CMOS ANALOGUE VLSI IMPLEMENTATION OF A KOHONEN MAP

THÈSE N° 1174 (1993)

PRÉSENTÉE AU DÉPARTEMENT D'ÉLECTRICITÉ

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES TECHNIQUES

PAR

PASCAL HEIM

Ingénieur électricien diplômé EPFL
originaire d'Oppens (VD)

acceptée sur proposition du jury:

Prof. E. Vittoz, rapporteur
Dr X. Arreguit, corapporteur
Prof. D. Del Corso, corapporteur
Prof. J.-D. Nicoud, corapporteur

Lausanne, EPFL
1993

– Oui, dit-elle lentement en anglais avec un fort accent germanique.

William BOYD

ACKNOWLEDGEMENTS

I wish to express my deepest appreciation to my research advisor Professor Eric Vittoz for his outstanding teaching and guidance during this work. He convinced me to undertake graduate studies at a time when finding a good job in the industry was a piece of cake. This was to all my satisfaction since I have seldom met someone with such broad technical knowledge, combined with a natural sense of humour. I hope many other students will have the opportunity to work with him.

I would also like to express my thanks to Professor Dante Del Corso, Professor Martin Hasler, Professor Jean-Daniel Nicoud and Doctor Xavier Arreguit for agreeing to be my referees.

I am most grateful to my colleagues of the *Laboratoire d'Electronique Générale* who made possible the most fruitful collaboration during these four years. In the lot are François Krummenacher, Gilles Van Ruymbeke, Christian Enz, Xavier Arreguit, George Wegmann, Vincent Peiris, Bertrand Hochet, François Dorel, Norbert Joël, Alessandro Mortara, Maher Kayal, Vlado Valencic, Mihaela Grigorie, Hussein Ballane, Tristan Reimann, Laurent Lemaître...

I must express particular thanks to Oscar Buset who checked the spelling of the whole manuscript, letting believe that I am a fluent English writer.

Thanks also to the permanent employees Danièle Gallay, Paul Debefve, André Decurnex, Joseph Guzzardi, Roland Jaques and Jean-Claude Zapf for their invaluable support in the Lab.

I would also like to express my gratitude to the *Fondation Suisse pour la Recherche en Microtechnique* (FSRM) which has supported the major part of the project (3 years and all the integrations and conferences), and Professor Michel Declercq, Director of the Lab, who allowed me to finish this work.

Last but not least, I would like to thank a good friend of mine, Carlos Andrade, who gave me substantial psychological support during this past year, while I was heavily disturbed by personal problems.

RÉSUMÉ

Le domaine des réseaux de neurones artificiels a rapidement évolué au cours des dix dernières années, et de nombreux groupes travaillent actuellement à la recherche de nouveaux algorithmes neuronaux ainsi qu'à leurs potentiels d'applications technologiques.

L'idée de s'inspirer de modèles biologiques pour réaliser des systèmes intelligents repose sur la constatation que les êtres vivants, de par leur adaptation à l'environement, ont évolué vers des structures robustes et fiables, tolérant les imperfections ou la destruction d'un certain nombre de leurs cellules. De plus, ces structures semblent particulièrement bien adaptées aux tâches de perception. Ces propriétés sont imputables à la grande redondance que leur confère leur parallélisme massif. De nombreux modèles ont été validés à l'aide d'ordinateurs, mais le fonctionnement séquentiel de ces derniers nécessite des temps de calcul prohibitifs. La réalisation de nouvelles architectures, permettant de disposer de supports matériels mieux adaptés au parallélisme des modèles, est freinée d'une part par la complexité de certains opérateurs dans le domaine du numérique, et d'autre part par l'abondance des interconnexions entre cellules, qui concerne aussi bien le domaine de l'analogique que du numérique.

L'objet de ce travail de thèse est la réalisation d'un réseau de neurones, à l'aide de technologies intégrées analogiques, afin d'évaluer les potentialités et les faiblesses de ce type de réalisations. Le réseau de Kohonen a été choisi comme

base de ce travail exploratoire à cause de sa relative simplicité. En effet, il s'agit d'un réseau non supervisé, ce qui simplifie ses interfaces avec l'extérieur. De plus, des solutions permettant de limiter le nombre de ses interconnexions étaient connues, qui permettent de s'affranchir des limitations inhérentes aux supports planaires des technologies intégrées.

L'étude débute par un bref rappel de l'algorithme de Kohonen, suivi par la description d'une architecture adaptée à l'intégration du réseau au moyen de technologies CMOS analogiques. Avant d'aborder le développement des circuits nécessaires à l'intégration du réseau, les effets sur le comportement de l'algorithme de certaines imperfections inhérentes aux réalisations analogiques sont analysés qualitativement à l'aide de simulations. Cette étape permet de préciser les spécifications des circuits qui sont parfois très différentes de celles que l'on s'attend à trouver dans des domaines plus classiques de l'électronique analogique. Vient ensuite la description des différents circuits utilisés pour la réalisation d'un réseau complet. Un réseau non-linéaire, composé de transistors reliant les cellules voisines entre elles, définit la structure du réseau et permet de générer le voisinage d'apprentissage. Un circuit de type "Winner-Take-All" est utilisé pour sélectionner le neurone dont le vecteur synaptique est le plus proche du vecteur d'entrée. Un élément particulièrement important est la synapse. Celle-ci permet de mémoriser et de modifier selon la règle d'apprentissage l'information élémentaire appelée poids synaptique. La mémorisation à long terme d'une information sous la forme analogique nécessite des technologies spéciales (EEPROM) et la modification de cette information est lente et mal contrôlée. Afin de surmonter ce handicap, une mémoire moyen-terme avec une perte correspondant à 0,1% de la pleine échelle par seconde a été développée. Ce temps de rétention permet au réseau de fonctionner correctement sous apprentissage continu, et permet aussi une lecture periodique des poids synaptiques.

Tous les circuits développés sont analysés en fonction des exigences du réseau et la plupart ont été intégrés et mesurés. En particulier, les mesures de la synapse effectuées sur plusieurs circuits, confirment précisément les prévisions analytiques. Finalement, un circuit d'évaluation, comprenant quatre neurones avec chacun trois synapses, a été intégré dans le but de réaliser un réseau complet pouvant comprendre jusqu'à une centaine de neurones. Les mesures effectuées sur les chips démontrent la faisabilité du système, malgré quelques erreurs tactiques facilement corrigibles lors d'un redesign.

SUMMARY

The domain of artificial neural networks has evolved rapidly during the last decade, and many research groups are presently working on new neuronal algorithms and investigating their potential for technological applications.

The idea to use biologically inspired models to implement intelligent systems is issue from the fact that animals, through their adaptation to the environment, have evolved towards robust and reliable structures, well adapted to the imperfections or even the destruction of some of their cells. In addition, these structures are particularly well adapted to perception tasks. These properties arise from the large redundancy inherent to their massive parallelism. Many models have been validated with computers, but the sequential operation of the latter leads to prohibitive computing time. The implementation of new architectures, leading to hardware that is better suited to the parallelism of the models, is slowed both by the complexity of some digital operators, and by the huge number of interconnexions between cells in the analogue and digital domains.

The goal of this thesis is the implementation of a neural network, using analogue integrated technologies, to evaluate the potential and the weaknesses of such implementations. The Kohonen network has been chosen as a basis for this exploratory work because of its relative simplicity. As a matter of fact,

this is a non-supervised network, which greatly simplifies the interfaces with the outside world. Furthermore,methods for limiting the number of interconnexions were known, thus overcoming the inherent limitations of intrinsically two-dimensional VLSI technologies.

The study begins with a brief recall of the Kohonen algorithm, followed by the description of an architecture adapted to the integration of the network by means of standard CMOS analogue VLSI technologies. Before looking at the design of circuits needed to implement the network, the effects of some inaccuracies inherent to analogue circuits on the behaviour of the algorithm are analysed qualitatively by means of simulations. This analysis is needed to set up the specifications of the circuits, which may sometimes be quite different from the specifications that are encountered in more classical domains of analogue electronics. Then, the various circuits used in the implementation of the network are described. A nonlinear network, made of transistors connecting the nearest neighbour cells, defines the topology of the network and is used to generate the learning neighbourhood. A Winner-Take-All circuit is used to select the neuron whose synaptic vector is closest to the input vector. The most important element is certainly the synapse. The latter memorizes and updates, according to the learning rule, the elementary information called synaptic weight. Long term storage of an analogue value requires special technologies (EEPROM) and the update of this value is slow and badly controlled. To overcome this drawback, a medium term memory has been developed that has a leakage corresponding to 0.1% of full scale per second. This retention time is sufficient to operate the network under continuous learning, and also sufficient to periodically read the synaptic weights.

All the proposed circuits are analysed with respect to the requirements of the network, and most of them have been integrated and measured. In particular, measurements of the synapse, made on several chips, are in good accordance with the analytical previsions. Finally, an evaluation chip, including four neurons with three synapses each, has been integrated. This chip can be used to build a complete network containing up to a hundred or so neurons. Measurements of single chips demonstrate the feasibility of the system, despite some tactical errors that can be easily corrected for an eventual redesign of the chip.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

The Kohonen map belongs to the large family of artificial neural networks. The concept of a formal neuron was first proposed by McCulloch and Pitts [1] in the early 1940s to try to model the operation of the nervous system. However, activity in the field really started in the 1980s, and has been increasing exponentially during the last decade.

Among the various algorithms that have emerged to date, Kohonen [2] proposed the self-organizing map, also called Kohonen map or network, which is inspired by the meaningful order of information that can be found in the brain. The algorithm executes an unsupervised self-organizing process that exhibits very interesting pattern classification and clustering properties. These networks offer attractive solutions to some complex problems in fields like speech recognition, pattern classification [2] and robotics [3].

Neural networks in general have often been credited with inherent fault tolerance thanks to their massively parallel mode of computation. This suggests that analogue circuits should lead to more efficient implementations in terms of speed and density than their digital counterparts. Although this assumption has been widely verified for particular sensor-driven networks such as

artificial retinas [4] [5] or optical motion sensors [4], there is no evidence that it is true for the more general class of neural networks used for pattern classification. Indeed, since the algorithms are first developed for software implementations due to the absence of dedicated hardware, there is no evidence a priori that they are suited to analogue VLSI implementations. Furthermore, software implementations simply avoid the problem of the large number of interconnexions faced by VLSI implementations.

Among the wide class of neural networks used for pattern classification, the Kohonen map appears to be a good candidate for an exploratory analogue VLSI implementation. The algorithm is rather simple and not supervised, which limits the complexity of both the circuits and the interfaces. In addition, the interconnexion problem has already been investigated [6], and solutions that are compatible with the intrinsically two-dimensional VLSI processes have been proposed.

The primary objective being pursued in this work is to investigate the potential of analogue circuits for implementing a fully autonomous Kohonen map. The goal is to develop a feasible implementation, either by taking advantage of the real collective properties of the network, or by finding adapted circuits whose weaknesses are tolerated by the algorithm. The performance of the resulting implementation and the way in which they have been attained should provide better insight into the real nature of the Kohonen map. It should be pointed out that this work put strong emphasis on circuit design approaches involved in the field of neural networks.

This work is organized as follows:

Chapter 2 summarizes the basic principles of the Kohonen network, from the formal neuron to the simplified computational algorithm. Some qualitative aspects concerning the learning parameters are also developed. It is shown that the use of a particular learning neighbourhood can improve the organization phase of the process and facilitates the choice of the temporal evolution of the learning parameters.

Chapter 3 describes the general architecture of the network and discusses the possible analogue implementations of the different blocks. The problem of multichip implementations is analysed, and a suitable general scheme is proposed. Finally, the adopted architecture is described and gives an overview of the contents of this work.

Chapter 4 deals with the effects of circuit inaccuracies on the algorithm. This is an important part of the work that investigates the accuracy requirements of the algorithm, so that the circuit specifications are clearly defined. It is shown that, in some aspects, the accuracy requirements may be compared with those of D/A converters, and guide-lines are given for the choice of an economically acceptable size of network.

Chapter 5 proposes a nonlinear diffusion network, inspired from the linear resistance-conductance (RG) network proposed by Mead [4]. This nonlinear network is well adapted to generate the learning neighbourhood in implementations based on the simplified algorithm.

Chapter 6 recalls the principle of the Winner-Take-All circuit proposed by Lazzaro [7]. The static analysis is reviewed for its current mode of operation, and it is shown that better performance can be attained in strong inversion. A modified version of the WTA is also proposed, which is compatible with multichip implementations, and can handle larger bus capacitance.

Chapter 7 proposes a new offset-insensitive current-mode rectifier that has been developed especially to fit some requested specifications of the network.

Chapter 8 proposes a precise analogue synapse with learning capability that behaves in accordance with the simplified computational algorithm. A test chip is also described, and measurements on the chip confirm the theoretical analysis

Chapter 9 describes a chip that contains four neurons with three synapses per neuron. The chip can be used to build a larger two-dimensional network.

Chapter 10 provides summarizing remarks and conclusions.

Appendix A summarizes the MOS transistor model used in this thesis for hand calculations. It contains the basic definitions that are usually not repeated in the chapters. It is therefore advised to read this appendix before the chapters concerned with MOS circuits.

Appendix B presents two biasing techniques under voltage reference control that are needed for a multichip implementation of the network. The first fixes the pinch-off voltage of the transistors and is required for the

proper operation of the nonlinear diffusion network. The second fixes the same saturation voltage of differential pairs independently of the technological parameters.

Appendix C describes a low-voltage cascode mirror in strong inversion.

Appendix D summarizes the technological parameters of the analogue CMOS SACMOS 3μm technology that has been used to implement the chips.

References

[1] W. S. McCulloch and W. A. Pitts: "A logical calculus of the ideas immanent in nervous activity", Bulletin of Mathematics and Biophysics, 5, 1943, pp. 115-133.

[2] T. Kohonen: "Self-organization and associative memory", Springer Verlag, Berlin, 1988.

[3] T. M. Martinetz, H. J. Ritter and K. J. Schulten: "Three-dimensional neural net for learning visuomotor coordination of a robot arm", IEEE Transactions on Neural Network, 1, No 1, 1990, pp. 131-136.

[4] C. Mead: "Analog VLSI and Neural Systems", Addison-Wesley, Reading, 1989.

[5] M. A. Mahowald: "VLSI Analogs of Neuronal Visual Processing: A Synthesis of Form and Function", Ph.-D. Thesis, Caltech-CS-TR-92-15, 1992.

[6] E. A. Vittoz, P. Heim, X. Arreguit, F. Krummenacher and E. Sorouchyari: "Analog VLSI Implementation of a Kohonen Map", Journées d'électronique 1989, Lausanne, 1989.

[7] J. Lazzaro, S. Ryckebusch, M. A. Mahowald and C. A. Mead: "Winner-Take-All Networks of Order N Complexity", in Proc. 1988 IEEE Conf. on Neural Information Processing - Natural and Synthetic, Denver, 1988, pp. 703-711.

CHAPTER 2

THE KOHONEN NETWORK

2.1 INTRODUCTION

This chapter is used to recall the basic principles of the Kohonen network that are necessary to understand the contents of this document. For a more detailed description, the reader can consult the Kohonen's well-known reference [1]. The network is described from the formal neuron to the most simplified algorithm which may be used for computer simulations or hardware implementations. It is shown that the biologically inspired structure is made of spatially organized formal neurons totally interconnected through a lateral feedback function which has the form of a mexican hat. This interconnecting function, together with an adaptive process, lead to the process of self-organization. The ability of the network to map multidimensional data into a meaningful order of its processing units is commonplace in the brain where information is compressed into reduced representations of the most relevant facts, without loss of knowledge about their interrelationships [1]. The original "neuromorphic" structure is then replaced by a more classical structure best suited to simulations or hardware implementations. Finally, the effects of the parameters settings (adaptation gain, radius of the neighbourhood and type of distance measurement for the selection of the winner) on the network behaviour are analysed qualitatively.

2.2 THE FORMAL NEURON

The symbol used to represent a single neuron i is shown in Figure 2.1. An n-dimensional input vector $x_i = (x_{i1}, x_{i2},...x_{in})$ is applied to the neuron through the weighting elements $m_i = (m_{i1}, m_{i2},...m_{in})$ called synapses. The output activity of such a neuron is often described using the weighted sum of the above vectors and a nonlinear operator, and can be expressed as follows:

$$Y_i = \sigma\left(\sum_{j=1}^{n} m_{ij}\, x_{ij} - \theta_i\right) \tag{2.1}$$

In this relation, the sum represents the scalar product $x_i \cdot m_i$, θ_i is a threshold value and $\sigma(\cdot)$ is a sigmoid-like saturating function which models the nonlinear behaviour of the neuron. A particular choice of the parameters m_i and θ_i makes the neuron sensitive to input vectors belonging to a defined class. A set of several neurons with different parameters is thus able to make a classification of input vectors x (the index i is now omitted because the same vectors are input to all neurons). The fundamental problem now is finding an optimal choice for the synaptic weights m_{ij}. One possible answer to this problem is the self-organizing process that is implemented by means of the Kohonen network.

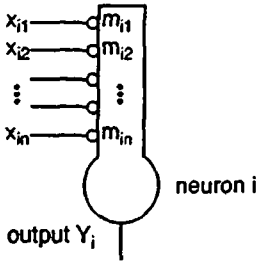


Fig. 2.1 Formal neuron.

2.3 NEUROMORPHIC STRUCTURE WITH LATERAL FEEDBACK

The classical Kohonen map is a set of neurons ordered as a one- or two-dimensional array that defines a fixed topological relationship between units. The structure used throughout this work is a two-dimensional

orthogonal array topology. A structure containing M neurons is shown in Figure 2.2 in its one-dimensional form for simplicity.

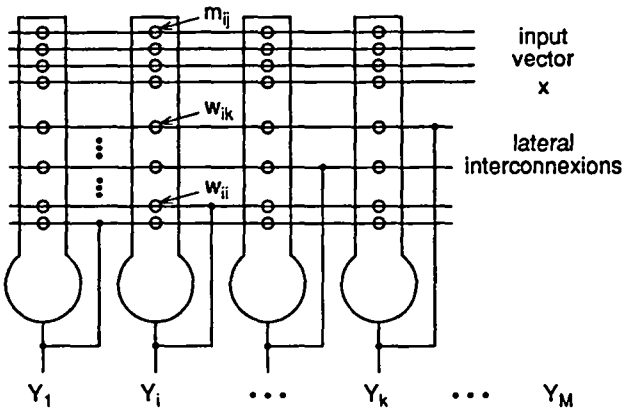


Fig. 2.2 Basic structure of the Kohonen network.

Each neuron receives the same input vector x at its own set of plastic synapses m_{ij}. The neurons also have a second set of M fixed-value synapses w_{ik} that links all the neurons together through a lateral interaction function. This function has the typical form of a mexican hat shown in Figure 2.3. It implements excitatory interactions at short distances, and inhibitory interactions at longer distances.

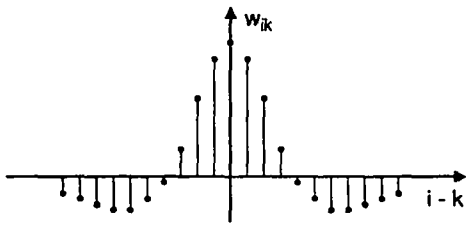


Fig. 2.3 "Mexican hat" lateral interaction function

When an input vector is presented to the network, each neuron computes an output activity Y_k which is a function of the similarity between its synaptic weights and the input vector. The lateral interaction function induces the above mentioned excitation/inhibition effect which results in a "clustering" of the

activity where the primary excitation has a maximum. The size of this cluster, which is also called "bubble", depends on the relative amplitude and spread of the excitatory and inhibitory areas and is stabilized by the saturated outputs of the neurons. Next, the weight vectors of the bubble's cells are adapted towards the input vector. The formation of the activity bubble in the most responding area of the network together with the sensitization of its cells to the input vector are responsible for the self-organizing process.

2.4 THE SIMPLIFIED COMPUTATIONAL ALGORITHM

For a large size two-dimensional network, the number M^2 of connections needed to implement the lateral interaction function is prohibitive. Although there are some methods to implement this function in connecting the nearest neighbours only, such as the use of linear resistive (RG) networks [2] or pseudo-linear networks [3] [4], it is worth recalling the simplified *computational algorithm* best suited to simulations or simple hardware implementations. It is stated in [1] that the formation of the activity bubble is the central phenomenon which is responsible for self-organization. Therefore, as the bubble is obviously formed where the map has its maximum activity, it is computationally advantageous to find the most responding unit and *define* the bubble around it. This new computational algorithm can be expressed in the discrete-time formalism by the following steps:

1. Initialize the synaptic weights with random values.
2. Select an input vector $x(t_k)$ from the data base.
3. Find the most responding unit c by means of an appropriate proximity or distance measurement.
4. Generate a learning neighbourhood $N_c(t_k)$ (bubble) centred around unit c.
5. Update the synaptic weights of the neurons belonging to $N_c(t_k)$ with the following rule:

$$m_{ij}(t_{k+1}) = m_{ij}(t_k) + \alpha(t_k)\left[x_j(t_k) - m_{ij}(t_k)\right] \qquad (2.2)$$

6. Back to number 2.

where t_k is an integer representing the discrete-time index or the number of iterations. The algorithm is greatly accelerated with a proper temporal evolution of the learning parameters $N_c(t_k)$ and $\alpha(t_k)$. At the beginning of the process, $N_c(t_k)$ may contain as much as half of the network's cells, and $\alpha(t_k)$ is close to 1. Then $N_c(t_k)$ is decreased until it includes the most responding unit with its nearest neighbours only, and $\alpha(t_k)$ is decreased to an arbitrarily small value depending on the expected degree of convergence. There are no exact sequences for these parameters and certain empiric rules may have to be determined by experience. The gain function of the learning neighbourhood is generally "cylindrical", and is characterized by its radius R and its "height" α. The temporal evolution of these parameters is often described with the following relations:

$$\alpha(t_k) = \frac{\alpha_0}{1 + t_k/\tau_\alpha} \tag{2.3}$$

$$R(t_k) = \frac{R_0}{1 + t_k/\tau_R} \tag{2.4}$$

where α_0 and R_0 are the initial values of the parameters and τ_α and τ_R their time constant. As $R < 1$, the bubble is reduced to the most responding unit c only and the network looses its ability to organize. This is acceptable only if the network is already organized when it happens, provided that the network does not have any inaccuracies that can affect its synaptic weights. If the computational precision is high (simulations) α can decrease to a very low value, thus increasing the accuracy of the final convergence phase. However, if inaccuracies, which are inherent in analogue hardware implementations, are taken into account, both α and R must be kept above a minimum value under which the network is no longer controllable. For this reason, the relations (2.3) and (2.4) will be practically limited to some minimum values α_{min} and R_{min}.

Simulation results of the network's mapping are presented using a lattice representing the topology of the map or, similarly, the neighbouring relations between cells. The nodes of the lattice represent the position of the weight vectors in the same coordinate system as the data base. As an example, Figure 2.4 shows the evolution of a map containing 12×12 neurons at

several learning iteration steps t_k. The parameters are $\alpha_0 = 0.8$, $\tau_\alpha = 12.5$, $R_0 = 8$ and $\tau_R = 1000$. The data base is limited to a triangle containing uniformly distributed vectors. The figure also includes the values of R and α for each iteration step t_k.

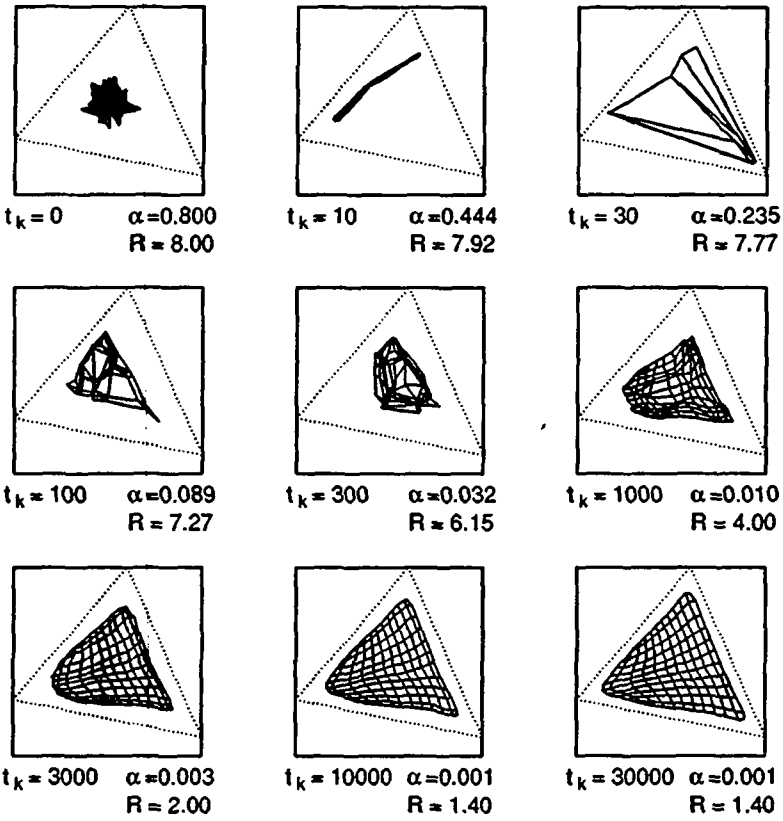


Fig. 2.4 Simulation using the standard "cylindrical" neighbourhood.

The formation of the map has two slightly different phases: the *organization phase* and the *final convergence phase*. It can be seen from the simulation that the organization phase is completed between $t_k = 1000$ and $t_k = 3000$ iterations, i.e. when the map looks totally unfolded and ordered. At the beginning of the process, the evolution of the map seems somewhat perturbed.

This is due to a non-optimal choice of the learning parameters, which can be avoided by using the modified neighbourhood proposed in the next paragraph. On the other hand, the triangular shape of the data base has been especially chosen to illustrate the ability of the map to reproduce the probability distribution of the data base.

2.5 THE CHOICE OF THE LEARNING PARAMETERS

There are some degrees of freedom in the choice of the temporal evolution of the learning parameters. The requirements may be different for a simulation or for a hardware implementation, depending on the availability of some functions and the presence of inaccuracies or not. This paragraph proposes a particular neighbourhood which improves the speed of the organization phase and discusses the influences of the final adaptation gain and neighbourhood size on the final convergence phase.

2.5.1 The cone-shaped neighbourhood

As suggested in the previous paragraph, the choice of the temporal evolution of the learning parameters $\alpha(t_k)$ and $R(t_k)$ is somewhat critical for the organization phase. Referring to the adaptation rule (2.2), this may result from the largest update values $\alpha(x_j - m_{ij})$ at the periphery of the bubble. To overcome this problem, the gain $\alpha(t_k)$ must be a decreasing value of the distance to the most responding unit c within the bubble, which can be implemented by means of the cone-shaped neighbourhood shown in Figure 2.5.

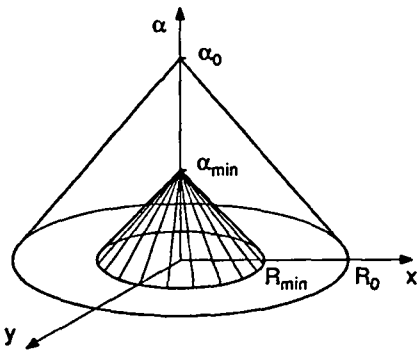


Fig. 2.5 The cone-shaped neighbourhood.

Therefore, due to its computational simplicity and its similarity with an available analogue hardware implementation [5] (see also Chapter 5), this cone-shaped neighbourhood will be used for the simulations shown throughout this document.

The simulation results shown in Figure 2.6 have been obtained under the same conditions as those of Figure 2.4, except for the shape of the neighbourhood. The organization phase has been reduced by a factor 10, and its evolution is much more regular.

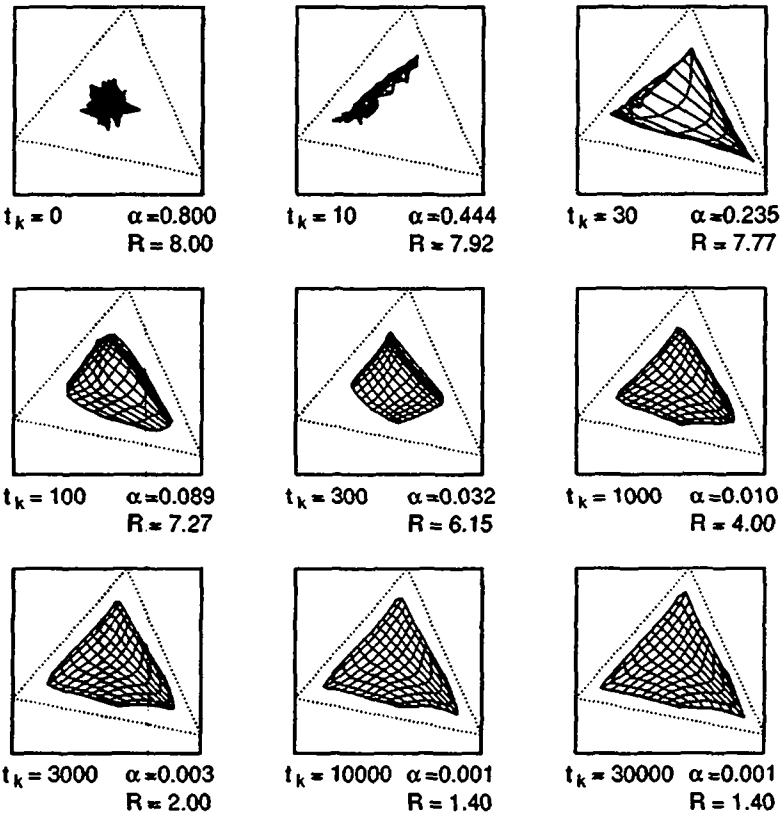


Fig. 2.6 Simulation using the modified "cone-shaped" neighbourhood.

2.5.2 Selection of the most responding unit

To select the most responding unit, Kohonen suggests the scalar product or some appropriate distance measurement. The scalar product requires the normalization of the input vectors and does not seem well adapted to practical cases. On the contrary, a distance measurement allows the use of the full input vector space. The simplest distance measurement implementable with analogue circuits is the manhattan distance:

$$D_M = \sum_{j=1}^{n} | x_j - m_j | \tag{2.5}$$

On the other hand, the euclidean distance is the most natural from a geometric point of view and can be realized by means of a simple self-biased analogue circuit [6] that is free of absolute error at the origin. The euclidean distance is formulated as follows:

$$D_E = \sqrt{\sum_{j=1}^{n} (x_j - m_j)^2} \tag{2.6}$$

The difference resulting from the choice of the manhattan or euclidean distance can be illustrated in two different ways. Figure 2.7 shows a hypothetical map in a two-dimensional space.

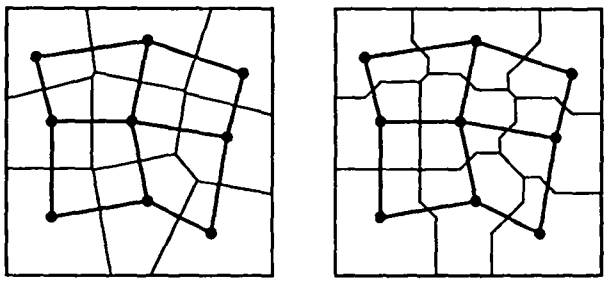


Fig. 2.7 Classes defined by euclidean and manhattan distances.

The thin lines separating the dots are the medians that define the classes represented by the neurons. There is a clear difference between the shapes of

the classes but their areas are very similar. The particular shape of the classes defined by the manhattan distance medians has an influence on the behaviour of the network, which can be highlighted by means of an adequate data base such that representing a lozenge of uniform probability used for the simulations of Figure 2.8. It can be seen that the manhattan distance forces the orientation of adjacent neurons along the referential axes, because this arrangement minimizes globally the distances between neurons. In opposition, the euclidean distance makes the network insensitive to direction, but is computationally and materially more expensive.

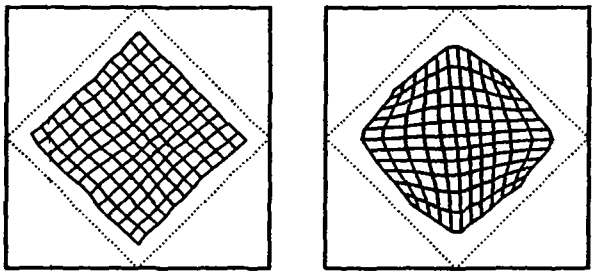


Fig. 2.8 Distance measurement effect on network's behaviour.

2.5.3 Influence of the absolute neighbourhood's size and gain value

The final convergence phase is carried out with constant neighbourhood size R_{min} and gain value α_{min}. These values are chosen using some criteria which depend on circuit performances and will be discussed in Chapter 4. Practically, the neighbourhood can be limited to only the most responding unit. However, in order to preserve the organizing behaviour of the network in the presence of inaccuracies, the smallest neighbourhood considered includes the four nearest neighbours. As a consequence, for the conic-shaped neighbourhood, the smaller radius must be just lower than $\sqrt{2}$ and the value 1.4 is chosen. Under this condition, the most responding neuron has a gain α_{min} and its four neighbours $2\alpha_{min}/7$, which reduces considerably the border effect (see [1], pp. 143-155).

In order to show qualitatively the influence of the learning parameters R_{min} and α_{min} in the convergence phase, 9 simulation cases are reported in Figure 2.9. The data base is a uniform distribution over the square representing the whole two-dimensional input range, and the network is made of a 10×10 neuron orthogonal array. There are two clearly non-correlated phenomena that depend respectively on the two parameters. Independently of R_{min}, the higher α_{min}, the more "wavy" is the network. Wavy means that the network moves locally a lot after each learning cycle, while keeping its global shape. The second effect, independent of α_{min}, is a contraction of the network with an increasing bubble size. This corresponds also to an increase of the *relative* border effect with increasing R_{min}.

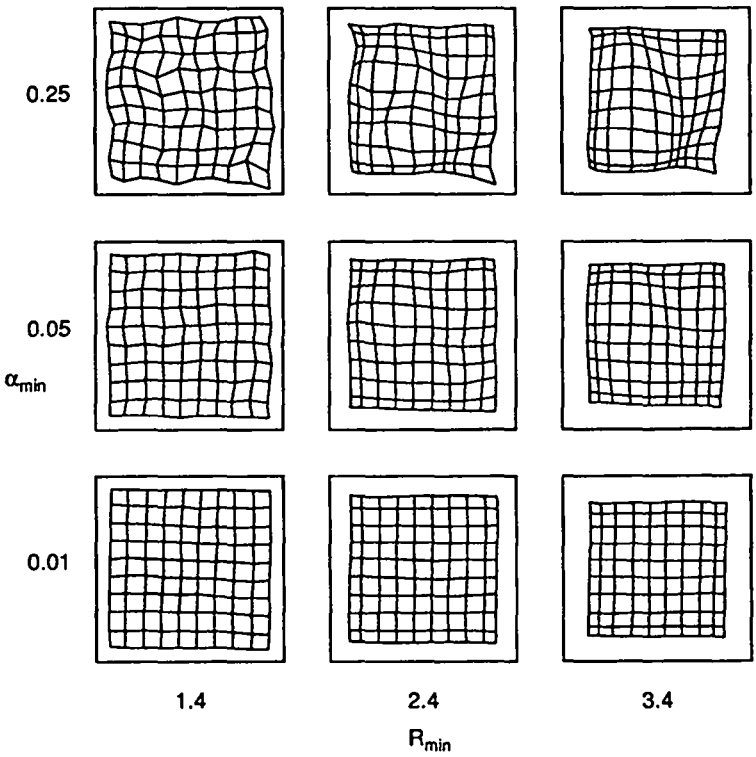


Fig. 2.9 Network's behaviour under various R_{min} and α_{min}.

2.5.4 Influence of the relative neighbourhood's size

In order to compensate the effects of some inaccuracies in hardware implementations, it may be useful to keep a non-minimum neighbourhood for the final convergence phase. This may be particularly helpful to compensate the leakages of medium-time memories in large networks since the total leakage is proportional to the network area (number of cells). For this purpose, several cases with various network and bubble sizes were simulated. The results are summarized in Figure 2.10 as a function of R_{min} and the size of the network.

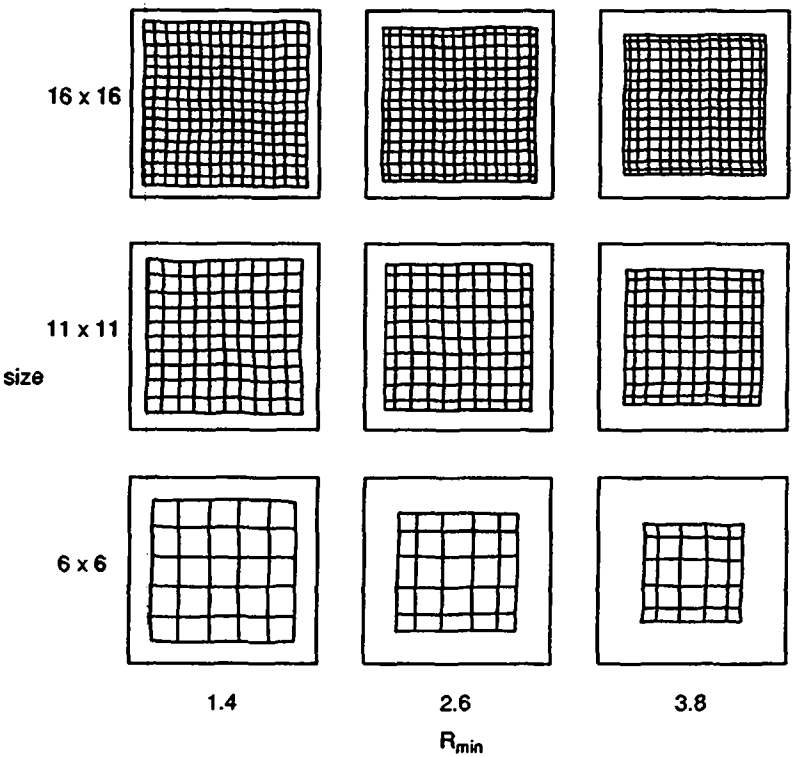


Fig. 2.10 Network's behaviour under various bubble to network area ratio.

For $R_{min} = 1.4$, the border effect is negligible and the network maps regularly the whole input data base. On the other hand, it has been already said that the border effect increases with increasing R_{min}, but what is more interesting is the nearly constant *absolute* border effect with constant bubble to network area ratio (B/N). This is noticeable considering the simulations on the diagonal of Figure 2.8, where B/N is about 17%.

2.6 CONCLUSION

The principle of the Kohonen network has been succinctly explained. Some practical aspects concerning the influence of the learning parameters on the network behaviour have been developed. These aspects are particularly important to set the specifications and requirements of the hardware implementations.

2.7 REFERENCES

[1] T. Kohonen: "Self-organization and associative memory", Springer Verlag, Berlin, 1988.

[2] E. A. Vittoz: "Analog VLSI Implementation of Neural Networks", Journées d'électronique 1989, Lausanne, 1989.

[3] E. A. Vittoz: "Analog VLSI signal processing: why, where and how?", to be published in the *Journal of VLSI Processing*, Kluwer, 1993.

[4] E. A. Vittoz and X. Arreguit: "Linear networks based on transistors", Electronics Letters, 4th February 1993 Vol. 29 No. 3.

[5] P. Heim, B. Hochet and E. Vittoz: "Generation of learning neighbourhood in Kohonen feature maps by means of simple nonlinear network", Electronics Letters, 31st January 1991 Vol. 27 No. 3.

[6] O. Landolt, E. A. Vittoz and P. Heim: "CMOS selfbiased Euclidean distance computing circuit with high dynamic range", Electronics Letters, 13th February 1992 Vol. 28 No. 4.

CHAPTER 3

ANALOGUE VLSI IMPLEMENTATION OF THE NETWORK

3.1 INTRODUCTION

The principle of the Kohonen map has been described in the previous chapter, and the tasks it has to accomplish may seem rather simple. Actually, considering VLSI implementations, both digital and analogue, functions such as clustering of the output activities, selection of the winner cell, storage of the synaptic weights or simply the multiplication involved in the algorithm, may need prohibitive silicon area and/or computing time. This work has been deliberately oriented towards a fully analogue implementation of the map, and the choices and compromises that have led to the final structure are developed in the present chapter. Emphasis is put on circuit design considerations, avoiding non conventional technologies, and exploiting as much as possible that which can be implemented with standard analogue CMOS technologies. As there are few clearly defined applications at the moment, certainly because of the lack of working hardware implementations, this work is not aimed at implementing a network dedicated to a particular application, but rather at finding a feasible implementation. This will clarify what can reasonably be expected in terms of precision, circuit area and power consumption. The proposed implementation is compatible with the results presented in Chapter 4, which gives information about the structures that must be avoided and the precision required to get acceptable performance. This chapter also deals with the signals representation so that multichip implementations are possible in the presence of chip-to-chip process variations.

3.2 NETWORK ARCHITECTURE

A general architecture of the map, including all the functions, can be inferred from the basic structure depicted in Figure 2.2 and from the simplified algorithm described in paragraph 2.4. The block diagram shown in Figure 3.1 includes the following elements:

- an input vector bus
- a weights vector bus to read the synaptic weights
- an output selector controlled by a digital decoder (not shown)
- a set of n synapses per neuron, with learning capability
- a set of M neurons, including activity computation and the collective and control circuits
- the interconnexions to next neighbours that are used for the collective functions and define the map physical topology
- a digital learning control signal that fixes the time period of the learning process
- two analogue signals to control the learning gain and the radius of the neighbourhood
- the output of the neurons that can be the processed activity (bubble) or simply the winning cell.

For some applications, the synaptic weights do not need to be known and the weights vector bus can therefore be omitted. In this case, the network is generally used for pattern classification, and the only information needed is the location of the winning cell or the position of the bubble. In spite of this, as long as the validity of an implementation has not been demonstrated, an output bus to read the synaptic weights is necessary.

Although some applications could be implemented using fixed synaptic weights, the network should be made not only programmable, but also adaptive, in the sense that the learning process takes place on-chip. For this purpose, various ways to implement plastic analogue memories are possible [1], such as sample-and-hold, EEPROM, UV-alterable floating gate devices or self-refreshing schemes. They are all affected by their own imperfections such

as leakage, limited number of operations, ultra-low speed or quantified values. Long-term memories such as EEPROM or UV memories need a special technology, and are therefore not considered in this work. In any case, the chosen memory implementation must satisfy some minimum specifications that can be inferred from the results presented in Chapter 4, so that the network behaves satisfactorily.

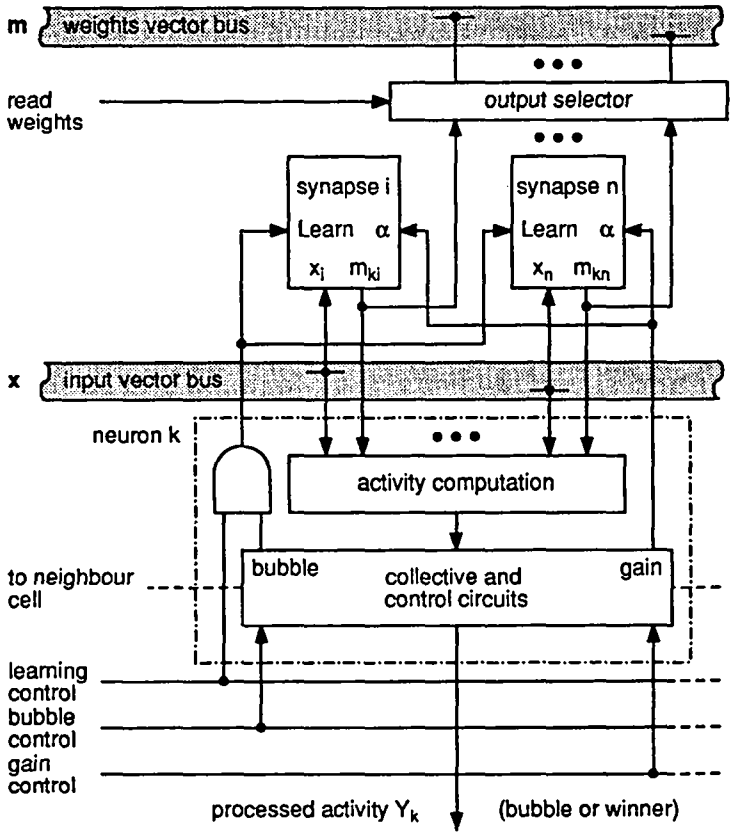


Fig. 3.1 General architecture of an analogue Kohonen map implementation.

The neurons include the activity computation and the collective and control circuits. The activity computation leads to rather simple circuitry that can be limited to a simple current summation. The purpose of the collective circuits is the generation of the learning neighbourhood and thereby to define the network's physical topology, which means that each cell has a well defined position within the network. Usual topologies can be one-dimensional, two-dimensional or eventually tri-dimensional. Since silicon VLSI processes are essentially two-dimensional, it is natural to build a two-dimensional rectangular lattice, as was presented in Chapter 2. The collective circuits can be implemented either by mimicking the action of the "Mexican hat" lateral interaction function, or by first selecting the most responding unit and then defining a bubble around it.

Finally, there are three signals used to control the learning process of the network. A digital signal or clock fixes the learning rate of the network, and must therefore respect the delays needed by the various circuits involved in the process to perform their operations. There are also two analogue signals used to control the time-dependent values of the learning gain α and the bubble radius R.

3.3 IMPLEMENTATION OF THE NEURONS

Considering the basic structure of the map depicted in Figure 2.2, the output processed activities Y_k are the result of both the internal activities due to the input vector and the action of the lateral interconnexions. The neurons thus contain the whole activity computation.

3.3.1 Computing the activity due to the input vector

The internal activity of a neuron must be a measure of *similarity* [2] between the input vector and the synaptic weight vector. The choice of this measure depends on some practical considerations concerning the circuit implementation, and on the implementation of the collective functions. As stated in Chapter 2, the dot product $x_i \cdot m_i$ is not suited for practical implementations as the input vectors may have to be normalized because colinear vectors do not necessarily carry the same information. As a matter of fact, if the vectors are normalized, an extra component with the value 1 must be added before normalization, to keep the information about the original norm of the vectors. This means that one extra synapse per neuron is needed.

On the other hand, the dot product requires a multiplier that may need large silicon area. It is therefore more convenient to use a distance or proximity measure, and normalize each component of the input vector so that they fit the voltage range of the synapses.

The block diagram of Figure 3.1 suggests that the neuron receives both the input and the weight vectors. On the other hand, the synapses need the values x_i-m_i for their own update computation. These values can therefore be fed back to the neuron and used directly to compute the euclidean or the Manhattan distance. Furthermore, if these values are represented by currents, the Manhattan distance is simply the sum of the components, and the euclidean distance can be computed using a simple circuit proposed by Landolt [3].

3.3.2 The neuromorphic "Mexican hat" approach

Implementing the Mexican hat lateral interaction function as shown in Figure 2.2 is practically impossible, as the number of interconnexions increases with the square of the number of neurons. A simple solution proposed by Vittoz [4] exploits the linear superimposition principle to implement a Mexican hat-like function by means of two resistance-conductance (RG) networks [5], as shown in Figure 3.2.

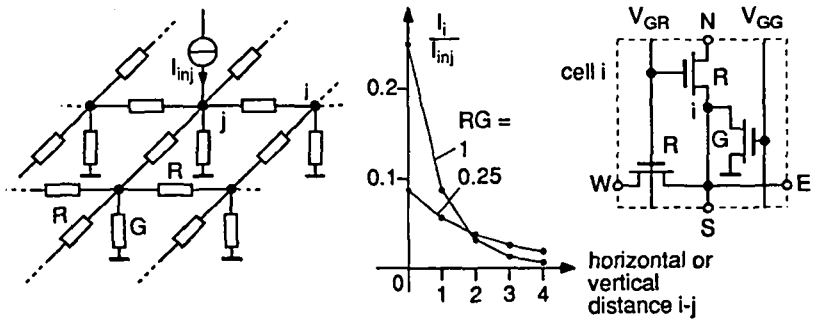


Fig. 3.2 Resistance-conductance network with its characteristics and MOS implementation.

A Mexican hat function can be implemented by subtracting the characteristics of two RG networks having different RG products. Each node of the RG network is associated to a neuron, which defines the physical topology of the map. The effects of the M neurons are superimposed and the M^2 lateral

interconnexions are thus replaced by the nearest neighbour connexions of the RG networks.

An experimental chip containing 12×12 cells has been made to test this technique [4]. To mimic realistic conditions, the map is implemented with fixed precalculated four-dimensional synaptic weights. A first excitatory RG network is made with MOS transistors working in their linear region to implement the resistances and conductances. The second inhibitory RG network is degenerated into a single conductance G_i (R=0). The gate voltages V_{GR} and V_{GG} of the transistors as well as the conductance G_i can be varied to modify the shape of the Mexican hat function. Since the map is of limited size, a solution had to be found to treat the terminations of the RG network. The most obvious solution is to close them by connecting the nodes of the first and last row and column. The network is thus continuous (it is a torus), with a periodic weighting function that is independent of the position on the map.

When presenting different input vectors, a bubble is formed at different locations on the map. Before obtaining a stable bubble, however, it is necessary to adjust the different parameters, and since their variation scale is rather limited, the possible variation of the bubble's size is also limited. In consequence, this approach has not been further investigated, because the algorithm needs a neighbourhood with a wide range variable size. On the other hand, a new technique has been proposed recently [6], for implementing with transistors, any network of linear resistors in which only currents are considered. It may be thus interesting to investigate this technique, since the value of the so-called pseudo-resistors can be controlled by their gate voltage in weak inversion.

3.3.3 The Winner- or Loser-Take-All approaches

Another approach, which is inspired from the simplified computational algorithm, is based on the selection of the most responding neuron c, followed by the generation of the learning neighbourhood centred around it. A Loser-Take-All circuit is used to select the most responding unit if the internal activities are computed by means of distance measurements, and a Winner-Take-All circuit is used if the activities are proximity measurements. The generation of the neighbourhood could be implemented using a linear RG network into which the most responding cell injects a current. Each cell then compares its own node voltage with a common reference voltage to know if it

is within the neighbourhood or not. This technique is however very imprecise and inconvenient due to the limited voltage range of the transistors in conduction and the rapidly decreasing function of the node voltages from the injection node. A more convenient nonlinear network, which uses the same number of transistors, is proposed in Chapter 5. Its main advantages are the generation of a well-controlled variable size neighbourhood and an easy detection of the bubble boundary.

Winner- or Loser-Take-All can be based on a circuit proposed by Lazzaro [7]. Its simplest implementation needs only two transistors per cell as shown in Figure 3.3 (a). However, it is not suited for multichip implementations with their inherent process variations between devices. For example, a difference between the β parameters could be compensated with a proper biasing technique, but a difference between the threshold voltages of transistors M_A cannot be compensated by simple means, and leads to a comparison error that depends on the input current levels because of the nonlinear characteristic of the transistors, as shown in Figure 3.3 (b).

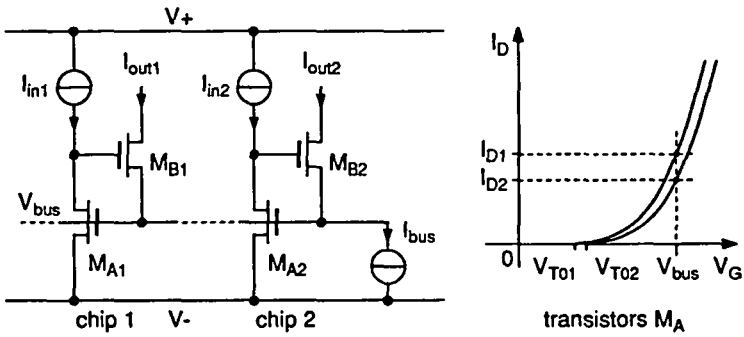


Fig. 3.3 Basic Winner-Take-All circuit proposed by Lazzaro [7].

To overcome these problems, the transconductance implemented by transistors M_A must be replaced by a differential pair polarized with a current that will be defined later. A more elaborate circuit based on this technique, which is multichip compatible and optimized for large bus capacitance, is proposed in Chapter 6.

3.4 IMPLEMENTATION OF THE SYNAPSES

The synapses may or may not include the learning capability. If learning is not implemented, the possible applications of the network are limited to analogue "look-up tables" for which the weights have been previously calculated by computer simulation. It can be used to implement multi-variable function interpolation, exploiting the position of the bubble as the output variable, which leads to continuous interpolation. For this purpose, the collective clustering function forming the bubble acts as a smoothing function against the unavoidable weights errors as well as the intrinsically localized storage of information. Although the potential of such fixed-weights networks is limited, their implementation is not necessarily straightforward. In addition, as neural networks should be able to handle corrupted or incomplete data without explicit error correction, on-chip learning capability is a key feature that must be implemented to fully exploit their potential.

3.4.1 Basic learning synapse for the Kohonen map

A synapse with learning capability must be able to update its weight according to a rule that depends on the neural network considered. The updating rule (2.2) used for the Kohonen algorithm is the discrete-time formulation of a low-pass filter that could be implemented using a resistor and a capacitor, as shown in Figure 3.4.

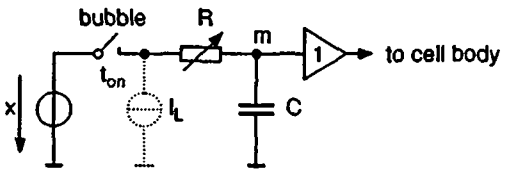


Fig. 3.4 Simple RC low-pass filter implementing a learning synapse.

In this version of the synapse, the learning gain α can be controlled by means of the on-time t_{on} of the switch and/or the resistance value. A simpler version, which avoids the switch and thus suppresses the leakage current I_L, might be implemented if a variable resistor with a range reaching extremely high value was available. Unfortunately, the implementation of such a device is difficult, if not impossible, with available standard CMOS technologies. This scheme is

however applicable to UV-light alterable memories, since the resistance of the oxide layer becomes extremely large in the absence of light.

3.4.2 Learning with medium-term memories

In the present context, medium-time storage refers to storage devices that can be used without perturbing the behaviour of the network significantly. In other words, the rate at which information is forgotten must be small enough to be compensated by the network under continuous learning. As an example, the analysis made in Chapter 4 shows that a forgetting rate of 5 to 20 ppm of full range per learning step is acceptable under some conditions. If the full range of the synapse is 1 volt and the learning rate 10,000 s^{-1}, then the corresponding acceptable forgetting rate is 50 to 200 mV/s. Forgetting rates that are two orders of magnitude smaller than the above-mentioned values can be attained with the synapse shown in Figure 3.5 [1].

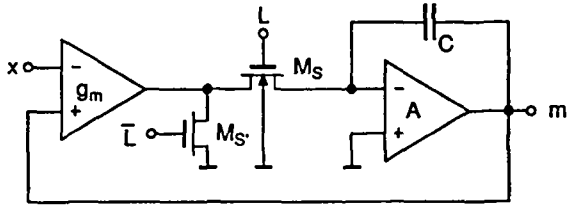


Fig. 3.5 Synapse based on a low-leakage sample-and-hold.

The synapse is based on a low-leakage sample-and-hold that reduces the voltage across the drain-to-bulk junction of the switch transistor M_S to the input offset voltage of amplifier A. This technique drastically reduces the leakage current compared to standard sample-and-hold configurations. A transconductance amplifier g_m is used in place of the resistor to generate an update current proportional to $(x\text{-}m)$, and the learning gain α is controlled by the switch's on-time. During storage, the current delivered by g_m must be diverted by the additional switch $M_{S'}$ to avoid activating a lateral bipolar action in M_S by forward-biasing its source-to-bulk junction.

Several drawbacks are incident to this implementation of the synapse. The circuit must be completed with a charge injection reduction technique that might be difficult to design because the switch is supplied with the current source g_m and turned off during the updating action. As a matter of fact, the

source voltage of M_S, when it is turned off, depends on the value and the sign of $(x\text{-}m)$ through its channel on resistance, which may influence the channel charge distribution between drain and source. In addition, the delay between the two non-overlapping clocks L and $\overline{L}$ must be very short and well controlled to avoid the above-mentioned lateral bipolar action in M_S.

3.5 SIGNAL REPRESENTATION IN MULTICHIP NETWORKS

Signals in analogue circuits are represented by physical variables, i.e. voltages, currents, charge, frequency or time duration. A signal x is represented by the value of a physical variable X related to the corresponding reference value X_{ref}. The reference of any signal must either be produced internally, with its origin clearly identified to keep it under control with respect to process and ambience variations, or provided from outside. For some operations, the number of physical references required by the operator may be reduced to just one or even none at all. On the other hand, all circuit architectures should be based on ratios of matched components values, to eliminate the dependency on any absolute value other than the required normalization reference X_{ref} [8].

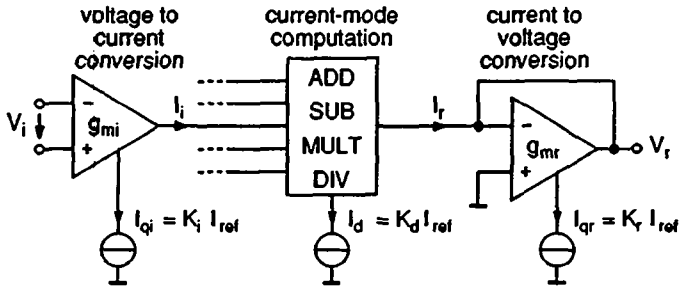


Fig. 3.6 General principle of voltage distribution of signals and current-mode computation.

In the implementation of a Kohonen map, the input signals are distributed in parallel to all the cells, and can therefore be advantageously represented by voltages. The Winner-Take-All circuit also has a bus that carries the same voltage to all the cells. On the other hand, the operations performed within the cells are usually best adapted to current-mode implementations, as for example summation or subtraction of signals, as well as all the possibilities carried out

by translinear circuits implemented with MOS transistors or compatible lateral bipolar transistors (CLBT). Under these conditions, when the network has to be implemented on several chips, a general scheme can be derived in which the I/O ports of the chips carry information represented by voltages and the internal operators are current-mode circuits. For this purpose, voltage-to-current (V/I) converters are used at the inputs and current-to-voltage (I/V) converters are used at the outputs of the chips, as shown in Figure 3.6.

The V/I and I/V converters are OTAs implemented by means of differential pairs in strong inversion to ensure large input and output voltage ranges. Each OTA i has a bias current I_{qi} that is a weighted copy of ratio K_i of a reference current I_{ref}. The OTAs are also characterized by the transconductance parameter β_i of their differential pairs. The transconductance and the asymptotic saturation voltage of a single-ended version of an OTA are given respectively by:

$$g_{mi} = \sqrt{\frac{\beta_i I_{qi}}{n}} = \sqrt{\frac{\beta_i K_i I_{ref}}{n}} \tag{3.1}$$

$$V_{idsat\,i} = \sqrt{\frac{n I_{qi}}{\beta_i}} = \sqrt{\frac{n K_i I_{ref}}{\beta_i}} \tag{3.2}$$

If the current-mode computation block executes current additions and subtractions only, then the resulting output voltage $V_{r,\pm}$ can be expressed as:

$$V_{r,\pm} = \frac{I_r}{g_{mr}} = \frac{1}{g_{mr}} \sum_n \pm g_{mn} V_n = \sum_n \pm \sqrt{\frac{\beta_n K_n}{\beta_r K_r}}\, V_n \tag{3.3}$$

This result shows that the output voltage is scaled by a ratioed term and independent of the reference current I_{ref}. If the computation block contains only sum and subtract operators, the scheme of Figure 3.6 is thus suitable for multichip implementations using any polarization technique.

Multiplying or dividing two input currents requires a third normalizing bias current $I_d = K_d I_{ref}$ as suggested in Figure 3.6. The multiply and divide operations can thus be written as follows:

$$I_{r,\,mult} = \frac{I_j I_k}{I_d} \tag{3.4}$$

$$I_{r,\,div} = \frac{I_l}{I_m} I_d \tag{3.5}$$

where I_j, I_k, I_l and I_m are input currents. The resulting output voltage of these operations are, respectively:

$$V_{r,\,mult} = \frac{I_{r,\,mult}}{g_{mr}} = \frac{g_{mj} V_j \, g_{mk} V_k}{g_{mr} I_d} = \sqrt{\frac{\beta_j \beta_k K_j K_k}{\beta_r^2 K_r K_d}} \sqrt{\frac{\beta_r}{n K_d I_{ref}}} V_j V_k \tag{3.6}$$

$$V_{r,\,div} = \frac{I_{r,\,div}}{g_{mr}} = \frac{g_{ml} V_l}{g_{mm} V_m} \frac{I_d}{g_{mr}} = \sqrt{\frac{\beta_l^2 K_l K_d}{\beta_m \beta_r K_m K_r}} \sqrt{\frac{n K_d I_{ref}}{\beta_l}} \frac{V_l}{V_m} \tag{3.7}$$

For both operators, the output voltage depends on a ratioed term (first square root) and on a scaling term (second square root) that is function of the technology parameters β_0 and n, and the reference current I_{ref}. Consequently, an arbitrary reference current I_{ref} is not adapted to multichip implementations since the outputs of different chips are not scaled with the same absolute reference. The scaling term containing I_{ref} has the form of the saturation voltage V_{idsat} of a differential pair biased by I_{ref} given by (3.2). The reference current I_{ref} must therefore be generated by means of a polarization that forces the saturation voltage V_{idsat} of a differential pair to be equal to an external reference voltage V_{ref}. Such a polarization is described in Appendix B and leads to a reference current that can be expressed as follows:

$$I_{ref} = V_{ref}^2 \frac{\beta_{ref}}{n K_{ref}} \tag{3.8}$$

where β_{ref} and K_{ref} stand for the polarization circuit. Substituting I_{ref} in (3.6) and (3.7) leads to:

$$V_{r,\,mult} = \frac{I_{r,\,mult}}{g_{mr}} = \frac{g_{mj} V_j \, g_{mk} V_k}{g_{mr} I_d} = \sqrt{\frac{\beta_j \beta_k K_j K_k K_{ref}}{\beta_r \beta_{ref} K_r K_d^2}} \frac{V_j V_k}{V_{ref}} \tag{3.9}$$

$$V_{r,\,div} = \frac{I_{r,\,div}}{g_{mr}} = \frac{g_{ml} V_l}{g_{mm} V_m} \frac{I_d}{g_{mr}} = \sqrt{\frac{\beta_l \beta_{ref} K_l K_d^2}{\beta_m \beta_r K_m K_r K_{ref}}} \frac{V_l}{V_m} V_{ref} \tag{3.10}$$

These expressions are now solely dependent on ratios and on an external reference voltage V_{ref} that can be distributed in parallel to all the chips. In

addition, it can be easily verified that the principle can be generalized to other current-mode operators.

Using a polarization that controls V_{idsat} is also interesting for this particular implementation because the input voltage range of the OTAs of all the chips are the same. This facilitates the worst case design when this voltage range is an important issue. Furthermore, the spread of the bias currents and the transconductance values among chips is that of the parameters β and n, which is limited to ±15% for actual standard CMOS technologies.

3.6 THE SELECTED ARCHITECTURE

The various criteria that led to the final choice for the implementation of an analogue Kohonen network have been developed in the previous paragraphs. The different circuit implementations that have been selected are the result of several trials with conventional techniques, which have generally failed due to their insufficient precision. To finally get a clear insight into the precision required by the algorithm, a program has been written to simulate the various inaccuracies that affect the circuits. The rather high accuracy that was needed for some functions led to the search for dedicated circuits implementations that will be developed in the following chapters. The general architecture of one cell is illustrated in Figure 3.7, which summarizes the basic principles without giving the details of the circuits. The important design strategies are developed below.

The synapse includes the analogue memory that stores the weight component, the update circuitry and the input V/I converter g_m that is compatible with the multichip communication scheme. The latter is an OTA that provides the value $(x\text{-}m)$ by means of a current. This OTA has an unavoidable input offset voltage V_{os} that does not perturb the behaviour of the map, but just affects the weight value m^* that is effectively stored into the memory and read at the output. The explanation of this arises from the following definitions:

- The coordinate system of the data base defines the referential for the input vectors x.

- The **internal value** m of a synaptic weight is defined with the same referential as the data base. Therefore, m **is equal to** x **when the physical update value** $\alpha g_m(x\text{-}m)$ **is equal to zero.**
- The **external value** m^* of the synaptic weight is the measurable value that is effectively stored into the memory. Therefore, $m^* = m + V_{os}$.

The internal value m of the synaptic weights will be implicitly used throughout this work.

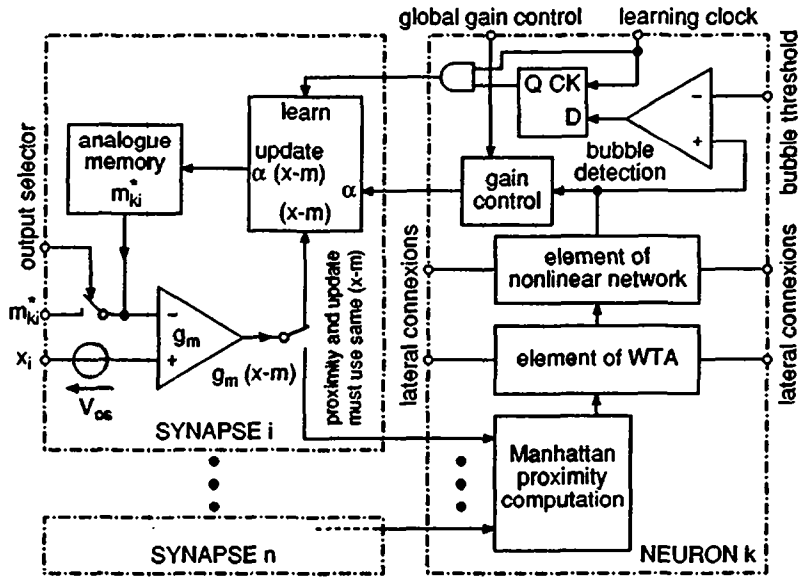


Fig. 3.7 Block diagram of one cell detailing the structure of the neuron and the synapses.

The output current $g_m(x\text{-}m)$ of the OTA is used alternatively for the update value computation and the proximity computation to avoid an error on the selection of the most responding cell. As a matter of fact, if the proximity is computed using a copy of this current, which is affected by the mismatches of the transistors, then the proximity function no longer reaches its maximum when $x = m$. This can be interpreted as a position error in the sense that the weight is not seen at its real internal coordinates by the proximity computation circuit.

A synapse that fulfils the above-mentioned considerations is described in Chapter 8. It uses a low-leakage technique, a simple charge injection compensation, and performs exactly the updating rule (2.2) thanks to a double storage technique that memorizes the present value $m(t_k)$ of the weight while computing its updated value $m(t_{k+1})$.

The neuron is based on a Winner-Take-All circuit that selects the cell with the largest Manhattan proximity value. The so-called winning cell then injects a current into a nonlinear diffusion network that generates a cone-shaped bubble. The bubble is detected by means of a comparator to define the learning neighbourhood, and its shape is exploited to define a position-dependent learning gain. To make the gain control more convenient, a global control is also provided for. Finally, the learning clock controls the process timing. It is gated by the previously latched bubble information to generate the update steps for the cells belonging to the neighbourhood.

The implementation of the Winner-Take-All circuit is described in detail in Chapter 6. As the activity of the neurons are compared by means of a single line carrying a voltage, the WTA is compatible with the multichip baising scheme described above. The nonlinear diffusion network is described in Chapter 5. Finally, the implementation of the synapse, which partially includes the proximity computation, is described in Chapter 8, and the implementation of the neurons is included in Chapter 9, which describes a complete multichip implementation of a Kohonen map.

3.7 CONCLUSION

In this chapter, the network architecture has been described and the possibilities for implementing the various blocks have been discussed. After having considered various approaches, some basic principles have been retained, and a more detailed architecture, which summarizes the global approach considered in this work, has been proposed. In addition, the proposed architecture is fully compatible with multichip implementations of the network.

3.8 REFERENCES

[1] E. A. Vittoz et al.: "Analog storage of adjustable synaptic Weights", proc. of the ITG/IEEE Workshop on Microelectronics for Neural Networks, Dortmund, Germany, June 25-26, 1990.

[2] T. Kohonen: "Self-organization and associative memory", Springer Verlag, Berlin, 1988.

[3] O. Landolt, E. A. Vittoz and P. Heim: "CMOS selfbiased Euclidean distance computing circuit with high dynamic range", Electronics Letters, 13th February 1992 Vol. 28 No. 4.

[4] E. A. Vittoz, P. Heim, X. Arreguit, F. Krummenacher and E. Sorouchyari: "Analog VLSI Implementation of a Kohonen Map", Journées d'électronique 1989, Lausanne, 1989.

[5] C. Mead: "Analog VLSI and Neural Systems", Addison-Wesley, Reading, 1989, pp. 107-116, 339-351.

[6] E. A. Vittoz and X. Arreguit: "Linear networks based on transistors", Electronics Letters, 4th February 1993 Vol. 29 No. 3.

[7] J. Lazzaro, S. Ryckebusch, M. A. Mahowald and C. A. Mead: "Winner-Take-All Networks of Order N Complexity", in Proc. 1988 IEEE Conf. on Neural Information Processing - Natural and Synthetic, Denver, 1988, pp. 703-711.

[8] E. A. Vittoz: "Analog VLSI signal processing: why, where and how?", to be published in the *Journal of VLSI Processing*, Kluwer, 1993.

CHAPTER 4

EFFECTS OF CIRCUIT INACCURACIES ON THE ALGORITHM

4.1 INTRODUCTION

Although it has been stated that analogue implementations of neural networks should lead to very dense circuits thanks to the high computation potential of MOS transistors [1], some limitations due to circuit inaccuracies must be expected [2]. It is certainly true that high precision should not be necessary to carry out cognitive tasks, as long as the network operates in a truly collective and continuous manner. However, contrary to supervised neural networks or hard-wired collective circuits like artificial retinas, the self-organizing process of Kohonen is not constrained by an expected output value or by separate input signals. A single input vector is compared to all the "free" weight vectors at each learning step. As a consequence, any error in the computation of the weights or in the selection of the winner has an influence on the next learning steps and can thus induce a mechanism which causes the network to diverge. This chapter analyses qualitatively the effects of circuit inaccuracies on the behaviour of the network by means of simulations. The analysis puts emphasis on the inaccuracies that are relevant to the particular implementation described in Chapter 3.

4.2 CLASSIFICATION OF INACCURACIES

The inaccuracies that may affect the considered circuit implementation can be separated into two classes, depending on whether they perturb the synaptic weights or the selection of the most responding neuron. The first class includes the imperfections of the synapse as a memory element, which are

- leakage current
- charge injection

and the following inaccuracies due to the learning circuits:

- offset (constant difference) on the gain α relative to x and m
- asymmetry of the gain α with respect to the sign of $(x\text{-}m)$.

The second class concerns the neuron circuits, i.e. the distance or proximity measurement and the Winner-Take-All, from which the most relevant inaccuracies are:

- error of position on the distance or proximity measurement
- error in the selection of the winner.

Each type of inaccuracy is analysed and simulated separately. In addition, random and systematic inaccuracies of the same type may also be simulated separately, because they can sometimes give rise to distinct effects. The values are chosen so that the network just converges to a satisfactory mapping, i.e. which still looks organized from an aesthetic point of view. For this purpose, the simulations may be repeated using three increasing values, the last one being usually too large. This criterion is totally subjective and does not reflect the exact requirements of a particular application, but will be perceived as natural by most readers. Another aspect that is interesting to investigate is the usefulness of using large networks if the accuracy limits the resolution to a value that can be obtained using a smaller network. This is an economic criterion: as an example, it is useless to build a 12-bit D/A converter with a technique that ensures a precision of 8 bits only. Although the implementation of the four extra bits does not degrade the overall 8 bits of precision, it is a waste of silicon area, and can also induce nonmonotonicity that may perturb the behaviour of the application.

4.3 SIMULATION PARAMETERS

From a practical point of view, the simulations must show the mapping obtained after a sufficient number of learning steps during the convergence phase, when both the gain and the neighbourhood's radius have reached their minimum value. The data base, which is as simple as possible, is a uniform distribution covering the whole range of the synaptic weights in a two-dimensional space; it is thus a square, the size of which is normalized to unity for simplicity.

To illustrate several effects, networks size from 6×6 to 16×16 neurons will be used. In addition, the low limit values α_{min} and R_{min} of the parameters may be different depending on the considered inaccuracy. Therefore, to keep some consistency among the simulations, the evolution of the parameters are chosen such as the low saturation values of the gain and the bubble radius are reached respectively after 5000 and 2000 learning steps, in any case. For this purpose the respective values of the time constants τ_α and τ_R are calculated according to relations (2.3) and (2.4). Under these conditions, and using $\alpha_0 = 0.8$ and $R_0 = 6$, all the simulations behave satisfactorily. The results are read after 20000 learning steps.

The Kohonen map features a projection which tends to reproduce the probability distribution of the data base, in the sense that larger mapping areas are assigned to the regions of the input space which are presented more frequently. On the other hand, the effects of the inaccuracies must be much lower than the update values to preserve the network behaviour, and the average update value can be deduced from the average distance between neurons, which depends on both the probability distribution of the data base and the number M of neurons in the network. If the probability distribution of the data base is uniform over the whole vector space allowed by the range of the synapses, the average distance between neurons is proportional to $(M)^{-1/n}$, and increases with the dimension n of the input data base. Since the data base is generally not uniform, this distance cannot be used to deduce the average update of the weights. In fact, the average update value can be directly approximated by averaging the successive update values $\alpha |x - m_c|$ of the winning cells during the learning process. This is a good approximation provided the cone-shaped neighbourhood is used. Indeed, during the convergence phase, the neighbourhood to network area ratio (B/N) is usually small and the limited map area it covers is approximately uniform.

Consequently, the term $(x\text{-}m)$ is linearly increasing while α is linearly decreasing, and the weights updates are thus of the same order among the neighbourhood. The time-averaged update $E\{\alpha\,|x - m_c|\}$ is computed by the simulation program by means of a low-pass filter with a time constant of 100 learning steps, and will be compared to the values of the inaccuracies. On the other hand, it may be interesting to control this value by adjusting α during the process. For this pupose, a simple feedback loop is also implemented in the program. To ensure that the initial value of $E\{\alpha\,|x - m_c|\}$ is large enough, the initial setting of the weight vectors is randomly distributed in a limited area as shown in Figures 2.4 and 2.6.

The simulation results are presented in the same way as in Chapter 2. Below the plots are the following indications: the value of $E\{\alpha\,|x - m_c|\}$ and the value of α_{min} at $t_k = 20000$, and the value of the considered inaccuracy. The two first terms are designated by the letters E and α for simplicity. If not specified, the radius of the neighbourhood has the default value $R_{min} = 1.4$ (the unit is the cell index).

4.4 INACCURACIES OF THE MEMORY

Charge injection and leakage are inherent to sampling on a capacitor. Considered in the context of the Kohonen algorithm, they lead to significantly distinct effects. The charge injection affects only the cells being updated, whereas the charge leakage affects all the cells permanently. The leakages can be considered as constant current sources and the charge injection as a constant charge injected at each weight's update. This is a good approximation for the particular implementation of the synapse described in Chapter 8.

4.4.1 Leakages

Because the leakages affect all the cells, their relative effect depends on the bubble to network ratio B/N since the "refresh" of the weights is made by means of the learning. The value of the leakage is expressed in ppm per iteration, and thus depends on the learning frequency. Each weight has its own leakage value which is kept constant during the whole learning process.

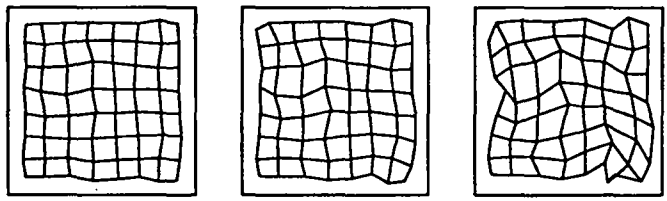


Fig. 4.1 Constant leakages normally distributed, with zero mean value.

Figure 4.1 shows the results using randomly distributed leakages with σ ranging from 5 to 20 ppm/it. The same file was used for the three simulations with different scaling factors. Random leakages lead to some "crumpling" of the map, which is acceptable as long as it doesn't look disorganized.

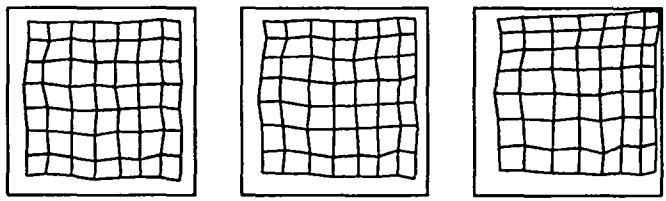


Fig. 4.2 Constant leakages L identical for all the synapses.

Figure 4.2 shows the results under constant leakages L, identical for all the synapses. As the value is positive, the map is shifted towards the top right corner. For larger values, some cells may be pushed out of the data base boundary and thus discarded from the process. This results in a diminishing of the effective neuron density that increases the mean update value E.

If maps of different size are used with the same B/N ratio, the relative effect of the leakage is similar, as can be seen on Figure 4.3, where the dotted lines have the same coordinates for the three cases.

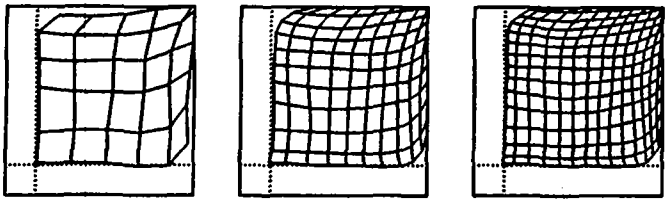


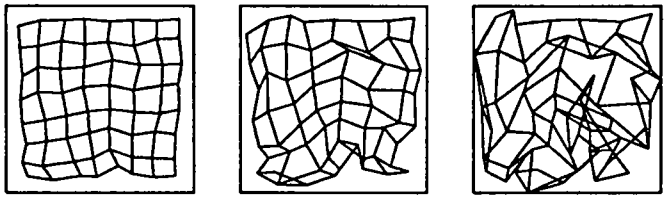
Fig. 4.3 Constant leakages of **50 ppm/it.**, networks with equal B/N ratios.

The above results account for rather large values. As an example, 10 ppm/it. corresponds to a leakage of 100 mV/s, with a synapse having a full range of 1 V, at a learning frequency of 10 KHz. The synapse proposed in Chapter 8 is nearly two orders of magnitude better than the above value. Therefore, leakage will not be a problem under continuous learning.

4.4.2 Charge injection

Charge injection is the most difficult inaccuracy to minimize, because it depends on the matching of minimum size transistors. The values used for the simulations depicted in Figures 4.4 to 4.7 correspond to what can be reasonably attained. As an example, the synapse of Chapter 8 has a sigma of 0.4 mV for a voltage range of 2 V, which corresponds to the lower value used hereafter.

Figures 4.4 and 4.5 show that the transition between a satisfactory and an unacceptable result is very abrupt. For the case of Figure 4.5 the ratio of the perturbation to the mean update value ranges from about 8% to about 20 %. This suggests that above some limit of accuracy, some phenomena can appear suddenly, as for example nonmonotonicity in D/A converters. The result can be improved by increasing α_{min}. This can be done for example by controlling the value of E as shown in Figures 4.6 and 4.7. In this case, the update values are increased, which reduces the effect of charge injections. On the other hand, simulations have shown that the result is neither improved nor degraded by increasing the minimum radius. This may be because the increased collective behaviour is compensated by reduced update values due to the shrink of the map and the increased number of perturbated cells.

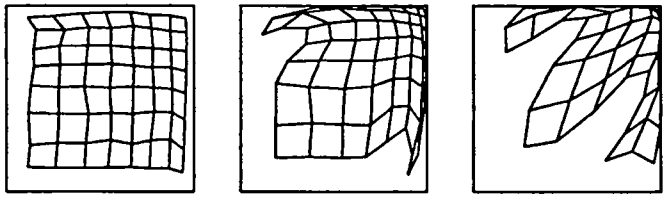


E = 2.547E-3 E = 2.679E-3 E = 2.863E-3

$\alpha = 0.05$ $\alpha = 0.05$ $\alpha = 0.05$

$\sigma = 0.0002$ $\sigma = 0.0005$ $\sigma = 0.001$

Fig. 4.4 Constant charge injections normally distributed, with zero mean value.

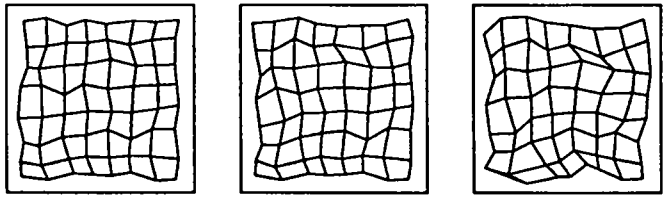


E = 2.664E-3 E = 3.520E-3 E = 4.720E-3

$\alpha = 0.05$ $\alpha = 0.05$ $\alpha = 0.05$

c = 0.0002 c = 0.0005 c = 0.001

Fig. 4.5 Constant charge injections, identical for all the synapses.



E = 0.01 E = 0.01 E = 0.01

$\alpha = 0.193$ $\alpha = 0.192$ $\alpha = 0.191$

$\sigma = 0.0002$ $\sigma = 0.0005$ $\sigma = 0.001$

Fig. 4.6 Constant charge injections normally distributed, with zero mean value.

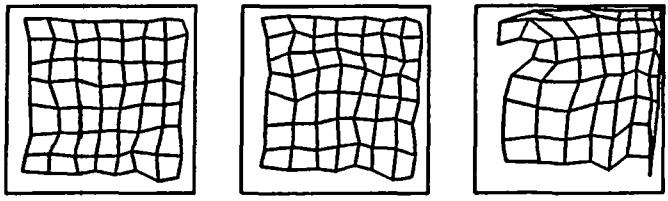


Fig. 4.7 Constant charge injections, identical for all the synapses.

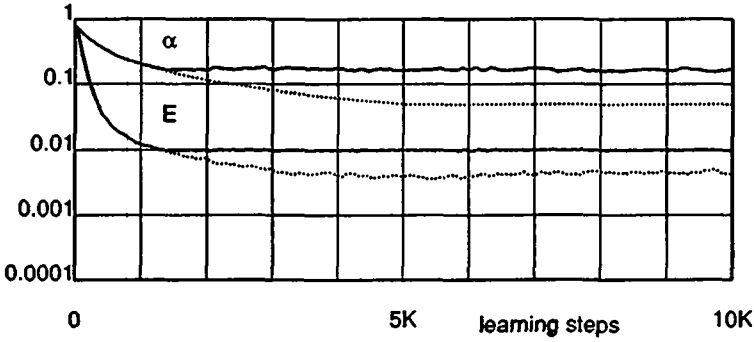


Fig. 4.8 Evolution of E and α, with and without control of E.

Figure 4.8 shows the evolution of the parameters α and E for the case $\sigma = 0.001$. The dotted lines stand for the programmed evolution of the parameters, and the plain lines for their controlled evolution, which clamps E at 0.01. The use of the control loop can be useful to find the minimum required value of the gain when the data base is unknown a priori, because a given level of charge injection imposes a minimum value of E to get an acceptable mapping. Therefore, if the mapping is still not acceptable with a rather large gain value, it is most likely that the network contains too many cells, as will be shown.

The above results used the minimum bubble radius $R = 1.4$, and it has been stated that it is useless to increase the radius to compensate the charge injection. But what happens if the network size is increased while keeping the B/N ratio constant? The answer is given by the simulation results depicted in

Figure 4.9. The network contains four times more neurons (twice the number of rows and columns) compared to the above simulations, and the bubble also contains four times more neurons (R = 2.8 instead of 1.4).

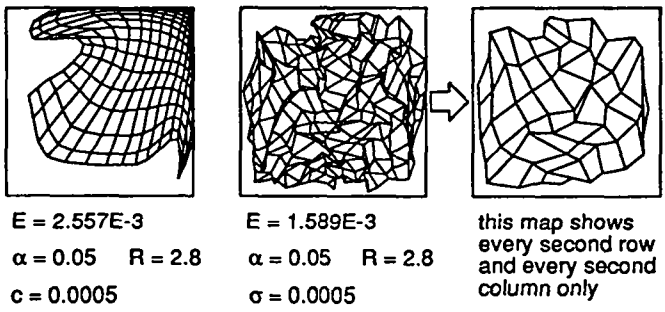


Fig. 4.9 Effect of charge injection (equally and normally distributed) on a 16 × 16 neuron network with R = 2.8. These results must be compared with those of Fig. 4.4 and 4.5 ($c = \sigma = 0.0005$), for which the bubble to network ratio B/N is the same.

The result is easily interpreted for the case of equally distributed charge injection. The global shape of the map is the same as that of Figure 4.5 for the corresponding charge injection value (compare the "wing" at the bottom right corner). On the contrary, the result concerning the normally distributed charge injection is not evident because the map looks rather disorganized. It becomes clearer when only every second row and every second column of the map are considered as shown in the third drawing. The resulting 8 × 8 neuron map doesn't look more disorganized than that of Figure 4.4 for the corresponding charge injection value. Some interesting properties can be deduced from these results. In general, the global behaviour of the map is not modified by the number of neurons, provided the B/N ratio is kept constant. However, for randomly distributed charge injection, the apparent disorganization is in fact due to an excessive resolution of the map relative to its accuracy. The phenomenon is therefore similar to the presence of nonmonotonicity in D/A converters. As a consequence, the use of a large B/N ratio is not suited to estimating the network behaviour under the presence of charge injection, and the minimum radius R = 1.4 must be used to find the economically optimized number of neurons.

4.5 INACCURACIES OF THE LEARNING CIRCUITS

4.5.1 Offset on the gain α relative to x and m

This inaccuracy is analysed here to show that it is risky to try to implement the algorithm using analogue circuits based on the following equivalent formulation of the update process:

$$m_{ij}(k+1) = (1 - \alpha)\, m_{ij}(k) + \alpha\, x_j(k) \tag{4.1}$$

This formulation [3] shows that the terms m and x might be processed separately and thus suggests nice implementations using for example two differential pairs in weak inversion or variable duty cycle switched currents to implement the weighting terms $(1-\alpha)$ and α, x and m being unipolar currents. In any case, however, there will be an error that is likely to make the sum of these two weighting terms non-unity. This can be for instance offsets on the differential pairs or phase delays on the switched currents, leading to different values of α for m and x. This can be formulated as follows:

$$\alpha_x = \alpha_m + \Delta\alpha \tag{4.2}$$

where α_x and α_m are the gains applied to x and m, respectively, and $\Delta\alpha$ is a constant offset term between the two gains.

Then rewriting the update process in its original formulation yields:

$$m_{ij}(k+1) = (1 + \Delta\alpha)\, m_{ij}(k) + \alpha_x \left(x_j(k) - m_{ij}(k) \right) \tag{4.3}$$

In vectorial terms, m should move towards x along a straight line. The excess vectorial term $\Delta\alpha m$ diverts m with an angle δ_m given by:

$$\delta_m = \operatorname{atan} \frac{\Delta\alpha \left| \vec{m} \right|}{\alpha_x \left| \vec{x} - \vec{m} \right|} \tag{4.4}$$

The angle thus increases both as m approaches x, and with the norm of m. As a consequence, the divergence of the update value, and thus the effect on the behaviour of the map, must depend on the sign of the inaccuracy $\Delta\alpha$ (the weights are restricted to positive values by the particular implementations).

Figure 4.10 shows the results using a positive, negative and random value of 0.001 for $\Delta\alpha$, which represents only 2% of α_{min} in this case.

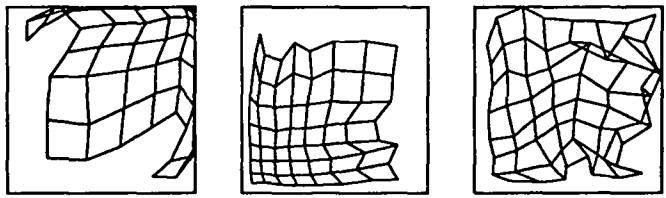


Fig. 4.10 Offset on the gain α relative to x and m (positive, negative and random values).

The effects are surprisingly important, taking into account the relatively low value of the inaccuracy (particularly if it stands for the input offset voltage of differential pairs in weak inversion). There is also a clear difference between the positive and negative values. A positive value pushes several cells outside the data base boundary, giving rise to an effect similar to that of systematic charge injections. On the contrary, a negative value does not push any cell outside the data base, but just increases the neuron density towards the left bottom corner of the data base. Finally, random values can significantly disorganize the map. These results show that implementations based on relation (4.1) must be avoided.

4.5.2 Asymmetry of the gain α with respect to the sign of $(x\text{-}m)$

This inaccuracy is present when the circuit implied in the generation of the update value is not symmetrical, but exempt from offset. The current rectifier presented in Chapter 7 can be adequately characterized by this kind of inaccuracy as can be seen in Figure 7.2. The resulting effect on the learning gain can be modeled using two different values α_n and α_p with constant ratio α_n/α_p , for the negative and positive values of $(x\text{-}m)$ respectively.

The simulation results presented in Figure 4.11 use values that are larger than what can be expected from the matching of a current mirror. Consequently, this type of inaccuracy is not critical for the implementation of the network, and the use of the above-mentioned rectifier for computing the update value in the implementation of the synapse described in Chapter 8 is justified.

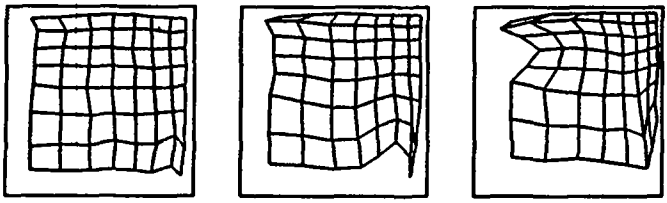


Fig. 4.11 Asymmetry of the gain α with respect to the sign of $(x\text{-}m)$.

4.6 INACCURACIES OF THE NEURON CIRCUITS

The neuron circuits include the common features of a single cell. These are essentially the distance or proximity measurement and the selection of the most responding cell by means of a Looser-Take-All or a Winner-Take-All circuit respectively. As both approaches are similar, the analysis will be limited to the case of a proximity measurement associated with a Winner-Take-All, corresponding to the choice made for the practical implementation.

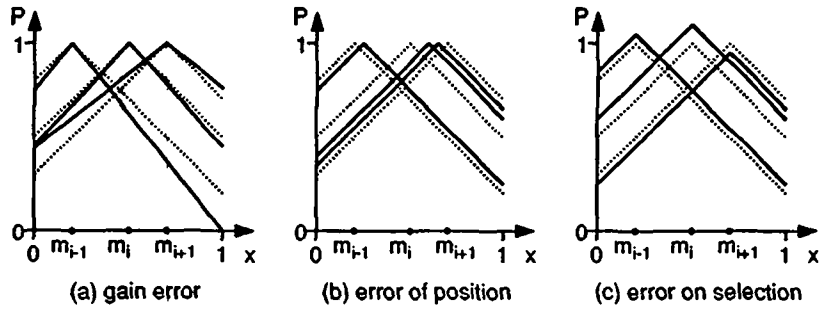


Fig. 4.12 Inaccuracies affecting the selection of the winner (proximity measurement P).

Proximity measurements can be affected in two ways: the scale may be different among cells or an error (generally an offset) may appear between the measured and the internal value of the components. This is illustrated in Figure 4.12 (a) and (b) in the one-dimensional case (the dotted lines stand for the ideal case). Because the first inaccuracy cannot lead to the permutation of two cells, it is considered as a soft one and will not be considered in this work.

The Winner-Take-All, considering the implementation described in Chapter 6, is affected by an offset on its differential pair, which affects the selection of the winner as shown in Figure 4.12 (c). This offset also includes the accumulation of all mirrors' mismatches along the path from the proximity measurement to the Winner-Take-All.

4.6.1 Error of position on the distance or proximity measurement

This inaccuracy arises from particular circuit architectures. If, for instance, the circuit uses two different paths to compute the term $(x\text{-}m)$ for the proximity measurement and the update of the weights, an offset term may appear between the two computed values, and the network sees the synaptic weight at a virtual position which is different from its real internal coordinate, as already stated in paragraph 3.6. It will be shown later that the proposed circuit implementation is free of this kind of error.

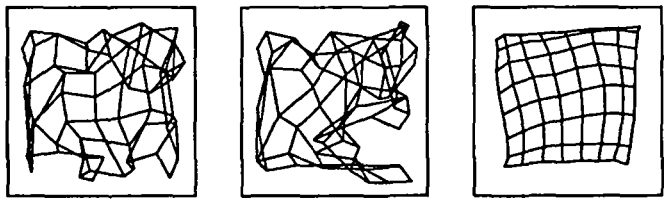


Fig. 4.13 Error of position with $\sigma = 0.05$ on the components. (a) simulation with standard parameters, (b) simulation using control of E at 0.01, (c) simulation using an increased minimum radius $R_{min} = 2.5$ (the bubble contains 21 cells).

The value used for the simulations of Figure 4.13 (5% mismatch, including all contributions) is representative of what can be expected from the circuits, taking into account their limited areas. The resulting effect is an unacceptable disorganization of the map. Increasing the gain worsens the result because it is obvious that the position errors cannot be compensated by means of increased update values. The only way to compensate the position errors is to ensure that the bubble includes the cell whose internal coordinates are the closest to the input vector, i.e. the cell that should have won. Since this error is referenced to the data base coordinate and the bubble radius is referenced to the indices of

the cells, the required size of the bubble depends on the highest local neuron concentration, which in turn depends on the data base structure. The simulation of Figure 4.13 (c) uses a minimum radius of 2.5, leading to a bubble containing 21 cells. The resulting map is totally organized and smooth, at the expense of a slight shrink of the map, as was described in Chapter 2.

4.6.2 Error in winner selection

This inaccuracy does not introduce an error in the position but leads to a similar effect because a neighbouring cell can be selected instead of the closest cell to the input vector. Although the considered implementation is free of error in the position, the error in the selection of the winner cannot be avoided practically.

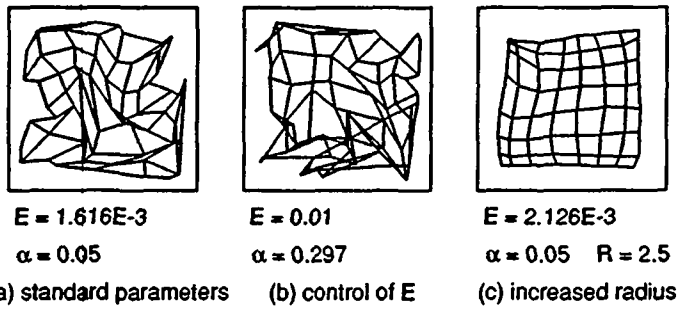


Fig. 4.14 Error in selection of the winner with $\sigma = 0.05$. (a) simulation with standard parameters, (b) simulation using control of E at 0.01, (c) simulation using an increased minimum radius $R_{min} = 2.5$ (the bubble contains 21 cells).

The simulation depicted in Figure 4.14 is made in the same conditions as those of the above inaccuracy, and leads to similar effects. The mapping is also improved by increasing the minimum radius of the neighbourhood, and the same criteria stand for the choice of the minimum size of the bubble. For a given value of the inaccuracy, the behaviour of a map containing more neurons will not be modified if the same B/N ratio is used. Therefore, there is some similarity with charge injection, which is also an inaccuracy referenced to the data base coordinate. Since this inaccuracy is unavoidable, care must be taken when designing the whole path for the selection of the winner.

4.7 REALISTIC 3-DIMENSIONAL EXAMPLE

In order to show the behaviour of a Kohonen network containing several realistic inaccuracies in a more general case, a data base with a dimension larger than two should be used. Since it is not possible to represent graphically a dimension larger than three, a three-dimensional data base representing an open cylinder has been chosen. Figure 4.15 shows the envelope of the data base and the mapping performed by a 10×10 neuron network that is free of inaccuracies.

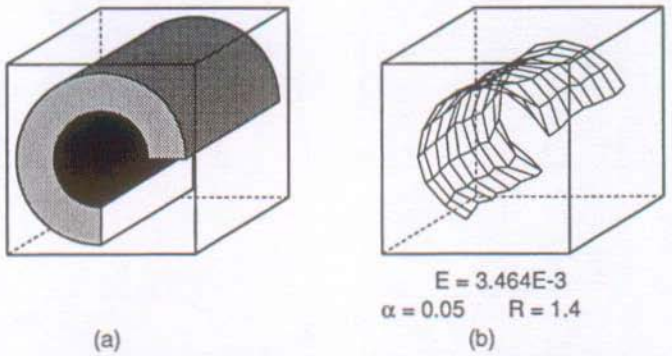


Fig. 4.15 (a) envelope of the data base (b) mapping without inaccuracies.

The thickness of the cylinder does not appear on the mapping because it is too small compared to the relative quantization of the map. This example is somewhat particular since the map is in fact constrained within a surface, and does not lead to an effective dimension reduction. This choice was however necessary for the readability of the results.

It is now interesting to see how the network behaves under the presence of several inaccuracies. For this purpose, the most relevant inaccuracies affecting the considered implementation are taken into account and summarised in Table 4.1. As suggested in the above paragraphs, there is no offset on the gain relative to x and m, and the asymmetry of the gain with respect to the sign of $(x - m)$ is negligible. Furthermore, the network will be exempt from offset on position thanks to the use of the rectifier described in Chapter 7. The values listed in the table are compatible to what can be reasonably expected from circuit implementations.

Table 4.1 (units are related to the full scale of the synapses)

inaccuracy	mean value	sigma	unit
leakages	1	1	[ppm/it.]
charge injections	0.0002	0.0002	[-]
error on selection	0	5	[%]

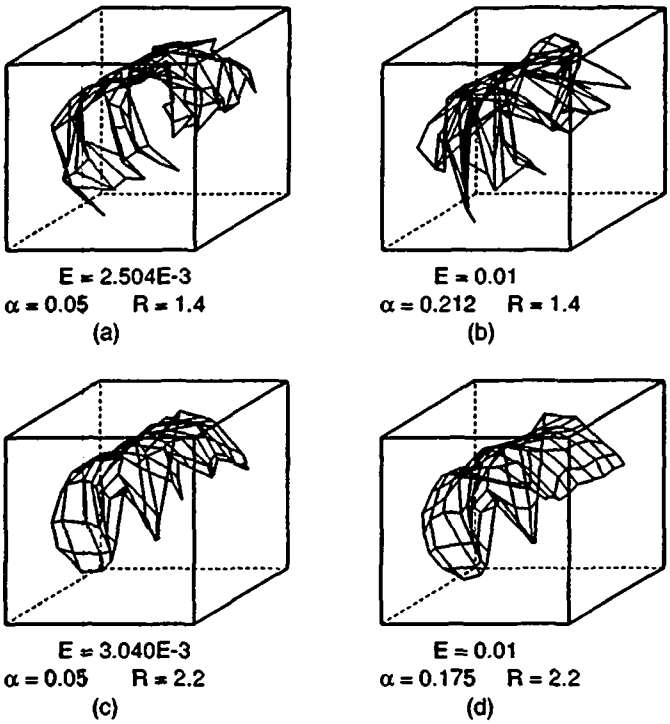


Fig. 4.16 Simulations (a) with inaccuracies of Table 4.1 (b) with control of E (c) with increased radius (d) with increased radius and control of E.

The simulation results depicted in Figure 4.16 are all made with the inaccuracies listed in Table 4.1. The result using the standard small gain and radius shows a mapping locally disorganized, although having a global cylindrical shape still recognizable. The result is worse if the gain is increased by controlling E to 0.01, which is due to the errors in the selection of the

winner. With increased radius only, the map is locally better organized, but there are some peaks due certainly to the charge injections. Some shrinkage of the map is also noticeable, due to the increased radius. Finally, using both increased radius and control of E, improves significantly the result. The shrinkage and the peaks of the previous case are still noticeable, but the map is better organized and looks smoother. It is also interesting to notice that the gain in case (d) is smaller than the gain in case (b), despite the more regular mapping. This example shows that in the presence of several type of inaccuracies, a compromise must be made between both compensation techniques to get an optimal result.

4.8 OPTIMAL CHOICE OF THE NETWORK'S SIZE

The simulations have clearly shown that precision has a key importance on the results. There are two dominant inaccuracies, practically unavoidable, that limit the performances of analogue implementations of the network: charge injection and error on the selection of the winner. These inaccuracies both lead to absolute errors that are referenced to the functional range of the synapses. As already stated, these errors strictly limit the resolution of the network in the same way as the resolution of a D/A converter is limited by the accuracy of its LSB. However, whereas the structure of a D/A converter is set a priori, that of the network's mapping depends on the data base distribution, so that the optimal number of cells that just fits the resolution of the network for a particular application cannot be known a priori. Therefore, the estimation of the optimal number of cells requires first, that the inaccuracies of the circuits are well characterized, and second, that some trials are made by simulation for each type of inaccuracy. The charge injection level will fix the optimal number of cells for a given learning gain α_{min}, and then the error on the selection of the winner will fix the minimum B/N ratio. The first operation must be done before the second, with the minimum bubble radius $R_{min} = 1.4$, because a larger B/N ratio can hide the underlying behaviour of the map, as was stated in paragraph 4.4.2.

On the other hand, it is worth noticing that both the square data base and the cylindrical data base used in the above simulations do not formally lead to dimension reduction. This property however corresponds to what is expected from the Kohonen network, as most practical applications will use multi-dimensional data bases. In these conditions, the map can evolve within a

larger volume which will allow the use of larger networks for the same circuit performance. In addition, since the map performs a projection of the data base, the mapping is no longer comparable to some exact geometrical shape. This suggests that the use of quite large neighbourhoods may be tolerated, so that the topology preserving ability of the network is kept as much as possible by means of increased collective behaviour.

Finally, as charge injection is inherent to sampling on capacitors, better performances may be attained if the learning was continuous in time. However, as stated in paragraph 3.4.1, a variable resistance reaching extremely high value cannot be implemented with actual standard CMOS technologies. On the other hand, the use of the truly collective Mexican hat lateral interaction function should also lead to better performance than the Winner-Take-All approach, since the most responding cell is certainly included within the bubble, whatever its size, when the network is organized.

4.9 CONCLUSION

This chapter has clarified some aspects of the Kohonen algorithm in the presence of circuit inaccuracies. It is shown that some types of inaccuracies are unacceptable, and thus gives information about the circuit structures that must not be used. Some other inaccuracies can be compensated by means of the learning parameters. The resulting effects of the compensations have also been described. In addition, the use of realistic values is essential to set the specifications of circuit implementations. Finally, some practical considerations about the size of the network have been proposed that must be further experimented within real situations.

4.10 REFERENCES

[1] E. Vittoz et al.: "Analog implementation of a Kohonen map", Journées d'électronique 1989, Ecole Polytecnique Fédérale de Lausanne (EPFL), 1989.

[2] P. Heim: "Influence des défauts inhérents aux réalisations VLSI analogiques du réseau de Kohonen", Cours Postgrade en Informatique Technique, Ecole Polytechnique Fédérale de Lausanne (EPFL), 1992.

[3] O. Landolt. Private communication.

CHAPTER 5

NONLINEAR DIFFUSION NETWORK FOR NEIGHBOURHOOD GENERATION

5.1 INTRODUCTION

In Chapter 2 it was emphasized that the topology of a Kohonen map is defined by means of its lateral interconnections. A possible implementation of the neuromorphic approach that needs the Mexican hat's lateral excitation-inhibition coupling function has been successfully demonstrated [1] with the help of two-dimensional linear resistance-conductance (RG) networks (Figure 5.1), originally proposed by Mead to compute local average in artificial retinas [2]. For this purpose, a first RG network was used to propagate the excitatory signals and a second degenerated network was used to compute the global inhibitory level. Another approach, which is closer to the simplified computational algorithm, uses a "winner-take-all" network [3] to select the most responding unit and a single RG network to define the topological neighbourhood around it. However, the rapidly decreasing slope of the RG network's characteristic is not suited to defining large neighbourhoods. In this chapter, a variation of this network is proposed that exploits nonlinearities to generate well-controlled variable size neighbourhoods [4].

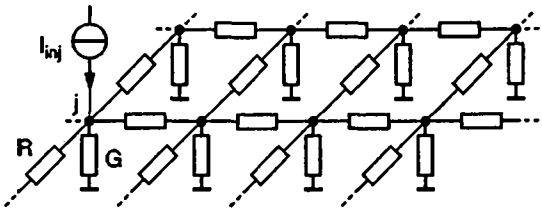


Fig. 5.1 Linear resistance-conductance (RG) network

5.2 MODIFIED RG NETWORK

The linear RG network exhibits a sharp and rapidly decreasing exponential characteristic. This means that most of the injected current I_{inj} is sunk near the injection node. It is therefore difficult to use such a network to define large neighbourhoods. This drawback can be eliminated by replacing the conductances with current limiting devices, e.g. the outputs of a current mirror of unit value I_u. Since the outputs of the mirror cannot sink more than I_u, the injected current I_{inj} must spread within an area of I_{inj}/I_u units. Furthermore, the current that spreads laterally through the resistors causes a voltage drop that forces this area to be circular.

In the realisation of a Kohonen map, the most interesting topology is the orthogonal array. However, this is the most difficult to analyse because it has no radial symmetry. For this reason, the analysis will be based on a continuous model of the network. The unit current sources I_u are replaced by an ideal constant current density J_u [A/m^2] that saturates at $V = 0$ and the horizontal discrete resistors by a sheet of resistance ρ [Ω/square]. The current flowing radially at radius r is the difference between the current injected and the current absorbed to ground within r, i.e.

$$I(r) = I_{inj} - \pi J_u r^2 \tag{5.1}$$

which gives the radius r_n of the neighbourhood

$$r_n = \sqrt{\frac{I_{inj}}{\pi J_u}} \tag{5.2}$$

The resistance of an annulus of radius r and width dr in polar coordinates is

$$dR(r) = \frac{\rho}{2\pi r}\, dr \tag{5.3}$$

and the decreasing potential from the injection node is described as

$$dV(r) = -\,dR(r)\,I(r) = \frac{\rho}{2\pi}\left(\pi J_u r - \frac{I_{inj}}{r}\right)dr \tag{5.4}$$

Solving this differential equation with the limit condition $V(r_n) = 0$, leads to the following normalized relation:

$$\frac{4\pi}{\rho I_{inj}} V(r) = \left(\frac{r}{r_n}\right)^2 - 1 - 2\ln\frac{r}{r_n} \tag{5.5}$$

Figure 5.2 shows that $V(r)$ still exhibits a rapidly decreasing slope, despite the modified current distribution. This is due to the linear lateral resistance and can be understood intuitively because the radial current $I(r)$ is decreasing (5.1) while the elementary resistance dR is decreasing (5.3).

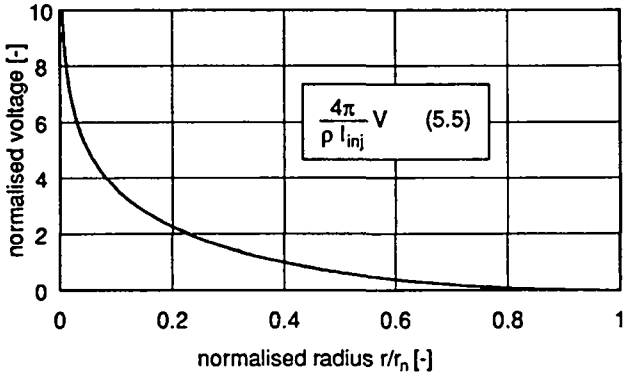


Fig. 5.2 Profile of the nonlinear continuous network with linear radial resistance.

Although the neighbourhood's size is well controlled by means of the ratio I_{inj}/I_u, the resulting characteristic is not suited to generate a wide range of neighbourhood sizes because of its rapidly increasing voltage with injected current. On the other hand, linear resistors are cumbersome elements in CMOS technologies and it is worth considering any nonlinear element which can lead to a more constant slope, i.e. some voltage-dependent resistance that decreases with increasing voltage.

5.3 DISCRETE TWO-DIMENSIONAL NONLINEAR NETWORK

The nonlinear resistance suggested above can be realised with transistors. With the help of (A.29), the resistance defined by the ratio of the differential voltage $\Delta V = (V_D - V_S)$ centred around a common mode voltage $V_{CM} = (V_D + V_S)/2$ to the drain current I_D is

$$R(V_{CM}) = \frac{1}{\beta n (V_{PR} - V_{CM})} \tag{5.6}$$

where V_{PR} is the pinch-off voltage of the resistors. The discrete network is shown in Figure 5.3 in its one-dimensional form for simplicity, together with the relevant characteristics of the transistors, i.e. the inverse of the normalised resistance of the resistors and the output characteristic of the mirror. The nonlinear resistance of M_R decreases with increasing voltage and is infinite below an "offset" voltage $V_{off} = V^+ - |V_{PR}|$. This offset voltage must be larger than V_{PG} to keep the current sources I_u saturated.

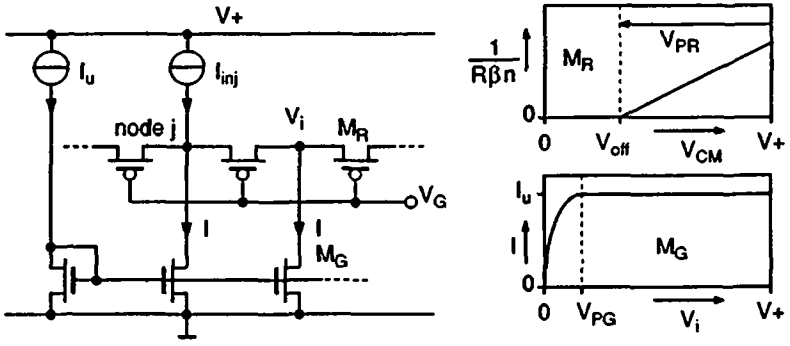


Fig. 5.3 Proposed nonlinear network and relevant characteristics of M_R and M_G.

If the drain voltage V_D of M_R is kept below V_{off}, the transistor enters into saturation and its drain current is controlled by V_S. This can happen only at the boundary of the neighbourhood if the remaining current is smaller than I_u, thus desaturating the NMOS mirror's output. If the drain to source conductances g_{DS} of both M_R and M_G are small enough, the probability of finding a node voltage between V_{PG} and V_{off} is very low. On the other hand, the voltage drop $V_{off} - V_{PG}$, which can be made quite large (one volt or more), allows an easy logic detection of the neighbourhood.

5.4 ANALYSIS OF THE TWO-DIMENSIONAL NONLINEAR NETWORK

For the same reasons as stated previously, the analysis is based on a continuous network in which the constant sheet resistance has been replaced by a voltage-dependent sheet resistance. This leads to the following incremental resistance:

$$dR = \frac{1}{\beta\, n\, V} = \frac{dr}{2\pi r\, \beta_0\, n\, V} \tag{5.7}$$

where $\beta_0 = \mu C_{ox}$ and the second expression develops β in terms of r. This resistance characteristic is consistent with that of the PMOS resistor, provided that the offset voltage $V_{off} = 0$. For this pupose, the constant current density J_u is assumed with a zero-volt saturation voltage. Introducing this new resistance characteristic into the differential equation leads to the following normalized solution, which is simply the square root of the previous one:

$$\sqrt{\left(\frac{2\pi\, n\beta_0}{I_{inj}}\right)}\, V(r) = \sqrt{\left(\frac{r}{r_n}\right)^2 - 1 - 2\ln\frac{r}{r_n}} \tag{5.8}$$

This relation is shown in Figure 5.4 together with relation (5.5) for comparison. There is also a vertical asymptote at the origin, but the voltage will remain finite if the current is injected into a finite area of radius r_{inj}. On the other hand, the slope is nearly constant except close to the origin.

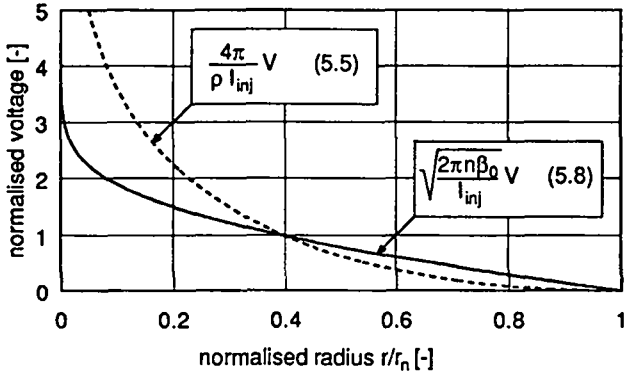


Fig. 5.4 Profile of the nonlinear continuous network with nonlinear radial resistance.

A link between the continuous network and the rectangular discrete network must be found to size the aspect ratio W_R/L_R of transistors M_R. To be consistent with the discrete network, the injection area of the continuous network is chosen such that $r_{inj}/r_n = \sqrt{I_u/I_{inj}}$, which corresponds to the area where one unit current I_u has been absorbed to ground. The correspondence of the four transistors M_R connected to the injection node is an annular transistor that has an inner radius $r_{inn} = r_{inj}/r_n$ and an outer radius $r_{out} = \sqrt{5} r_{inj}/r_n$, as shown in Figure 5.5. At the periphery of this annular transistor, five current units I_u have been absorbed to ground.

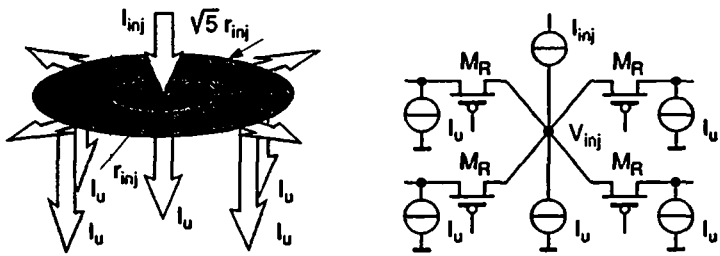


Fig. 5.5 Correspondence between the continuous and the discrete networks.

The voltage V_{inj} at the injection node is calculated using eqn (5.8) with the injection radius r_{inj}. On the other hand, the term 2π which appears in the normalisation factor and arises from the definition (5.7) of the incremental resistance, must be replaced by the equivalent value corresponding to the annular transistor defined in Figure 5.5. The equivalent aspect ratio of the annular transistor is $2\pi/\ln(r_{out}/r_{inn})$, and its counterpart for the discrete network is $4W_R/L_R$. The injection node voltage V_{inj} for the discrete network can therefore be approximated by

$$V_{inj} = \sqrt{\frac{I_{inj}}{2n\frac{W_R}{L_R}\beta_0 \ln 5}} \sqrt{\frac{I_u}{I_{inj}} - 1 - \ln\frac{I_u}{I_{inj}}} + V_{off} \qquad (5.9)$$

with $I_{inj} > 5I_u$, to ensure there are at least five cells in the neighbourhood.

This relation, which can be used to size the transistors M_R, is compared with simulation results in the next paragraph.

5.5 SIMULATIONS OF THE DISCRETE NETWORK

Figure 5.6 shows the simulated profiles of the neighbourhood for several values of injected current ranging from 10μA to 200μA. The unit current I_u = 1μA. SPICE model 2 has been used with the parameters listed in Table 5.1 (n is not a SPICE parameter). Long transistors have been used to minimize the effect of drain to source conductance g_{DS}. Relation (5.8) has been fitted with the largest profile at the two black dots, and shows the good similarity between large discrete networks and the continuous network.

Table 5.1

	n-type	p-type	unit
β_0	56	22	[μA/V2]
V_{T0}	1	0.9	[V]
n	1.3	1.2	[-]
W	80	160	[μm]
L	80	80	[μm]

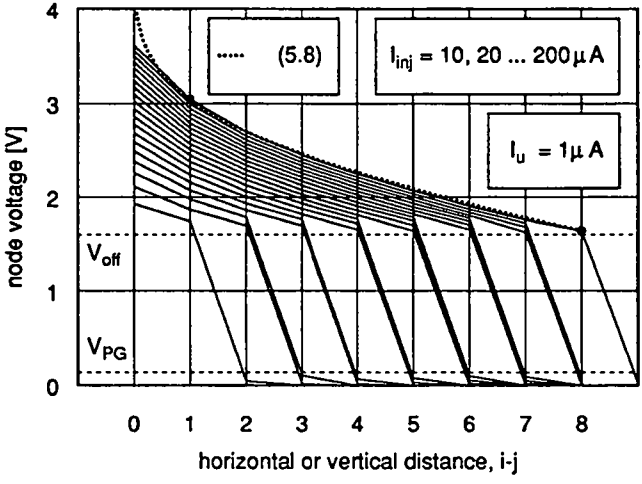


Fig. 5.6 SPICE simulations of the rectangular nonlinear network.

In Figure 5.7, eqn. (5.9) is compared with the injection node voltages arising from the simulations of Figure 5.6. The theoretical curve uses the values listed in table 5.1. It is interesting to notice that the two curves are very close for small neighbourhoods (< 100 cells). However, the slightly larger value given by eqn. (5.9) for larger neighbourhoods ensures a safety margin when sizing transistors M_R.

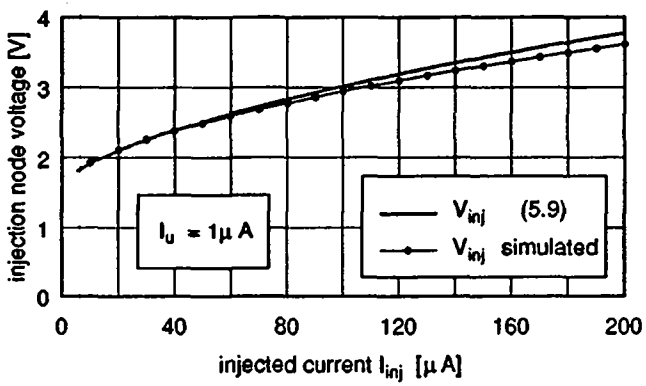


Fig. 5.7 V-I characteristic at the injection node.

5.6 EXPERIMENTAL RESULTS

An experimental circuit of 32 × 32 nodes has been made by B. Hochet. The technology used was ES2 2μm, the parameters of which are listed in Table 5.2 (n is not a SPICE parameter).

Table 5.2

	n-type	p-type	unit
β_0	50	18	[μA/V²]
V_{T0}	0.9	0.8	[V]
n	1.2	1.15	[-]
W	5.5	11	[μm]
L	5.5	5.5	[μm]

The unit current I_u = 100 nA. Figure 5.8 shows the oscillographs of three bubbles of different sizes. The first row shows the profiles (containing several cross sections but not all) of the bubbles. The two other rows are views from above representing the nodes' voltages using the Z input (luminance) of the scope (notice that the scale is not the same as in the first row). With the values indicated at the bottom of the figure, the bubbles should contain respectively 100, 300 and 800 units, however the high output conductances of the mirrors make these values about 10% smaller. It can be seen from these views that the bubbles are remarkably circular.

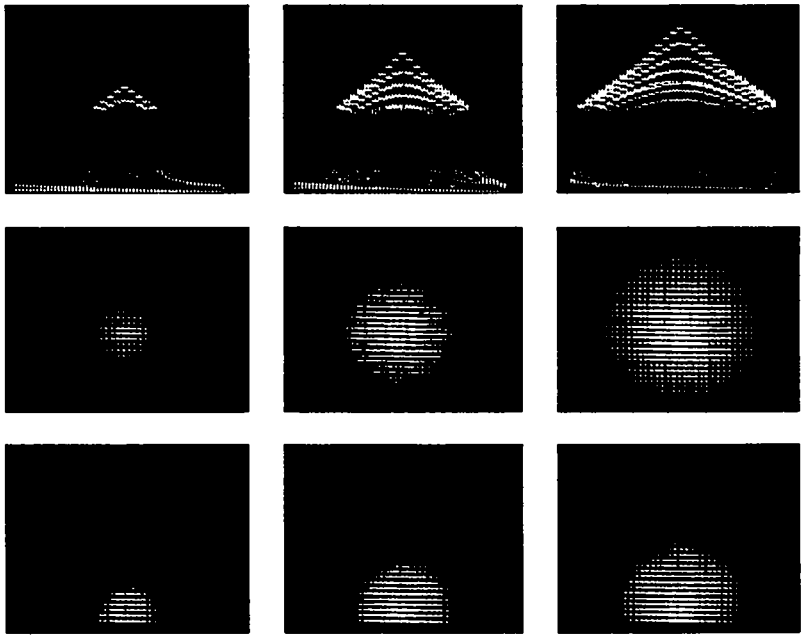


Fig. 5.8 Measured profiles and shapes of three bubbles.

To be consistent with the simplified algorithm of Kohonen, the radius of the bubble should be independent of its position in the network, even if the bubble is cut at the borders. However, if the network is not terminated, it is the bubble's area that is independent of the position. Although such an effect has not been simulated, it is not likely to be fatal to the map's behaviour. As for

the linear RG network, the ideal terminations that render the network uniform have not been found. The third row of Figure 5.8 shows the case of an injection at the border of the network, the terminations of which being made of transistors having twice the length of M_R, with their drains connected to ground. The result is a nearly independent radius versus position in the network, except for large bubbles.

5.7 MULTICHIP IMPLEMENTATION

If the nonlinear network is to be implemented with several chips, the resulting non-uniformity of the technology parameters will result in an alteration of the bubbles' shapes. Because it plays a key role in the formation of the neighbourhood by fixing the edge of the bubble, the pinch-off voltage V_{PR} of transistors M_R is the most important parameter to control among the chips. For this purpose, the polarization that controls V_P described in Appendix B is well adapted, and needs a single external voltage reference which can be applied in parallel to all the chips. The consequence of this polarisation is a spread for the resistors values being equal to the spread of the product $n\beta$, as shown by eqn. (A.28). This spread, which is about ±10% to ±15% for actual standard CMOS technologies, is not sufficient to induce unacceptable distortion in the bubble's shape.

Concerning the unit currents I_u, there are several possibilities. The best solution would be to distribute matched currents to every chip, but this needs an external circuit containing matched current mirrors, and is limited to polarising a small number of chips. However, such precision is not required, and the best compromise here is to use a current that is derived from a polarization that controls V_P driven from a common external voltage reference. The spread of these currents is that of the product $n\beta$, as for the resistors, and does not depend on the spread of the threshold voltages V_{T0}, which can reach ±150 mV.

5.8 EXPLOITING THE NEIGHBOURHOOD'S SHAPE

The most obvious purpose of the nonlinear network is the selection of the cells belonging to the neighbourhood. On the other hand, it has been pointed out in Chapter 2 that the organization phase of the learning process is greatly accelerated if the adaptation gain α is a decreasing function of r within

the neighbourhood. For this purpose, a conic-shaped neighbourhood has been proposed, which can be implemented by exploiting the nearly conic shape of the nonlinear network to set the learning gain. The extra circuitry needed to implement this function improves the performances of the Kohonen network by accelerating its organization phase.

5.9 TRANSIENT BEHAVIOUR

The settling time t_s of the bubble has to be known to establish the time period for the learning process. Before the injection current I_{inj} is applied to the network, all the nodes are at zero potential (see Figure 5.3). As for the DC characteristic, the analytical approach is based on a continuous model of the network which includes a surfacic capacitance C' [F/m^2] to ground. The injected current propagates radially and the current available to charge the elementary (annular) capacitance $dC(r) = 2\pi r C' dr$ is given by (5.1). The corresponding annular "node" must reach the potential V_{off} before the next node can start to charge. This takes the following elementary amout of time:

$$dt(r) = \frac{dC(r)\, V_{off}}{I(r)} = \frac{2\pi\, r\, C'\, V_{off}}{I_{inj} - \pi J_u r^2}\, dr \tag{5.10}$$

Integration of this equation gives the time t_s needed to reach the radius r

$$t_s(r) = -\frac{C'\, V_{off}}{J_u} \ln\left(1 - \left(\frac{r}{r_n}\right)^2\right) \tag{5.11}$$

with r_n given by (5.2). This relation, which is depicted in Figure 5.9, is somewhat optimistic and would be correct only if the lateral resistance was very low, thus leading to the "flattened" neighbourhood as shown in Figure 5.10 (a). Under this condition, it is interesting to notice that (5.11) does not depend on I_{inj} and this means that the settling time does not depend on the size of the bubble as long as the size is controlled by I_{inj} only, I_u being kept constant.

Although (5.11) is not exact, its vertical asymptote at $r = r_n$ shows clearly that the time needed to complete the neighbourhood is theoretically infinite. In the case of the real discrete network, the problem is similar because the last discrete nodes to be charged at the boundary of the neighbourhood dispose of a current in between zero and I_u, leading to indefinite rise time. However, to

ensure a proper operation of the synapses during the learning steps, a logical latching of the neighbourhood is necessary. On the other hand, except during the convergence phase, the exact number of cells is not important. Consequently, it is useless to wait for the slow boundary nodes before latching the neighbourhood. Under this condition, it is possible to calculate the rise time t_s using the simplified shape of the neighbourhood depicted in Figure 5.10 (b), in which the exact shape has been replaced by a cone.

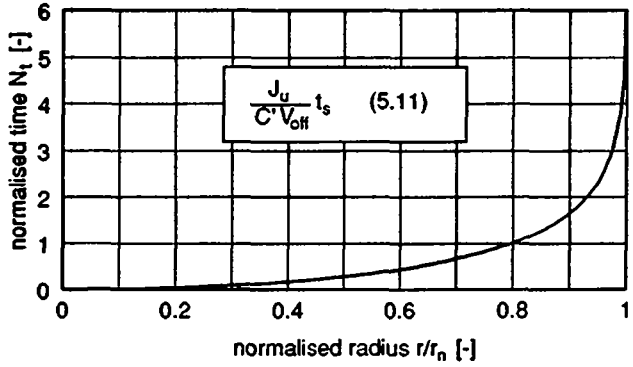


Fig. 5.9 Settling time evolution of normalised bubbles.

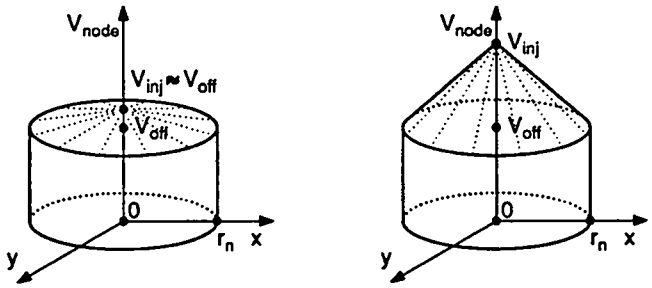
The rise time is the time needed to charge the "volume" of the neighbourhood. The contribution due to the cylindrical part is independent of the size, as confirmed by relation (5.11). The contribution of the cone depends on the size through its corresponding height V_{inj}-V_{off} only, which can be calculated by means of relation (5.9). Summing these two contributions leads to the following settling time:

$$t_s = N_t \frac{C'}{J_u}\left(V_{off} + \frac{V_{inj} - V_{off}}{3}\right) \tag{5.12}$$

where the normalized time N_t is the function of r/r_n shown in Figure 5.9. For the discrete network case, C' must be replaced by the node's capacitance C_n and the current density J_u by the point current I_u. On the other hand, considering that for the largest possible bubble the injection's node voltage cannot exceed the positive supply voltage V^+, the worst case settling time is given by:

$$t_s \leq N_t \frac{C_n}{I_u}\left(\frac{2}{3} V_{off} + \frac{1}{3} V^+\right) \tag{5.13}$$

For example, the normalized time N_t for a largest possible bubble of 1000 nodes, which ensures that all but the peripheral nodes are settled within t_s, can be calculated as follows: for $I_{inj}/I_u = 1000$, $r_n = 17.8$. The number of peripheral nodes is equal to the circumference $2\pi r_n \approx 112$. The remaining 888 nodes are contained within a radius $r = 16.8$. Therefore, $r/r_n = 0.94$ and the corresponding value of N_t is approximately 2.



(a) low lateral resistance "flat" network

(b) simplified network for settling time calculation

Fig. 5.10 Simplified shapes of the neighbourhood.

The knowledge of the fall time t_f is also necessary to establish the time period of the learning process. The worst case is simply the time needed to discharge the node capacitor C_n from V^+ (the maximum value) to ground by means of the point current I_u, i.e.:

$$t_f = \frac{C_n V^+}{I_u} \tag{5.14}$$

It is interesting to notice that the settling time is shorter than the fall time in any case, since V_{inj} is always larger than V_{off}. As an example, consider a node capacitor $C_n = 0.1$ pF (valid for a network made on a single chip), a unit current $I_u = 1$ μA and a supply voltage $V^+ = 5$ V. The sum of the settling time and the rise time is only 1 μs and a bubble containing 1000 cells can be generated with an injected current of only 1 mA. If now the network is made

of several chips, the node capacitance can reach 10 pF, the unit current may be increased to 10 μA, thus leading to a total time of 10 μs, but the bubble size could be limited by the injection current.

5.10 CONCLUSION

A nonlinear network has been proposed that is well adapted to the implementation of the learning neighbourhood for the Kohonen network. In spite of its extreme simplicity (only three transistors per cell), it features both the generation of a very wide range variable neighbourhood and the required shape for the implementation of a faster learning algorithm. Furthermore, the realised function is well suited to the low precision available in analogue circuits.

5.11 REFERENCES

[1] E. A. Vittoz, P. Heim, X. Arreguit, F. Krummenacher and E. Sorouchyari: "Analog VLSI Implementation of a Kohonen Map", Journées d'électronique 1989, Lausanne, 1989.

[2] C. Mead: "Analog VLSI and Neural Systems", Addison-Wesley, Reading, 1989, pp. 107-116, 339-351.

[3] J. Lazzaro, S. Ryckebusch, M. A. Mahowald and C. A. Mead: "Winner-Take-All Networks of Order N Complexity", in Proc. 1988 IEEE Conf. on Neural Information Processing - Natural and Synthetic, Denver, 1988, pp. 703-711.

[4] P. Heim, B. Hochet and E. Vittoz: "Generation of learning neighbourhood in Kohonen feature maps by means of simple nonlinear network", Electronics Letters, 31st January 1991 Vol. 27 No. 3.

CHAPTER 6

WINNER-TAKE-ALL WITH MULTICHIP COMPATIBILITY

6.1 INTRODUCTION

The Winner-Take-All (WTA) is a circuit that selects the largest of m input signals. A current-mode implementation of the WTA was first proposed by Lazzaro et al. [1]. This is certainly the simplest possible implementation of the WTA, since it needs only two transistors per cell. Its operation is based on global nonlinear inhibition, which is computed by means of a single wire for the entire circuit. Lazzaro analysed his circuit with the transistors operating in the weak inversion region. It will be shown that this may be a limitation if the common bus is loaded with large capacitance, and that better performance can be attained if the transistors are operating in strong inversion. In addition, the extreme simplicity of this circuit has some drawbacks. The most important of which is its incompatibility with multichip implementations because the variation in technological parameters, particularly that of the threshold voltage V_{T0}, can lead to excessive comparison errors and cannot be compensated by simple means. This chapter analyses the static and dynamic behaviour of the WTA, and proposes a modified version of the circuit that is multichip compatible and optimized to work with the large bus capacitance inherent to multichip implementations. A design procedure is also given that leads to a more comprehensive approach thanks to the use of normalized curves.

6.2 THE LAZZARO WINNER-TAKE-ALL IMPLEMENTATION

The original WTA circuit of Lazzaro is reproduced in Figure 6.1. In his analysis, Lazzaro exploited the internal node voltages V_k as the output values, but the drain currents of transistors M_B can also be exploited advantageously, particularly if current-mode circuits are used.

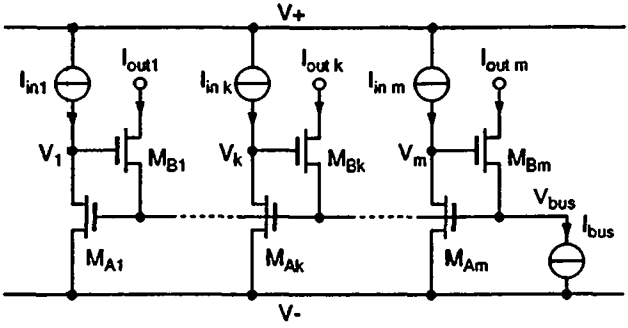


Fig. 6.1 Lazzaro Winner-Take-All circuit.

The operation of the WTA circuit is rather simple, and can be explained as follows. Each cell is made of two transistors M_A and M_B. Transistors M_A of all the cells have the same gate voltage $V_G = V_{bus}$, and therefore the same forward current I_{FA} (see Appendix A). Transistors M_B are source followers sharing the same bias current I_{bus}. For different input currents I_{in}, only the largest input current I_{ink} is able to maintain the corresponding transistor M_{Bk} as a follower, which in turn fixes the bus potential so that $I_{FA} = I_{ink}$. Indeed, all the other input currents are smaller than I_{FA}, which effectively cuts off the $(m\text{-}1)$ corresponding transistors M_B.

6.3 STATIC ANALYSIS OF THE WTA

As all the cells but the winner are cut off for any configuration of the input currents, provided the largest current is unique, the analysis can be based on the two-cell WTA depicted in Figure 6.2. The circuit is thus limited to the present and the future winning cells, since the other cells don't take part in the process. The schematic also shows the conductances g_{01} and g_{02} of the internal

nodes, which include the contributions of both the input current sources I_{in} and transistors M_A. These conductances are linear in first approximation within the voltage range limited by the saturation of both the input current sources and transistors M_A, which includes the whole operating range of interest. The most interesting parameter of the static behaviour of the WTA is its resolution, i.e. the difference $I_{res} = |I_{in1} - I_{in2}|$ needed to derive the whole bias current I_{bus} through a single transistor M_B.

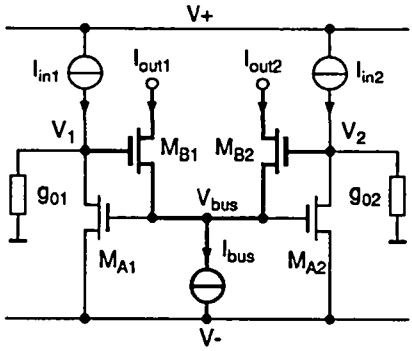


Fig. 6.2 Schematic diagram of a two-cell WTA circuit.

The static analysis is very simple when considering the differential pair made of transistors M_{B1} and M_{B2}, and biased by the current I_{bus}. Its transfer characteristic does not depend on the input currents, and can be characterized by its transconductance g_{mB} and its asymptotic saturation voltage V_{idsat} (see Appendix B, § B.4). Its exact shape is not important. The differential pair is therefore saturated for an input differential voltage equal to $\pm V_{idsat}$, which is reached for:

$$I_{res} = \left| I_{in1} - I_{in2} \right| = g_0 V_{idsat} \tag{6.1}$$

This relation completely defines the resolution of the WTA through the size of the transistors and the current values. If, in addition, the output conductance of the transistors is modelled using the Early voltage V_E, then g_0 is proportional to the drain current I_{in} (A.22), and a relative resolution of the WTA can be defined, which is independent of the input current level:

$$\frac{I_{res}}{I_{in}} = \frac{V_{idsat}}{V_E} \tag{6.2}$$

To increase the resolution, both the conductances and the asymptotic saturation voltage V_{idsat} must be decreased. Low conductances can be obtained using long-channel transistors or by cascoding both transistors M_A and the input current sources. The smallest saturation voltage of a differential pair is $2nU_T$ in weak inversion. These parameters also concern the dynamic behaviour of the circuit, as will be shown in the following paragraphs, and some compromises may be necessary between the static and the dynamic performances.

The input/output current transfer characteristics $I_{out1,2}(I_{in1}, I_{in2})$ have the same shape as the V/I transfer characteristic of the differential pair, with their inputs scaled by the term g_0. To illustrate this, simulation results using ESACAP [2] are reported in Figure 6.3. The model used was developed at the EPFL [3], and the parameters used are those of the SACMOS 3µm technology listed in Appendix D. The transistor parameters are summarized in Table 6.1, and will be used for all the simulations reported in this chapter.

Table 6.1

parameter	M_A	M_B	unit
W/L	10/10	100/5	[µm]
β	51.1	1240	[µA/V2]
I_D	1.0	100	[µA]
g_m	8.54	421	[µA/V]
C_G	70	350	[fF]

The current I_{in2} is fixed at 1µA, and the current I_{in1} is varied from 0.98 to 1.02 µA. The dotted curve representing $V_1 - V_2$ confirms that the conductances g_0 are linear within the range of interest. This conductance has a value of 8 nA/V. Extrapolating the saturation asymptote of the output currents gives $I_{res} = 2.67$ nA. Using relation (6.1), yields $V_{idsat} \approx 333$mV, which is in accordance with the calculated value:

$$V_{idsat} = \sqrt{\frac{n\, I_{bus}}{\beta_B}} = \sqrt{\frac{1.4 \times 100\text{E-}6}{\frac{99.2}{4.0}\,50\text{E-}6}} = 336\ \text{mV}$$

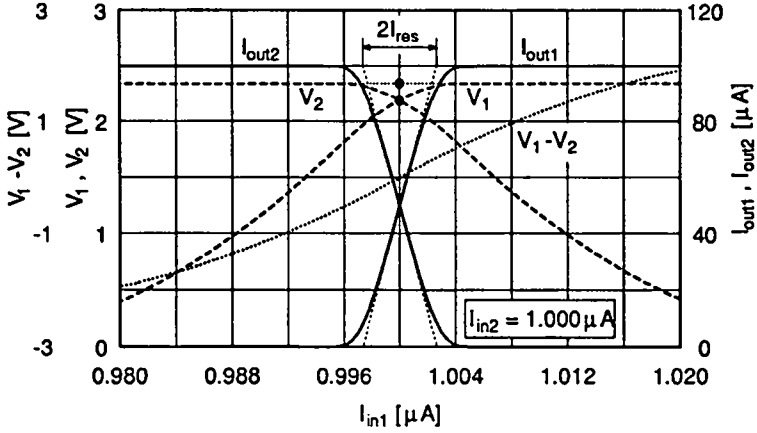


Fig. 6.3 Simulation of a two-cell WTA with $(W/L)_{MA}$ = 10/10 and $(W/L)_{MB}$= 100/5, in µm.

The simulation clearly shows that the current transfer characteristics are derived from that of the differential pair. The I/V transfer characteristics of the two-cell WTA, which are represented by the dashed curves, are also related to the transfer characteristic of the differential pair since the difference $V_1 - V_2$ depends linearly on the input current difference $I_{in1} - I_{in2}$. These I/V transfer characteristics are cumbersome to calculate and of little interest if the current outputs of the WTA are used. Therefore, these functions can be well enough characterized by considering that they saturate at the same abscissa as the output currents for the winning cell, and decrease linearly with a slope g_0 for the losing cell. In addition, the voltage difference between the saturated value $V_{max}(I)$ (the upper black dot on the figure) and the equilibrium point $V_{eq} = V_1 = V_2$ (the lower black dot on the figure) can be estimated simply by the difference between the gate voltage of M_B driving I_{bus} and the gate voltage of M_B driving $I_{bus}/2$, which gives respectively:

$$V_{max} - V_{eq} = n\, U_T \ln 2 \qquad \text{in weak inversion} \qquad (6.3)$$

$$V_{max} - V_{eq} = \left(\sqrt{2} - 1 \right) V_{idsat} \qquad \text{in strong inversion} \qquad (6.4)$$

6.4 STABILITY ANALYSIS OF THE WTA

The WTA must be stable and should not exhibit damped oscillations in response to input changes. This paragraph analyses the stability criteria of the circuit. For this purpose, the analysis can be based on two different circuits, as

shown in Figure 6.4. The first circuit contains two cells that are at equilibrium, i.e. both input currents are equal to I. Although statically stable, this situation is not very representative of the circuit behaviour, particularly if its resolution is high. Indeed, after a transient it is most likely that a single cell takes all the bias current I_{bus}. This situation is the most probable, and the stability analysis will show that it is also the most constraining.

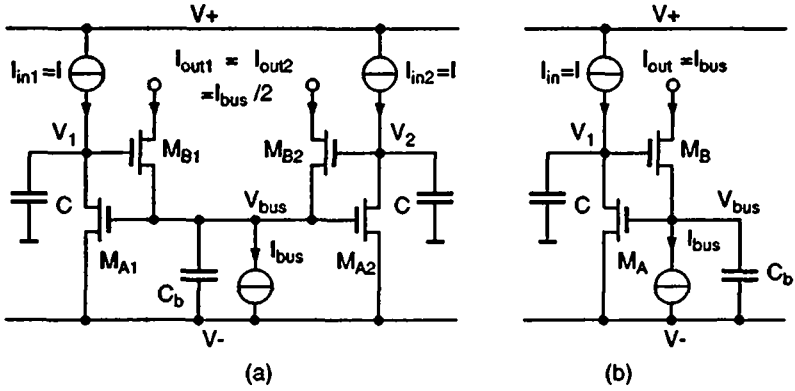


Fig. 6.4 Schematic diagrams of (a) a two-cell WTA near equilibrium and (b) a single cell (the winner), including the node capacitances used for the stability analysis.

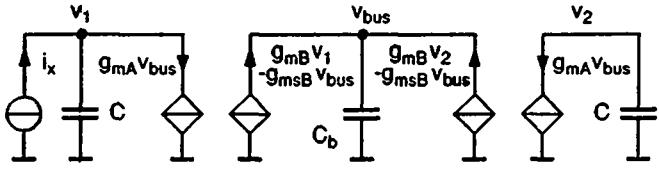


Fig. 6.5 Small-signal equivalent circuit for the two-cell WTA of Figure 6.4 (a).

The small signal equivalent circuits of the two circuits above are shown in Figures 6.5 and 6.6 respectively. They include an input current source i_X used to calculate the transimpedance $Z_X = v_{bus}/i_X$. The stability condition of the circuits can be imposed on the poles of Z_X, which must be real to avoid damped oscillations.

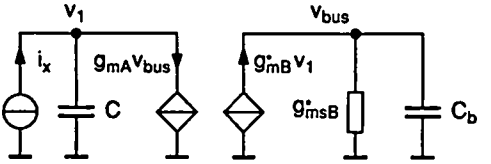


Fig. 6.6 Small-signal equivalent circuit for the WTA single-cell of Figure 6.4 (b).

The impedances Z_{xa} for the two-cell circuit and Z_{xb} for the single cell circuit are respectively:

$$Z_{xa} = \frac{1}{2g_{mA}} \frac{1}{1 + \frac{g_{mSB}\,C}{g_{mA}\,g_{mB}} p + \frac{C\,C_b}{2\,g_{mA}\,g_{mB}} p^2} \tag{6.5}$$

$$Z_{xb} = \frac{1}{g_{mA}} \frac{1}{1 + \frac{g_{mSB^*}\,C}{g_{mA}\,g_{mB^*}} p + \frac{C\,C_b}{g_{mA}\,g_{mB^*}} p^2} \tag{6.6}$$

where g_{mB} and g_{mSB} refer to transistors M_B biased with a current $I_{bus}/2$, and g_{mB}^* and g_{mSB}^* refer to transistor M_B biased with a current I_{bus}. Since $g_{mS} = ng_m$, the conditions to have real poles can be formulated as follows, for g_{mB} and g_{mB}^* respectively:

$$g_{mB} \geq \frac{2}{n^2} \frac{C_b}{C} g_{mA} \tag{6.7}$$

$$g_{mB^*} \geq \frac{4}{n^2} \frac{C_b}{C} g_{mA} \tag{6.8}$$

These conditions can be reformulated by substituting the values of the transconductances given by relations (A.20). The result is the same for the two circuits if the transistors are operating in weak inversion:

$$I_{bus} \geq \frac{4}{n^2} \frac{C_b}{C} I \tag{6.9}$$

Conversely, different results are obtained for the two-cell and the single cell circuits respectively, if the transistors are operating in strong inversion:

$$I_{bus} \geq \frac{8}{n^4} \frac{C_b^2}{C^2} \frac{\beta_A}{\beta_B} I \tag{6.10}$$

$$I_{bus} \geq \frac{16}{n^4} \frac{C_b^2}{C^2} \frac{\beta_A}{\beta_B} I = \left(\frac{4}{n^2} \frac{C_b}{C} \right)^2 \frac{\beta_A}{\beta_B} I \tag{6.11}$$

The condition for the single cell circuit is more restrictive and must therefore be used for the design. In addition, the squared term of (6.11) is the same as that appearing in (6.9), which gives more consistency to the results. On the other hand, the quality factor and the resonant (or cut-off) frequency of the single cell equivalent circuit are given, respectively, by:

$$Q = \frac{1}{n} \sqrt{\frac{g_{mA}}{g_{mB^*}} \frac{C_b}{C}} \tag{6.12}$$

$$f_0 = \frac{1}{2\pi} \sqrt{\frac{g_{mA} g_{mB^*}}{C\ C_b}} \tag{6.13}$$

which shows that for a given quality factor ($Q \leq 0.5$ to get real poles), the bandwidth increases with the current level.

As an example, the maximum number of cells for a WTA implemented with transistors having the parameters listed in Table 6.1 can be calculated using condition (6.11), since the transistors are in strong inversion. The circuit exhibits a first order response for $C_b \leq 24C = 8.4\text{pF} = 120\ C_{GA}$. If some margin is taken into account for the bus capacitance, this design is suitable for a hundred cells on a single chip. The cut-off frequency for the single cell is $f_0 = 560$ KHz

Although the stability analysis is rather simple, the design of the circuit for given specifications is not easy since the parameters appearing in the conditions of stability are not independent. There is a particular case, however, which makes the stability condition independent of the inversion coefficient IC of the transistors. It is certainly not an optimal case, but gives a good basis for the design approach. The conditions are the following:

- transistors M_A and M_B have the same inversion coefficient IC, which means that $I_{bus}/\beta_B = I/\beta_A$

- transistors M_A and M_B have the same length L, which further restricts the above condition to $I_{bus}/W_B = I/W_A = K_I$, with:

$$K_I = IC\,\frac{2\,n\,\beta_0\,U_T^2}{L} \tag{6.14}$$

- the capacitances C and C_b are dominated by the gate capacitance of the respective transistors, i.e. $C = K_C W_B$ and $C_b = mK_C W_A$, m is the number of cells and K_C is given by:

$$K_C = L\,C'_{ox} \tag{6.15}$$

With these assumptions, the transconductances of both transistors, expressed continuously from weak to strong inversion, using (A.25) or (A.26), are given by the following relation:

$$g_{mi} = \frac{I_i}{nU_T\sqrt{1 + 0.5\sqrt{IC} + IC}} = \frac{I_i}{nU_T\,G(IC)} \tag{6.16}$$

Substituting the respective values of $g_{mB}{}^*$ and g_{mA}, which have the same denominator, in relation (6.8), yields:

$$I_{bus} \geq \frac{4}{n^2}\frac{C_b}{C}\,I \tag{6.17}$$

This stability condition is similar to (6.9), but is valid from weak to strong inversion. Using now the above-mentioned conditions concerning currents, transistor sizes and capacitances, yields:

$$W_B \geq \frac{4}{n^2}\frac{mW_A}{W_B}\,W_A \quad \Rightarrow \quad \frac{W_B}{W_A} = \frac{I_{bus}}{I} \geq \frac{n}{2}\sqrt{m} \tag{6.18}$$

which shows that both the ratio of the transistor widths and the ratio of the currents increase with the square root of the number of cells. If I is the maximum input current value, then the circuit remains stable for smaller values. On the other hand, since the capacitance values are fixed by the size of the transistors, the highest frequency response is obtained with the largest possible current level. Indeed, using the above relations, the cut-off frequency can be formulated as follows:

$$\omega_0 = \frac{1}{n\,U_T\,G(IC)}\frac{K_I}{K_C}\frac{1}{\sqrt{m}} = \frac{2\,\mu\,U_T\,IC}{L^2\sqrt{m}\,G(IC)} \tag{6.19}$$

where μ is the carrier mobility. Therefore, in strong inversion the cut-off frequency asymptotically increases with the square root of the current level.

6.5 TRANSIENT ANALYSIS AND TIME RESPONSE

Because the biasing conditions of the cells are modified during a transient, the small-signal equivalent circuit cannot be used to predict the dynamics of the circuit. The transient analysis can however be carried out independently of the transistor characteristics. For this purpose, a simulation can help the explanation, and the two-cell WTA of Figure 6.2 will be used with the parameters listed in Table 6.1. There are two ideal cases to be distinguished a priori: a cell can win because its input current becomes larger than that of the other cell, or it can win because the input current of the other cell becomes smaller than its own input current. The two situations are simulated together by keeping the input current of cell number 2 constant and equal to 1 µA, while changing the input current of cell number 1 by $\pm I_w$ from this value. In the initial situation ($I_{in1} < I_{in2}$), provided the input current difference is much larger than the resolution current I_{res}, $V_2 = V_{max}(I_{in2})$ and V_1 is close to V- (0 V in the following simulations) since M_{A1} is desaturated. When I_{in1} becomes larger than I_{in2}, their difference $I_w = I_{in1} - I_{in2}$ is available to charge the capacitive node V_1 up to the value $V_{max}(I_{in1})$, and is then available to discharge the capacitive node V_2 down to a value close to V-. A qualitative distinction can be made between large and small input current differences I_w. If I_w/g_0 is large compared to the voltage variation $\Delta V \approx V_{max} - V$-, the equivalent current source is quasi-ideal and the voltage transient $V_1 - V_2$ is linear in time. On the contrary, for low I_w, the response is exponential with a time constant $C/g_0 = CV_E/I$. The two cases have been simulated and are reported in Figures 6.7 and 6.8 respectively. The total bus capacitance C_b has been set to 8.4 pF, which corresponds to the stability limit given by relation (6.11).

In the simulation of Figure 6.7, I_w/g_0 is larger than $V_{max} - V$-, and the voltage transient $V_2 - V_1$ is linear in time. In the first transient, cell 1 wins

because its input current becomes larger than the input current of cell 2. The winning time t_w is reached when $V_1 = V_2 = V_{eq}$, and is simply:

$$t_w = \frac{C\,\Delta V}{I_w} \approx \frac{C\left(V_{max} - V_{\text{-}}\right)}{I_{in1} - I_{in2}} \tag{6.20}$$

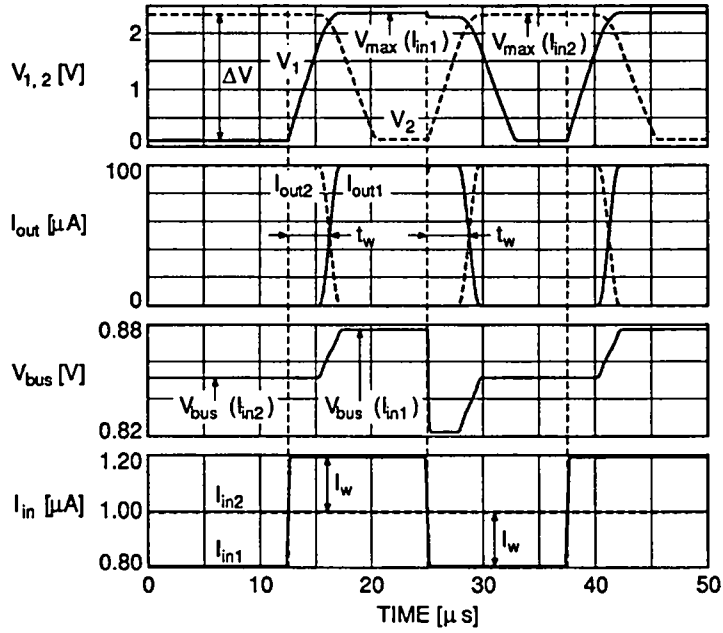


Fig. 6.7 WTA transient simulation with $I_w = 0.2$ μA.

In the second transient, cell 2 wins because the input current of cell 1 becomes smaller than its own input current. The difference with respect to the first transient is noticeable on the bus voltage V_{bus}. At the beginning of the transient, an initial current $2I_w$ discharges the node V_1, and the source of M_{B1} *follows* until V_{bus} reaches the value for which $I_{FA} = I_{in1}$. The winning current I_w is now available at cell 2 to charge the node V_2, which leads to a transient similar to that of the previous case. Therefore, the present winner first loses before the new winner wins, as shown on the figure.

Under the stability condition (6.17), $I_{bus}/C_b \approx I/C$ so that $I_{bus}/C_b >> 2I_w/C$. The voltage drop of both V_1 and V_{bus} during the losing transient are therefore

dominated by the slew-rate $2I_w/C$ of the internal node. In addition, since this voltage drop is much smaller than ΔV, the losing time is generally much smaller than the winning time, and so can be neglected.

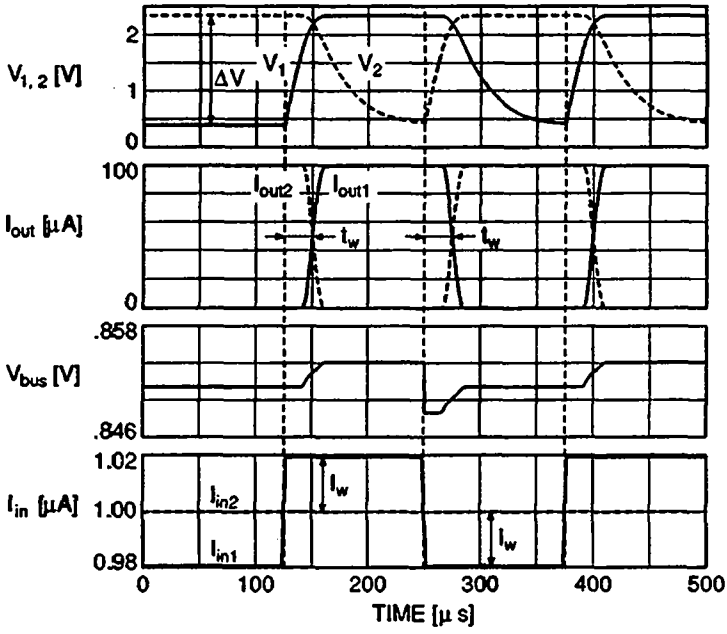


Fig. 6.8 WTA transient simulation with $I_w = 0.02$ μA.

The simulation of Figure 6.8 uses small input current steps of 20 nA, which leads to an exponential variation of $V_2 - V_1$. The winning time can be calculated from the simple linear equivalent circuit made of a current source I_w, a conductance g_0 and a capacitor C, which leads to:

$$t_w = -\frac{C}{g_0} \ln\left(1 - \frac{g_0 \Delta V}{I_w}\right) \tag{6.21}$$

The simulated winning time is somewhat shorter than the value predicted by this relation. This can be explained by the relatively small step current. Indeed, as the major part of the initial current difference $I_{in2} - I_{in1}$ is absorbed by the conductance, transistor M_A is not desaturated, and the effective step current is closer to $2I_w$ than I_w. If the input step current is further reduced, the relative

winning time is also reduced thanks to the reduction of ΔV. Therefore, relation (6.20) can be safely used to estimate the winning time t_w for any input step current. The losing time could be deduced from this relation in the same way as before, but since the voltage change at node V_1 is very small, this time is relatively shorter than for large input step currents, and can be neglected, as shown in the figure for $t = 250$ µs.

The transient analysis was based on a two-cell WTA. It may be useful to see what happens qualitatively when several cells are in competition. For this purpose, a simulation result including a third cell with a constant input current is reported in Figure 6.9.

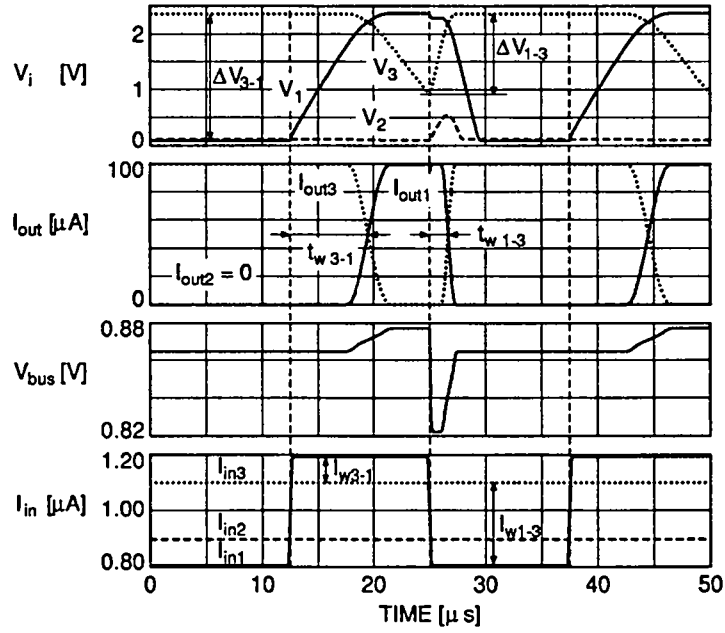


Fig. 6.9 Three-cell WTA transient simulation with two constant inputs.

This result shows that cell 2 never wins and does not interfere in the computation of the winner. Furthermore, the winning time is a function of the difference between the input currents of the future and the present winners. This assumption is confirmed with the following simulation result for which all three input currents are changed simultaneously.

The result of Figure 6.10 further confirms that the winning time is a function of the difference between the input currents of the future and the present winners, even if both currents are changed. This generalizes the definition of the winning current I_w. The result also leads to the conclusion that a loser transient is present when the input current of the present winner is decreased.

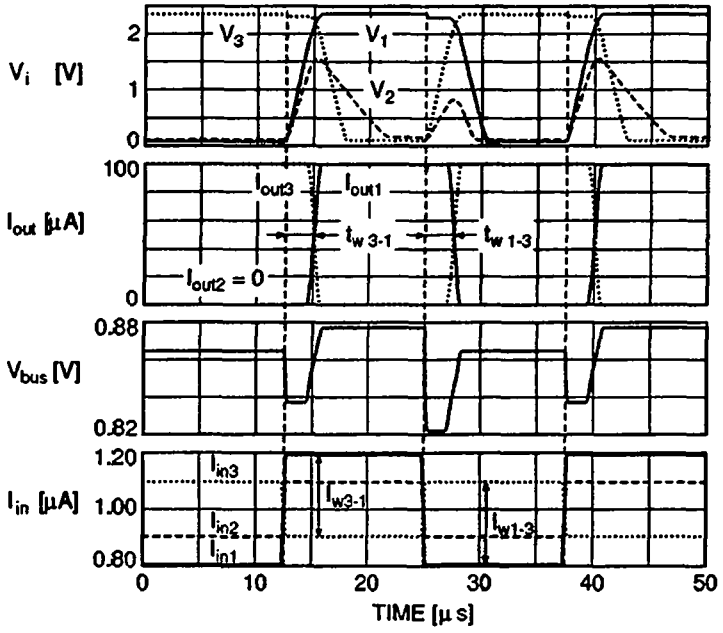


Fig. 6.10 Three-cell WTA transient simulation with variable input currents.

The transient analysis, as well as the qualitative results on the behaviour of the WTA, show that the circuit is well adapted for the selection of the largest proximity in analogue implementations of a Kohonen network. As a matter of fact, the winning time is roughly proportional to the inverse of the difference between the input currents of the future and the present winners I_w. Since the learning algorithm requires that the input vectors are selected randomly from the data base, large values of I_w are the most probable. When two successive input vectors are close together, the winning time can be longer than the fixed wait time, and the wrong cell can be selected. This is however

acceptable as long as the selected cell is included within the bubble, which gives a basis for estimating the wait time.

6.6 WTA SUITED TO MULTICHIP IMPLEMENTATIONS

As already suggested, the simple WTA circuit of Lazzaro is not suited to multichip implementations, because the large spread of threshold voltages among chips leads to unacceptable comparison errors. This drawback can be avoided if the transconductance implemented with M_A is replaced by a better suited element. Since the bus carries the same voltage to all the chips, the multichip scheme presented in Chapter 3 can be used, and M_A must be replaced by a differential pair.

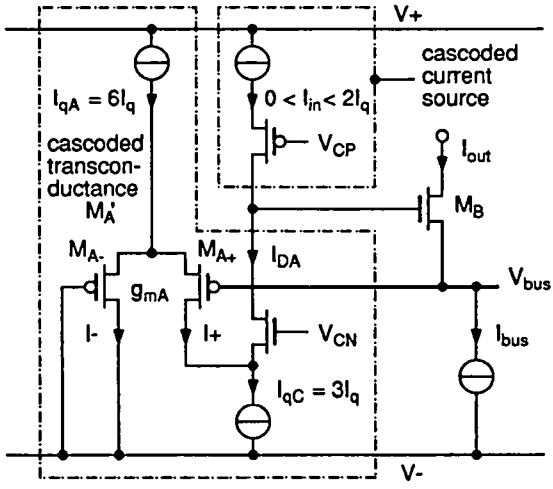


Fig. 6.11 Modified WTA circuit, compatible with the multichip scheme.

The modified version of the WTA has basically the same structure as the original, except transistor M_A has been replaced by a linear transconductor M_A' made of a differential pair and a folded cascode structure, as shown in Figure 6.11. Since the transconductor is cascoded, the input current source has also been cascoded to ensure a high resolution. The operation of the circuit as well as the choice of the bias current values are easily explained with the help of the diagram of Figure 6.12.

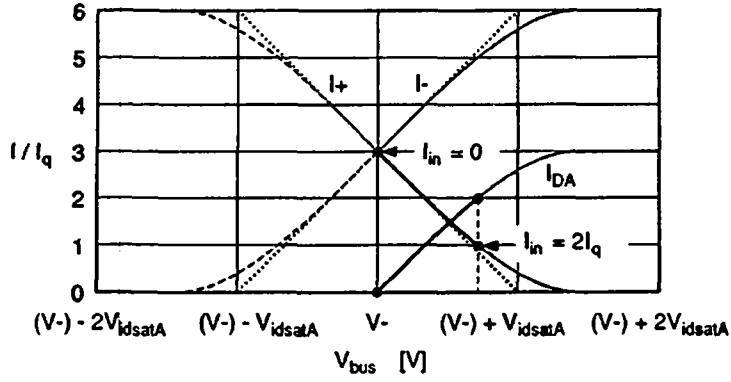


Fig. 6.12 *Currents of the modified WTA circuit versus V_{bus}.*

The input current range has been chosen between 0 and $2I_q$, where I_q is a bias current issued from a polarization that controls the saturation voltage V_{idsat} of a differential pair (see Appendix B). Therefore, the current to voltage conversion V_{bus} [MAX(I_{in})] performed by the WTA must use a transconductance based on a differential pair that is biased with a multiple of I_q to be compatible with the multichip scheme proposed in Chapter 3. The bias current I_{qA} of g_{mA} has been chosen equal to $6I_q$ for two reasons. First, only half of this bias current is available at the drain of M_{A+}, since the bus voltage V_{bus} is not allowed to be more negative than V-. Second, the input current must not be larger than the maximum drain current of M_{A+} to prevent the differential pair from reaching saturation. These two conditions are fulfilled using the bias currents of Figure 6.11. The transfer characteristic $I_{DA}(V_{bus})$ of the resulting transconductance is shown in Figure 6.12, and ranges from V- for $I_{in} = 0$ to about $(V\text{-}) + 2V_{idsatA}/3$ for $I_{in} = 2I_q$.

This modified version of the WTA has some advantages over the original version. The transconductance g_{mA} is constant and independent of the input current level, and so is the dynamic behaviour of the circuit. Relation (6.8) is suitable for setting the stability conditions for this WTA, because the effects of the parasitic poles due to the cascode transistors are negligible. On the other hand, the transient winning time t_w is approximately proportional to $C\Delta V$. Therefore, since the transconductance g_{mA} of this WTA is exempt of the threshold voltage inherent to the version of Lazzaro in strong inversion, the body effect of transistors M_B is reduced and its gate voltage is smaller for a

given transconductance g_{mB}*, which leads to a substantial reduction of ΔV. Conversely, if the same ΔV is used, the width of M_B can be reduced and so is the capacitance C. Since multichip implementations can have a large bus capacitance C_b, due to the off-chip bus interconnexions, the above properties are also helpful for getting larger transconductances g_{mB}* with smaller transistors M_B.

The accuracy of the WTA depends essentially on the mismatch of the transistors. The input offset voltage of the differential pair and the mismatch of the bias current source I_{qC} both contribute to constant error which is independent of the input current level. A mismatch in the bias current source I_{qA} affects the transconductance value g_{mA} and leads to an error that depends linearly on the input current level.

6.7 DESIGN PROCEDURE

As already stated, the design parameters are not independent, and it may be particularly laborious to get an optimal design. On the other hand, the input current level may be dictated by other circuits and the precision requirements can have a significant importance on the choice of g_{mA}. These choices in turn fix the bus capacitance C_b, which can be evaluated by:

$$C_b = m\, C_{GA} + C_{board} \tag{6.22}$$

where C_{GA} is the gate capacitance of M_{A+} and C_{board} is the total capacitance of the board, including the contributions of the bounding and the packaging of the chips. Therefore, if C_b and g_{mA} are fixed, then the stability condition can be reformulated as follows:

$$g_{mB}{}^{*}C \geq \frac{4}{n^2}\, C_b\, g_{mA} = K_S = \text{constant} \tag{6.23}$$

The design procedure assumes that C_b and g_{mA} are known a priori, and the evolution of the performance will be analysed as a function of the inversion coefficient IC_B of M_B under the constraint $g_{mB}{}^{*}C = K_S$, or equivalently $Q = 0.5$. Using the interpolation function (A.24), the transconductance g_{mB}* can be expressed as:

$$g_{mB^*} = \frac{I_{bus}}{nU_T\, G\left(IC_B\right)} \tag{6.24}$$

The internal node capacitance C and the inversion coefficient IC_B of transistor M_B are respectively given by:

$$C = W_B L_B C'_{ox} \tag{6.25}$$

$$IC_B = \frac{I_{bus}}{2n\,\beta_B\, U_T^2} = \frac{I_{bus}}{2n\,\frac{W_B}{L_B}\,\mu\, C_{ox}\, U_T^2} \tag{6.26}$$

Using the above constraint and relations, the following parameters can be formulated as a function of IC_B:

$$W_B = \sqrt{\frac{G\left(IC_B\right)}{IC_B}}\, NF_W \tag{6.27}$$

$$C = \sqrt{\frac{G\left(IC_B\right)}{IC_B}}\, NF_C \tag{6.28}$$

$$G_{mB^*} = \sqrt{\frac{IC_B}{G\left(IC_B\right)}}\, NF_g \tag{6.29}$$

$$f_0 = \sqrt{\frac{IC_B}{G\left(IC_B\right)}}\, NF_f \tag{6.30}$$

where NF_i are normalizing factors respectively equals to:

$$NF_W = \sqrt{\frac{K_S}{2\,\mu\, C_{ox}^2\, U_T}} \tag{6.31}$$

$$NF_C = L_B \sqrt{\frac{K_S}{2\,\mu\, U_T}} \tag{6.32}$$

$$NF_g = \frac{1}{L_B}\sqrt{2\, K_S\, \mu\, U_T} \tag{6.33}$$

$$NF_f = \frac{n}{4\,\pi\,C_B\,L_B}\sqrt{2\,K_S\,\mu\,U_T} \tag{6.34}$$

These results show that the internal node capacitance C and the cut-off frequency f_0 evolve in the inverse ratio, both leading to better performance in strong inversion. The normalized relations C/NF_C and f_0/NF_f are shown in Figure 6.13.

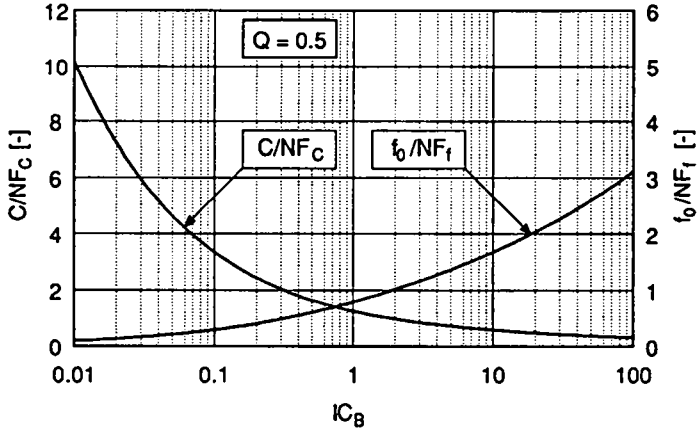


Fig. 6.13 Normalized internal node capacitance and cut-off frequency of the WTA, as a function of the inversion coefficient IC_B.

As C decreases, the gate voltage of M_B, which is approximately equal to the internal node voltage swing ΔV, is increasing. In order to show the evolution of the winning time t_w, the continuous formulation of the drain current from weak to strong inversion proposed by Oguey [4], can be used in estimating ΔV:

$$\Delta V \approx V_{GB} = 2nU_T \ln\left(\exp\sqrt{IC_B} - 1\right) + nV_S + V_{T0} \tag{6.35}$$

This relation is depicted in Figure 6.14 with $V_S = 1$ V and $V_{T0} = 0.6$ V as an example (in fact, V_S is function of g_{mA} and the input current level, and its maximum value must be chosen for the calculation). The normalized curve $\Delta VC/NF_C$, which is in first approximation proportional to the winning time t_w for a given winning current I_w, is also shown in the figure for comparison. This curve must be interpreted carefully. In strong inversion, if V_S and V_{T0}

were equal to zero, the gate voltage V_{GB} would be a function of the square root of IC_B and ΔVC function of $IC_B{}^{1/4}$, thus increasing with IC_B. In this case, there would be an optimum winning time for $IC_B = 1$. However, because of the extra term $nV_S + V_{T0}$, this optimum is practically transposed to a much higher value of IC_B. To reach the theoretical optimum for $IC_B = 1$, which may be important to limit the power consumption, additional circuitry should be used to clamp the internal node voltage of all the losing cells to a value $V_{idsat}(IC_B)$ smaller than V_{max}.

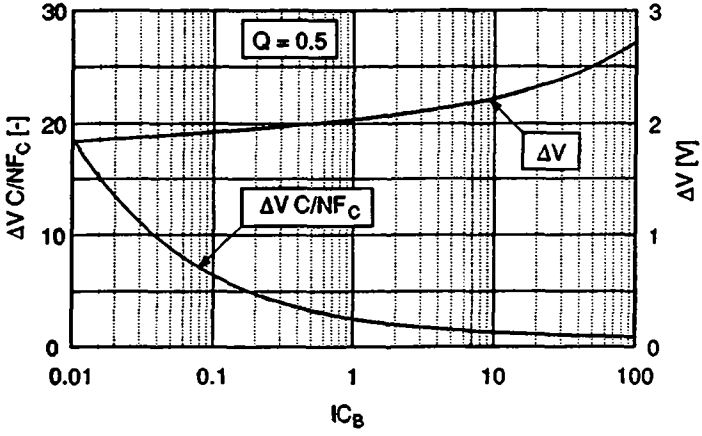


Fig. 6.14 Approximation of ΔV by the gate voltage of transistor M_B for a particular case, and evolution of the winning time t_w, as a function of the inversion coefficient IC_B.

Finally, the bus current I_{bus} is given by:

$$I_{bus} = \sqrt{IC_B\, G\,(IC_B)}\; NF_{bus} \tag{6.36}$$

$$\text{with} \quad NF_{bus} = \frac{n}{L_B}\sqrt{2\,\mu\, U_T^3\, K_S} \tag{6.37}$$

Therefore, in strong inversion, the bus current increases with $IC_B{}^{3/4}$, which may seem excessive because the winning time t_w is not improved. On the other hand, large bus currents may be necessary when the bus capacitance is large. In this case, strong inversion must be used to limit the width of transistors M_B, which would be too large in weak inversion. When both the number of cells and the bus capacitance are limited, for example in single chip

implementations, the best compromise is to operate M_B in the medium inversion region, i.e. IC_B between 1 and 10, where the bus current level is not excessive.

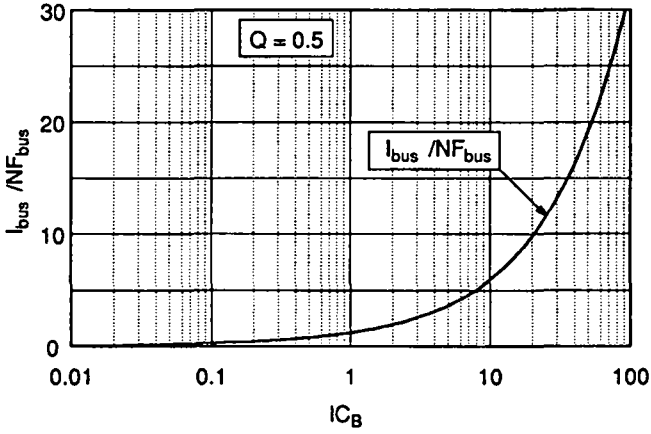


Fig. 6.15 Normalized bus current I_{bus}, as a function of the inversion coefficient IC_B.

6.8 CONCLUSION

In this chapter, the analysis of the Winner-Take-All has been reconsidered from a point of view different from those that can be found in the literature. The static and dynamic behaviour of the circuit is simply described, and a simplified design procedure using some justified assumptions has been described. The latter leads to a more comprehensive approach thanks to the use of normalized curves as a function of the inversion coefficient IC_B of transistor M_B. On the other hand, a modified version of the WTA has been proposed which is multichip compatible and well suited to handle a large bus capacitance.

6.9 REFERENCES

[1] J. Lazzaro, S. Ryckebusch, M. A. Mahowald and C. A. Mead: "Winner-Take-All Networks of Order N Complexity", in Proc. 1988 IEEE Conf. on Neural Information Processing - Natural and Synthetic, Denver, 1988, pp. 703-711.

[2] P. Stangerup: "ESACAP–a PC-implemented general-purpose circuit simulator", IEEE Circuits and Devices Magazine, 1988, 4, pp. 20-25.

[3] C. C. Enz: "High Precision CMOS Micropower Amplifiers", Ph.D. Thesis No 802, Ecole Polytechnique Fédérale de Lausanne, 1989.

[4] H. J. Oguey ans S. Cserveny: "MOS modelling at low current density", Summer Course on Process and Device Modelling, ESAT Leuven-Heverlee, Belgium, June 1983.

CHAPTER 7

ZERO-CROSSING CURRENT RECTIFIER WITH SIGN RECOVERY

7.1 INTRODUCTION

The primary objective of this rectifier was to compute Manhattan distance in a VLSI analogue implementation of a Kohonen map, while avoiding accurate device matching requirements. The idea of this circuit [1] was born after several failures using current CMOS circuits for computing absolute value. These circuits suffer from several major drawbacks, one of which being the lack of precision at the zero-crossing point of their DC transfer characteristic. Circuits based on the translinear principle, implemented with compatible lateral bipolar transistors (CLBT) [2], have a good precision but occupy large circuit area. Another implementation based on the MOS generalized translinear principle [3] suffers from an inconvenient input configuration. The circuit proposed in this chapter has a flat section at the origin of its DC transfer characteristic that depends on transistor output conductances only. When associated with a differential pair, the circuit constitutes a transconductance rectifier whose equivalent input offset voltage is similar to that of a conventional single-ended OTA.

7.2 CIRCUIT DESCRIPTION

The basic configuration of the complete OTA-based circuit is shown in Figure 7.1(a). The output structure consists of a current mirror M_1-M_3 whose input connexion is selected either by switch M_5 or by switch M_7 , and clamp transistors M_6-M_8 that drive the output current I_0. The bias string made of current source $2I_q$ and transistor M_{10} fixes the gate voltage of M_6-M_8 so that their source voltages are near ground (GND) when sinking a current $2I_q$.

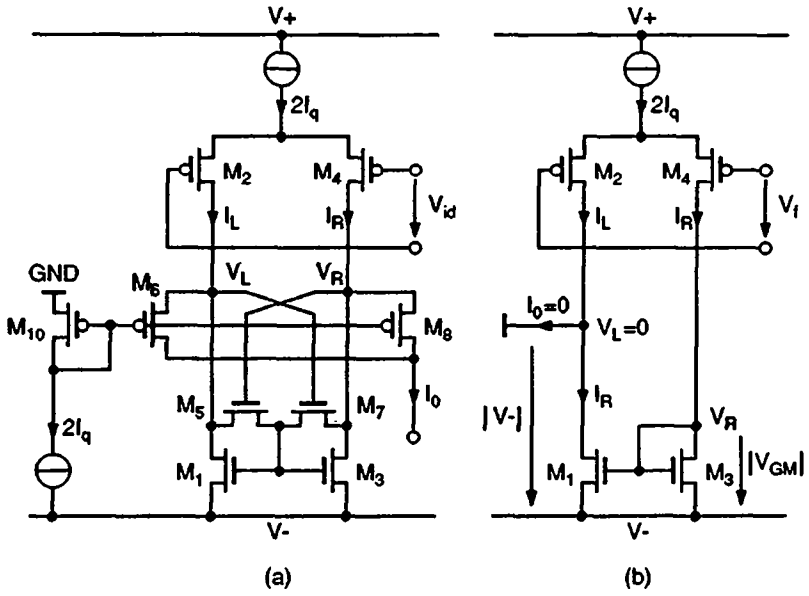


Fig. 7.1 (a) Circuit diagram of OTA-based full-wave rectifier, and (b) Equivalent circuit for calculating width of flat section of DC transfer characteristic.

Let V_{id} be large enough such that $I_L = 2I_q$ and $I_R = 0$. In this case, the node V_L is clamped at (virtual) ground by transistor M_6, which drives I_L to the output. This turns transistor switch M_7 on, thus making transistor M_3 the input device of the mirror, such that $V_{D3} = V_{G3} = V_R$. If I_R is then increased (by reducing V_{id}), it is subtracted from I_L through the mirror to make the output current I_0 equal to I_L - I_R. When I_R becomes larger than I_L, V_L drops and V_R rises thus inverting the mirror's input connexion. Since the circuit is symmetrical, similar behaviour occurs for the opposite input voltage polarity,

with I_0 equal to I_R - I_L. The transistors must be sized such that at zero crossing, the clamp transistors M_6-M_8 are both cut off.

7.3 INFLUENCE OF CHANNEL LENGTH MODULATION

The input differential voltage swing necessary to perform the whole inversion of the mirror corresponds to a small flat section at the origin of the DC transfer characteristic (see Figure 7.2). This flat section is due to the channel length modulation of both the source-coupled pair and the output mirror transistors. This means that the width of the flat section can be fixed through design parameters, i.e. the length of the transistors. When the mirror is about to invert its connexion, we have the ideal situation depicted in Figure 7.1(b), where V_f is the differential input voltage corresponding to the equilibrium at one extremity of the flat section. Consequently, the inversion of the mirror's connexion is completed for $\Delta V_{id} = 2V_f$ and $\Delta V \equiv \Delta V_L = \Delta V_R = |V\text{-}| - |V_{GM}(Iq)|$, which is the voltage swing of V_L and V_R, $|V_{GM}(Iq)|$ being the gate voltage of the mirror driving the current I_q. A first order analysis can be made considering the conductances g_0 at nodes V_L and V_R. These conductances can be calculated using any model describing the channel length modulation effect. This approach leads to a simple formulation based on the intrinsic voltage gain g_m/g_0 of the OTA-based structure:

$$V_f = \frac{g_0}{g_m}\Delta V = \frac{g_0}{g_m}\left(|V\text{-}| - |V_{GM}(I_q)|\right) \tag{7.1}$$

Considering now the asymptotic input saturation voltage of the cross-coupled pair defined by $V_{idsat} = 2I_q/g_m$, the relative flatness of the DC transfer characteristic can be defined as:

$$\frac{V_f}{V_{idsat}} = \frac{g_0\,\Delta V}{2I_q} \tag{7.2}$$

Because the drain voltages of the source-coupled pair are fixed by the output structure, V_f does not depend on the input common-mode voltage as long as it is within its functional range.

7.4 INFLUENCE OF DEVICE MISMATCH

The only devices of the output structure that have significant influence on the DC transfer characteristic are those of the current mirror. Any difference in their threshold voltage V_{T0} or their transconductance parameter β will lead, in first approximation, to a non-unity ratio r for a given mirror's connexion, and $1/r$ for the opposite one. The output current can be for example $I_0^+ = I_L - I_R/r$ for one polarity of the input differential voltage and $I_0^- = I_R - rI_L$ for the other polarity. After some calculations and normalization to V_{idsat}, this leads to the following relative input offset voltage:

$$\frac{V_{os}}{V_{idsat}} = \frac{1-r}{1+r} \tag{7.3}$$

This offset voltage contribution must be added to that of the source-coupled pair in terms of their RMS values.

As can be seen from the dotted curves of Figure 7.2, the asymptotic saturation voltage V_{idsat} is not modified by the offset of the output mirror, but the DC transfer characteristic shows two different slopes, which can be found to be respectively:

$$g_m^+ = \frac{r+1}{2r} g_m \tag{7.4}$$

$$g_m^- = \frac{r+1}{2} g_m \tag{7.5}$$

On the other hand, it is interesting to notice that the node voltages V_L and V_R can be used to restore exactly the original sign of the output current, even in presence of mismatches. This property will be used for the implementation of the synapse described in Chapter 8. Since the sign can be exactly recovered, the only term affecting the behaviour of the network in the particular implementation is the difference in the slopes of the transfer characteristic. This inaccuracy has been analysed in paragraph 4.5.2 and was found to be rather innocuous to the network's behaviour.

7.5 SIMULATION RESULTS

The structure of Figure 7.1(a) has been simulated with ESACAP [4], using the MOS transistor model developed by Enz [5]. The transistor

parameters are those of the SACMOS 3μm technology listed in Appendix D. The bias current is $2I_q$ = 1μA, and the voltage supplies are $V+$ = 2.5V and $V-$ = -2.5V. All the transistor sizes are W/L = 3μm/3μm, except those summarized in Table 7.1. The plain curves of Figure 7.2 represent a circuit with short transistors, showing a wide flat section at the origin of the DC transfer characteristic. The dotted curves show a smaller flat section due to the increased length of the concerned transistors, and the dashed curves show an offset resulting from a mismatch of about 25% in the current mirror.

Table 7.1 Transistor sizes in μm

curve	$(W/L)_{M2,4}$	$(W/L)_{M1}$	$(W/L)_{M3}$	remark
——	3.0/3.0	3.0/3.0	3.0/3.0	large output conductances
.....	**7.4/7.0**	3.0/**7.0**	3.0/**7.0**	output cond. reduced by 3
------	3.0/3.0	**3.5**/3.0	3.0/3.0	mismatch: $\beta_1 \approx 1.25\beta_3$

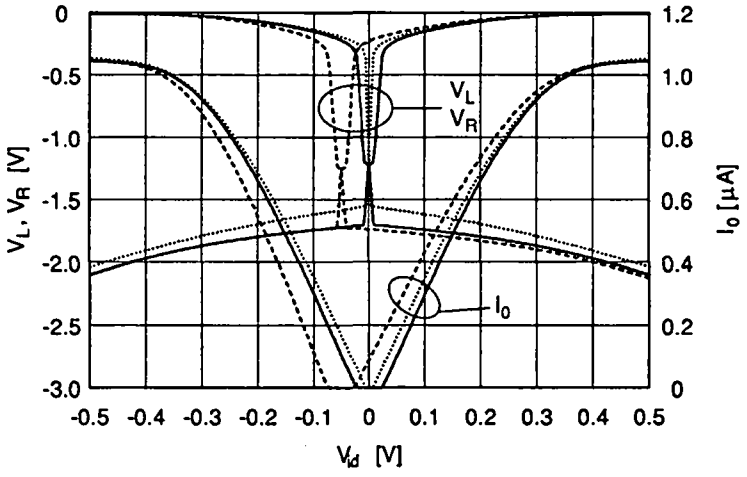


Fig. 7.2 DC transfer characteristics showing influences of channel length modulation and mirror mismatch (see text).

7.6 TRANSIENT BEHAVIOUR

As long as the zero-crossing point is not reached, the circuit behaves like a conventional single-ended OTA since the voltages at nodes V_L and V_R are kept nearly constant. However, if the mirror has to switch, these two nodes have to reach the large voltage swing $\Delta V = |V_{\dashv}| - |V_{GM}(Iq)|$ before the mirror can complete its operation. During the whole switching time, both transistors M_6 and M_8 are blocked and the output current is nulled, thus resulting in extra distortion of the output signal. A second limitation appears when one of the currents I_R or I_L is equal to zero at the beginning of a transient. In this case, the gate-to-source voltage of the mirror is zero, and a voltage step equal to V_{T0} has to be reached before the mirror is operational. Therefore, care must be taken to minimize the capacitance at nodes V_R and V_L if the circuit is intended to handle high-frequency signals.

If the circuit is used to compute Manhattan distance in analogue implementations of Kohonen maps, its poor transient behaviour is inherently compensated by the algorithm. As a matter of fact, when a new input vector is presented, the network must select the neuron with the smallest distance, which is close to zero and can thus lead to very large settling time. Therefore, if the fixed waiting time is shorter than the settling time, the apparent distance of the winner will be zero. On the other hand, all the other neurons have a distance larger than some I_ε, which depends on the structure of the data base, so that the settling time t_R of the rectifier for this application is practically limited to the following value:

$$t_R < 2\frac{C_{L,R}}{I_\varepsilon}\Delta V = 2\frac{C_{L,R}}{I_\varepsilon}\left(|V_{\dashv}| - |V_{GM}(I_q)|\right) \tag{7.6}$$

where $C_{L,R}$ is the capacitance at nodes V_L and V_R. Simulations have shown that the value of the capacitance at the gates of the mirror M_1-M_3 has an effect on the transient waveforms, but does not significantly affect the settling time t_R. It is necessary to know the settling time t_R in order to set the learning period of the network. The current I_ε must be evaluated taking into consideration the number of neurons, the dimension of the vectors and the structure of the data base to estimate the average distance between neurons. Finally, the second transient limitation is naturally avoided by restricting the

input differential voltage range to the linear fraction of the transfer characteristic, smaller than $2V_{idsat}$, which prevents both current I_R and I_L from reaching a value of zero.

7.7 CONCLUSION

The proposed CMOS full-wave rectifier is well-suited for the particular application of distance measurement in analogue Kohonen map implementations. Its major limitations are tolerated by the algorithm and its well controlled DC transfer characteristic allows low matching requirements and high circuit density.

7.8 REFERENCES

[1] P. Heim, F. Krummenacher and E. A. Vittoz: "CMOS Full-Wave Operational Transconductance Rectifier with Improved DC Transfer Characteristic", Electronis Letters, 30th January 1992, Vol. 28 No. 3.

[2] E. A. Vittoz: "MOS Transistors Operated in the Lateral Bipolar Mode and Their Application in CMOS Technology", JSSC, VOL. SC-18, NO. 3, June 1983, pp 273-279.

[3] E. Seevinck and R. J. Wiegerink: "Generalized Translinear Circuit Principle", IEEE JSSC, Vol. 26, No. 8, pp 1098-1102, August 1991.

[4] P. Stangerup: "ESACAP - A PC-Implemented General-Purpose Circuit Simulator", IEEE Circuits and Devices Magazine, vol. 4, July 1988, pp 20-25.

[5] C. Enz: "High-Precision CMOS Micropower Amplifiers", Ph.D. Thesis No 802, EPF-Lausanne, 1989, pp 5-61.

CHAPTER 8

PRECISE ANALOGUE SYNAPSE WITH LEARNING CAPABILITY

8.1 INTRODUCTION

Implementing a synapse for the Kohonen network is difficult because the learning algorithm is very sensitive to imperfections. For example, the effect of the charge injection or the leakage current inherent to sampling on capacitor has been pointed out in chapter 4. The charge injection can be compensated by the use of dummy switches [1], which requires a complementary control signal, and the leakage current can be minimized by maintaining the drain-to-bulk junction voltage of the access transistor near zero [2]. The precision of these techniques is in any case limited by transistor mismatches. Another problem arises from the sequential nature of the Kohonen algorithm, which requires the storage of the current weight value to compute the next one. If the present value cannot be stored, the weight update may be done by injecting a current proportional to $(x\text{-}m)$ during a fixed time. However, this technique can be very imprecise and asymmetrical, because the switches are opened during the weight update. Simulations have shown clearly the high sensitivity of the Kohonen algorithm to any non-symmetrical behaviour. Consequently, the proposed synapse has been developed especially to overcome the above mentioned drawbacks and is based on a symmetrical differential input-output modified switched-capacitor (SC) structure [3]. The weight update converges to an equilibrium state so that the switches are opened in well defined conditions. The technique is also based on the matching of transistors for the compensation of charge injection, but does not require complementary control signals and profits from all the advantages of fully differential circuits, such as good CMRR and PSRR.

8.2 DIFFERENTIAL SYNAPSE WITH LEARNING CAPABILITY

The rule (2.2) used to update the synaptic weights describes a discrete-time low-pass filter that could be implemented with a classical switched-capacitor (SC) technique if α were constant. To make α variable, a modified technique uses current equalization, instead of the usual charge redistribution, to modify the charge on a capacitor. To implement (2.2) with this technique, the actual weight value $m = m(t_k)$ must be maintained while computing the new weight value $m' = m(t_{k+1})$, which needs a second capacitor C'. Although the final structure is fully differential, the operation of the circuit can be better explained by first considering a single-ended version as depicted in Figures 8.1 and 8.2, which emphasize the two steady states of the synapse. The synapse is made of two transconductance amplifiers (OTA), a high gain amplifier A, a current invertor, two capacitors and four switches.

8.2.1 Phase 1: storage of the synaptic weight

The first steady state of the synapse corresponds to the weight storage (Phase 1), and is emphasized in Figure 8.1 by using plain lines for the active paths and dotted lines for the inactive paths. The weight m is held on capacitor C in the feedback loop of amplifier A. In the same time, the weight m is also copied to capacitor C', the resistor-connected OTA g_{m2} imposing a virtual ground for C'. During this phase, the output current $g_{m1}(x\text{-}m)$ is directed to the cell body to be used for the proximity computation and the selection of the winning cell.

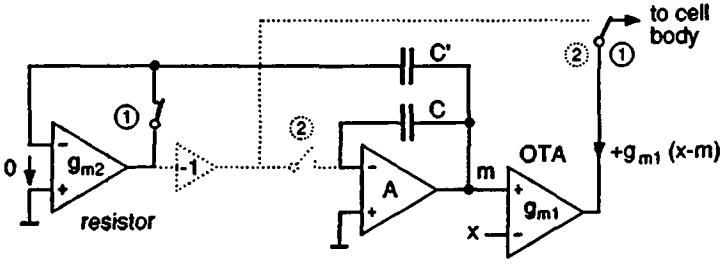


Fig. 8.1 Equivalent circuit of phase 1 (weight storage).

8.2.2 Phase 2: update of the synaptic weight

The updating phase starts with the opening of switches 1, holding the synaptic weight m on capacitor C'. Then switches 2 are closed, and the output current of OTA g_{m1} charges capacitor C. The resulting variation of m is applied at the input of g_{m2} by means of C', which generates a current $-g_{m2}(m'-m)$ at the output of g_{m2}. This current is then inverted and subtracted from the other charging current. When equalization occurs, the synapse reaches its second steady state as shown in Figure 8.2.

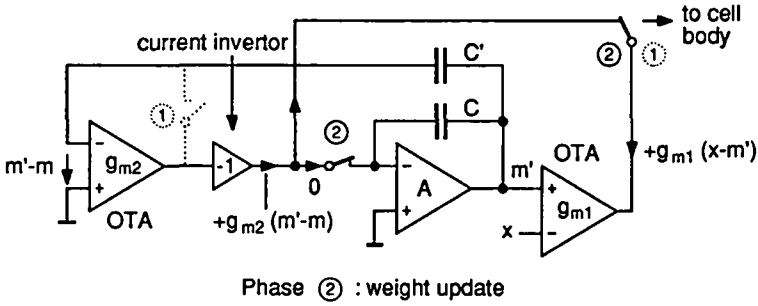


Fig. 8.2 Equivalent circuit of phase 2 (weight update).

At equilibrium, the two currents are equal, i.e.:

$$g_{m2}(m' - m) = g_{m1}(x - m') \tag{8.1}$$

Rearranging (8.1) results in the following updated weight:

$$m' = \frac{g_{m2}}{g_{m1} + g_{m2}} m + \frac{g_{m1}}{g_{m1} + g_{m2}} x \tag{8.2}$$

Identification with the update rule of the Kohonen algorithm

$$m' = m + \alpha (x - m) \tag{8.3}$$

leads to:

$$\alpha = \frac{g_{m1}}{g_{m1} + g_{m2}} \tag{8.4}$$

Which shows that α can be controlled by varying either g_{m1}, g_{m2} or both.

8.2.3 Differential version of the synapse

The purpose of the single-ended version of the synapse was to describe the principle of operation. The difficulty with this structure lies in the implementation of an offset-free bidirectional current invertor. Apart from the above-mentioned benefits, a fully differential input-output structure further simplifies the circuit because the current invertor can be avoided by simply crossing two branches, as shown in Figure 8.3.

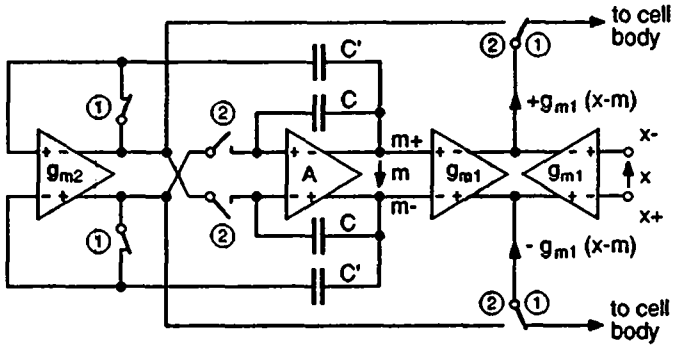


Fig. 8.3 Differential version of the synapse.

The major benefit of the differential technique, which is commonplace in SC implementations, is its inherent rejection of common-mode signals. This property will be exploited to compensate the charge injection of the switches, taking advantage of having a single control signal, as will be shown later.

On the other hand, this implementation requires one more OTA, two more capacitors and the amplifiers need output common-mode feedback (CMFB) circuitry. An eventual difference between the two g_{m1} values only affects the external synaptic weight m^* (see paragraph 3.6) but does not compromise the network behaviour.

The problem encountered with this synapse is the difficulty to implement variable transconductance OTAs while keeping a large and constant input voltage range. In addition, the input offset voltage may also vary with the transconductance value. As the variation range of the learning gain α must be quite large, a modified version of the synapse, using fixed transconductance OTAs only, has been developed.

8.3 SYNAPSE WITH FIXED TRANSCONDUCTANCES

In this version of the synapse, the variation of the gain is not implemented through the transconductance of the OTAs but by means of a one-quadrant translinear current multiplier [4] used as a variable gain current mirror. For this purpose, the output current of g_{m1} is rectified (Chapter 7 and [5]) and multiplied by a coefficient γ to produce the update current I_α. The sign is regenerated with a source-coupled pair which directs the current I_α to the correct node. The subsequent disequilibrium is compensated by the action of the CMFB of g_{m2}, which mimics complementary symmetrical injected currents of value $I_\alpha/2$.

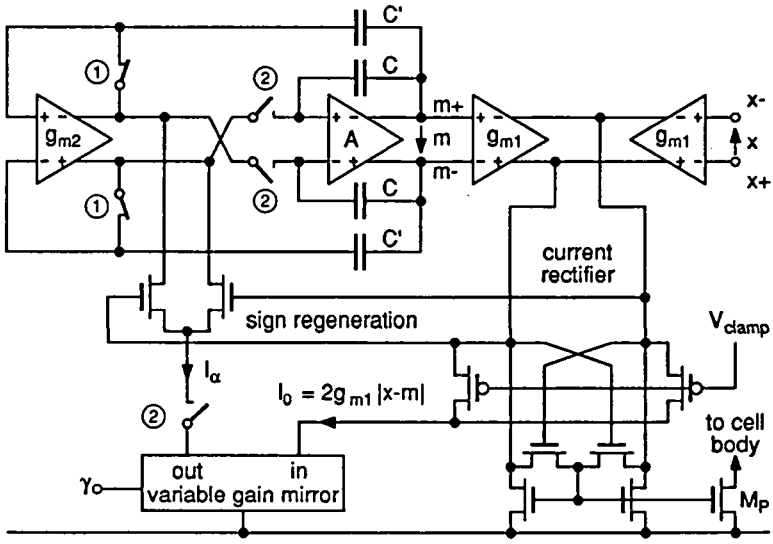


Fig. 8.4 Synapse version with fixed transconductances (see text).

At the output of the rectifier, the current I_0 takes the value

$$I_0 = 2\, g_{m1}\, |x\text{-}m| \tag{8.5}$$

where the coefficient 2 arises from the two OTAs g_{m1}. After multiplication by γ and division by 2 through the above-mentioned action of the CMFB, the (equivalent) injected currents at the outputs of g_{m2} are given by:

$$\frac{I_\alpha}{2} = \pm \gamma g_{m1} |x-m| \tag{8.6}$$

As in the previous version of the synapse, these injected currents will be compensated by the output currents of g_{m2}, driving the synapse to a new steady state. Equilibrium is reached when

$$g_{m2} (m - m') = \gamma g_{m1} (x - m') \tag{8.7}$$

which finally leads to the following value of α:

$$\alpha = \frac{\gamma}{g_{m2}/g_{m1} + \gamma} \tag{8.8}$$

Therefore α is an increasing and weakly nonlinear function of γ and crosses the origin. It may therefore not be necessary to take the nonlinearity into account when programming the temporal evolution of α. Figure 8.5 shows this function for several values of the parameter g_{m2}/g_{m1}.

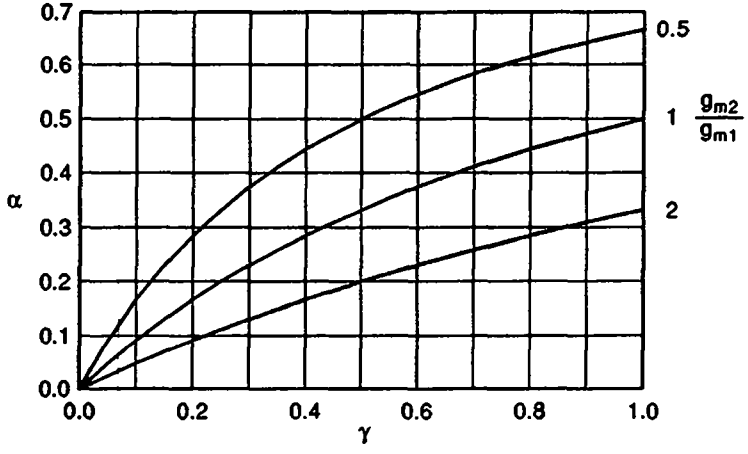


Fig. 8.5 Transfer function of the gain command.

The function of transistor M_P is to provide a proximity measure to the cell body, i.e. a current which is maximum when $x = m$. If for instance the polarization current of OTAs g_{m1} is $2I_q$, the maximum value of the proximity current will be $2I_q$ for $x = m$, provided M_P has the same size than the

mirror's transistors. Furthermore, this proximity measure is perfectly centred on the zero-crossing of the rectifier, i.e. it has its maximum value for $I_0 = 0$. This means that a position error on the selection of the winner is avoided, and the extra switches to the cell body as shown in Figures (8.1) to (8.3) are no longer needed since the proximity measure is permanently available.

8.4 SECOND ORDER EFFECTS

8.4.1 Influence of parasitic capacitors

Except for the voltage to current conversion due to g_{m2}, the synapse acts like a switched-capacitor integrator. There are two criteria that determine whether such a circuit is sensitive to parasitic capacitors [6] :

- Besides the reference node (generally the ground), the circuit can have only two types of nodes, i.e. outputs of amplifiers and virtual grounds.
- The terminals of the transfer capacitors are, at the exclusion of any other possibility, switched either between two amplifier's outputs, between the ground and a virtual ground, or between two virtual grounds.

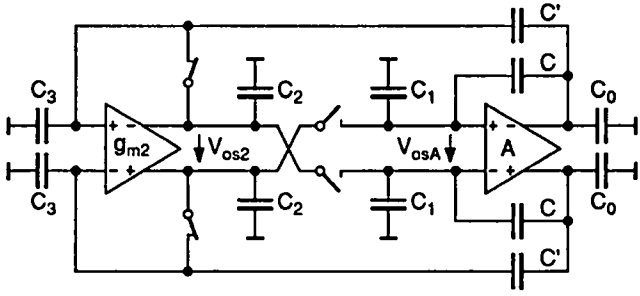


Fig. 8.6 Synapse with its parasitic capacitors and offset voltages.

The above figure shows the synapse with its parasitic capacitors. The circuit is considered to be perfectly symmetrical in first approximation. The transfer capacitors C' are not in accordance with the second above-mentioned rule. During the weight's updating phase, the capacitors C_3 are connected to the floating inputs of g_{m2}, which are high impedance nodes. The result is that the variation m'-m of the synaptic weight, which should be applied at the input of

g_{m2}, is in fact reduced by the capacitive divider C'-C_3. Consequently, relation (8.7) must be modified as follows:

$$g_{m2} \frac{C'}{C' + C_3} (m - m') = \gamma\, g_{m1} (x - m') \tag{8.9}$$

and leads to the following learning gain:

$$\alpha = \frac{\gamma}{\frac{C'}{C' + C_3} \frac{g_{m2}}{g_{m1}} + \gamma} \tag{8.10}$$

This results equivalently in sizing the parameter g_{m2} by the factor $C'/(C'+C_3)$ in relations (8.8).

An eventual asymmetry between a couple of parasitic capacitances, which cannot be totally avoided on the layout, may lead to other effects that will be further developed.

8.4.2 Influence of the amplifiers' input offset voltages

When the switches of phase 2 are closed, the outputs of g_{m2} are connected to the inputs of A. The resulting common-mode voltage is fixed by the CMFB of g_{m2} (the common mode voltage at the inputs of A is restored to ground by the leakage currents during the storage phase, as will be explained in the next paragraph). In the presence of non-zero and independent offset voltages, the feedback loop forces the differential output voltage of g_{m2} to be equal to V_{osA}. This must be accomplished through a modification Δm_o of the synaptic weight, which is weighted by the capacitive divider C'-C_3 at the input of g_{m2} and amplified by the open loop gain $A_2 = g_{m2}/g_{o2}$ to give the sum of the two offset voltages. The weight modification can thus be expressed as follows:

$$\Delta m_o = \frac{V_{os1} + V_{os2}}{A_2} \frac{C' + C_3}{C'} \tag{8.11}$$

The term containing the offset voltages can be calculated statistically by summing their variances. Because it is easy to make OTAs with high open loop gains (60 dB or more), the error Δm_o on the synaptic weight is expected to be kept in the tens of microvolts, and is therefore negligible with respect to other inaccuracies.

8.4.3 Leakage currents

The leakage current reduction technique used for the sample-and-hold presented in Chapter 3 [2] can be used for the synapse. The differential structure, including the switches and the input offset voltage of the amplifier is shown in Figure 8.7 (a).

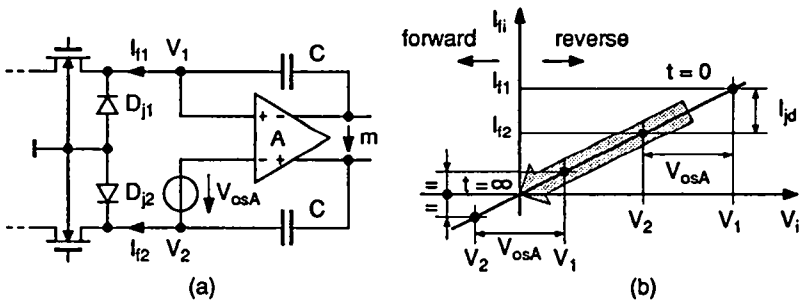


Fig. 8.7 (a) Low-leakage sample-and-hold. (b) Input common-mode voltage recovery.

The input offset voltage of amplifier A appears across the back-to-back junction diodes D_{j1} and D_{j2}, leading to two different leakage currents I_{f1} and I_{f2}. When the weight is updated, the input common-mode voltage of amplifier A is fixed by the CMFB of OTA g_{m2}, and can be somewhat different from the reference ground. Therefore, the two junction diodes may be either both forward-biased, both reverse-biased or one reverse-biased and the other forward-biased. Figure 8.7 (b) shows a simplified case where the diode characteristics are equal and linear near the origin, having an equivalent conductance g_j. If, for example, the common-mode voltage fixed during the updating phase is greater than ground, there is a common-mode leakage current $(I_{f1}+I_{f2})/2$ that discharges the two capacitors with the large time constant g_j/C, until the two leakage currents reach the same value with opposite polarity. It cannot be otherwise because the CMFB of A has to supply this common-mode leakage current until it vanishes. Therefore, the only leakage that affects the synaptic weight is the differential component I_{jd}.

Figure 8.8 shows more realistic cases in which the junction diodes have different leakage values. It can be seen that in any case, when the common-mode leakage current has been cancelled, the largest fraction of the offset voltage V_{osA} is used to bias the diode which has the lowest leakage value. Since

the spread of diode leakage currents can be very large due to local defects, this property is very interesting because it can cancel the large leakage of a bad diode. Therefore, as the probability of having two bad diodes should be very low, the probability distribution of the leakages should depend mostly on that of the input offset voltage of the amplifier.

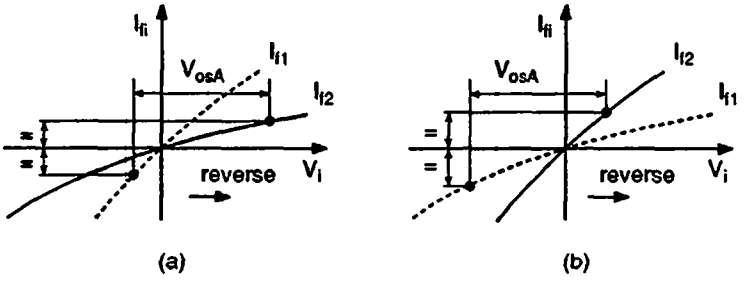


Fig. 8.8 Differential leakage at $t = \infty$ (a) $I_{f1} > I_{f2}$. (b) $I_{f2} > I_{f1}$.

8.4.4 Charge injection compensation

When the switching transistors are turned off, the minority carriers that are present in the channel during conduction are released and alter the voltages at both ends of the transistors. The amount of charge injected into the hold capacitor has been extensively analysed for the case of a feedforward sample-and-hold [1], and some compensation techniques have been proposed in [7]. A classical technique uses dummy switches to compensate the excessive charge on the holding capacitor, but requires an adequate design of the switching parameters, complementary clock phases and a carefully optimized layout. These requirements need, however, excessive circuit area and are therefore not adapted for neural network implementations.

The charge injection can be compensated using the inherent rejection of common-mode signals of differential circuits. The resulting error will then depend on the mismatch of the elements, i.e. those of the switches and the capacitors. The structure of the synapse is well adapted to this technique because the two switches of each pair are operating under the same conditions when they are turned off. The parameters needed to estimate the differential contribution of the injected charges are the following [1], [7]:

- the total charge in the channel

$$Q_{tot} = C_G (V_{GON} - V_{TE}) \tag{8.12}$$

where C_G is the total gate capacitance, including the overlap capacitances, V_{GON} is the on value of the gate voltage, and V_{TE} is the effective gate threshold voltage which depends linearly on the (equal) access voltages $V_{D,S}$, according to:

$$V_{TE} = V_{T0} + n V_{D,S} \tag{8.13}$$

- the normalized switching parameter B, which is given by:

$$B = (V_{GON} - V_{TE}) \sqrt{\frac{\beta}{a C_H}} \tag{8.14}$$

where a is the slope of the switching off gate voltage in V/s, and C_H the hold capacitance.

- the ratio of the hold to source capacitances C_H / C_S

References [1] and [7] give a normalized diagram showing the amount of normalized charge Q_H / Q_{tot} injected in C_H as a function the switching parameter B and the capacitance ratio C_H / C_S. The diagram shows that for sufficiently fast gate voltage swing (in fact for $B < 0.1$), the channel charge is shared equally between drain and source, independently of the ratio C_H / C_S. For larger values of B, the channel charge tends to go into the largest capacitor. Another reference [8] analyses the case of the simple current copier, taking into account the feedback loop introduced by the copying transistor when the switch in series with the storage capacitor opens. Although the result cannot be totally normalized, it is shown that for $B < 0.1$ the channel charge is still shared equally, which means that the feedback effect does not operate. A difference is noticeable for larger values of B, for which the channel charge is increasingly returned to the source capacitor.

The synapse has two pairs of switches that are included in feedback loops. The switches of phase 1 that short the OTA g_{m2} are in a simple feedback loop similar to that of a current copier, and could thus be analysed using the curves presented in [8]. On the other hand, the sampling switches of phase 2 are in a double feedback loop including g_{m1} and g_{m2}, which makes the analysis particularly cumbersome, if not impossible. In addition, as will be seen from the measurements of the test chip, a slow transition of the gate voltage leaves

enough time to the feedback to operate and gives rise to a troublesome systematic effect which is certainly due to the asymmetry of the parasitic capacitors. Therefore, it is recommended to use the fastest possible gate voltage transition. This is practically easier to implement, and the analysis can be based on the knowledge of the total gate capacitance C_G and the mismatch of the switching transistors only.

In differential mode, the resulting voltage alteration due to charge injection is given by the difference between the two sides of the circuit. If all the mismatch terms affecting the total charge in the channel are included into a single parameter ε_q, the differential voltage variation is given by:

$$\Delta V_{diff} = \varepsilon_q \frac{Q_H}{C_H} = \varepsilon_q \frac{Q_{tot}}{2C_H} \qquad \text{if } B < 0.1 \tag{8.15}$$

The contribution of phase two directly affects the synaptic weight at the end of the updating phase, while the contribution of phase 1 affects the temporary value used at the beginning of the updating phase. The latter contribution to the updated value is weighted by the ratio g_{m2}/g_{m1}. The total alteration on the synaptic weight Δm_q resulting from the contribution of the two phases can thus be expressed as:

$$\Delta m_q = \varepsilon_{q1} \frac{g_{m2}}{g_{m1}} \frac{Q_{tot1}}{2C'} + \varepsilon_{q2} \frac{Q_{tot2}}{2C} \tag{8.16}$$

Since the two mismatches are statistically independent, they can be summed in terms of their RMS values, and the standard deviation $\sigma(\Delta m_q)$ is given by:

$$\sigma(\Delta m_q) = \sqrt{\left(\sigma(\varepsilon_{q1}) \frac{g_{m2}}{g_{m1}} \frac{Q_{tot1}}{2C'}\right)^2 + \left(\sigma(\varepsilon_{q2}) \frac{Q_{tot2}}{2C}\right)^2} \tag{8.17}$$

The relative contribution of the mismatch of the capacitors to the error Δm_q is equal to the value of the mismatch itself, and can be neglected.

8.5 IMPLEMENTATION OF A TEST CHIP

8.5.1 General considerations

The synapse seems rather complicated considering the functions it implements. Consequently, it is of major importance to chose the best suited

circuits to implement the different functions. The most important criterion is circuit area, and the goal is to make a working synapse with the maximum precision in a limited area. Therefore, the following assumptions are made a priori:

- noise considerations are avoided by using the largest possible voltage range for the synaptic weights, which is also recommended to minimize the relative effect of charge injection and junction leakages
- stability considerations are avoided by using a folded cascode structure to implement the amplifiers
- except for the switches, minimum-sized transistors are never used because this would lead to the poorest accuracy and the smallest ratio of the total active area to the total circuit area
- the bias currents are not chosen a priori, but will depend on the transistor sizes of the OTAs and the requested input voltage range.

After some trials at both the layout and design levels, it was decided to make a cell of 250μm × 250μm, for a density of 16 synapses per mm^2. Once the size of the cell is fixed, most of the work consists of optimizing the layout so that the critical specifications are respected.

8.5.2 Circuit design

The test chip is implemented using the SACMOS 3μm technology, the parameters of which are listed in Appendix D. This is a self-aligned technology that has no overlap around the contacts, and the circuit density is therefore comparable to that of 2μm standard technologies. The circuit uses ±2.5V supply voltages. The ground (GND) is the reference voltage for the CMFBs and fixes the switches' steady-state potentials and the virtual ground at the inputs of amplifier *A*. The whole schematic of the synapse is split between Figures 8.9 to 8.11.

The schematic including the input OTAs g_{m1}, the current rectifier, the translinear multiplier and the differential pair that is used to regenerate the sign of the update current is depicted in Figure 8.9, and the transistor parameters are listed in Table 8.1. The choice of the transistor types is imposed by the type of the technology. Since SACMOS 3μm is a p-well technology, only NPN compatible lateral bipolar transistors (CLBT) are

available, which imposes the transistor types of the other blocks to avoid unnecessary current inversions.

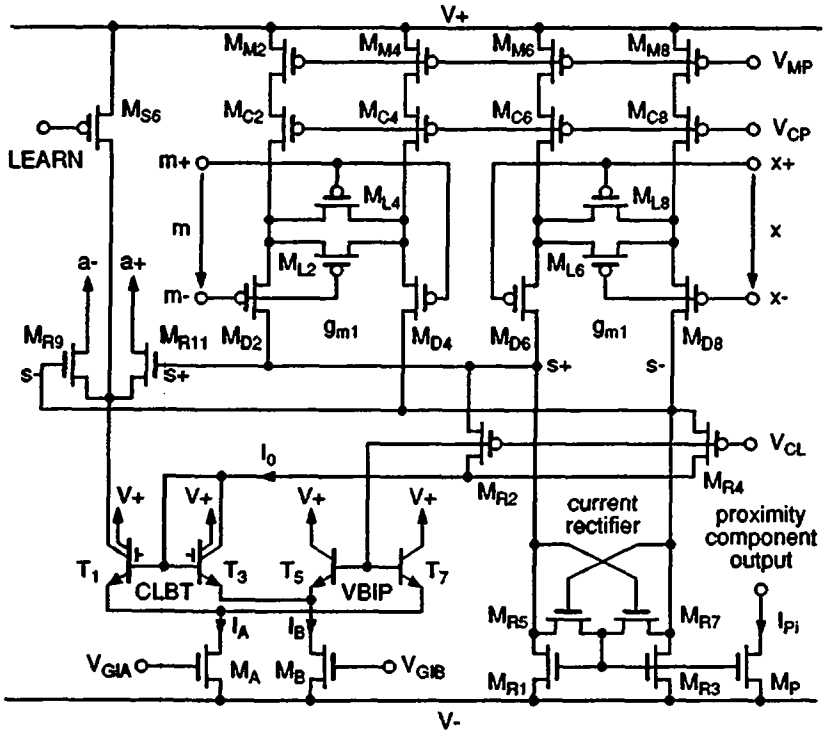


Fig. 8.9 Schematic of the input OTAs g_{m1}, the current rectifier, the one-quadrant translinear multiplier and the sign regeneration differential pair.

The OTAs g_{m1} are implemented with linearized differential pairs [9]. For the same bias current $2I_q$, and therefore the same gate overdrive voltage $n(V_P\text{-}V_S)$ of transistors M_D, the input saturation voltage V_{idsat} is multiplied by 2.5, provided $\beta_D/\beta_L = 6$. In addition, the linearity of the transfer characteristic is improved and a larger input voltage range can be exploited. To ensure a differential input voltage range $V_{range} = 2$ V, a nominal saturation voltage $V_{idsat1} = 1.25$ V has been chosen. With the transistor sizes used, the nominal bias current I_q is found to be 2.0 µA.

The offset-insensitive current rectifier is described in Chapter 7. It is interesting to notice that its particular use as a load for OTAs g_{m1} avoids the need for a CMFB circuit.

The one-quadrant translinear multiplier is made of two octagonal CLBTs and two vertical bipolar transistors (VBIP). It is used as a variable-gain current mirror controlled by means of currents I_A and I_B. For this purpose, I_B is a fixed bias current and I_A controls the gain as:

$$\gamma = \frac{I_{out}}{I_{in}} = \frac{I_A}{I_B} \tag{8.18}$$

The current I_B must be larger than the maximum input current $I_{0,\,max} = 4I_q$, divided by the common-base current gain α_{CLBT} of the transistors. For this purpose, I_B is generated using a CLBT in a feedback loop that forces its collector current to be equal to $5I_q$. With the above-mentioned value $I_q = 2\mu A$, the CLBTs are still operating below their high injection level.

Table 8.1

Device	Type	W/L [μm]	β [μA/V^2]	Function
M_{M2}-M_{M8}	p	13/13	19.5	mirror
M_{C2}-M_{C8}	p	13/5	54.0	cascode
M_{L2}-M_{L8}	p	5/2×13	3.2	diff. pair lin. transistor
M_{D2}-M_{D8}	p	3×5/13	19.2	differential pair
M_{R2}, M_{R4}	p	19/3 (U-shaped)	146.0	"clamp" transistors
M_{R1}, M_{R3}	n	11/11	52.0	rectifier mirror
M_{R5}, M_{R7}	n	3/3	55.0	switches
M_P	n	11/11	52.0	proximity output
M_A	n	10/10	52.1	gain control
M_B	n	2×10/5	234.6	multiplier polarization
M_{R9}, M_{R11}	n	6/3	132.6	sign regeneration
M_{S6}	p	3/3	16.7	phase 2 shunt switch
T_1, T_3	CLBT	S_E = 4×4	octagonal	one-quadrant multiplier
T_5, T_7	VBIP	S_E = 3×3	min. size	one-quadrant multiplier

To save an extra wire, the bases of transistors T_5 and T_7 are biased with the gate voltage of the clamp transistors M_{R2} and M_{R4} of the rectifier. This voltage must be sufficiently above V- so that M_B is kept saturated. The output current of the variable mirror is then directed to the correct node a+ or a- with the differential pair made of M_{R9} and M_{R11}. The differential pair is driven with the node voltages s+ and s- that give exact information on the zero-crossing point of the rectifier. Transistor M_{S6} is a switch that diverts the output current to V+ during the storage phase (LEARN = 0).

Finally, transistor M_P provides the proximity measure I_{Pi} for the weight component i. As already stated, the maximum value I_{Pi} is $2I_q$ and is reached exactly at the zero-crossing point of the rectifier, i.e. when $x = m$.

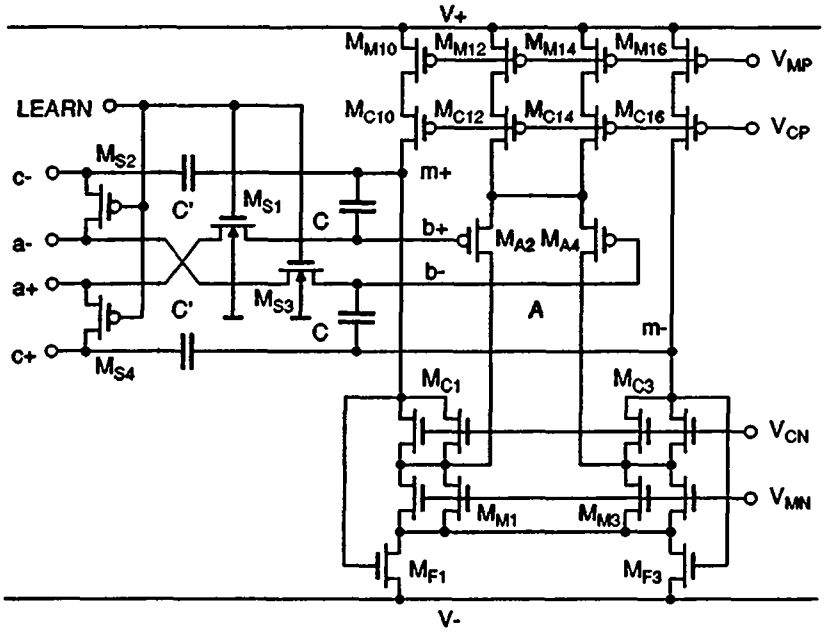


Fig. 8.10 Schematic of amplifier A and switches of phases 1 and 2.

The schematic of amplifier A together with that of the switches are shown in Figure 8.10, and the transistor parameters are listed in Table 8.2. The amplifier is based on a folded cascode structure, and is biased with the same current value $2I_q$ used for g_{m1}. The transistors of the differential pair

M_{A2} and M_{A4} are octagonal, which minimizes the gate-to-drain capacitance and gives a large aspect ratio W/L so that the intrinsic gain $A = g_{mA}/g_0$ is maximized. The common-mode feedback is implemented with transistors M_{F1} and M_{F3}, operating in the linear mode. This structure has been chosen because a fast action is not required, and it occupies the smallest possible silicon area.

The switches of phase 2 are n-channel transistors in a separate well connected to GND, as already shown in Figure 8.7 (a). The switches of phase 1 are p-channel transistors so that the same command signal LEARN can be used. This structure strictly limits the minimum supply voltage to the following value, which must be respected for both transistor types in the worst case:

$$V_+ - V_- \geq \frac{V_{T0}}{1 - n/2} \tag{8.19}$$

This relation assumes that the positive and negative supply voltages are symmetrical with respect to ground, but another choice can be made depending on the transistor parameters. In fact, some margin must be provided to the supply voltage to ensure a minimum conductance to the switches when both their source and drain terminals are near ground. In this situation, the two phases are non-overlapping, which is the case at both transitions of the command signal, provided the respective steady states have been previously attained.

Table 8.2

Device	Type	W/L [μm]	β [μA/V^2]	Function
M_{M10}-M_{M16}	p	13/13	19.5	mirror
M_{C10}-M_{C16}	p	13/5	54.0	cascode
M_{A2}, M_{A4}	p	32/5 (octagonal)	131.7	differential pair
M_{C1}, M_{C3}	n	2×11/5	255.0	cascode
M_{M1}, M_{M3}	n	2×11/11	102.0	mirror
M_{F1}, M_{F3}	n	5/24	9.1	CMFB resistors
M_{S1}, M_{S3}	n	3/3	52.0	phase 2 switches
M_{S2}, M_{S4}	p	3/3	16.7	phase 1 switches

At the beginning of the update phase, since the CMFB of g_{m2} does not react instantaneously to compensate the asymmetrical injected current at node a+ or a-, this current can be temporarily diverted through the source-to-well

junction diode of M_{S1} or M_{S3}. Apart from delaying the update process, this situation will not perturb the final weight value, since it is based on current equalization only.

The schematic of OTA g_{m2} with its CMFB circuit is shown in Figure 8.11, and the transistor parameters are given in Table 8.3. The OTA is also based on a folded cascode structure to avoid stability problems, and its differential pair, as well as its bias current, are the same as those of g_{m1}, so that $g_{m2} = g_{m1}$ in this implementation of the synapse.

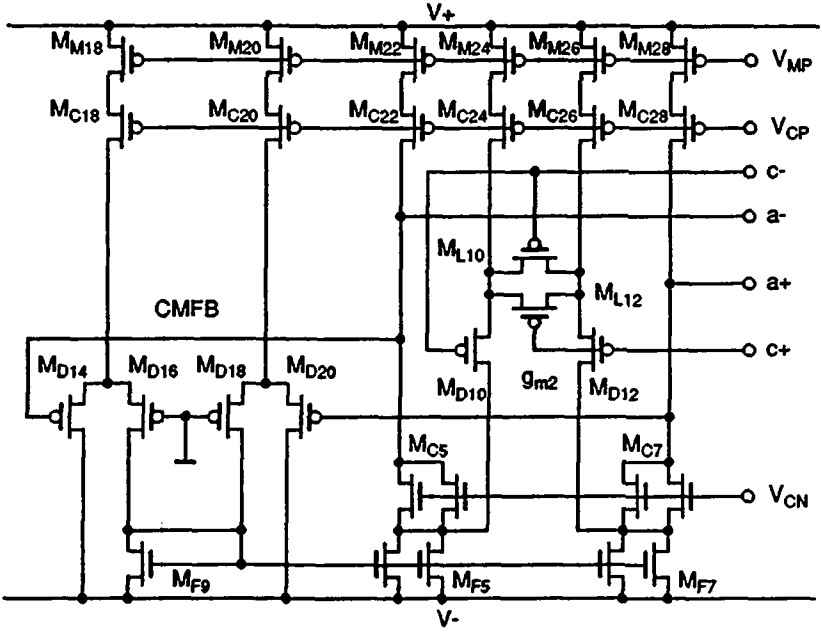


Fig. 8.11 Schematic of OTA g_{m2} with its CMFB circuitry.

Due to the asymmetrical injection of the update current I_α into a single output of g_{m2}, the action of the CMFB circuit must be fast, and the simple structure used for amplifier A is not suitable. The proposed circuit uses two differential pairs M_{D14}-M_{D16} and M_{D18}-M_{D20}, biased with I_q, to compare the output voltages with the ground reference. The differential pairs are loaded with the same diode-connected transistor M_{F9}, which directly controls the current loads

M_{F5} and M_{F7} of g_{m2} so that the total load current is equal to the total bias current $4I_q$. When I_α is injected into an output, the CMFB circuit decreases the load current to maintain equilibrium, and the output common-mode voltage is also decreased. This statism could be avoided by replacing M_{F9} by a current mirror to load the two differential pairs, and by controlling the gates of M_{F5} and M_{F7} with the resulting high impedance output node. This increases the open loop gain of the CMFB stage, but it also introduces an extra pole that causes instability. This solution was therefore rejected due to the excessive silicon area needed for compensation of the circuit.

Table 8.3

Device	Type	W/L [μm]	β [μA/V^2]	Function
M_{M18}-M_{M28}	p	13/13	19.5	mirror
M_{C18}-M_{C28}	p	13/5	54.0	cascode
M_{L10}, M_{L12}	p	5/2×13	3.2	diff. pair lin. transistor
M_{D10}, M_{D12}	p	3×5/13	19.2	differential pair
M_{C5}, M_{C7}	n	2×11/5	262.5	cascode
M_{F5}, M_{F7}	n	2×11/11	105.0	mirror
M_{D14}-M_{D20}	p	7/16	7.6	CMFB differential pair

The common-mode voltage drop ΔV_{CM} resulting from the injection of I_α must be limited to prevent the conduction of the source-to-bulk junction diodes of M_{S1} and M_{S3}. The value of I_α for $x = -m = V_{range}$ is given by:

$$I_\alpha \leq \gamma 4 I_q \frac{V_{range}}{2V_{idsat1}} \tag{8.19}$$

where V_{range} is the full differential voltage range of the synapse, and $4I_q$ the total bias current of the two OTAs g_{m1}. The injection of I_α is compensated by a reduction in ΔV_{CM} through the transconductance g_{mF} of the CMFB circuit. Since the mirror M_{F9}-(M_{F5}&M_{F7}) has a gain of 4, this can be expressed as:

$$I_\alpha = -4\, g_{mF}\, \Delta V_{CM} = -\frac{4 I_q}{V_{idsatF}} \Delta V_{CM} \tag{8.20}$$

which finally leads to:

$$|\Delta V_{CM}| < \frac{\gamma}{2} \frac{V_{idsatF}}{V_{idsat1}} V_{range} \tag{8.21}$$

For this design, the ratio V_{idsatF}/V_{idsat1} is 0.45, which may seem quite large. However, as will be seen in the next paragraph, γ is practically limited to 0.45.

Finally, the polarizations of the cascoded mirrors are based on the low-voltage technique proposed in Appendix C. For the n-channel mirrors, a transistor M_{Fb} similar to M_{F1} and M_{F3} has been inserted in the source of the input transistor of the mirror. The gate of M_{Fb} is tied to GND, to fix the common-mode voltage reference for amplifier A. The resulting cascode gate voltage V_{CN} is suitable for transistors M_{C5} and M_{C7}, since it is larger than the requested value. The gate voltage of the clamp transistors M_{R2} and M_{R4} of the rectifier is generated within the cell (not shown on the schematic, see Chapter 7), to save a wire that would occupy an area of 150 μm^2.

8.5.3 Circuit limitations

Most limitations of the circuit arise from the limited silicon area, which imposed some restrictions on the layout. In particular, g_{m1} and g_{m2} have the same value, which limits the maximum possible learning gain for the following reason. The update current I_α must be smaller than I_q to prevent the concerned output of g_{m2} from reaching saturation (in fact, the output will be clamped by the source-to-well junction diode of the switch). Using relation (8.19) with the above mentioned typical values, this limits γ to:

$$\gamma < \frac{V_{idsat1}}{2V_{range}} = \frac{1.25}{4} \approx 0.31 \tag{8.22}$$

From relation (8.8), the maximum corresponding value of the learning gain α is 0.24. However, since the principle of the synapse is based on current equalization, its operation will not be perturbed if the final value of I_α, which corresponds to the initial value weighted by (1-α), is smaller than I_q. Taking this into account and using (8.8) and (8.19) leads to the following simple limit for the learning gain:

$$\alpha < \frac{V_{idsat1}}{2V_{range}} \tag{8.23}$$

which surprisingly is identical to relation (8.22). The learning gain is therefore limited to about 0.31 (which corresponds to $\gamma = 0.45$), but a larger value can be imposed in the context of the Kohonen algorithm since $(x\text{-}m)$ will practically never reach V_{range} within the learning neighbourhood.

8.5.4 Layout of the cell

Layout considerations had a large impact on the design, and as such are an important aspect of this work. The layout of a complete cell is shown in Figure 8.12. It may not seem, at first sight, that the cell contains no less than 79 MOS transistors, 2 CLBTs, 2 VBIPs and 4 capacitors of about 0.8 pF.

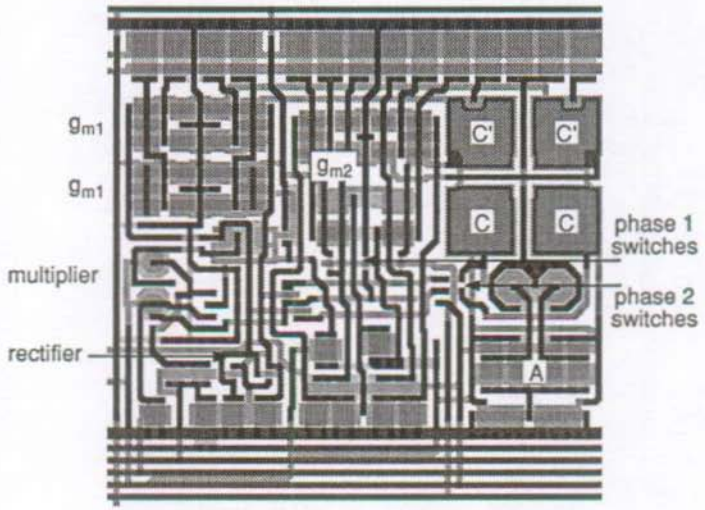


Fig. 8.12 Layout of the synapse ($250 \times 250\ \mu m^2$).

The disposition of the various blocks is globally influenced by the schematics, in order to optimize the routing paths, which does not lead to a symmetrical layout. Some trials to make the layout more symmetrical led to excessive silicon area. As a consequence, the parasitic capacitors due to the interconnexions are not balanced, and this may lead to systematic effects that must be identical for all the chips. A very compact layout has been found for the linearized differential pairs used for the OTAs. The ratio $\beta_D/\beta_L = 6$ is implemented with three unit transistors in parallel for M_D and two unit transistors in series for M_L. This structure is also interesting because metal

wires can cross over the gates of the transistors. The translinear one-quadrant multiplier is also very dense thanks to the use of two vertical bipolar transistors. Its area is 3400 μm^2, which represents 5.5% of the cell area.

A photograph of the test chip is shown in Figure 8.13. The chip contains two synapses, a circuit for the generation of the learning gain (not described in this chapter), the various polarizations, and a separate one-quadrant translinear multiplier. On the top of the chip are four source followers used to output the synaptic weights (m_1+, m_1-) and (m_2+, m_2-) of the two cells. Ten chips were available for the measurements, and all of them were operating properly.

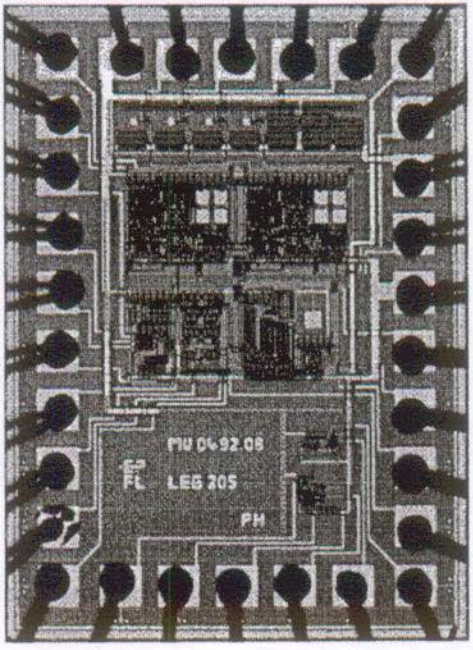


Fig. 8.13 Photograph of the chip (1.2 × 1.6 mm^2).

8.6 TEST CHIP MEASUREMENTS

The synapse has been measured several ways, and this helped sometimes to understand some of its unexpected behaviours. Although the results globally correspond to what was expected, there are some phenomena that have not been explained at all.

8.6.1 Operating the synapse as a time constant

To first check the overall operation of the synapse, it is used as a variable time constant. For this purpose, a square wave learning clock signal with a frequency of 5 KHz was used (t_{LEARN} = 100 µs). The input signal is also a square wave, with an amplitude of ±1 V with respect to GND, which corresponds to the full voltage range V_{range}. The period of the input signal is 64 times larger than that of the clock. Three measurements with very different values of the learning gain are reported in Figure 8.14, to show the wide range of operation.

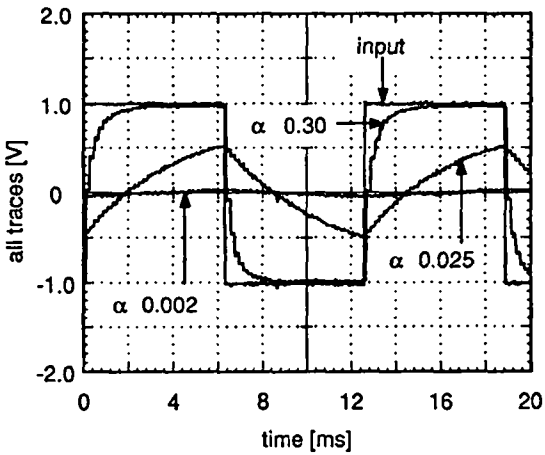


Fig. 8.14 Measurement of the synapse as a variable time constant.

The first measurement used the largest possible gain for which the synapse was operating properly. Looking at the first update step following an input signal transition (about 1 V), the learning gain value would be around 0.5. However, looking at the next update steps, it falls to about 0.3, which corresponds to the

calculated value. Therefore, it is useless to analyse what happens during the first update step, since it can be practically avoided in a real situation. The third measurement used the smallest gain value for which the weight was still maintained within the midrange of the input signal, and has a value of about 0.002. Finally, the second measurement uses a value in between just to show the continuous operation of the synapse.

An unforeseen problem was found during the measurements. For small amplitudes of the input signal the synaptic weight did not reach the expected value, but saturated at a smaller amplitude. Further investigation showed a dependence on both the learning time t_{LEARN} and the update value $\alpha(x\text{-}m)$. The problem is due to the parasitic capacitor C_X at the collector of T_1. During the storage phase, this node is held to the positive supply by transistor M_{S6} (see Figure 8.9). At the beginning of the update phase, C_X is discharged by the update current I_α until transistor M_{R9} or M_{R11} diverts it to an output of g_{m2}. For low update currents (under 245 nA these transistors operate in weak inversion), this happens approximately when the source of M_{R9} or M_{R11} reaches its pinch-off voltage. Therefore, for a given value of the update current, the requested learning time is given by:

$$t_{LEARN} > \frac{C_X \Delta V_X}{I_\alpha} \tag{8.24}$$

where ΔV_X is the voltage swing at the collector of T_1. Using relation (8.6) for the update current and assuming that $\alpha \approx \gamma$ for the low values, the minimum weight's update value can be expressed as follows:

$$\alpha |x - m| > \frac{C_X \Delta V_X}{2\, g_{m1}\, t_{LEARN}} \tag{8.25}$$

For the above-mentioned typical values (and $C_X \approx 134$ fF), weight update values of 0.1% of full scale need a learning time of 100 µs, which seems somewhat excessive. However, the learning time can be shortened with a proper gain command generated off-chip. Indeed, during the update phase, the gain can be modified without affecting the operation of the synapse if it is returned to the correct value before the phase completion. Therefore, a short pulse of increased gain value can be added at the beginning of the update phase to shorten the charging time of C_X. The amplitude of this pulse can be easily calculated using (8.25), or experimented on the chip.

8.6.2 Charge injection measurement

The measurement of charge injection is made under continuous learning with the input $x = 0$ (or any constant value). Under these conditions, the weight m will converge to a value for which the update value $\alpha(x\text{-}m)$ is just sufficient to compensate for the injected charge. Figure 8.15 shows a typical result for a synapse with an injected charge corresponding to about 0.4 mV on the synaptic weight. The figure shows the clock signal and the differential outputs $m+$ and $m-$. The output offset has been cancelled with the scope DC adjustment.

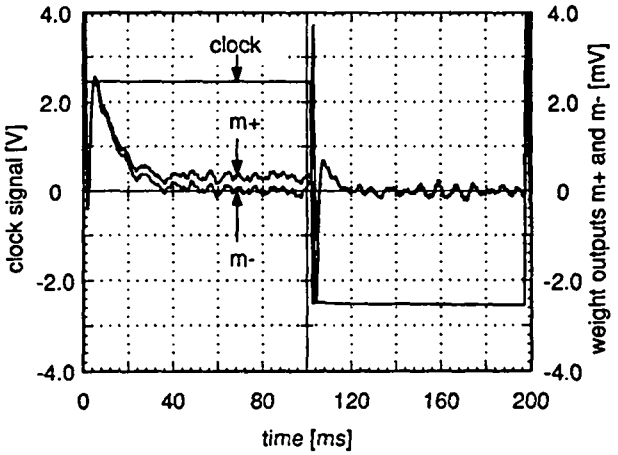


Fig. 8.15 Measurement of charge injection with a clock rise and fall time of 10 ns.

As previously stated, the asymmetry of the layout may lead to systematic effects. Since the sampling switches are inserted into feedback loops, this asymmetry is most likely to affect the injected charges. To illustrate this, measurements have been made on nine synapses of the same run for different rise and fall time. The results are reported in Table 8.4 and Figure 8.16. On the figure, absolute values are used due to the logarithmic scale. For large rise and fall times, i.e. above 10 ns, a systematic effect dominates with a relatively small standard deviation. This is a good indication that the effect is due to layout asymmetry. What is more surprising is the linear dependence of the weight error on the rise and fall time. Although it has not been explained, it

might be the integration of a current resulting from some DC transfer characteristic that depends on the gate voltage of the switches.

Table 8.4

$t_{r,f}$ [ns]	1	10	100	1000
E(Δm_q) [mV]	-0.27	0.27	3.01	29.0
$\sigma(\Delta m_q)$ [mV]	0.18	0.23	0.21	1.02

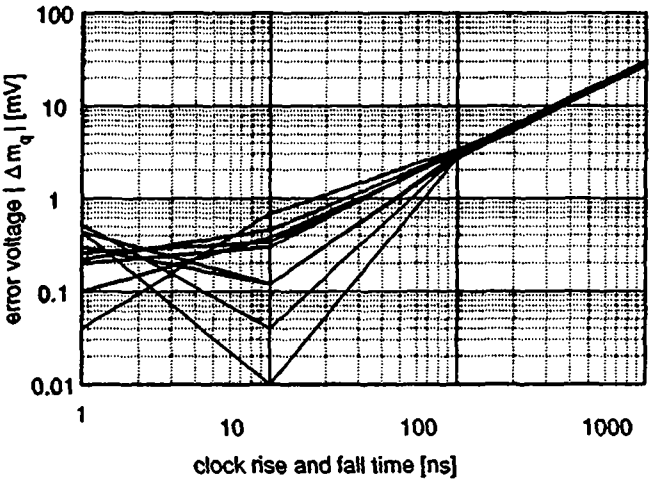


Fig. 8.16 Charge injection of 9 synapses, function of the clock rise and fall time.

For small rise and fall times, i.e. under 10 ns, the systematic effect is smaller than the contribution due to the mismatches only. Although the mean value for $t_{r,f} = 1$ ns is not equal to zero, showing another systematic effect due to the layout asymmetry, the relatively large standard deviation must be due to the mismatches.

Table 8.5 summarizes the parameter values needed to estimate the error on the synaptic weight given by relation (8.17). The capacitances C_S and C_H are the mean values of the two sides of the circuit, and have been obtained with an extract program. ΔV is the resulting voltage step on C_H, with the assumption of charge equipartition, i.e. using $Q_{tot}/2$. Since minimum-sized transistors have large mismatch, the estimation is made with $\varepsilon_{q1} = \varepsilon_{q2} = 10\%$, which gives

$\sigma(\Delta m_q) = 0.85$ mV. The measurements are below the estimated results, indicating that 10% mismatch is perhaps somewhat pessimistic.

Table 8.5

switch	V_{GON}-V_{TE} [V]	C_G [fF]	β [μA/V2]	C_S [fF]	C_H [fF]	Q_{tot} [fC]	ΔV [mV]
M_{S1}-M_{S3}	1.9	6.96	55	226	942	13.2	7.0
M_{S2}-M_{S4}	1.4	7.14	17	226	1030	10.0	4.9

8.6.3 Leakage measurement

The leakage measurements on 20 synapses have been grouped in Figure 8.17. The distribution of the results is more concentrated near the small values, and can be considered as a gaussian distribution. The mean value is close to zero and the standard deviation is about 2 mV/s at room temperature. This corresponds to a leakage current of about 2 fA with the total 1 pF capacitance obtained with the extract program.

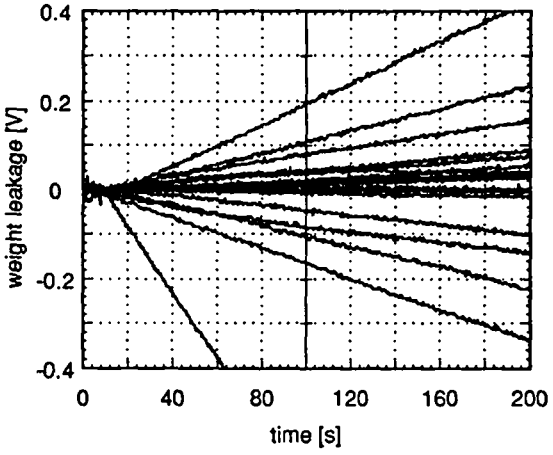


Fig. 8.17 Leakages of 20 sample synapses (about 3 minutes elapsed time).

The SACMOS technology has quite large junction reverse currents compared to other technologies. Measurements of these reverse currents [10] show quasi-linear V/I characteristics, but accurate statistics of their values are not available. Using the data listed in Appendix D, the leakage of a n+/p junction

of $3 \times 4\ \mu m^2$ would be about 940 pA at room temperature with a reverse voltage of 8 V. Therefore, the reduction of nearly three orders of magnitude attained with this low leakage technique is in accordance with the resulting reverse voltage reduction, which limits the junction voltage to the input offset voltage of amplifier *A*. The retention time is still orders of magnitude smaller than that of floating gate memories, but it is suitable to implement a Kohonen network operating under continuous learning, as stated in Chapter 4.

8.7 COMMENTS AND POSSIBLE IMPROVEMENTS OF THE DESIGN

The proposed synapse may seem excessively complicated for an element that should be ideally as simple as possible. This assumption must however be tempered if its performances are objectively considered. Indeed, its precision is suitable to implement a network with a few hundreds of cells, depending on the number of synapses per cell. As an example, a 10×10 neuron network with 10 synapses per neuron can be implemented on a single chip of less than 100 mm^2, including the pads. On the other hand, the current consumption of a single synapse can be estimated to about 70 µA (more than 50% for the multiplier) with the typical parameters, which leads to a power consumption of 350 mW for 1000 synapses with ±2.5 V power supplies. This consumption, which should not exceed 0.5 W with the neurons, is reasonable for a chip of 100 mm^2.

Since it is useless to exceed the number of neurons that is economically justified, the only improvement of the synapse must be found in improving its accuracy and/or density. This can be done by finding more efficient circuit techniques or by using submicron technologies with two metal layers. For this purpose, the differential structure should not be abandoned since it is the basis of both the charge injection compensation and the leakage reduction technique. In addition, the circuit measurements showed very good PSRR, although the test board was a simple experiment board, full of "flying" wires, with several standard CMOS logic chips connected to the same power supplies.

The major drawback of this implementation is the speed limitation due to the particular use of the shunting switch M_{S6} (see Figure 8.9). As no simple and dense solution has been found to solve this problem, the off-chip gain command described in paragraph 8.6.1 must be used.

In order to limit the errors due to charge injection when the weight's updates are small, the learning command could be disabled when the update current I_α is smaller than a fixed value. The extra circuitry needed to implement this function could be advantageously compensated by improved performance of the network, and this technique should therefore be investigated.

8.8 CONCLUSION

A plastic analogue medium-term memory implementing a synapse that can fulfil the requirements for the implementation of a Kohonen map has been proposed. The use of a differential switched capacitor-like structure together with an efficient leakage reduction technique has led to an accurate cell using a single control signal. Its accuracy is adapted to the implementation of a medium-sized network based on a single or multichip basis. Its apparent complexity must be viewed in the context of the required accuracy, or compared with an equivalent digital implementation with the same versatility.

8.9 REFERENCES

[1] G. Wegmann, E. A. Vittoz and F. Rahali: "Charge injection in analog MOS switches", IEEE JSSC, vol. 22, No 6, p. 1091, December 1989.

[2] E. A. Vittoz et al.: "Analog storage of adjustable synaptic Weights", proc. of the ITG/IEEE Workshop on Microelectronics for Neural Networks, Dortmund, Germany, June 25-26, 1990.

[3] P. Heim and E. A. Vittoz: "Precise analogue synapse for Kohonen feature maps", ESSCIRC, Sevilla, September 1993.

[4] E. Seevinck: "Analysis and Synthesis of Translinear Integrated Circuits", Elsevier, 1988.

[5] P. Heim, F. Krummenacher and E. A. Vittoz: "CMOS full-wave operational transconductance rectifier with improved DC transfer characteristic", Electronics Letters, 30th January 1992 Vol. 28 No. 3.

[6] F. Krummenacher: "Optimisation des filtres à capacité commutées a très faible consommation", Ph.-D. Thesis No 581, Ecole Polytechnique Fédérale de Lausanne, 1985.

[7] G. Wegmann: "Design and analysis techniques for dynamic current mirrors", Ph.D. Thesis No 890, Ecole Polytechnique Fédérale de Lausanne, 1990.

[8] D. Macq and P. Jespers: "Charge injection in current copier cells", Electronics Letters, 29th April 1993, Vol. 29 No. 9, pp780-781.

[9] F. Krummenacher and N. Joehl: "A 4 MHz CMOS continuous-time filter with on-chip automatic tuning", IEEE JSSC, vol. sc-23, pp. 750-758, June 1988.

[10] F. Krummenacher, personal communications.

CHAPTER 9

IMPLEMENTATION OF AN EVALUATION CHIP

9.1 INTRODUCTION

All the basic blocks needed for the analogue VLSI implementation of a Kohonen map have been detailed in the previous chapters. These blocks have been analysed in the context of the Kohonen algorithm, and it has been shown that all their weaknesses are tolerated by the algorithm. Therefore, the next obvious step is the implementation of a chip that uses the above blocks to build a working map. This chapter describes the design of a chip containing four neurons with three synapses each, which is intended to build a larger two-dimensional multichip network. However, some unexpected errors on the layout, which necessitated the preparation of a new metal mask, have delayed the fabrication of the chip. As a consequence, measurements of the whole network are not available yet, but preliminary results on single chips are given that demonstrate the proper operation of all the blocks.

9.2 GENERAL CONSIDERATIONS

The choice of the size of the network has been dictated by the cost and the dimensions of the chip. Nevertheless, the limited number of neurons per chip should not be a problem, since the multichip scheme is used. However, due to the expected precision, the three-dimensional data base practically limits the useful size of the network to about one hundred neurons, as shown in Chapter 4 with the simulations. Each cell contains three horizontally abutted synapses and the neuron functions that are distributed in three blocks fitting the width of the synapses. This disposition leads to efficient routing paths, and extra synapses can easily be added vertically and horizontally to fit the application requirement. Some tactical design weaknesses that are easily corrected have been found during measurements of single chips, and guidelines will be given for a successful redesign of the chip.

9.3 MODIFICATIONS TO THE SYNAPSES

The synapses are based on the design described in the previous chapter. The layout has been slightly modified by the addition of output source followers and output selectors, as shown in Figure 9.1.

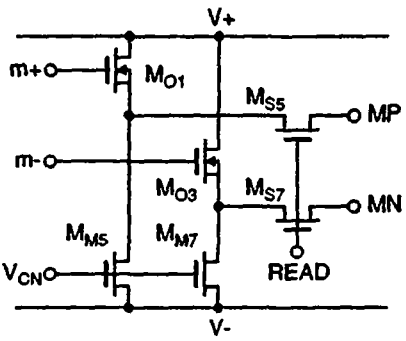


Fig. 9.1 *Added output source followers and output selectors.*

To avoid increasing the area of the synapse, some transistors have been placed under the routing path, and the transistor lengths of the linearized differential pairs have been shortened. For the same voltage range $V_{range} = 2$ V, the latter modification leads to a nominal bias current $I_q = 2.2$ µA, which is 10% larger than for the previous version. Table 9.1 summarizes the parameters of

the added and modified elements with respect to the above version of the synapse, and the new layout is shown in Figure 9.2. The one quadrant multiplier has been redrawn and better symmetry has been attained. The overall circuit density has been increased and the capacitor values have been increased to about 1 pF.

Table 9.1

Device	Type	W/L [μm]	β [μA/V^2]	Function
M_{M5}, M_{M7}	n	6/16	17.3	current sources
M_{O1}, M_{O3}	n	6/3	130.0	source followers
M_{S5}, M_{S7}	n	3/3	55.0	switches
M_{M2}-M_{M28}	p	12/12	18.5	mirror
M_{C2}-M_{C28}	p	12/5	47.3	cascode
M_{L2}-M_{L12}	p	5/2×11	3.8	diff. pair lin. transistor
M_{D2}-M_{D12}	p	3×5/11	22.8	differential pair
M_{D14}-M_{D20}	p	7/13	9.4	CMFB differential pair
M_{A2}, M_{A4}	p	34/4	180.2	amp. A differential pair

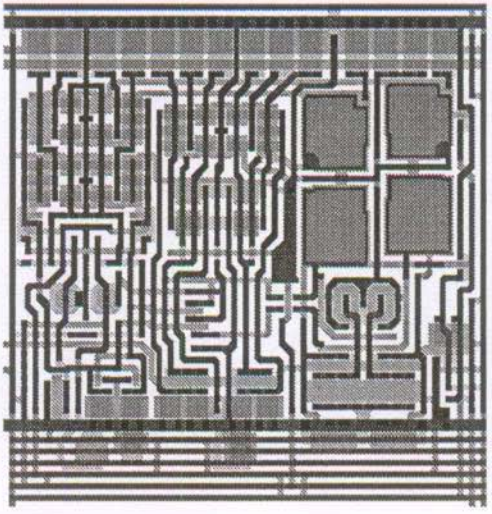


Fig. 9.2 New layout of the synapses.

9.4 IMPLEMENTATION OF THE NEURONS

The neurons have been split into three blocks that fit the width of the synapses. These three blocks contain respectively the WTA, the nonlinear diffusion network with the learning gain control circuitry, and the bubble logic detection with the associated latch. Due to the limited size of the chip, the output current of the WTA is directly used as the injection current for the nonlinear network. As a consequence, the voltage range available for both circuits is reduced by half, and the bubble size cannot be controlled by means of the injection current.

9.4.1 Design of the Winner-Take-All

The WTA has been designed to fulfil the capacitive load requirements for the 25 chips needed to implement a 10×10 neuron network. Due to the lack of accurate data, the capacitive load contribution has been estimated at 5 pF per chip. With the contribution of a gate capacitance of g_{mA}, the total bus capacitance has been estimated at 160 pF.

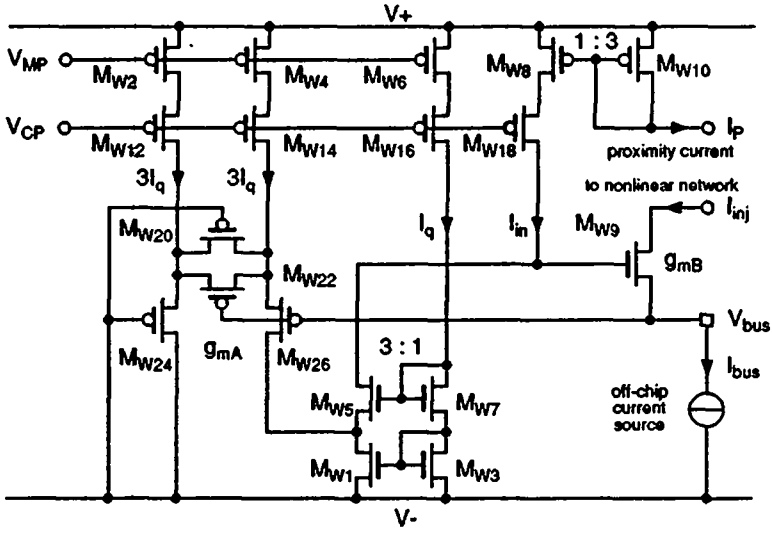


Fig. 9.3 Schematic of the Winner-Take-All.

The transconductance g_{mA} is implemented with a linearized differential pair polarised with $6I_q$. The sizes of the transistors have been chosen to fit the available area so that their mismatch is minimized. With a nominal value of bias current I_q = 2.2 µA, the transconductance is g_{mA} = 2.51 µA/V, and the input saturation voltage is V_{sWTA} = 2.11 V, leading to a maximum bus voltage of 1.75 V with respect to the negative supply voltage *V-*.

Table 9.2

Device	Type	W/L [µm]	β [µA/V2]	Function
M_{W2}, M_{W4}	p	3×12/12	55.5	mirror output $3I_q$
M_{W12}, M_{W14}	p	3×12/5	141.9	cascode
M_{W6}	p	12/12	18.5	mirror output I_q
M_{W16}	p	12/5	47.3	cascode
M_{W8}	p	2×10/12	30.4	mirror output $I_P/3$
M_{W10}	p	6×10/12	91.2	mirror input I_P
M_{W18}	p	2×10/5	77.7	cascode
M_{W20}, M_{W22}	p	10/2×16	5.6	diff. pair lin. transistor
M_{W24}, M_{W26}	p	3×10/16	33.8	differential pair
M_{W1}	n	3×19/12	248.2	mirror output $3I_q$
M_{W5}	n	3×19/5	682.5	cascode
M_{W3}	n	19/12	82.7	mirror input I_q
M_{W7}	n	19/5	227.5	cascode
M_{W9}	n	121/3	3000.0	WTA transistor "M_B"

The current source $I_{qC} = 3I_q$ (see Figure 6.11) is implemented with mirror M_{W3} - M_{W1} due to the absence of n-type current source I_q. The mismatch of both this mirror and the current source M_{W6} contribute to increasing the error in the selection of the winner. The large contribution of (small) transistor M_{W6} could be avoided by implementing the current source I_{qC} by simply connecting the gate voltages of M_{W1} and M_{W5} to the same bias reference of $3I_q$, at the expense of two extra routing wires. The proximity currents I_{Pi} of the three synapses are summed at the input of mirror M_{W8} - M_{W10}, and divided by 3 to respect the input current range $2I_q$ of the WTA. Finally, transistor M_{W9} (M_B in Chapter 6) has been drawn as large as possible to minimize its pinch-off voltage V_{P9} for the required bus current.

The purpose of this is to get the maximum voltage range for the nonlinear network. Using relation (A.8), this pinch-off voltage is given by:

$$V_{P9} = \sqrt{\frac{2\,I_{bus}}{n\,\beta_9}} + V_{bus} = 21.8\sqrt{I_{bus}} + V_{bus} \qquad (9.1)$$

To satisfy the stability condition (6.8), the product Cg_{mB*} must be greater than 820E-18. With $C = C_{G26} = 336$ fF (the gate capacitance of M_{W26}), the required bus current is 1.39 mA and the resulting pinch-off voltage is 2.56 V using $V_{bus} = 1.75$ V. Since it is somewhat excessive, an additional capacitor has been added so that $C \approx 600$ fF, which reduces the required bus current to 436 µA and the pinch-off voltage to 2.2 V.

To estimate the minimum wait time of the WTA, the following assumptions are made. From the three-dimensional example of paragraph 4.7, the smallest distance between neurons is estimated at about 8% of full scale, which corresponds to an input current change (i.e. I_W) of 94 nA. From relation (6.35), the internal node voltage swing is 3.8 V, and finally, relation (6.20) gives a winning time of 24 µs.

9.4.2 Design of the nonlinear diffusion network

The schematic of the nonlinear network and the logical detection of the bubble is shown in Figure 9.4.

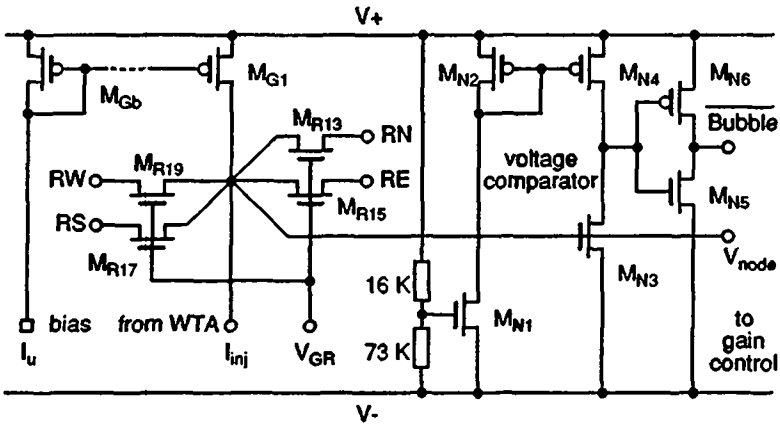


Fig. 9.4 Schematics of the nonlinear diffusion network and bubble detection.

Similarly to the other blocks, the nonlinear network has been fitted to the layout, while keeping some functional margin. The maximum injection current has been fixed at 450 μA and the larger bubble limited to 50 cells. From relation (5.9), the transconductance parameter of the resistors M_R is given by:

$$\beta_R > \frac{I_{inj}\left(I_u/I_{inj} - 1 - \ln\left(I_u/I_{inj}\right)\right)}{2\, n \ln 5 \left(V_{inj} - V_{off}\right)^2} \tag{9.2}$$

From Appendix D, the maximum pinch-off voltage for a n-type transistor with $V_G = 5$ V is approximately 3 V. Therefore, with a total supply voltage of 5 V, the voltage range $(V_{inj} - V_{off})$ allowed for the cone-shaped part of the characteristic is only 0.8 V. Using relation (9.2), β_R must be greater than 267 μA/V^2. Each resistor is made of two transistors M_R in series, which explains the value reported in Table 9.3. The purposes of which are to make the transient behaviour uniform in all the directions (because of the large capacitance associated with the multichip routing on board), and to average the "resistor" values at the chip interconnexions. The logical detection of the bubble uses a simple voltage comparator made of M_{N1} to M_{N4}, and proper digital levels are regenerated by means of invertor M_{N5} - M_{N6}. The bubble detection level has been fixed by means of two p-well resistors to save a routing wire.

Table 9.3

Device	Type	W/L [μm]	β [μA/V^2]	Function
M_{Gb}, M_{G1}	p	2×42/5	348.0	unit current source I_u
M_{R13}, M_{R19}	n	34/4	553.0	1/2 nonlinear resistor
M_{N1}, M_{N3}	n	3/28	4.1	voltage comparator
M_{N2}-M_{N4}	p	9/17	9.4	mirror
M_{C2}-M_{C28}	p	3/3	16.7	invertor
M_{L2}-M_{L12}	n	3/3	55.0	invertor

Since the injection current must be kept constant, the largest settling time of the network is given for the maximum number of cells in the bubble. There are 8 pads per chip for the interconnexions of the network, so that two pad capacitances of 5 pF are associated with each cell. For a bubble containing 50 cells ($N_t \approx 1$, see § 5.9), the settling time given by relation (5.13) is therefore:

$$t_s = \frac{2 \times 50 \times 5\ \text{pF}}{450\ \mu\text{A}} \left(\frac{2}{3} 2\ \text{V} + \frac{0.8\ \text{V}}{3} \right) = 1.8\ \mu\text{s}$$

9.4.3 Design of the learning gain control circuit

The learning gain must be controlled both by the bubble node voltage to implement the cone-chaped neighbourhood, and by an external signal V_{DS} common to all the cells. For this purpose, the simple variable transconductor [1] shown in Figure 9.5 has been used. The circuit on the left side of the schematic is repeated in all the cells, while that on the right side is a biasing circuit used to set the threshold voltage V_{TH} for the bubble dependant gain.

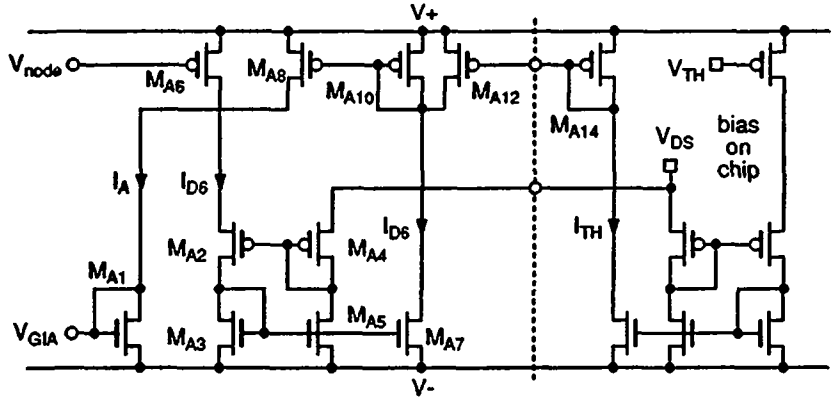


Fig. 9.5 Schematic of the gain control circuit.

Table 9.4

Device	Type	W/L [μm]	β [μA/V^2]	Function
M_{A1}	n	11/11	51.0	mirror input I_A
M_{A3}-M_{A7}	n	22/11	106.0	current conveyor
M_{A2}, M_{A4}	p	22/11	18.7	current conveyor
M_{A6}	p	2x10/14	25.9	transconductor
M_{A8}-M_{A14}	p	22/11	18.7	current subtractor

Transistors M_{A3}, M_{A5}, M_{A7}, M_{A2} and M_{A4} constitute a current conveyor [2]. The voltage V_{DS} applied at the source of M_{A4} is copied at the source of M_{A2} and therefore fixes the drain-to-source voltage of transistor M_{A6}. If the pinch-off voltage of M_{A6} is larger than V_{DS} it operates in conduction mode, and its drain current depends linearly on the node voltage V_{node}. To get a gain function that reaches zero at the boundary of the bubble, a second circuit is used to produce a current I_{TH} that corresponds to a node voltage V_{TH}, chosen equal to V_{off}. The current I_{TH} is then subtracted from I_{D6} by means of the unipolar current subtractor M_{A8} to M_{A14}, to produce the current I_A that controls the learning gain. I_A is therefore given by:

$$I_A = \beta_6 \left(V_{node} - V_{TH} \right) V_{DS} \tag{9.3}$$

For the nominal values used, the maximum allowed gain $\alpha = 0.3$ is obtained with $V_{DS} = 0.5$ V for $V_{node} - V_{TH} = 0.8$ V.

9.4.4 Design of the learning control logic

To avoid useless learning outside the bubble, the clock signal must be gated by the bubble logic detection. In addition, the bubble information is latched by means of the simple static latch shown in Figure 9.6 to avoid the activation of a cell during the learning process.

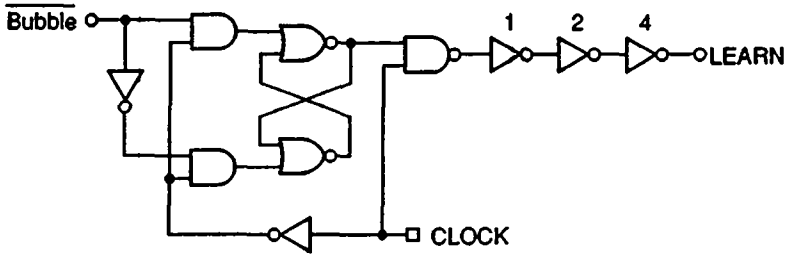


Fig. 9.6 Schematic of the learning control logic.

The learning signal is buffered by a cascade of three exponentially sized invertors that ensure rise and fall times smaller than 10 ns for an equivalent capacitive load of up to 20 synapses.

9.4.5 Layout of a complete neuron

The layout of the neuron is shown in Figure 9.7. The bottom left block contains the WTA, the bottom middle block contains the nonlinear diffusion network, the learning gain control circuitry and the bubble detection, and the bottom right block contains the learning control logic. The size of the cell is $750 \times 415\ \mu m^2$, but additional vertical routing paths are needed for the bias and I/O connexions.

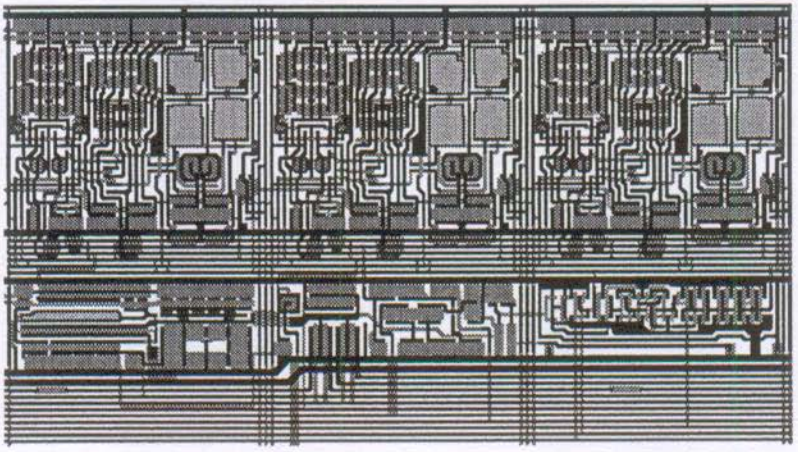


Fig. 9.7 Layout of the neuron.

9.5 POLARIZATIONS AND I/O INTERFACES

Apart from those already mentioned, the chip needs two important polarization circuits. The first generates the bias current I_q and fixes the saturation voltage of the differential pairs to be compatible with the multichip scheme described in Chapter 3. The second controls the pinch-off voltage of the transistors that implement the resistors of the nonlinear network. As these two biasing schemes are extensively described in Appendix B, they will not be developed in the present chapter. However, measurement of the chips has been possible thanks to a direct input pad for the bias current I_q, because the bias circuit was inoperational due to an unintentional short-circuit.

The I/O interfaces concern mainly the read access to the synaptic

weights, as the input vector components are directly connected to input pads. Due to the limited number of pads, a simple decoding scheme, based on row ($R0$, $R1$) and column ($C0$, $C1$) selection, has been chosen as shown in Figure 9.8. Each neuron includes an AND gate for the selection of its synaptic weights. In addition, a simple circuit is used to detect if the chip is selected, so that the outputs of several chips can be connected in parallel. Therefore, the synaptic weights of a complete multichip network can be accessed using a simple row/column sweeping scheme.

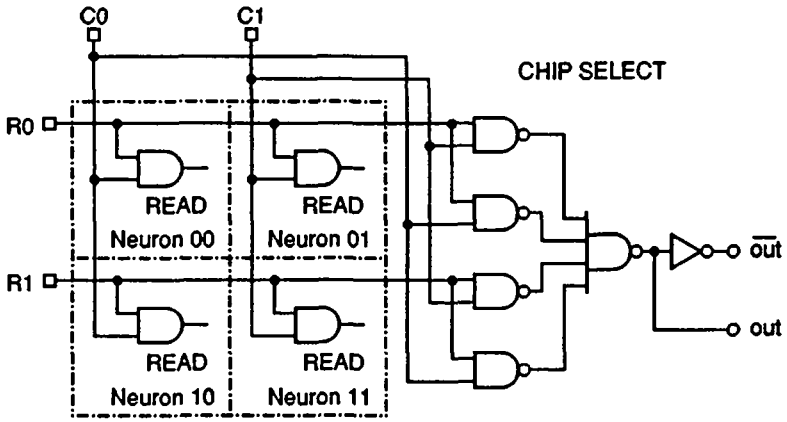


Fig. 9.8 Schematic of the decoder used to read the synaptic weights.

The outputs of the synaptic weights are made by means of currents. For this purpose, linearized transconductances similar to that of the WTA are used, as shown in Figure 9.9. The switches M_{S9} and M_{S11} connect the differential value of the synaptic weight (MN, MP) to the inputs of the differential pair when the chip is selected, and M_{S10} directs the corresponding weight current to the output pad. When the chip is not selected, switches M_{S12} and M_{S14} connect the inputs of the differential pair to the source voltage of a source follower similar to those of the synapses, with its gate voltage at ground. The weight current is also diverted to the negative supply by transistor M_{S8}. This circuit is repeated for each synaptic weight component, and the voltage conversion of the weights is made externally by means of a simple opamp-based I/V converter. It is worth noting that this current-mode output is not really suited to the multichip scheme, since the current scales may be different

between chips. In addition, as will be shown by the measurements, the external I/V converters are very sensitive to parasitic coupling with the surrounding digital signals. Therefore, further implementations of the network must use direct voltage-mode outputs.

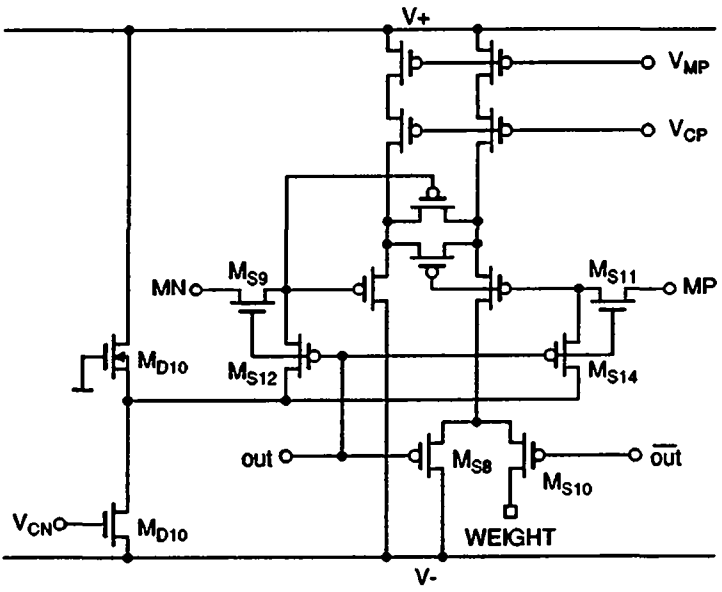


Fig. 9.9 Schematic of the current-mode synaptic weight output.

Table 9.5

Device	Type	W/L [μm]	β [μA/V^2]	Function
M_{S9}, M_{S11}	n	3/3	55.0	switch
M_{S8}, M_{S10}	p	27/3	199.0	switch
M_{S12}, M_{S14}	p	3/3	16.7	switch

9.6 MEASUREMENT OF THE CHIP

Since only isolated chips can be measured, it is not evident what can be done with a four-neuron Kohonen network. In addition, the learning parameters are under control of the network state and cannot be controlled

directly as was possible for a single synapse. A solution has been found with the particular sequence of input vectors shown in Figure 9.10. The components x_1 and x_2 are simply a two-bit binary sequence generated from the learning clock, and scaled to the input voltage range. Component x_3 is kept constant and equal to zero.

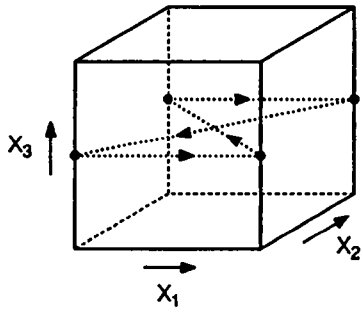


Fig. 9.10 Simplified input data base for single chip measurement.

Since there are four different input vectors and four neurons, it seems evident that each synaptic weight converges to its own input vector. Although attempts to realize this with the simulator were unsuccessful, it is possible with the circuit. For this purpose, the bubble must first contain the four neurons and the learning gain must be large enough so that all the weight vectors are maintained within the input vector space. Indeed, due to leakage, the weights can escape from the functional voltage range of the synapses. Then the bubble and the gain are slowly decreased until the network converges to the expected state. Nevertheless, several trials may be necessary before it succeeds.

In the following measurements, the input vectors are changed at the rising edge of CK_{WTA}. The high-state duration of CK_{WTA} is the waiting time for the WTA, and the high-state duration of CK_{LEARN} is the learning time. The first measurement uses a learning frequency of 20 KHz, which corresponds to the largest possible frequency deduced from the above calculations. The other measurements use a learning frequency of 10 KHz to improve the readability of the results in the presence of parasitic coupling induced glitches. To replace the capacitive load of a 25-chip implementation, a 120 pF capacitor has been added to the WTA bus.

Figure 9.11 shows the synaptic weights of one neuron when the network has

converged to the above-mentioned four-vector data base. The bubble contains the winner only, and the neurons have their synaptic weights successively refreshed to their respective input vector. The weights are therefore constant, and the glitches that are noticeable on m_2 and m_3 are due to parasitic coupling with the clock signal on the test board. On the other hand, since the weight vectors are equal to the input vectors, each neuron provides its maximum proximity value when it wins. This is a very interesting property because the mismatches that lead to the error on the selection of the winner can be deduced from the WTA bus voltage. For the measured circuit, the largest difference between the four neurons is about 0.1 V, which, referred to the average bus voltage of 1.4 V, leads to an error of 7%. A more accurate statistical value would necessitate measurements on several chips, but this chip was not the worst and the standard deviation of this error is certainly larger than the 5% used in Chapter 4 for the simulations, for the reason evoked in paragraph 9.4.1.

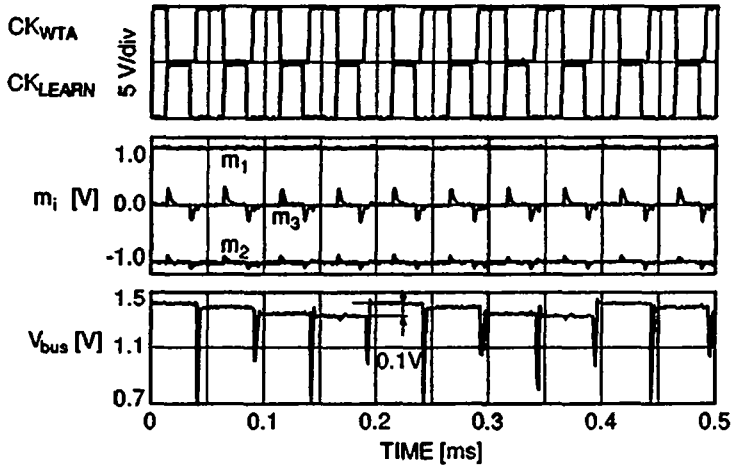


Fig. 9.11 Weight components of one neuron and WTA bus voltage for the four-neuron network mapping the data base depicted in Figure 9.10.

For the two following measurements, the input data base has been modified so that components x_1 to x_3 are a three-bit binary sequence generated from the learning clock. Figure 9.12 shows the evolution of the synaptic weights of one neuron for the largest possible learning gain and a bubble including the four

neurons. In these conditions, it is possible to maintain the process in a periodic state so that the weights follow the inputs. The evolution of the WTA bus voltage shows a decrease in the proximity value when a new input vector is presented and an increase that follows the weights update. The fact that the proximity value always decreases after an input vector change is particular to the four-neuron network and the data base used, because each update increases the distance of all the weights from the next input vector.

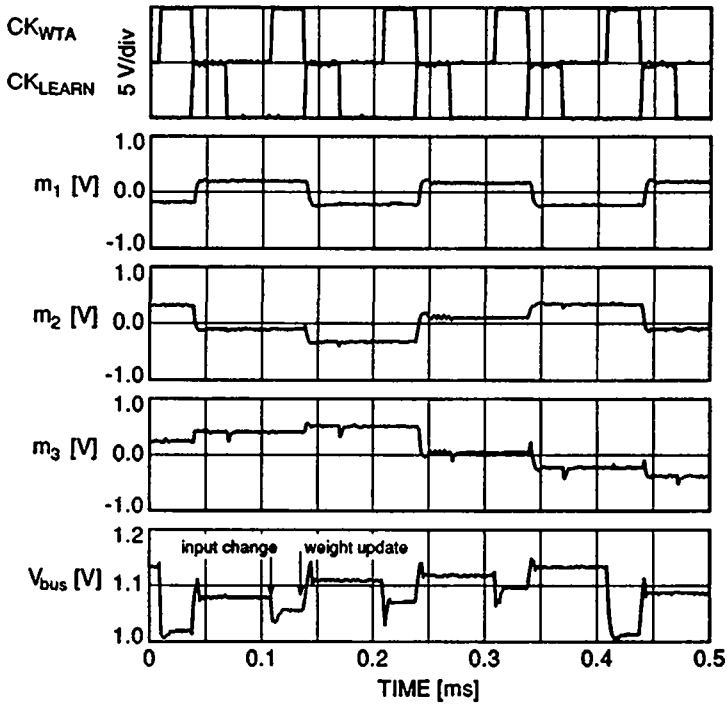


Fig. 9.12 Weight components of one neuron and WTA bus voltage for the four-neuron network running freely with the 8-vector data base (see text).

The measurement of Figure 9.13 was made under the same conditions as the previous one, except the learning gain has been set to the minimum possible value (about 0.02). For this purpose, the technique described in paragraph 8.6.1, which consists of adding a short pulse of increased gain at the beginning of the update phase, has been used. In this case, the pulse

corresponds to a gain of about 0.25 and has a duration of 10 μs. The spikes due to these pulses are clearly visible on the figure. Since the components of the input vector have an average value equal to zero, the synaptic weights are held near zero. The results also show that the learning time can be decreased to 20 μs without affecting the process.

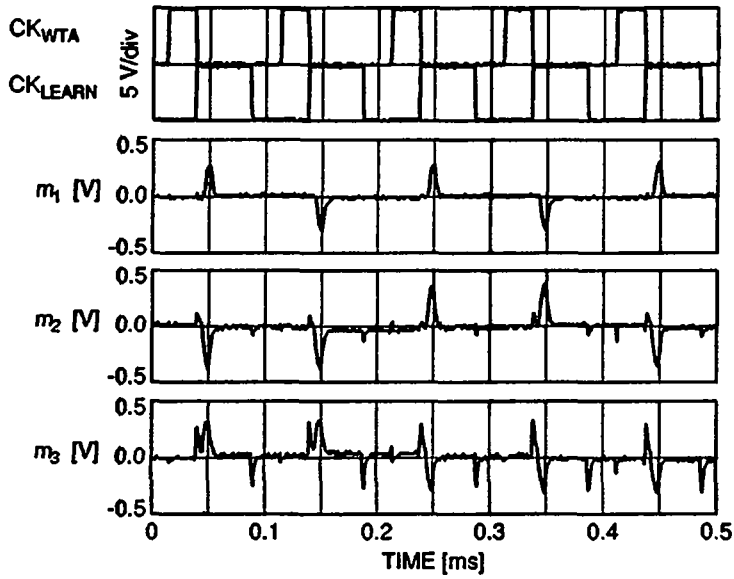


Fig. 9.13 Weight components of one neuron for the four-neuron network running freely with the 8-vector data base, using gain enhancement pulse.

9.7 CONCLUSION

A first evaluation chip for implementing an analogue Kohonen map has been described and measurement results have been presented. The proper operation of a single chip has been demonstrated for learning frequencies up to 20000 cycles per second. It has been pointed out that the precision in the selection of the winner should be enhanced to get a fully operational multichip implementation of the map.

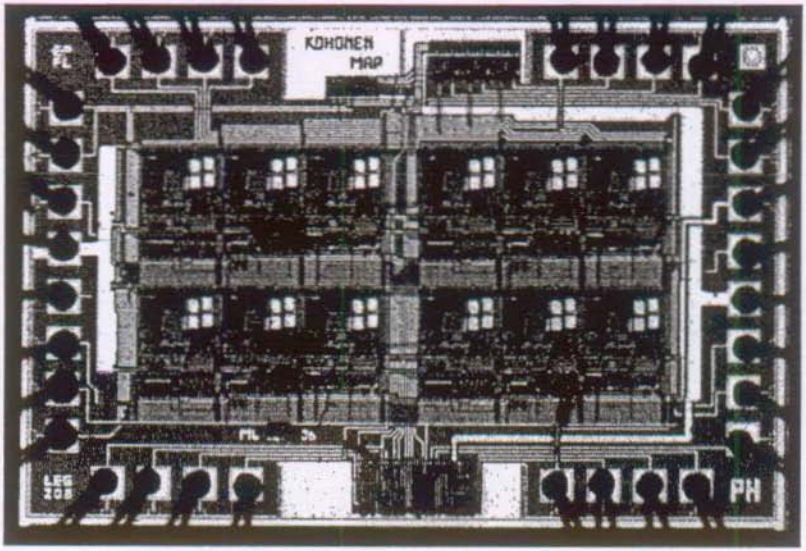


Fig. 9.14 Photograph of the chip.

9.8 REFERENCES

[1] B. Nauta, E. Klumperink and W. Kruiskamp: "A CMOS triode transconductor", ?

[2] A. S. Sedra: "The current conveyor: history and progress", International Symposium on Circuit and Systems (ISCAS), 1989.

CHAPTER 10

CONCLUSION

This work has culminated in the design of a fully analogue implementation of a Kohonen map, based on the simplified computational algorithm. To this end, the effects of circuit inaccuracies on the behaviour of the map were analysed, and circuits meeting the specific requirements were developed. The proposed architecture is suitable for the implementation of a medium-sized network, i.e. one containing a few hundred neurons and a few thousand synapses. This limitation is inherent to analogue circuits and is due to the capacitive load handling capability of the collective circuits (nonlinear network and Winner-Take-All). The expected learning frequency has been estimated at 20,000 cycles per second, depending slightly on the number of cells through the bus capacitance of the Winner-Take-All. Measurements of a complete multichip network are not available yet, but preliminary results on the different blocks demonstrate the feasibility of the proposed implementation.

To counter problems associated with a multichip implementation of the network, a general scheme which includes the required polarization circuits is suitable and has been proposed for the particular implementation. On the other hand, the underlying properties of the related polarization techniques may find applications in other fields.

A nonlinear diffusion network, well adapted to controlling the variable size of the learning neighbourhood, has been developed. It has the same structure as that of a linear resistance-conductance network, which limits interconnexions to nearest neighbours only. In addition, its nearly cone-shaped characteristic has been advantageously exploited to generate a gain function that accelerates the organization phase of the algorithm.

The Winner-Take-All circuit has been extensively analysed. A simplified design procedure, using a more comprehensive approach, has been derived that can be used to optimize the circuit. In addition, a modified version of the WTA has been proposed which is compatible with the multichip scheme, and suited to handle large bus capacitance.

A zero-crossing current-mode rectifier has been developed, which solves the problem of position error on the selection of the winner. Its poor transient behaviour is inherently tolerated by the algorithm. This shows that sometimes circuits that are not suited to more standard applications may find opportunistic applications in the domain of neural networks, and thus encourage the search for new circuits or, why not, a second look at circuits that have fallen into oblivion.

The synapses are certainly the most critical elements in analogue implementations of neural networks in general. Although these synapses should be as simple as possible, this work has been oriented towards a precise implementation due to the apparent lack of robustness of the Kohonen algorithm with respect to some types of errors. This has, however, stimulated the search for efficient techniques to minimize the circuit area. The proposed synapse implements the weight update rule over a wide range of learning gain. Its leakage value at room temperature represents 0.1% of full scale per second, and the charge injection represents an error of 0.02% of full scale, which is suitable for handling weight updates down to 0.2% of full scale. Although it is often claimed that the lack of long-term memories is a major drawback of analogue circuits, this work has shown that the Kohonen network can operate satisfactorily under continuous learning, using medium-term memories. This

may be true for other types of neural networks as well. The proposed leakage reduction technique confirms the feasibility of medium-term memories with standard CMOS technologies. The above-mentioned retention time, which may seem quite modest, must be compared with the learning period of the network.

The most apparent collective action of the network may be the effect of a large bubble against errors in the selection of the winner. This is the only error of this type, since the proposed implementation inherently avoids the errors of position. For the collective action to be effective, the relative bubble to network area ratio must be large enough so that the cell that should have won if no error was present is well inside the bubble. In the proposed implementation, the errors on the selection of the winner are determined by the accumulated inaccuracies along the path from the synapses to the Winner-Take-All, and cannot be avoided. The influence of these errors, and therefore the required bubble to network area ratio, depends on the local average distance between neurons. A similar dependence applies for the effects of charge injection, which is also a critical source of errors. It cannot be compensated by the algorithm, and leads to the most constraining accuracy requirements, which in turn limits the circuit density. This does not mean, however, that the sequential "simplified computational algorithm" is not suited to an analogue implementation of the map. A continuous version of the algorithm would certainly increase the precision of the cells thanks to the suppression of the sampling effect, but would not basically modify the behaviour of the network in the presence of other circuit inaccuracies.

As a matter of fact, there is nothing wrong with the analogue implementation of a Kohonen map, if it is used correctly. A particularity of the Kohonen network is that each cell is assumed to represent a class of the input data base that must be distinguishable from the other classes. In other words, the Kohonen network needs high accuracy because it has no intrinsic redundancy, although redundancy can be added by increasing the number of cells above the expected number of classes. The comparison with the nonmonotonicity of a D/A converter is pertinent, because it shows that increasing the number of cells above the limit imposed by the precision is not only useless, but also tends to mix the closest classes together. Therefore, since the effects of circuit inaccuracies depend on the average distance between neurons, a trade-off exists between the precision of the circuit and the dimension of the data base (for a given number of neurons). Very accurate software implementations are suitable for applications that use data bases with a small dimension, such as

those used for illustrative simulations. Conversely, low precision hardware implementations are suitable when the input vectors are sufficiently orthogonal, which means that the dimensionality of the data base must be relatively large. To obtain such data bases, the information available may first need to be processed. This does not mean that the Kohonen map is no longer useful. It is primarily intended to perform a dimension reduction, and its unique topology preserving abilities may be advantageously exploited in some applications.

The proposed high precision analogue implementation constitutes a good compromise for applications that need a hundred neurons and use input vectors with about 5 to 20 components. Under 5 components the mean update values become too small with respect to the circuit precision, and above 20 components this circuit's precision, and so its area, are no longer justified. Taking these considerations into account, this work constitutes a good basis for estimating the required circuit precision and area for a given application.

APPENDICES

APPENDIX A

THE MOS TRANSISTOR MODEL FOR HAND CALCULATION

A.1 INTRODUCTION

The MOS transistor model used in this work was developed by Enz and Krummenacher [1] and was based on the previous works of Vittoz [2] [3] and Oguey and Cserveny [4] on the modelling of the MOS transistor from weak to strong inversion. It is valid in all regions of operation for the large-signal and small-signal parameters, including weak inversion, moderate inversion, strong inversion, conduction and saturation. The aim of this appendix is to summarize the equations dedicated to the design and analysis of analogue circuits which are used throughout this work. The originality of the model is its symmetry with respect to source and drain which leads to a more comprehensive understanding of the transistor's behaviour. A minimum number of parameters makes it easy to use for hand calculations while accurately modelling second order effects such as length modulation and body effect. A very useful graphical model valid in strong inversion is also presented, which was proposed by Memelink [5] and published by Wallinga and Bult in [6]. This graphical model is compatible with the model presented here since it is also symmetrical with respect to the source and the drain of the transistor.

A.2 TRANSISTOR SYMBOLS AND PARAMETERS DEFINITION

The symbols of both an n- and a p-channel MOS transistor are shown in Figure A.1 together with their respective terminal voltage definitions.

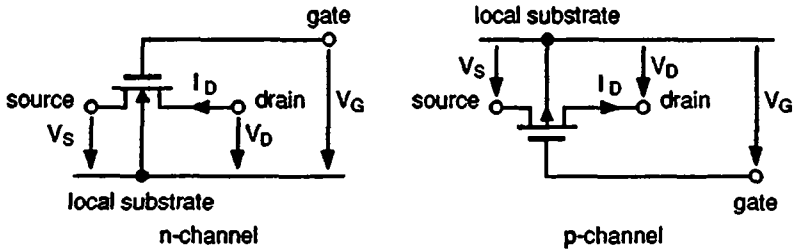


Fig. A.1 Symbols and definitions of an n- and a p-channel MOS transistor

The terminal voltages are all referred to the local substrate. With the conventions used, all the relations usually given for the n-channel transistor are also applicable to the p-channel transistor with the same positive values.

The following technology parameters are needed to describe the model:

- the **gate threshold voltage** V_{T0} for $V_S = 0$
- the **slope factor** n which represents the reduction of the gate voltage effect due to fixed charges in the channel
- the **transconductance parameter** $\beta = \mu C'_{ox} W/L$, where μ is the mobility of charge carriers, C'_{ox} the gate oxide capacitance per unit area and W and L the effective width and length of the transistor
- the **channel length modulation parameter** λ [V/µm] which is used to calculate the equivalent Early voltage as $V_E = \lambda L$

The **pinch-off voltage** V_P is the channel potential corresponding to the limit of weak inversion for a given gate voltage V_G. The factor n mentioned above is the slope of the function $V_G(V_P)$ and is a slowly decreasing function of V_G. Therefore, for hand calculation, n can be chosen constant and V_P can be approximated as follows:

$$V_P = \frac{V_G - V_{T0}}{n} \qquad \text{(A.1)}$$

The transistor being symmetrical, drain and source can be chosen freely. However, once the nodes have been defined, the drain current I_D arise from the superposition of two modes: a **forward mode** controlled by V_S and a **reverse mode** controlled by V_D, as shown in Figure A.2.

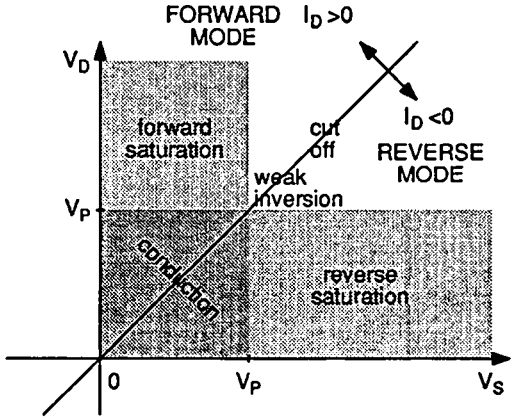


Fig. A.2 Modes of operation of the transistor [7].

The drain current I_D is the difference between a forward current I_F and a reverse current I_R:

$$I_D = I_F(V_S) - I_R(V_D) \tag{A.2}$$

The **forward saturation** arises as the reverse current vanishes ($V_D > V_P$):

$$I_{Dsat} = I_F(V_S) \tag{A.3}$$

The limit between weak and strong inversion is set by the **specific current:**

$$I_s = 2n\beta U_T^2 \tag{A.4}$$

$$\text{with} \quad U_T = kT/q \tag{A.5}$$

The **inversion coefficient** IC defines the mode of operation

$$IC = I_{Dsat}/I_S \tag{A.6}$$

- $IC \ll 1$ **weak inversion**
- $IC \gg 1$ **strong inversion:**

A.3 STRONG INVERSION LARGE-SIGNAL MODEL (FORWARD MODE)

conduction $$I_D = \frac{\beta n}{2}\left((V_P - V_S)^2 - (V_P - V_D)^2\right) \quad \text{(A.7)}$$

saturation $$I_{Dsat} = \frac{\beta n}{2}(V_P - V_S)^2 \quad \text{(A.8)}$$

The gate potential V_G can appear explicitly with the use of (A.1):

conduction $$I_D = \beta\left(V_G - V_{T0} - \frac{n}{2}(V_D + V_S)\right)(V_D - V_S) \quad \text{(A.9)}$$

saturation $$I_{Dsat} = \frac{\beta}{2n}(V_G - V_{T0} - nV_S)^2 \quad \text{(A.10)}$$

The transistor operates in conduction if (V_S and V_D) < V_P. It is saturated if $V_S < V_P < V_D$. The saturation voltage V_{Dsat} is thus:

$$V_{Dsat} = V_P \quad \text{(A.11)}$$

A.4 WEAK INVERSION LARGE-SIGNAL MODEL (FORWARD MODE)

conduction $$I_D = 2n\beta U_T^2\left(exp\left(\frac{V_P - V_S}{U_T}\right) - exp\left(\frac{V_P - V_D}{U_T}\right)\right) \quad \text{(A.12)}$$

saturation $$I_{Dsat} = 2n\beta U_T^2\, exp\left(\frac{V_P - V_S}{U_T}\right) \quad \text{(A.13)}$$

The gate potential V_G can appear explicitly with the use of (A.1):

conduction $$I_D = 2n\beta U_T^2\, exp\left(\frac{V_G - V_{T0}}{nU_T}\right)\left(exp\left(-\frac{V_S}{U_T}\right) - exp\left(-\frac{V_D}{U_T}\right)\right) \quad \text{(A.14)}$$

saturation $$I_{Dsat} = 2n\beta U_T^2\, exp\left(\frac{V_G - V_{T0} - nV_S}{nU_T}\right) \quad \text{(A.15)}$$

The saturation voltage in weak inversion (V_D and $V_S > V_P$) is reached for the following value, i.e. when the second exponential term becomes negligible:

$$V_{Dsat} = V_S + (4 \text{ to } 6)\, U_T \quad \text{(A.16)}$$

A.5 GENERAL SMALL-SIGNAL MODEL [3]

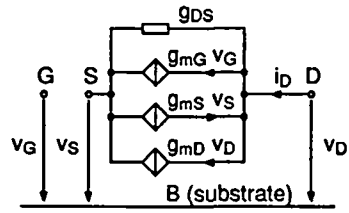


Fig. A.3 Small-signal equivalent circuit

	strong inversion		weak inversion	
	conduction	saturation		
g_m	$\beta(V_D - V_S)$	$\beta(V_P - V_S)$	$\frac{I_D}{nU_T}$	(A.17)
g_{mS}	$n\beta(V_P - V_S)$	$n\beta(V_P - V_S)$	$\frac{I_D/U_T}{1 - exp(-(V_D - V_S)/U_T)}$	(A.18)
g_{mD}	$n\beta(V_P - V_D)$	$\frac{I_D}{V_E}$	$\frac{I_D/U_T}{exp((V_D - V_S)/U_T) - 1} + \frac{I_D}{V_E}$	(A.19)

A.6 SMALL-SIGNAL MODEL IN FORWARD SATURATION [3]

	strong inversion	weak inversion	
g_m	$\beta(V_P - V_S) = \frac{2I_D}{n(V_P - V_S)} = \sqrt{\frac{2\beta I_D}{n}} = \frac{I_D}{nU_T\sqrt{IC}}$	$\frac{I_D}{nU_T}$	(A.20)
g_{mS}	ng_m	$\frac{I_D}{U_T} = ng_m$	(A.21)
g_{DS}	$\frac{I_D}{V_E}$	$\frac{I_D}{V_E}$	(A.22)
$A_0 = \frac{g_m}{g_{DS}}$	$\frac{2V_E}{n(V_P - V_S)} = V_E\sqrt{\frac{2\beta}{nI_D}} = \frac{V_E}{nU_T\sqrt{IC}}$	$\frac{V_E}{nU_T}$	(A.23)

A_0 is the **intrinsic voltage gain** of the transistor and reaches its largest value in weak inversion.

A.7 CONTINUOUS MODEL OF TRANSCONDUCTANCE

Because the spread of moderate inversion (the smooth transition between weak and strong inversion) is quite large, it is useful to have a good interpolation function for the small-signal parameters, since the asymptotic relations for weak and strong inversion are no longer valid (more than 50% error in excess for g_m at $IC = 1$). A good interpolation function has been proposed in [1] and will be used throughout this work:

$$G(IC) = \sqrt{1 + 0{,}5\sqrt{IC} + IC} \tag{A.24}$$

It is now possible to express the transconductance in saturation using its asymptotic values in weak and strong inversion:

$$\frac{g_m}{g_{mweak}} = \frac{1}{\sqrt{1 + 0{,}5\sqrt{IC} + IC}} \quad with \quad g_{mweak} = \frac{I_{Dsat}}{nU_T} = 2\beta U_T IC \tag{A.25}$$

$$\frac{g_m}{g_{mstrong}} = \frac{\sqrt{IC}}{\sqrt{1 + 0{,}5\sqrt{IC} + IC}} \quad with \quad g_{mstrong} = \sqrt{\frac{2\beta I_{Dsat}}{n}} = 2\beta U_T \sqrt{IC} \tag{A.26}$$

A.8 CONDUCTANCES

MOS transistors are often used as switching elements controlled by their gate potentials. The switch conductance can be approximated by the small-signal conductance for $V_D = V_S$:

$$G_{on} = 2n\beta U_T \exp\left(\frac{V_P - V_F}{U_T}\right) \quad for \quad V_S = V_D = V_F > V_P \tag{A.27}$$

$$G_{on} = n\beta(V_P - V_F) \quad for \quad V_S = V_D = V_F \leq V_P \tag{A.28}$$

A.9 THE MEMELINK GRAPHICAL MODEL IN STRONG INVERSION

This graphical model is an exact geometric representation of relations (A.7) to (A.10). In this context, it is primarily intended to avoid tedious algebraic transformations and allows a more comprehensive understanding of the static behaviour of MOS circuits. The $V_{TB}(V_{ch})$ line is the threshold

voltage referred to the substrate. The $V_{TS}(V_{ch})$ value would be the threshold voltage referred to the source if the source were the reference potential. The voltage $-Qinv/Cox$ is proportional to the inversion charge in the channel and is null as the channel is pinched off, i.e. when $V_{ch} \geq V_P$.

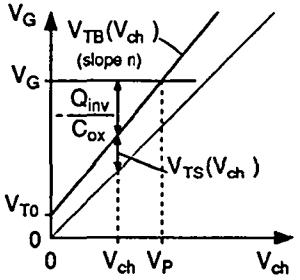


Fig. A.4 The Memelink graphical model in strong inversion

The shaded areas in Figure A.5 are equal to the drain current divided by the β of the transistor. The effect of a change in any of the terminal voltages is clearly visualised. In addition, the body effect is taken into account, and it is easy to study its influence on circuit behaviour. In a circuit, if a transistor is in a separate well not connected to the most negative voltage V^-, its graphical representation must be shifted along the diagonal $V_G = V_{ch}$ the value of its well potential with respect to V^-.

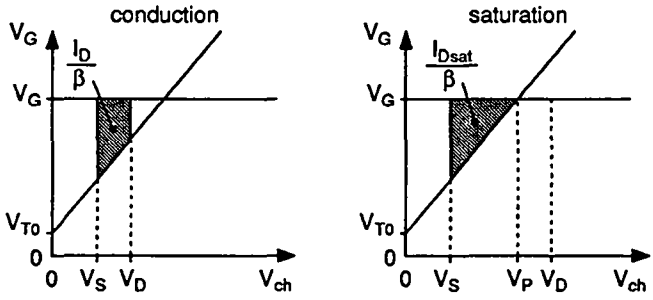


Fig. A.5 Graphical representation of the drain current in conduction and saturation.

A.10 BIBLIOGRAPHY

[1] C. C. Enz, "High Precision CMOS Micropower Amplifiers", Ph.D. Thesis No 802, Ecole Polytechnique Fédérale de Lausanne, 1989.

[2] E. A. Vittoz, "The Design of High Performance Analog Circuits on Digital CMOS Chips", IEEE JSSC, vol. SC-20, No 3, pp 657-665, June 1985.

[3] E. A. Vittoz, "MOS Transistors", Intensive Summer Course on CMOS VLSI Design: Analog & Digital, EPFL, 1989.

[4] H. J. Oguey and S. Cserveny, "MOS modelling at low current density", Summer Course on Process and Device Modelling, ESAT Leuven-Heverlee, Belgium, June 1983.

[5] J. A. van Nielen and O. W. Memelink: "Influence of substrate upon the DC characteristics of MOS transistors", Philips Research Reports, vol. 22, pp. 55-71, 1967.

[6] H. Wallinga and K. Bult: "Design and Analysis of CMOS Analog Signal Processing Circuits by Means of a Graphical MOST Model", IEEE JSSC, vol. 24, No 3, pp 672-680, June 1989.

[7] E. A. Vittoz, "Conception de circuits intégrés analogiques", Cours à option 7ème et 8ème semestres, EPFL, 1991-1992.

APPENDIX B

CONTROL OF V_P AND V_{idsat}

B.1 INTRODUCTION

A problem common to collective circuits is the need for global references to all the chips that have to work together. As stated in Chapter Δ, the signals are represented by physical variables that must be related to their own corresponding references. The only available references that are independent of technology parameters are the bandgap and the PTAT (proportional to absolute temperature) voltage references. Having these basic references is however not sufficient to make the corresponding circuits from several chips to work in the same way, because of the spread in technology parameters. For these reasons, and because the parallel distribution of signals is made by means of voltages, it is relevant to look for polarizations that fix the voltage scale of a circuit under control of an external reference voltage. Under these conditions, the polarization of multichip implementations can be controlled by means of a single wire.

This appendix proposes two polarizations under voltage reference control. The first one fixes the pinch-off voltage V_P of the transistors, and the second one fixes the same voltage scale (V_{idsat}) for the transfer characteristics of corresponding differential pairs among several chips. Both polarizations produce a current that depends on the technology parameters of each chip, which gives rise to the same value V_P or V_{idsat} for all the chips.

B.2 EXTRACTION OF THE PINCH-OFF VOLTAGE

Making a polarization to control the pinch-off voltage V_P implies it is possible to extract this voltage from the transistor. The simple circuit of Figure B.1 (a) can be considered as a single transistor with a tap providing an access to measure the potential at some location in the channel [1] [2].

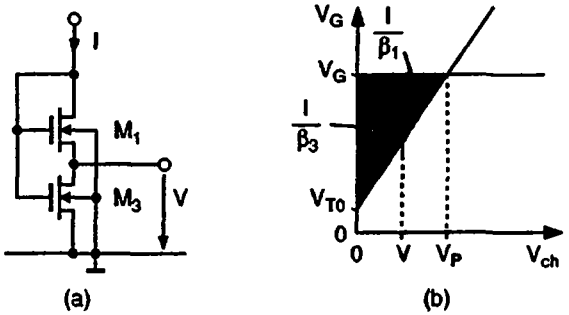


Fig. B.1 (a) circuit for extracting the channel potential and (b) graphical model.

The graphical model of Figure B.1 (b) shows clearly that the voltage V at the tap is a fraction ζ of the pinch-off voltage V_P. Furthermore, this fraction is independent of the current I, and depends only on the aspect ratios $(W/L)_1$ and $(W/L)_3$ of the transistors. Using relation (A.8) for M_1, which is saturated, and relation (A.7) for M_3, which is in conduction, yields:

$$I = \frac{\beta_1 n}{2}(V_P - V)^2 \tag{B.1}$$

$$I = \frac{\beta_3 n}{2}\left((V_P)^2 - (V_P - V)^2\right) \tag{B.2}$$

which lead to the following result:

$$V = \zeta V_P = \left(1 - \sqrt{\frac{\beta_3}{\beta_1 + \beta_3}}\right) V_P \tag{B.3}$$

If the widths of the two transistors are equal, relation (B.3) can rewritten as:

$$V = \zeta V_P = \left(1 - \sqrt{\frac{L_1}{L_1 + L_3}}\right) V_P \tag{B.4}$$

These expressions are valid in strong inversion only, because V_P becomes zero when the gate potential is equal to the threshold voltage. In weak inversion, the tap voltage is proportional to U_T and independent of the current, which suppresses any possibility of control.

In practice, to improve the accuracy of scaling factors, it is commonplace to restrict the choice to integer numbers and use combinations of identical transistors in series or parallel to implement them [1] [3]. In this context, a good choice is $\zeta = 0.5$, which requires only four elementary transistors, M_3 being made of three of them in series.

B.3 CONTROL OF V_P

The principle consists of controlling the tap voltage ζV_P of the circuit depicted in Figure B.1 (a) with an external reference voltage V_{ref} by means of a feedback loop. For this purpose, a current mirror M_2-M_4 imposes an equal current I_P to a second branch made of a single transistor M_1', similar to M_1, whose source is polarized by the voltage source V_{ref}, as shown in Figure B.2.

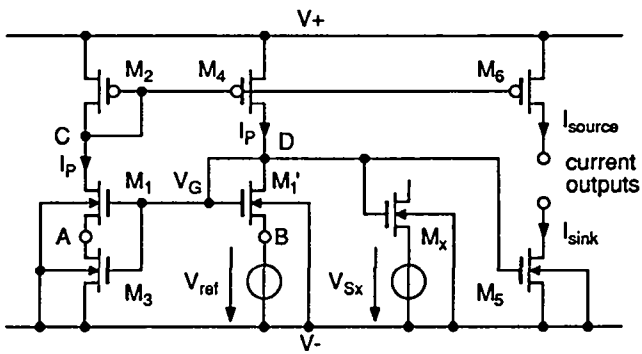


Fig. B.2 Circuit that controls V_P of n-channel transistors

Since transistors M_1 and M_1' are identical and have the same gate voltage and drain currents, their sources must be at the same potential, and therefore $\zeta V_P = V_{ref}$. The connexion of the mirror can be explained by considering that the circuit made of transistors M_1, M_1', M_2 and M_4 is a current conveyor [4], which works as follows: the access B is a voltage controlled access, i.e. an equal voltage appears at A, independent of the current supplied to A, and the

current flowing through B is equal to the current supplied to A, independent of the voltage applied to B.

As the pinch-off voltage is a function of the gate voltage only, any n-channel transistor M_X that has a gate voltage equal to V_G, has a pinch-off voltage equal to V_P. This is true whatever the size, the technology parameters and the source voltage of the transistor. In addition, this polarization is usable for both n-channel and p-channel transistors since no separate well is required. The need for such a polarization can be found in multichip collective circuits that require a well defined pinch-off voltage, as for example the non-linear network described in Chapter £.

The current I_P generated by the circuit has some properties that are interesting to analyse. According to relations (B.2) and (B.3), it is given by:

$$I_P = \frac{\beta_1 \beta_3}{\beta_1 + \beta_3} \frac{n}{2} \left(\frac{V_{ref}}{\zeta} \right)^2 = \frac{n\beta_{eq}}{2} V_P^2 \qquad \text{(B.5)}$$

where β_{eq} is the equivalent transconductance parameter of transistors M_1 and M_3 together. This result shows that if different chips are polarized with the same external voltage source V_{ref}, the spread of the currents I_{Pi} among chips (mismatches excepted) is that of the product $n\beta_0$, which does not exceed ± 15% in standard CMOS technologies. The currents I_{Pi} are also totally independent of the threshold voltages, which are generally given with large tolerance margins (up to ± 30% for low V_T technologies).

For the simultaneous polarization of several chips, the voltage source V_{ref} can be replaced by a single external resistor R_{ext}, thus transforming the polarization to the well-known polarization based on a degenerated mirror [1], [3]. In this way, all the above-mentioned properties are conserved because the accesses B of all the chips are connected together. For a number m of chips, their common pinch-off voltage is given by:

$$V_P = \frac{2\zeta}{n\langle\beta_{eq}\rangle R_{ext}} \qquad \text{(B.6)}$$

where $\langle\beta_{eq}\rangle$ is the mean value of β_{eq} for the m chips. On the other hand, the individual currents I_{Pi} are given by:

$$I_{Pi} = \frac{2\zeta^2}{n\, m\beta_{eqi} R_{ext}^2} \tag{B.7}$$

Another interesting property concerns the inversion coefficient IC. A transistor M_X of the considered type, polarized with a current I_X issued from the current I_P, will have an inversion coefficient IC_X independent of the technology parameters. Indeed, using the relations (A.4), (A.6) and (A.8), the inversion coefficient IC_X can be expressed as follows:

$$IC_x = \frac{(V_{Px} - V_{Sx})^2}{4U_T^2} = \frac{I_x}{I_P}\frac{\beta_{eq}}{\beta_x}\frac{V_P^2}{4U_T^2} \tag{B.8}$$

where V_P is the controlled pinch-off voltage, V_S being equal to zero. If, for instance, the reference voltage source V_{ref} is PTAT and generated on chip, the inversion coefficient IC_X of transistor M_X can be fixed by geometrical ratios only. It will be identical for any chip at any temperature and fabricated with any technology.

B.4 POLARIZATION OF DIFFERENTIAL PAIRS (CONTROL OF V_{idsat})

The asymptotic saturation voltage V_{idsat} of a differential pair is defined by the ratio of its bias current to its transconductance $2I_q/g_m$. It is a continuous function of the inversion coefficient IC, and thus depends on the bias current $2I_q$. Using the transconductance interpolation given by (A.25) or (A.26), a continuous relation for V_{idsat}, valid from weak to strong inversion, can be formulated as follows:

$$V_{idsat} = \frac{2I_q}{g_m} = 2nU_T\sqrt{1 + 0.5\sqrt{IC} + IC} \tag{B.9}$$

This function is drawn in Figure B.3 and shows the asymptotic value $2nU_T$ in weak inversion. Since in weak inversion V_{idsat} no longer depends on the bias current, it cannot be controlled and the proposed polarization is limited to the strong inversion region. The saturation voltage in strong inversion corresponds to the "gate overdrive voltage" of the transistors. Using the model presented in Appendix A, it can be expressed as:

$$V_{idsat} = n\,(V_P - V_S) \tag{B.10}$$

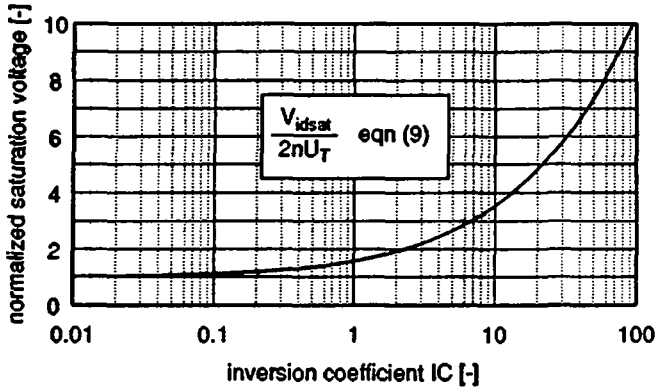


Fig. B.3 V_{idsat} from weak to strong inversion.

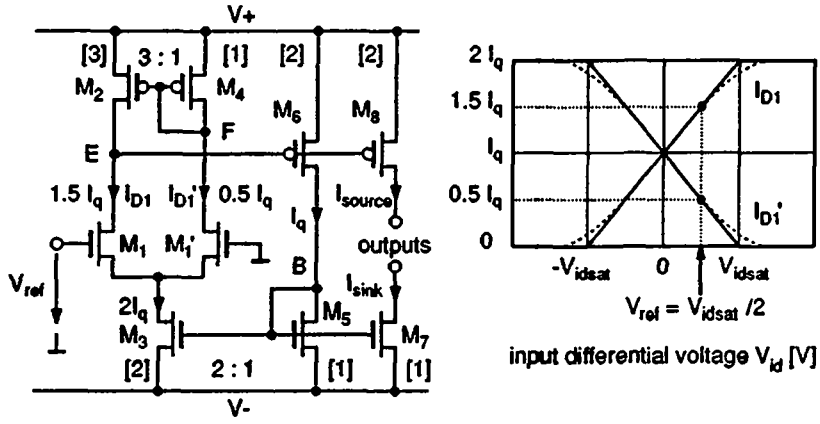


Fig. B.4 (a) Circuit that controls V_{idsat} and (b) graphical representation.

Figure B.4 shows a simple feedback loop configuration using a differential pair similar to those to be polarized. The numbers in brackets are the numbers of identical transistors in parallel. A reference voltage V_{ref} is applied at the input of the differential pair M_1-M_1', which splits the bias current $2I_q$ into two currents I_{D1} and I_{D1}'. The ratio of these two currents, which is a decreasing function of the bias current $2I_q$, is then compared by means of a current mirror M_4-M_2 of fixed ratio 1 to 3. The output of the so-

formed OTA (node E) is used to control the bias current $2I_q$ of the differential pair by means of M_6 and the mirror M_5-M_3. Transistor M_6 is made of two unit transistors in parallel to get $V_E = V_F$, thus minimizing the asymmetry of mirror M_4-M_2 due to its finite output conductance. Apart from being an integer, the ratio 1 to 3 has been chosen for several reasons. It fixes the saturation voltage at twice the reference voltage, as shown in Figure B.4 (b), which is still on the linear portion of the transfer characteristic (at this point, the output current is 6.5% smaller than its asymptotic value). On the other hand, it is large enough to limit the error due to the input offset voltage of the differential pair to a reasonable value.

This polarization can be implemented for both n-channel and p-channel differential pairs, and the value of n is that of differential pairs having their input common mode voltage around the reference ground. In addition, the reference input of the polarization has a very high impedance, which greatly simplifies the implementation of the voltage reference on board.

B.5 START-UP CONSIDERATIONS

It is worth considering whether the proposed polarizations are self starting or not when power supplies are applied to the circuit. Although the two polarizations are different, their ability to start up is related to the leakage currents I_{Rj} of the reverse biased junctions at the drains of the transistors. These leakage currents are generally comparable to or larger than the weak inversion currents at $V_G = 0$, even for low V_T technologies. As an example, Table B.1 summarises the calculated values using the typical parameters (the only one available for the leakage currents) for transistors having W/L = 10μm/10μm and drain areas of 10μm × 4μm, implemented in SACMOS technology (Appendix D).

Table B.1

	n-channel	p-channel	unit
$I_D(V_G=0)$	21.2	0.1	[pA]
I_{Rj}	440	188	[pA]

Assume that the polarization controlling V_P is cut off. There may exist a stable point at the origin ($I_P = 0$), if the equivalent leakage current drawn from node C is smaller than that drawn from node D. The polarization controlling V_{idsat} is much more difficult to analyse because there are four nodes in the circuit. It will however start-up if node E is pulled down by the resulting leakage current. In both cases, larger drain diffusion areas may be needed for some transistors to ensure proper start-up conditions. This concerns the drains of M_1 and/or M_4 for the first polarization, and the drains of M_1, M_3 and/or M_6 for the second polarization. This method is not very safe due to the large spread in leakage currents, and an extra start-up circuit should be added.

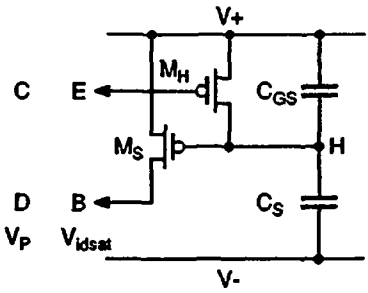


Fig. B.5 Start-up auxiliary circuits.

A suitable circuit is shown in Figure B.5. Before applying the power, the gate capacitance C_{GS} of transistor M_S is discharged thanks to the drain-to-bulk junction of transistor M_H, which is the only conducting path to node H. Transistors M_S and M_H can be minimum-size transistors. Capacitor C_S must be much larger than C_{GS} so that node H follows the negative supply voltage when the circuit is powered on. As a consequence, M_S injects a current at node D/B that has just to be sufficient to overtake the eventual parasitic stable state due to the leakage currents. As soon as the reference has started, transistor M_H cuts the start-up current off. Because this is a dynamic circuit, there is a minimum slew-rate value for the settling of the power supplies, and the capacitor C_S must be sized in consequence. As an example, a value of 100 fF is sufficient for a very low 1V/ms slew-rate.

B.6 STABILITY ANALYSIS

Because the polarizations are used to bias the circuits, they must not only be stable, but it is also desirable that they be slightly overdamped so that no oscillations occur when a perturbation is induced. The polarization that controls V_P contains two internal nodes only. There is no problem of stability and the analysis aims at obtaining real poles. Concerning the polarization that controls V_{idsat}, there is a stability problem because it contains three poles that are likely to be close to each other.

The small-signal equivalent circuit of the polarization controlling V_P is shown in Figure B.6. The condition to get two real poles can be imposed on the impedance seen from a node, e.g. node C. The analysis is made under the assumption that the current mirror M_2-M_4 has a ratio of one and M_3 is made of three transistors similar to M_1 in series. This leads to the simplifications written within the figure.

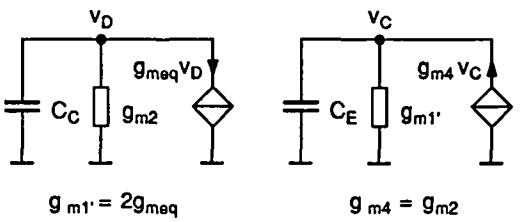


Fig. B.6 Small-signal equivalent circuit for the polarization controlling V_P.

The impedance Z_C seen from node C can be written in the following terms:

$$Z_C = \frac{2g_{m2} + sC_C}{3g_{m1'}g_{m2} + (g_{m1'}C_C + 2g_{m2}C_D)s + C_C C_D s^2} \tag{B.11}$$

To ensure that the impedance Z_C has two real poles, the discriminant Δ_C of the second order polynomial denominator must be positive, i.e.:

$$\Delta_C = g_{m1'}^2 C_C^2 - 10 g_{m1'} g_{m2} C_C C_D + 4 g_{m2}^2 C_D^2 > 0 \tag{B.12}$$

Using the algebraic sum and product rules for second order polynomials, this equation can be rewritten as follows:

$$\Delta_C = \left(g_{m1'}C_C - (5 + \sqrt{21})\, g_{m2} C_D\right)\left(g_{m1'}C_C - (5 - \sqrt{21})\, g_{m2} C_D\right) > 0 \tag{B.13}$$

This equation has two solutions depending on the signs of the two terms. To have both terms positive, it is sufficient that the smaller be positive (the left one), which yields:

$$C_C > (5 + \sqrt{21}) \frac{g_{m2}}{g_{m1'}} C_D \approx 10 \frac{g_{m2}}{g_{m1'}} C_D \qquad \text{(B.14)}$$

Similarly, to have both terms negative, it is sufficient that the larger be negative (the right one), which yields:

$$C_D > \frac{1}{(5 - \sqrt{21})} \frac{g_{m1'}}{g_{m2}} C_C \approx 2.5 \frac{g_{m1'}}{g_{m2}} C_C \qquad \text{(B.15)}$$

It seems more interesting to compensate the reference by means of C_D if the n-channel and p-channel transistors have the same size. However, in practice the choice of the compensation capacitor may depend on the relative number of n-channel or p-channel extra current output transistors (like M_5 or M_6) since they contribute to the capacitance value.

The stability of the polarization controlling V_{idsat} can be analysed by means of its open loop transfer function $T(s)$. For this purpose the loop can be open in any point where a voltage source v_x can be inserted without affecting the impedance loading [5]. This can be done by opening the circuit at the gate of transistor M_6, while keeping the total capacitance C_E at node E, as shown in the small-signal equivalent circuit of Figure B.7. In this case the open loop transfer function $T(s)$ is simply $-v_E / v_{G6}$.

To establish the small-signal equivalent circuit, the action of the differential pair must be considered carefully. As the small-signal parameters must be calculated for the equilibrium point of the circuit, the differential pair acts as a splitting element for the current source M_3, and its own transconductance will not appear in the small-signal equivalent circuit. At first sight, considering Figure B.4 (b), the split may be expressed as follows:

$$S^+ = \frac{1 + \delta_S}{2}; \quad S^- = \frac{1 - \delta_S}{2}; \quad \text{with } \delta_S = 0.5 \qquad \text{(B.16)}$$

However, when increasing the bias current of a differential pair, its total output current increases by the same value, but its input saturation voltage increases with the square root of the bias current. Therefore, as V_{ref} is

constant, the splitting is reduced with increased bias current, and vice versa. Using the approximation

$$\sqrt{1+\varepsilon} \approx 1 + \frac{\varepsilon}{2} \tag{B.17}$$

relations (B.16) must be rewritten as:

$$S_+ = \frac{1 + \delta_S/2}{2}; \quad S_- = \frac{1 - \delta_S/2}{2}; \quad \text{with } \delta_S = 0.5 \tag{B.18}$$

Finally, the differential pair together with transistor M_3 will be replaced by two transconductances of value:

$$g_{m+} = S_+ g_{m3} = \frac{5}{8} g_{m3} = \frac{5}{4} g_{m5} \tag{B.19}$$

$$g_{m-} = S_- g_{m3} = \frac{3}{8} g_{m3} = \frac{3}{4} g_{m5} \tag{B.20}$$

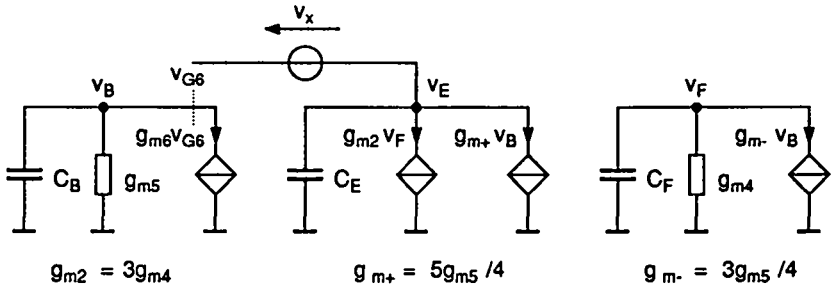


Fig. B.7 Small-signal equivalent circuit for the polarization controlling V_{idsat}.

From the resulting small signal equivalent circuit, the open loop transfer function $T(s)$ is found to be:

$$T(s) = \frac{\left(1 - 5s\dfrac{C_F}{g_{m4}}\right)}{s\dfrac{C_E}{g_{m6}}\left(1 + s\dfrac{C_B}{g_{m5}}\right)\left(1 + s\dfrac{C_F}{g_{m4}}\right)} \tag{B.21}$$

To compensate the circuit with high enough damping, the integrating time constant C_E/g_{m6} must be sized so that the excess phase contribution of the

other time constants does not exceed 30°, which leads to a phase margin of 60°. For this purpose, the integrating time constant must be at least twice the sum of the others, i.e.:

$$\frac{C_E}{g_{m6}} > 2\left(\frac{C_B}{g_{m5}} + \frac{C_F}{g_{m4}} + 5\frac{C_F}{g_{m4}}\right) \tag{B.22}$$

Since C_B/g_{m5} is just a bit smaller than C_F/g_{m4}, this relation can be further simplified to:

$$C_E > 14\frac{g_{m6}}{g_{m4}}C_F \tag{B.23}$$

Considering that $g_{m6} = 2g_{m4}$ and C_F is made of 4 (unit) gate capacitances C_G, the total capacitance C_E at node E must be larger than $112C_G$, which may appear quite large if circuit area is limited. On the other hand, it is worth noticing that the additional capacitor to C_E must be connected to the positive voltage supply.

B.7 CONCLUSION

Two polarization techniques have been presented that are suited to the implementation of multichip collective circuits. The main advantage of these polarizations is that a single wire connecting all the chips gives them some similar properties. The polarization that controls V_P is necessary for the proper operation of the nonlinear network described in Chapter £. It can also be used to control the inversion coefficient of the transistor by means of geometrical ratios, independently of the technology used. The polarization that controls V_{idsat} makes all the corresponding differential pairs work at the same voltage scale among chips, thus solving the problem of ratioed current comparison.

B.8 REFERENCES

[1] E. A. Vittoz, "Conception de circuits intégrés analogiques", Cours à option 7ème et 8ème semestres, EPFL, 1991-1992.

[2] C. C. Enz, private communication, 1990.

[3] E. A. Vittoz, "Elementary Building Blocs", Intensive Summer Course on CMOS VLSI Design: Analog & Digital, EPFL, 1989.

[4] A. S. Sedra: "The current conveyor: history and progress", International Symposium on Circuit and Systems (ISCAS), 1989.

[5] R. D. Middlebrook: "Design-oriented circuit analysis and measurement techniques, Part 1: Structured analog design", SIFU, Stockholm, December 5-8, 1988.

APPENDIX C

LOW-VOLTAGE CASCODE MIRROR IN STRONG INVERSION

C.1 INTRODUCTION

High impedance current sources or current mirrors are basic building blocs for analogue circuit implementations. To get high output impedance, the cascode technique is often the best suited solution. However, conventional cascode mirror structures such as stacked mirrors or Wilson mirrors have excessively high output saturation voltage, which is not acceptable for low voltage applications (5V or less). There are several techniques to reduce the saturation voltage [1] [2] [3], which use proper polarisation of the cascode transistor, but they usually need an extra branch or transistors in separate wells. This appendix proposes a cascode mirror that has low saturation voltage. The cascode transistor is polarised by means of a simple structure inserted in the input branch. In addition, design rules are proposed that ensure proper operation of the mirror within a given range of currents.

C.2 BASIC PRINCIPLE OF THE MIRROR

The basic configuration of the cascode mirror is shown in Figure C.1. The cascode transistor M_9 is polarized by means of the circuit described in Appendix B for the extraction of the pinch-off voltage. Since the latter is inserted in the input branch, no extra current source or extra branch are needed.

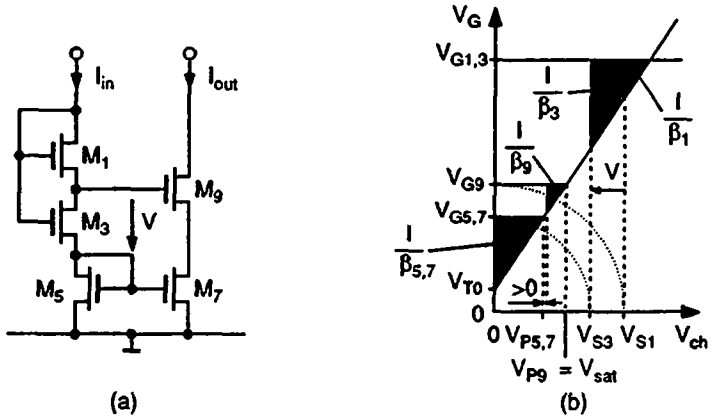


Fig. C.1 (a) Cascode mirror and (b) graphical representation.

The operation of the circuit is explained by first considering the structure made of M_1 and M_3. It is similar to that of Figure B.1, except its source voltage is not connected to ground, and relation (B.3) is modified as follows:

$$V = \zeta\left(V_{P1,3} - V_{S3}\right) = \zeta\sqrt{\frac{2I}{n\beta_{eq}}} \tag{C.1}$$

$$\text{with} \quad \zeta = \left(1 - \sqrt{\frac{\beta_3}{\beta_1 + \beta_3}}\right) \tag{C.2}$$

$$\text{and} \quad \beta_{eq} = \frac{\beta_1\beta_3}{\beta_1 + \beta_3} \tag{C.3}$$

On the other hand, the voltage V must be larger than the gate overdrive voltage of M_9, represented by the height of the triangle concerning M_9 on the graphical representation, which can be stated as:

$$V = \zeta \sqrt{\frac{2I}{n\beta_{eq}}} > n\left(V_{P9} - V_{S9}\right) = n\sqrt{\frac{2I}{n\beta_9}} \tag{C.4}$$

This relation, leads to

$$\beta_9 > \frac{\beta_1 n^2}{\left(\sqrt{\frac{\beta_1 + \beta_3}{\beta_3}} - 1\right)^2} = \frac{n^2}{\zeta^2}\beta_{eq} \tag{C.5}$$

For a worst case design, the largest possible value of n must be chosen (usually $n_{max} = 2$ if a more accurate value is not available). Relation (C.5) is valid for any current I and any value of $\beta_5 = \beta_7$, provided all the transistors are working in strong inversion. The saturation voltage V_{sat} of the cascode mirror corresponds to the pinch-off voltage of $M9$, and can be formulated as follows:

$$V_{sat} = \left(\frac{1}{\sqrt{\beta_5}} + \frac{\zeta}{n\sqrt{\beta_{eq}}}\right)\sqrt{\frac{2I}{n}} \tag{C.6}$$

which leads to the same value as that which could be obtained with any other biasing scheme that minimizes $V_{S9} - V_{P7}$.

C.3 CONCLUSION

A simple structure to polarize cascode mirrors for low voltage applications in strong inversion has been presented. The design of the basic implementation is extremely easy, and the mirror is operational over the whole range of currents for which the transistors are in strong inversion.

C.4 REFERENCES

[1] E. A. Vittoz, "The Design of High Performance Analog Circuits on Digital CMOS Chips", IEEE JSSC, vol. SC-20, No 3, pp 657-665, June 1985.

[2] E. A. Vittoz, "Elementary building blocs", Intensive Summer Course on CMOS VLSI Design 90': Analog & Digital, EPFL, Lausanne, 1990.

[3] P. J. Crawley and G. W. Roberts: "Designing Operational Transconductance Amplifiers For Low Voltage Operation", IEEE International Symposium on Integrated Circuits (ISCAS), Chicago, 1993.

APPENDIX D

SACMOS TECHNOLOGICAL PARAMETERS

D.1 TRANSISTOR PARAMETERS

Parameter	n-channel	p-channel	Units
V_{T0}	0.55 ± 0.15	0.60 ± 0.15	[V]
β_0	50 ± 5	19 ± 2	[μA/V²]
ΔW	-0.8	-0.8	[μm]
ΔL	-1.0	-0.5	[μm]
$n(V_G$=2.5V)	1.4	1.2	[-]
λ	5	5	[V/μm]

D.2 DIODE LEAKAGE CURRENTS (TYP. @ T=27 [C], $\|V_R\|$=8 [V])

Paremeter	p+/n	n+/p	p-well/sub	Units
bottom	30.0	60.0	1.0	[pA/mm²]
PS-edge	0.7	15.0	10.0 (edge)	[pA/mm]
LOCOS-edge	10.0	16.0		[pA/mm]

D.3 CAPACITANCES

D.3.1 Single plate dielectric capacitors

Parameter	min.	typ.	max.	unit
gateoxyde	0.665	0.700	0.735	
PS-LOCOS-sub	0.068	0.077	0.086	
PS-LOCOS-p-well	0.080	0.089	0.098	
IN-LOCOS-sub	0.048	0.054	0.061	
IN-LOCOS-p-well	0.051	0.058	0.065	[fF/µm^2]
IN-diff (n+ or p+)	0.105	0.126	0.153	
IN-PS	0.090	0.105	0.120	
IN-PS (CO on PS)	0.640	0.710	0.780	

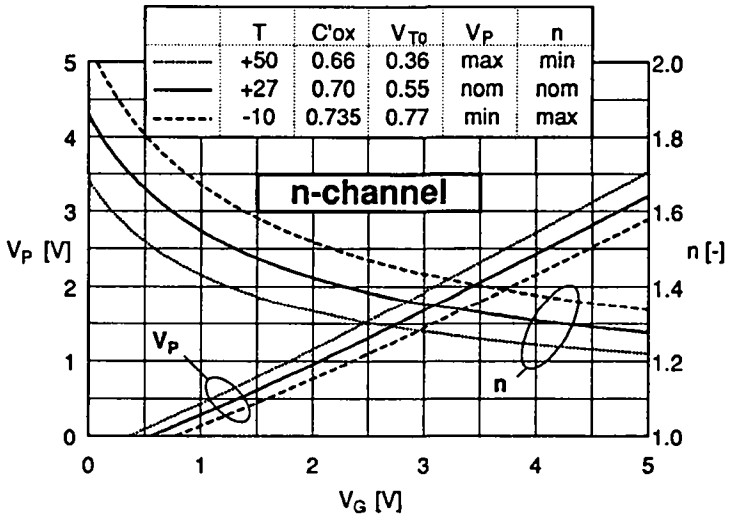
D.3.2 MOS transistor capacitances

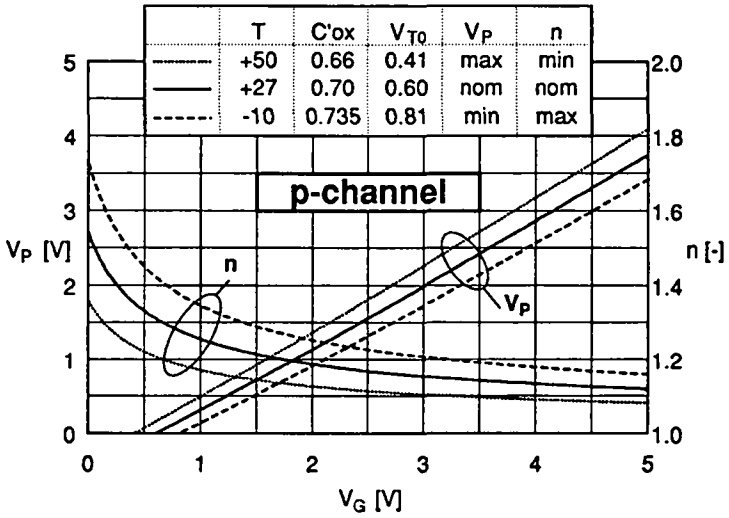
Parameter	typ.	typ.	typ.	unit
C_{ov} n-channel	0.110	0.140	0.460	[fF/µm]
C_{ov} p-channel	0.140	0.170	0.480	[fF/µm]
at overlap	-1.00	0.00	+1.00	[µm]

D.3.3 Junction capacitors

Parameter	min.	typ.	max.	unit
p+/n- bottom	0.130	0.160	0.190	[fF/µm^2]
PS-edge	0.120	0.150	0.180	[fF/µm]
LOCOS-edge	0.135	0.165	0.195	[fF/µm]
n+/p- bottom	0.360	0.450	0.540	[fF/µm^2]
PS-edge	0.200	0.250	0.300	[fF/µm]
LOCOS-edge	0.240	0.300	0.360	[fF/µm]
p-weel/sub.	0.060	0.078	0.095	[fF/µm^2]
LOCOS-edge	1.50	2.00	2.50	[fF/µm]

D.4 V_P and n versus V_G





D.5 REFERENCE

[1] C. Bleiker: "ZM17/19 parameters, FZV-3-10-51/646, 21-12-88, Version 6.1.1", FASELEC A.G., Zürich, 1988.

List of Publications

[1] E. A. Vittoz, P. Heim, X. Arreguit, F. Krummenacher and E. Sorouchyari: "Analog VLSI Implementation of a Kohonen Map", Journées d'électronique 1989, Lausanne, 1989.

[2] P. Heim, B. Hochet and E. Vittoz: "Generation of learning neighbourhood in Kohonen feature maps by means of simple nonlinear network", Electronics Letters, 31st January 1991 Vol. 27 No. 3.

[3] P. Heim, F. Krummenacher and E. A. Vittoz: "CMOS full-wave operational transconductance rectifier with improved DC transfer characteristic", Electronics Letters, 30th January 1992 Vol. 28 No. 3.

[4] O. Landolt, E. A. Vittoz and P. Heim: "CMOS selfbiased Euclidean distance computing circuit with high dynamic range", Electronics Letters, 13th February 1992 Vol. 28 No. 4.

[5] P. Heim, X. Arreguit et E. A. Vittoz: "Implémentation VLSI analogique des réseaux de Kohonen", Bulletin ASE/UCS 83 (1992)5, 13 mars.

[6] P. Heim: "Influence des défauts inhérents aux réalisations VLSI analogiques du réseau de Kohonen", rapport de projet du Cours Postgrade en Informatique Technique: Réseaux de Neurones Biologiques et Artificiels, EPFL, 1992.

[7] A. Mortara, E. A. Vittoz and P. Heim: "Simple PFM demodulator to be used by analogue artificial neural networks which communicate through pulses", Electronics Letters, 18th February 1993 Vol. 29 No. 4.

[8] P. Thiran, V. Peiris, P. Heim and B. Hochet: "Quantization Effects in Digitally Behaving Circuit Implementations of Kohonen Networks", accepted for publication in IEEE Transactions on Neural Networks.

[9] P. Heim and E. A. Vittoz: "Precise analogue synapse for Kohonen feature maps", Nineteenth European Solid-State Circuits Conference (ESSCIRC'93), Sevilla, 1993.

CURRICULUM VITAE

Surname:	Heim	**First name:**	Pascal
Date of birth:	21 February 1962	**Place of birth:**	Lausanne
Nationality:	Swiss	**Civil status:**	single

Addresses:

Home:	Sentier de la Gare 7 1020 RENENS Switzerland	**Professional:**	EPFL EL-LEG ELB-Ecublens 1015 LAUSANNE Switzerland + 41 21 693 46 03

Languages:

- French — first language
- English — First Certificate of Cambridge, grade B, 5 months in Australia in 1989

Studies:

- Diplôme d'ingénieur électricien ETS (technician level) 1982
- Diplôme d'ingénieur électricien EPFL (M.S) 1989

Postgraduate Studies:

- Intensive summer course on CMOS VLSI design: Analog & Digital (EPFL) 1989
- Cours postgrade en informatique technique: Réseaux de neurones biologiques et artificiels (EPFL) 1992

Professional Activities:

- Ing. ETS in a small electronics company: discrete electronics design using microcontrollers 1983–1984
- Ing. ETS at the ceramics laboratory at EPFL: some electronics and Fortran programming 1985/6 mths
- Research assistant at the electronics laboratory at EPFL: CMOS VLSI design related to neural networks 1989–1993