

1991

# An architecture for multi-layer feed-forward neural networks.

Aria. Nosratinia  
*University of Windsor*

Follow this and additional works at: <http://scholar.uwindsor.ca/etd>

---

## Recommended Citation

Nosratinia, Aria., "An architecture for multi-layer feed-forward neural networks." (1991). *Electronic Theses and Dissertations*. Paper 980.

This online database contains the full-text of PhD dissertations and Masters' theses of University of Windsor students from 1954 forward. These documents are made available for personal study and research purposes only, in accordance with the Canadian Copyright Act and the Creative Commons license—CC BY-NC-ND (Attribution, Non-Commercial, No Derivative Works). Under this license, works must always be attributed to the copyright holder (original author), cannot be used for any commercial purposes, and may not be altered. Any other use would require the permission of the copyright holder. Students may inquire about withdrawing their dissertation and/or thesis from this database. For additional inquiries, please contact the repository administrator via email ([scholarship@uwindsor.ca](mailto:scholarship@uwindsor.ca)) or by telephone at 519-253-3000ext. 3208.



National Library  
of Canada

Bibliothèque nationale  
du Canada

Canadian Theses Service    Service des thèses canadiennes

Ottawa, Canada  
K1A 0N4

## NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

## AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

# **AN ARCHITECTURE FOR MULTI-LAYER FEED-FORWARD NEURAL NETWORKS**

by

Aria Nosratinia

A Thesis

Submitted to the Faculty of Graduate Studies through the  
Department of Electrical Engineering in Partial Fulfillment  
of the Requirements for the Degree of  
Master of Applied Science at the  
University of Windsor

Windsor, Ontario, Canada

August 1991



National Library  
of Canada

Bibliothèque nationale  
du Canada

Canadian Theses Service    Service des thèses canadiennes

Ottawa, Canada  
K1A 0N4

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-315-69905-1

Canada

© Aria Nosratinia 1991

All Rights Reserved

## ABSTRACT

Feed-forward neural networks can perform classifications and generalizations that are difficult to achieve with any other known method, and their performance matches or surpasses that of the conventional methods. To utilize the potential of these networks to the fullest, however, an efficient hardware implementation is needed. In this thesis, an architecture for efficient implementation of feed-forward multi-layer neural networks is introduced. The interconnection congestion problem is addressed by a multiplexing scheme, which reduces the number of physical interconnections without any loss of generality. The building blocks are mostly in current mode analog CMOS, and the connection strengths of the network are stored in a digital memory. Also included in this thesis is a performance analysis of the architecture and a study of the effects of quantization and truncation of connection strengths on network performance.

*To my parents*

## ACKNOWLEDGEMENTS

*I would like to express my sincere thanks and appreciation to my supervisors, Dr. M. Ahmadi and Dr. M. Shridhar, for their support and guidance throughout the progress of this thesis. I would also like to thank Mr. Jun Cao for his help in running the simulations of chapter 5. Thanks must also go to the Mostafapour family for their unfailing support and encouragement during my stay in Windsor.*



## TABLE OF CONTENTS

ABSTRACT . . . . .	iv
DEDICATION . . . . .	v
ACKNOWLEDGEMENTS . . . . .	vi
LIST OF FIGURES . . . . .	ix
LIST OF TABLES . . . . .	xii
1. Introduction . . . . .	1
1.1 Goals, objectives and organization of this thesis . . . . .	2
1.2 A brief history of research in neural models . . . . .	4
1.3 Some major points on nervous system and neural models . . . . .	6
1.4 Implementation of neural networks . . . . .	7
1.5 VLSI implementations . . . . .	9
1.6 Handwritten character recognition and feed-forward networks . . . . .	12
1.7 Summary . . . . .	14
2. An Architecture for Multi-Layer Neural Networks . . . . .	15
2.1 The architecture . . . . .	18
2.2 Operation . . . . .	20
2.3 Modifications . . . . .	23
2.4 Units and scales . . . . .	25
2.5 Input/output considerations . . . . .	27
2.6 Performance analysis on an example . . . . .	30
2.7 Summary . . . . .	40
3. Neuron and Synapse . . . . .	42
3.1 Neuron . . . . .	42
3.1.1 Current mode neurons . . . . .	43
3.1.2 Voltage mode neurons . . . . .	48
3.2 Synapse . . . . .	50
3.2.1 Multiplier . . . . .	50
3.2.2 ROM . . . . .	53
3.3 Summary . . . . .	56

4. Support Circuitry . . . . .	57
4.1 Trans-impedance amplifier . . . . .	57
4.2 Voltage to current converter . . . . .	60
4.3 Address generation . . . . .	62
4.4 Analog buffers and output pad drivers . . . . .	63
4.4.1 A buffer with high capacitive drive . . . . .	64
4.4.2 A rail-to-rail buffer for resistive and capacitive loads . . . . .	69
4.5 Summary . . . . .	75
5. Quantization and Truncation Effects . . . . .	76
5.1 A simple example; XOR-XNOR . . . . .	76
5.2 A complex example;handwritten character recognition . . . . .	77
5.3 Summary . . . . .	81
6. Conclusion . . . . .	82
Appendix A A Brief Overview of the Nervous System . . . . .	86
Appendix B Layout Diagrams . . . . .	95
Appendix C Hardware and Software Tools . . . . .	106
REFERENCES . . . . .	109
VITA AUCTORIS . . . . .	118

## LIST OF FIGURES

Figure 1.1:	Character Recognition . . . . .	13
Figure 2.1:	Multi-layer feed-forward network . . . . .	16
Figure 2.2:	Layers and stages in the architecture . . . . .	18
Figure 2.3:	Internal structure of one stage . . . . .	19
Figure 2.4:	The adder . . . . .	21
Figure 2.5:	Modified architecture . . . . .	23
Figure 2.6:	Second modification for further improvement . . . . .	24
Figure 2.7:	Conversion of the signal mode at input and output . . . . .	28
Figure 2.8:	Second solution to the I/O problem . . . . .	29
Figure 2.9:	Changing the I/O mode by displacing the neurons . . . . .	30
Figure 2.10:	XOR and XNOR . . . . .	31
Figure 2.11:	XOR-XNOR using the original architecture . . . . .	32
Figure 2.12:	Waveforms of the original architecture . . . . .	33-35
Figure 2.13:	XOR-XNOR with the modified architecture . . . . .	37
Figure 2.14:	Waveforms of the modified architecture . . . . .	38-40
Figure 3.1:	The sigmoidal logistics function . . . . .	43
Figure 3.2:	Current mode neuron with variable threshold . . . . .	44
Figure 3.3:	The differential pair and its transfer characteristics . . . . .	44
Figure 3.4:	Simulation results of the neural cell with a threshold of zero . . . . .	46
Figure 3.5:	Test results of the neural cell with various threshold values . . . . .	46
Figure 3.6:	Current mode neural cell with fixed threshold . . . . .	47
Figure 3.7:	Voltage mode neuron with variable threshold . . . . .	48

Figure 3.8:	Voltage mode neuron with fixed threshold . . . . .	48
Figure 3.9:	Transfer characteristics of the voltage mode neuron . . . . .	49
Figure 3.10:	Cascode and Wilson current mirrors . . . . .	51
Figure 3.11:	Characteristics of cascode and Wilson current mirrors . . . . .	51
Figure 3.12:	The multiplying D/A . . . . .	52
Figure 3.13:	Output of the D/A vs. input current . . . . .	53
Figure 3.14:	Schematic diagram of a section of the ROM . . . . .	54
Figure 3.15:	Data access delay in the ROM . . . . .	54
Figure 3.16:	A possible source of error in the ROM circuit . . . . .	55
Figure 4.1:	Trans-impedance amplifier . . . . .	58
Figure 4.2:	Simulated and actual transfer characteristics . . . . .	59
Figure 4.3:	Step response of the X-impedance amplifier with a load of 0.5pF	59
Figure 4.4:	Voltage to current converter . . . . .	60
Figure 4.5:	Transfer characteristics of the voltage to current converter . . . . .	61
Figure 4.6:	A suggested circuit for the ring counter . . . . .	63
Figure 4.7:	Schematic diagram of the buffer with high capacitive drive . . . . .	65
Figure 4.8:	Test results: A) Step response with $C_L=5000\text{pF}$ B) Stability test with $C_L=200\text{pF}$ . . . . .	67
Figure 4.9:	Transfer characteristics with various loads (test results) . . . . .	68
Figure 4.10:	Normalized harmonics of the output at 20kHz with $V_{in}=2\text{V(p-p)}$	68
Figure 4.11:	Frequency response of the buffer with $C_L=5000\text{pF}$ $R_L=10\text{k}\Omega$ . . . . .	68
Figure 4.12:	Schematic diagram of the buffer for resistive and capacitive load	70
Figure 4.13:	Test results: A) step response with $C_L=5000\text{pF}$ B) Stability test with $C_L=200\text{pF}$ . . . . .	72
Figure 4.14:	Gate voltages of M16 and M17 as the output slews . . . . .	73
Figure 4.15:	Transfer characteristics with various loads (test results) . . . . .	73

Figure 4.16:	Frequency response at different operating points with $C_L=5000\text{pF}$	74
Figure 4.17:	Normalized harmonics at 20kHz with $V_{in}=2\text{V(p-p)}$ and $R_L=10\text{k}\Omega$	74
Figure 4.18:	Total harmonic distortion vs. load at various operating points	74
Figure 5.1:	XOR-XNOR with quantized weights	77
Figure 5.2:	Distribution of the synaptic weights in Example 2	79
Figure A.1:	Neuron and its different sections	89
Figure A.2:	Different neuron types	90
Figure A.3:	A) A typical nerve impulse    B) Output of a neuron	93
Figure B.1:	Layout diagram of the current mode neuron with variable threshold	96
Figure B.2:	Layout diagram of the current mode neuron with fixed threshold	97
Figure B.3:	Layout diagram of the voltage mode neuron with variable threshold	98
Figure B.4:	Layout diagram of the voltage mode neuron with fixed threshold	99
Figure B.5:	The multiplying D/A	100
Figure B.6:	ROM with buffers	101
Figure B.7:	Trans-impedance amplifier	102
Figure B.8:	Voltage to current converter	103
Figure B.9:	Output buffer for high capacitive loads	104
Figure B.10:	Output buffer for high capacitive and resistive loads	105

## LIST OF TABLES

Table 3.1: Transistor geometries of the neural cells . . . . .	49
Table 4.1: Device geometries for the trans-impedance amplifier . . . . .	58
Table 4.2: Device geometries for the voltage to current converter . . . . .	62
Table 4.3: Device geometries for the buffer for capacitive loads . . . . .	65
Table 4.4: Worst case pole-zero configurations at $V_{in}=4V$ (simulation) . . . . .	66
Table 4.5: Device geometries for the second buffer . . . . .	70
Table 4.6: Pole-zero configurations (simulation) . . . . .	71
Table 5.1: Recall accuracy with the original connection strengths . . . . .	78
Table 5.2: Recall accuracy with the weights quantized but not truncated . . . . .	80
Table 5.3: Recall accuracy with weights truncated at 20 and quantized . . . . .	80
Table 5.4: Recall accuracy with weights truncated at 14 and quantized . . . . .	80

## **Introduction**

---

Even in the age of supercomputers with processing powers in the range of tens to hundreds of megaflops, there are many tasks at which the human brain surpasses the computer. These tasks fall mainly in the area of associative memory, generalization, perception and classification. The brain itself is an amazingly complex machine, a systematic study and understanding of which started only a century ago. A brief overview of the structure and biology of the nervous system is presented in appendix A.

Ever since the mid-twentieth century, the capabilities of the brain in the areas of associative memory and robust decision making has drawn the interest of cognitive and physiological psychologists, as well as computer scientists. A variety of models to describe different behavioral aspects of the brain have been proposed since then. Each of these models, at best, reflects only a few of the properties of the nervous system.

In most of the cases where the brain holds the edge over the computer, a number of different pieces of information have to be processed at the same time. Examples include perception of written words, where one or more of the letters are partially or completely obscured [1-3], and spoken word or phrases where a part of the phonemes are removed [4]. These examples suggest a mechanism in which a number of processes occur simultaneously, influencing and being influenced by each other. This leads to a class of models that have most appropriately been called Parallel Distributed Processing (PDP) [3].

In these models, the processing of information is performed by a large number of relatively simple, highly interconnected elements. Psychologists usually consider these elements as certain hypotheses about such things as the letters in a particular display, and the activities would stand for the relative strengths of these hypotheses. In other cases, they may consider them as possible goals, objectives or actions. Electrical Engineers and others involved in physical sciences, however, prefer to consider these units as physical cells with measurable, quantitative inputs and outputs. From the viewpoint of the engineer, a neural network in the most general case consists of a number of nonlinear, highly interconnected elements, each of which acts on the weighted sum of its inputs. This view is particularly attractive because of its biological plausibility. In either case, the PDP models have radically changed our perceptions of the mechanisms of learning and recall.

### **1.1 Goals, objectives and organization of this thesis:**

This work concerns itself with the development of architectures and circuits for feed-forward multi-layer neural networks. A hybrid architecture for this type of network will be presented. It will be shown that this architecture results in a decrease in the number of physical interconnections on the chip without any loss of generality. The multiplexing



scheme used will also make possible multi-chip systems without a great number of connections. A complete set of cells for this architecture has been designed in a full custom CMOS VLSI technology.

The motivation for this work was a desire for a real-time solution for handwritten character recognition. However, all architectures and circuits presented in this thesis can be applied to any feed-forward network regardless of application. Multi-layer feed-forward networks have shown a great deal of promise as robust and reliable classifiers and it is in this capacity that they are used in the solution of the character recognition problem. A brief description of the character recognition problem is presented later as a typical application.

In chapter 1, the objectives of the thesis are defined, a brief overview of the history of research in neural models and implementation methods is presented, and some important points about neural networks are reviewed. The original form of the architecture is proposed and analyzed in chapter 2. In this scheme, the number of interconnections and multipliers is reduced considerably from a full implementation. Also, the problem of matching the units and scales in the network, and the I/O mode of the network and related issues, are discussed. A number of solutions to these problems and the advantages and drawbacks of each of them are presented.

In chapter 3, realizations of the two main building blocks, neuron and synapse, are introduced. Several variations of the neuron and also the issues pertinent in the design of the synapse will be discussed. In chapter 4, the other building blocks needed in the architecture are presented. These include a trans-impedance amplifier, a voltage to current converter, and a series of high-power output buffers with low silicon area consumption. A suggestion for efficient address generation for the ROM and demultiplexers is given in section 4.3.

In chapter 5, the effects of quantization and truncation of the weights on two examples are investigated. Finally, in chapter 6, conclusions about this work of research are presented and some areas of promise for future work are pointed out.

## **1.2 A Brief history of research in neural models:**

The history of research on nervous systems goes as far back as the discovery of the electrical nature of nervous transmission by Galvani, and the early experiments of Helmholtz. Significant advances in the area of nervous modeling, however, had to wait until late nineteenth century. At this time, cognitive psychologists like James [5] outlined the fundamental concepts of nervous activity that are still in use today.

The famous paper of McCulloch and Pitts [6] in 1943 is usually cited as the beginning of serious work on neural modeling. They showed that a neural-like network of linear threshold elements can compute any logical function. Actually, the impact of this paper on computer science was much more pronounced than on neural networks [7-8]. In 1949, Donald Hebb published his epoch making work [9], explicitly stating the correlation update law that James had only hinted at half a century ago. Frank Rosenblatt [10] proposed the first computationally oriented network and also devised the perceptron convergence procedure. Widrow and Hoff [11] with their ADALINE (1960) helped bring adaptive systems and neural networks together; and also attracted the interest of electrical engineers to this relatively young subject.

At this time, it was clear that a dichotomy in the understanding of the brain function was forming. One group, inspired by Rosenblatt, believed in the power of the perceptron and in general considered a connectionist architecture essential to the types of behavior observed

in human learning and recall processes. In other words, this group regarded as important not only *what* the brain does, but also *how* it is done. The other group attached no great importance to the internal mechanisms of the nervous system; they were mostly in favour of serial symbol processing. The area of artificial intelligence as we know it today is a reflection of the beliefs of this second group.

In 1960, Minsky and Papert published their famous book, *Perceptrons* [12]. This book is a brilliant mathematical analysis of the limitations of networks consisting of a single layer of linear threshold units. The only mistake of the authors, if it may be called so, is that they went a step further and surmised that these limitations extend to the general multi-layer and non-linear case. A thing that most people accepted at the time - and just happens not to be true. This misconception resulted in a considerable slackening of the pace of research in neural networks for the next two decades.

Between this time and the late 1970's, when a new interest in neural networks was aroused, the works of Kohonen in correlation matrix memories [13] and Grossberg in mathematical modeling [14] were most prominent.

The 1980's saw a renaissance of neural networks. Some of the interesting developments in this period include; studies in letter perception [1-2], Hopfield networks [15-16], Self-organized networks [17-19], Neocognitron [20-23], Boltzmann machines [24], and last but not far from least, the back error propagation learning algorithm [25-28]. The topic continues to be of great interest and has attracted researchers from various areas of science and engineering.

### 1.3 Some major points on nervous system and neural models:

There are a number of characteristics of the brain [29] that are best taken into account when either forming models for cognitive processes or pursuing the engineering design of a hardware implementation of these models. Most of these points have had a deciding influence on the way the architectures and circuits in this thesis were developed.

1. *Neurons are slow:* Much slower than conventional computing components. The speed of most operations in modern computers is in the order of nanoseconds, whereas in the neurons it is four orders of magnitude slower. This is the basis of the so-called “100-step program” [30], meaning that most useful operations in the brain have to be performed in less than 100 serial steps. Evidently, this points to massive parallelism in the brain.
2. *There are a large number of neurons:* The number of neurons in the brain is estimated at  $10^{10}$  to  $10^{11}$ .
3. *Neurons receive inputs from a large number of neurons:* The number of inputs to each neuron can be between 1000 to 100,000.
4. *Neurons communicate by activation and inhibition:* The information flowing in the nervous system is quantitative, i.e. signed numbers of limited precision and range.
5. *Information is continuously available:* There is no appreciable decision phase during which a unit provides no response. Alternatively, it can be said that the system is asynchronous.
6. *Graceful degradation with damage and information overload:* Partial damage to the nervous system, in the form of disconnection of a few neurons, does not cause a

breakdown of the whole system or even the specific function. Neither does information overflow result in irreparable loss of information.

7. *Distributed control*: All processes are local. There is no central executive overseeing the general flow of information.

#### **1.4 Implementation of neural networks:**

Mathematical modeling and the work of such people as Grossberg and Hebb was fundamental in the development of neural networks. It was only after the advent of the digital computer and availability of numerical simulation, however, that investigators of nervous system and behavioral psychology could actually evaluate the performance of their models in any accuracy or detail. One of the first attempts in this direction was made by Farley and Clark in 1954 [31]. Others soon followed.

Among the more sophisticated software implementations is the Neocognitron by Fukushima [20-23]. This was a multi-layer neural network with 9 layers. Four different types of neural units were used and the learning was supervised. The network was trained to recognize handwritten characters regardless of size and position.

Another successful example in software implementation of neural networks was the NETTALK, a network that learned how to read. This feed-forward, 3-layer network was developed in 1986 by Sejnowsky and Rosenberg [32]. Boltzmann machine and back propagation learning rules were both applied with comparable results, the difference being that back propagation was faster in learning. After 12 CPU hours on a DEC VAX, it was able to produce phonemes from the training set which were correct 95% of the time.

It is generally not very difficult to simulate a known network on a digital computer. The amount of time needed to run such a program, however, can be immense. This stems from modeling a highly interconnected parallel system with serial hardware. Implementation of neural networks with special purpose hardware has been addressed many times during the past 3 decades. Probably the first attempt at making a learning machine was made by Marvin Minsky in 1951. This machine, made at Harvard, had a memory consisting of 40 control knobs, which were moved by a single motor through electric clutches. It had some 300 tubes and in Minsky's own words [33] was never thoroughly debugged, but worked nonetheless (robustness).

Rosenblatt established himself as one of the pioneers of this area in early 1960's when he and his colleagues in Cornell University built a "Mark I" perceptron with 400 photoreceptive sensors on a 20x20 array. This perceptron had 152 associative units and 8 binary response units for the final classification. Each sensory unit had up to 40 random connections to the associator units. Some important results from this machine appeared in a later paper by Block [34].

The major problem in the realization of neural networks is the sheer amount of hardware necessary to implement even the simplest of functions. In recent years, a great deal of interest has been aroused in optoelectronic hardware as a means of implementing neural networks. The Hopfield network and the linear associator are among the networks successfully implemented using this method [35-36]. Optoelectronics is well suited to neural networks. Optical systems are fully parallel and allow a high degree of interconnection, because crossing light beams do not interfere. Also, a large number of active elements can be implemented with comparative ease, since optical devices like lenses and gratings can operate on a number of light beams simultaneously, without increasing the complexity. Mixing optics with electronics combines the best of two worlds: massive

parallelism of optics and flexibility, high gain and decision making properties of electronics. It seems more attractive to form analog neural hardware by completely optical means. However, in the absence of fully optical decision making devices (e.g. sensitive optical bistable devices) the capabilities of the optoelectronic approach remain very attractive [37].

### **1.5 VLSI implementations:**

VLSI is also a suitable medium for the implementation of neural networks. Neural networks typically contain a large number of similar, relatively simple units, which goes very well with VLSI. At the present time, the most successful semiconductor technology for computer logic and memory is digital CMOS. Other technologies cannot match the outstanding integration scale and low power dissipation of CMOS. Its only weakness is speed, and that is improving steadily. Because of the maturity of this technology, some attempts have been made to use it for the implementation of neural networks [38-39]. Among these are wafer scale designs [40], and those trying to achieve a similar mode of transmission as biological neural networks by using pulse stream arithmetic [41]. The advantages of the digital approach are:

- A) Design techniques are advanced, automated, and well understood.
- B) Programming of weights can be managed easily.
- C) Interchip communication and possible exchange of information with a host computer can be readily performed.
- D) Digital memories are relatively easy to build and the weight storage problem, from which analog implementations suffer, does not appear in the digital case.

On the other hand, neural networks are analog in nature and, intuitively, it seems that an analog implementation would be more suitable and elegant. One of the properties of neural networks is that they use signals that have limited precision and range, and their environment can be very noisy. Therefore, precision of variables and noise immunity, the most powerful aspects of digital systems, do not seem essential.

Also, many of the functions needed in the network may be approximated by a combination of a few semiconductor devices on a chip, making use of their characteristic curves. In a digital system, the implementation of a non-binary function usually necessitates a cumbersome lookup table approach. The number of interconnection wirings will also be much smaller in an analog system than in a parallel-bus digital version. Finally, digital multipliers occupy larger silicon areas than comparable analog multipliers.

Carver Mead [42-44] and others [45] have tried to apply analog CMOS to the simulation of biological neural networks, especially sensory organs and early processing elements. Some other works [46-47] have been directed to the development of networks and algorithms that are more compatible with the strengths and limitations of CMOS VLSI.

Before any practical engineering application of analog neural networks can be realized, the problem of weight storage has to be addressed. Almost all of the information in a neural network is stored in the connection weights, and to store these analog values is by no means trivial. At present, no reliable nonvolatile analog memory is available, and the choices are more or less limited to the following:

- A) Programmable threshold devices, i.e. floating gate and metal-nitride-oxide-semiconductor (MNOS) devices [48]. The attractiveness of nonvolatile storage on these devices is reduced by the difficulty in making small weight changes [49], limitation to a finite number of WRITE cycles, and inaccessibility of the technology



[50]. In neural networks, unlike conventional EEPROMs, a large number of bits have to be written at the same time. This makes the WRITE circuitry even more complex. Also, the aging problems of floating gate and especially MNOS devices have yet to be resolved.

- B) Dynamic analog storage on a node or a MOS capacitor. Storage on a node can be useful for the purpose of demonstrating the feasibility of an algorithm, or when the system is adapting constantly. Experiments on this method [51] have shown that by cooling the chip down to cryogenic temperature, weights can be maintained for hours. Dynamic storage on MOS capacitors can be attractive once some compromises are made in terms of layout area and environmental requirements of the circuit [50] (to reduce the effect of leakage mechanisms). If the capacitors are large enough, and a reasonable control over the temperature is maintained, leakage can be reduced to a level low enough so that analog refresh from digital memory using conventional D/A techniques will be possible.
- C) Multi-level storage on a capacitor [52]. In this method, the capacitor holding the analog value is allowed to take any of a set of  $U$  discrete voltages, which are usually equally spaced. If  $U_{i-1}$  and  $U_i$  are the elements of  $U$ , such that  $U_{i-1} < V_{C0} < U_i$ , then the system sets the voltage  $V_C$  to  $U_i$ . Once  $V_C=U_i$ , it remains locked at that value by regular refreshes.
- D) Hybrid solutions by digital storage of weights. In this method, multiplying digital to analog converters (MDAC) are used to perform synaptic multiplications. This has the advantage of reliability and permanent preservation of weights without any great compromise. The drawback of this approach is the limited number of bits in dynamic range, which stems from mismatching of the devices on the chip. This will have an adverse effect on the convergence of most of the learning algorithms.

This method is ideal for applications where constant adaption is unnecessary. It will be shown later that fairly complex recall problems can be solved with as little as 8 bits of precision (including the sign bit).

## **1.6 Handwritten character recognition and feed-forward networks:**

Handwritten character recognition is an engineering problem with applications in such diverse areas as postal services and banking. Most of the character recognition problems follow the solution flow as described below [53]. Each of these steps may be performed in a number of different ways .

The raw data is usually in the form of digitized images, with multiple gray levels. This image goes through a preprocessing stage which consists of thresholding, normalization, and smoothing, not necessarily in that order. Thresholding reduces the multiple gray level image to a binary image, easier to handle for the later stages of the process. Normalization brings all the data samples to the same size so that differences in size will not affect the final decision on character type. Smoothing will add or subtract some dark pixels in the image so as to remove the roughness of the edges of the characters. This is actually analogous to a form of integration and serves to reduce spurious, useless data in the image.

The next step is feature extraction. The binary image at this point still contains a huge amount of information. In a typical 512x512 binary image, this amounts to 262,144 bits. If all this information is used, it will lead to over-classification. Insignificant and/or unimportant differences in bit patterns may cause two different images of the same character to be recognized as different characters, or result in other undesirable effects. Extraction of

features reduces the data to a manageable size and retains only those fragments of information that are most necessary and helpful in classification.

In the last stage, the features of the particular image are compared to the features of one or more known images from a "training set", and a decision is arrived at as to the type of the character. In the statistical classifiers like n-nearest-neighbor and its variants, a number of distance measures from the present features to the features of the training set are computed and used as a criterion in decision making. In feed-forward neural networks, the features fed at the input are changed to an internal representation [54] at the hidden layer(s) and the answer appears at the output. The information content of the training set is distributed throughout the network in the form of connection strengths and no explicit comparison takes place. Neural networks can compete very well with the traditional classifiers.

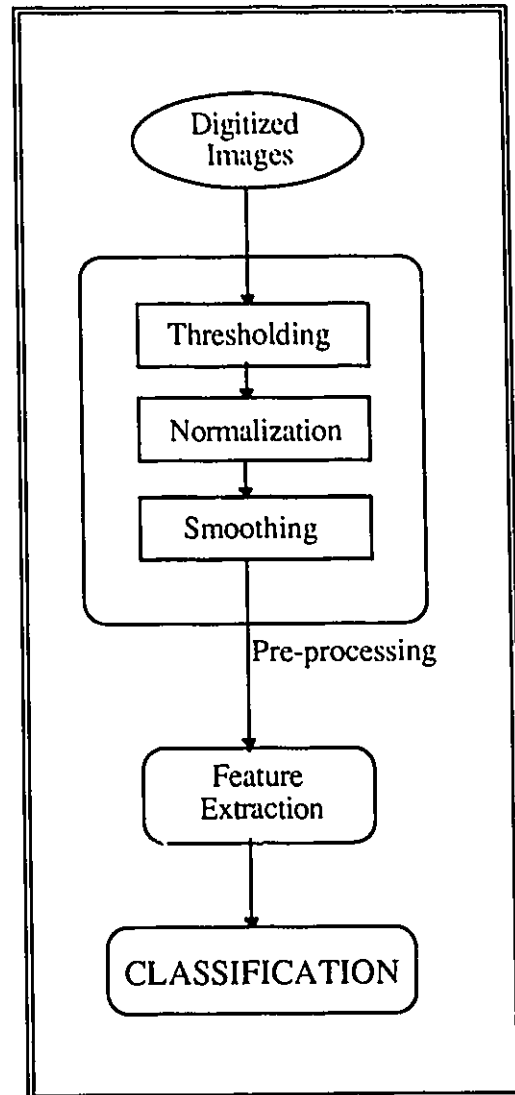


Figure 1.1: Character Recognition

Although it would be very attractive and elegant to perform the recognition without feature extraction, i.e. directly from the visual data, attempts in this direction have been unable to

produce recognition rates acceptable for practical applications [55]. Also, it is worth mentioning that the feature extraction approach has some merit in the way of being nearer to what happens in nature. It has been shown that many preprocessing actions take place in the early vision system, including, for instance, edge extraction [56].

### **1.7 Summary:**

The objective of this thesis, in the next few chapters, is to introduce an architecture for feed-forward neural networks. This type of network is very suitable for classification and can be used in applications such as pattern recognition. In this architecture, the synaptic weights are stored in a digital memory structure and a multiplexing scheme is used to reduce the number of physical interconnections. In this chapter, a general overview of neural networks and some historically significant research efforts in modelling and implementation of these networks were presented. Some important points about neural models were given and, as the context of this work, a general flow of the solution to character recognition problem was outlined.

---

## An Architecture for Multi-layer Neural Networks

---

Multi-layer networks are a popular and important subclass of neural networks. Because of their simple dynamics, which stems from a lack of feedback paths, these networks are inherently stable. Also, the existence of powerful learning and adaptation algorithms for these networks makes them even more attractive from an engineering point of view. Supervised learning schemes on feed-forward multi-layer networks, such as back error propagation, are particularly attractive for applications such as pattern and speech recognition, waveform classification, etc.

Analog VLSI is a promising vehicle for the implementation of these networks. Large scale integration makes it possible to put many electronic elements on a semiconductor chip with better reliability and lower cost. VLSI fabrication is particularly compatible with designs with a large number of similar and simple elements; a central characteristic of neural networks. However, another inherent property of neural networks, high interconnectivity, has proven to be one of the major obstacles in the way of hardware implementation of large networks. It has been known for some time that the interconnections limit the size of

neural networks on chips. Analog technology has the advantage that the maximum information capacity of a single line is only limited by noise and other uncertainties<sup>1</sup>. When the signals are represented digitally and transmitted in parallel, each activation or node value needs several wires for interconnection, whereas one wire suffices in an analog system.

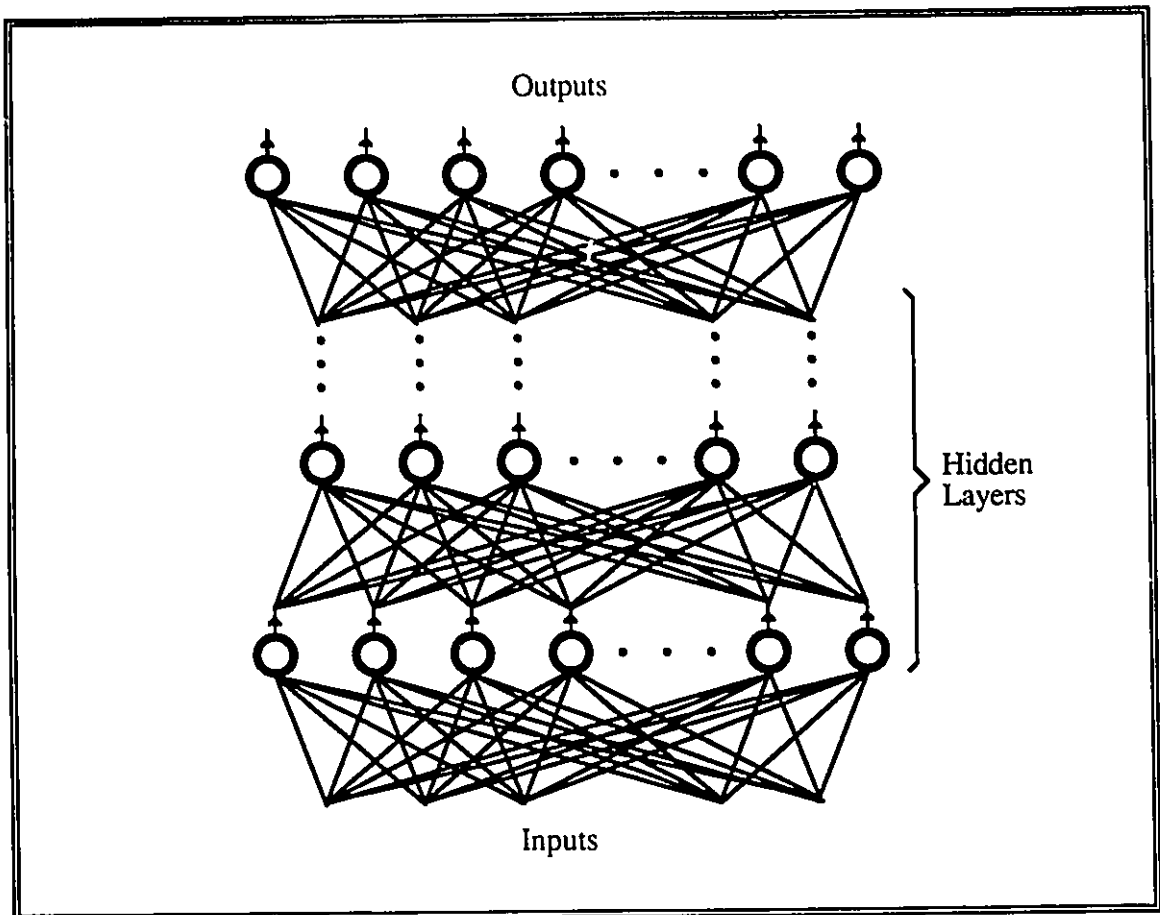


Figure 2.1: Multi-layer feed-forward network

In the design of analog neural networks, one of the most difficult issues is the storage of synaptic weights. Although some studies have been made in this area, a reliable, compact analog memory in CMOS technology, that can preserve data with acceptable accuracy for

<sup>1</sup> In the strict sense, the bandwidth should be included in any information analysis. Here, however, we only refer to the information content at a given point in time based on the allowable levels on a line.

long periods of time, is not available. As indicated in chapter 1, a number of analog memory designs have been presented in the literature. Furman and Abidi [51] presented a design for a feed forward network with back error propagation. The weights are stored as charges on small capacitors on the nodes, and can be preserved for hours by keeping the chip at cryogenic temperatures. Borgstrom, Ismail and Bibyk [48] presented a circuit technique involving floating gate MOS transistors. The weights are stored as charges in the floating gates. Schwartz, Howard and Hubbard [50] introduced a circuit utilizing dynamic charge storage on MOS capacitors. The weights can be preserved on the order of seconds. Some other designs [57-60] are also present in the literature.

All these methods present problems that make the design of a practical network, with suitable size for solving practical problems, very difficult. A realistic engineering design has to have at least tens of neurons on a chip, work in normal temperatures, and preserve connection strengths during the operation of the system. This necessitates a certain degree of efficiency in the utilization of chip real estate. The designs that use dynamic charge storage on capacitors have to use up large areas for the capacitive elements to ensure a reliable preservation of weights at different temperatures. Multilevel storage on capacitors involves a considerable amount of overhead in circuitry to keep the capacitors at the locked voltages. Floating gate memories show large variations in transistor characteristics and need large voltages on the chip for programming.

Digital weight storage has been addressed by Raffel et al. [61] and others. This is the method chosen for the architecture presented in this thesis. The most general form of multi-layer networks, without feedback, is implemented (figure 2.1). Most of the functions are realized in analog current mode circuitry, which eliminates the need for adder hardware.

## 2.1 The architecture:

The general form of the architecture is shown in figure 2.2. Each of the layers has a structure similar to that shown in figure 2.3. Here, we define a stage as a layer of neurons and their corresponding connection strengths. Each set of connection strengths is associated with the neurons of the next layer, instead of those of the preceding layer. Therefore, each of these stages consists of the neurons at that layer plus all the connection strengths from the previous layer. In this way, the number of physical interconnections between any two stages  $i$  and  $i+1$  are reduced to the number of neurons in stage  $i$  (figure 2.2). This possibility should be apparent intuitively: once the weights are fixed, the amount of information passed to a layer does not exceed the information present at the output of the preceding layer, and the minimum number of lines needed to carry this information is no more than  $m_i$ .

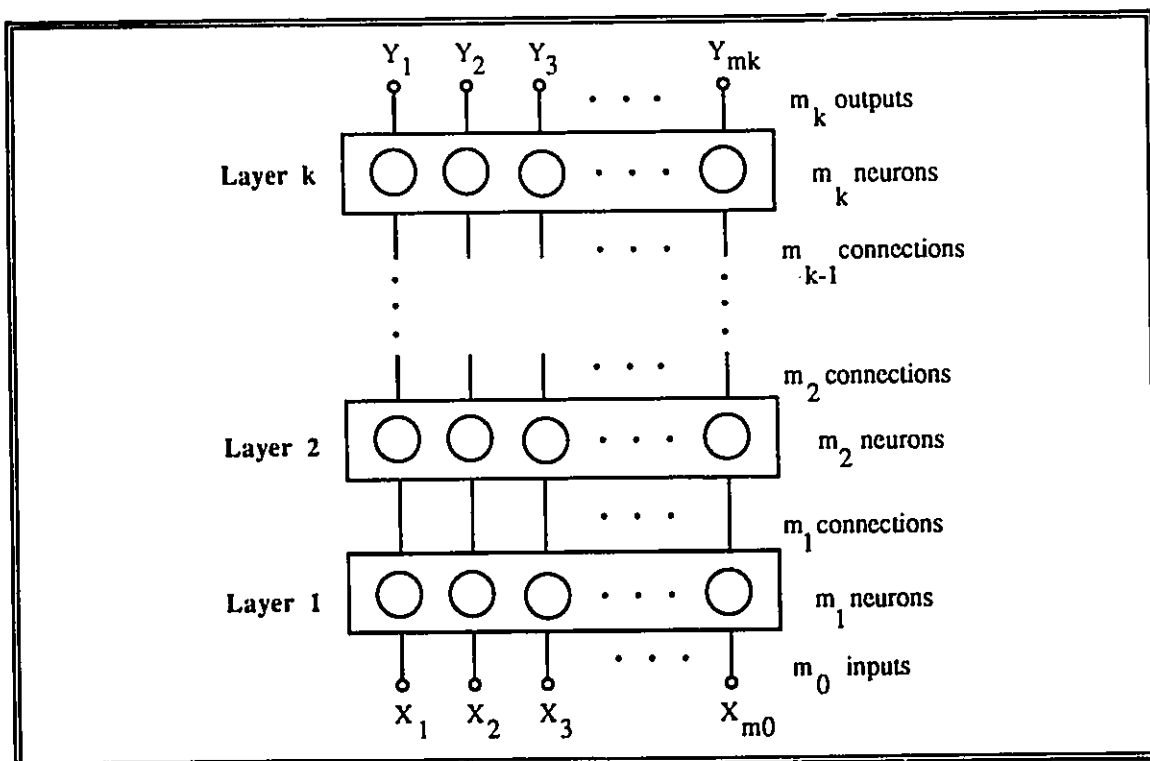


Figure 2.2: Layers and stages in the architecture



With this setup, a multi-chip system can also be considered. In such a system, each of the stages would be implemented on a separate chip. In other schemes, this has to be avoided because of the high number of interconnections. In this architecture, interstage wiring has been dramatically reduced and much larger systems can be designed using a number of chips, each of which constitutes one stage of the network. This can also lead to a semi-custom approach to the design of these networks, where only one generic chip containing one stage is designed. This chip will contain the maximum number of neurons possible in the technology in use (e.g.  $3\ \mu$  or  $1.2\ \mu$  CMOS). It will also provide the maximum number of interconnections to another stage with the same number of neurons. The user can order these chips and program his/her own network. If the number of neurons in any stage is greater than needed, they can be masked out by assigning zero to the incoming weights.

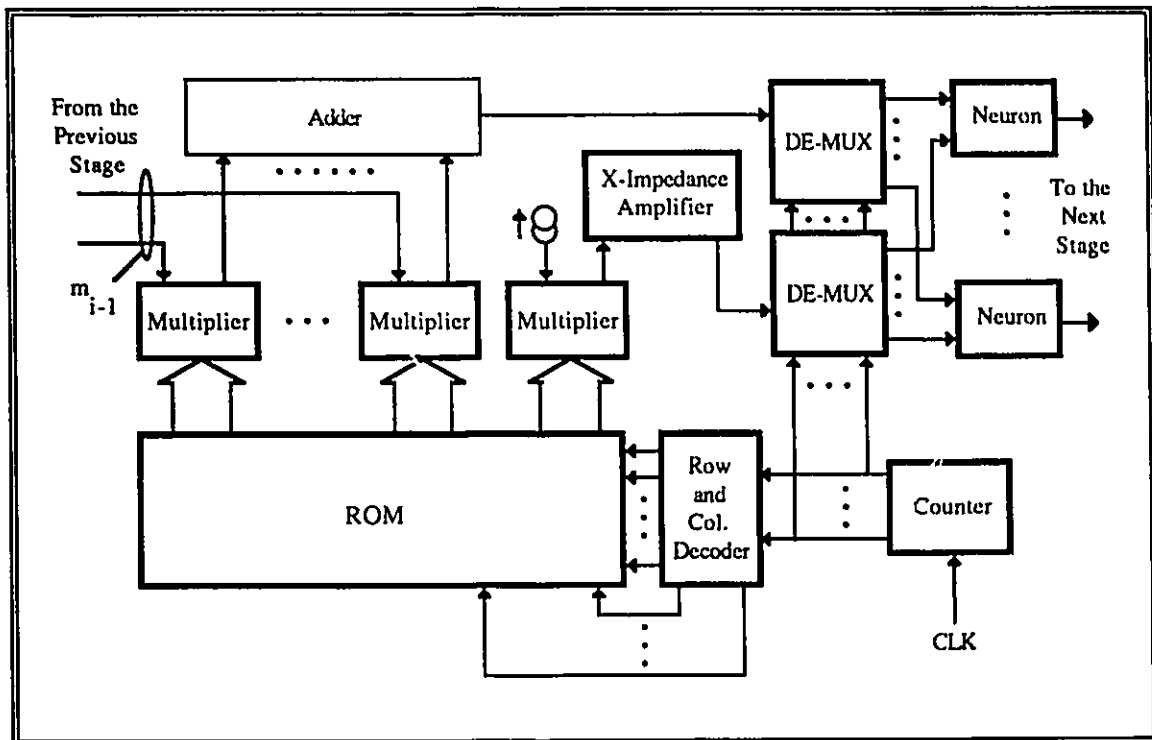


Figure 2.3: Internal structure of one stage

The central idea in this scheme (figure 2.3) is an imbedded multiplexing of the connection strengths. This is possible because the bandwidth of the analog signals in the network is, or can be made, much smaller than the clocking frequencies available in digital CMOS. The feed-forward networks that we consider do not have feedback. Therefore, the delays introduced by multiplexing do not disrupt the network dynamics and only increase the time latency. The most important constraint is that the analog nodes should be refreshed in time so that the output of the neurons are kept valid at all times. The minimum clocking speed depends on the leakage of the nodes, itself dependent on temperature, and is no more than a few hundred KHz; quite practical.

Currently, the value of the weights is stored in a mask-programmed static ROM. Eventually, this can be changed to a PROM or EPROM for field programmability and easy customizing. They can also be stored in a RAM for a programmable version.

## 2.2 Operation:

The ROM storage, at the logical level, can be considered as an  $(m_{i-1} + 1) \times m_i$  two dimensional array. The counter is reset after each  $m_i$  clock cycles. Suppose that at an arbitrary point in time, the output of the counter is equal to  $j$ ,  $0 \leq j \leq m_i - 1$ . The  $j$ th column of the ROM is fed to the digital inputs of the multiplier array. The multipliers are current mode multiplying digital to analog converters with the multiplier in digital form and the multiplicand in analog (current mode) form.

At the same time, the output node values of stage  $i-1$ , in the form of current, are available at the analog inputs of the multipliers. The weighted current mode outputs are added on a node, and the result of this addition is converted to voltage by a trans-impedance amplifier

(figure 2.4). This value is then passed to the  $j$ th neuron by a demultiplexer, and the activity of this neuron forms one element of the input of the next stage. The input nodes of the neurons have capacitances that hold the value of their input until the next demultiplexing cycle.

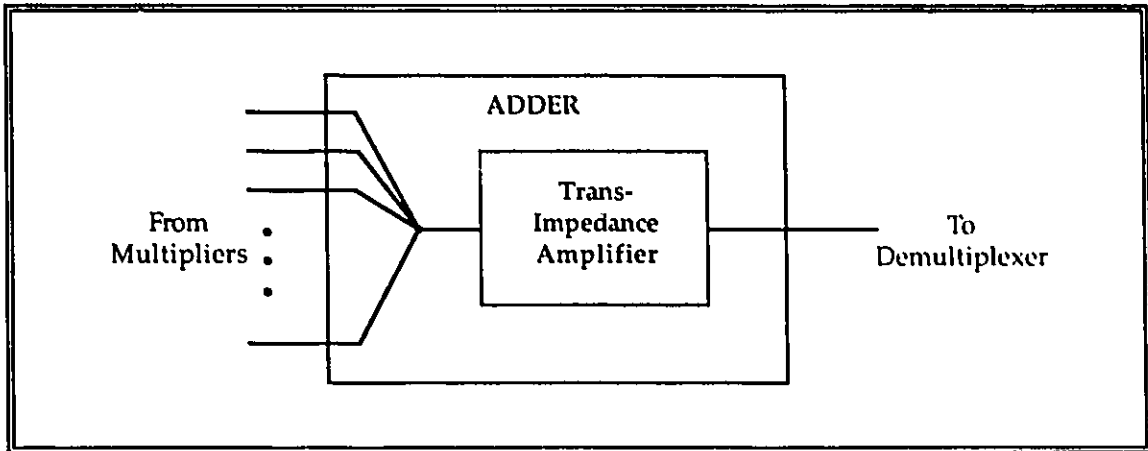


Figure 2.4: The adder

Each of the neurons has two inputs: post-synaptic activation and threshold value. The threshold of the neurons can be considered as a negative signal coming from a source with a strength of unity and a weight equal to the threshold value. One row of the storage is allocated to the threshold values and the signal is distributed using the same demultiplexing method described above. The input of the first  $m_{i-1}$  multipliers are summed and fed to a demultiplexer. The output of the multiplier  $(m_{i-1}+1)$ , which represents the threshold, is input to the other demultiplexer. At each clock cycle, the two demultiplexers forward these signals to the inputs of one neuron, providing the refresh of both the activation and threshold at the same time. In order to avoid mismatching, the current source shown in figure 2.3 is actually another neuron with signal and bias inputs connected to  $V_{DD}$  and ground respectively.

This multiplexing scheme enables the reduction of the number of multipliers in each stage from  $(m_{i-1} + 1) \times m_i$  to  $m_{i-1} + 1$ . The same reduction factor holds for the interconnections between layers.

As always, there is a price to be paid for convenience. The settling time of the network is the factor that is adversely affected when going from a full implementation to one with multiplexing. In the worst case, where all connection strengths are non-zero and all inputs to all neurons change at a point in time, the settling time will be:

$$T = \sum_{i=1}^K \frac{n_i}{f} \quad (2.1)$$

Where:

$T$  = Total settling time

$K$  = Number of layers

$n_i$  = Number of neurons in stage  $i$

$f$  = Clocking frequency

In a typical application for handwritten character recognition, a network with the following specifications was designed and trained with the back error propagation algorithm:

Number of active layers	3
Number of inputs	36
Number of neurons	20, 15, 10

Setting the clocking frequency to 6MHz, the settling time will be equal to 7.5 microseconds. During this time, 3195 multiplications are performed, which translates to  $426 \times 10^6$  multiplications per second.

### 2.3 Modifications:

In the original form, the neurons have two inputs each; post-synaptic activation and threshold. This is similar to the structure of the nervous system, where the threshold is the property of individual neurons and is applied locally. A more effective and accurate approach, however, is also possible, as shown in figure 2.5.

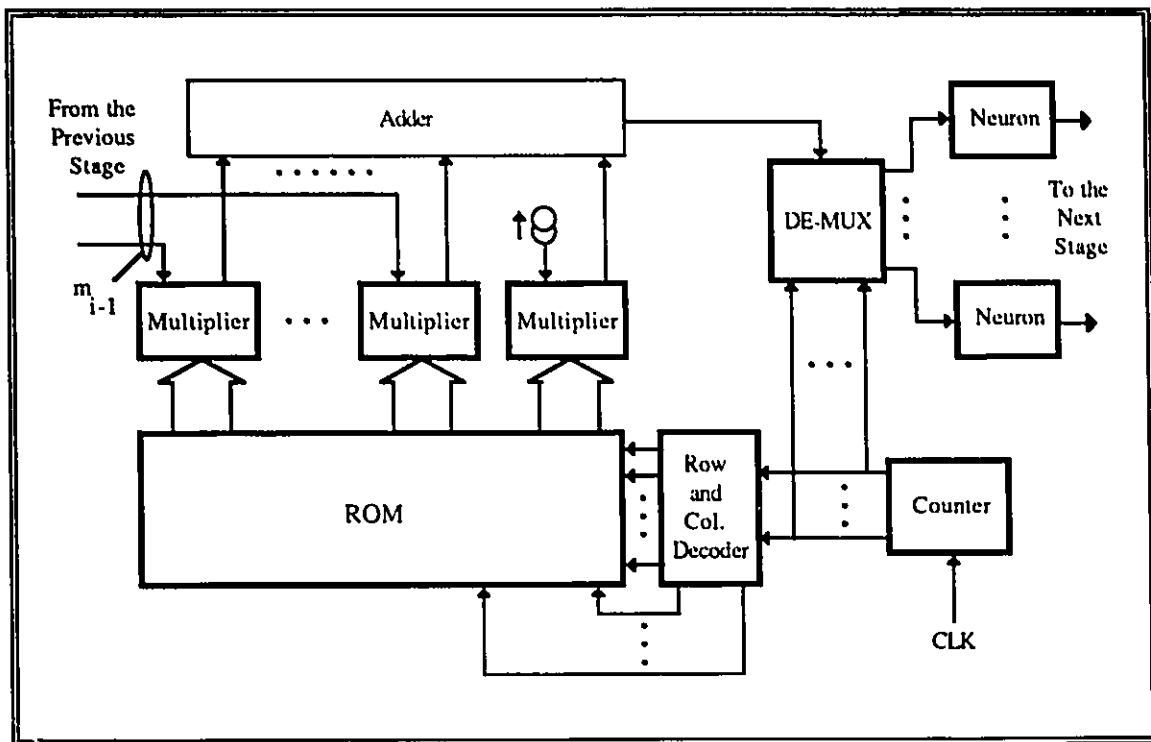


Figure 2.5: Modified architecture

The variable threshold neuron of the original structure makes use of a differential pair of MOS transistors at its input stage; this will be discussed in greater detail in later chapters. The transfer characteristics of this neuron, as a function of  $(v_1 - v_2)$ , is not the same over a large range of  $v_2$  (figures 3.3 through 3.5). In other words, the characteristics of the neuron changes at different threshold values. This undesirable effect can be prevented by transferring the subtraction performed in thresholding back to the adder node. In this

approach, neurons will have only one input each, and the threshold values will be internally fixed to zero. The thresholds are added, with a negative sign, to the post-synaptic activations. This modification not only results in uniform transfer characteristics and a linear subtraction by current mode circuits on a node, but also saves some hardware. By a comparison of figures 2.3 and 2.5, it is seen that a trans-impedance amplifier, a demultiplexer, and a considerable amount of wiring at the input of the neurons have been eliminated.

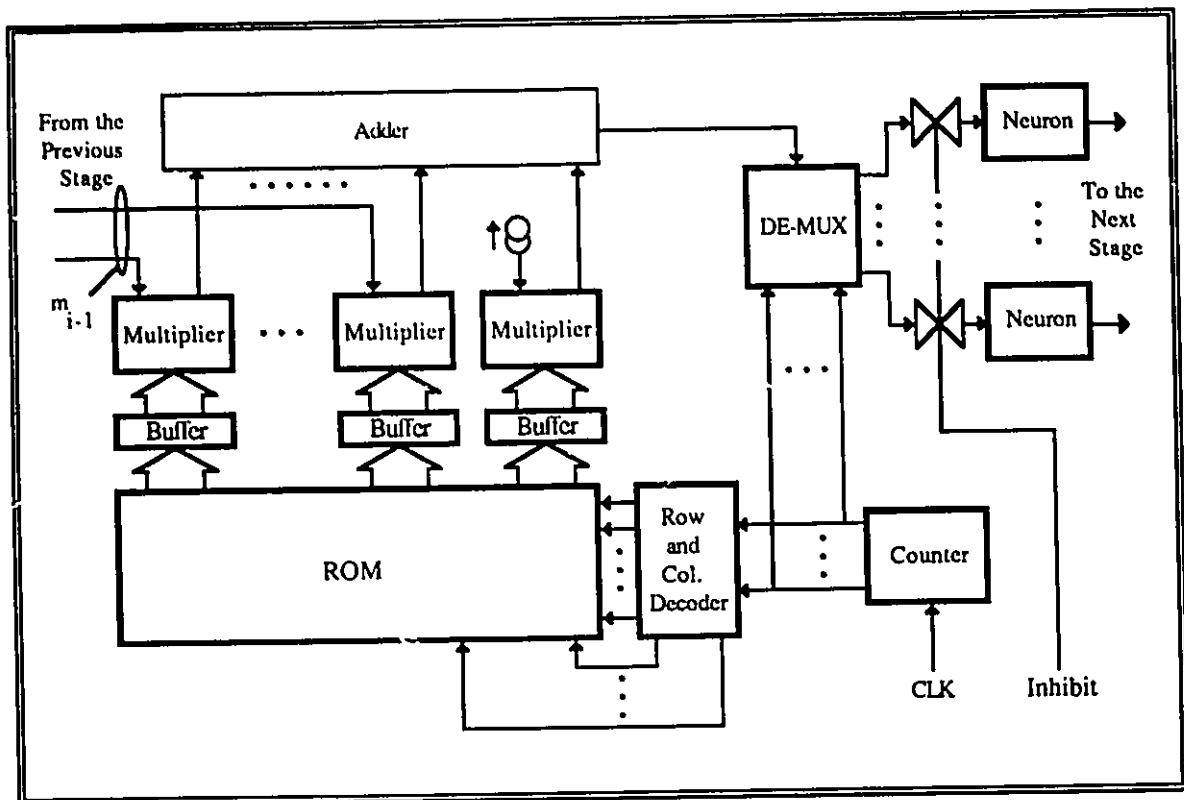


Figure 2.6: Second modification for further improvement

Another set of changes that would improve the performance are shown in figure 2.6. At the output of the ROM array, a series of static buffers are added. The objective is to reduce the length of bit lines to a minimum so that cell and pullup transistors in the ROM have a smaller load capacitance to charge and discharge, hence an acceleration of the response of the ROM. This will allow greater settling time for the rest of the circuit.

An inhibit signal and a switching circuit are also added before the neurons. Because of the finite settling time of the multipliers and specifically the trans-impedance amplifier inside the adder block, the signals at the output of the demultiplexer are not stable at the time the address becomes available. If this inhibition circuitry is not provided, the capacitors at the input nodes of the neurons will receive erroneous charging voltages at the beginning of each refresh cycle. The corresponding voltage spikes in the signals effectively increase the settling time. This will be discussed in greater detail later in the thesis.

## 2.4 Units and scales:

The transfer characteristic generally preferred for the neuron is sigmoidal (S-like). The feed-forward model for the back-error propagation learning algorithm is usually associated with a version of the sigmoid; the so-called logistics function:

$$f(\text{net}) = \frac{1}{1 + e^{-\text{net}}} \quad (2.2)$$

The transfer characteristic of the physical circuit representing the neuron can be approximated with:

$$I_{\text{out}} = \frac{A}{1 + e^{-\text{net}/B}} \quad (2.3)$$

The range of the connection weights also has to be fixed. This range clearly depends on the application. Suppose that the choice is made and the largest weight to be represented is  $W$ . Also, suppose that the designed multiplying D/A will deliver an output current of  $K$  times its input current when all the bits are set to 1, i.e. the largest connection strength  $W$  will result in a multiplication by  $K$ .

The transfer characteristic of the trans-impedance amplifier can be expressed in terms of the amplification factor  $R$ , i.e. the slope of the transfer characteristic at the linear region.

Having the four parameters  $A$ ,  $B$ ,  $W$ ,  $K$  and  $R$ , the mapping relation can be expressed as follows: The unit value of the post-synaptic activation is clearly  $B$  volts. The other parameters have to be adjusted so that an activation of strength unity at the output of a stage, namely  $A$  amps, multiplied by a connection strength of unity and passed through the trans-impedance amplifier will result in a post-synaptic activity of  $B$ . This means:

$$A \frac{K}{W} R = B \quad (2.4)$$

The choice of  $B$  is limited by the power supply voltages and the desire to have as much resolution within this potential difference as possible.  $K$  is determined by VLSI layout considerations (in our case is 2), and  $W$  is determined by the application.  $A$  and  $R$  can be used to balance the equation.

## 2.5 Input/Output considerations:

In each of the stages defined above, the input and output signals are in current mode (figures 2.3, 2.5 and 2.6). Although this does not create any problems inside the chip, it is preferable that the I/O communication is performed in voltage mode. The current levels are in the microamp range and would be severely affected by the noise that can be coupled to the relatively long connection wires to the chip package. A possible remedy to this problem



is the use of current mode output buffers, but the design of these buffers is very difficult<sup>1</sup>. Also, current mode circuits have high input and output impedances. High impedance nodes, especially outside the chip, are susceptible to voltage noise spikes of large amplitude, which will take the current mode circuits to saturation if the voltage of the node goes beyond the power supply voltage.

To change the signals at the I/O of the chip to voltage mode, a trans-impedance amplifier at the input and a voltage to current converter at the output may be used (figure 2.7). The voltage to current converter and the trans-impedance amplifier are designed so that their combined transfer characteristic has a slope of unity. The magnitude of the current mode outputs should also be taken into account when these two blocks are designed, so that none of them will be driven into a nonlinear zone in the normal course of the operation of the network.

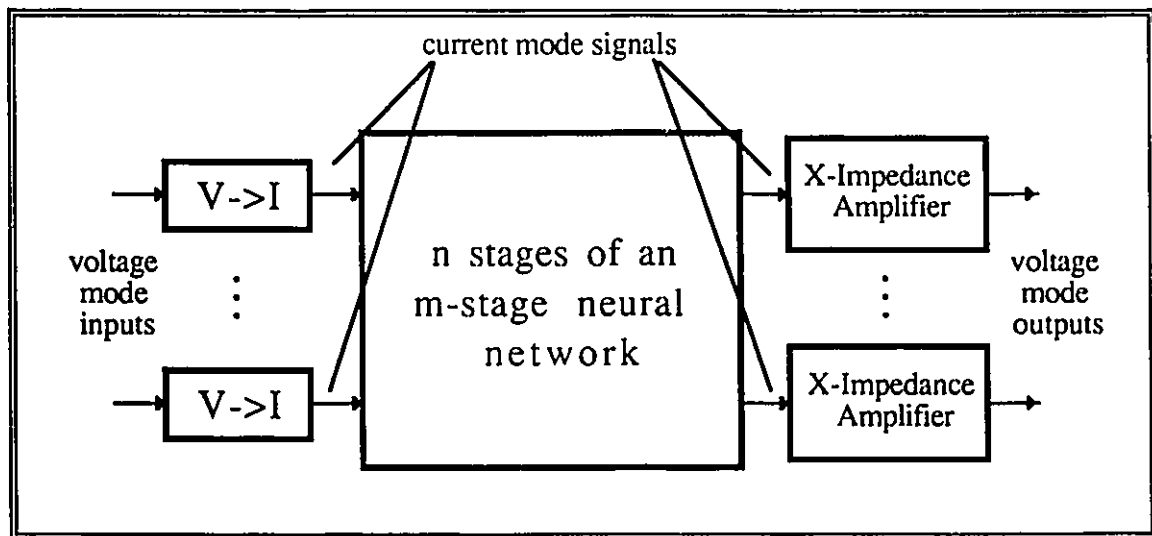


Figure 2.7: Conversion of the signal mode at input and output

<sup>1</sup> Generally, feedback is needed to insure the linearity of a buffer. In this case, the appropriate feed back would be current-parallel, which needs linear resistors, almost impossible to make in CMOS. Stability would also be a problem.

At the output, another variation is possible which saves some silicon area. Voltage mode neural blocks<sup>1</sup> can replace the current mode neurons and the trans-impedance amplifiers at the last stage. This approach has an additional advantage: Feedback is used in trans-impedance amplifiers to achieve linear, stabilized I-V characteristics. Feedback, however, also results in performance degradation in the network dynamics. Most of the propagation delay of the network is due to the settling time of the trans-impedance amplifiers. By eliminating these extra amplifiers, the total network delay can be reduced.

There is a disadvantage associated with the incorporation of the voltage mode neuron into the network. The transfer characteristic of the voltage mode neuron is slightly different from the current mode neuron and is almost linear<sup>1</sup>. This small error may sometimes not be acceptable and the choice of one solution or another will depend on the particular application.

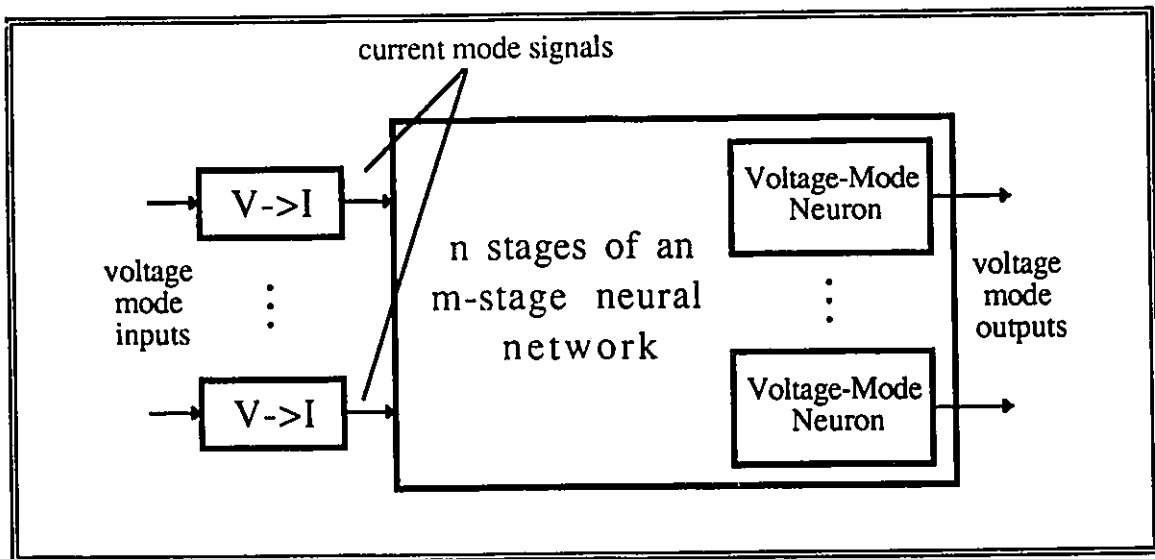


Figure 2.8: Second solution to the I/O problem

<sup>1</sup> Ref. to chapters 3 and 4 for complete description of this and other circuit blocks

In large multi-chip networks and in applications where absolute economy in silicon usage is necessary, another approach becomes attractive: Consider a case where the neurons of each stage are transferred to the next layer, i.e. neurons are located before the synaptic multipliers instead of after (figure 2.9). Since the input of the neurons are voltage mode, and the demultiplexer switches voltage signals, this new stage will have voltage mode inputs and outputs.

This end is achieved without using any extra blocks. The regularity of the architecture, however, is disrupted. In this modification, the first and last stages will be different from the structure of figure 2.9, because there are no neurons at the input of the overall network and there should be a layer of neurons at the output of the last stage. This scheme is therefore incompatible with the idea of a generic programmable chip for all stages, for at least three types of stages are now necessary: first, last and intermediate.

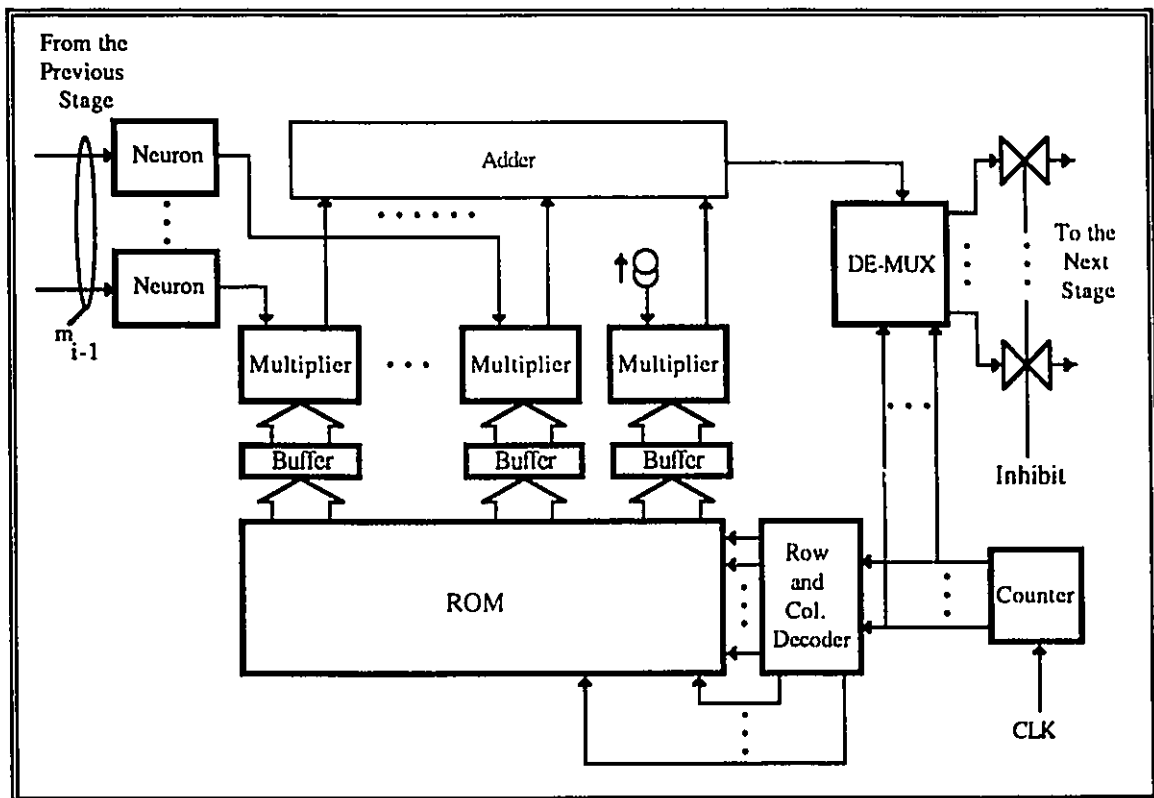


Figure 2.9: Changing the I/O mode by displacing the neurons

## 2.6 Performance analysis on an example:

It is well known that the XOR logic function and its generalized form, the parity function, are among the more difficult functions to implement on neural networks. This difficulty arises from the non-connected regions in the domain of these functions, which necessitate a separate internal representation of the data set. This function was implemented using the proposed architecture to show the validity and workability of the concepts.

The network consists of inputs, a hidden layer and an output layer. The 8 connection strengths and 4 threshold values are to be determined in the learning process. These values were iteratively computed using the back error propagation learning algorithm with the following vectors:

U	V	XOR	XNOR
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

and the result of this training is shown in figure 2.10.

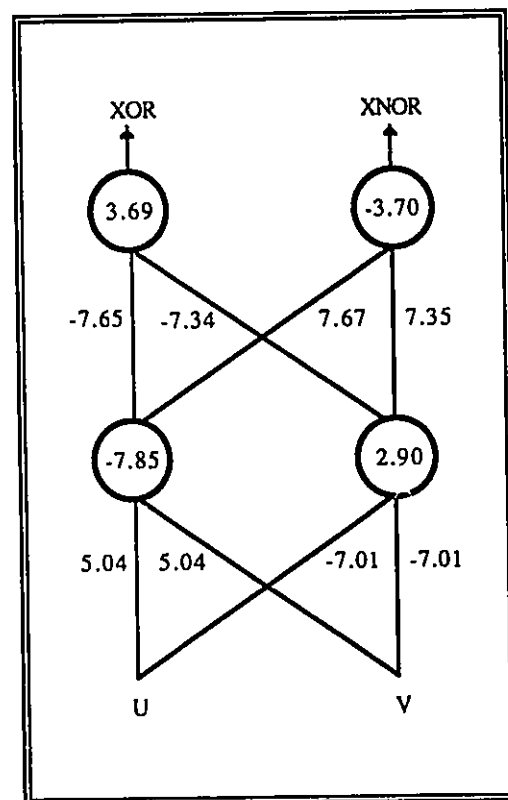


Figure 2.10: XOR and XNOR

The scales in the circuit can be determined in the following manner: The maximum level of synaptic strength and threshold value is set at  $W=8$ , and as mentioned before,  $K=2$ . The neurons have the following characteristics<sup>1</sup>:

$$A=33\mu\text{A}$$

$$B=0.5 \text{ Volts}$$

Therefore:

$$A \frac{K}{W} R = B$$

$$R = \frac{B W}{A K} = 60 \text{ K}\Omega$$

To have voltage mode output, voltage mode neurons were used in the second stage (figures 2.11 and 2.13). Instead of voltage to current converters, two current mode neurons are used at the input. This is permissible because the test vectors are digital and the behavior of this block is unimportant between 0 and 1.

The circuit shown in figure 2.11 is the original architecture, i.e. without inhibition circuitry. Waveforms of the signals named in figure 2.11 are given in figure 2.12. Note the ringing on V1 and V2, the outputs of the trans-impedance amplifiers. There are also large spikes on M1, M2, L1, and L2, as explained in section 2.3. Despite spikes on the output signals, XOR and XNOR functions are clearly visible. Output transitions happen at most 4 data cycles - or, alternatively, 2 clock cycles<sup>2</sup> - after each input transition. The data is presented to the circuit with a frequency of 2MHz. This is a maximum, since even at this frequency, the outputs of the trans-impedance amplifiers are barely stable when the next data cycle arrives (fig 2.12).

---

<sup>1</sup> Ref. chapter 3

<sup>2</sup> Since there are only two memory addresses in each ROM section in this design, the Clock signal and its complement were used for selection and address decoders were avoided.

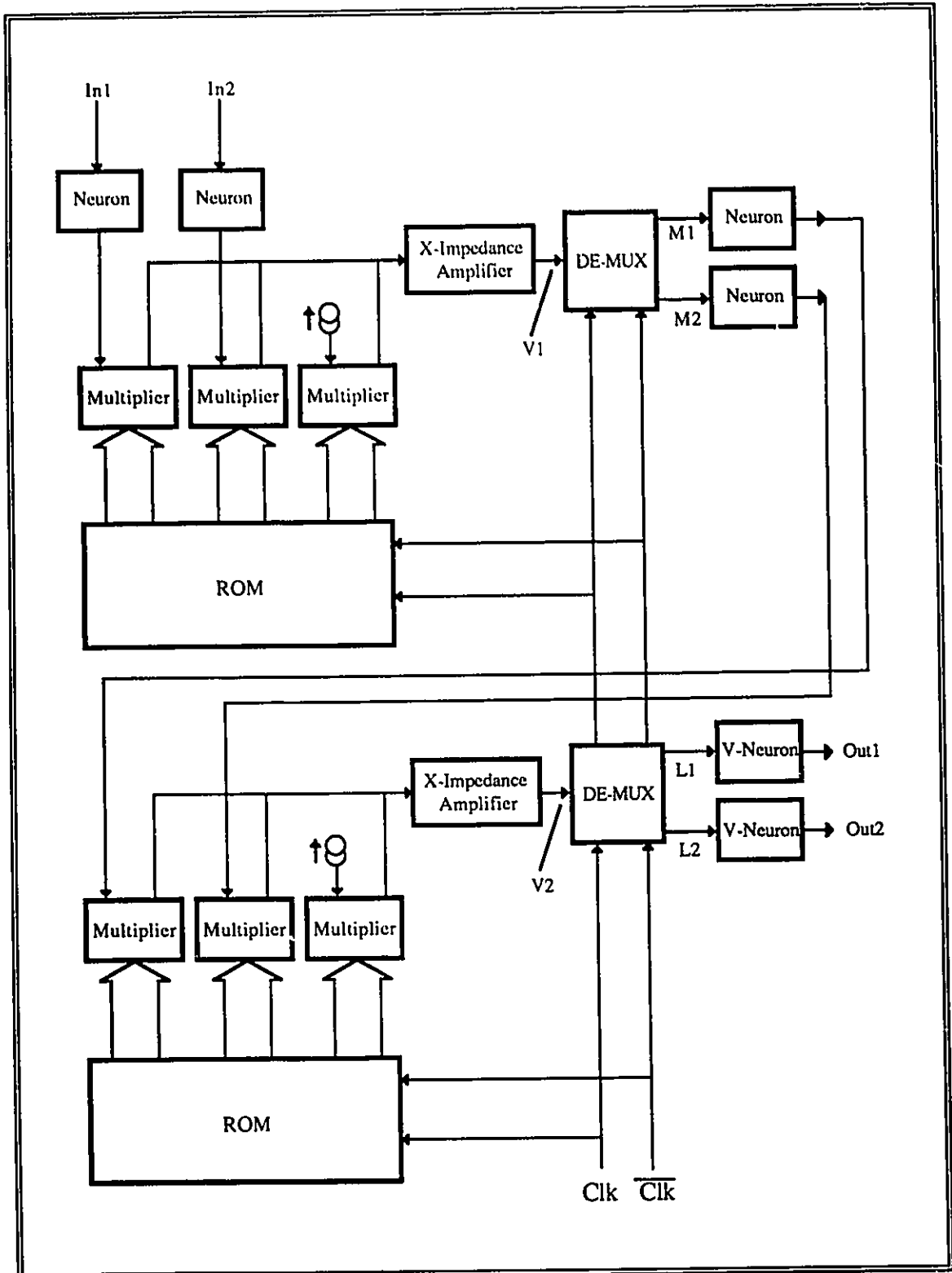


Figure 2.11: XOR-XNOR using the original architecture

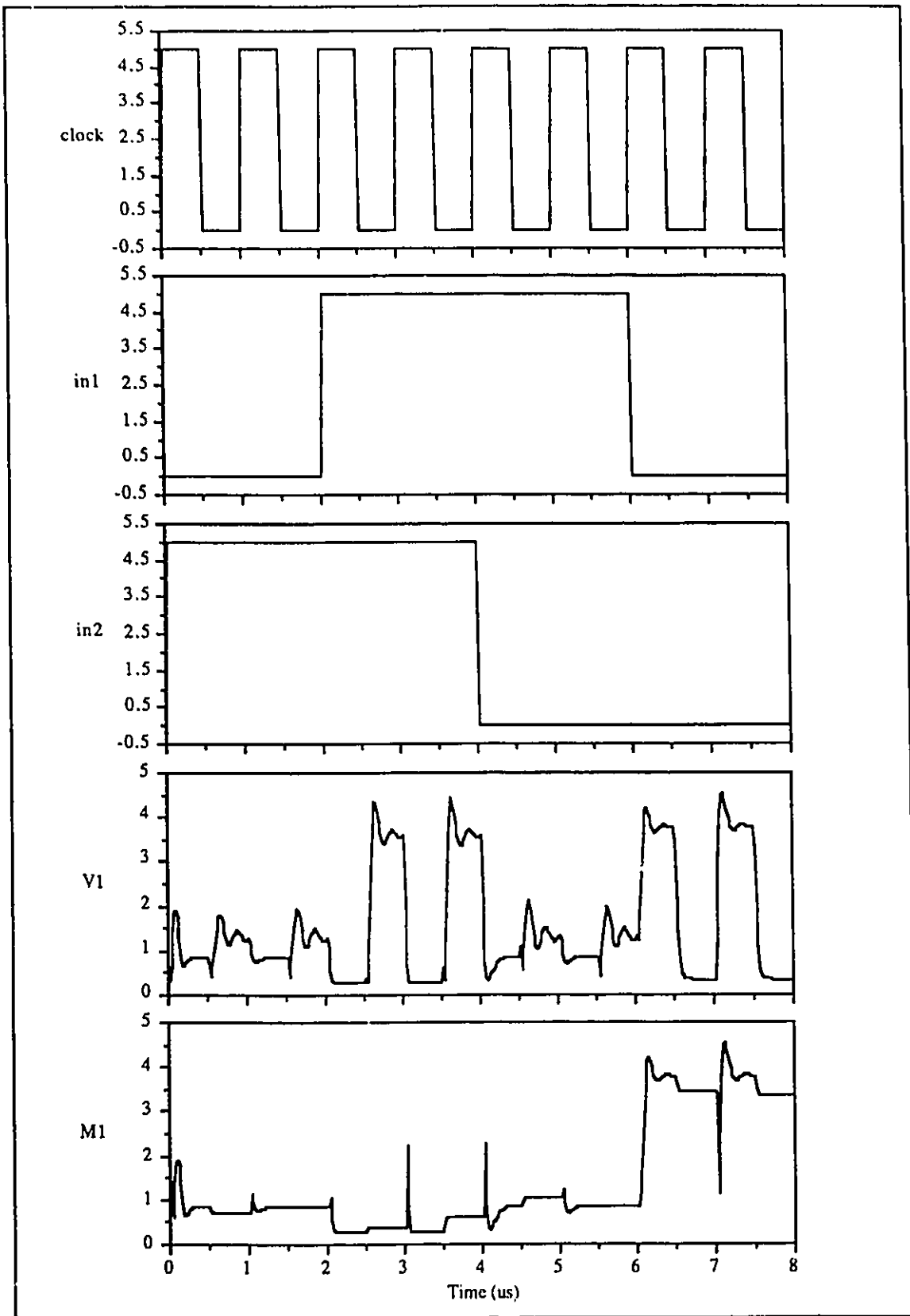


Figure 2.12: (Continued below)

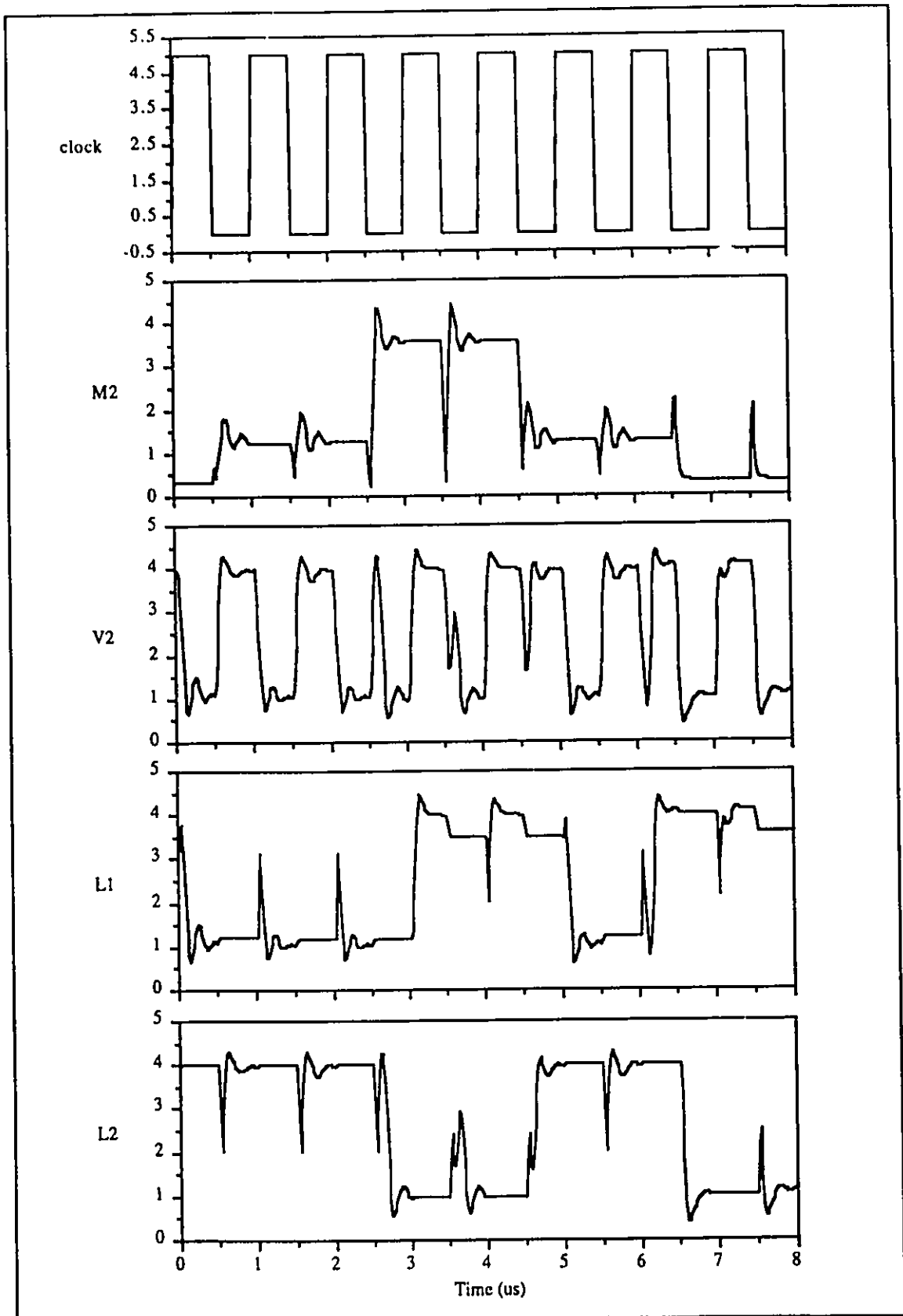


Figure 2.12: (Continued below)



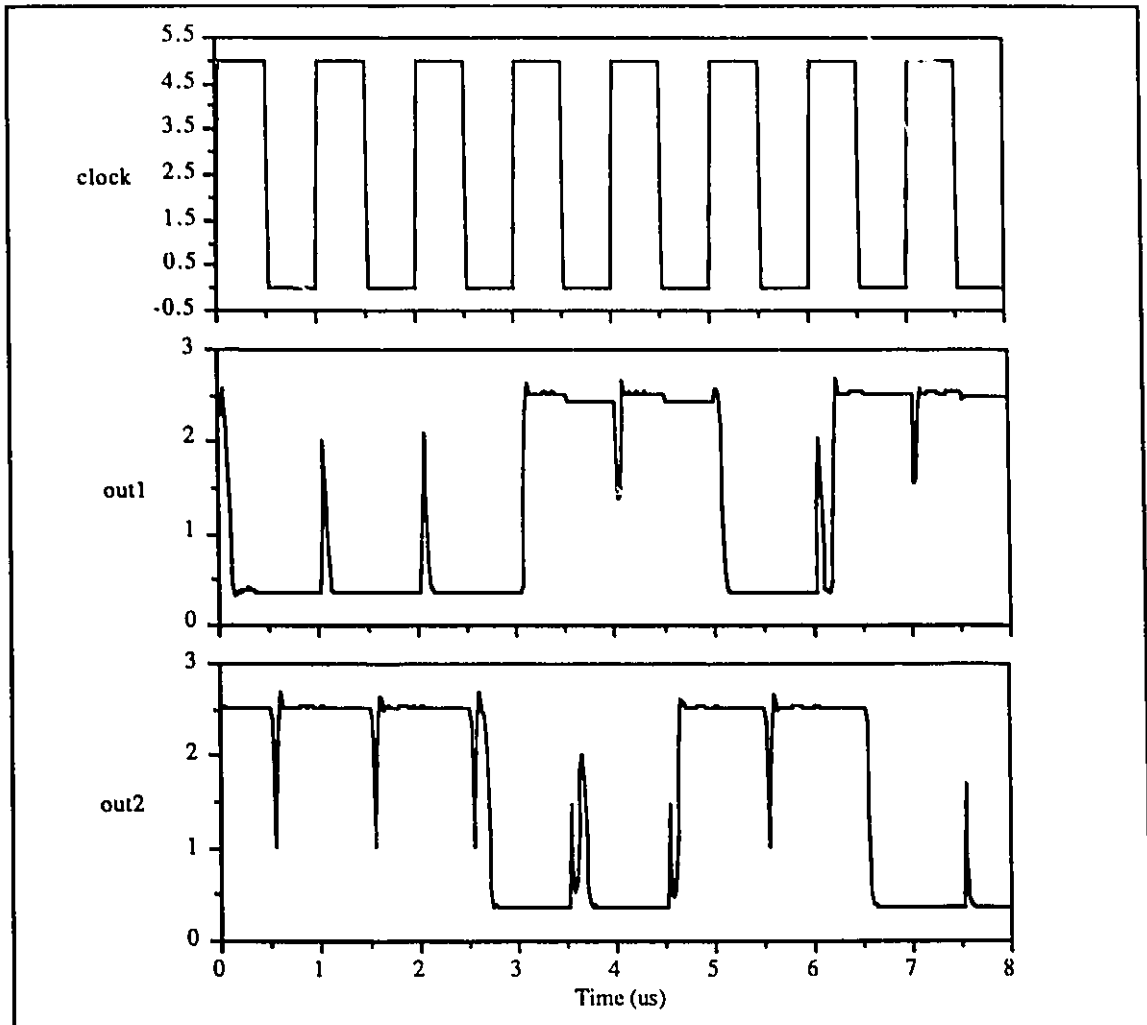


Figure 2.12: Waveforms of the original architecture

Figure 2.13 shows the same example in the modified architecture of section 2.3. A modified circuit for the trans-impedance amplifier, with lower settling time, is used in this version. In order to show the improvements, the waveforms of the last stage are presented in figure 2.14. The clocking frequency is 2MHz, twice the previous case, and the Inhibit signal disconnects the switches around the transition of the clock.

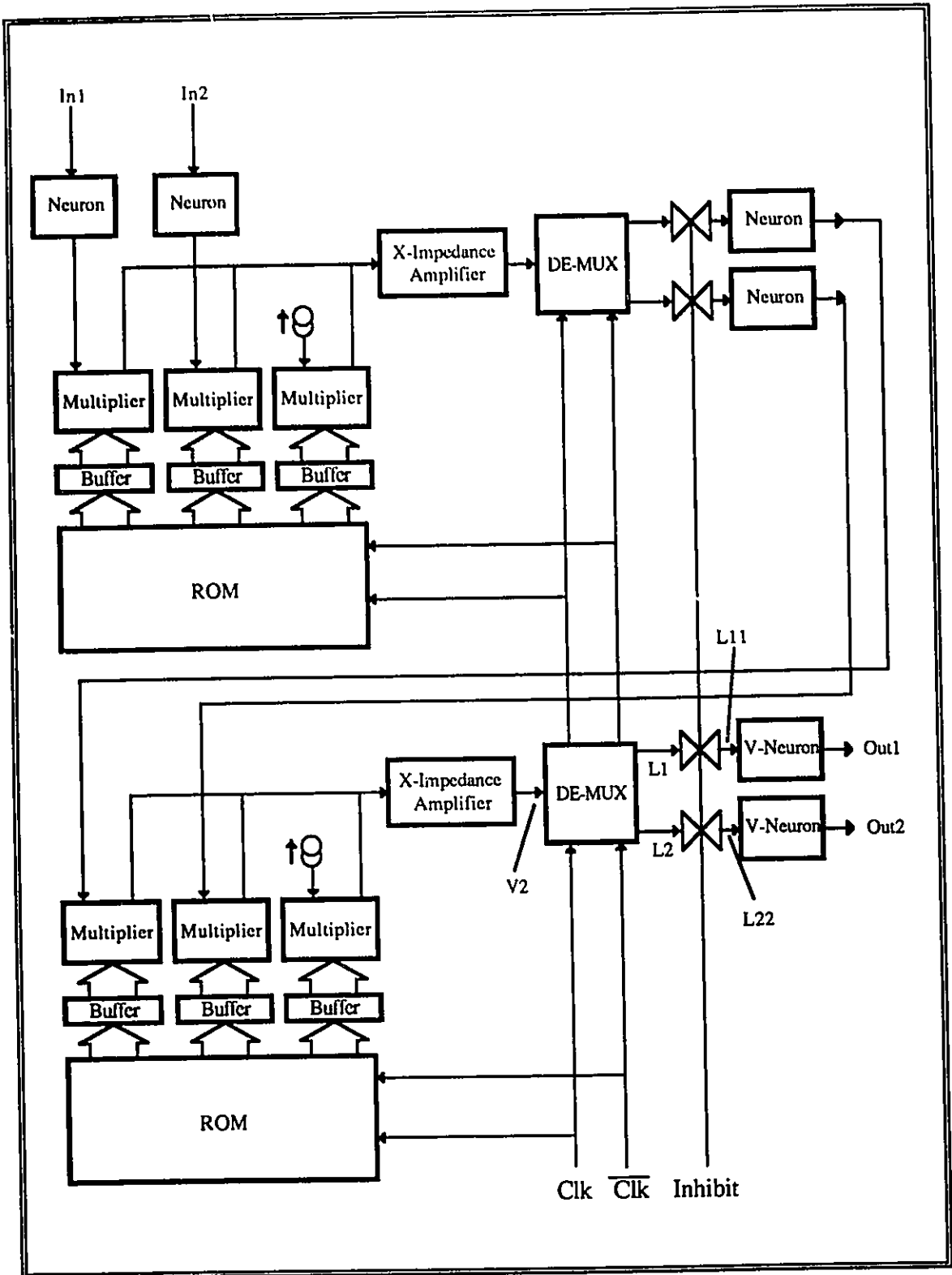


Figure 2.13: XOR-XNOR with the modified architecture

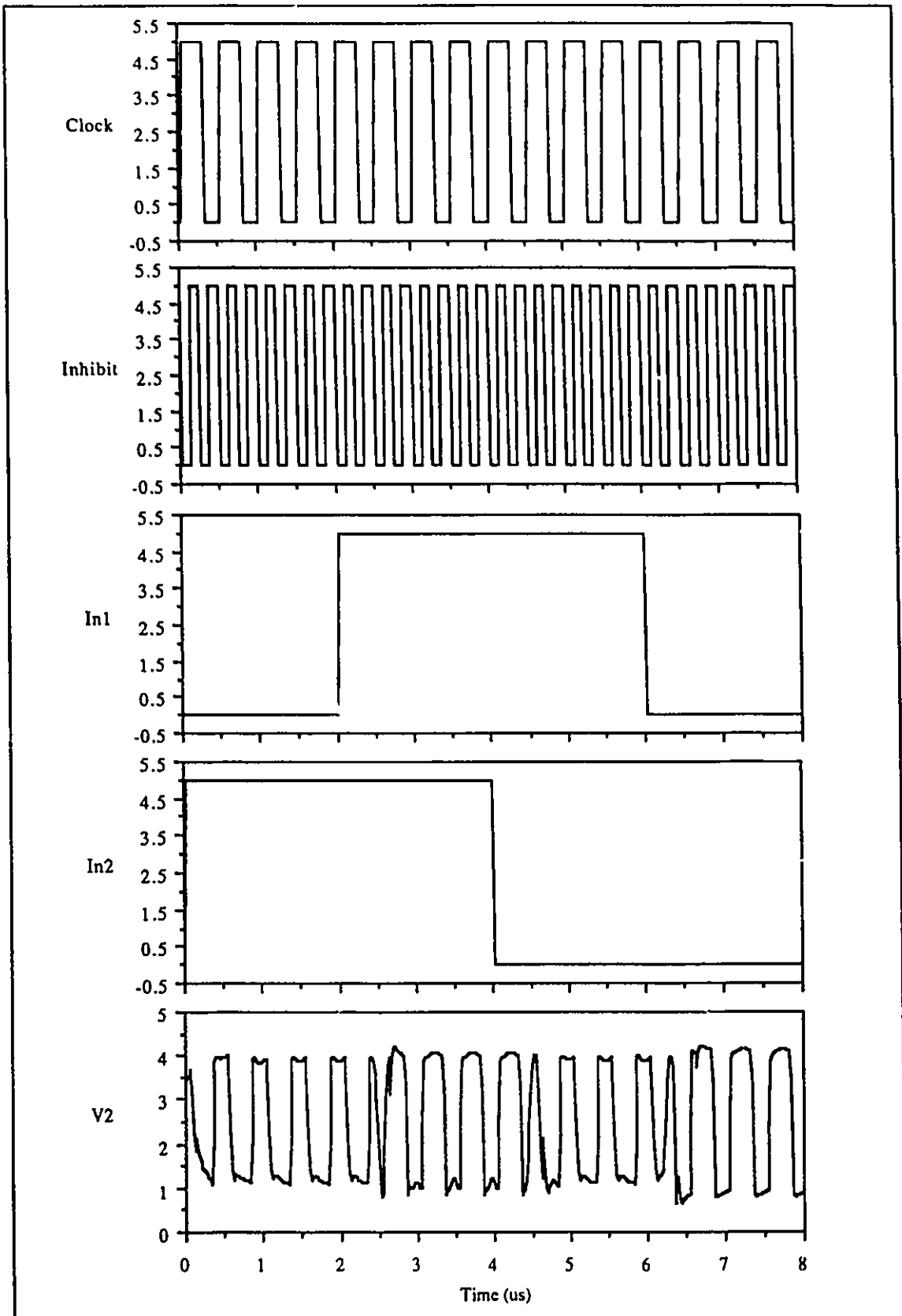


Figure 2.14: (Continued below)

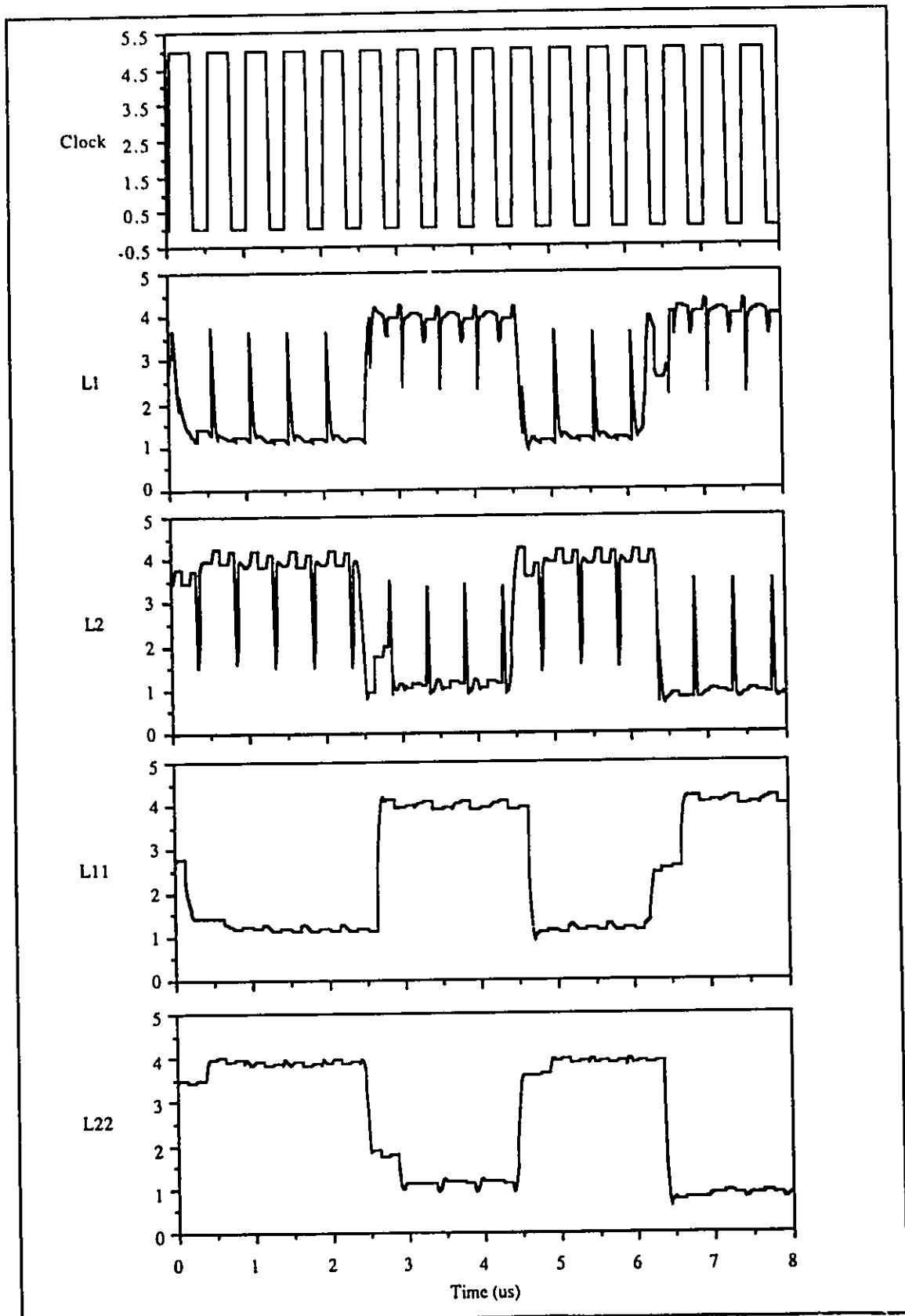


Figure 2.14: (Continued below)

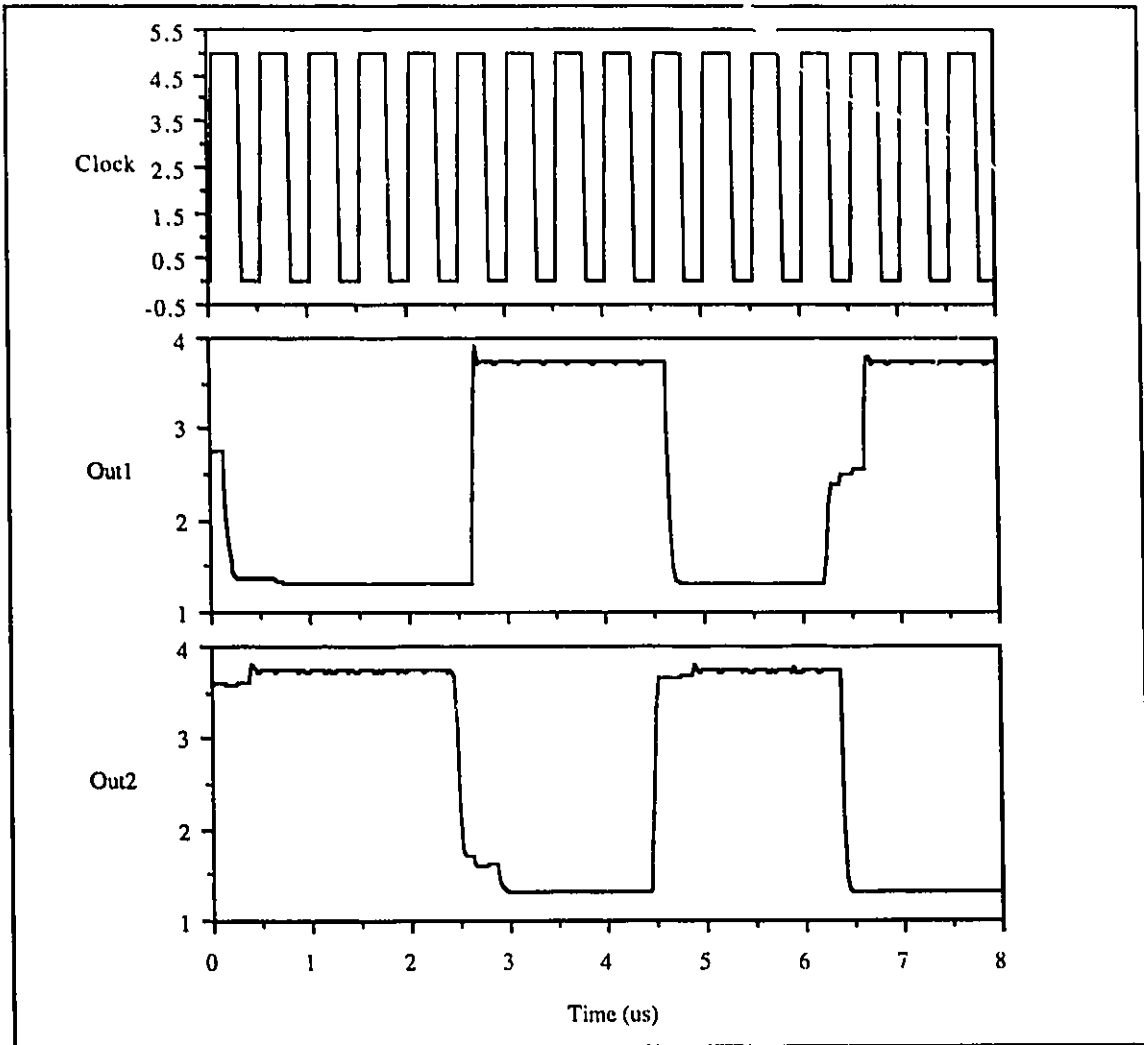


Figure 2.14: Waveforms of the modified architecture

The spikes are still seen in the signals L1 and L2, but L11 and L22, the inputs of the voltage mode neurons, are smooth. Notice that V2 is settled down well before the next transition of the clock. The speed of the circuit can therefore be easily increased to 8 mega refreshes per second.

The waveforms presented in this section are the result of full-scale transistor level SPICE simulations. They serve to show the validity of concepts and to insure that the circuit

blocks work well when put together. The simulation and test results of the individual blocks will be presented in later chapters.

## 2.7 Summary:

Analog CMOS, a promising technology for the implementation of neural networks, has been used as the basis of the architecture presented in this chapter. The main object in this design is to trade-off computation time for circuit complexity. The multiplexing scheme imbedded in the architecture allows the reduction of the number of synaptic multipliers and physical interconnections. This trade-off can be justified by the fact that biological neural networks are generally much slower than CMOS circuitry (point #1 in section 1.3), and yet have remarkable performance unparalleled by digital computers. It seems reasonable to trade the speed of CMOS for interconnection complexity, which is a major hindrance.

By making the architecture as regular as it is, and by reducing the wiring between stages, it becomes possible to have multi-chip networks. In this way, one of the constraints on the size of network is relaxed. These chips can all be similar and differ only in the ROM contents. A generic programmable chip containing one stage can be developed to facilitate the design of feed-forward networks and reduce the turn-around time in the development process.

The original architecture of figure 2.3 was composed with the assistance of ideas from biological neural networks. Some modifications were made afterwards to improve the overall performance and save silicon area. In all these versions, a relationship between the scales of the building blocks is necessary for the correct operation of the system. This relationship is summarized in the equation:

$$A \frac{K}{W} R = B \quad (2.7)$$

Examples with two versions of the introduced structure were presented in this chapter and the analysis of the waveforms demonstrated the necessity of the changes in the original version.

---

## Neuron and Synapse

---

### 3.1 Neuron:

In many neural net models, the transfer characteristic of the neuron is in the form of a sigmoidal ( S-like ) function. Beside being well behaved and continuously differentiable, it is a good approximation to the activation function of the biological neural cells. In feed-forward networks that rely on a variant of the steepest descent algorithm for learning, e.g. back error propagation, a specific type of sigmoid becomes very attractive:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (3.1)$$

This function, sometimes called the logistics function, has the advantage that its derivative can be expressed in terms of itself and its shifted version. This is useful, since the derivative is used in the update law and is not always easy to compute.

$$f'(x) = f(x) f(1-x) \quad (3.2)$$



The transfer characteristics of the VLSI neuron cells presented in this chapter are approximations to this function.

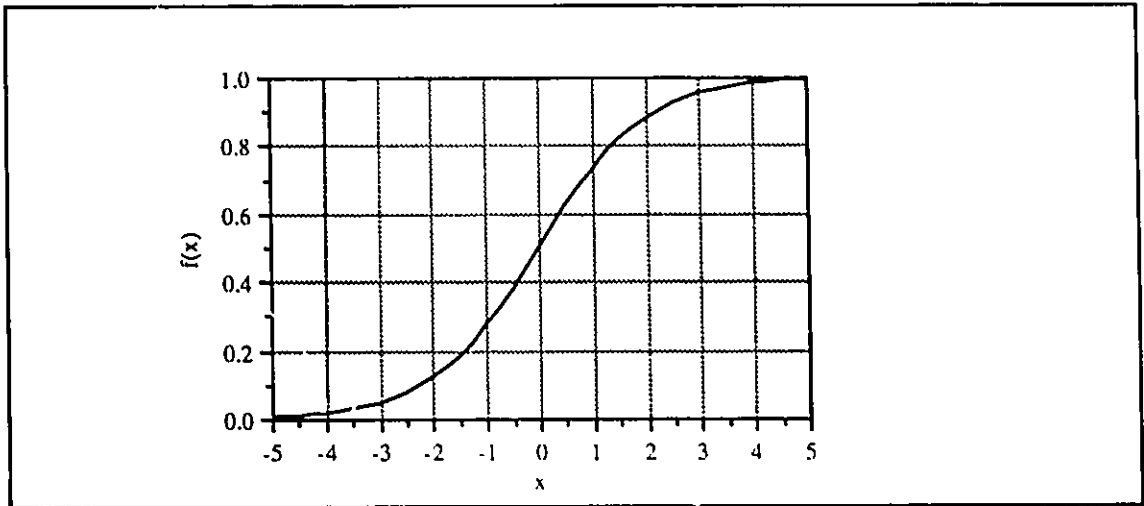


Figure 3.1: The sigmoidal logistics function

### 3.1.1 Current mode neurons:

The basic current mode neuron is shown in figure 3.2. This is a cell designed for the original architecture of figure 2.3. This cell has two inputs: activation and threshold. The S-like transfer characteristic is obtained through a MOS differential pair, as explained below. The node  $I_{out}$  will be coupled to the input of the synapse.

The large signal characteristics of this circuit can be obtained analytically using a simplified model for the differential pair. Assume  $M5$  to be an ideal current source, and  $M3$  and  $M4$  always to be in saturation. These assumptions are reasonable for the larger part of the operating region and illustrate the behavior even when the assumptions are not exactly valid. The simplified circuit is shown in figure 3.3.

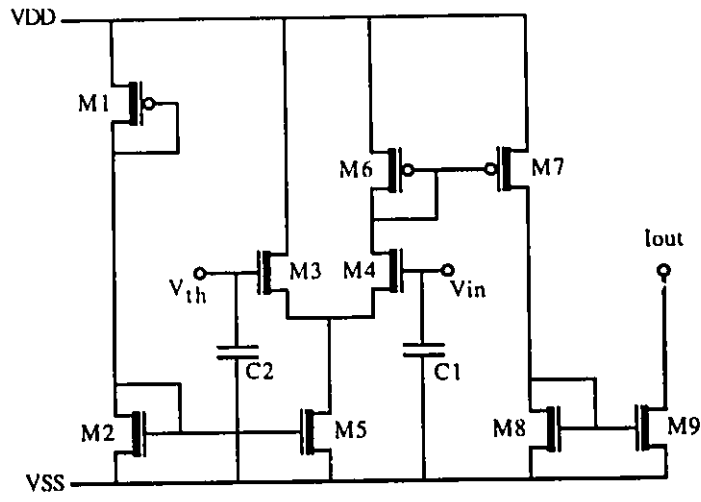


Figure 3.2: Current mode neuron with variable threshold

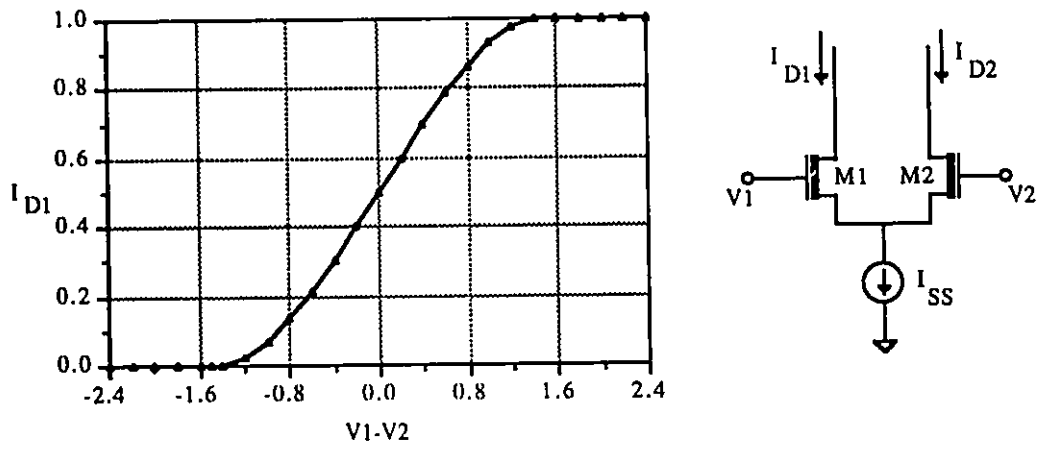


Figure 3.3: The differential pair and its transfer characteristics

The relationships describing the large signal behavior are [62]:

$$V_{ID} = V_1 - V_2 = \left( \frac{2I_{D1}}{\beta} \right)^{0.5} - \left( \frac{2I_{D2}}{\beta} \right)^{0.5} \quad (3.3)$$

and

$$I_{SS} = I_{D1} + I_{D2} \quad (3.4)$$

Where it has been assumed that M1 and M2 are matched. The solution for  $I_{D1}$  and  $I_{D2}$  is obtained by substituting (3.4) in (3.3) and forming a quadratic equation. There are four regions of operation. For  $V_{ID}^2 < -\frac{2I_{SS}}{\beta}$ :

$$I_{D1} = 0 \quad I_{D2} = I_{SS} \quad (3.5)$$

For  $-\frac{2I_{SS}}{\beta} < V_{ID}^2 < 0$ :

$$I_{D1} = \frac{I_{SS}}{2} - \frac{I_{SS}}{2} \left( \frac{\beta V_{ID}^2}{I_{SS}} - \frac{\beta^2 V_{ID}^4}{4I_{SS}^2} \right)^{0.5} \quad (3.6)$$

$$I_{D2} = \frac{I_{SS}}{2} + \frac{I_{SS}}{2} \left( \frac{\beta V_{ID}^2}{I_{SS}} - \frac{\beta^2 V_{ID}^4}{4I_{SS}^2} \right)^{0.5} \quad (3.7)$$

For  $0 < V_{ID}^2 < \frac{2I_{SS}}{\beta}$ :

$$I_{D1} = \frac{I_{SS}}{2} + \frac{I_{SS}}{2} \left( \frac{\beta V_{ID}^2}{I_{SS}} - \frac{\beta^2 V_{ID}^4}{4I_{SS}^2} \right)^{0.5} \quad (3.8)$$

$$I_{D2} = \frac{I_{SS}}{2} - \frac{I_{SS}}{2} \left( \frac{\beta V_{ID}^2}{I_{SS}} - \frac{\beta^2 V_{ID}^4}{4I_{SS}^2} \right)^{0.5} \quad (3.9)$$

And for  $\frac{2I_{SS}}{\beta} < V_{ID}^2$ :

$$I_{D1} = I_{SS} \quad I_{D2} = 0 \quad (3.10)$$

The transfer characteristic of the differential pair in the normalized form ( $I_{SS}=1$ ,  $\beta=1$ ) is given in figure 3.3. The shape of this curve is a good approximation to the sigmoid function of figure 3.1. Simulation of the neuron cell with SPICE results in the characteristic given in figure 3.4. Test results of the same cell fabricated with the Northern Telecom™ 3 $\mu$ m double-metal single-polysilicon technology are presented in figure 3.5.

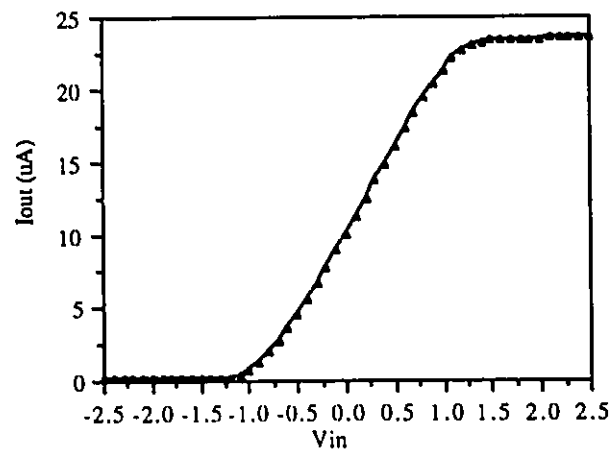


Figure 3.4: Simulation results of the neural cell with a threshold of zero

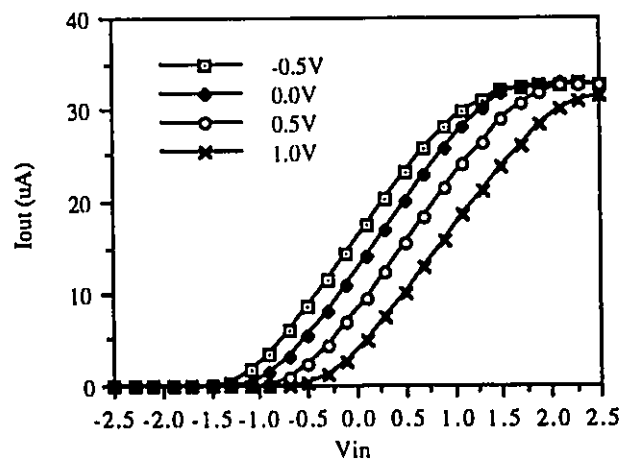


Figure 3.5: Test results of the neural cell with various threshold values

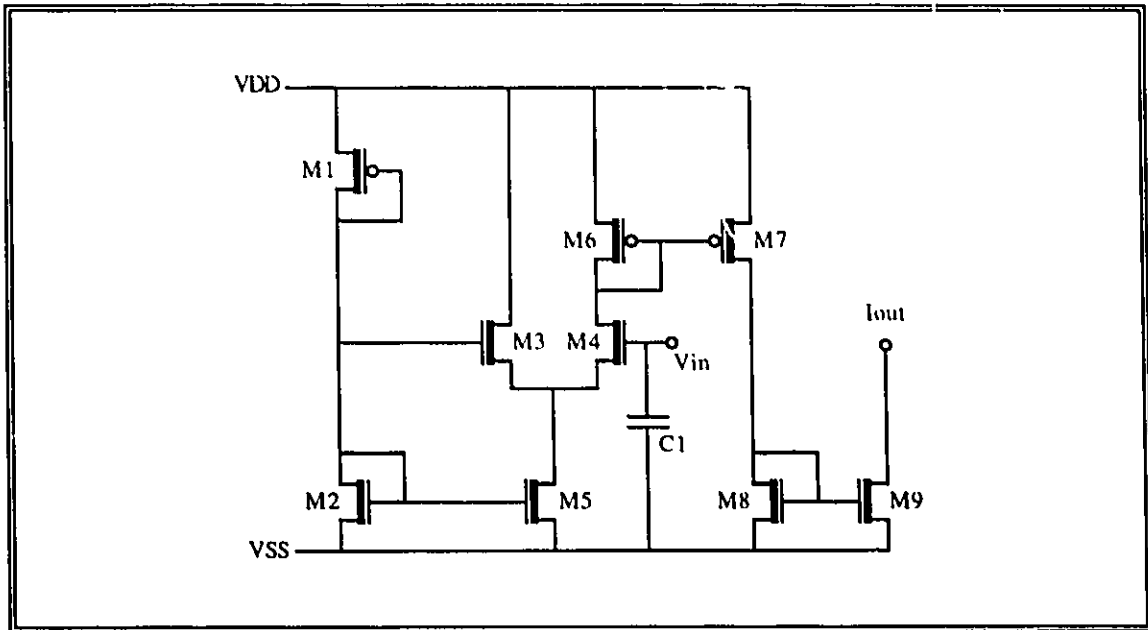


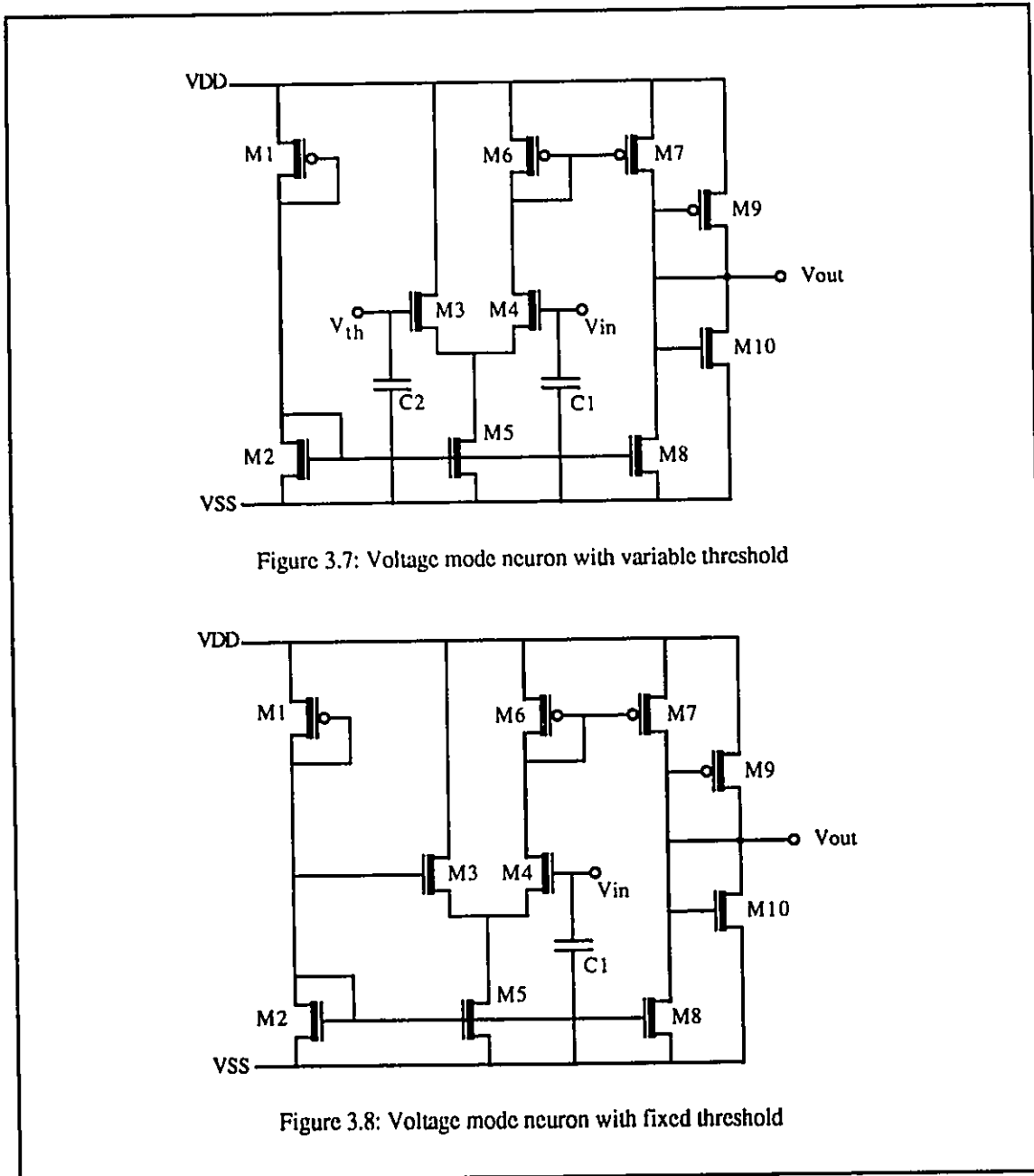
Figure 3.6: Current mode neural cell with fixed threshold

For the modified form of the architecture as shown in figure 2.5 and 2.6, a neural cell with only an activation input and no threshold input is needed. This cell is given in figure 3.6. The transistors M1 and M2 have W/L ratios so that the drain of these two transistors is at 0V. This cell is the same as that of figure 3.2 with the difference that the gate of M3 is connected to the gates of M1, M2 and M5, and the capacitor C2 is eliminated.

The capacitors in both of the cells serve to reduce the voltage drop of the input nodes of these cells in the time between two refresh cycles. Charge leakage in capacitive nodes has two main components: through capacitive oxide and through the reverse biased junction of the transistor(s) charging the node. The first component is proportional to the oxide area and cannot be reduced by increasing the capacitance. The second mode depends only on the voltage of the node and the width of the charging transistor. Since the second mode is dominant in normal temperatures, an increase in the capacitance of the node will result in a lower voltage drop between refresh cycles.

### 3.1.2: Voltage mode neurons:

Voltage mode neurons (figures 3.7 and 3.8) are designed by subtracting a constant current from the drain current of M4 and feeding the result to a resistive MOS pair (M9 and M10). The transfer characteristics of this neuron is given in figure 3.9.



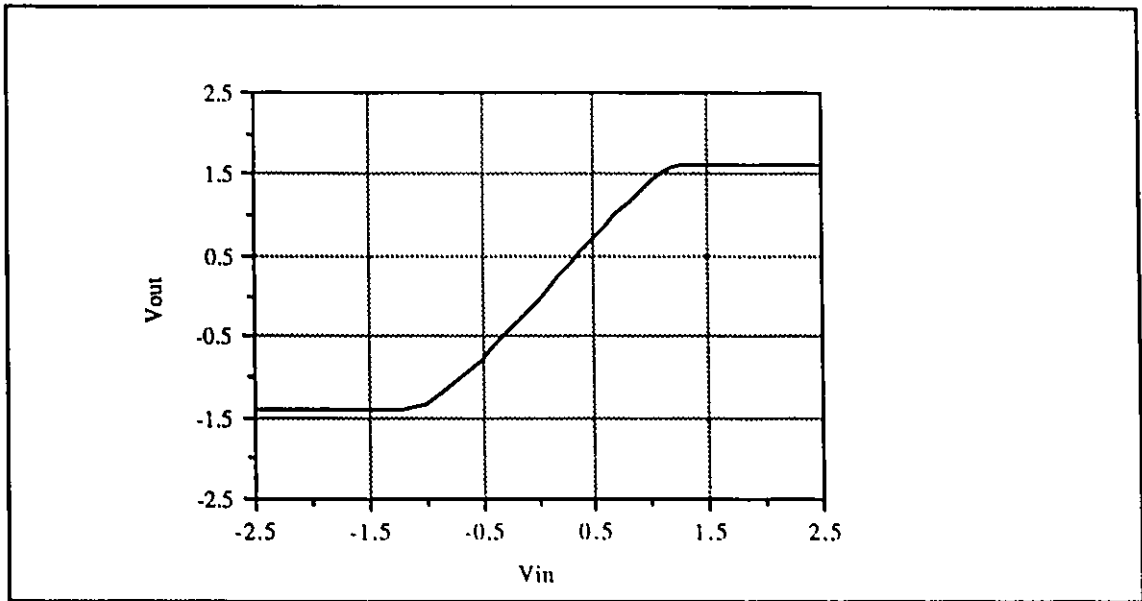


Figure 3.9: Transfer Characteristics of the voltage mode neuron (Simulation)

Current Mode Neurons ( $\mu\text{m}$ )		Voltage Mode Neurons ( $\mu\text{m}$ )	
M1	5.4/5.4	M1	5.4/5.4
M2	5.4/9.6	M2	5.4/9.6
M3-M4	5.4/9.6	M3-M4	5.4/9.6
M5	5.4/10.2	M5	5.4/10.2
M6-M7	6/3	M6	6/3
M8-M9	6/3	M7	18.6/3
Capacitors		M8	3/4.2
C1	0.433 pF	M9	6/9
C2	0.519 pF	M10	3/15

Table 3.1: Transistor geometries of the neural cells

Note that in the present form, the voltage mode neuron is only suitable for a final stage in the network. To be used in the intermediate stages, as proposed in section 2.5 and shown in figure 2.8, the cell has to be modified to have a minimum output activation of zero. Removal of transistor M8 would result in a minimum output of zero, since the M9-M10 pair is matched. The transistor geometries for the 4 neural cells presented so far are given in table 3.1.

## **3.2 Synapse:**

By definition, the synapse is a junction between two neurons. Almost all of the information in a neural system is stored in the synapses. Some methods of designing a synapse in analog CMOS were described in section 1.5. Below we present a hybrid realization with current mode multiplying D/A converters and digital ROM for storage of pre-trained connection weights.

### **3.2.1 Multiplier:**

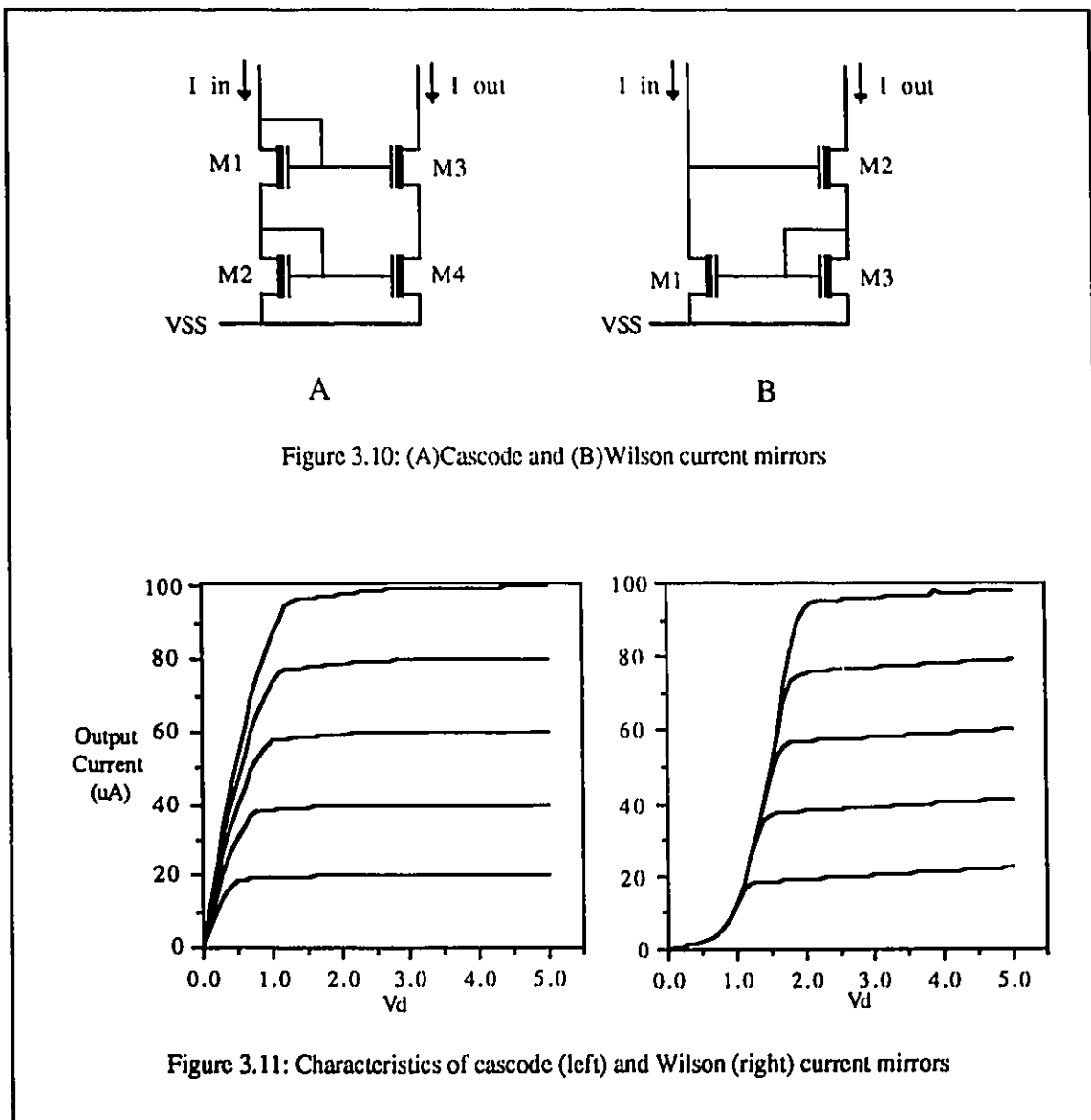
Current mode digital to analog converters are constructed with a series of current mirrors, each dividing the current in half, and a series of switches that forward these currents to an addition node. The main criteria in the choice of the current mirrors are:

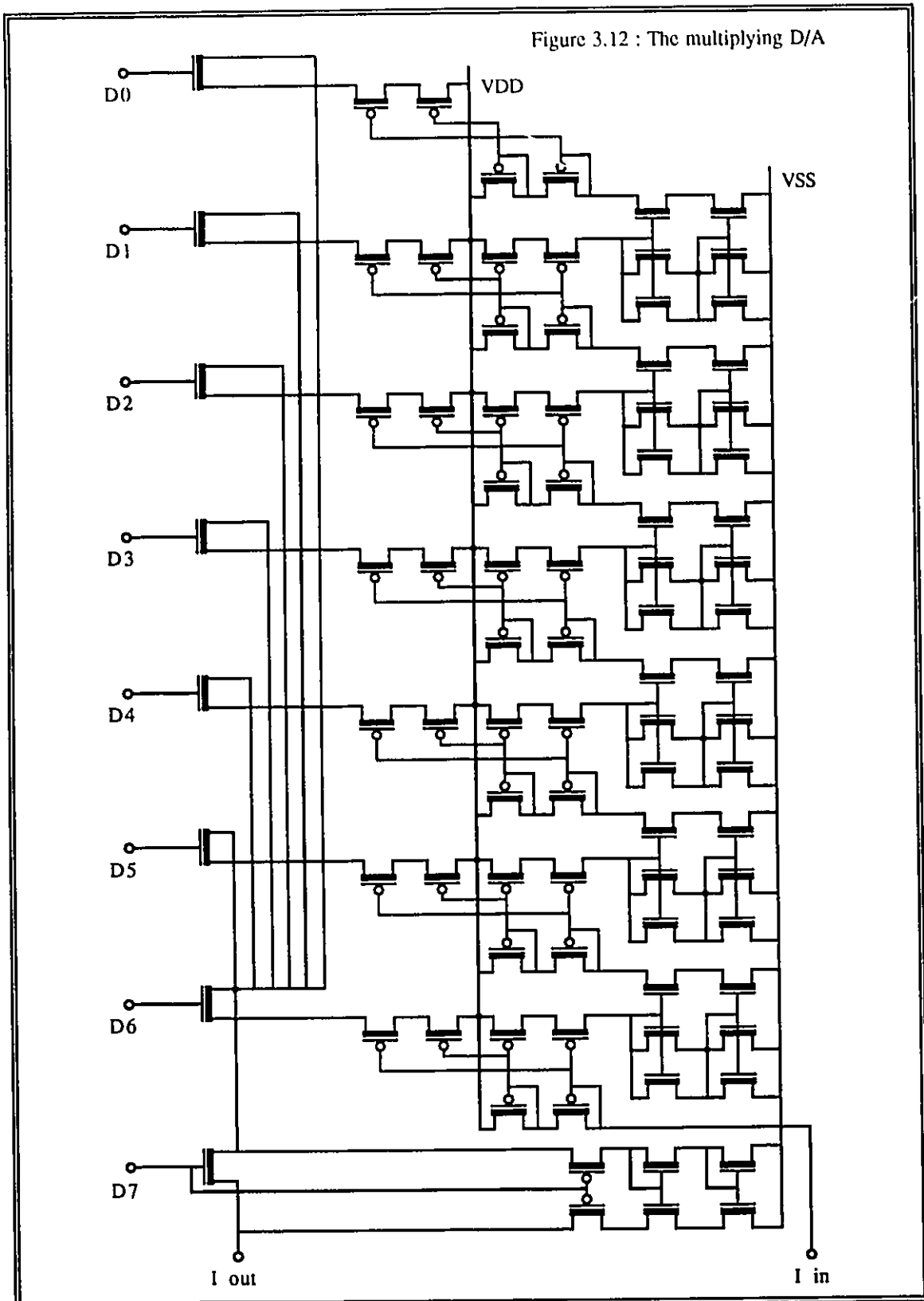
- a) High small signal output resistance
- b) Low minimum operating voltage for wider operational range

A simple current mirror, because of channel length modulation effects, does not have enough output resistance. The best choices are Wilson and cascode mirrors (figure 3.10).



Both of these circuits utilize feedback to compensate the channel length modulation effects and hence increase the output resistance. Although Allen and Holberg [62] assert that the Wilson current mirror has a slightly higher output resistance, this was not observed in our SPICE simulations (figure 3.11). However, it is seen both in [62] and figure 3.11 that the cascode current mirror has a more graceful degradation of linear transfer characteristics as  $V_D$  decreases. The cascode mirror was chosen for the implementation of the D/A. The schematic diagram of the D/A is shown in figure 3.12.





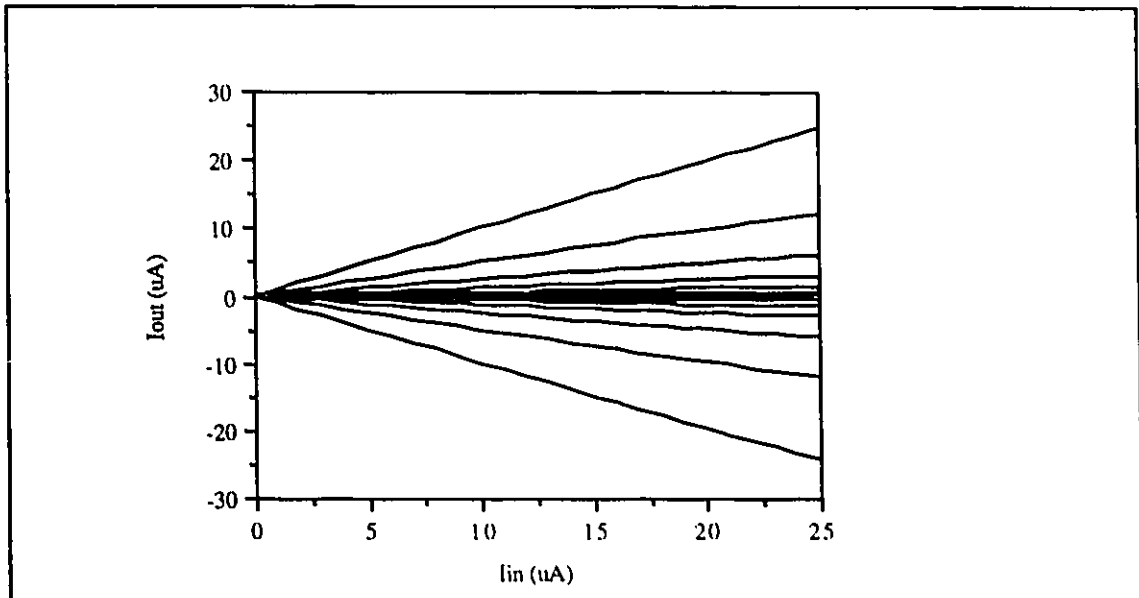


Figure 3.13: Output of the D/A vs. input current as the input bits are set one by one (simulation)

### 3.2.2 ROM:

The ROM structure, a section of which is shown in figure 3.14, consists of p-channel pull-up transistors and n-channel transistors as memory elements. To avoid large pull-up transistors and save silicon area, the gates of the pull-up transistors are held at a voltage provided by the M25-M26 divider. Although it is possible to obtain greater speed with dynamic circuits, the added complexity of dynamic memories is not justified, since the delay introduced into the network by the analog blocks is many times higher than that of the ROM.

The waveforms of figure 3.15 are test results of the ROM cells. The delay is 33ns for the falling edge and 18ns for the rising edge of the clock pulse. However, this is only an upper bound on the actual delays of the cell, since it includes the delay introduced by the output pad driver on the chip. Simulations confirm that the cell itself is faster, another indication that a static ROM suffices for the proposed architecture.

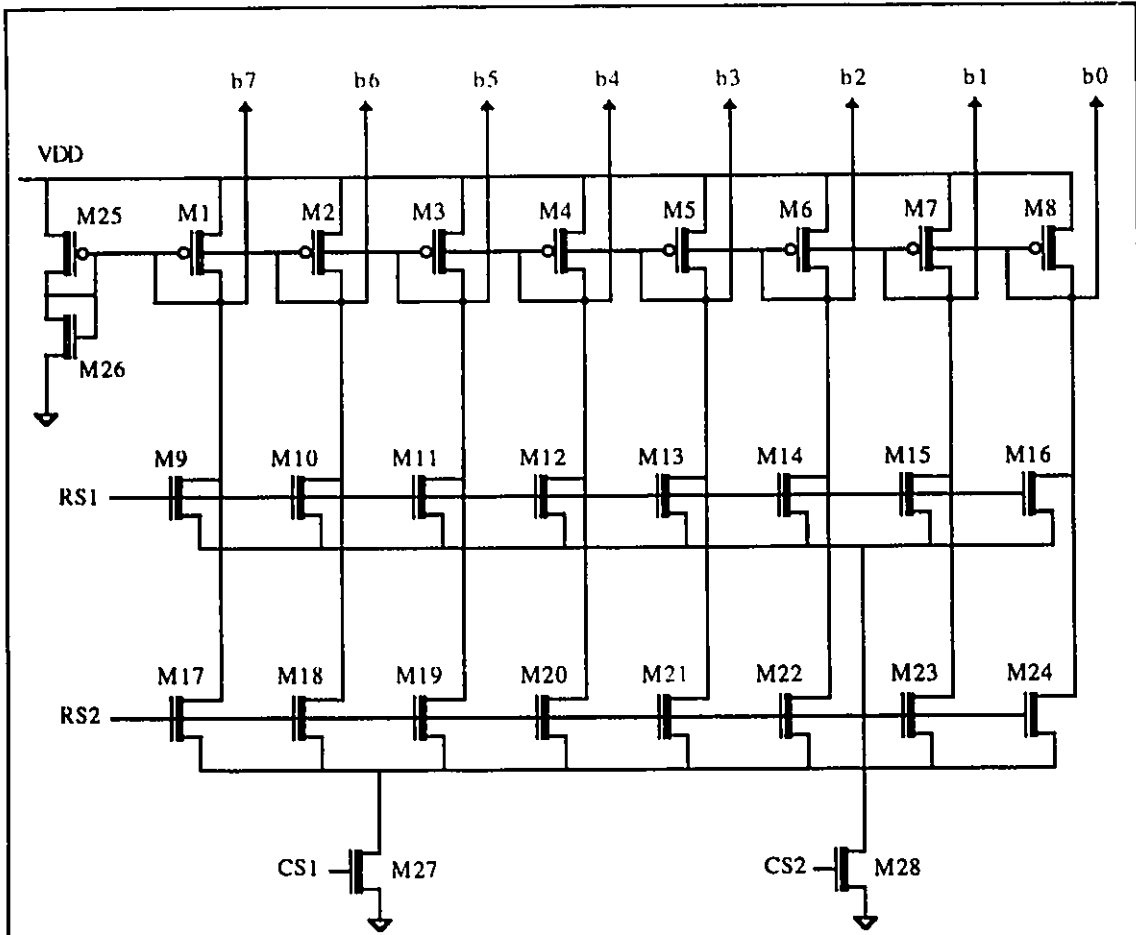


Figure 3.14: Schematic diagram of a section of the ROM

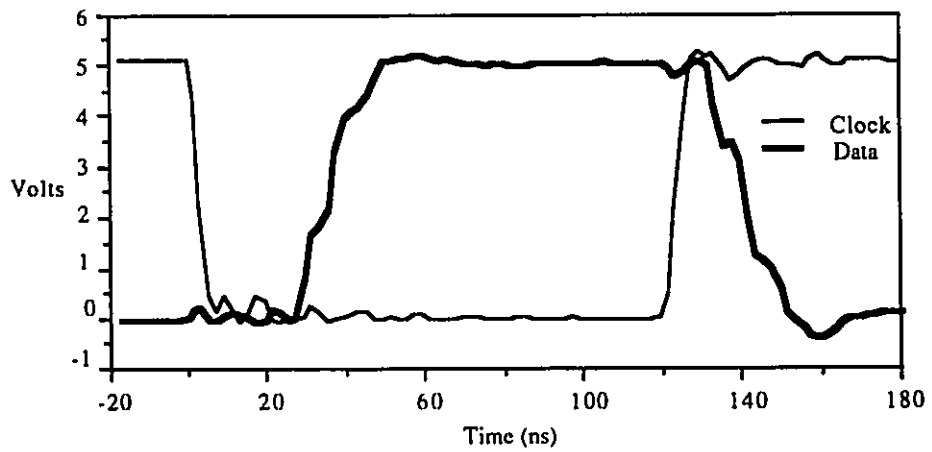


Figure 3.15: Data access delay in the ROM (test result)

The ROM of figure 3.14 is a 2-dimensional structure, with row select and column select lines. It should be noted that, in the simpler cases where only a 1-dimensional structure is needed, the row selects should be used for selection and *not* the column selects. The effect of activating more than one row select and using the column select for addressing is shown in figure 3.16. An arbitrary data pattern<sup>1</sup> is stored in the two locations. Suppose that CS1 is selected. The output bit pattern should be 10110001. But the sources of the upper row of cells are also grounded via the highlighted path, and an erroneous output will result.

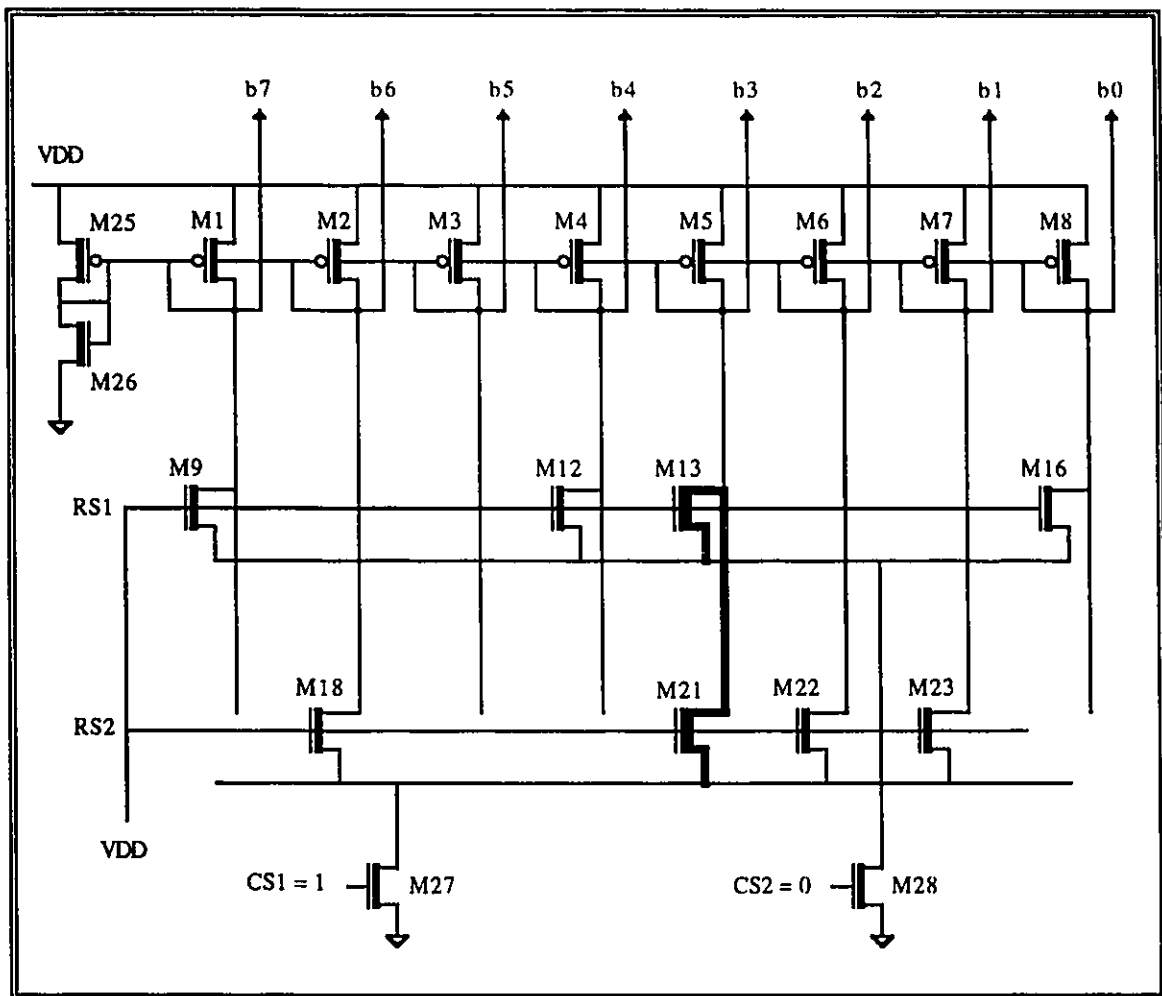


Figure 3.16: A possible source of error in the ROM circuit

<sup>1</sup> For the error to appear, there should be at least two data words with zero bits at the same location.

### 3.3 Summary:

For various reasons, neural models usually use a sigmoidal characteristic for the neuron. The differential MOS pair has a characteristic that resembles the sigmoid, and has been used here to develop current and voltage mode neurons. These cells have been fabricated and the tests comply with theoretical and simulation results.

The synapse is constructed from a multiplying D/A converter and a ROM. The D/A is designed with cascode mirrors and the ROM has a static structure. In the addressing of the ROM, no two row selects should ever be active at the same time. If it is desired to have a 1-dimensional structure, the column selects should be connected to  $V_{DD}$  and addressing be performed by the row select lines.

---

## Support Circuitry

---

### 4.1 Trans-impedance amplifier:

This building block is used whenever there is a need to convert current mode signals to voltage mode. The design (figure 4.1) consists of a differential amplifier with a resistive feedback element; transistors M12 and M13. The feedback configuration is voltage-parallel, which results in a predictable  $I$ - $V$  characteristic and low input and output impedance. A low input impedance decreases the variations of input voltage and therefore helps the current mode circuit of the previous stage to remain in the normal range of operation and avoid saturation effects. Figure 4.2 shows the simulation and actual transfer characteristics of the trans-impedance amplifier.

As in all feedback systems, the stability problem has to be addressed. Originally, an attempt was made to stabilize the circuit with the load capacitance. But even with a load of

1pF, settling time was in excess of 300ns. In the present configuration, with a compensation capacitor of 0.1pF across the feedback resistor, settling times of around 30ns have been achieved (figure 4.3).

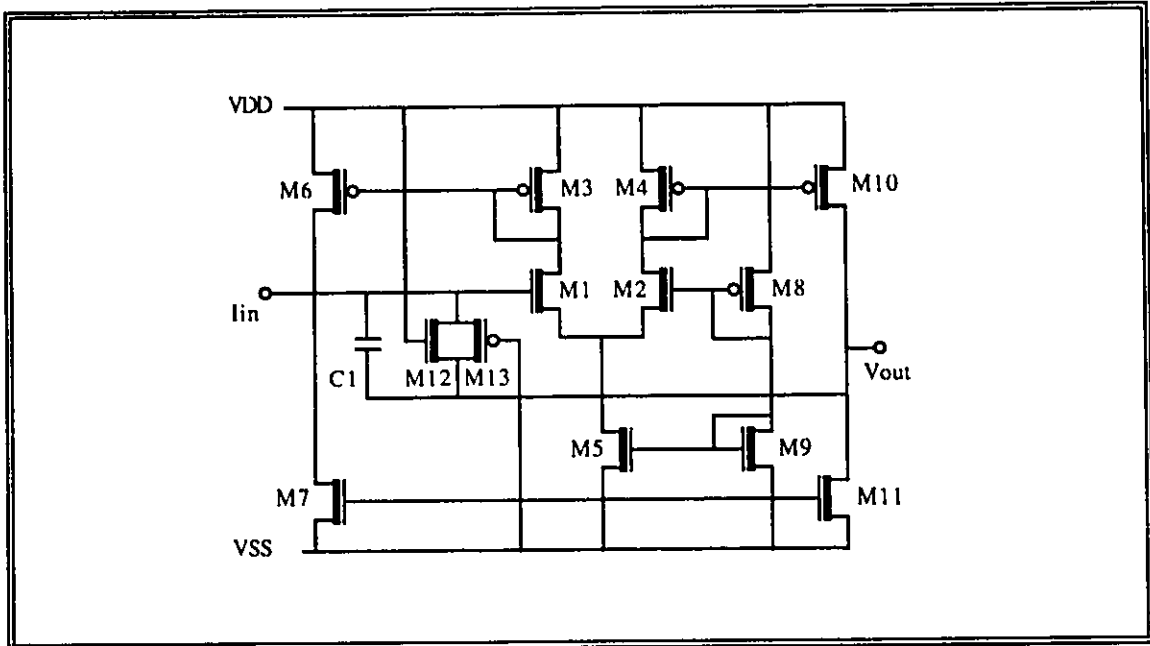


Figure 4.1: Trans-impedance amplifier

Device W/L Dimensions ( $\mu\text{m}$ )			
M1-M2	30/3	M8	5.4/3.6
M3-M4	9/3	M9	5.4/9.6
M5	5.4/3	M10	18/3
M6	18/3	M12	3/23.4
M7,M11	6/3	M13	3/7.8

Table 4.1: Device geometries for the trans-impedance amplifier



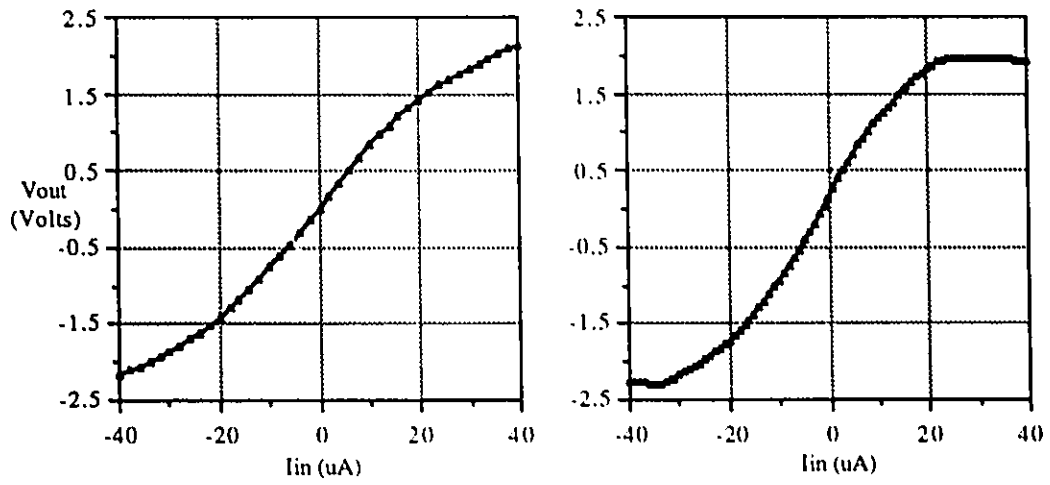


Figure 4.2: Simulated (left) and actual (right) transfer characteristics

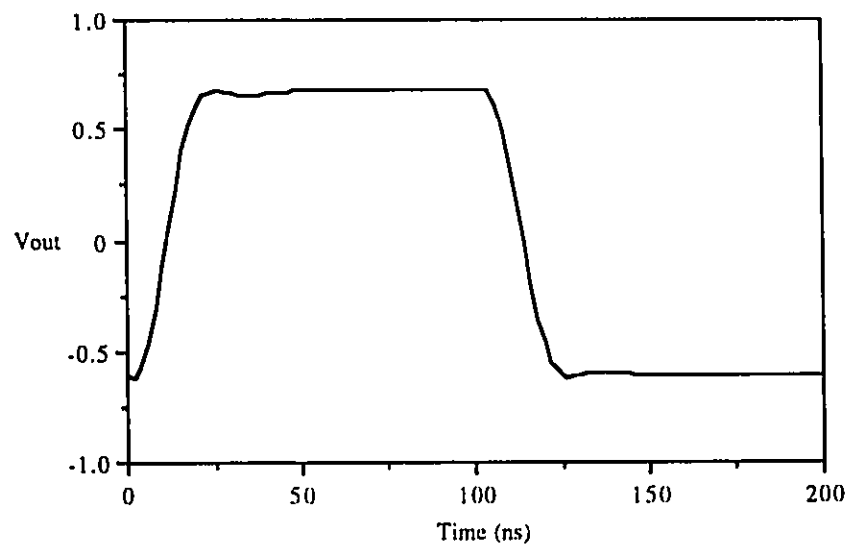


Figure 4.3: Step response of the trans-impedance amplifier with a load of 0.5pF

## 4.2 Voltage to current converter [57] :

Unlike the trans-impedance amplifier introduced in the previous section, this circuit does not have any feedback. Its operation is based on the square law characteristics of the MOS transistors. The currents are added and subtracted in a way that the quadratic terms cancel out and the output current is a linear function of the input voltage, as shown in the following analysis. This analysis is based on the assumption that all devices are similar and in saturation.

Considering that the drain current of M1 (M2) is the same as M3 (M4), ignoring the body effect and assuming identical transistors:

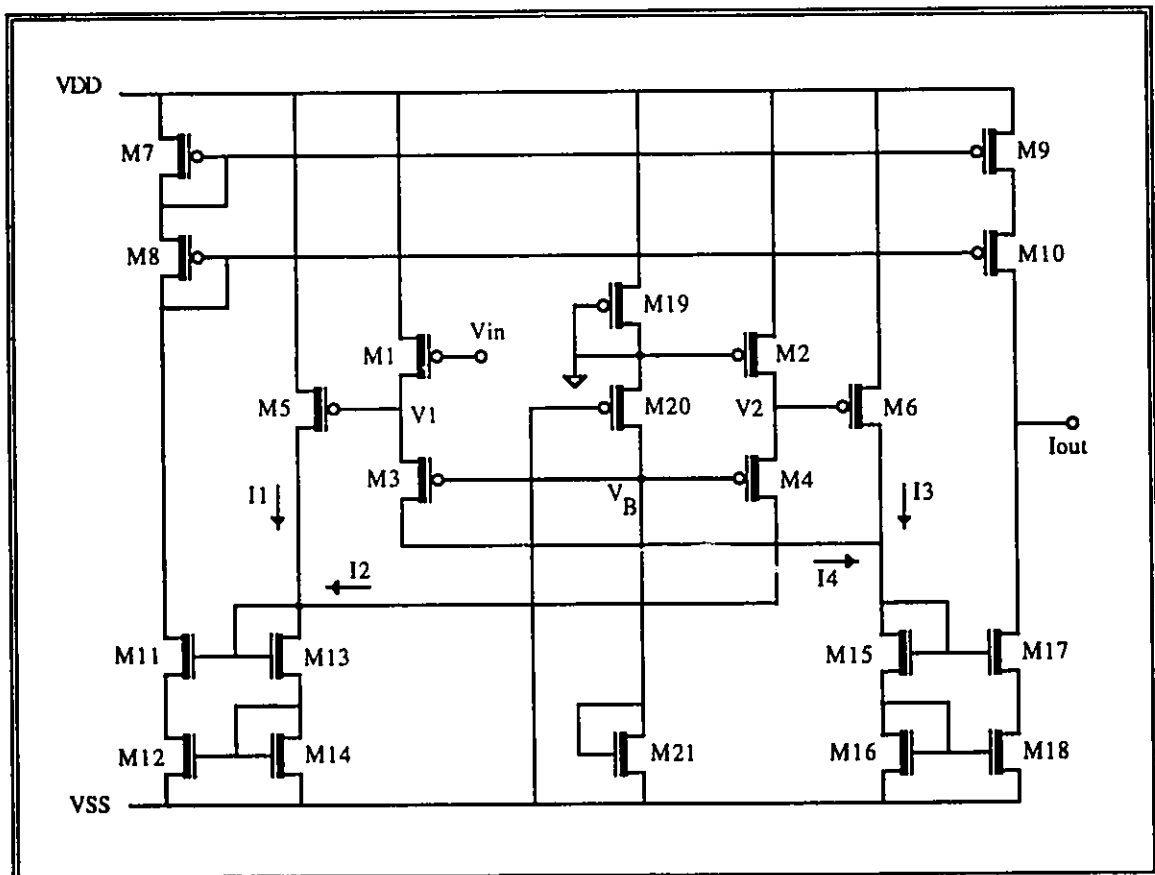


Figure 4.4: Voltage to Current Converter

$$V_1 - V_B = V_{DD} - V_{in} \quad (4.1)$$

$$V_2 - V_B = V_{DD} - 0 \quad (4.2)$$

$$I_1 = \beta (V_{DD} - V_1 - V_{th})^2 \quad (4.3)$$

$$I_2 = \beta (V_2 - V_B - V_{th})^2 = \beta (V_{DD} - V_{th})^2 \quad (4.4)$$

$$I_3 = \beta (V_{DD} - V_2 - V_{th})^2 \quad (4.5)$$

$$I_4 = \beta (V_1 - V_B - V_{th})^2 = \beta (V_{DD} - V_{in} - V_{th})^2 \quad (4.6)$$

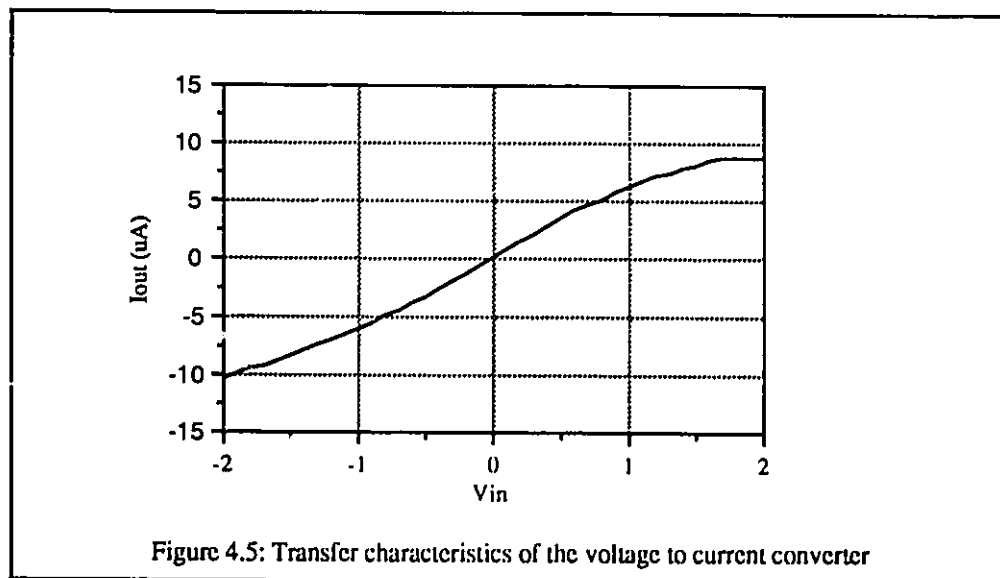
$$I_1 + I_2 = \beta (V_1^2 + 2V_{DD}^2 + 2V_{th}^2 - 2V_{DD}V_1 + 2V_1V_{th} - 4V_{DD}V_{th}) \quad (4.7)$$

$$I_3 + I_4 = \beta (V_2^2 + V_{in}^2 + 2V_{DD}^2 + 2V_{th}^2 - 2V_{DD}V_2 + 2V_2V_{th} - 4V_{DD}V_{th} - 2V_{DD}V_{in} + 2V_{in}V_{th}) \quad (4.8)$$

Substituting (4.1) and (4.2) in (4.7) and (4.8) and subtracting:

$$I_{out} = (I_1 + I_2) - (I_3 + I_4) = 2\beta V_{in} (V_{DD} - V_B - 2V_{th}) \quad (4.9)$$

In the actual case, the multiplicative constant will be different because the transistors are not all the same size.



Device W/L Dimensions ( $\mu\text{m}$ )			
M1-M2	4.2/12	M11-M18	12/3
M3-M4	7.2/3	M19	6/16.2
M5-M6	6.6/9	M20	4.8/5.4
M7-M10	36/3	M21	3.6/15

Table 4.2: Device geometries for the voltage to current converter

### 4.3 Address generation:

In the proposed architecture (figures 2.3, 2.5 and 2.6), address generation is performed in a systematic manner with a counter and standard decoders for the ROM and demultiplexers. However, this may not be the most efficient way. The goal is to select the ROM locations and the corresponding demultiplexer channels one by one. Since the ROM locations and demultiplexer channels have a one-to-one correspondence, separate decoders are not necessary. Moreover, the decoders and the counter can be removed altogether and replaced with a ring counter, which will provide the line selection signals directly. If the ROM employs a 2-dimensional selection scheme, two independent ring counters will be used and the selection signals of the demultiplexer can be obtained through a set of AND gates. A suggested circuit for the ring counter is shown in figure 4.6

Taking  $\Phi 1$  and Hold to  $V_{DD}$  and  $\Phi 2$  to  $V_{SS}$  will isolate every two inverter into a flip flop. If these three signals are kept at the said levels while powering up, and the n-channel transistors of the inverter pairs are mismatched properly, the counter can be initialized so that only one of the output bits is at level 1.

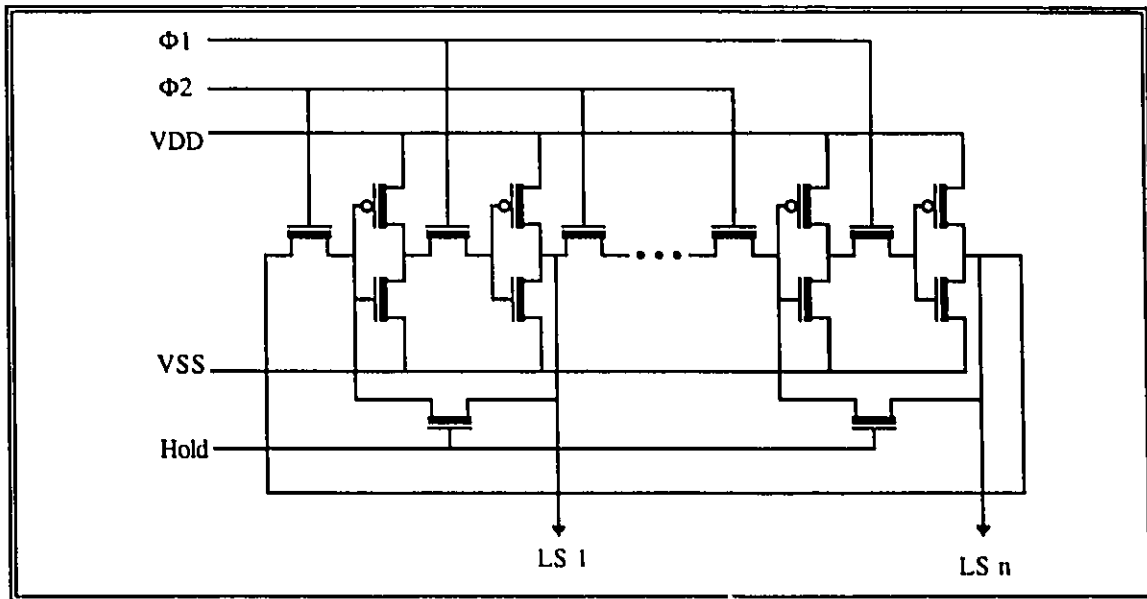


Figure 4.6: A suggested circuit for the ring counter

#### 4.4 Analog buffers and output pad drivers:

CMOS OpAmps with low drive capability are used extensively in applications such as switched capacitor filters and the procedures and trade-offs in the design of these amplifiers have been more or less clear for some time [63]. There are numerous design examples in this category reported in the literature. However, with increasing circuit density and growing interest in very large scale analog and mixed mode circuits, high drive analog buffers with efficient die area utilization and wide dynamic range will be needed as well.

A high-drive amplifier is typically required to handle capacitive loads in excess of 1000pF and resistive loads of 1k $\Omega$  or less with acceptable slew rate, phase margin and total harmonic distortion. Although designs of power OpAmps and buffers have been reported in the literature, the overall performance of most of them suffer from individual shortcomings. Low capacitive load-handling capability [66, 67], high quiescent current

[68, 72], low dynamic range [64, 69-71], stability problems [70], and most importantly, large circuits with high die area consumption [65, 67-68, 71-72] are some of the more common weaknesses that make most of these designs unsuitable for a general purpose high power output pad driver.

In our survey, using circuit simulation, we found the circuit of Wong and Salama [70] the most appropriate for our application. Dynamic range, resistive load drive, and stability of the circuit (both verified and reported in [72]), however, were still outside the acceptable range required for our purposes. In this section, a series of buffers with improved overall performance will be presented. These buffers were designed as standard cells to be used in large scale analog circuits. They were implemented in a 3 $\mu$ m p-well CMOS process.

#### **4.4.1 A buffer with high capacitive drive [74-75]:**

The schematic of the circuit is given in figure 4.7. The transistors M1-M5 and M9-M13 form two complementary differential stages. The inputs of these stages are connected in parallel. Each of them drives one-half of the output common source push-pull stage. By using two input stages in parallel, the intermediate level-shifting stage, normally needed to feed a push-pull output stage, is eliminated. The drawback of this approach is that it slightly increases the harmonic distortion, since the input stages cannot be exactly matched. This increase in distortion is tolerated for our applications. The alternative is a larger circuit, possibly including compensation capacitors, which will significantly increase buffer area. Transistors M14 and M15 are connected as resistors. Their function is to decrease the impedance at the gates of output transistors, M16 and M17, and compensate the frequency response.

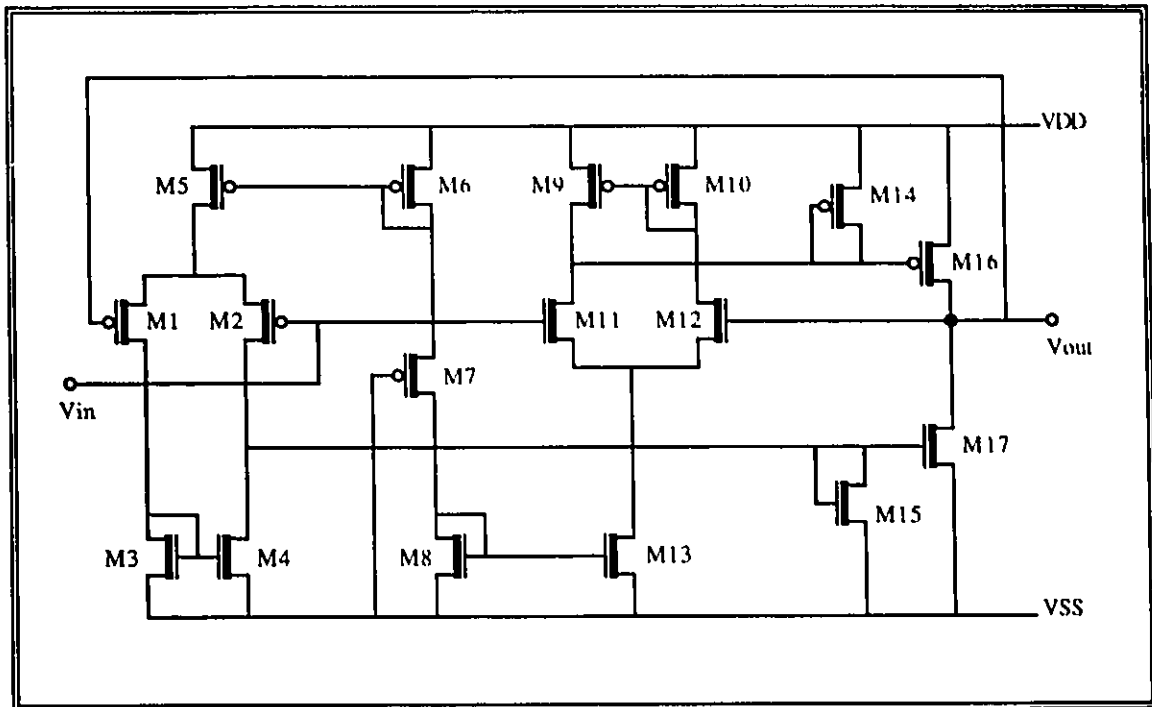


Figure 4.7: Schematic diagram of the buffer with high capacitive drive

Device W/L Dimensions ( $\mu\text{m}$ )			
M1	142.8/3	M11	48/3
M2	139.2/3	M12	50.4/3
M3, M4	5.4/3	M13	21/3
M5	27/3	M14	5.4/9,6
M6	31.2/3	M15	5.4/27
M7	3/9	M16	504/3
M8	12/3	M17	252/3
M9, M10	11.4/3		

Table 4.5: Device geometries

The pulse response of the buffer with a 5000pF capacitive load is shown in figure 4.8A. Slew rates of 1.2v/ $\mu$ s and 0.5% settling time of 3 $\mu$ sec have been achieved with this load. Figure 4.9 shows the transfer characteristic of the buffer. Output voltage span is 94% of the supply range with a 5k $\Omega$  load and rail-to-rail for  $R_L > 15k\Omega$ . Although the input circuitry is designed in two half-parts, total harmonic distortion is less than 3% at 20kHz. Normalized harmonics of the input at this frequency for a 2v(p-p) signal are given in Figure 4.10. The quiescent current is only 125  $\mu$ A.

The frequency response of the buffer is shown in figure 4.11. Dominant poles and zeros of the circuit are given in Table 4.4 for load conditions of:

$$R_L = 10k\Omega, C_L = 500pF$$

$$R_L = 10k\Omega, C_L = 5000pF$$

In both cases, the modulus of the closest pole (zero) to those in the Table are at least 5 (3) octaves higher. With the 5000pF load, the complex conjugate poles have a quality factor of  $Q=1.75$ , and with  $C_L=500pF$ ,  $Q=7.38$ . The circuit operates acceptably over an order of magnitude variation of the capacitive load.

Load	Poles (rad/sec)		Zeros (rad/sec)	
	Real	Imaginary	Real	Imaginary
$C_L = 500pF$ $R_L = 10K\Omega$	$-1.344 \times 10^6$	$1.991 \times 10^7$	$-3.369 \times 10^6$	0
	$-1.344 \times 10^6$	$-1.991 \times 10^7$	$-1.467 \times 10^8$	0
	$-3.369 \times 10^6$	0	$-1.758 \times 10^8$	0
$C_L = 5000pF$ $R_L = 10K\Omega$	$-1.675 \times 10^6$	$6.123 \times 10^6$	$-3.369 \times 10^6$	0
	$-1.675 \times 10^6$	$-6.123 \times 10^6$	$-1.467 \times 10^8$	0
	$-3.370 \times 10^6$	0	$-1.758 \times 10^8$	0

Table 4.4: Worst case pole-zero configurations at  $V_{in}=4V$  (simulation)



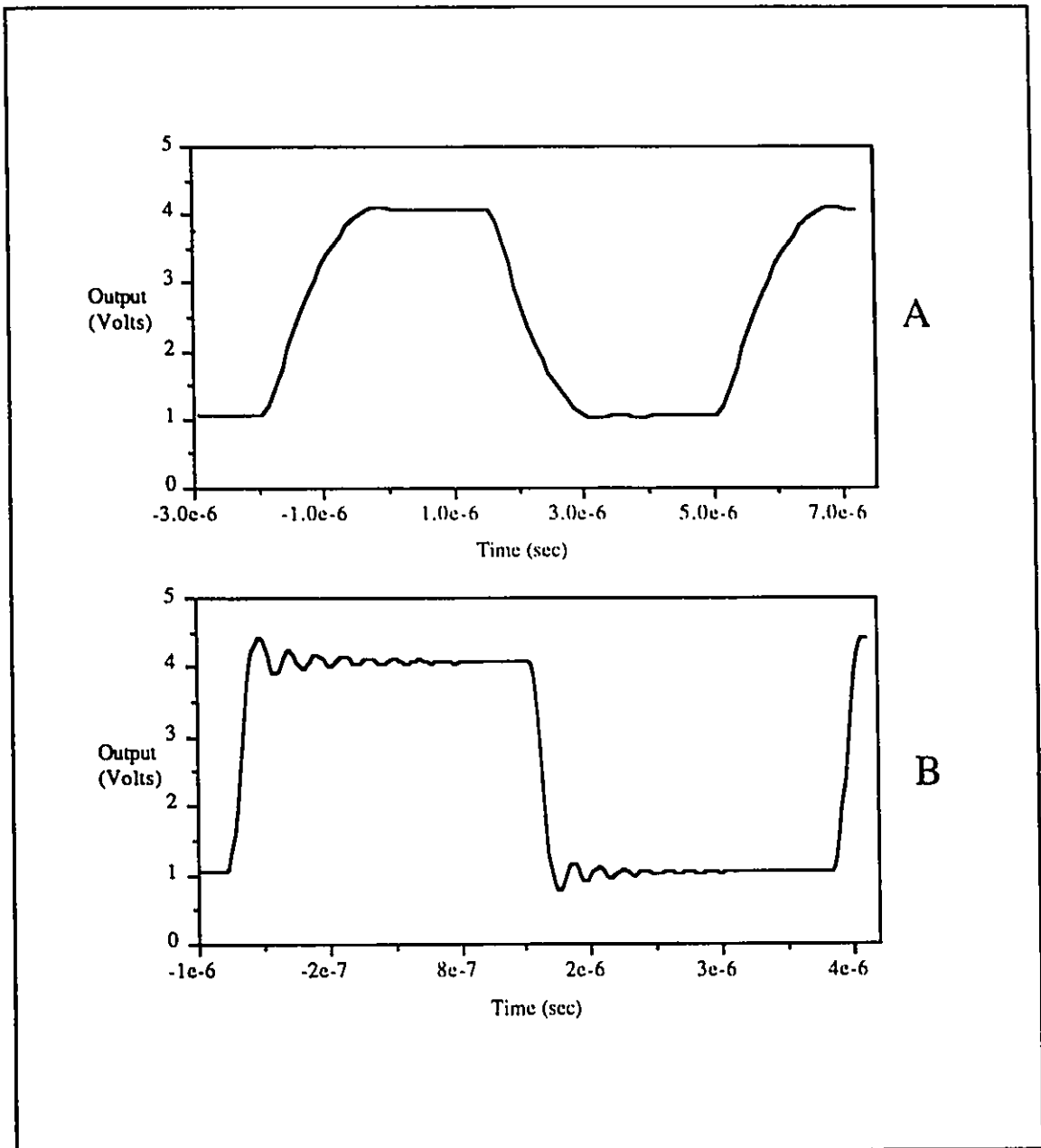


Figure 4.8: Test results: A) step response with  $C_L = 5000\text{pF}$

B) stability test with  $C_L = 200\text{pF}$

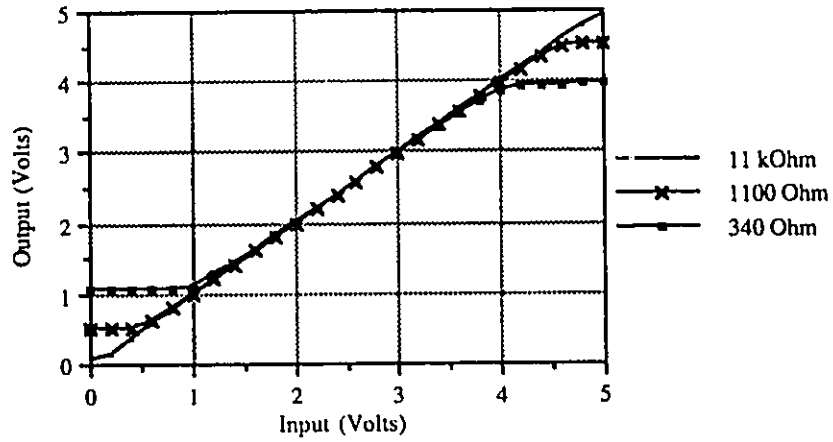


Figure 4.9: Transfer characteristics with various loads (test results)

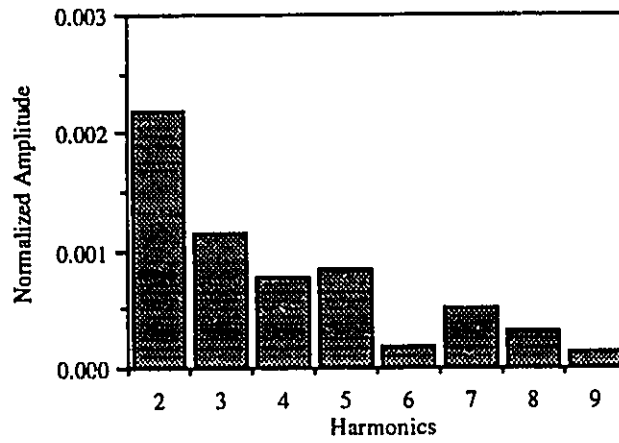


Figure 4.10: Normalized harmonics of the output at 20kHz with  $V_{in}= 2V(p-p)$  (simulation)

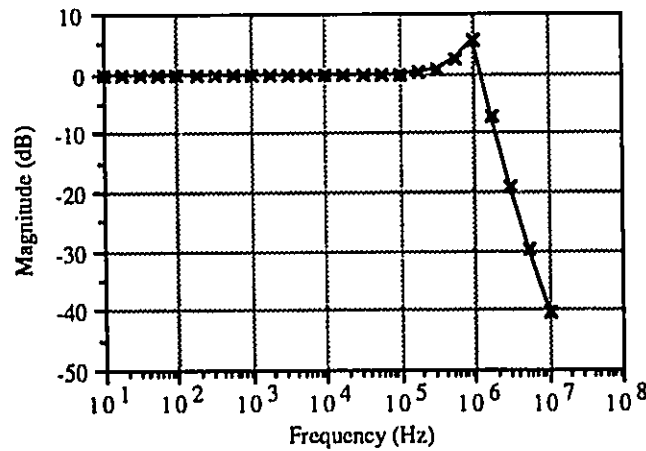


Figure 4.11: Frequency response of the buffer with  $C_L=5000pF$   $R_L=10k\Omega$  (simulation)

#### 4.4.2 A rail-to-rail buffer for resistive and capacitive loads :

In a high-swing 5v design, the choice of output stage is limited to the common source configuration. Although the source follower and its variants provide better frequency response characteristics, the gate-source voltage overhead imposed by MOS transistor threshold voltage puts a severe limitation on dynamic range in these configurations. Since high output swing was a high priority in this and other designs in this section, a common source output stage was chosen.

The schematic of the buffer [76] is shown in figure 4.12. The dimensions of the devices are given in table 4.5. A complementary differential pair is used in the input stage to provide large input common mode range. The signal is then fed to an intermediate stage which does not contribute to the gain, but enhances the stability. Transistors M19 and M21-M23 operate in the linear mode and are basically used as resistors. These resistors reduce the impedance seen at the points A and B. Therefore, the poles produced by gate-drain capacitance of M24 and M25 will be shifted to a higher frequency. Since the pole produced by the load capacitance is now dominant, the stability will be improved. A careful choice of W/L ratios for these devices results in good stability and yet acceptable open loop gain and large voltage span at the gates of M24 and M25.

Because of large voltage variations at nodes A and B, transistors M19 and M22 are connected in a complementary configuration so that a linear I-V characteristic is maintained for the range of operations. Maintaining a symmetrical structure will also reduce the second harmonic to a minimum and decrease the total harmonic distortion considerably.

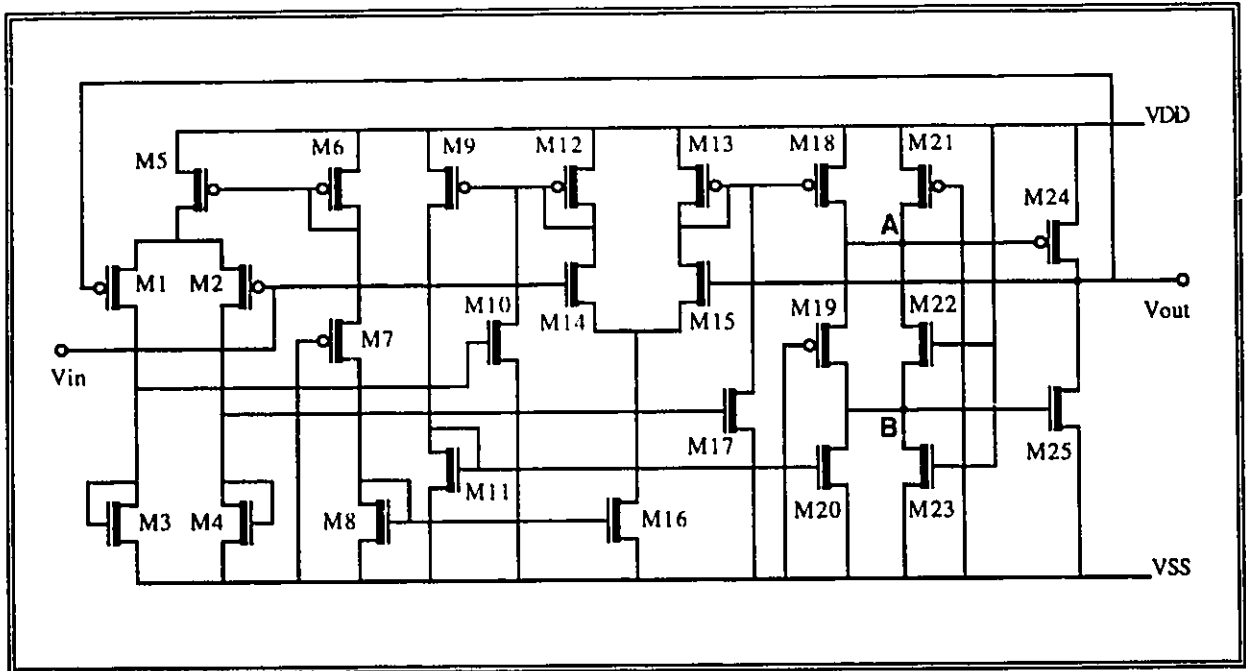


Figure 4.12: Schematic diagram of the buffer amplifier

Device W/L Dimensions ( $\mu\text{m}$ )			
M1, 2, 14, 15	50.4/3	M11, M20	4.2/3
M3, M4	5.4/3	M19	3/9
M5, M16	5.4/3	M21	3/15
M6	7.2/3	M22	3/27
M7	3/6	M23	3/45
M8	6/6	M24	576/3
M9,12,13,18	12/3	M25	252/3
M10, M17	15/3		

Table 4.5: Device geometries for the second buffer

The circuit was realized in  $3\mu\text{m}$  single-polysilicon double-metal p-well process. Die area, excluding the bonding pad area, is less than  $110\text{ mils}^2$ . Maximum sinking and sourcing currents are  $6.1\text{mA}$  and  $5.1\text{mA}$  respectively. With a load of  $5000\text{pF}$ , a down-going slew rate of  $2\text{V/ms}$  was obtained (figure 4.13). The quiescent current is only  $165\mu\text{A}$ .

High die area consumption of most previous designs stems mainly from inefficient utilization of output transistors, i.e. the voltage span on the gates of these transistors are small. The voltage span of the nodes A and B is more than 3 volts (figure 4.14). The circuit is absolutely stable for  $C_L > 130\text{pF}$ .

Load	Poles (rad/sec)		Zeros (rad/sec)	
	Real	Imaginary	Real	Imaginary
$C_L = 200\text{pF}$	$-5.386 \times 10^6$	$1.469 \times 10^7$	$-3.713 \times 10^7$	0
$R_L = 10\text{k}\Omega$	$-5.386 \times 10^6$	$-1.469 \times 10^7$	$-1.531 \times 10^8$	0
$V_{in}=2.5\text{V}$	$-3.938 \times 10^7$	0		
$C_L = 5000\text{pF}$	$-8.783 \times 10^5$	0	$-3.684 \times 10^7$	0
$R_L = 10\text{k}\Omega$	$-1.092 \times 10^7$	0	$-1.533 \times 10^8$	0
$V_{in}=2.5\text{v}$	$-4.027 \times 10^7$	0		
$C_L = 200\text{pF}$	$-5.409 \times 10^5$	$2.426 \times 10^7$	$-3.917 \times 10^7$	0
$R_L = 10\text{k}\Omega$	$-5.409 \times 10^5$	$-2.426 \times 10^7$	$-9.486 \times 10^7$	0
$V_{in} = 4\text{v}$	$-3.866 \times 10^6$	0	$-1.692 \times 10^8$	0
$C_L = 5000\text{pF}$	$-1.650 \times 10^6$	$3.838 \times 10^6$	$-3.156 \times 10^7$	0
$R_L = 10\text{k}\Omega$	$-1.650 \times 10^6$	$-3.838 \times 10^6$	$-9.139 \times 10^7$	0
$V_{in} = 4\text{v}$	$-3.251 \times 10^7$	0	$-1.602 \times 10^8$	0

Table 4.6: Pole-zero configurations (simulation)

The measured transfer characteristics at various loads is given in figure 4.15. Note that the swing is improved compared to the design of section 4.4.1. The frequency response at various loads and operating points are given in figure 4.16. As shown in figure 4.17, the second harmonic is very low, which is a result of the symmetrical structure of the amplifier. Total harmonic distortion curves in a variety of load conditions appear in figure 4.18. The buffer has a systematic offset of 0.54mv and open loop gain of 43dB.

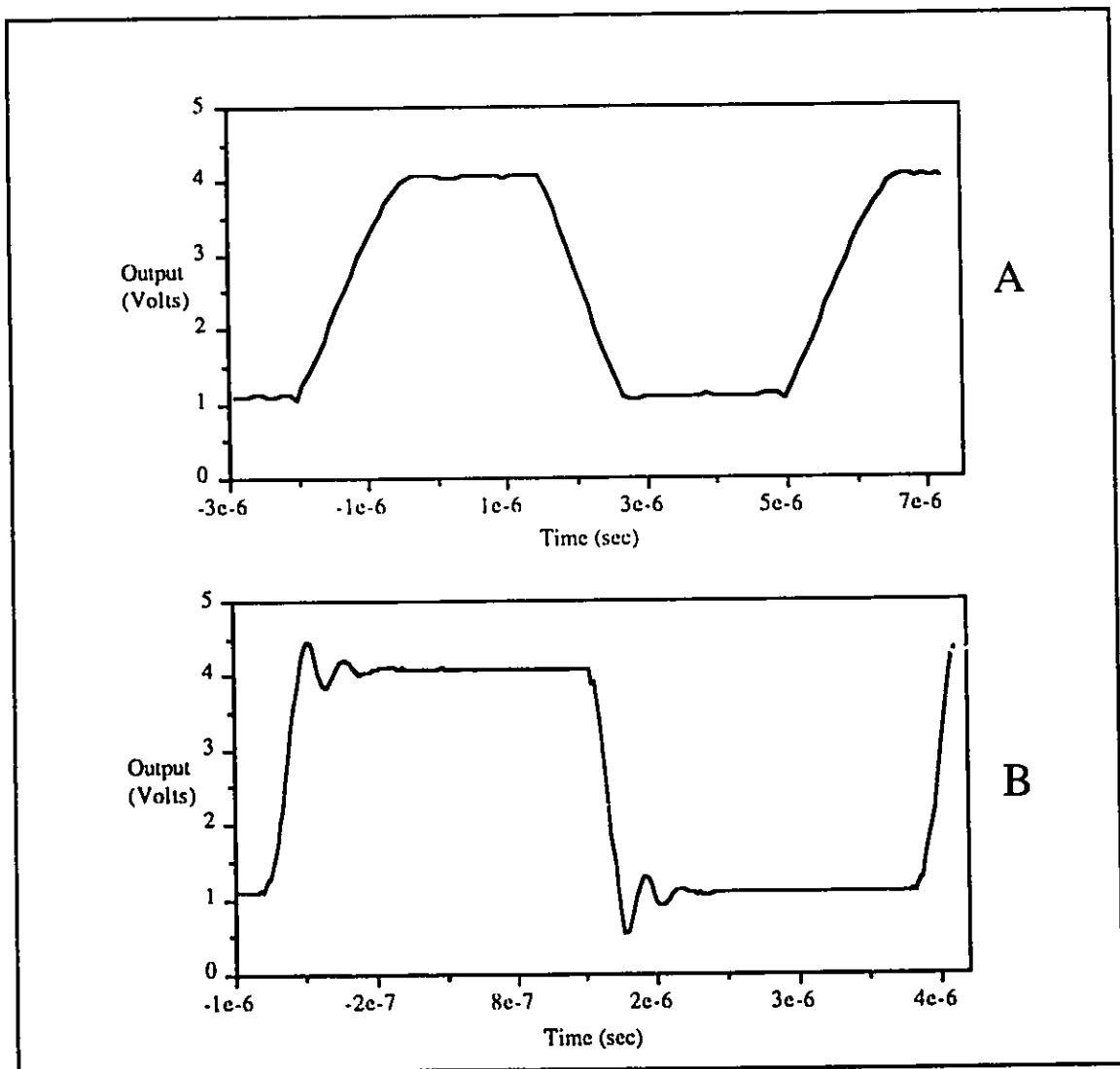


Figure 4.13 : Test results: A) step response with  $C_L = 5000\text{pF}$  B) stability test with  $C_L = 200\text{pF}$

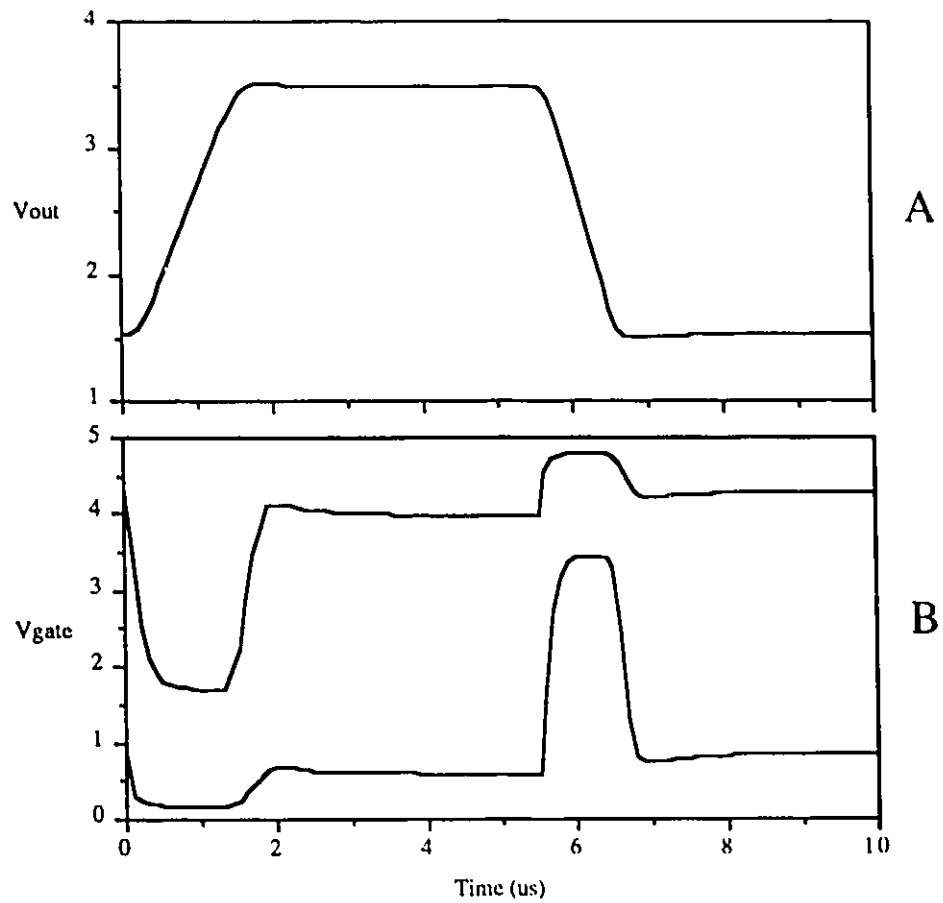


Figure 4.14: Gate voltages M16 and M17 (B) as the output (A) slews (simulation)

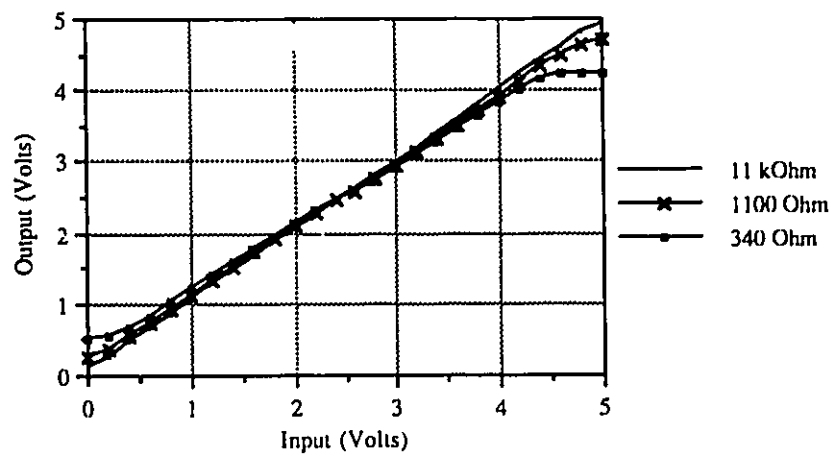


Figure 4.15: Transfer characteristics with various loads (test results)

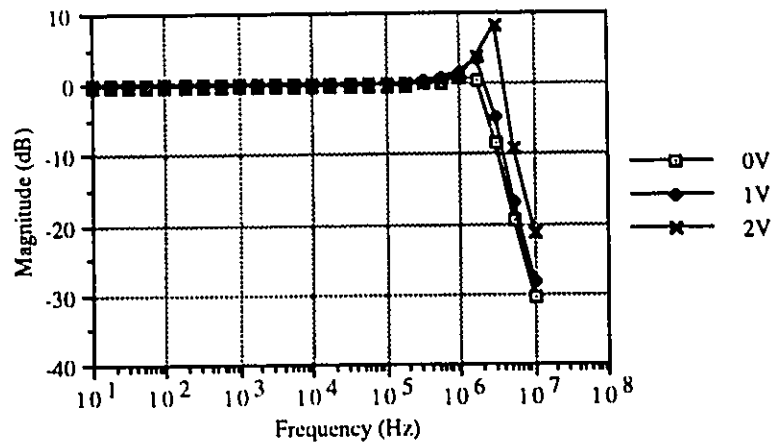


Figure 4.16: Frequency response at different operating points with  $C_L=5000\text{pF}$  (simulation)

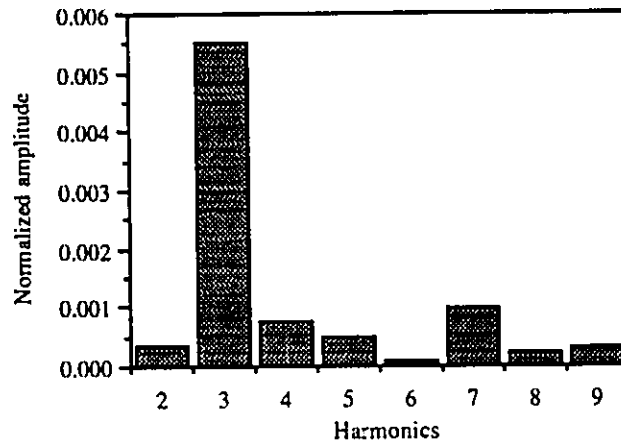


Figure 4.17: Normalized harmonics with  $R_L=10\text{k}\Omega$  (simulation)

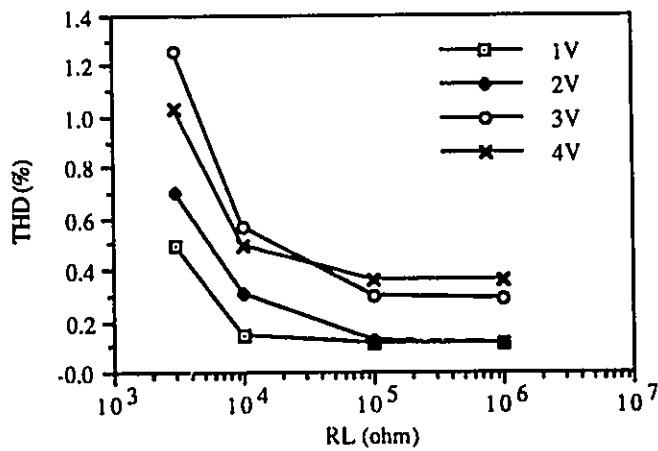


Figure 4.18: Total Harmonic Distortion vs. load at various operating points (simulation)



## 4.5 Summary:

In this chapter, several building blocks, needed for the construction of a multi-layer neural network, were presented. The layouts are in Northern Telecom™ 3 $\mu$ m p-well double-metal single-polysilicon CMOS technology.

The trans-impedance amplifier of section 4.1 is used wherever signals have to be converted from current to voltage mode. It is designed using a differential amplifier and voltage-parallel feedback; compensation is performed by a capacitive element of 0.1pF across the feedback resistor. The voltage to current converter in section 4.2 makes use of the square law characteristics of MOS transistors in saturation and has no feedback. An analysis of this circuit and its transfer characteristic was presented. In section 4.3, a scheme for address generation for the ROM and demultiplexers of the architecture was suggested which results in a more efficient utilization of silicon area. In this scheme, the normal oscillator and address decoders are discarded and a ring oscillator takes the place of both. Finally, in section 4.4, two high-power output buffers were presented. These buffers were designed as output pad drivers and occupy silicon areas of 100 and 110mils<sup>2</sup>.

## Quantization and Truncation Effects

---

While learning algorithms generate continuous valued connection strengths, in the proposed architecture of chapter 2 the weights are represented in a fixed precision format. This means that the weights obtained from the learning algorithms have to be quantized and limited in range before they can be incorporated into the neural network architectures. Because of the non-linear nature of the networks under study, a general analytical investigation of the effects of quantization and truncation is prohibitively difficult. In this chapter, the effects of these two operations, using two simulation examples, will be investigated.

### 5.1 A simple example; XOR-XNOR:

In this example, the XOR-XNOR network of section 2.6 will be revisited. The network and its connection strengths and thresholds are shown in figure 2.10. The dynamic range

of the numbers is limited and there is a concentration of weights around  $\pm 8$ . The maximum range is chosen as  $\pm 8$  and no truncation will be necessary. With 8 bits of accuracy, the resolution will be equal to 0.0625.

Figure 5.1 shows the XOR-XNOR network with quantized weights and thresholds. These are actually the values stored in the ROM in the circuit level simulations of section 2.6. System level simulations show almost no change in the behavior of the circuit and the outputs are the same as the original up to two significant digits.

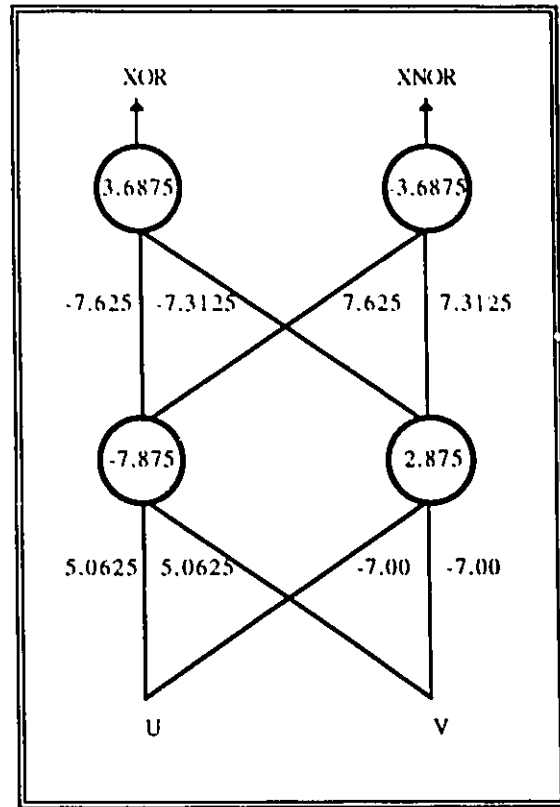


Figure 5.1: XOR-XNOR with quantized weights

## 5.2 A complex example; handwritten character recognition:

The objective, in this second example, is to build a neural network that assigns digitized images of handwritten numerals to the correct class. To achieve this end, the methodology discussed in section 1.6 for handwritten character recognition will be used. 11,100 handwritten numerals were run through the preprocessing and feature extraction [53] stages. Out of this, 5900 samples were used for training and 5200 were assigned to the test set. Feature vectors of the samples in the training set were then used to train a maximally connected 3-layer feed-forward network with the back propagation learning

algorithm. The network has 36 inputs, each corresponding to one feature, and 3 layers of 20, 15 and 10 neurons each. All the thresholds were set to zero throughout the training, which took place in 1.1 million epoches.

At the time of recall, the outputs of the last stage are thresholded. If one and only one of the outputs is higher than  $T_1$ , and all others are lower than  $T_2$ , then a recognition has been effected. Multiple membership occurs if more than one output is higher than  $T_3$ , and no output lies between  $T_2$  and  $T_3$ . In this case, the sample is recognized as a member of a subset of the character set. For example, a handwritten sample may not be fully recognizable, but be identified as either 1 or 7. In all other cases, the results are inconclusive and the sample is rejected. A successful recognition occurs when one and only one output is higher than  $T_1$  and it is the correct output.

The percentage of error and rejection are related. It is possible to avoid some errors by increasing  $T_1$  and/or decreasing  $T_2$ , putting more of the samples in the rejected set. The error and rejection rates for the test set are given in table 5.1. Histogram of the connection strengths is shown in figure 5.2. Distribution of the weights is well behaved and has an almost Gaussian shape. The weights are all between -27 and 23.

T <sub>2</sub> /T <sub>1</sub>	%error / %reject					
	0.98	0.95	0.90	0.85	0.75	0.65
0.01	0.56/8.31	0.62/7.60	0.69/7.23	0.73/7.06	0.77/6.92	0.77/6.90
0.05	0.62/4.52	0.67/3.44	0.79/2.88	0.83/2.69	0.88/2.54	0.88/2.52
0.10	0.62/4.08	0.69/2.94	0.81/2.29	0.87/2.06	0.92/1.90	0.92/1.85
0.20	0.62/3.98	0.73/2.81	0.87/2.12	0.92/1.88	0.98/1.71	0.98/1.69
0.30	0.63/3.88	0.75/2.69	0.88/2.00	0.94/1.73	1.02/1.54	1.02/1.50
0.65	0.63/3.69	0.77/2.46	0.90/1.75	0.96/1.46	1.04/1.25	1.04/1.15

Table 5.1: Recall accuracy with the original connection strengths  
 $T_3 = 0.65$  Multiple Membership %0.15

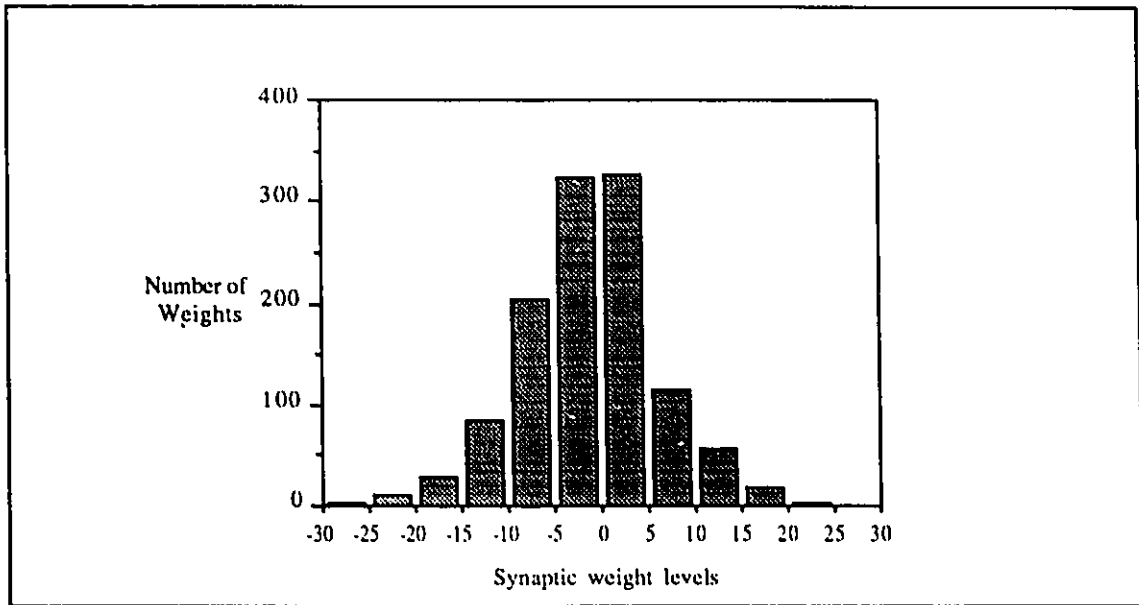


Figure 5.2: Distribution of the synaptic weights in Example 2

We now try to fit the network into the proposed architecture. Before anything else, a truncation level should be decided upon. A lower truncation level will result in higher error for weights above that level. But at the same time, those weights below the truncation level will have higher accuracy, since the number of bits is constant (lower quantization error).

The tails of the distribution contain relatively few elements. This suggests that truncation of these elements may be a good price to pay for enhanced accuracy of all others. Tables 5.2, 5.3 and 5.4 show the results of running the test set through the network with weights truncated at various values and quantized with 8 bits. By an examination of error and rejection rates, it is evident that it is best to have no truncation at all. Even truncation at 20, which affects less than 1% of the connection weights, will have a perceptible adverse effect on recognition accuracy. In other words, the network seems to be more sensitive to maximum error in the weights than to mean squared error.

		%error / %reject					
T2NT1		0.98	0.95	0.90	0.85	0.75	0.65
0.01		0.54/8.17	0.58/7.35	0.63/7.10	0.67/6.94	0.71/6.83	0.73/6.79
0.05		0.58/4.65	0.63/3.54	0.73/3.04	0.77/2.79	0.83/2.62	0.87/2.52
0.10		0.60/4.29	0.67/3.13	0.79/2.56	0.83/2.27	0.90/2.04	0.94/1.94
0.20		0.60/4.13	0.67/2.98	0.79/2.37	0.83/2.08	0.90/1.81	0.94/1.71
0.30		0.62/4.00	0.71/2.83	0.85/2.17	0.88/1.85	0.98/1.52	1.02/1.42
0.65		0.62/3.87	0.73/2.65	0.87/2.98	0.90/1.60	1.00/1.27	1.08/1.10

Table 5.2: Recall accuracy with the weights quantized but not truncated  
T3 = 0.65 Multiple Membership %0.13

		%error / %reject					
T2NT1		0.98	0.95	0.90	0.85	0.75	0.65
0.01		0.56/8.96	0.67/8.33	0.71/8.02	0.73/7.85	0.75/7.67	0.79/1.62
0.05		0.58/5.19	0.69/4.31	0.73/3.77	0.77/3.56	0.81/3.29	0.87/3.13
0.10		0.62/4.77	0.79/3.73	0.87/3.12	0.90/2.88	0.94/2.56	1.00/2.38
0.20		0.63/4.62	0.83/3.50	0.90/2.87	0.96/2.48	1.02/2.13	1.08/1.92
0.30		0.65/4.42	0.85/3.31	0.96/2.63	1.06/2.21	1.13/2.81	1.19/1.60
0.65		0.67/4.29	0.87/3.13	0.98/2.44	1.08/2.00	1.15/1.52	1.27/1.19

Table 5.3: Recall accuracy with weights truncated at 20 and quantized  
T3 = 0.65 Multiple Membership %0.23

		%error / %reject					
T2NT1		0.98	0.95	0.90	0.85	0.75	0.65
0.01		0.71/22.13	0.85/21.31	0.92/20.79	0.98/20.54	1.06/20.08	1.12/19.88
0.05		0.90/12.44	1.10/11.35	1.17/10	1.25/10.50	1.35/9.92	1.46/9.60
0.10		0.98/9.65	1.21/8.38	1.29/7.77	1.37/7.40	1.54/6.73	1.65/6.40
0.20		1.08/8.06	1.35/6.69	1.42/6.06	1.50/5.67	1.67/4.94	1.83/4.58
0.30		1.10/7.40	1.37/5.96	1.44/5.27	1.52/4.88	1.77/4.06	1.96/3.65
0.65		1.25/6.56	1.54/5.00	1.63/4.27	1.88/3.65	2.19/2.71	2.42/2.21

Table 5.4: Recall accuracy with weights truncated at 14 and quantized  
T3 = 0.65 Multiple Membership %1.12

Even more surprising is the fact that in most threshold combinations, the network with quantized weights shows superior performance in comparison with the original network (compare tables 5.1 and 5.2). This happens despite the loss of information in the quantization process.

### **5.3 Summary:**

In this chapter, the effects of quantization and truncation of connection strength values on the performance of feed-forward networks were investigated. Because of the nonlinearities of the network, a formal analysis is extremely difficult and so an empirical approach, using two examples, has been used to illustrate the effects numerically.

Using these examples, we have found that feed-forward networks are highly sensitive to maximum error values in connection strengths resulting from truncation, whereas distributed errors are tolerated. Most interestingly, it is seen that quantization of the weights can even result in an improvement in network performance. More work is needed, however, to determine whether this behavior is an isolated case or representative of a general phenomenon.

## Conclusion

---

Neural networks are an outcome of the studies of human brain and behavior. They promise to be more robust, reliable and fault tolerant than other engineering solutions in many applications [77]. The list of such applications grows as the research into neural models continues.

Multi-layer feed-forward networks are an important subclass of neural nets, historically and otherwise. Their roots go back to the first quantitative neural model: the Perceptron [10]. Feed-forward networks have the advantage of inherent stability, since they have no feedback paths, and yet they are powerful and general to a large degree: It has been shown that any non-decreasing continuous function of many variables can be represented by a composition of no more than 3 layers of continuous functions of one variable [78]. Also, feed-forward networks have one of the most powerful and (from an engineering viewpoint) useful learning algorithms; back error propagation. They have been successfully used in numerous pattern recognition applications.



Although software implementations of neural networks are useful in exploring the behavior of neural models, it takes a long time for these software simulations to run, mainly because they are an attempt to model a highly parallel system on serial hardware. To utilize the potential of neural networks to the fullest, a hardware realization is needed. Analog CMOS VLSI provides a suitable vehicle for such an implementation, in a highly cost effective manner.

In this thesis, an architecture for feed-forward multi-layer neural networks is proposed and its building blocks introduced and analyzed. The architecture is based on, and the building blocks are designed with, analog and hybrid CMOS circuitry. The two main objectives in the design process were: a) Reducing the number of physical interconnections b) generating reliable means of storing the weights. The first objective led to an imbedded multiplexing approach, and the second to a hybrid architecture.

The accompanying reduction in the number of interconnections creates new potentials, namely the possibility of multichip systems without any hardware overhead or extensive interchip wiring. This can lead to a semi custom approach where only one generic chip containing one stage of the proposed architecture is designed, with a maximum number of neurons and connections. These chips can be independently programmed and cascaded to form the desired network, thus dramatically reducing the turn around time in the development process.

The original form of the architecture was proposed and analyzed in chapter 2. It was seen that in this scheme, the number of interconnections and multipliers is reduced considerably from a full implementation. Also, some modifications were proposed to improve the speed and output waveforms. The problem of matching the units and scales in the network was

addressed in section 2.4. The I/O mode of the network and related issues were discussed in section 2.5.

Realizations of the two main building blocks, neuron and synapse, were introduced in chapter 3. The neuron was presented in four variations: current and voltage mode, variable and fixed threshold. The MOS differential pair was used to construct the desired sigmoidal transfer characteristics. The synapse is made of a ROM structure and a multiplying D/A converter. The former stores the connection strengths and the latter performs the synaptic multiplication. The cascode current mirror was preferred in the design of the D/A because of its graceful degradation of linear characteristics at lower drain voltages.

In chapter 4, the other building blocks needed in the architecture were presented. The trans-impedance amplifier of section 4.1 is constructed with a differential amplifier and voltage-parallel resistive feedback. Compensation is performed by a capacitive element across the feedback resistor. A voltage to current converter was discussed in section 4.2. This converter uses the MOS transistor square law characteristics. The currents of different branches are added and subtracted in a way that the quadratic terms cancel out and the input/output characteristic is linear.

A set of high drive, low footprint output buffers were introduced in section 4.4. The improved performance is obtained through the use of a novel compensation technique. The main idea in these designs is to move the critical pole of the system to higher frequencies by reducing the impedance of relevant nodes. In this way, large internal compensation capacitors are avoided.

Finally, in chapter 5, the effects of quantization and truncation of the weights on two examples are investigated. The two main results are:

- a) Feed-forward networks seem to be sensitive to the maximum value of error in the connection strengths, but not so to the distributed quantization error.
- b) Quantization of the weights can sometimes improve the overall performance of the network.

This work of research has paved the way to the development of practical hardware implementations of feed-forward neural networks. Future improvements are possible in the area of circuit design, especially in the case of the trans-impedance amplifier. Going to higher integration technologies adds some factors to circuit design, and the current building blocks may have to be modified before they can be used in more advanced integrated circuit technologies. The effect of deviation of the building blocks from ideal characteristics on network performance is not yet well understood and more work in this area is needed before a practical, completely general purpose feed-forward network can be developed. Also, the studies in chapter 5 on quantization and truncation are suggestive of important results, but not conclusive. It is certainly an area worth further investigation.

---

## A Brief Overview of the Nervous System<sup>1</sup>

---

The nervous system is the network of organs and tissue that controls and coordinates all the activities of an animal's body. Except for some one-celled organisms, nearly all animals have a nervous system of some kind. In fact, all the coordinated activities of a multicellular animal are made possible by a nervous system. As the scale of development is ascended from lower to higher animal forms, the arrangement of nervous tissue into organs and systems becomes increasingly complex.

In humans, the central nervous system, consisting of the brain and spinal cord, controls all the activities of the body, including internal organ function, sensory perception, conscious movement, and complex functions such as thought and memory. The central nervous system exercises its control through the peripheral nervous system, where nerves connect the brain and the spinal cord with all other parts of the body.

---

<sup>1</sup> The material and figures in this appendix were taken from [79-81]

To carry out its task of determining the many aspects of behavior and controlling directly or indirectly the rest of the body, the nervous system possesses an immense number of lines of communication provided by the nerve cells (Neurons). These are the fundamental units or building blocks of the brain. These cells function through electrochemical signals known as *impulses*. Sensory impulses arising in special receptor organs travel along *sensory neurons* to the central nervous system, keeping the body aware of its environment. *Motor neurons*, carrying motor impulses, run from the central nervous system out to muscles and glands of the body, controlling their activities. Within the central nervous system, *associative neurons* relay impulses from sensory neurons to motor neurons.

That neurobiologists who study single nerve cells should be able to discuss higher functions of the brain, such as perception, is a very recent and somewhat unexpected development. It has long been realized that knowledge of cellular properties of neurons is essential for any detailed study of the brain. Nevertheless, it seemed quite possible that the workings of the cerebral cortex would still remain a mystery even if a great deal was known about signaling in individual neurons, especially because brain is a very diversified organ, in which different cells perform different specific tasks. A good example is provided by the neurons that excite or start signals as contrasted to neurons that inhibit or suppress signaling. In addition to their different chemistry, inhibitory and excitatory neurons obey different plans of connection. Moreover, the cells in the brain are connected with one another according to a complicated but specific design that is of far greater complexity than the connections between cells in other organs.

Fortunately, there are many simplifying features in the nervous system. First, it has only two basic types of signals, one for short and the other for long distances. Second, these signals are virtually identical in all nerve cells of the body, whether they carry messages to or from centers, or are the result of painful stimuli or touch, or simply interconnect various

portions of the brain. The signals are remarkably similar even in different animals. The signals themselves cannot be endowed with special properties because they are stereotyped and much the same in all nerves. The mechanisms by which signals are generated are also similar, though with interesting variants. Thus, the brain deals with symbols of external events, which do not resemble the real objects in any way.

Although a great deal of information can be conveyed by a series of agreed upon symbols, the frequency or pattern of discharges in the nervous system does not serve as a code. The reason is that even though impulses and frequencies are much the same in different cells responding to light, touch, or other stimuli, the content of the information is quite different. The meaning of a signal depends on the origin and destination of the nerve cells, that is, their connections. The arrangement of these connections accounts for the almost infinite wealth of information that reaches us from the world. This point has been confirmed by many experiments [79].

### **A.1 Neuron:**

The basic structural unit of the nervous system is the neuron. Several other types of supportive cells are also found in the nervous tissue. Although the neuron resembles all other cells in its metabolism, it differs from ordinary cells in that it cannot reproduce itself. A neuron has three distinct parts: a cell body, a single axon, and one or more dendrites (figure A.1). The cell body contains the nucleus. A very thin membrane encloses the contents of the cell body.

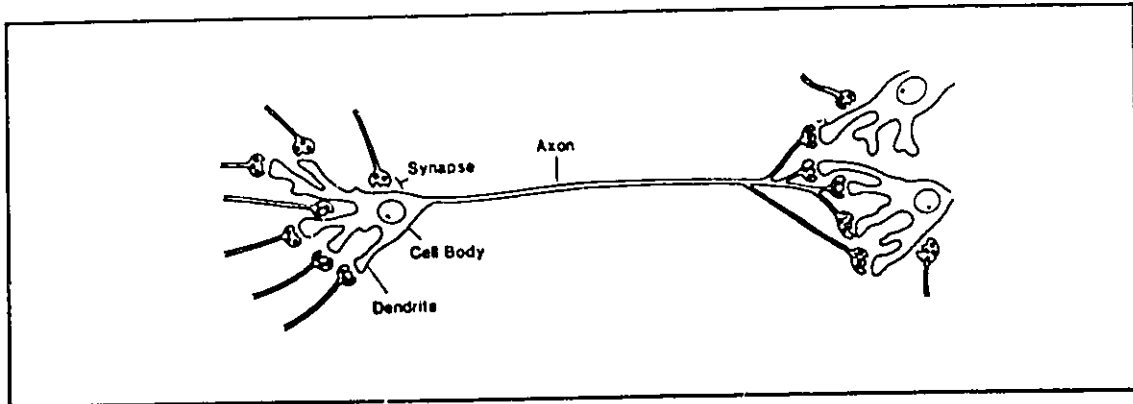


Figure A.1: Neuron and its different sections

Human neuron cell bodies vary greatly in shape and size. The smallest, in cerebellum of the brain, are only about 4 microns in diameter. The largest, found in certain motor neurons of the spinal cord, are about 135 microns in diameter. Sensory cells are usually round, motor cells tend to be star shaped, and those in the cerebral cortex are pyramidal or diamond-shaped (figure A.2)

The nerve cell extensions -axon and dendrite- are also known as nerve fibers. Dendrites are branching structures. Dendrites carry nerve impulses toward the cell body and are called *afferent processes*. Axons are longer processes with fewer branches. They carry impulses away from the cell body and are called *efferent processes*.

## A.2 Neural connections: The synapse:

When a receptor is stimulated, the impulse travels along the dendrite through the cell body and along the axon. The impulse must then be transferred to other cells in order to reach the central nervous system. This transfer occurs at a junction between the axon of one neuron and the dendrite or the cell body of the next neuron. This junction is called a synapse. There is a gap of a few microns between the cellular membranes at the synapse.

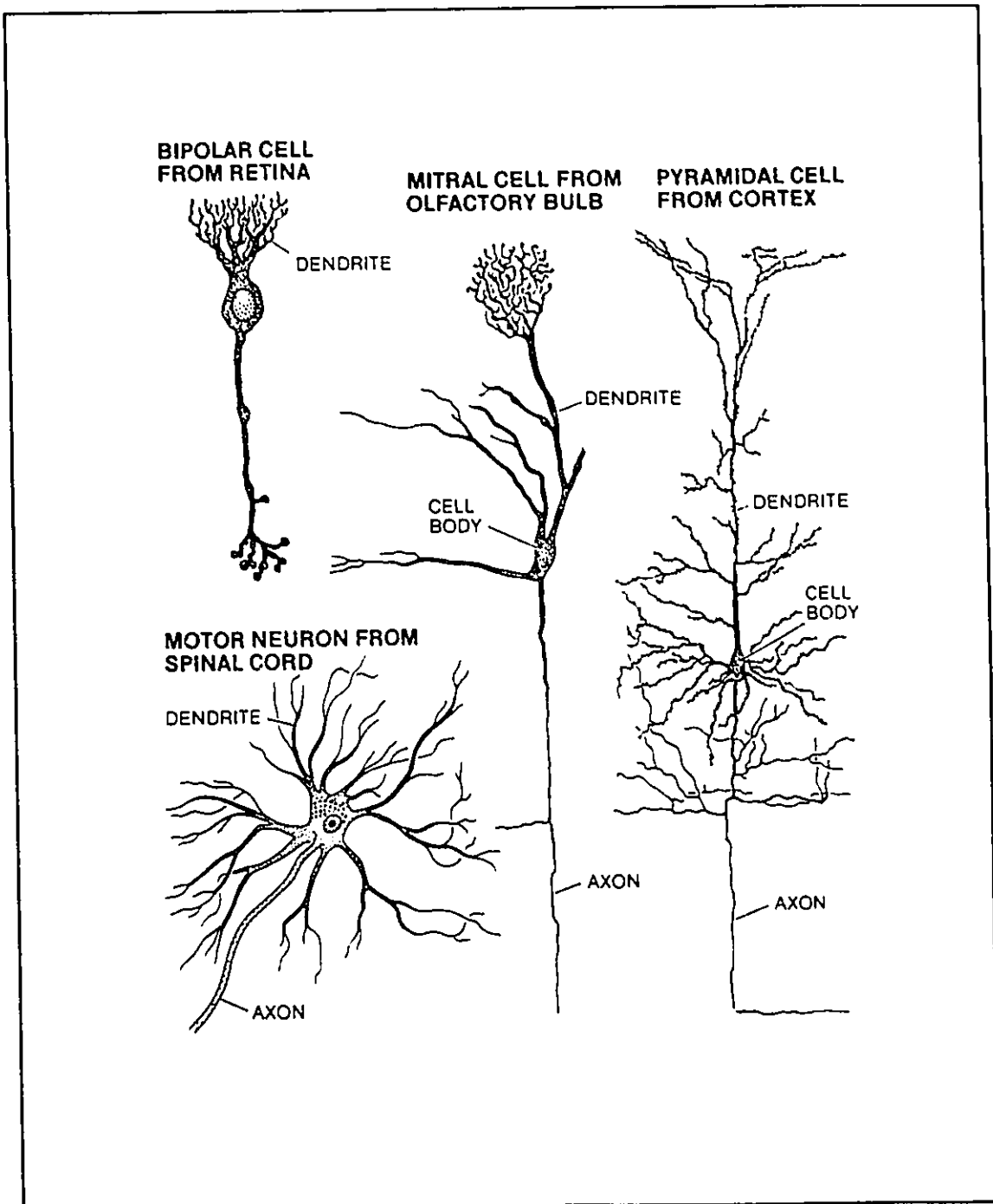


Figure A.2: Different neuron types



Transmission of the nerve impulse across synapses is effected by the release of a tiny amount of a chemical transmitter substance. One such substance, acetylcholine, is released by motor axon nerve endings at muscle-nerve junction. In about one millisecond, the action of acetylcholine is terminated by the enzyme cholinesterase. Thus there may be excitatory and inhibitory effects in action throughout the nervous system.

### **A.3 Electrical nature of a nerve impulse:**

Measurements of electrical activity have revealed much about the nerve impulse and the reaction of a neuron to a stimulation. It was once thought that the passage of a nerve impulse and of an electrical current through a wire were similar. However, it is now known that there are two important differences between a nerve impulse and electric current in a conductor. In the first place, the speed of a nerve impulse is about 120m/sec. This is much slower than the propagation of electrical energy. In the second place, there is an electrochemical change in the membrane surface of the axis cylinder of the neuron process that, once started, is self-propagating. Thus the neuron itself supplies the energy for the transmission of a nerve impulse.

Actually, passive propagation of electrical impulses along the nerve would be quite impossible. A nerve fiber can be considered as a tube containing axoplasm, a watery solution of salts and proteins. The cell membrane restricts the diffusion of ions and separates the axoplasm from the outside fluid, which has the same ionic strength but a different composition. Axoplasm as a conductor of electricity is about  $10^7$  times worse than copper. Considering the small diameter of the nervous fibers, typically between  $0.1\mu$  and  $10\mu$ , the resistance of these fibers would be quite high. A simple calculation shows that in a  $1\mu$  fiber containing axoplasm with a resistivity of 100 Ohm cm, the resistance per

unit length is about  $10^{10}$  Ohm per cm. A signal propagating passively in this medium would be dispersed completely within a fraction of a millimeter.

As stated before, the action of a neuron on another can be excitatory or inhibitory. The process that sorts out these various influences, some acting in concert, some opposing each other, is called the *integrating action of neurons*. Integration at the cellular level is simply the way in which impulses (long-range signals) that converge on a cell become translated into postsynaptic potentials (local or short-range signals). These then determine the firing of the neuron, that is, the generation of a new impulse that contains the synthesis of all the various inputs.

Each neuron has a threshold level for the generation of an impulse. It is a critical point and any single stimulus below this critical level will not start a nerve impulse. If, however, two or more stimuli, each of less than threshold intensity, are received within a short period of time, they reinforce each other and can start an impulse (integrating action).

The characteristic pattern of the nerve impulse is the action potential, or spike potential. During the action potential, the excitability of the neuron is reduced to zero so that it is impossible to send another impulse along the fiber. This is the *absolute refractory period*, which lasts for about 1msec. Therefore the theoretical rate at which impulses can be transmitted along the nerve is 1000 per second. Then, for about 5msec, there is a *relative refractory period*, during which a much stronger than normal stimulus is needed for the transmission of the impulse. Following this, there is a subnormal period, or *negative afterpotential*, when the neuron has increased excitability and the threshold is lower than normal. This in turn is followed by the *positive afterpotential*, with the neuron in a state of decreased excitability before the neuron returns to normal.

Electrical condition in the nerves depends on the movement of sodium ions ( $\text{Na}^+$ ) and of potassium ions ( $\text{K}^+$ ). Sodium is ordinarily in higher concentration outside the nerve fiber membrane, while potassium is higher inside. During an action potential, there is a movement of sodium ions into the fiber and of potassium ions out of the fiber. This movement of the ions is the main generator of the action potential.

The strength of an impulse carried over a nerve fiber does not depend on the strength of the stimulus. Once threshold is reached, the fiber responds to its full capacity. If a section of a nerve is exposed to alcohol vapor, conduction of nerve impulses through this section is depressed, so that fewer impulses are transmitted through this section. However, if an impulse gets through at all, it is as strong beyond the depressed section as it was before. This phenomenon illustrates the *all-or-none* law. The strength of the stimulus does not affect the amplitude of the impulse, but is instead reflected in the frequency of the firing of the nerve (figure A.3). The neuron is therefore not an amplifier, but a voltage to frequency converter.

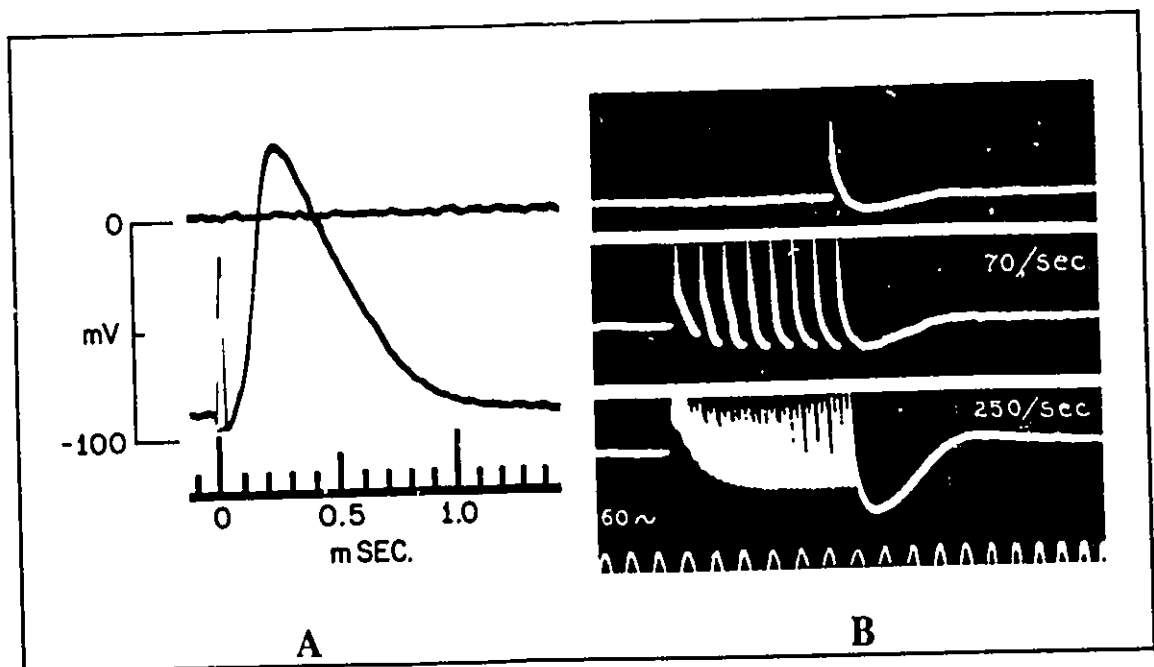


Figure A.3: A) A typical nerve impulse B) Output of a neuron

#### A.4 Summary:

Nervous system is the network of organs and tissue that controls and coordinates the activities of the body. The fundamental building block of this system is the neural cell or neuron. Neurons are made of three parts: cell body, dendrite and axon. They generate and propagate nervous electrochemical signals known as impulse. These impulses are transmitted along the neuron by ion concentration inversion, which is an active process. The neuron itself supplies the energy needed for the propagation of the signal.

Impulses are transmitted from one neuron to another at junctions known as synapse. Synapses are places where the axon of one cell gets close to dendrites or cell bodies of other neurons. Signals are transmitted through the synaptic gap by the release of chemical transmitter substances. The effect of the activities of one neuron on the next can be either exhibitory or inhibitory, depending on the type of neuron and synapse.

In each neuron, the effect of the signals received through synapses are combined in a process called integrating action, and the neuron fires if the combined result is higher than a certain threshold. The amplitude of the impulse does not depend on the strength of the stimulus. Instead, the strength of the stimulus is reflected in the firing rate of the neuron. Also, the shape and amplitude of the impulse is the same across the nervous system. This shows that the impulse itself does not carry any information regarding the type of the stimulus or its properties. This information is stored in the topology of the network, the synaptic weights, and the firing rate of the neuron.

## Layout Diagrams

---

This appendix contains layout diagrams of the building blocks of the architecture proposed in this thesis. They are all implemented in the Northern Telecom™ 3 $\mu$  double-metal single-polysilicon p-well CMOS process. The cells are:

1. Current mode neuron with variable threshold
2. Current mode neuron with fixed threshold
3. Voltage mode neuron with variable threshold
4. Voltage mode neuron with fixed threshold
5. Multiplying D/A
6. ROM with corresponding line buffers
7. Trans-impedance amplifier
8. Voltage to current converter
9. Output buffer for high capacitive loads
10. Output buffer for high resistive and capacitive loads

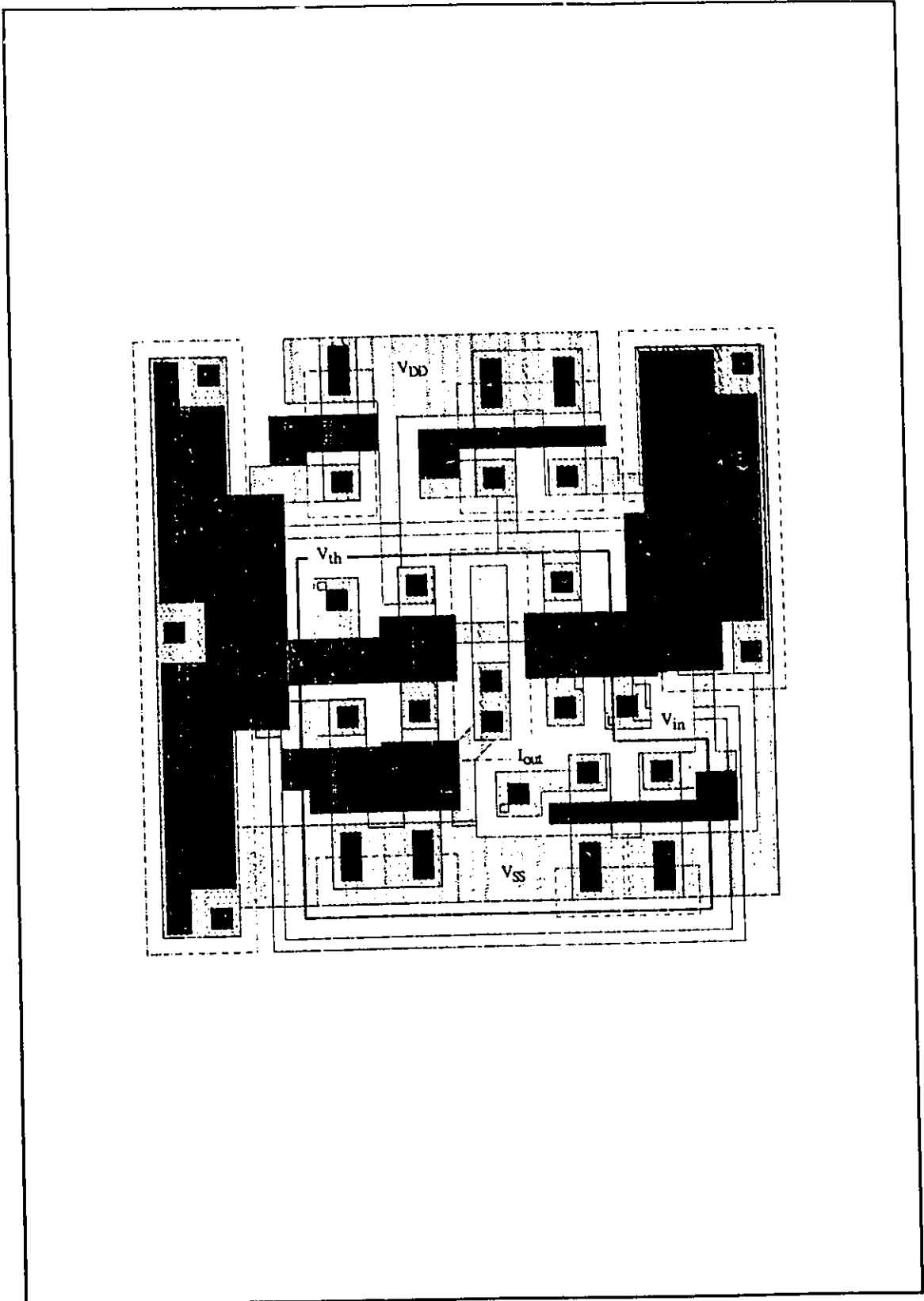


Figure B.1: Layout diagram of the current mode neuron with variable threshold

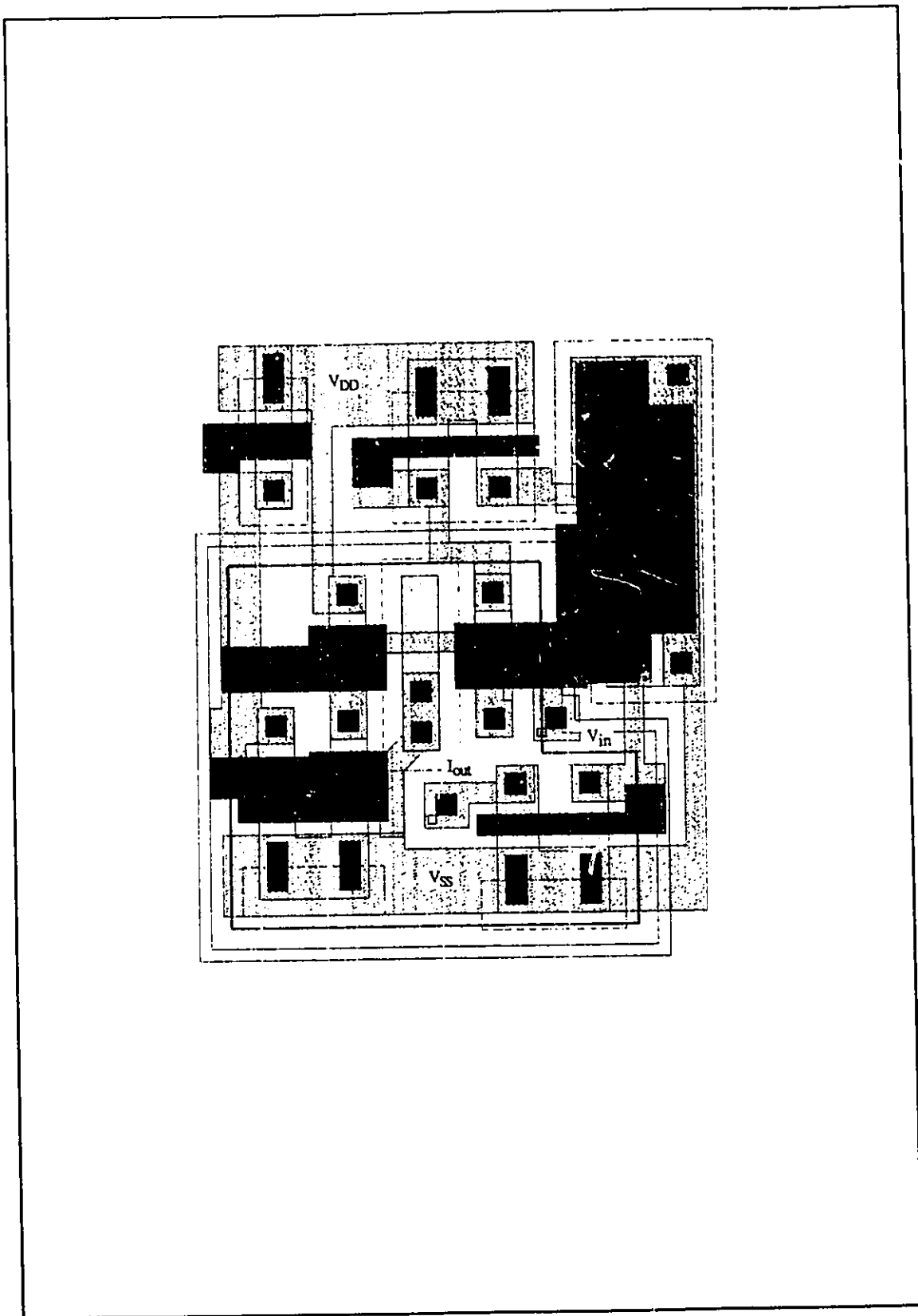


Figure B.2: Layout diagram of the current mode neuron with fixed threshold

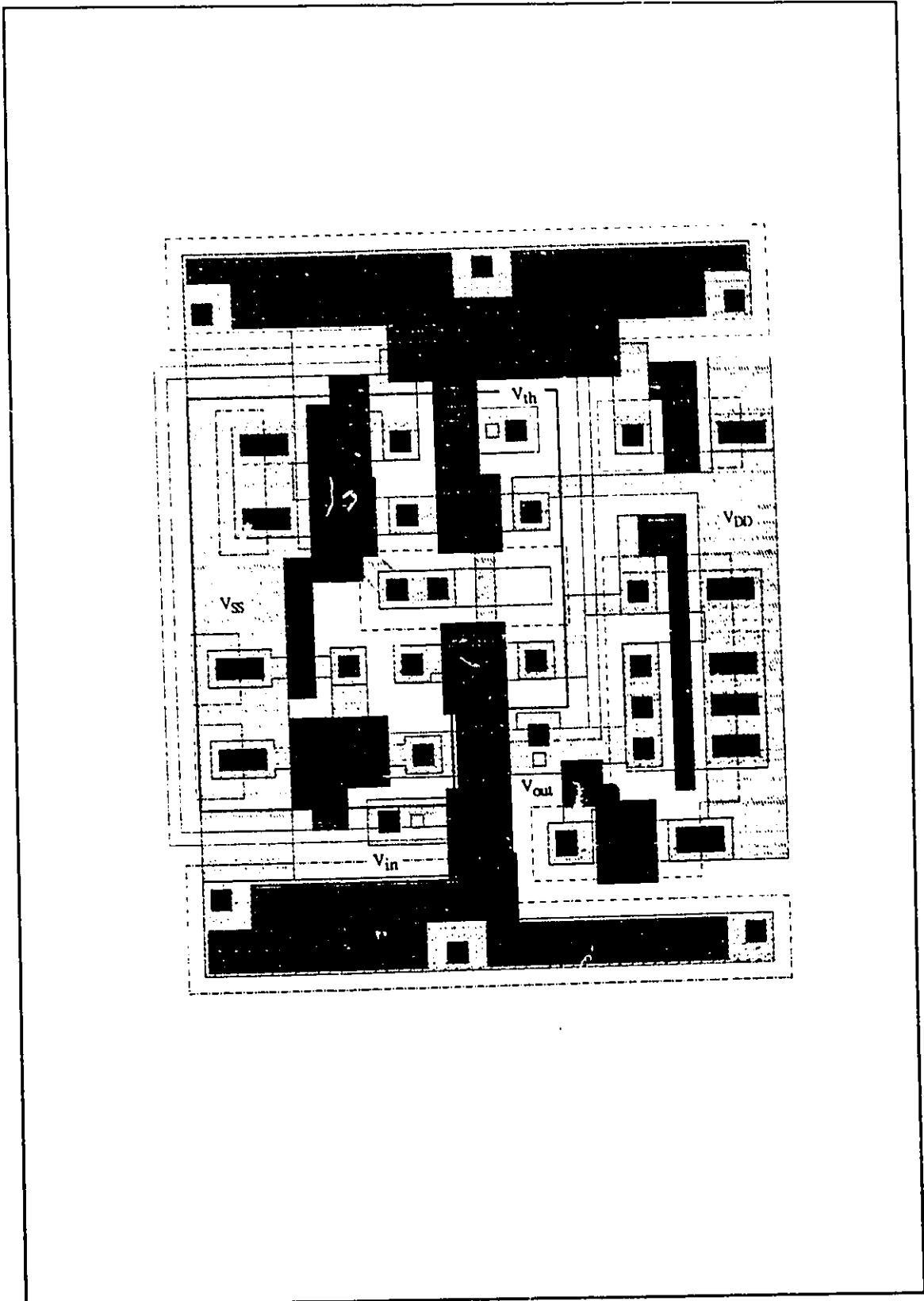


Figure B.3: Layout diagram of the voltage mode neuron with variable threshold



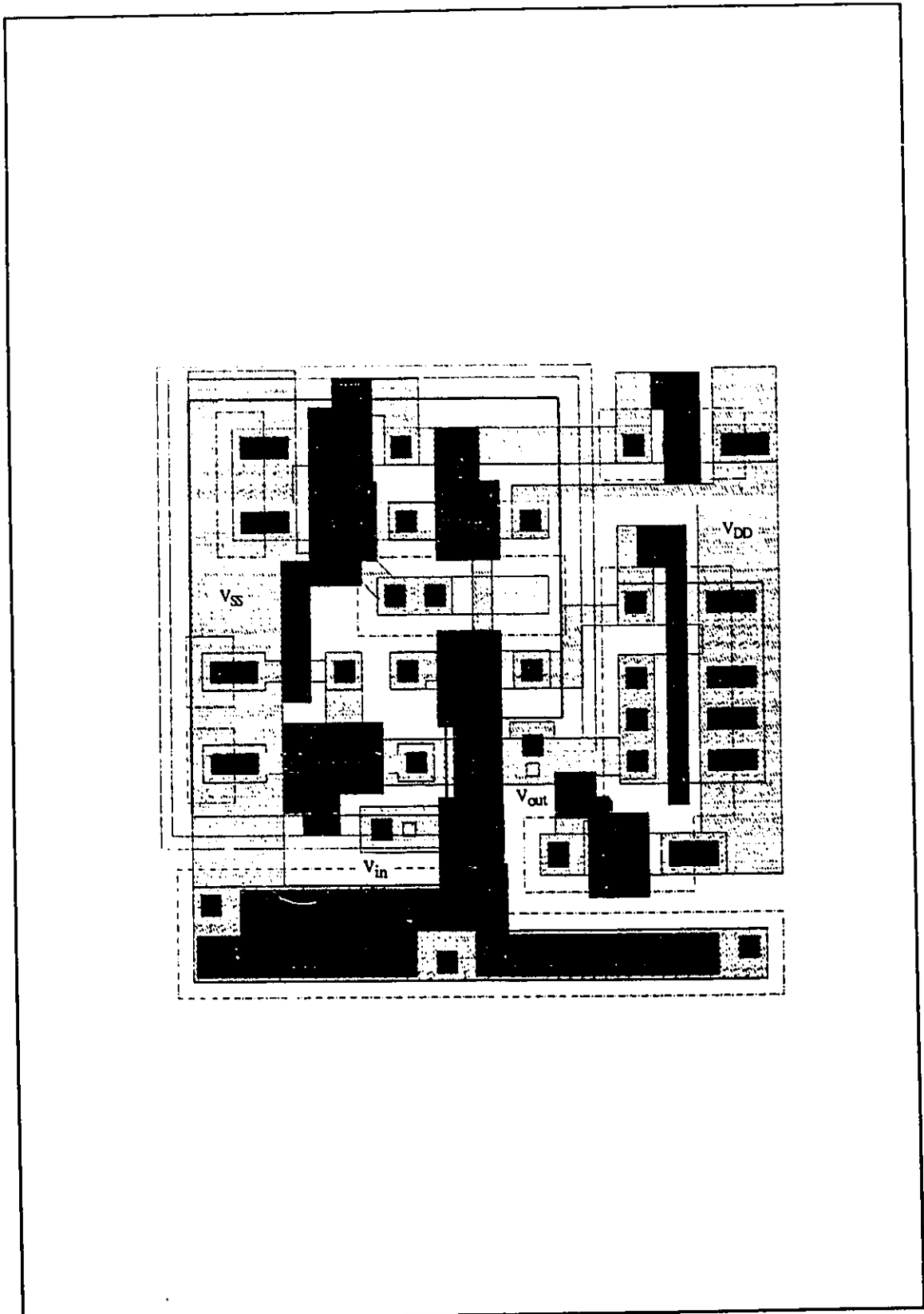


Figure B.4: Layout diagram of the voltage mode neuron with fixed threshold

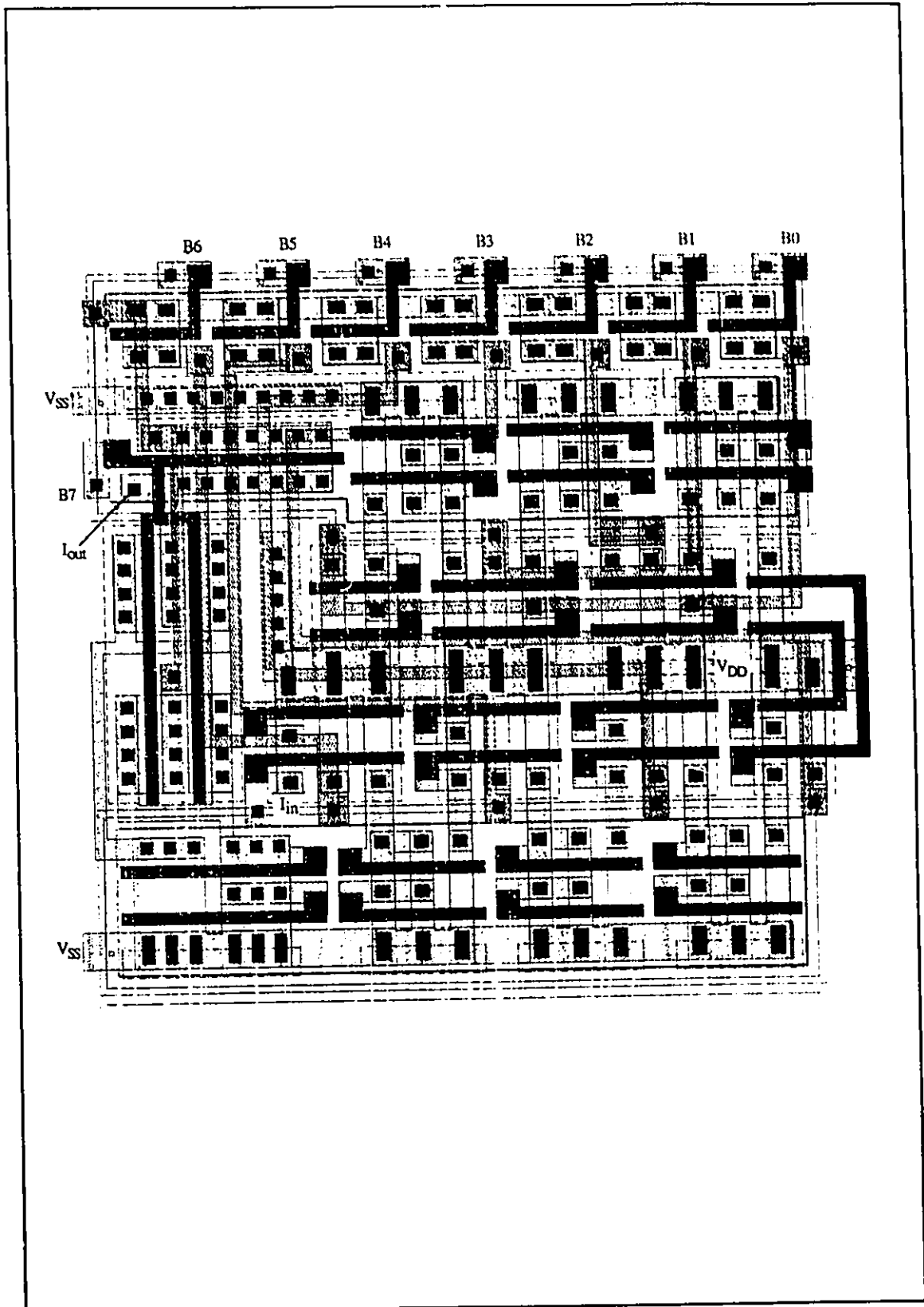


Figure B.5: The multiplying D/A

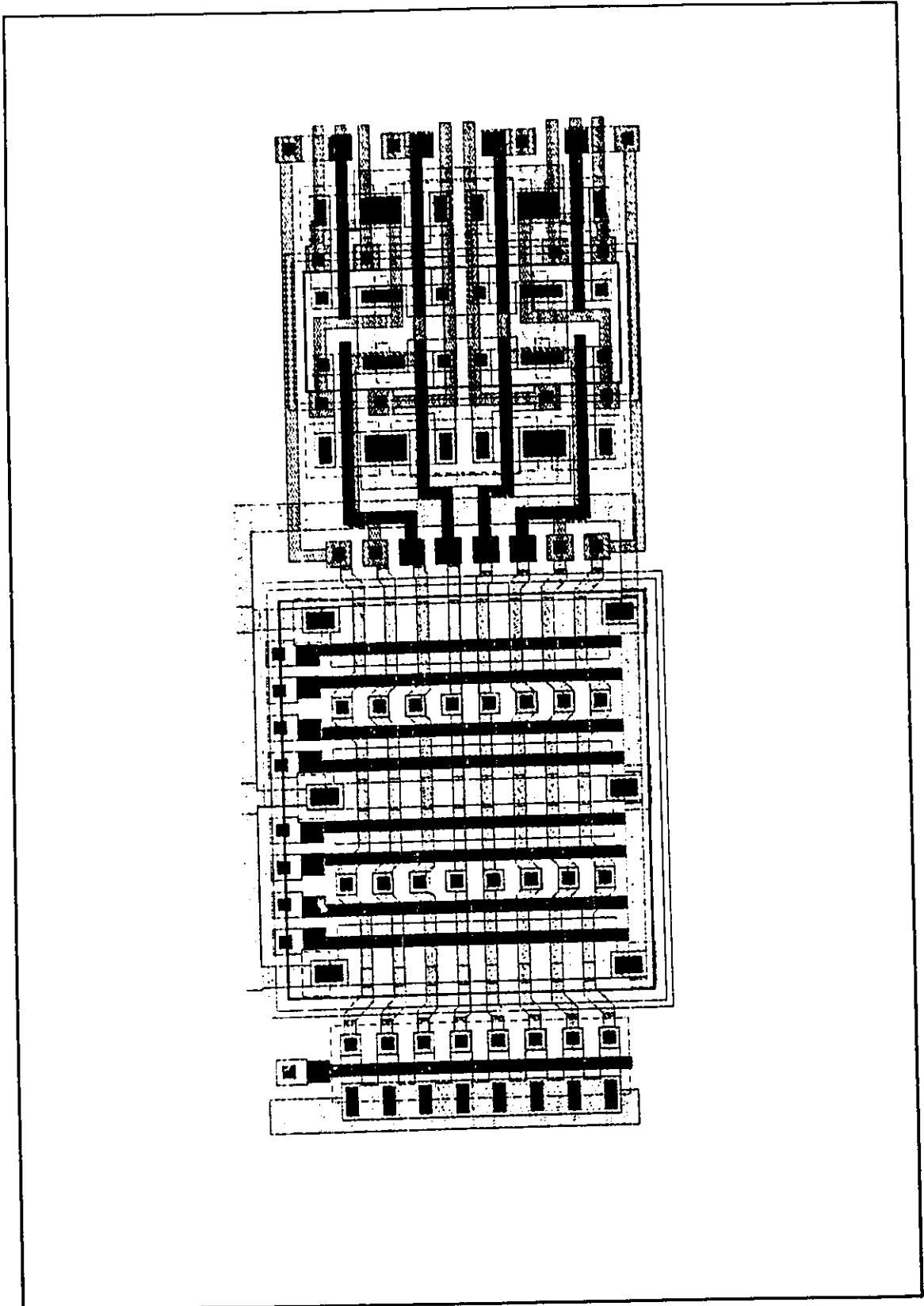


Figure B.6: ROM with buffers

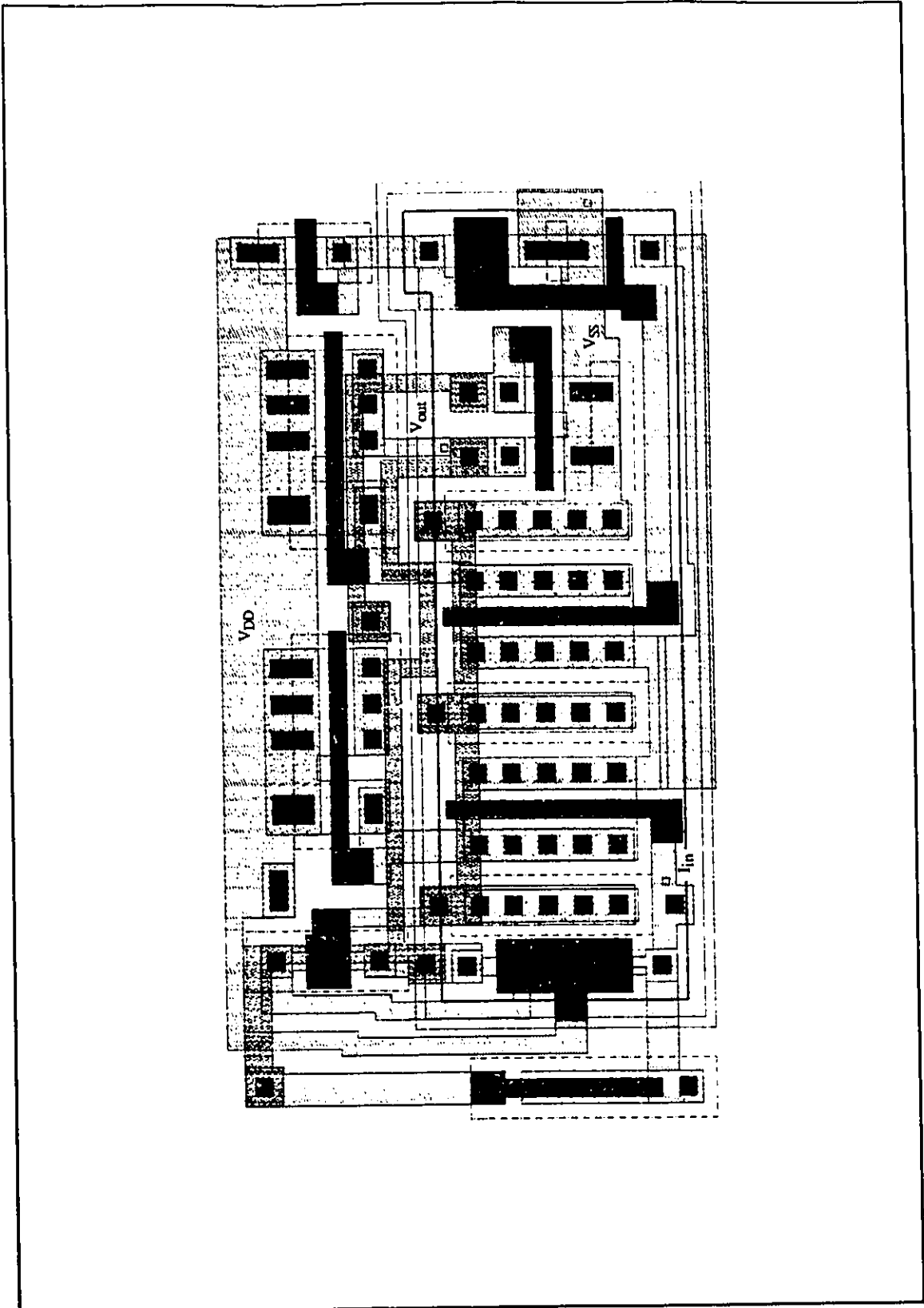


Figure B.7: Trans-impedance amplifier

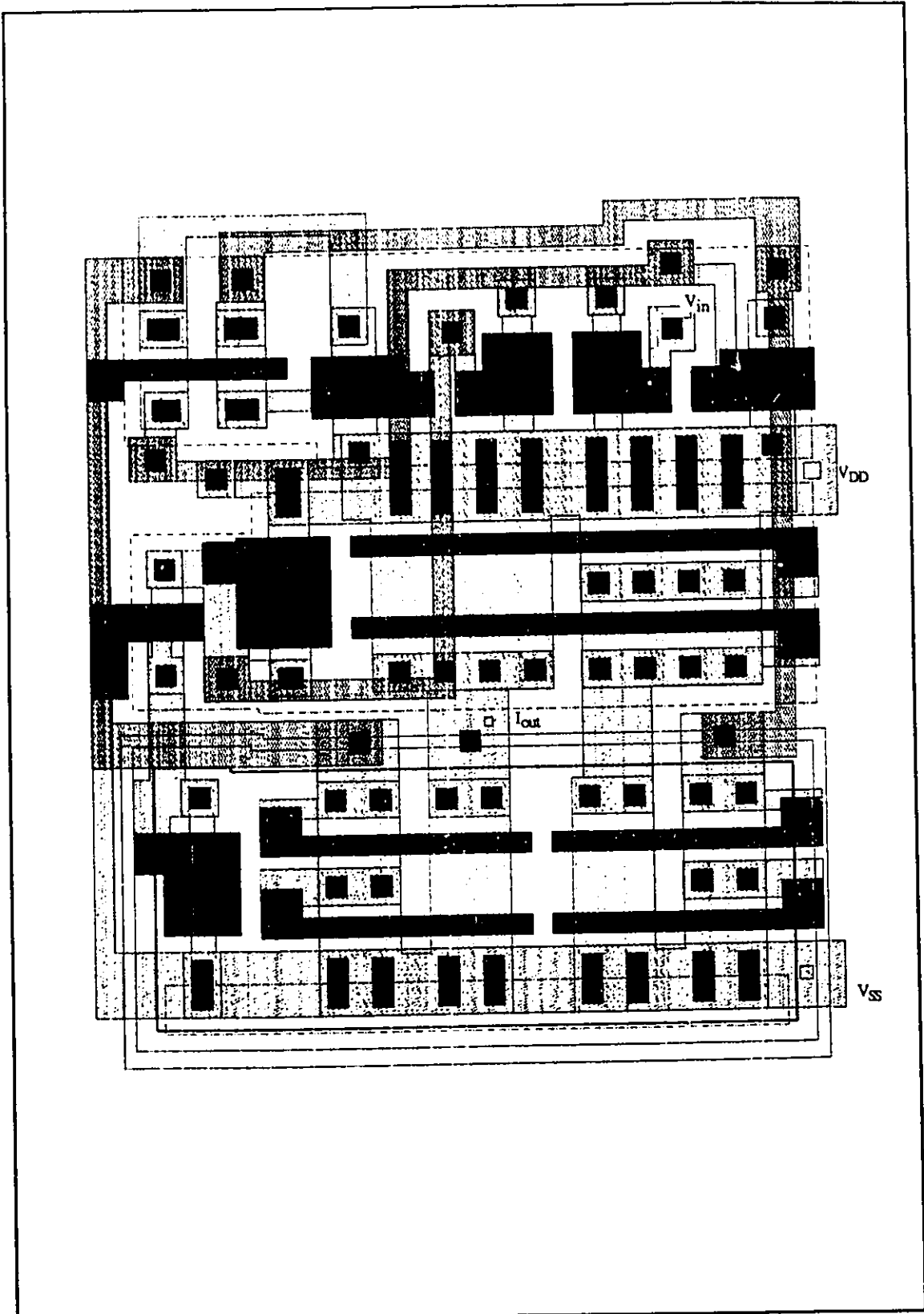


Figure B.8: Voltage to current converter

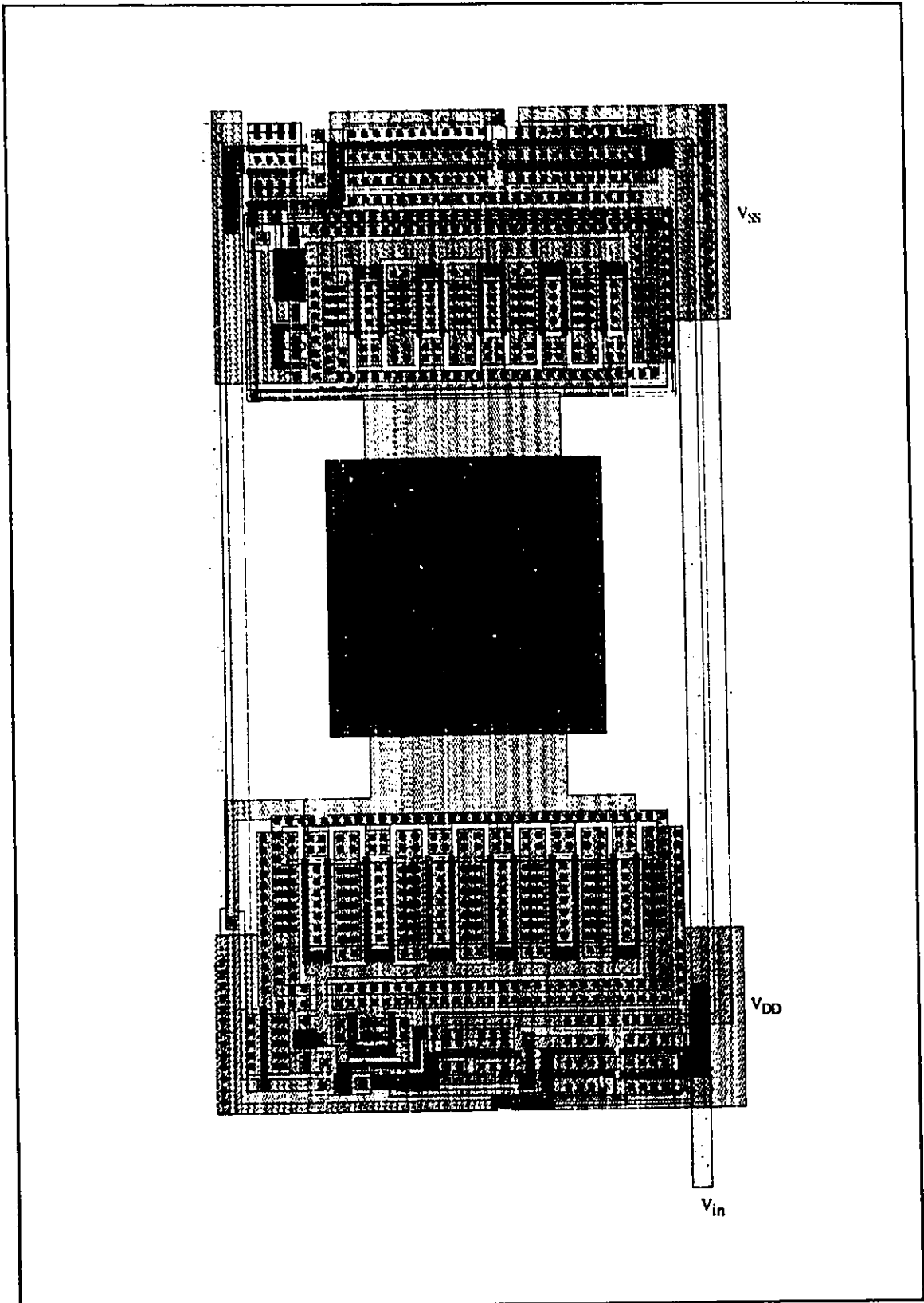


Figure B.9: Output Buffer for high capacitive loads

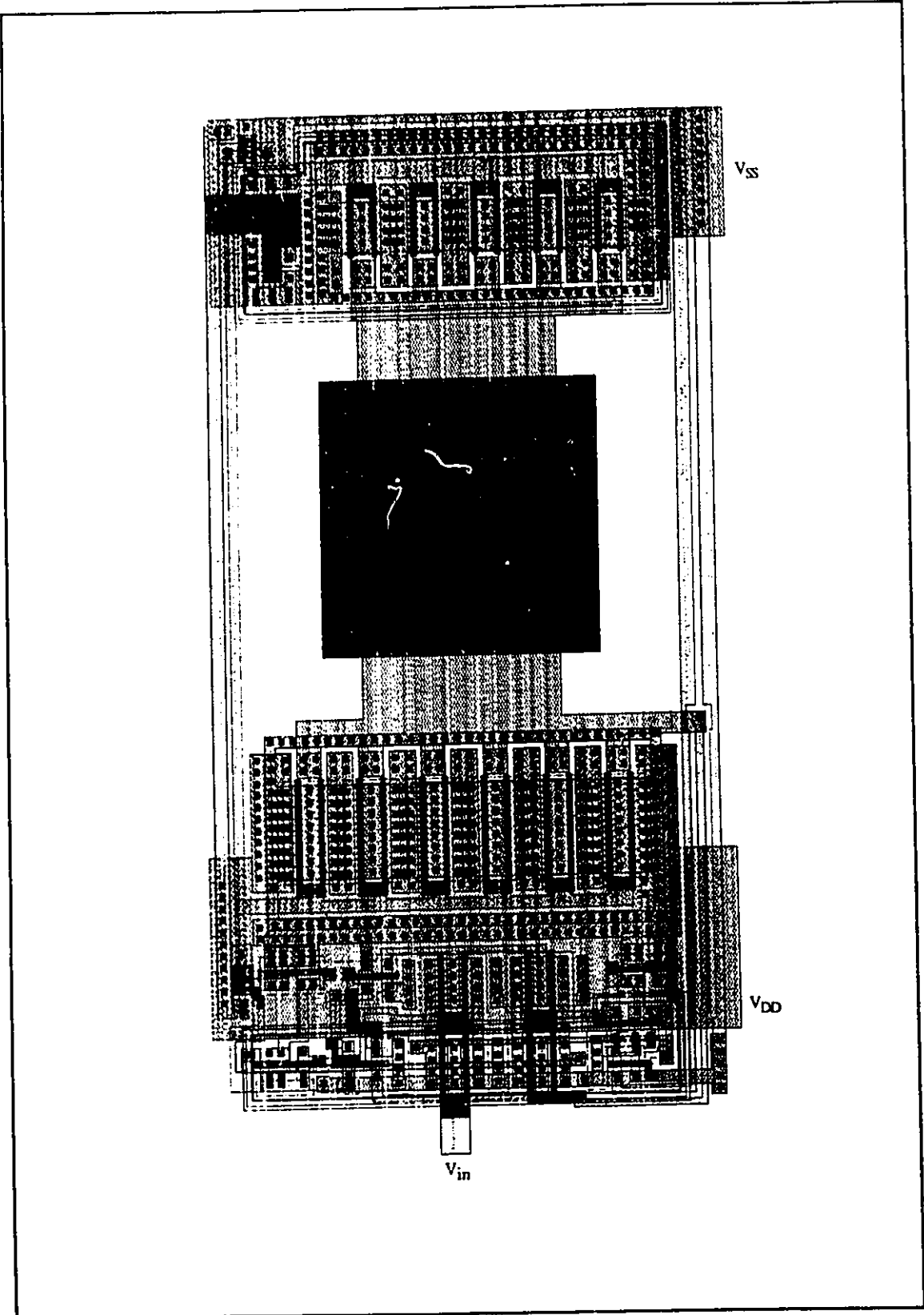


Figure B.10: Output buffer for high capacitive and resistive loads

---

## Hardware and Software Tools

---

A variety of design and verification tools and testing equipment were used throughout this work. The following is a list of these tools and equipment.

The greater part of the physical (geometric) layout of the cells were done on LOGICIAN systems. LOGICIAN is an integrated hardware and software system specifically developed to increase the productivity of VLSI design by Daisy Systems Corp. It has a proprietary operating system, DAISY-DNIX, which is a derivative of the standard Berkeley 4.2 UNIX. It also has a series of upper case commands compatible with DOS. Although this system has circuit and logic design, schematic entry, design verification, simulation, and test vector generation capabilities, the only part used was CHIPMASTER, the software for physical layout of integrated circuits.

Design verification and analysis was done on two VAX machines; a VAX 11/750 and a VAX 11/785. The software used was MASKAP II, by Phoenix Data Systems Inc. This



software was used for capture, analysis, debugging and verification of IC designs. The advantage of this software to comparable features in the Daisy workstation was mainly speed. After layouts were completed on the Daisy, they were converted to standard CIF files and transferred to the VAX machines, where they were used to create a MASKAP database. The errors were displayed graphically on Tektronix 4110 graphics engines, connected to the VAX machines.

MASKAP also includes a netlist extractor that can produce SPICE compatible netlists. After adding the model definition cards and analysis cards to the netlists, the completed SPICE decks were run on SPICE2G.6 on the same VAX machines.

The systems mentioned above were phased out in 1991 and the VLSI Research group at the University of Windsor moved to CADENCE integrated design and verification environment, by CADENCE Inc. (formerly SDA by SDA Inc.). This system was first installed on a MicroVAX and Sun 4/150, to which SparcStation 1 and SparcStation 2 machines were later added. The main advantage of this system was that all the tools needed for design and verification are built into one environment, so that no tedious data conversions are necessary as design information is moved from one tool to another. Also, interactive design rule checking (DRC) reduces design turn around time considerably.

On the Sun workstations, circuit simulation was done with HSPICE. HSPICE is an optimizing analog circuit simulator by Meta-Software Inc. which is input-compatible with SPICE, but has some added features. The main advantage of this simulator with respect to others used previously is higher speed and better convergence. Simulations of chapter 2 on circuits with more than 2700 circuit elements were made possible by HSPICE.

System level simulations of neural networks were performed using the software package included in [82] and also NeuralWorks by HNC Inc.

The following equipments were used in the testing of the building blocks:

1. ASIX-2 automatic IC tester by ASIX Systems Corp.
2. HP 4145B Semiconductor Parameter Analyzer
3. Tektronix 11402A Digitizing Oscilloscope

The digitizing oscilloscope was used mainly for obtaining the transient response of the buffers. A microscope and probing devices were used when testing the cells with the semiconductor analyzer. Both the oscilloscope and the semiconductor analyzer were connected to an IBM PC through GPIB buses and the data obtained in testing was directly translated into ASCII files. This information, after being changed into proper format by custom programs, appears in the figures of chapters 3 and 4\*.

---

\* LOGICIAN, DAISY, DNIX, and CHIPMASTER are trademarks of Daisy Systems Inc. VAX and MicroVAX are trademarks of Digital Equipment Corporation. MASKAP and MASKAP II are trademarks of Phoenix Data Systems Inc. SDA and CADANCE are trademarks of CADANCE Inc. Meta-Software and HSPICE are trademarks of Meta-Software Inc. NeuralWorks is a trademark of HNC Inc. ASIX and ASIX-2 are trademarks of ASIX Systems Corp. IBM is a trademark of International Business Machines Corp.

---

## References

---

- [1] J.L. McClelland and D.E. Rumelhart, "An interactive activation of context effects in letter perception: Part I. An account of basic findings," *Psychological Review*, Vol. 88, pp.375-407, 1981.
- [2] D.E. Rumelhart and J.L. McClelland, "An interactive activation of context effects in letter perception: Part II. The contextual enhancement effect and some tests and extensions of the model," *Psychological Review*, Vol. 89, pp.75-112, 1982.
- [3] D.E. Rumelhart and J.L. McClelland, *Parallel Distributed Processing Explorations in the Microstructure of Cognition Vol. 1: Foundations*. Cambridge, Massachusetts: MIT Press, 1986.
- [4] R.M. Warren, "Perceptual restoration of missing speech sounds," *Science*, Vol.167, No. 3917, pp.392-393, 23 January 1970.
- [5] W. James, *Psychology (Briefer Course)*, New York: Holt, Chapter XIV "Associations," 1890, pp.253-279.
- [6] W.S. McCulloch and W. Pitts, "A logical calculus of ideas immanent in nervous activity," *Bulletin of Mathematical Biophysics*, Vol.5, pp.115-133, 1943.

- [7] J. von Neumann, *The Computer and the Brain*. New Haven: Yale University Press, 1958, pp.66-82.
- [8] J. von Neumann, "First draft of a report on the EDVAC," in *The Origins of the Digital Computers: Selected papers*. B. Randall (Editor), Berlin: Springer Verlag, 1945/1982.
- [9] D.O. Hebb, *The Organization of Behavior*. New York: Wiley, 1949.
- [10] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain," *Psychological Review*, Vol. 65, pp.386-408, 1958.
- [11] B. Widrow and M.E. Hoff, "Adaptive switching circuits," 1960 IRE WESCON Convention Record, IRE, New York, pp.96-104.
- [12] M. Minsky and S. Papert, *Perceptrons*. Cambridge, Massachusetts: MIT Press, 1969.
- [13] T. Kohonen, "Correlation matrix memories," *IEEE Transactions on Computers*, Vol. C-21, pp.353-359, 1972.
- [14] S. Grossberg, "Adaptive pattern classification and universal recoding: I. Parallel development and coding of neural feature detectors," *Biological Cybernetics*, Vol. 23, No.4, pp.121-134, July 1976.
- [15] J.J. Hopfield, "Neural Networks and physical systems with emergent collective computational abilities," *Proceedings of the National Academy of Science*, Vol.79, pp. 2554-2558, April 1982.
- [16] J.J. Hopfield, "Neurons with graded response have collective computational properties like those of two-state neurons," *Proceedings of the National Academy of Science*, Vol.81, pp.3088-3092, May 1984.
- [17] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological Cybernetics*, Vol. 43, pp.59-69, 1982.

- [18] T. Kohonen, "Adaptive, associative, and self-organizing functions in neural computing," *Applied Optics*, Vol. 26, No. 23, pp.4910-4918 December 1987.
- [19] T. Kohonen, *Self Organization and Associative Memory*. Berlin: Springer Verlag, 1984.
- [20] K. Fukushima, "Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological Cybernetics*, Vol. 36, pp.193-202 April 1980.
- [21] K. Fukushima, S. Miyake, and T. Ito, "Neocognitron: a neural network model for a mechanism of visual pattern recognition," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-13, No. 5, pp.826-834, Sept./Oct.1983.
- [22] K. Fukushima, "Neocognitron: a hierarchical neural network capable of visual pattern recognition," *Neural Networks*, Vol.1, No.2, pp.109-130, 1988.
- [23] K. Fukushima, "A neural network for visual pattern recognition," *IEEE Computer Magazine*, pp.65-75, March 1988.
- [24] D.H. Ackley, G.E. Hinton, and T.J. Sejnowsky, "A learning algorithm for Boltzmann machines," *Cognitive Science*, Vol.9, pp.147-169, 1985.
- [25] D.E. Rumelhart, G.E. Hinton, and R.J. Williams, "Learning representations by back-propagating errors," *Nature*, Vol.323, No. 6088, pp.533-536, 9 October 1986.
- [26] Y. Le Cun, "Learning processes in an asymmetric threshold network," *Disordered Systems and Biological Organization*, E. Bienenstock, F. Fogelman Souli, and G. Weisbuch (Editors), Berlin: Springer Verlag, 1986.
- [27] D. Parker, "Learning Logic," Technical report TR-87, Center for Computational Research in Economics and Management Sciences, MIT, Cambridge, MA, 1985.
- [28] P.J. Werbos, "Beyond regression: new tools for prediction and analysis in the behavioral sciences," Ph.D. thesis, Harvard University, Cambridge, MA, 1974.

- [29] G.E. Hinton, J.L. McClelland, and D.E. Rumelhart, "Distributed representations," in *Parallel Distributed Processing Explorations in the Microstructure of Cognition Vol. 1: Foundations*. J.L. McClelland and D.E. Rumelhart (Editors), Cambridge, Massachusetts: MIT Press, 1986.
- [30] J.A. Feldman, "Connectionist models and their applications: Introduction," *Cognitive Science*, Vol.9, pp.1-2, 1985.
- [31] B.G. Farley and W.A. Clark, "Simulation of self-organizing systems by digital computer," *IRE Transactions on Information Theory*, Vol. 4, pp.76-84, 1954.
- [32] T.J. Sejnowski and C.R. Rosenberg, "NETtalk: a parallel network that learns to read aloud," The Johns Hopkins University Electrical Engineering and Computer Science Technical Report JUH/EECS-86/01, 1986.
- [33] J. Bernstein, "profiles: AI, Marvin Minsky," *The New Yorker*, pp.50-126, 14 December 1981.
- [34] H.D. Block, "The Perceptron: a model for brain functioning. I," *Reviews of Modern Physics*, Vol. 34, pp.123-135, 1962.
- [35] N.H. Farhat, D. Psaltis, A. Prata and E. Paek, "Optical implementation of the Hopfield model," *Applied Optics*, Vol. 24, No. 10, pp.1469-1475, 15 May 1985.
- [36] N. Farhat and D. Psaltis, "Optoelectronic analogs of self-programming neural nets," *Applied Optics*, Vol.26, No. 23, pp.5093-5103, 1 December 1987.
- [37] N. Farhat, "Optoelectronic neural networks and learning machines," *IEEE Circuits and Devices Magazine*, Vol.5, No.5, pp.32-41, September 1989.
- [38] L.E. Atlas and Y. Suzuki, "Digital systems for artificial neural networks," *IEEE Circuits and Devices Magazine*, Vol. 5, No.6, pp.20-24, November 1989.
- [39] A. Masaki, Y. Hirai and M. Yamada, "Neural networks in CMOS: a case study," *IEEE Circuits and Devices Magazine*, Vol. 6, No.4, pp.13-17, July 1990.

- [40] M. Yasunaga, et al., "A wafer scale integration neural network utilizing completely digital circuits," Proc IJCNN 1989, pp.II/213-217, 1989.
- [41] A.F. Murray and A.V.W. Smith, "Asynchronous VLSI neural networks using pulse stream arithmetic," *IEEE Journal of Solid State Circuits*, Vol.23, No.3, pp-688-697, June 1988.
- [42] M.A. Sivilotti, M.A. Mahowald, and C.A. Mead, "Real-time visual computations using analog CMOS processing arrays," *Advanced Research in VLSI: proceedings of the 1987 Stanford Conference*. P.Losleben (Editor), Cambridge, MA: MIT Press, pp.295-312, 1987.
- [43] C.A. Mead, "Adaptive retina," in *Analog VLSI Implementation of Neural Systems*. C. Mead and M. Ismail (Editors), Dordrecht: Kluwer, pp.239-246, 1989.
- [44] J. Lazzaro and C.A. Mead, "Circuit models of sensory transduction in the cochlea," in *Analog VLSI Implementation of Neural Systems*. C. Mead and M. Ismail (Editors), Dordrecht: Kluwer, pp.85-102, 1989.
- [45] J. Van Der Spiegel, et al., "A foviated retina-like sensor using CCD technology," in *Analog VLSI Implementation of Neural Systems*. C. Mead and M. Ismail (Editors), Dordrecht: Kluwer, pp.189-212, 1989.
- [46] S. Eberhardt, A. Moopenn and A. Thakoor, "Considerations for Hardware Implementations of Neural Networks", *Proceedings of the 22nd ASILOMAR annual conference on signals, systems and computers* , pp. 649-653, 1988.
- [47] S. Bibyk and M. Ismail, "Issues in analog VLSI and MOS techniques for neural computing," in *Analog VLSI Implementation of Neural Systems*. C. Mead and M. Ismail (Editors), Dordrecht: Kluwer, pp.103-134,1989.
- [48] T.H. Borgstrom, M. Ismail and S.B. Bibyk, "Programmable current mode network for implementation in analogue VLSI," *IEE Proceedings*, part G, Vol. 137, No. 2, pp.75-84, April 1990.

- [49] V. Hu, A. Kramer, and P.K. Ko, "EEPROMS as analog storage devices for neural nets," *Neural Networks*, Suppl., Vol.1, p.387, 1988.
- [50] D.B. Schwartz, R.E. Howard and W.E. Hubbard, "A programmable analog neural network chip," *IEEE Journal of Solid-State Circuits*, Vol. SC-24, No. 2, pp. 313-319, April 1989.
- [51] B. Furman and A.A. Abidi, "An analog CMOS backward error-propagation LSI," *Proceedings of the 22nd ASILOMAR annual conference on signals, systems and computers*, pp.645-648, 1988.
- [52] B. Hochet, V. Peiris, S. Abdo, and M.J. Declercq, "Implementation of a learning Kohonen neuron based on a new multilevel storage technique," *IEEE Journal of Solid-State Circuits*, Vol. 26, No. 3, pp. 262-267, March 1989.
- [53] F. Kimura, Z. Chan , and M. Shridhar, " An integrated character recognition algorithm for locating and recognizing ZIP codes," *Proc. of the 4th advanced Technology Conf.*, Washington D.C., August 1990.
- [54] D.E. Rumelhart, G.E. Hinton, and R.J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing Explorations in the Microstructure of Cognition Vol. 1: Foundations*. J.L. McClelland and D.E. Rumelhart (Editors), Cambridge, Massachusetts: MIT Press, 1986.
- [55] Y. Le Cun, et al., "Constrained neural network for unconstrained handwritten character recognition," in *Frontiers in Handwriting Recognition*. C.Y. Suen (Editor), Montreal: CENPARMI, Concordia University, pp.145-151, 1990.
- [56] P.C. Dodwell, *Visual Pattern Recognition*. New York: Holt, Rinehart and Winston, Chapter 3:Neurophysiological evidence on contour coding, 1970.
- [57] P. Mueller, et al., "A general purpose analog neural computer," in *IEEE Second International Conference on Neural Networks*, vol II, pp. 177-182, 1988.



- [58] H.P. Graf and L.D. Jackel, "Analog Electronic Neural network Circuits," *IEEE Circuits and Devices Magazine*, Vol. 5, No. 4, pp. 44-49, July 1989.
- [59] E.A. Vittoz, "Analog VLSI Implementation of Neural Network," *Proc. of the IEEE International Symposium on Circuits and Systems (ISCAS)*, pp.2524-2527, 1990.
- [60] F.M.A. Salam and M.R. Choi, "An All-MOS Analog Feedforward Neural Circuit with Learning," *Proc. of the IEEE International Symposium on Circuits and Systems (ISCAS)*, pp.2508-2511, 1990.
- [61] J. Raffel, et al., "A Generic Architecture for Wafer-Scale Neuromorphic Systems," in *Proc. IEEE 1st Int. Conference on Neural Networks*, 1987, Vol. III, p. 501.
- [62] P.E. Allen and D.R. Holberg, *CMOS Analog Circuit Design*. New York: Holt, Rinehart and Winston, 1987.
- [63] P.R. Gray and R.G. Meyer, "MOS operational amplifier design- a tutorial overview," *IEEE Journal of Solid-State Circuits*, Vol. SC-17, No. 6, pp. 969-982, Dec 1982.
- [64] A. Nosratinia, "A small high-power buffer with high efficiency," Internal Research Group Report, University of Windsor.
- [65] K.E. Brehmer and J.B. Wieser, "Large swing CMOS power amplifier," *IEEE Journal of Solid-State Circuits*, Vol. SC-18, No. 6, pp. 624-629, Dec 1983.
- [66] M. Steyaert and W. Sansen, "A high dynamic range CMOS Op Amp with low-distortion output structure," *IEEE Journal of Solid-State Circuits*, Vol. SC-22, No. 6, pp. 1204-1207, Dec 1987.
- [67] J.A. Fisher and R. Koch, "A highly linear CMOS buffer amplifier," *IEEE Journal of Solid-State Circuits*, Vol. SC-22, No. 3, pp. 330-334, June 1987.
- [68] J.N. Babanezhad, "A rail-to-rail CMOS Op Amp," *IEEE Journal of Solid-State Circuits*, Vol. SC-23, No. 6, pp. 1414-1417, Dec 1988.

- [69] K. Nagaraj, "Large-swing CMOS buffer amplifier," *IEEE Journal of Solid-State Circuits*, Vol. 24, No. 1, pp. 181-183, Feb 1989.
- [70] S.L. Wong and C.A. Salama, "An efficient CMOS buffer for driving large capacitive loads," *IEEE Journal of Solid-State Circuits*, Vol. SC-21, No. 3, pp. 464-469, June 1986.
- [71] F.N.L.Op't Eynde, et al., "A CMOS large-swing low-distortion three-stage class AB power amplifier," *IEEE Journal of Solid-State Circuits*, Vol. 25, No. 1, pp.265-273, Feb. 1990.
- [72] M.D. Paroden and M.G. Degrauwe, "A rail-to-rail input/output CMOS amplifier," *IEEE Journal of Solid-State Circuits*, Vol. 25, No. 2, pp.501-504, April 1990.
- [73] L.G.A. Callewart and W.M.C. Sansen, "Class AB CMOS amplifiers with high efficiency," *IEEE Journal of Solid-State Circuits*, Vol. 25, No. 3., pp. 684-691 June 1990.
- [74] A. Nosratinia, M. Ahmadi, G.A. Jullien and M. Shridhar, " A high-drive CMOS buffer for high capacitive loads," *Proc. of the IEEE Conf. on Circuits and Systems*, June 1991, Shenzhen, China.
- [75] A. Nosratinia, M. Ahmadi, G.A. Jullien and M. Shridhar, " A high-drive CMOS buffer for high capacitive loads," *Electronics Letters*, vol.26, No.12, pp.1044-1046, June 1991.
- [76] A. Nosratinia, M. Ahmadi, G.A. Jullien and M. Shridhar, "A high-drive, High-swing buffer amplifier," *Proc. of the IEEE Symp. on Circuits and Systems*, June 1991, Singapore, pp.2220-2223.
- [77] DARPA Neural Network Study, Lincoln Laboratory, MIT, 1988.
- [78] A.N. Kolmogorov, "On representation of continuous functions of many variables by superposition of continuous functions of one variable and addition," *Dokl. Akad. Nauk. USSR*, Vol. 114, pp. 953-956, 1957.

- [79] S.W. Kuffler and J.G. Nichols, *From Neuron to Brain*, Sutherland, MA: Sinauer Associates, 1976.
- [80] G.M. Shepherd, *Neurobiology*, New York: Oxford University Press, 1988.
- [81] T.C. Ruch, H.D. Patton, J.W. Woodbury, and A.L. Towe (Editors), *Neurophysiology*, Philadelphia: W.B. Saunders, 1969.
- [82] J.L. McClelland and D.E. Rumelhart, *Explorations in Parallel Distributed Processing: A Handbook of Models, Programs and Exercises*, Cambridge, Massachusetts: MIT Press, 1988.

## VITA AUCTORIS

Aria Nosratinia was born in Tehran, Iran in 1965. He graduated from Alborz highschool in 1983. He obtained his B.S. in Electrical Engineering (Electronics) from Tehran University in 1988. From 1984 to 1988, he had a part-time job in the area of maintenance of industrial controllers. From 1988 to 1989 he worked on the design of a numerical controller of cutting tools. In 1989, he joined the graduate program at the University of Windsor and obtained his M.A.Sc. degree in Electrical Engineering in August 1991.