VLSI Systems for Analog and Hamming Parallel Computation

Thesis by

Volnei A. Pedroni

In Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

California Institute of Technology

Electrical Engineering Department

Pasadena, California

1995

Acknowledgments

I would like to express my gratitude to my advisor, Amnon Yariv, for providing this unique opportunity for education and growth here at the California Institute of Technology. Prof. Yariv has always been an unconditional source of support and encouragement, providing an ideal environment for the pursuit of this research. Under his guidance, my early interest in the technological aspects involved in physical realizations of parallel signal processing systems was allowed to expand and materialize.

I am also grateful to Gert Cauwenberghs and Charles Neugebauer, whom I deeply enjoyed working with. Our many long discussions on MOS/VLSI, CCD's, and related issues will always be remembered. My experience was equally enriched by Jose Tierno, through our many discussions on CS-related matters.

My gratitude goes also to the many members of the Caltech staff who made my time here a little easier by providing assistance on many occasions. Special thanks to Jana Mercado, Paula Samazan, Helen Carrier, Linda Dozsa, and Bette Linn.

I want also to thank all the members of my committee, professors Amnon Yariv, Yaser Abu-Mostafa, Carver Mead, David Middlebrook, and Axel Scherer, for both the time they took to be in the committee and for the positive impact that each one had on my work.

Finally, I wish to express my gratitude to my wife and children for their endless patience and support. They were and are the actual motivation behind it all.

Abstract

This thesis explores the vast field of physically implementing parallel-computing algorithms. In this research, we introduce a series of new circuit architectures and new technology applications, which implement multidimensional functions that are at the heart of many parallel signal processing systems, e.g., neural and Hamming networks, vector quantizers, and median filters. The functions are realized using low-cost, low-power, high-density technologies (CMOS and CCD), fully compatible with current industrial processes. The systems are either analog or hybrid, allowing lower time and/or storage complexities in many types of applications when compared to fully-digital systems. Special emphasis is placed on circuit modeling, with the purpose of thoroughly understanding the potentialities - and limitations - of each alternative. The models are verified experimentally on most occasions. As a consequence, the results presented in this dissertation are expected not only to provide new technological alternatives, but also new means of evaluating the technologies themselves.

Chapter 1 presents an introductory discussion on parallel systems. It has three main purposes. One is to describe some of the parallel functions whose implementations we are interested in. Another is to present a graphical discussion on how certain multidimensional systems work, which is probably the best way of describing - and appreciating - systems of this kind. And finally to describe basic guidelines concerning this research.

Chapter 2 discusses a function that is inherent to most analog parallel processors, the *winner-take-all* function. The reason for it to be developed first is that this function is part of many other function realizations. A global discussion is presented, which provides an overview on the

potentialities of most implementations available in CMOS technology, followed by high-resolution alternatives. The use of this function to implement other functions and systems is also illustrated.

Chapter 3 presents a detailed discussion on charge-coupled device (CCD) technology and its applications to parallel signal processing systems. This technology, compatible with conventional double-poly CMOS, is of interest due to its low power consumption and very high integration density, allowing the construction of very efficient vector-matrix multipliers and Hamming networks. To overcome its main limitation (i.e., charge-transfer inefficiency), a locally-controlled architecture is introduced. Several chips and extensive measurements are shown, with the purpose of concretely evaluating the performance of this technology when performing signal processing tasks.

Finally, Chapter 4 describes further research on CMOS cells that compute distance-based functions. These circuits allow the construction of LMS and other distance-based parallel processors, and provide additional valuable means of further examining the use of MOS technology for analog computation. Once again experimental results are presented, and the systems are illustrated through vector quantizers, Hamming networks, vector multipliers, and median filters. This chapter also provides further applications of the winner-take-all function to the construction of more complex functions.

Contents

List of Acronyms

BCCD - Buried Channel-Coupled Device

CAM - Content Addressable Memory

CCD - Charge-Coupled Device

CMOS - Complementary Metal-Oxide-Semiconductor (device)

D/AC - Digital-to-Analog Converter

GND - Ground

HN - Hamming Network

LAD - Least Absolute Distance

LCCD - Locally-Controlled Charge-Coupled Device

LMS - Least Mean Square

MOS - Metal-Oxide-Semiconductor (device)

MOSFET - Metal-Oxide-Semiconductor Field-Effect Transistor

NN - Neural Network

SCCD - Surface Charge-Coupled Device

VDD - Supply voltage

VF - Voltage Follower

VLSI - Very Large Scale Integration

VMM - Vector-Matrix Multiplier

VQ - Vector Quantizer

WTA - Winner-Take-All

Chapter 1

Introduction

Many are the applications in engineering in which not one but a series of signals must be processed in order for the system to produce the desired output. Classical examples are in the fields of neural networks, pattern recognition, and data compression, among others. Parallelism is highly desirable in these cases, for it reduces the time complexity of the system (and often the storage complexity as well). Many are also the aspects to be evaluated when considering hardware realizations of this sort. The ultimate goal of this evaluation is to attain a link between the competing algorithms and their final implementations in VLSI form. Well-established procedures are available for analyzing either side of the problem. However, linking mathematical models to circuit models is an interdisciplinary matter, making the task somehow intuitive and often complex. In this initial chapter we describe some of the parallel functions in whose implementations we are interested. Additionally, we present a series of graphical discussions on how certain multidimensional systems work, motivated by the fact that this is probably the best way of understanding (and appreciating) the operation of parallel systems. Finally, some basic guidelines concerning this research and the VLSI environment are described.

1.1 Analog vs. Digital Parallel Signal Processors

Many are the problems faced in engineering in which a system must process a multitude of signals in order to produce the appropriate output. Popular examples are in the fields of neural networks, pattern recognition, and data compression, among others. Additionally, data loss occurs inherently in many of these systems, requiring therefore no absolute computing accuracy, and the signals are generally analog rather than binary. These features directly contrast with von Neumann machines and other conventional digital implementations, which are essentially serial, thus presenting time complexities which are often greater than that of their parallel analog counterparts. Moreover, it is generally the case that VLSI implementations of the latter require less silicon area and consume less power. As a result, the use of parallel analog circuits has proven advantageous over conventional systems in many highly parallel, moderate accuracy applications. Additionally (and especially), analog models allow a deeper investigation on technology performance.

The growing interest in parallel signal processing systems, stirred mainly by the advent of neural networks, has caused an intense research effort aimed at systems of low time and storage complexities, with special emphasis on their VLSI hardware realizations. In this thesis we discuss a series of algorithms/implementations of this kind, notably Hamming networks, vector-matrix multipliers, and vector quantizers, among other multidimensional functions. The models and discussions are generally accompanied by experimental measurements realized on prototype chips fabricated through the MOSIS service.

1.2 Multidimensional Functions

The central operation in any parallel computing system consists essentially of some kind of vector manipulation. Three of the most commonly encountered functions of this type are the inner product, the rank-detection function, and distance-measure functions.

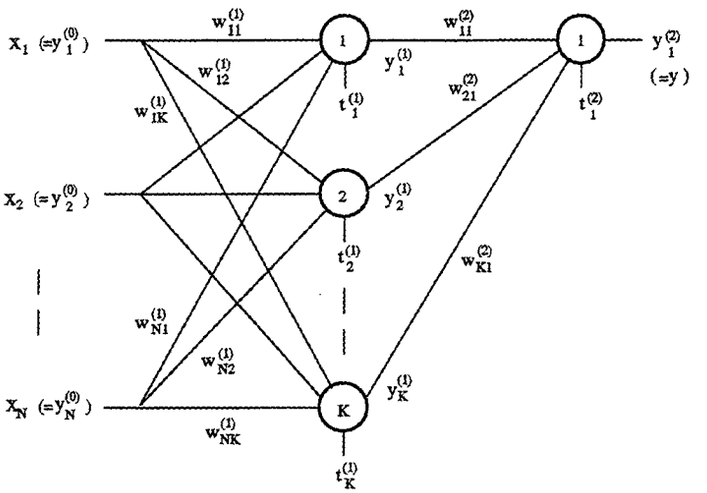


Fig. 1.1: Single hidden layer, single output feedforward neural net.

Inner product: Let us consider the feedforward neural network (NN) of Fig. 1.1. The basic operation performed by neuron j of layer k in computing its output is the recursive function

$$y_j^{(k)} = f(\sum_i y_i^{(k-1)} w_{ij}^{(k)}), \tag{1.1}$$

where $f()$ represents a threshold function (linear, sigmoid, etc.) and $(\mathbf{y}^{(k-1)}, \mathbf{w}_j^{(k)}) = \sum_i y_i^{(k-1)} w_{ij}^{(k)}$ is the inner product between the neuron's input vector $\mathbf{y}^{(k-1)}$ and its weight vector $\mathbf{w}_j^{(k)}$. During learning, vector-matrix multiplications are again the central operation performed by the system, no matter whether the training procedure is a supervised (e.g., back propagation [Rumelhart 86]) or an unsupervised one (e.g., Hebb's rule [Muller 90]); for instance, error gradient computation in supervised learning is done by means of successive vector-matrix multiplications. As a result, inner product is the single most commonly encountered function in the NN field.

Inner product is also the central operation in second order statistics computations (auto- and cross-correlation, covariance, etc.) and other matrix transformations. It frequently appears in the binary domain as well, as is the case of Hamming networks (HN's). The operation of a HN consists basically of, given a set $\mathbf{W}=\{\mathbf{w}^{(1)},\mathbf{w}^{(2)},...,\mathbf{w}^{(M)}\}$ of template vectors (codewords), where $\mathbf{w}^{(k)}=(w_1^{(k)},w_2^{(k)},...,w_N^{(k)})$, and an input vector $\mathbf{x}=(x_1,x_2,...,x_N)$, all with components $\{-1,1\}^N$, finding the element of $\mathbf{W}$ that is closest to $\mathbf{x}$ (i.e., that presents the smallest Hamming distance with respect to $\mathbf{x}$). This computation can be performed as depicted in Fig. 1.2, with $y=f()$ representing a pointer to the winning vector, that is,

$$y=f(\max_k\{\sum_i x_i w_i^{(k)}\}), \tag{1.2}$$

for, since all the vectors are of same length $\sqrt{N}$, the best-matching codeword is the one that yields the largest inner product with $\mathbf{x}$; or, in other words, the larger the inner product the smaller the

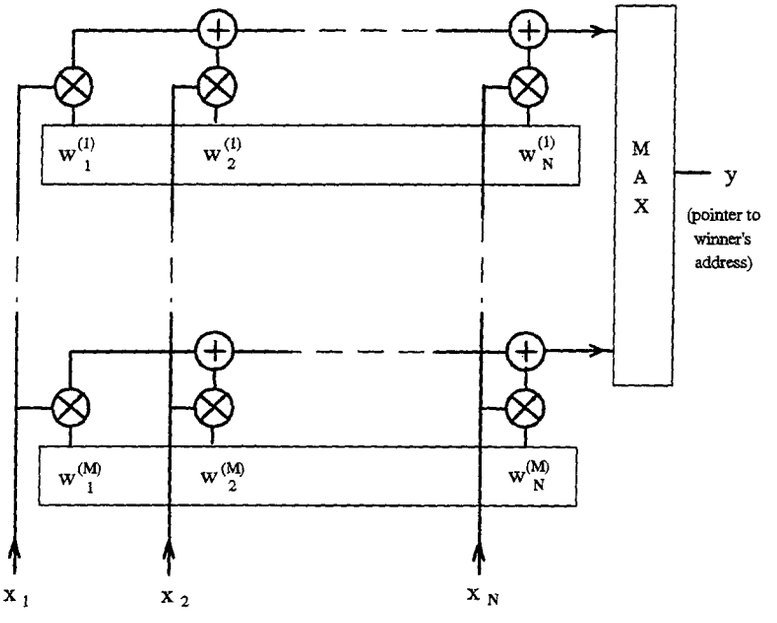


Fig. 1.2: Symbolic architecture of a Hamming network.

spatial angle between the corresponding vectors. As in (1.1), the primary computation in (1.2) is the inner product.

Rank-detection functions: Given a real-valued scalar set $X=\{x_1,x_2,...,x_N\}$, where we can assume, without loss of generality, that $x_1 \le x_2 \le ... \le x_N$, the *rank k* element of X is defined as

$$y=x_k\,. \tag{1.3}$$

Three particular cases of the rank function are of special interest to parallel systems: (a) $k=N$, in which case the maximum is detected; (b) $k=1$, in which case the minimum is detected; and (c) $k=(N+1)/2$ (N odd), which corresponds to finding the median of X. Applications of (a)-(b), also known as the *winner-take-all* (WTA) function, are very common; for instance, this is the case in Eq. (1.2). Applications of (c), on the other hand, can be found in the field of image processing, where median filters have experienced growing interest in recent years due to their ability to remove impulsive noise without great edge disturbance [Wendt 86][Yli-Harja 91]. Such is the importance of rank functions to parallel system realizations that a whole chapter will be dedicated to them in this dissertation.

Distance-measure functions: Many are the cases in signal processing where vectors must be compared. Hamming networks, described above, are one example of this kind. For real-valued vector components, however, the distance metrics are inevitably more complex and, in general, also more difficult to compute. As an example let us consider the case of vector quantization [Gray 84][Nasrabadi 88]. Similarly to Fig. 1.2, the basic computation performed by a vector quantizer (VQ) consists of, given an input vector $\mathbf{x}$, choosing from a set $\mathbf{W}=\{\mathbf{w}^{(1)},\mathbf{w}^{(2)},...,\mathbf{w}^{(M)}\}$ of template vectors the one that best represents $\mathbf{x}$ according to some pre-defined distance-measure criterion. Two of the most commonly used criteria are the *least absolute distance* (LAD) and the *least mean square* (LMS), defined by

$$y = \min_k \{\frac{1}{N}\sum_i |x_i - w_i^{(k)}|\} \tag{1.4}$$

and

$$y = \min_k \{\frac{1}{N}\sum_i (x_i - w_i^{(k)})^2\}, \tag{1.5}$$

respectively. Hardware implementations of these functions are relatively complex due to the need for multisignal multiplications and signed additions.

In the following chapters we discuss several alternatives to parallel VLSI implementations of the functions described above and other multidimensional functions, which constitute fundamental building blocks to most signal processing systems. The CMOS and CCD circuits presented in those chapters are intended not only as alternatives to hardware realizations of analog parallel processors, but also as a means of further understanding and evaluating the potentialities - and limitations - of this kind of technology.

1.3 Graphical Interpretation of Parallel Systems

We present in this section a series of graphical discussions on how certain multidimensional systems work. In the first part we discuss the operation of neural systems, while in the second part we develop an example of our own.

1.3.1 Threshold Functions and Neural Networks

The use of threshold functions for computation was intensely examined in the 1960's (e.g., [Lewis 67]), as an alternative replacement to conventional digital gates. This was followed by a period of relative obscurity until the resurgence of neural networks in the mid 1980's. A threshold gate can

be represented in the same way as a single neuron of Fig. 1.1, with $y=f()$ representing a step function and t representing the threshold decision level. Hence, for the threshold gate j of layer k,

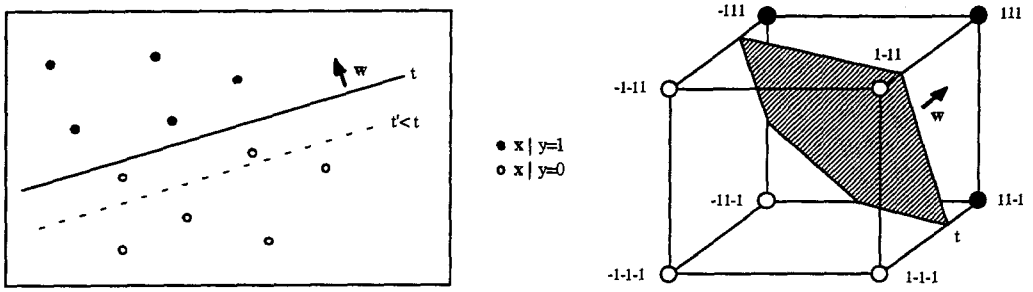
$$y_j^{(k)}=1 \text{ if } \sum_i y_i^{(k-1)} w_{ij}^{(k)} \geq t_j^{(k)}$$

$$y_j^{(k)}=0 \text{ (or -1) otherwise.} \qquad (1.6)$$

Although differentiable forms of $f()$ (notably sigmoidal functions of the form $1/(1+e^{-ax})$) are generally preferred from the viewpoint of learning [Rumelhart 86], (1.6) is easier to compute and simpler to implement in hardware. Besides, since the outputs are digital, it offers better scalability, making multichip networks easier to build.

Much of the analysis carried out in the field of neural networks during recent years was based on the model above; in particular, information capacity analysis, number of neurons needed to compute arbitrary functions, and size requirements for generalization. Function counting arguments, based on spatial analysis of threshold functions, proved to be powerful tools in the derivation of formal results.

A graphical interpretation of Eq. (1.6) is presented in Fig. 1.3(a). The weight vector $\mathbf{w}$ and the threshold t define a separating hyperplane in the N-dimensional space. The weights are responsible for the plane orientation ($\mathbf{w}$ is normal to the plane), whereas t determines its relative position (the plane moves parallel to itself when only t changes; for instance, $t'<t$ in the figure). The input vectors $\mathbf{x}^{(1)},\mathbf{x}^{(2)},\ldots,\mathbf{x}^{(M)}$ are represented by spatial points in the E^N. If a point lies either on top of the hyperplane or on the side pointed by $\mathbf{w}$ it produces an output 1; otherwise, it yields a 0 (or -1, being 0 and -1 considered equivalent for the purposes of this discussion). In other words, a point lies on the upper side of the plane if its inner product with $\mathbf{w}$ is at least t, as indicated by Eq. (1.6) and the dark points in Fig. 1.3(a).



(a) (b)

Fig. 1.3: (a) Graphical interpretation of threshold functions; (b) Particular case: a Hamming space (with N=3 and M=8).

Hamming spaces constitute a particular case of input variable distribution in which the input vector coordinates are restricted to $\{-1,1\}^N$. Consequently, every point lies in one vertex of a hypercube of dimension N, as illustrated in Fig. 1.3(b) (for N=3 and M=8). Threshold functions, as originally defined [e.g., Lewis 67], are functions of this type.

The well-defined structure of Hamming spaces makes mathematical analysis more feasible. One of the first discussions on the subject referred to the maximum number of distinct threshold functions that can be realized by a single gate (i.e., a single hyperplane). An upper bound for this number was derived in [Winder 60], resulting $O(M^N / N!)$, where M is again the number of points (specified inputs) in N-dimensional space. The bound above was obtained by counting the number of distinct ways in which a single plane can slice the cube. This result, applied to the NN of Fig. 1.1, shows that a feedforward net of this type cannot compute more than $O(M^{NK} / N!^K K!)$ dichotomies.

Another problem of equal interest relates to the number of bits needed to represent one neuron (one threshold gate in this case). This can be derived from the largest value needed for the components of **w** when constrained to be integers. It has been shown [Hong 87] that $O(N \log_2 N)$ bits suffice to

represent a weight in general. It has also been shown [Myhill 61] that certain functions cannot be implemented if all weights are less than $O(N/2)$ bits.

From the viewpoint of function implementation, network size can be linked to the degree of separability of the input variables. The set $\mathbf{X} = \{\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(M)}\}$ is said to be linearly separable (LS) if there is a single hyperplane that correctly classifies all of the inputs. Computing in this kind of space obviously requires just one neuron, but this is rarely the case in actual applications. Non-LS sets require multiple neurons in the hidden layer and at least two layers, as in Fig 1.1, being the size of the network expected to grow as the space becomes more intricate. It has been shown [Winder 63] that at least $O(\frac{M}{N}/\log_2\frac{eM}{N})$ neurons are required in the hidden layer for the circuit to be able to compute arbitrary functions, being $O(M/N)$ sufficient if the points are in general position [Baum 88]. Classical examples of non-LS spaces are the *parity detection* and the *symmetry detection* functions, which can be implemented as in Fig. 1.1, with K=N and K=2 neurons in the hidden layer, respectively (e.g., [Pedroni 93]). From the point of view of Fig. 1.1, however, there still is only one neuron at the output, which, as we know, can only separate sets that are LS. This indirectly describes the role of the hidden layer(s), which is to map the non-LS input set into a LS set, which can then be trivially separated into two groups by the output neuron. So, graphically speaking, the hidden layer implements a *remapping* function (non-LS to LS), as illustrated in Fig. 1.4.

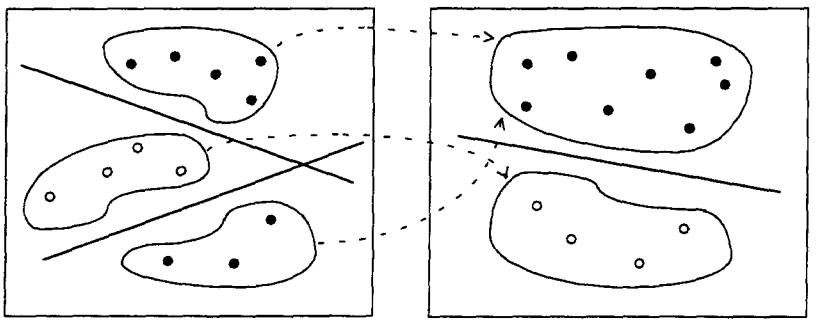


Fig. 1.4: Remapping action of the hidden layer.

From the point of view of neural networks, however, the circuit-size issue is more complex, for it must encompass not only the ability to learn, but also the ability to generalize. As the size of a network grows, so grows its information capacity [Abu-Mostafa 85][Olafsson 88], making learning more feasible. However, it is unlikely that the rate of successes on the examples will hold in general if the network is too large, for memorization is then expected rather than generalization [Abu-Mostafa 89][Baum 89][Baum 91]. For instance, the information capacity of the feedforward NN of Fig. 1.1 is $C \approx NK$, which is linearly dependent on network size; however, in order for the NN to be expected to achieve an error rate ε in general, it should provide the same error rate on $M' \approx NK / \varepsilon$ $(M' \leq M)$ examples [Abu-Mostafa 93], in which case large networks are obviously undesirable.

Threshold functions and neural networks can be used to implement many parallel signal processing tasks. A NN can directly emulate, for example, a programmable Hamming network (though with major limitations, as we show later). A modular system architecture, consisting of N separate circuits similar to that of Fig. 1.1, each one providing one of the output bits, can be employed in this case. During the training phase, a series of examples, drawn from E^N according to some underlying probability distribution, are presented to the network, which has its weights adapted to produce the closest template vector at the output. If training and network size are in good agreement (meaning that generalization is expected), the circuit is expected to operate appropriately in general.

1.3.2 "Rough-Estimator" Experiment

Many signal processing applications are inherently susceptible to data loss, as is the case of quatized data compression. Since error is inevitably present in this kind of system, even if it never

fails to retrieve the best internal representation, one might consider using a "rough" estimator instead (one that is allowed to make a wrong retrieval from time to time), in order to reduce the time complexity of the system, or the storage complexity, or both. Although we have no intention to provide a formal discussion on the matter, the discussion presented below is expected to be helpful in further illustrating the operation of parallel systems.

The NN implementation of a Hamming network described at the end of Sec. 1.3.1 is an example of this kind of estimator. As mentioned before, the system is expected to fail to retrieve the best vector representation in ε % of the cases. The time complexity of this implementation, for fully parallel neurons, is *O(2)*, therefore small. However, the storage complexity can be as high as $O(2^N)$, therefore often higher than the storage complexity $O(NK)$ of other (error-free) implementations. Another limitation of this system, when compared to the system described below, is that a wrongly retrieved vector is not necessarily in the vicinity of the ideal choice.

A different rough-estimator example is presented in Fig. 1.5. In this case, each codeword $\mathbf{w}^{(i)} \in \mathbf{W}$ is surrounded by a cloud of attractors (spacewords) $\mathbf{s}^{(i1)},\ldots,\mathbf{s}^{(iK_i)} \in \mathbf{S}^{(i)}$. When an input vector $\mathbf{x}$ is received, it is attracted to the closest spaceword, which is then decoded to the corresponding codeword by means of a lookup table. An error occurs when the closest attractor and the input are not in the same domain, as indicated in Fig. 1.5, where the input is attracted to $\mathbf{s}^{(jk)}$, producing $\mathbf{w}^{(j)}$ at the output instead of $\mathbf{w}^{(i)}$. The motivation for running this experiment is twofold: first, the mapping $\mathbf{X} \rightarrow \mathbf{S}$ can be of low storage complexity, even if the set $\mathbf{S}$ is large, for systematic procedures can be employed; second, a wrong retrieval does not mean a catastrophic error, because the correct and the chosen domains are necessarily adjacent. The latter implies that, even if a high vector-error rate occurs, the bit-error rate might still be acceptable. Say that N represents the space dimension, as before, and $|\mathbf{W}|$ and $|\mathbf{S}|$ represent the cardinality of $\mathbf{W}$ and $\mathbf{S}$, respectively. An input is correctly classified if either it is attracted to the right domain or it directly coincides with an element of $\mathbf{W}$ or $\mathbf{S}$. For $|\mathbf{W}| << |\mathbf{S}| << 2^N$, the probability of the latter is negligible, whereas the first

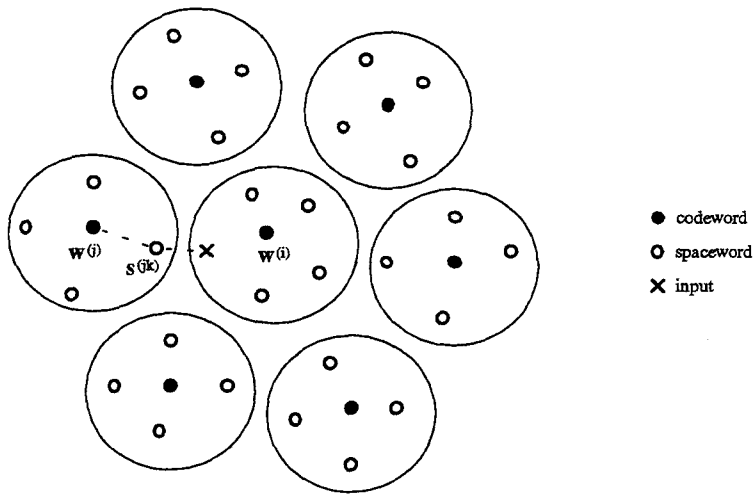


Fig. 1.5: Graphical interpretation of the rough-estimator experiment.

rapidly decreases toward 0.5. It is lower bounded, however, and the bit error rate is expected to be relatively low.

Figures 1.6 and 1.7 describe an experiment of this kind, which was run on a system with N=16, $|\mathbf{W}|$=32, and $|\mathbf{S}|$=1024. Two separate sets of codewords were used: a pre-defined one, which is shown in Fig. 1.6 (plus the 16 complementary words), and a set of random codewords. A Viterbi decoder [Viterbi 79] was used to provide the (systematic) mapping from $\mathbf{X}$ to $\mathbf{S}$, with code constraint K=3 (i.e., $2^{K-1} = 4$ nodes) and 2 bits per trellis step (therefore achieving the desired low storage complexity). (Notice also that, even for larger system parameters, a low-constraint Viterbi decoder can still be used.) The retrieval error rate for three runs of the experiment on each set of codewords is presented in Fig. 1.7(a), being the average bit-error rate for the random codewords plotted in Fig. 1.7(b). As can be seen, though the former is close to 0.5, the second adds little to the error inherently present in the system.

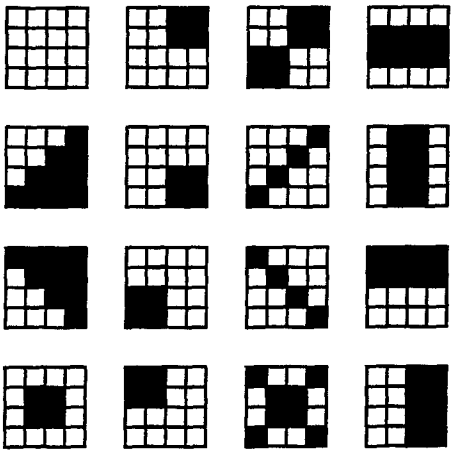


Fig. 1.6: One of the codeword sets (plus complementary words) used in the experiment described above.

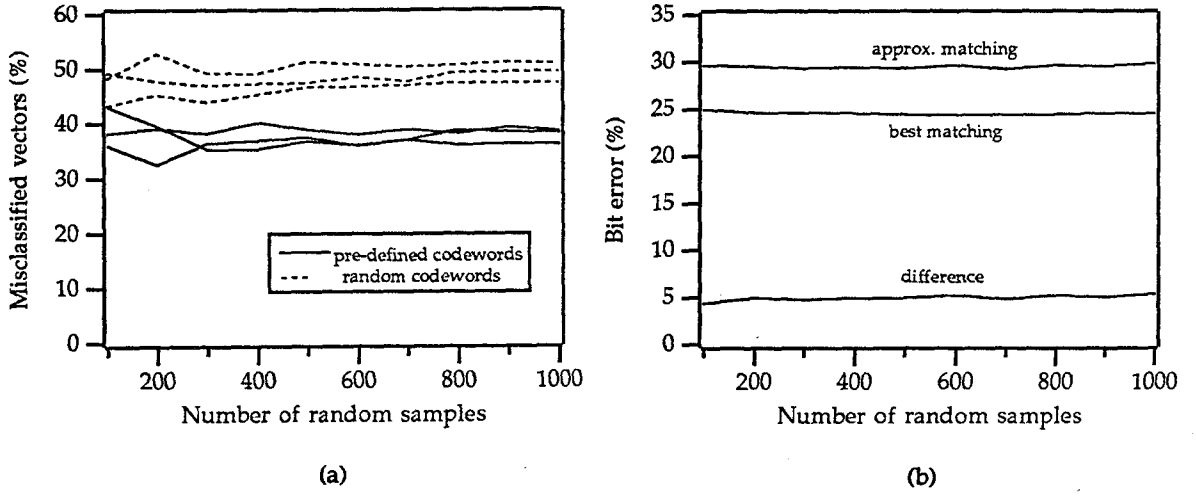


Fig. 1.7: Six runs of the rough-estimator experiment, with the vector-error rate plotted in (a) and the bit-error rate in (b).

1.4 VLSI Perspective

Before we start the discussions on particular system implementations, we present some remarks regarding the technologies and basic guidelines that were adopted in this research.

1.4.1 Work Premises

The search for VLSI alternatives capable of implementing functions like those describe in Sec. 1.2 is a very broad, interdisciplinary task. Although several technological options are available, low-cost, low-power, high-density technologies (e.g., CMOS and CMOS-compatible) are of greater interest. Another premise regarding this research is that the circuits and systems should be application-oriented, and the models and discussions should be experimentally verified as often as possible. Finally, the work should provide not only new system alternatives, but also means of further understanding and evaluating the potentialities of silicon-implemented analog parallel signal processors, with the mathematical results expressed always in low-entropy form.

1.4.2 MOSIS Parameters

As mentioned earlier, all of the chips referred to in this work were fabricated through the MOSIS service. The CMOS/VLSI technology adopted in these chips was either 1.2μm or 2.0μm (mostly n-well, though some p-well chips were also fabricated), with the latter also used for the fabrication of the charge-coupled devices (CCD's) described in Chapter 3. In analyzing experimental results, as well as in presenting theoretical examples, typical MOSIS parameters were employed, being the main parameters of interest summarized in Table 1.1.

Table 1.1: Typical MOSIS parameters for 1.2μm and 2.0μm n-well CMOS processes.

Name	Symbol	Typical values 1.2μm CMOS n-trans.	p-trans.	Typical values 2.0μm CMOS n-trans.	p-trans.	Unit
Threshold voltage (@ $V_{SB} = 0$ V)	V_{TO}	0.7-0.8	0.9-1.0	0.8-0.9	0.9-1.0	V
Transconductance coefficient (@ W/L=1)	β	70-100	20-40	35-55	15-25	$\mu A/V^2$
Body-effect (gamma factor)	γ	0.4 - 0.7		02-0.3	0.6-0.8	$V^{1/2}$
Channel-length modulation	λ	10^{-1} - 10^{-3}		10^{-1} - 10^{-3}		V^{-1}
Doping concentration	N_{SUB}	10^{15} - 10^{16}		10^{15} - 10^{16}		cm^{-3}
Gate-oxide thickness	t_{ox}	180 - 220		380 - 420		A°
Gate capacitance	C_g	1.3 - 1.5		0.7 - 0.9		$fF/\mu m^2$
Poly capacitance (* linear cap.; ** poly1-poly2 cap.)	C_p	1.0 - 1.2 *		0.4 - 0.5 **		$fF/\mu m^2$

1.4.3 Comments on Appendix A

Although the reader is assumed to be familiar with MOS physics and CMOS circuit modeling (as well as with the main aspects governing the construction of VLSI systems), a brief review is presented in Appendix A. This, however, is solely intended as a review on the physical parameters that appear more frequently in this dissertation. For more detailed discussions one may refer, for example, to [Sze 81][Allen 87][Tsividis 87][Mead 89][Weste 93]. For the particular case of charge-coupled device (CCD) technology, the basic concepts are reviewed in Chapter 3.

A final remark refers to the design parameter k adopted in all our derivations. k represents the *ratio of transistor geometry ratios.* The reason for adopting it is that it allows the final results to be expressed in low-entropy form, as desired, thus making the physical interpretations simpler and more objective. (A more complete justification is presented in Sec. A.7 of Appendix A.)

Chapter 2

MOS Winner-Take-All Function Implementations

Many are the algorithms in parallel signal processing that require the identification of a specific rank element among a given collection of variables. The most common case corresponds to identifying the largest (or smallest) element in the set, a function also known as the *winner-take-all* (WTA) function. Application examples of this function are abundant in the fields of data compression and pattern recognition, among many others. For instance, anytime multivector comparisons are carried out in order to find an approximate representation, a WTA circuit is needed. High-resolution implementations of the WTA function are usually non-linear and non-stable (or, using an analogy to multivibrators, monostable, meaning that a preset signal is needed and that the system runs to a monostable state as soon as this signal is released). In this chapter we discuss a series of models and VLSI realizations of this function, which are always accompanied by experimental results. We also present some application examples, in which we include a generic rank function implementation as well.

2.1 Introduction

Rank functions are fundamental building blocks to many parallel signal processing systems, no matter whether the implementation is in analog or digital form. However, we are mainly interested in fully-parallel alternatives (which are inevitably analog), for they present minimum time complexity; additionally, analog implementations usually demand significantly less silicon area, allowing larger arrays to be built on a single chip.

The rank detection function can be formally described as follows (Sec. 1.2). Given a real-valued scalar set $X=\{x_1, x_2, ..., x_N\}$, and assuming that $x_1 \leq x_2 \leq ... \leq x_N$, the *rank k* element of X is defined as

$$y=x_k \ , \tag{2.1}$$

that is, it is the kth smallest component of X. Three particular cases of the rank function are of special interest: (a) $k=N$, in which case the maximum is detected; (b) $k=1$, in which case the minimum is detected; and (c) $k=(N+1)/2$ (N odd), which corresponds to the median of X. Cases (a)-(b), also known as the *winner-take-all* function, are by far the most common, being one example shown in Fig. 1.2. Case (c), on the other hand, finds applications in fields like image processing, where median filters are employed to reduce impulsive noise (Sec. 1.2). As we show later, (a)-(b) can be used to implement *any* rank function, therefore including (c).

In this chapter we present a detailed discussion on CMOS/VLSI realizations of the WTA function. We start by showing, in Sec. 2.2, that many hardware realizations of this function share a common ground; specifically, we investigate the inhibitory mechanism implicitly present in this kind of single-cycle multidimensional system, and verify that it is often implemented by means of a voltage-follower circuit. This discussion will lead to a systematic process for analyzing the

performance of any WTA system that makes use of this type of intercell communications channel. After this study on system similarities is concluded, we present, in Sec. 2.3, a design-oriented analysis of one particular implementation, which not only presents high resolution, but also employs a global-feedback circuitry that will allow a connection between implementations with and without the kind of inhibitory channel mentioned above. Experimental results for either case are also presented, with special emphasis on system resolution. We conclude the chapter with two application examples of WTA circuits, which are described in Sec. 2.4.

2.2 Inhibitory-Mechanism Analysis of MOS Winner-Take-All Networks

In order for a system to be able to identify one among a series of N input signals (candidates) in just one clock cycle, it is necessary that all the N computing cells be interconnected thorough some kind of communications channel. We verify that, in VLSI realizations of the WTA function, this channel is almost invariably composed of a conventional voltage-follower (VF) circuit [Pedroni 94a] (see Figs. 2.1 and 2.3).

According to the number of interconnects, WTA circuits can be classified as complexity $O(N)$ or complexity $O(N^2)$, though VF circuits of length $O(N)$ are normally used in either case for the transmission of the inhibitory signal over the network. Basic VF circuits present low gain, however, and therefore low resolution, what has caused several modifications to been proposed in order to enhance the system performance. These modifications yet only contribute with additional local positive feedback in $O(N)$ systems, being the global feedback still transmitted by an underlying VF circuit. We start our analysis by examining, in Sec. 2.2.1, the basic VF itself, for it is, in fact, the simplest form of $O(N)$ WTA network. This will permit to establish some comparison parameters that are helpful in better understanding the potentialities of the modified configurations, and in deriving a general method for performance enhancement analysis as well, as shown in Sec.

2.2.2. The discussions are further illustrated by means of experimental results, which are presented in Sec. 2.2.3.

2.2.1 Voltage-Follower Circuit

Fig. 2.1 shows a basic nMOS VF circuit, which is constructed by simply connecting N conventional source followers in parallel, having inputs $V_1, V_2, ..., V_N$, output V_O, and a common bias voltage V_B. In what follows we consider the transistors operating initially in saturation, though the conclusions are extendible to the subthreshold regime; then, $I_i = (\beta_i/2)(V_{GS_i} - V_T)^2$, where $\beta_i = (\mu C_{OX} W/L)_i$ is the conductance factor of transistor i and V_T is the threshold voltage (Appendix A). If we consider the network initially in the condition $V_1 = V_2 = ... = V_N \equiv V_I$, it is easy to verify that

$$V_O = V_I - V_T - \sqrt{k}(V_B - V_T) = V_I - \text{constant}, \tag{2.2}$$

where the design parameter k represents the transistor parameter ratio $k = \beta_1/\beta_2$. From the viewpoint of the WTA operation, the worst case corresponds to having only one of the input signals distinct from the others. Doing so, that is, slightly increasing one of the inputs (say V_1) while keeping all the others (call them V_2) fixed and equal, we obtain that the output voltage given in (2.2) increases to

$$V_O = \bar{V}_I - V_T - \sqrt{k(V_B - V_T)^2 - \frac{N-1}{N^2}(V_1 - V_2)^2}\ , \tag{2.3}$$

where $\bar{V}_I = \sum_{i=1}^{N} V_i / N$ represents the arithmetic mean of the input values.

The behavior of equation (2.3) is illustrated in Fig. 2.2. As can be seen, if the differential voltage

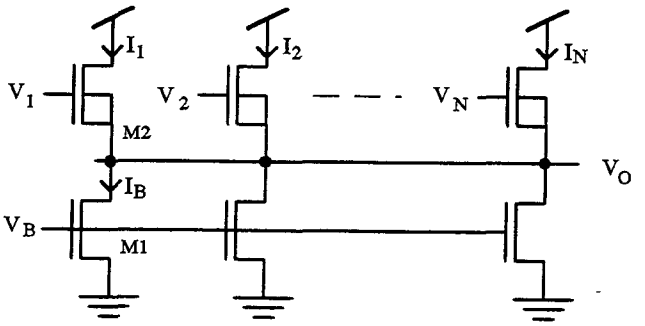


Fig. 2.1: N-cell nMOS voltage-follower circuit. In the WTA region, $V_O = \max\{V_1, V_2, ..., V_N\}$-constant.

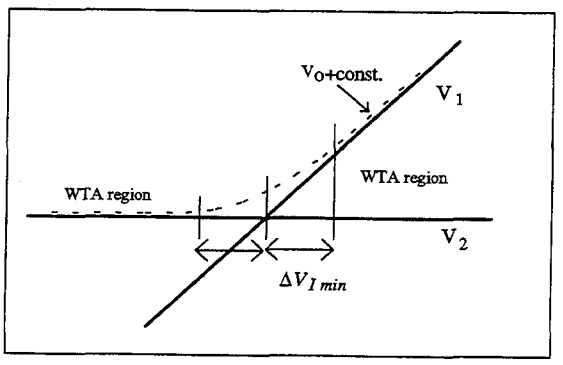


Fig. 2.2: Output voltage of the VF of Fig. 1 (for N=2). The circuit operates in the WTA region when $|V_1 - V_2| \geq \Delta V_{I\,\min}$.

$\Delta V_I = V_1 - V_2$ is small, V_O is proportional to $\bar{V}_I$, in which case the circuit behaves approximately as an *adder*. If, on the other hand, ΔV_I is large, it may cause the output to be $V_O \geq V_2 - V_T$, in which case all upper transistors in Fig. 2.1 will tend to cutoff, except that of cell 1 (the winner); this causes the circuit to operate in the *WTA region*, for then the output follows the highest input linearly and exclusively.

The WTA region: Ideally, V_O should present a sharp corner in Fig. 2.2 (i.e., $V_O \propto \max\{V_1, V_2\}$), what is not possible, for obvious reasons. We are then interested in finding how "wide" the round part of Fig. 2.2 is for the circuit of Fig. 2.1, when operating under worst-case conditions (i.e., $V_1 > V_2 = V_3 = \ldots = V_N \equiv V_2$). We define the *WTA region* as that in which $V_O \geq V_2 - V_T$, for then the output of the circuit follows the winning input linearly and exclusively, as discussed above (i.e., $V_O = V_1 - V_T - \sqrt{Nk}(V_B - V_T) = V_1 - \text{constant}$). So, applying $V_O = V_2 - V_T$ to (2.3), we obtain the desired system resolution in terms of the minimum differential input voltage $\Delta V_I = |V_1 - V_2|$ needed for the circuit to operate in this region, that is,

$$\Delta V_{I\,\min} = \sqrt{Nk}(V_B - V_T). \tag{2.4}$$

The first conclusion that can be drawn from this result is that the smaller the bias current (i.e., k and/or V_B) the smaller the differential voltage needed for the network to operate in the WTA region, hence higher resolutions are expected as the circuit moves toward subthreshold. If, however, strong inversion is required, as is often the case in high-speed systems (the high-frequency poles of the network grow with I_B), there are no parameters left in (2.4) to overcome the system's slow roll-off. Also another cause to contribute to the slow roll-off is that the limiting voltage $V_O = V_2 - V_T$ considered above, though providing small-error agreement between input and output, does not provide absolute zero current, for it does not account for the subthreshold region, which would require $V_O = V_2$. Therefore, minimum differential input voltages in the hundreds of

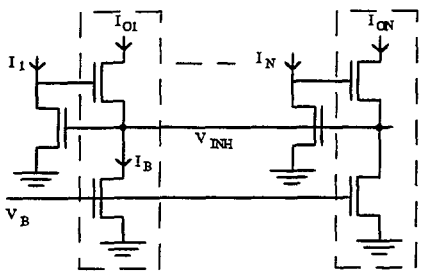
milivolts are typically necessary for this kind of network to work as a WTA. This result (Eq. 2.4) is verified experimentally in Sec. 2.2.3.

2.2.2 Modified Voltage-Follower Circuits

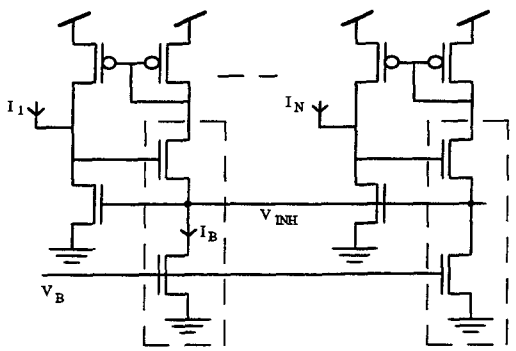
Fig. 2.3 shows four distinct MOS implementations of WTA circuits. Network (a) was introduced in [Lazzaro 89][Andreou 89] (operating in subthreshold), (b) in [Starzyk 93], (c) in [Cauwenberghs 94], and (d) in [Pedroni 94b]. The first two operate in current mode, with the winning cell sinking all the output current ($=NI_B$) and all the remaining cells shut off when in the WTA region; the other two circuits operate in voltage mode, with the winning output at high voltage and all the others at low voltage. Some circuit details were omitted in (c)-(d), for they are not relevant to the present discussion. Notice also that, though the positive feedback mechanism is $O(N^2)$ in (d), the network utilizes the same $O(N)$ VF circuit used in (a)-(c). This is, indeed, the main point to be noticed about these networks: they all employ an underlying VF circuit (dashed lines in Fig. 2.3) for the intercell transmission of the inhibitory signal (here denoted by V_{INH}). The basic effect of the modifications introduced onto this circuit is that they provide local positive feedback, thus increasing the system resolution.

The principle of operation of this kind of network can be verified with the help of Fig. 2.4, which, except for transistors M41 and M42, is a copy of Fig. 2.3(a) (with N=2). If the input currents are equal, that is, $I_1 = I_2$, so obviously are the output currents, $I_{O1} = I_{O2}$ $(= I_B)$. If now one of the inputs (say I_1) grows, the gate voltage of M31 must grow too. However, this voltage, V_{INH}, is common to all cells, what would cause the current driven by M32 to increase. Input I_2 has not changed, however, what forces M32 to switch to triode mode. Then, $V_2' < V_{INH} - V_T$ and, consequently, $V_{GS22} < 0\text{V}$, causing M22 to be completely cut off. In this case, the winning cell will drive all of the output current ($I_{O1} = NI_B$). Notice that the global inhibitory mechanism is still the same as in Sec. 2.2.1, with V_{INH} acting on the source of the VF transistors, but now it also

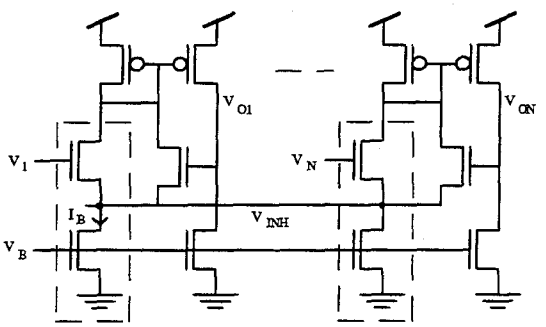
(a)



(b)



(c)



(d)

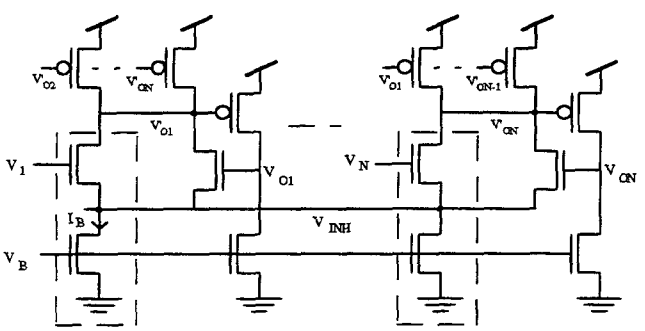


Fig. 2.3: Modified VF circuits: (a)-(b) operate in current mode, whereas (c)-(d) operate in voltage mode. In all of the cases, the inhibitory signal, V_{INH}, is transmitted by a nested VF network (dashed lines).

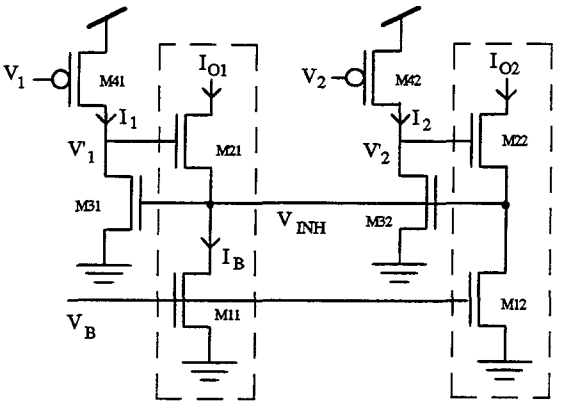


Fig. 2.4: A voltage-coupled version of Fig. 2.3(a) (for N=2).

activates local feedback loops (V_1', V_2' in the example above), which operate on the transistor gates as well. As a result, smaller variations of V_{INH} are now sufficient for the circuit to operate in WTA mode.

Resolution gain: The networks of Fig. 2.3(a)-(c) can be represented as in Fig. 2.5, with A denoting the gain of an input stage, which is placed between the actual inputs and the VF circuit and is controlled by V_{INH}. A then summarizes the (generic) effects of the modifications introduced onto the basic VF configuration. However, looking into the circuit from the $V_1', \ldots, V_N'$ terminals in Fig. 2.5, the problem is exactly the same as in Sec. 2.2.1, that is, these voltages must be at least the amount given by (2.4) apart for the network to operate as a WTA. It is obvious then that A represents the *resolution gain*, which, applied to (2.4), gives

$$\Delta V_{I\,\min} = \sqrt{Nk}\,(V_B - V_T)/A. \tag{2.5}$$

Still another way of appreciating the effect of A in Fig. 2.5 is by observing that, even if a certain input (say V_i) stays constant while another grows, the corresponding signal (V_i') decreases, due to

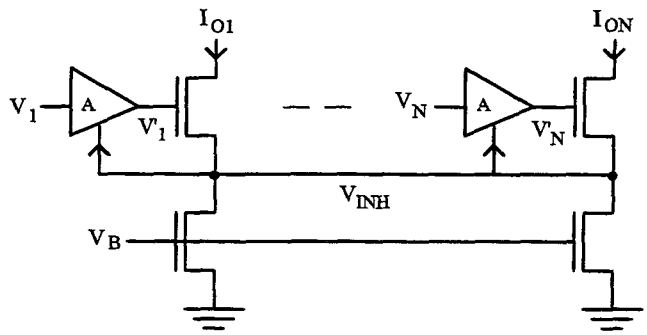


Fig. 2.5: A generalized representation of the modified VF networks. *'A'* represents the global effect (*resolution gain*) caused by the positive feedback loop introduced onto the basic VF circuit.

the action of V_{INH}. This additional feedback causes the required minimum differential input voltage to be smaller.

Gain factors in the one to two orders of magnitude range are typical. Let us consider, for example, the case of Fig. 2.4. We want to derive an expression for $A = V_i' / V_i$. This expression, however, must be valid in the central (non-linear) part of Fig. 2.2. But, since the effect of A is reciprocal, as described above, we can simply take the gain of a single cell in the linear region, which is approximately the same as considering both (or all) cells at once when operating in the non-linear region (the effect is symmetric). The solution to the problem then becomes straightforward. It is easy to verify that the gain in Fig. 2.4 is

$$|A| \approx g_{m4} . r_{d4} \,//\, r_{d3} \approx \frac{2}{(\lambda_3 + \lambda_4)(V_{DD} - V_T - V_I)} \quad , \tag{2.6}$$

where $g_{m_j} = \sqrt{2\beta_j I_j}$, $r_{d_j} = 1/\lambda_j I_j$, and V_I is the worst-case input voltage. Therefore, by using long transistors (low λ), high values of A can be achieved. For example, if V_{DD}=5V, V_T=1V, V_I=2V, and $\lambda_3 = \lambda_4$=0.02, then (2.6) gives $A = 50$. The obvious consequence of these modifications is that

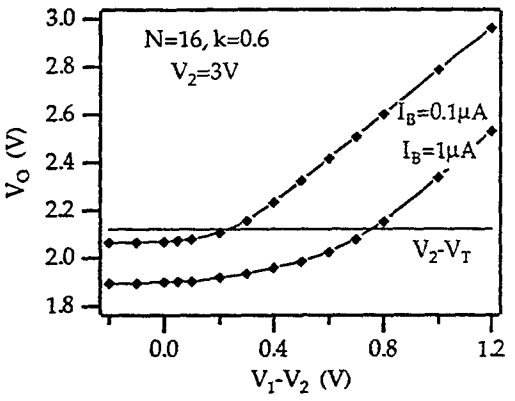
voltage differences of typically just a few milivolts can now be detected. It is important to notice, however, that though this kind of architecture makes possible to attain high-resolution systems, this is not always the case; for instance, [Johnson 91] uses a VF network, but the incorporated circuitry has a relatively low A.

2.2.3 Experimental Results

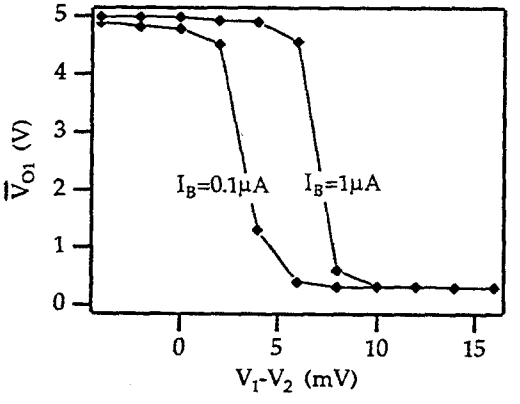
Fig. 2.6 shows measurements realized on an N=16 modified VF circuit similar to that of Fig. 2.3(d), constructed on a standard 2.0μm CMOS chip. The nested VF transistors were $(W/L)_1$=6/6 and $(W/L)_2$=10/6 (k=0.6), and the measured V_T of the bias transistor was approximately 0.88V. Two sets of measurements are presented, for $I_B = 0.1\mu A$ (V_B=0.96V) and 1μA (1.14V). Initially, the local feedback loop was kept disabled, so the circuit operated as a basic VF (Fig. 2.1). Only one input (V_1) was allowed to vary, while all the others were kept constant and equal (V_2=3V). The measurements are plotted in Fig. 2.6(a). As discussed earlier, the circuit operates in the WTA region when $V_O \geq V_2 - V_T$, that is, when V_O is above the horizontal line in Fig. 2.6(a). The estimated values for $\Delta V_{I\ \min}$ (Eq. 2.4) are 0.25V and 0.80V, respectively, therefore in good agreement with the experimental results (intersection points in Fig. 2.6(a)) and quite large, as predicted. Next, the local excitation was enabled, resulting the behavior depicted in Fig. 2.6(b). As expected, a large reduction of $\Delta V_{I\ \min}$ was achieved, resulting a resolution under 10mV in both cases.

2.3 Global-Feedback WTA Implementation

In the section above we have verified that there is an important similarity between VLSI implementations of the winner-take-all function: they often rely on a voltage-follower (VF) circuit for intercell communication. We have also derived a general method for analyzing what the consequences are when circuit modifications are introduced onto a basic VF network. In this



(a)



(b)

Fig. 2.6: Experimental results from an N=16 WTA circuit similar to that of Fig. 2.3(d). In (a), the local feedback was disabled, in which case the predicted values of $\Delta V_{I\ \min}$ (Eq. 2.4) are 0.25V and 0.80V (for I_B=0.1μA and 1μA, respectively). The resolution is enormously increased when the local excitation is enabled (b), being both voltages reduced to under 10mV.

section we investigate one of these circuits. More specifically, we investigate the network of Fig. 2.3(d) [Pedroni 94b], whose interconnect architecture is responsible not only for local, but also for global additional feedback. This arrangement provides a high value for A and a sensitivity to transistor mismatches that is $\sqrt{N}$ smaller than that of $O(N)$ local feedback systems with same transistor sizes. The network also allows an additional experiment regarding system resolution, that is, due to its globally interconnected synapses, the circuit can operate without the nested VF circuit. Although this is obviously not the way the circuit is used regularly, it will serve to once again illustrate the performance degradation that occurs when A is decreased. Experimental results are also presented and compared to the analytical predictions.

2.3.1 Circuit Analysis and Design

Fig. 2.7 shows the complete schematics of the global feedback WTA circuit [Pedroni 94b]. As can be seen, the synaptic matrix (left-hand side) reminds that of a Hopfield network [Hopfield 82], with just one transistor (M3) per synapse, while the output circuit (right-hand side) is composed of a bias transistor (M1), an input transistor (M2), an output amplifier (M4-M5) and a local positive feedback transistor (M6). A preset transistor (M7) is also shown. The signals $V_1, V_2, \ldots, V_N$ are the inputs to the circuit, $V_{O1}, V_{O2}, \ldots, V_{ON}$ are the interconnect outputs, and $\bar{V}_{O1}, \bar{V}_{O2}, \ldots, \bar{V}_{ON}$ are the (digital) outputs of the system.

The basic operation of the network can be summarized as follows. If we suppose initially that $V_1 = V_2 = \ldots = V_N$, then $V_{O1} = V_{O2} = \ldots = V_{ON}$ (theoretically only, due to transistor mismatches). If we let now one of the inputs, say V_1, increase, V_{O1} has to decrease in order to bring M2 of row 1 into linear mode, needed to keep I_1 constant ($V_{O2}, \ldots, V_{ON}$ initially unchanged), so increasing the current driven by the leftmost p-transistor of rows 2 to N. However, the total current must remain the same, what forces the gate voltages of all the other p-transistors in rows 2 to N to increase, thus

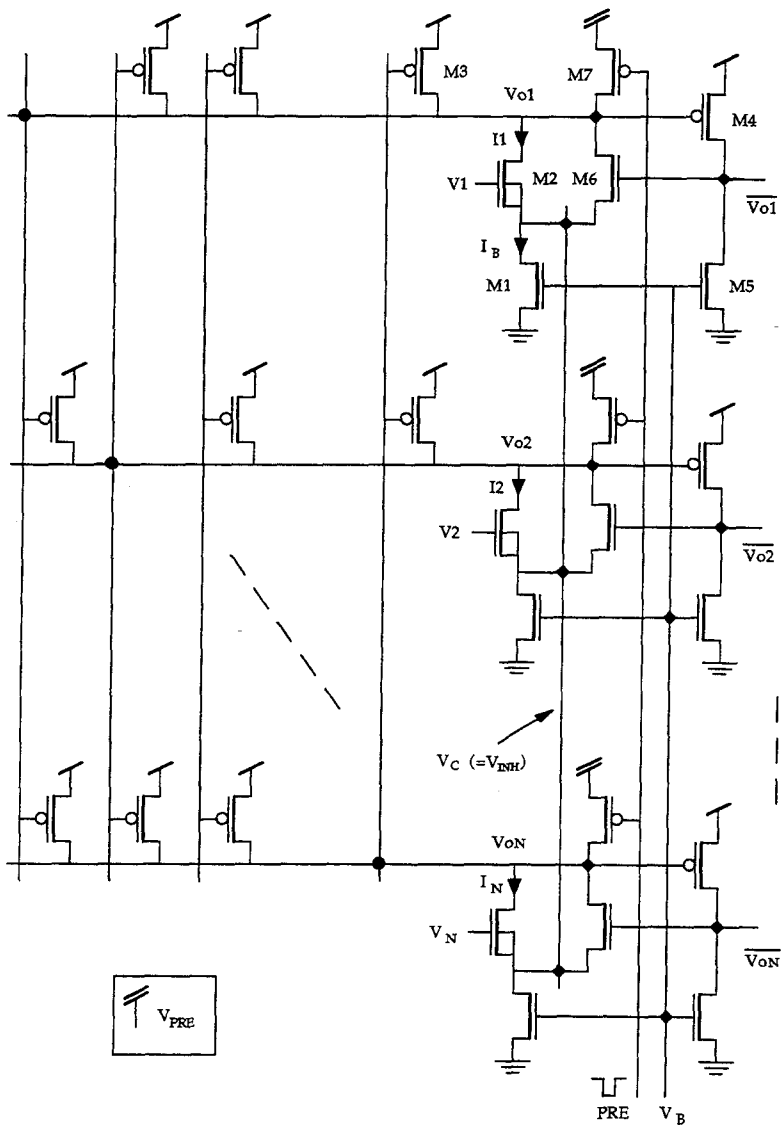


Fig. 2.7: Global-feedback WTA network.

making the total current driven by the p-transistors of cell 1 even smaller and forcing V_{O1} deeper down. Depending on the geometry of the transistors and on the applied voltages, it might happen that V_{O1} becomes low enough to make the other outputs greater than $V_{DD} - V_T$, in which case all p-transistors of row 1 will tend to cutoff, producing an output vector that can be directly converted to a digital representation. If the system is designed with this (desirable) monostable behavior, a preset mechanism (M7) is necessary in order to set the system's initial condition. The time response of the system is then governed by the time needed for V_1 to discharge node V_{O1}, which is initially charged with V_{PRE}. Since the total bias current is constant, so is the discharge rate. An optional local feedback loop (M6) can be employed in order to accelerate this process (by a factor of ~3-5), though with small effect on the digital outputs. It is clear then that, due to time response constraints, operation in strong inversion is normally preferred, establishing a straightforward trade-off between power consumption and speed.

Quantitatively, the circuit behaves as follows. If $V_1 = V_2 = \ldots = V_N \equiv V_I$, then $V_{O1} = V_{O2} = \ldots = V_{ON} \equiv V_{O=}$ (theoretically) and $V_C \equiv V_{C=}$. With transistors in saturation and $\beta_j = (\mu C_{ox} W / L)_j$, we obtain that

$$V_{C=} = V_I - V_T - k_1(V_B - V_T) \quad \text{and} \tag{2.7a}$$

$$V_{O=} = V_{DD} - V_T - k_2(V_B - V_T), \tag{2.7b}$$

where $k_1 = \sqrt{\beta_1 / \beta_2}$, $k_2 = \sqrt{\beta_1 / (N-1)\beta_3}$, and, for simplicity, $V_{Tn} \approx V_{Tp} \equiv V_T$. As expected, the voltage on the common line (Eq. 2.7a) is linearly dependent on the input voltages, whereas the output voltage (Eq. 2.7b) is not affected by them when all the input voltages are equal. This behavior is illustrated in Fig. 2.8(a) for a network of size N=16, where the transistor parameter ratio k_1 is kept fixed while k_2 varies. The plots indicate that the discrimination between the winning output and the others is expected to be more difficult as k_2 grows (we will demonstrate this later).

From (2.7) we verify that, for the saturation condition (i.e., $V_{C=} \geq V_B - V_T$ and $V_{O=} \geq V_I - V_T$) to be satisfied for any input level, the transistor ratios must obey

$$k_1 \leq \frac{V_{i\min} - V_B}{V_B - V_T} \quad \text{and} \tag{2.8a}$$

$$k_2 \leq \frac{V_{DD} - V_{i\max}}{V_B - V_T}, \tag{2.8b}$$

therefore establishing the following input dynamic range,

$$V_B + k_1(V_B - V_T) \leq V_i \leq V_{DD} - k_2(V_B - V_T). \tag{2.9}$$

If we now let one of the inputs (say V_1) grow, with all the others still alike (worst case), V_{O1} decreases toward $V_C \approx V_{C=}$, whereas $V_{O2} = V_{O3} = \ldots = V_{ON}$ increase, i.e.,

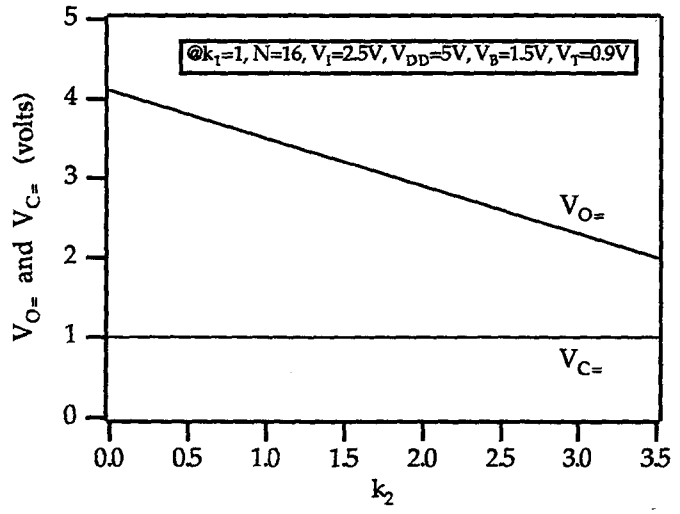
$$V_{O1} \approx V_{C=} \quad \text{and} \tag{2.10a}$$

$$V_{O2} \approx \frac{(N-2)(V_{DD} - V_T) + k_1(V_B - V_T) - V_2}{N-3} -$$

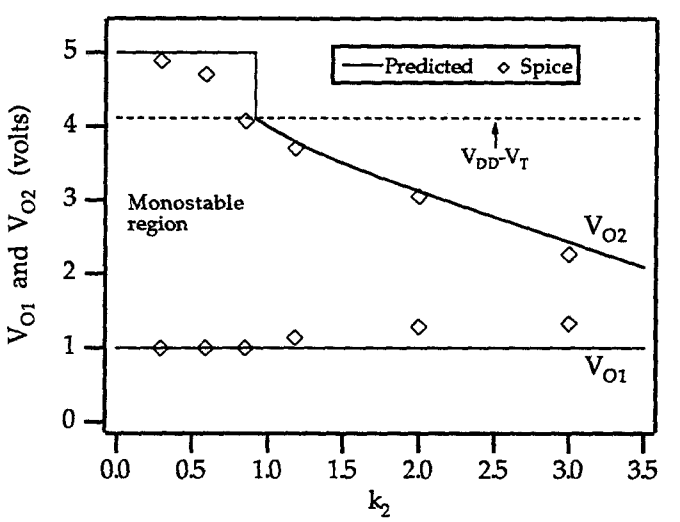
$$\sqrt{\left(\frac{V_{DD} - (N-2)V_T + k_1(V_B - V_T) - V_2}{N-3}\right)^2 - \frac{N-1}{N-3}k_2^2(V_B - V_T)^2 - \frac{N-2}{N-3}V_T^2}\ . \tag{2.10b}$$

As mentioned earlier, if the transistor parameter ratios are small enough, they may cause V_{O2} to be greater than $V_{DD} - V_T$. From (2.10b) we verify that this situation exists if

$$(N-1)k_2^2 = \frac{\beta_1}{\beta_3} < \frac{(2V_{DD} - 2V_C - 3V_T)V_T}{(V_B - V_T)^2}, \tag{2.11}$$



(a)



(b)

Fig. 2.8: (a) Common voltage $V_{C=}$ and output voltage $V_{O=}$ for the network of Fig. 2.7, for N=16 and $V_1 = V_2 = \ldots = V_N$. (b) Output voltages of the winning (V_{O1}) and loosing (V_{O2}) outputs when $V_1 > V_2 = \ldots = V_N \equiv V_2$. In this case, full enlargement is achieved when k_2<0.94.

which is independent from N, as expected, since in this state each active cell has only one active transistor load. This behavior is illustrated in Fig. 2.8(b), which shows both the theoretical predictions given by (2.10)-2(11) and SPICE simulations for a circuit of N=16 cells. As can be seen, the output enlargement is immediately achieved when the network is designed with parameters such that (2.11) is fulfilled (i.e., $k_2 < 0.94$ in this case).

We return now to the time response. Although the total bias current NI_B is constant, only a fraction δI_B $(1 < \delta < N)$ is destined to discharging the winning node (say V_{O1}), the magnitude of δ depending on how far V_1 is from the other input voltages. The settling time of the system is therefore given by

$$T = \frac{C\Delta V}{\delta I_B}, \tag{2.12}$$

where C is the capacitance of the output node (~$(N\text{-}1)(\text{WL})_3$ x0.5fF/μm^2, Table 1.1) and ΔV (~V_T) is the difference between V_{PRE} (V_{DD}) and the transition voltage of the output inverter (~$V_{DD} - V_T$). A typical time response is illustrated in Fig. 2.9 for practical lower and upper bounds of δ. Equation (2.12), together with (2.5), permit to formalize the trade-off mentioned earlier between resolution, time response, and power consumption, where we observe that while the time performance grows with power consumption (2.12), the resolution tends to decrease (2.5).

2.3.2 Experimental Results

The performance of the network was measured on an N=32 WTA circuit fabricated on a 2μm CMOS chip (shown in Fig. 2.10), with L=10μm, $k_1 = 0.7$, and $k_2 = 0.4$. The experiments agreed consistently with the predictions, showing always only one winner present at a time, as imposed by

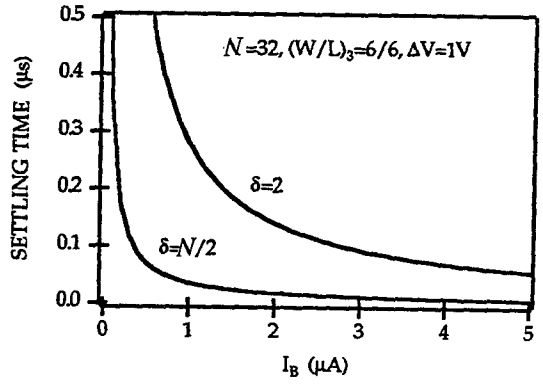


Fig. 2.9: Settling time versus power consumption of circuit of Fig. 2.7.

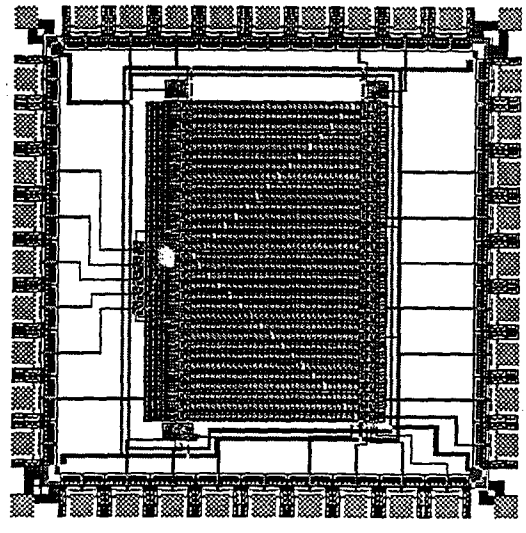


Fig. 2.10: N=32 2μm CMOS implementation of WTA network of Fig. 2.7.

the equilibrium conditions of the circuit. A wide system resolution was obtained (~50 dB), as expected, with a sensitivity better than 10mV in the worst case. Two separate gain measurements are shown in Fig. 2.11(a), where the high resolution of the system is apparent. The tests were performed with $V_B = 1.0\,\mathrm{V}$ and V_{REF}(all inputs but one)=2.0V. An oscilloscope picture is also presented (Fig. 2.11(b)), showing the dynamic (monostable) behavior of the winning and losing outputs. Notice in Fig. 2.11(b) that all the outputs are high during preset (PRE=0V), with the winning output going to low voltage when PRE is released. The time required for the transition in this case ($I_B = 0.2\mu\mathrm{A}$) is approximately 3μs, therefore in good agreement with (2.12). Notice also that this time can be drastically reduced by using either higher I_B or smaller synaptic transistors.

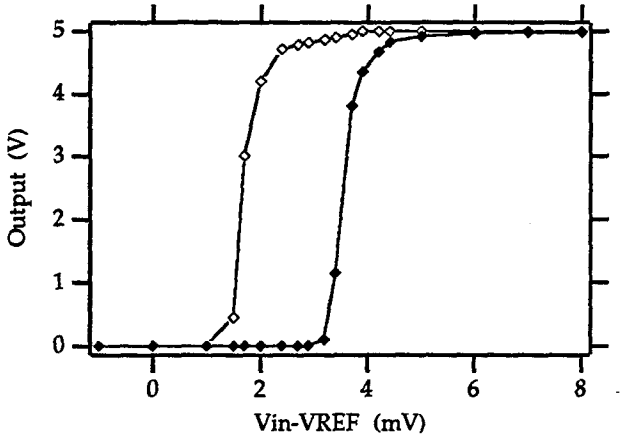
2.3.3 Moderate-Resolution Architectures

As mentioned before, the network of Fig. 2.7 can operate *without* the underlying VF circuit, for the added feedback is global in this case. The VF sub-circuit can be disabled by simply connecting the V_C signal to GND in Fig. 2.7 (and removing M1, M6). The consequence of this is that the total current driven by the input transistors is no longer constant (contrary to Fig. 2.7 or any of the circuits of Fig. 2.3, where it was constant and equal to NI_B), therefore reducing the system gain and resolution.

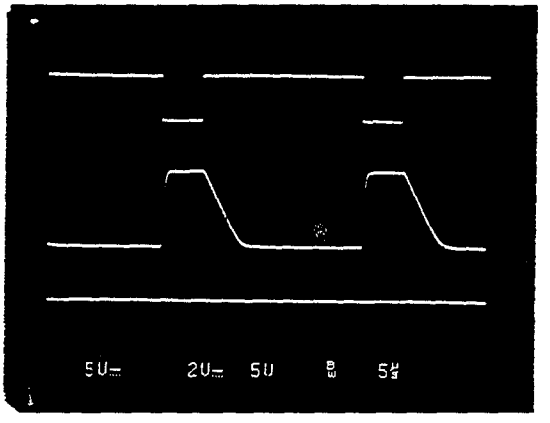
Inspecting the altered circuit configuration described above, we verify that the output voltage, for the condition of equal inputs, is now given by

$$V_{O=} = V_{DD} - V_T - k(V_I - V_T)\ , \qquad (2.13)$$

while the input dynamic range is determined by



(a)



(b)

Fig. 2.11: Test results from chip of Fig. 2.10: (a) System gain and (b) winning versus losing outputs, with the preset signal on the upper channel, the winning output in the middle, and one of the other outputs on the lower channel.

$$V_T \le V_i \le \frac{V_{DD} + kV_T}{1+k}\,, \qquad (2.14)$$

where $k = \sqrt{\beta_1 / (N-1)\beta_2}$. As can be seen in (2.13), the output voltage is now input-dependent, contrary to (2.7b).

Tests conducted on an N=32 MOS circuit of this type, designed with L=10μm and $k = 0.58$, and with applied input voltages according to (2.14), showed resolutions in the 30-60mV range, therefore poor, as expected, when compared to the original circuit. Consequently, for signal voltages that are typical to V_{DD} =5V implementations, the resolution of this (degenerate) network is expected to be not higher than 6 bits in general.

Other examples of WTA networks that do not employ VF circuits can be found in [Cilingiroglu 93][Yuping 93][Moon 94]. As far as the system resolution is concerned, the performance of the architecture adopted in [Cilingiroglu 93][Yuping 93] can be immediately evaluated based on the situation just described, whereas [Moon 94] relies on a neural configuration that has circuit parameter assumptions which are inherent causes of high offsets (besides requiring a wide silicon area for its implementation).

2.4 A Temporal Network

Despite the moderate resolution of the examples described in Sec. 2.3.3, high-performance analog WTA systems are not necessarily built onto voltage-follower circuits (though VF's are generally present in implementations that are of high resolution *and* of small size). Let us consider the circuit of Fig. 2.12 [Pedroni 92a,b] as an example. It consists of a *temporal* WTA network (i.e., it

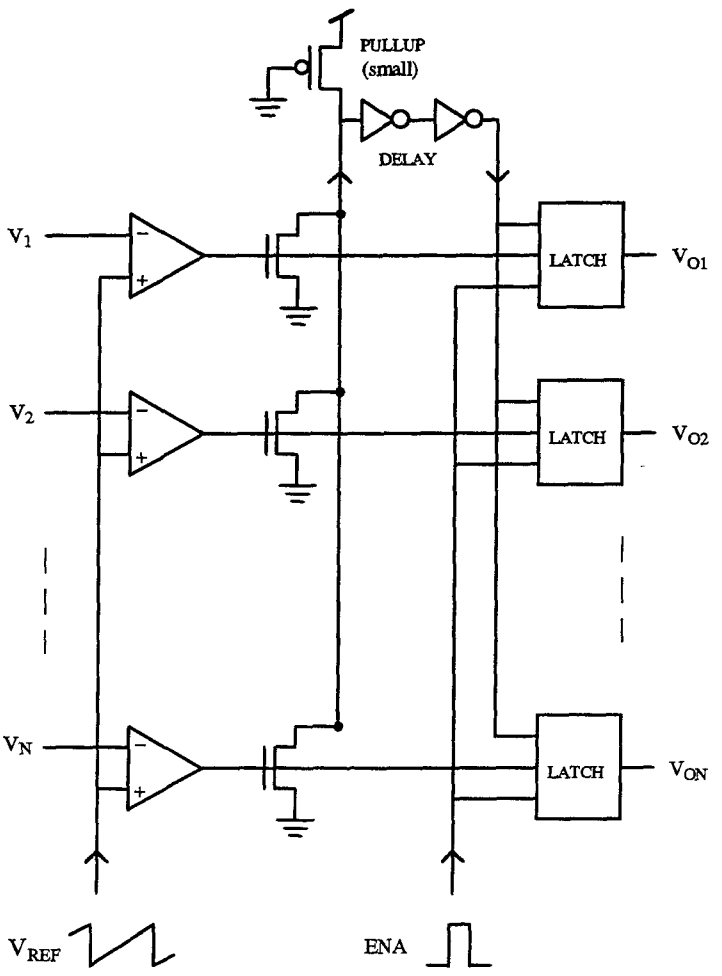


Fig. 2.12: No feedback, time-dependent *O(N)* WTA system.

detects the first input signal to reach - or to be reached by - the reference voltage V_{REF}). The circuit has no feedback, and the inhibitory mechanism consists simply of freezing the memory (latches) immediately after one of the voltage comparators switches to '1'. Due to the open architecture of this system, high performance can be achieved, depending basically only on how much one is willing to spend on the individual voltage comparator implementations. An N=16 network of this

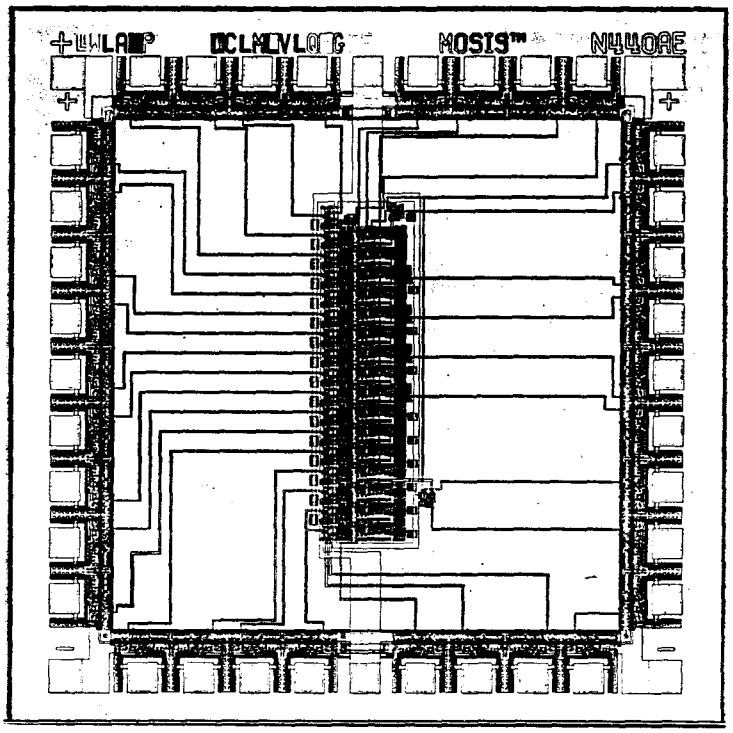
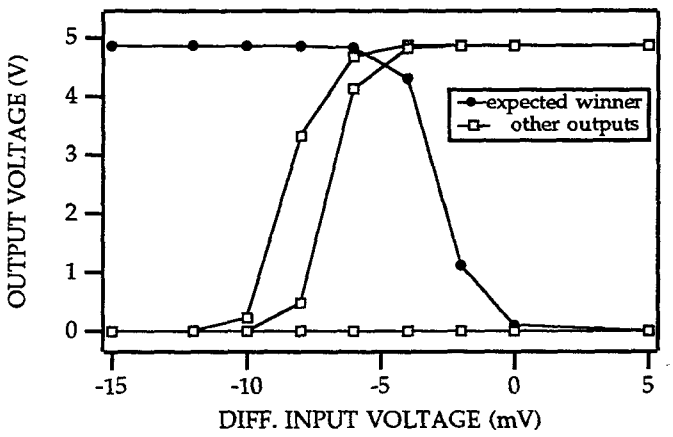
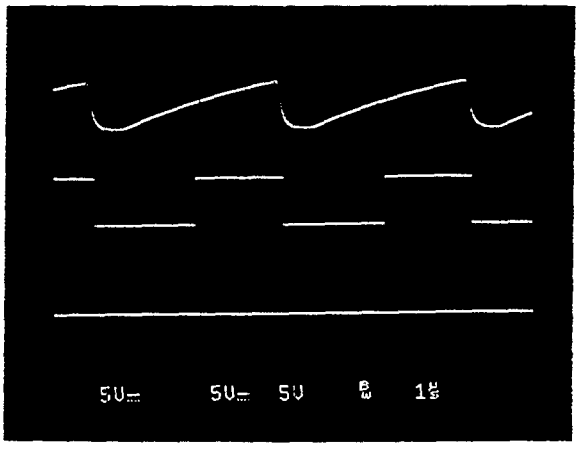


Fig. 2.13: N=16 1.2μm implementation of circuit of Fig. 2.12.

type was fabricated using 1.2μm CMOS technology (Fig. 2.13). Conventional 2-stage comparators [Allen 87] were used at the inputs, with a time-response of approximately 1MHz and measured offsets (for all inputs) in the -5mV/10mV range. The tests were realized with the time-dependent (ramp) signal applied to V_{REF}, as depicted in Fig. 2.12. All inputs were kept equal (worst case) and fixed at 3V, except one, which was allowed to vary. The varying input signal was applied to the input with highest offset. The measurements, realized with a ramp period of 4μs, are plotted in Fig. 2.14(a). As can be seen, the system correctly and uniquely decodes the varying signal when it is 12mV (or more) below the other inputs. An oscilloscope picture is also presented in Fig. 2.14(b), showing V_{REF} on the upper channel, the winning output in the middle, and one of the other outputs (the first to switch to high in Fig. 2.14(a)) on the lower channel.



(a)



(b)

Fig. 2.14: Test results from chip of Fig. 2.13: (a) System resolution and (b) oscilloscope picture showing V_{REF} (4μs) on the upper channel, the winning output in the middle, and one of the other outputs on the lower channel.

2.5 Application Examples

We conclude this chapter by presenting two application examples of WTA circuits. The first one consists of a conventional application, in which the circuit is connected to the output of a Hamming classifier in order to identify the highest among its output voltages (best matching). In the second example, a non-conventional application is described, in which case a pair of complementary WTA networks is used to implement a *generic* rank function.

2.5.1 Application Example One: Hamming Classifier

The WTA circuit of Fig. 2.7 was applied to the output of a 32x32 Hamming classifier (similar to that described in Chapter 3) in order to detect the winning output. All rows of the system processed the same random vector (being one of the samples shown in Fig. 2.15), except one row, which processed the vector shown on the right-hand side of that figure. Notice that the latter vector has only one bit distinct from the former, which circulates over the 8x4 block and is always in agreement with the corresponding input bit. During the tests, the circuit yielded a fixed winner, as expected, for the resolution of this system is well above 5 bits, as verified in Sec. 2.3.

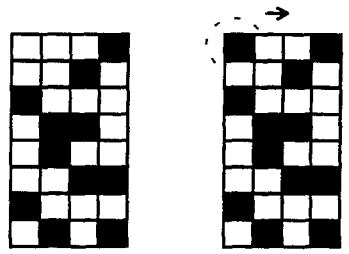


Fig. 2.15: Example of data vectors used in the Hamming classifier application.

2.5.2 Application Example Two: WTA-Based Rank Filter

Rank and median filters are members of a class of non-linear circuits known as *order statistic* filters. Their use for one- and two-dimensional signal processing applications has been subject to growing interest by virtue of their ability to remove impulsive noise [Wendt 86][Yli-Harja 91]. The vector elements (inputs to the filter) can be binary, integral (M-valued), or more generally real numbers. In the case of binary inputs, VLSI implementations are straightforward, but of little interest for practical applications. If, however, the inputs are integers (or reals), conventional VLSI implementations rely on threshold decompositions of the analog values into M-1 binary strings of length N, which are then binarily filtered and later added together [Wendt 86][Yli-Harja 91]. These implementations are necessarily hybrid and the digital operations involved cannot be performed completely in parallel, therefore demanding computation times that grow with filter parameters.

The VLSI implementation presented in this section is fully-parallel, therefore requiring only one clock cycle for a complete 1D real-domain rank computation, regardless of M or N. The circuit needs no A/DC or threshold decompositions. A simplified block diagram of the system is shown in Fig. 2.16. The only analog units are the winner-take-all circuits, which detect the greatest (detected by WTA) and the smallest (by $\overline{\text{WTA}}$) among the N analog input voltages. The control units (CTRL) are responsible for the positions of the input switches, which depend on the respective outputs of the two WTA circuits. If the upper switch of a pair of switches is ON, then that input is connected to the corresponding input signal (input active); if, on the other hand, the lower switch is the one that is ON, the input is connected to either $V_{\min}$ or $V_{\max}$, which are, respectively, the lower and upper bound of the WTA input dynamic range, implying that the input is not a candidate for the median (or rank k) selection (input blocked).

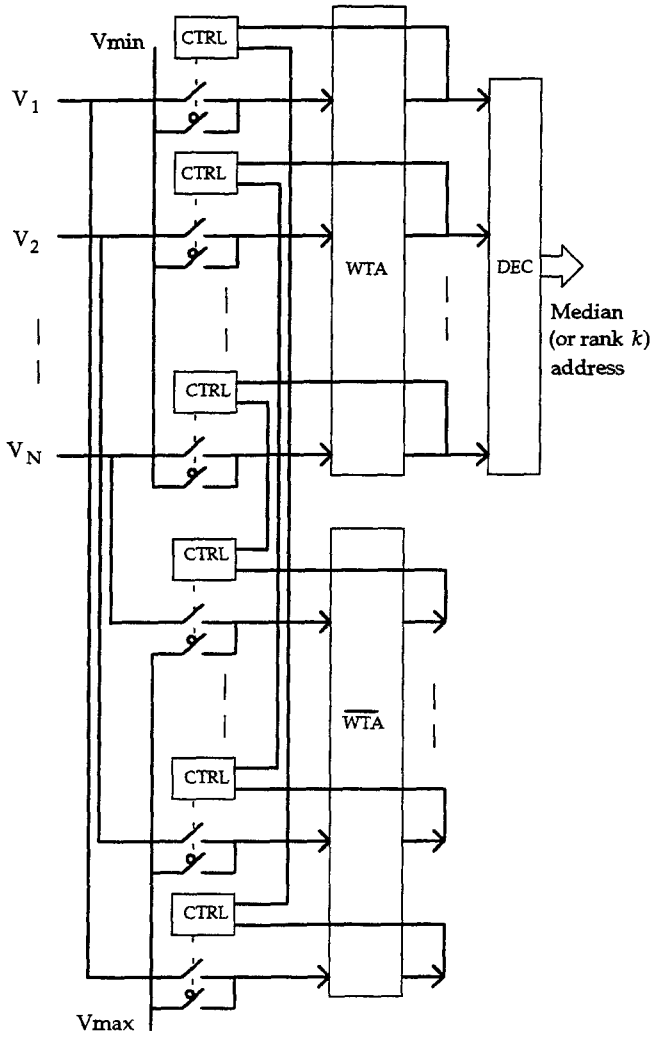


Fig. 2.16: Simplified system diagram of the WTA-based generic rank filter.

The reason for using WTA units to implement rank filters arises from the following observation. Say that the set $\mathbf{X} = \{x_1, x_2, \ldots, x_N\}$ of input elements (represented by $\mathbf{V} = \{V_1, V_2, \ldots, V_N\}$ in Fig. 2.16, though it must be noted that i refers to rank position in $\mathbf{X}$ whereas it refers to physical position in $\mathbf{V}$) lying within the filter window has been sorted in ascendant order, i.e., $x_1 \leq \ldots \leq x_k \leq \ldots \leq x_N$, and that the desired output, x_k, has been identified. If now we shift the window by one position, letting a new component, x_i, in, the new output will necessarily be in $\{x_i, x_{k-1}, x_k, x_{k+1}\}$, that is, besides the new input and the old output, the only candidates are x_k's

immediate neighbors. This suggests the use of WTA circuits for implementing the rank-order function, for they can select any rank element in the set by means of successive select-inhibit cycles, that is, if the first winning output, x_N, is inhibited, the next winner will be x_{N-1}, and so on until x_k is reached. This procedure, however, requires *N-k* cycles, for WTA circuits are essentially unidirectional. This limitation can be overcome with the use of two circuits, computing in opposite directions (Fig. 2.16), hence implementing a single-clock-cycle 1D filter.

The operation of the circuit of Fig. 2.16 can be summarized as follows. Say that the system has been initialized as described above, starting with all switches in the upper position (i.e., all CTRL='1'); the initialization is concluded after *N-k* cycles, with the first output selected, leaving the system switches blocked (lower position) for ranks $i > k$ and active (upper position) for $i \leq k$, being these switch positions then copied into the other set of switches ($\overline{\text{WTA}}$) in complementary form, that is, blocked for $i \leq k$ and active for $i > k$. From this point on, each new median (or rank *k*) computation requires only one cycle, which is performed in the following way. Say that the filter window is shifted, so letting a new element, x_i, in, which overwrites one of the previous components of the input vector. If i is a blocked input in WTA, this is released and the WTA circuit is activated just once, producing the desired selection; then the WTA switches are copied to $\overline{\text{WTA}}$ in complemented form. If, on the other hand, i is an active input in WTA, it is a blocked input in $\overline{\text{WTA}}$; then a similar procedure is applied to $\overline{\text{WTA}}$, that is, the input is released and $\overline{\text{WTA}}$ is clocked once, being the switch positions copied to WTA in complemented form, which then produces the desired output automatically.

Finally, we observe that the system application described above can be used for 2D applications as well, in which case *n* clock cycles are necessary for a complete rank computation on an $N=n\text{x}n$ pixel block, for every time the window is shifted *n* new elements enter the signal vector.

Chapter 3

CCD-Based Analog and Hamming Matrix Processors

Many signal processing algorithms can be mathematically described in terms of vector-matrix multiplication. Hardware implementations of vector-matrix multiplier (VMM) circuits are generally expensive, however, for they usually demand large silicon areas and high power consumption. Attempts to reduce these two factors, while keeping others (e.g., speed and accuracy) at acceptable levels, have been a constant challenge. In this search for alternatives that are more "economical" for a specific type of application, the use of charge-coupled device (CCD) technology has been considered, motivated by the fact that it allows the fabrication of very small, highly interconnected computing cells. Although these devices are better known for their applications to imaging systems than to computing, the discussions on their physical properties presented in this chapter are exclusively from a signal processing perspective. We also introduce a locally-controlled CCD (LCCD), which overcomes one of the main limitations of conventional devices, namely their poor charge-transfer efficiency, hence allowing the construction of larger computing arrays. Application examples of this technology to vector-matrix multipliers, Hamming networks, and threshold functions are also described, along with chip implementations and experimental results.

3.1 Introduction

One of the basic and most important features of neural processors, pattern classifiers and other analog parallel signal processing systems is their highly dense, heavily interconnected architecture. Due to their compact cell size and interconnecting simplicity, charge-coupled devices (CCD's) [Sequin 75][Beynon 80] have been considered for system implementations of this kind [Agranat 90][Chiang 91][Fossum 91][Neugebauer 92][Pedroni 92a,b]. Although this technology is better known for its applications to imagers [Amelio 71][Barbe 75] than to signal processors, the systems differ immensely in architecture and performance specifications. For instance, while in an imager the charge packets are generated by the incident light, suffering no processing and being destroyed upon frame reading, in a signal processor the charge is normally generated electronically, for this is faster and more controllable than optical methods [Neugebauer 92]; furthermore, the system must be equipped with some kind of computing mechanism, and the computations must be performed without destroying the information stored on the chip.

Since in a CCD the information is encoded in the form of charge packets, it is fundamental that each individual packet, representing one of the matrix nodes, be preserved as accurately as possible during either the loading/refreshing the matrix phase or the computing phase. In conventional CCD's, matrix loading is accomplished by injecting each charge packet at one end of a row of electrodes and then shifting it along that row to its storage address. This process, however, represents a major limitation for the use of CCD's in large arrays, for the charge transfer efficiency is inevitably lower than unity, meaning that some charge is always left behind after each transfer, adding to the trailing packets. To overcome this problem, a locally-controlled CCD (LCCD) cell [Pedroni 94d,e] is introduced, which is capable of generating, handling, and dumping charge locally.

We start the analysis by reviewing, in Sec. 3.2, the basic principles of operation of charge-coupled devices, and by investigating the basic types of signal processing functions that can be directly implemented with this technology. In Sec. 3.3, the LCCD cells are described, along with discussions on their properties, applications, and circuit sizes. Finally, in Sec. 3.4, a series of experimental results and additional technical discussions are presented. Part of the work reported here (on conventional CCD's) was developed in collaboration with C. Neugebauer, whose dissertation [Neugebauer 93] contains a detailed mathematical description of the physical properties of these devices. (Additional information can also be found in all the other references mentioned along this chapter.) Due to this fact, a general mathematical modeling will be omitted here. Rather, we concentrate our efforts on specific aspects which have proven to be most limiting in our previous experiences (e.g., charge transfer inefficiency). Besides introducing a new CCD architecture, we also describe new applications for this technology and conclude with an extensive series of experiments and discussions on its performance.

3.2 Conventional Signal-Processing CCD's

In a simple way, a CCD can be viewed as a MOS transistor, with the differences that the former has multiple gates and that charge is not readily available at its source terminal; rather, charge is injected in a controlled (switched or shifted) way, being conserved thereafter over a certain period of time. Probably the best way to describe the basic principle of operation of this kind of device is through potential well diagrams [Beynon 80], as illustrated in Fig. 3.1. The gates are constructed of overlapping poly1-poly2 electrodes. If a gate voltage is pulsed high, a potential well is created under that electrode, which remains empty if both the neighboring gates are closed (low voltage) and the gate voltage does not stay high too long (typically under 100ms [Beynon 80][Sze 81],

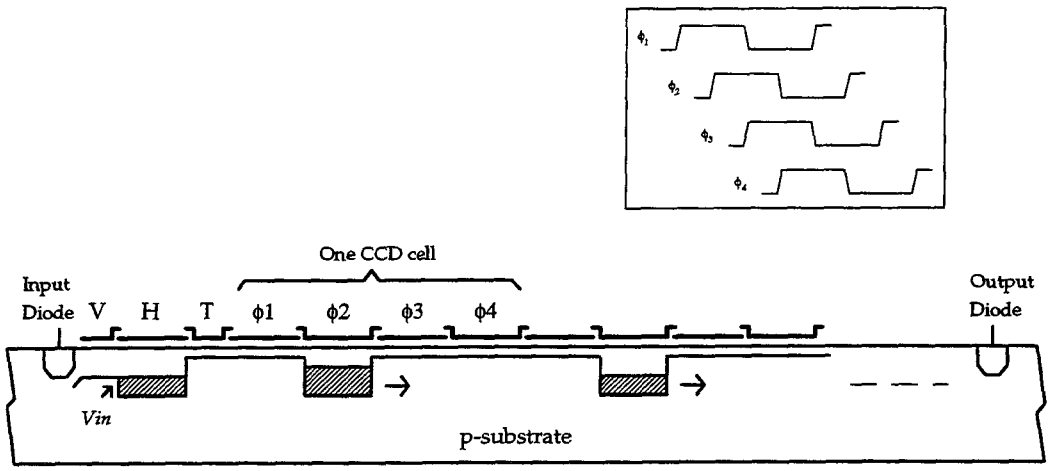


Fig. 3.1: Conventional 4-phase CCD. Charge packets of size proportional to V_{in} are generated by pulsing the input diode low and are then sequentially shifted to their destinations along the N-cell row of electrodes. During write/refresh, old packets are dumped into the output diode as new ones enter the chip. Each CCD cell is composed of four gates, whose driving clocks, for charge shifting, are depicted in the inset.

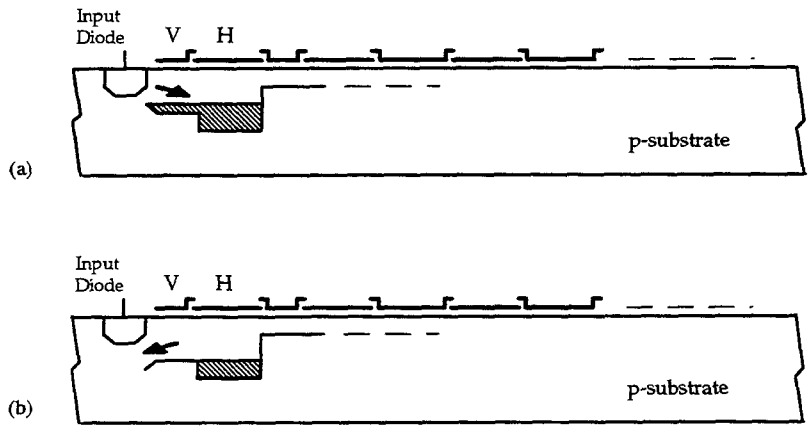


Fig. 3.2: Fill (a) and spill (b) charge injection process.

otherwise thermally generated inversion charge may accumulate). Therefore, by sequentially creating and removing wells, charge can be easily moved back and forth, thus emulating a bi-directional analog shift register.

The creation of charge packets can be accomplished in several ways, including optical methods. One of the most accurate among these processes, known as *fill-and-spill* [Beynon 80], is depicted in Fig. 3.2. As can be seen in Fig. 3.2(a), the input diode (which is normally at high voltage) is pulsed low momentarily, filling (in excess) the first two wells (V and H). When the diode voltage returns to high, only the charge under H is retained (Fig. 3.2(b)), being the size of the charge packet proportional to the voltage difference between H and V (V_{in} in Fig. 3.1).

Since a CCD is a volatile storage medium, periodical refresh of the weight matrix is necessary. During this process (or when a new matrix is loaded), the charge packets previously stored on the chip must be removed. The removal of charge is normally done by means of an output diode, installed at the opposite end of each row from the input diode, as shown in Fig. 3.1. This diode stays permanently high, so charge is automatically dumped during the write/refresh cycles.

Contrary to loading/refreshing the matrix, which requires global charge shifting, computing requires only local transfers of charge [Neugebauer 92][Pedroni 92a]. The computations are performed by moving (or not moving) a determined charge packet to a previously created well which resides under a (common) sensing gate. This process is illustrated in Fig. 3.3, which shows one matrix row, with all the weights initially stored under electrodes $\phi_2^{(i)}$ (where $i = 1,2,...,N$ denotes the matrix columns). Say that the voltage of ϕ_3 is slightly raised, whereas that of ϕ_1 is kept low, as indicated by potential wells in Fig. 3.3(a). With V_{PRE} high, an empty well is created under the sense electrode, ϕ_4, when PRE, the precharge switch, is (momentarily) closed. When PRE is opened, ϕ_4 is left floating at a voltage given by V_{PRE}. If now a well $\phi_2^{(i)}$ collapses, as in Fig. 3.3(b), the charge packet will be transferred to the common gate, ϕ_4, whose voltage then decreases

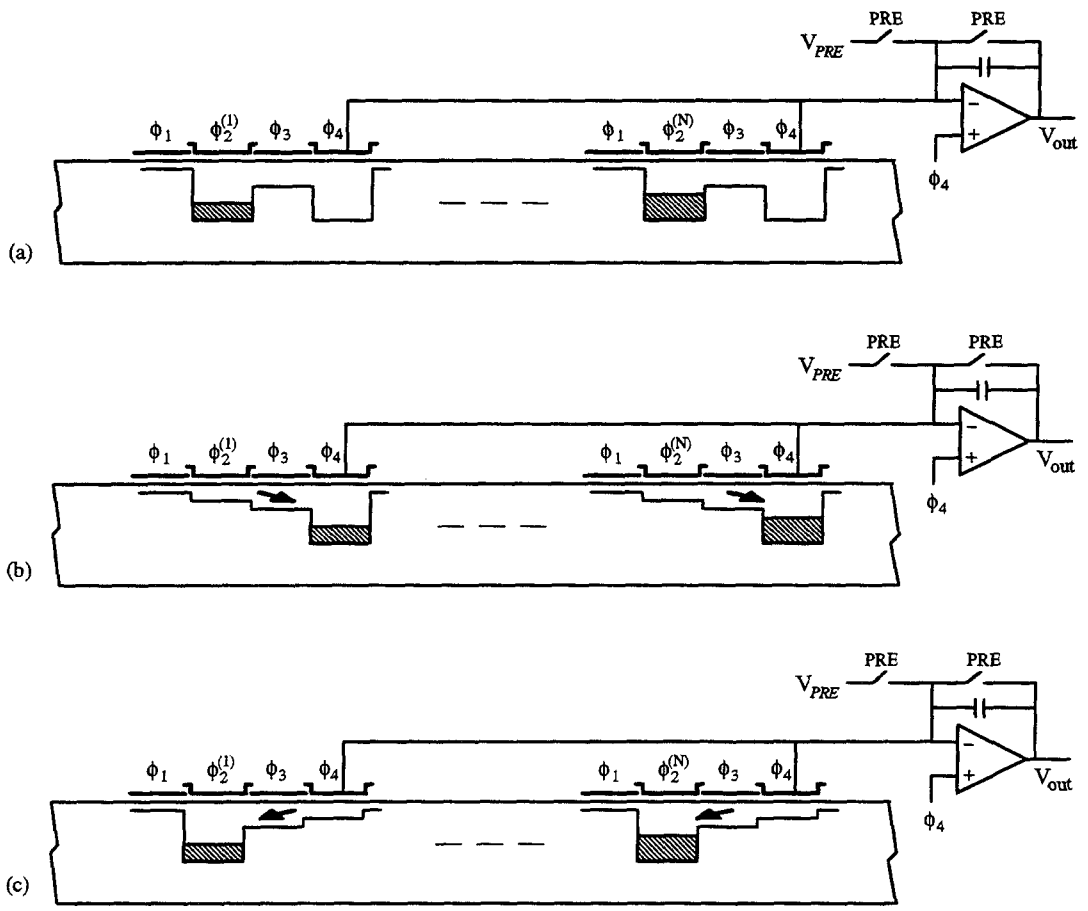


Fig. 3.3: Basic CCD computing sequence: The charge packets are all initially stored under electrodes ϕ_2. In (a), the voltage of ϕ_3 is slightly raised and, by the action of PRE, the precharge switches, a floating well is created under each one of the sense electrodes, ϕ_4. If a ϕ_2 well collapses, as shown in (b), its charge will move towards ϕ_4, therefore causing a voltage variation on the common electrode, ϕ_4, proportional to the total amount of charge received. After the voltage variation is sensed at the output, the charge packets are all returned to their original sites, as shown in (c), leaving the system ready for the next computation.

proportionally to the total amount of charge received. Then, calling **x** the input vector, with components $x_i = \{0,1\}$ (which control whether $\phi_2^{(i)}$ should or should not collapse), and $\mathbf{w}^{(k)}$ the weights stored in row k, with components $w_i^{(k)}$, the voltage variation computed at the output of that row is given by

$$y^{(k)} = \sum_i x_i w_i^{(k)}. \tag{3.1}$$

As equation (3.1) shows, the CCD circuit described above directly implements a binary vector-matrix multiplier (VMM) when binary weights are used, being therefore useful for the construction of Hamming networks, for example. If, on the other hand, the charge packets are analog, the circuit immediately produces a binary-analog VMM (or a threshold function). If, additionally, a divide-by-two circuit is used at the output [Neugebauer 92], a $\log_2 b$-cycle fully-analog VMM is implemented (where b is the number of bits needed to represent the input vector components).

3.3 Locally-Controlled CCD (LCCD) Cells

As noticed above, in a conventional CCD (say an NxN matrix of CCD cells, being one row depicted in Fig. 3.1), the charge packets, representing the (possibly analog) information to be stored at each cell, is input at one end of a row and then shifted along that row up to its storage address. Hence, the write operation requires each charge packet to be moved over up to N (or even $2N$ if a single input pin is used) cells before getting to its final destination. Since the charge transfer efficiency [Carnes 72][Tompsett 73][Berglund 73][Brodersen 75] is lower than unity, that is, some fraction ε of free charge is inevitably left behind after every shift (because of insufficient time for complete charge diffusion and possibly backward spill, as we discuss later), adding to the trailing packets, charge-transfer loss poses one of the major limitations for the use of this technology in

large computing arrays. In a simplistic model, the amount of free charge left behind in a four-phase CCD after N shifts is

$$Q_{LOSS} \approx 4N\varepsilon Q, \tag{3.2}$$

where Q is the charge initially contained in the packet. Therefore, to keep b bits of accuracy, the loss must be less than 1/2LSB, resulting

$$\varepsilon < \frac{1}{N.2^{b+3}} \tag{3.3}$$

(just for loading the matrix); for example, if b=7 bits and N=128, the charge transfer inefficiency must obey $\varepsilon < 2^{-17} \approx 8\text{x}10^{-6}$, which is far below typical values attainable with current Surface-CCD (SCCD) technology. (Experimental results reported in [Fossum 91] and [Neugebauer 93] show $\varepsilon \sim 10^{-3}$ and $\varepsilon \sim 2\text{x}10^{-4}$, respectively.) Buried-CCD's (BCCD's) [Beynon 80], on the other hand, have a smaller ε, but they also have a much poorer linearity, an even more limiting factor; additionally, they require a special fabrication process. We are interested in the higher linearity and simplicity of fabrication (directly compatible with standard double-poly CMOS processes) of SCCD technology, while overcoming its charge-transfer inefficiency problem.

The cells shown in Fig. 3.4 contrast with those of Fig. 3.1 for the fact that charge generation is local, thus requiring no charge shifting. As the measurements will show, this circuit is very linear and properties are well matched over the whole array. LCCD's are suited for binary as well as analog computations. A basic LCCD cell is shown in Fig. 3.4(a), which implements a capacitive memory (analog or binary) plus an AND gate (multiplier). The input/output diode (D) can be shared by adjacent cells, and so can the blocking gate (B), which stays permanently at low voltage

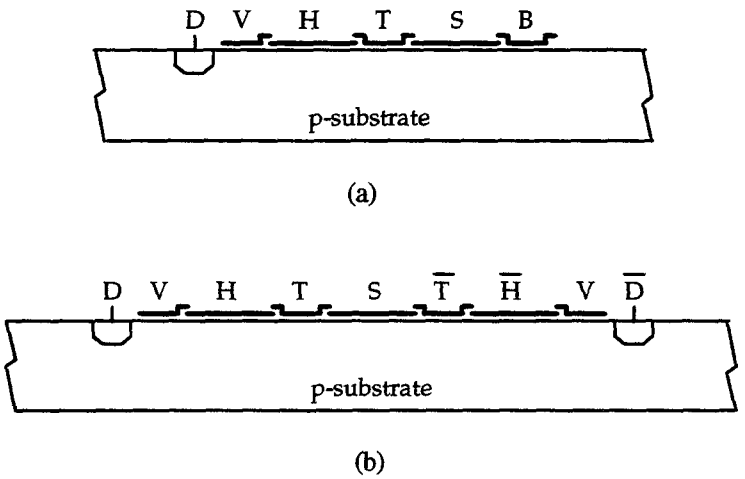


Fig. 3.4: (a) Basic LCCD cell implementation, comprising a memory (analog or binary) plus an AND gate (multiplier). The input/output diode (D) can be shared by adjacent cells, and so can the blocking gate (B), which is kept permanently at low voltage to provide cell isolation. (b) shows an application example where two symmetrically connected cells implement a Hamming unit (a memory plus a complete XOR gate). A preferred Hamming implementation utilizes two separate LCCD cells of (a), in which case charge transfer efficiency is optimized.

in order to provide cell isolation. As in the conventional case, this cell immediately implements a binary VMM if the stored charge packets are binary (i.e., full='1', empty (or 'fat zero' [Beynon 80])='0') or a binary-analog VMM (or a threshold function) if the charge packets are analog. It can also implement a fully-analog VMM if a divide-by-two circuit is used at the output. The cell shown in Fig. 3.4(b), on the other hand, is just a derivation of that of Fig. 3.4(a), and implements a capacitive memory plus an XOR gate.

We describe now the operation of the XOR cell of Fig. 3.4, from which the description of the basic LCCD cell is straightforward. The circuit of Fig. 3.4(b) is approximately symmetric. It is composed of two diodes (D, $\overline{D}$), two vertically controlled gates (V), two horizontally controlled gates (H, $\overline{H}$), two transfer gates (T, $\overline{T}$) and, finally, a sense gate (S). The sense electrode is

common to all cells of one column, so charge moved under that gate by the action of the neighboring electrodes is capacitively converted to a common voltage which is then sensed by an output circuit. After sensing, S is pulsed low, so charge is moved back to its original site, which is either H or $\overline{H}$, leaving the system ready for the next computation. The operation of the circuit can be summarized as follows. During the write phase, if a binary '1' is to be written a full charge packet is stored on one side of the cell, say left, while a '0' (an empty packet or a background charge is stored on the opposite side. During the computing phase, one side (left) receives the input bit, while the other receives its complement, with the side receiving the '0'-bit being the one to cause the charge transfer. In this way, if the input bit and the stored bit are different, they will cause an actual packet (a '1') to be transferred to S, otherwise they will cause a '0', hence performing a complete XOR operation between the input and the stored vectors in just one clock cycle. Each output voltage is then linearly proportional to p, the total number of packets sent to S, which is equal to that row's Hamming distance to the input vector. A winner-take-all network can then be used at the output to identify the best matching (that is, the lowest or highest output voltage, depending on whether the data are in regular or complemented form, respectively). As mentioned before, the size of the charge packets is controlled by the differential voltage $V_{in} = H - V$.

A binary application of LCCD cells is presented in Fig. 3.5, which shows an *M*x*N* Hamming classifier. Each node of the network, consisting of a binary memory plus a complete XOR gate, can be constructed by using either two cells of Fig. 3.4(a) or a single cell of Fig. 3.4(b), as described before. Though charge-transfer efficiency is higher in the former case (as we show later), the latter will be used in one of the prototype chips, for it provides a means to further illustrating the charge-transfer problem. It is easy to verify that, if the second option is adopted, with cells symmetrically connected to all bus lines (thus requiring no input signal twisting) and shared adjacent diodes, each node requires 1 diode (on average) and 7 electrodes, 4 of which can be small (V's, T, $\overline{T}$), for they do not have to hold charge, against 8 full-size electrodes of conventional CCD implementations of

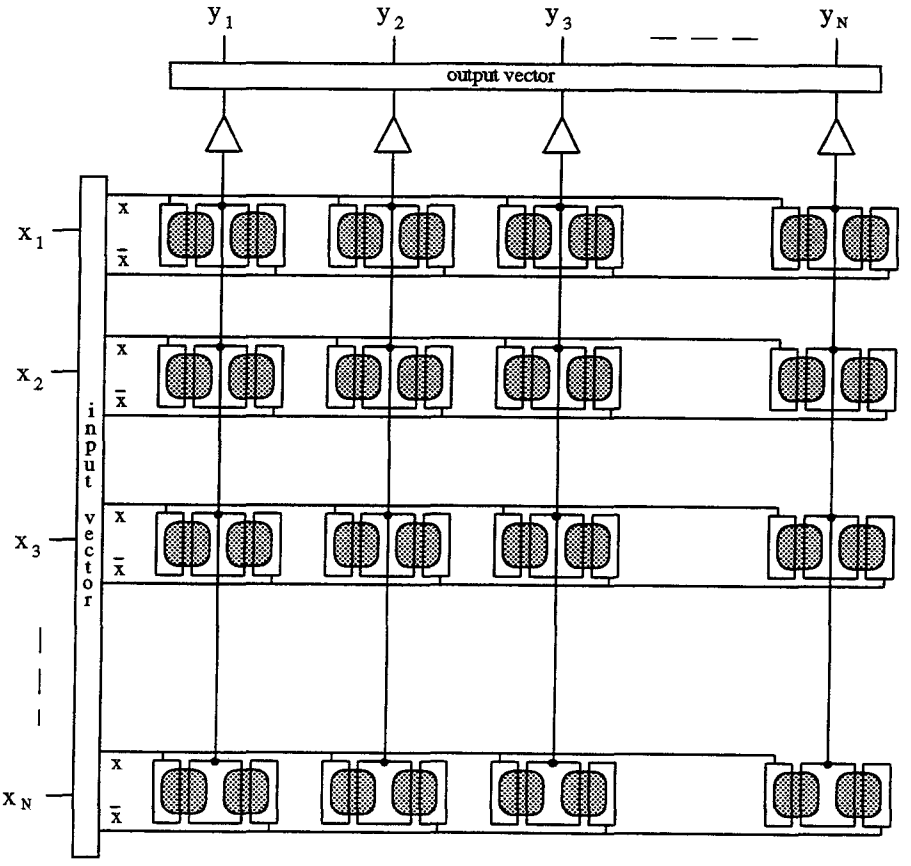


Fig. 3.5: *NxN* Hamming network utilizing the LCCD cell of Fig. 3.4(b) (or two separate cells of Fig. 3.4(a)).

this function [Pedroni 92a]. If, on the other hand, option one is adopted, each node requires 1 diode and 9 electrodes (5 small), on average. For purely analog weight applications, a similar analysis can be made, resulting, on average, 1/2 diode and 4½ gates (2½ small), against 4 full-size electrodes of conventional four-phase CCD's. LCCD implementations of Fig. 3.4(a) and 3.4(b) in standard double-poly 2.0μm CMOS process, including all array interconnections, are typically of the order of 45x30μm^2 and 60x35μm^2, respectively, which are very small areas for complete

memory-compute-accumulate cells, much smaller than similar analog (or digital) implementations utilizing MOS transistors.

Another advantage of LCCD's is the number of "analog clocks" needed for their operation. (The reason for this designation is that charge-transfer efficiency is affected by clock fall rates [Beynon 80], that is, if the gate voltage goes to low faster than charge can accommodate into the new destination site, it may cause charge to spill backwards, so clock edges must be controllable; additionally, these clocks have low and high voltages which are normally different from GND and VDD.) In LCCD implementations this number is just two, against at least four of conventional CCD's. Moreover, controlling backward spill is simpler, since 3-level analog clocks are never necessary in LCCD's.

The charge-sensing circuit adopted in the experiments described in Sec. 3.4 is shown in Fig. 3.6 [Neugebauer 93]. As can be seen, it differs slightly from the version depicted in Fig. 3.3, though, as in that case, a floating-gate technique is used in order to fulfill the information preserving requirement. In the Hamming implementation described above, a feedback amplifier is installed at one end of each column, having its input connected to that column's common gate S. In the schematics of Fig. 3.6, C_F represents the feedback capacitor (normally constructed with poly1-poly2), C_S denotes the total oxide capacitance of the sense gate, C_D the (non-ideal) depletion capacitance of the substrate, and C_C the coupling capacitance. The sensing sequence is very similar to that of Fig. 3.3 and can be summarized as follows. First, one of the transfer gates (T or $\overline{T}$, depending on the input bit) is selected and has its voltage raised, while S is pre-charged to a (high) positive voltage V_{PRE} (PRE switches ON). The presence of C_C makes possible the use of a low reference voltage, V_{REF}, thus increasing the dynamic range of the system when compared to other feedback sensing systems, like that of Fig. 3.3. Next, the PRE switches are turned OFF, what causes S to float. Then, by the action of the H (or $\overline{H}$), charge is transferred to S, therefore decreasing V_S proportionally to the amount of charge received and increasing V_{out} with a voltage

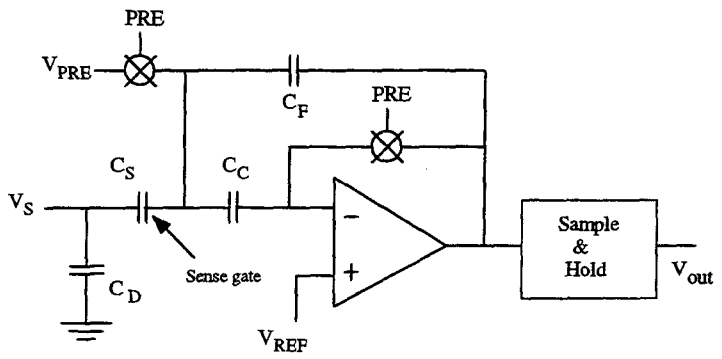


Fig. 3.6: Wide dynamic range, floating gate, capacitive feedback charge-sensing circuit.

gain given by C_S / C_F. After sensing, the charge packets are immediately returned to their original sites by turning the PRE switches ON again and then lowering V_{PRE}.

The overall operation of the LCCD of Fig. 3.4(b), as described along this section, is illustrated through potential well diagrams in Fig. 3.7. During writing, the cells are initially cleaned (dump) by pulsing V high, then the positions of the '1's and the '0's are decided based on the input data (select) and the corresponding diodes are momentarily pulsed low (fill-and-spill), leaving an amount of charge proportional to $V_{in} = H - V$ on the '1'-side. During the computing phase, the sides to be moved are initially selected according to the input vector, causing one of the transfer gates of each cell (T in Fig. 3.7) to have its voltage slightly raised, and the sense gates are pre-charged. Then charge is transferred accordingly, sensed, and returned to its original storage wells.

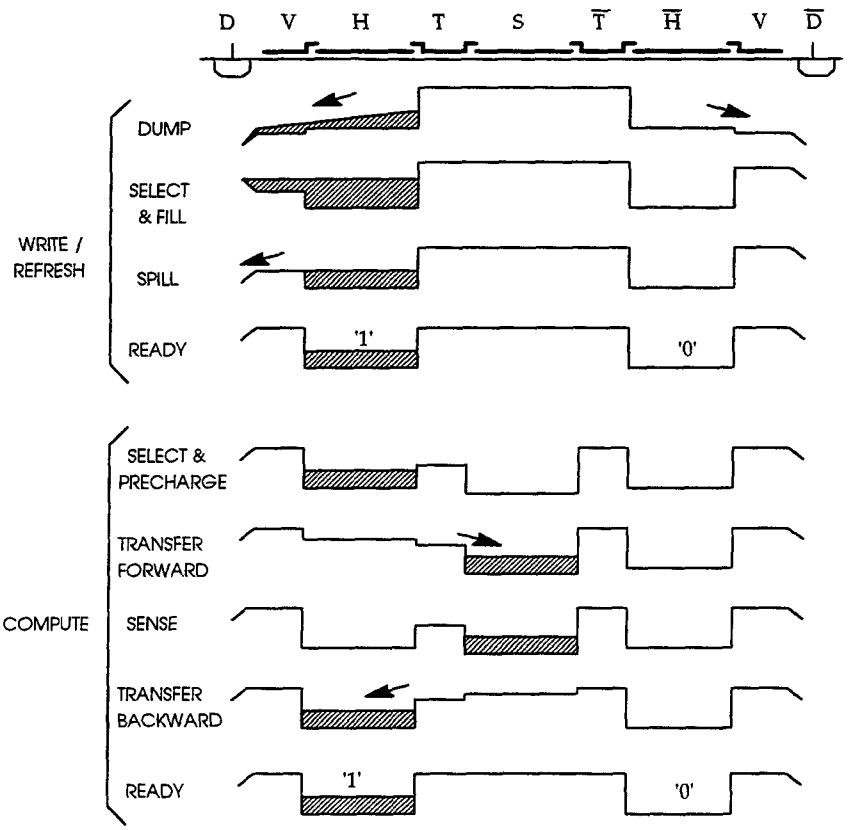


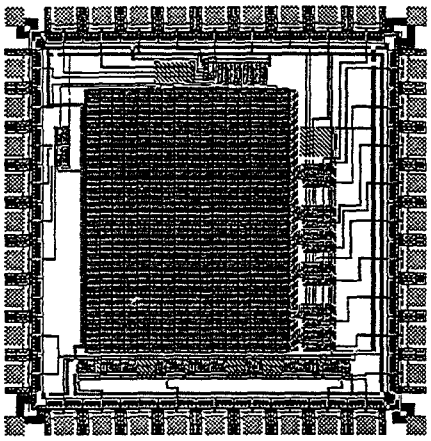
Fig. 3.7: Potential well diagrams describing the operation of the LCCD of Fig. 3.4(b). Only two analog clocks are needed (for forward and backward transfers).

3.4 Experimental Results and Discussion

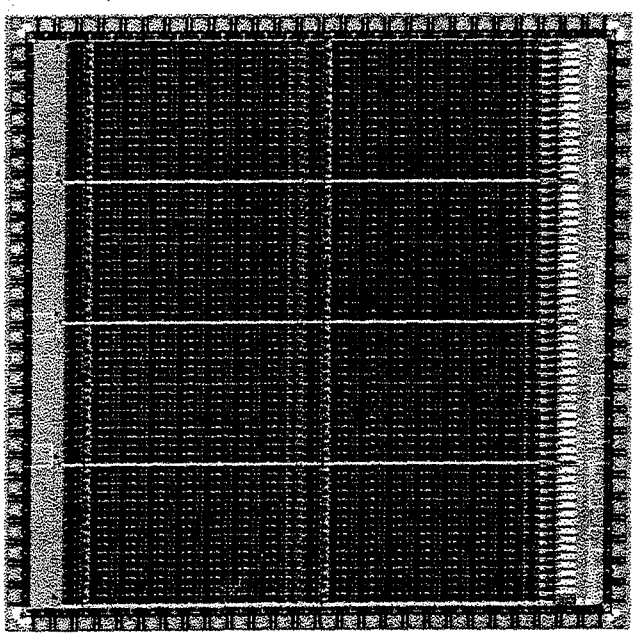
A 16x32 (16 vectors of length 32) LCCD chip was fabricated through MOSIS using conventional double-poly 2.0μm CMOS technology (Fig. 3.8(a)). Each cell has the configuration shown in Fig. 3.4(b), with symmetric bus connections (shared diodes). The cell size, as mentioned earlier, is very small when compared to other analog implementations, what is obviously even more so when compared to digital realizations. (Just to illustrate this fact, a 6-bit 8x16 digital systolic VMM chip, using 1.2μm technology, is presented in Fig. 3.8(b); each cell is $80x1100\lambda^2$, therefore between one and two orders of magnitude larger than LCCD cells.) The output (sense) circuit used in the LCCD chip is similar to that depicted in Fig. 3.6, with $C_F \approx C_S / 2$. The tests were performed with fall times (analog clocks) of 200ns.

3.4.1 Charge-Transfer Inefficiency

Although self-induced drift is the dominant factor in the beginning of the charge-transfer process, the transfer of the last few electrons is governed by thermal diffusion, which is exponentially related to time through large time constants [Carnes 72][Tompsett 73][Beynon 80]. This implies that some charge will inevitably be left behind, whose amount can be approximated as a constant fraction ε of the original charge packet. This charge loss can be aggravated by backward charge spill [Beynon 80], as described earlier. Besides these fractional effects on free charge transfer, there are also other effects which are approximately packet-size independent, and were not considered in our initial discussion in Sec. 3.2. The main effect is the trap of electrons in fast states at the Si-SiO_2 interface. Although the capture of electrons by these states is basically instantaneous, their re-emissions happen in times raging to milliseconds [Carnes 72][Beynon 80]. Then, if a CCD shift register has received a long series of 'zeros', these will cause the traps to be relatively empty, provoking a large amount of charge to be captured from the first 'one' following this sequence. This effect is illustrated in the picture of Fig. 3.9(a), which shows a test realized on

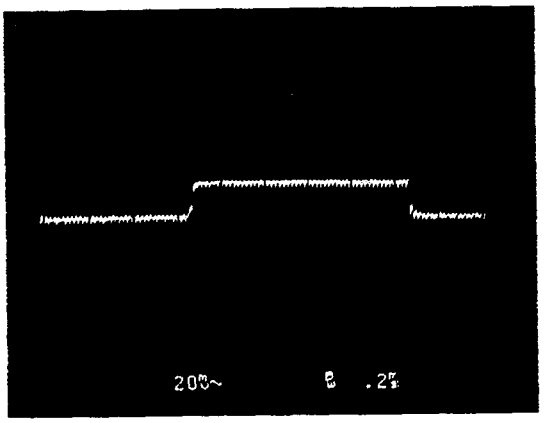


(a)

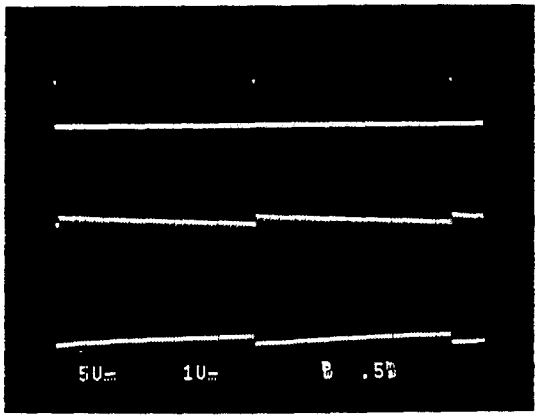


(b)

Fig. 3.8: (a) 16x32 2.0μm Hamming LCCD CMOS prototype chip; (b) a digital 6-bit 8x16 1.2μm VMM chip. Notice that (a) and (b) are not to same scale, which is apparent through the number of pins of each chip.



(a)



(b)

Fig. 3.9: Charge transfer loss in a conventional CCD: (a) A 52-cell/row chip with 52 '1's followed by 52 '0's. As can be seen, a large amount of charge is captured from the leading packet, being re-emitted later, thus affecting the leading '0's; (b) A 32-cell CCD emulated with the LCCD of Fig. 3.4(b) for measurement purposes. There is no background charge and the number of computations before refreshing (upper channel) is 122. As can be seen, the '1's decrease progressively while the '0's increase.

a 52-cell conventional CCD chip, with two long series of '0's and '1's being alternately applied to the chip. As can be seen, a considerable amount of charge is taken from the leading packet, being re-emitted later, hence affecting the leading '0's. This amount of charge, Q_0, is almost independent of packet size, and so a background charge (also called *fat zero* or bias charge) is normally used in order to keep these states permanently filled [Carnes 72][Tompsett 73][Beynon 80]. This solution is not complete, however, because of statistical variations of the capture-emission rate and also because of the edge effect [Carnes 72][Tompsett 73] originated from the fact that the area under an electrode occupied by a fat zero is still smaller than that occupied by a full packet, thus not completely filling all of the surface states. As a consequence of Q_0, it is expected that the gain of a CCD circuit be linear only for V_{in} above a certain voltage which creates enough charge to cause both the fixed and the fluctuating parts of the trapping effect to be negligible (typically a fraction of a volt).

As mentioned earlier, the reason for choosing the cell of Fig. 3.4(b) in one of the prototype chips is because it allows the simulation of a pure LCCD cell (Fig. 3.4(a)) when only one side is used, as well as it allows charge transfer inefficiency measurements, as in a conventional CCD, if both sides are operated alternately. We consider first the basic LCCD cell. One can verify that the free charge loss in this case is approximately

$$Q_{LOSS} \approx \varepsilon(Q_H + Q_S), \tag{3.4}$$

where Q_H and Q_S (=0) are the charges initially stored under H and S, respectively. This result shows that the charge loss is practically zero (independent of the number of cycles, for $Q_H + Q_S$=constant), and so charge-transfer inefficiency basically does not affect the system. Besides, the use of background charge is simpler in LCCD's and the edge effect basically does not exist. If we consider now a conventional CCD, emulated by Fig. 3.4(b) as described above, we verify a charge loss of approximately $i\varepsilon Q_H$, therefore dependent on i, the number of computation

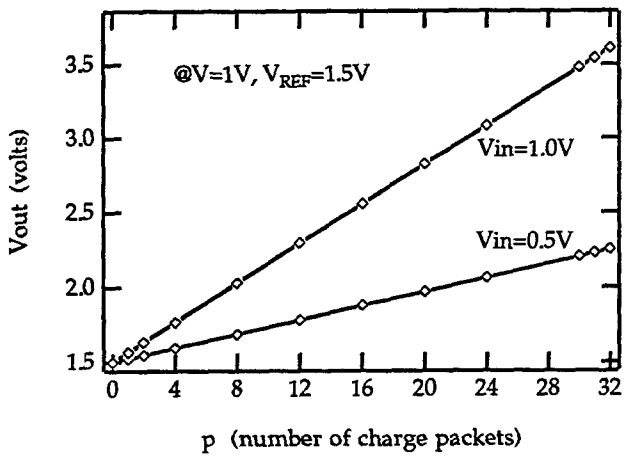
cycles (which is similar to the charge loss given by (3.2)). The experimental behavior of the latter is pictured in Fig. 3.9(b), which shows a series of 120 alternating '1'-'0' computations, with no background charge, and with the upper channel of the scope showing the refresh clock. As can be seen, the '1's decrease progressively whereas the '0's increase. The measured charge transfer inefficiency was $\varepsilon \sim 2.2\text{x}10^{-4}$.

3.4.2 Charge-Voltage Linearity

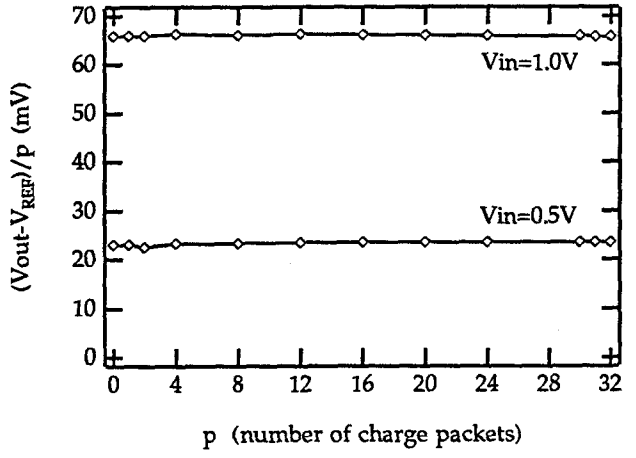
Apart from the charge-transfer problem, the single most important experiment of interest for binary applications is the measurement of the linearity between the number of fixed-size charge packets, p, sent to S and the corresponding variation of the output voltage, V_{out} (charge-voltage linearity). This test was performed for V_{in}=0.5V and V_{in}=1.0V, with V fixed at a voltage slightly above threshold ($V = 1.0\text{V}$) and p varying from 0 to 32. The measurements are plotted in Fig. 3.10(a), while the normalized results are shown in Fig. 3.10(b). As can be seen in Fig. 3.10(a), $V_{out} = V_{REF}$ when p=0, growing linearly as p is incremented. In Fig. 3.10(b) the voltage per packet is 23±0.5mV for V_{in}=0.5V and 66±0.5mV for V_{in}=1.0V. Measurements on the prototype chip show a system resolution of approximately 44.3dB (a little over 7 bits), therefore suitable for handling vectors of length N=128, for example. A qualitative result is also shown through the picture of Fig. 3.11(a). Here, p is gradually reduced from 32; when in the middle, the input vector is complemented, producing a same-value output, as expected, responsible for the flat part in the center of the staircase. A detail of the staircase is also shown in Fig. 3.11(b).

3.4.3 Voltage-Voltage Linearity

For analog applications, the most fundamental set of results comes from the combination of charge-voltage linearity measurements (described above) with measurements of the linearity between V_{in} and V_{out} (voltage-voltage linearity). The voltage-voltage linearity test was performed

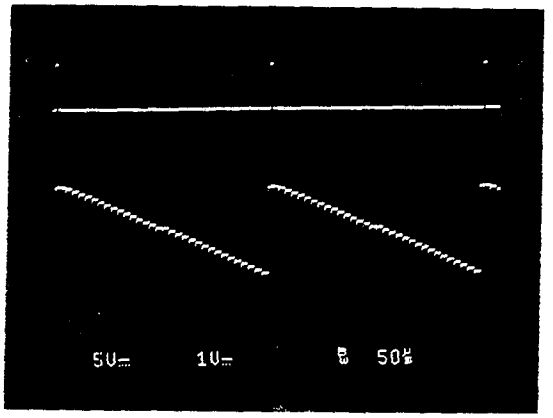


(a)

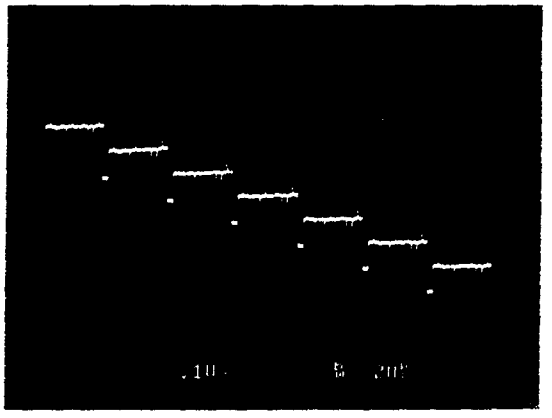


(b)

Fig. 3.10: Plots of measured output voltages versus the number of fixed-size charge packets transferred to the sense gate (charge-voltage linearity). As can be seen in (a), $V_{out} = V_{REF}$ when p=0, growing linearly with p thereafter. This behavior is further illustrated through the normalized results of (b) that show voltages per packet within 23±0.5mV for V_{in}=0.5V and 66±0.5mV for V_{in}=1.0V. The measured resolution of the system is over 7 bits.



(a)



(b)

Fig. 3.11: A qualitative verification of linearity: (a) p is reduced from 32 in steps of size 1, with a (intentional) system inversion (flat part) at the mid-point of the staircase; (b) A detail of the staircase.

for p=1 and p=16, as shown in Fig. 3.12, with $V = 1.0\text{V}$ and $0.4\text{V} \le V_{in} \le 2.0\text{V}$. Fig. 3.12(a) shows the global results, where the highly linear behavior of the system is apparent. The ratio of the variations of V_{out} and V_{in} (gain), relative to the mid-point of the voltage range and normalized to p, is presented in Fig. 3.12(b). The measured average voltage-voltage linearity of the system is approximately 7 bits.

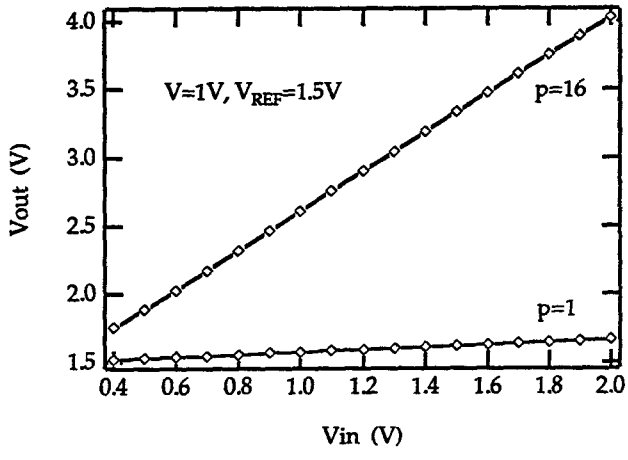
3.4.4 Input Reference Voltage Effect

Although in practice the value of the reference voltage V used for writing is normally kept fixed, we investigate the effect of its variation on the output voltage of the system. Since $C_{SiO_2} >> C_D$ for $V > V_T$, it is expected that this effect be almost negligible, that is, that V_{out} be little dependent on H or V when the input voltage $V_{in} = H - V$ is kept constant. Besides, the (small) variation of C_D with V is negative, hence the (little) variation of V_{out} is expected to be positive. The experimental behavior of the circuit, for V_{in}=0.5V and V_{in}=1.0, with p=32, is plotted in Fig. 3.13(a), for V varying between V_T (threshold) and $3V_T$, while the normalized voltage per packet is plotted in Fig. 3.13(b). As can be seen, the circuit behaves as predicted, and the results coincide with those of Fig. 3.10(b) when V=1.0V.

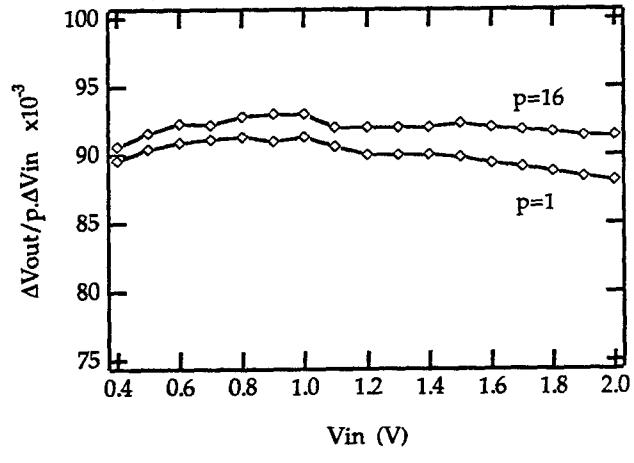
3.4.5 Power Consumption

Because of the charge conservation mechanism of CCD's, the power consumption of this type of circuit is basically that of the driving clocks plus that of the sense amplifier. The power consumption in the LCCD matrix is very small; each vector consumes approximately

$$2C_d V_{clock} f, \tag{3.5}$$

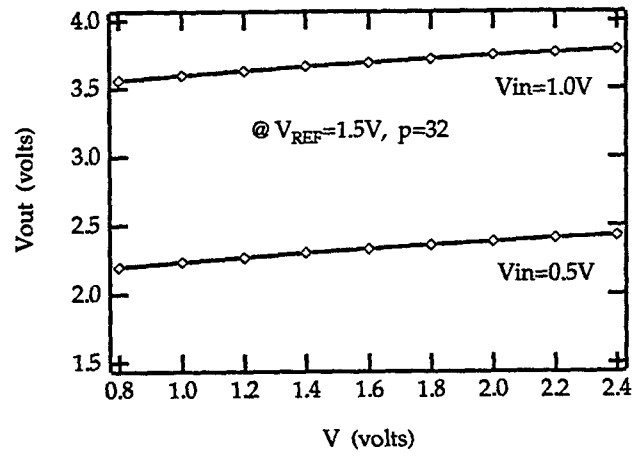


(a)

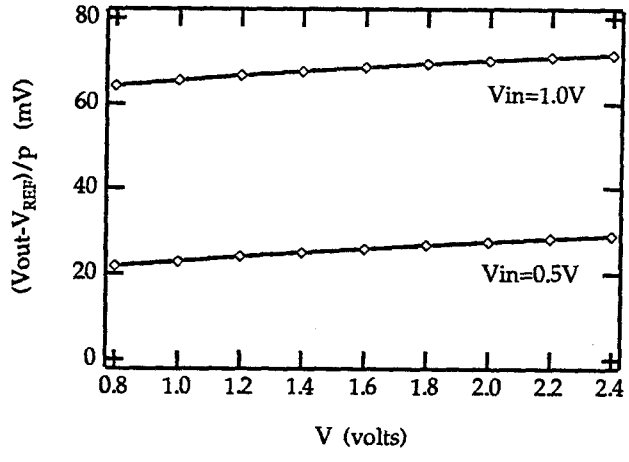


(b)

Fig. 3.12: Plots of measured output versus input voltages (voltage-voltage linearity). The measurements were realized for p=1 and p=32, with global results plotted in (a), where the system's linear behavior is apparent. This is further illustrated through the normalized results (gain per packet) plotted in (b). The measured voltage-voltage linearity of the system is approximately 7 bits.



(a)



(b)

Fig. 3.13: Plots of measured output voltages versus reference voltages for a fixed input. As expected, the size of the charge packets is little sensitive to variations of V or H when V_{in} is kept constant, as can be seen in the global results of (a) and through the normalized error per packet of (b).

where V_{clock} is the voltage swing on the H and S gates (for forward and backward charge transfer, respectively, as depicted in Fig. 3.7), resulting an energy per bit in the pJ range, typically. The power consumption in the sense circuit, on the other hand, has two components, one related to the driving clocks (to charge/discharge the capacitors) and another due to the bias current of the amplifier, which is frequency dependent too, since the high frequency poles of the amplifier grow with I_{BIAS}. The total power consumption tends therefore to be dominated by the sense circuit. The measured power consumption of the prototype chip was approximately 0.8mW/vector at 250k matrix computations per second, being mostly destined to the sense circuits.

3.4.6 Dark Current and Refreshing

One last consideration refers to the need for refreshing the weights. No matter how accurate the matrix-loading process is, long-term effects will eventually obliterate the charge encoded information. The main effect is the thermal generation of minority carriers, an effect also known as *dark current* [Beynon 80]. Calling N_{DC} the average electron generation rate due to this effect and N_S the number of electrons contained in the largest signal packet, the maximum time allowed before the generated charge introduces a b-bit error is

$$T_{\max} = (N_S / N_{DC}) 2^{-(b+1)}. \tag{3.6}$$

For instance, for a dark current of ~10nA/cm^2 [Sequin 75][Beynon 80], about 600 electrons/μm^2.s are generated, while N_S~3,000 electrons/μm^2.V for a typical 2μm CMOS process; then, if $V_{in\max} = 2V$ and b=7 bits, $T_{\max} = 20\text{ms}$. Current high quality fabrication processes tend to attenuate this effect, however. For instance, this effect only affected our experiments after periods well over this estimate, hence allowing hundreds of computations to be typically performed before refresh was needed. Notice also that Hamming LCCD's are differential in nature, so partially compensating this and other long-term charge deterioration effects.

Chapter 4

MOS Distance Processors and Related Cells

In the previous chapters we have introduced a series of circuit models and VLSI implementations of functions that are at the heart of most analog parallel signal processing systems. If we divide this kind of system into two parts, one corresponding to the output unit and the other to the computing unit, we may say that Chapter 2 was devoted to the former, whereas Chapter 3 deals with the latter. Although we have dedicated a great research effort to the development of these new techniques, especially those of Chapter 3, where the potentiality of CCD technology for signal processing was objectively delineated, other alternatives to compose the computing unit referred to above were also devised. These circuits make use of conventional CMOS technology to compute certain kinds of distance metrics, namely the absolute-distance and the LMS metrics, and to perform vector-matrix multiplication as well. Needless to say that these cells allow a wide variety of algorithm implementations. In this chapter we introduce these distance-based circuits and other related cells, which are normally accompanied by experimental results, as usual.

4.1 Introduction

As discussed earlier, cells that compute certain kinds of distance metrics are essential for the implementation of a wide variety of signal processing algorithms. In this chapter we introduce a series of new circuits of this kind and illustrate their applications to Hamming networks, vector quantizers, and median filters, among others. Other cells that are related to the distance metrics or that are typical of systems of this kind are also included. The circuits make use exclusively of conventional CMOS technology, being therefore of low cost and low power consumption. The models are experimentally verified through chips fabricated using either 1.2μm or 2.0μm CMOS technology.

4.2 Balanced MOS Vector-Matrix Multiplier and Vector Quantizer

Simplicity is a key factor in the development of high-density systems. We discuss in this section a balanced, four-quadrant, fully-analog vector-matrix multiplier (VMM) and a vector quantizer (VQ) [Pedroni 94f] which require very small silicon area for their implementations, while presenting high linearity, a flexible input dynamic range, a symmetric power consumption behavior, and are inherently suitable for parallel operations. The circuits require only four transistors per synapse in the VMM and two in the VQ, plus two (small) write/refresh transistors.

4.2.1 Linear Transconductor Array

As we know, the central operation performed in many parallel signal processing systems can be described in terms of vector-matrix multiplication. In this section we present a parallel VMM circuit and its extension to a VQ implementation [Pedroni 94f], both demanding very small silicon area, therefore suitable for high-density analog systems. The latter requires only two transistors per

cell, while the former requires four transistors per synapse for complete four-quadrant computations. The area required for this implementation is close to that of [Neugebauer 92], which is one of the smallest VMM implementations reported to date. As shown below, the alternative presented here overcomes a major limitation of other MOS implementations in saturation mode [Szczepanski 93]. It is also superior to triode-mode implementations [Kub 90] for it has a more flexible dynamic range and, mainly, requires no output approximations for current sensing.

A block diagram of the system is shown in Fig. 4.1, which computes

$$y_i = \Sigma_j x_j . w_j^{(i)} \quad \text{or} \tag{4.1a}$$

$$y_i = \Sigma_j (x_j - w_j^{(i)})^2 \ , \tag{4.1b}$$

depending on whether the individual cells are VMM or VQ units, respectively. Let us consider initially the vector-matrix multiplier case. A linear transconductor was proposed in [Szczepanski 93], which can be used to emulate a multiplier, as shown in Fig. 4.2. With all transistors in saturation, i.e., $I = 0.5\beta(V_{GS} - V_T)^2$, and utilizing conventional MOS notation, where $\beta = \mu C_{ox} W / L$ (Appendix A), it is easy to verify that the current components of this circuit are

$$\begin{aligned} I_1 &= I_1' + I_1'' \\ I_1' &= \frac{\beta}{2}(V_2 - \varepsilon)^2 \\ I_1'' &= \frac{\beta}{2}(V_1 + \varepsilon)^2 \end{aligned}$$

and

$$\begin{aligned} I_2 &= I_2' + I_2'' \\ I_2' &= \frac{\beta}{2}(V_1 - V_2 + \varepsilon)^2 \ , \\ I_2'' &= \frac{\beta}{2}\varepsilon^2 \end{aligned} \tag{4.2}$$

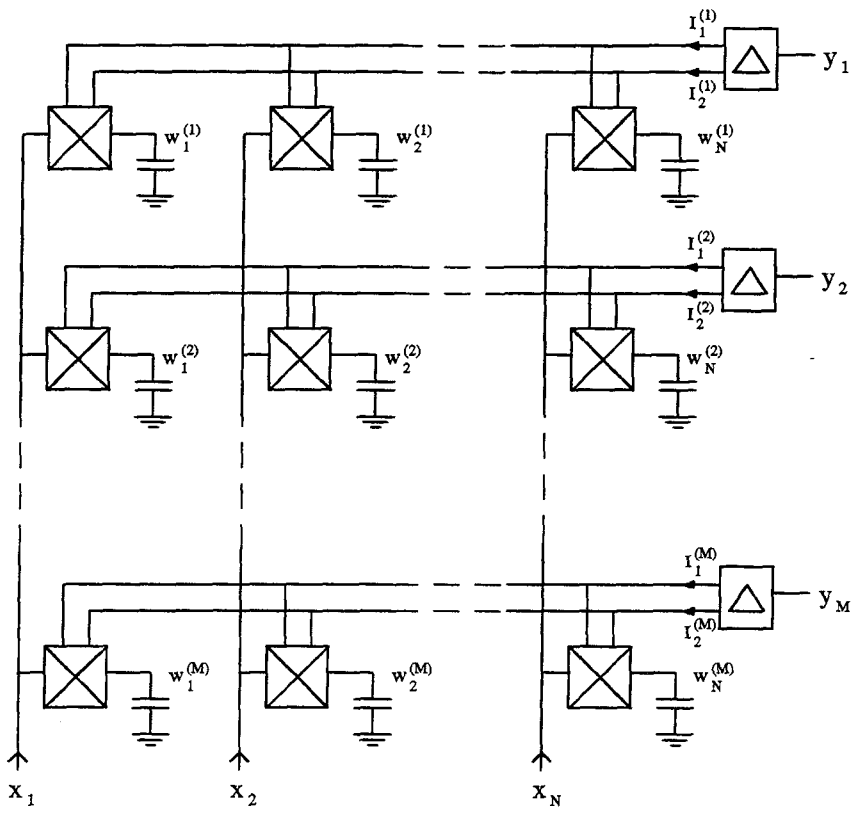


Fig. 4.1: *MxN* parallel analog vector-matrix multiplier (or vector quantizer) architecture. $w_i^{(j)}$ represents the (stored) matrix components, whereas **x** and **y** represent the input and output vectors, respectively.

where $\varepsilon = V_{REF} - V_T \geq V_{2\max} - V_{1\min}$. V_{REF} is used to adapt the input dynamic range to this constraint, thus giving the circuit great flexibility and allowing V_1 and V_2 to be bipolar (4-quadrant). From (4.2) we obtain that the differential current is exactly equal to

$$\Delta I = I_1 - I_2 = \beta V_1 V_2 , \tag{4.3}$$

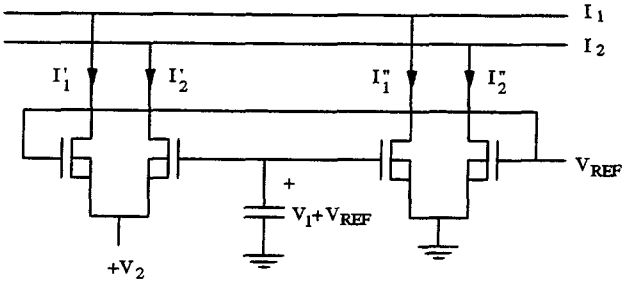


Fig. 4.2: Original transconductor cell. The differential output current is given by $\Delta I = \beta V_1 V_2$.

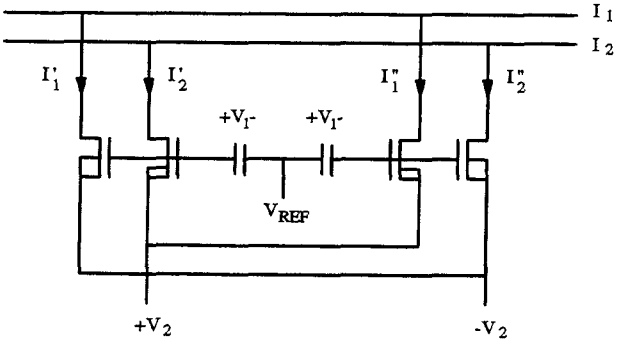


Fig. 4.3: Balanced 4-quadrant multiplier cell. The differential output current is $\Delta I = 4\beta V_1 V_2$.

therefore implementing (4.1a) directly. However, the quadratic law used to model the transistors in the analysis above did not consider second-order effects, like channel-length modulation (Appendix A), which are all non-linear and non-symmetric. If we plot the current components of equation 4.2, we verify that those that are not constant are either all small or all large (see Fig. 4.4(a)). This non-symmetric behavior causes the actual output of the circuit to diverge significantly from the theoretical prediction given by (4.3). This discrepancy will be verified experimentally later.

4.2.2 Balanced Multiplier Cell

Let us now consider the circuit of Fig. 4.3, which has its two pairs of n-channel transistors built inside p-type wells of symmetrical potentials. The current components of this circuit are

$$I_1 = I_1' + I_1''$$
$$I_1' = \frac{\beta}{2}(V_1 + V_2 + \varepsilon)^2$$
$$I_1'' = \frac{\beta}{2}(V_1 + V_2 - \varepsilon)^2$$

and

$$I_2 = I_2' + I_2''$$
$$I_2' = \frac{\beta}{2}(V_1 - V_2 + \varepsilon)^2 \quad . \qquad (4.4)$$
$$I_2'' = \frac{\beta}{2}(V_1 - V_2 - \varepsilon)^2$$

The first difference between this cell and that of Fig. 4.2 is that the current components are now symmetric, as immediately revealed by the equations in (4.4), which are plotted in Fig. 4.4(b) for $-1V \leq V_1, V_2 \leq 1V$ and $\varepsilon = 2V$. This behavior will cause an almost complete cancellation of transistor non-idealities. The second difference is that the differential output current is now given by

$$\Delta I = 4\beta V_1 V_2, \qquad (4.5)$$

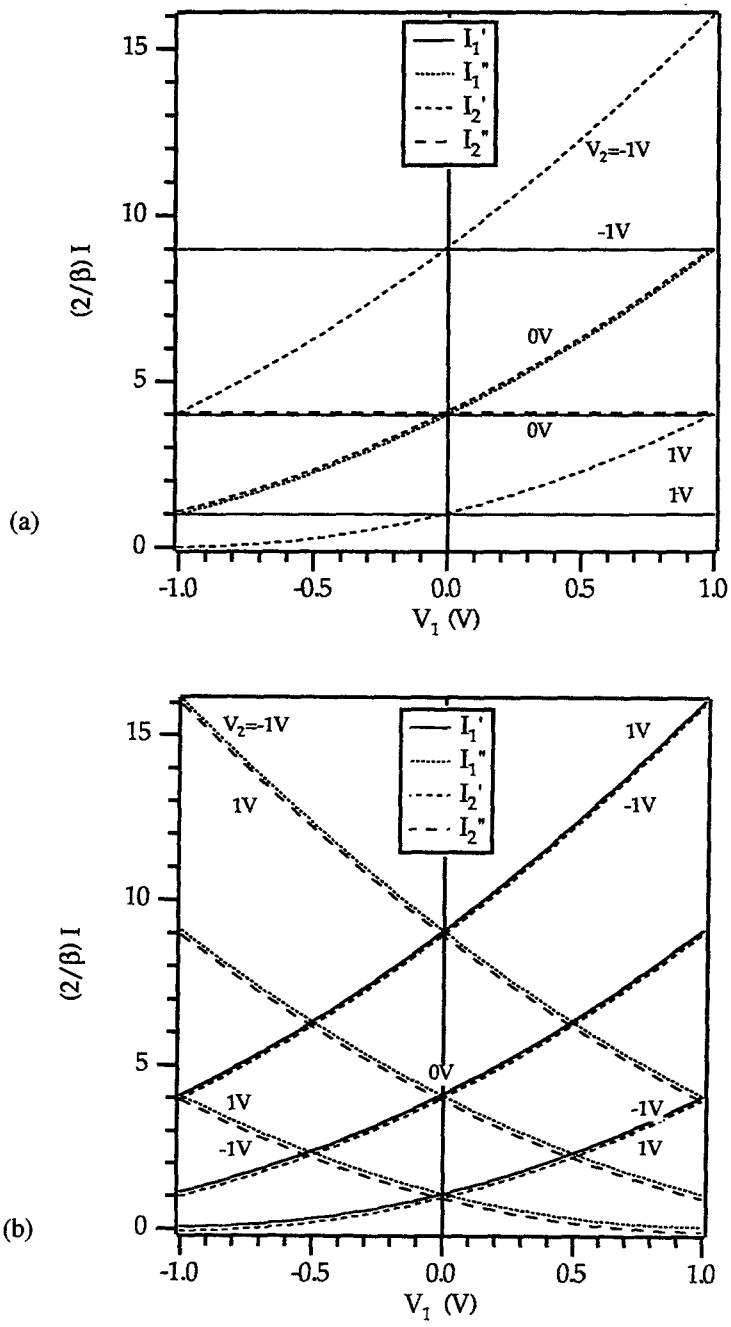


Fig. 4.4: Current components of the original (a) and the balanced (b) VMM cells. As can be seen, they are symmetric in the latter, whereas the components that are not constant are either all small or all large in the former. The adopted input dynamic range is [-1V, 1V], with $\varepsilon = V_{REF} - V_T = 2V$.

that is, the circuit not only implements (4.1a) directly and in a symmetrical fashion, but it does it with a gain four times as big as in the previous case (4.3). Additionally, if we compare the power consumption of each cell (i.e., $I_1 + I_2$), we verify that they are exactly the same at the idle point $V_1 = V_2 = 0\,\mathrm{V}$, being that of the balanced cell (cell 2) normally smaller than that of the non-balanced (cell 1) circuit, in spite of its higher gain. This behavior is illustrated in Fig. 4.5(a). Moreover, if we determine the output signal to power consumption ratio (SPR) of each cell, we obtain

$$\frac{SPR_{cell2}}{SPR_{cell1}} = 2\frac{(V_1+\varepsilon)^2+(V_2-\varepsilon)^2-V_1V_2}{V_1^2+V_2^2+\varepsilon^2}. \tag{4.6}$$

Recalling that $V_{2\max} - V_{1\min} \leq \varepsilon$, we verify that (4.6) is never smaller than unity, therefore indicating a superior signal to power consumption performance of cell 2. In fact, this ratio is approximately 4 for small signals, as can be seen in Fig. 4.5(b). Still another advantage of this symmetric behavior is that it makes the design of the differential output circuit (current mirror) easier because of the smaller current peak of the balanced cell.

A complete VMM cell for VLSI implementations is presented in Fig. 4.6, in which the "write" transistors (M3) were also included. The dashed lines indicate that the $\pm V_2$ bus (input-vector components) can be used to load the $\pm V_1$ signals (matrix components), though this would cause additional charge injection, as we show later. As discussed above, one important property of this implementation is that circuit errors tend to be *locally* compensated (within each cell), having therefore little effect over the array as a whole, what is rarely the case in VLSI implementations of analog parallel processors.

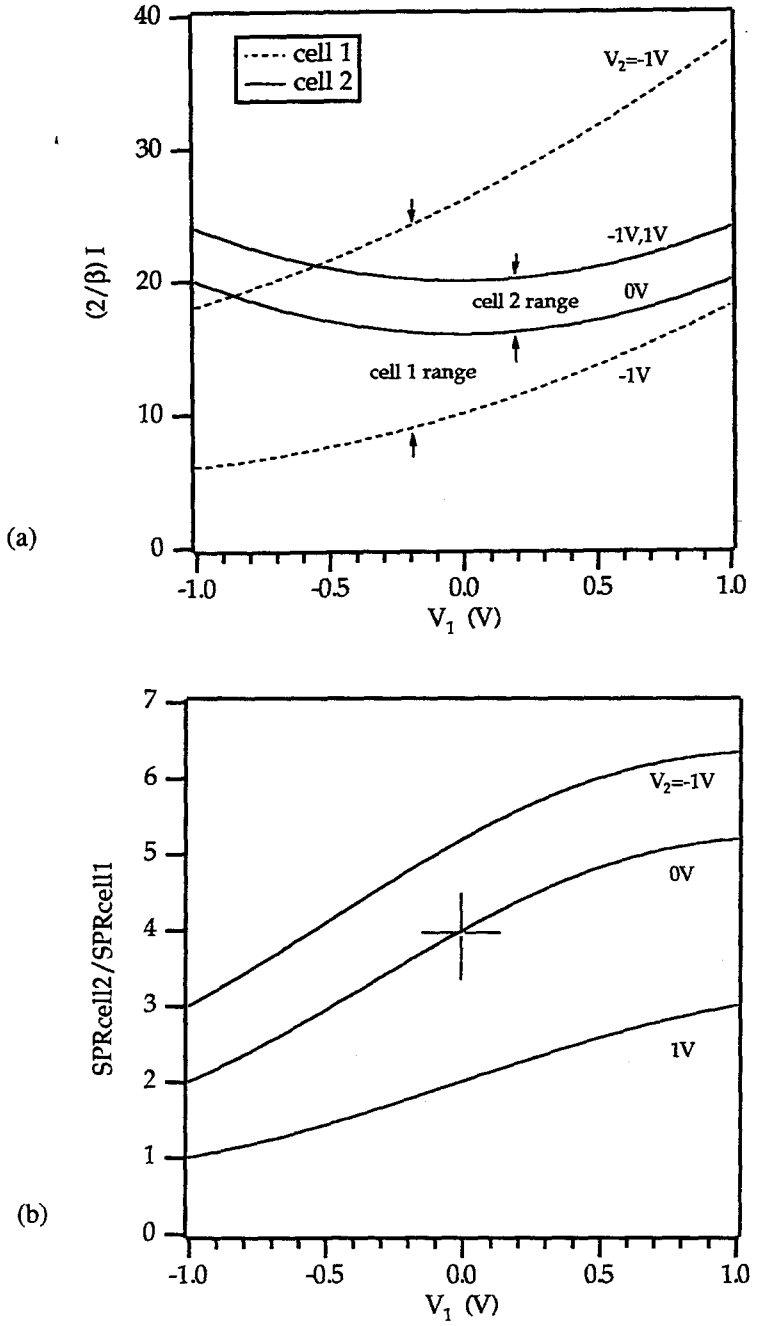


Fig. 4.5: Gain versus power consumption of the conventional (cell 1) and balanced (cell 2) circuits. As shown in (a), the maximum current range of the latter is almost always lower that the former. Since its gain is four times as big, this gives rise to an output signal to power consumption ratio (SPR) which is always better in the latter, as shown in (b). As can be seen, this ratio is about 4 for small signals.

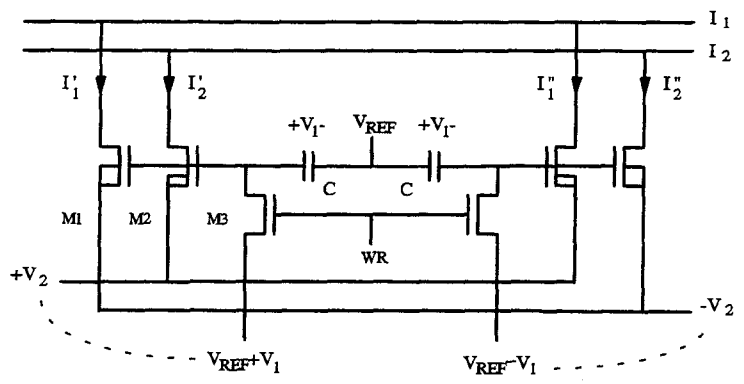


Fig. 4.6: VLSI implementation of the VMM cell of Fig. 4.3, including a pair of write transistors. The dashed lines indicate that the $\pm V_2$ bus can be used to load the $\pm V_1$ signals.

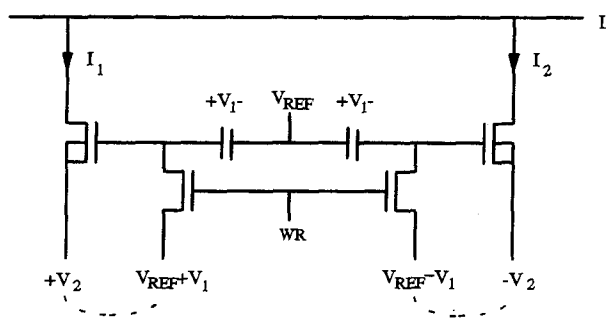


Fig. 4.7: Balanced 4-quadrant VQ cell. The output current is given by $I = \beta(V_1 - V_2)^2 + \text{constant}$.

4.2.3 LMS Vector Quantizer

Contrary to the non-balanced case, with a small modification the balanced cell can implement also equation (4.1b), i.e., an LMS vector quantizer. The complete VQ circuit, including the write/refresh transistors, is shown in Fig. 4.7. As before, the dashed lines indicate that the $\pm V_2$ bus can be used for loading the $\pm V_1$ signals. Comparing Fig. 4.7 to Fig. 4.6, we verify that two transistors were removed and that the differential-current circuit is no longer needed at the output. It is easy to verify that the output current of this circuit is given by

$$I = I_1 + I_2 = \beta(V_1 - V_2)^2 + \beta\varepsilon^2, \tag{4.7}$$

which is a direct implementation of (4.1b), for the term $\beta\varepsilon^2$ =constant is common to all matrix rows, therefore not affecting the winner-take-all detection at the output.

4.2.4 Charge Injection

One of the main concerns in the design of systems that require the storage of analog signals is with charge injection. Although some charge injection is inevitable during writing/refreshing the weights, it might also occur during the computing phase in systems where the voltages surrounding the storage media change constantly. This is the case of all circuits presented above, for the input signals (V_2's) may vary before each computation. It is therefore necessary to verify how these variations affect the stored signals (V_1's), for it is desirable that the system be able to perform as many computations as possible before the need for refreshing arises.

Write-mode charge injection: Let us consider the complete VMM circuit of Fig. 4.6. During writing/refreshing, there are two possible sources of charge injection, one due to clockfeedthrough (WR pulse) and another due to the voltage variation of either V_2 or V_{REF} (though the latter only

occurs if the dashed-line option is adopted). The capacitances of one-half the circuit, for charge-injection purposes, are shown in Fig. 4.8(a). We want to investigate the effects of charge injection on the stored signals (V_1's), starting with the clockfeedthrough component. In this case, the circuit can be modeled as in Fig. 4.8(b), where all fixed voltages were simply replaced by GND, since they represent low-impedance paths. Clockfeedthrough only happens when the write pulse is turned OFF, because in the other cycle a low-impedance path exists. As can be seen in the circuit, the charge injection effect on V_1 is given by

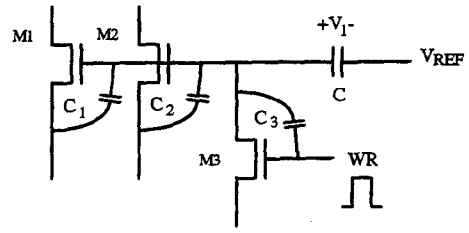
$$\Delta V_1 = -\frac{C_3}{C + C_1 + C_2 + C_3} V_{DD} \,. \tag{4.8}$$

This error component can be easily controlled by using small write transistors (small C_3). (We will return to the discussion on clockfeedthrough in Appendix B.) As mentioned above, a second charge-injection component occurs when the dashed-line option is used to enter the weights. In that case, writing can be done with either $\pm V_1$ applied to the input bus and $V_{REF} = 0\text{V}$, or $V_{REF} \pm V_1$ applied to the inputs. In either of these cases the storage media will suffer a voltage variation in its surroundings equal to V_{REF} right after WR is applied. This is illustrated in Fig. 4.8(c), where we verify that the effect on the stored signals is given by

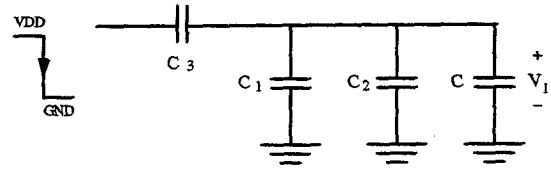
$$\Delta V_1 = \frac{C_1 + C_2}{C + C_1 + C_2} V_{REF} \,. \tag{4.9}$$

Since transistors M1 and M2 are in general not of minimum size (for parameter-matching purposes), this term is more difficult to control, requiring C to be relatively large. Therefore, for high accuracy systems, the use of a separate bus for loading $\pm V_1$ is advisable, as indicated in Fig. 4.6. One important observation about expressions (4.8) and (4.9), however, is that, though they affect the stored voltages, the variations have the same sign on both sides of the circuit. As a result, this is equivalent to simply replacing V_{REF} with some V'_{REF} in (4.4), having therefore no effect on

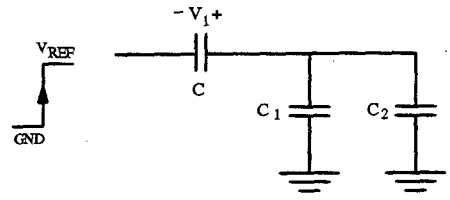
(a)



(b)



(c)



(d)

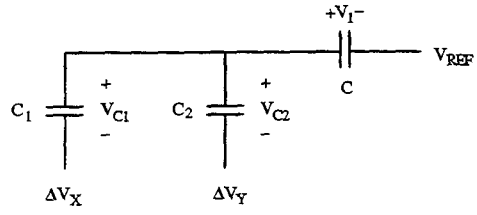


Fig. 4.8: Charge-injection analysis of the balanced VMM circuit.

(4.5). On the other hand, these variations of $\pm V_1$ represent a break of symmetry, so second-order effects are not expected to be as well canceled as in the case where charge injection is well controlled.

Compute-mode charge injection: When computing, the voltages on the $\pm V_2$ terminals change constantly. The equivalent capacitive circuit in this case is that of Fig. 4.8(d), resulting

$$\Delta V_1 = \frac{C_1.\Delta V_X + C_2.\Delta V_Y}{C + C_1 + C_2} . \tag{4.10}$$

Recall, however, that M1=M2 and that the input signals are symmetric (i.e., $\pm V_2$). Therefore, $C_1 = C_2$ and $\Delta V_X = -\Delta V_Y$, resulting $\Delta V_1 = 0$ in (4.10). Hence, the internal charge of the system is not affected by the computations.

4.2.5 Experimental Results

We describe in this section experimental measurements realized on conventional and symmetric VMM cells constructed on a 2μm CMOS chip, with transistors of size 10/8 and a measured β of approximately $34\mu A/V^2$.

Fig. 4.9(a) shows two sets of measurements, one realized on a conventional VMM circuit and another realized on a symmetric one. The current of the latter was divided by 4 in order to provide a direct comparison. The solid line represents equation (4.3). As can be seen, the symmetric circuit has a superior accuracy, as expected, being the discrepancies hardly noticeable graphically. Since all second-order effects not considered in the transistor model are non-symmetric, the symmetric behavior of the second cell automatically compensates for such errors, what obviously does not occur in the other circuit. The measurements on the symmetric cell are further illustrated in Fig.

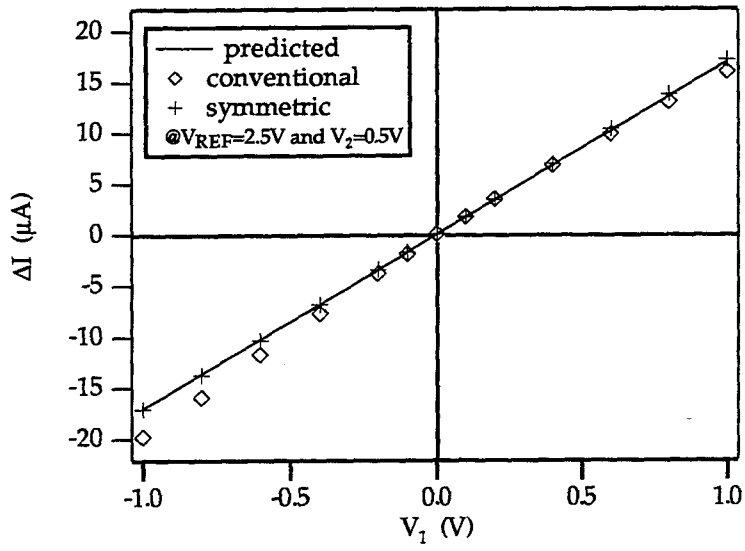


Fig. 4.9: Experimental measurements realized on conventional and symmetric VMM cells constructed on a 2μm CMOS chip, with transistors of size 10/8 and a measured β of approximately 34μA/V2. The solid line is given by equation (4.3). The current of the latter was divided by 4 in order to provide a direct comparison. As can be seen, the symmetric circuit has superior accuracy, being the discrepancies hardly noticeable graphically, as predicted.

4.10, which shows two sets of results (for V_2 =0.5V and 1.0V). Here the solid line (predicted output) is given by equation (4.5).

Finally, we show, in Fig. 4.11, experimental results obtained from a complete N=16 VMM array. The multipliers are those of Fig. 4.6, and the dashed-line option was adopted. Since the computing transistors are quite large (10x8) for the capacitors used in the implementation (30x30), a significant amount of charge injection is expected, thus affecting the symmetry of the current components, as described above. This will cause the cells to present an error in between that of the balanced and that of the non-balanced cell. This expected behavior is verified in the measurements plotted in Fig. 4.11, where the solid line represents equation (4.5) once again.

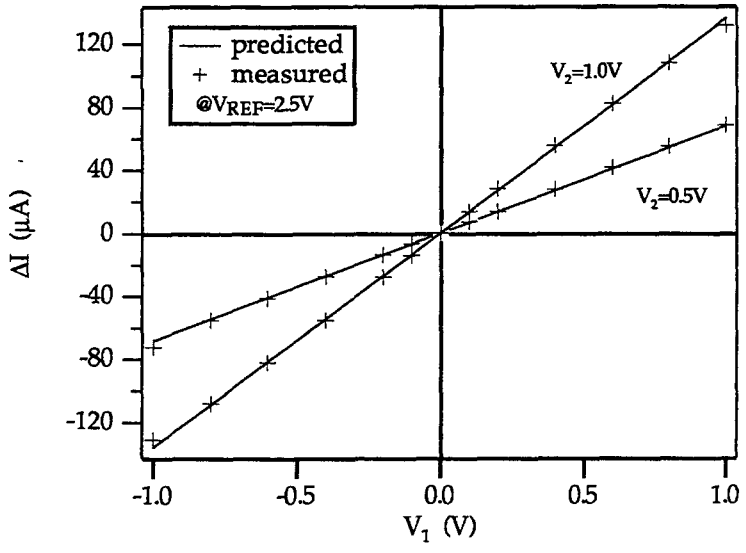


Fig. 4.10: Further measurements realized on a symmetric VMM cell (for V_2 =0.5V and 1.0V). The solid line is given by equation (4.5).

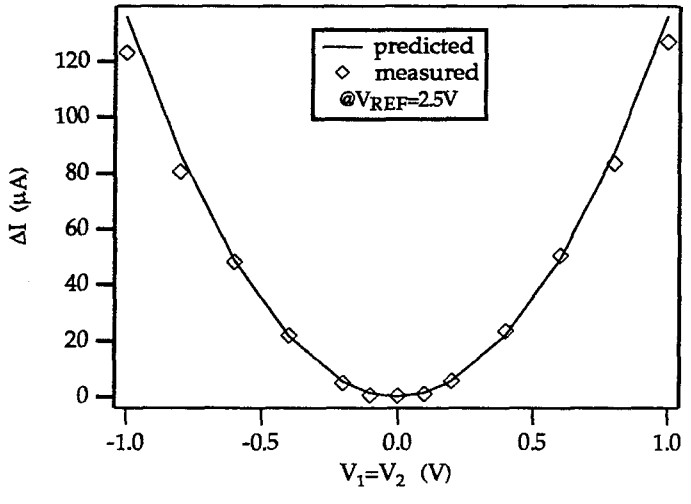


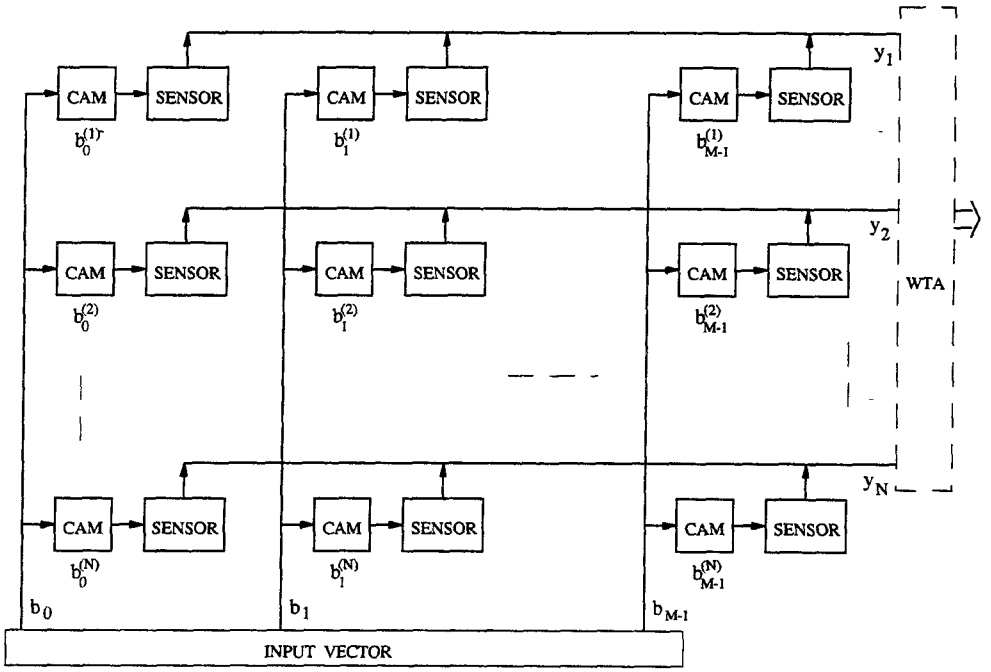
Fig. 4.11: Experimental results obtained from a complete N=16 VMM array. The multipliers are those of Fig. 4.6, and the dashed-line option was adopted. Since the computing transistors are quite large (10x8) for the capacitors used in the implementation (30x30), a significant amount of charge injection is expected, thus affecting the symmetry of the current components and causing the cells to present an error in between that of the balanced and that of the non-balanced cell. This expected behavior is verified in the measurements plotted above.

4.3 WTA-Based Hamming Networks

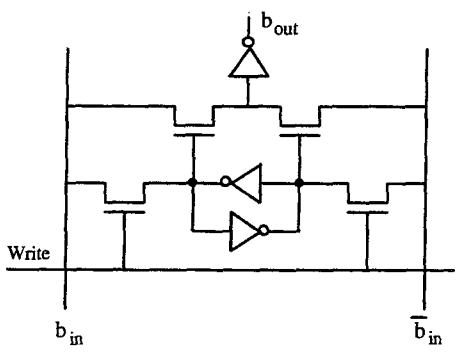
Hamming networks constitute one of the most basic parallel signal processing systems. As discussed earlier, they perform a (in principle) non-linear function which consists of, given an input vector $\mathbf{x}$, with components $x_i = \{0,1\}$ $(i = 1,2,...,M)$, finding, within a set $\mathbf{W} = \{\mathbf{w}^{(1)}, \mathbf{w}^{(2)}, ..., \mathbf{w}^{(N)}\}$ of (stored) vectors of the same kind, the one that best matches $\mathbf{x}$. For analytical purposes, the vector components can be replaced by $\{-1,1\}$, in which case they all have the same Euclidean length $\sqrt{M}$. Therefore, by "best matching" one may refer to the *smallest Hamming distance*, or to the *LMS distance*, or to the *least absolute distance*, or to the *largest inner product*, which are all equivalent to saying that the best representation for $\mathbf{x}$ is the vector $\mathbf{w} \in \mathbf{W}$ that has the *smallest spatial angle* with respect to $\mathbf{x}$. This variety of analytical interpretations is certainly useful from the algorithm implementation point of view, for it allows a larger number of hardware alternatives for solving the same problem. For instance, a vector-matrix multiplier, which is a linear system, can be used to implement this non-linear function if the inner-product approach is chosen. In this section we discuss MOS implementations that are based on the Hamming-distance metric and are constructed directly onto winner-take-all circuits (Chapter 2). As a consequence, the systems are hybrid, having a digital section, which performs the XOR operations, and an analog section, which includes a sensor/adder and the WTA unit.

4.3.1 Hamming Architecture and Input Sensors

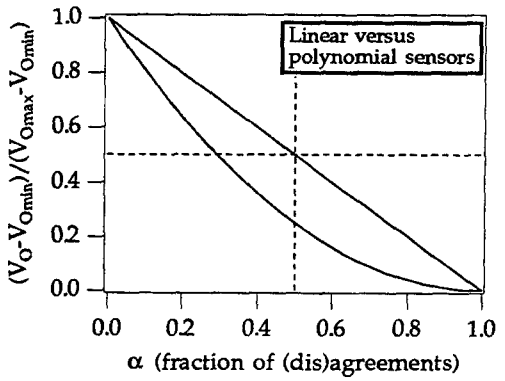
Fig. 4.12(a) shows a WTA-based Hamming network (HN) architecture, where the input vector is represented by bits $b_0, b_1, ..., b_{M-1}$ and the templates (codewords) by $b_0^{(i)}, b_1^{(i)}, ..., b_{M-1}^{(i)}$ $(i = 1,2,...,N)$. The content-addressable memory (CAM) circuits are responsible for the vector storage and for the XOR operation. A conventional CAM circuit is shown in Fig. 4.12(b), where the lower pair of transistors is used for writing the weights to the static latch formed by the pair of inverters, and the upper pair of transistors is used to perform the XOR operation between the input



(a)



(b)



(c)

Fig. 4.12: (a) WTA-based Hamming network architecture, where $y_i = f(\sum_j b_j \oplus b_j^{(i)})$; (b) Example of a content-addressable memory (CAM) circuit; (c) Linear versus polynomial sensors.

and the stored bits. The sensors, placed at the output of the CAM circuits, are analog units responsible for generating a combined output signal which is proportional to the row's total Hamming distance. This sensor/adder circuit can be linear or, more commonly, polynomial, as depicted in Fig. 4.12(c). A linear sensor can be easily obtained by means of M switched current sources connected to a common ohmic load. However, networks that utilize exclusively MOS transistors are normally of polynomial behavior, due to the quadratic nature of the current-voltage relationship. This polynomial behavior is highly desirable in systems where the best-matching vector is expected to have a small Hamming distance (say $d < M/2$), for the circuit is then more sensitive (i.e., presents higher gain) in the range of interest, as can be easily verified in Fig. 4.12(c). On the other hand, if the space is a degenerate one (i.e., the internal vectors are all located in a "corner" of the Euclidean space rather than having some kind of uniformity), and additionally it is admitted that the input vector may be anywhere in the space, then d can be very large. For instance, if the minimum Hamming distance between the N codewords, each of length M, as defined above, is d_{code}, then the Hamming distance between **x** and its best matching vector can be as large as

$$d_{\max} = M - D\,, \tag{4.11}$$

where

$$D = \{\min\ i \mid \sum_{i} \binom{M}{i.d_{code}} \geq N\}\,. \tag{4.12}$$

Since (4.12) grows almost exponentially with i, D turns out to be quite small even when d_{code} is relatively large, causing $d_{\max}$ in (4.11) to be large. In this kind of system the linear-type sensor is preferred, but this is fortunately not the case in general.

4.3.2 Moderate-Resolution HN

We start our discussion on WTA-based HN implementations by presenting a very simple $O(N^2)$ system, which is depicted in Fig. 4.13. Each row sensor is composed of M p-type pull-up transistors, whose input bits, denoted by H/L (high/low), are generated by individual CAM circuits. In this case, only transistors with L on the gate will be ON (generally in triode mode), whereas all the others will be OFF. m_i represents the number of transistors in the ON state in row i, where $m_i \leq M$ and $i = 1,2,...,N$. These sensors are polynomial and are directly connected to an analog grid of n-type feedback transistors, which perform the winner-take-all function. Although

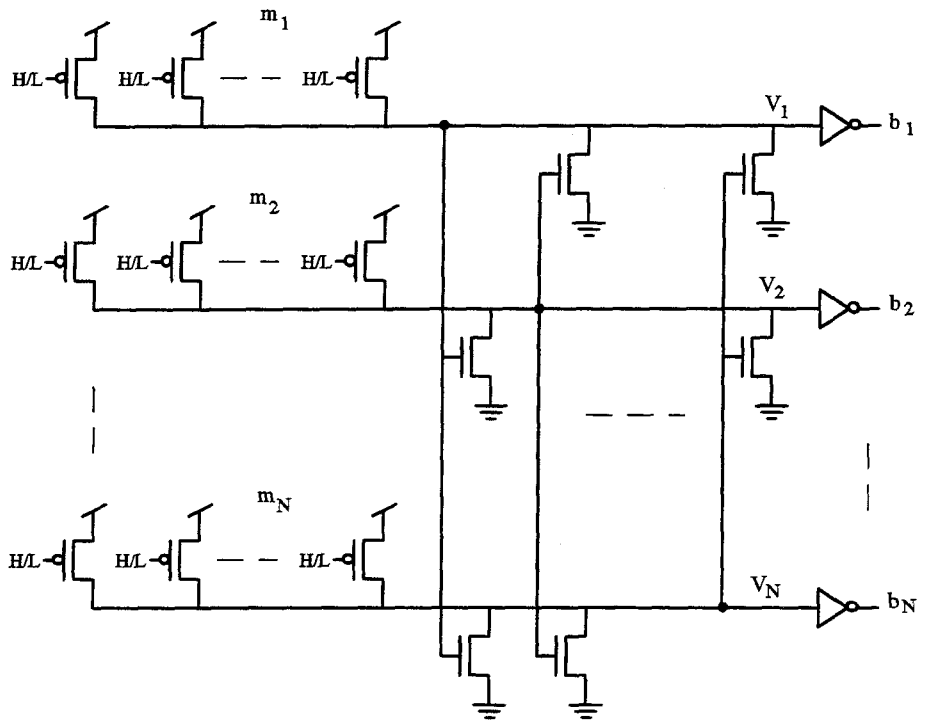


Fig. 4.13: A moderate-resolution Hamming circuit. H/L (high/low) are binary values originated by the individual CAM cells, and $m_i \leq M$ represents the number of 0V bits in row i (i=1, 2,..., N). Voltages V_i form an internal positive-feedback analog grid, which is converted to binary output signals b_i by the output inverters. Biased-type CMOS inverters are preferred over common-gate in this kind of implementation.

this network falls in the category discussed in Sec. 2.3.3, its simplicity and efficient output enlargement procedure are attractive when the vector length (M) is relatively small (typically under 64 bits); moreover, its analysis is helpful in understanding more sophisticated configurations, like the one described in Sec. 4.3.3. The output enlargement is related to the need of having the internal voltages ($V_1, V_2, ..., V_N$ in Fig. 4.13) well apart in order for the output inverters to be able to convert them to digital values unmistakably. In the circuits introduced in Chapter 2, as well as in those presented here, the enlargement is achieved in an elegant way, by means of the appropriate design of the transistor parameter ratios, thus avoiding popular approaches that require power switching and storage capacitors [Cilingiroglu 93][Yuping 93], which are inevitably of relatively low accuracy.

In analyzing the circuit of Fig. 4.13 we adopt the same approach used in Chapter 2. It is easy to verify that the nMOS transistors can operate in either saturation or triode mode, depending on the internally generated feedback voltages (which depend on m_i), being therefore governed by a relatively complex set of equations. There are, however, two regions of operation which are of main interest, being one for the case when all voltages are very close to each other, and another for the case when they are well apart and full enlargement can thus be achieved. Then, considering initially the condition $m_1 = m_2 = ... = m_N \equiv m$, and observing that the n-type transistors are in saturation and the p-type are in triode mode, we obtain that $V_1 = V_2 = ... = V_N \equiv V_=$ is given by

$$V_= = V_T + \frac{V_{DD} - V_T}{\sqrt{1 + k/\alpha}}, \tag{4.13}$$

where $\alpha = m/M$ represents the fraction of agreements (or disagreements, depending on whether the CAM circuits are of $\overline{\text{XOR}}$- or XOR-type, respectively - notice that the circuit of Fig. 4.12(b) corresponds to the latter) between $\mathbf{x}$ and the corresponding stored vector, and $k = (N\beta_1)/(M\beta_2)$ represents the transistor parameter ratio (β_1 for the n-type transistor and β_2 for the p-type, being

$\beta = \mu C_{OX} W / L$, as usual); also, $V_{TN} \approx V_{TP} \equiv V_T$ for simplicity. If we now let one of the m's grow, then the corresponding analog output will grow too. Say that $m_2 = ... = m_N \equiv m$ and $m_1 = m + d$, where $d \geq 1$; then $V_1 > V_2 = ... = V_N \equiv V_2$. If $V_1 \leq V_2 + V_T$, all nMOS transistors are still in saturation and we obtain that

$$V_1 \approx V_T + (V_{DD} - V_T)\sqrt{\frac{1+\frac{k}{\alpha}(1-x)}{1+\frac{k}{\alpha}(1-\frac{k}{\alpha N}x)}} \quad \text{and} \tag{4.14}$$

$$V_2 \approx V_T + (V_{DD} - V_T)\sqrt{\frac{1-\frac{k}{\alpha N}}{1+\frac{k}{\alpha}(1-\frac{k}{\alpha N}x)}}, \tag{4.15}$$

where

$$x = \frac{m}{m_1} = \frac{\alpha M}{\alpha M + d} \tag{4.16}$$

is also a function of $\alpha = m / M$, the fraction of agreement bits in rows 2 to N. As mentioned before, these equations are only valid for $V_1 \leq V_2 + V_T$, i.e., when the voltages are all close to each other (within $V_T \sim 1$V), in which case detecting the best match is most difficult. As indicated by (4.14)-(4.15), as α gets smaller (or larger k factors are used in the design) the voltages spread apart and detection becomes easier. For instance, the equations above show that the voltage separation is well above V_T when α decreases to

$$\alpha = \frac{k}{N}. \tag{4.17}$$

The other region of interest is that in which V_2 approaches V_T, for then all nMOS transistors of row 1 will tend to cut off, allowing V_1 to grow (in floating mode) towards its maximum value, V_{DD}. This wide voltage enlargement occurs when

$$\frac{k}{\alpha N} \geq \frac{(V_{DD}-V_T)^2}{(2V_{DD}-3V_T)V_T}, \tag{4.18}$$

or, equivalently, if

$$m \leq \frac{\beta_1}{\beta_2}\frac{(2V_{DD}-3V_T)V_T}{(V_{DD}-V_T)^2}, \tag{4.19}$$

which, for typical values of V_{DD} and V_T, can be approximated as $m \leq 0.4\beta_1/\beta_2$.

From the discussion above we conclude that the operation of the network of Fig. 4.13 can be divided into three main regions, depending on the value of α, as depicted in Fig. 4.14. If α is close

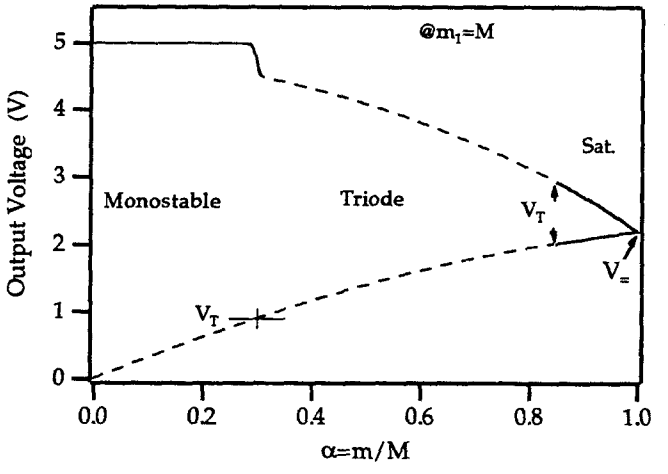


Fig. 4:14: Qualitative behavior of the network of Fig. 4.14, for m_1 fixed at its maximum value ($m_1 = M$). $\alpha = m/M$ represents the fraction of 0V bits in rows 2 to N ($m \equiv m_2 = m_3 = \ldots = m_N$, $m = 0,1,\ldots,M$). For large values of α all nMOS transistors are in saturation and the circuit is governed by equations (4.14)-(4.15) (solid lines), which converge to (4.13) as $\alpha \to 1$. When α decreases, V_1 and V_2 separate, causing one of the n-type transistors in rows 2 to N to operate in triode mode when this separation is greater than V_T. Decreasing α even further may cause V_2 to reach V_T, in which case all nMOS transistors of row 1 will tend to cutoff, causing full enlargement of the corresponding output voltage ($V_1 \to V_{DD}$).

to one (i.e., $m \rightarrow M$), then all nMOS transistors operate in saturation and the circuit behaves according to expressions (4.14)-(4.15), being reduced to (4.13) when $m = M$. For smaller values of α, such that $V_1 - V_2 > V_T$, one of the n-type transistors of each row, except the winner, will operate in triode mode. If α is decreased even further, such that m obeys (4.19), then the circuit operates in the monostable region. (Notice that, in the latter case, a preset transistor - not shown in Fig. 4.13 - is necessary, similar to that used in Fig. 2.7.)

Experimental Results: A large network (N=M=64), similar to that of Fig. 4.13, was fabricated using 2μm CMOS technology. The chip, shown in Fig. 4.15, is a complete (hybrid) system, including all the individual CAM circuits. Due to area constraints, transistor dimensions are relatively small (4μm on the smallest side). $k \approx 54$ was adopted in the design, with the purpose of

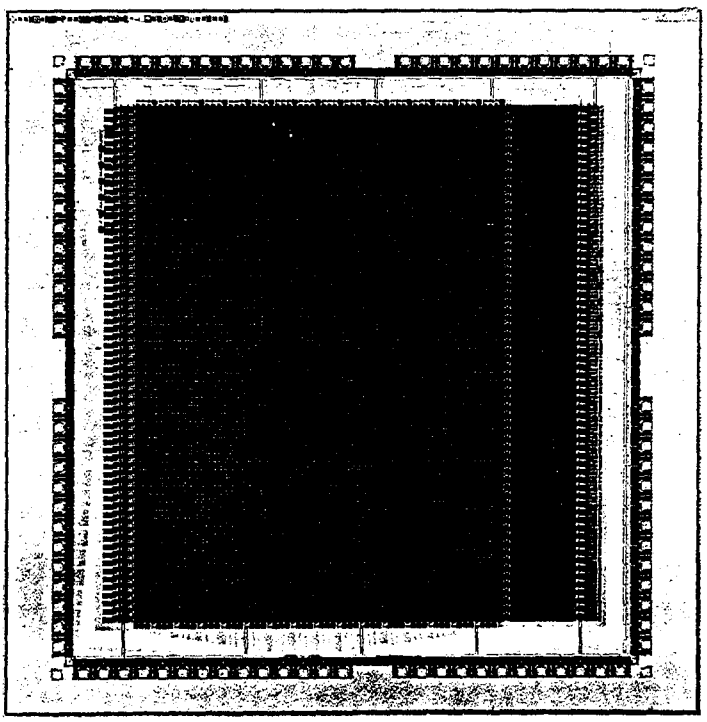


Fig. 4.15: Complete N=M=64 system, fabricated on a CMOS chip utilizing 2μm technology.

inspecting all three main regions of operation. The tests were performed for $V_{DD} = 5\,\mathrm{V}$, with $m_1 = M = 64$ and m decreasing progressively from this limit. In Fig. 4.16, the upper channel of the oscilloscope indicates the position of the expected winner, whereas the lower (time multiplexed) channel shows the 64 analog output voltages. In Fig. 4.16(a), $m = m_1$, so no winner is expected ($V_1 = V_2 \equiv V_=$). In Fig. 4.16(b), m=62, and the circuit is governed by (4.14)-(4.15), with V_1 already well separated from V_2. In Fig. 4.16(c), m=60, and the separation is larger than V_T. Finally, in Fig. 4.16(d), m=24, causing $V_2 \approx 0.8\,\mathrm{V}$ and $V_1 \approx 5\,\mathrm{V}$ (monostability), as predicted by (4.19).

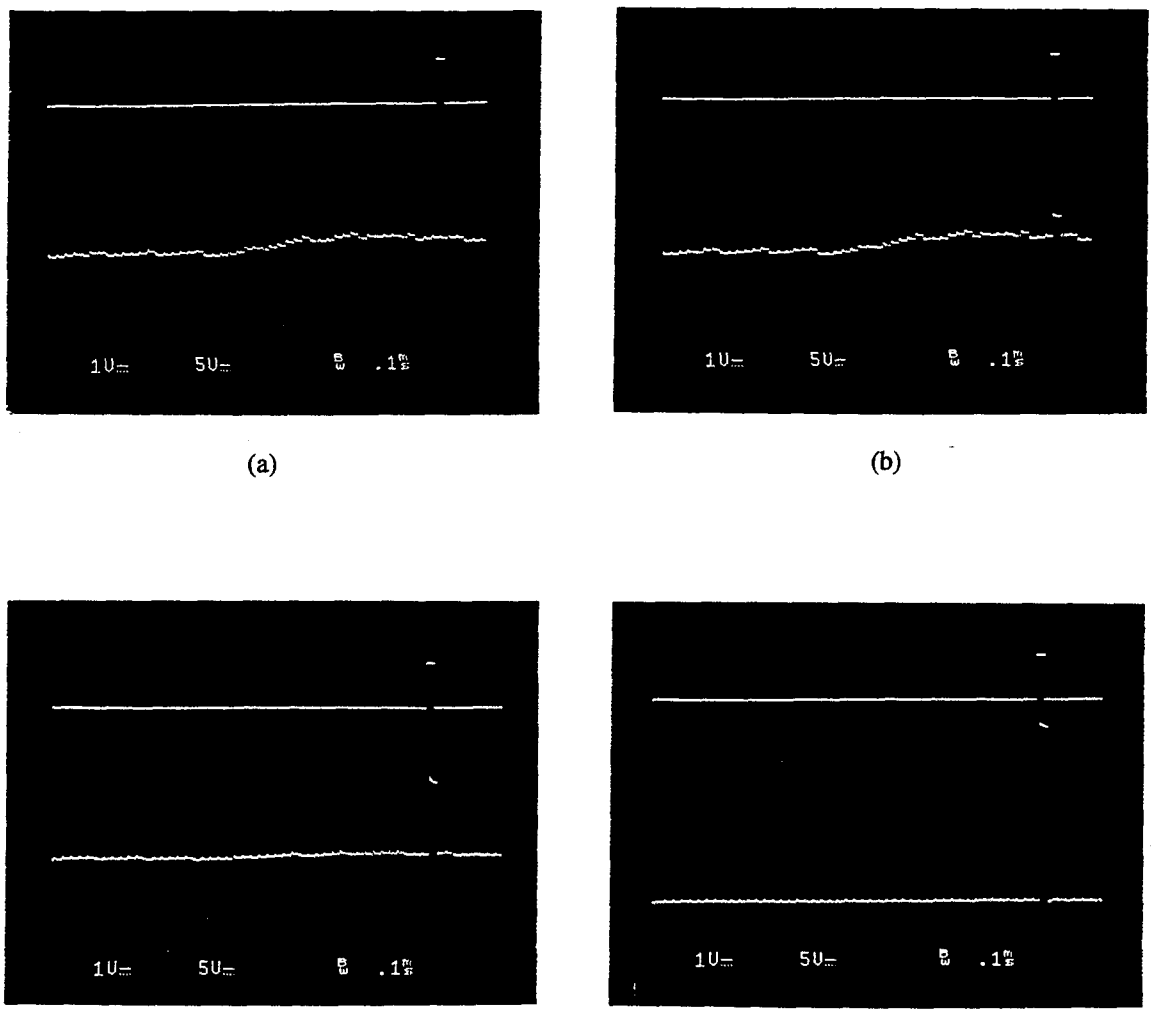
4.3.3 Global Feedback HN

We make use now of a more sophisticated WTA circuit, which was introduced and examined in detail in Sec. 2.3, in order to produce a high-resolution Hamming network. Such a system can be derived directly from the winner-take-all circuit of Fig. 2.7 by replacing transistor M2 of each row by a sensor similar to that described in the previous section, but using n-type transistors, as depicted in Fig. 4.17. Also, in order to provide a higher gain, the sensor operates in saturation rather than in triode mode; thus, V_H and V_L are adopted as the high and low bit voltages instead of VDD and GND. Now, m_i represents the number of transistors at level V_H in sensor i ($i = 1, 2, \ldots N$), whereas $M - m_i$ indicates the number of transistors at level V_L.

With the circuit modifications described above, and using strong inversion current equations, we verify that, for the initial condition $m_1 = m_2 = \ldots = m_N \equiv m$, the common-line and output voltages are given by

$$V_{C=} = \alpha V_H + (1-\alpha)V_L - V_T - \sqrt{k_1^2 (V_B - V_T)^2 - \alpha(1-\alpha)(V_H - V_L)^2} \quad \text{and} \tag{4.20}$$

$$V_{O=} = V_{DD} - V_T - k_2 (V_B - V_T)\,, \tag{4.21}$$



(a) (b)

(c) (d)

Fig. 4.16: Experimental results from the chip of Fig. 4.15. The circuit was designed with relatively small transistor dimensions (4μm on the smallest side), causing a modest parameter matching. The transistor ratio adopted in the design was $k \approx 54$. In the experiments above, m_1 was kept fixed at 64, while m was decreased progressively from 64. In (a), $m = m_1$, so no winner is present. The irregularities of $V_=$ were expected, for reasons discussed in Chapter 2. The average voltage is $V_= \approx 1.8\,\mathrm{V}$ (against 1.5V of eq. 4.13). The upper channel of the oscilloscope indicates the position of the expected winner when $m < m_1$. In (b), $m = 62$ and the circuit is governed by (4.14)-(4.15), with V_1 already well separated. In (c), $m = 60$, and the separation is larger than V_T. Finally, in (c), $m = 24$, causing $V_2 \approx 0.8\,\mathrm{V}$ and $V_1 \approx 5\,\mathrm{V}$, being therefore in good agreement with (4.19), which predicts monostability for $m < 22$ in this case.

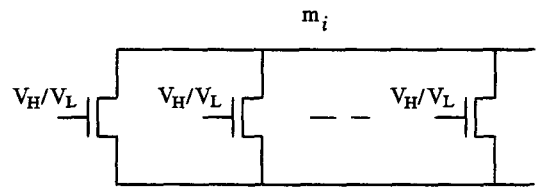


Fig. 4.17: A high-resolution Hamming network can be directly obtained from the WTA circuit of Fig. 2.7 by replacing the transistor M2 of each row by M "binarily" controlled transistors as shown above. Here, m_i $(m_i \leq M)$ represents the number of high-voltage bits in row i (i=1,2,...,N). However, to ensure a higher gain, the low and high bit voltages are V_L and V_H rather than GND and VDD.

where $\alpha = m/M$ and, analogous to Sec. 2.3, $k_1 = \sqrt{\beta_1 / M\beta_2}$ and $k_2 = \sqrt{\beta_1/(N-1)\beta_3}$. For the saturation condition to be satisfied, i.e., $V_{C=} \geq V_B - V_T$ and $V_{O=} \geq V_H - V_T$, we obtain from (4.20) and (4.21) that the transistor parameter ratios must obey

$$k_1 \leq \frac{\sqrt{(V_B - V_T)^2 + \alpha(V_H - V_L)(V_H + V_L - 2V_B)}}{V_B - V_T} \quad \text{and} \tag{4.22}$$

$$k_2 \leq \frac{V_{DD} - V_H}{V_B - V_T} \, . \tag{4.23}$$

Notice, however, that (4.20) reduces to

$$k_1 \leq \frac{V_L - V_B}{V_B - V_T} \tag{4.24}$$

as $\alpha \rightarrow 0$, being therefore (4.24) and (4.23) in full agreement with (2.8a) and (2.8b), respectively, as expected. Thus the quantitative analysis of this network reduces almost completely to that presented in Sec. 2.3, being the output enlargement governed by equation (2.11). As an example,

let us consider that $V_L = 2.5\,\mathrm{V}$ and $V_H = 3.5\,\mathrm{V}$ are adopted on a HN of N=64 rows, with $V_{DD} = 5\,\mathrm{V}$, $V_B = 1.5\,\mathrm{V}$, and $V_T = 0.9\mathrm{V}$. We obtain from (4.24) and (4.23) that the transistor parameters must obey $k_1 \leq 1.67$ and $k_2 \leq 2.5$. Say that $k_1 = 1$ is adopted; then, if $k_2 < 0.36$ is used (2.11), the circuit will operate in the monostable regime and full enlargement will be immediately achieved.

4.4 Absolute Distance-Based Two-Dimensional Median Filters

As mentioned in Sec. 1.2 and Sec. 2.5.2, median filters are of interest to image processing due to their ability to remove impulsive noise. They constitute a particular case of rank filters in which the central element of the ordered set is to be selected. Performing the median function in a low time complexity fashion is a very challenging exercise from the viewpoint of hardware implementation. Digital realizations of this function can obviously not be of *O(1)*, even for 1D data streams. As illustrated in Fig. 4.18, digital implementations normally rely on threshold decompositions of the M-valued components of the data block (of size N=nxn) into M-1 binary strings of length N, which are then binarily filtered and later added together [Wendt 86][Yli-Harja 91][Murthy 92][Jarske 93], therefore requiring computation times that inevitably grow with filter parameters.

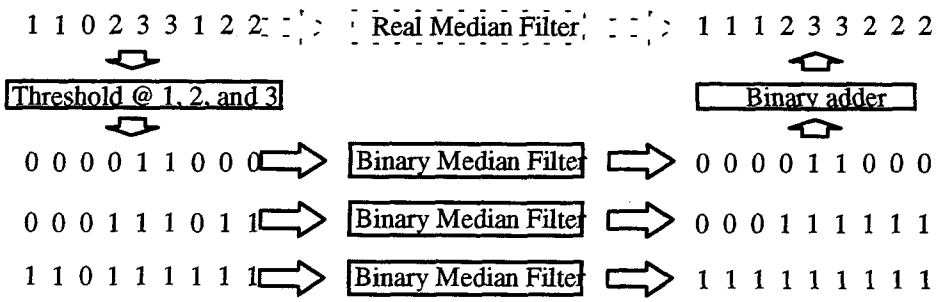


Fig. 4.18: Conventional median filter implementation, with 4-valued inputs and window of width 3 [Wendt 86].

In Sec. 2.5.2 we introduced a real-domain 1D rank filter of time complexity *O(1)*. In this section we introduce a 2D median filter which has also minimum time complexity. To do so, we observe that, given a real-valued set $\mathbf{X} = \{x_1, x_2, ..., x_N\}$, the median of $\mathbf{X}$ can be computed as

$$\text{med}(\mathbf{X}) = \min_j \{\sum_{i=1}^{N} |x_j - x_i|\} . \qquad (4.25)$$

Equation (4.25) indicates that an array of absolute-distance cells can be used to implement the median function directly and in a single clock cycle. Such a system is illustrated in Fig. 4.19. Notice that, contrary to most architectures presented so far, this network does not require any signal storage. As indicated in the inset of Fig. 4.19, each individual cell has two inputs and two outputs; one output transmits the highest input, whereas the other transmits the smallest. Then, by applying a control signal HI/LO (high/low), the outputs can be independently selected. While LO is active, the PRE (preset) signal precharges all rows to a reference voltage V_{PRE}. After PRE is released, HI is enabled, causing the global row bus to (capacitively) compute $y_j = \sum_{i=1}^{N} |x_j - x_i|$, $j = 1, 2, ..., N$. Then, by applying these signals to a winner-take-all circuit, the computation of (4.25) is completed.

Two examples of MOS absolute-distance cells are depicted in Fig. 4.20, both operating as voltage-followers, with outputs $V_{O1} \propto \max\{V_1, V_2\}$ and $V_{O2} \propto \min\{V_1, V_2\}$. In Fig. 4.20(a), a single transistor would suffice on each side of the upper branch, but two were used in order to provide symmetry with respect to the lower branch, such that $V_{O1} = V_{O2}$ when $V_1 = V_2$. This cell has very little power consumption, since there is no bias current (one side of each branch is normally OFF), but an additional transistor (not shown) is needed to provide a discharge path for each of the central nodes. The circuit of Fig. 4.20(b), introduced in [Cauwenberghs 94], has a smaller transistor count and operates in biased mode, with a discharge path readily available. The performance of either cell can be estimated from our discussion on voltage-follower (VF) circuits

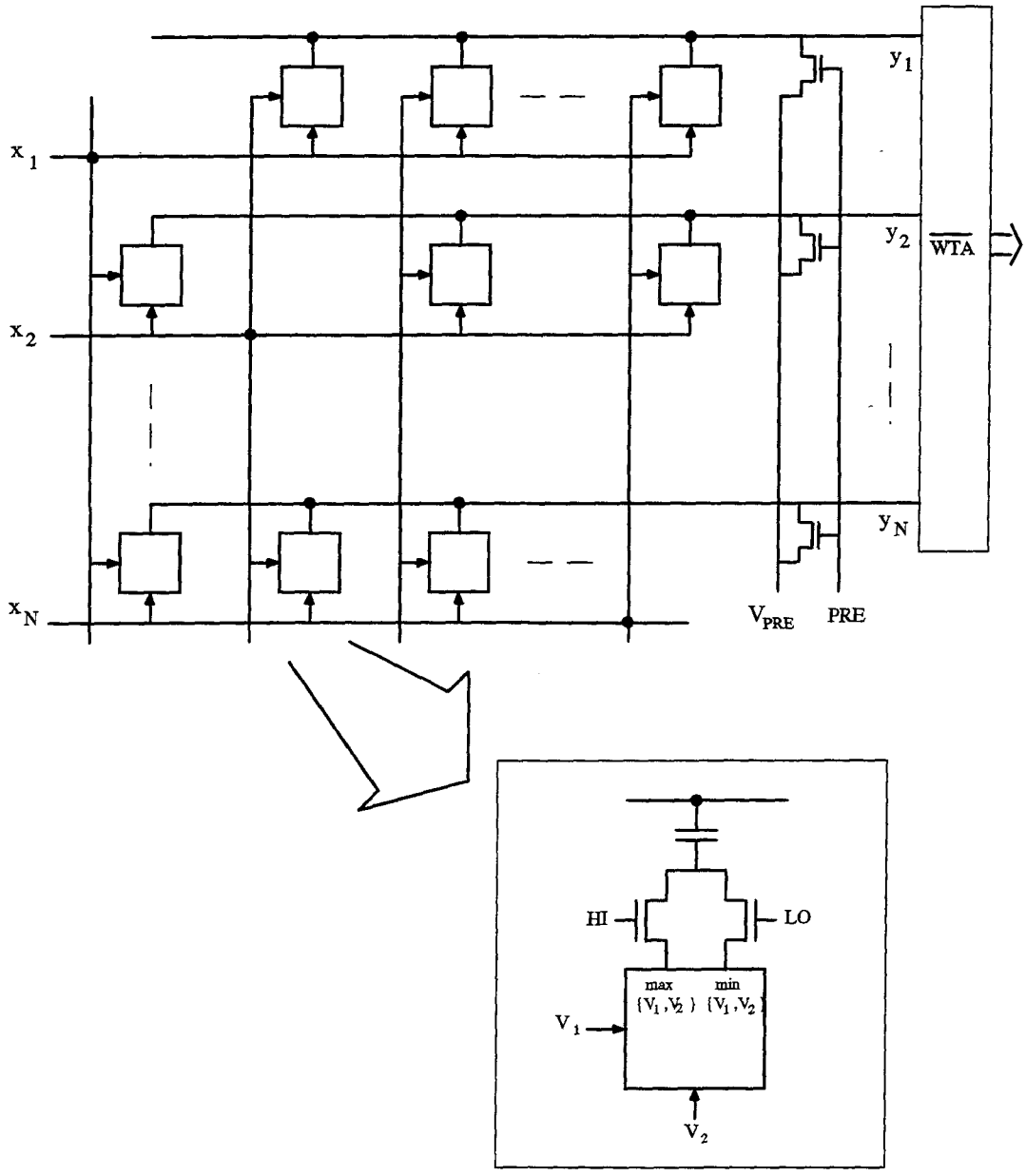
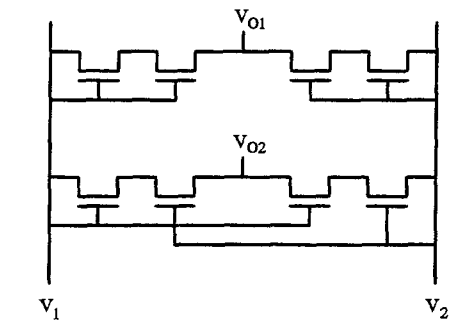
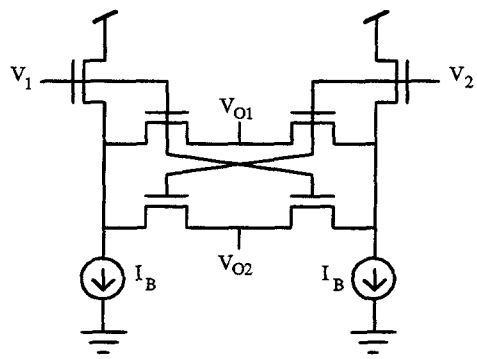


Fig. 4.19: Absolute distance-based 2D median filter of time complexity *O(1)*. Each row computes $y_j = \sum_i |x_j - x_i|$, being the smallest sum detected by the winner-take-all circuit.



(a)



(b)

Fig. 4.20: Two absolute-distance cell implementations of similar performance. Both operate as voltage-followers and fall in the category discussed in Sec. 2.2.1. Cell 1 (a) requires an additional transistor (not shown) at each output node in order to provide a discharge path, whereas cell 2 (b) corresponds to a pair of transistor-connected conventional VF circuits.

presented in Sec. 2.2. Indeed, that discussion is very helpful in understanding many analog processors, for many depend, on one way or the other, on VF circuits. (Commercially available voltage comparators and opamps are good examples.) A look at Fig. 4.20(a)-(b) immediately reveals that both circuits fall in the sub-category discussed in Sec. 2.2.1, being therefore expected to present a moderate system resolution. The performance of this kind of implementation can be aggravated by the *body effect* (Appendix A) in case the transistors are not constructed inside appropriate wells (i.e., when the source-body voltage is $V_{SB} \neq 0\,V$). This effect causes the threshold voltage V_T to grow with the source-body bias, resulting a non-symmetric output performance. For instance, Fig. 4.21 shows experimental measurements realized on both circuits, with transistors of size 6/4 and no source-body connection. The observed body effect was typical for MOSIS parameters (Table 1.1), causing a variation of V_T over 0.5V for V_{SB} between 0V and 3V (though it is partially canceled by the differential computation $V_{O1} - V_{O2}$).

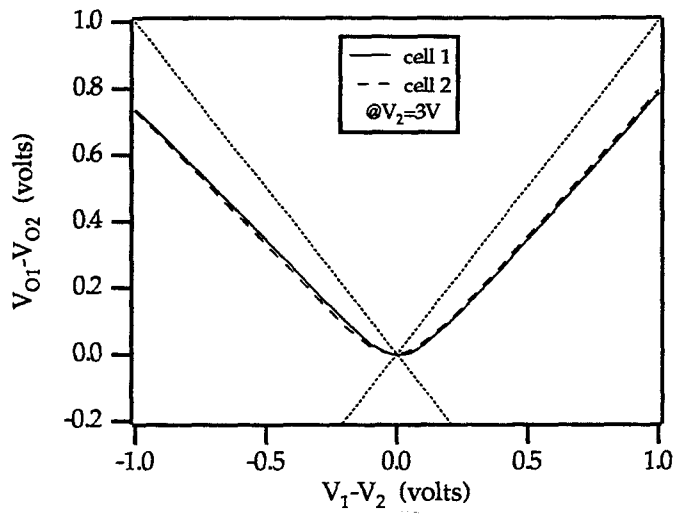


Fig. 4.21: Experimental results from both cells of Fig. 4.19, with transistors of size 6/4 and $V_{SB} \neq 0\,V$. The output voltages are very similar and, since a body bias was allowed to exist, a significant body effect can be observed in both circuits.

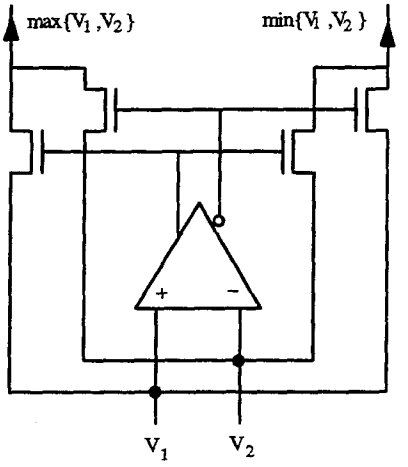


Fig. 4.22: High-resolution absolute-distance cell. The comparator represents a modified-VF circuit, which can be any of those discussed in Sec. 2.2.2 or similar. The circuit operates as a pointer rather than as a voltage transmitter.

As in Sec. 2.2, higher resolution cells can be obtained with the use of modified-VF circuits. These modified cells operate as pointers to the winner rather than transmitting the signal itself (Sec. 2.2.2). This kind of system is less sensitive to the body effect, causing the total silicon area to be sometimes even smaller than that of Fig. 4.20. As shown in Chapter 2, it also presents higher gain, offering therefore a highly improved system resolution. A typical absolute-distance cell configuration of this type is depicted in Fig. 4.22, which provides a higher quality min-max signal for capacitive computation.

4.5 Comments on Appendix B

With the advent of neural networks, there has been in recent years a growing effort towards the development of analog-storage media that are compact, easily programmable, error robust, and are capable of retaining the analog information in a long-term fashion [Hochet 90][Vittoz 91][Lee 91][Cauwenberghs 93]. Massive VLSI implementations of analog memories are indeed one of the

greatest challenges faced in the realization of neural networks and other high-density parallel signal processing systems. While the simplest form of analog memory consists of storing the information as a voltage level on a capacitor, the drift due to current leakage and noise limits this approach to short-term computations. Floating-gate techniques, on the other hand, allow for long-term storage, but lack a write access which is system compatible.

Although the basic structure of programmable analog-memory circuits is somewhat different from those presented in this and in the preceding chapters, the many physical aspects governing their operation offer a rich opportunity of further understanding the potentialities and limitations of analog silicon systems, especially regarding the *charge-injection* problem. For this reason, we have included, in Appendix B, a discussion on the subject, which is illustrated through a MOS analog memory implementation [Pedroni 94c]. This *N*-valued memory was developed under the premises of being small (high integration density), self-contained (no external communications channels needed), and equipped with some kind of error-correction capability.

Appendix A

MOS Basics

We present in this appendix a brief review on the main physical aspects governing the operation and modeling of the MOS transistor, especially those that appear more frequently in this dissertation. For a more detailed discussion the reader may refer, for example, to [Sze 81][Tsividis 87][Mead 89][Pierret 89][Weste 93].

A.1 MOS Capacitor

Probably the most efficient way of describing the operation of the MOS (metal-oxide-semiconductor) transistor, a four-terminal structure, is by describing first the MOS capacitor, a two-terminal structure. A MOS capacitor is presented in Fig. A.1. As can be seen, it is composed of a field plate (gate), made of metal (Al) or, more generally, polycrystalline silicon (poly), which is

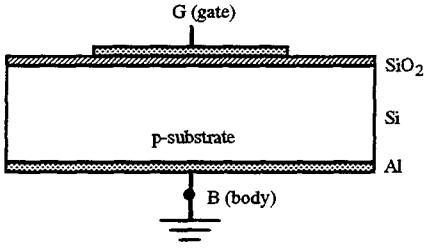


Fig. A.1: MOS capacitor structure.

separated from the lightly doped p- or n-type silicon substrate by an insulating oxide layer (SiO_2). The substrate (body) is externally connected through a metallic ohmic contact placed at the bottom of the structure. The oxide thickness, t_{ox}, is typically in the 200-400 A° range for the processes referred to in our experiments (Table 1.1), and the substrate doping concentration (N_A or N_D) is typically 10^{15}-10^{16} atoms/cm^{-3}.

Ideal MOS Capacitor

In order to facilitate the description of the MOS capacitor of Fig. A.1, we assume initially that it is an ideal structure. In such a case, there are no charge centers either in the oxide or at the oxide-semiconductor interface, the substrate is uniformly doped, no current flows through the insulator under any bias condition, the workfunction (energy difference from the vacuum level, E_{vac} , to the Fermi level, E_F) of the gate metal and of the semiconductor are equal ($\Phi_M=\Phi_S$), and the body ohmic contact (semiconductor-metal contact at the bottom of Fig. A.1) has zero Schottky barrier.

The workfunction of the metal is a fixed property of the material, for since the metal plate can be considered equipotential its E_F is the same at any position inside the material. The Fermi level of the semiconductor, on the other hand, varies with doping and band bending. For this reason, Φ_S is normally expressed in terms of the electron affinity χ (the energy difference between the vacuum level and the conduction level, E_C), which is independent from doping or bias voltage, being therefore fixed for a given semiconductor. Hence, $\Phi_S = \chi+(E_C-E_E)_{bulk}$, where the "bulk" designation refers to the substrate volume opposite to the oxide-semiconductor interface (which presents zero electric field).

Fig. A.2(a) shows the individual energy band diagrams of the materials that compose a basic ideal MOS structure: metal (say Al), insulator (SiO_2), and substrate (p-type in this case; notice that E_F is in between the intrinsic energy level, E_i, and the valence level, E_V). When the three materials are brought into intimate contact, with zero volts applied to the structure, their Fermi energy levels must align, for there is no current flow. Then the only effect of the oxide is to reduce slightly the

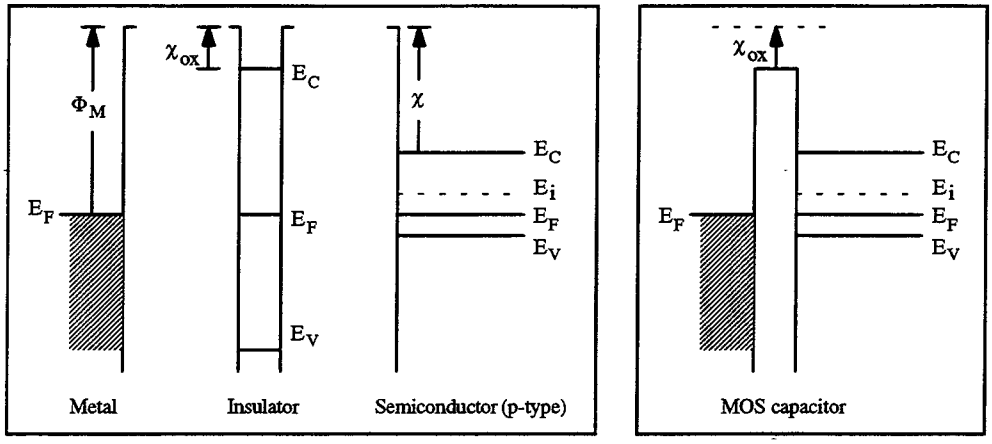


Fig. A.2: MOS capacitor energy band diagrams: (a) individual materials; (b) materials in intimate contact, in equilibrium (V_{GB}=0V).

energy barrier between the metal and the semiconductor (by an amount χ_{ox}, as indicated in Fig. A.2). The resulting *equilibrium* energy band diagram is depicted in Fig. A.2(b). Since the electric field inside the structure is zero for $V_{GB} = 0\,\mathrm{V}$, the *flat-band* voltage of an ideal MOS capacitor is said to be V_{FB}=0V.

We consider now what happens when V_{GB} varies. Say that the body voltage is kept fixed and the gate voltage is changed. If $V_{GB} < 0\,\mathrm{V}$, E_F of the metal is displaced upward by an amount -qV_{GB}. Since E_F of the semiconductor is still the same (and is level for no current flows through the structure), it causes the other p-substrate bands to bend upward by the same amount. This is illustrated in Fig. A.3(a). Consequently, the difference E_i-E_F grows, causing the hole concentration $p = n_i \exp[(E_i - E_F)/kT]$, where $n_i \sim 10^{10}\mathrm{cm}^{-3}$ is the intrinsic carrier density, to also increase at the semiconductor inner surface. In other words, applying $V_{GB} < 0\,\mathrm{V}$ causes *additional* majority carriers (holes in this case) to accumulate at the silicon surface. For obvious reasons, this regime of operation is referred to as *accumulation*.

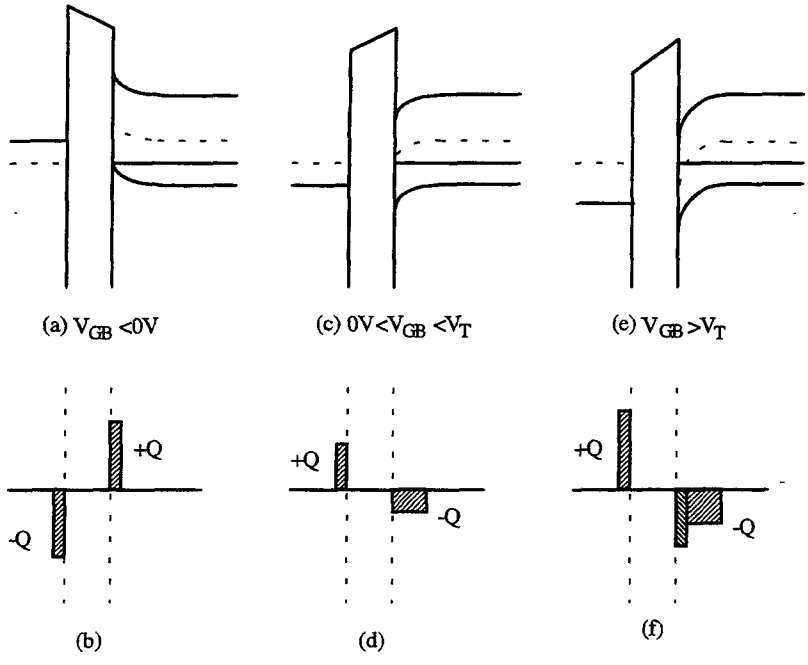


Fig. A.3: Energy band diagrams and charge density diagrams of an ideal two-terminal MOS structure as a function of the applied voltage: (a)-(b) *accumulation* mode (V_{GB}<0V); (c)-(d) *depletion* mode (0V<V_{GB}< V_T); (e)-(f) *inversion* mode (V_{GB}≥V_T).

An alternative way of describing the microscopic behavior of semiconductors is by means of charge-density diagrams, as shown in the lower part of Fig. A.3. In the accumulation mode described above, a negative voltage $-Q$ is applied to the gate. This charge causes a $+Q$ charge of *accumulated* holes to appear in the substrate. The accumulation mode can then be described as in Fig. A.3(b).

Let us consider now the case of $V_{GB} > 0$ V. When a positive voltage V_{GB} is applied to the metal, it causes its Fermi level to be displaced downward by an amount $-qV_{GB}$ with respect to its equilibrium condition. As a consequence, the energy bands bend downward on the semiconductor side. This condition is depicted in Fig. A.3(c). If V_{GB} is small, the only significant effect of this positive voltage is the repulsion of holes from the region immediately below the gate, creating therefore a region which is *depleted* of majority carriers. This mode of operation is known as

depletion. A representation of this mode of operation in terms of the charge density is presented in Fig. A.3(d).

With the growth of V_{GB}, electrons (minority carriers in this case) are attracted to the semiconductor surface, forming a very thin negatively charged layer. The electron density $n = n_i \exp[(E_F - E_i)/kT]$ then grows systematically from less than n_i when $E_F < E_i$, to n_i when $E_F = E_i$, and to greater than n_i when E_i goes below E_F. When V_{GB} reaches a voltage such that $E_F - E_i|_{surface} = E_i|_{bulk} - E_F$, the electron concentration at the surface equals the acceptor impurity density, that is, $n|_{surface} = N_A$, after what an abundant layer of electrons is available at the semiconductor surface. The value of V_{GB} for which this occurs is called V_T, the threshold voltage of the MOS structure. The regime of operation for $V_{GB} \geq V_T$ is referred to as *inversion*. This mode of operation is illustrated in Figs. A.3(e)-(f). Indeed, the possibility of creating and controlling an inversion layer of charge carriers constitutes the basic principle for the construction of the MOS transistor.

Non-Ideal MOS Structure

Although the ideal structure described above is a good approximation to real MOS devices, it is important to be aware of the component non-idealities, for they might affect the device performance under particular conditions. In an actual MOS structure, net charge is inevitably present in the oxide and at the semiconductor-oxide interface, thus originating internal electric fields even if the applied gate voltage is zero volts. Moreover, contact potentials exist, and $\Phi_M \neq \Phi_S$. As a result, the actual flat-band voltage is $V_{FB} \neq 0V$.

The net charge in the oxide is due mainly to fixed and mobile (mostly positive) ions trapped within the oxide during the fabrication process. Since this charge density is approximately fixed and is low for high-quality processes, its effects on the device performance tend to be negligible. The (non-ideal) charge distribution tend therefore to be dominated by the charge traps localized at the semiconductor-oxide interface (also known as *surface states*). Although the origin of these interfacial charge centers has not been completely determined, probable causes are the incomplete

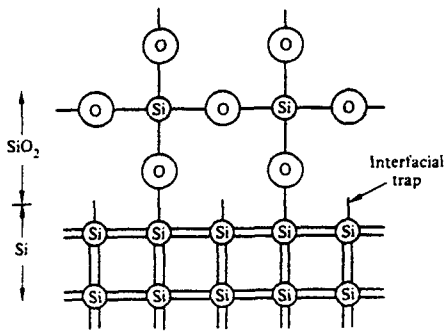


Fig. A.4: Incomplete silicon bonds at the Si-SiO_2 interface (after [Pierret 89]).

silicon bonds at the surface (Fig. A.4), fabrication defects, and high energy particles injected from the substrate. The overall behavior is similar to having allowed energy levels in the forbidden band gap, which are then filled-emptied according to the applied voltage, and is aggravated by fluctuations in the capture-release rate due to different time constants. The internal electric field created by these charges displaces the flat-band voltage from zero volts. A crude way of representing this effect quantitatively is by considering that the total charge Q_{trap} trapped at the semiconductor-oxide interface is approximately constant. In this case, the voltage drop across the oxide is

$$\phi_{ox} = Q_{trap}/C_{ox}\,, \tag{A.1}$$

where $C_{ox} = \varepsilon_{ox}/t_{ox}$ is the oxide capacitance (per unit area) and $\varepsilon_{ox} = 0.035\text{fF}/\mu\text{m}$ is the permitivity of SiO_2.

Another approximation used in the ideal MOS structure was the equality $\Phi_M = \Phi_S$, which certainly does not hold for actual materials. A more realistic case is presented in Fig. A.5. Fig. A.5(a) shows the energy band diagrams for the individual materials, where, contrary to Fig. A.2(a), the top ledges are not at the same level. When the materials are brought into intimate contact, once again

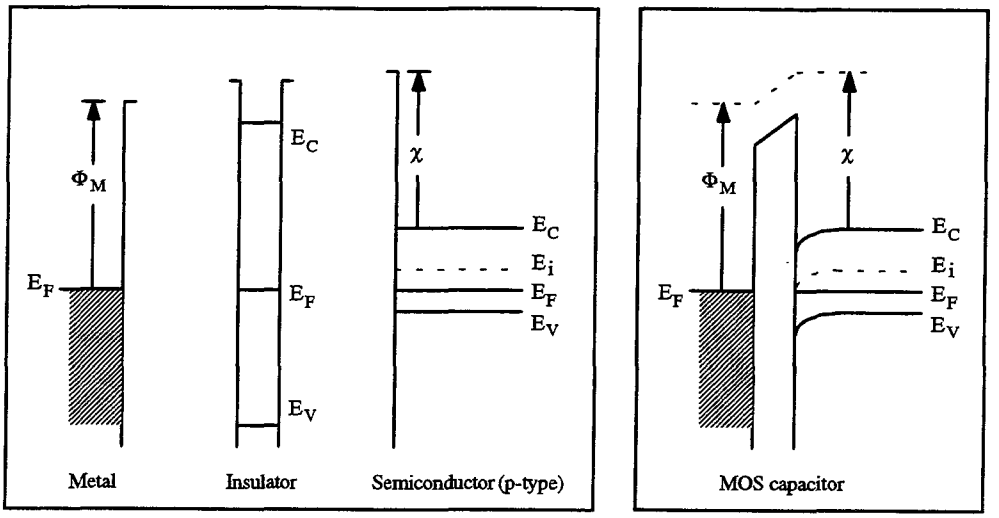


Fig. A.5: Energy band diagrams of a non-ideal MOS capacitor: (a) individual materials; (b) MOS structure.

their Fermi levels must align, resulting the energy distribution of Fig. A.5(b). As can be seen in Fig. A.5(b), an internal electric field (band bending) exists in the equilibrium condition ($V_{GB} = 0\,V$), hence also contributing for a flat-band voltage $V_{FB} \neq 0V$. The consequence of this effect is that a different value of V_{GB} is needed in order to produce the same charge conditions in the substrate. The difference between the new gate voltage and that of the ideal structure depends on the difference between the metal and semiconductor workfunctions, and is determined by

$$\phi_{MS}=(\Phi_M-\Phi_S)/q\,. \qquad (A.2)$$

(Notice that the voltage ϕ_{MS} is expressed in volts, whereas the energies Φ_M and Φ_S are given in electron volts.) As a result of $\phi_{MS} \neq 0V$, the threshold voltage changes (by the amount ϕ_{MS}). This can be easily verified in the MOS structure of Fig. A.5, which presents $\Phi_M<\Phi_S$. In this case, the energy bands are already bent downward with the system still in equilibrium conditions. Consequently, a smaller value of V_T is sufficient to create an inversion layer of same charge density in the substrate (compare Figs. A.5 and A.2).

Combining (A.1) and (A.2) results the following approximation for the flat-band voltage of a practical MOS system,

$$V_{FB} = \phi_{MS} + \phi_{ox} \ . \qquad \text{(A.3)}$$

Typical values of V_{FB} are in the $0V<|V_{FB}|<1V$ range.

MOS Structure Capacitance

The total capacitance (C_{GB}) of the structure of Fig. A.1 can be written as a series combination of two main terms, one due to the oxide (C_{ox}) and the other due to the substrate (C_{sub}). For a given process, the oxide term is constant and given by $C_{ox} = \varepsilon_{ox} / t_{ox}$. The other term, however, is not, for the charge distribution in the semiconductor varies with the applied voltage (recall the modes of operation: accumulation, depletion, and inversion). Due to the large charge densities at the semiconductor surface, the substrate capacitance is very large in the cases of strong accumulation or strong inversion, therefore causing the total capacitance to be determined by C_{ox} (i.e., C_{GB} approaches its maximum value, C_{ox}). On the other hand, for values of V_{GB} in the depletion region, the "thickness" of the substrate capacitor is large, implying that C_{sub} is small. As a result, the total capacitance is smaller in this region, peaking a little below V_T, where the depth of the depleted region is at its maximum. This behavior is illustrated qualitatively in Fig. A.6. For a typical CMOS process (say t_{ox}=400A°=0.04μm), and using ε_{ox}=0.035fF/μm, one obtains C_{ox}=0.88fF/μm^2. Notice that this value is of the same order of magnitude as those in Table 1.1.

A final remark regarding the MOS capacitance is that minority carriers, needed to form the inversion layer, do not have time to form if the applied signal (V_{GB}) is of high frequency (times in the milliseconds range are needed). In this case, the total capacitance stays at its low level, as indicated by the broken line in Fig. A.6. This is indeed the basic principle that allows the construction of charge-coupled devices (CCD's), as discussed in Chapter 3.

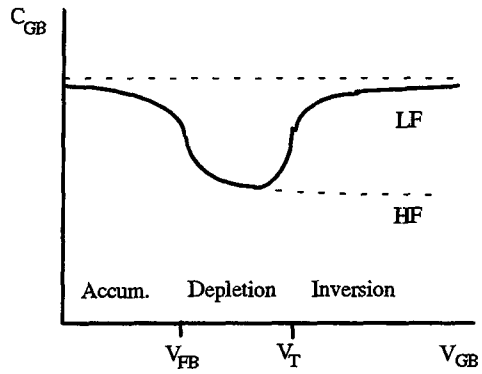


Fig. A.6: Capacitance of the two-terminal MOS structure as a function of the applied voltage.

A.2 MOS Transistor

The basic structure of a MOSFET (MOS field-effect transistor) is shown in Fig. A.7(a). Contrary to the two-terminal structure discussed in Sec. A.1, minority carriers, needed to form the inversion layer, are readily available from the source terminal, thus allowing high-frequency operation. The MOS transistor of Fig. A.7(a) is said to be of n-type (or n-channel) because the channel (inversion layer) between the drain and the source diffusions is of n-type (electrons). Analogously, a transistor is said to be of p-type (or p-channel) when the substrate (or well) is n-doped. Fig. A.7(c) shows a pair of CMOS (complementary MOS) integrated transistors in an n-well process, implementing the inverter of Fig. A7(b). In this kind of process, the p-transistors are constructed inside n-type well diffusions, with the substrate connected to GND and the wells to VDD.

Let us consider the n-channel transistor of Fig. A.7(a). For simplicity, we assume that V_{SB} (source-to-body voltage) and V_{FB} (flat-band voltage) are both zero. We are interested in examining the behavior of the region underneath the gate in the presence of $V_{GS} \geq 0\,\mathrm{V}$, and determining the corresponding current I_D that flows between drain and source as a consequence of this voltage (for a fixed voltage V_{DS}). We know from our previous discussion that, for a (large) current to flow, an inversion channel must be formed underneath the gate, a situation that exists if $V_{GS} \geq V_T$. In a

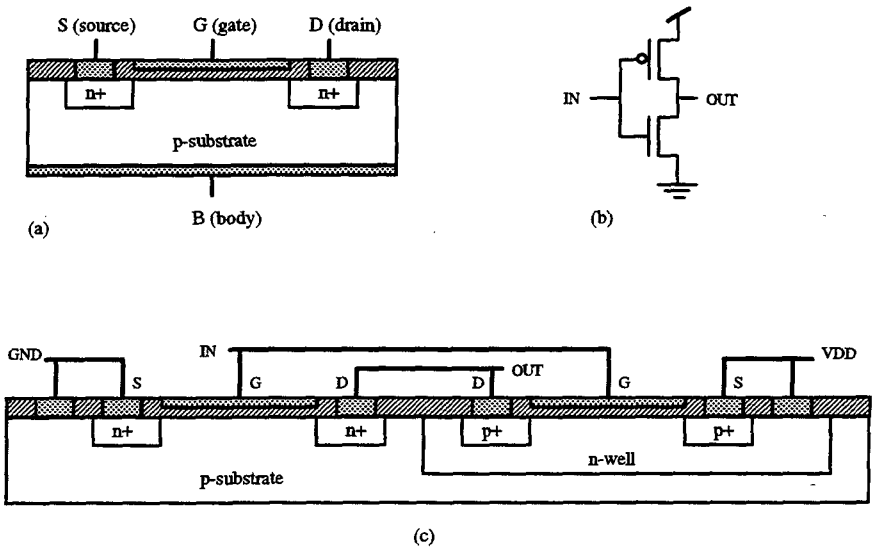


Fig. A.7: (a) n-channel MOSFET; (b) CMOS inverter circuit; (c) Integrated CMOS inverter implementation.

MOSFET, this mode of operation is normally referred to as *strong inversion* (above threshold). In this case, the current I_D is approximately linear with respect to V_{GS} (and V_{DS}) if V_{DS} is small ($V_{DS} < V_{GS} - V_T$). This sub-mode of operation is called the *linear* (or *triode*) mode. If, on the other hand, V_{DS} is large ($V_{DS} \geq V_{GS} - V_T$), the transistor is said to be in *saturation*, in which case I_D varies with the *square* of V_{GS} and is practically independent from V_{DS}. These regimes of operation are illustrated in the I_DxV_{DS} characteristic of Fig. A.8(a).

Even though the conductivity of the channel is enormously reduced when the transistor operates in the subthreshold (depletion) region (i.e., $V_{GS} < V_T$), a small diffusion current can still flow through it. If V_{GS} is relatively far from V_T (typically a few hundred milivolts), the dependence of I_D on V_{GS} is exponential, and is almost independent of V_{DS}. This sub-mode of operation is referred to as *weak inversion*. If, on the other hand, V_{GS} is near V_T, it is said to be in *moderate inversion*. In the latter case, the current-to-voltages relationship is rather complex, being therefore of little interest for (analog) computation. These modes of operation are also indicated in Fig. A.8(a).

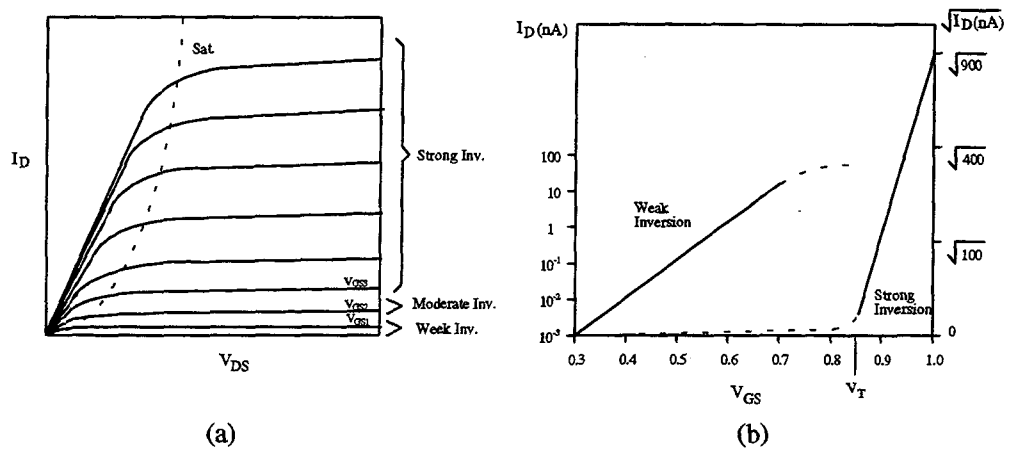


Fig. A.8: Drain current in an n-channel MOSFET: (a) Typical I_D versus V_{DS} transistor characteristic; (b) I_D versus V_{GS} plot illustrating the exponential behavior in the *weak inversion* region, and the quadratic behavior in the *strong inversion* regime (when in saturation mode).

A. 3 Drain Current Equations

The basic equations for I_D in an n-channel MOSFET in the several regimes of operation described in Sec. A.2 are summarized below.

Weak inversion (subthreshold, $V_{GS} < V_T$, exponential region):

$$I_D = I_0 e^{-\kappa \frac{q}{kT} V_{GS}} (1 - e^{-\kappa \frac{q}{kT} V_{DS}}) , \tag{A.4}$$

where $I_0 = n_i (W / L) \exp(-V_{bi} / \phi_t)$ is a constant determined by transistor parameters, $kT / q = \phi_t$ is the thermal voltage (26mV at room temperature), κ (~0.7) is a correction factor accounting for the effect of the ionized acceptors (or donors) of the substrate on the effectiveness of V_{GS} in controlling the barrier energy, n_i is the intrinsic carrier concentration, W and L are the transistor width and length, respectively, and V_{bi} is the built-in voltage at the source junction.

Strong inversion, in saturation mode ($V_{GS} \geq V_T$ and $V_{DS} \geq V_{GS} - V_T$):

$$I_D = \frac{\mu C_{ox}}{2}\frac{W}{L}(V_{GS} - V_T)^2 = \frac{\beta}{2}(V_{GS} - V_T)^2 \ , \tag{A.5}$$

where μ is the electron (or hole) mobility at the channel, $C_{ox} = \varepsilon_{ox} / t_{ox}$ is the oxide capacitance (per unit area), and $\beta = \mu C_{ox} W / L$ is the transconductance coefficient.

Strong inversion, in triode (linear) mode ($V_{GS} \geq V_T$ and $V_{DS} < V_{GS} - V_T$):

$$I_D = \beta[(V_{GS} - V_T)V_{DS} - \frac{V_{DS}^2}{2}] \ . \tag{A.6}$$

Notice that, for a p-channel device, the signs of all voltages are reversed.

A.4 Body Effect

Two of the main non-idealities in MOS transistors are the *body effect* and the *channel-length modulation effect*. The body effect refers to the property of the threshold voltage being dependent on the source-to-body voltage. For instance, if $V_{SB} > 0\text{V}$, a higher threshold voltage is needed in order to produce the same amount of channel charge (same I_D). The displacement of V_T caused by $V_{SB} \neq 0V$ is given by

$$V_T = V_{T0} + \gamma(\sqrt{\phi_B + V_{SB}} - \sqrt{\phi_B}) \ , \tag{A.7}$$

where $\gamma = \sqrt{2q\varepsilon_{Si}N_{SUB}} / C_{ox}$ and $\phi_B \approx 2\phi_F + 6\phi_t$ are process parameters, V_{T0} is the threshold voltage for $V_{SB} = 0\,\text{V}$ ($V_{T0} = V_{FB} + \phi_B + \gamma\sqrt{\phi_B}$), N_{SUB} is the substrate doping density (N_A or N_D), and $\phi_F = E_F / q$ is the Fermi voltage. Therefore, if $V_{SB} \neq 0\,\text{V}$, the threshold voltage that appears in (A.5)-(A.6) must be adjusted according to (A.7).

As an example of the body effect, we consider an n-channel transistor with the following process parameters: $N_{SUB} = 5.10^{15}\text{cm}^{-3}$, $t_{ox} = 400\text{A}^{\circ}$, and $V_{FB} \approx -0.5\text{V}$. This gives $C_{ox} = 0.88\text{fF}/\mu\text{m}^2$, $\gamma = 0.47\text{V}^{1/2}$, and $V_{T0} = 0.75\text{V}$. Consequently, $\Delta V_T = (0.47\sqrt{V_{SB} + 0.82} - 0.426)$ volts, resulting ΔV_T=0.21V, 0.36V, and 0.49V for V_{SB}=1V, 2V, and 3V, respectively.

A.5 Channel-Length Modulation (Early Effect)

This effect refers to the non-zero slope of the I_D versus V_{DS} transistor characteristic in the saturation region (Fig. A.8(a)). As we show below, this effect is more serious in short transistors. The channel-length modulation effect occurs because the effective transistor length changes with V_{DS}, that is,

$$L_{eff} = L - \Delta L \,, \tag{A.8}$$

where

$$\Delta L \approx \sqrt{\frac{2\varepsilon_{Si}}{qN_{SUB}}(V_{DS} - V_{GS} + V_T)} \,. \tag{A.9}$$

Thus the effective *W/L* ratio of the transistor increases with V_{DS}, causing I_D to grow. This effect is represented by an empirical factor λ, which allows (A.5) to be re-written as

$$I_D = \frac{\beta}{2}(V_{GS} - V_T)^2(1 + \lambda V_{DS}) \,. \tag{A.10}$$

For large values of L (say above 15 length units) this effect is small, but it can be strong in minimum-size transistors (i.e., L=2). Typical values of λ are in the 10^{-1}V^{-1} to 10^{-3}V^{-1} range.

Still another effect caused by short channels is the reduction of the threshold voltage, but this will not be discussed here. In fact, V_T is also affected by the width W of the transistor. However, contrary to short channels, narrow channels tend to increase the threshold voltage instead of reducing it.

A.6 SPICE Parameters

Most of the transistor parameters presented above are used in SPICE simulations. Table A.1 provides a correspondence between some of these parameters and the way they appear in the simulations, with their corresponding SPICE designations.

Table A.1

SPICE parameter	Physical parameter correspondence
VTO	V_{T0}
NSUB	N_A (for acceptors) or N_D (for donors)
TOX	t_{ox}
LAMBDA	λ
GAMMA	γ
PHI	$2\phi_F$
KP	β $(@W/L=1)$

A.7 The Design Parameter *k*

A final remark refers to the design parameter k that we have adopted in the derivations presented in the main chapters of the dissertation. As can be seen in the equations of I_D presented above, the dependence of the drain current on the transistor geometry is expressed in terms of the transistor ratio W/L rather than W or L separately. Furthermore, in multitransistor circuits the global behavior of the currents and voltages is often dependent on *ratios of transistor ratios* (e.g., $(W/L)_{trans.A}/(W/L)_{trans.B}$, or β_A/β_B), rather than isolated transistor ratios. The design parameter k was introduced in order to represent such ratios. The initial motivation for doing so is that expressing all W's and L's separately is of no help. The second (and main) reason is that k allows

reducing the *entropy* [e.g., Middlebrook 91] of the expressions enormously, therefore providing mathematical results that are much simpler to be interpreted physically. Once the value of k is determined (to fulfill primary system specifications), the individual values of W and L are normally made as small as possible, within the constraint that second-order system specifications (parameter matching, channel-length modulation, threshold fluctuation, etc.) are also satisfied. For instance, in digital implementations minimum-size transistors (W / L=3/2) are normally employed, for there are no second-order specifications to fulfill in general (that is, speed, power consumption, and layout area are normally all that matters).

Appendix B

Self-Contained *N*-Valued Memory

We present in this appendix a discussion on the problem of building programmable systems for long-term analog information storage. The analysis of this kind of circuit offers a rich opportunity of further understanding the potentialities of analog silicon systems, especially regarding the charge-injection problem. Massive VLSI implementations of analog memories are indeed one of the greatest challenges faced in the realization of neural networks and other high-density parallel signal processing systems.

B.1 Analog Memory

With the advent of neural networks, there has been a growing effort towards the development of easily programmable, high-density multi-valued memories in recent years [Hochet 90][Vittoz 91][Lee 91][Cauwenberghs 93]. While the simplest form consists of storing the information as a voltage level on a capacitor, the drift due to current leakage and noise limits this approach to short-term computations. Floating-gate techniques, on the other hand, allow for long-term storage, but lack a write access which is system compatible.

The basic premises in the development of the N-valued memory described below [Pedroni 94c] is that it must be small (high integration density), self-contained (no external communications channels needed), and equipped with some kind of error-correction capability. Although the basic principle adopted in most approaches to this problem is essentially the same, that is, to quantize the voltage to be stored to the nearest among $N=2^b$ possible discrete levels, where b is the number of quantization bits, and then periodically refresh this value accordingly, the algorithms (and consequently the MOS implementations) differ immensely. In [Hochet 90] and [Vittoz 91], for example, there is no error-control mechanism, thus any quatization error during refresh will destroy the information permanently. [Lee 91] and [Cauwenberghs 93], on the other hand, have error compensation schemes, with full range in [Cauwenberghs 93], but require permanent I/O communications with external units, so violating the self-containment premise of the implementation described here. In the present approach the voltage increments are computed at every cycle rather than being fixed, therefore adapting to the error that has been encountered. Also, if a quantization error occurs within a certain neighborhood of the correct level, it is automatically compensated by the system. The self-containment and the robustness against errors are achieved within a still small silicon area, 40% smaller than that of [Hochet 90], which, as mentioned before, has no error-preventing capabilities.

The circuit of the multi-valued memory is shown in Fig. B.1. It consists of three main parts: a voltage comparator, a control unit (pass transistors M1-M8 and two inverters), and two capacitors (C_p, the pre-storage capacitor, and the main capacitor, C). The system bus is composed of two write signals (row select, WRi, and column select, WRj) and of four refresh signals, which are all global (self-contained memory). Two of the refresh signals are digital (PREcharge and ENAble), whereas two are analog. The analog signals are steplike outputs of an external single slope D/A converter, being $V(n)$ the voltage at time n and $V(n-1)$ its previous value, which has been sampled and held. Finally, the switches controlled by SELect are common to all cells, thus

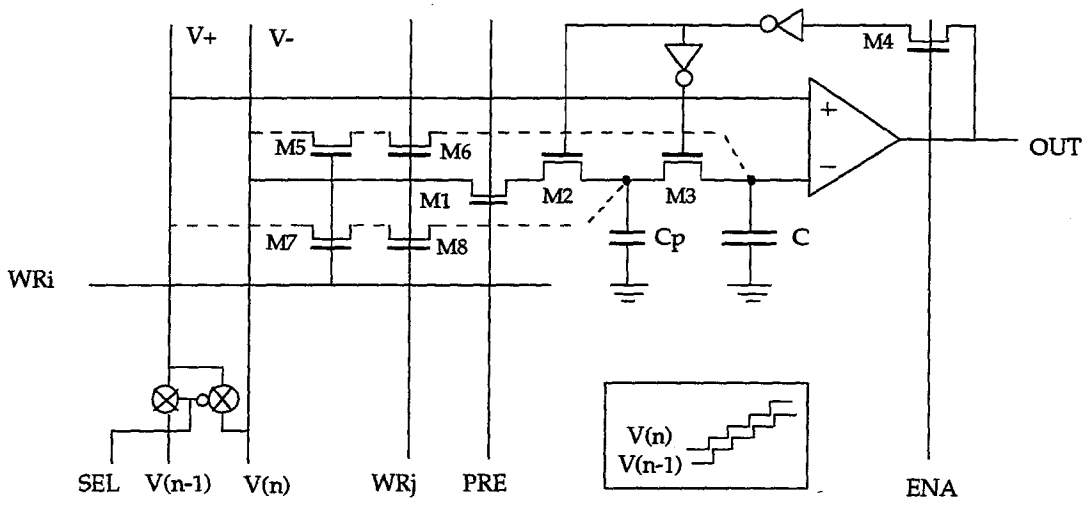


Fig. B.1: Analog-memory circuit.

making $V^+ = V(n-1)$ if SEL='1' or $V^+ = V(n)$ if SEL='0', with $V^- = V(n)$ always. The operation of the circuit is summarized below.

Write cycle: The four transistors connected through dashed lines (M5-M8) are active only during the write operation. Writing is done with SEL='1' and PRE='0', therefore charging C_p to $V(n-1)$ and C to $V(n)$. This ensures OUT='0' and thus M3=OFF. As the D/AC is running freely, within the next one or two clocks after the write pulses are applied it will cause the output of the comparator to switch to '1', turning M3 ON and thus causing C_p and C to share their charges. As $V_{Cp} < V_C$, the combined voltage will be a value in between these two voltages, given by

$$V_C = \frac{C_p.V(n-1) + C.V(n)}{C_p + C} = V(n) - \lambda.\Delta \qquad \text{(B.1)}$$

where λ is the capacitor ratio $\lambda = C_p / (C_p + C)$ and Δ is the voltage step of the signal generated by the D/A converter, i.e., $\Delta = V(n) - V(n-1)$. The stored voltage will be locked within this

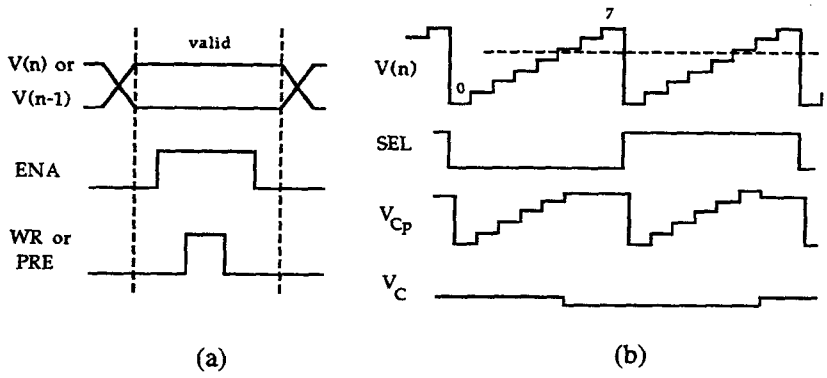


Fig. B.2: Control signals.

interval, being asymptotically approximated to its mid-point during refresh. The clock waveforms are illustrated in Fig. B.2(a).

Refresh cycle: Now only transistors M1-M4 are active. Transistor M4 acts as a protection against switching noise, which is likely to happen during the transitions of the D/AC, during which the output of the comparator is then disabled and the pair of inverters acts as a DRAM. PRE is pulsed high every time the input signals are ready (Fig. B.2(a)), while SEL is alternately '0' or '1' for a full D/AC cycle. This is exemplified in Fig. B.2(b) for N=8. If, for example, the "written" voltage is in the interval 4-5, C_p is precharged to 4 in the cycle with SEL='0', making the global voltage come down when OUT goes to '1'. Conversely, C_p is precharged to 5 in the next cycle (SEL='1'), causing the combined voltage to go up. Since the increments are applied in both directions alternately and are proportional to the error, this approach prevents the use of extra hardware for gradient detection.

We consider now the asymptotic value of the voltage stored on C, which is illustrated in Fig. B.3. The computation performed by the pair of capacitors is given by (see Fig. B.3(a))

$$V_{Cafter} = V_{Cbefore} + \lambda.\Delta^{+} \quad \text{or} \quad V_{Cafter} = V_{Cbefore} - \lambda.\Delta^{-} , \tag{B.2}$$

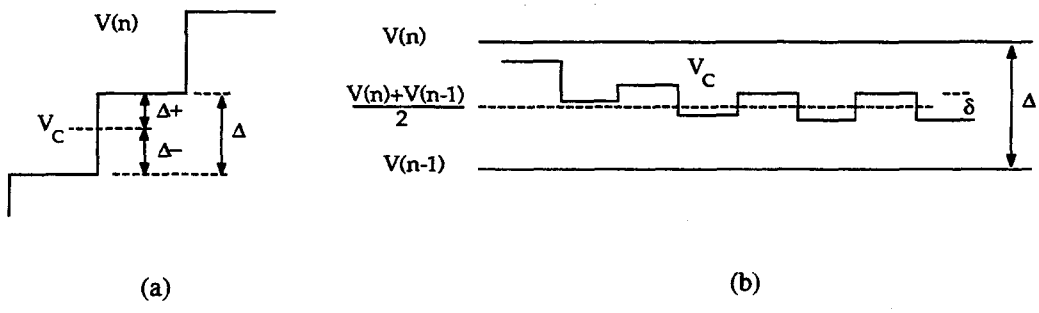


Fig. B.3: Asymptotic value of the stored voltage.

depending on whether it is an incremental (SEL='1') or decremental (SEL='0') cycle, respectively, with Δ^+ and Δ^- representing the absolute difference between the stored voltage and the neighboring levels. Taking the limit for a large number of refresh cycles gives (see Fig. B.3(b))

$$V_C = \frac{V(n)+V(n-1)}{2} \pm \frac{\delta}{2} \quad , \tag{B.3}$$

where $\delta = (\lambda / (2-\lambda))\Delta$. As $C_p \le C$, $\lambda \le 1/2$ and so $\delta \le \Delta / 3$.

Finally, we consider the sensitivity of the circuit to errors. The basic way that noise may affect the system is by switching the comparator at the wrong time (through line V^+), causing an unexpected charge sharing between C_p and C. We know from (B.2) that the variation of V_C after each cycle is proportional to the capacitor ratio $\lambda = C_p / (C_p + C)$. Hence, if λ is small, so are the voltage increments and decrements. This implies that, even if the comparator is switched when the precharge voltage is at a certain distance from its correct value, the combined voltage may still lie within its original limits. Due to amplitude limitations, noise superimposed to V^+ is more likely to affect the comparator in the vicinities of V_C. So, the error control range (ECR) is obtained by designing λ appropriately small. In the present case, we verify that

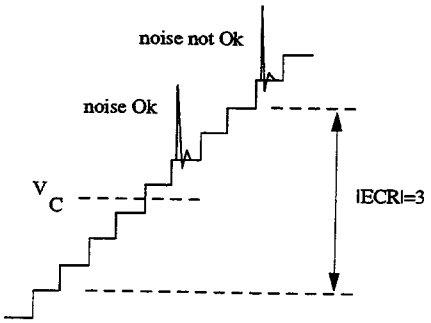


Fig. B.4: Error-control range.

$$ECR = \pm\frac{1}{2\lambda} \quad . \tag{B.4}$$

This is illustrated in Fig. B.4 for $C=5C_p$, what gives $\lambda=1/6$ and, consequently, ECR=$\pm$3.

B.2 Design Considerations and Experimental Results

Similarly to any other switched-capacitor implementation, clock-feedthrough [Wilson 85] [Wegmann 87] plays a major role in the performance of the system. There are again two basic situations to consider: *write* and *refresh*.

During the write cycle, the condition for each capacitor is similar to that depicted in Fig. B.5(a). Due to the low impedance of the signal source (V^- in this case), there is no charge-feedthrough when the transistor is turned on; however, there is when it is turned off. If C' were constant and the transistor switched off instantaneously, the voltage change would be easily obtained as $\Delta V_C \approx -(C'/C)V_{DD}$. However, C' is dependent on several factors, including the input voltage itself, being not constant and not even linear, while the conductivity of the channel depends, among others, on the fall time of the control clock [Wegmann 87][Wilson 85]. However, these factors

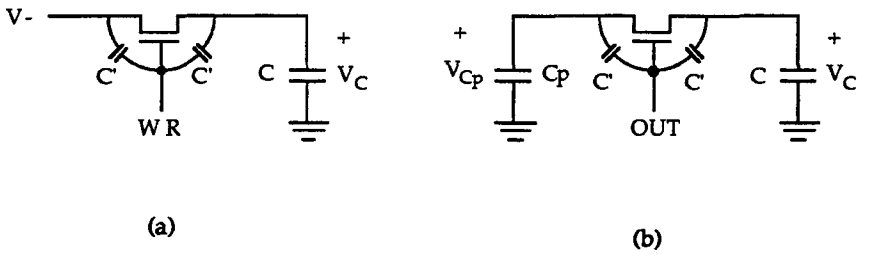


Fig. B.5: Clock-feedthrough analysis.

cause practical voltage variations to be somewhat smaller than the prediction above. Notice also that one of the write transistors could be replaced by a p-type one, so acting as a dummy transistor [Song 93], but neither this nor complementary switches [Wegmann 87] are very effective in canceling the injected charge.

During refresh, the situation is similar to that of Fig. B.5(b), in which case there are no low-impedance paths during either the ON or the OFF operation. While this implies that the turn-off clock feedthrough will be bigger than in the previous case, it also implies that there will be charge injection in the turn-on operation, thus partially compensating for the increase in turn-off error. Indeed, if the premises above were true once again, the net voltage change would be exactly the same as that predicted for the write cycle.

As a result, we consider a practical approximation for estimating the clock-feedthrough voltage variation which is given by

$$\Delta V_C \approx -\frac{A_G}{2A_C} V_{DD} \tag{B.5}$$

with A_G and A_C representing the area of the transistor gate and of the capacitor, respectively, while V_{DD} is the conventional rail-to-rail clock voltage. Since the memory can operate at very low refresh rates (typically well below 1kHz), minimum-size transistors can be used without compromising the time-response of the system. Thus, (B.5) can be used to design C such that ΔV_C is significantly smaller than Δ, the voltage step of the input signals.

Experimental Results: A 64-level memory array, with an error control range ECR=±4, was fabricated on a double-poly chip using 2.0μm CMOS technology. The supply voltage was V_{DD}=5V, thus easily accommodating a Δ=50mV signal (3.15V dynamic range). With C of size 5,000μm² (~2.5pF) and minimum-size transistors, we obtain from (B.5) that the expected clock-feedthrough on C is of the order of -3mV. (The asymptotic limit in Fig. B.3(b) will be displaced downward by this amount.) As λ=1/8, the size required for C_p is 715μm² (~0.36pF), resulting Δ.ECR=±200mV of noise protection. Finally, we observe that Δ also determines the maximum offset acceptable at the comparator. A conventional two-stage p-type comparator [Allen 87], with an offset under 10mV and the appropriate dynamic range, was employed in the implementation.

The complete memory cell, as described above, is shown in Fig. B.6. It occupies an area of 203x89μm², which is 40% smaller than that of [Hochet 90] (with same technology and same storage capacitor size). The power consumption of the control unit is very small, due to the very low operating speed of the system, causing the overall power consumption of the system to be basically that of the comparator. Due to noise associated with switching and the $V(n)$ and $V(n-1)$ signals, a minimum Δ of 65mV was necessary in order for the circuit to operate properly. The tests are illustrated in the oscilloscope picture of Fig. B.7, which shows the signal $V(n)$ on the upper trace, the voltage on C in the middle, and the output voltage on the bottom. As can be seen, the voltage on C stays constant after $V(n)$ reaches the stored voltage, in which case the output voltage switches from low to high, as described in the operation of the circuit.

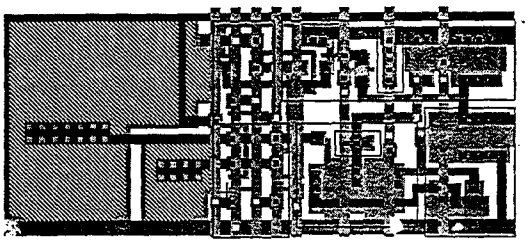


Fig. B.6: Layout of circuit of Fig. B.1.

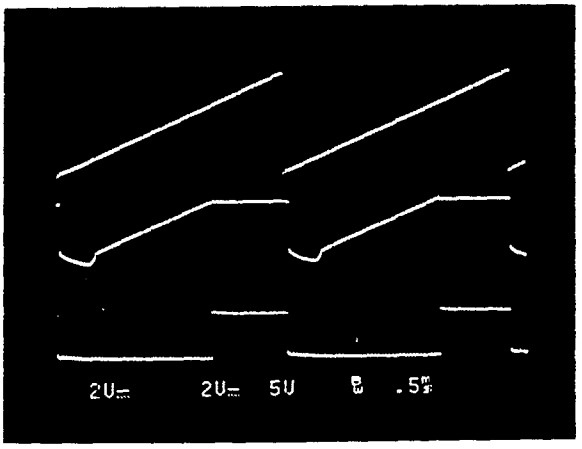


Fig. B.7: Measurements on a 64-level memory fabricated on a 2.0μm CMOS chip. Signal *V(n)* is shown on the upper channel of the scope, while the voltage on *C* is shown in the middle and the output voltage on the bottom.

References

[Abu-Mostafa 85] Y. Abu-Mostafa, J. Saint Jacques, "Information capacity of the Hopfield model," IEEE Trans. Information Theory, pp. 461-464, 1985.

[Abu-Mostafa 89] Y. Abu-Mostafa, "The Vapnik-Chervonenkis dimension: information versus complexity in learning," Neural Computation, pp. 312-317, 1989.

[Abu-Mostafa 93] Y. Abu-Mostafa, "Pattern recognition," class notes, Caltech, winter 1992.

[Agranat 90] A. Agranat, C. Neugebauer, R. Nelson, A. Yariv, "The CCD neural processor: a neural network integrated circuit with 65536 programmable analog synapses," IEEE Trans. on Circuits and Systems, pp. 1073-75, 1990.

[Allen 87] P. Allen, D. Holberg, "CMOS analog circuit design", N. York: Holt, Rinehart, Winston, 1987.

[Amelio 71] G. Amelio, W. Bertram, M. Tompsett, "Charge-coupled imaging devices: design considerations," IEEE Trans. on Electron Devices, pp. 986-992, 1971.

[Andreou 89] A. G. Andreou, K. A. Boahen, "Synthetic neural circuits using current-domain representations," Neural Computation, pp. 489-501, 1989.

[Andreou 91] A. G. Andreou, K. A. Boahen, P. O. Pouliquen, A. Pavasovic, R. E. Jenkins, K. Strohbehn, "Current-mode subthreshold MOS circuits for analog VLSI neural systems," IEEE Trans. on Neural Networks, pp. 205-213, 1991.

[Barbe 75] D. Barbe, "Imaging devices using the charge coupled concept," Proc. IEEE, vol. 63, pp. 38-97, 1975.

[Baum 88] E. B. Baum, "On the capabilities of multilayer perceptrons," Journal of Complexity, pp. 193-215, 1988.

[Baum 89] E. B. Baum, D. Hausller, "What size net gives valid generalization?" Neural Computation, pp. 151-160, 1989.

[Baum 91] E. B. Baum, "Neural net algorithms that learn in polynomial time from examples and queries," IEEE Trans. Neural Networks, pp. 5-19, 1991.

[Berglund 73] C. Berglund, K. Thomber, "Incomplete transfer in charge transfer devices," IEEE Journal of Solid State Circuits, pp. 108-116, 1973.

[Beynon 80] J. Beynon, D. Lamb, "Charge coupled devices and their applications," London: McGraw-Hill, 1980.

[Brodersen 75] R. Brodersen, D. Buss, Al Tasch, Jr., "Experimental characterization of transfer efficiency in charge-coupled devices," IEEE Trans. on Electron Devices, pp. 40-46, 1975.

[Carnes 72] J. Carnes, W. Kosonocky, E. Ramberg, "Free charge transfer in charge-coupled devices," IEEE Trans. on Electron Devices, pp. 798-808, 1972.

[Cauwenberghs 93] G. Cauwenberghs, "Analog VLSI autonomous systems for learning and optimization," Ph.D. dissertation, Caltech, 1993.

[Cauwenberghs 94] G. Cauwenberghs, V. A. Pedroni, "A charge-based CMOS parallel analog vector quantizer," Neural Inf. Proc. Systems (NIPS), Denver, 1994.

[Chan 92] W. Chan, S. Gupta, A. Gersho, "Enhanced multistage vector quantization by joint codebook design," IEEE Trans. on Communications, pp. 1693-1697, 1992.

[Chiang 91] A. M. Chiang, M. Chuang, "A CCD programmable signal processor and its neural network applications," IEEE Journal of Solid State Circuits, pp. 1894-1901, 1991.

[Cilingiroglu 93] U. Cilingiroglu, "A charge-based neural Hamming classifier," IEEE J. Solid-State Circuits, pp. 59-67, 1993.

[Fossum 91] E. Fossum, "Wire transfer of charge packets using a CCD-BBD structure for charge-domain signal processing," IEEE Trans. on Electron Devices, pp. 291-298, 1991.

[Gray 84] R. M. Gray, "Vector quantization," IEEE ASSP Mag., pp. 4-29, 1984.

[Hochet 90] B. Hochet, V. Peiris, S. Abdo, M. Declercq, "Implementation of a learning Kohonen neuron based on a new multilevel storage technique", IEEE Journal of SSC, pp. 262-267, 1990.

[Hong 87] J. Hong, "On connectionist models," Technical report, Dept. of Computer Science, University of Chicago, 1987.

[Jarske 93] T. Jarske, O. Vainio, "A review of median filter systems for analog signal processing," in Analog Integrated Circuits and Signal Processing, Boston: Kluwer Acad. Publ., 1993.

[Johnson 91] L. Johnson, S. Jalaleddine, "MOS implementation of winner-take-all network with application to content-addressable memory," Electronics Letters, pp. 957-958, 1991.

[Kub 90] F. J. Kub, K. K. Moon, I. A. Mack, F. M. Long, "Programmable analog vector-matrix multipliers," IEEE JSSC, pp. 207-213, 1990.

[Lazzaro 89] J. Lazzaro, S. Ryckebush, M. A. Mahowald, C. Mead, "Winner-take-all networks of O(N) complexity," Neural Inf. Proc. Systems (NIPS), Denver, 1989.

[Lee 91] E. Lee, P. Gulak, "Error correction technique for multivalued MOS memories", Elec. Letters, pp. 1321-1322, 1991.

[Lewis 67] P. M. Lewis, C. L. Coates, "Threshold logic," N.York: Wiley, 1967.

[Mead 89] C. Mead, "Analog VLSI and neural networks," N. York: Addison-Wesley, 1989.

[Middlebrook 91] R. D. Middlebrook, "Low-entropy expressions: the key to design-oriented analysis," Proc. IEEE Frontiers in Education, 1991, pp. 399-403.

[Moon 94] G. Moon, M. Zaghloul, R. Newcomb, "CMOS design of two winner-take-all circuits using pulse duty cycle synaptic weighting," Proceedings of ISCAS'94, pp. 379-382, London, 1994.

[Muller 90] B. Muller, J. Reinhardt, "Neural networks: An introduction," New York: Springer-Verlag, 1990.

[Murthy 92] N. Murthy, M. Swamy, "On the VLSI implementation of real-time order statistic filters," IEEE Trans. Signal Proc., pp. 1241-1252, 1992.

[Myhill 61] J. Myhill, "On the size of weights required for linear-input switching functions," IEEE Trans. on Electronic Computers, pp. 288-290, 1961.

[Nasrabadi 88] N. M. Nasrabadi, R. A. King, "Image coding using vector quantization: a review," IEEE Trans. on Communications, pp. 957-969, 1988.

[Neugebauer 92] C. Neugebauer, A. Yariv, "A parallel analog CCD/CMOS signal processor," Neural Information Processing Systems, Denver, pp. 748-755, 1992.

[Neugebauer 93] C. Neugebauer, "Parallel analog computation with charge coupled devices," Ph.D. dissertation, Caltech, Dept. of Applied Physics, 1993.

[Olafsson 88] S. Olafsson, Y. Abu-Mostafa, "The capacity of multilevel threshold functions," IEEE Trans. Pattern Analysis and Machine Intelligence, pp. 277-281, 1988.

[Pedroni 92a] V. Pedroni, A. Agrant, A. Yariv, "Pattern matching and parallel processing with CCD technology," International Joint Conference on Neural Networks, Baltimore-USA, 1992.

[Pedroni 92b] V. Pedroni, A. Agrant, A. Yariv, "Hamming distance calculation and pattern matching with CCD technology," US Patent granted, filed March 1993.

[Pedroni 93] V. A. Pedroni, "Learning in the hypercube," IEEE Int. Conf. on Neural Networks, San Francisco, 1993.

[Pedroni 94a] V. A. Pedroni, "Inhibitory-mechanism analysis of complexity *O(N)* MOS winner-take-all networks," IEEE Trans. on Circuits and Systems - accepted for publication; submitted 01/94.

[Pedroni 94b] V. A. Pedroni, "Neural *n*-port voltage comparator network," Electronics Letters, pp. 1774-1775, 1994.

[Pedroni 94c] V. A. Pedroni, "Self-contained error-compensated N-valued memory for neural applications," 4th Int. Conf. on Microelectronics and Fuzzy Logic Systems, Italy, Sept. 1994.

[Pedroni 94d] V. A. Pedroni, A. Yariv, "Locally-controlled CCD's and their applications to parallel and Hamming computation," Submitted to IEEE Trans. On Circuits and Systems, 08/94.

[Pedroni 94e] V. A. Pedroni, A. Yariv, "Locally-controlled charge-coupled devices (LCCD's)," U.S. Pattent Application 08289581, 1994.

[Pedroni 94f] V. Pedroni, "Highly linear high-density vector quantiser and vector-matrix multiplier," Electronics Letters, pp. 945-946, 1994.

[Pierret 89] R. Pierret, "Field effect devices," N. York: Addison-Wesley, 1989.

[Pucknell 88] D. Pucknell, K. Eshraghian, "Basic VLSI design," N. York: Prentice Hall, 1988.

[Rumelhart 86] D. E. Rumelhart, G. E. Hinton, R. J. Williams, "Learning representations by back-propagating errors," Nature, pp. 533-536, 1986.

[Sequin 75] C. Sequin, M. Tompsett, "Charge transfer devices," New York: Academic Press, 1975.

[Song 93] M. Song, Y. Lee, W. Kim, "Clock feedthrough reduction circuits for switched-current systems," IEEE Journal of SSC, pp. 133-137, 1993.

[Starzyk 93] J. A. Starzyk, X. Fang, "CMOS current mode winner-take-all circuit with both excitatory and inhibitory feedback," Electronics Letters, pp. 908-910, 1983.

[Szcepanski 93] S. Szcepanski, A. Wyszynski, R. Schaumann, "Highly linear voltage-controlled CMOS transconductors," IEEE Trans. on Circuits and Systems, pp.258-262, 1993.

[Sze 81] S. M. Sze, "Physics of semiconductor devices," N. York: Wiley, 1981.

[Tompsett 73] M. Tompsett, "The quantitative effects of interface states on the performance of charge-coupled devices," IEEE Trans. on Electron Devices, pp. 45-55, 1973.

[Tsividis 87] Y. Tsividis, "Operation and modeling of the MOS transistor," N. York: McGraw-Hill, 1987.

[Viterbi 79] A. J. Viterbi, J. K. Omura, "Principles of digital communication and coding," New York: McGraw-Hill, 1979.

[Wegmann 87] G. Wegmann, E. Vittoz, F. Rahali, "Charge injection in analog MOS switches," IEEE Journal of SSC, pp. 1091-1097, 1987.

[Wendt 86] P. Wendt, E. Coyle, C. Gallagher, "Stack filters," IEEE Trans. Acoustics, Speech, and Signal Proc., pp. 898-911, 1986.

[Weste 93] N. Weste, K. Eshraghian, "Principles of CMOS VLSI design," 2nd edition, New York: Addison-Wesley, 1993.

[Wilson 85] W. Wilson, H. Massoud, E. Swanson, R. George, R. Fair, "Measurement and Modeling of Charge Feedthrough in n-Channel MOS Analog Switches", IEEE Journal of SSC, pp. 146-152, 1985.

[Winder 60] R. O. Winder, "Threshold logic," Ph.D. dissertation, Princeton University, 1960.

[Winder 63] R. O. Winder, "Bounds on threshold gate realizability," IEEE Trans. on Computers, pp. 561-564, 1963.

[Yli-Harja 91] O. Yli-Harja, J. Astola, Y. Neuro, "Analysis of the properties of median and weighted median filters using threshold logic and stack filter representation," IEEE Trans. Signal Proc., pp. 395-410, 1991.

[Yuping 93] H. Yuping, U. Ciringiroglu, "A charge-based on-chip adaptation Kohonen neural network," IEEE Trans. Neural Networks, pp. 462-469, 1993.