2-1-1994

A Hardware approach to neural networks silicon retina

Arif K. Golwalla

A Hardware Approach to Neural Networks Silicon Retina

by

Arif K. Golwalla

A Thesis Submitted
in
Partial Fulfillment of the
Requirements for the degree of
MASTER OF SCIENCE
in
Computer Engineering

Approved by: ____________________
Graduate Advisor - Prof. George A. Brown

____
Department Head - Dr. Roy Czernikowski

____
Committee Member - Dr. P. R. Mukund

DEPARTMENT OF COMPUTER ENGINEERING
COLLEGE OF ENGINEERING
ROCHESTER INSTITUTE OF TECHNOLOGY
ROCHESTER, NEW YORK
FEBRUARY 1994

THESIS RELEASE PERMISSION FORM

ROCHESTER INSTITUTE OF TECHNOLOGY
COLLEGE OF ENGINEERING

Title of Thesis: A Hardware Approach to Neural Networks - Silicon Retina.

I, Arif K. Golwalla, hereby grant permission to the Wallace Memorial Library of RIT to reproduce my thesis in whole or in part.

Date: 03/02/94

ABSTRACT

The primary goal of this thesis was to emulate the function of the biological eye in silicon. In both neural and silicon technologies, the active devices occupy approximately 2 percent of the space, *wire* fills the entire remaining space.The silicon retina was modeled on the distal portion of the vertebrate retina. This chip generates, in real time, outputs that correspond directly to signals observed in the corresponding levels of the biological retinas. The design uses the principles of signal aggregation. It demonstrates a tolerance for device imperfection that is characteristic of a collective system. The digital computer is extremely effective at producing precise answers to well-defined questions. The nervous system accepts fuzzy, poorly conditioned input, performs a computation that is ill-defined, and produces approximate output.

Table of Contents

Table of Figures

GLOSSARY

VLSI	Very Large Scale Integrated Circuit
IC	Integrated Circuit
RC	Resistor-Capacitor
NN	Neural Network
ANN	Artificial Neural Network
CMOS	Complementary Metal-Oxide Semiconductor
NMOS	N-Type Metal-Oxide Semiconductor
VHDL	VHSIC Hardware Description Language
VHSIC	Very High Speed Integrated Circuit
TTL	Transistor Transistor Logic

1.0 INTRODUCTION

Living Systems are made from three-dimensional soft cells, however computers are constructed of rigid inorganic matter in flat, two-dimensional sheets. Living systems are powered by metabolic bio-chemistry, while computers are powered by electrical energy from the power mains. The destruction of a few percent of the cells in a brain will cause no discernible degradation in performance, but the loss of even a single transistor may cause complete loss of functionality.

A close look reveals some degree of similarities between the two kind of systems. They both *process information*. Signals are represented as differences in electrical potential, and are conveyed on *wires* formed by surrounding a conducting path with an excellent electrical insulator. Active devices cause electrical current to flow in a second *output* conductor due to potential in a first *input* conductor. A *power supply* maintains a near-constant average difference in electrochemical potential across the active devices. The active devices are formed of extremely thin *energy barriers* that prevent the flow of current between two electrical nodes. The passage of current is mediated by the potential on a third control electrical node. That current varies exponentially with the potential on the control node. Neurons perform Boolean AND and OR operations on the way to firing off a nerve pulse to the next stage of computation.

The visual system of a single human being does more image processing than do the entire world's supply of supercomputers. The digital computer is extremely effective at producing precise answers to well-defined questions. The nervous system accepts fuzzy, poorly conditioned input, performs a computation that is ill-defined, and produces approximate output.

Perhaps the most important aspect of analog computation is the extent to which the elementary computational primitives are a direct consequence of fundamental laws of physics. We will see that a single transistor can take at its gate a voltage type signal and produces at its drain a current type signal that is exponential in the input voltage. This exponential function is a direct result of the Boltzmann distribution. We will see that the addition and subtraction of currents follows directly from the conservation of charge.

The complexity of a computational system derives not from the complexity of its component parts, but rather from the multitude of ways in which a large collection of these components can interact. Even if we understand in elaborate detail the operation of every nerve channel and every synapse, we will not by so doing have understood the neural computation as a system. It is not the neural devices themselves that contain the secret of thought. It is rather, the organizing principles by which vast numbers of these elementary devices work together.

Two barriers have historically blocked the way from creating a nervous system in silicon,

(1) Neural systems have far greater connectivity than has been possible in standard computer hardware. Many early attempts to create neural systems failed simply because no workable technology existed for realizing systems of the requisite complexity.

(2) Sufficient knowledge of the organizing principles involved in neural systems was not available.

The rapidly developing technology of very large scale integrated circuits has given us a medium in which it is possible to fabricate tens of millions of devices interconnected on a single silicon wafer. In terms of discovering neural organizing principles, we are less well off. Although a great deal of progress has been made in recent years, there is still no global view of the principles and representations on which the nervous system is organized. Many hypothesis have been proposed about the way computation is performed in these systems. To date, it has proved difficult if not possible either to verify or to disprove any given hypothesis concerning the operating principles of even the simplest neural system.

A new approach is adopted, in the sense that we know all the elementary operations found in the nervous system can be realized in silicon. Also neural areas are thin sheets, and carry two-dimensional representations of their computational space. The retina is the most obvious example of this organization. In both neural and silicon technologies, the active devices occupy no more than 1 to 2 percent of the space, *wire* fills the entire remaining space. Thus the limitation of connectivity will force the solution into a particular form.

The constraints on our analog silicon systems are similar to those on neural systems; wire is limited, power is precious and robustness and reliability are essential. We shall in later chapters describe the relevant aspects of neural netware at the level of abstraction where we will be working. We will then develop the operations that are natural to silicon, and examine how they can be used to implement certain known neural functions.

It is a general belief that the ability to realize simple neural functions is strictly limited by our understanding of their organizing principles, and not by difficulties in implementation. If we really understand a system, we will be able to build it. Conversely, we can be sure that we do not fully understand a system until we have synthesized and demonstrated a working model. The success of this venture will create a bridge between neurobiology and the information sciences, and will bring us a much deeper view of computation as a physical process.

2.0 INTRODUCTION TO NEURAL NETWORKS

Neural networks provide a unique computing architecture whose potential has only begun to be tapped. Used to address problems that are intractable or cumbersome with traditional methods, these new computing architectures inspired by the structure of the brain, are radically different from the computers that are widely used today. Neural networks are massively parallel systems that rely on dense arrangements of interconnections and surprisingly simple processors.

Artificial neural networks attempt to model the networks of nerve cells in the brain. Although a great deal of biological detail is eliminated in these computing models, the artificial neural networks retain enough of the structure observed in the brain to provide insight into how biological neural processing may work.

Neural networks provide an effective approach for a broad spectrum of applications. They excel at problems involving patterns: pattern mapping, pattern completion, and pattern classification. Neural networks may be applied to translate images into keywords, to translate financial data into financial predictions, or to map visual images to robotic commands. Noisy patterns, those with segments missing, may be completed with a neural network that has been trained to recall the completed patterns. For example, a neural network might receive an input of the outline of a vehicle that has been partially obscured, and produce an outline of the complete vehicle.

Possible applications for pattern classification abound: Visual images need to be classified during industrial inspections; medical images, such as magnified blood cells, need to be classified for diagnostic tests; sonar images may be input to a neural network for classification; speech recognition requires classification and identification of words and sequences of words. Even diagnostic problems, where results of tests and answers to questions are classified into appropriate diagnosis, are promising areas for neural networks. The process of building a successful neural network application is complex, but the range of possible applications is impressively broad.

Neural networks utilize a parallel processing structure that has a large number of processors and many interconnections between them. These processors are much simpler than typical central processing units (CPU's). In a neural network each processor is linked to many of its neighbors (typically hundreds or thousands) so that there are many more interconnects than processors. The power of the neural network lies in the tremendous number of interconnections.

2.1 TRADITIONAL VERSUS NEURAL NETWORK ARCHITECTURE

Neural network architectures are strikingly different from traditional single processor computers. Traditionally Von Neumann machines have a single CPU that performs all of its computations in sequence. A typical CPU is capable of a hundred or more basic commands, including adds, subtracts, loads, and shifts, among others. The commands are executed one at a time, at successive steps of a time clock. In contrast, a Neural Network (NN) processing unit may do only one or, at most, a few calculations. A summation function is performed on its inputs and incremental changes are made to parameters associated with interconnections. This simple structure nevertheless provides a NN with the capability to classify and recognize patterns, to perform pattern mapping, and to be useful as a computing tool.

The processing power of a NN is measured mainly by the number of interconnection updates per second; in contrast, Von Neumann machines are benchmarked by the number of sequential instructions that are performed per second by a single processor. Neural Networks, during their learning phase, adjust parameters associated with the interconnections between neurons. Thus, the rate of learning is dependent on the rate of interconnection updates.

Neural Network architectures depart from typical parallel processing architectures in some basic respects. First, the processors in a NN are massively interconnected. As a result, there are more interconnections than there are processing units. State of the art parallel processing architectures typically have a smaller ratio of interconnections to processing units. In addition, parallel processing architectures tend to incorporate processing units that are comparable in complexity to those of Von Neumann machines. Neural Network architectures depart from this organization scheme by containing simpler processing units, which are designed for summation of many inputs and adjustment of interconnection parameters.

2.2 BIOLOGICAL NEURAL SYSTEMS - THE ORIGINAL NEURAL NET

Neural Network architectures are motivated by models of our own brains and nerve cells. Although our current knowledge of the brain is limited, we do have much detailed anatomical and physiological information. The basic anatomy of an individual nerve cell or neuron is known, and the most important biochemical reactions that govern its activities have been identified.

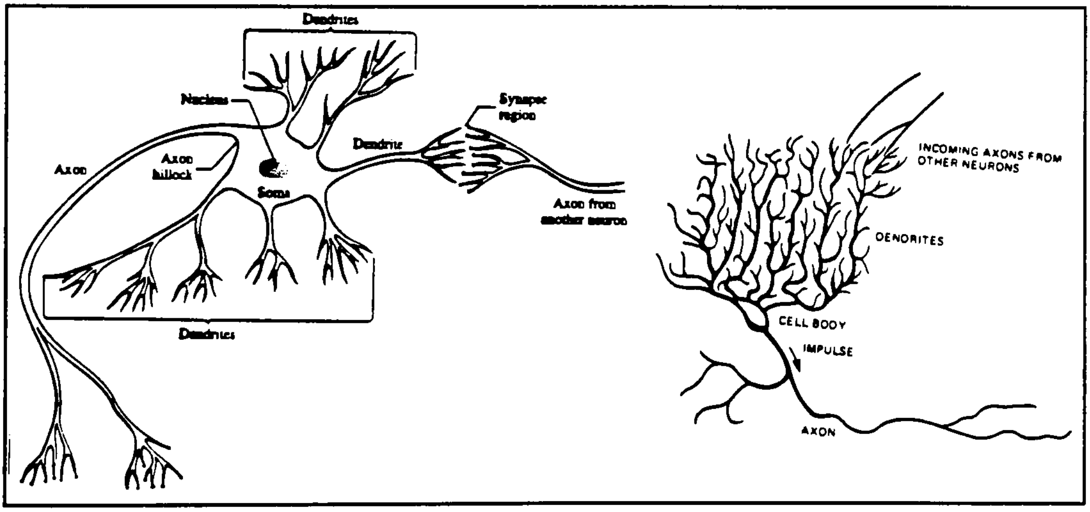

Figure 2-1: Schematic drawing of a biological nerve cell.[13]

The processes happening inside biological neurons are not known in detail. Nevertheless, there are a number of ways in which the electronic model of the neuron approximates the behavior of neural cells. As shown in figure 2-1, a living neuron receives multiple inputs from other neurons via branching input paths called dendrites. The combined stimuli from these input signals activate a region called an axon hillock, where an outgoing tendril called an axon connects to the cell body. The axon then transmits the neuron's output to still other neurons through their dendrite. Or, in some cases, the output that the neuron transmits along its axon goes directly to muscle or gland cells in order to activate or inhibit the functions that those cells perform.

The gap between an axon of one neuron and the input dendrites of another is the location of the synapses. Information transfer across a synapse is controlled by biochemical agents, a process that is modeled in electronic neurons by the changing of synaptic weights.

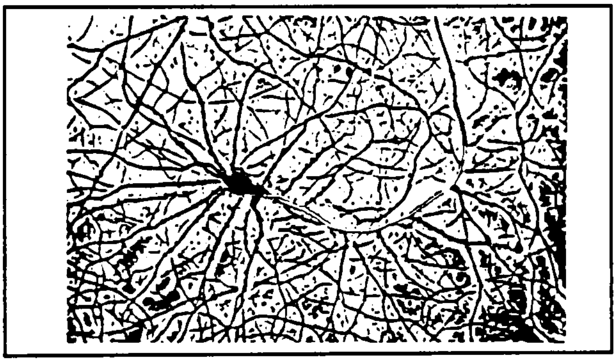

Figure 2-2: A biological neuron magnified 400X with the dendritic tree in the foreground.[13]

Apart from their function in receiving and transmitting nerve impulses, neurons are more or less like other cells of the body. Unlike other body cells, however, most neurons do not reproduce. Similarly, their metabolic functions are largely taken care of by attendant glial cells that transport nutriments and waste products to and from the neurons, regulate their chemical environment, and remove and digest the neurons when they are dead or damaged.

The neuron shown in figure 2-2, was photographed from a tissue culture of embryonic nerve cells. Although the axon is hidden, the dendritic tree is apparent. The many larger fibers in the foreground are dendritic branches; the smaller fibers that crisscross in thebackground are axons that synapse onto the dendrites, bringing incoming pulses from other neurons.

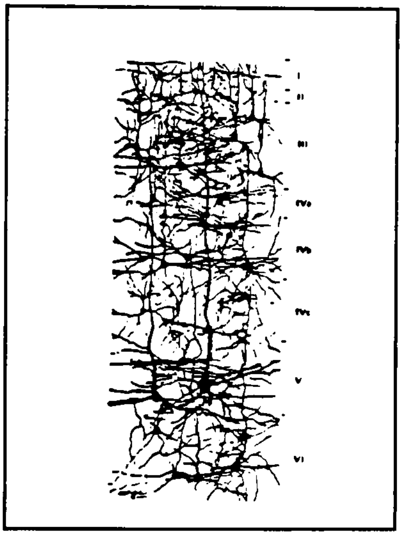

Figure 2-3: A golgi-stained preparation from the visual cortex of a two year old child [13]

Figure 2-3, shows a typical network of neurons, traced from the human visual cortex. These neurons appeared when a thin section of the cortex was impregnated with a Golgi stain, which is taken up by only 2% of the neurons. The resulting picture indicates the nature of the biological NN present, with densely placed neurons and myraid intersecting nerve branches. The actual biological network is much more dense than that shown in the figure because of the sparsity of cells that take up the Golgi stain. This picture exemplifies the vast interconnected arrays of neurons that appear in biological neural networks.

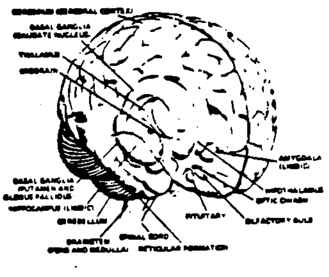

Figure 2-4: Major structures of the human brain. [14]

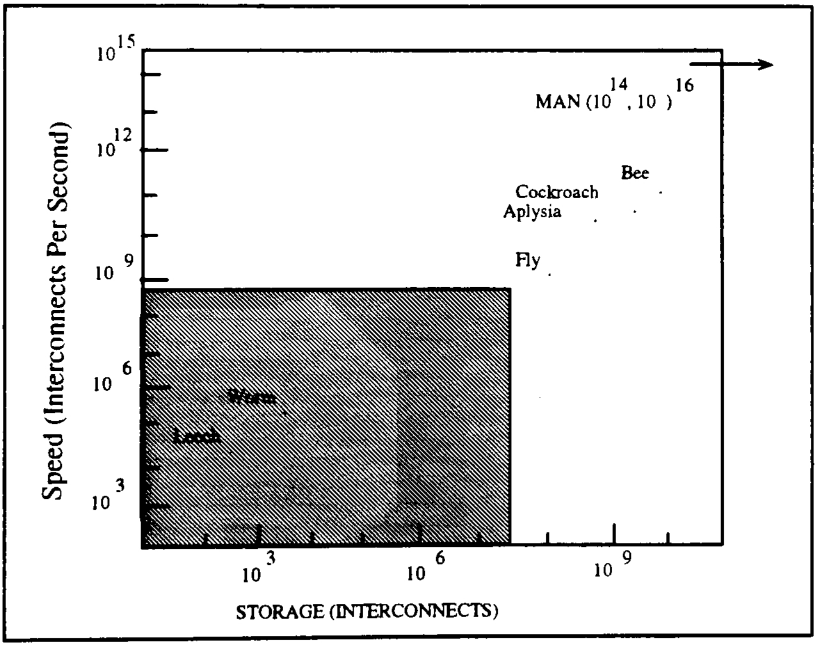

Figure 2-5: Speed versus storage for a variety of systems. [14]

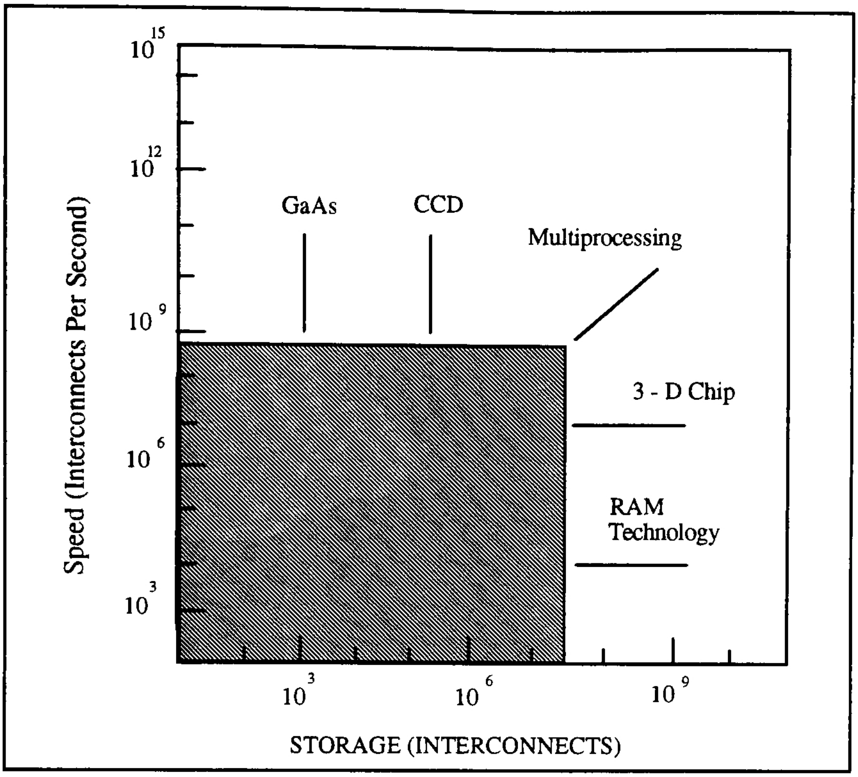

Figure 2-6 : Estimation of the Resources available to Neural Networks in 1980. [14]

Figure 2-4, depicts the human brain. The brain is a dense neural network in which the neurons are highly interconnected. The total number of neurons in the human brain is estimated at 100 billion. Each neuron is connected to perhaps 10,000 other cells, meaning such biological neuron can send impulses that may be received by as many as 10,000 targets cells.

Figure 2-5, and 2-6, shows a comparison of different biological nervous systems with artificial neural networks[14]. Speed, in term of interconnections processed per second, is plotted against storage, measured in terms of interconnections. The shaded area represents neural network sizes that are within the reach of today's artificial NN simulations. The leech and worm, relatively primitiva invertebrates, have nervous systems that appear within the range of existing simulators having fewer than 10^8 interconnections.

More complex organisms, such as the fly, bee, cockroach, and aplysa (a sea slug), have nervous systems with considerable more speed and storage capacity. They appear to exceed the computational capabilities presently available in simulations. The human nervous system is far larger than the other systems plotted, and would appear beyond the top right of the graph.

2.3 ARTIFICIAL NEURAL NETWORKS - THE BASIC STRUCTURE

The figure 2-7, depicts an example of a typical processing unit for an artificial neural network. On the left are the multiple inputs to the processing unit, each arriving from another unit, which is connected to the unit shown at the center. Each interconnection has an associated connection strength, given as w1, w2, . . ., w_n. The processing unit performs a weighted sum on the inputs and uses a nonlinear threshold function, f, to compute its output. The calculated result is sent along the output connections to the target cells shown at the right. The same output value is sent along all the output connections.

The neural network shown in figure 2-8, has three layers of processing units, It is the typical organization for the neural net paradigm known as back-error propagation. The first layer of input units assume the values of a pattern, represented as a vector, that is input to the network. The middle, "hidden", layer of this network consists of "feature detectors", units that respond to particular features that may appear in the input pattern. Sometimes there is more than one hidden layer. The last layer is the output layer. The activities of these units are read as the output of the network. In some applications, output units stand for different classification of patterns.

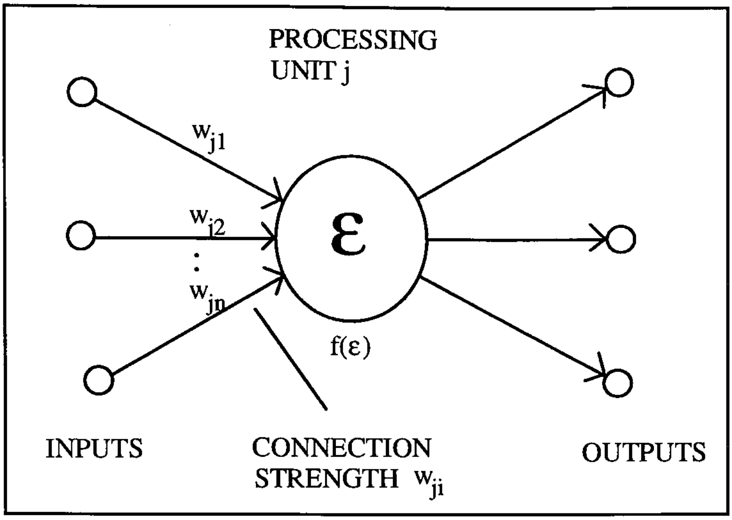

Figure 2-7: Schematic processing unit from an artificial neural network [16]

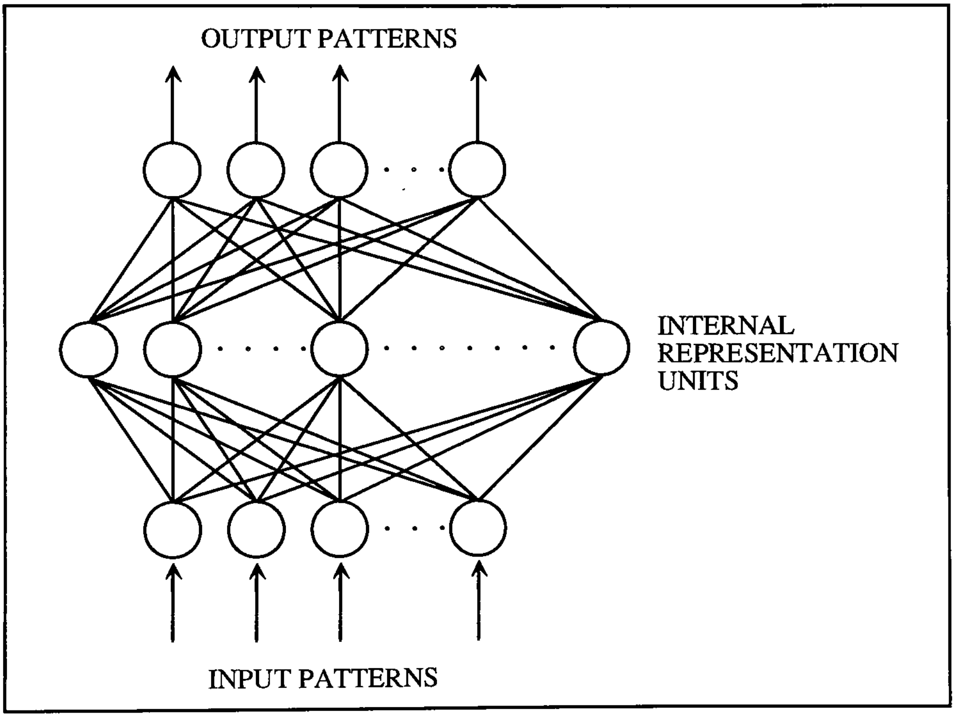

Figure 2-8: An artificial neural network with three fully interconnected layers. [16]

A larger neural network, in which each layer is organized as a two - dimensional slab of neurons, is shown in figure 2-9. Neural networks are not limited to three layers, and may utilize a huge number of interconnections.

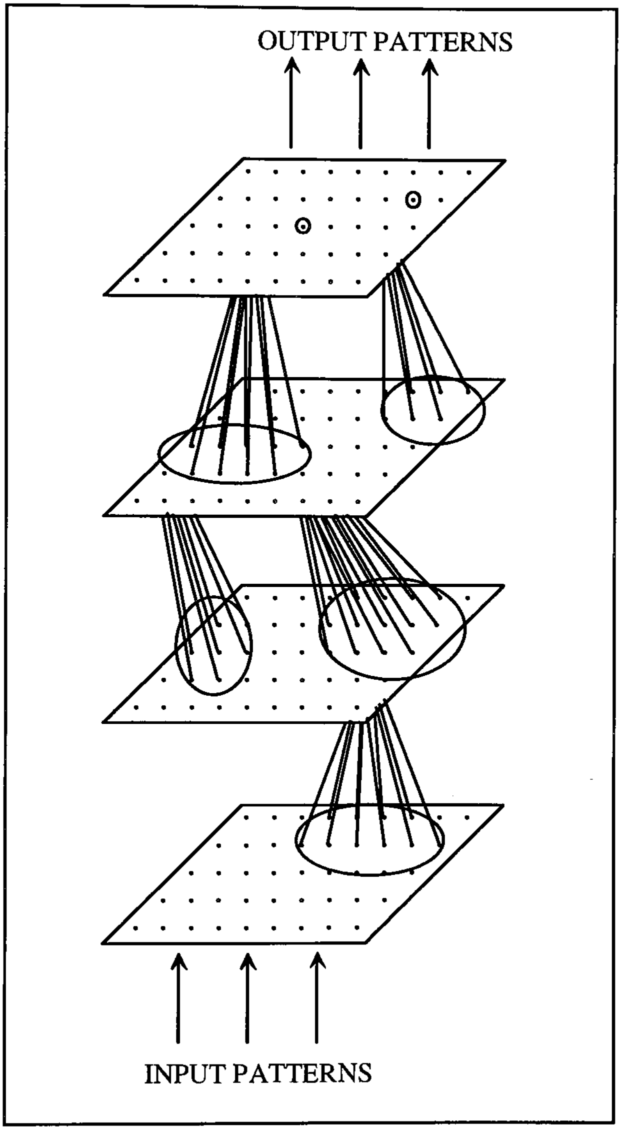

Figure 2-9: A multilayered network with slabs of processing units that are interconnected with adjacent layers. [16]

Each interconnection between processing units acts as a communication route. Numeric values are passed along these interconnections from one processing unit to another. These values are weighted by a connection strength when they are used computationally by the target processing unit. The connection strengths that are associated with each interconnection are adjusted during training to produce the final neural network.

Some neural network applications have fixed interconnections weights; these network operate by changing activity levels of neurons without changing the weights. Most networks, however, undergo a training procedure during which the network weights are adjusted. Training may be supervised, in which case the network is presented with target answers for each pattern that is input. In some architectures, training is unsupervised, the network adjusts its weights in response to input patterns without the benefit of target answers. In unsupervised learning, the network classifies the input patterns into similar categories.

2.4 NEURAL NETWORK CHARACTERISTICS

Neural networks are not programmed; they learn by example. Typically, a neural network is presented with a training set consisting of a group of examples from which the network can learn. These examples, known as training patterns, are represented as vectors, and can be taken from such sources as images, speech signals, sensor data, robotic arm movements, financial data, and diagnosis information.

The most common training scenarios utilize supervised learning, during which the network is presented with an input pattern together with the target output for that pattern. The target output usually constitutes the correct answer, or correct classification for the input pattern. In response to these paired examples, the neural network adjusts the values of its internal weights. If training is successful, the internal parameters are then adjusted to the point where the network can produce the correct answers in response to each input pattern. Usually the set of training examples is presented many times during training to allow the network to adjust its internal parameters gradually.

Because they learn by example, neural networks have the potential for building computing systems that do not need to be programmed. This reflects a radically different approach to computing compared to traditional methods, which involve the development of computer programs. In a computer program, every step that the computer executes is specified in advance by the programmer, a process that takes time and human resources. The neural network, in contrast, begins with sample inputs and outputs, and learns to provide the correct outputs for each input.

Figures 2-10a, and 2-10b, contrast two different approaches to a pattern classification problem. The task here is to classify pictures of a cat, a dog, and a rabbit. Figure 2-10a illustrates the traditional approach, and is compared to a neural network approach shown in figure 2-10b. In the traditional approach, preprocessing of the image is performed, followed by a human analysis of the data to identify the important features. Human resources are then utilized in developing algorithms and programs that make use of those features to identify the cat, dog, and rabbit. The result is a program that may classify the three types of pictures, or three different programs that each recognize a single picture type. The same programs cannot then be used to classify new types of pictures.

Figure 2-10b, illustrates the neural network approach. A single neural network is drawn three times in the figure. The net has already been implemented as a simulation, and may use special purpose hardware to accelerate its computations. Preprocessing of the image data is recommended. The network is presented with the picture of the cat as an input, and with the text string "cat" as an output. The weights are readjusted automatically. The same network is then presented with the picture of a dog as an input, and "dog" as an output, and the picture of a rabbit, with "rabbit" as the output. After each presentation the weights are again readjusted automatically. This training procedure is repeated many times. After training, the same network can identify all three types of pictures. The same neural network can then be retrained to classify additional picture types, or a completely new set of pictures.

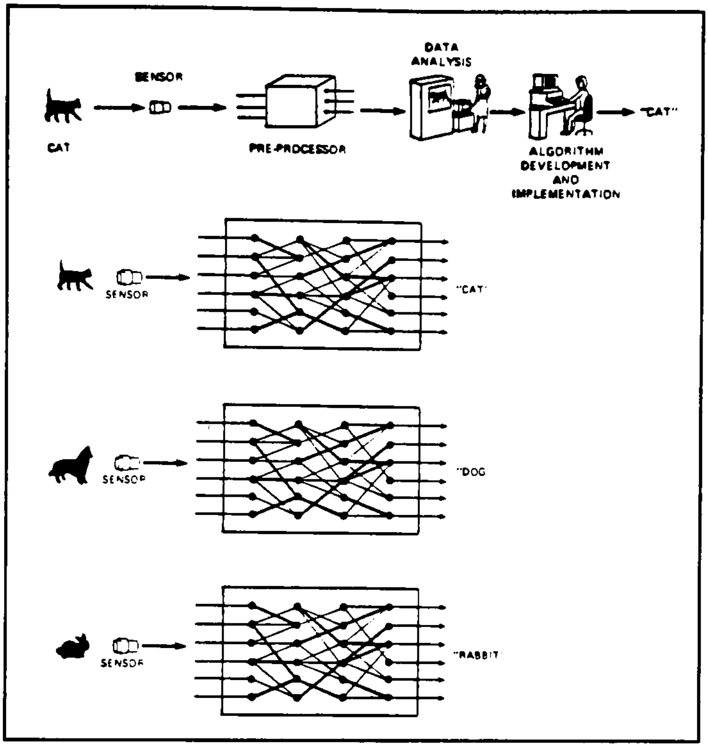

Figure 2-10: (a) Traditional approach to pattern classification.
(b) Neural Network approach. [16]

The neural network approach does not require human identification of features, or human development of algorithms and programs that are specific to the classification problem at hand, suggesting that time and human effort can be saved. There are drawbacks to the neural network approach, however. The time to train the network may not be known ahead of time, and the process of designing a network that successfully solves an application problem may be difficult. The potential of the approach, however, appears significantly better than past approaches.

Neural network architectures encode information in a distributed fashion. Typically the information that is stored in a neural net is shared by many of its processing units. This type of coding is in stark contrast to traditional memory schemes, where particular pieces of information are stored in particular locations of memory. Traditional speech recognition systems, for example, contain a lookup table of template speech patterns (individual syllables or words) that are compared one by one to spoken inputs. Such templates are stored in a specific location of the computer memory. Neural networks, in contrast, identify spoken syllables by using a number of processing units simultaneously. The internal representation is thus distributed across all or part of the network. Furthermore, more than one syllable or pattern may be stored at the same time by the network.

Distributed storage schemes provide many advantages, the most important being that the information representation can be redundant. Thus a neural network system can undergo partial destruction of the network and may still be able to function correctly. Although redundancy can be built into other types of systems, the neural network has a natural way to organize and implement this redundancy; the result is a naturally fault or error tolerant system.

It is possible to develop a network that can generalize on the tasks for which it is trained, enabling the neural network to provide the correct answer when presented with a new input pattern that is different from the inputs in the training set. To develop a neural network that can generalize, the training set must include a variety of examples that are good preparation for the generalization task. In addition, the training session must be limited in iterations, so that no "overlearning" takes place. Thus, special considerations in constructing the training set and the training presentations must be made to permit effective generalization behavior from a neural network.

A neural network can discover the distinguishing features needed to perform a classification task. This discovery is actually a part of the network's internal selforganization. The organization of features, for example, takes place in back-propagation. A network may be presented with a training set of pictures, along with the correct classification of these pictures into categories. The network can then find the distinguishing features between the different categories of pictures. These features can be read off from a "feature detection" layer of neurons after the network is trained.

A neural network can be "tested" at any point during training. Thus it is possible to measure a learning curve for a neural network. All these characteristics of neural networks may be explained through the simple mathematical structure of the neural net models. Although we use broad behavioral terms such as learn, generalize, and adapt, the neural network's behavior is simple and quantifiable at each node. The computations performed in the neural net may be specified mathematically, and typically are similar to other mathematical models already in use. Although large neural network systems may sometimes act in surprising ways, their internal mechanisms are neither mysterious nor incomprehensible.

2.5 AMARI AND HOPFIELD NETWORKS

2.5.1 AMARI NETWORK

In the early seventies, Amari proposed two self-organizing random networks [16], a Non-recurrent network for association and a Recurrent network for concept formation. The recurrent network, shown in figure 2-11a, is a sequential network containing n bistable elements (neuron pools) $\{v_1(t), v_2(t), ..., v_n(t)\}$. Each element, shown in figure 2-11b, consists of mutually connected neurons. The outputs of the network are connected to its inputs. The bistable element can be in one of the two states: $v_i(t) = 1$ (firing state) if many neurons in the pool are active and $v_i(t) = 0$ (resting state) if many of the neurons in the pool are off. Each element $v_i(t)$ receives weighted input signals from all the elements at time t. After summing the weighted inputs and comparing with a threshold h_i, the state of the element $v_i(t + 1)$ at time t + 1 is determined by,

$$v_i(t+1) = g(\sum_{j=1}^{n} T_{i,j} v_j(t) - h_i)$$

where $T_{i,j}$ is the weight (synaptic interconnection strength) from element j to element i and g(x) is the activation function defined by

$$g(x) = 1 \quad if \quad x \geq 0$$
$$0 \quad if \quad x \leq 0$$

which is a step function. The stable states are reached if

$$v_i(t+1) = v_i(t) \qquad \text{for } i = 1, 2, ..., n.$$

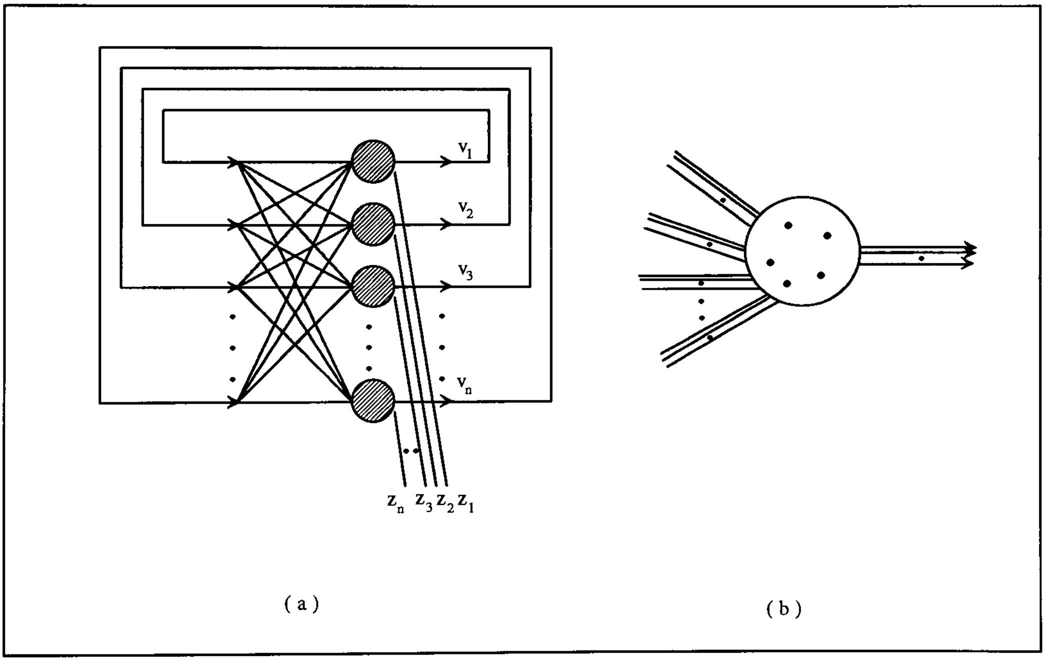

Figure 2-11: Amari recurrent network (a) Recurrent network (b) Bistable element [16]

Since the weight $T_{i,i}$ is nonzero, all the elements of the network have feedback. The recurrent network was actually derived from the McCulloch-Pitts formal neuron, the simplest form of neural network.

2.5.2 THE BINARY HOPFIELD NETWORK

2.5.2.1 BASIC STRUCTURE

The binary Hopfield net [16] has a single layer of processing units. Each processing unit has an activity value, or "state" that is binary - one of the two possible values. Here we use the binary states 0 and 1 (the network works the same way if values of +1 and -1 are used but slight changes in the equations are required).

The entire network is considered to have a "state" at each moment. The state is a vector of 0's and 1's. Each entry in the vector corresponds to an individual processing units in the network. Thus at any given moment, the state of the network is represented by a state vector such as:

$$U = (u1, u2, ..., un) = (+ +.....+.......+)$$

This vector reflects a network of n processing units, where element i has state u_i. In this notation, a '+' represents a processing unit with the binary value 1, and a '-' represents a processing unit with the value 0. Figure 2-12 shows a diagram of the processing units in a Hopfield network, together with an example state. The state of the network can change over time as the values of individual units change.

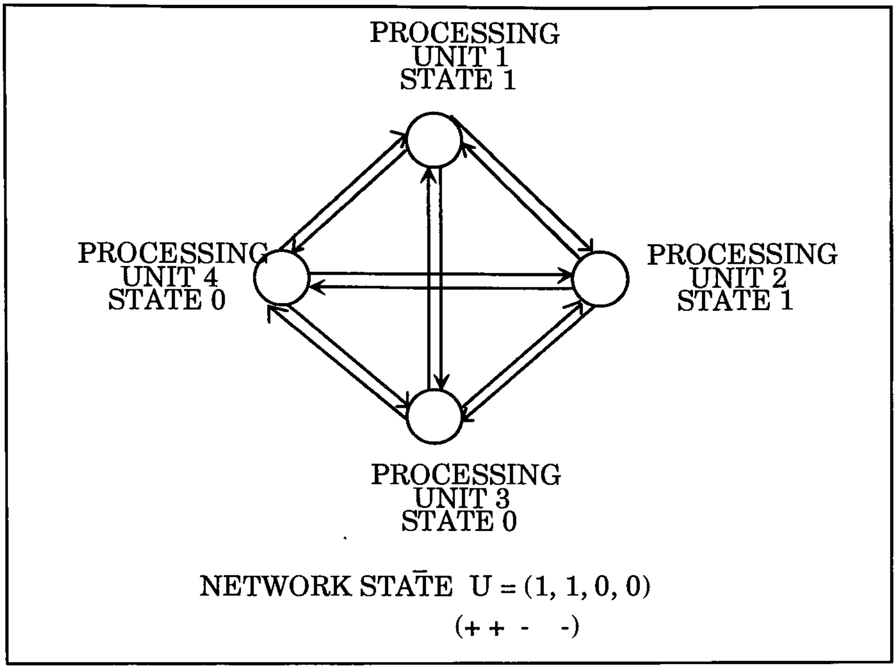

Figure 2-12: A binary Hopfield network [16]

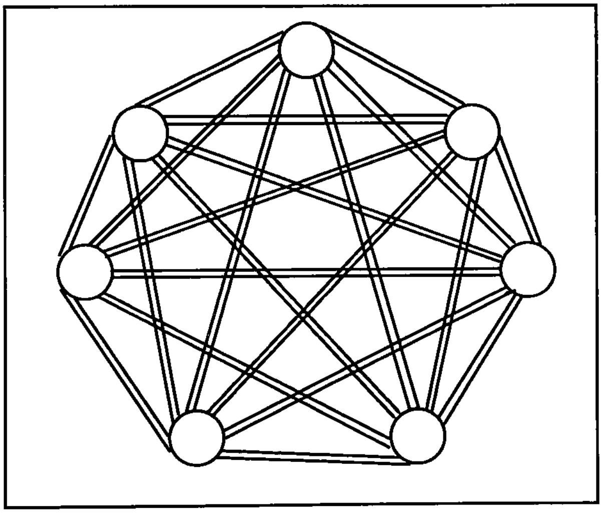

Figure 2-13: Fully interconnected one-layered networks, with connections in both directions between each pair of processing units. [16]

The processing units in the Hopfield network are fully interconnected - each unit is connected to every other unit. In fact, the connections are "directed", and each pair of processing units has a connection in each direction, as shown in figure 2-13. This interconnection topology makes the network "recursive" because the outputs of each unit feed into inputs of other units in the same layer. It will be seen that this recursive organization will allow the network to relax into a stable state in the absence of external input.

Each interconnection has an associated weight. This weight is a scalar value, considered intuitively to be the connection strength. Let T_{ji} denote the weight to unit j from unit i. In the Hopfield network, the weights T_{ji} and T_{ij} have the same value, therefore

$$T_{ji} = T_{ij}$$

Mathematical analysis [16] has shown that when this equality is true, the network is able to converge - that is, it eventually attains a stable state. Convergence of the network is necessary in order for it to perform useful computational tasks such as optimization and associative memory. Many networks with unequal weights ($T_{ji} \neq T_{ij}$) also converge successfully.

2.5.2.2 CONVERGENCE

Each state of the Hopfield network has an associated "energy" value. This value is defined by [16],

$$E = -\frac{1}{2}\sum_{j}\sum_{\substack{i \\ i \neq j}} T_{ji} u_j u_i \qquad (2\text{-}3)$$

The equation is referred to as "energy", although it does not represent the real energy of any physical system. The energy function in the equation is an objective function that is minimized by the network.

The successive updating of the Hopfield network provides a "convergence" procedure whereby the energy of the overall network gets smaller and smaller. Eventually the network goes into a stable state, at this stable state, the energy is at a minimum. This minimum may be local or global.

It is possible to prove that each time a processing unit is updated, the energy of the network either stays the same or decreases. As a result, this updating procedure will always allow the energy of the network to converge to a minimum.

There is also an argument that the updating procedure either decreases the energy of leaves it the same. Suppose that unit j is the next processing unit to be updated. Then, the portion of E affected by processing unit j is given by:

$$E_j = -\frac{1}{2}\sum_{\substack{i \\ i \neq j}} T_{ji} u_j u_i \tag{2-4}$$

which rearranges to

$$E_j = -\frac{1}{2} u_j \sum_{\substack{i \\ i \neq j}} T_{ji} u_i \tag{2-5}$$

When unit j is updated, if there is no change in its state, then energy E_j remains the same. If there is a change in its state, then the difference in E_j is:

$$\Delta E_j = E_{j_{new}} - E_{j_{old}} = -\frac{1}{2}\Delta u_j \sum T_{ji} u_i \qquad \text{where } \Delta u_j = u_{j_{new}} - u_{j_{old}} \tag{2-6}$$

If u_j changes from 0 to 1, then $\Delta u_j = 1$ and $\sum_i T_{ji} u_i \geq 0$ after updating. Plugging these nonnegative values into Eq. (2-6), we get $\Delta E_j \leq 0$

If u_j changes from 1 to 0, then $\Delta u_j = -1$ and $\sum_i T_{ji} u_i < 0$

after updating, by (2-2). Plugging these two negative values into (2-6), we get $\Delta E_j < 0$ Since ΔE_j is the product of three negative numbers. Thus, the change in E is always negative or 0 no matter what change there is in the state of unit j upon updating. The network is guaranteed to converge, with E taking on lower and lower values until the network reaches a steady state.

2.6 A DISCRETE NEURAL NETWORK FOR VISION

To deal with visual problems, we use several multi-dimensional neural networks. For illustration purposes, we present only a one-dimensional network. It is noticeable that since images are two-dimensional data, even if we use one neuron to represent each image pixel, then a huge number of neurons are needed to represent the whole image. For instance, for a 256 x 256 image, a total of 65, 536 neurons are needed. If each pixel needs multiple neurons, say m neurons, then a total of 65, 536 x m neurons are required. Figure 2-11a, shows a 1-D neural network.

2.6.1 A DISCRETE NETWORK

The network consists of n mutually interconnected binary neurons {v1, v2, ..., vn}. Each neuron takes the value 0 for resting and 1 for firing. Let $T_{i,j}$ denote the strength (possibly negative) of the interconnection between neuron i and neuron j. The interconnections are assumed to be symmetric

$$T_{i,j} = T_{j,i} \qquad \text{for} \qquad 1 \leq i,\ j \leq n$$

and the self-connection is not necessarily zero, the self-connection could be non zero

$$T_{i,i} \neq 0.$$

The non zero self-connection means there is a self-feedback for each neuron. In this network, each neuron (i, k) synchronously, or randomly and asynchronously receives inputs $\Sigma\ T_{i,j}v_j$ from all neurons including itself and a bias input I_i.

$$u_i = \sum_{j}^{n} T_{i,j}\, v_j + I_i \qquad (2.7)$$

Each neuron u_i is fed back to corresponding neurons after either thresholding or maximum evolution

$$v_i = g(u_i) \qquad (2.8)$$

where $g(x_i)$ is an activation function whose form is taken either as (2.8) for thresholding or

$$g(x_i) = \begin{cases} 1 & \text{if } x_i = \max(xk;\ \forall k \in \Omega_l) \\ 0 & \text{otherwise} \end{cases} \qquad (2.9)$$

for maximum evolution, where Ω_l's are disjoint subsets of index set $\Omega = \{1, 2, ..., n\}$ and $\bigcup \Omega_l = \Omega$, and $i \in \Omega_l$. The synchronous updating scheme uses information about the old states of all the neurons. By contrast, the asynchronous updating scheme uses the latest information about the states of the other neurons to update the state of the present neuron, which means that any state change in a neuron will immediately affect the state of all the neurons.

2.6.2 DECISION RULES

As mentioned above, this network has a self-feedback, $T_{i,i} \neq 0$. As a result of having the self-feedback, this network does not always converge to stable states. This can be explained as follows. Let E denote the energy function of the network. According to Hopfield [14], by setting thresholds $\{h_i\}$ to zero the energy function of the network can be found as,

$$E = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} T_{i,j}v_i v_j - \sum_{i=1}^{n} I_i v_i \qquad (2.10)$$

Let the state change of neuron i be,

$$\Delta v_i = v_i^{new} - v_i^{old}$$

and the energy change of the network be,

$$\Delta E = E^{new} - E^{old}$$

2.6.2.1 CASE 1: STEP FUNCTION

For simplicity of analysis, we assume that at each step, only one neuron changes its state either from 1 to 0 or from 0 to 1. When a step function is used as the activation function, the energy change ΔE due to a state change Δv_i of neuron i is given by

$$\Delta E = -(\sum_{j=1}^{n} T_{i,j}v_j + I_i)\Delta v_i - \frac{1}{2}T_{i,i}(\Delta v_i)^2 \qquad (2.11)$$

By (2.7), ΔE can be written as

$$\Delta E = -u_i \Delta v_i - \frac{1}{2}T_{i,i}(\Delta v_i)^2 \qquad (2.12)$$

When u_i is greater than zero, v_i changes its state from 0 to 1 and hence $\Delta v_i = 1$ which leads to

$$\Delta E = -u_i - \frac{1}{2}T_{i,i} \qquad (2.13)$$

If $T_{i,i} < -2u_i$, then $\Delta E > 0$. Similarly, when u_i is less than 0, v_i changes its state from 1 to 0. The state change Δv_i is then -1 and

$$\Delta E = u_i - \frac{1}{2}T_{i,i} \qquad (2.14)$$

If $T_{i,i} < 2u_i$, then $\Delta E > 0$. Hence, whenever

$$T_{i,i} < -2 \mid ui \mid$$

we have $\Delta E > 0$ which means that the energy changes are not always negative and the energy function does not decrease monotonically with a transition. E is not a

Lyapunov function and the network may not be stable. Consequently, the convergence of the network is not guaranteed.

2.6.2.2 CASE 2: MAXIMUM EVOLUTION FUNCTION

When a maximum evolution function is used, a batch of m neurons $\{v_k; k \in \Omega_t\}$is simultaneously updated at each step. Since the maximum evolution function only allows one neuron to be active and the others to be inactive, at most two neurons can change their state at each step, the active neuron becomes inactive and one of the inactive neurons becomes active. Suppose neurons i and i' change their states. The energy change ΔE due to the state changes of neurons i and i' is then given by

$$\Delta E = -(\sum_{j=1}^{n} T_{i,j} v_j + I_i)\Delta v_i - \frac{1}{2} T_{i,i}(\Delta v_i)^2 - (\sum_{j=1}^{n} T_{i',j} v_j + I_{i'})\Delta v_{i'} - \frac{1}{2} T_{i',i}(\Delta v_{i'})^2 - T_{i,i'}(\Delta v_i v_{i'}^{new} + \Delta v_{i'} v_i^{new}) \tag{2.15}$$

Similarly, by (2.7), ΔE can be written as

$$\Delta E = -u_i \Delta v_i - \frac{1}{2} T_{i,i}(v_i)^2 - u_{i'} \Delta v_{i'} - \frac{1}{2} T_{i',i'}(\Delta v_{i'})^2 - T_{i,i'}(\Delta v_i v_{i'}^{new} + \Delta v_{i'} v_i^{new}) \tag{2.16}$$

By properly setting v_i and $v_{i'}$, it is easy to show that the energy changes are not always negative. To ensure convergence of the network to a minimum, one can design some decision rules for updating the state of neurons. Depending on whether convergence to a local minimum or a global minimum is desired, a deterministic or stochastic decision rule can be used, respectively. In some cases, for example when the energy function is convex, the deterministic decision rule will ensure that the network will converge to a global minimum.

2.6.2.3 DETERMINISTIC DECISION RULE:

The deterministic rule is to take a new state v_i^{new} of neuron i if the energy change ΔE due to state change Δv_i is less than zero. If ΔE due to the state change is > 0, no state change is affected.

2.6.2.4 STOCHASTIC DECISION RULE

A stochastic rule is similar to the one used in simulated annealing techniques. We define a Boltzmann distribution by,

$$\frac{p_{new}}{p_{old}} = e^{\frac{-\Delta E}{T}}$$

where p_{new} and p_{old} are the probabilities of the new and old global state respectively, ΔE is the energy change and T is the parameter which acts like temperature. A new state v_i^{new} is taken if

$$\frac{p_{new}}{p_{old}} > 1, \quad \text{or if} \quad \frac{p_{new}}{p_{old}} \leq 1 \quad \text{but} \quad \frac{p_{new}}{p_{old}} > \xi$$

where ξ is a random number uniformly distributed in the interval [0,1].

3.0 SIGNAL AGGREGATION PRINCIPLES

3.1 STATISTICAL COMPUTATION

Statisticians have various criteria by which, if a data point is sufficiently out of line, its effect may be reduced. A common procedure is to develop some notion of reasonable behavior. In an experiment, we often have sound theoretical reasons to believe that the output should be some smooth function of some independent variable (the input). The transistor curves and the amplifier transfer curves can be considered as examples. In both cases, there is a voltage scale given by kT/(qK). (Where q is the charge, k is the Boltzmann constant, T is the temperature and K is the process constant. kT is the thermal energy per charge carrier and has units of potential, thus it is called *Thermal Voltage,* and its magnitude is equal to 0.025 volt at room temperature. K is approximately equal to 0.7, and it remains reasonably constant among transistors in a single fabrication batch. We will treat this kT/(qK) term as the unit of voltage.) If we change the input less than this amount, we do not expect the output to change abruptly. Hence, if we take several data points within each kT/(qK) voltage interval, we have a great deal of redundancy in the input. If the distance from a single data point to a smooth curve passing through the average of other points in the neighborhood is relatively large, we should certainly check out that maverick data point. Any such scheme relies on four important features,

- We know the size of a "region of smoothness" within which, for some fundamental reason, the data cannot change abruptly.
- Many data points are available within the region of smoothness.
- A method, consistent with the nature of the expected smoothness, is available for fitting a smooth function through the data points.
- Some method of estimating the average deviation of the data from the smooth function is available.

Once we have formulated a computation with these attributes, we can use it to identify unexpected data points. These may be "bad" points, or they may be items of exceptional interest. Sensory processing is replete with examples of spatially and temporally smoothed signals. These smooth functions are used to provide a reference for local computations. The most widely known example is the center-surround organization of many visual areas, from retina to cortex.

3.2 FOLLOWER AGGREGATION

The simplest circuit for computing a smooth function is shown in the figure 3.1. It consists of n follower stages, all driving the single wire labeled V_{out}. As shown in the figure, the output of each individual amplifier is a current, whereas the output of the entire aggregation is a voltage. That voltage is the outcome of a collective interaction of the entire set of amplifiers.

There are n amplifiers, each responsible for the contribution of its V_i input to the common output. Each amplifier has a transconductance: G_1 for A_1, G_2 for A_2, and so on to G_n for A_n. The G's are set by the current controls on the transconductance amplifiers. By Kirchoff's current law the sum of all currents into the node V_{out} is zero.

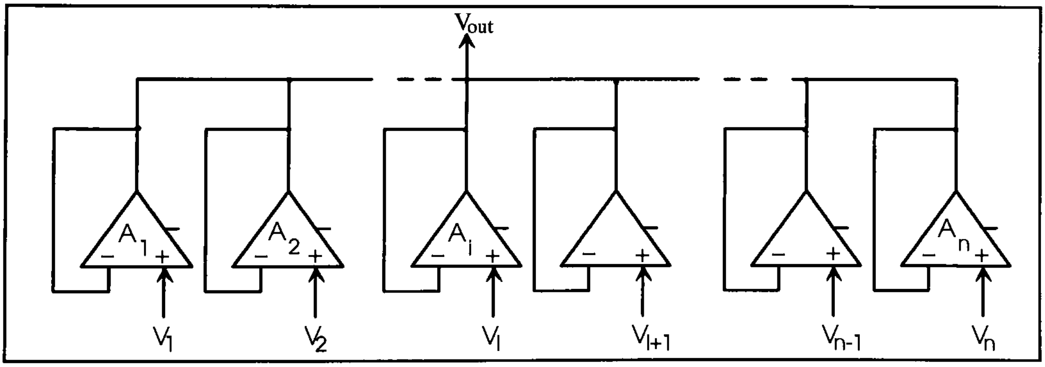

Figure 3.1: Schematic of the follower-aggregation circuit. [1]

The total current is the sum of the currents out of each amplifier. The current for the first amplifier is $G_1(V_1-V_{out})$, that for the second is $G_2(V_2-V_{out})$ and for the nth is $G_n(V_n-V_{out})$.

$$\sum_{i=1}^{n} G_i(V_i - V_{out}) = 0$$

Transferring the V_{out} terms to the other side of the equation and rearranging, we obtain,

$$V_{out} = \frac{\sum_{i=1}^{n} G_i V_i}{\sum_{i=1}^{n} G_i}$$

In other words, V_{out} is the weighted average of the Vi inputs, each input weighted by its transconductance G_i.(Provided $G_1 = G_2 = G_i$)

3.3 ROBUSTNESS

The follower-aggregation circuit computes the weighted average of the input voltages $V_1, \ . \ . \ . \ , V_n$. Up to this point, our analysis has assumed a linear relation between input voltages and output current. The important thing to keep in mind is that the follower implementation of a neural network has great robustness against bad data points.

Transconductance amplifiers have a strictly limited current output. This limit is evident in their tanh transfer characteristics. The robustness of collective networks made with these circuits is a direct result of this current limitation. If any one input voltage is way off scale, it does not matter. The off_scale voltage will not pull any harder on the wire than would a voltage a few kT/(qK) different from the intended voltage of the wire. As long as all inputs are close to the average value, V_{out} will assume an average, with the inputs weighted by the current in their amplifier.

From a statistical viewpoint, the tanh characteristics changes the computation done by the network. It implements what statisticians call a *resistant transformation.* The weighting assigned to outlying data points is reduced. For all signals close to V_{out}, we have seen that the circuit computes a weighted average, or *mean*. Signal values that are scattered by many kT/(qK) are treated as inputs to a weighted *median* calculation. In both cases, the data are weighted by the transconductances of their respective amplifiers. To ensure that no single amplifier contributes more than its share to the output, we use wide-range amplifiers to avoid the V_{min} problem.

3.4 RESISTIVE NETWORKS

The follower-aggregation performs well in computing an average that can be used as a reference against which to measure exceptional events. There is a problem, however, with this kind of average. The average is represented by the voltage on a single wire, and that wire is a single electrical node. The average, therefore, will be a global average. It is highly desirable in the visual systems to have a local average, one in which the contribution of spatially distant inputs is less than that of inputs in close proximity to the point at which the average is used. The illumination level within a visual scene often varies from one point to another by several orders of magnitude. If the visual system used a global average as a reference, details in very bright and very dark areas would be invisible.

A locally weighted average signal, from which local differences can be measured, is computed by a layer of *horizontal cells* in the retina. These cells are linked together by high resistance connections called *gap junctions* [17], and form an electrically continuous resistive network just below the photoreceptors. Propagation of signals in resistive networks is generically referred to as the *electronic spread.*

3.4.1 ELECTRONIC SPREAD

The simplest example of electronic spread occurs in a long, straight, passive neural process of constant diameter. We can model the process as a resistive ladder network, as shown in the figure 3.2. The R resistances correspond to the axial resistance per unit length of the cytoplasm, and the G conductance represent the leakage conductance per unit length through the membrane to the extracellular fluid. A potential V_0 is generated by an input at the left end of the process. The voltage V(x) generated by the input decreases with distance x from the input, because some of the current injected by the input is shunted to ground by the G conductance.

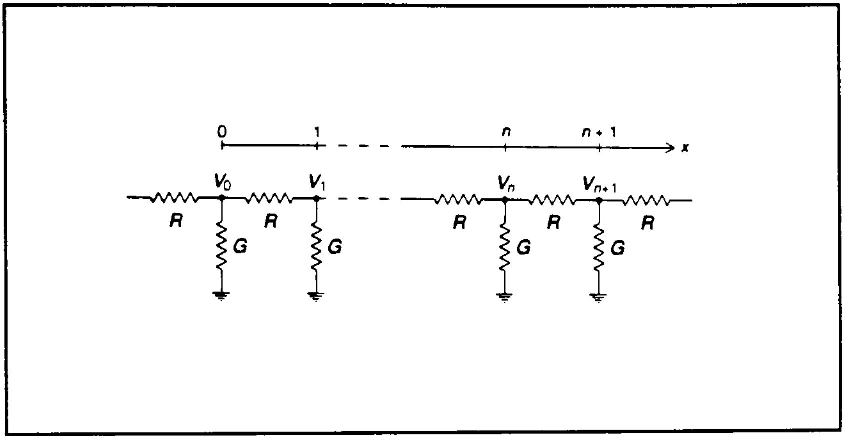

Figure 3.2: Resistive model of passive electronic spread in a neural process [17]

For Uniform, continuous networks, the voltage has the form

$$V = V_0 e^{-\alpha|x|} = V_0 e^{-\frac{1}{L}|x|} \qquad (3.1)$$

where α is the space constant and L is the characteristic length or diffusion length of the process.

$$\alpha = \frac{1}{L} = \sqrt{RG} \qquad (3.2)$$

A signal injected into a linear resistive ladder network decays exponentially with distance from the source. If a signal is injected into a node in the middle of a very long process, the influence of that input spreads out in both directions, not just in the +x direction.

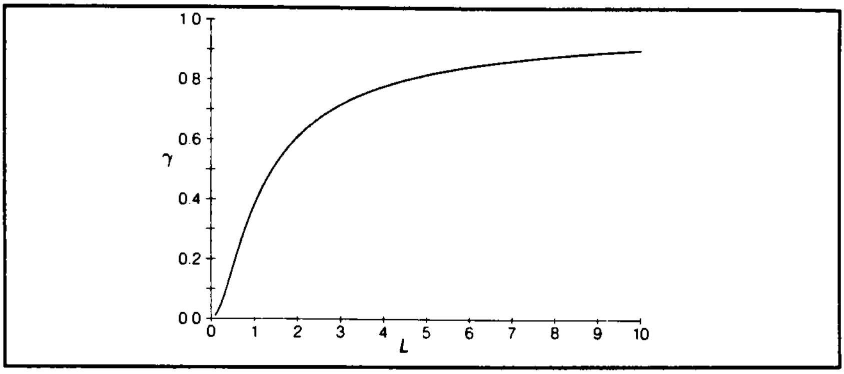

Figure 3.3: The exponent γ as a function of L [17]

For discrete networks, the decay also is exponential. For a node n sections away from the source, the voltage will be

$$V_n = \gamma^n V_0 \qquad \text{where } \gamma = \frac{V_1}{V_0} = 1 + \frac{1}{2L^2} - \frac{1}{L}\sqrt{1 + \frac{1}{4L^2}} \tag{3.3}$$

where 1/L is equal to $\sqrt{RG}$ as before, but in the discrete case the values of R and G are given per section rather than per unit length. A plot of γ as a function of L is shown in figure 3.3. For large values of L, γ approaches 1, and the continuous approximation of the equation is valid. For values of L less than about 10, the magnitude of the decrement per stage given by the discrete solution differs markedly from that obtained from the continuous approximation.

3.4.2 MULTIPLE INPUTS

Multiple signal inputs to a network can be provided in the form of either voltage or current type signals. If we inject currents at many places, the network performs an automatic weighted average, the farther away the inputs are, the less weight they are given, in accordance with equation 3.3. The voltage at any given point k due to a number of inputs is just,

$$V_k = \frac{1}{2G_o}\sum_n \gamma^{|n-k|} I_n$$

In other words, by the principal of superposition the voltage at any given point due to a number of inputs is just the sum of the voltages that would have been measured at that point had each input been presented individually, with all other inputs held at zero. A convenient way to generate inputs to the network is to connect voltage sources in series with the conductance, as shown in figure 3.4. Using the principle of superposition, we need compute only the node voltage due to a single input. It is seen that the node voltage V_i generated by a voltage source v_i in the middle of a very long, uniform, discrete, one dimensional network is,

$$\frac{V_i}{v_i} = \frac{1}{\sqrt{4L^2+1}}$$

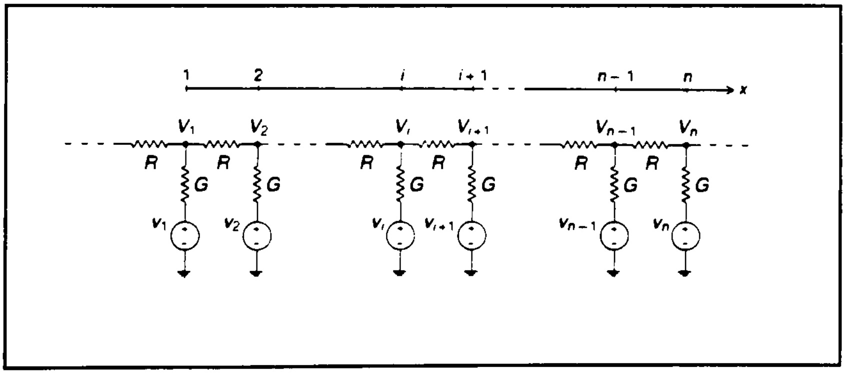

Figure 3.4: Electronic network in which input signals are supplied by voltage sources [17]

As the effective length over which the network averages increases, the effect of any given input decreases. For large characteristic lengths, the voltage due to any particular input is proportional to 1/L. The total effect of a set of uniformly spaced inputs included in one characteristic length is therefore constant, independent of the value of L, because the number of inputs is proportional to L. Thus it is seen that, when the voltage at all inputs is the same, the output voltage anywhere in the network is equal to the input voltage.

3.5 DENDRITIC TREES & SYNAPTIC INPUTS

Many types of neurons have no axon whatsoever, so their primary role cannot be to produce action potentials [17]. Many type of neurons, those with axons and those without, have been shown to have *synaptic outputs* as well as inputs on their dendrites. This finding suggests that much of the lateral communication in the nervous system is extremely local, and is mediated by graded analog potentials rather than by more digital nerve pulses. The dendrites convey two way information rather than merely collecting current into the soma.

If enough current is injected into the dendritic tree, then the neuron will release neurotransmitter from any output synapses it has on its dendrites. If the current into the cell as a whole reaches a high enough level, the nerve can initiate pulses in its axon. *Depolarizing* inputs cause the release of neurotransmitter from dendritic synapses and, if sufficiently intense and prolonged, can cause the axon to fire as well. These inputs are called *excitatory.* Inputs that *hyperpolarize* the neuron act to cancel out the effect of excitatory inputs, they are therefore called *inhibitory.*

If the entire path from the leaves of the dendrites to the axon hillock is less than L in length, the neuron is said to be *electrically compact.* such a cell can be assumed to be equipotential throughout its dendrites, and therefore can be modeled as a wire. A neuron with dendritic processes much longer than L can have very different potentials at different locations in the dendritic tree. The dendrites of such a neuron can be modeled as linear resistive networks.

3.5.1 SHUNTING INHIBITION

We have used the voltage sources of figure 3.4 to model excitatory and inhibitory input synapses to the network. Inputs also may be injected as currents, one sign of current being excitatory and the other inhibitory. There is a third class of inputs, often called veto synapses, that neither hyperpolarize nor depolarize the neuron, but instead partially short-circuit to ground any activity present in the process. This kind of inhibition is called shunting inhibition.

The simplest realization of shunting inhibition is implemented directly by the network of figure 3.4, we merely make one conductance, G_{shunt}, very large compared with the others. This arrangement will attenuate a signal traveling in either direction in the process. The attenuation suffered by a signal as it passes such a shunt is given by,

$$V_{out} = \frac{V_o}{1 + \frac{G_{shunt}}{2G_o}}$$

Where V_0 is the voltage that would have been present without the shunt. As G_{shunt} becomes large compared with the network effective conductance G_0, the operation performed by such a synapse resembles a division by G_{shunt}.

3.6 TWO - DIMENSIONAL NETWORKS

The horizontal network in the retina is a flat mesh of dense processes that are highly interconnected by resistive gap junctions. These interconnections are somewhat random in number and direction. Any given cell is connected with many others, and there is a great deal of overlap among interconnected cells. In silicon, discrete two-dimensional networks are very useful, and generally are implemented in a regular array by interconnection of nearest neighbors. We have mentioned that this kind of network computes an average that is a nearly ideal way to derive a reference with which local signals can be compared. In the figure 3.5, we see six resistors coming into each node. A resistance R is connected between neighboring nodes, and a conductance G is connected from each node to ground.

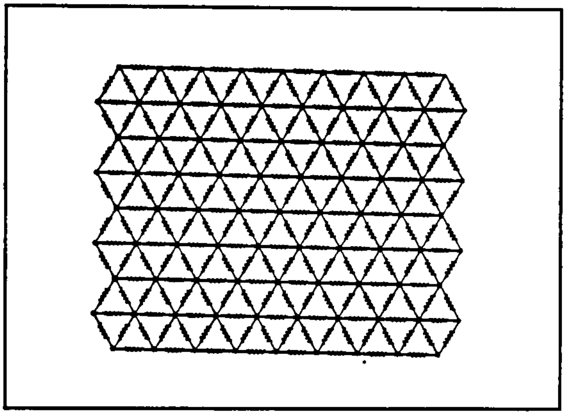

Figure 3.5: Topology of a hexagonal network. [3]

This network is particularly interesting, because it has the highest symmetry and connectivity of any regular, two dimensional structure. If we introduce a current into a node of the network (let us call it node 0), the resulting voltage decays exponentially with distance from that node. We can derive an approximate solution for the decay law in the following manner. As we progress outwards from node 0 following a row of resistors, we encounter nodes that are vertices of larger and larger hexagons centered on node 0. The index of hexagon *'n'* (its radius) is just the number of resistors we must pass through on the direct path from node 0 to a vertex. Our circular approximation assumes that all nodes on the perimeter of a given hexagon have the same voltage. Under this approximation we can write a finite difference equation for the current into hexagon n in terms of the voltage relative to that of hexagon n - 1 and to that of hexagon n + 1. We notice that there are 6n nodes on the perimeter of hexagon n, and that there are 12n-6 resistors from hexagon n-1 to hexagon n, and 12n+6 resistors from hexagon n to hexagon n+1. The current I into hexagon n is therefore,

$$I = \frac{(12n-6)(V_{n-1}-V_n)-(12n+6)(V_n-V_{n+1})}{R} - 6nGV_n \qquad (3.4)$$

In steady state, this current must be zero. Simplifying the above equation for zero current, we obtain the finite difference equation for the steady state node voltage.

$$(2n+1)V_{n+1} - n(RG+4)V_n + (2n-1)V_{n-1} = 0 \qquad (3.5)$$

3.6.1 SOLUTION FOR THE HEXAGONAL NETWORK

The equation 3.5 relates the vertex voltages of three consecutive hexagons of the network. Solving for V_{n+1}, we obtain the forward recursion relation,

$$V_{n+1} = \frac{n(RG+4)V_n - (2n-1)V_{n-1}}{2n+1} \qquad (3.6)$$

which produces the voltage on a given hexagon in terms of the voltages on the two smaller concentric hexagons. If we know V_0, the voltage at the center of the network, and V_1, the voltage at the first hexagon, we can solve for V_2. We can now iterate this procedure, given V_1 and V_2, we can determine V_3, and so on.

Determining V_1 for a given V_0 is the hard part of solving the network. The correct choice for V_1 leaves V_n finite as n approaches infinity. Any other choice for V_1 causes V_n to diverge as n approaches infinity. The values for V_1/V_0 as a function of L, computed by evaluating the exact expression, are shown in figure 3.6. Once we know V_0 and V_1, we can use equation 3.6 to evaluate the next few V_n. Knowing V_0 and V_1 also allows us to compute the current through the six resistors radiating from node 0, and hence to compute the input conductance of the network.

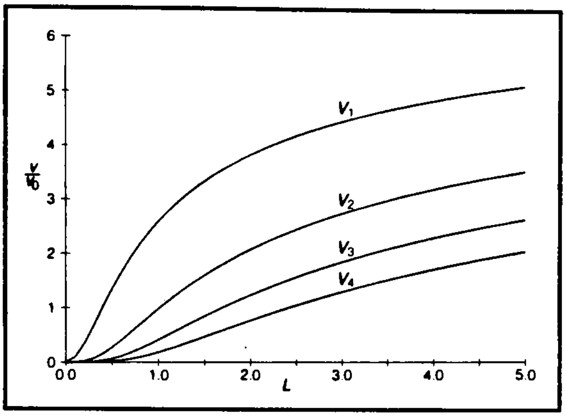

Figure 3.6: Voltages of the first few hexagons as a function of L. [3]

For n larger than 2L, the successive iteration of equation is subject to rapid erosion of numerical precision, and it is best to calculate V_n from the asymptotic relation,

$$V_n \approx \gamma^n \frac{V}{\sqrt{n}} \quad \text{where } \gamma = 1 - \frac{2}{1+\sqrt{1+8L^2}} \tag{3.7}$$

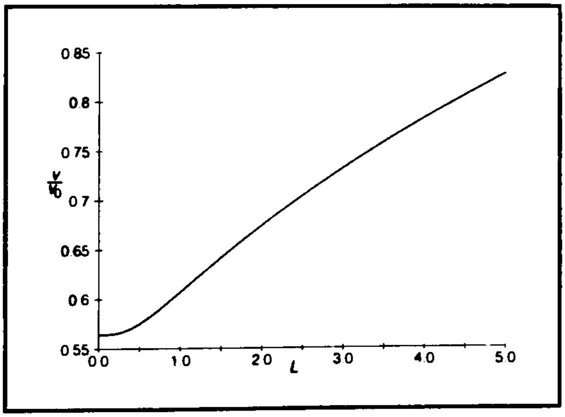

Figure 3.7: The value of V in equation 3.7 as a function of L. [3]

Equation 3.7 is the two dimensional equivalent of the expression given in equation 3.3 for the one dimensional network. The value of V in equation 3.7 is a complicated function of L, and is plotted in figure 3.7.

3.6.2 ROUTING COMPLEXITY

It is seen that a resistive network computes a smooth average over a number of neighbors, with the neighbors farther away contributing less to the average. If we were to replace this network with a set of circuits that did a completely separate computation at each location, the amount of wire required would proliferate enormously. For a computation centered at a given point, we would need to run a wire to every signal source that formed an input to the computation. A computation centered at a different location would require wires back to its sources also. By the time we were done, we would have duplicated many levels of wiring.

To obtain an efficient design, we must share as many signal paths as possible. In that way, we avoid duplication of wire, and also share the maximum amount of processing circuitry. The resistive network is the ultimate example of a shared function. Every location can put signals into the network, read voltages off of the network, and use the same network to sense this weighted sum over its neighbors, including itself.

If we are willing to include the location itself in the average, letting it make the greatest contribution to the average, then we can use a resistive network to compute this kind of weighted average. This type of an arrangement might play an important role in neural circuitry, because the computation is shared by every location, we need only one network, and every location gets taken into account.

3.7 HORIZONTAL RESISTOR CIRCUIT

We will describe a resistor with a control input that allows us to set the resistance electronically.

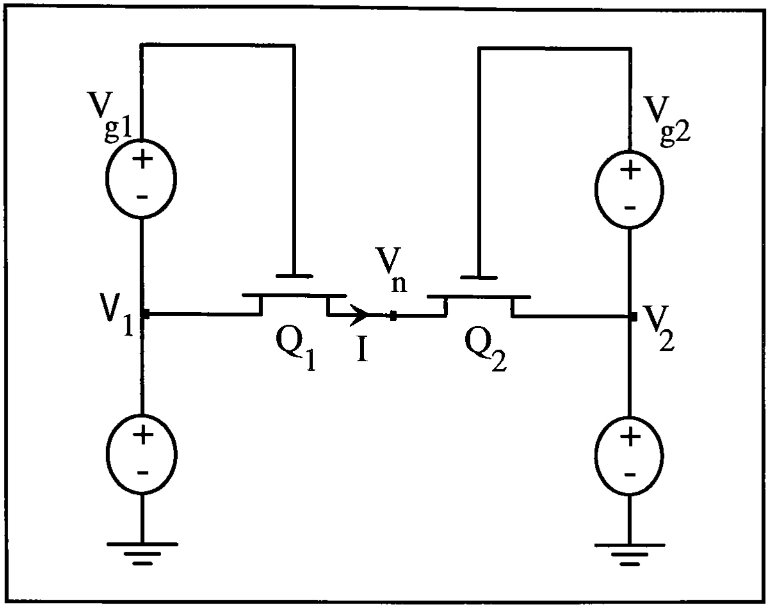

Figure 3.8: Schematic of the resistive connection of the horizontal resistor circuit. [17]

The most elementary resistive connection is implemented by two pass transistors in series, as shown in the figure 3.8. The gate voltage of each transistor is set at a fixed value V_q above the input voltage V_1 or V_2. This bias voltage controls the saturation current I_0 of the Q_1 and Q_2 pass transistors and therefore sets the effective resistance of the connection. The current I through a transistor is,

$$I_0 e^{kV_g}(e^{-V_s} - e^{-V_d}) = e^{kV_g - V_s}(1 - e^{V_s - V_d}) \qquad (3.8)$$

where all voltages are expressed in units of kT/(qK). This equation is completely asymmetrical under the interchange of source and drain terminals. For V_1 greater than V_2, V_1 acts as the drain of Q_1, and the intermediate node V_n acts as the source of Q_1 and the drain of Q_2. The current I is limited by Q_2, and saturates for V_1-V_2 much greater than kT/q because the gate source voltage of Q_2 is set by the bias voltage. For V_2 greater than V_1, the roles of Q_1 and Q_2 are reversed, and I is negative. For V_1 approximately equal to V_2, the circuit acts like a resistor.

The current through Q_1 must be negative of that through Q_2. Writing equation 3.8 for Q_1 and Q_2, and setting the currents equal and opposite, we obtain

$$I = I_0 e^{V_q}(e^{V_1 - V_n} - 1) = I_0 e^{V_q}(1 - e^{V_2 - V_n}) \qquad (3.9)$$

From equation 3.9, we can determine the voltage V_n at the junction between the two pass transistors,

$$2e^{V_n} = e^{V_1} + e^{V_2} \qquad (3.10)$$

Substituting equation 3.10 into equation 3.9, we obtain,

$$\frac{I}{I_{sat}} = \frac{e^{V_1} - e^{V_2}}{e^{V_1} + e^{V_2}} \qquad \text{where} \qquad I_{sat} = I_0 e^{V_q} \qquad (3.11)$$

Multiplying top and bottom of equation 3.11, by $e^{\frac{-(V_1+V_2)}{2}}$ and simplifying, we obtain the final expression for the current,

$$I = I_{sat} \tanh(\frac{V_1 - V_2}{2})$$

The slope of the tanh function at the origin is unity, therefore, the effective resistance R of this kind of resistive connection is $R = \frac{2kT/q}{I_{sat}}$.

3.8 NETWORKS IN CMOS

The horizontal resistor circuit is the simplest element with which we can build an electronic analog of the neuron's dendritic tree. Let us start implementing a one dimensional network analogous to the linear arrangement of figure 3.4. Such a network is shown in figure 3.9. Each local signal drives the network with a follower. The local current into the network is thus proportional to the difference between the signal and the local potential of the network.

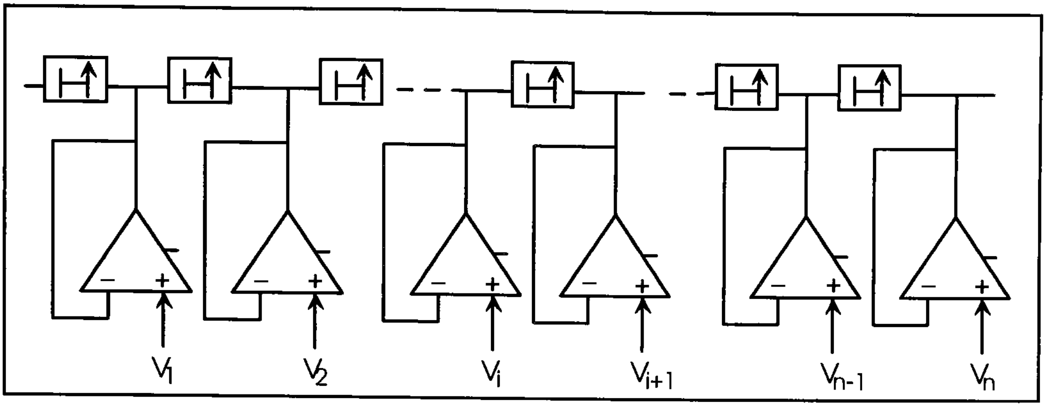

Figure 3.9: CMOS implementation of the abstract network of figure 3.4. [17]

The inputs to the transconductance amplifiers correspond to the voltage sources in figure 3.4, the transconductance of the amplifiers corresponds to G in figure 3.4, and the horizontal resistor circuits take the place of the resistors. The value of L is controlled by the ratio of the current in the bias circuit of the horizontal resistor circuits to the current in the transconductance amplifiers. Because the network voltage can be very different from the input voltage, it is desirable to use wide range amplifiers for the followers in a network of this type.

The structure shown in figure 3.9 can be used to implement two dimensional networks like the hexagonal topology shown in figure 3.5. The horizontal resistor circuit is ideal for networks in which many resistive connections converge on each node. Only one biasing circuit is required per node, the node is connected to the voltage V_{node} of the biasing circuit. All pass transistors connected to that node have their gates connected to the Vg output of the biasing circuit. The larger connectivity required by the hexagonal network is thus achieved at low incremental cost. In addition to the bias circuit, each node requires only one pass transistor per connection.

3.8.1 SMOOTH AREAS

It is seen that the follower aggregation circuit computes node voltages that are a smooth approximation to the input data, as long as each element is operating within its linear range. The output of this computation is a single voltage.

The linear resistive network computes a set of node voltages from a set of input voltages. A given node voltage is a weighted sum of the inputs, the weight of each input decreases exponentially with the distance from the node in accordance with equation 3.3. If we view the inputs and outputs as functions in one dimension, the node voltages are a smooth approximation to the inputs. For small L, inputs within a small region around any given output node contribute to the output value, and the smoothing will be minimal. For larger L, proportionally more smoothing will occur.

Similarly, a two dimensional resistive network computes node voltages that are a smooth approximation to a two dimensional set of inputs. We can think of the network computing a smooth fit at each point to the data included in a region of diameter approximately equal to L.

Smoothing in two dimensions is an important computation in image processing. Often, objects in an image have some property, such as color or velocity, that is smooth over the object but changes discontinuously at the boundary of the object. A two dimensional horizontal resistor network can smooth a signal over a large region in a visual image, even though a signal representing some property of the image changes by many kT/(qK) voltage units over that region. The voltage difference between any two neighboring nodes can be less than kT/(qK), even though the total difference across the smooth areas of the image can be much greater. An abrupt discontinuity, however, as occurs when we try to put many kT/q voltage units across one resistor, will simply cause a discontinuity in the network voltage. The current out of the horizontal resistor circuits will limit at I_{sat}, no matter how large the voltage drop across the resistive connection is. So a horizontal resistor network computes a smooth approximation to the inputs as long as the drop across any one element is less than approximately kT/q. It allows a larger voltage discontinuities by limiting the current through each element.

3.8.2 SEGMENTATION

For signals larger than approximately kT/(qK), both the followers and the resistors saturate. A small amount of saturation can be seen in the first data points in figure 3.10. This saturation property leads to the same kind of robustness that we had observed for the follower aggregation circuit.

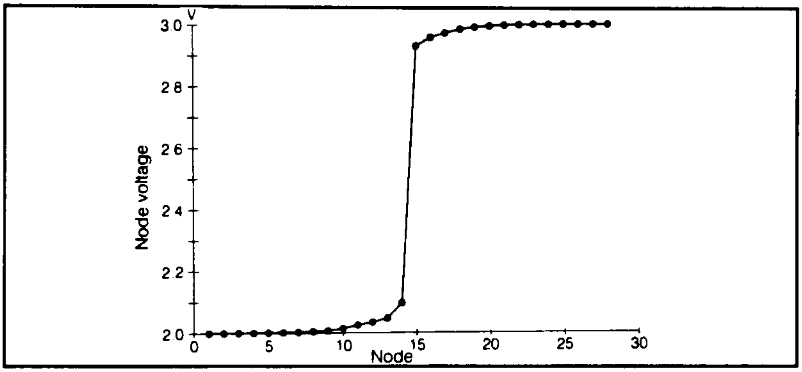

Figure 3.10: Segmentation due to saturation of resistive connections. [17]

In many situations, discontinuities are not only desirable but also necessary. An image is made up of smooth regions separated by discontinuities. The discontinuities carry the most information about the image. A network built of horizontal resistor circuits allow arbitrarily large discontinuities. Suppose, for example, we have a high contrast edge in an image. There is a set of signals pulling up on one part of the network, and another set of signals pulling down on another part. Voltages at various positions in the network will look like figure 3.10. A resistor on the high contrast edge cannot supply enough current to keep the network within the linear range, there will be a big drop across that resistor, and then the network voltage will go off smoothly on the other side, as shown in the figure. That kind of discontinuity cannot occur in a network made of linear resistors, but it is a natural result of the non linear nature of a network of horizontal resistor circuits. The resulting image is *segmented* into regions over which the property represented by the input voltages V(x,y) is smooth. We can easily identify the boundaries of these regions by finding the positions with voltage differences larger than kT/(qK). Computation of the natural boundaries in an image is called segmentation. Thus it is seen that the physics of a simple nonlinear circuit enables us to perform a complex computation in a simple way.

3.9 SINGLE STAGE OPERATIONAL TRANSCONDUCTANCE AMPLIFIER

The major problems in the realization of micro power operational amplifiers is that of achieving reasonable speed and acceptable dynamic range. Battery operation allows more relaxed requirements on power supply rejection, since all power noise is generated on the chip and may be more easily filtered out at very low current.

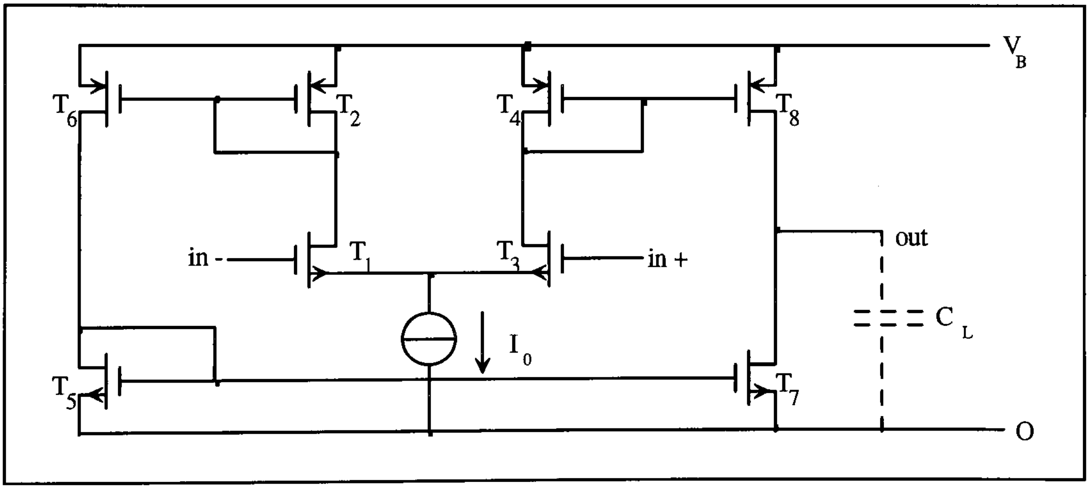

Figure 3.11: Simple OTA with differential input.

Assuming a ratio B of mirror T_4 - T_8, the total transconductance is

$$g_{mt} = Bg_{m1(3)} \tag{3.12}$$

The circuit, loaded by C_L, behaves essentially as an integrator with time constant

$$\tau_u = \frac{C_L}{g_{mt}} \tag{3.13}$$

which is the inverse of the unity gain angular frequency ω_u. This corresponds to a dominant pole at a very low frequency that depends on DC gain A_0, as shown below.

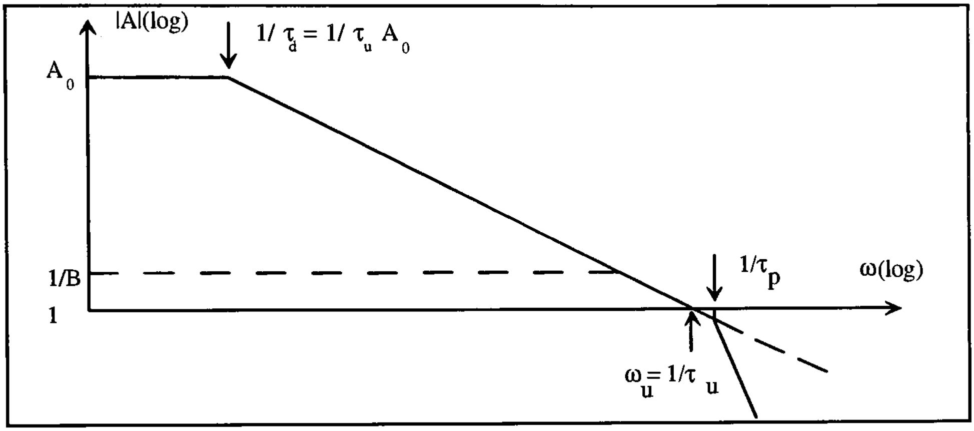

Figure 3.12: Frequency behavior of gain A.

Let us assume a single non-dominant parasitic pole with time constant Γ_p. This time constant may be that of the output nodes of the differential pair (T_1 - T_3). It may also represent the effect of all parasitic poles in any amplifier, which can be shown to be equivalent to a single pole,

$$\omega \leq \frac{1}{2\Gamma_p}$$

The open loop high frequency gain may thus be expressed as

$$A = \frac{1}{s\Gamma_u(1+s\Gamma_p)} \tag{3.14}$$

An opamp is usually used with an amount of voltage feedback $1 \geq \beta >> 1/A_0$.

The settling time T_S necessary to reach equilibrium with a residual error ε after application of a small unit step may be calculated with the gain given by equation 3.14.

$$T_s \cong (2\Gamma_p + \frac{\Gamma_u}{\beta}) \ln \varepsilon^{-1} \tag{3.15}$$

For small bias currents (I0 ≅ 0.5 μA), 1/f noise usually still dominates in the audio frequency range, even for large sized input transistors T_1 and T_3. However, in most practical applications, noise bandwidth is larger than 100 kHz and white noise predominates in the total noise power. The input referred equivalent noise bandwidth in closed loop is

$$\Delta f = \int_0^\infty \frac{1}{(1+\frac{1}{\beta A})} df \tag{3.16}$$

which yields, with gain A given by,

$$\Delta f = \frac{\beta}{4\Gamma_u} \tag{3.17}$$

The noise bandwidth is independent of parasitic time constant Γ_p. This can be explained qualitatively by the fact that any reduction of noise at high frequencies due to an increase of Γ_p is compensated by some peaking at lower frequencies.

Total equivalent input noise is then,

$$V_N^2 = 4kTR_{Nt}\Delta f = \frac{\beta^{kT} R_{Nt}}{\Gamma_u} \tag{3.18}$$

Now, by inspection RNt may be expressed as a function of total transconductance g_{mt} given by ,

$$R_{Nt} = \frac{\gamma}{g_{mt}} \tag{3.19}$$

where γ is a factor ranging from nB when noise of pair T1 and T3 predominates to n∗(2B + 1 + B/C) at very low current. Combination of 3.19, 3.18, 3.13 and 3.12 yields,

$$V_N^2 = \frac{\gamma\beta_{kT}}{C_L} \tag{3.20}$$

The amount of noise introduced by a single stage transconductance amplifier is thus inversely proportional to load capacitance C_L and independent of current (except for small possible variations of γ).

Total supply voltage	V_B = 3 V
Total current	I_{tot} = 2.5 μA (I_0 = 0.5 μA)
DC gain	A_0 = 97 dB
Unity gain frequency	f_u = 135 kHz
Slew rate	S.R. = 0.1 V/μs
Total noise above 10 Hz	60 μs
Noise corner frequency	30 kHz
Output swing	2.2 V
Dynamic range	82 dB
Input capacitance	0.7 pF
Current excess factor [28]	CEF = 13
Noise excess factor	NEF = 10 dB

Figure 3-13: Characteristics of a Transconductance Amplifier.

Noise can also be reduced by increasing C_L/B, which will increase settling time T_S. Examination of 3.15 shows that,

$$\frac{\Gamma_u}{\beta} \le \frac{T_S}{\ln \varepsilon^{-1}}$$

which can be introduced in 3.18 to give,

$$V_N^2 \ge \frac{kTR_{Nt}}{T_S} \ln \varepsilon^{-1} \qquad (3.21)$$

The minimum noise for a given settling time T_S depends on R_{Nt} and is thus increased at low current. If current is fixed, according to 3.19, noise is minimum when the input pair is in weak inversion. To achieve high value of voltage gain A_0, cascode transistors must be added to output pair T_7 - T_8.

If low frequency noise is cumbersome, all noisy n-channel transistors can be replaced by compatible bipolars. An equivalent input noisy density below 0.1 $\mu V/\sqrt{Hz}$ for frequencies as low as 1 Hz has been obtained with tail current $I_0 = 2$ μA and minimum sized devices.

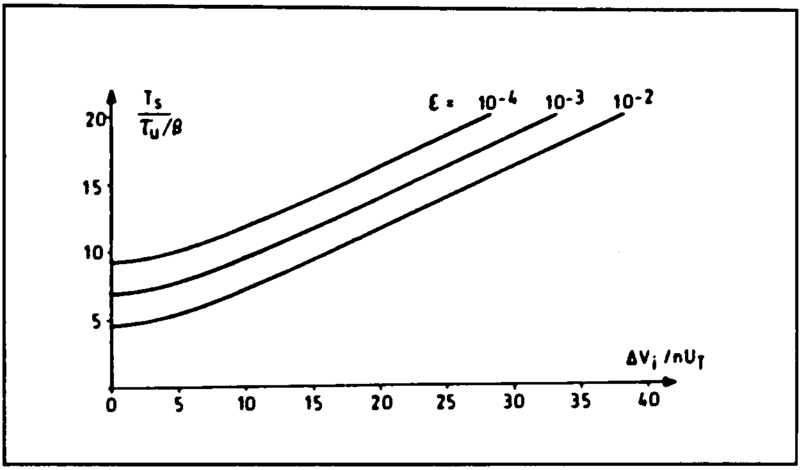

Figure 3.13: Large signal settling time

Due to the weak inversion operation of its input stage, the amplifier is able to settle within a short period of time inspite of its very low power consumption. This is only true for small input steps. Calculation of settling time for large input step ΔV_i and $\Gamma_p << \Gamma_u/\beta$ yields,

$$T_S \cong \frac{\Gamma_u}{\beta} \ln \frac{\sinh(\Delta V_i / 2nU_T)}{\sinh(\varepsilon \Delta V_i / 2nU_T)} \tag{3.22}$$

This relation is plotted on figure 3.13. Settling time is increase by a factor 2 to 3 for $\Delta V_i \cong 1$ V. Better power efficiency is obtained by means of amplifiers that operate in class AB, and therefore able to supply additional current only when it is required.

4.0 CIRCUIT IMPLEMENTATION

4.1 TRANSCONDUCTANCE AMPLIFIER

This is the most important circuit in the model, we will use it for almost everything we do. The amplifier is a device that generates as its output a current that is a function of the difference between the two input voltages, V_1 and V_2, and that difference is called the *differential input* voltage. The circuit is called a *differential transconductance amplifier.* An ordinary conductance turns a voltage difference across two terminals into a current through the same two terminals. A *transconductance* turns a voltage difference somewhere into a current somewhere else. In the transconductance amplifier, a voltage difference between the two inputs creates a current as the output.

4.1.1 DIFFERENTIAL PAIR

Since many circuits take an input signal represented as a difference between two voltages as shown in the figure 4.1, we will first analyze its characteristics and then show how it is used in the transconductance amplifier.

The bottom transistor Q_b is used as a current source. Under normal circumstances, its drain voltage V is large enough that the drain current I_b is saturated at a value set by the gate voltage V_b. The manner in which I_b is divided between Q_1 and Q_2 is a sensitive function of the difference between V_1 and V_2, and is the essence of the operation of the stage.

We know that the saturated drain current Isat is exponential in the gate and source voltages,

$$I_{sat} = I_0 e^{kV_g - V_s}$$

Applying this expression to Q_1 and Q_2, we obtain

$$I_1 = I_0 e^{kV_1 - V} \quad \text{and} \quad I_2 = I_0 e^{kV_2 - V} \tag{4-1}$$

The sum of the drain currents must be equal to I_b,

$$I_b = I_1 + I_2 = I_0 e^{-V}\left(e^{kV_1} + e^{kV_2}\right)$$

We can solve this equation for the voltage V,

$$e^{-V} = \frac{I_b}{I_0}\frac{1}{e^{kV_1} + e^{kV_2}} \qquad (4\text{-}2)$$

Substituting equation 4-2 into equation 4-1, we obtain expressions for the two drain currents,

$$I_1 = I_b \frac{e^{kV_1}}{e^{kV_1} + e^{kV_2}} \quad \text{and} \quad I_2 = I_b \frac{e^{kV_2}}{e^{kV_1} + e^{kV_2}} \qquad (4\text{-}3)$$

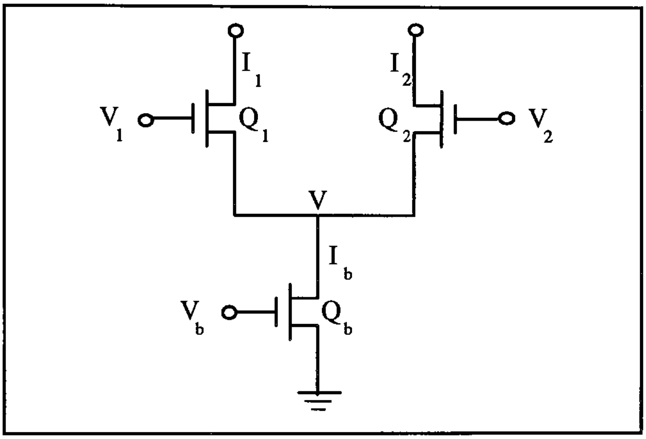

Figure 4.1: Schematic of the differential pair.

If V_1 is more positive than V_2 by many kT/(qK), transistor Q_2 gets turned off, so essentially all the current goes through Q_1, I_1 is approximately equal to I_b, and I_2 is approximately equal to 0. Conversely if V_2 is more positive than V_1 by many kT/(qK), Q_1 gets turned off, I_2 is approximately equal to I_b, and I_1 is approximately equal to 0. The two currents out of a differential pair are shown as a function of $(V_1 - V_2)$ in figure 4-2.

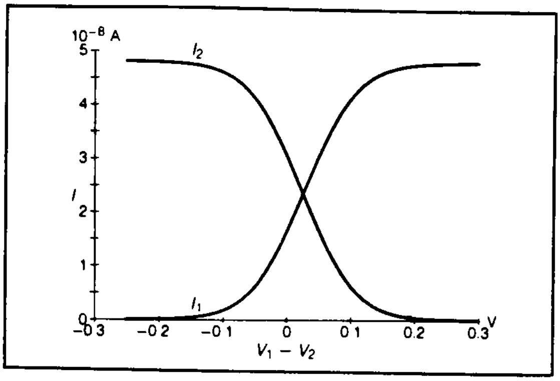

Figure 4-2: Output currents of the differentiator as a function of diff. input voltage. [12]

The differential transconductance amplifier uses various kinds of current mirrors to generate an output current that is proportional to the difference between the two drain currents. This difference is,

$$I_1 - I_2 = I_b \frac{e^{kV_1} - e^{kV_2}}{e^{kV_1} + e^{kV_2}} \tag{4-4}$$

Multiplying both the numerator and denominator of equation 4-4 by $e^{-(V_1+V_2)/2}$, we can express every exponent in terms of voltage differences. The result is,

$$\begin{aligned} I_1 - I_2 &= I_b \frac{e^{k(V_1-V_2)/2} - e^{-k(V_1-V_2)/2}}{e^{k(V_1-V_2)/2} + e^{-k(V_1-V_2)/2}} \\ &= I_b \tanh \frac{k(V_1 - V_2)}{2} \end{aligned} \tag{4-5}$$

The tanh is one of the few functions which follows a normal behavior. It goes through the origin with unity slope, becomes +1 for large positive arguments, and becomes -1 for large negative arguments. Let us observe as to what happens when only small changes are made in V_1 and V_2. We increase V_1 and decrease V_2 such that V is kept constant. The current through Q_2 goes down exponentially and the current through Q_1 goes up exponentially. The difference in voltages, however, is twice as large as V_2 relative to V, or as V_1 relative to V. That is why the curves of figure 4-2 take twice as much voltage to saturate as do the single transistor curves.

4.2 SIMPLE TRANSCONDUCTANCE AMPLIFIER

The schematic for the transconductance amplifier is shown in the figure 4.3. The circuit consists of a differential pair and a single current mirror, which is used to subtract the drain currents I_1 and I_2. The current I_1 drawn out of Q_3 is reflected as an equal current out of Q_4. The output current is thus equal to $(I_1 - I_2)$, and is therefore given by equation 4.5.

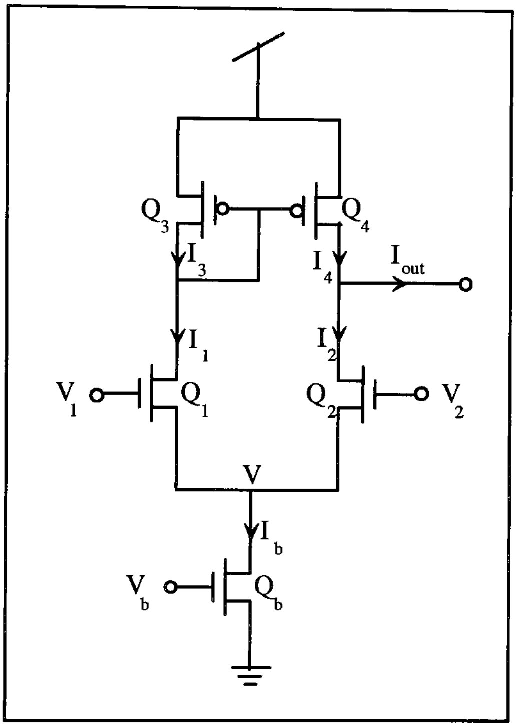

Figure 4-3: Schematic of the simple transconductance amplifier. [12]

The current out of the simple amplifier is plotted as a function of V_1-V_2 in figure 4.4. The curve is very close to a tanh, as expected. The layout of the simple amplifier is as shown in figure 4.5. We can determine the effective value of kT/(qK) by extrapolating the slope of the curve at the origin to the two asymptotes. In terms of the circuit variables,

$$G_m = \frac{\partial I_{out}}{\partial V_{in}} = \frac{I_b}{2kT/(qk)} \tag{4-6}$$

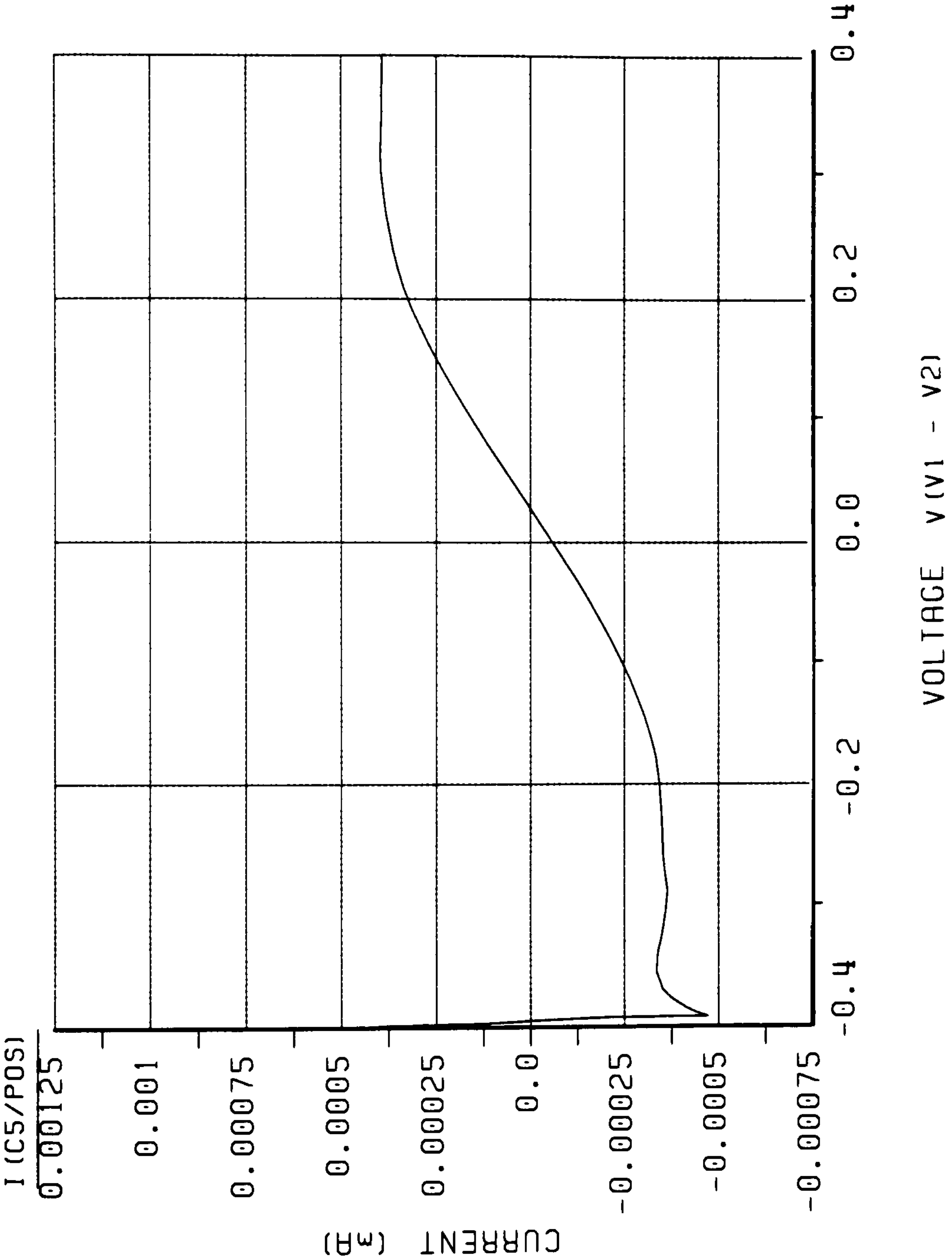

Figure 4-4: Output Current of the Amplifier as a Function of Differential Input Voltage.

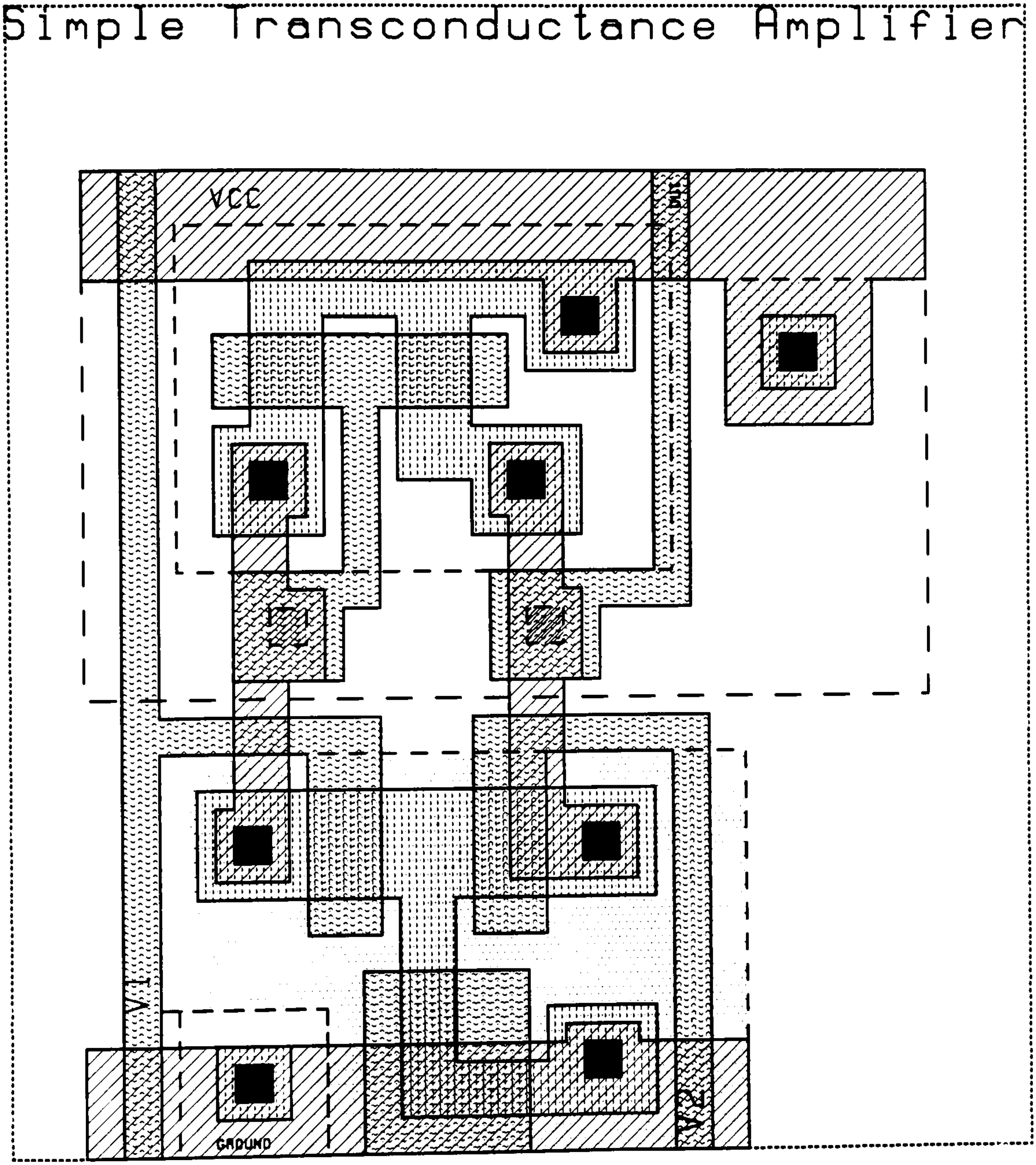

Figure 4-5: Layout of the Simple Transconductance Amplifier.

4.2.1 CIRCUIT LIMITATIONS

Deviations from ideal behavior are of two basic sorts,

(1) Mismatch between transistors

(2) Deviation of a transistor from perfect current - source behavior, this second class of non-ideality is further classified into two ways as,

(a) Voltage limitations due to transistors coming out of saturation

(b) Finite slope of the drain curves in saturation

4.2.2 TRANSISTOR MISMATCH

In general all transistors are not equal, some have higher values of I_0 than others. The tanh curve is shifted by about 25 millivolts. In addition, the saturated current coming out of Q_4 is not the same as the current coming out of Q_2. In other words, the negative asymptote is not the same as the positive asymptote. In figure 4.4, the difference is about 6 %.

The Q_3 - Q_4 current mirror does not have 100 percent reflectivity. What we take out of Q_3 does not necessarily come out of Q_4, because Q_4 may have a slightly larger or smaller value of I_0 than does Q_3. Differences of a factor of two between I_0 values of nominally identical transistors are observed in such circuits.

4.2.3 UPPER LIMIT

It is expected, that this device will not be able to put out a constant current at voltages larger than V_{DD} or smaller than zero. This limitation is important, for it means that the circuit cannot generate a voltage for the next stage that is outside those limits. If we raise the output voltage above V_{DD}, the drain of Q_4 becomes the source, and we start draining current out of the ammeter up through Q_4 to V_{DD}. Q_4 will be turned on, because the voltage on its gate is less than V_{DD}, because that voltage is generated by Q_3. Even if Q_1 is turned off, the worst that can happen is that the voltage on the gate of Q_4

will approach V_{DD}. So, if V_{out} gets a few tenth's of a volt above V_{DD}, we will start to get an exponential negative increase in I_{out}.

4.2.4 LOWER LIMIT

We have to be a bit careful with the lower limit for V_{out}. Let us first consider the case where V_1 is greater than V_2 by several kT/(qK). Under these conditions, V is approximately equal to $k(V_1 - V_b)$, I_2 is approximately equal to 0, and I_{out} is positive, approximately equal to I_4. As we lower the output voltage, all is well until V_{out} decreases to less than V, after which the output node becomes the source of Q_2, and the V node becomes the drain. The interchange of source and drain of Q_2 results in a reversal of current through Q_2. I_2 becomes negative instead of positive. The reversal occurs when V_{out} is equal to $k(V_1 - V_b)$, but is not noticeable in the output current until V_{out} is approximately equal to $k(V_2 - V_b)$, where I_2 becomes comparable with I_1. A further decrease in output voltage results in an exponential increase in I_{out}, because the gate source voltage of Q_2 is increasing. This negative I_2 is supplied by an increase in I_1, which results in an equal increase in output current through Q_4. The output current thus increases from two equal contributions of the same sign.

If V_2 is greater than V_1 by several kT/(qK), the same effect can be observed. The output current is negative, and V is equal to $k(V_2 - V_b)$. As we decrease the output voltage, we make the voltage between the source and the drain of Q_2 smaller and smaller, Q_2 comes out of saturation, and V begins to decrease. As both V_{out} and V decrease, the gate source voltage of Q_2 increases, causing Q_2 to conduct more current. The voltage V follows V_{out} more and more closely. There is a noticeable change in output current, however, until V approaches $k(V_1 - V_b)$, at which point the current through Q_1 becomes comparable to I_b. As we decrease the voltage at the output node further, I_1 exceeds I_b, and V does not decrease as fast as does V_{out}. Once V is greater than V_{out}, the drain and source of Q_2 are interchanged, and the situation is exactly as it was for V_2 greater than

V_1. Transistor Q_2 starts taking charge away from the V node, and the output current increases exponentially.

This limitation on the operation of the simple transconductance amplifier imposed by this behavior is sometimes referred to as the V_{min} *problem.* We can express the minimum output voltage as the,

$$V_{\min} = k(\min(V_1, V_2) - V_b) \tag{4-7}$$

In other words, the amplifier will work with its output voltage up to nearly V_{DD}, and down to V_b below the lowest input signal that we have applied to it, but not lower than that.

We run into two walls, one on the top and one on the bottom. The wall on the top side is not serious, all it does is to prevent us from going right up to V_{DD}. When V_1 is greater than V_2, the current comes out of Q_4, so, if we make the output node equal to V_{DD}, we will not get any current out. We cannot quite work up against the rail, but we can get close. As long as we stay a few kT/(qK) below V_{DD}, we are fine.

The bottom V_{min} limit is much more serious. It is the biggest problem with this circuit. It forms a hard limit below which the circuit does not work, and that limit depends on the input voltage.

4.2.6 VOLTAGE OUTPUT

The transconductance amplifiers can also be used to take a difference in voltage at the input, and turn it into a voltage at the output. Instead of measuring I_{out} with an ammeter, we measure V_{out} with a voltmeter. The drain conductance of Q_2 and Q_4 are used to convert the output current into an output voltage.

The drain current of a transistor is not completely independent of its drain voltage, even in saturation. There is a finite slope of I_d versus V_d given by the *Early effect* [12]. This effect is responsible for the dependence of output current on output voltage seen between the two limits in figure 4.6.

4.2.7 VOLTAGE GAIN

The voltage gain A is defined as $\partial V_{out}/\partial V_{in}$, where V_{in} is equal to $V_1 - V_2$. An enlargement of the steep part of the V_2 = 2.5 volt curve in figure 4-8 is shown in figure 4.9.

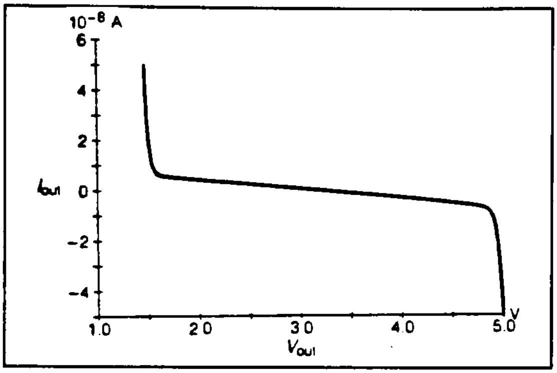

Figure 4.6: Current out of Q2-Q4 output transistors as a function of output voltage. [12]

An enlargement of the intersection of the Q_2 and Q_4 drain curves in figure 4-6 is shown in figure 4-9. For a certain input voltage difference, the curves are marked I_2 and I_4. When the input voltage difference is increased by Δv, both curves change. I_2 decreases to I'_2 and I_4 increases to I'_4. Because the bias current I_b is constant, an increase ΔI in I_4 due to a change in input voltage will result in an equal decrease ΔI in I_2, as shown. The total change in output current per unit change in input voltage difference was defined in equation 4-6 as the transconductance G_m of the circuit.

When the output is open circuited, the total increase 2ΔI in output current due to an increase ΔV in input voltage difference is compensated by an equal decrease in output current due to the increase ΔV in the output voltage.

$$2\Delta I = \frac{\partial I_{out}}{\partial V_{in}} \Delta v = -\frac{\partial I_{out}}{\partial V_{out}} \Delta v \qquad (4\text{-}8)$$

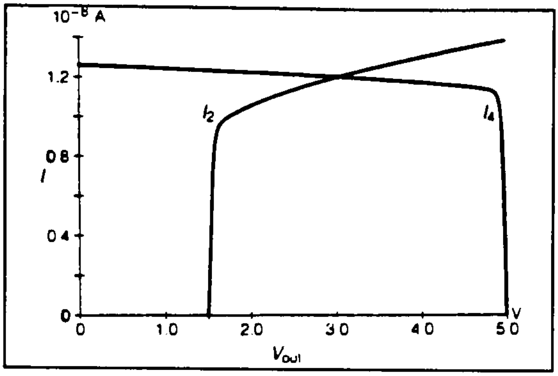

Figure 4-7: Open circuit output voltage of the amp. as a function of V_1, for different V_2.

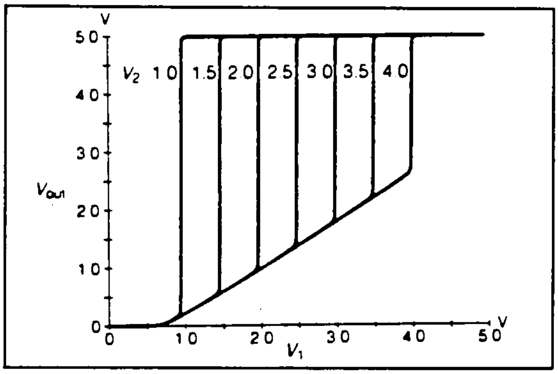

Figure 4.8: Expanded view of the center curve for V2 equal to 2.5 volts. [17]

4.3 WIDE RANGE AMPLIFIER

A simple transconductance amplifier will not generate output voltages below V_{min}, which, in turn, is dependent on the input voltages. This limitation often is a source of problems at the system level, because it is not always possible to restrict the range of input voltages. We can remove this restriction, however, by a simple addition to the circuit, as shown in the figure 4-9. The simulated result and layout is as shown in figure 4-10 and 4-11.

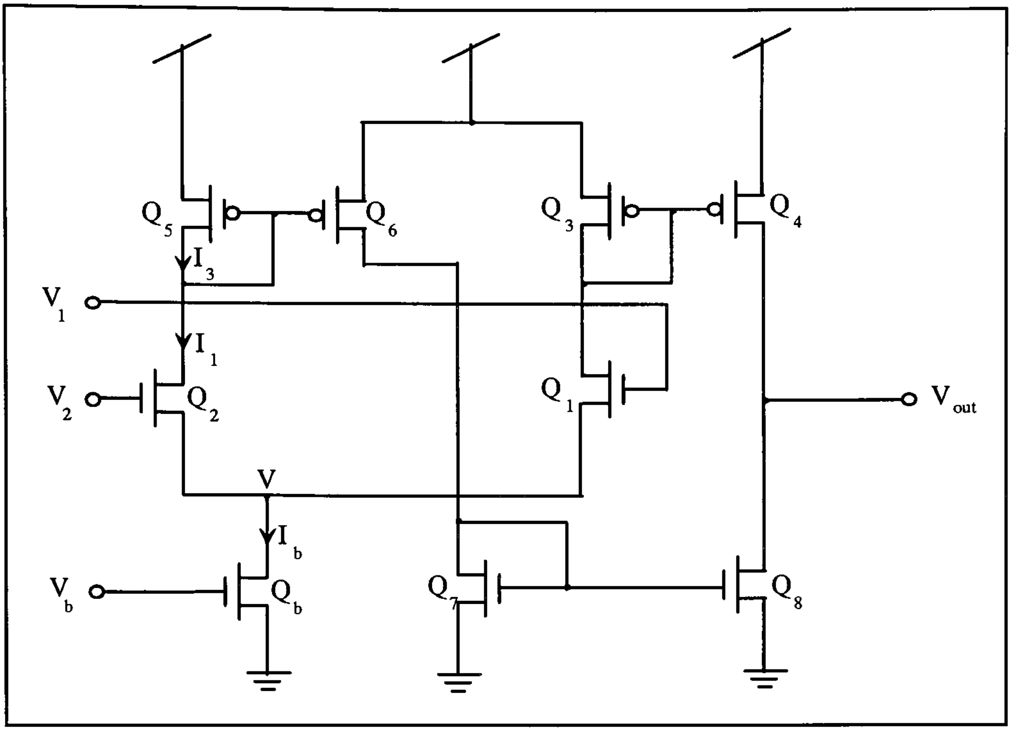

Figure 4-9: Schematic of the Wide-Range Transconductance Amplifier. [12]

Instead of feeding the output directly, the drain of Q_2 is connected to the current mirror formed by Q_5 and Q_6. The current coming out of Q_4 and Q_6 now are just the two halves of the current in the differential pair. We then reflect the Q_6 current one more time, through Q_7 and Q_8, and subtract it from I_4 to form the output. As in the simple circuit, the output current is just the difference between I_1 and I_2.

The major advantage of the wide range amplifier over the simple circuit is that both the input and output voltages can run almost upto V_{DD} and almost down to ground, without affecting the operation of the circuit. In other words, the problem of V_{min} seems to be eliminated.

The other nice thing about this circuit is that the current mirrors, such as Q_3 and Q_5, hold the drain voltages of Q_1 and Q_2 very nearly constant. In diode connected transistors, the current increases exponentially with the gate voltage, so the drain voltage never gets very far below V_{DD}. For that reason Q_2 no longer has a problem associated with its drain conductance, its source drain voltage is nearly equal to that of Q_1. So the drain conductance of Q_1 and Q_2 are not critical in this circuit. The same thing is true of Q_6, Q_7 is a diode connected transistor, it holds the drain voltage of Q_6 very nearly constant. The only transistors that work over a large voltage range are Q_b, Q_4, and Q_8, and we can make their channels long to get a low drain conductance i.e. the output current that is nearly independent of the output voltage. Because of their low output conductance, long Q_4 and Q_8 transistors give the circuit a high voltage gain. Such wide range amplifiers have about 10 times the gain of the simple amplifier, and they work all the way down to ground and all the way up to V_{DD}.

4.4 LOGARITHMIC AMPLIFIER

We know that a diode connected transistor creates a voltage that is proportional to the logarithm of the input current. This voltage can be used to control the output currents of other transistors, but it is below the range of usable inputs for circuits such as the transconductance amplifiers. A voltage that is well within the operating range of these circuits can be generated by two diode connected transistors in series, as show in the figure 4-12(a). The inverse operation, creating a current proportional to the exponential as a voltage, is accomplished by the circuit of figure 4-12(b). The relationship between voltage and current for these circuits is shown in figure 4-13.

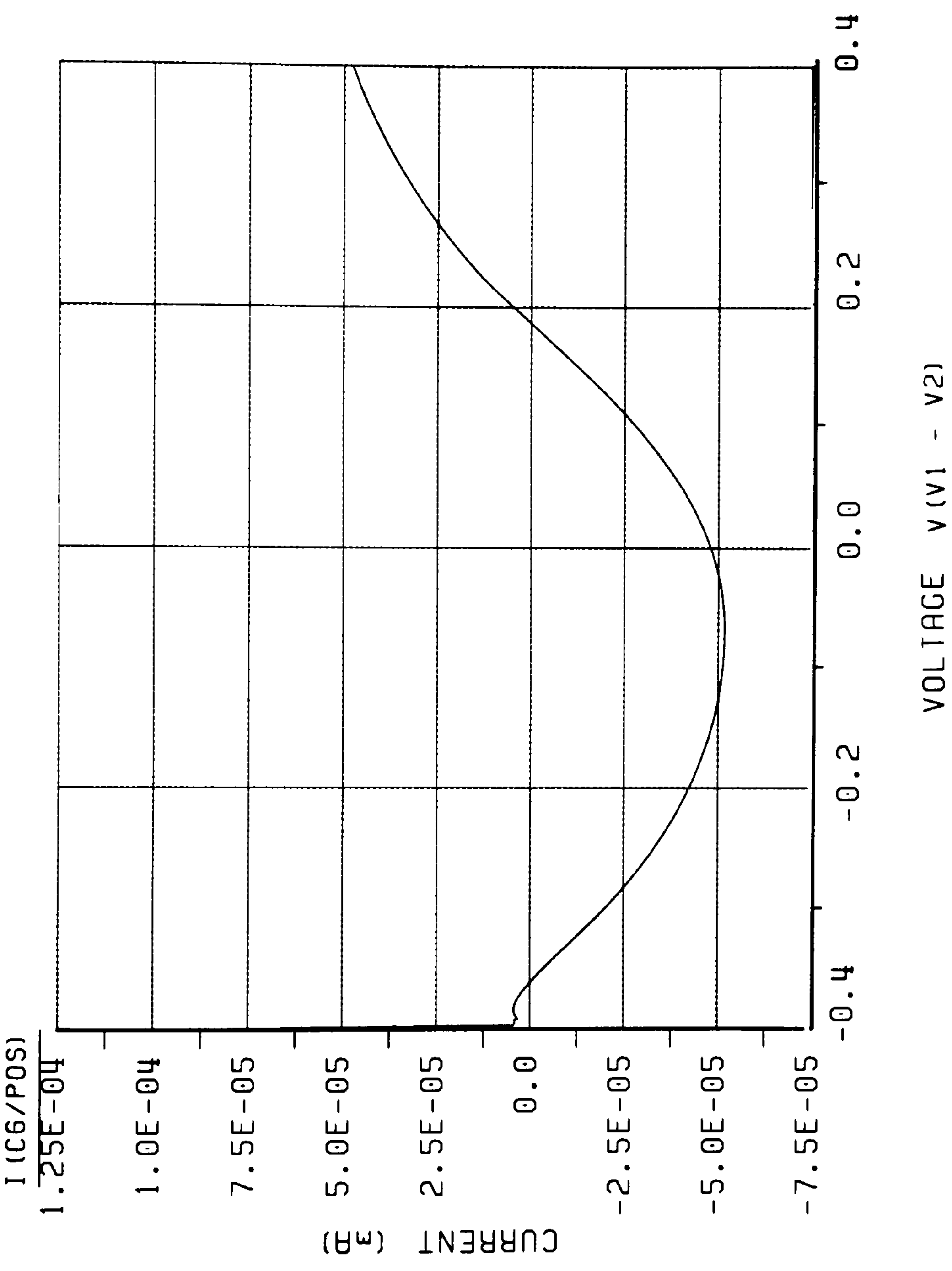

Figure 4-10: Output Current of the Amplifier as a Function of Differential Input Voltage

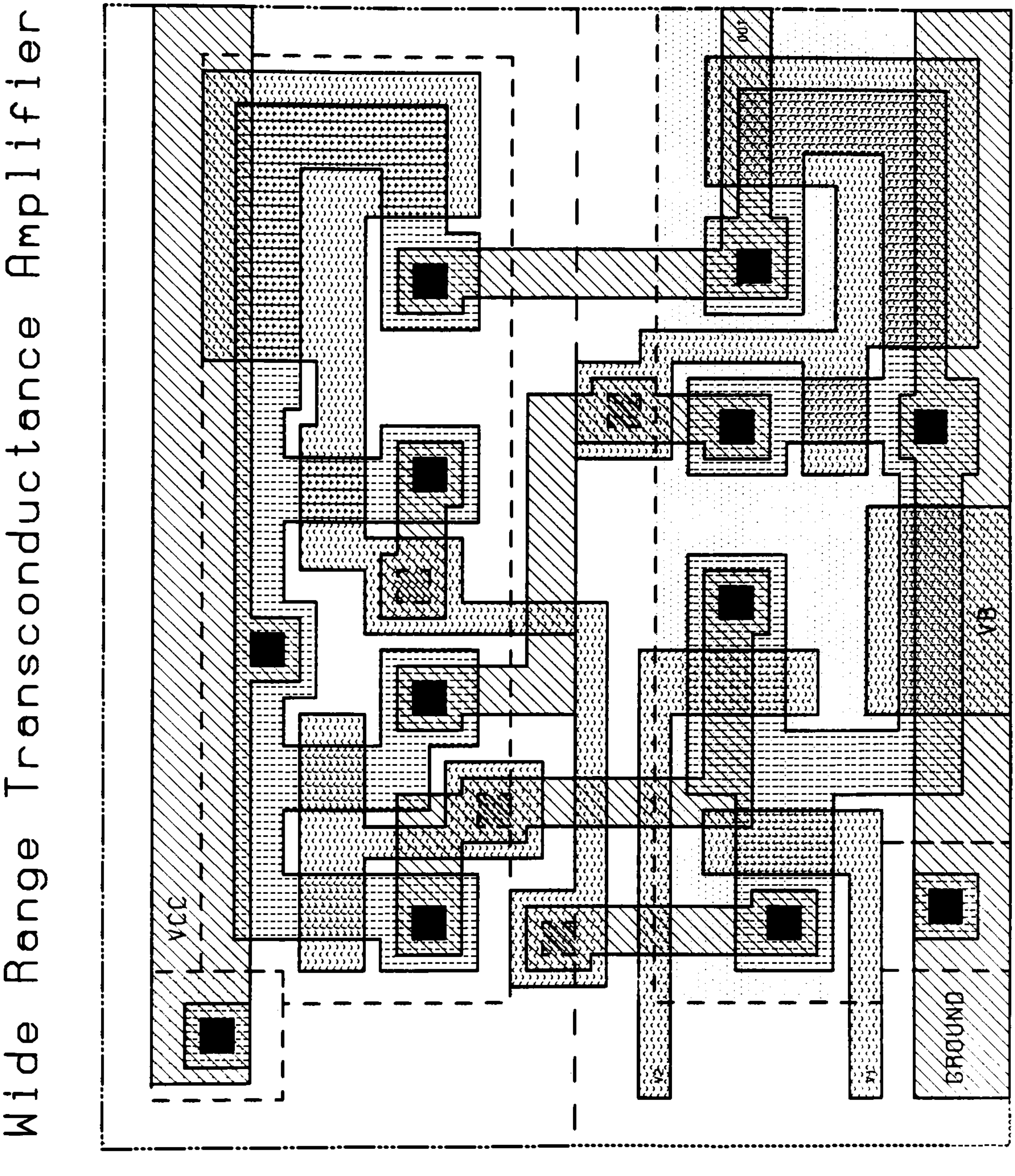

Figure 4-11: Layout of the Wide-Range Transconductance Amplifier.

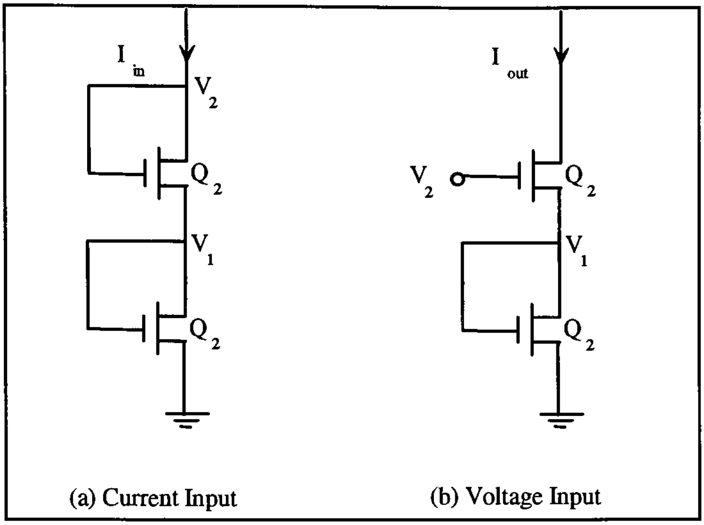

Figure 4-12: Two circuits that use the natural logarithmic voltage current characteristics of the MOS transistor. The current input version (a) generates an output voltage that is proportional to the logarithm of the input current. The voltage input version (b) generates an output current that is exponentially related to the input voltage.

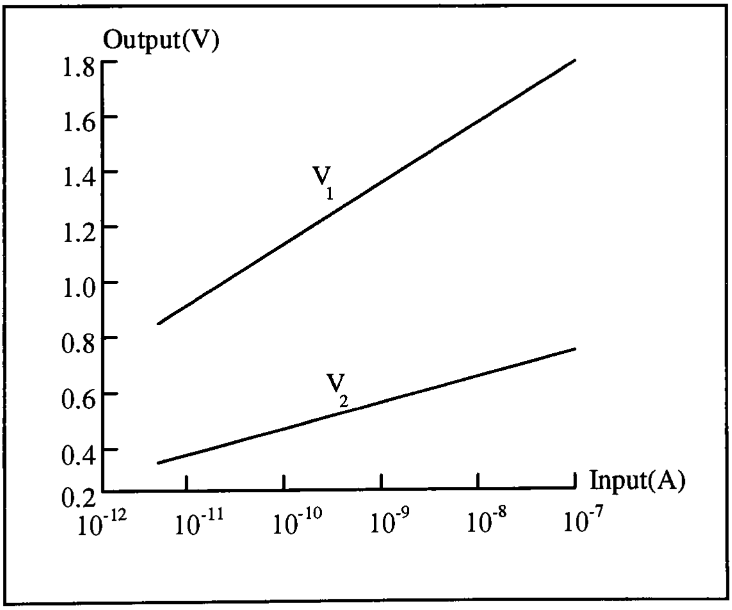

Figure 4-13: Relationship between voltage and current for circuit in figure 4-12.

For an n-channel transistor,

$$I = I_0 e^{KV_g}(e^{-V_s} - e^{-V_d}) = I_{sat}(1 - e^{-V_{ds}})$$

We know that the saturated drain current Isat is exponential in the gate source voltage Vgs,

$$I_{sat} = I_0 e^{kV_g} e^{-V_s}$$

Applying this expression to Q1 and Q2, we obtain,

$$I = I_0 e^{kV_1} = I_0 e^{kV_2 - V_1} \qquad (4\text{-}10)$$

Taking logarithms of the last two terms,

$$V_2 = \frac{k+1}{k} V_1$$

From which we conclude,

$$\ln \frac{I}{I_0} = \frac{k^2}{k+1} V_2 \qquad (4\text{-}11)$$

In the ideal case where k is equal to one, equation 4-11 has the solution,

$$I = I_0 e^{\frac{V_2}{2}}$$

and we would expect the slope of the upper curve of figure 4-13 to be twice that of the lower curve.

4.4.1 PHOTORECEPTOR CIRCUIT

The primary function of the photoreceptor is to transduce light into an electrical signal. For intermediate levels of illumination, this signal is proportional to the logarithm of the incoming light intensity. The logarithmic nature of the output of the biological photoreceptor is supported by psychological and electrophysiological evidence. Psychological investigations of human visual sensitivity thresholds show that the threshold increment of illumination for detection of a stimulus is proportional to the background illumination over several orders of magnitude. Physiological recordings show that the photoreceptor's electrical response is logarithmic in light intensity over the central part of the photoreceptor's range, as are the responses of other cells in the distal retina. The logarithmic nature of the response has two important system level consequences,

(1) An intensity range of many orders of magnitude is compressed into a manageable excursion in signal level.

(2) The voltage difference between two points is proportional to the *contrast ratio* between the two corresponding points in the image. In a natural image, the contrast ratio is the ratio between the reflectance's of two adjacent objects, reflectance's which are independent of the illumination level.

The silicon photoreceptor circuit of a photoreceptor, which transduces light falling onto the retina into an electrical photo current, and a logarithmic element, which converts the photo current into an electrical potential proportional to the logarithm of the local light intensity. Our photodetector is a *vertical bipolar transistor.* The base of the transistor is an isolated section of well, the emitter is a diffused area in the well, and the collector is the substrate. Photons with energies greater than the band gap of silicon create electron hole pairs as they are absorbed. Electrons are collected by the n-type base of the pnp phototransistor, thereby lowering the energy barrier from emitter to base, and increasing the flow of holes from emitter to collector. The gain of this process is determined by the number of holes that can cross the base before one hole recombines with an electron in the base. The photodetector in this silicon photoreceptor produces several hundred electrons for every photon absorbed by the structure.

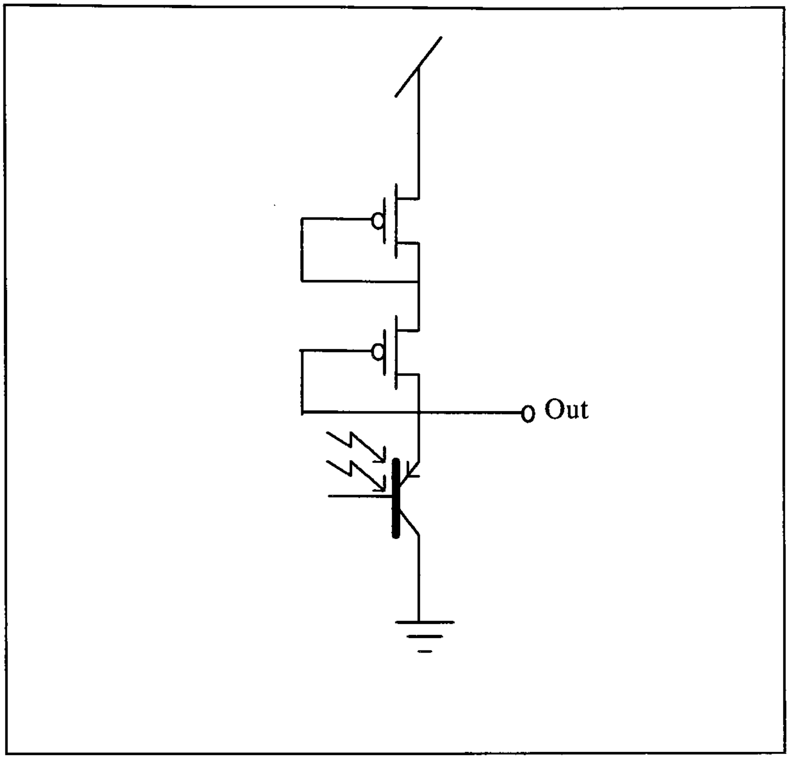

Figure 4-14: Measured response of a logarithmic photoreceptor.

The current from the photocurrent is fed into the two diode-connected MOS transistors in series. It produces a voltage proportional to the logarithm of the current, an therefore to the logarithm of the incoming intensity. We use two transistors to ensure that, under normal illumination conditions, the output voltage will be within the limited allowable voltage range of the resistive network. Even so, at very low light levels, the output voltage of the photoreceptor may be close enough to V_{DD} that the resistor bias circuit cannot adequately bias the horizontal resistive connections.

The voltage out of this photoreceptor circuit is logarithmic over four to five orders of magnitude of incoming light intensity, as shown in the figure 4-15. The layout of this circuit is as shown in figure 4-16.

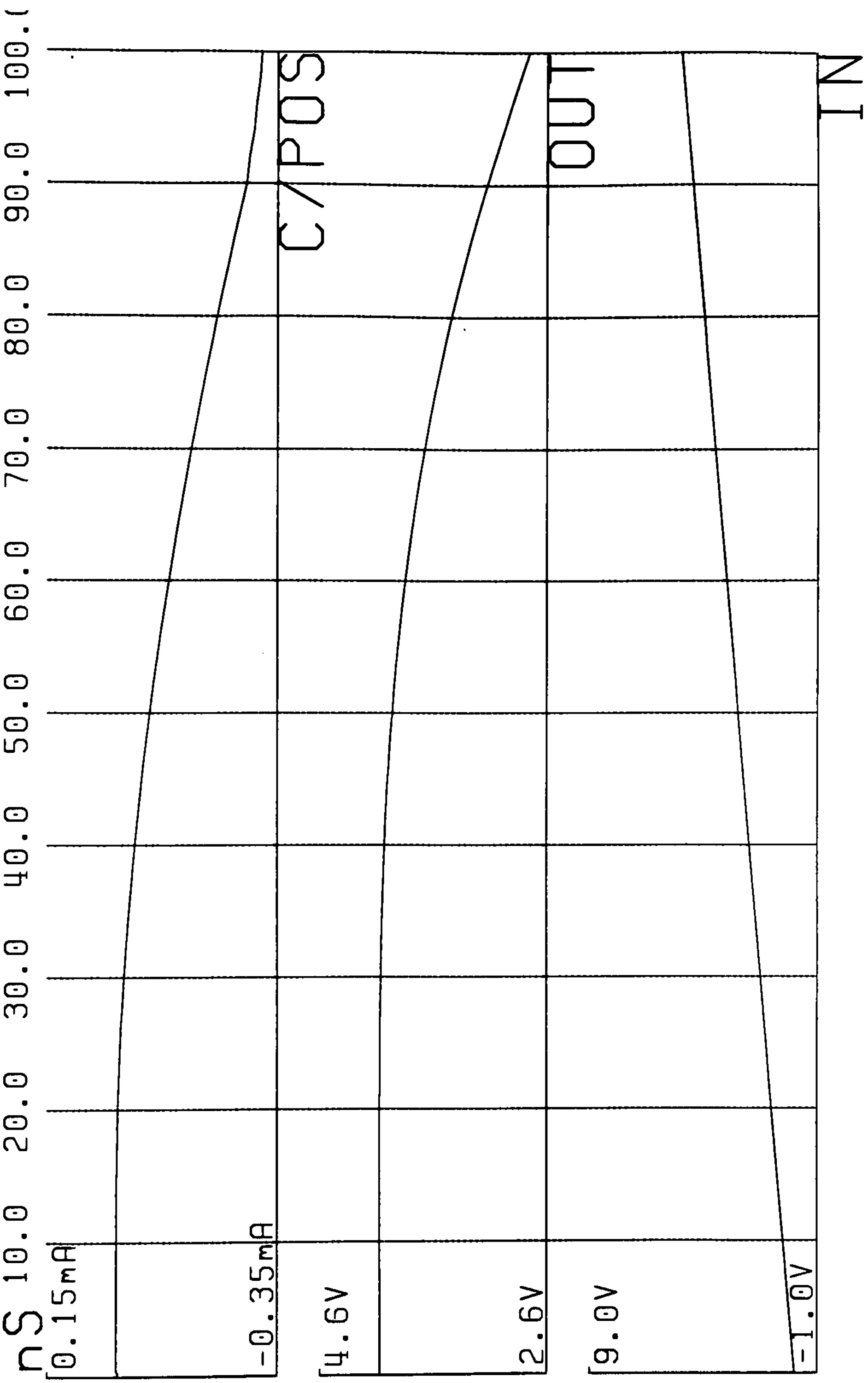

Figure 4-15: Response versus Intensity Curve of a Logarithmic Photoreceptor

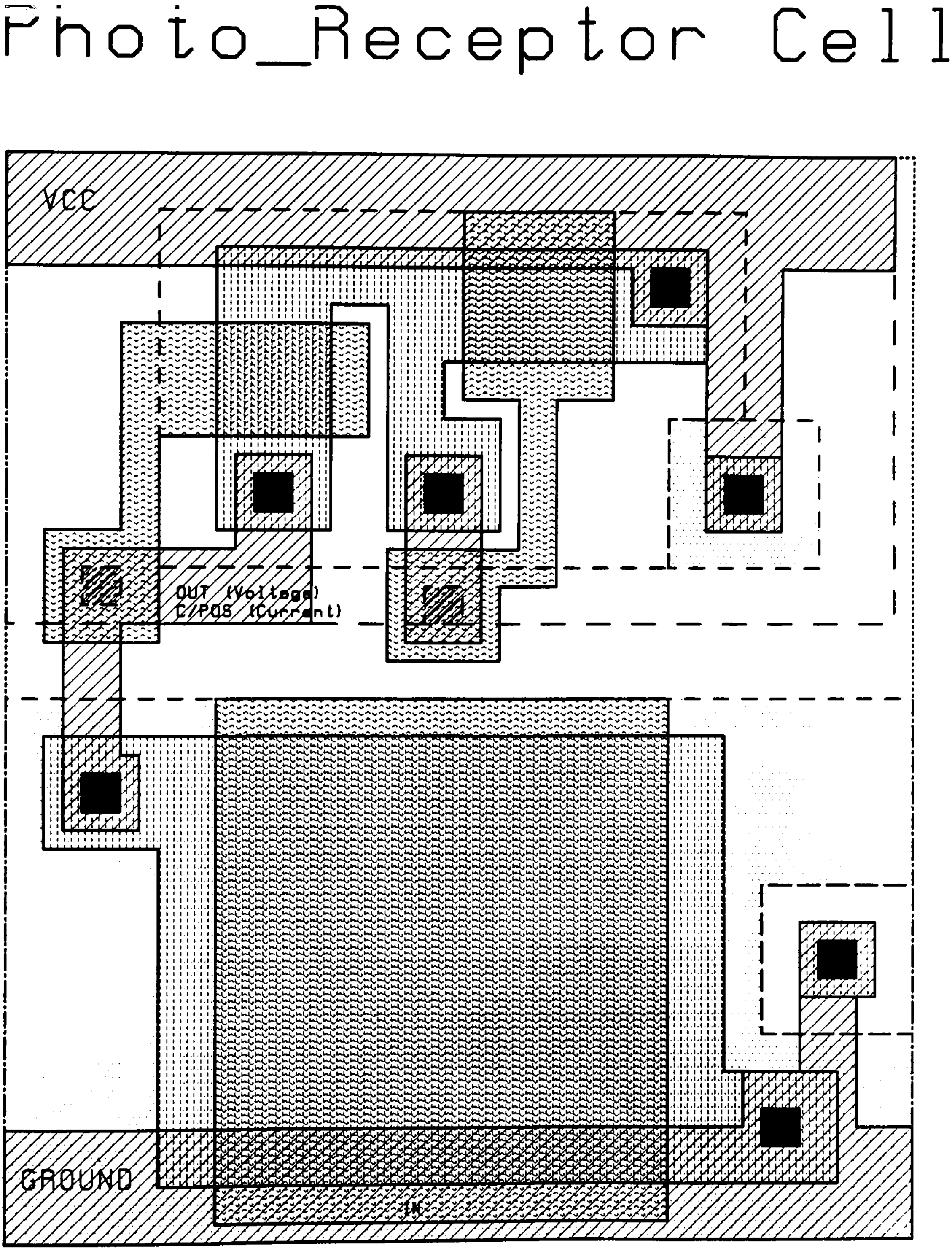

Figure 4-16: Layout of the Photoreceptor Cell.

4.5 BIAS CIRCUIT

For the kind of resistive connection shown in figure 4-17(a), we have to find a way to implement the V_q bias voltage sources. The bias voltage generator should adjust the value of V_q such that the saturation current of the resistive connection can be set by an external control, but not vary as the voltage level in the network changes.

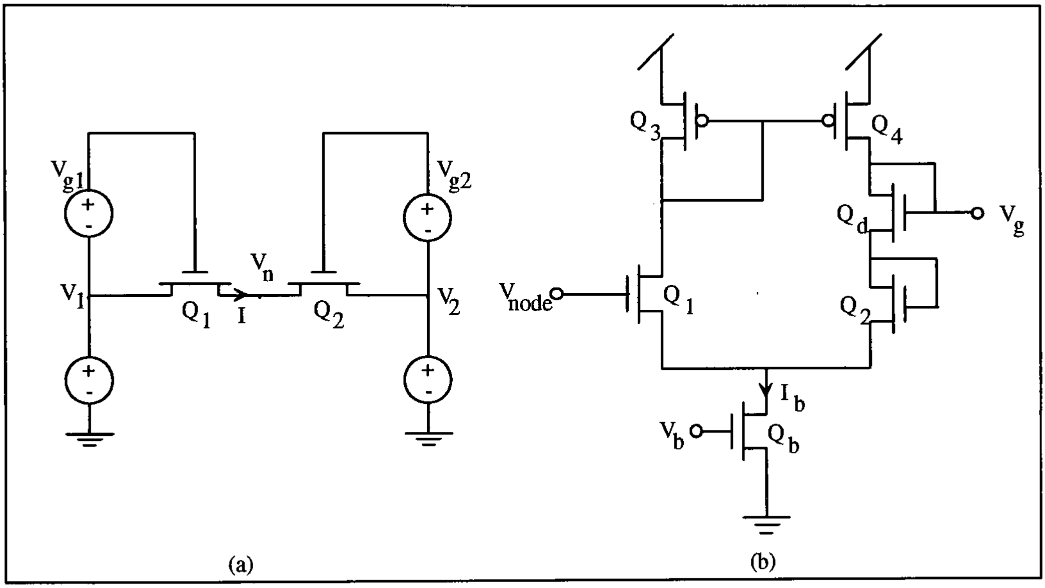

Figure 4-17: Schematic of the bias circuit for the horizontal resistor circuit. [17]

A biasing circuit that achieves these properties is shown in figure 4-17(b). The node labeled V_{node} senses the network voltage at a network node, for example, V_1, and the circuit generates an output voltage V_g to bias the gates of all pass transistors connected to that node. This is nothing but an ordinary transconductance amplifier connected as a follower, with the addition of the diode connected transistor Q_d. Because of the follower action, the voltage at the gate of Q_2, which is connected to the source of Q_d, follows the node voltage V_{node}. The output voltage V_g will follow the node voltage, but with a positive offset equal to the voltage across Q_d.

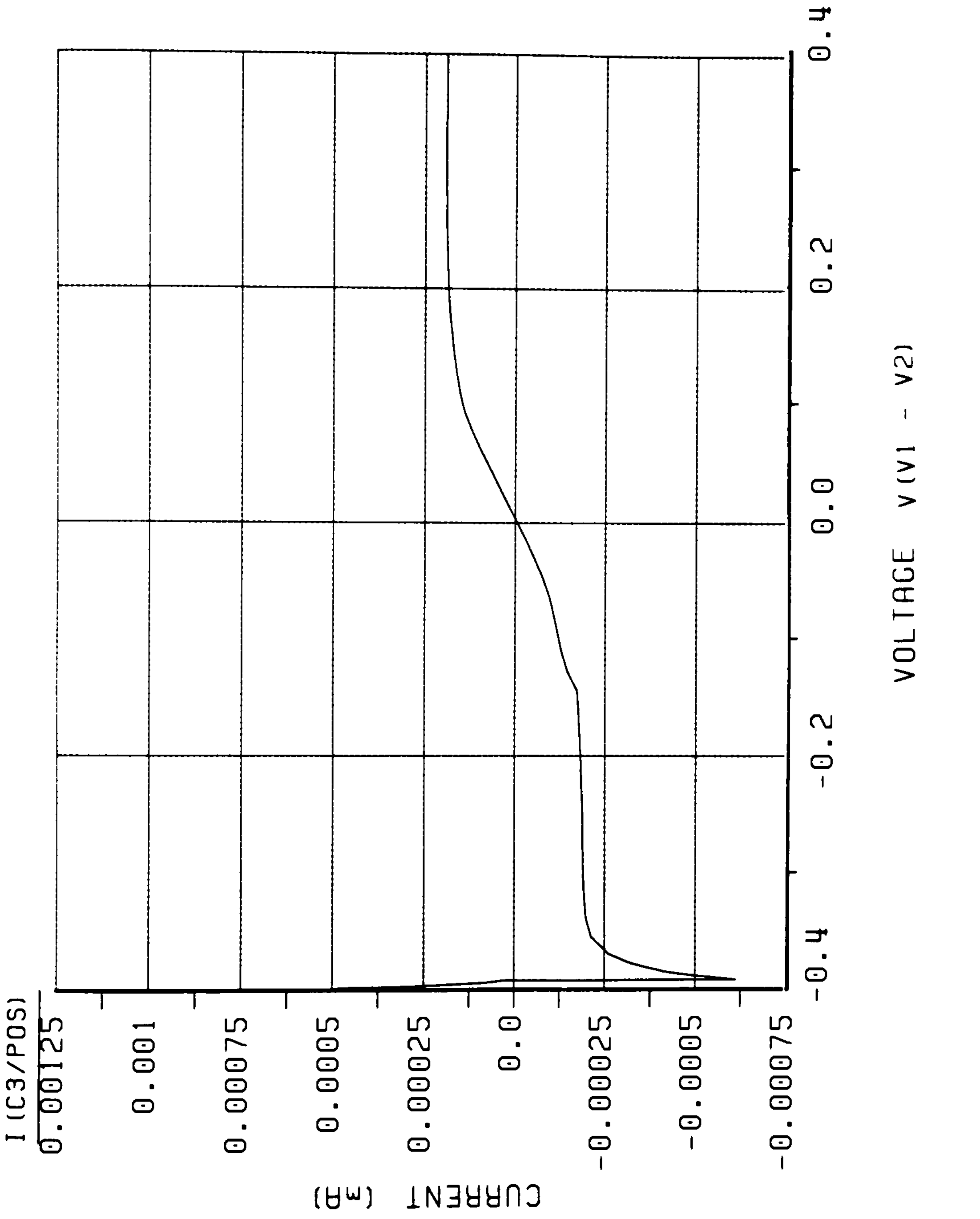

Figure 4-18: Measured Current-Voltage Characteristics of Horizontal Resistor Circuit

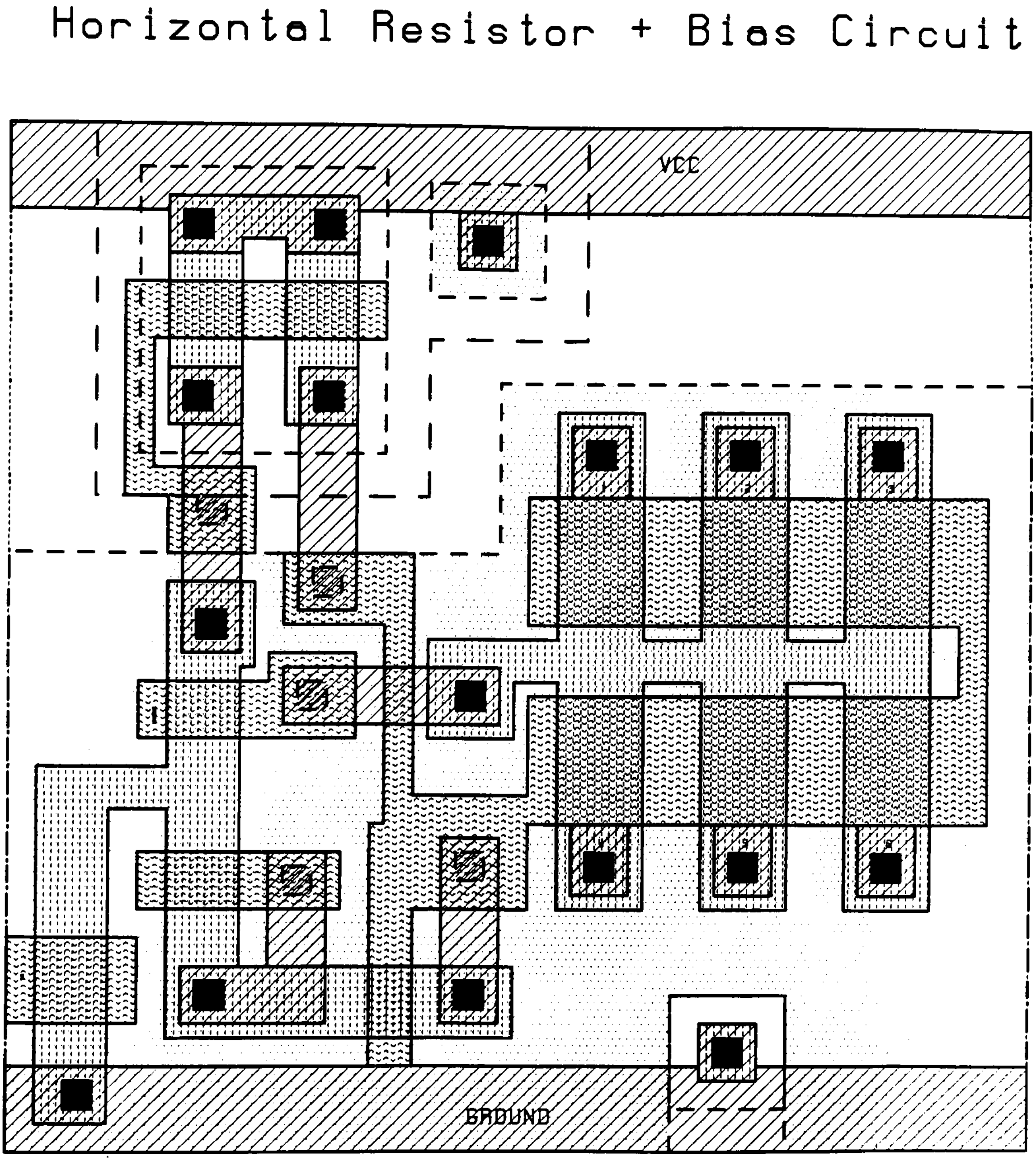

Figure 4-19: Layout of the Horizontal Resistor Circuit.

The diode connected transistor Q_d has both its source and gate voltages equal to those of the pass transistor. Half of the bias current I_b is flowing in Q_d. Writing equation for Q_d in each of the bias sources, we obtain,

$$\frac{I_b}{2} = e^{kV_{g1}-V_1} = e^{kV_{g2}-V_2}$$

Thus, the saturation current of Q_d will be the saturation current of the resistive connection, independent of the node voltage. We have accomplished this without drawing any current out of the network. The bias current I_b serves two purposes in this circuit. First it enables the follower to operate, and second it biases the diode connected transistor Q_d. The voltage across Q_d, and hence the gate source voltage of the pass transistor, is set by the bias current. We therefore can use I_b to control the conductance of the resistive connection.

The measured current voltage curve for the horizontal resistor circuit is shown in the figure 4-18. The layout for the horizontal circuit is as shown in figure 4-19. The current is linear with voltage across the resistor for differential voltages less than approximately ± 100 millivolts, and saturates at Isat for larger voltages. The negative saturation current is not equal to the positive saturation current, due to the mismatch between transistors in the bias circuit on the right. In spite of this mismatch, the current flowing from one circuit to the other, except for leakage current of the source and drain regions to substrate, is guaranteed to pass through zero at zero voltage. The leakage currents usually are negligible compared with Isat.

The key processing elements in the outer- plexiform layer is the triad synapse, which is found in the base of the photoreceptor. The triad synapse is the point of contact among the photoreceptor, the horizontal cells, and the bipolar cells. We can describe our model of the computation performed at the triad synapse in terms of the synapse's three elements,

(1) The photoreceptor takes the logarithm of the intensity

(2) The horizontal cells form a resistive network that spatially and temporally averages the photoreceptor output.

(3) The bipolar cell's output is proportional to the difference between the photoreceptor signal and the horizontal cell signal.

4.6 HORIZONTAL RESISTIVE LAYER

The retina provides an excellent example of the computation that can be performed using a resistive network. The horizontal cells in most species are connected to one another by gap junctions to form an electrically continuous network in which signals propagate by electronic spread. The lateral spread of information at the outer-plexiform is thus mediated by the resistive network formed by the horizontal cells. The voltage at every point in the network represents a spatially weighted average of the photoreceptor inputs. The farther away an input is from a point in the network, the less weight it is given. The horizontal cells usually are modeled as passive cables, in which the weighing function decreases exponentially with distance.

Our silicon retina includes the passive resistive network, patterned after the horizontal cells of the retina. Each photoreceptor in the network is linked to its six neighbors with resistive elements, to form the hexagonal array shown in figure 3.5. Each node of the array has a single bias circuit to control the strength of the six associated resistive connections. The photoreceptors act as voltage inputs that drive the horizontal network through conductance. This method of providing input to a resistive network is shown in figure 3.9. By using a wide range amplifier in place of a bi-directional conductance, we have turned the photoreceptor into an effective voltage source. No current can be drawn from the output node of the photoreceptor, because the amplifier input is connected to only the gate of a transistor.

The horizontal network computes a spatially weighted average of photoreceptor inputs. The spatial scale of the weighting function is determined by the product of the lateral resistance and the conductance coupling the photoreceptors into the network. Varying the conductance of the wide range amplifier or the strength of the resistors changes the space constant of the network, and thus changes the effective area over which the signals are averaged.

Both biological and silicon resistive networks have associated parasitic capacitances. The integrated resistive element in our case have an unavoidable capacitance to the silicon substrate, so they provide the same kind of time integration as do their biological counterparts. The effects of delay due to electronic propagation in the network are most apparent when the input image changes suddenly.

4.7 TRIAD SYNAPSE COMPUTATION

The receptive field of the bipolar cell shows an averse center surround response. The center of the bipolar cell receptive field is excited by the photoreceptors, whereas the surround is due to the horizontal cells. In this model, the center surround computation is a result of the interaction of the photoreceptors, the horizontal cells, and the bipolar cells in the triad synapse.

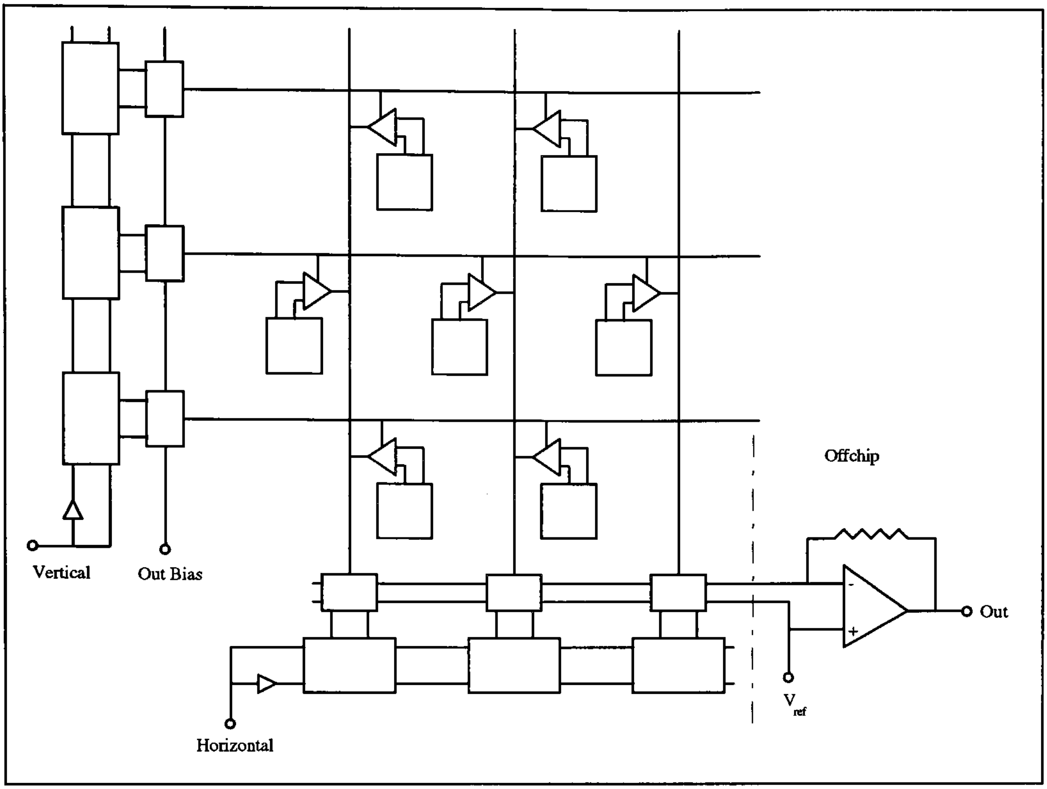

Figure 4-20: Schematic Layout of the retina chip.

The output of the silicon retina is analogous to the output of a bipolar cell in a vertebrate retina. Our triad synapse consists of two elements,

(1) A wide range amplifier provides a conductance through which the resistive network is driven towards the photoreceptor output potential.

(2) A second amplifier senses the voltage difference across the conductance, and generates an output proportional to the difference between the photoreceptor output and the network potential at that location.

The output of the bipolar cell thus represents the difference between a center intensity and a weighted average of the intensities of surrounding points in the image.

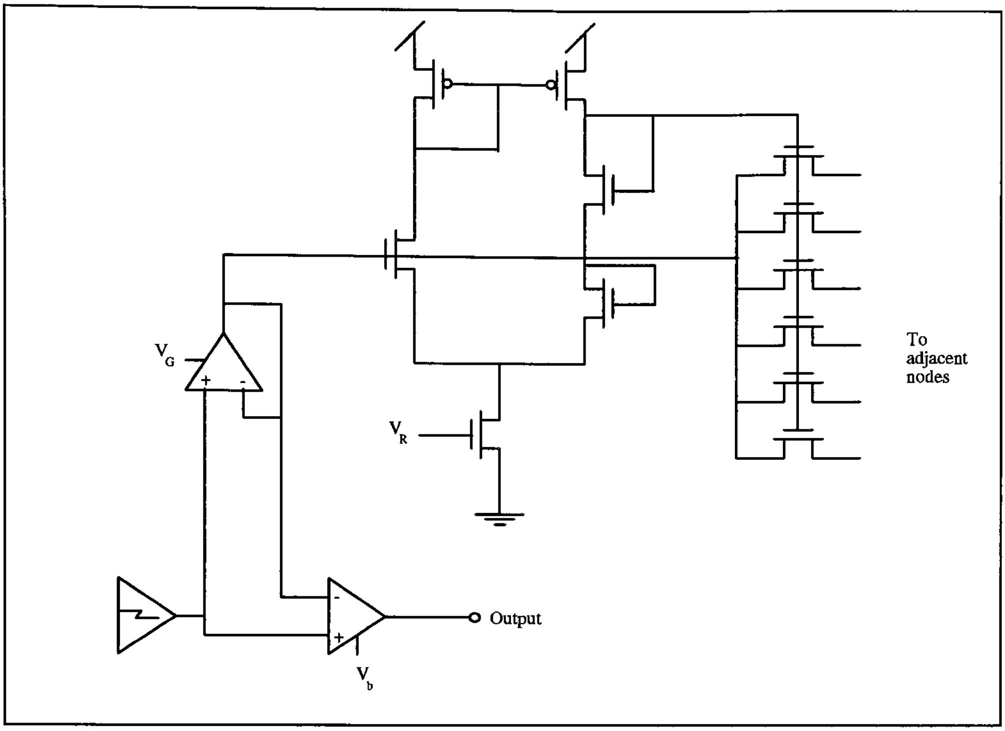

Figure 4-21: Detailed schematic of all circuitry within an individual pixel of the retina [17]

4.8 IMPLEMENTATION

The floor plan for the retina is shown in figure 4-20. The chip consists of an array of pixels, and a scanning arrangement for reading the results of retinal processing. The output of any pixel can be accessed through the scanner, which is made up of a vertical scan register along the left side of the chip and a horizontal scan register along the bottom of the chip. Each scan register stage has a 1-bit of shift register, with the associated signal selection circuits. Each register normally is operated with a binary 1 in the selected stage, and binary 0's in all other stages. The selected stage of the vertical scan register connects the out-bias voltage to the horizontal scan line running through all pixels in the corresponding row of the array. The deselected stages force the voltage on their horizontal scan lines to ground.

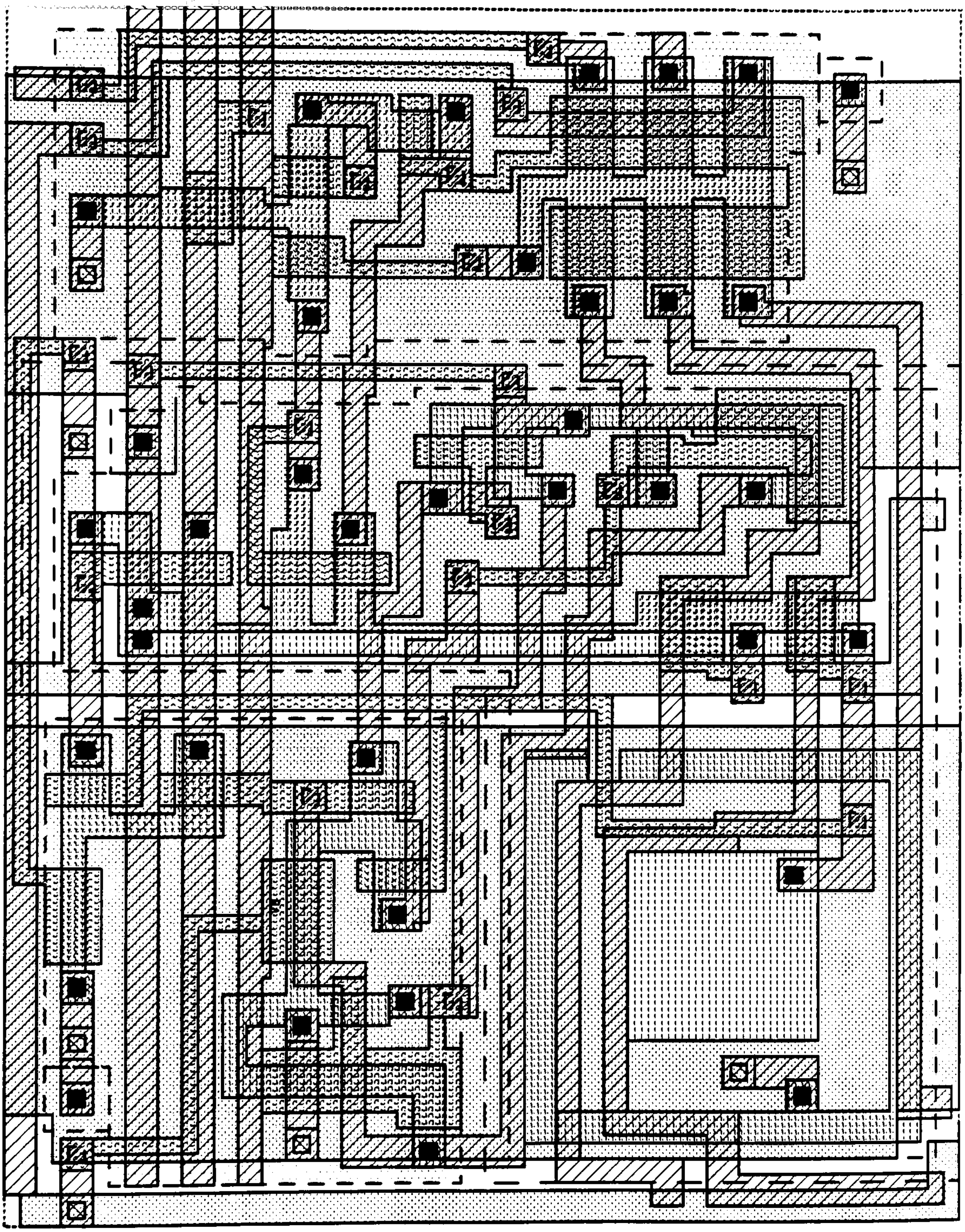

Figure 4-22 Layout of the Pixel of the Retina

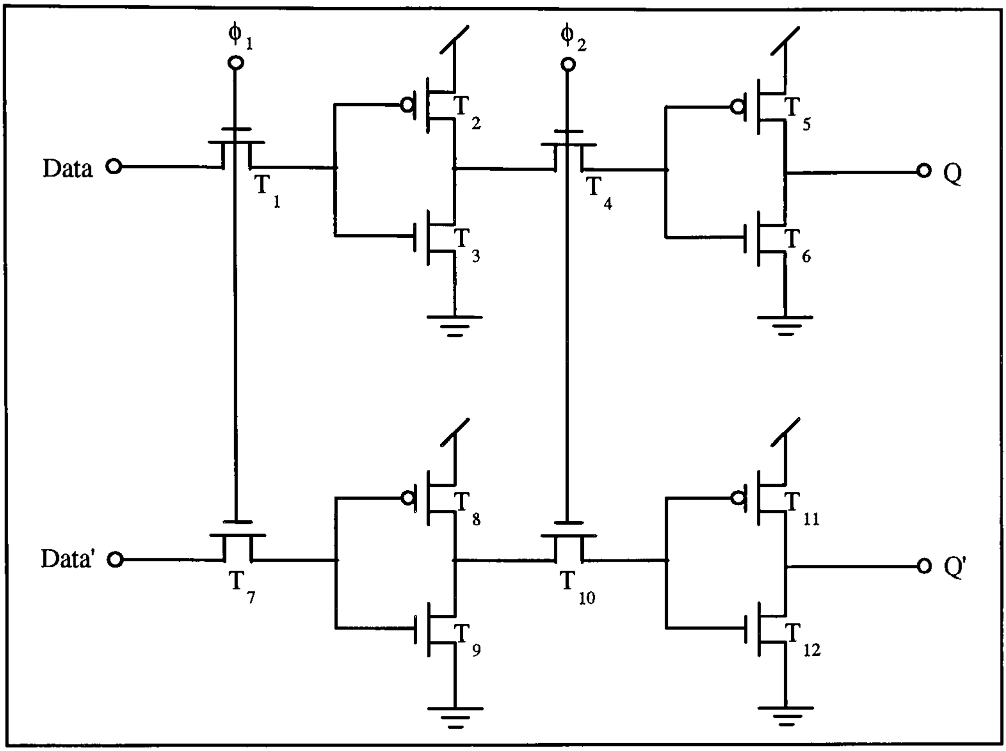

Figure 4-23: Shift register used in the horizontal and vertical scanner.

Each horizontal scan line is connected to the bias control (V_b) of the output amplifiers of all pixels in the row. The output of each pixel in a selected row is represented by a current, and this current is enabled onto the vertical scan line by the V_b bias on the horizontal scan line. The current scale for all outputs is set by the outbias voltage, which is supplied from offchip. A schematic diagram of all the circuits in the pixel is as shown in figure 4-21.

4.8.1 SHIFT REGISTER

The schematic for the shift register is as shown in figure 4-23. It is made up of a single pass transistor followed by a inverter, this in effect sums up to be a half bit shift register. We thus need two identical blocks to get a 1-bit shift register. Input to the

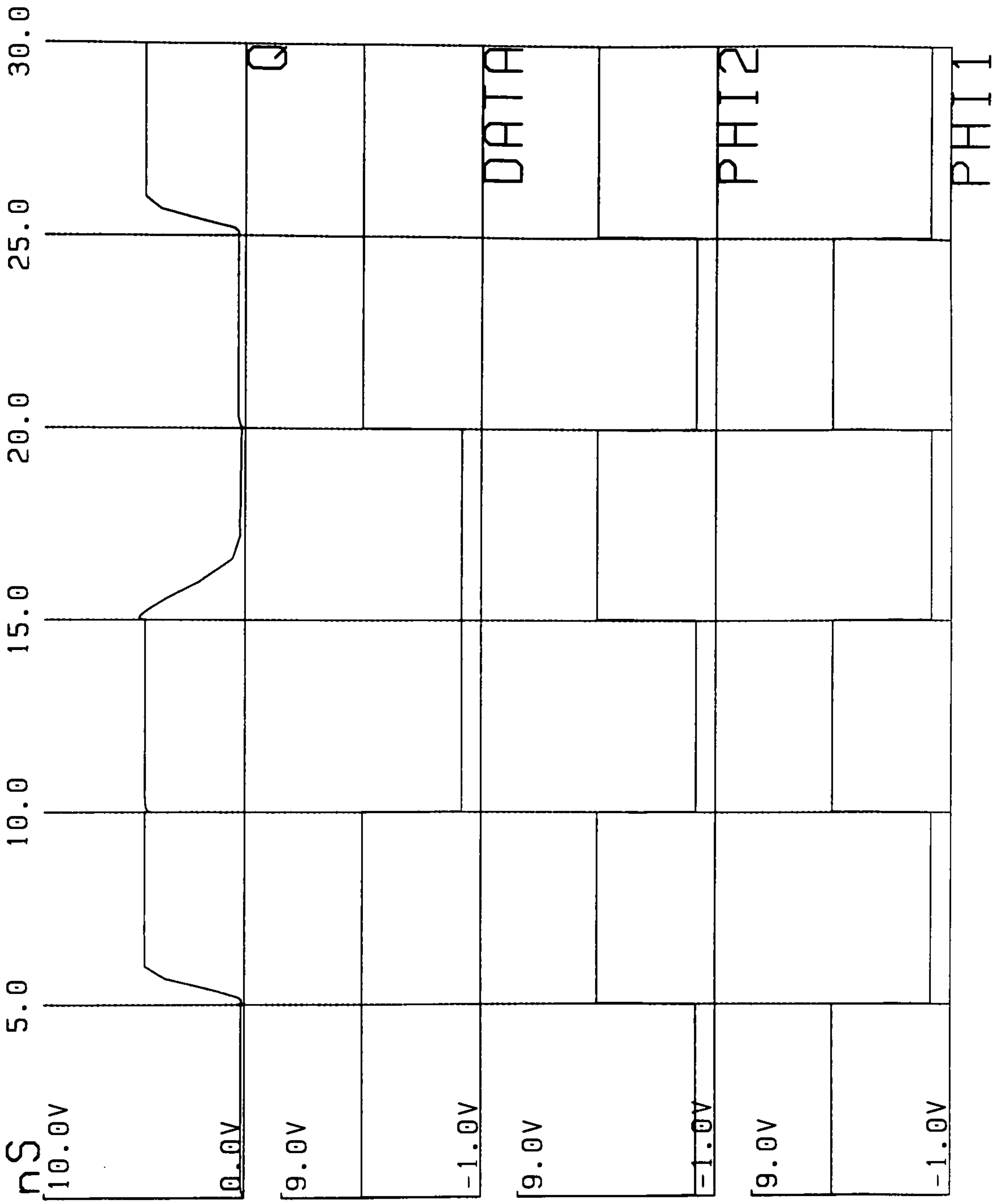

Figure 4-24: Simulated Results of the Shift Register.

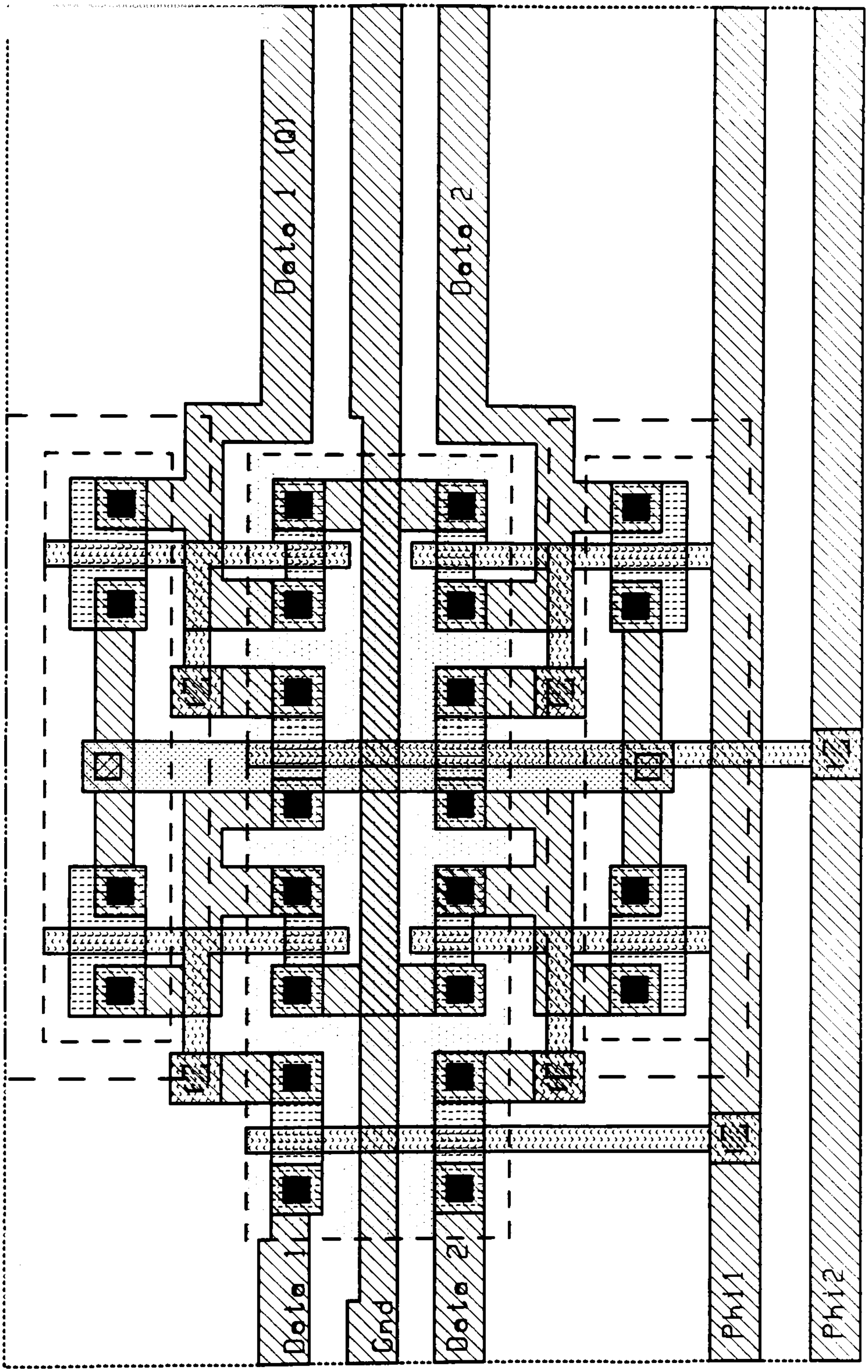

Figure 4-27: Layout of the Shift Register.

comes on the *data line,* which is taken in on the positive half of the clock (Φ_1) cycle. On the negative half of the clock (Φ_2) cycle, the inverted signal from the output of the first stage, is taken in and we thus get the final output of the 1 bit shift register after one clock cycle. Both $\Phi_1$and Φ_2 are the opposite halves of the same clock pulse. The simulated result and the layout of the shift register is as shown in figure 4-26 and 4-27.

4.8.2 HORIZONTAL & VERTICAL SCANNER

The circuit associated with driving a horizontal scan line and selecting data from a vertical scan line are shown in figure 4-28. The current in a vertical scan line is connected to one of the two output lines through a pair of complementary pass transistor analog switches. If a binary 1 is stored in the corresponding stage of the horizontal shift register, the vertical scan line is connected to the line labeled *out.* If a binary 0 is stored in the stage, the vertical scan line is connected to the line labeled V_{ref}. The current from the selected column thus flows in the *out* line, and the current from all unselected columns flows in the V_{ref} line. The chip is designed to be used with the off-chip current sense amplifier shown to the right of the broken line in figure 4-22. The *out* line is held at the Vref potential by negative feedback from the amplifier output through the resistor. The principal advantage of this arrangement is that all vertical scan lines, selected and unselected, are held at the same potential. Thus, no transient is introduced as the vertical scan line is selected. In addition, capacitive transients due to the charge in the pass transistor channels are minimized by the complementary nature of the analog switches.

The scanners can be operated in one of the two modes, static probe or serial access. In static probe mode, a single row and column are selected, and the output of a single pixel is observed as a function of time, as the stimulus incident on the chip is changed. In serial access mode, both vertical and horizontal shift registers are clocked at regular intervals to provide a sequential scan of the processed image for display on a television monitor. A binary 1 is applied at *horizontal* and is clocked through the horizontal shift register in the time required by a single scan line in the television display. A binary 1 is applied at *vertical,* and is clocked through the vertical shift register in the time required by one frame of the television display. The vertical scan lines are accessed in sequential order via a single binary 1 being clocked through the horizontal shift register. After all pixels in a given row have been accessed, the single binary 1 in the vertical shift register is advanced to the next position, and the horizontal scan is repeated. The horizontal scan can be fast because it involves current steering and does not require voltage changes on the capacitance of a long scan wire. The vertical selection, which involves the settling of the output bias on the selected amplifiers, has the entire horizontal flyback time of the television display to settle, before it must be stable for the next horizontal scan.

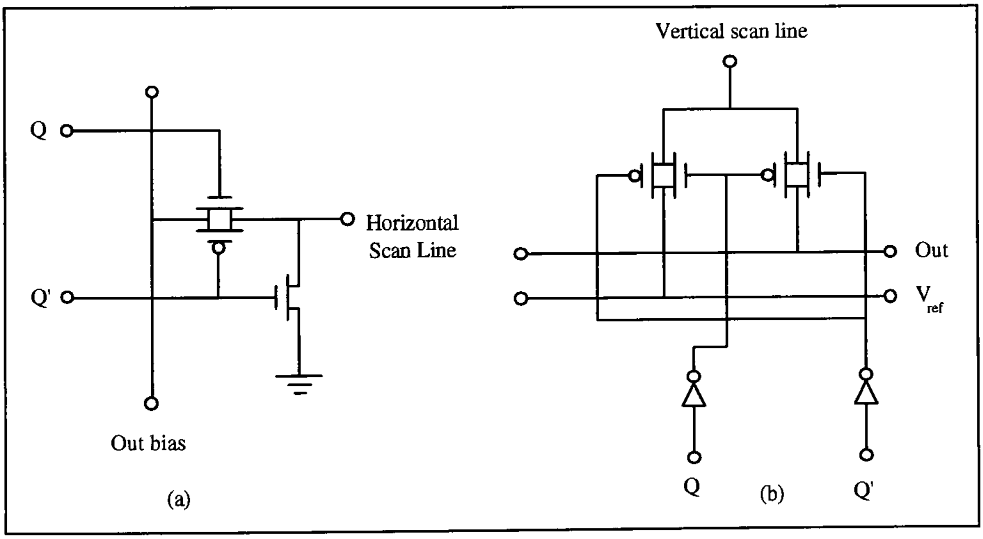

Figure 4-26: (a) Schematic of the driver for the horizontal scan line.
(b) Schematic of a multiplexer for the vertical scan line.

The core of the chip is made up of rectangular tiles with dimensions of 120x154 μ. Each tile contains the circuitry for a single pixel, as shown in figure 4-15, with the wiring

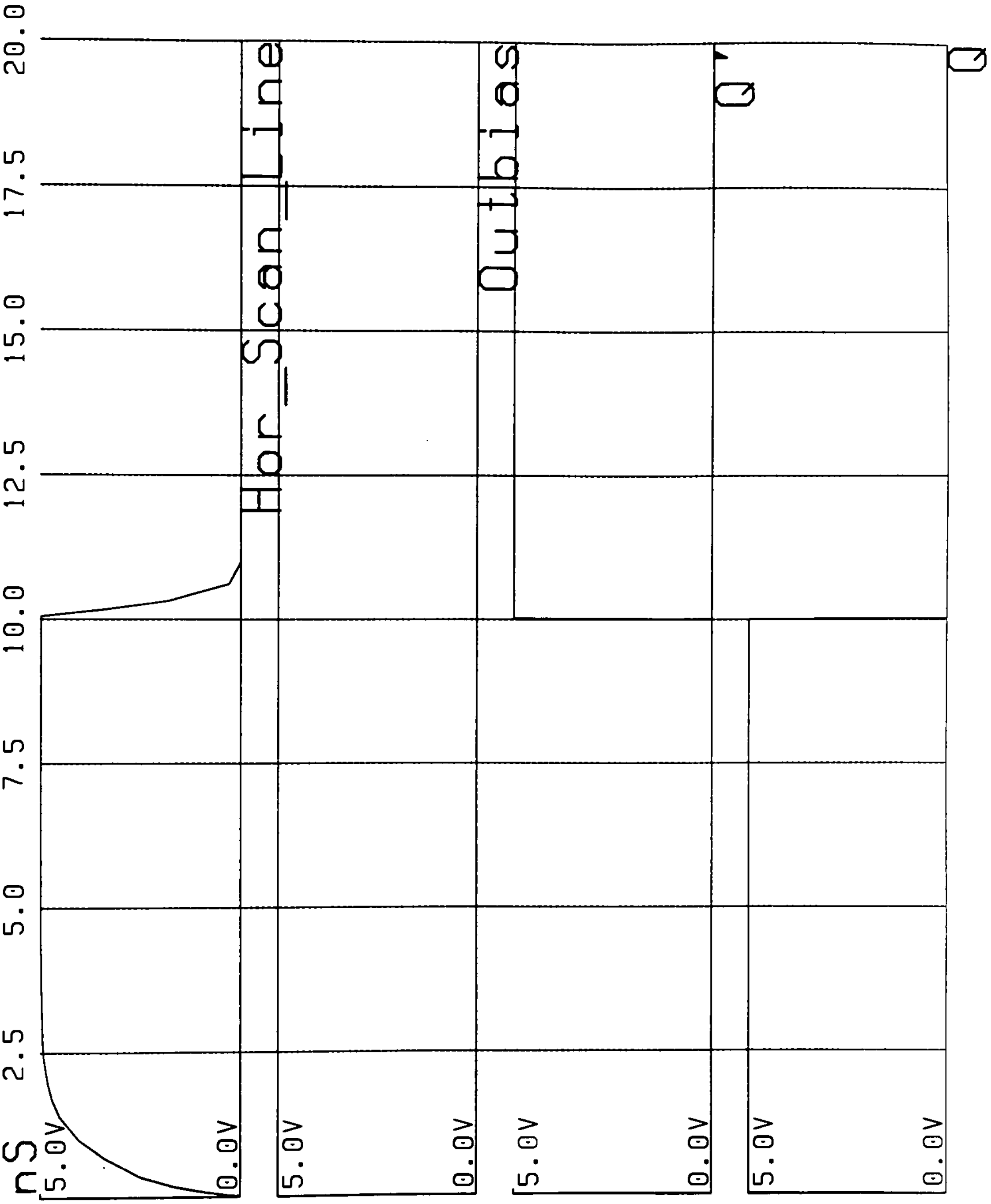

Figure 4-27: Simulation Results of Driver Circuit Required for the Horizontal Scanner.

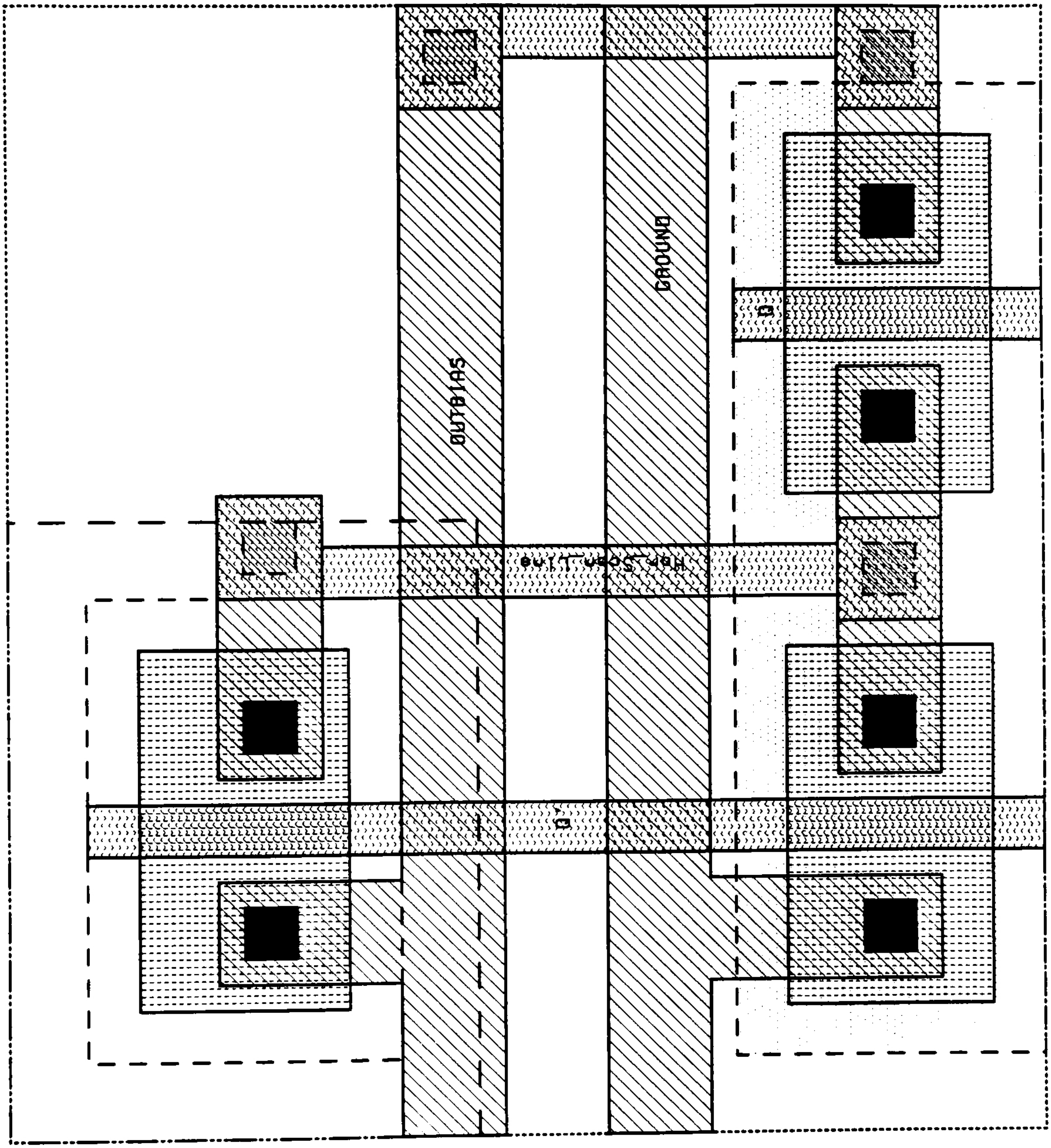

Figure 4-28: Layout of the Driver Circuit for the Horizontal Scanner

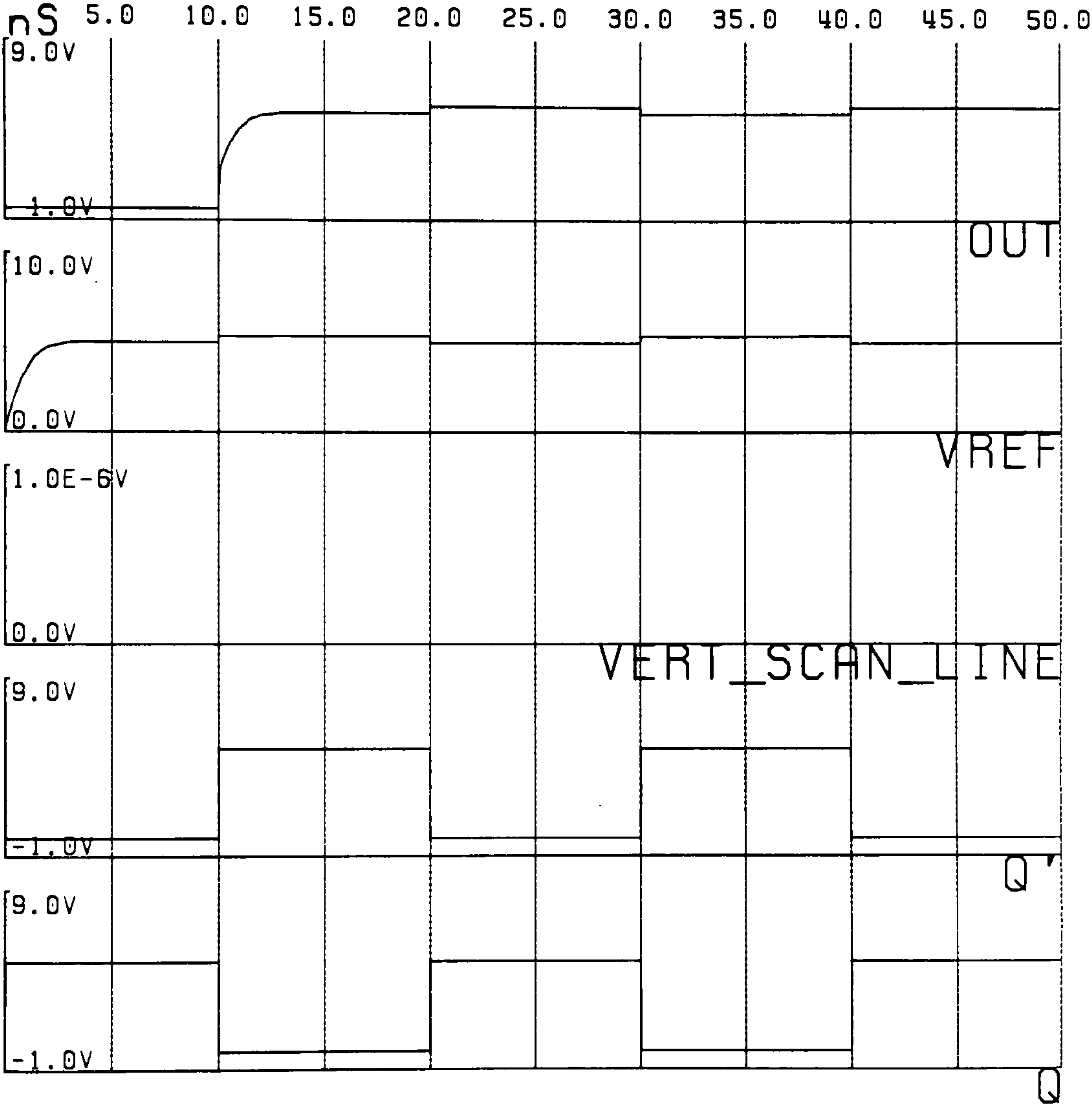

Figure 4-29: Simulation of the Multiplexer Required for the Vertical Scanner.

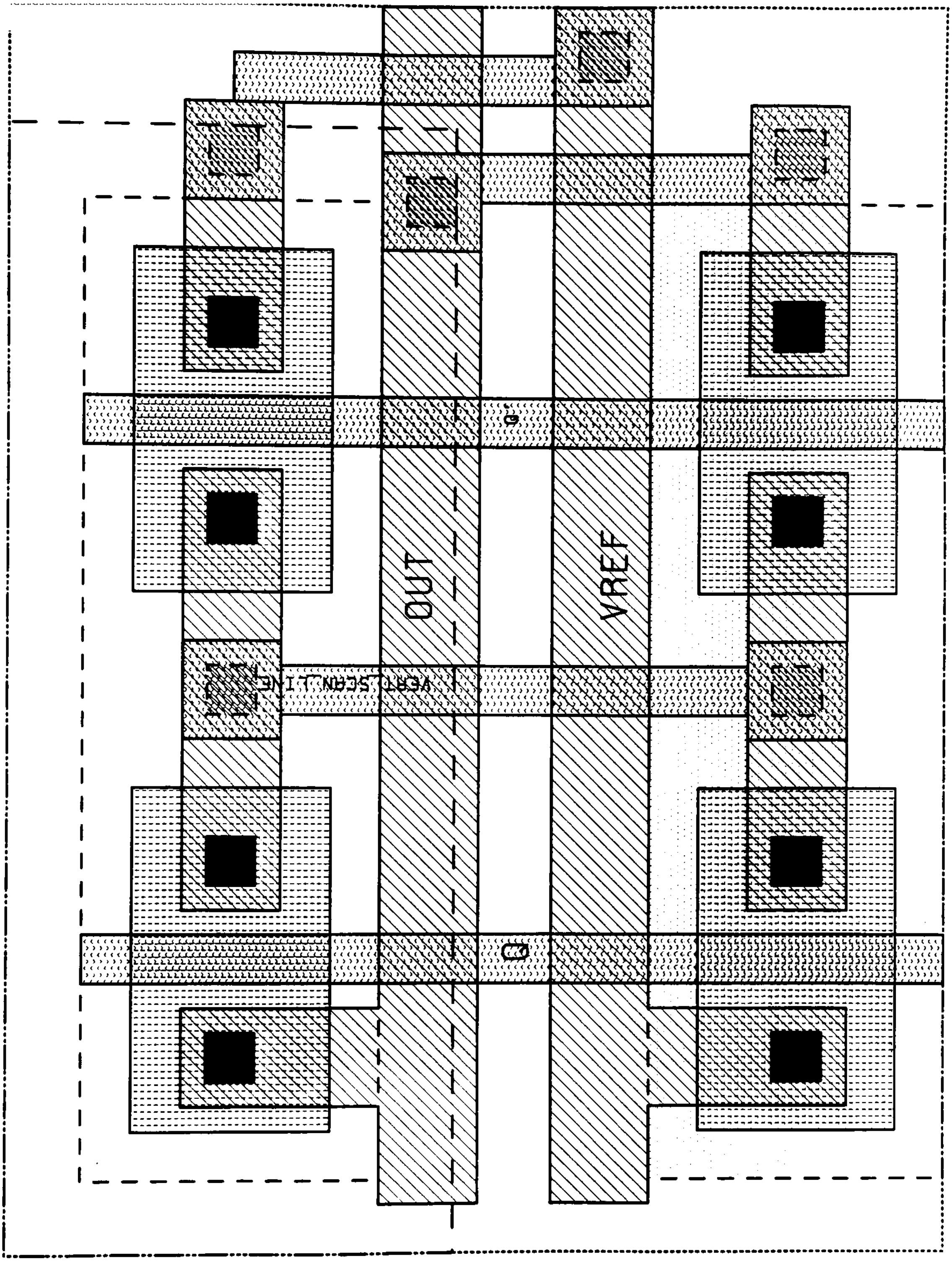

Figure 4-30: Layout of the Multiplexer needed for the Vertical Scanner.

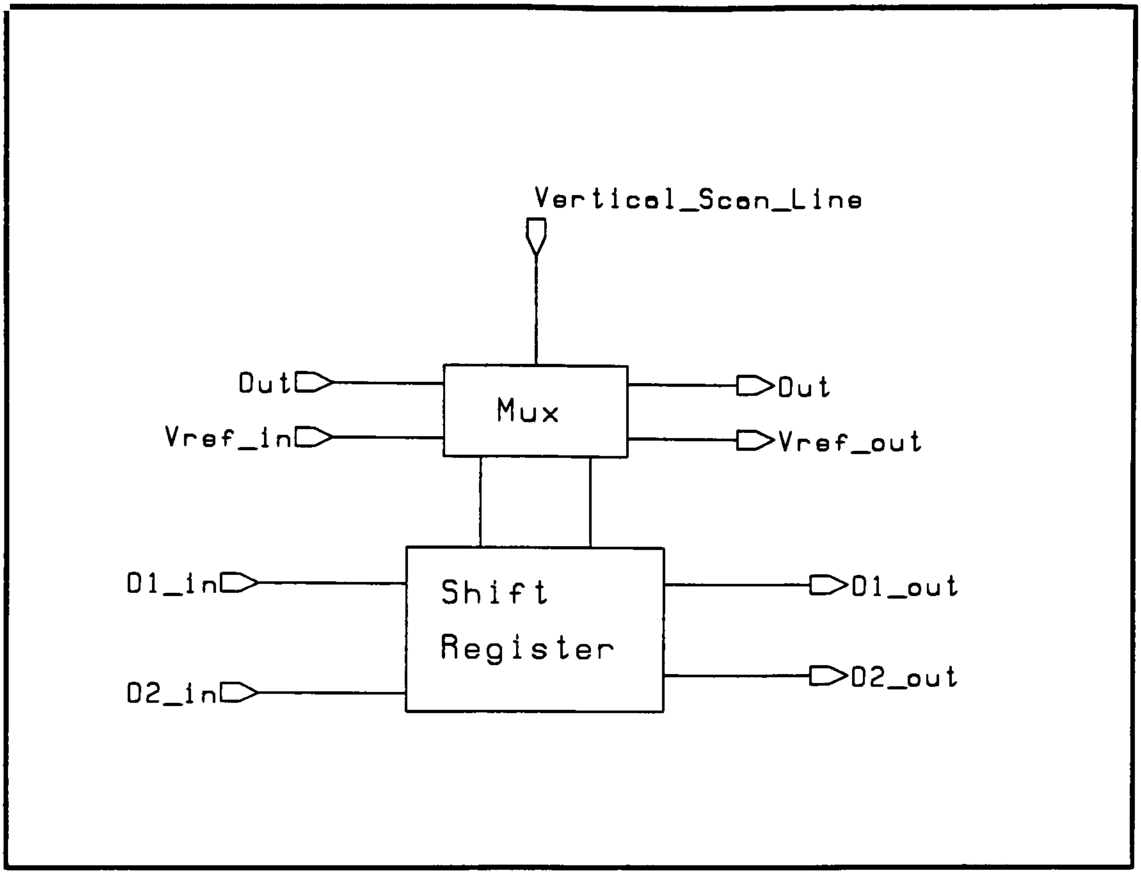

Figure 4-31: Schematic Arrangement of the Vertical Scanner

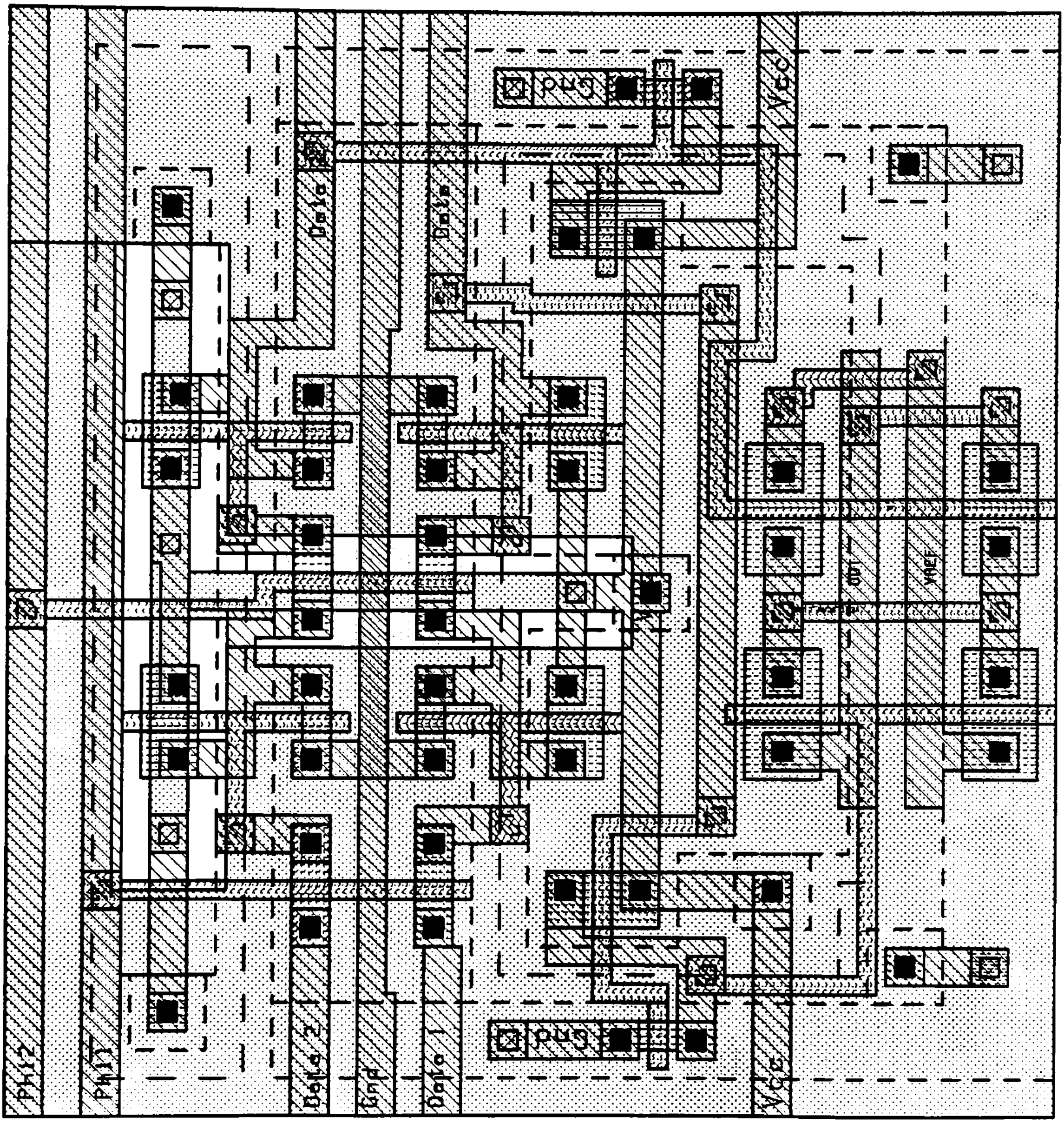

Figure 4-32: Layout of the Vertical Scanner.

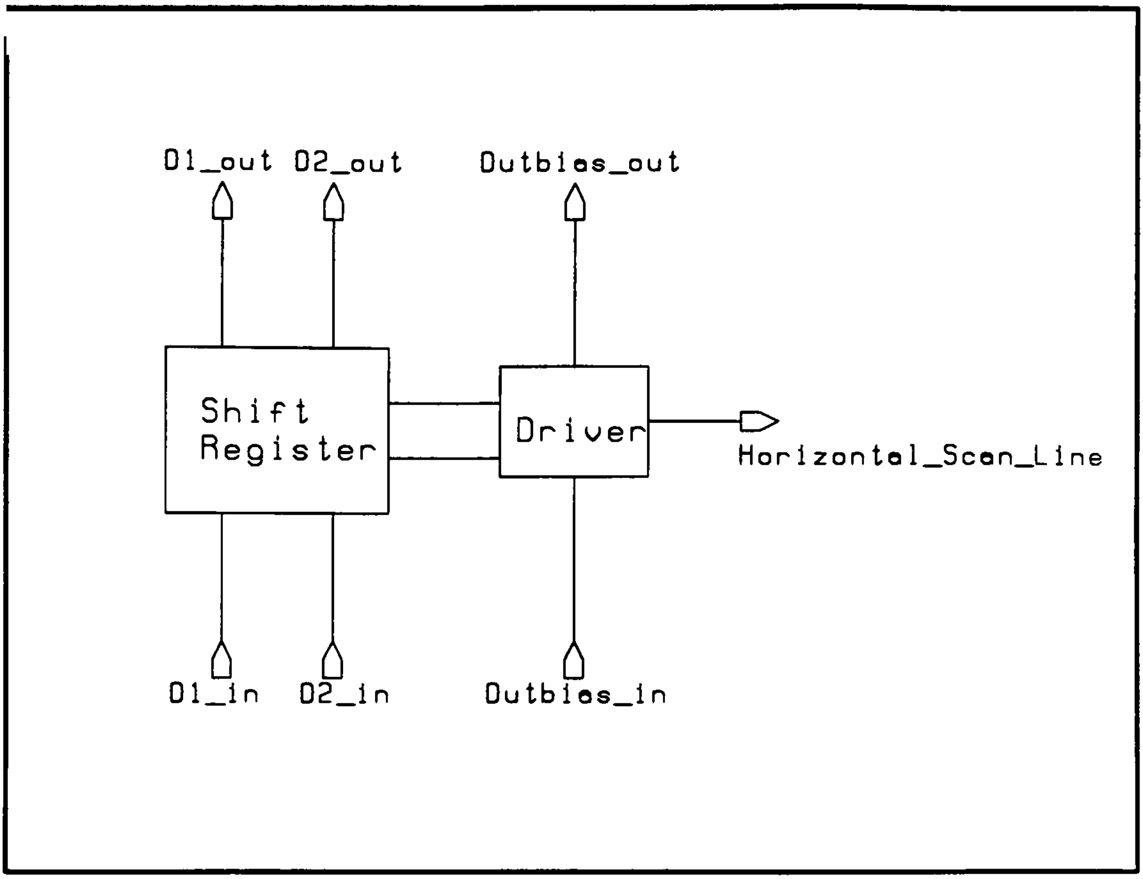

Figure 4-33: Schematic Arrangement of the Horizontal Scanner.

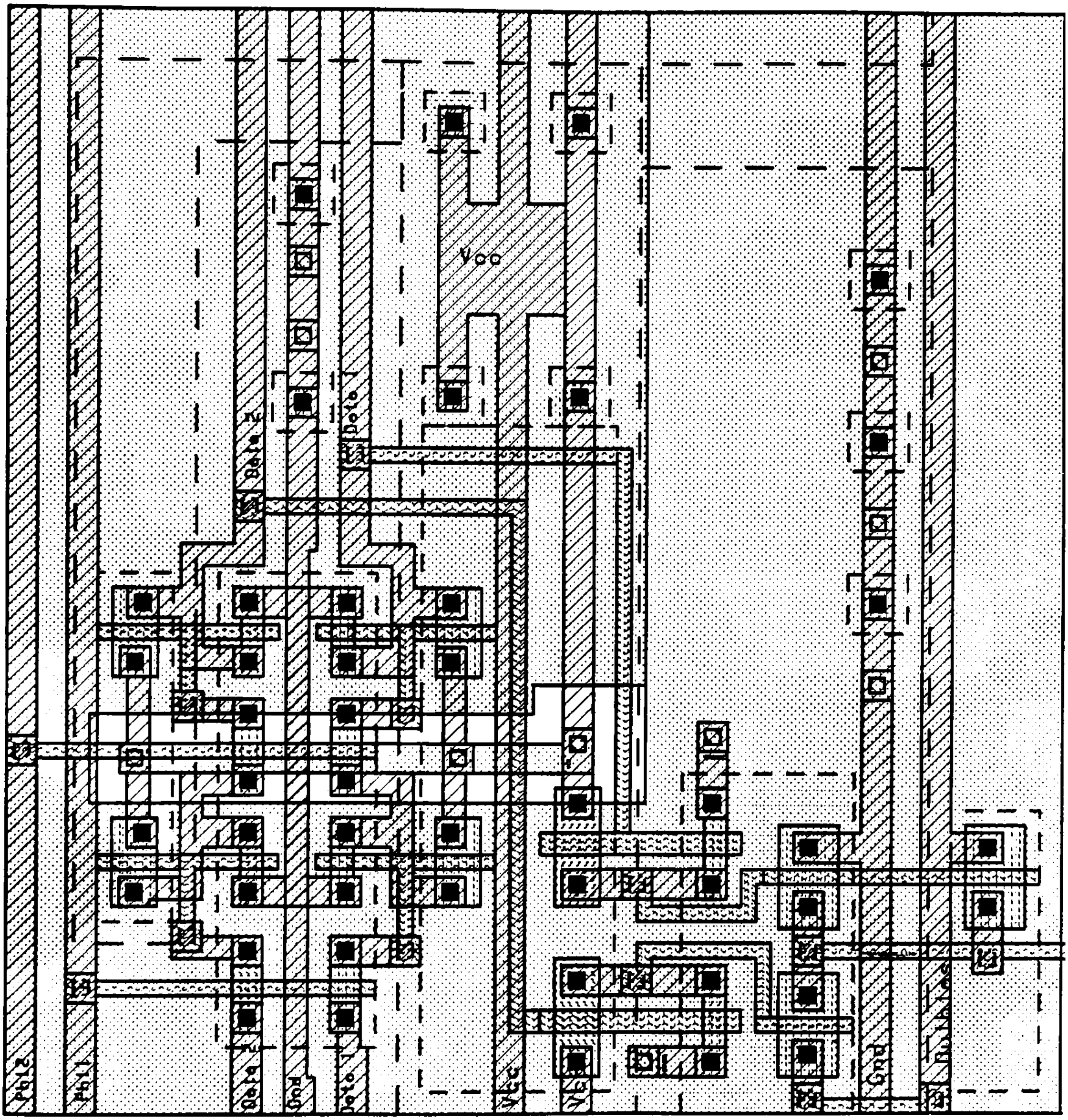

Figure 4-34: Layout of the Horizontal Scanner.

necessary to connect the pixel to its nearest neighbors. Each tile also contains the sections of global wiring necessary to form signal nets for V_{DD}, the bias controls for the resistive network, and the horizontal and vertical scan lines. The photoreceptors are located near the vertical scan line such that alternating rows of left and right facing cells form a hexagonal array. This arrangement allows the vertical scan wire to be shared between adjacent rows, being accessed from the left by the odd rows, and from the right by even rows. To protect the processing circuitry form the effects of stray minority carriers, the entire chip has been covered with a solid sheet of second layer of metal, with openings directly over the photoreceptors. This layer is used for distributing ground to the pixels.

4.8.3 PERFORMANCE

The experiments on the silicon retina have yielded results remarkably similar to those obtained from biological systems. From an engineering point of view, the primary function of the computation performed by the retina is to provide an automatic gain control that extends the useful operating range of the system. It is essential that a sensory system be sensitive to changes in its input, no matter what the viewing conditions. The structure executing this gain control operation can perform many other functions as well, such as computing the contrast ratio or enhancing edges in the image. Thus, the mechanisms responsible for keeping the system operating over an enormous range of image intensity and contrast have important consequences with regard to the representation of data.

4.8.4 SENSITIVITY CURVES

The computation performed in the distal portion of the retina prevents the output from saturating over an incredible range of illumination levels. By logarithmically compressing the input signals, the photoreceptor takes the first step toward increasing the retina's dynamic range. The next step is a level normalization, implemented by means of the resistive network. The horizontal cells of the retina provide spatially averaged version of the photoreceptor outputs, with which the local photoreceptor potential can be compared. The triad synapse senses the difference between the photoreceptor output and the potential of the horizontal cells, and generates a bipolar cell output from this difference. The maximum response occurs when the photoreceptor potential is different from the space-time averaged outputs of many photoreceptors in the local neighborhood. This situation occurs when the image is changing rapidly in either space or time.

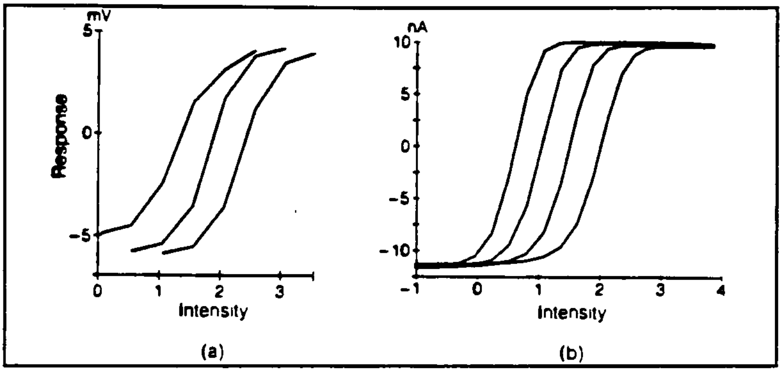

Figure 4-35: Intensity response curves shift to higher intensities at higher background illuminations [17]

Figure 4-35, shows the shift in operating point of the bipolar cell output of both biological and a silicon retina, as a function of surround illumination. At a fixed surround illumination level, the output of the bipolar cell has a familiar tanh characteristic, it saturates to provide a constant output at very low or very high center intensities, and it is sensitive to changes in input over the middle of its range. Using the potential of the resistive network as a reference centers the range over which the output responds on the signal level averaged over the local surround. The full gain of the triad synapse can thus be used to report features of the image without fear that the output will be driven into saturation in the absence of local image information.

In the retina, the operating point of the system is the local average of intensity as computed by the horizontal cells. Because it uses a local rather than a global average, the eye is able to see detail in both light and dark areas of high contrast scene, a task that would overwhelm a television camera, which uses only global adaptation.

4.8.5 TIME RESPONSE

Time is an intrinsic part of an analog computation. In analog perception systems, the time scale of the computation must be matched to the time scale of external events, and to other real time parts of the system. The body and the eye movements are an important part of the computation.

Figure 4-36, shows the response of a single output to a sudden increase in incident illumination. Output from a bipolar cell in a biological retina is provided for comparison. The initial peak represents the difference between the voltage at the photoreceptor caused by the step input and the old averaged voltage stored on the capacitance of the resistive network. As the resistive network equilibrates to the new level, the output of the amplifier diminishes. The final value is a function of the size of the stimulus, which changes the average value of the intensity of the image as computed by the resistive network. Having computed a new average value of intensity, the resistive network causes the output of the amplifier to overshoot when the stimulus is turned off. As the network decays to its former value, the output returns to the baseline.

The temporal response of the silicon retina depends on the properties of the horizontal network. The voltage stored on the capacitance of the resistive network is the temporally as well as the spatially averaged output of the photoreceptors. The horizontal network is like the follower integrator circuit, which weights its input by an amount that decreases exponentially into the past. The time constant of integration is set by the bias voltages of the wide range amplifier and the resistors. The time constant can be varied independently of the space constant, which depends on only the difference between these bias voltages, rather than on their absolute magnitude.

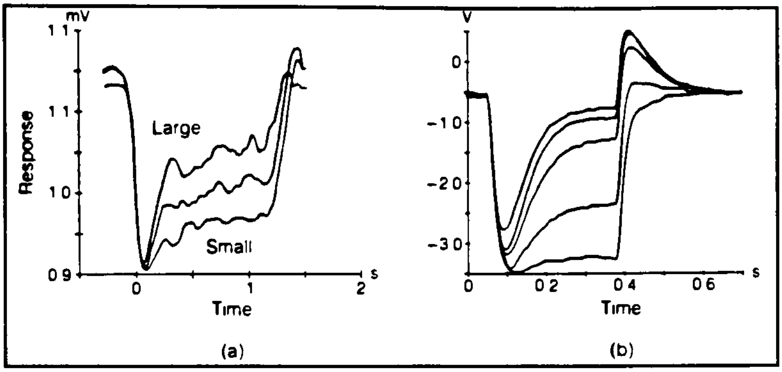

Figure 4-36: Temporal response to different sized test flashes. [17]

The form of time response of the system varies with the space constant of the network. When the resistance value is low, γ approaches one, and the network is computing the global average. A test flash of any limited size will produce a sustained output. Conversely, when the resistance value is high, γ approaches zero, and the triad synapse is just a differentiator circuit, which has no sustained output.

4.8.6 EDGE RESPONSE

The outputs of the bipolar cells directly drive the sustained X-type retinal ganglion cells of the mud puppy. The receptive fields of these cells are described as antagonistic center-surround fields. Activation of the center of the receptive field stimulates the cell's response, and activation of the surround produces inhibition. Cells with this organization are strongly affected by *discontinuities in intensity*. The response of a sustained X-type ganglion cell to a contrast edge placed at different positions relative to its receptive field is shown in figure 4-37(a). The spatial pattern of activity found in the cat is similar to the response of our silicon retina to a spatial intensity step, as shown in figure 4-37(b).

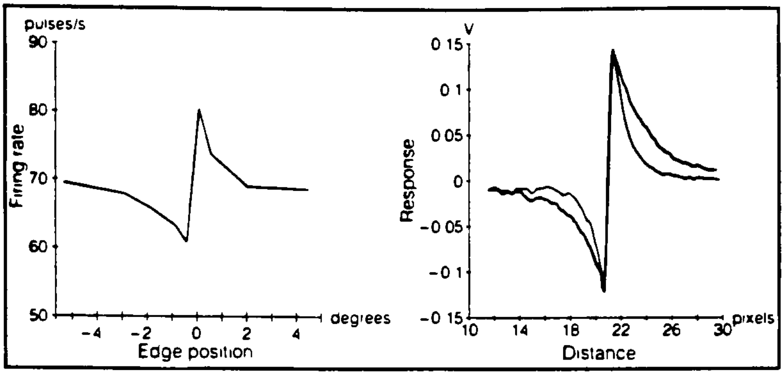

Figure 4-37: Spatial derivative response of a retinal ganglion cell and of a pixel to a contrast edge [17]

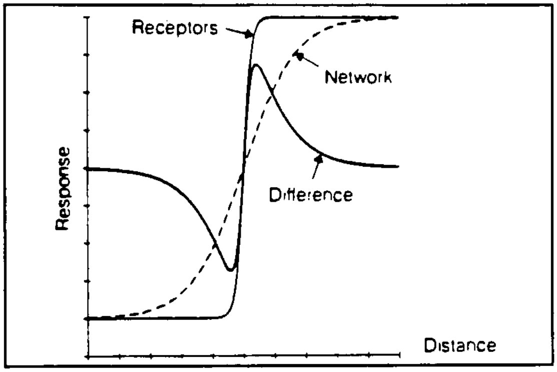

Figure 4-38: Illustration of the mechanism of generation of pixel response to spatial edge in intensity. [17]

The way the second spatial derivative is computed is illustrated in figure 4-38. The surround value computed by the resistive network reflects the average intensity over a restricted region of the image. As the sharp edges passes over the receptive field center, the output undergoes a sharp transition from lower than the average to above the average. Sharp edges thus generate large output, whereas smooth areas of the image produce no output, because the local center intensity matches the average intensity.

4.8.7 MACH BANDS

In the visual systems of higher animals, the center-surround response to a local stimuli is responsible for some of the strongest visual illusions. For example, Mach bands, the Hermann-Hering grid illusion, and the Craik-O'Brian-Cornsweet illusion may all be traced to simple inhibitory interactions among elements of the retina.

The response of a pixel to a ramp stimulus is plotted in figure 4-39. Because the retina performs a second order filtering of the image, changes in the first derivative of intensity are enhanced. *Mach bands* are illusory bright and dark bands that appear at the edges of an intensity ramp. The positions of the illusory bands correspond to the positions where the retinal output is enhanced due to changes in the first derivative of the intensity.

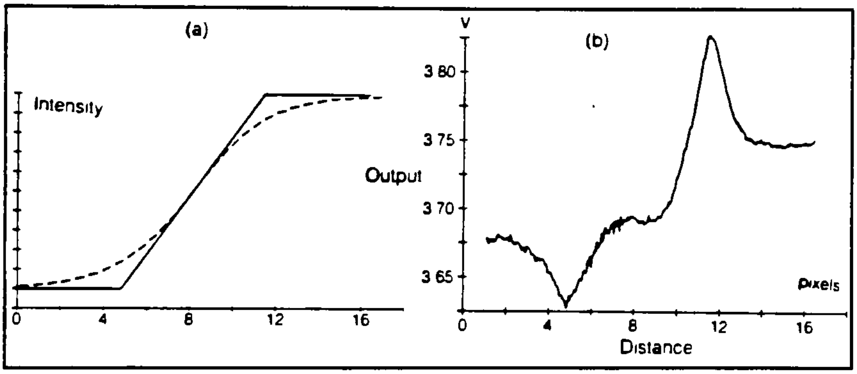

Figure 4-39: (a) Ramp stimulus illustrates the function of a second order filter. (b) Response of a pixel to ramp stimulus. [9]

The retina, as the first stage in the visual system, provides gain control and image enhancement, as well as transduction of light into electrical signals. From an engineering viewpoint, the retina greatly reduces the signal bandwidth required to transmit visual information to the brain, thereby greatly reducing the size of the optic nerve and allowing more effective computation at the next level.

5.0 PARAMETER EXTRACTION

5.1 PARASITIC RESISTANCE AND CAPACITANCE

Parasitic capacitance effects can be divided into two classes, *intrinsic* which is the capacitance between a conduction layer and the base, and *coupling* which is the capacitance between two different nets in the conduction layers. The two constituents of intrinsic capacitance are body that is proportional to the area of the conduction geometry, and *fringe* (sidewall) that is proportional to the perimeter of the conduction geometry. There are three constituents for coupling capacitance. Crossover is the effect between two different nets on two different layers which overlap each other. *Crossover overlap* capacitance is proportional to the area of overlap and *crossover fringe* capacitance is proportional to the perimeter of the overlap. When two conduction edges of two different nets are coincident, there would be involved both crossover fringe capacitance between the two nets and intrinsic fringe for each layer. The third coupling effect is *near-body* capacitance which is between two different parallel nets on the same conduction layer. It is proportional to the length of the opposing net edges and inversely proportional to the distance separating them. The process parameters provide the proportionality constants for capacitance. Intrinsic parameters may be specified for each conduction and base layer pair. Crossover overlap parameters may be specified for each unique pair of conduction layers, while crossover fringe parameters may be specified for each unique ordered pair of conduction layers. Near;-body parameters may be specified for each conduction layer.

Shielding is a complication caused by the vertical ordering of the conduction layers. When more than two conduction layers overlap, layers in-between shield the capacitance effect between the outer layers. In figures 5.1 and 5.2, an example of shielding and the corresponding capacitances is seen. It is noted that the poly is shielding a portion of metal1 from the base layers. The area of metal1 has been split to contribute separately to C5 and C8. Also the area of poly has been split due to different base materials for C2 and C3. All the capacitors may be calculated with different parameters since they each deal with a unique effect and layer pair. In particular, C7 and C9 may have different parameters because they have different ordered pairs of layers. The parameters for C7 is a function of the thickness of metal1, while the parameter for C9 is a function of the thickness of poly.

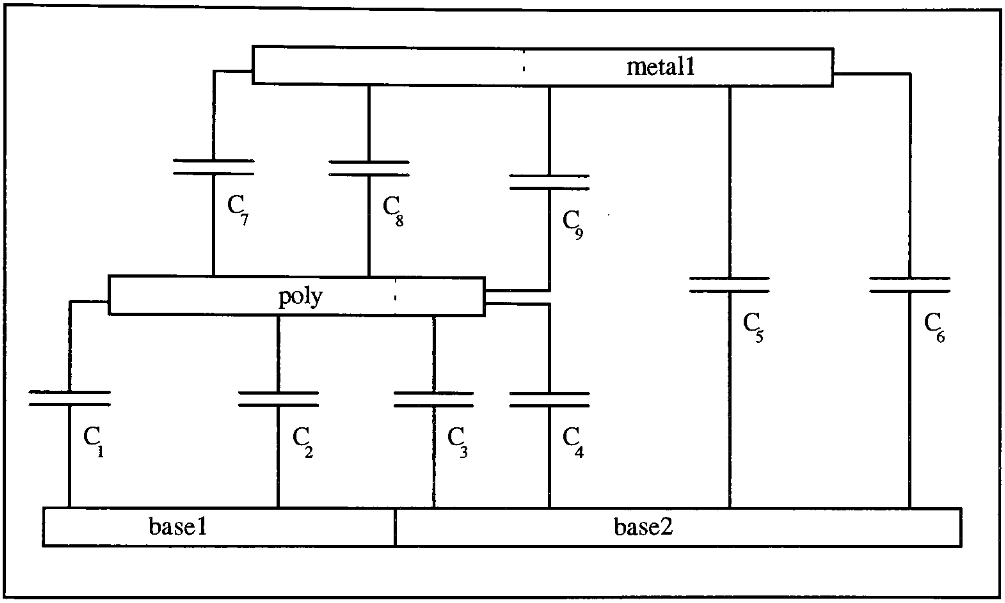

Figure 5.1: Shielding Example.

Parasitic resistance has two constituents, *sheet* resistance and *connection* resistance. Sheet resistance is present for current flowing within a conduction layer from one boundary to another. It is measured by sheet resistivity, which is in units of Ohms per square. Connection resistance is present for connection layers, such as contacts and via's, which connect different conduction layers. To efficiently calculate resistance values, a net on a conduction layer must be fractured into smaller elements. These smaller elements are constructed such that good resistance approximations can be calculated using simple equations involving sheet resistivity. Once a net has been fractured and the resistance of each element determined, the capacitance is calculated for each element. These elements, with their R and C value pairs, constitute a sub network of the original net. This sub network can be reported in total or passed to an analysis/reduction function which reports an equivalent RC pair for each path in the sub network. A path is a pairing of net's source pin with a net's sink pin. Thus there are as many paths as there are sink pins.

NAME	EFFECT	LAYER1	LAYER2
C1	fringe	poly	base1
C2	body	poly	base1
C3	body	poly	base2
C4	fringe	poly	base2
C5	body	metal1	base2
C6	fringe	metal1	base2
C7	crossover fringe	metal1	poly
C8	crossover overlap	metal1	poly
C9	crossover fringe	poly	metal1

Figure 5.2: Capacitance Effects.

Parasitic capacitance may be calculated, without regard to resistance, for all or any of the capacitance effects. These effects can be reported individually or summed to create a *lumped capacitance* for each net. This lumped value may be back-annotated to a schematic design. Based on the total areas and perimeters of the conduction layers for a net, an approximate resistance can be calculated for a rectangle having the same area and perimeter. The sum of this resistance plus connection resistances in the net yield an approximate lumped resistance for each net. Using the lumped values for each net as selection criteria, threshold values may be specified to filter nets that are not to be extracted for parasitic resistance.

5.2 DATA REDUCTION

In this method we process the extracted distributed parasitic circuits and reduce the network to an equivalent network that provides a good approximation of behavior but uses a significantly smaller number of parasitic circuit elements and nodes. Since circuit simulation time is proportional to the number of nodes in the circuit, this reduction of nodes will reduce that time.

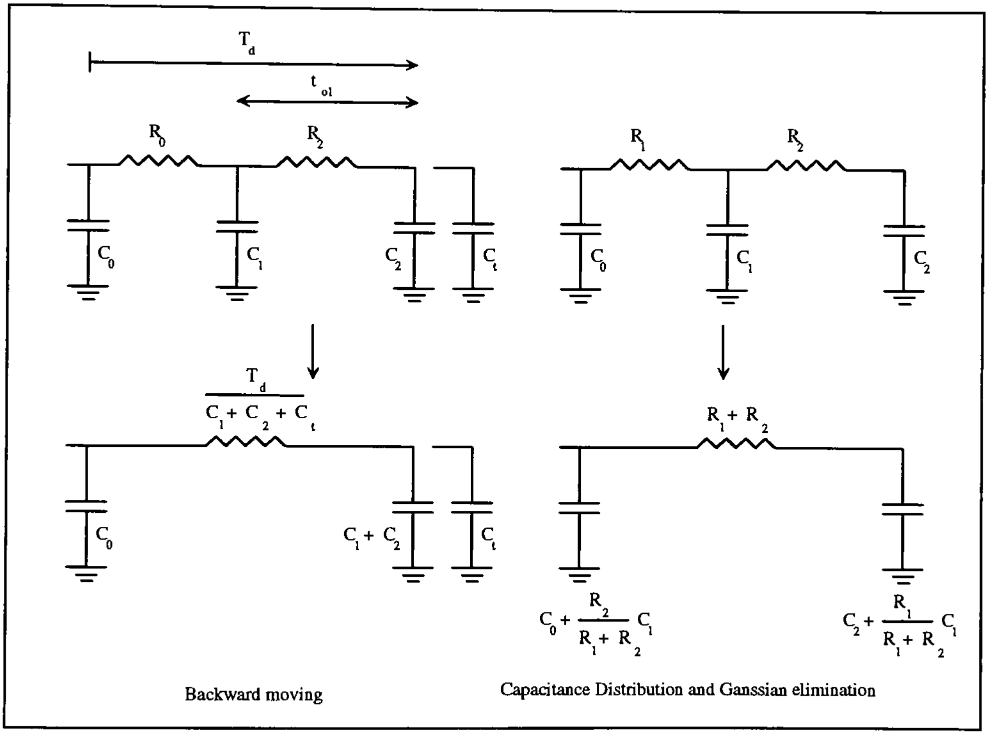
T_d
t_{ol}
R_0
R_2
C_0
C_1
C_2
C_t
R_1
R_2
C_0
C_1
C_2
$\frac{T_d}{C_1 + C_2 + C_t}$
C_0
$C_1 + C_2$
C_t
$R_1 + R_2$
$C_0 + \frac{R_2}{R_1 + R_2} C_1$
$C_2 + \frac{R_1}{R_1 + R_2} C_1$
Backward moving
Capacitance Distribution and Ganssian elimination

Figure 5.3: Network reduction algorithm.

This network reduction is necessary since the extracted circuit from layout may contain many more RC parasitic elements than active ones, which causes unacceptable overhead for simulation and other verification tools.

5.3 CIRCUIT LAYOUT AND CAPACITANCES

Before we can analyze the transient characteristics of MOS inverters, we must know the actual physical dimensions of the transistors and their interconnections so that the capacitances which limit switching speed can be calculated. For manual analysis, device dimensions together with the capacitance per unit area for junctions and oxides permit calculation of all capacitances. Circuit simulators such as SPICE usually require entry of dimensions so that the program may calculate capacitances on circuit nodes.

5.4 LAYOUT EXAMPLE AND SPICE INPUT DATA

The essential features of a practical NMOS silicon gate inverter circuit layout are shown in figure 5.4. As is usual in integrated circuits work, vertical dimensions are exaggerated compared to horizontal dimensions so that small details is seen clearly. For clarity, the final layer of metal interconnections is not shown in the figure.

We shall now examine several features of figure closely. The drain of the inverter shares a common n^+ diffused region with the source of the load device. This saves area over using a separated diffusions, each with its own contact. For the depletion load circuit shown here, the gate of the load device must be connected to its source by metal. The output of this inverter may be connected to the following circuitry using diffusion or polysilicon or metal, since all of these are in contact with the output node. It should be noted that the diffused and polysilicon conductors cannot cross, because the intersection of these layers forms a transistor.

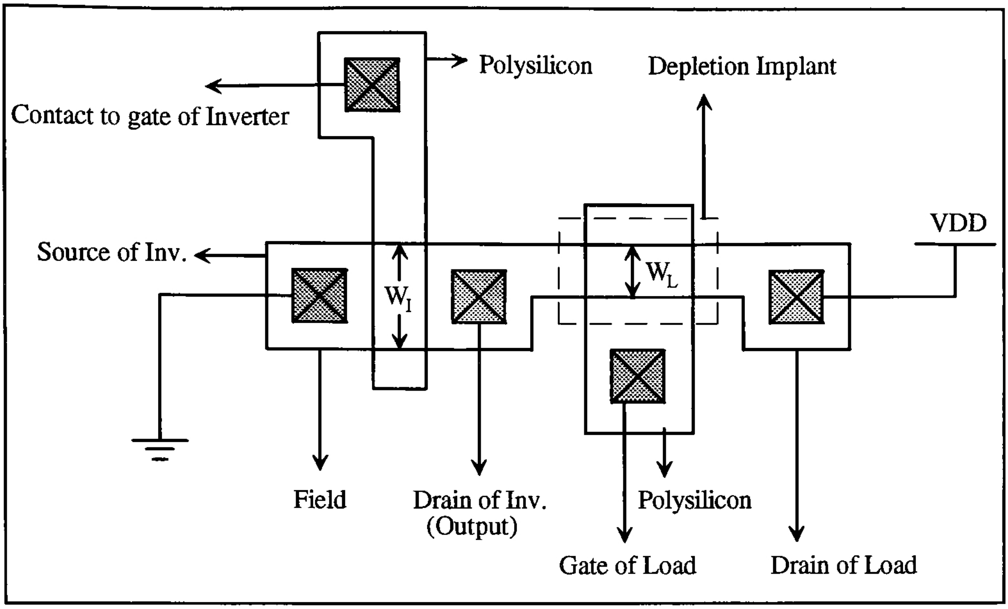

Figure 5.4: Layout of the NMOS Inverter

Figure 5.4 shows gate widths of 10 and 5 μm for inverter and load devices respectively. These are the distances between the walls of the field diffusion and are entered in SPICE as channel width W. Also seen in figure are gate lengths of 7 and 12 μm for inverter and load devices, respectively. These are assumed to be actual dimensions of the polysilicon gate electrodes after fabrication, and are entered in SPICE as length L. During fabrication, the n+ source and drain diffuse toward each other under the gate electrode, resulting in an overlap of the nominally self-aligned gate above source and drain, and an electrical channel length shorter than the gate length. For this example, we assume 1 μm of lateral diffusion at each source. Gate-source and gate-drain overlap are 1 μm each.

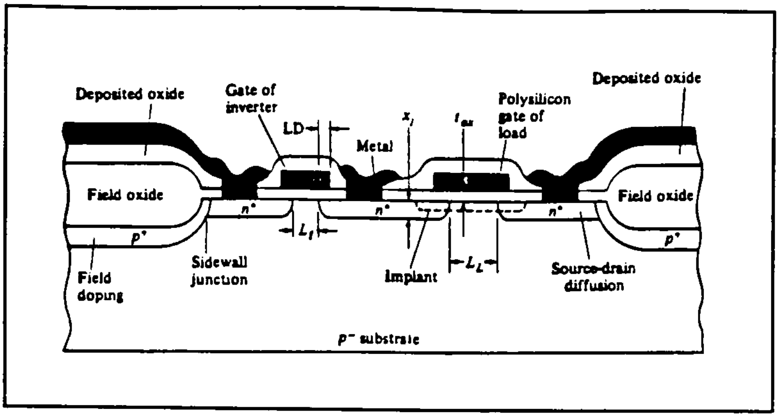

Figure 5.5: Section view of the NMOS Inverter.

The SPICE program calculates all device capacitances provided the necessary data is entered. The values for CJ and CJSW are entered, the zero bias capacitances for the bottom and sidewalls of the source body and drain body junctions. The capacitance unit area of an abrupt n$^+$p junction is

$$C_{j0} = \sqrt{\frac{q\varepsilon_{si}N_A}{2\phi_0}} = CJ \tag{5.1}$$

The sidewall capacitance per unit area is higher than CJ because n$^+$ source and drain about the p+ field diffusion. Unless other data are available, we assume that field doping is 10 times higher than body doping, so sidewall capacitance per unit area is higher by the square root of 10. Final SPICE requires that CJSW be given per unit of diffusion perimeter. Therefore sidewall capacitance per unit area is multiplied by junction depth XJ to obtain the value for CJSW to be entered into the MODEL file.

The built-in junction potential PB (ϕ_0) is given as,

$$\phi_0 = V_T \ln \frac{N_A N_D}{n_i^2} = PB \tag{5.2}$$

SPICE takes the same value of PB for bottom and sidewall junctions, causing only a very minor error. It is required that TOX be entered in the MODEL file if SPICE is to calculate the gate capacitances.

To summarize, the minimum set of data to enter in the MODEL file for MOS devices comprises the dc parameters VTO, KP, and GAMMA, gate oxide thickness TOX, and the capacitance parameters CJSW, and PB. Other parameters may take on default values without any serious errors.

5.5 CAPACITANCE CALCULATION FOR HAND ANALYSIS

A number of simplifications are necessary to facilitate hand analysis of MOS digital circuits because of the many nonlinear dc parameters and nonlinear capacitances in even a simple inverter.

Each MOS transistor has five separate voltage dependent capacitances coupling its four electrodes. Manual analysis of MOS transistor circuits in which each capacitor is considered individually is virtually impossible. However, approximate calculations of switching times becomes feasible if all capacitance effects are lumped into a single total capacitor C_T which is connected to the output node of each inverter or gate.

Voltage-dependent effects of junction capacitances are removed by defining equivalent linear capacitances C_{eq} which require the same change in charge as the nonlinear capacitors for a transition between two voltage levels V_1 and V_2.

With $V_2 > V_1$,

$$C_{eq} = \frac{\Delta Q}{\Delta V} = \frac{Q(V_2) - Q(V_1)}{V_2 - V_1} \equiv K_{eq} C_{j0} \qquad (5.3)$$

The depletion layer capacitance per unit area C_{j0} is calculated according to equation 5.1. Sidewall capacitances cannot be ignored for modern MOS processes. The sidewall capacitance per unit area is calculated as shown above. The dimensionless parameter K_{eq} for an abrupt junction is,

$$K_{eq} = \frac{-2\phi_0^{1/2}}{V_2 - V_1}[(\phi_0 - V_2)^{1/2} - (\phi_0 - V_1)^{1/2}] \qquad (5.4)$$

It is noted that the voltage applied to the junction is $V_2 = -(V_{OL} - V_{BB})$ in the low state and $V_1 = -(V_{OH} - V_{BB})$ in the high state. VBB is the (zero or negative) body bias voltage applied to the body with respect to the sources of inverter transistors. By convention, voltages applied to junctions are defined as positive for forward bias and negative for reverse bias.

Figure 5.6 shows calculated values of K_{eq} as a function of supply voltage V_{DD} and body voltage V_{BB}, on the assumption that $V_{OL} = 0$, $V_{OH} = V_{DD}$ and $\phi_0 = 1V$. These data are used in calculating C_T. The result of the linearization of junction capacitances using K_{eq} is a minor distortion of the shape of transient voltage waveforms at circuit nodes.

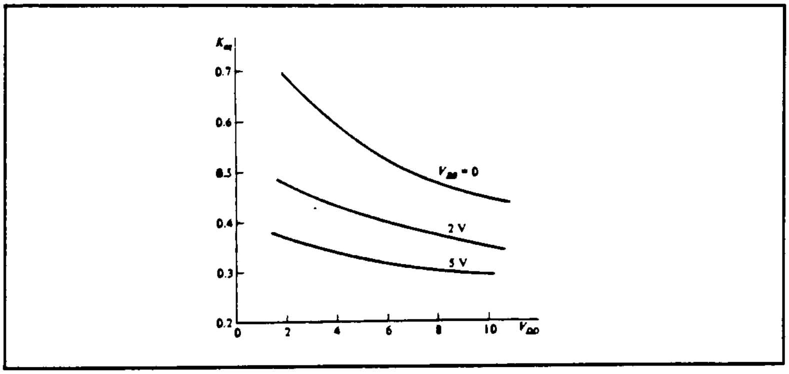

Figure 5.6: Equivalent Capacitance calculation

Figure 5.7 shows how the device capacitances in a circuit comprising two cascaded inverters can be lumped at the inverter output nodes. First, all capacitances across which there is no voltage change are ignored, since they have no effect on circuit performance. The device capacitance which must be considered are shown in figure 5.7(a). In this figure, the capacitances CL represent the capacitances of interconnecting wiring connected to inverter outputs.

The capacitance CT1 is made up as follows,

$$C_{T1} = K_{eq}(C_{db1} + C_{sb2}) + C_{gd1} + C_{gd2} + C_{g3} + C_L \qquad (5.5)$$

Transistors M_1 and M_2 are saturated or cut off during a large part of the switching cycle. Hence capacitances C_{gd} for these two devices are defined to include only the gate overlap capacitances to the drain of each. The total gate capacitance C_{g3} loads the output of the first inverter, so it is included without breaking it into separate components.

The major consequence of lumping all capacitance to ground is that the effects of capacitive coupling between input and output are ignored. This simplification is necessary for hand calculations.

Specific values of all capacitances are found from calculated values of capacitance per unit area and areas of nodes determined from the circuit layout at hand. In addition to the planar areas which are obvious in figure 5.5, sidewall areas where source and drain diffusions meet field doping (as seen in figure 5.4) are very significant in modern circuits. Sidewall capacitance per unit area is typically 3 to 5 times greater than capacitance along the bottom of the source-drain diffusion because the doping in the field regions is typically 10 to 25 times greater. Adequate accuracy is achieved by taking the sidewall area as the product of diffusion perimeter and junction depth, neglecting the curvature of the sidewall and the gradient in field doping. The capacitance for small areas of metal or polysilicon outside transistor regions, including the gate-body overlap capacitance CGBO, may be neglected.

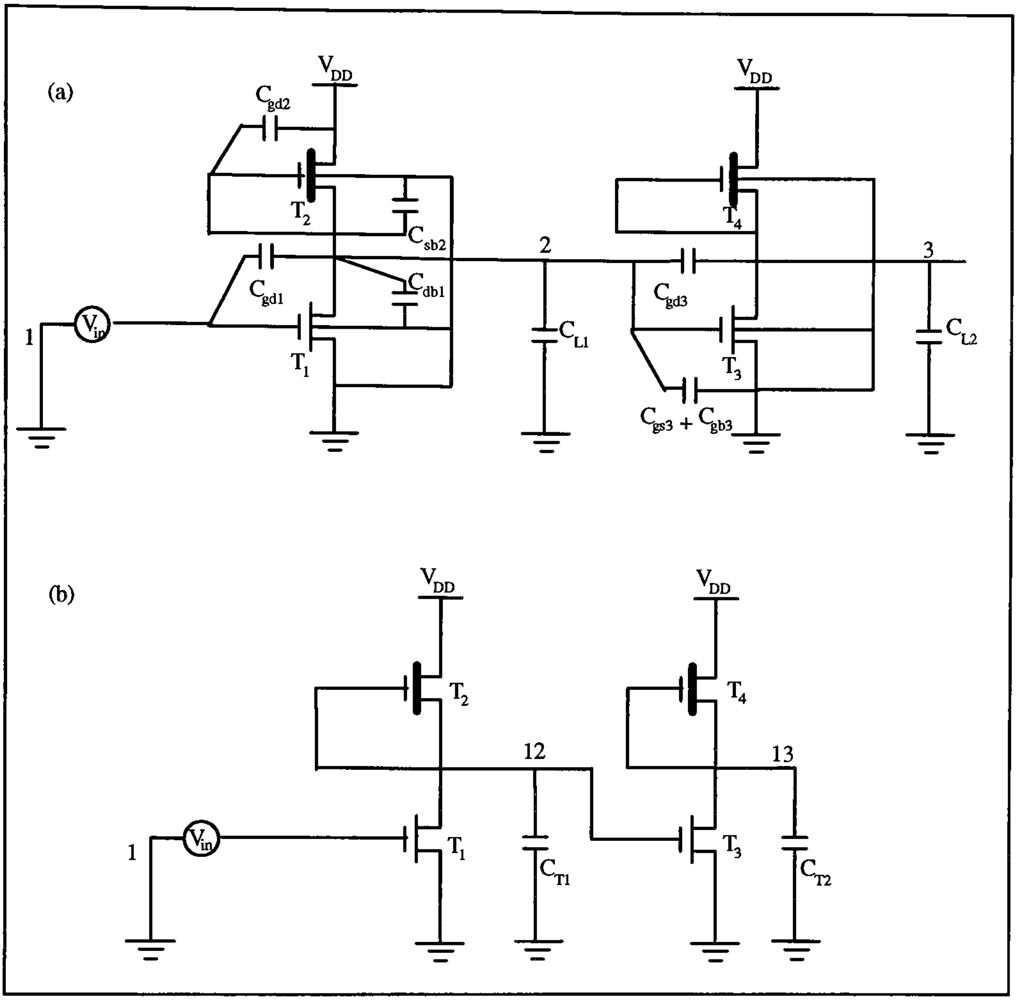

Figure 5.7: (a) Cascoded NMOS Inverters(Capacitance to be Calculated is shown only)

(b) Method of Lumping Capacitances to Inverter Output Nodes.

5.6 TRANSIENT ANALYSIS OF CMOS INVERTER

Transient analysis of CMOS inverters is carried out in very much the same way as for the NMOS inverters. A single lumped linear load capacitance at each output node is defined. Then the average currents available for charging and discharging are calculated.

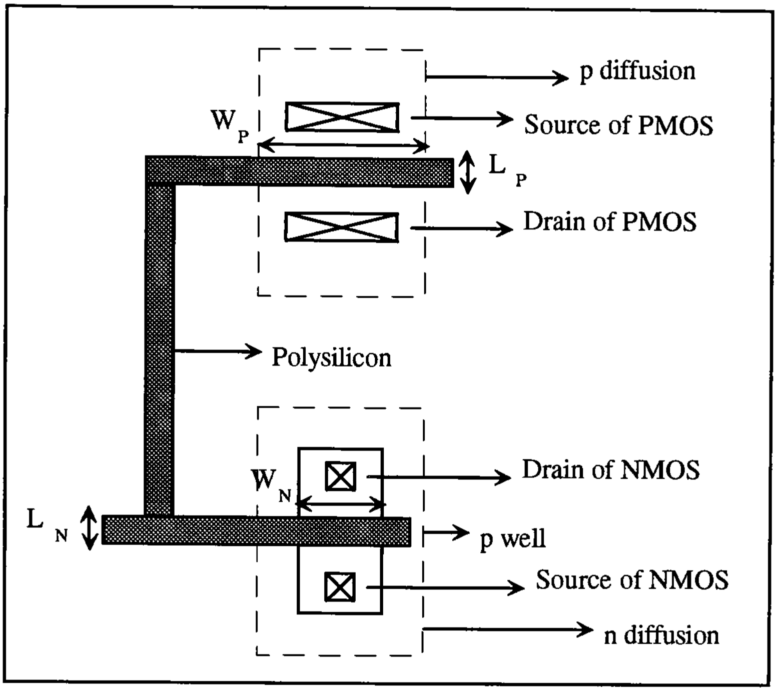

Figure 5.8: CMOS Inverter Layout

Capacitances may be calculated from a layout such as that shown in figure 5.8, provided that information on junction depths and doping concentrations is available. Either the NMOS or the PMOS device (sometimes both) must be formed in an approximately doped well, since the devices require bodies of opposite conductivity type. Figure 5.8 shows the NMOS in a p-type well, with the PMOS formed in the n-type substrate. The well will always be more heavily doped than the substrate, because it must be formed by overcompensating the initial substrate doping concentration. Consequently, junction capacitances per unit area are higher for the devices formed in wells.

Channel widths and lengths W and L shown in the figure, are typical for a near minimum size CMOS inverter. W_P/L_P must be about 2.5 times W_N/L_N to obtain approximately equal values of K_P and K_N. This equality is often desired in order to obtain equal rise and fall times when driving capacitive loads.

In figure 5.9 are shown the results of two spice simulations of a cascade of two CMOS inverters of the design shown in the layout of figure 5.8. One simulation used the full SPICE capacitance model, including all nonlinearities. For the second simulation, all model capacitances were forced to zero. Capacitive effects were calculated by hand and lumped at the output node of each inverter in the manner described for NMOS circuits. It is seen that the two simulations give similar results.

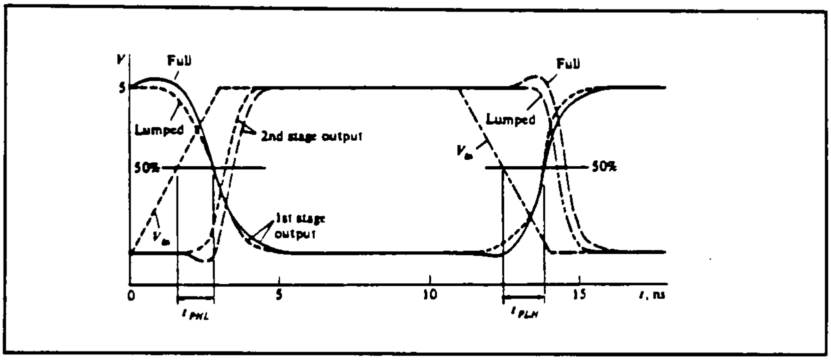

Figure 5.9: Spice Simulation of Transient Response of Two Cascaded CMOS Inverters. The Two cases are: (a) Full Model & (b) Lumped Model.

5.7 Parasitic Effects on our Circuits

We will use the above mentioned lumped model for the calculation of our circuits. Specifications for resistance and capacitance was taken from the MOSIS handout. The value of a resistance was 0.05 Ohms/square and for capacitance it was 36 aF/um^{2}. The interconnect lengths between active devices being very small, gives us low values such as 3.1 ohms for resistance and 1.723 femto-farads for capacitance. After substituting the RC networks in the appropriate place, the circuit was re simulated and a negligible change in response was observed as can be seen from the following waveforms.

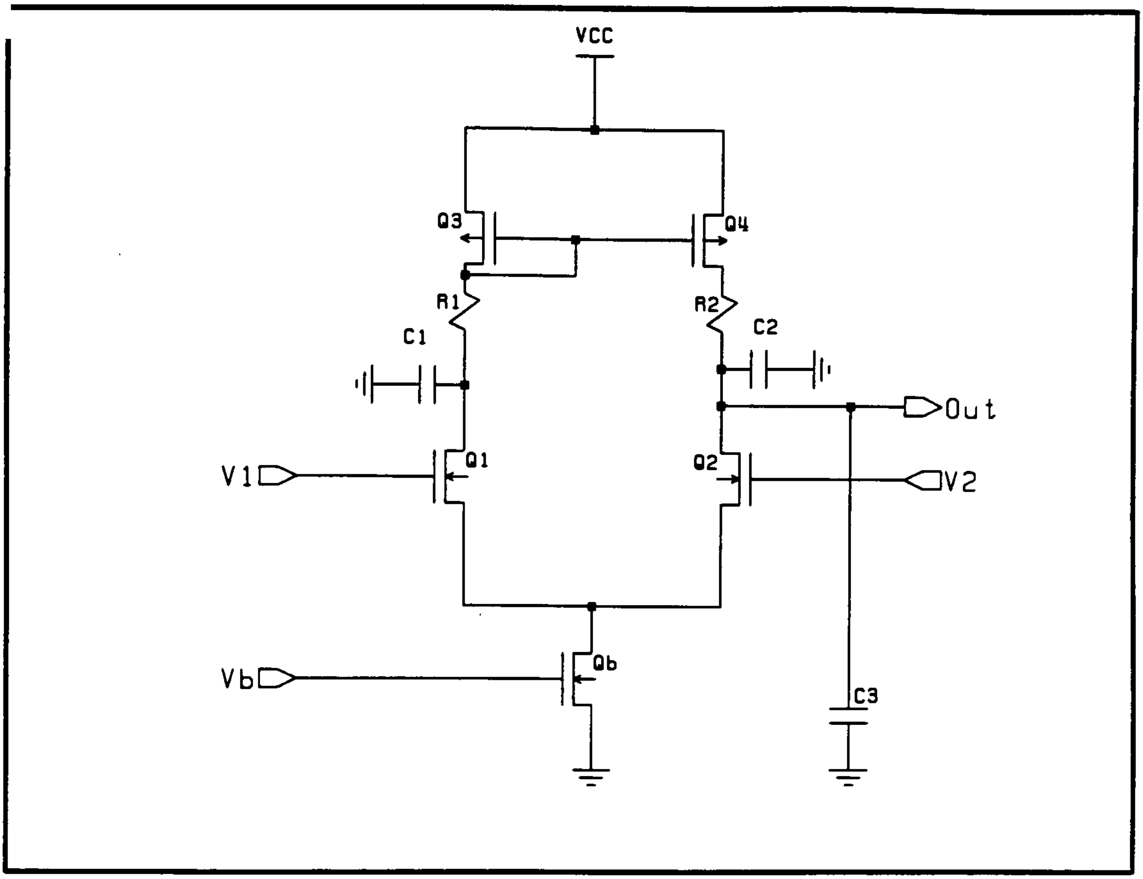

Figure 5-10: Schematic of the Simple Amplifier with Parasitic Effects.

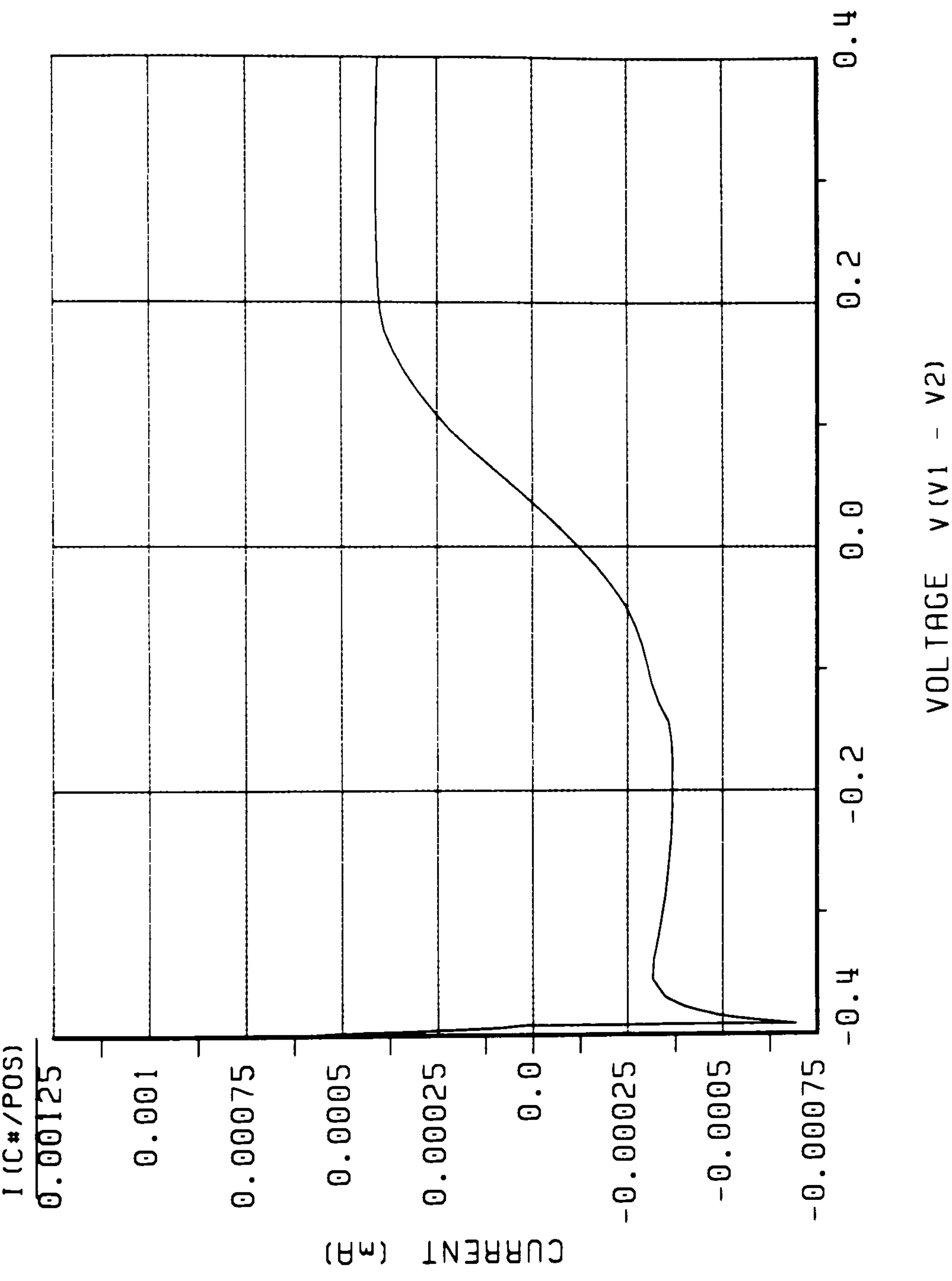

Figure 5-11: Simulated Result of a Simple Transconductance Amplifier.

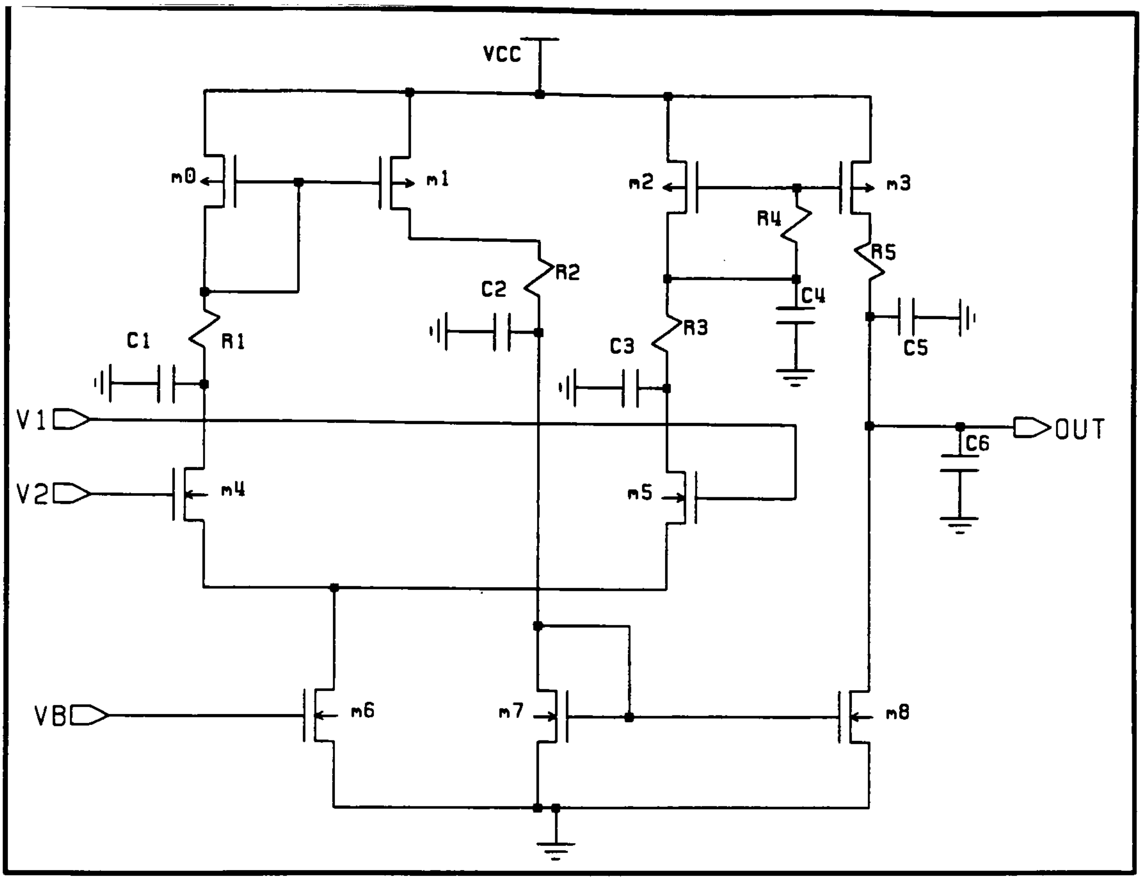

Figure 5-12: Schematic of the Wide-Range Amplifier with Parasitic Effects.

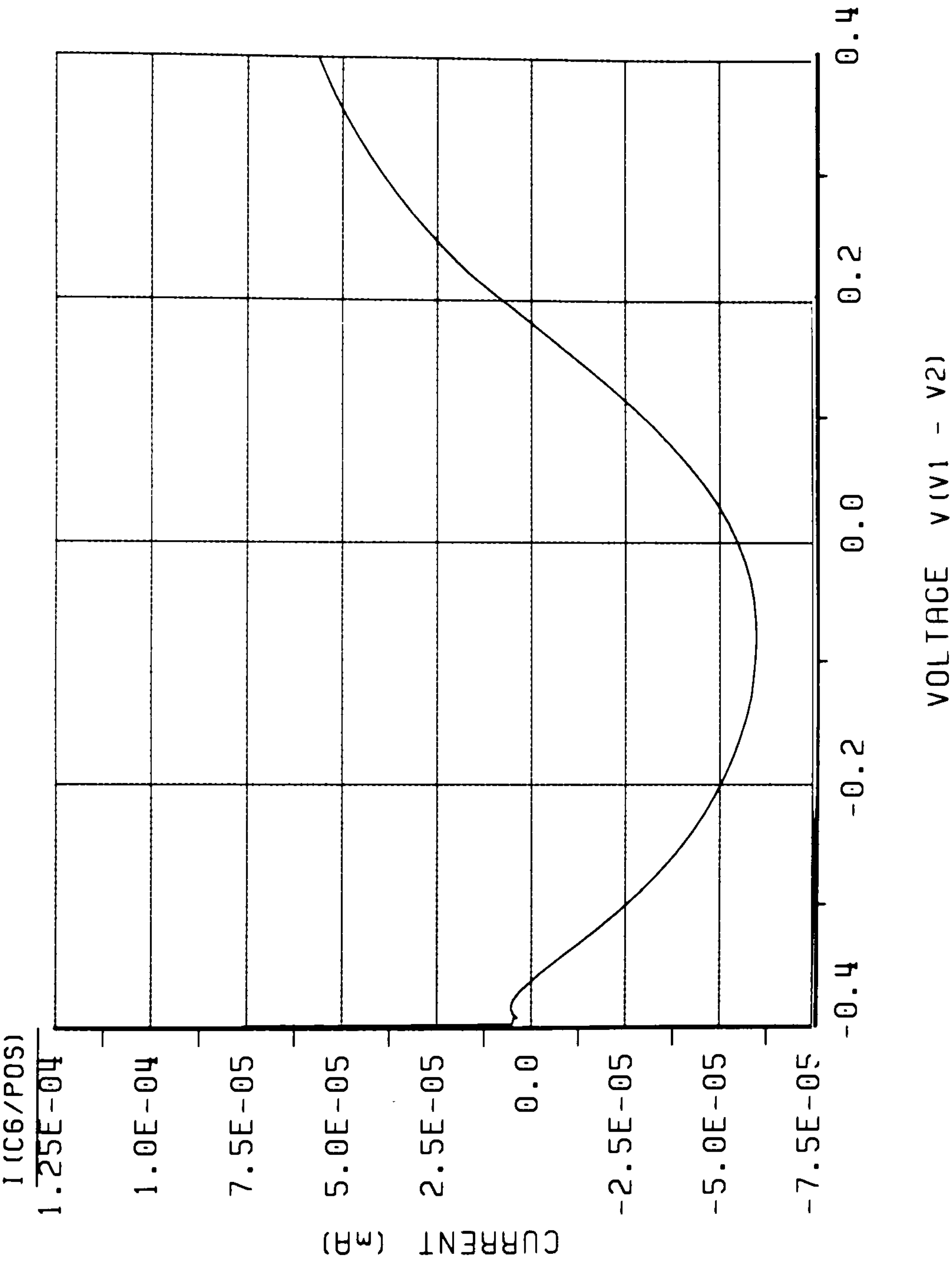

Figure 5-13: Simulation Results for the Wide-Range Transconductance Amplifier.

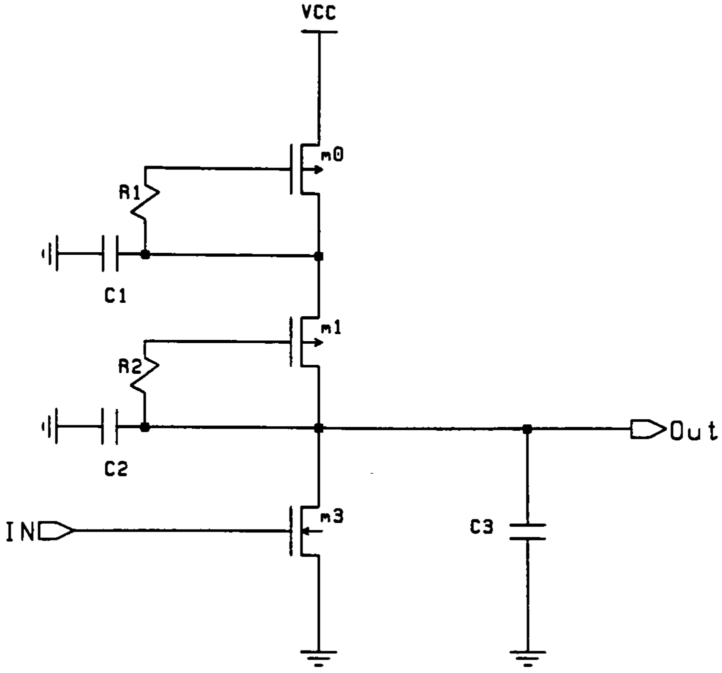

Figure 5-14: Schematic of the Photoreceptor Cell with Parasitic Effects.

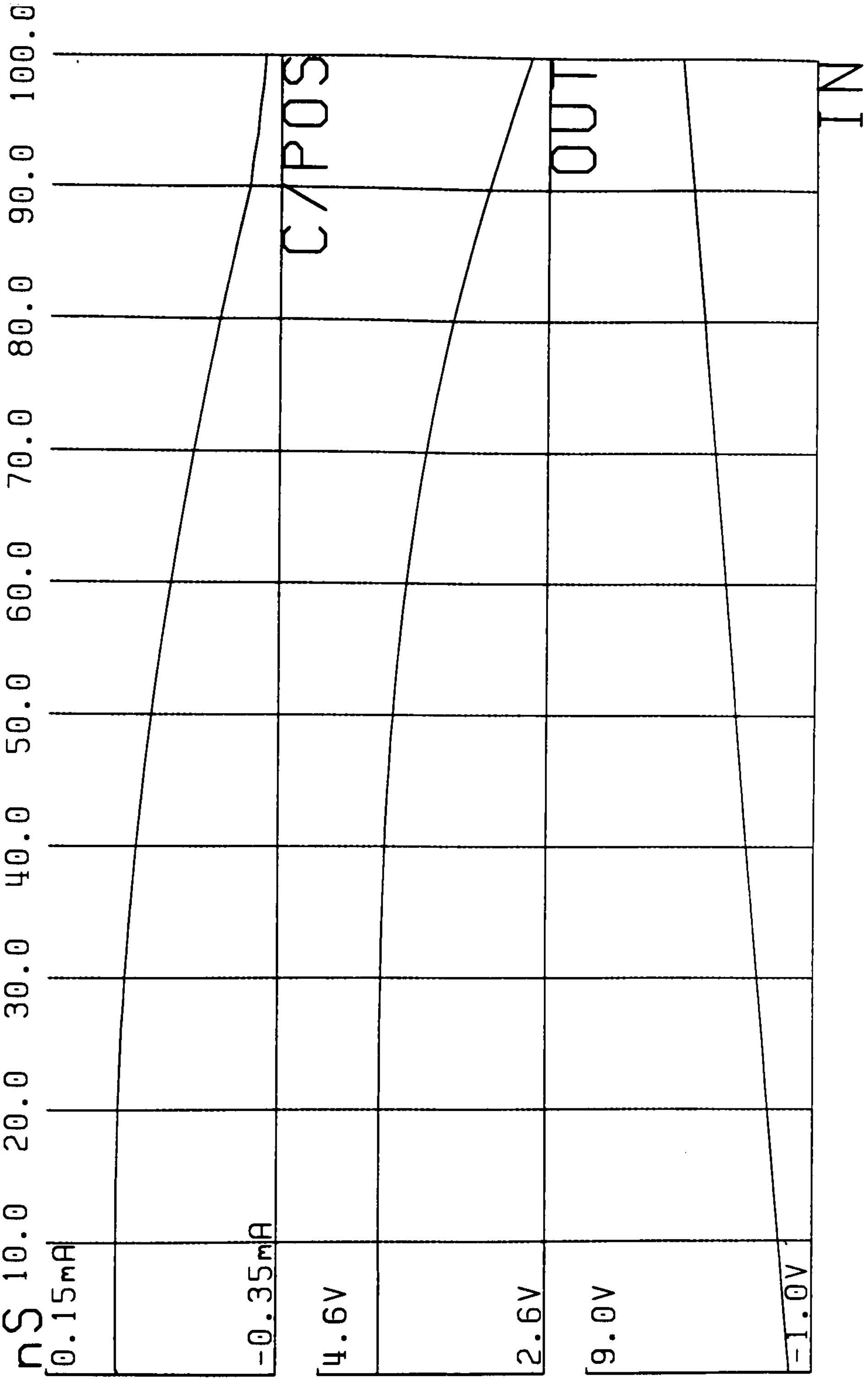

Figure 5-15: Simulation Results for the Photoreceptor cell.

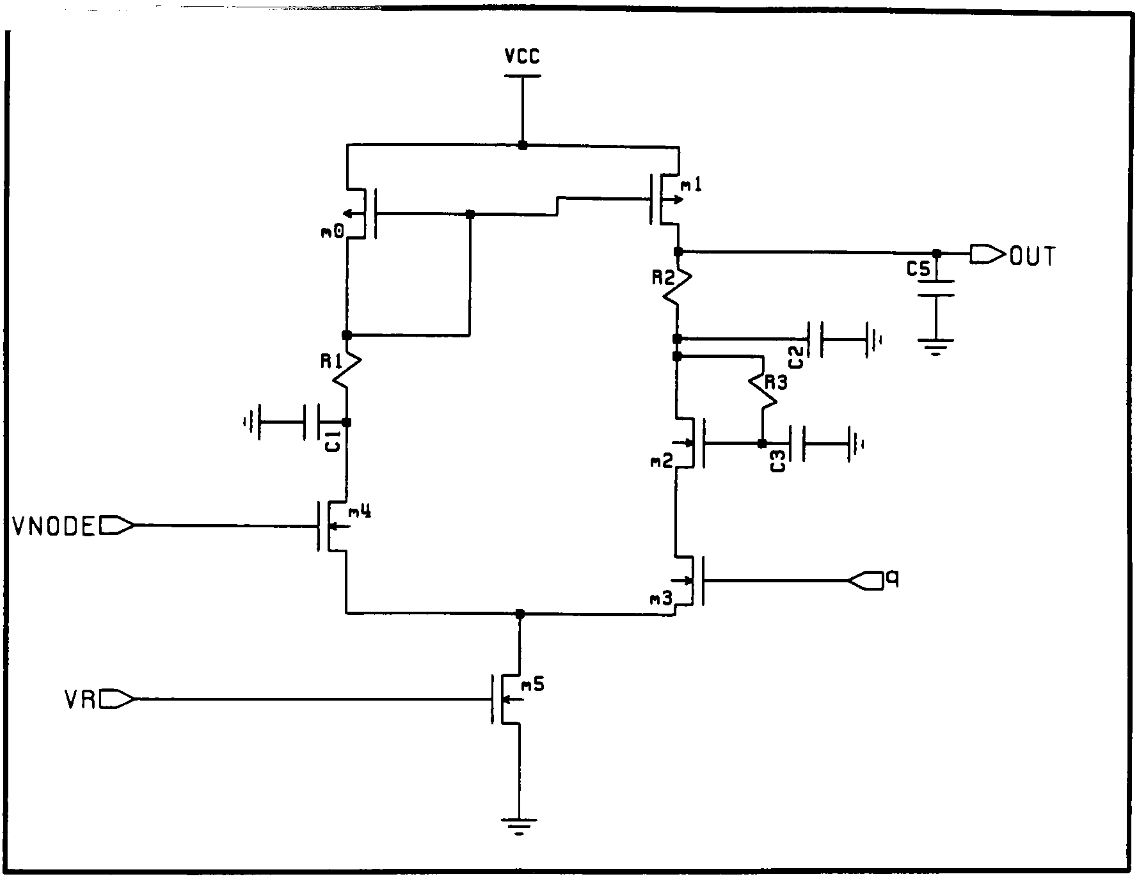

Figure 5-16: Schematic for the Horizontal Resistor Cell with Parasitic Effects.

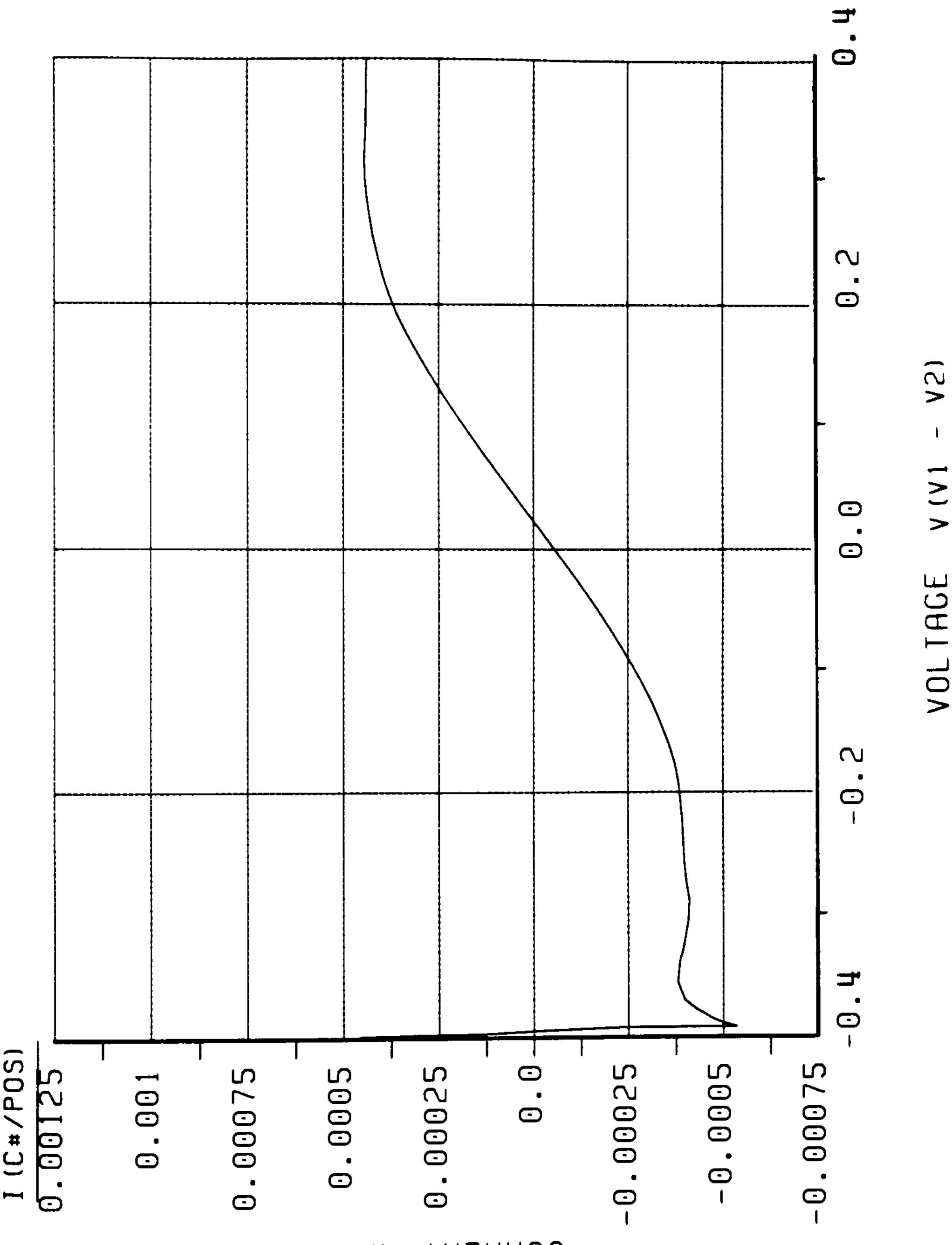

Figure 5-17: Simulation Results for the Horizontal Resistor Circuit.

Simple Amplifier	**Wide Amplifier**	**Photo-Receptor**	**Horizontal Amp.**
Q1 l=4u w=6u	Q1 l=4u w=4u	Q1 l=8u w=6u	Q1 l=4u w=5u
Q2 l=4u w=6u	Q2 l=4u w=4u	Q2 l=6u w=6u	Q2 l=4u w=5u
Q3 l=4u w=4u	Q3 l=4u w=4u	Q3 l=24u w=24u	Q3 l=3u w=5u
Q4 l=4u w=4u	Q4 l=28u w=4u	R1=2.0Ω C1=1.16ff	Q4 l=4u w=5u
Qb l=12u w=3u	Q5 l=4u w=6u	R2=5.45Ω C2=3.16ff	Q5 l=4u w=5u
R1=4.25Ω C1=2.5ff	Q6 l=4u w=6u		Q6 l=6u w=5u
R2=4.85Ω C1=2.81ff	Q7 l=13u w=4u		R1=4.21Ω C1=2.43ff
	Q8 l=4u w=4u		R2=3.4Ω C2=1.97ff
	Q9 l=28u w=4u		R3=2.6Ω C1=1.45ff
	R1=7.15Ω C1=4.14ff		
	R2=6.6Ω C2=3.82ff		
	R3=3.1Ω C3=1.79ff		
	R4=2.05Ω C4=1.18ff		
	R5=6.15Ω C5=3.56ff		

Figure 5-18: Spice Parameters for different circuits used to make the pixel.

6.0 CONCLUSION

The first and foremost problem encountered was the conversion of a single neuron into an equivalent circuit. The transconductance amplifier which is the heart of every pixel in the system, had to be very accurate and had to provide high gain and display the characteristic *tanh* curve. Since our model assumes a hexagonal resistive mesh, interconnection plays a very important role, in the sense that we try to share as many lines as possible and try to conserve silicon area. Each pixel contains the sections of global wiring necessary to form signal nets for V_{DD}, the bias controls for the resistive network, and the horizontal and vertical scan lines. The photoreceptors were located near the vertical scan line, such that alternating rows of left and right facing cells form a hexagonal array. This arrangement allows the vertical scan line wire to be shared between adjacent rows, being accessed from the left by odd rows, and from the right by even rows.

All the individual circuits were *designed* and *simulated* separately and also the *layout* for the same was carried out. Subsequently *Layout versus Schematic* checks were performed on each cell and then on the entire circuit. The final step included the *Back-Annotation*, here the parasitic capacitances and resistances were calculated by hand and were fed back to the original circuit, to observe the effect of layout routing on the actual performance of the design. It became evident that since the individual circuits were quite small, the parasitic' contribution was negligible.

We have taken the first step in simulating the computations done by the brain to process a visual image. A medium was used, that has a structure in many ways similar to neurobiological structures. Following the biological metaphor has led us to develop a system that is nearly optimal from many points of view. The constraints on our silicon system are similar to those on neurobiological systems. As in the biological retina, density is limited by the total amount of wire required to accomplish the computation. The retina, like many other areas of the brain, minimizes wire by arranging the signal representation such that as much wire as possible can be shared. The resistive network is the ultimate example of shared wiring. By including a pixel's own input in the average, we can compute the weighted average over a neighborhood for every position in the image, using the same shared structure.

It has become evident that the powerful organizing principles found in the nervous system can be realized in our most commonly available technology, namely silicon integrated circuits. Integrated circuit fabrication has evolved to the point where systems of the scale of small, but identifiable parts of the nervous system can be emulated on a single piece of silicon. The efficient mapping of a system onto its implementation medium, be it neuron or silicon, is the essence of the design problem. Once we are able to design systems of this kind, we will have extended our notion of computation into application areas that are intractable for even the largest digital computers.

Bibliography

[1] Gregorian, R. and Temes, G.C., "Analog MOS Integrated Circuits for Signal Processing", New York, Wiley, 1986

[2] Vittoz, E.A. and Fellrath, J., "CMOS Analog Integrated Circuits Based on Weak Inversion Operation", *IEEE Journal of Solid-State Circuits*, SC-12:224, 1977

[3] Feinstein, D., "The Hexagonal Resistive Network and the Circular Approximation", *Caltech Computer Science Technical Report*, Caltech-CS-TR-88-7, California Institute of Technology, Pasadena, CA, 1988

[4] German, S. and German, D., "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6:721, 1984

[5] Hutchinson, J.M. and Koch, C. "Simple Analog and Hybrid Networks for Surface Interpolation", Neural Networks for Computing, New York, American Institute of Physics, 1986

[6] Dowling, J.E. "The Retina: An Approachable Part of the Brain", Belknap Press of Harvard University Press, Cambridge, MA, 1987

[7] Enroth-Cugell, C. and Robson, J.G. "The Contrast Sensitivity of Retinal Ganglion Cells of the Cat", *Journal of Physiology*, 187:517, 1966

[8] Mead, C. and Wawrzynek, J."A new Discipline for CMOS Design", In Fuchs, H.(ed), 1985 *Chapel Hill Conference on Very Large Scale Integration. Chapel Hill*, NC, Computer Science Press, 1985.

[9] Ratliff, F. "Mach Bands", *Quantitative Studies on Neural Networks in the Retina. San Francisco*, Holden-DAY, 1965

[10] Sivilotti, M.A., Mahowald, M.A. and Mead, C.A. "Real-Time Visual Computations using Analog CMOS Processing Arrays". In Losleben, P. (ed), *Stanford Conference on Very Large Scale Integration, Cambridge*, MA, 1987

[11] Srinivasan, M.V., Laughlin, S.B., and Dubs, A. "Predictive Coding: A Fresh View of Inhibition in the Retina". *Proceedings of the Royal Society of London*, Series B, 216:427, 1982

[12] Gray, Paul R. and Meyer, Robert G. "Analysis and Design of Analog Integrated Circuits", 2nd ed., Wiley, New York, 1984.

[13] Dayhoff, Judith E. "Neural Network Architecture: An Introduction". Van Nostrand Reinhold, New York, 1990.

[14] Carpenter, Gail A. "Neural Network for Vision and Image Processing". Cambridge, Massachusetts, MIT press 1992.

[15] Wechsler, Harry. "Neural Network for Perception". Academic Press, Boston, 1992.

[16] Chester, Michael. "Neural Networks: A Tutorial". Prentice Hall, Englewood Cliffs, NJ, 1993.

[17] Mead, Carver. "Analog VLSI and Neural Systems". Addison-Wesley Publishing Company, Inc. 1989