amount of silicon area especially in the case of digital weights. Therefore analog circuits are most appropriate for learning algorithms with high fault–tolerance and requiring moderate or low precision, while digital circuits are used for high–resolution learning algorithms.

Optoelectronics can offer a solution to the adaptation and high data rate problems in neural networks by integrating light waveguides and photo diodes into silicon [9, 10].

1.3 Motivation and Approach

A Cellular Neural Network (CNN) as proposed by Chua and Yang, is a special type of analog nonlinear processor array. Due to their continuous–time dynamics and parallel processing features, analog CNN circuits are very effective in real time image processing applications such as noise removal, edge

detection and feature extraction [11]. The regularity, the parallelism and the local connectivity found in CNN circuits architecture make it suitable for VLSI implementations.

This work presents the implementation of cellular neural network structure using analog CMOS circuits. One of our major goals is design simplicity. To achieve this, the design is reduced to CMOS transconductance elements. One can easily adapt this realization to various types of applications by just choosing the appropriate transconductance parameters according to the predetermined coupling coefficients between the neighboring cells. These coefficients may be either set according to a computer simulation or chosen based on prominent kernels for image processing.

Another important reason for using CMOS transconductance elements is the requirement of adaptability. To achieve programmable coupling coefficient, the transconductance parameters can be adjusted with external voltage sources.

In the implementation of "fix function" cellular neural networks that performs one or a related set of processing function using fixed coefficients, the number of transistors is reduced further by a new multi–input voltage–controlled current source.

Chapter 2

Cellular Neural Networks

A novel class of information processing system called Cellular Neural Network (CNN), possesses some of the key features of neural networks like asynchronous parallel processing, continuous–time dynamics and global interaction of network elements. They have important potential applications in such areas as image processing and pattern recognition [12]. In this chapter we will briefly review the architecture, stability and applications of CNN proposed by Chua et.al.

2.1 Architecture of Cellular Neural Network

In a cellular neural network structure, the basic circuit unit called *cell*, is interconnected directly to the nearest cells in a neighborhood. The cells which are not directly connected together may effect each other indirectly because of the propagation effects of the continuous–time dynamics. An example of $M \times N$ cellular neural network structure is shown in Fig. 2.1. The ith row and jth column cell is indicated as $C(i,j)$. The r- neighborhood N_r of radius r of cell $C(i,j)$ in a $M \times N$ cellular neural network is defined by :

$$N_r(i,j) = \{C(k,l)|max\{|k-i|,|l-j|\} \leq r, \quad 1 \leq k \leq N \qquad (2.1)$$

where r is an positive integer number.

The principle circuit model of cell $C(i,j)$ is shown in Fig. 2.2, where the suffixes u,x and y denote the input, state and output, respectively. It is constructed from linear and nonlinear dependent sources, linear resistors and a linear capacitor.

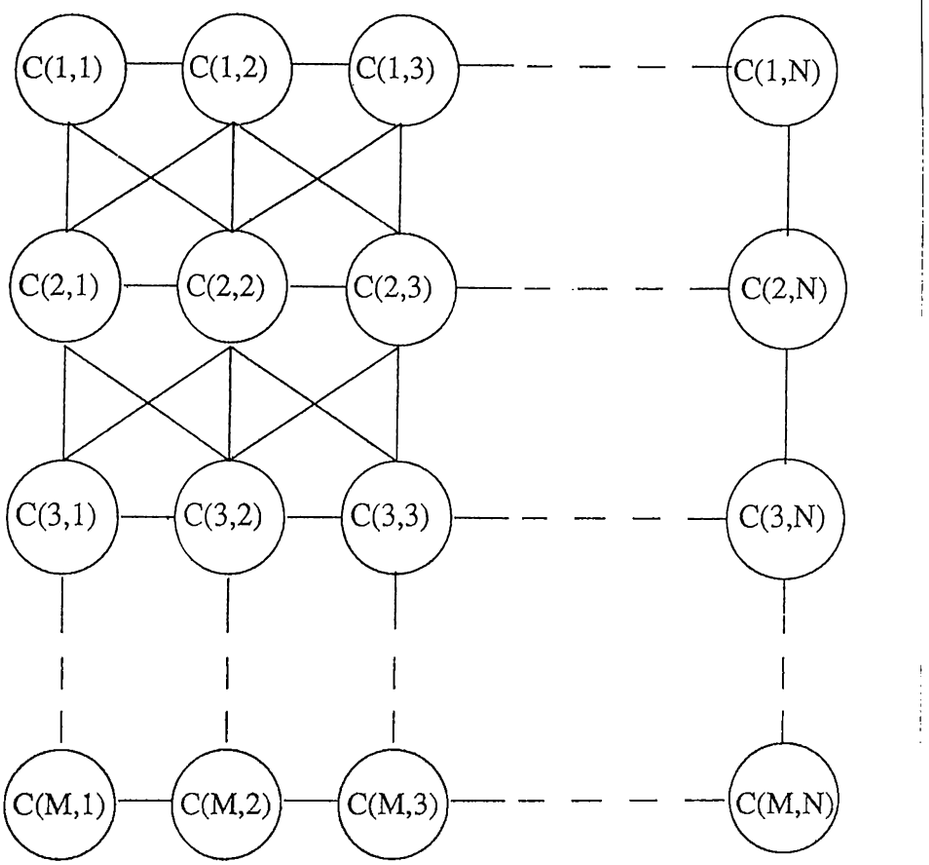


Figure 2.1: $M \times N$ Cellular Neural Network Structure.

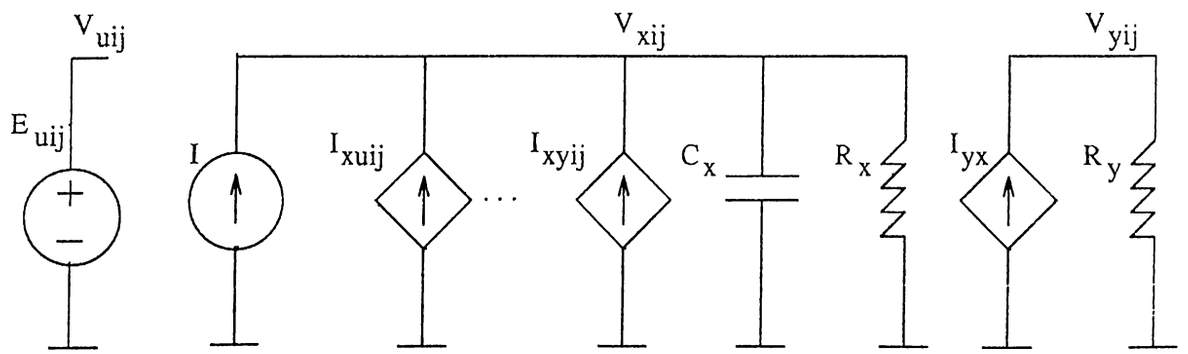


Figure 2.2: The principle circuit model of cell $C(i, j)$.

The node voltage v_{uij} is called the input of the cell $C(i,j)$ and is assumed to be a constant with magnitude less than or equal to 1. The node voltage v_{xij} of $C(i,j)$ is defined as the state of the cell whose initial condition is assumed to be a constant with magnitude less than or equal to 1. Finally, the node voltage v_{yij} is defined as the output.

I_{xu} and I_{xy} are the voltage–controlled current sources which are coupled to its neighboring cells via the input v_{ukl} and the output v_{ykl} of each neighbor cell $C(k,l)$ with the characteristics

$$I_{xu}(i,j;k,l) = B(i,j;k,l)v_{ukl} \tag{2.2}$$

and

$$I_{xy}(i,j;k,l) = A(i,j;k,l)v_{ykl} \tag{2.3}$$

Therefore, the input control and the output feedback of the CNN architecture depend on the transconductances $B(i,j;k,l)$ and $A(i,j;k,l)$, respectively. The nonlinear output voltage–controlled current source $I_{yx}(i,j)$ has the characteristic

$$I_{yx}(i,j) = \frac{1}{R_y} f(v_{xij}) \tag{2.4}$$

where $f(\cdot)$ is either a piece-wise linear or a sigmoid type of function as shown in Fig. 2.3. I is the biasing current and $E_{i,j}$ is the constant input voltage.

By applying Kirchhoff's current and voltage laws, the circuit equations of a cell in an $N \times M$ CNN can be easily derived as follows:

State equation:
$$\begin{aligned} C\frac{dv_{xij}(t)}{dt} &= -\frac{1}{R_x}v_{xij}(t) + \\ &+ \sum_{C(i,j)\in N_r(i,j)} A(i,j;k,l)v_{ykl}(t) + \\ &+ \sum_{C(i,j)\in N_r(i,j)} B(i,j;k,l)v_{ukl} + I \end{aligned} \tag{2.5}$$

Input and Output equations : $v_{uij}(t) = E_{ij} \;\; ; \;\; v_{yij}(t) = f(v_{xij}(t))$ (2.6)

Constraint conditions: $|v_{xij}(0)| \leq 1 \; ; \; |v_{uij}| \leq 1$ (2.7)

Parameter assumption: $A(i,j;k,l) = A(k,l;i,j)$ (2.8)

where $1 \leq i,k \leq M; 1 \leq j,l \leq N$.

In practice, the magnitude of the signals can always be normalized to satisfy the constraint conditions and the parameter assumption is reasonable because of the symmetry property of the neighborhood system.

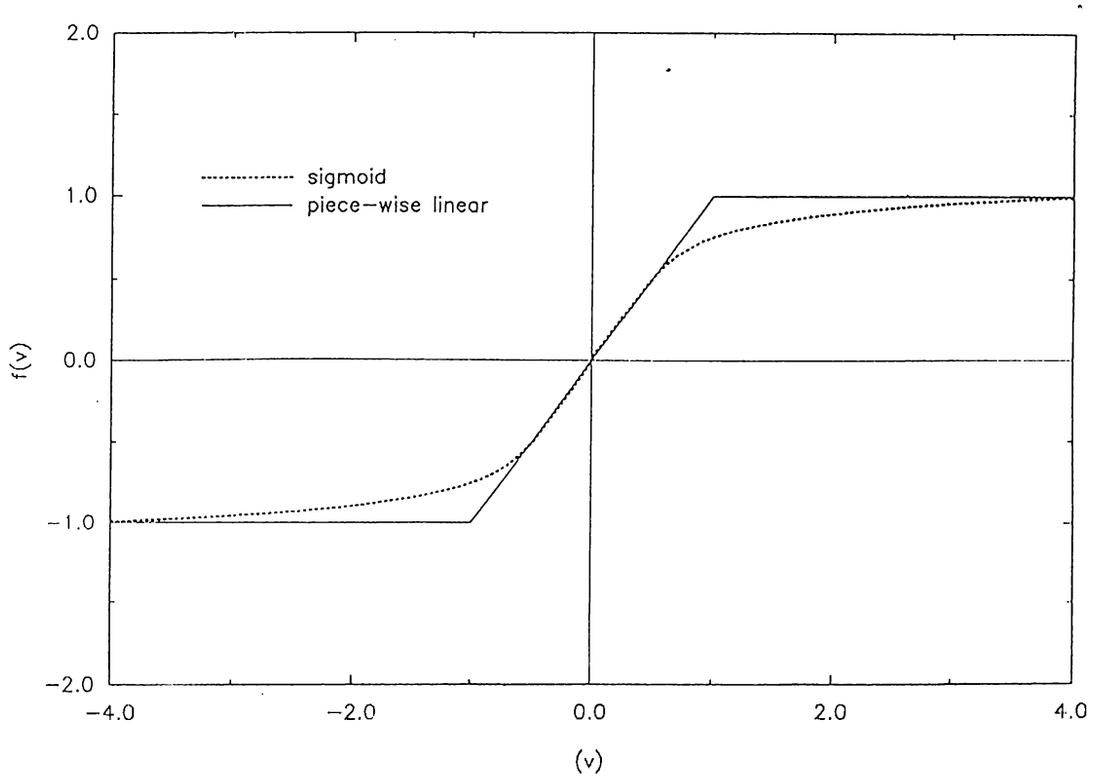


Figure 2.3: The nonlinear characteristics of $f(\cdot)$.

2.2 Stability

In order to determine the dynamic range of all node voltages in the network, it is proven that, the state v_{xij} of each cell in a CNN is bounded for all time $t \geq 0$ and the bound v_{max} can be computed by the following formula for *any* CNN [11] :

$$v_{max} = 1 + R_x|I| + R_x \max_{1\leq i\leq M,\ 1\leq j\leq N}[\sum_{C(i,j)\in N_r(i,j)} (|A(i,j;k,l)| + |B(i,j;k,l)|)] \tag{2.9}$$

For any CNN, the parameters $R_x, C, I, A(i,j;k,l)$ and $B(i,j;k,l)$ are finite constants, therefore the bound on the states of the cells, v_{max}, is finite and can be computed via formula (2.9).

In stability analysis, it can be easily proven that [11], every cell output has two equilibrium points (± 1). Therefore, after the transient has settled down, a CNN always approaches one of its equilibrium points. In other words,

$$\lim_{t\to\infty} v_{xij}(t) = constant; \quad 1\leq i \leq M,\ 1\leq j\leq N \tag{2.10}$$

or

$$\lim_{t\to\infty} \frac{dv_{xij}(t)}{dt} = 0 \tag{2.11}$$

Moreover if the circuits parameters satisfy

$$A(i,j;k,l) > \frac{1}{R_x} \tag{2.12}$$

then

$$\lim_{t\to\infty} |v_{xij}| \geq 1 \tag{2.13}$$

or equivalently,

$$\lim_{t\to\infty} v_{yij}(t) = \pm 1. \tag{2.14}$$

The equation (2.14) is significant for cellular neural networks, because it implies that the circuit will not oscillate or become chaotic. This equation also guarantees that CNN converges to a binary value output which is a crucial property for solving classification problems in image processing applications.

2.3 Application to Image Processing

In order to understand the image transform mechanism in cellular neural networks, let us rewrite the state equation (2.5) in its equivalent integral form as follows:

$$v_{xij}(t) = v_{xij}(0) + \frac{1}{C}\int_0^t [\frac{-1}{R_x} v_{xij}(\tau) + f_{ij}(\tau) + g_{ij}(\tau) + I]d\tau \tag{2.15}$$

where

$$f_{ij}(t) = \sum_{C(i,j)\in N_r(i,j)} A(i,j;k,l)v_{ykl}(t) \tag{2.16}$$

and

$$g_{ij}(t) = \sum_{C(i,j)\in N_r(i,j)} B(i,j;k,l)v_{ukl} \tag{2.17}$$

Equation (2.15) represents the image at time t, which depends on the initial image $v_{xij}(0)$ and the dynamic rules of a cellular neural network. Therefore, cellular neural networks can be used to obtain a *dynamic transform* of an initial image at any time t. In special case $t \to \infty$, the state variable $v_{xij}(t)$ tends to constant and the output $v_{yij}(t)$ tends to either $+1$ or -1 as stated in equation (2.14).

The result of this dynamic transform depends on the choice of the cell equivalent circuit element values, i.e. transconductance of linear current sources

($A(i,j;k,l)$ and $B(i,j;k,l)$), the bias current (I), the resistances (R_x and R_y), capacitance C and the radius of the direct interactions between cells (r). Elements mentioned above are called CNN parameters and how to choose these parameters to achieve a desired transformation is currently still an active research problem [25], [26]. Some application possibilities like noise–reduction and edge detection are mentioned in Chapter 4, and others can be found in the references [12],[17]–[20].

Chapter 3

Analog CMOS Implementation of CNN

3.1 CMOS Transconductance Element

The linear CMOS transconductance element (voltage–to–current transducer) which is shown in Fig. 3.1.(a), resembles in most respects that of the CMOS inverter but without the matching problems between PMOS and NMOS transistors and with the additional advantage of tunability [21]. In this section, DC operation of this CMOS transconductance element is analyzed and some SPICE simulations are shown to demonstrate the performance.

In DC analysis of the linear region defined in Fig. 3.1.(b), using the standard square–law model for MOS in their saturation region and assuming a perfect matching between the geometrically identical NMOS transistors $M1, M2$ and between PMOS devices $M2, M4$, the currents I_a and I_b defined in Fig. 3.1.(a) are easily derived as

$$I_a = k_{eff}(V_{G1} - V_{in} - V_{Tn1} - |V_{Tp2}|)^2 \tag{3.1}$$

$$I_b = k_{eff}(V_{G4} + V_{in} - V_{Tn3} - |V_{Tp4}|)^2 \tag{3.2}$$

where

$$k_{eff} = \frac{k_n k_p}{(\sqrt{k_n} + \sqrt{k_p})^2} \tag{3.3}$$

$$k_{n,p} = \frac{1}{2}(\mu_{eff} C_{ox}(\frac{W}{L}))_{n,p} \tag{3.4}$$

and $V_{Tni} \geq 0$, $V_{Tpj} \leq 0$. These parameters have their usual meanings.

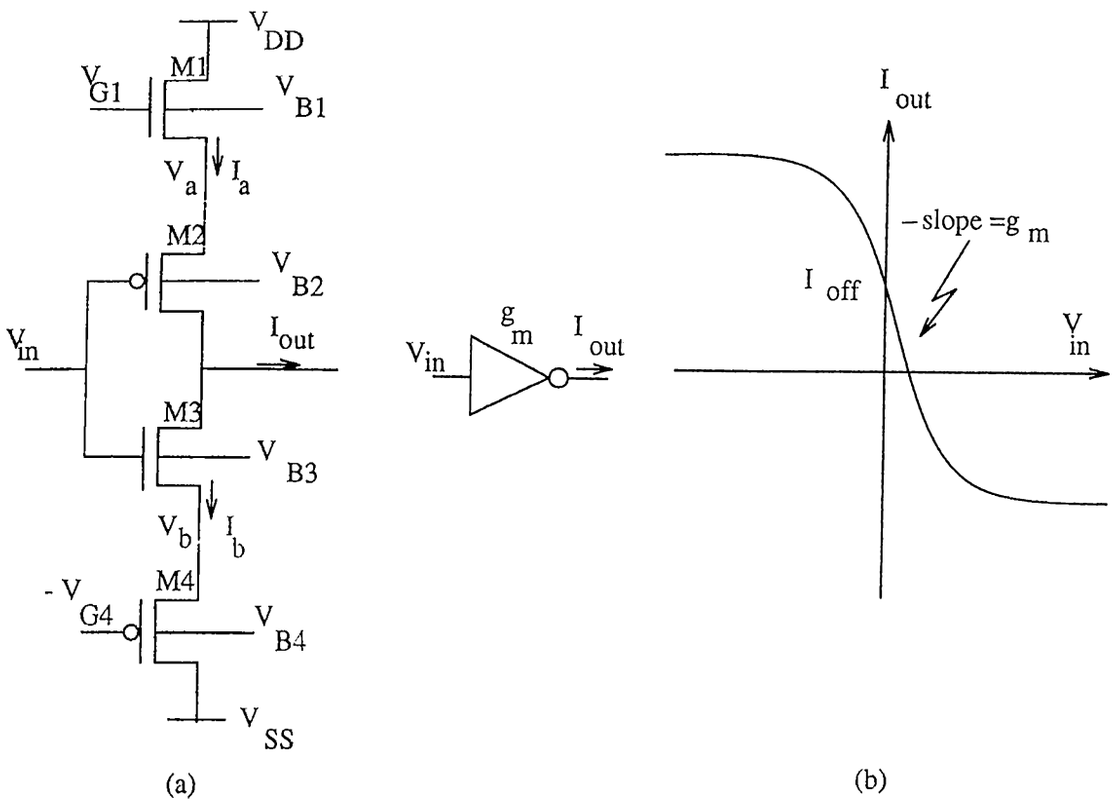


Figure 3.1: The transistor schematic (a) and the input–output characteristic (b) of CMOS transconductance element.

Thus with equations 3.1 and 3.2, the output current $I_{out} = I_a - I_b$ equals

$$I_{out} = -g_m V_{in} + I_{off} \tag{3.5}$$

where the abbreviations

$$g_m = 2k_{eff}[V_{G1} + V_{G4} - (V_{Tn1} + V_{Tn3} + |V_{Tp2}| + |V_{Tp4}|)] \tag{3.6}$$

$$I_{off} = \frac{g_m}{2}[(V_{Tn3} - V_{Tn1}) + (|V_{Tp4}| - |V_{Tp2}|) + (V_{G1} - V_{G4})] \tag{3.7}$$

are introduced for the transconductance parameter and the offset current, respectively.

Although the offset current I_{off} is not equal to zero due to the body effect, it can be easily eliminated by an appropriate setting of V_{B4} in an n–well process and V_{B1} in a p–well process for $V_{G1} = V_{G4} = V_G$ and $V_{B2} = V_a$ where V_a is a node voltage as defined in Fig. 3.1.

For linear operation of the transconductance element, all the transistors must stay in their saturation region, as was assumed in the derivation of (3.5), i.e. the conditions $V_{DS} \geq V_{GS} - V_{Tn}$ for n–channel and $V_{SD} \geq V_{SG} - |V_{Tp}|$ for

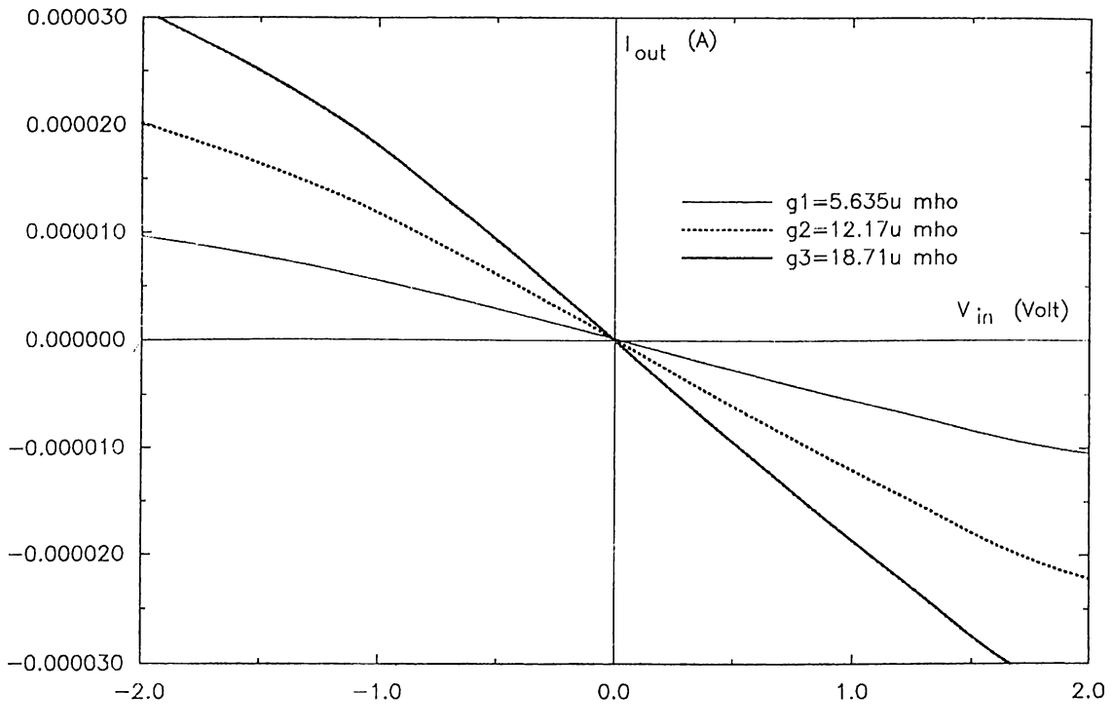


Figure 3.2: Simulation results of the transconductance element : $(W/L)_n = 0.38$, $(W/L)_p = 0.29$ for g_1,$(W/L)_n = 0.77$, $(W/L)_p = 0.57$ for g_2 and $(W/L)_n = 1.15$, $(W/L)_p = 0.85$ for g_3.

p–channel transistors must be satisfied. This leads the requirements

$$-|V_{Tp2}| \leq (V_{in} - V_{out}) \leq V_{Tn3} \tag{3.8}$$

$$V_{DD} \geq V_{G1} - V_{Tn1}, \qquad V_{SS} \leq -V_{G4} + |V_{Tp4}| \tag{3.9}$$

in addition to

$$-V_{G4} + V_{Tn3} + |V_{Tp4}| \leq V_{in} \leq V_{G1} - V_{Tn1} - |V_{Tp2}| \tag{3.10}$$

SPICE simulation results for different values of transconductace parameters, using 1.5–μ SPICE model parameters are shown in Fig. 3.2. In these simulations $V_{B1} = V_{B3} = V_{SS}$,$V_{G1} = V_{G4} = 3.5V$ and bulk voltage V_{B4} is chosen appropriately to eliminate the offset current defined in equation (3.7).

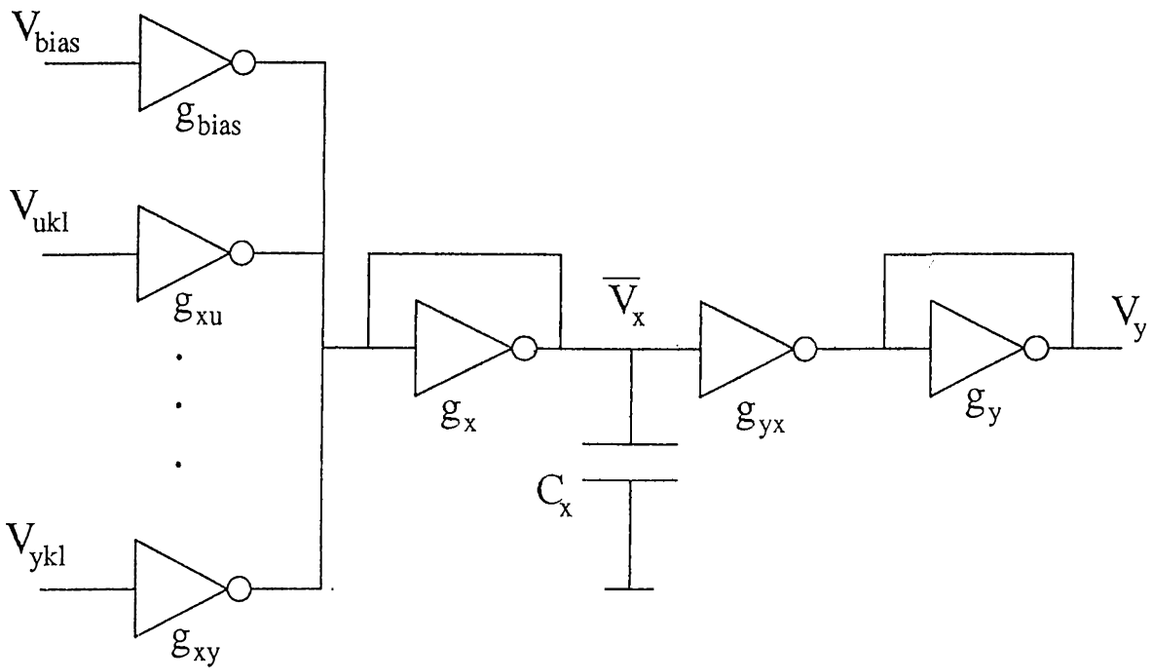
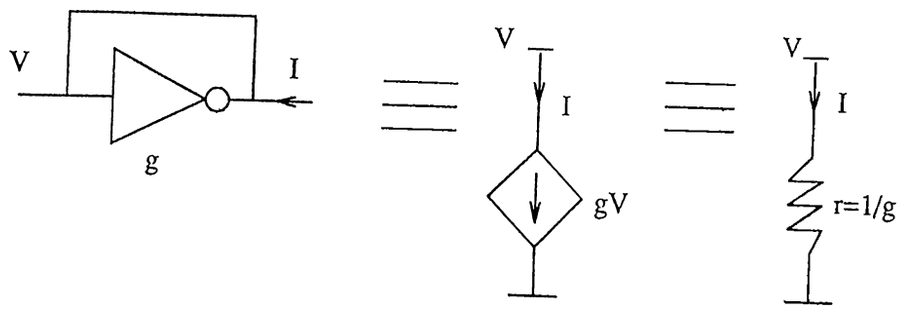


Figure 3.3: The cell circuit realization with CMOS transducers.

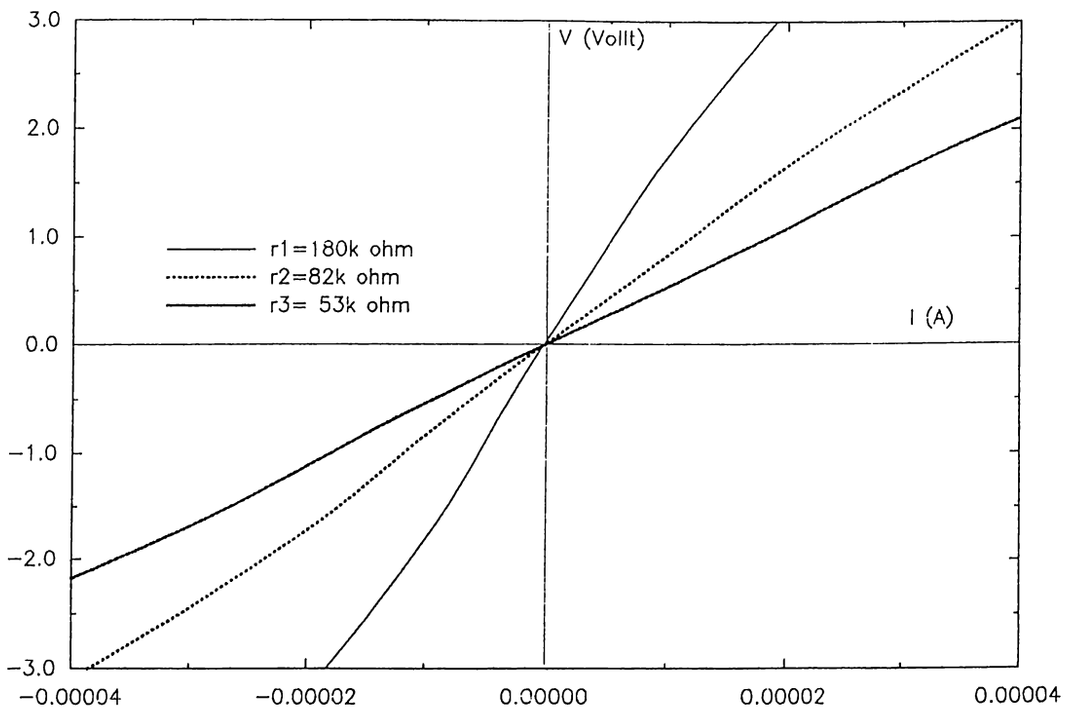
3.2 Realization of a Cell with CMOS Transconductances

Because of its simplicity, the CMOS transconductance element is chosen as the basic building block for the integrated circuit realization of the cell circuit. The circuit diagram of the integrated circuit realization of one CNN cell with CMOS transconductance element is shown in Fig. 3.3. It consists of the summation node, where all the input currents and the bias current are summed, the state and output resistors, the input and output voltage–controlled current sources and the sigmoid function.

The main problem in VLSI circuits is the implementation of the resistors that are not commonly used in standard CMOS technology. They usually occupy a large chip area which makes it impossible to implement networks with huge number of resistors. In order to eliminate this problem, we have implemented the cell circuit resistors R_x and R_y defined in Fig. 2.2, by simply connecting the input of the transconductance to its output as shown in Fig. 3.4.(a). The desired resistance values can be easily achieved by choosing the appropriate transconductance parameters. SPICE simulation results for different resistance values are shown in Fig. 3.4.(b).



(a)



(b)

Figure 3.4: The circuit diagram (a) and the characteristic (b) of a CMOS resistor for different resistance values.

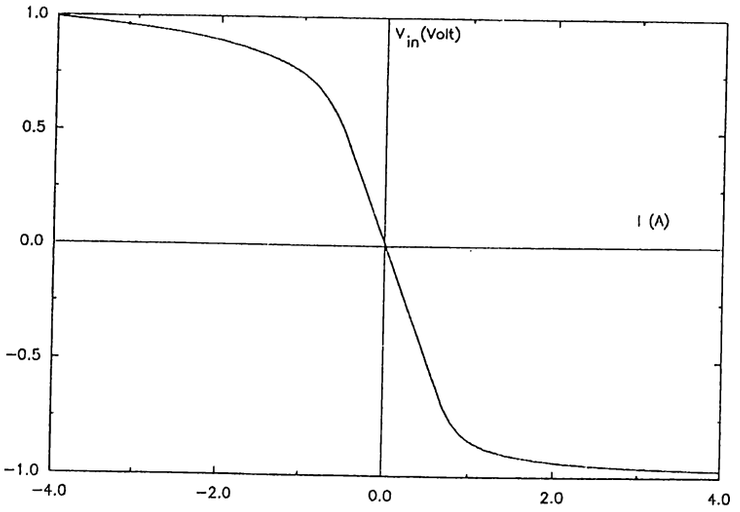


Figure 3.5: Output nonlinearity with CMOS transconductance elements.

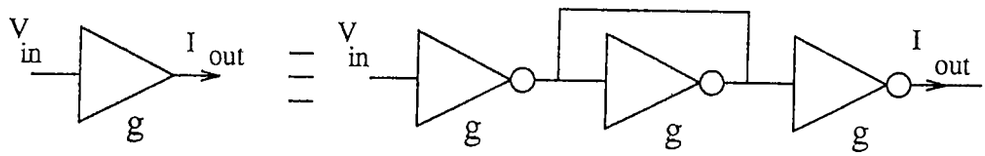


Figure 3.6: Inhibitory coupling coefficients with CMOS transconductance elements.

The sigmoid type nonlinear transformation, needed at the output of the cell circuit, is performed with two transconductance element as shown in Fig. 3.5. The second transconductance element whose input is connected to its output, acts like the resistor R_y and the output voltage v_y drives individual current sources whose outputs are coupled to the neighbors.

Fig. 3.2 shows the linear behavior of the transconductance parameter for $|V_{in}| \leq 1volt$. Since input voltages v_{uij} and the output voltages v_{yij} are bounded with $1volt$, the input control $B(i,j;k,l)$ and the output feedback $A(i,j;k,l)$ voltage–controlled current sources are obtained by using the transconductance elements in their linear region.

The desired coupling coefficient can be easily achieved by simply choosing the appropriate transconductance parameters. The negative (inhibitory) coupling coefficients can be obtained by inverting the positive (excitatory) input with a cascaded transconductance element pair as shown in Fig. 3.6.

By the following appropriate definitions it can easily be shown that the

realization shown in Fig. 3.3 satisfies the same state equation (2.5);

$$A(i,j;k,l) = g_{xy}(i,j;k,l) \tag{3.11}$$

$$B(i,j;k,l) = g_{xu}(i,j;k,l) \tag{3.12}$$

$$R_{xij} = g_{xij}^{-1} \tag{3.13}$$

$$R_{yij} = g_{yij}^{-1} \tag{3.14}$$

$$I = g_{bias}V_{bias} \tag{3.15}$$

$$v_{xij} = -\overline{v_{xij}} \tag{3.16}$$

In order to perform one or a related set of processing functions using fixed coefficients, the transconductance parameter variation can be achieved by choosing the gate-width-to-gate-length ratio (W/L) appropriately after setting $V_{G1} = V_{DD} = 5V$ and $-V_{G4} = V_{SS} = -5V$.

3.3 Programmable Coupling Coefficients

The requirement of adaptability is difficult to achieve in most neural network VLSI implementations [13],[22]. In the realization of the CNN structure with CMOS transconductance elements, we can achieve a programmable implementation by varying the transconductance parameters with external voltage sources connected to the gate voltages V_{G1} and V_{G4} of each cell defined in equation (3.6). SPICE simulation results of the transconductance parameter variation for different gate voltages is shown in Fig. 3.7. But one of the major problems in VLSI programmable implementations of CNN is the wiring needed for changing the template coefficients of each cell. Since all the cells of a cellular neural network have the same coupling coefficients $A(i,j;k,l)$ and $B(i,j;k,l)$, the wiring problem can be reduced by controlling the transconductance parameter variation of all cells with the same set of external voltages as shown in Fig. 3.8

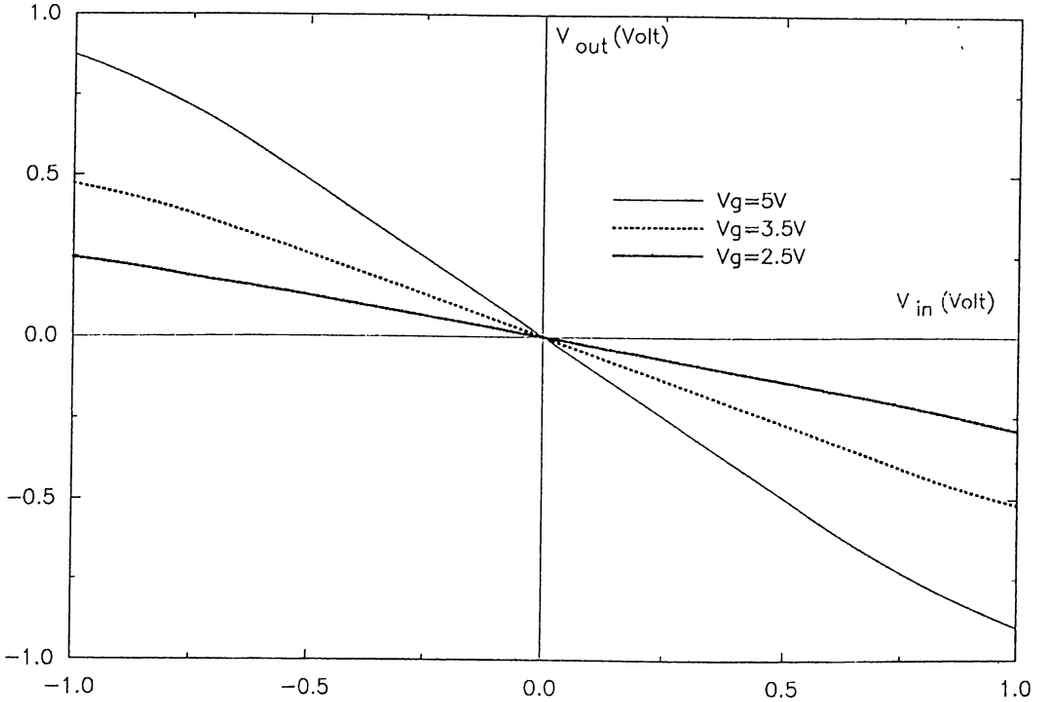


Figure 3.7: Simulation results of the transconductance parameter variation.

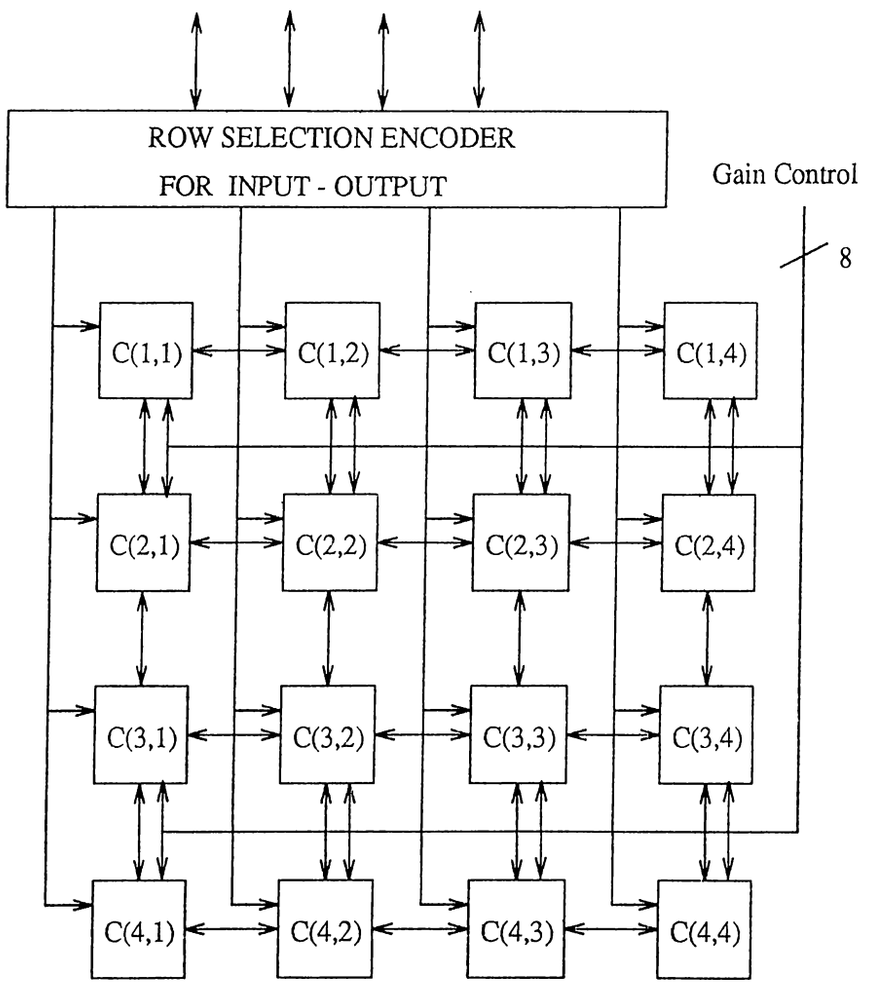


Figure 3.8: Programmable 4×4 CNN structure using CMOS transconductance elements.

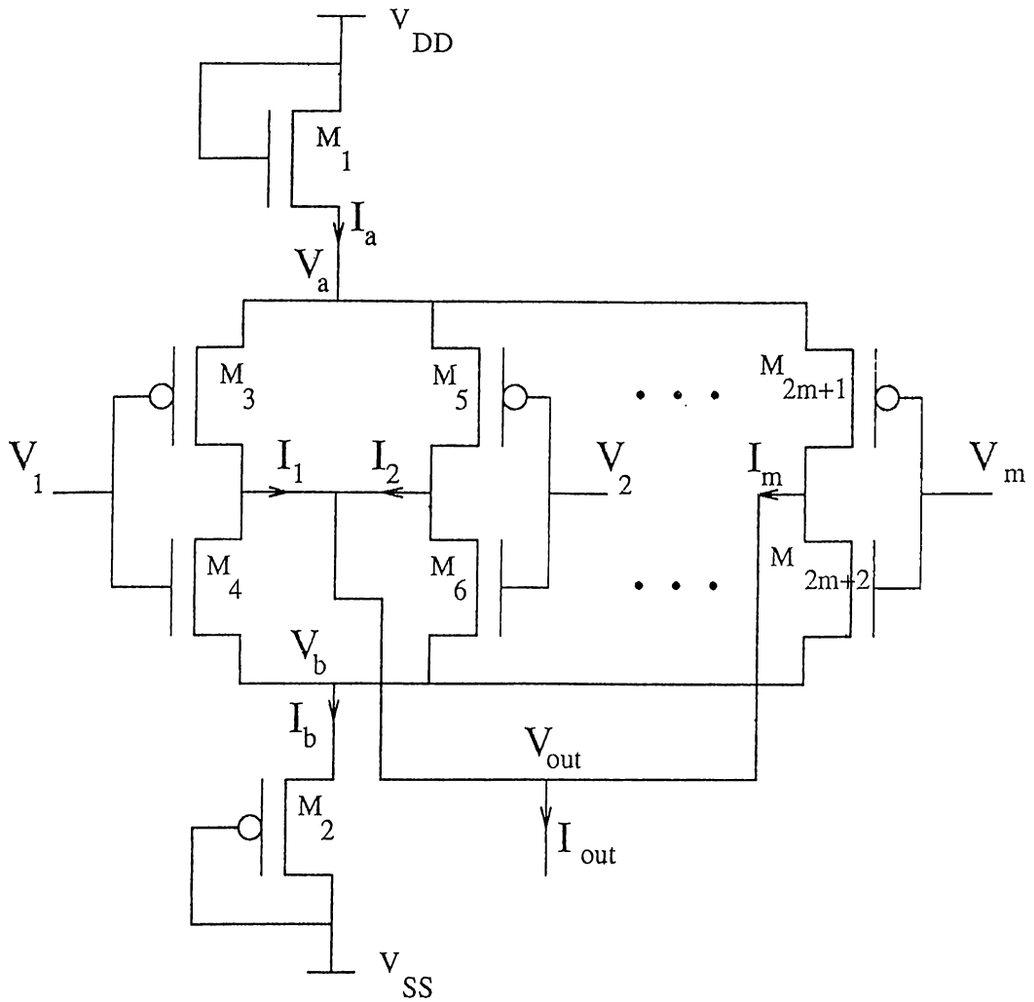


Figure 3.9: M–input voltage–controlled current source.

3.4 Reduced Hardware

In order to decrease the number of transistors in the realization of cellular neural network structure, we have presented a new type of multi–input voltage–controlled current source (VCCS) which is shown in Fig. 3.9. This multi–input VCCS is similar to m transconductance elements with common output, except upper NMOS and lower PMOS transistors are common.

In DC analysis of this new m–input voltage–controlled current source, assuming matching between NMOS transistor $M1$ and PMOS transistor $M2$ and using the standard square–law model for MOS transistors in their saturation region, the current I_{out} can be derived as

$$I_{out} = [-g_{eff} \sum_{i=1}^{m} w_i V_i + I_{off}][1 - \Delta] \tag{3.17}$$

where

$$g_{eff} = \frac{k}{2}[V_{DD} - V_{SS} - V_{Tn1} - |V_{Tp2}| - \sum_i w_i(|V_{Tp(2i+1)}| + V_{Tn(2i+2)})] \quad (3.18)$$

$$I_{off} = \frac{g_{eff}}{2}[V_{DD} + V_{SS} - V_{Tn1} + |V_{Tp2}| - \sum_i w_i(|V_{Tp(2i+1)}| - V_{Tn(2i+2)})] \quad (3.19)$$

$$\Delta = \frac{\{\sum_i w_i V_i^2 - [\sum_i w_i V_i]^2\}^2}{[-V_{DD} + V_{Tn1} + \sum_i w_i(V_i + |V_{Tpi}|)]^2[[-V_{SS} - |V_{Tp2} + \sum_i w_i(V_i - V_{Tni})]^2} \quad (3.20)$$

The conditions that should be satisfied in the derivation of equation (3.17) are

$$k = k_{n1} = k_{p2} = \frac{k_{n(2i+2)}}{w_i} = \frac{k_{p(2i+1)}}{w_i} \qquad i = 1, 2, 3, ..., m. \quad (3.21)$$

and

$$\sum_{i=1}^{m} w_i = 1 \qquad \text{and} \qquad w_i > 0.; \quad (3.22)$$

where k_{ni} and k_{pj} are the i^{th} NMOS and j^{th} PMOS transistor parameters which are defined in equation (3.4), w_i's are the scaling factors and m is the number of input voltages.

The inhibitory coupling coefficients can be obtained by inverting the positive (excitatory) input with a cascaded transconductance element pair shown in Fig. 3.5. It can be easily shown that, the maximum of the Δ term defined in equation (3.20) is less then 0.015, that is

$$\max_{|V_i| \leq 1, \sum w_i = 1} \text{percentage error} \leq 1.5\%. \quad (3.23)$$

The derivations of the output current and maximum percentage error are given in Appendix A.

Since the maximum error is less the 1.5%, this multi-input VCCS can also be used in the implementation of fixed function CNN structures. SPICE simulation results for two-input VCCS which defines the plane

$$I_{out} = g_{eff}(w_1 V_1 + w_2 V_2) + I_{off}$$

is shown in Fig. 3.10.

To obtain a perfect matching between k_{n1} and k_{p2} is a difficult task to achieve in practice, since the electron and hole mobilities μ_n and μ_p depend

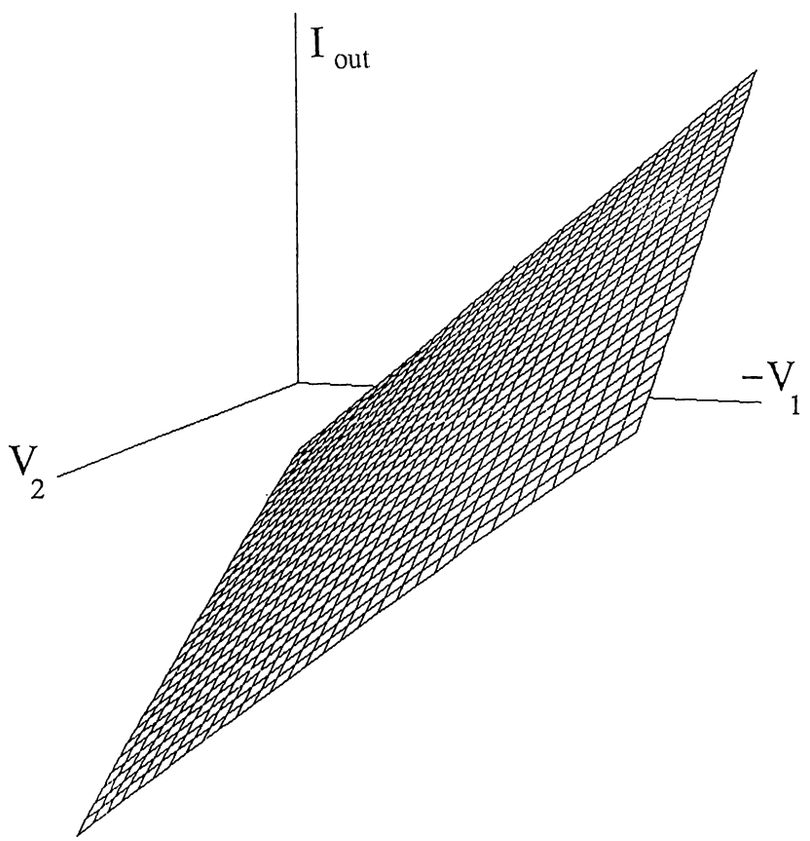


Figure 3.10: Simulation results for w_1=0.6,w_2=0.4 and g_{eff}=7e−5℧.

on doping, bias voltages and temperatures. Therefore, to see the effects of mismatching between k_{n1} and k_{p2} on this new multi-input voltage-controlled current source, a SPICE simulation for the case $k_{n1} = 0.75k_{p2}$ is performed and the result, that is shown in Fig. 3.11, again has defined a plane similar to the previous one (Fig. 3.10) showing that matching between k_{n1} and k_{p2} is a tolerable requirement.

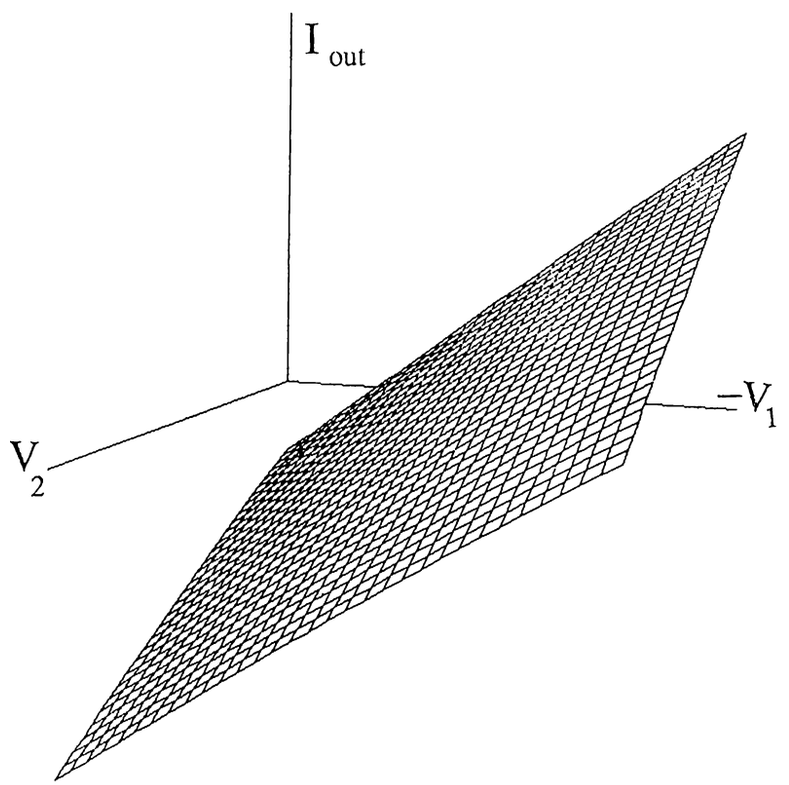


Figure 3.11: Simulation results for $k_{n1} = 0.75k_{p2}$.

Chapter 4

Simulations

One of the important problems in image processing is image segmentation. In image segmentation, each pixel of the image is classified into two or more classes. From the mathematical point of view, pixel classification can be considered as a map F, which maps a continuous vector space into a discrete vector space as defined below:

$$F : [a, b]^{M \times N} \rightarrow \{A, B, C, ...\}^{M \times N} \tag{4.1}$$

where $M \times N$ is the number of pixels in an image and $A, B, C, ..$ stand for different classes. For cellular neural network applications, we wish to assign to each pixel in the array one of the two values -1 and 1 based on some classification rules and the original pixel values. So F is defined by

$$F : [-1, 1]^{M \times N} \rightarrow \{-1, 1\}^{M \times N} \tag{4.2}$$

In the following sections, some examples of this kind of image transform performed by our analog CMOS realizations of cellular neural networks is presented. In the simulations of these networks, the circuit simulator SPICE2g6 is used. We have developed a preprocessor software called "cnn2spice" to generate the input circuit files for SPICE2g6 automatically. This program generates $N \times M$ CNN circuit file according to the predefined cell circuit and the input images are presented to the network as a set of initial conditions to the state capacitors. We have also developed two postprocessors, called "spice2plot" and "plot", which map the standard outputs of SPICE2g6 between 0 and 255 uniformly and then feed them to a color graphics terminal as gray level images.

In our CNN implementations, a large state capacitor value of $10pF$ is chosen in order to dominate the dynamics of the cell. Although this may slow the circuit down somewhat (most of the simulations have settling time about $4\mu sec$), it makes the circuit less sensitive to parasitic capacitances and resistances that can occur in any fabricated circuit.

In 12×12 noise–removing and 8×8 edge detection CNN circuit simulations, in order to reduce the memory requirement by computer, the four–transistor transconductance elements was first simulated using 1.5–μ SPICE model parameters and then its characteristics was modeled with 10^{th} order polynomials by minimizing the sum of squared deviations [23]. Finally, the simulation of the CNN architecture is performed using these 10^{th} order polynomials as nonlinear voltage–controlled current sources in the SPICE input file.

4.1 Line Detection

Although line detection is a very simple example image processing problem, it gives some intuitive ideas on how to design a Cellular Neural Network for solving a practical image processing problem.

For the "horizontal line detector" circuit, a very simple dynamic rule is chosen. The circuit element parameters of the cell $C(i,j)$ are as follows :

$$\begin{aligned} &C = 10pF, \quad I = 0, \quad R_x = g_x^{-1} = 180k\Omega, \quad B(i,j,k,l) = 0, \\ &A(i,j,i-1,j-1) = A(i,j,i-1,j) = A(i,j,i-1,j+1) = 0, \\ &A(i,j,i,j-1) = g_x, \quad A(i,j,i,j) = 2g_x, \quad A(i,j,i,j+1) = g_x, \\ &A(i,j,i+1,j-1) = A(i,j,i+1,j) = A(i,j,i+1,j+1) = 0. \end{aligned} \tag{4.3}$$

for a 3×3 neighborhood system. The feedback operator $A(i,j,k,l)$ are space invariant, that is $A(i,j,k,l) = A(i-k,j-l)$, therefore as in image processing filters, we can use a cloning template matrix to describe the feedback operator of the cell as follows :

$$A = \begin{bmatrix} 0 & 0 & 0 \\ g_x & 2g_x & g_x \\ 0 & 0 & 0 \end{bmatrix} \tag{4.4}$$

where the center entry of the cloning template correspond to $A(i,j,i,j)$; the upper left corner entry correspond to $A(i,j,i-1,j-1)$ and so forth. Since it is extremely convenient and clear to characterize the interactions of a cell with its neighbors by means of a cloning template matrix, we will use the cloning template matrix expressions in the following sections.

The dynamic equations of the cellular neural network corresponding to the above parameters are given by:

$$\frac{dv_{xij}(t)}{dt} = \frac{g_x}{C}[-v_{xij}(t) + v_{yij-1}(t) + 2v_{yij}(t) + v_{yij+1}(t)] \tag{4.5}$$

and

$$v_{yij}(t) = f(v_{xij}(t)) \qquad \text{for } 1 \leq i \leq 4, \quad 1 \leq j \leq 4 \tag{4.6}$$

where $f(\cdot)$ is a sigmoid type nonlinear function. The condition

$$A(i,j;k,l) > \frac{1}{R_x}$$

defined in Equation (2.12) is satisfied.

From the circuit Equations (4.5) and (4.6), it can be easily seen that the derivative of the pixel values depends on their left and right neighbors, but not on the upper and lower neighbors. This particular dynamic rule will therefore enhance the detection of horizontal lines in the original image.

As mentioned before, every cell in a Cellular Neural Network has the same connections as its neighbors. Therefore, the circuit equation of each cell is the same as those of the other cells in the same circuit. (Without loss of generality, the boundary effects are ignored). Hence, we can understand the global properties of a cellular neural network by studying the local properties of a single cell. This approach is extremely useful for analysis and design of cellular neural networks.

With the circuit parameters defined in Equation (4.3), a 4×4 cellular neural network is simulated using 1.5-μ model card in SPICE2g6 at the transistor level. From the first simulation results of this simple example, that is shown in Fig. 4.1, it can be easily seen that the *row* 3 stands as a black horizontal line with value 1.0 and flanked by white background with value -1.0. Hence the circuit is capable of extracting the horizontal lines in the given image in

upper left corner of Fig. 4.1. This simple example have shown that, CNN circuits can recognize and extract some special patterns from input images, by choosing the circuit parameters (i.e., the dynamic rule) appropriately.

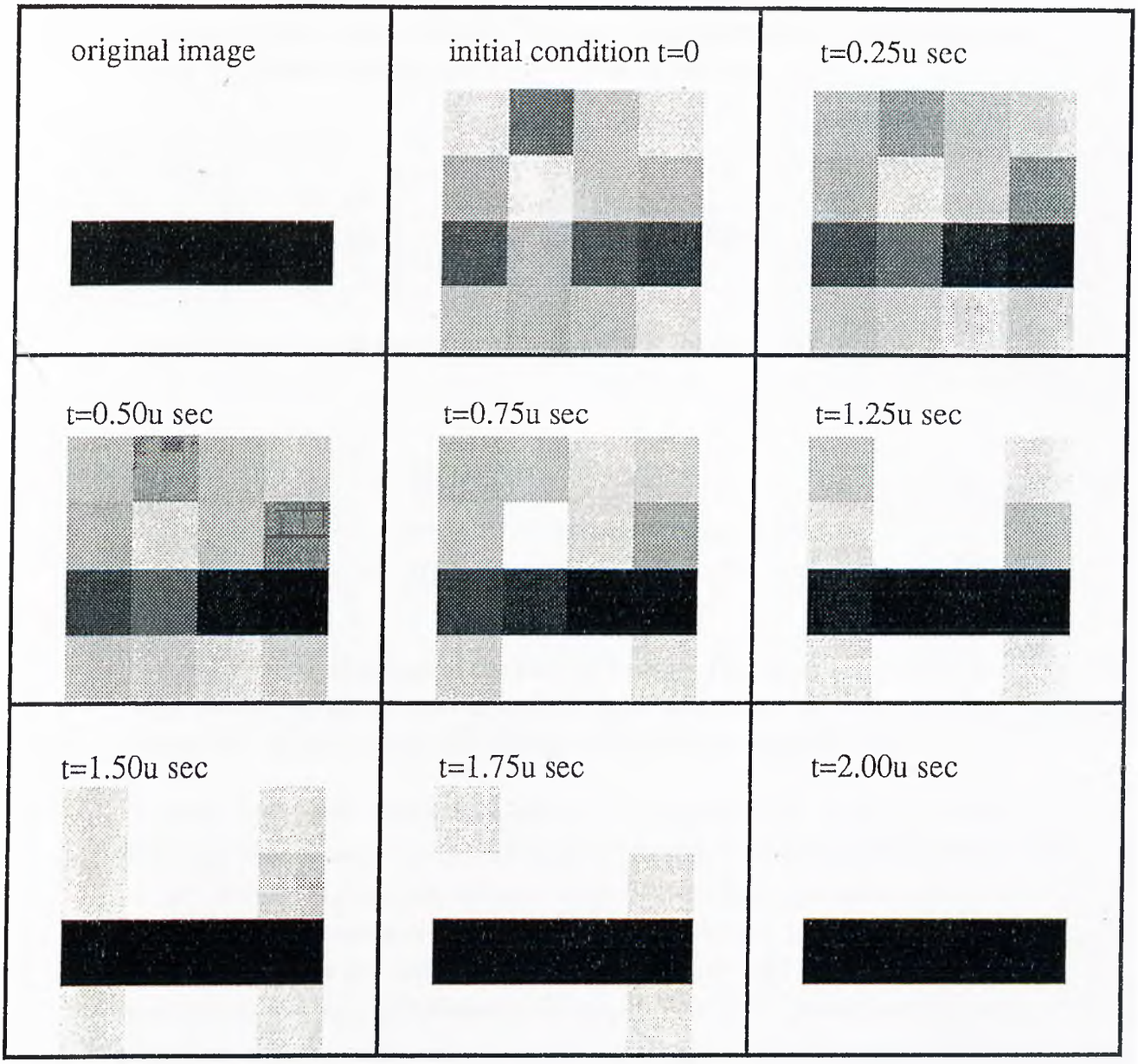


Figure 4.1: Simulation results for horizontal line detection.

4.2 Noise Reduction

One of the well-known techniques to reduce noise from the image is to use an averaging operator which corresponds to a lowpass filtering [12], [24]. Therefore, the averaging operator is chosen as the feedback operator of the noise-reducing cellular neural network. The circuit parameters of our implementation for 12 × 12 noise-reducing CNN are chosen as follows:

$$A = \begin{bmatrix} 0 & g_x & 0 \\ g_x & 2g_x & g_x \\ 0 & g_x & 0 \end{bmatrix}; \quad B = 0; \quad I = 0; \quad R_x = 180k\Omega; \quad C = 10pF; \qquad (4.7)$$

and the resulting cell circuit equations for the are given by

$$\begin{aligned} \frac{dv_{xij}(t)}{dt} &= \frac{g_x}{C}[-v_{xij}(t) + v_{yi-1j}(t) + \\ &\quad +v_{yij-1}(t) + 2v_{yij}(t) + v_{yij+1}(t) + v_{yi+1j}(t)] \\ v_{yij}(t) &= f(v_{xij}(t)) \qquad \text{for} \qquad 1 \leq i \leq 12, \;\; 1 \leq j \leq 12 \end{aligned} \qquad (4.8)$$

Since the rate of change of the state of cell $C(i,j)$ is approximately proportional to the average of the outputs of the neighborhood $N_1(i,j)$, the steady state of $C(i,j)$ depends on the average of those of its neighbor cells.

With the circuit parameters defined in equation (4.7), a 12 × 12 noise-removing cellular neural network circuit is simulated using the circuit simulator SPICE2g6. Figures 4.2, 4.3 and 4.4 show the three simulation results of 12 × 12 noise-removing cellular neural network circuit. In these figures, the noisy input images are defined at the upper left part and the rest of the pictures are the outputs at different time steps. This CNN circuit has the same properties as a two-dimensional low-pass filter. It retains the low-frequency components while eliminating the high-frequency components. In the spectrum of an image, the high frequency components contain information about the corners of objects. These high-frequency components are removed along with the high-frequency noise because of the low-pass filter effect. Therefore, the pixel classification is not always correct at the corners of the objects as seen in the Fig. 4.3.

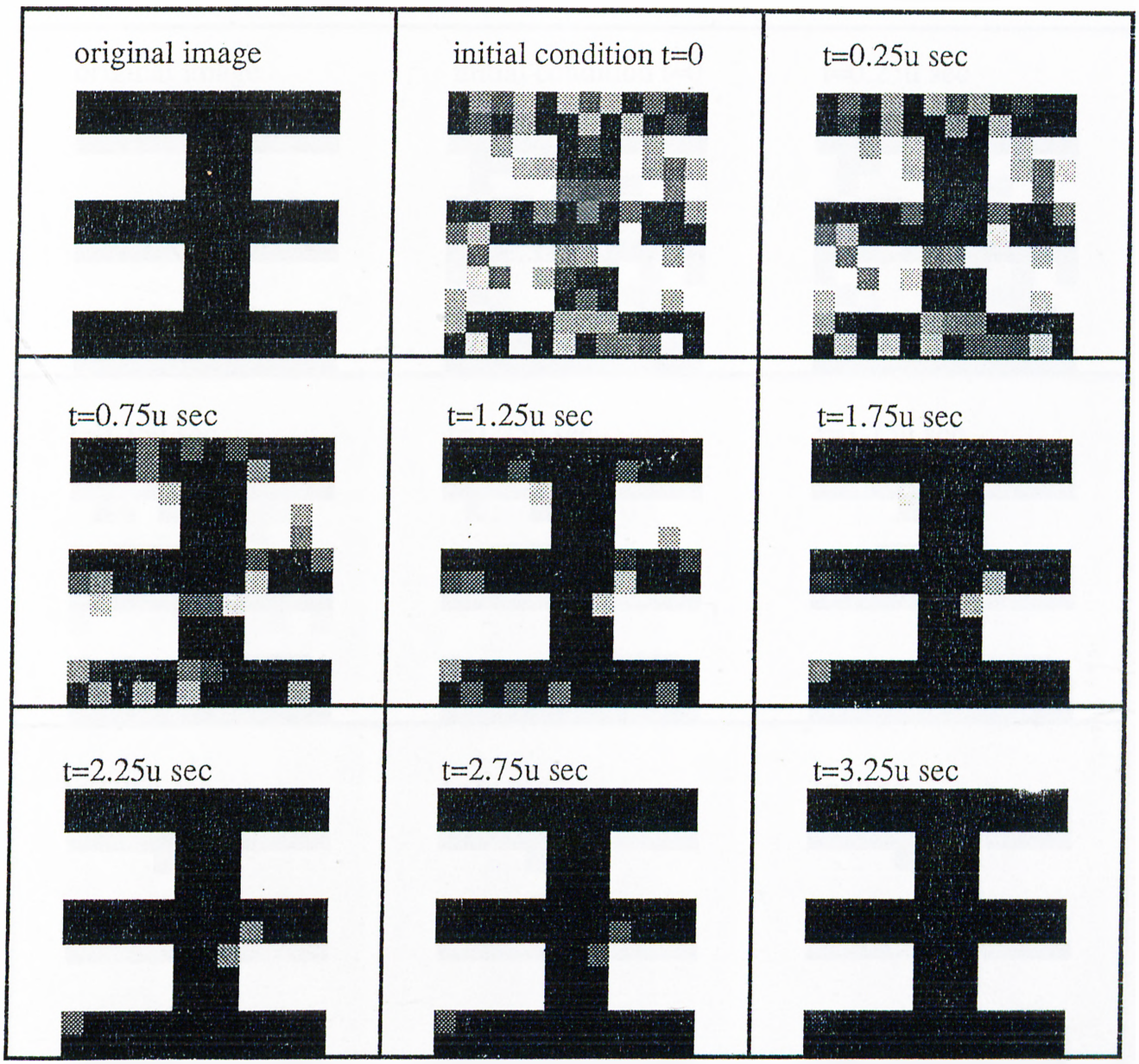


Figure 4.2: Simulation results of a noise removing cellular neural network.

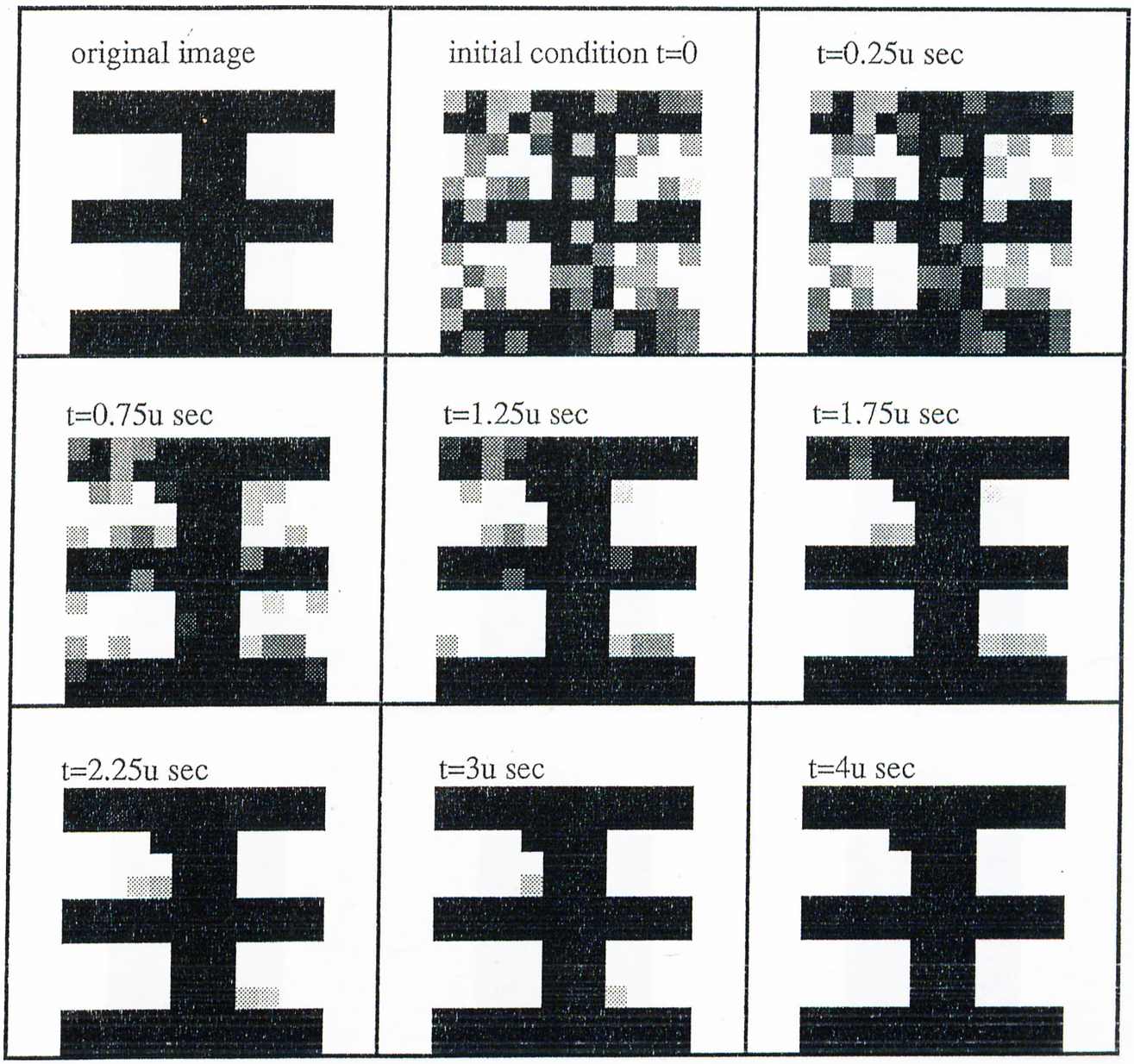


Figure 4.3: Simulation results of a noise removing cellular neural network.

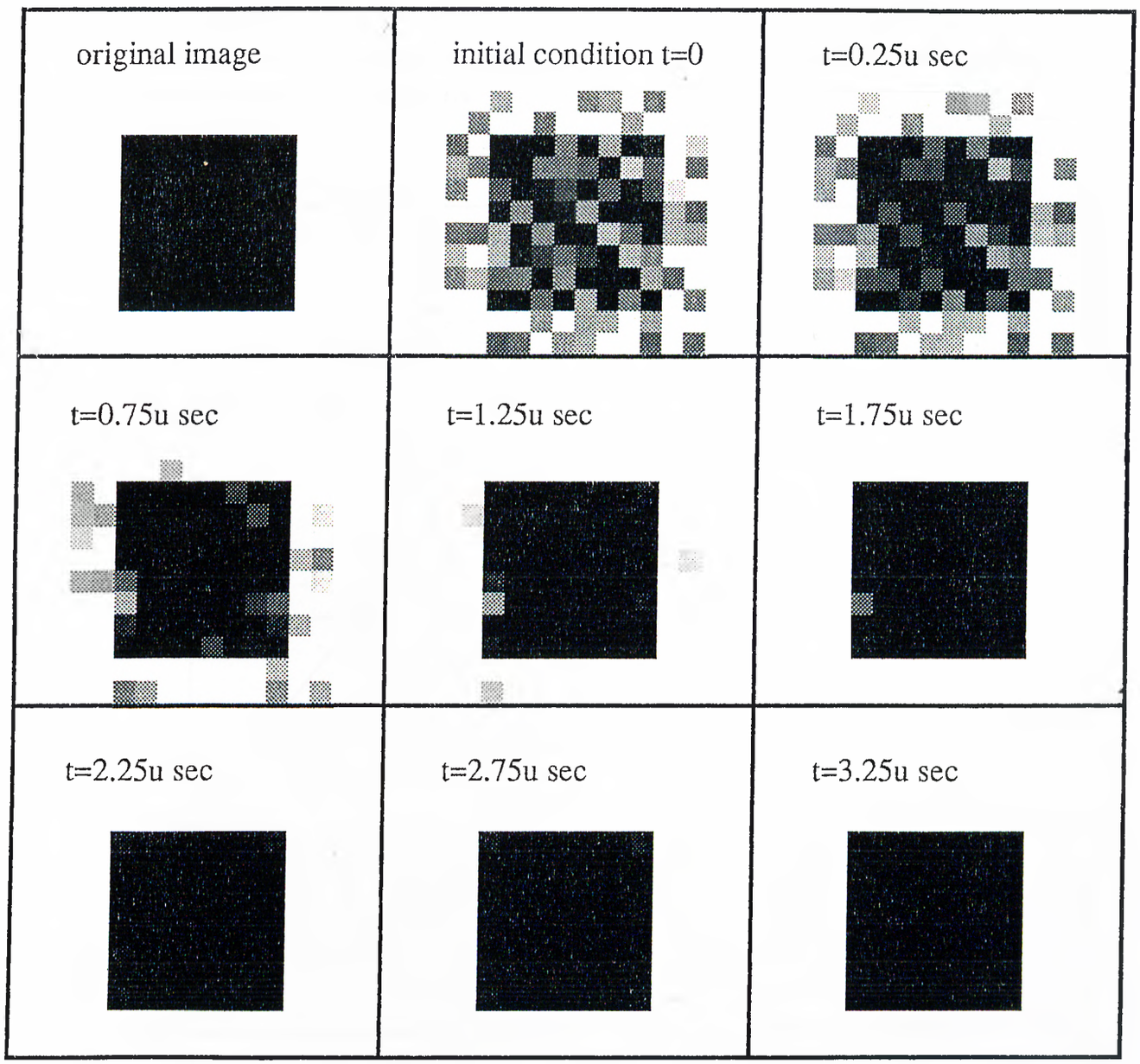


Figure 4.4: Simulation results of a noise removing cellular neural network.

To see the dynamic behavior of the circuit in more detail, the output transient characteristics of cells $C(1,7), C(7,2)$ and $C(8,2)$ are displayed in Fig. 4.5. These transient characteristics which are taken from the simulation results in Fig. 4.2, shows that the cell outputs reach their appropriate steady state values depending on both their neighbor cells and initial conditions.

From the simulation results, it can be seen that cellular neural network circuit defined with the circuit parameters in equation (4.7) is effective for removing noise in image processing, especially for images with large objects and few corners as in Fig. 4.4.

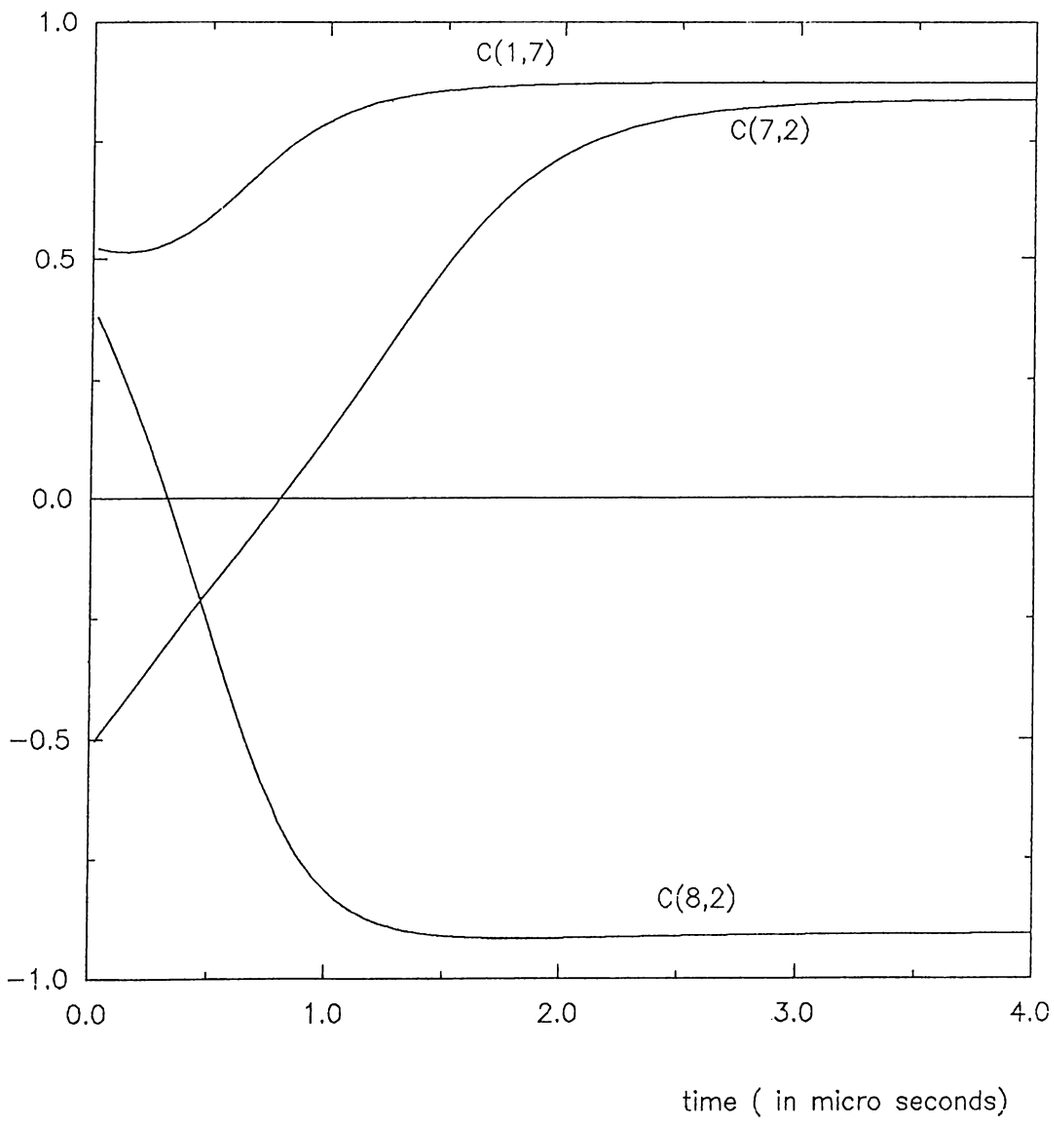


Figure 4.5: The transient response of cells $C(1,7), C(7,2)$ and $C(8,2)$.

4.3 Edge Detection

In this application, we have used another two-dimensional filter called *Laplacian operator* as the feedback operator to detect the edges of a square. The circuit parameters for this 8×8 cellular neural network circuit is defined below

$$A = \begin{bmatrix} 0 & -g_x & 0 \\ -g_x & 4g_x & -g_x \\ 0 & -g_x & 0 \end{bmatrix}; \quad B = 0; \quad I = 8\mu A; \quad R_x = 180k\Omega; \quad C = 10pF; \tag{4.9}$$

and

$$\begin{aligned} \frac{dv_{xij}(t)}{dt} &= \frac{g_x}{C}[-v_{xij}(t) - v_{yi-1j}(t) + \\ &\quad -v_{yij-1}(t) + 4v_{yij}(t) - v_{yij+1}(t) - v_{yi+1j}(t)] \\ v_{yij}(t) &= f(v_{xij}(t)) \qquad \text{for} \qquad 1 \leq i \leq 8, \;\; 1 \leq j \leq 8 \end{aligned} \tag{4.10}$$

The result of the circuit simulation is shown in Fig. 4.6. The parameter I in this example can control the derivatives of the state variables, and thus affects the dynamics of the circuit.

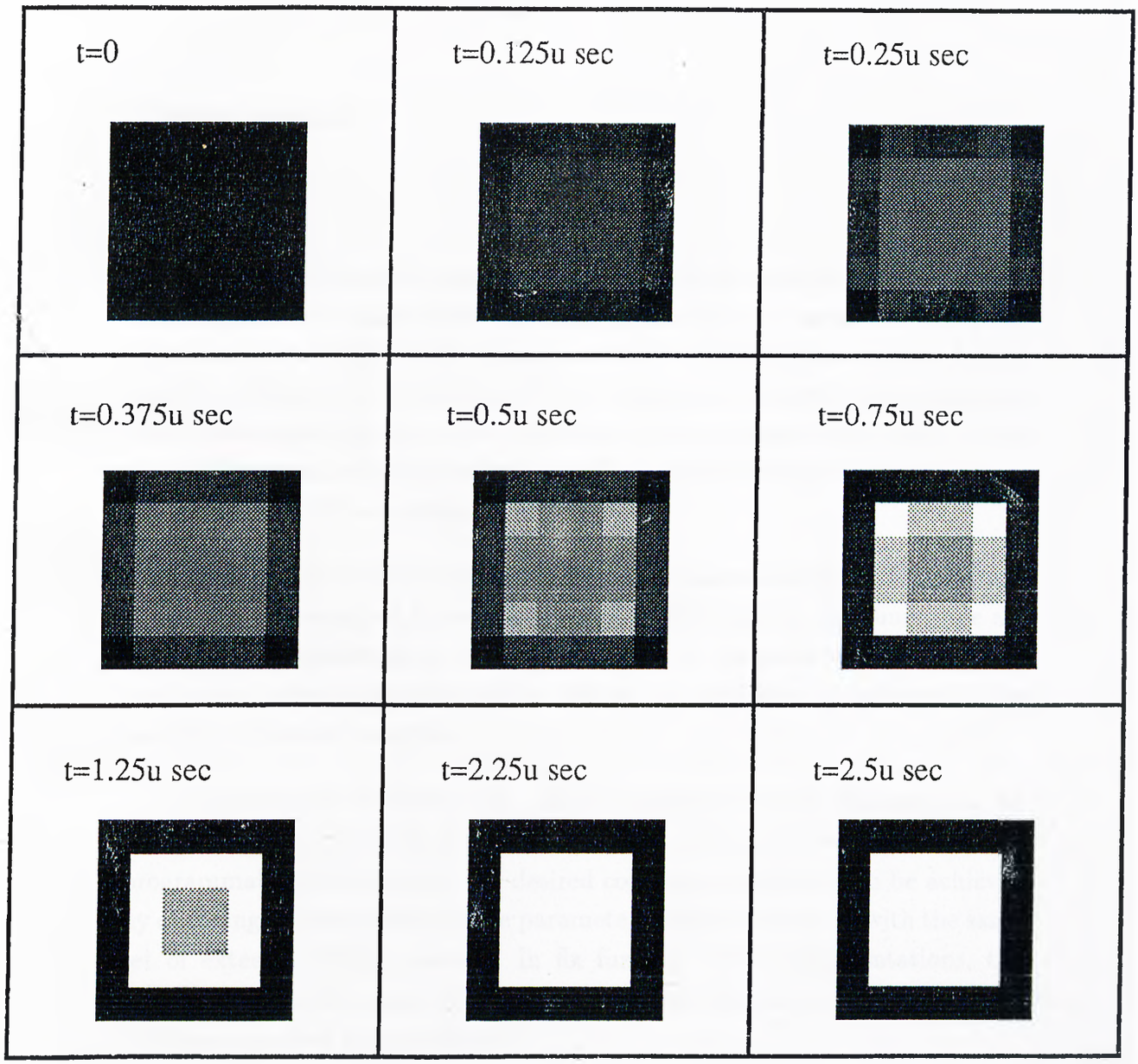


Figure 4.6: Simulation results of edge detection of a square.

Chapter 5

Conclusion

An analog CMOS circuit implementation of cellular neural network has been realized and some applications in image processing are presented. These applications show that cellular neural network implementation proposed here, can be used as a two–dimensional filter. Moreover, its parallel processing and continuous time features make it possible to process large–size images in real time. The estimated chip area for a 20×20–neuron CNN is about $4mm \times 4mm$ using 1.5μ CMOS technology.

The design of CNN circuits is reduced to a transconductance element and it can be easily adapted to various types of applications by just tuning the appropriate transconductance elements according to the predetermined coupling coefficients between the neighboring cells as it is performed in noise–removing and edge detection examples.

The realization of CNN with CMOS transconductance elements can be either programmable or fix function. In order to reduce the wiring problems in programmable CNN circuits, the desired coupling coefficients can be achieved by changing the transconductance parameters of each cell circuit with the same set of external voltage sources. In fix function CNN implementations, the number of transistors are reduced further by introducing a new multi–input voltage–controlled current source.

This analog CMOS realization of Cellular Neural Network can be integrated by optoelectronic sensors and/or charge–coupled devices to feed the input pattern to the neural processors.

Appendix A

Derivations of Multi–Input VCCS

For the m–input voltage–controlled current source that shown in Fig. 3.9, let

$$k = k_{n1} = k_{p2} = \frac{k_{n(2i+2)}}{w_i} = \frac{k_{p(2i+1)}}{w_i} \qquad i = 1, 2, 3, ..., m. \tag{A.1}$$

and

$$\sum_{i=1}^{m} w_i = 1 \qquad \text{and} \qquad w_i > 0.; \tag{A.2}$$

where k_{ni} and k_{pj} are the i^{th} NMOS and j^{th} PMOS transistor parameters which are defined in equation (3.4), w_i's are the scaling factors and m is the number of input voltages.

In DC analysis, using the standard square–law model for MOS transistors in their saturation region, the currents I_a and I_b, defined in Fig. 3.9, are easily derived as

$$I_a = \frac{k}{4}(\xi + \frac{\delta}{\xi})^2 \tag{A.3}$$

and

$$I_b = \frac{k}{4}(\eta + \frac{\delta}{\eta})^2 \tag{A.4}$$

where

$$\xi = -V_{DD} + V_{Tn1} + \sum_{i=1}^{m} w_i(V_i + |V_{Tp(2i+1)}|) \tag{A.5}$$

$$\eta = -V_{SS} - |V_{Tp2}| + \sum_{i=1}^{m} w_i(V_i - V_{Tn(2i+2)}) \tag{A.6}$$

and

$$\delta = \sum_{i=1}^{m} w_i V_i^2 - [\sum_{i=1}^{m} w_i V_i]^2 \tag{A.7}$$

Thus with equations (A.3) and (A.4), the output current $I_{out} = I_a - I_b$ equals

$$\begin{aligned} I_{out} &= \frac{k}{4}[(\xi + \frac{\delta}{\xi})^2 - (\eta + \frac{\delta}{\eta})^2] \\ &= \frac{k}{4}[\xi + \eta + \delta(\frac{1}{\xi} + \frac{1}{\eta})][\xi - \eta + \delta(\frac{1}{\xi} - \frac{1}{\eta})] \\ &= \frac{k}{4}[\xi + \eta][\xi - \eta][1 - \frac{\delta^2}{(\xi\eta)^2}] \end{aligned} \tag{A.8}$$

which implies

$$I_{out} = [-g_{eff} \sum_{i=1}^{m} w_i V_i + I_{off}][1 - \Delta] \tag{A.9}$$

where

$$\begin{aligned} g_{eff} &= \frac{k}{2}[V_{DD} - V_{SS} - V_{Tn1} - |V_{Tp2}| - \textstyle\sum_i w_i(|V_{Tp(2i+1)}| + V_{Tn(2i+2)})] \\ I_{off} &= \frac{g_{eff}}{2}[V_{DD} + V_{SS} - V_{Tn1} + |V_{Tp2}| - \textstyle\sum_i w_i(|V_{Tp(2i+1)}| - V_{Tn(2i+2)})] \\ \Delta &= \frac{\delta^2}{(\xi\eta)^2} \end{aligned} \tag{A.10}$$

The maximum percentage error can be written as

$$\begin{aligned} \text{max percentage error} &= \max\Delta \times 100 \\ &= \frac{\max|\delta|^2}{\min|\xi\eta|^2} \times 100 \end{aligned} \tag{A.11}$$

Since $|V_i| \leq 1$ and $\sum w_i = 1$

$$\begin{aligned} \max|\delta|^2 &= \max[\sum_{i=1}^{m} w_i V_i^2 - (\sum_{i=1}^{m} w_i V_i)^2]^2 \\ &= 1 \end{aligned} \tag{A.12}$$

and assuming $V_{T(n,p)i} \approx 1V$, $V_{DD} = -V_{SS} = 5V$

$$\begin{aligned} \min|\xi\eta|^2 &= \min[(\sum_{i=1}^{m} w_i V_i)^2 - 9]^2 \\ &= 64 \end{aligned} \tag{A.13}$$

Inserting (A.12) and (A.13) into (A.11), the maximum percentage error is be obtained as

$$\begin{aligned} \text{max percentage error} &\leq \frac{1}{64} \times 100 \\ &\leq 1.5\% \end{aligned} \tag{A.14}$$

References

[1] D.A. Anderson and E. Rosenfeld, (Eds.), "Neurocomputing", MIT Press, 1988.

[2] R. Echmiller and C. Maloburg, " Neural Computers", Springer–Verlag, 1988.

[3] J.J. Hopfield, " Neural Networks and Physical Systems with Emergent Collective Computational Abilities,"*Proc. Nat'l Academy Sci.*, Vol.79, Apr. 1982, pp 554-558.

[4] D. Rumelhart et.al., "Parallel Distributed Processing", MIT Press, 1986.

[5] T. Kohonen, " Self–Organization and Associative Memory,"Springer–Verlag, 1984.

[6] U. Rueckert and K. Goser, " VLSI Design of Associative Networks," *VLSI for Artificial Intelligence*, Kluwer Academic Publisher, pp227–235,1989.

[7] P. Treleaven, M. Pacheco and M. Vellasco, "VLSI Architectures for Neural Networks", *IEEE Micro*, pp.8-27, Dec. 1989.

[8] J.P. Sage, K. Thompson and R.S. Withers, "An Artificial Neural Network Integrated Circuit Based on MNOS/CCD Principles," *Proc. Neural Networks for Computing Conf.*, Vol. 151, pp 381–385, 1986.

[9] D.G. Hall, " Survey of Silicon–Based Integrated Optics," *Computer*, Vol.20, No.12, pp 26, 1987.

[10] A. Torinmi et.al., " A Study of Photon Emission from n–channel MOSFETS," *IEEE Trans. on Electronic Devices*, Vol.23, No.12, pp 55-60, 1986.

[11] L.O. Chua and L. Yang, "Cellular Neural Networks : Theory", *IEEE Trans. Circuits Syst.*, Vol. 35, no.10, pp. 1257-1272 , October 1988.

[12] L.O. Chua and L. Yang, "Cellular Neural Networks : Applications", *IEEE Trans. Circuits Syst.*, Vol. 35, no.10, pp. 1273-1290 , October 1988.

[13] L. Yang, L.O. Chua and K.R. Kried, "VLSI Implementation Of Cellular Neural Networks", *Proc. IEEE ISCAS*, pp. 2425-2427, 1990.

[14] V. Vemuri,"Artificial Neural Networks", Computer Society Press, 1988.

[15] H.P. Graf and L.D. Jackel, "Advance in Neural Network Hardware," *Proc. Int. Electron Devices Meeting*, 1988.

[16] C. Mead, "Analog VLSI and Neural Networks," Addision-Wesley, 1989.

[17] T. Matsumoto, T. Yokohama, H. Suzuki, R. Furukawa, A. Oshimoto, T. Shimmi, Y. Matsushita, T. Seo and L.O. Chua, "Several Image Processing Examples by CNN", *Proc. IEEE CNNA*, pp. 100-106, 1990.

[18] J. Denker (Ed.), " Computing with Neural Networks", American Institute of Physics, 1986.

[19] G. Serker, " Small Object Counting with CNN,*Proc. IEEE CNNA*, pp. 114-123, 1990.

[20] A. Dzielinski, S. Skoneczng and R. Zbikowski, "CNN Application to MOIRE pattern Filtering", *Proc. IEEE CNNA*, pp. 139-148, 1990.

[21] C.S. Park and R. Schaumann, "A High Frequency CMOS Linear Transconductance Element," *IEEE Trans. Circuits Syst.*, Vol. 33, no.11, pp. 1131-1138, November 1986.

[22] M. Verleysen and P.G.A. Jespers, "An Analog Implementation of Hopfield's N.N.", *IEEE Micro*, pp.46-55, Dec. 1989.

[23] P. Lancaster and K. Salkauskas, "Curve and Surface Fitting", Academic Press, 1986.

[24] A. Jain, "Fundementals of Digital Image Processing",Prentice–Hall.

[25] F. Zou, S. Schwarz and J.A. Nossek, "Cellular N.N. Design Using a Learning Algorithm", *Proc. IEEE CNNA*, pp. 73-81, 1990.

[26] K. Slot, "Determination of Cellular N.N. Parameters For Feature Detection of Two Dimensional Images", *Proc. IEEE CNNA*, pp.82-91, 1990.