This is to certify that the

dissertation entitled

IMPLEMENTATION OF
FEEDFORWARD ARTIFICIAL NEURAL NETWORKS
WITH LEARNING USING STANDARD CMOS TECHNOLOGY

presented by

Myung-Ryul Choi

has been accepted towards fulfillment
of the requirements for

Ph.D degree in Electrical Engineering

Major professor

Date Jan 18, 1991

IMPLEMENTATION OF FEEDFORWARD ARTIFICIAL NEURAL NETWORKS WITH LEARNING USING STANDARD CMOS TECHNOLOGY

By

Myung-Ryul Choi

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Department of Electrical Engineering

1991

ABSTRACT

IMPLEMENTATION OF FEEDFORWARD ARTIFICIAL NEURAL NETWORKS WITH LEARNING USING STANDARD CMOS TECHNOLOGY

By

Myung-Ryul Choi

A feedforward artificial neural network with a supervised learning rule is implemented using standard CMOS technology. Programmable synapse cells are implemented by employing a simple tunable all-MOS analog multiplier, where its inputs and its output are given in voltage units. Employing this multiplier, we can implement large scale artificial neural networks (ANNs) using fewer MOS transistors than are required by implementations using the so-called Gilbert multiplier.

A modified error back-propagation learning rule is used in order to realize on-chip learning circuits using standard CMOS technology. Using this learning rule, sigmoid-derivative circuits are not necessarily required for the implementation of feedforward artificial neural networks (ANNs) with learning. This learning circuit has learned a desired input/output pattern successfully for arbitrary initial weights.

A modular design is described for a large scale implementation of feedforward artificial neural networks (ANNs) with learning. Two implementations are designed

using the MAGIC VLSI editor: 4x1 module chip on the MOSIS Tinychip and 9x3 module chip on the MOSIS small standard chip. The module chips can be connected vertically and horizontally to realize large scale feedforward artificial neural networks (ANNs) with optional use of on-chip or off-chip learning capability.

To my parents,

Yun-Jin, Jae-Joon, and Jae-sig

for their love, support, and sacrifice

ACKNOWLEDGMENTS

I would like to thank my thesis advisor, Dr. Fathi M. A. Salam for his patience, guidance and encouragement throughout the course of this research.

I would like to express my gratitude to all the members of my Ph.D. guidance committee, Dr. Frank Hoppenstadt, Dr. Hassan K. Khalil, Dr. Lionel M. Ni and Dr. Gregory Wierzba for their comments and suggestions.

I would like to express my sincere thanks and appreciation to my parents, my wife parents, my sister, and my brother-in-law for their continuous love, understanding, and encouragement. Special thank to my wife, Yun-Jin, and my children, Jae-Joon, Jae-Sig, for help and understanding throughout my studies.

I also gratefully acknowledge the partial support of this work under grant from ONR and the Michigan Research Excellence Fund (REF).

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Conventional digital computers are very useful in solving well-defined problems since they can be represented by sequences of instructions. However, conventional computers can not compete with the performance of human beings in solving ill-defined random problems such as pattern recognition, classification, speech understanding, vision and so on.

The human brain can solve certain ill-defined problems easily, even though it usually takes much longer time to solve well-defined problems than does the digital computer. The human nervous system and the brain have the following characteristics: adaptive learning, generalization, error correction, robustness, and creativity. Its computational power is postulated to result from its massive interconnection and asynchronous parallel communication among neurons.

In the human brain, neuron cells become defective or dying out everyday. Yet, the human brain still functions correctly without any apparent degradation in performance. However, in a digital computer if one digital circuit element is not functioning correctly, the whole computer may be malfunctioning. The human nervous system is the best existing intelligent system because of its adaptive and learning ability. It can learn or adapt by itself.

In the nervous system, a neuron is the basic unit. It consists of dendrites, a cell body, an axon, and synapses. Synapses appear to be the most crucial component in

neural systems since the interaction between neurons appears to result from the synaptic connection and the behavior of the neural system depends critically on them. Synapses are often considered to be the distributed storage elements and their operational mechanism is still under investigation. The concept of distributed storage is quite different from that of present digital computers.

With the development of VLSI technology and with improved understanding of the human nervous system, it is possible to implement models of neural nets by mimicking some aspects of the nervous system of mammals. Models of neural nets are considerably simplified from their biological counterparts, however. When a nervous system is imitated and implemented, it is usually called artificial neural networks (ANNs). The implementation of artificial neural networks is based on a large number of simple computational components. The proposed artificial neural networks available have quite different architectures from the ones used in present day digital computers.

Many researchers have worked on the implementation of artificial neural networks (ANNs). Currently, there are many artificial neural network models. These models can be implemented via software or hardware. Software implementation usually employs an algorithm, which is based on the architecture and understanding of neural networks, mapped onto conventional digital computers. Software implementation is very flexible since one can easily modify codes or algorithms in order to implement another artificial neural network model or learning mechanism.

There are at least two disadvantages to software implementations using present day digital computers. One is that the computations among a large number of simple processors in artificial neural nets require huge communication paths between processors, which can not be supplied or which may take huge communication time by present parallel or serial digital computers. The other is that when artificial neural networks (ANNs) are modeled with differential equations, solving the differential equations of the neural network model may take a great deal of computation time.

Hardware implementation is suitable to implementing large ANNs and its computation speed is much faster than that of software implementation. Huge computation paths between processors can be supplied by connecting those processors with wire. Hardware implementation can be realized via electronics, optics, or electro-optics. This implementation, however, is not flexible but is specifically dedicated for a given application or a given artificial neural network model.

Among the many hardware implementations of artificial neural network models, the Hopfield-type feedback ANN model and the feedforward ANN model are frequently used. Most implementations are distinguished by the particular implementation of the synapse. Some implementations have implemented the synaptic weight multiplication with binary inputs and discrete set of weights. Here are some hardware implementations of artificial neural network in the literature. In [1], only one fixed weight is used. In [2, 3], three different weights, -1 (inhibitory), 0 (no connection) and +1 (excitatory), are used. Other implementations have employed the synaptic weight multiplication with binary inputs and analog programmable weights [4] or with analog inputs and binary programmable weights [5] or analog synaptic weights [6-10].

The ENN (Electronic Neural Network) by Hubbard et al. [1] has been designed with 256 neurons on a single chip using a mixture of analog and digital VLSI technology and a custom microfabrication process. This circuit employs amorphous silicon for a resistor material. Accordingly, only one fixed connection weight is available and its weight strength should be decided and fixed before fabrication.

The ANN (Artificial Neural Network) by Sage et al. [4] has been designed with the Charge-Coupled Device (CCD) and the Metal-Nitride-Oxide-Semiconductor (MNOS). This circuit operates with binary inputs and analog programmable synaptic weights and executes analog operations, but it can not be implemented with standard VLSI technology.

The ASSOCMEM (Associative Memory) by Sivilotti et al. [2, 3] has been implemented and tested in NMOS technology. This circuit has been implemented with 22 neurons on 6700 x 5700 μm^2 with 53 I/O pads. This circuit has on-chip learning circuits by using the so-called truncated Hebb rule. During learning, all the desired memory contents load onto the chip, the outer product rule is applied, and its ressults are truncated one at a time. Finally, three different synaptic weights are obtained, specifically -1 (inhibitory), 0 (no connection), and +1 (excitatory).

Using the Hebb rule, the interconnection weights are calculated from all the stored vectors. Based on simulation studies [11], the Hebb outer-product rule has been found to succeed in storing data vectors equal to, $0.15\,N$, where N is the number of neurons.

The AT&T group [10] uses SRAM (static random access memory) cells to store prespecified memories and one bit of each memory is stored at one SRAM cell. Here the stored data are presented as the interconnection matrix and only one output, which has the largest inner product value between the input vector and the stored vectors, is chosen. This design does not generate spurious steady states since it physically stores the desired data as weights. However, it requires a large number of extra memory cells to store the desired data.

At Naval Research Laboratory, researchers [6, 7] have illustrated several approaches for implementing programmable synapse multiplier circuits that perform multiplication with analog inputs and analog synapse weights. The synapse weights are calculated with a host computer using a suitable neural network algorithm and are then stored in digital form in EPROM or RAM. The digital data are loaded to digital-to-analog converter by a memory controller and thus analog voltages are written to each synapse cell by a decoding circuit, i.e., an analog voltage supplies a charge to the capacitor at the gate of an MOS transistor at each synapse cell.

At Jet Propulsion Laboratory, researchers [8] have developed hardware implementation of feedforward neural systems using several custom analog VLSI building block chips. They have implemented three different kinds of chips: a multiplexing input neuron chip, a synapse chip, and a hidden/output layer variable neuron chip. All three chips are using CMOS n-well 2 micron process with 8.2 x 8.2 mm^2 die size and 84-pin package. The overall operation is similar to the operation of the Naval Research Laboratory group [6, 7]. The synapse weights are calculated with a host computer using a suitable neural network algorithm. The basic programmable synapse cell consists of a pair of sample-and-hold circuits with address select circuitry and a folded Gilbert four-quadrant multiplier [21]. The input voltages and synapse weight voltages are stored on the gate of a 5550 μm^2 transistor and the output currents of the synapses are summed via wire connection, which is simply connected together. That is, the charge stored on the capacitor is converted into conductance through the multiplier. Then a variable-gain neuron chip sums current signals coming from the previous layer and then produces output voltages which are used for the input voltages of the next layer.

Furman and Abidi have described an analog CMOS backward error-propagation LSI (Large Scale Integration) [9]. This design is a novel cascadable CMOS IC chip with on-chip supervised learning. Each IC chip consists of 48 analog inputs and 10 analog outputs which are fully connected together. Each pin is used for two signals : one is feedforward signal as voltage and the other is backward error-propagation signal as current. Using this common pin for both processing signals, they can save a large number of pins. This IC chip can be added vertically or horizontally to increase the number of neurons per layer between a hidden layer and the next hidden layer or the output layer. An analog Gilbert multiplier [19] is used for the implementation of each synapse cell. The synapse weight voltages are stored as charges on capacitors with values 0.7 pF.

For the implementation of artificial neural nets (ANNs) using standard CMOS VLSI technology, the implementation of synapses is the most crucial and difficult one. A synapse can be implemented using an analog multiplier which multiplies incoming analog input by the stored synaptic weight. Currently, the analog MOS multiplier is implemented with the inputs in voltage and the output in current [6-9]. Then its output current is converted to voltage via current-voltage converters. Moreover, for higher dimensions, the implementation becomes complex and requires larger number of transistors.

In this thesis, feedforward artificial neural net (ANN) is implemented with on-chip learning circuit by using a modified error back-propagation rule [43]. We employ a simple tunable all-MOS analog multiplier whose inputs and output are given in voltage units in order to implement the programmable synapse cell which operates with analog inputs and analog synaptic weights. This multiplier can be used for a scalar multiplication and a vector-vector multiplication. In the implementation of feedforward and feedback neural circuit, this multiplier executes the product of a vector of (output) signals and their corresponding weights, i.e., a vector-vector multiplication. With this multiplier, we can implement larger scale artificial neural networks using fewer MOS transistors than are required by other implementations.

In this thesis, an implementation of feedforward ANNs with a supervised learning using standard analog CMOS VLSI/LSI technology is presented. Contributions of this thesis are as follows:

1. A simple tunable analog multiplier is designed using enhancement-mode MOS transistors for the implementation of feedforward ANNs with learning.
2. A CMOS operational amplifier from the literature [33] is considered and designed for the multiplier. The multiplier with CMOS operational amplifier is simulated using the PSPICE circuit simulator and is implemented using the MAGIC VLSI editor.

3. A voltage follower is used for the driving inputs for the drain of the enhancement MOS transistors of the simple multiplier.

4. Two voltage shifters/attenuators are considered and designed for the following two reasons: The operating voltage ranges are mismatched between the inputs and the output of the multiplier. In order to reduce the nonlinearity error of the multiplier, which may be caused by mobility of MOS transistor, the operating range of the multiplier is reduced in order to reduce the variation of mobility.

5. The multiplier is designed with additional circuits such as voltage followers and voltage shifters/attenuators. The performance of the multiplier is simulated using the PSPICE circuit simulator. A very high dimensional multiplier may be designed hierarchically, i.e., connecting lower-dimensional multiplier cells vertically and horizontally together.

6. A derivative of the sigmoid function is implemented as one circuit element of the feedforward learning circuitry using one 2-dimensional multiplier by assuming the sigmoidal function to be a hyperbolic tangent function, $S(x) = \tanh x$. That is, we have implemented the function, $S(x)' = (1 + S(x))(1 - S(x))$.

7. Implemented the modified error back-propagation learning rule [43] for a prototype two-layer feedforward ANNs with learning using standard MOS technology. PSPICE simulation is performed in order to demonstrate the learning capability of the circuits.

8. A module chip is considered for implementation of feedforward ANNs with on-chip learning. A module chip for a hidden layer and one for the output layer are separately designed using MOS transistors, since the learning circuitry of the output layer has sigmoid-derivative circuits but the hidden layer does not. In this case two different chips should be available to implement a two-layer feedforward ANNs with learning. Another module chip has been considered for the use of any layer since the design of two different module chips has required more design

effort than that of one module chip. The module chip has been implemented and simulated using the PSPICE circuit simulator.

9. In order to provide the flexibility of the learning capability, the module chip is designed with two learning features: on-chip learning and off-chip learning. The on-chip learning is executed by using on-chip learning circuits which realize the modified learning rule [43]. The off-chip learning is executed by using software on an off-line personal computer (PC). Given any learning rule, a PC computes the learning process and controls, via interface circuitry, the modified weights written on the on-chip analog storages.

10. A MOS capacitor is employed as a analog storage device in order to store the modified weights via the learning process. Due to leakage currents, these MOS capacitors are required to be refreshed regularly in order not to lose the stored information. This refreshing is done periodically through interface circuitry under the control of a PC or a microprocessor.

11. Two chips have been designed on a 40-pin TINYCHIP with analog pads using the MAGIC VLSI editor and have been sent for fabrication via MOSIS using 2 μm n-well CMOS process. One is an 11-dimensional multiplier chip which is implemented using 39 out of the 40 pins. The other is a 4x1 module chip which is implemented using 32 out of the 40 pins.

12. A 9x3 module chip is designed on a 64-pin MOSIS small chip with analog pads using the MAGIC VLSI editor and ready to be submitted for fabrication via MOSIS using 2 μm n-well CMOS process. This chip has used 46 out of the 64 pins.

This thesis is organized as follows. Chapter 2 reviews the model of a neuron and two artificial neural network models, namely, feedback and feedforward models. In Chapter 3, analog subcircuits are discussed for the implementation of the basic standard cells. In Chapter 4, analog multiplier cells are implemented and designed using

the MAGIC VLSI editor. PSPICE simulation is executed for the performance of this multiplier and its applications are discussed. In Chapter 5, a modified error back-propagation rule [43] is discussed and two-layer feedforward ANNs with a modified learning rule are designed and simulated using a PSPICE circuit simulator. In chapter 6, a modular design is described, then a design using the MAGIC VLSI editor is discussed. This module chip can be connected vertically and horizontally for the implementation of a large scale artificial neural system. Chapter 7 summarizes the conclusions of this research work.

CHAPTER 2

BACKGROUND

2.1 The Model of a Neuron

The basic building block in the brain is the neuron which can be abstractly and schematically depicted in Figure 2.1.1. Each neuron consists of synapses, dendrites, a cell body (or soma), and axons. Neurons communicate across synapses, receive inputs via dendrites, integrate the received information over the cell body, and transmit the signal through their axon.

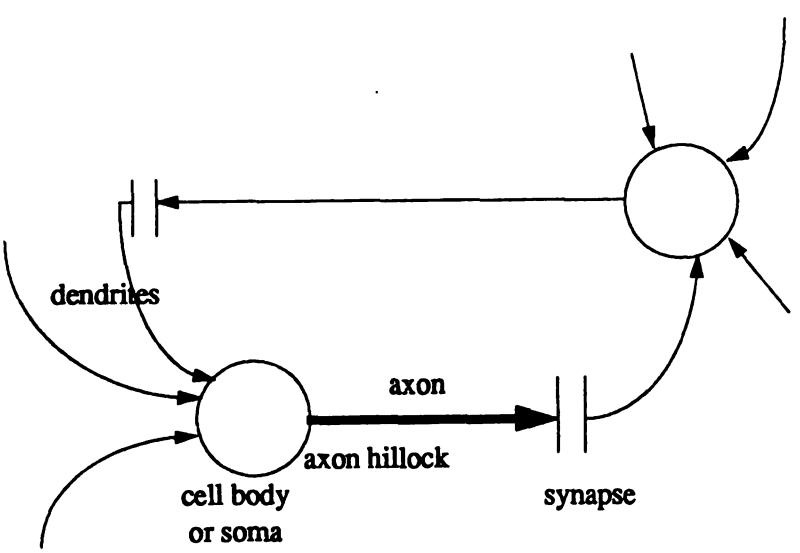


Figure 2.1.1 A biological neuron

Among these processing elements, synapses are believed to be the primary information-processing element in the neuron. The human brain is said to have about 10^{12} neurons [48] and each neuron has on average about 10^3 synaptic connections to other neurons. The collective functioning through synapses is determined largely by the way in which neural units are connected physically.

Dendrites are intricate tree of largely passive "process" or elements which aggregate synaptic input current from other neurons. This aggregate form of computation is done with slowly varying potentials control by chemical activities of neurotransmitter ions. The input charges the cell body and changes the cell membrane potential. If this potential, say u_i, is larger than a threshold voltage, say U_o, then an action potential is fired at the axon hillock located at the end of the cell body and the beginning of the axon.

This action potential arrives at the presynaptic side of synapses, and consequently neurotransmitters are released from the inside of the presynaptic membrane. The synapse is thus said to be excitatory if the postsynaptic potential becomes more positive, and inhibitory if the postsynaptic potential becomes more negative, than before.

The rate of firing of action potentials is determined by the net synaptic input current. Let T_{ij} be the synaptic conductance between the input of neuron i and the output of neuron j, and let u_i be the cell membrane input potential of neuron i. Then action potential pulses are generated at a rate dependent on the value of u_i. If the neuron has a large input, then these pulses are generated at a high rate. If the input is weak or absent, then the pulses are generated at a very low rate. The firing rate is usually represented by a smooth sigmoid function of the input u_i, say $S_i(u_i)$.

The augmented neural system is now modeled as a continuous dynamical system represented by a set of differential equations. Let C_i be the cell membrane capacitance. Then the rate of change of u_i is represented by

$$C_i \frac{du_i}{dt} = \sum_j T_{ij} V_j - \frac{u_i}{R_i} + I_i \tag{2.1.1}$$

$$u_i = S_i^{-1}(V_i)$$

$$R_i^{-1} = \rho_i^{-1} + \sum_j |T_{ij}|,$$

where T_{ij} is the linear conductance modeling the synaptic weight, ρ_i is the local resistance at the input of the cell body, I_i is an external stimulation or excitation at the input of the cell body, and $T_{ij}V_j$ is the postsynaptic current from neuron j to neuron i.

This mathematical model can be described by the neural model circuit in electrical elements, which is shown in Figure 2.1.2. A cell body is replaced by an amplifier whose input-output characteristics is a sigmoid function. Axons and dendrites are each replaced by a transmission wire and synapses are substituted by variable conductance devices.

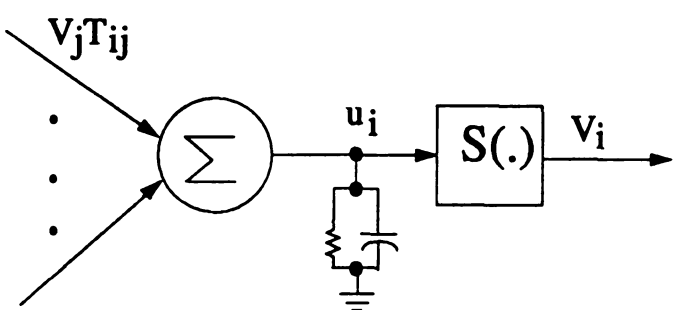


Figure 2.1.2 Neural model circuits

In the neural model circuit, excitatory and inhibitory synapses can be represented by two approaches: One is by using the sign of a conductance, i.e., a positive value for an excitatory synapse and a negative value for an inhibitory synapse. The other is by connecting a linear resistor to the output of a normal operational amplifier for an excitatory synapse and to the output of an inverting operational amplifier for an inhibitory

synapse.

2.2 Neural Network Models

Many researchers have tried to model neural network systems with current knowledge of biological neurons since it is still under research. Neural network models can be specified by the network topology, learning rule, and computational element characteristics. Currently, there are several artificial neural network models. Among them, the feedback model and feedforward model, which are discussed next.

2.2.1 Feedback Models

Two feedback neural network models are briefly reviewed: the Hopfield model and the Hoppensteadt model. The Hopfield model is shown in Figure 2.2.1.

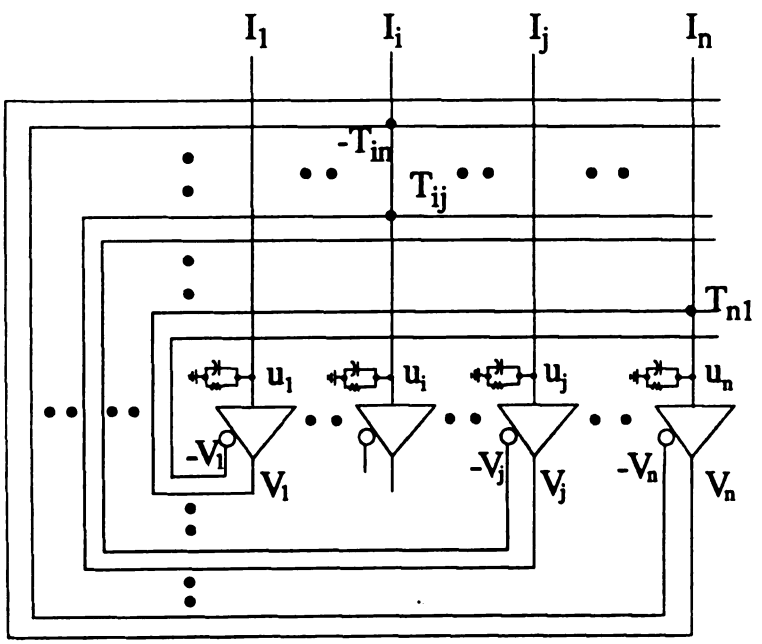


Figure 2.2.1 The Hopfield feedback model

The Hopfield model has a simple topology and is represented with binary or analog inputs and outputs. Each neuron has a nondecreasing sigmoid nonlinearity at each node. Its output is fed back to all other neurons via synaptic weights. Each weight denoted by T_{ij} connects the output of neuron j to the input of neuron i. There is a symmetry requirement on the connections, namely, we must have $T_{ij} = T_{ji}$. I_i is the external input current of neuron i.

The dynamic behavior of the Hopfield model can be examined by considering the energy function of equation (2.1.1). Equation (2.1.1) can be rewritten then as

$$C_i \frac{du_i}{dt} = - \frac{\partial E}{\partial V_i}, \tag{2.2.1}$$

where the energy function, E, is given by

$$E = - \frac{1}{2} \sum_i \sum_j T_{ij} V_i V_j + \sum_i \frac{1}{R_i} \int_0^{V_i} S_i^{-1}(V) dV - \sum_i I_i V_i. \tag{2.2.2}$$

The time derivative of the energy function along trajectories is

$$\begin{aligned} \frac{dE}{dt} &= - \sum_i \frac{\partial E}{\partial u_i} \frac{du_i}{dt} \\ &= - \sum_i \frac{\partial E}{\partial V_i} \frac{dV_i}{du_i} \frac{du_i}{dt} \\ &= - \sum_i \frac{1}{C_i} \frac{dV_i}{du_i} \left[\frac{\partial E}{\partial V_i} \right]^2 \\ &= - \sum_i \frac{1}{C_i} \frac{dS_i(u_i)}{du_i} \left[\frac{\partial E}{\partial V_i} \right]^2 \\ &\leq 0, \end{aligned} \tag{2.2.3}$$

since $C_i \geq 0$ and $S_i(u_i)$ is a monotone nondecreasing sigmoid function. Therefore, this system is a gradient-like system. That is, this energy function decreases along trajectories and its time-derivative equals zero at $\frac{\partial E}{\partial V_i} = 0$, which is an equilibrium point

of this system.

The Hopfield model, as proposed by Hopfield [11], employs the Hebb rule as its learning law. The Hebb rule is simply described by Hopfield [11] as

$$T_{ij} = \sum_{s=1}^{M} V_i^s V_j^s, \tag{2.2.4}$$

where T_{ij} is the synapse weight from the output of neuron j to the input of neuron i, V_i^s is the ith bit of stored memory s and M is the number of the stored memories. Since equation (2.2.4) is the addition of the outer product terms, equation (2.2.4) can be represented in matrix form as

$$T = \sum_{s=1}^{M} V^{s^T} V^s. \tag{2.2.5}$$

If further learning is required with additional desired pattern V^a, then equation (2.2.5) can be recursively written as

$$T_{new} = T_{old} + V^{a^T} V^a, \tag{2.2.6}$$

where T_{new} is the new synapse matrix and T_{old} is the old synapse matrix which was modified by the new desired patterns.

The operation of this neural network model is described follows: First, synapse weights are set using some learning, such as the Hebb rule, from desired patterns. Then an unknown pattern is applied to initialize the net. After initialization, the net evolves through its dynamics. Finally, the outputs of the net converge to their steady state values.

The Hopfield model is proposed as an associative memory or to solve optimization problems [11-13]. However, the Hopfield model with the Hebb rule is severely limited by the number of desired patterns which can be stored and accurately remembered when it is used as an associative memory [11].

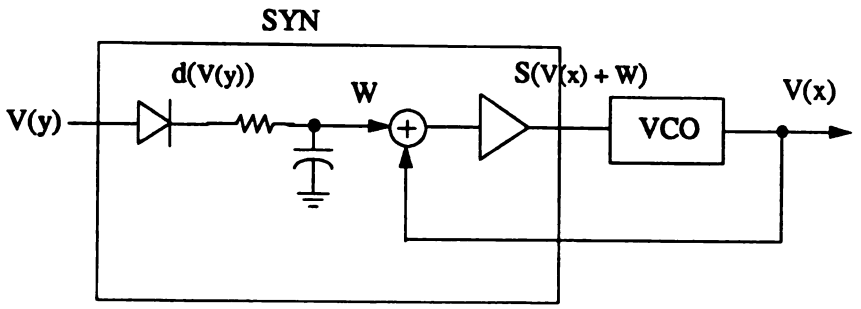


Figure 2.2.2 The Hoppensteadt model

Another feedback neural network model is introduced by Hoppensteadt, say the Hoppensteadt model, which is closer to the biological neuron. Figure 2.2.2 shows a voltage-controlled oscillator neuron (VCON), which is the neuron circuit model of the Hoppensteadt model. A VCO behaves analogous to the cell body, where repetitive action potentials are generated. A SYN is emulating the synapse function. A SYN is realized with a low-pass filter, which models the synaptic gap, and a comparison amplifier using a voltage adder and a linear amplifier. A comparison amplifier sums the output of the low-pass filter, say W, and the postsynaptic potential, say U. This circuit model is represented by the following mathematical model [32]:

$$RC\frac{dW}{dt} = -W + d(V(y)) \tag{2.2.7}$$

$$\frac{dx}{dt} = \omega_o + P[V(x) + W], \tag{2.2.8}$$

where ω_o is the VCO's center frequency, P is a bounded continuous differential monotone increasing function, and $d(V)$ is the positive part of V through a rectifying diode.

A VCON generates its output, $V(x)$, at the rate of the phase x which is the function of W and $V(x)$. That is, $V(x)$ is generated at a high rate when W is positive (excitatory) and generated at a low rate when W is negative (inhibitory). In other

wards, W is positive if the synapse is excitatory and negative if the synapse is inhibitory.

The VLSI implementation of the Hoppensteadt model is not presently available since a VCON circuit is not simple to implement using standard MOS technology.

2.2.3 Feedforward Models

Multilayer feedforward neural nets have one or more layers of computational nodes between the input and output nodes. Consider the basic structure of multilayer feedforward ANNs in Figure 2.2.3, where outputs of any layer are weighted and summed as an input to a neuron in the next layer [31].

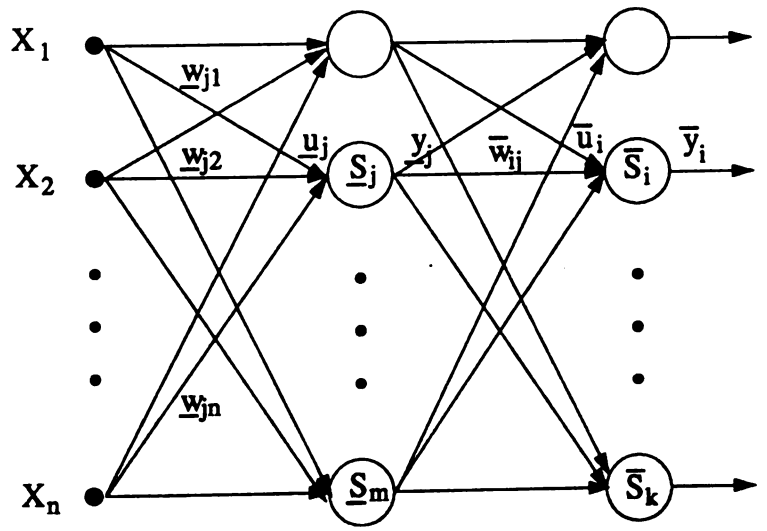


Figure 2.2.3 General two-layer feedforward ANNs

An external input is applied to the first layer, which is called the input layer and fed forward to the last layer, which is called the output layer. Any layer between the input layer and the output layer is called a hidden layer. These layers are cascaded to form

the multilayer feedforward ANNs. The governing static equation for each neuron unit in any layer may be described as

$$y_j = S_j(\sum_i w_{ji} x_i + \theta_j), \tag{2.2.9}$$

where y_j is the output of the j-th neuron, $S_j(.)$ is a nonlinear monotone nondecreasing sigmoid function, w_{ji} is the connection weight from the i-th output of the previous layer to the j-th neuron input, x_i is the i-th output of a neuron unit in the previous layer, and θ_j is a threshold at the input of the j-th neuron.

For a supervised learning rule of feedforward ANNs, the error back-propagation rule was proposed by Rumelhart et al. [31]. For training the desired output or target vector, the input vector is applied to this network and its output vector is produced, which is compared with the desired output vector. If there is an error, the weights are modified to reduce the error. And this process is iteratively continued until its error is minimized.

The total squared error function is given by

$$E = \sum_p E_p \tag{2.2.10}$$

$$= \frac{1}{2} \sum_p \sum_j (t_{pj} - y_{pj})^2, \tag{2.2.11}$$

where E_p is the squared error function for the desired target p, say $t_p^T = [t_{p1} \cdots t_{pn}]$ and y_{pj} is the actual output for the target p.

The error back-propagation rule is given as follows [31]:

$$w_{ji}^k = w_{ji}^{k-1} + \sum_p \Delta_p w_{ji}^k \tag{2.2.12}$$

$$\Delta_p w_{ji} = -\eta \frac{\partial E_p}{\partial w_{ji}}$$

$$= \eta \delta_{pj} y_{pi}, \tag{2.2.13}$$

where $\Delta_p w_{ji}^k$ is the change of the weight w_{ji} at the k-th iteration for the desired target p, η is the learning rate which is sufficiently small positive value, and y_{pi} is the output of the previous layer.

If a unit j is in the output layer, δ_{pj} is given by

$$\delta_{pj} = \frac{dS_j}{du_j}[t_{pj} - y_{pj}]. \tag{2.2.14i}$$

If a unit j is in the hidden layer, δ_{pj} is given by

$$\delta_{pj} = \frac{dS_j}{du_j} \sum_k \delta_{pk} w_{kj}, \tag{2.2.14ii}$$

where k is the index for the units in the next layer to which a unit j is connected.

This learning rule is iteratively computed in discrete time using equation (2.2.13) and equation (2.2.14).

2.3 MOS Transistor

The MOS transistor plays a central role for all our analog implementation of artificial neural nets [6, 7, 8, 9, 27, 28, 29, 30]. Here we present its basic features as available in the literature for completeness. Consider the simplified 3-dimensional structure shown in Figure 2.3.1.

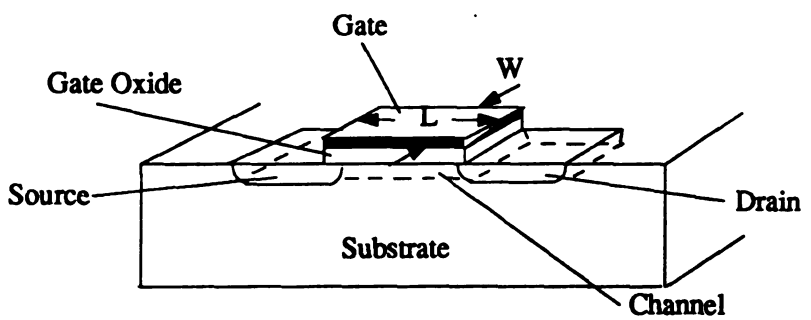


Figure 2.3.1 A simplified 3-dimensional view of a MOS transistor

It consists of several layers: from top to bottom, it contains a metal layer, a silicon dioxide (S_iO_2) layer, a p-type or n-type silicon layer, and another metal layer. It is called a metal-oxide-semiconductor (MOS) structure, where the silicon dioxide layer is an excellent insulator and the bottom metal layer is connected to the lowest voltage, V_{ss}, for an n-well process and the highest voltage, V_{dd}, for a p-well process.

There are two heavily doped n-type (n^+-type) regions in the p-type material, which are respectively called drain, whose voltage is denoted by v_D, and source, whose voltage is denoted by v_S. These are separated by L, which is called channel length. The top metal electrode is called the gate whose voltage is denoted by v_G. The body of the semiconductor is usually called the substrate or bulk whose voltage is denoted by v_B. Since this structure is symmetric, the roles of the source and the drain are interchangable. This overall device is called the nMOS transistor.

For the nMOS transistor, when a positive voltage is applied to the gate, electrons start to deplete the substrate near the surface under the gate and form a depletion region under the gate. If more positive voltage is applied to the gate, then more electrons will be attracted to the substrate surface under the gate and this region is changed from p-type to n-type, i.e., inverted. This n-type region is called an inversion layer or channel. The gate-source voltage which is required to create the inversion layer is called the threshold voltage, V_T.

The pMOS transistor is similarly obtained by forming p^+-type region in the n-type material. Both nMOS and pMOS transistors are four-terminal devices. The MOS transistor size is represented by the channel width, W, and the channel length, L, which are depicted in Figure 2.3.1. W/L is called the aspect ratio of the MOS transistor.

The convention of electrical variables of MOS transistors are shown in Figure 2.3.2. Along with this convention, a mathematical MOSFET dc model is obtained in order to predict the experimental characteristics for a given aspect ratio, process

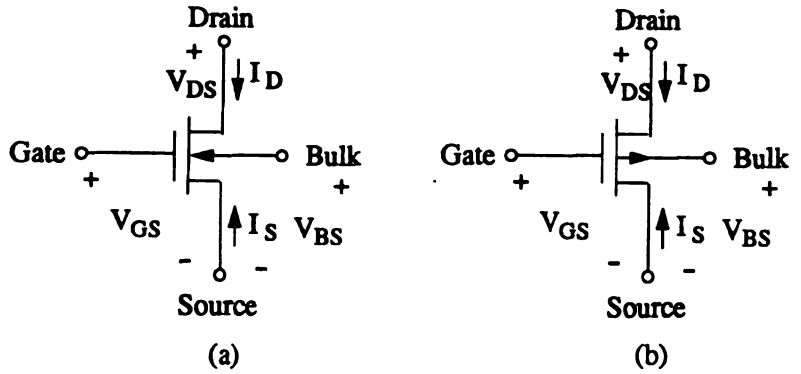


Figure 2.3.2 Convention for electrical variables for MOS transistors
(a) nMOS transistor (b) pMOS transistor

parameters, and applied voltages. There are three different operating regions: cutoff region, ohmic (or triode) region, and saturation region.

Assuming that the channel is sufficiently long and wide and the substrate is uniformly doped. For an nMOS transistor, the drain current is approximated by the following quadratic function:

Cutoff: if $(v_{GS} - v_T) \leq 0$

$$i_D = 0 \tag{2.3.1}$$

Triode: if $(v_{GS} - v_T) \geq (v_{DS})$

$$i_D = u_o C_{ox}(W/L)\left[(v_{GS} - v_T)v_{DS} - \frac{v_{DS}^2}{2}\right](1 + \lambda V_{DS}) \tag{2.3.2}$$

Saturation: if $(v_{GS} - v_T) \leq (v_{DS})$

$$i_D = \frac{u_o C_{ox}}{2}(W/L)(v_{GS} - v_T)^2 (1 + \lambda V_{DS}) \tag{2.3.3}$$

with the threshold voltage

$$v_T = V_{TO} + \gamma(\sqrt{\phi - v_{BS}} - \sqrt{\phi}), \tag{2.4.4}$$

where μ, C_{ox}, λ, V_{T0}, γ, and ϕ are process parameters :

μ = channel mobility

C_{ox} = capacitance density of the gate-channel capacitor

λ = channel length modulation

V_{T0} = threshold voltage for v_{BS}=0

γ = bulk threshold parameter

ϕ = strong inversion surface potential.

For a pMOS transistor, we can obtain the drain current i_D similarly by taking the negative sign of the current and voltage values, i.e., i_D is substituted for $-i_D$ and v_x is replaced by $-v_x$, where v_x is any one of v_D, v_S, and v_T. That is,

Cutoff: if $(v_{GS} - v_T) \geq 0$

$$i_D = 0 \tag{2.3.5}$$

Triode: if $(v_{GS} - v_T) \leq (v_{DS})$

$$i_D = -u_o C_{ox}(W/L)\left[(v_{GS} - v_T)v_{DS} - \frac{v_{DS}^2}{2}\right](1 - \lambda V_{DS}) \tag{2.3.6}$$

Saturation: if $(v_{GS} - v_T) \geq (v_{DS})$

$$i_D = -\frac{u_o C_{ox}}{2}(W/L)(-v_{GS} + v_T)^2(1 - \lambda V_{DS}) \tag{2.3.7}$$

with the threshold voltage

$$v_T = V_{TO} - \gamma(\sqrt{\phi + v_{BS}} - \sqrt{\phi}). \tag{2.3.8}$$

nMOS devices become more popular because of higher mobility of electrons, which makes nMOS transistors faster than pMOS transistors.

As VLSI MOS technology is developing, the device minimum size is getting smaller. In analog circuit designs using MOS technology, the channel length and the channel width are very important. The above equations are derived with the assumption of a sufficiently long and sufficiently wide channel. If the channel is not sufficiently long or sufficiently wide, these equations are changed a little [36].

Moreover, for short channels, MOS transistors face another problem, namely, limitations on voltages. For example, when an analog circuit is designed using 2 μm minimum feature size of a transistor, as supplied via MOSIS, the maximum allowable voltage difference is only 10 volts.

CHAPTER 3

BASIC ANALOG SUBCIRCUITS

The following basic analog subcircuits are described in order to integrate analog multiplier and feedforward artificial neural nets (ANNs) with learning as building block scheme: a simple tunable analog MOS multiplier, a CMOS operational amplifier, and voltage shifters/attenuators. These basic analog MOS subcircuits have been designed and implemented using standard CMOS technology:

A simple tunable analog scalar/vector multiplier [15, 16, 17, 23] is used to implement an artificial synapse cell by performing a multiplication analog inputs by analog synaptic weights. In order to implement an artificial synapse cell of large scale ANNs, the multiplier is designed with a CMOS operational amplifier, a voltage follower, and voltage shifters/attenuators. A follower integrator is used to perform the integration of the learning algorithm and is realized with a voltage follower and a MOS capacitor. A MOS capacitor is also used as an analog storage element to store a synaptic weight. A CMOS double inverter is used to realize a cell body, whose input-output characteristics are represented by a sigmoid function. An analog multiplexer is addressed to read or write synaptic weights on on-chip analog memories via off-chip interface circuitry.

All the basic analog MOS subcircuits are simulated using the PSPICE circuit simulator, which is a commercial circuit simulator.

3.1. A Simple Tunable MOS Analog Multiplier

The output voltage of a multiplier is uniquely determined by any pair of inputs which are in the operating range. Four-quadrant multipliers respond to all four combinations of input signals and provide output signals of correct polarity. Presently, some analog MOS multipliers have been implemented using the MOS version of the Gilbert multiplier which performs multiplication with very good accuracy. However, these multipliers seem to take a large number of transistors for the implementation of a vector-vector multiplier which implements a synapse cell for large scale ANNs.

Here we employ a simple tunable four-quadrant analog multiplier [15, 16, 17, 23] in order to implement a synapse cell of a large scale ANN. This multiplier is easy to implement as high dimensional vector-vector multiplier with less number of transistors than the existing multipliers.

Consider four matched nMOS transistors which are inside the box in Figure 3.1.1 [15, 16, 17, 23].

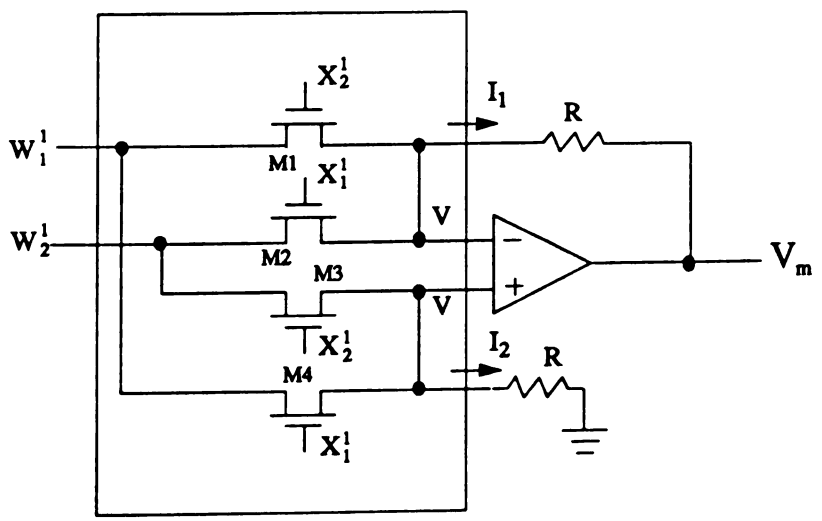


Figure 3.1.1 A simple four-quadrant analog multiplier

The complete expression of the drain current in the nonsaturation ohmic region is represented by [36]

$$I_D = g(V_G,V_D) - g(V_G,V_S), \tag{3.1.1}$$

which shows the symmetry of drain and source and where

$$g(V_x,V_y) = K[\,(V_x - V_B - V_{FB} - \phi_B)V_y - \frac{1}{2}(V_y - V_B)^2 - \frac{2}{3}\gamma(V_y - V_B + \phi_B)^{\frac{3}{2}}\,]$$

with $K = \mu C_{ox}\frac{W}{L}$ and $\gamma = \frac{\sqrt{2qN_A\varepsilon_s}}{C_{ox}}$.

Assuming all transistors are matched, operate in the ohmic region, have the same constant mobility, and the operational amplifier is ideal. Then given terminal inputs X_1, X_2, W_1 and W_2 [16],

$$\begin{aligned}
I_1 - I_2 &= I_{D_1} + I_{D_2} - I_{D_3} - I_{D_4} \\
&= g(X_2,W_1) - g(X_2,V) + g(X_1,W_2) - g(X_1,V) \\
&\quad - g(X_2,W_2) + g(X_2,V) - g(X_1,W_1) + g(X_1,V) \\
&= K[\,(X_2 - V_B - V_{FB} - \phi_B)W_1 - \frac{1}{2}(W_1 - V_B)^2 - \frac{2}{3}\gamma(W_1 - V_B + \phi_B)^{\frac{3}{2}} \\
&\quad + (X_1 - V_B - V_{FB} - \phi_B)W_2 - \frac{1}{2}(W_2 - V_B)^2 - \frac{2}{3}\gamma(W_2 - V_B + \phi_B)^{\frac{3}{2}} \\
&\quad - (X_2 - V_B - V_{FB} - \phi_B)W_2 + \frac{1}{2}(W_2 - V_B)^2 + \frac{2}{3}\gamma(W_2 - V_B + \phi_B)^{\frac{3}{2}} \\
&\quad - (X_1 - V_B - V_{FB} - \phi_B)W_1 + \frac{1}{2}(W_1 - V_B)^2 - \frac{2}{3}\gamma(W_1 - V_B + \phi_B)^{\frac{3}{2}}\,] \\
&= K\,(X_2 - X_1)(W_2 - W_1).
\end{aligned}$$

That is,

$$I_1 - I_2 = K\,(X_2 - X_1)(W_2 - W_1). \tag{3.1.2}$$

Therefore, its output V_m can be represented by

$$V_m = R\ (I_1 - I_2)$$

$$= K\ R\ (W_1 - W_2)\ (X_1 - X_2), \tag{3.1.3}$$

where $K = \mu C_{ox}(W/L)$ is the transconductance parameter of the nMOS transistors for the feedforward devices. Since equation (3.1.1) is subject to operate in the triode region, the following operating conditions should be satisfied [27]:

$$W_1, W_2 \leq \min\ [\ (X_1 - V_T), (X_2 - V_T)\ ]$$

or

$$X_i - W_j \geq V_T > 0 \qquad \text{for } all\ i, j = 1,2. \tag{3.1.4}$$

The multiplier shown in Figure 3.1.1 can be implemented with only MOS transistors by replacing the two resistors by another matched four nMOS transistors. This all-MOS analog multiplier is depicted in Figure 3.1.2.

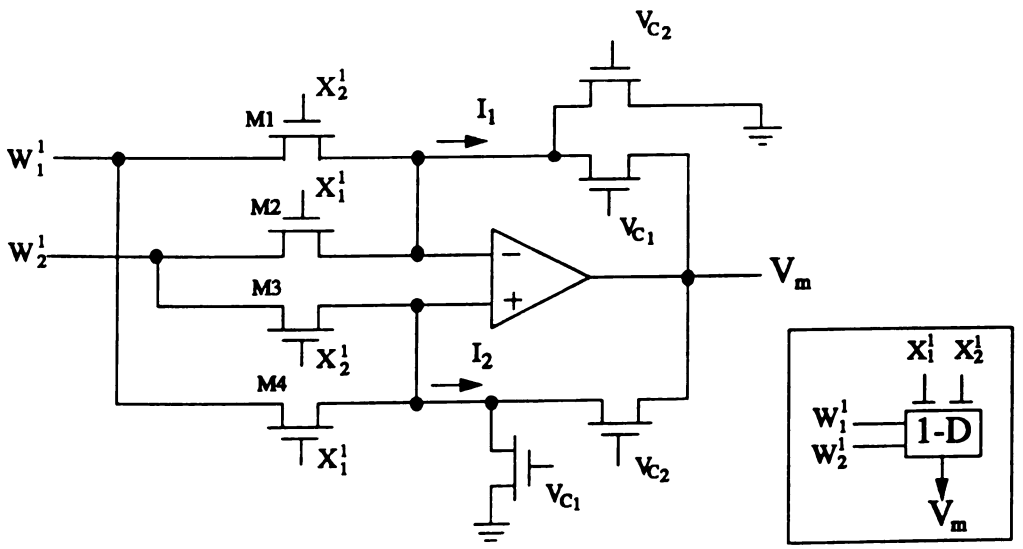


Figure 3.1.2 All-MOS simple four-quadrant analog multiplier

It is assumed that all transistors are matched, all transistors operate in the ohmic region, have the same constant mobility, and the operational amplifier is ideal. For the terminal voltages V_{c_1}, V_{c_2}, V_m, and $V_k = 0$, we can derive the following equation from equation (3.1.1):

$$-(I_1 - I_2) = K_r(V_{c_1} - V_{c_2})(V_m - V_k). \tag{3.1.5}$$

Therefore, the output of this multiplier can be represented by [27, 28]

$$V_m = \frac{K_1(W_1 - W_2)(X_1 - X_2)}{K_r(V_{c_1} - V_{c_2})}, \tag{3.1.6}$$

where $K_1 = \mu C_{ox}(W/L)_1$ is the transconductance parameter of the nMOS transistor for the feedforward devices and $K_r = \mu C_{ox}(W/L)_r$ is the transconductance parameter of the nMOS transistor for the feedback devices. V_{c_1} and V_{c_2} are used for tuning and to specify the multiplier constant for given K_1 and K_r. This multiplier may be used as a divider when V_{c_1} and V_{c_2} are used as the dividend with fixed $W_{i,j}$. The operating conditions of this multiplier are given by [27, 28]

$$\begin{aligned} &X_i - W_j \geq V_T > 0 \qquad \text{for } \textit{all } i, j = 1,2, \text{ and} \\ &V_{c_i} - V_T \geq 0 \text{ and } V_{c_i} - V_T \geq V_m \qquad \text{for } \textit{all } i = 1,2. \end{aligned} \tag{3.1.7}$$

The multipliers in Figure 3.1.1 and Figure 3.1.2 can be extended for the multiplication of two n-dimensional vectors, i.e., n-D vector inner product when additional $(n-1)$-four matched nMOS transistors are connected to the input nodes of the operational amplifier [27]. This multiplier will be called the n-D multiplier. Then equation (3.1.3) can be extended to read as follows [27]:

$$V_m = R \sum_{i=1}^{n} K_i \, (W_1^i - W_2^i)\,(X_1^i - X_2^i), \tag{3.1.8}$$

where the superscript i is used as an index. Assume all nMOS transistors have the same aspect ratio (W/L), equation (3.1.8) becomes [27]

$$V_m = \mu C_{ox} R (W/L) \sum_{i=1}^{n} (W_1^i - W_2^i)(X_1^i - X_2^i) \tag{3.1.9}$$

under the following operating conditions [27, 28]:

$$X_i^k - W_j^k \geq V_T > 0 \quad \text{for } \textit{all } i, j = 1,2 \text{ and } k = 1,2,...,n. \tag{3.1.10}$$

Similar to the above n-D analog multiplier, this multiplier can be implemented using MOS transistors only. An all-MOS n-D four-quadrant multiplier is shown in Figure 3.1.3 [27, 28]. Similarly with n-four matched nMOS transistors, equation (3.1.6) can be extended to the vector inner product [27, 28]

$$V_m = \frac{1}{K_r (V_{c_1} - V_{c_2})} \sum_{i=1}^{n} K_i (W_1^i - W_2^i)(X_1^i - X_2^i), \tag{3.1.11}$$

where the superscript i is used as an index. Assume all the nMOS transistors of the feedforward devices have the same aspect ratio (W/L), equation (3.1.11) becomes [27, 28]

$$V_m = \frac{(W/L)_i}{(W/L)_r (V_{c_1} - V_{c_2})} \sum_{i=1}^{n} (W_1^i - W_2^i)(X_1^i - X_2^i). \tag{3.1.12}$$

Similar to equation (3.1.7), the operating conditions are given by [27, 28]

$$X_i^k - W_j^k \geq V_T > 0 \quad \text{for } \textit{all } i, j = 1,2 \text{ and } k = 1,2,...,n \text{ and}$$

$$V_{c_i} - V_T \geq 0 \text{ and } V_{c_i} - V_T \geq V_m \quad \text{for } \textit{all } i = 1,2. \tag{3.1.13}$$

Since the output of this multiplier ranges from negative to positive voltages, the above constraints can be simplified by

$$X_i^k - W_j^k \geq V_T > 0 \quad \text{for } \textit{all } i, j = 1,2 \text{ and } k = 1,2,...,n \text{ and}$$

$$V_{c_i} - V_T \geq V_m \quad \text{for } \textit{all } i = 1,2. \tag{3.1.14}$$

This analog multiplier has complete cancellation of nonlinearities except for a mobility term. A mobility is represented using SPICE parameters as follows:

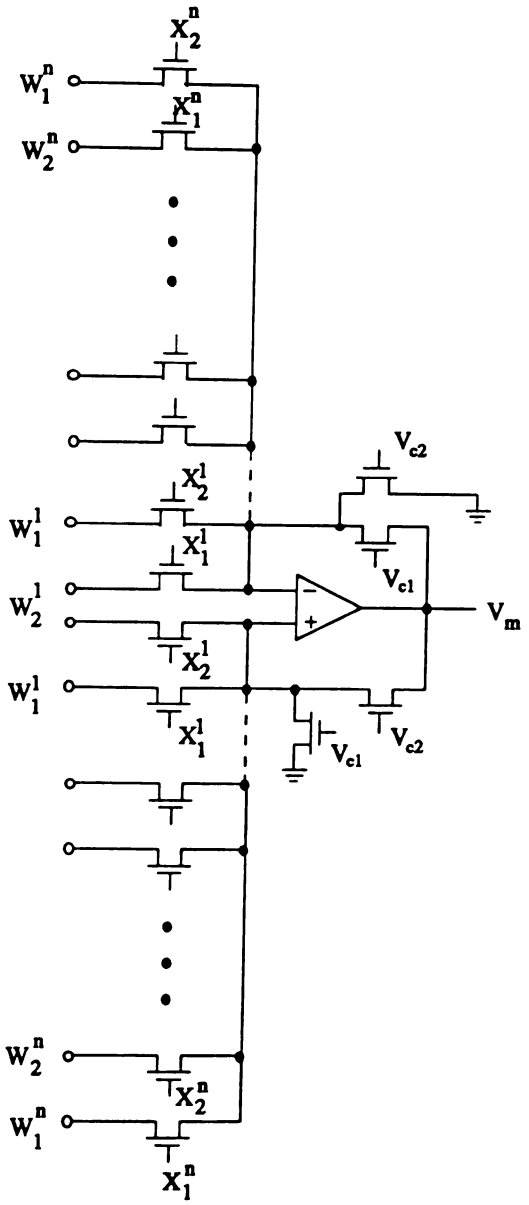


Figure 3.1.3 All-MOS n-D analog multiplier

$$\mu_s = \mu_o [\ \frac{(UCRIT)\varepsilon_{si}}{C_{ox}[V_{GS} - V_T - (UTRA)V_{DS}]}\ ]^{UEXP},$$

$$\text{if } \frac{(UCRIT)\varepsilon_{si}}{C_{ox}} < V_{GS} - V_T - (UTRA)V_{DS}$$

$$\mu_s = \mu_0, \qquad \text{otherwise,}$$

where *UCRIT*, *UTRA*, and *UEXP* are process parameters :

UCRIT = mobility degradation critical field

UTRA = mobility degradation transverse field coefficient

UEXP = mobility degradation exponent.

That is, this term is still dependent upon the terminal voltages, V_G, V_D, and V_S, and this attributes to the imperfection of the multipliers in real-world implementation. A complete implementation of the analog scalar/vector multiplier is discussed in Chapter 4.1.

3.2. A CMOS Operational Amplifier

A CMOS operational amplifier is considered in order to implement the above simple tunable all-MOS analog multiplier using standard CMOS technology. We have designed a CMOS operational amplifier to approximate an ideal operational amplifier with a sufficient open-loop gain, very small input offset voltage, and a quite large output current. However, a real operational amplifier is implemented with finite gain, nonzero output resistance and nonzero input offset voltage. These factors attribute to some errors or imperfection of the multipliers.

Presently, many CMOS operational amplifiers are available in the literature. Figure 3.2.1 depicts the schematic circuit of a simple p-channel input, unbuffered two stage CMOS operational amplifier [33].

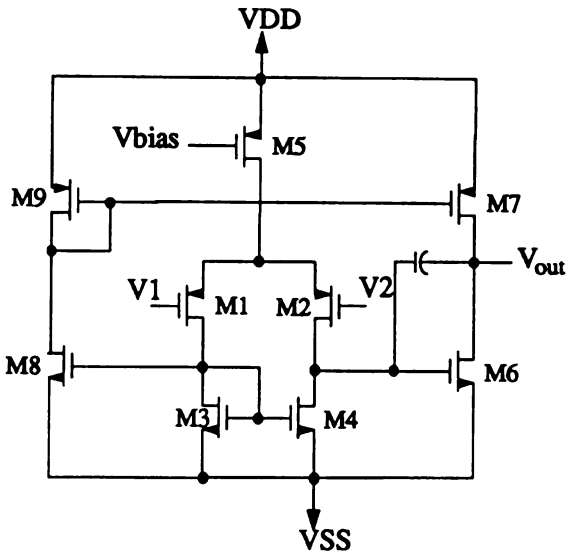


Figure 3.2.1 A CMOS operational amplifier

In the design of this operational amplifier, it is crucial to select a proper bias voltage, V_{bias}, and the aspect ratios of the MOS transistors. MOS transistor M1 and M2 are assumed identical and M3 and M4 are also assumed identical. Then the overall linear gain of the operational amplifier is given by [33]

$$A_v = A_{v_1} A_{v_2}$$

$$= \frac{g_{m_1}}{g_{o_2} + g_{o_4}} \frac{g_{m_6}}{g_{o_6} + g_{o_7}},$$

where $g_m = \left[I_{ss} \mu C_{ox} \frac{W}{L} \right]^{\frac{1}{2}}$ and $g_o = \lambda I_d$.

This operational amplifier is usually designed with the aspect ratio relationship [33]:

$$\frac{(W/L)_3}{(W/L)_6} = \frac{(W/L)_4}{(W/L)_6} = \frac{1}{2} \frac{(W/L)_5}{(W/L)_7}.$$

For low noise, M3 and M4 are designed with low transconductance, i.e., with small

aspect ratios. For a better frequency response, M6 is designed with high transconductance, i.e., with large aspect ratio.

The bias voltage of the CMOS operational amplifier is chosen to 3.2 v. The device sizes and the characteristics of the CMOS operational amplifier, based on our PSPICE simulations, are summarized in Table 3.2.1. Figure 3.2.2 shows the PSPICE simulation result of the open loop input/output characteristics of the CMOS operational amplifier and its frequency response. The PSPICE input file for the frequency response of the CMOS op-amp is shown in Appendix B.1.

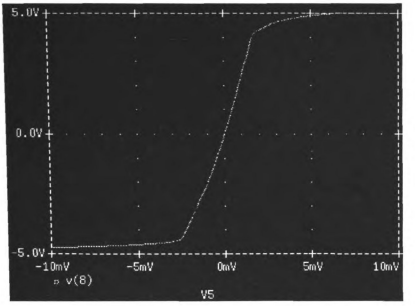
Table 3.2.1 (a) The device sizes (unit: μm) and (b) the characteristics of the CMOS op-amp

(a)

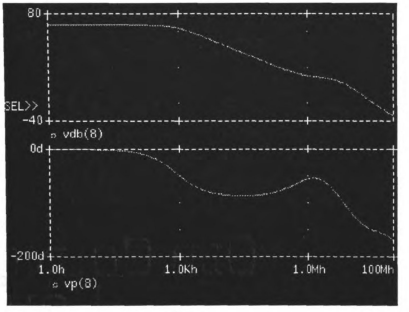
M1 and M2	$\frac{16}{4}$	M3 and M4	$\frac{4}{16}$	M5	$\frac{4}{10}$	-	-
M6	$\frac{240}{4}$	M7	$\frac{200}{4}$	M8 and M9	$\frac{4}{4}$	M10	$\frac{40}{40}$

(b)

voltage supplies	V_{dd} = 5.0 v and V_{ss} = –5.0 v
input voltage offset	–0.1 mv
input common-mode range	–4.5 v to 4.5 v
open-loop voltage gain	70dB
unit-gain bandwidth (GB)	8 MHz
phase margin	70 degree
slew rate	2.5 $v(\mu sec)^{-1}$
Output current at $V^+ - V^- = 0$	390 μA



(a)



(b)

Figure 3.2.1(a) The open loop transfer function and
(b) the frequency response of the CMOS op-amp

3.2.1. A Voltage Follower

The vector multiplier circuit is suitable to implement an artificial synapse in artificial neural nets since it is simple and executes linear multiplication. However, the multiplier circuit has some practical difficulties when it is implemented. The multiplier circuit has a driving input to the drain of an MOS transistor, whose input impedance is low. This will create problems of loading.

In order to overcome this problem, a voltage follower is desired. The output impedance of the voltage follower is so small that it can supply sufficient current to the low input impedance node of the drain of the MOS transistor of the vector multiplier circuit.

A voltage follower can be implemented with any standard (MOS) operational amplifiers. The voltage follower has been designed as the same structure as Figure 3.2.1. It has been implemented with a lower output impedance than the impedance of the driven MOS transistor in order to supply sufficient current to low input impedance node of the drain of the MOS transistor of the multiplier. The PSPICE simulation has been executed and has shown proper operation of the voltage follower supporting our contention. The device sizes of the voltage follower according to our design are summarized in Table 3.2.2 with setting the bias voltage of the voltage follower to 3.2v.

Table 3.2.2 The device sizes (unit: μm) of a voltage follower

M1 and M2	$\frac{8}{4}$	M3 and M4	$\frac{4}{16}$	M5	$\frac{4}{16}$	-	-
M6	$\frac{60}{4}$	M7	$\frac{47}{4}$	M8 and M9	$\frac{4}{4}$	M10	$\frac{40}{5}$

3.2.2. A Voltage Integrator

A voltage integrator will be used to perform the integration of the learning algorithm. This is identical to a voltage follower except that it has a capacitor load [30], which is shown in Figure 3.2.3. The capacitor and the output impedance of the operational amplifier will determine the time constant for the learning dynamics. The device sizes of a voltage integrator is the same as those of a voltage follower which are shown in Table 3.2.2. The PSPICE simulation has been executed to verify its proper function.

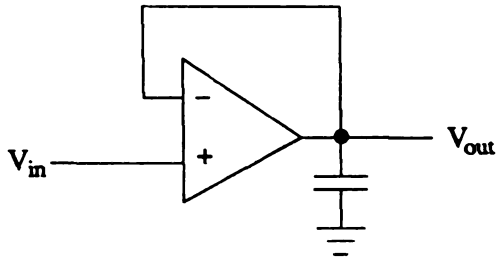


Figure 3.2.3 The voltage integrator circuit

3.3. Voltage Shifters/Attenuators

In order to use the vector multiplier in the implementation of a synapse cell, the operating range of its input must be considered carefully. The input of the multiplier in ANNs is applied from either an external voltage source or the output of a neuron. However, the output and input of the (vector) multiplier are not properly matched. In addition to input-output matching or compatibility, one must always satisfy the proper operation constraints of the multipliers.

The multipier has been designed to operate in a small operating range in order to reduce the nonlinearity-error effects which are caused by the variation in the

mobility parameters. Therefore, the operating range of the multiplier is required to be amplified to the level of range of a neuron in order to use the output of a neuron as an input signal. The difference in unmatched voltages and the increase in the operating range can be adjusted by employing a voltage shifter/attenuator.

There are several kinds of level shifters in the literature [17, 25, 26]. A level shifter can be implemented by using an operational amplifier and resistors. However, it takes a large area and is difficult to implement using MOS technology because of the implementation of the linear resistors. Simple and small-area level shifters are suitable for the implementation of neural nets [17, 26].

Two level shifter/attenuator circuits are considered. One shifter/attenuator circuit is for the input of the vector multiplier w_i. We label this shifter the w_shifter. The other shifter/attenuator circuit is for x_i. We label this shifter the x_shifter. Figure 3.3.1 shows the level shifter/attenuator for the w_i input. It is assumed that the transistors, M1, M5, and M7 are operating in the saturation region and the transistor M3 is operating in the ohmic region [25]. Then, its drain current is represented as follows:

$$i_{d_1} = \frac{K_1}{2} (V_{in} - V_3 - V_{T_1})^2 (1 + \lambda(V_{dd} - V_3))$$

$$i_{d_3} = K_3 (V_{in} - V_{ss} - V_{T_0} - \frac{V_3 - V_{ss}}{2}) (V_3 - V_{ss}) (1 + \lambda(V_3 - V_{ss}))$$

Since $i_{d_1} = i_{d_3}$, and assuming $\lambda = 0$, one obtains

$$\frac{K_1}{2} (V_{in}-V_3-V_{T_1})^2 = K_3 (V_{in} - V_{ss} - V_{T_0} - \frac{V_3 - V_{ss}}{2}) (V_3 - V_{ss})$$

$$= \frac{K_3}{2} [(V_{in}-V_{T_0}-V_{ss})^2 - (V_{in}-V_{T_0}-V_3)^2].$$

Simplifying the above equations, one gets

$$(K_1 + K_3) (V_{in}-V_3-V_{T_1})^2 = K_3 (V_{in}-V_{T_0}-V_{ss})^2$$

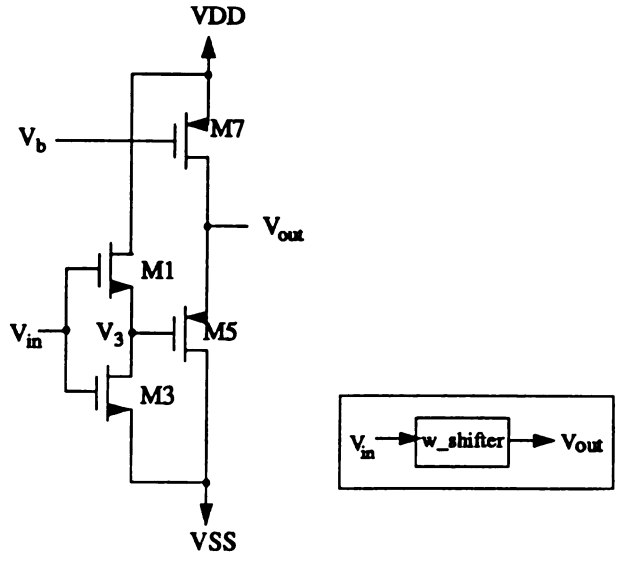


Figure 3.3.1 The w_shifter

$$V_{in} - V_3 - V_{T_1} = \sqrt{\frac{K_3}{K_1 + K_3}}\,(V_{in} - V_{T_0} - V_{ss}).$$

Rearranging the last equation containing V_3, and assuming $V_{T_1} = V_{T_0}$, one obtains

$$V_3 = (\,1 - \sqrt{\frac{K_3}{K_1 + K_3}}\,)\,(\,V_{in} - V_{T_0}\,) + \sqrt{\frac{K_3}{K_1 + K_3}}\,V_{ss}. \qquad (3.3.1)$$

Since the pMOS transistors, M5 and M7, are in the saturation region, then $i_{d_5} = i_{d_7}$.

Assuming $\lambda = 0$, one gets

$$\frac{K_5}{2}(V_{out} - V_3 + V_{T_5})^2 = \frac{K_7}{2}(V_{dd} - V_b + V_{T_7})^2$$

$$V_{out} = V_3 - V_{T_5} + \sqrt{\frac{K_7}{K_5}}\,(\,V_{dd} - V_b + V_{T_7}\,)$$

$$= V_3 + \sqrt{\frac{K_7}{K_5}}\,(\,V_{dd} - V_b\,) + [\,\sqrt{\frac{K_7}{K_5}}V_{T_7} - V_{T_5}\,]. \qquad (3.3.2)$$

Substituting equation (3.3.1) for equation (3.3.2),

$$V_{out} = (\,1 - \sqrt{\frac{K_3}{K_1 + K_3}}\,)\,V_{in} + \sqrt{\frac{K_7}{K_5}}\,(\,V_{dd} - V_b\,) + \sqrt{\frac{K_3}{K_1 + K_3}}\,V_{ss}$$

$$+ \left(\sqrt{\frac{K_3}{K_1 + K_3}} - 1 \right) V_{T_0} - V_{T_5} + \sqrt{\frac{K_7}{K_5}} V_{T_7}. \tag{3.3.3}$$

That is, the attenuation factor, a_w, is given by

$$a_w = 1 - \sqrt{\frac{K_3}{K_1 + K_3}}. \tag{3.3.4}$$

The level shifting, V_{sh}, is given by

$$V_{sh} = \sqrt{\frac{K_7}{K_5}} \left(V_{dd} - V_b \right) + \sqrt{\frac{K_3}{K_1 + K_3}} V_{ss}$$
$$+ \left(\sqrt{\frac{K_3}{K_1 + K_3}} - 1 \right) V_{T_0} - V_{T_5} + \sqrt{\frac{K_7}{K_5}} V_{T_7}. \tag{3.3.5}$$

Figure 3.3.2 shows the level shifter/attenuator for the x inputs, namely the x_shifter. This circuit is the as same as the first stage of Figure 3.3.1 except that pMOS transistors are used, in stead of nMOS transistors, for two input transistors. Similarly, the output of this level shifter/attenuator is given as

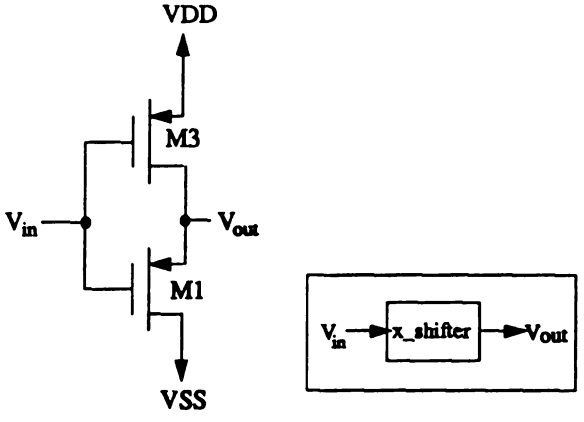


Figure 3.3.2 The x_shifter

$$V_{out} = \left(1 - \sqrt{\frac{K_2}{K_1 + K_2}} \right) \left(V_{in} - V_{T0} \right) + \sqrt{\frac{K_2}{K_1 + K_2}} V_{dd}. \tag{3.3.6}$$

The attenuator factor for the x_shifter, a_x, is given in the same form as a_w. However, these values will be different since the K_i values are dependent upon the transistor type and the aspect ratio of the MOS transistors used.

These level shifters/attenuators have been simulated using the PSPICE circuit simulator. Their device sizes and PSPICE simulation results are summarized in Table 3.3.1 with setting the bias voltage of the w_shifter to 3.0v.

Table 3.3.1 (a) The operating ranges and (b) the device sizes (unit: μm) of the w_shifter and the x_shifter

(a)

shifter	range of inputs	shifting	attenuation factor	max. error(%)
w_shifter	-2.5 to 2.5	-1.566	0.0813	0.0019 (0.47)
x_shifter	-2.5 to 2.5	4.686	0.0766	0.0013 (0.34)

(b)

	M1 & M2	M3 & M4	M5 & M6	M7 & M8
w_shifter	$\frac{11}{22}$	$\frac{10}{4}$	$\frac{4}{4}$	$\frac{4}{4}$
x_shifter	$\frac{8}{16}$	$\frac{20}{4}$	-	-

3.4. A MOS Capacitor

Theoretically, the capacitance is represented as

$$C = \varepsilon \frac{A}{d}, \tag{3.4.1}$$

where A is the area of the electric plate and d is the distance between two electric plates. A capacitor can be designed by several different ways using standard MOS technology [35]. One method uses double metals or double poly. However, this method requires large chip area due to the thick oxide layer between double metals or double poly. Another method uses the gate oxide capacitance when a MOS transistor is operated in the ohmic region. This method requires less chip area than the first method because of the thin oxide layer of the gate of a MOS transistor.

We have employed the second method in order to implement an MOS capacitor which is shown in Figure 3.4.1. The MOS transistor is biased in the ohmic region by connecting its drain, source, and substrate nodes to the same voltage level, say V_{ss} and by satisfying $V_{gs} - V_T > 0$. Then the gate forms one plate and the source, drain, and channel form another plate. This capacitor can be used very effectively in non-critical applications. However, the sheet resistance of the bottom plate formed by the channel is high since the underlying substrate is lightly doped.

Using equation 3.4.1, the gate oxide capacitance, C_{gox}, is represented as

$$C_{gox} = A\ C_{ox} = A\ \frac{\varepsilon_{ox}}{T_{ox}}. \tag{3.4.2}$$

Therefore, if we want to determine the actual transistor size for a capacitance, say C_1, then the actual chip area, A is found by using

$$A = \frac{C_1}{C_{ox}} = \frac{C_1\, T_{ox}}{\varepsilon_{ox}}, \tag{3.4.3}$$

where $A = WL$.

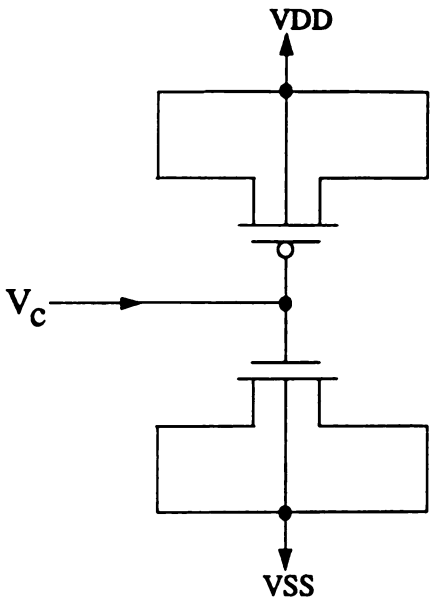


Figure 3.4.1 A MOS capacitor

Given the SPICE parameters in Appendix A.2, Table 3.4.1 below summarizes the device size corresponding to different capacitance value.

Table 3.4.1 The device size of MOS capacitors

capacitance	1 pF	2 pF	3 pF	4 pF	5 pF
area (μm^2)	1200	2400	3600	4800	6000

3.5. A CMOS Double Inverter

A CMOS push-pull double inverter is shown in Figure 3.5.1(a). This can be used to implement a cell body by cascading two inverters and forming a double inverter since its i/o characteristics shows a sigmoid function (Figure 3.5.1(b)). Sometimes a simple operational amplifier is used for the implementation of a cell body.

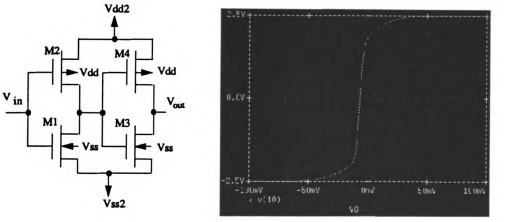


Figure 3.5.1 (a) A CMOS double inverter and
(b) Transfer characteristics of the double inverter

The maximum and the minimum output of this double inverter are given as +2.5 v and -2.5 v, respectively, in order to be matched with the maximum and the minimum input of the analog multiplier. For latch-up protection of MOS transistors, p-well contact and n-well contact are connected by +5 v and -5 v, respectively.

3.6. An Analog Multiplexer

An analog multiplexer is one of the important elements for the implementation of ANNs. This is especially true in the design of ANNs with a large number of neurons since standard chips have a limited number of pins. In Chapter 6, an analog multiplexer is used to be addressed to read or write synaptic weights stored on on-chip analog memories via off-chip interface circuitry. This analog multiplexer can be easily designed with CMOS analog switches and a digital decoder. When an address is applied to the decoder, its corresponding analog switch is turned on and an analog signal is passed through the analog switch from the Digital/Analog converter (DAC) or Analog/Digital converter(ADC) on the interface circuitry. Figure 3.6.1 shows the

block diagram of a 1x8 analog multiplexer, where switches are implemented by a CMOS analog switch.

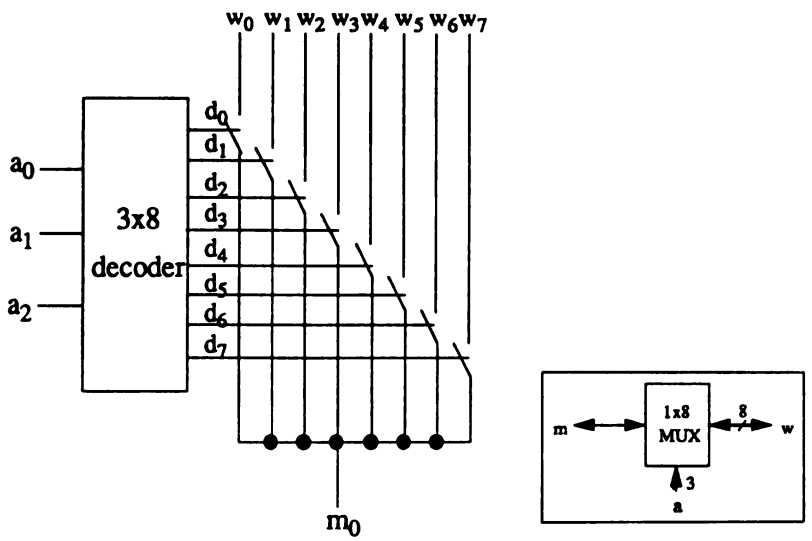


Figure 3.6.1 A 1x8 analog multiplexer

CHAPTER 4

ANALOG SCALAR/VECTOR MULTIPLIERS

The following standard cells are described in this chapter: analog multiplier cells, analog adder cells, programmable synapse cells, and a sigmoid-derivative cell.

4.1. Analog Multiplier Cells

An all-MOS four-quadrant analog multiplier is implemented using the basic analog MOS circuits which are described in Chapter 3. The output of this n-dimensional analog vector multiplier is given by [27, 28]

$$V_m = \frac{(W/L)_i}{(W/L)_r (V_{c_1} - V_{c_2})} \sum_{i=1}^{n} (W_1^i - W_2^i)(X_1^i - X_2^i) \tag{4.1.1}$$

with the following operating constraints

$$X_i^k - W_j^k \geq V_T > 0 \quad \text{for } all\ i,j = 1,2 \text{ and } k = 1,2,...,n \text{ and}$$

$$V_{c_i} - V_T \geq V_m \quad \text{for } all\ i = 1,2. \tag{4.1.2}$$

Due to the operating constraints of the multiplier, equation (4.1.2), the output range of the multiplier does not match its operating input range. The multipier has been designed to operate in the small operating range in order to reduce the nonlinearity-errors caused by the variation of the mobility. The operating range of the multiplier is required to be compatible with the operating range of a neuron in order to

use the multiplier as an artificial synapse.

There are two problems to be considered for this multiplier in order to implement it as a synapse in artificial neural nets : One is the multiplier's driving input to the drain of a floating MOS transistor, whose input impedance is low; the other is the operating range of the multiplier and its input-output comparability.

The first problem is solved by employing a voltage follower, which is described in section 3.2.1. In order to make the operating input range match the output range of the multiplier and/or increase the operating input range of the multiplier, voltage shifters/attenuators are employed, which are described in section 3.3.

The 1-D multiplier cell is designed with additional circuits such as voltage followers and voltage shifters/attenuators, which is depicted in Figure 4.1.1. The inputs are applied through the voltage shifters/attenuators. A voltage follower is placed in prior to the drain input of the multiplier in order to drive sufficient current to the low-input-impedance drain inputs of the multiplier. The operating range of the multiplier is given by $-2.5 \leq w_i, x_i, V_m \leq 2.5$.

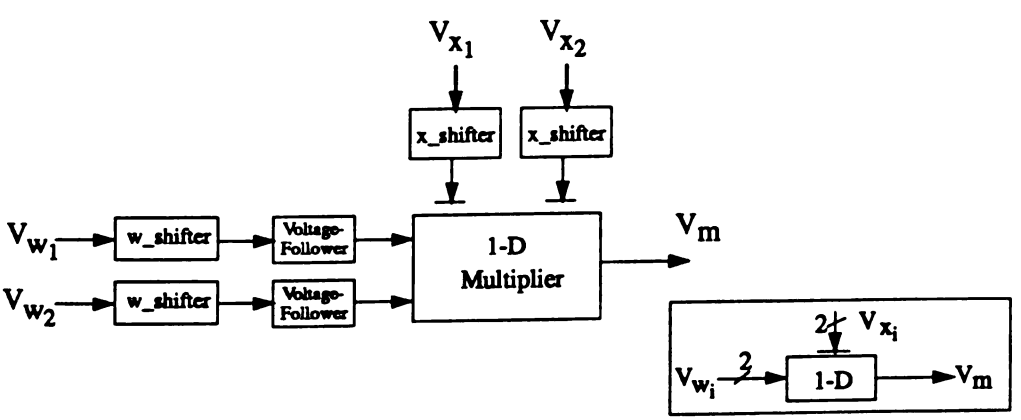


Figure 4.1.1 A 1-D multiplier cell

This multiplier is defined and designed as a standard analog multiplier cell which may be up to 20-dimensional vector multiplier, called n-D multiplier. The PSPICE

simulations have been executed up to 20-D vector multiplier with the SPICE parameter in Appendix A.2. The device sizes and the tuning voltages are summarized in Table 4.1.1.

Table 4.1.1 The device sizes (unit: μm) and the applied tuning voltages (unit: volts) of each standard analog multiplier cell

n-D	$\left(\frac{W}{L}\right)_i$	$\left(\frac{W}{L}\right)_r$	$V_{C_1} - V_{C_2}$
1-D to 10-D	$\frac{4}{24}$	$\frac{4}{48}$	0.041 n
11-D to 20-D	$\frac{4}{48}$	$\frac{4}{48}$	0.02 n

All the standard multiplier cells are designed with the same size of the feedforward nMOS transistors and the feedback nMOS transistors for 1-D to 10-D and for 11-D to 20-D, respectively. They are controlled by setting $V_{C_1} - V_{C_2}$ to $0.041n$ volts for the former and to $0.02n$ volts for the latter. However, these tuning voltage values are determined by the SPICE parameters used.

When a very high dimensional vector multiplier is required, this design requires large chip area. That is, from equation (3.1.12), as n is increased, L_i is also increased with fixed W_i, W_r, and L_r. This results in a large chip area.

Hierarchical design may be suitable to design a very high dimensional vector multiplier. The standard multiplier cells, n-Ds and the standard adder cells, n-DAs, are used for hierarchical design. The standard adder cells are implemented using the standard multiplier cells, which will be described in section 4.4.3.

Figure 4.1.2 shows one example with the implementation of a 100-dimensional vector multiplier. This multiplier is implemented using ten 10–Ds and one 10–DA. Each 10–D cell multiplies two 10-dimensional vectors, i.e., 10-dimensional vector

inner product. A 10–*DA* cell combines ten vector inner products which are obtained from ten 10–*DM* cells and results in 100-dimensional vector inner product.

Using this hierarchical design with standard cells, we can reduce the chip area and its design time for a very high dimensional vector multiplier. However, this design has two disadvantages. One is a time delay since the computation is done through more than one layer. The time delay is proportional to the number of layers. The other is that the error becomes larger than that of the non-hierarchical design since each layer generates errors.

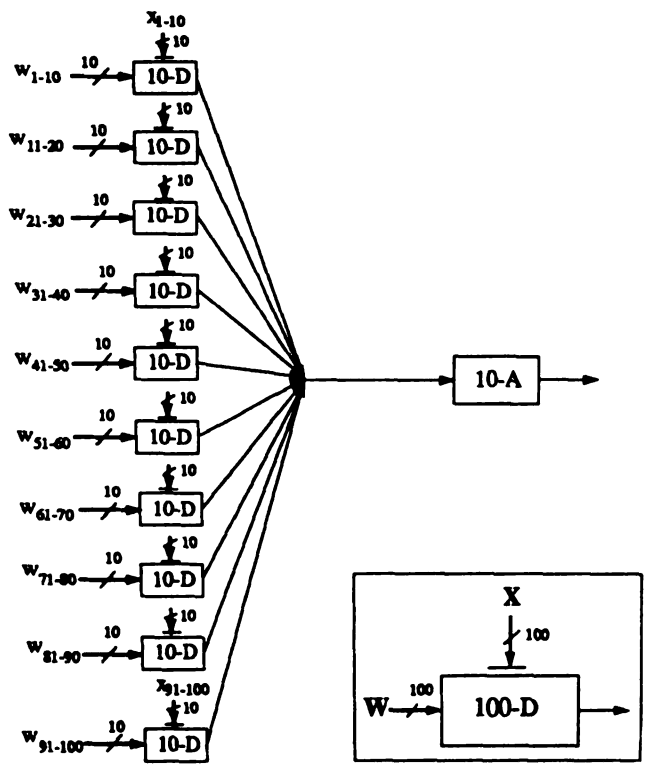


Figure 4.1.2 A 100-D multiplier cell

4.2. Performance of Analog Multipliers

The performance of the multiplier [47] is usually checked by finding the following measurements : output offset, X or W nonlinearity, X or W feedthrough, small signal bandwidth, slew rate, and settling time.

First, the output function is determined by the anticipated functional relationship among two inputs, ΔV_x, ΔV_w, the output, $V_{anticipated}$, and the scale constant, V_r. The multiplier in Section 4.1 produces the following output function.

$$V_{anticipated} = \frac{\Delta V_x \, \Delta V_w}{V_r}$$

$$= \frac{\Delta V_x \, \Delta V_w}{5.0}, \tag{4.2.1}$$

where

$$\Delta V_x = V_{x_1} - V_{x_2}$$

$$\Delta V_w = V_{w_1} - V_{w_2}$$

with the operating range, $-2.5 \le V_{x_{1,2}}, V_{w_1} \le 2.5$ volts and $V_{w_2} = 0$.

All errors are defined as the deviations from this output function. *Percentage error* is calculated as follows:

$$\%\ Error = 100\ \frac{\varepsilon}{FS}, \tag{4.2.2}$$

where $\varepsilon = V_{measured} - V_{anticipated}$ and FS means full-scale range.

The *output offset* is measured at the output voltage by applying $V_x = V_w = 0$.

X or W nonlinearity is measured with the maximum percentage error of full-scale between the actual multiplier output and the anticipated output with the full-scale range of the corresponding input and the full-scale dc value on the other input. Specifically, *X nonlinearity* is measured with a 10.0 V_{p-p} 50-Hz triangular waveform on ΔV_x and

2.5 dc voltage on ΔV_w. *W nonlinearity* is measure with a 5.0 V_{p-p} 50-Hz triangular waveform on ΔV_w and a 5.0 dc voltage on ΔV_x.

X or W feedthrough is usually specified by applying a 50 Hz full-scale sine wave on one input and zero on the other.

Small-signal bandwidth is achieved by measuring the frequency at which the output has decreased to -3 dB of the full-scale output with a full-scale dc voltage (5 volts) on one input and 5% of full-scale (0.25 volts) sine wave on the other. *Slew rate* is the maximum rate of change of output voltage and *settling time* is the time that the output takes to approach within a specified percentage of its final value in response to a full-scale step input and a full-scale dc voltage.

The above specifications are measured using the PSPICE circuit simulator for 1-dimensional analog multiplier which is shown in Figure 4.1.1. Table 4.2.1 compares the specifications of 1-dimensional analog multiplier and a commercial multiplier AD532J [51], which is designed with a Gilbert cell using bipolar technology. Scalar to 11-D vector multiplier is simulated to measure their output offset and maximum error. Its results is summarized in Table 4.2.2. Figure 4.2.1 depicts the graphs which are achieved from the PSPICE simulation: the multiplication in Figure 4.2.1(a), the x-nonlinearity error in Figure 4.2.1(b), the w-nonlinearity error in Figure 4.2.1(c), and the slew rate in Figure 4.2.1(d). The PSPICE inputs for the above measurements are shown in Appendix B.2-B.6.

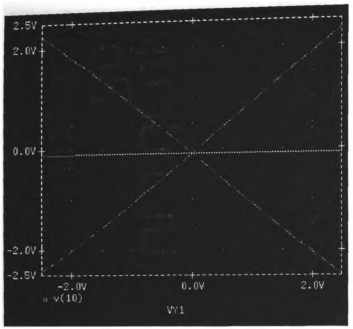


Figure 4.2.1 (a) Multiplication graphs of 1-D multiplier

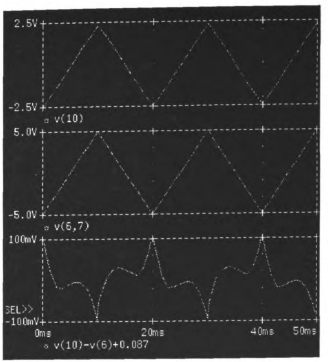


Figure 4.2.1 (b) The x-nonlinearity of 1-D multiplier for $\Delta V_x = 10 v_{p\text{-}p}$ and $\Delta V_w = 2.5v$.
Top trace: Output signal
Middle trace: x-input signal
Bottom trace: The x-input nonlinearity voltage

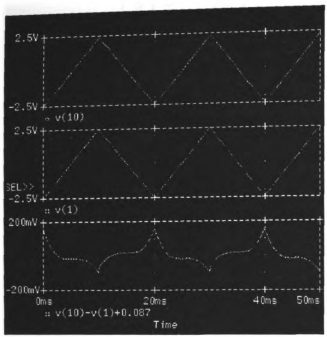


Figure 4.2.1 (c) The W-nonlinearity of 1-D multiplier for ΔV_w=5.0v_{p-p} and ΔV_x=5v.
Top trace: Output signal
Middle trace: w-signal
Bottom trace: The w-input nonlinearity voltage

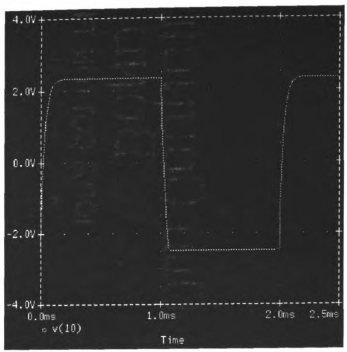


Figure 4.2.1 (d) Step response of 1-D multiplier ΔV_x = 10 v_{p-p} 500Hz square wave and ΔV_w=2.5v.

Table 4.2.1 The specifications of 1-D multiplier and AD532J multiplier [51]

	1-D multiplier	AD532J
Power Supply	± 5v	± 15v
Input-Output Range	$5.0v_{p-p}$	$10v_{p-p}$
Output offset voltage	-0.087v	±0.04v
X-input nonlinearity	2.0%FS	± 0.8%
W-input nonlinearity	3.0%FS	± 0.3%
X-feedthrough	0.02 mv_{p-p}	50 mv
W-feedthrough	9 mv_{p-p}	30 mv
-3dB Bandwidth for the entire circuit multiplier cell only	 5kHz 2MHz	1MHz
Slew Rate	0.1 $v(\mu sec)^{-1}$	45 $v(\mu sec)^{-1}$
Settling time	117 μsec to 2%	1 μsec to 2%

Table 4.2.2 Output offset voltage and maximum percentage error of analog multiplier cells

	1-D	2-D	3-D	4-D	5-D	6-D
Output Offset (v)	-0.087	-0.094	-0.096	-0.097	-0.0976	-0.0977
% Error (%)	2.26	3.82	4.14	4.34	4.49	4.596
	7-D	8-D	9-D	10-D	11-D	-
Output Offset (v)	-0.0976	-0.0975	-0.0973	-0.097	-0.0967	-
% Error (%)	4.69	4.75	4.83	4.88	4.95	-

4.3. Implementation of an 11-D Analog Multiplier

An 11-D analog multiplier is designed using the MAGIC VLSI editor and sent for fabrication on a 40-pin MOSIS TINYCHIP with analog pads using 2 μm CMOS n-well process. The 11-dimensional vector multiplier is the largest vector multiplier we can obtain because of the pin limitation of the 40-pin on the MOSIS TINYCHIP. Figure 4.3.1 shows its layout in the 40-pin MOSIS TINYCHIP pad frame. The 40-pin MOSIS TINYCHIP pad frame contains 34 analog i/o pins, 3 pins for Vss, and 3 pins for Vdd.

33 pins out of 34 pins are used and Table 4.3.1(a) summarizes the number of pins used for inputs and outputs, where their notations are the as same as we used in Figure 4.1.1. Table 4.3.1(b) tabulates the pin-assignment of this chip. This chip contains three different cells to test an up to 11-dimensional vector multiplier, with on-chip CMOS operational amplifier or with off-chip operational amplifier, and a CMOS operational amplifier. For the purpose of testing each cell, some of the control voltages are set as in Table 4.3.2.

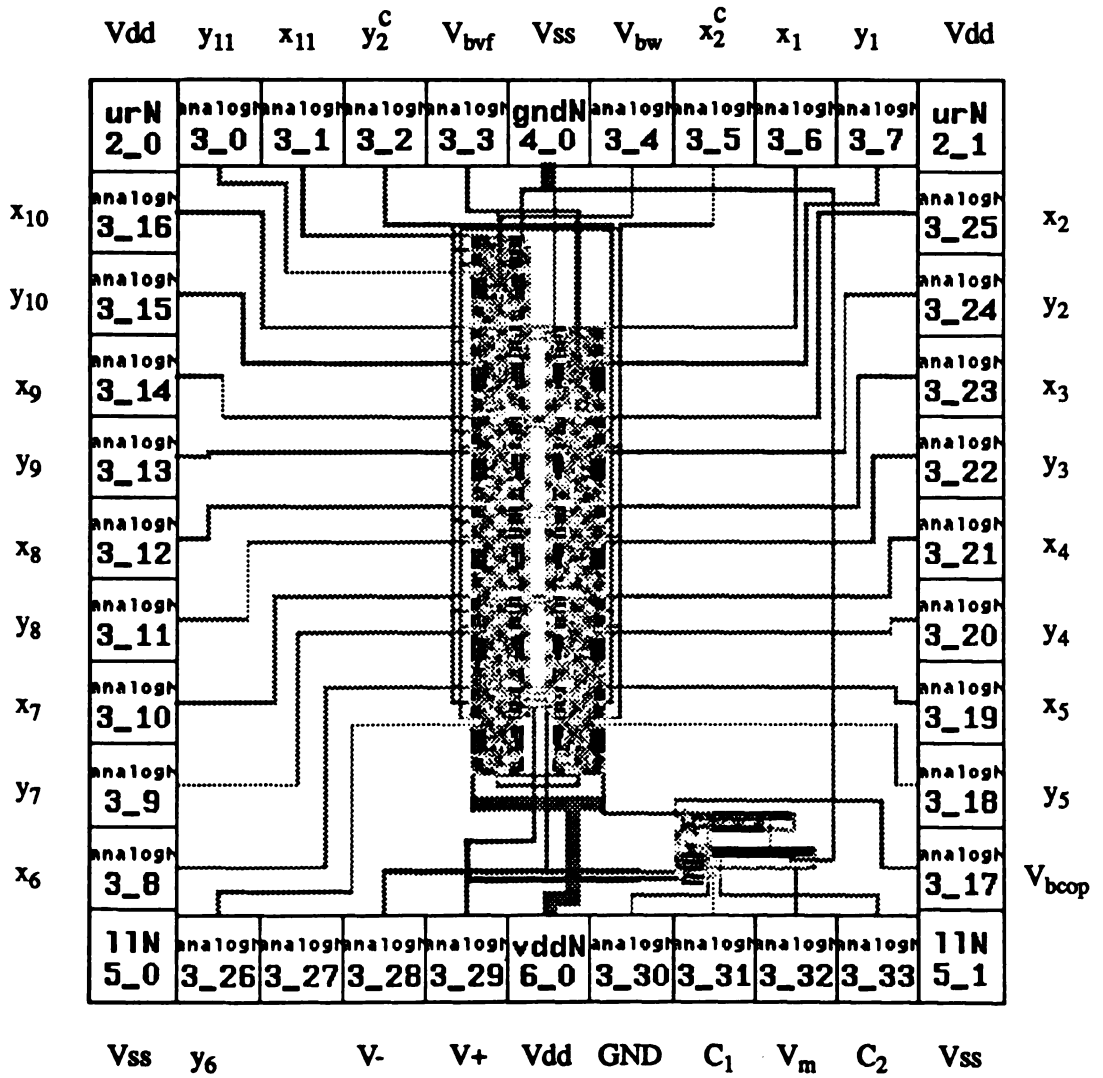


Figure 4.3.1 The 11-D vector multiplier chip on a 40-pin Tinychip

Table 4.3.1 (a) The number of pins used and (b) pin-assignment of the 11-D chip

(a)

Signal	x_1^i	w_1^i	x_2^i	w_2^i	V_{c_1}	V_{c_2}	V^+	V^-	V_k	V_m	V_{cop}	V_{vf}	V_w	total
# of pins	11	11	1	1	1	1	1	1	1	1	1	1	1	33

(b)

Pin #	1	2	3	4	5	6	7	8	9	10
Signal	w_3	x_3	w_2	x_2	Vdd	w_1	x_1	x_2^c	V_w	Vss
Pin #	11	12	13	14	15	16	17	18	19	20
Signal	V_{vf}	w_2^c	x_{11}	w_{11}	Vdd	x_{10}	w_{10}	x_9	w_9	x_8
Pin #	21	22	23	24	25	26	27	28	29	30
Signal	w_8	x_7	w_7	x_6	Vss	w_6	*	V^-	V^+	Vdd
Pin #	31	32	33	34	35	36	37	38	39	40
Signal	GND	V_{c_1}	V_m	V_{c_2}	Vss	V_{cop}	w_5	x_5	w_4	x_4

(* means "do not use.")

Table 4.3.2 The tuning signal for each cell in the 11-D chip (units are in volts)

Test Cell	V_{c_1}	V_{c_2}	V^+	V^-	V_{cop}	V_m
11-D with on-chip op-amp	**	**	*	*	**	output
11-D with off-chip op-amp	-5	-5	output	output	*	*
CMOS op-amp	-5	-5	input	input	**	output

(* means "do not use" and ** means "tune properly.")

4.4. Applications

Analog multipliers can be used to implement other circuits which are used for the implementation of feedforward ANNs with learning such as programmable synapse cells, a sigmoid-derivative cell, and analog voltage-adder cells.

4.4.1. Programmable Synapse Cells

Programmable analog synapse cells are implemented using standard analog MOS multiplier cells and MOS capacitors. *n*-S stands for the n-dimensional synapse cell and is realized with one *n*-D multiplier cell and *n* MOS capacitors. Figure 4.4.1 shows the circuit description of 1-S cell.

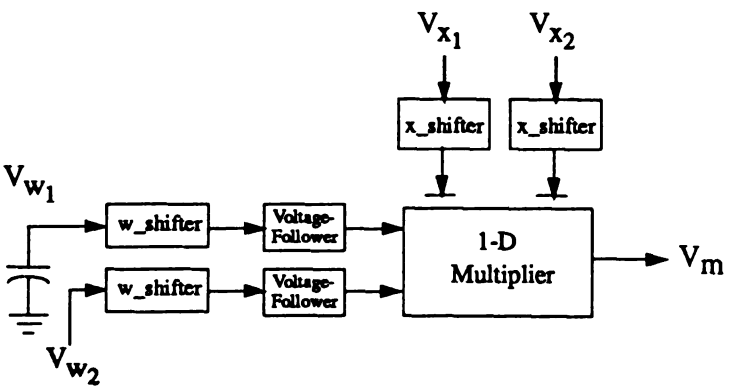


Figure 4.4.1 A 1-D programmable synapse cell

A MOS capacitor is employed as an analog storage device. In Table 3.4.1, in order to realize the 1 pF MOS capacitor, the transistor size is given by W/L= 40/20 for nMOS transistor and W/L= 40/20 for pMOS transistor.

A large scale ANN may require a huge number of synapses and neurons. In this case, a very high dimensional synapse cell should be available. A very high

dimensional synapse cell can be built with a very higher dimensional vector multiplier which is discussed in section 4.1.

4.4.2. A Sigmoid-Derivative Cell

A derivative of the sigmoid function of a neuron may be required to implement a learning rule for feedforward ANN. The sigmoid-derivative cell is the circuit which computes the differentiation of a sigmoid function by the input voltage of the sigmoid function, not by time.

Usually, a cell body is implemented using a double inverter or a simple operational amplifier. When a neuron is implemented on silicon, it is impossible to obtain the exact expression of the sigmoid function of a cell body. In order to obtain a derivative of the sigmoid function of a cell body, it is assumed that a sigmoid function of a cell body is represented by a hyperbolic tangent function such as

$$S(V_i) = m \tanh(kV_i). \tag{4.4.1}$$

Then its derivative can be represented

$$\frac{dS(V_i)}{dV_i} = k\ m\ sech^2(kV_i)$$

$$= k\ m\ (\ 1 - \tanh^2(kV_i)\ )$$

$$= \frac{k}{m}\ (\ m^2 - S^2(V_i)\ )$$

$$= \frac{k}{m}\ (\ m - S(V_i)\ )\ (\ m - (\ -S(V_i)\ ). \tag{4.4.2}$$

Equation (4.4.2) can be implemented using a 2-D analog multiplier by setting $x_1^1 = x_1^2 = m$, $x_2^1 = x_2^2 = 0$, $w_1^1 = w_2^2 = S(V_i)$, and $w_1^2 = w_2^1 = 0$, where $S(V_i)$ is

supplied from the output of a double inverter. Figure 4.4.2 shows the block diagram of a sigmoid-derivative cell. In order to achieve the derivative of the sigmoid function of a double inverter, this cell is simulated with the PSPICE input file in Appendix B.7 using the PSPICE circuit simulator. Figure 4.4.3 shows the sigmoid function of a double inverter and its derivative.

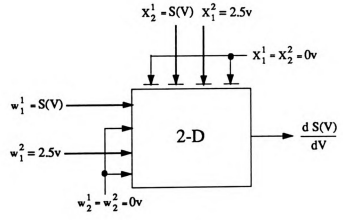


Figure 4.4.2 A sigmoid-derivative cell

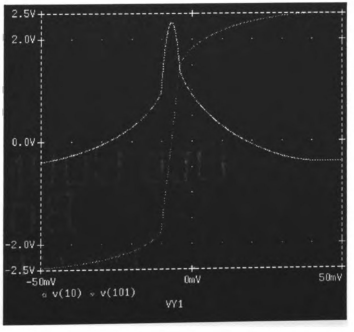


Figure 4.4.3 The input/output characteristics of the sigmoid-derivative cell

4.4.3. Analog Adder Cells

An analog voltage adder can be achieved using the analog multiplier cell by fixing $\Delta w_i = 2.5$. If w_i is supplied from the outside voltage source directly, two w_shifters and two voltage followers will not be required any more. Figure 4.4.4 shows the block diagram of an n-D analog adder, called n-DA. The operating range of an n-DA is given by

$$-2.5 \le x_j{}^i, V_m \le 2.5$$

with the fixed $w_j{}^1 = 2.5$v and $w_j{}^2 = 0$v.

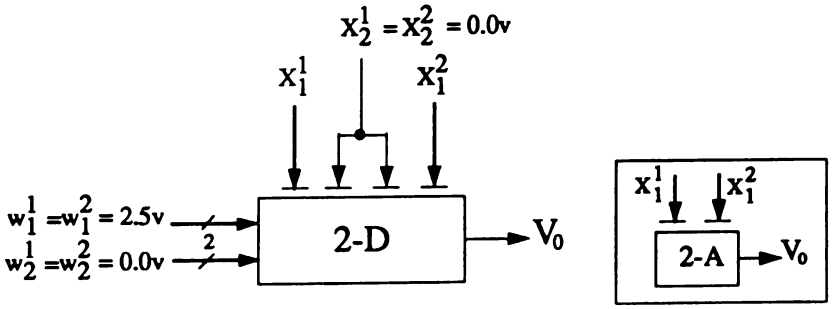


Figure 4.4.4 A voltage adder cell

CHAPTER 5

FEEDFORWARD ANNS WITH LEARNING

A modified error back-propagation learning rule [43] is realized with circuits which can be implemented using MOS transistors. The circuit is simulated using the PSPICE circuit simulator for a prototype two-layer feedforward ANNs with learning using standard CMOS VLSI/LSI technology.

Four feedforward ANNs with learning circuits are suggested by Dr. Salam. These circuits have been implemented and simulated using the PSPICE circuit simulator in order to prove this modified learning scheme is successfully implementable using standard CMOS technology. Each circuit is described with block diagram and its PSPICE simulation result is summarized. In order to save simulation time using PSPICE, control voltage sources are used to model an ideal operational amplifiers and level shifters in the circuits. A 100Ω resistor is used between an ideal operational amplifier and an integrating capacitor in order to ensure the integrating operation of the capacitor with the output of a multiplier.

5.1. A Modified Learning Rule for Feedforward ANNs

The squared error function and the governing static equation of each neuron are defined as follows [31]:

$$E_p = \frac{1}{2} \sum_{j=1}^{n} (t_{pj} - y_{pj})^2 \tag{5.1.1}$$

$$y_j = S_j(\sum_{i=1}^{N} w_{ji} x_i + \theta_j), \tag{5.1.2}$$

where t_{pj} is the value of the desired target p for the j-th output component, y_{pj} is the output of neuron j in the output layer for the pattern p, y_j is the output of neuron j, S_j is the sigmoid function of neuron j, N is the number of neurons in the previous layer, w_{ji} is the synapse weight from the output of neuron i in the previous layer to the input of neuron j, x_i is the output of neuron i in the previous layer, and θ_j is the threshold weight of neuron j.

A modified learning rule is obtained by removing the sigmoid derivative function term, dS_j/du_j, from the equation (2.2.14) in section 2.2.3, which is the error back-propagation rule [31]. Then the modified learning rule [29, 43] is given for any weight w_{ji} as follows:

$$\dot{w}_{ji} = -\eta \left(\frac{\partial S_j}{\partial u_j}\right)^{-1} \sum_p \frac{\partial E_p}{\partial w_{ji}} - \alpha_{ji} w_{ji}$$

$$= \eta \sum_p e_{pj}\, y_{pi} - \alpha_{ji} w_{ji}, \tag{5.1.3}$$

where, if the neuron j is in the output layer, then

$$e_{pj} = t_{pj} - \bar{y}_{pj} \tag{5.1.4}$$

and, if the neuron j is in any hidden layer, then

$$e_{pj} = \sum_k \frac{d\bar{S}_k}{d\bar{u}_k} e_{pk}\, \bar{w}_{kj}. \tag{5.1.5}$$

That is, when the neuron k is in the output layer, then

$$\dot{\bar{w}}_{kj} = \eta \sum_p (t_{pk} - \bar{y}_{pk})\, y_{pj} - \bar{\alpha}_{ji} \bar{w}_{ji} \tag{5.1.6}$$

and when the neuron j is in any hidden layer, then

$$\dot{\underline{w}}_{ji} = \eta \sum_k \frac{d\bar{S}_k}{d\bar{u}_k} \bar{w}_{kj} \sum_p (t_{pk} - \bar{y}_{pk})\, y_{pi} - \underline{\alpha}_{ji}\underline{w}_{ji}, \tag{5.1.7}$$

where $\bar{S}_k$ is the nondecreasing differentiable monotone sigmoid function of the neuron k for the output layer. $\bar{y}_{pk}$ is the output of the neuron k in the output layer, $\underline{y}_{pj}$ is the output of the neuron j in a hidden layer, and y_{pi} is the output of the neuron i in a previous layer. If the previous layer is the input layer, then y_{pi} is the sigmoidal output to the external input p-pattern x_{pi}.

It is also shown that the derivative terms $d\bar{S}_k/d\bar{u}_k$ in the equation (5.1.7) may be removed without loss of stability and convergence of the update law (5.1.6)-(5.1.7).

5.2. Feedforward ANNs with Sequential Learning

Two different feedforward ANNs with sequential learning are suggested by Dr. Salam and have been are implemented. Their PSPICE simulation results are discussed.

5.2.1. Feedforward ANNs with the Sequential Learning Circuit #1

The block diagram of the feedforward ANNs with the sequential learning circuit #1 is shown in Figure 5.2.1 [29]. This circuit is a 2x2x1 feedforward ANN employing a version of the modified learning rule. That is, it has two input nodes, two neurons in the hidden layer, and one neuron in the output layer. This circuit does not include circuits for the update of the threshold parameters. Its circuit implementation is described by the following equations:

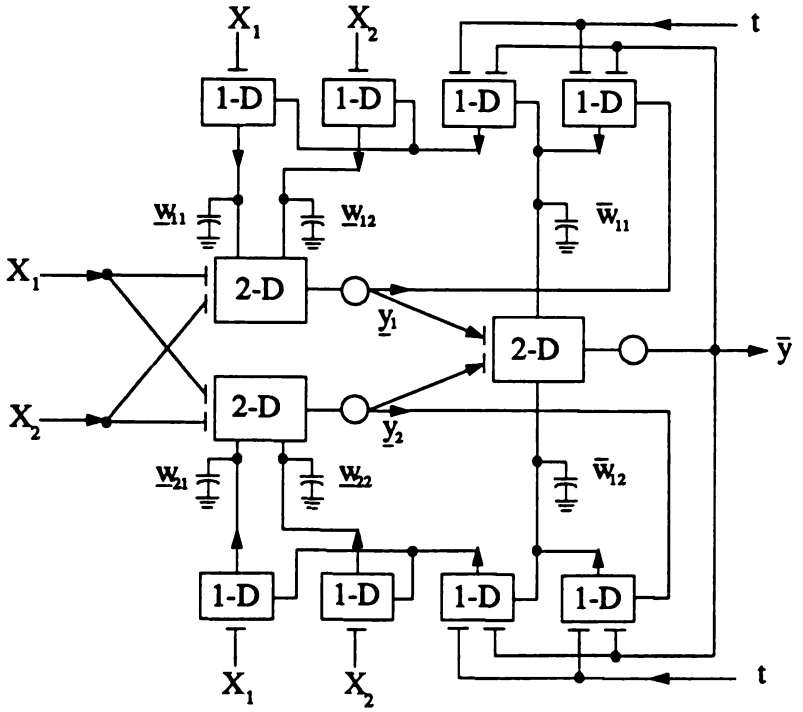


Figure 5.2.1 The feedforward ANN with the sequential learning circuit #1

$$RC\dot{\bar{w}}_{11} = k_1 [(t_{11} - \bar{y}_{11}) \underline{y}_{11}] - \bar{w}_{11} \tag{5.2.1i}$$

$$RC\dot{\bar{w}}_{12} = k_1 [(t_{11} - \bar{y}_{11}) \underline{y}_{12}] - \bar{w}_{12} \tag{5.2.1ii}$$

$$RC\dot{\underline{w}}_{11} = k_1 \bar{w}_{11} [(t_{11} - \bar{y}_{11}) x_{11}] - \underline{w}_{11} \tag{5.2.2i}$$

$$RC\dot{\underline{w}}_{12} = k_1 \bar{w}_{11} [(t_{11} - \bar{y}_{11}) x_{12}] - \underline{w}_{12} \tag{5.2.2ii}$$

$$RC\dot{\underline{w}}_{21} = k_1 \bar{w}_{12} [(t_{11} - \bar{y}_{11}) x_{11}] - \underline{w}_{21} \tag{5.2.2iii}$$

$$RC\dot{\underline{w}}_{22} = k_1 \bar{w}_{12} [(t_{11} - \bar{y}_{11}) x_{12}] - \underline{w}_{22}. \tag{5.2.2iv}$$

The circuit implementation of the governing equation results in

$$\underline{Y}_{p1} = S_n(k_2(\sum_j \underline{w}_{1j} x_{pj})) \tag{5.2.3i}$$

$$\underline{Y}_{p2} = S_n(k_2(\sum_j \underline{w}_{2j} x_{pj})) \tag{5.2.3ii}$$

$$\bar{Y}_p = S_n(k_2(\sum_j \bar{w}_{1j}\underline{y}_{pj})), \qquad (5.2.3iii)$$

where $S_n(.)$ is a sigmoid function of a neuron and k_n is a n-dimensional vector multiplier constant. In the above equations, $\underline{y}_{pj}$ and x_{pj} are defined as follows:

$$\underline{y}_{pj} \equiv \underline{Y}_{pj} - 2.5$$

$$x_{pj} \equiv X_{pj} - 2.5,$$

where $\underline{Y}_{pj}$ is an actual output of the hidden layer, $\bar{Y}_p$ is an actual output of the output layer, and X_{pj} is an actual input of the input layer for the pattern p. These same notations are used throughout this paper.

The full circuit is simulated using the PSPICE circuit simulator. Four learning tasks using four distinct input-output patterns have been executed separately as follows:

1. Initialize the dynamic feedforward ANN circuit with the same initial condition, ($\underline{w}_{11}$ $\underline{w}_{12}$ $\underline{w}_{21}$ $\underline{w}_{22}$ $\bar{w}_{11}$ $\bar{w}_{12}$) = (-0.5 0.6 -0.5 0.5 0.7 0.5) for all the learning tasks.
2. Each input data and its desired target (i.e., the input-output pattern) are applied and the PSPICE transient analysis is performed. Then measure the steady state weight values and the output error, $\tau - \bar{y}$, for each pattern pair.
3. The PSPICE dc analysis is performed by setting the weights to the steady state weight values. Then we measure the error $\tau - \bar{y}$ again.

The results are summarized in Table 5.2.1 and the PSPICE input files are shown in Appendix B.8-B.9. The results show that this ANN circuit succeeds to learn each applied input-output pattern. However, when the four distinct input-output patterns are applied sequentially, we found that the overall ANN circuit did not retain all previous patterns as the weights are updated to learn a new pattern. The PSPICE

simulation for the sequential learning circuit is executed as follows: we initialize the ANN circuit for an input-output pattern which it would learn successfully by converging to a set of (equilibrium) weights. We then use the attained (equilibrium) weights as initial weights for the next input-output pattern. Continuing this process, until we employ all the input-output patterns and reapply the first pattern again. In this process, we found that the ANN circuit would learn each pattern that is presently applied, but it does not necessarily retain other input-output patterns that it has previously learned.

The PSPICE transient analysis is executed for the ANN circuit with the PSPICE input file in Appendix B.10 by applying pulse input waveforms for the external inputs, X_1, X_2, and the desired target, τ, i.e., continuous inputs and target pulse signal. Figure 5.2.2 shows the actual output curve corresponding to an applied input vector and its desired target. The convergent weights are the as same as the results in Table 5.2.1.

Table 5.2.1 The PSPICE transient analysis results of the feedforward ANN with the sequential learning circuit #1. The initial and steady state weights are tabulated for each pattern (units are in volts).

X_1 X_2	0.5 0.5		0.5 4.5		4.5 0.5		4.5 4.5	
τ	1.0		4.5		4.5		1.0	
$\bar{y}$	1.81		5.0		5.0		1.81	
$\tau - \bar{y}$	-0.81		-0.5		-0.5		-0.81	
weights	init.	s.s.	init.	s.s.	init.	s.s.	init.	s.s.
$\underline{w}_{11}$	-0.5	0.053	-0.5	0.02	-0.5	-0.019	-0.5	-0.052
$\underline{w}_{12}$	0.6	0.053	0.6	-0.019	0.6	0.02	0.6	-0.052
$\underline{w}_{21}$	-0.5	0.053	-0.5	0.02	-0.5	-0.019	-0.5	-0.052
$\underline{w}_{22}$	0.5	0.053	0.5	-0.019	0.5	0.02	0.5	-0.052
$\bar{w}_{11}$	0.7	0.276	0.7	0.168	0.7	0.168	0.7	0.276
$\bar{w}_{12}$	0.5	0.276	0.5	0.168	0.5	0.168	0.5	0.276

s.s means steady state and init. means initial value.

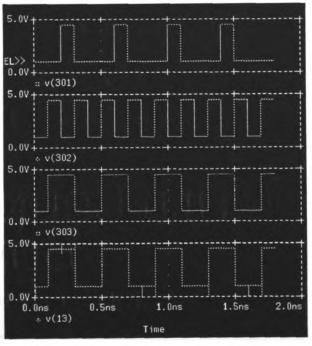


Figure 5.2.2 The PSPICE transient analysis of the circuit in Figure 5.2.1, where v(301) and v(302) are the input vector, v(303) is its desired target, and v(13) is the actual output of the circuit.

5.2.2. Feedforward ANNs with the Sequential Learning Circuit #2

The block diagram of the feedforward ANNs with the sequential learning circuit #2 is shown in Figure 5.2.3. This circuit is designed with a threshold weight learning circuit for each neuron and a nonlinearity for each weight, in addition to the feedforward ANNs with with sequential learning circuit #1.

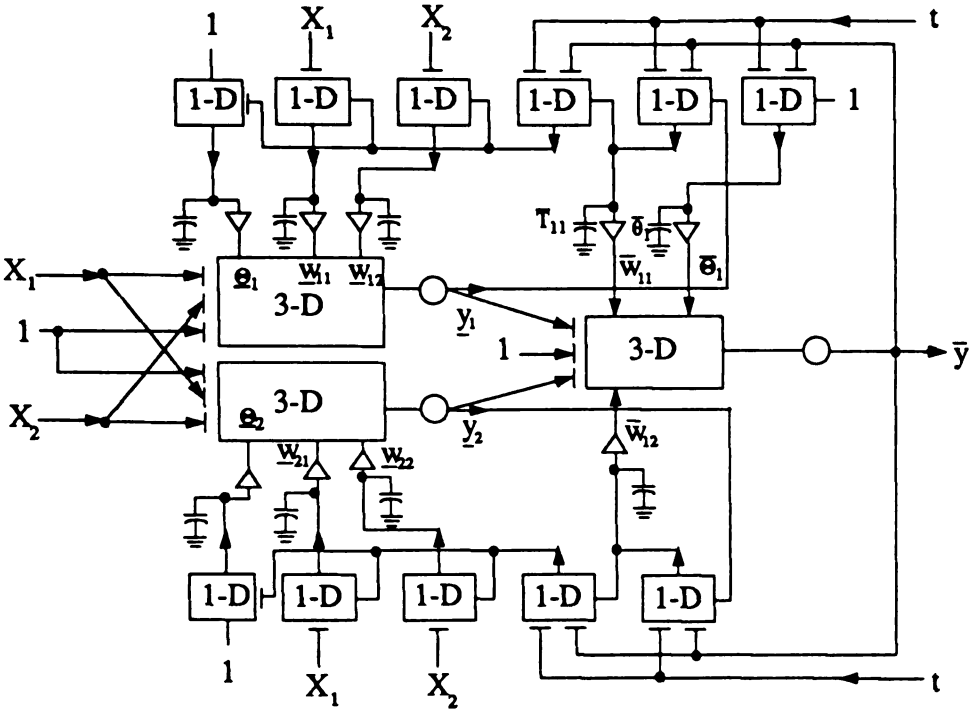


Figure 5.2.3 The feedforward ANN with the sequential learning circuit #2

The circuit implementation is described in the following equations [43]:

$$RC\ \dot{\bar{T}}_{11} = k_1\ (t_{11}-\bar{Y}_1)\ \underline{y}_{11} - \bar{T}_{11} \tag{5.2.4i}$$

$$RC\ \dot{\bar{T}}_{12} = k_1\ (t_{11}-\bar{Y}_1)\ \underline{y}_{12} - \bar{T}_{11} \tag{5.2.4ii}$$

$$RC\ \dot{\underline{T}}_{11} = k_1\ \bar{w}_{11}\ (t_{11}-\bar{Y}_1)x_{11} - \underline{T}_{11} \tag{5.2.5i}$$

$$RC\ \dot{\underline{T}}_{12} = k_1\ \bar{w}_{11}\ (t_{11}-\bar{Y}_1)x_{12} - \underline{T}_{12} \tag{5.2.5ii}$$

$$RC\ \dot{\underline{T}}_{21} = k_1\ \bar{w}_{12}\ (t_{11}-\bar{Y}_1)x_{11} - \underline{T}_{21} \tag{5.2.5iii}$$

$$RC\ \dot{\underline{T}}_{22} = k_1\ \bar{w}_{12}\ (t_{11}-\bar{Y}_1)x_{12} - \underline{T}_{22} \tag{5.2.5iv}$$

and

$$w_{ij} = S(T_{ij}).$$

The modified threshold weights are given as follows [43]:

$$\underline{\theta}_1 + RC\ \dot{\underline{\theta}}_1 = k_1\ \bar{w}_{11}\ (t_p - \bar{Y}_p)\ 1 \tag{5.2.6i}$$

$$\underline{\theta}_2 + RC\ \dot{\underline{\theta}}_2 = k_1\ \bar{w}_{12}\ (t_p - \bar{Y}_p)\ 1 \tag{5.2.6ii}$$

$$\bar{\theta}_1 + RC\ \dot{\bar{\theta}}_1 = k_1\ (t_p - \bar{Y}_p)\ 1 \tag{5.2.6iii}$$

and

$$\Theta_i = S(\theta_i),$$

where $S(.)$ represents a nonlinearity for Θ_i and w_{ij}, T_{ij} is the modified synaptic weight before it passes a nonlinearity, and θ_i is the modified threshold weight before it passes a nonlinearity.

The circuit implementation of the governing equations now results in

$$\underline{Y}_{p1} = S_n(\ k_3(\sum_j \underline{w}_{1j}x_{pj} + \underline{\Theta}_1)) \tag{5.2.7i}$$

$$\underline{Y}_{p2} = S_n(\ k_3(\sum_j \underline{w}_{2j}x_{pj} + \underline{\Theta}_2)) \tag{5.2.7ii}$$

$$\bar{Y}_p = S_n(\ k_3(\sum_j \bar{w}_{1j}y_{pj} + \bar{\Theta}_1)). \tag{5.2.7iii}$$

This circuit is simulated using the PSPICE circuit simulator. The learning tasks are executed sequentially in the following steps:

1. Initialize the ANN circuit with the initial condition which is at the top of each table.
2. The first input-output pattern is applied and the PSPICE transient analysis is executed. The measure the steady state weight values.
3. Use these steady state weight values as the initial condition and run the PSPICE transient analysis for the next input-output pattern. Then measure the steady state weight values.
4. Go to Step 3 until all the learning tasks are done.
5. After completing the learning tasks for all the input-output patterns, the PSPICE dc analysis is executed by setting the weights to the steady state weight values for each learning task. Then measure the output error $\tau - \bar{y}$.

Table 5.2.2 and Table 5.2.3 summarize the results of the PSPICE simulations. One of the PSPICE input files for Table 5.2.3 is shown in Appendix B.11. The simulation results show that this circuit does not learn all the desired targets simultaneously.

For example, let's consider Table 5.2.2. Table 5.2.2(a) contains the convergent connection weights of the feedforward ANNs with the sequential learning after executing the PSPICE transient analysis. With these weights, the PSPICE dc analysis is executed and the results are summarized in Table 5.2.2(b). Note that in Table 5.2.2(b), column #1,5 means that the dc analysis is performed using the data of column #1 and the data of column #5 in Table 5.2.2(a), separately, and their results are summarized in the same column since both results are the same.

In column #1,5, the pattern (X_1 X_2 τ)=(4.5 4.5 0.5) is learned with an error equal to -0.0002 and the pattern (X_1 X_2 τ)=(1.0 1.0 0.5) seems to be learned with an error equal to 0.5., but the patterns (X_1 X_2 τ)=(1.0 4.5 4.5) and (4.5 1.0 4.5) fail to be learned. In column #3,4, the patterns (X_1 X_2 τ)=(1.0 4.5 4.5) and (4.5 1.0 4.5) are

learned with an error equal to -0.001, the pattern (X_1 X_2 τ)=(1.0 1.0 0.5) seems to be learned with an error equal to 0.5, however the pattern (X_1 X_2 τ)=(4.5 4.5 0.5) fails to be learned.

Table 5.2.2 (a) The PSPICE transient analysis results of the feedforward ANN with the sequential learning circuit #2 (w/o amplification) (units are in volts). τ is a target and $\bar{y}$ is an actual output. The initial conditions are given by
($\underline{T}_{11}(0)$ $\underline{T}_{12}(0)$ $\underline{T}_{21}(0)$ $\underline{T}_{22}(0)$ $\bar{T}_{11}(0)$ $\bar{T}_{12}(0)$ $\underline{\theta}_1(0)$ $\underline{\theta}_2(0)$ $\bar{\theta}_1(0)$) = (0.5 -0.5 -0.5 0.5 0.5 0.5 -0.5 -0.5 0.5)
($\underline{w}_{11}(0)$ $\underline{w}_{12}(0)$ $\underline{w}_{21}(0)$ $\underline{w}_{22}(0)$ $\bar{w}_{11}(0)$ $\bar{w}_{12}(0)$ $\underline{\Theta}_1(0)$ $\underline{\Theta}_2(0)$ $\bar{\Theta}_1(0)$) = (1.0 -1.0 -1.0 1.0 1.0 1.0 -1.0 -1.0 1.0).

	Column #1	Column #2	Column #3	Column #4	Column #5
x_1	4.5	1.0	1.0	4.5	4.5
x_2	4.5	1.0	4.5	1.0	4.5
τ	0.5	0.5	4.5	4.5	0.5
$\bar{y}$	0.5002	0.4999	4.501	4.501	0.5002
$\tau-\bar{y}$	-0.002	0.001	-0.001	-0.001	-0.002
$\underline{T}_{11}$	4.210E-04	-2.511E-04	-2.901E-04	3.817E-04	4.565E-04
$\underline{T}_{12}$	4.210E-04	-2.511E-04	3.817E-04	-2.901E-04	4.565E-04
$\underline{T}_{21}$	4.210E-04	-2.511E-04	-2.901E-04	3.817E-04	4.565E-04
$\underline{T}_{22}$	4.210E-04	-2.511E-04	3.817E-04	-2.901E-04	4.565E-04
$\bar{T}_{11}$	3.410E-03	3.411E-03	3.410E-03	3.410E-03	3.410E-03
$\bar{T}_{12}$	3.410E-03	3.411E-03	3.410E-03	3.410E-03	3.410E-03
$\underline{\theta}_1$	4.210E-04	3.304E-04	3.817E-04	3.817E-04	4.565E-04
$\underline{\theta}_2$	4.210E-04	3.304E-04	3.817E-04	3.817E-04	4.565E-04
$\bar{\theta}_1$	-3.269E-03	2.739E-03	-2.723E-03	-2.723E-03	-3.771E-03
$\underline{w}_{11}$	-1	-1	-1	-1	-1
$\underline{w}_{12}$	-1	-1	-1	-1	-1
$\underline{w}_{21}$	-1	-1	-1	-1	-1
$\underline{w}_{22}$	-1	-1	-1	-1	-1
$\bar{w}_{11}$	-0.1393	0.1396	-0.1663	-0.1663	-0.1393
$\bar{w}_{12}$	-0.1393	0.1396	-0.1663	-0.1663	-0.1393
$\underline{\Theta}_1$	-1	-1	-1	-1	-1
$\underline{\Theta}_2$	-1	-1	-1	-1	-1
$\bar{\Theta}_1$	-1	-1	-1	-1	-1

Table 5.2.2(b) The PSPICE DC analysis results of the feedforward ANN circuit with the convergent weights in Table 5.2.2(a). τ is a target, $\underline{y}_j$ is an output of the hidden layer, and $\bar{y}$ is an output of the output layer.

			Column #1,5				Column #2				Column #3,4			
x_1	x_2	τ	$\underline{y}_1$	$\underline{y}_2$	$\bar{y}$	$\tau-\bar{y}$	$\underline{y}_1$	$\underline{y}_2$	$\bar{y}$	$\tau-\bar{y}$	$\underline{y}_1$	$\underline{y}_2$	$\bar{y}$	$\tau-\bar{y}$
1.0	1.0	0.5	5.0	5.0	0.0	0.5	5.0	5.0	0.4995	0.005	5.0	5.0	0.0	0.5
1.0	4.5	4.5	0.0	0.0	0.5002	3.998	0.0	0.0	0.0	4.5	0.0	0.0	4.501	-0.001
4.5	1.0	4.5	0.0	0.0	0.5002	3.998	0.0	0.0	0.0	4.5	0.0	0.0	4.501	-0.001
4.5	4.5	0.5	0.0	0.0	0.5002	-0.002	0.0	0.0	0.0	0.5	0.0	0.0	4.501	-4.001

Table 5.2.3(a) The PSPICE transient analysis results of the feedforward ANN with the sequential learning circuit #2 (w/o amplification) (units are in volts). τ is a target and $\bar{y}$ is an actual output. The initial conditions are given by

($\underline{T}_{11}(0)$ $\underline{T}_{12}(0)$ $\underline{T}_{21}(0)$ $\underline{T}_{22}(0)$ $\bar{T}_{11}(0)$ $\bar{T}_{12}(0)$ $\underline{\theta}_1(0)$ $\underline{\theta}_2(0)$ $\bar{\theta}_1(0)$) = (0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0)
($\underline{w}_{11}(0)$ $\underline{w}_{12}(0)$ $\underline{w}_{21}(0)$ $\underline{w}_{22}(0)$ $\bar{w}_{11}(0)$ $\bar{w}_{12}(0)$ $\underline{\Theta}_1(0)$ $\underline{\Theta}_2(0)$ $\bar{\Theta}_1(0)$) = (1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0).

	Column #1	Column #2	Column #3	Column #4	Column #5
x_1	4.5	1.0	1.0	4.5	4.5
x_2	4.5	1.0	4.5	1.0	4.5
τ	0.5	0.5	4.5	4.5	0.5
$\bar{y}$	0.5099	0.4928	4.51	4.51	0.5099
$\tau-\bar{y}$	-0.099	0.0072	-0.01	-0.01	-0.099
$\underline{T}_{11}$	4.607E-04	-2.507E-04	-2.85E-04	3.284E-04	4.535E-04
$\underline{T}_{12}$	4.607E-04	-2.507E-04	3.751E-04	-2.906E-04	4.535E-04
$\underline{T}_{21}$	4.607E-04	-2.507E-04	-2.85E-04	3.284E-04	4.535E-04
$\underline{T}_{22}$	4.607E-04	-2.507E-04	3.751E-04	-2.906E-04	4.535E-04
$\bar{T}_{11}$	3.410E-03	3.411E-03	3.410E-03	3.410E-03	3.410E-03
$\bar{T}_{12}$	3.410E-03	3.411E-03	3.410E-03	3.410E-03	3.410E-03
$\underline{\theta}_1$	4.607E-04	3.300E-04	3.751E-04	3.824E-04	4.535E-04
$\underline{\theta}_2$	4.607E-04	3.300E-04	3.751E-04	3.824E-04	4.535E-04
$\bar{\theta}_1$	-3.804E-03	2.748E-03	-2.723E-03	-2.723E-03	-3.739E-03
$\underline{w}_{11}$	-1	-1	-1	-1	-1
$\underline{w}_{12}$	-1	-1	-1	-1	-1
$\underline{w}_{21}$	-1	-1	-1	-1	-1
$\underline{w}_{22}$	-1	-1	-1	-1	-1
$\bar{w}_{11}$	-0.1396	0.1394	-0.1663	-0.1663	-0.1396
$\bar{w}_{12}$	-0.1396	0.1394	-0.1663	-0.1663	-0.1396
$\underline{\Theta}_1$	-1	-1	-1	-1	-1
$\underline{\Theta}_2$	-1	-1	-1	-1	-1
$\bar{\Theta}_1$	-1	-1	-1	-1	-1

Table 5.2.3(b) The PSPICE DC analysis results of the feedforward ANN circuit with the convergent weights in Table 5.2.3(a). τ is a target, y_j is an output of the hidden layer, and $\bar{y}$ is an output of the output layer.

			Column #1,5				Column #2,6				Column #3,4,7,8			
x_1	x_2	τ	y_1	y_2	$\bar{y}$	$\tau-\bar{y}$	y_1	y_2	$\bar{y}$	$\tau-\bar{y}$	y_1	y_2	$\bar{y}$	$\tau-\bar{y}$
1.0	1.0	0.5	5.0	5.0	0.0	0.5	5.0	5.0	0.4931	0.0069	5.0	5.0	0.0	0.5
1.0	4.5	4.5	0.0	0.0	0.5107	3.9893	0.0	0.0	0.0	4.5	0.0	0.0	4.511	-0.011
4.5	1.0	4.5	0.0	0.0	0.5107	3.9893	0.0	0.0	0.0	4.5	0.0	0.0	4.511	-0.011
4.5	4.5	0.5	0.0	0.0	0.5107	-0.0107	0.0	0.0	0.0	0.5	0.0	0.0	4.511	-4.011

5.3. Feedforward ANNs with the Simultaneous Learning

From the above simulation results, we may conclude that feedforward ANNs with the sequential learning do not (easily) completely learn all the desired patterns. In this section, feedforward ANNs with the simultaneous learning are considered by expanding the circuit for feedforward ANNs with the sequential learning, i.e., by employing a number of feedforward ANNs with the sequential learning circuit equal to the number of patterns.

Two different feedforward ANN with simultaneous learning circuits are suggested by Dr. Salam and have been implemented. These two implementations are the same except that there is an amplification unit before each capacitor (or after a multiplier). Accordingly, when these circuits are represented as model equations, they have the same form except this amplification constant, α.

Prototype 2x2x1 feedforward ANNs with the simultaneous learning are implemented when the number of pattern is four, i.e., $p = 4$. Figure 5.3.1 depicts the feedforward ANN with simultaneous learning circuit with an amplification unit ($\alpha = 50$) and without one ($\alpha = 1$), where the amplification unit is represented by the rectangular box. The circuit implementation is described by the following equations:

$$RC\,\dot{\bar{T}}_{11} = \alpha\, k_4\, [t_{11}-\bar{Y}_1\; t_{21}-\bar{Y}_2\; t_{31}-\bar{Y}_3\; t_{41}-\bar{Y}_4]\; [\underline{y}_{11}\; \underline{y}_{21}\; \underline{y}_{31}\; \underline{y}_{41}]^T - \bar{T}_{11} \qquad (5.3.1i)$$

$$RC\,\dot{\bar{T}}_{12} = \alpha\, k_4\, [t_{11}-\bar{Y}_1\; t_{21}-\bar{Y}_2\; t_{31}-\bar{Y}_3\; t_{41}-\bar{Y}_4]\; [\underline{y}_{12}\; \underline{y}_{22}\; \underline{y}_{32}\; \underline{y}_{42}]^T - \bar{T}_{12} \qquad (5.3.1ii)$$

$$RC\,\dot{\underline{T}}_{11} = \alpha\, k_4\, \bar{w}_{11}\, [t_{11}-\bar{Y}_1\; t_{21}-\bar{Y}_2\; t_{31}-\bar{Y}_3\; t_{41}-\bar{Y}_4]\; [x_{11}\; x_{21}\; x_{31}\; x_{41}]^T - \underline{T}_{11} \qquad (5.3.2i)$$

$$RC\,\dot{\underline{T}}_{12} = \alpha\, k_4\, \bar{w}_{11}\, [t_{11}-\bar{Y}_1\; t_{21}-\bar{Y}_2\; t_{31}-\bar{Y}_3\; t_{41}-\bar{Y}_4]\; [x_{12}\; x_{22}\; x_{32}\; x_{42}]^T - \underline{T}_{12} \qquad (5.3.2ii)$$

$$RC\,\dot{\underline{T}}_{21} = \alpha\, k_4\, \bar{w}_{12}\, [t_{11}-\bar{Y}_1\; t_{21}-\bar{Y}_2\; t_{31}-\bar{Y}_3\; t_{41}-\bar{Y}_4]\; [x_{11}\; x_{21}\; x_{31}\; x_{41}]^T - \underline{T}_{21} \qquad (5.3.2iii)$$

$$RC\,\dot{\underline{T}}_{22} = \alpha\, k_4\, \bar{w}_{12}\, [t_{11}-\bar{Y}_1\; t_{21}-\bar{Y}_2\; t_{31}-\bar{Y}_3\; t_{41}-\bar{Y}_4]\; [x_{12}\; x_{22}\; x_{32}\; x_{42}]^T - \underline{T}_{22} \qquad (5.3.2iv)$$

and

$w_{ij} = S(T_{ij})$.

The modified threshold weights are given as follows:

$$\underline{\theta}_1 + RC\ \dot{\underline{\theta}}_1 = \alpha\, k_4\, \bar{w}_{11} \sum_p (t_p - \bar{Y}_p)\, 1 \tag{5.3.3i}$$

$$\underline{\theta}_2 + RC\ \dot{\underline{\theta}}_2 = \alpha\, k_4\, \bar{w}_{12} \sum_p (t_p - \bar{Y}_p)\, 1 \tag{5.3.3ii}$$

$$\bar{\theta}_1 + RC\ \dot{\bar{\theta}}_1 = \alpha\, k_4 \sum_p (t_p - \bar{Y}_p)\, 1 \tag{5.3.3iii}$$

and

$$\Theta_i = S(\theta_i),$$

where $S(.)$ represents a nonlinearity for w_{ij} and Θ_i, T_{ij} is the modified synaptic weight, and θ_i is the modified threshold weight, before it passes through the non-linearity.

The circuit implementation of the governing equation now results in

$$\underline{Y}_{p1} = S_n(\, k_3(\sum_j \underline{w}_{1j} x_{pj} + \underline{\Theta}_1)) \tag{5.3.4i}$$

$$\underline{Y}_{p2} = S_n(\, k_3(\sum_j \underline{w}_{2j} x_{pj} + \underline{\Theta}_2)) \tag{5.3.4ii}$$

$$\bar{Y}_p = S_n(\, k_3(\sum_j \bar{w}_{1j} \underline{y}_{pj} + \bar{\Theta}_1)). \tag{5.3.4iii}$$

The feedforward ANN with simultaneous learning circuit with an amplification unit ($\alpha = 50$) is simulated using the PSPICE circuit simulator as follows:

1. Initialize the ANN circuit with the initial condition which is at the top of each table.

2. All the input-output patterns are applied and the PSPICE transient analysis is performed.

3. Measure the steady state weight values and the output errors, namely, $\tau - \bar{y}$.

The results are summarized in Table 5.3.1. Table 5.3.1(a) and Table 5.3.1(c) conclude that the feedforward ANN with simultaneous learning circuit is simulated with the same initial condition but with different *logic 0* value. And similarly for Table 5.3.1(b) and Table 5.3.1(d). As shown in Table 5.3.1, all four input-output patterns are successfully learned.

Next, the feedforward ANN with simultaneous learning circuit without an amplification unit ($\alpha = 1$) is simulated in the same manner. Four different initial condition sets are given for the simulation of this circuit. Two of them are the same as in Table 5.3.1 and the other two sets are obtained from dividing the initial conditions of Table 5.3.1 by the amplification factor of the first feedforward ANN with simultaneous learning circuit, i.e., 50. Their results are summarized in Table 5.3.2 and 5.3.3. However, depending on the initial conditions, this circuit may not always learn all the desired targets but learn some desired targets. For example in Table 5.3.2(a) and Table 5.3.2(c), Table 5.3.2(c) shows that this circuit learns all the desired targets but not in Table 5.3.2(a). The difference of these two simulations is the logic 0 value, i.e., Table 5.3.2(a) is the result of using 1.0v as the logic 0 value and Table 5.3.2(c) is the result of using 0.5v as the logic 0 value.

The PSPICE input files for Table 5.3.1 and Table 5.3.2 are shown in Appendix B.12 and Appendix B.13, respectively.

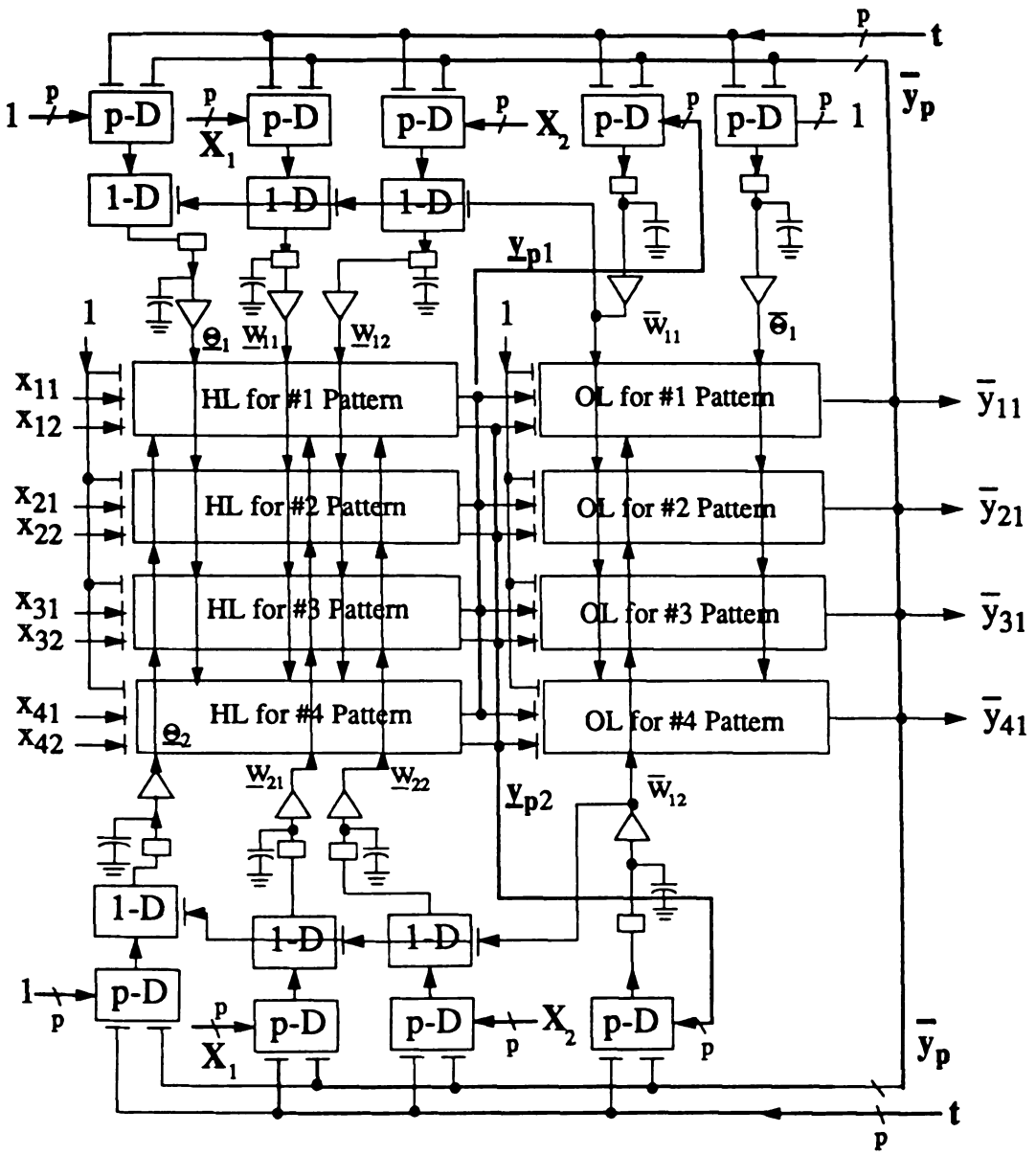


Figure 5.3.1 The feedforward ANN with the simultaneous learning circuit with threshold weight circuits, nonlinearity circuits, and amplification units

Table 5.3.1 The PSPICE transient analysis results of the feedforward ANN with the simultaneous learning when $\alpha = 50$ (units are in volts). τ is a target and $\bar{y}$ is an actual output.

(a) Logic 1 = 4.5 and logic 0 = 1.0

($\underline{T}_{11}(0)$ $\underline{T}_{12}(0)$ $\underline{T}_{21}(0)$ $\underline{T}_{22}(0)$ $\bar{T}_{11}(0)$ $\bar{T}_{12}(0)$ $\underline{\theta}_1(0)$ $\underline{\theta}_2(0)$ $\bar{\theta}_1(0)$) = (0.5 -0.5 -0.5 0.5 0.5 0.5 -0.5 -0.5 0.5)

($\underline{w}_{11}(0)$ $\underline{w}_{12}(0)$ $\underline{w}_{21}(0)$ $\underline{w}_{22}(0)$ $\bar{w}_{11}(0)$ $\bar{w}_{12}(0)$ $\underline{\Theta}_1(0)$ $\underline{\Theta}_2(0)$ $\bar{\Theta}_1(0)$) = (1.0 -1.0 -1.0 1.0 1.0 1.0 -1.0 -1.0 1.0)

τ	1.0		4.5		4.5		1.0	
$\bar{y}$	1.0		4.498		4.499		1.005	
$\tau-\bar{y}$	0.0		0.002		0.001		-0.005	
$\underline{T}_{11}$	$\underline{T}_{12}$	$\underline{T}_{21}$	$\underline{T}_{22}$	$\bar{T}_{11}$	$\bar{T}_{12}$	$\underline{\theta}_1$	$\underline{\theta}_2$	$\bar{\theta}_1$
8.440E-04	-1.287E-03	3.412E-03	3.412E-03	3.411E-03	3.410E-03	6.068E-05	1.012E-03	-3.819E-03
$\underline{w}_{11}$	$\underline{w}_{12}$	$\underline{w}_{21}$	$\underline{w}_{22}$	$\bar{w}_{11}$	$\bar{w}_{12}$	$\underline{\Theta}_1$	$\underline{\Theta}_2$	$\bar{\Theta}_1$
-1.0	-1.0	0.1661	0.1628	-0.01825	-0.314	-1.0	-1.0	-1.0

(b) Logic 1 = 4.5 and logic 0 = 1.0

($\underline{T}_{11}(0)$ $\underline{T}_{12}(0)$ $\underline{T}_{21}(0)$ $\underline{T}_{22}(0)$ $\bar{T}_{11}(0)$ $\bar{T}_{12}(0)$ $\underline{\theta}_1(0)$ $\underline{\theta}_2(0)$ $\bar{\theta}_1(0)$) = (-0.5 0.5 0.5 -0.5 -0.5 -0.5 -0.5 -0.5 0.5)

($\underline{w}_{11}(0)$ $\underline{w}_{12}(0)$ $\underline{w}_{21}(0)$ $\underline{w}_{22}(0)$ $\bar{w}_{11}(0)$ $\bar{w}_{12}(0)$ $\underline{\Theta}_1(0)$ $\underline{\Theta}_2(0)$ $\bar{\Theta}_1(0)$) = (-1.0 1.0 1.0 -1.0 -1.0 -1.0 -1.0 -1.0 1.0)

τ	1.0		4.5		4.5		1.0	
$\bar{y}$	1.0		4.499		4.499		1.004	
$\tau-\bar{y}$	0.0		0.001		0.001		-0.004	
$\underline{T}_{11}$	$\underline{T}_{12}$	$\underline{T}_{21}$	$\underline{T}_{22}$	$\bar{T}_{11}$	$\bar{T}_{12}$	$\underline{\theta}_1$	$\underline{\theta}_2$	$\bar{\theta}_1$
1.676E-04	-3.701E-03	3.412E-03	3.412E-03	3.411E-03	3.410E-03	-9.141E-05	1.552E-03	-5.825E-03
$\underline{w}_{11}$	$\underline{w}_{12}$	$\underline{w}_{21}$	$\underline{w}_{22}$	$\bar{w}_{11}$	$\bar{w}_{12}$	$\underline{\Theta}_1$	$\underline{\Theta}_2$	$\bar{\Theta}_1$
-1.0	-1.0	0.1646	0.1643	-0.01826	-0.314	-1.0	-1.0	-1.0

(c) Logic 1 = 4.5 and logic 0 = 0.5

($\underline{T}_{11}(0)$ $\underline{T}_{12}(0)$ $\underline{T}_{21}(0)$ $\underline{T}_{22}(0)$ $\bar{T}_{11}(0)$ $\bar{T}_{12}(0)$ $\underline{\theta}_1(0)$ $\underline{\theta}_2(0)$ $\bar{\theta}_1(0)$) = (0.5 -0.5 -0.5 0.5 0.5 0.5 -0.5 -0.5 0.5)

($\underline{w}_{11}(0)$ $\underline{w}_{12}(0)$ $\underline{w}_{21}(0)$ $\underline{w}_{22}(0)$ $\bar{w}_{11}(0)$ $\bar{w}_{12}(0)$ $\underline{\Theta}_1(0)$ $\underline{\Theta}_2(0)$ $\bar{\Theta}_1(0)$) = (1.0 -1.0 -1.0 1.0 1.0 1.0 -1.0 -1.0 1.0)

τ	0.5		4.5		4.5		0.5	
$\bar{y}$	0.5003		4.499		4.498		0.5041	
$\tau-\bar{y}$	-0.0003		0.001		0.002		-0.0041	
$\underline{T}_{11}$	$\underline{T}_{12}$	$\underline{T}_{21}$	$\underline{T}_{22}$	$\bar{T}_{11}$	$\bar{T}_{12}$	$\underline{\theta}_1$	$\underline{\theta}_2$	$\bar{\theta}_1$
3.412E-03	3.412E-03	-1.053E-03	-2.428E-03	3.410E-03	3.411E-03	2.161E-03	-2.443E-04	-5.490E-03
$\underline{w}_{11}$	$\underline{w}_{12}$	$\underline{w}_{21}$	$\underline{w}_{22}$	$\bar{w}_{11}$	$\bar{w}_{12}$	$\underline{\Theta}_1$	$\underline{\Theta}_2$	$\bar{\Theta}_1$
0.1714	0.1733	-1.0	-1.0	-0.3046	-0.02738	-1.0	-1.0	-1.0

(d) Logic 1 = 4.5 and logic 0 = 0.5

($\underline{T}_{11}(0)$ $\underline{T}_{12}(0)$ $\underline{T}_{21}(0)$ $\underline{T}_{22}(0)$ $\bar{T}_{11}(0)$ $\bar{T}_{12}(0)$ $\underline{\theta}_1(0)$ $\underline{\theta}_2(0)$ $\bar{\theta}_1(0)$) = (-0.5 0.5 0.5 -0.5 -0.5 -0.5 -0.5 -0.5 0.5)

($\underline{w}_{11}(0)$ $\underline{w}_{12}(0)$ $\underline{w}_{21}(0)$ $\underline{w}_{22}(0)$ $\bar{w}_{11}(0)$ $\bar{w}_{12}(0)$ $\underline{\Theta}_1(0)$ $\underline{\Theta}_2(0)$ $\bar{\Theta}_1(0)$) = (-1.0 1.0 1.0 -1.0 -1.0 -1.0 -1.0 -1.0 1.0)

τ	0.5		4.5		4.5		0.5	
$\bar{y}$	0.5197		4.498		4.499		0.4818	
$\tau-\bar{y}$	-0.0197		0.002		0.001		0.0182	
$\underline{T}_{11}$	$\underline{T}_{12}$	$\underline{T}_{21}$	$\underline{T}_{22}$	$\bar{T}_{11}$	$\bar{T}_{12}$	$\underline{\theta}_1$	$\underline{\theta}_2$	$\bar{\theta}_1$
3.410E-03	3.413E-03	3.414E-03	3.336E-03	3.411E-03	3.411E-03	7.826E-05	7.796E-05	3.410E-03
$\underline{w}_{11}$	$\underline{w}_{12}$	$\underline{w}_{21}$	$\underline{w}_{22}$	$\bar{w}_{11}$	$\bar{w}_{12}$	$\underline{\Theta}_1$	$\underline{\Theta}_2$	$\bar{\Theta}_1$
-0.2915	0.4228	0.6703	-0.9676	0.02708	0.02697	-1.0	-1.0	-0.1826

Table 5.3.2 The PSPICE transient analysis results of the feedforward ANN with the simultaneous learning when $\alpha = 1$ (units are in volts). τ is a target and $\bar{y}$ is an actual output.

(a) Logic 1 = 4.5 and logic 0 = 1.0

($\underline{T}_{11}(0)$ $\underline{T}_{12}(0)$ $\underline{T}_{21}(0)$ $\underline{T}_{22}(0)$ $\bar{T}_{11}(0)$ $\bar{T}_{12}(0)$ $\underline{\theta}_1(0)$ $\underline{\theta}_2(0)$ $\bar{\theta}_1(0)$) = (0.5 -0.5 -0.5 0.5 0.5 0.5 -0.5 -0.5 0.5)

($\underline{w}_{11}(0)$ $\underline{w}_{12}(0)$ $\underline{w}_{21}(0)$ $\underline{w}_{22}(0)$ $\bar{w}_{11}(0)$ $\bar{w}_{12}(0)$ $\underline{\Theta}_1(0)$ $\underline{\Theta}_2(0)$ $\bar{\Theta}_1(0)$) = (1.0 -1.0 -1.0 1.0 1.0 1.0 -1.0 -1.0 1.0)

τ	1.0		4.5		4.5		1.0	
$\bar{y}$	0.9617		3.332		3.332		3.332	
$\tau-\bar{y}$	0.0383		1.168		1.168		-2.332	
$\underline{T}_{11}$	$\underline{T}_{12}$	$\underline{T}_{21}$	$\underline{T}_{22}$	$\bar{T}_{11}$	$\bar{T}_{12}$	$\underline{\theta}_1$	$\underline{\theta}_2$	$\bar{\theta}_1$
-5.645E-04	-5.698E-04	1.380E-03	1.380E-03	3.411E-03	3.411E-03	1.143E-05	-2.820E-05	3.410E-03
$\underline{w}_{11}$	$\underline{w}_{12}$	$\underline{w}_{21}$	$\underline{w}_{22}$	$\bar{w}_{11}$	$\bar{w}_{12}$	$\underline{\Theta}_1$	$\underline{\Theta}_2$	$\bar{\Theta}_1$
-1.0	-1.0	-1.0	-1.0	3.955E-03	-9.753E-03	-1.0	-1.0	-0.2586

(b) Logic 1 = 4.5 and logic 0 = 1.0

($\underline{T}_{11}(0)$ $\underline{T}_{12}(0)$ $\underline{T}_{21}(0)$ $\underline{T}_{22}(0)$ $\bar{T}_{11}(0)$ $\bar{T}_{12}(0)$ $\underline{\theta}_1(0)$ $\underline{\theta}_2(0)$ $\bar{\theta}_1(0)$) = (-0.5 0.5 0.5 -0.5 -0.5 -0.5 -0.5 -0.5 0.5)

($\underline{w}_{11}(0)$ $\underline{w}_{12}(0)$ $\underline{w}_{21}(0)$ $\underline{w}_{22}(0)$ $\bar{w}_{11}(0)$ $\bar{w}_{12}(0)$ $\underline{\Theta}_1(0)$ $\underline{\Theta}_2(0)$ $\bar{\Theta}_1(0)$) = (-1.0 1.0 1.0 -1.0 -1.0 -1.0 -1.0 -1.0 1.0)

τ	1.0		4.5		4.5		1.0	
$\bar{y}$	0.9617		3.337		3.325		3.334	
$\tau-\bar{y}$	0.0383		1.163		1.175		-2.334	
$\underline{T}_{11}$	$\underline{T}_{12}$	$\underline{T}_{21}$	$\underline{T}_{22}$	$\bar{T}_{11}$	$\bar{T}_{12}$	$\underline{\theta}_1$	$\underline{\theta}_2$	$\bar{\theta}_1$
3.358E-04	3.288E-03	-2.532E-03	-2.508E-03	3.411E-03	3.411E-03	-6.659E-05	4.982E-05	3.410E-03
$\underline{w}_{11}$	$\underline{w}_{12}$	$\underline{w}_{21}$	$\underline{w}_{22}$	$\bar{w}_{11}$	$\bar{w}_{12}$	$\underline{\Theta}_1$	$\underline{\Theta}_2$	$\bar{\Theta}_1$
-0.9503	-0.9894	-1.0	-1.0	-0.02359	0.01954	-1.0	-1.0	-0.2586

(c) Logic 1 = 4.5 and logic 0 = 0.5

($\underline{T}_{11}(0)$ $\underline{T}_{12}(0)$ $\underline{T}_{21}(0)$ $\underline{T}_{22}(0)$ $\bar{T}_{11}(0)$ $\bar{T}_{12}(0)$ $\underline{\theta}_1(0)$ $\underline{\theta}_2(0)$ $\bar{\theta}_1(0)$) = (0.5 -0.5 -0.5 0.5 0.5 0.5 -0.5 -0.5 0.5)

($\underline{w}_{11}(0)$ $\underline{w}_{12}(0)$ $\underline{w}_{21}(0)$ $\underline{w}_{22}(0)$ $\bar{w}_{11}(0)$ $\bar{w}_{12}(0)$ $\underline{\Theta}_1(0)$ $\underline{\Theta}_2(0)$ $\bar{\Theta}_1(0)$) = (1.0 -1.0 -1.0 1.0 1.0 1.0 -1.0 -1.0 1.0)

τ	0.5		4.5		4.5		0.5	
$\bar{y}$	0.5126		4.434		4.434		0.677	
$\tau-\bar{y}$	-0.0126		0.066		0.066		-0.177	
$\underline{T}_{11}$	$\underline{T}_{12}$	$\underline{T}_{21}$	$\underline{T}_{22}$	$\bar{T}_{11}$	$\bar{T}_{12}$	$\underline{\theta}_1$	$\underline{\theta}_2$	$\bar{\theta}_1$
3.412E-03	3.412E-03	2.516E-04	3.047E-04	3.410E-03	3.411E-03	1.215E-03	9.873E-05	-4.714E-03
$\underline{w}_{11}$	$\underline{w}_{12}$	$\underline{w}_{21}$	$\underline{w}_{22}$	$\bar{w}_{11}$	$\bar{w}_{12}$	$\underline{\Theta}_1$	$\underline{\Theta}_2$	$\bar{\Theta}_1$
0.168	0.1679	-1.0	-1.0	-0.3032	-0.0253	-1.0	-1.0	-1.0

(d) Logic 1 = 4.5 and logic 0 = 0.5

($\underline{T}_{11}(0)$ $\underline{T}_{12}(0)$ $\underline{T}_{21}(0)$ $\underline{T}_{22}(0)$ $\bar{T}_{11}(0)$ $\bar{T}_{12}(0)$ $\underline{\theta}_1(0)$ $\underline{\theta}_2(0)$ $\bar{\theta}_1(0)$) = (-0.5 0.5 0.5 -0.5 -0.5 -0.5 -0.5 -0.5 0.5)

($\underline{w}_{11}(0)$ $\underline{w}_{12}(0)$ $\underline{w}_{21}(0)$ $\underline{w}_{22}(0)$ $\bar{w}_{11}(0)$ $\bar{w}_{12}(0)$ $\underline{\Theta}_1(0)$ $\underline{\Theta}_2(0)$ $\bar{\Theta}_1(0)$) = (-1.0 1.0 1.0 -1.0 -1.0 -1.0 -1.0 -1.0 1.0)

τ	0.5		4.5		4.5		0.5	
$\bar{y}$	0.4652		3.157		3.157		3.180	
$\tau-\bar{y}$	0.0348		1.343		1.343		-2.680	
$\underline{T}_{11}$	$\underline{T}_{12}$	$\underline{T}_{21}$	$\underline{T}_{22}$	$\bar{T}_{11}$	$\bar{T}_{12}$	$\underline{\theta}_1$	$\underline{\theta}_2$	$\bar{\theta}_1$
1.477E-03	1.477E-03	1.411E-03	1.411E-03	3.411E-03	3.411E-03	4.643E-07	1.176E-06	3.410E-03
$\underline{w}_{11}$	$\underline{w}_{12}$	$\underline{w}_{21}$	$\underline{w}_{22}$	$\bar{w}_{11}$	$\bar{w}_{12}$	$\underline{\Theta}_1$	$\underline{\Theta}_2$	$\bar{\Theta}_1$
-1.0	-1.0	-1.0	-1.0	-7.957E-03	-7.612E-03	-1.0	-1.0	-0.2835

Table 5.3.3 The PSPICE transient analysis results of the feedforward ANN with simultaneous learning when $\alpha = 1$ (units are in volts). τ is a target and $\bar{y}$ is an actual output.

(a) Logic 1 = 4.5 and logic 0 = 1.0

($\underline{T}_{11}(0)$ $\underline{T}_{12}(0)$ $\underline{T}_{21}(0)$ $\underline{T}_{22}(0)$ $\bar{T}_{11}(0)$ $\bar{T}_{12}(0)$ $\underline{\theta}_1(0)$ $\underline{\theta}_2(0)$ $\bar{\theta}_1(0)$) = (0.01 -0.01 -0.01 0.01 0.01 0.01 -0.01 -0.01 0.01)

($\underline{w}_{11}(0)$ $\underline{w}_{12}(0)$ $\underline{w}_{21}(0)$ $\underline{w}_{22}(0)$ $\bar{w}_{11}(0)$ $\bar{w}_{12}(0)$ $\underline{\Theta}_1(0)$ $\underline{\Theta}_2(0)$ $\bar{\Theta}_1(0)$) = (1.0 -1.0 -1.0 1.0 1.0 1.0 -1.0 -1.0 1.0)

τ	1.0		4.5		4.5		1.0	
$\bar{y}$	0.9617		3.331		3.330		3.335	
$\tau-\bar{y}$	0.0383		1.169		1.17		-2.335	
$\underline{T}_{11}$	$\underline{T}_{12}$	$\underline{T}_{21}$	$\underline{T}_{22}$	$\bar{T}_{11}$	$\bar{T}_{12}$	$\underline{\theta}_1$	$\underline{\theta}_2$	$\bar{\theta}_1$
2.464E-04	2.463E-04	5.757E-04	5.758E-04	3.411E-03	3.411E-03	-5.021E-06	-1.175E-05	3.410E-03
$\underline{w}_{11}$	$\underline{w}_{12}$	$\underline{w}_{21}$	$\underline{w}_{22}$	$\bar{w}_{11}$	$\bar{w}_{12}$	$\underline{\Theta}_1$	$\underline{\Theta}_2$	$\bar{\Theta}_1$
-1.0	-1.0	-1.0	-1.0	-1.737E-03	-4.066E-03	-1.0	-1.0	-0.2586

(b) Logic 1 = 4.5 and logic 0 = 1.0

($\underline{T}_{11}(0)$ $\underline{T}_{12}(0)$ $\underline{T}_{21}(0)$ $\underline{T}_{22}(0)$ $\bar{T}_{11}(0)$ $\bar{T}_{12}(0)$ $\underline{\theta}_1(0)$ $\underline{\theta}_2(0)$ $\bar{\theta}_1(0)$) = (-0.01 0.01 0.01 -0.01 -0.01 -0.01 -0.01 -0.01 0.01)

($\underline{w}_{11}(0)$ $\underline{w}_{12}(0)$ $\underline{w}_{21}(0)$ $\underline{w}_{22}(0)$ $\bar{w}_{11}(0)$ $\bar{w}_{12}(0)$ $\underline{\Theta}_1(0)$ $\underline{\Theta}_2(0)$ $\bar{\Theta}_1(0)$) = (-1.0 1.0 1.0 -1.0 -1.0 -1.0 -1.0 -1.0 1.0)

τ	1.0		4.5		4.5		1.0	
$\bar{y}$	1.008		4.429		4.429		1.184	
$\tau-\bar{y}$	-0.008		0.071		0.071		-0.184	
$\underline{T}_{11}$	$\underline{T}_{12}$	$\underline{T}_{21}$	$\underline{T}_{22}$	$\bar{T}_{11}$	$\bar{T}_{12}$	$\underline{\theta}_1$	$\underline{\theta}_2$	$\bar{\theta}_1$
3.412E-03	3.412E-03	1.905E-04	2.355E-04	3.410E-03	3.411E-03	1.017E-03	5.648E-05	-4.087E-03
$\underline{w}_{11}$	$\underline{w}_{12}$	$\underline{w}_{21}$	$\underline{w}_{22}$	$\bar{w}_{11}$	$\bar{w}_{12}$	$\underline{\Theta}_1$	$\underline{\Theta}_2$	$\bar{\Theta}_1$
0.1607	0.1610	-1.0	-1.0	-0.3122	-0.01631	-1.0	-1.0	-1.0

(c) Logic 1 = 4.5 and logic 0 = 0.5

($\underline{T}_{11}(0)$ $\underline{T}_{12}(0)$ $\underline{T}_{21}(0)$ $\underline{T}_{22}(0)$ $\bar{T}_{11}(0)$ $\bar{T}_{12}(0)$ $\underline{\theta}_1(0)$ $\underline{\theta}_2(0)$ $\bar{\theta}_1(0)$) = (0.01 -0.01 -0.01 0.01 0.01 0.01 -0.01 -0.01 0.01)

($\underline{w}_{11}(0)$ $\underline{w}_{12}(0)$ $\underline{w}_{21}(0)$ $\underline{w}_{22}(0)$ $\bar{w}_{11}(0)$ $\bar{w}_{12}(0)$ $\underline{\Theta}_1(0)$ $\underline{\Theta}_2(0)$ $\bar{\Theta}_1(0)$) = (1.0 -1.0 -1.0 1.0 1.0 1.0 -1.0 -1.0 1.0)

τ	0.5		4.5		4.5		0.5	
$\bar{y}$	0.5126		4.435		4.434		0.6764	
$\tau-\bar{y}$	-0.0126		0.065		0.066		-0.1764	
$\underline{T}_{11}$	$\underline{T}_{12}$	$\underline{T}_{21}$	$\underline{T}_{22}$	$\bar{T}_{11}$	$\bar{T}_{12}$	$\underline{\theta}_1$	$\underline{\theta}_2$	$\bar{\theta}_1$
3.412E-03	3.412E-03	2.674E-04	2.825E-04	3.410E-03	3.411E-03	1.206E-03	5.581E-05	-4.681E-03
$\underline{w}_{11}$	$\underline{w}_{12}$	$\underline{w}_{21}$	$\underline{w}_{22}$	$\bar{w}_{11}$	$\bar{w}_{12}$	$\underline{\Theta}_1$	$\underline{\Theta}_2$	$\bar{\Theta}_1$
0.1672	0.1687	-1.0	-1.0	-0.3033	-0.02531	-1.0	-1.0	-1.0

(d) Logic 1 = 4.5 and logic 0 = 0.5

($\underline{T}_{11}(0)$ $\underline{T}_{12}(0)$ $\underline{T}_{21}(0)$ $\underline{T}_{22}(0)$ $\bar{T}_{11}(0)$ $\bar{T}_{12}(0)$ $\underline{\theta}_1(0)$ $\underline{\theta}_2(0)$ $\bar{\theta}_1(0)$) = (-0.01 0.01 0.01 -0.01 -0.01 -0.01 -0.01 -0.01 0.01)

($\underline{w}_{11}(0)$ $\underline{w}_{12}(0)$ $\underline{w}_{21}(0)$ $\underline{w}_{22}(0)$ $\bar{w}_{11}(0)$ $\bar{w}_{12}(0)$ $\underline{\Theta}_1(0)$ $\underline{\Theta}_2(0)$ $\bar{\Theta}_1(0)$) = (-1.0 1.0 1.0 -1.0 -1.0 -1.0 -1.0 -1.0 1.0)

τ	0.5		4.5		4.5		0.5	
$\bar{y}$	0.4653		3.17		3.133		3.191	
$\tau-\bar{y}$	0.0347		1.33		1.367		-2.691	
$\underline{T}_{11}$	$\underline{T}_{12}$	$\underline{T}_{21}$	$\underline{T}_{22}$	$\bar{T}_{11}$	$\bar{T}_{12}$	$\underline{\theta}_1$	$\underline{\theta}_2$	$\bar{\theta}_1$
3.410E-03	3.410E-03	-3.892E-04	-4.647E-04	3.411E-03	3.411E-03	-5.265E-05	6.567E-06	3.410E-03
$\underline{w}_{11}$	$\underline{w}_{12}$	$\underline{w}_{21}$	$\underline{w}_{22}$	$\bar{w}_{11}$	$\bar{w}_{12}$	$\underline{\Theta}_1$	$\underline{\Theta}_2$	$\bar{\Theta}_1$
-0.2767	-0.2033	-1.0	-1.0	-0.01821	2.271E-03	-1.0	-1.0	-0.2844

5.4. Discussion

The modified learning rule [43] is employed in order to implement feedforward ANNs with learning in electronic VLSI/LSI circuits. Two feedforward ANNs with sequential learning circuits and two feedforward ANNs with simultaneous learning circuits have been realized and simulated using the PSPICE circuits simulator. The PSPICE simulation results show that the former circuits may not learn all the input-output patterns simultaneously but the latter ones can easily learn all the input-output patterns simultaneously.

Consider the total squared error function and the squared error function for the desired target p as follows:

$$E = \frac{1}{2} \sum_{p} E_p, \tag{5.4.1}$$

where $E_p = \frac{1}{2} \sum_{j} (t_{pj} - y_{pj})^2$.

For feedforward ANNs with simultaneous learning circuit, all the weights converge to an equilibrium point in order to achieve a minimum value of the total squared error function E. On the other hand, feedforward ANNs with sequential learning force all the weights to converge to an equilibrium point where the squared error function for the desired target p, E_p, has a minimum value. These two equilibrium points are not necessarily the same. That is, the final weights of the feedforward ANN with simultaneous learning circuit may not be a minimum for the feedforward ANN with sequential learning circuit. Therefore, feedforward ANNs with sequential learning circuit may forget or ignore the previous learned weights. This demonstrates the need for feedforward ANN with simultaneous learning circuits in the analog implementation of feedforward ANNs with learning. However, the feedforward ANNs with simultaneous learning circuit take larger chip area than the

feedforward ANNs with sequential learning circuit.

CHAPTER 6

A MODULAR DESIGN OF FEEDFORWARD ANNS WITH LEARNING

A module chip is designed in order to build a large scale feedforward ANNs with learning. The implementation of a module chip is initiated by designing two different module chips for feedforward ANNs with learning: One is for the hidden layer and the other is for the output layer. That is because the learning circuitry of the output layer has sigmoid-derivative circuits but the hidden layer does not.

Dr. Salam suggests one module chip for the use of any layer since the design of two different module chips has required more design effort than that of one module chip. Then the module chips are connected vertically in order to increase the number of neurons in any layer and horizontally for additional layers.

6.1. A Prototype 2x2x2 Feedforward ANNs with Learning

In the modified learning rule, the removal of sigmoid-derivative terms in the output layer does not affect the stability of the system. Dr. Salam suggests that another modification be taken from the modified learning rule by removing sigmoid-derivative terms in the output layer. A prototype 2x2x2 feedforward ANN with learning has been implemented without sigmoid-derivative circuits in the output layer

and simulated using the PSPICE circuit simulator in order to prove that this modified learning scheme is successfully implementable.

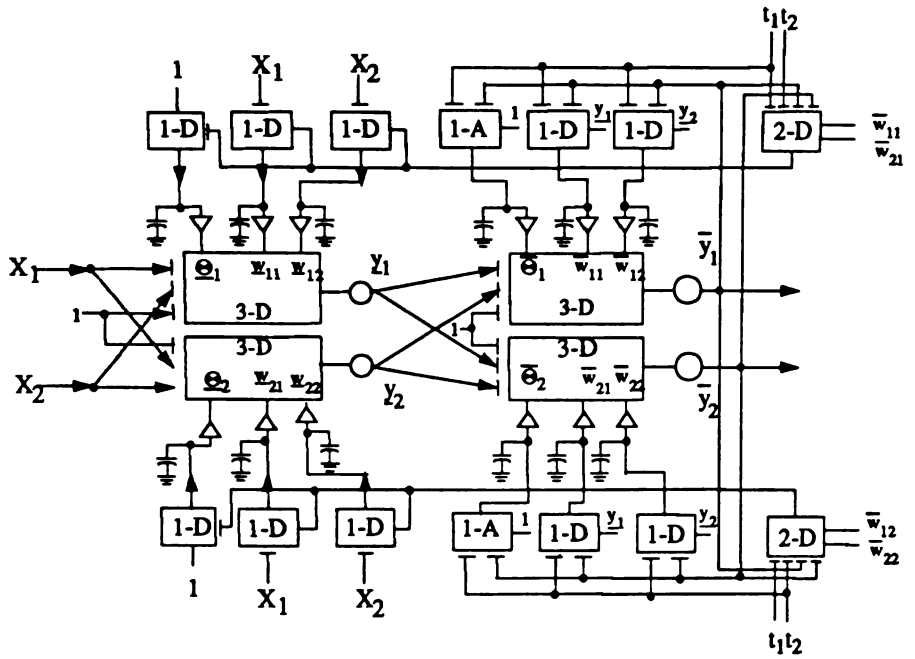


Figure 6.1.1 A 2x2x2 feedforward ANN with the sequential learning circuit #2 with threshold weight circuits and nonlinearity circuits

The feedforward ANN with the sequential learning circuit #2 in section 5.2.2 is employed to implement a prototype 2x2x2 feedforward ANN with learning, which is shown in Figure 6.1.1. There are two inputs, two outputs, and two neurons in the hidden layer. Since there are two inputs and two desired targets, all sixteen different learning tasks are available for the simulation of this ANN circuit using the PSPICE circuit simulator. The PSPICE simulation is performed along the following procedures :

1. Initialize the circuit with an initial condition, which is the same for all the tasks.
2. Each input-output pattern is applied and the PSPICE transient analysis is executed. Measure the steady state weight values and the output error, namely,

$\tau - \bar{y}$.

3. The PSPICE dc analysis is performed by setting the weights to the steady state weight values. Measure the output of each layer and the output error $\tau - \bar{y}$.

One of the PSPICE input files for Table 6.1.1(a) is shown in Appendix B.14 and one of the PSPICE input files for Table 6.1.1(c) is shown in Appendix B.15. The results are summarized in Table 6.1.1. This simulation result shows that each task has learned its own input-output pattern successfully although this ANN circuit does not include sigmoid-derivative circuits. That is, as expected, a sigmoid-derivative circuit does not affect the learning process of the implementation of feedforward ANN with learning.

Another PSPICE simulation is executed for the sequential learning of the feedforward ANN in Figure 6.1.1. Four training input-output vectors are given for this simulation. First, the first pair of input vector and its corresponding desired target vector are applied with a given initial condition. The PSPICE transient analysis is executed. The steady state weight values are measured and used as the initial condition for the next test vector. The same procedure is followed for the other three training vectors. After completing the fourth training process, the steady state weight values are used as the initial condition of the first training vector and this circuit is simulated again. This circuit then converged to the steady state weight values which are the same as those of the first simulation result of the same training pattern. Simulation of the next training pattern will result in the same steady state weight values for each case. This simulation results are tabulated in Table 6.1.2. This shows that the sequential learning appears to ignore the learning information of the previous training vector and has learned only the present training vector as we have encountered in Section 5.2.2.

Tabl

volts

($\underline{I}$

0.5 0

($\underline{w}$

-1.0

x_1
x_2
τ_1
τ_2
$\bar{y}_1$
$\bar{y}_2$
τ_1-
τ_2-
$\underline{I}$
$\underline{I}$
$\underline{I}$
$\underline{I}$
$\bar{I}$
$\bar{I}$
$\bar{I}$
$\bar{I}$
$\underline{\theta}$
$\underline{\theta}$
$\bar{\theta}$
$\bar{\theta}$
$\underline{w}$
$\underline{w}$
$\underline{w}$
$\underline{w}$
$\bar{w}$
$\bar{w}$
$\bar{w}$
$\bar{w}$
$\underline{\Theta}$
$\underline{\Theta}$
$\bar{\Theta}$
$\bar{\Theta}$

Table 6.1.1(a) Transient PSPICE simulation of a 2x2x2 feedforward ANN with the sequential learning circuit #1 (units are in volts). τ_i is a target and $\bar{y}_i$ is an actual output. The initial conditions are given by

($\underline{T}_{11}(0)$ $\underline{T}_{12}(0)$ $\underline{T}_{21}(0)$ $\underline{T}_{22}(0)$ $\bar{T}_{11}(0)$ $\bar{T}_{12}(0)$ $\bar{T}_{21}(0)$ $\bar{T}_{22}(0)$ $\underline{\theta}_1(0)$ $\underline{\theta}_2(0)$ $\bar{\theta}_1(0)$ $\bar{\theta}_2(0)$) = (0.5 -0.5 -0.5 0.5 0.5 0.5 0.5 0.5 -0.5 -0.5 0.5 0.5)

($\underline{w}_{11}(0)$ $\underline{w}_{12}(0)$ $\underline{w}_{21}(0)$ $\underline{w}_{22}(0)$ $\bar{w}_{11}(0)$ $\bar{w}_{12}(0)$ $\bar{w}_{21}(0)$ $\bar{w}_{22}(0)$ $\underline{\Theta}_1(0)$ $\underline{\Theta}_2(0)$ $\bar{\Theta}_1(0)$ $\bar{\Theta}_2(0)$) = (1.0 -1.0 -1.0 1.0 1.0 1.0 1.0 1.0 -1.0 -1.0 1.0 1.0).

	Column 1	Column 2	Column 3	Column 4	Column 5	Column 6	Column 7	Column 8
x_1	4.5	1.0	1.0	4.5	4.5	1.0	1.0	4.5
x_2	4.5	1.0	4.5	1.0	4.5	1.0	4.5	1.0
τ_1	0.5	0.5	4.5	4.5	4.5	4.5	0.5	0.5
τ_2	0.5	0.5	4.5	4.5	4.5	4.5	0.5	0.5
$\bar{y}_1$	0.5098	0.4928	4.510	4.510	4.513	4.490	0.5098	0.5098
$\bar{y}_2$	0.5098	0.4928	4.510	4.510	4.513	4.490	0.5098	0.5098
$\tau_1-\bar{y}_1$	-0.0098	0.0072	-0.01	-0.01	-0.013	0.01	-0.0098	-0.0098
$\tau_2-\bar{y}_2$	-0.0098	0.0072	-0.01	-0.01	-0.013	0.01	-0.0098	-0.0098
$\underline{T}_{11}$	5.477E-04	-2.997E-04	-3.476E-04	4.574E-04	3.411E-03	-3.490E-04	-4.178E-04	5.510E-04
$\underline{T}_{12}$	5.477E-04	-2.997E-04	4.574E-04	-3.476E-04	3.411E-03	-3.490E-04	5.510E-04	-4.178E-04
$\underline{T}_{21}$	5.477E-04	-2.997E-04	-3.476E-04	4.574E-04	3.411E-03	-3.490E-04	-4.178E-04	5.510E-04
$\underline{T}_{22}$	5.477E-04	-2.997E-04	4.574E-04	-3.476E-04	3.411E-03	-3.490E-04	5.510E-04	-4.178E-04
$\bar{T}_{11}$	3.410E-03	3.411E-03	3.413E-03	3.408E-03	6.925E-04	3.412E-03	3.411E-03	3.410E-03
$\bar{T}_{12}$	3.410E-03	3.411E-03	3.408E-03	3.413E-03	6.925E-04	3.412E-03	3.410E-03	3.411E-03
$\bar{T}_{21}$	3.410E-03	3.411E-03	3.413E-03	3.408E-03	6.925E-04	3.412E-03	3.411E-03	3.410E-03
$\bar{T}_{22}$	3.410E-03	3.411E-03	3.408E-03	3.413E-03	6.925E-04	3.412E-03	3.410E-03	3.411E-03
$\underline{\theta}_1$	5.477E-04	3.951E-04	4.574E-04	4.574E-04	3.411E-03	4.590E-04	5.510E-04	5.510E-04
$\underline{\theta}_2$	5.477E-04	3.951E-04	4.574E-04	4.574E-04	3.411E-03	4.590E-04	5.510E-04	5.510E-04
$\bar{\theta}_1$	-3.771E-03	2.752E-03	-2.716E-03	-2.716E-03	-3.412E-03	2.733E-03	-3.770E-03	-3.770E-03
$\bar{\theta}_2$	-3.771E-03	2.752E-03	-2.716E-03	-2.716E-03	-3.412E-03	2.733E-03	-3.770E-03	-3.770E-03
$\underline{w}_{11}$	-1	-1	-1	-1	-0.05046	-1	-1	-1
$\underline{w}_{12}$	-1	-1	-1	-1	-0.05046	-1	-1	-1
$\underline{w}_{21}$	-1	-1	-1	-1	-0.05046	-1	-1	-1
$\underline{w}_{22}$	-1	-1	-1	-1	-0.05046	-1	-1	-1
$\bar{w}_{11}$	-0.1396	0.1394	0.4721	-0.7921	-1	0.1664	0.03945	-0.3176
$\bar{w}_{12}$	-0.1396	0.1394	-0.7921	0.4721	-1	0.1664	-0.3176	0.03944
$\bar{w}_{21}$	-0.1396	0.1394	0.4721	-0.7921	-1	0.1664	0.03945	-0.3176
$\bar{w}_{22}$	-0.1396	0.1394	-0.7921	0.4721	-1	0.1664	-0.3176	0.03944
$\underline{\Theta}_1$	-1	-1	-1	-1	-0.05046	-1	-1	-1
$\underline{\Theta}_2$	-1	-1	-1	-1	-0.05046	-1	-1	-1
$\bar{\Theta}_1$	-1	-1	-1	-1	-1	-1	-1	-1
$\bar{\Theta}_2$	-1	-1	-1	-1	-1	-1	-1	-1

Table 6.1.1(b) Transient SPICE simulation of a 2x2x2 feedforward ANN with the sequential learning circuit #1 (units are in volts). τ_i is a target and $\bar{y}_i$ is an actual output. The initial conditions are given by

($\underline{T}_{11}(0)$ $\underline{T}_{12}(0)$ $\underline{T}_{21}(0)$ $\underline{T}_{22}(0)$ $\bar{T}_{11}(0)$ $\bar{T}_{12}(0)$ $\bar{T}_{21}(0)$ $\bar{T}_{22}(0)$ $\underline{\theta}_1(0)$ $\underline{\theta}_2(0)$ $\bar{\theta}_1(0)$ $\bar{\theta}_2(0)$) = (0.5 -0.5 -0.5 0.5 0.5 0.5 0.5 0.5 -0.5 -0.5 0.5 0.5)

($\underline{w}_{11}(0)$ $\underline{w}_{12}(0)$ $\underline{w}_{21}(0)$ $\underline{w}_{22}(0)$ $\bar{w}_{11}(0)$ $\bar{w}_{12}(0)$ $\bar{w}_{21}(0)$ $\bar{w}_{22}(0)$ $\underline{\theta}_1(0)$ $\underline{\theta}_2(0)$ $\bar{\theta}_1(0)$ $\bar{\theta}_2(0)$) = (1.0 -1.0 -1.0 1.0 1.0 1.0 1.0 1.0 -1.0 -1.0 1.0 1.0).

	Column 9	Column 10	Column 11	Column 12	Column 13	Column 14	Column 15	Column 16
x_1	4.5	1.0	1.0	4.5	4.5	1.0	1.0	4.5
x_2	4.5	1.0	4.5	1.0	4.5	1.0	4.5	1.0
τ_1	0.5	0.5	4.5	4.5	4.5	4.5	0.5	0.5
τ_2	4.5	4.5	0.5	0.5	0.5	0.5	4.5	4.5
$\bar{y}_1$	0.5098	0.5098	4.490	4.490	4.490	4.490	0.4928	0.4928
$\bar{y}_2$	4.510	4.510	0.4928	0.4928	0.4928	0.4928	4.490	4.490
$\tau_1-\bar{y}_1$	-0.098	-0.0098	0.01	0.01	0.01	0.01	0.0072	0.0072
$\tau_2-\bar{y}_2$	-0.01	-0.01	0.0072	0.0072	0.0072	0.0072	0.001	0.001
$\underline{T}_{11}$	-4.673E-01	7.776E-02	-2.591E-03	3.409E-03	3.411E-03	-1.675E-01	-6.074E-02	7.731E-02
$\underline{T}_{12}$	-4.673E-01	7.776E-02	3.409E-03	-2.589E-03	3.411E-03	-1.675E-01	8.001E-02	-5.872E-02
$\underline{T}_{21}$	-2.819E-01	1.406E-01	-2.459E-02	3.240E-02	3.240E-02	3.408E-03	-5.618E-02	9.231E-02
$\underline{T}_{22}$	-2.819E-01	1.406E-01	3.240E-02	-2.462E-02	3.240E-02	3.408E-03	7.397E-02	-7.011E-02
$\bar{T}_{11}$	3.413E-03	3.410E-03	-1.147E-03	-1.195E-03	-1.129E-03	3.414E-03	3.411E-03	3.411E-03
$\bar{T}_{12}$	3.406E-03	3.410E-03	3.411E-03	3.411E-03	3.411E-03	3.409E-03	3.411E-03	3.411E-03
$\bar{T}_{21}$	3.412E-03	3.411E-03	-1.194E-03	-1.189E-03	-1.193E-03	3.412E-03	3.412E-03	3.412E-03
$\bar{T}_{22}$	3.409E-03	3.410E-03	3.411E-03	3.411E-03	3.411E-03	3.411E-03	3.411E-03	3.412E-03
$\underline{\theta}_1$	-4.673E-01	-7.651E-02	3.409E-03	3.409E-03	3.411E-03	2.208E-01	8.001E-02	7.731E-02
$\underline{\theta}_2$	-2.819E-01	-1.382E-01	3.240E-02	3.240E-02	3.240E-02	-4.486E-03	7.397E-02	9.231E-02
$\bar{\theta}_1$	-3.763E-03	-3.757E-03	2.729E-03	2.730E-03	2.731E-03	2.732E-03	2.748E-03	2.750E-03
$\bar{\theta}_2$	-2.724E-03	-2.704E-03	2.748E-03	2.747E-03	2.747E-03	2.748E-03	2.731E-03	2.732E-03
$\underline{w}_{11}$	-1	1	-1	-0.5879	-0.05081	-1	-1	1
$\underline{w}_{12}$	-1	1	-0.5879	-1	-0.05081	-1	1	-1
$\underline{w}_{21}$	-1	1	-1	1	1	-0.6348	-1	1
$\underline{w}_{22}$	-1	1	1	-1	1	-0.6348	1	-1
$\bar{w}_{11}$	0.5867	-0.1396	-1	-1	-1	0.7427	0.1345	0.1394
$\bar{w}_{12}$	-0.8496	-0.1396	-8.946E-03	-8.933E-03	-8.947E-03	-0.4009	0.1443	0.1394
$\bar{w}_{21}$	0.1837	-0.08064	-1	-1	-1	0.2704	0.1996	0.1664
$\bar{w}_{22}$	-0.5124	-0.2517	-6.290E-02	-6.289E-02	-6.289E-02	8.942E-03	0.1332	0.1664
$\underline{\theta}_1$	-1	-1	-0.5879	-0.5879	-0.05081	1	1	1
$\underline{\theta}_2$	-1	-1	1	1	1	-1	1	1
$\bar{\theta}_1$	-1	-1	-1	-1	-1	-1	-1	-1
$\bar{\theta}_2$	-1	-1	-1	-1	-1	-1	-1	-1

Table
is an

Col	Col	Col	Col	Col	Col	Col	Col	Col	Col

Table 6.1.1(c) DC Analysis of a 2x2x2 feedforward ANN circuit with the convergent weights in Table 6.1.1(a). τ_i is a target, y_j is an output of the hidden layer, and y_i is an output of the output layer.

	x_1	x_2	τ_1	τ_2	y_1	y_2	y_1	y_2	$\tau_1 - y_1$	$\tau_2 - y_2$
Column 1	1.0	1.0	-	-	5.0	5.0	0.0	0.0	-	-
	1.0	4.5	-	-	0.0	0.0	0.5107	0.5107	-	-
	4.5	1.0	-	-	0.0	0.0	0.5107	0.5107	-	-
	4.5	4.5	0.5	0.5	0.0	0.0	0.5107	0.5107	-0.0107	-0.0107
Column 2	1.0	1.0	0.5	0.5	5.0	5.0	0.4931	0.4931	0.0069	0.0069
	1.0	4.5	-	-	0.0	0.0	0.0	0.0	-	-
	4.5	1.0	-	-	0.0	0.0	0.0	0.0	-	-
	4.5	4.5	-	-	0.0	0.0	0.0	0.0	-	-
Column 3	1.0	1.0	-	-	5.0	5.0	0.0	0.0	-	-
	1.0	4.5	4.5	4.5	0.0	0.0	4.511	4.511	-0.011	-0.011
	4.5	1.0	-	-	0.0	0.0	4.511	4.511	-	-
	4.5	4.5	-	-	0.0	0.0	4.511	4.511	-	-
Column 4	1.0	1.0	-	-	5.0	5.0	0.0	0.0	-	-
	1.0	4.5	-	-	0.0	0.0	4.511	4.511	-	-
	4.5	1.0	4.5	4.5	0.0	0.0	4.511	4.511	-0.011	-0.011
	4.5	4.5	-	-	0.0	0.0	4.511	4.511	-	-
Column 5	1.0	1.0	-	-	4.951	4.951	0.0	0.0	-	-
	1.0	4.5	-	-	4.881	4.881	0.0	0.0	-	-
	4.5	1.0	-	-	4.881	4.881	0.0	0.0	-	-
	4.5	4.5	4.5	4.5	2.091	2.091	4.511	4.511	-0.011	-0.011
Column 6	1.0	1.0	4.5	4.5	5.0	5.0	4.490	4.490	0.01	0.01
	1.0	4.5	-	-	0.0	0.0	0.0	0.0	-	-
	4.5	1.0	-	-	0.0	0.0	0.0	0.0	-	-
	4.5	4.5	-	-	0.0	0.0	0.0	0.0	-	-
Column 7	1.0	1.0	-	-	5.0	5.0	0.0	0.0	-	-
	1.0	4.5	0.5	0.5	0.0	0.0	0.5099	0.5099	-0.0099	-0.0099
	4.5	1.0	-	-	0.0	0.0	0.510	0.510	-	-
	4.5	4.5	-	-	0.0	0.0	0.5099	0.5099	-	-
Column 8	1.0	1.0	-	-	5.0	5.0	0.0	0.0	-	-
	1.0	4.5	-	-	0.0	0.0	0.5099	0.5099	-	-
	4.5	1.0	0.5	0.5	0.0	0.0	0.510	0.510	-0.01	-0.01
	4.5	4.5	-	-	0.0	0.0	0.5099	0.5099	-	-

Table 6.1.1(d) DC Analysis of a 2x2x2 feedforward ANN circuit with the convergent weights in Table 6.1.1(b). τ_i is a target, y_i is an output of the hidden layer, and $\bar{y}_i$ is an output of the output layer.

	x_1	x_2	τ_1	τ_2	y_1	y_2	$\bar{y}_1$	$\bar{y}_2$	$\tau_1 - \bar{y}_1$	$\tau_2 - \bar{y}_2$
Column 9	1.0	1.0	-	-	5.0	5.0	0.0	0.0	-	-
	1.0	4.5	-	-	0.0	0.0	0.5112	5.0	-	-
	4.5	1.0	-	-	0.0	0.0	0.5109	5.0	-	-
	4.5	4.5	0.5	4.5	0.0	0.0	0.5109	5.0	-0.0109	-0.5
Column 10	1.0	1.0	0.5	4.5	0.0	0.0	0.5107	4.511	-0.0107	-0.011
	1.0	4.5	-	-	0.05664	0.05664	0.4112	4.361	-	-
	4.5	1.0	-	-	0.05665	0.05665	0.4112	4.361	-	-
	4.5	4.5	-	-	5.0	5.0	0.0	0.0	-	-
Column 11	1.0	1.0	-	-	5.0	5.0	0.0	0.0	-	-
	1.0	4.5	4.5	0.5	1.646	5.0	4.594	0.6174	-0.094	-0.1174
	4.5	1.0	-	-	0.0	0.0	5.0	5.0	-	-
	4.5	4.5	-	-	0.0	5.0	5.0	5.0	-	-
Column 12	1.0	1.0	-	-	5.0	5.0	0.0	0.0	-	-
	1.0	4.5	-	-	0.0	0.0	5.0	5.0	-	-
	4.5	1.0	4.5	0.5	1.646	5.0	4.594	0.6174	-0.094	-0.1174
	4.5	4.5	-	-	0.0	5.0	5.0	5.0	-	-
Column 13	1.0	1.0	-	-	4.951	0.0	0.0	0.0	-	-
	1.0	4.5	-	-	4.881	5.0	0.0	0.0	-	-
	4.5	1.0	-	-	4.881	5.0	0.0	0.0	-	-
	4.5	4.5	4.5	0.5	1.661	5.0	4.498	0.5012	0.002	-0.012
Column 14	1.0	1.0	4.5	0.5	5.0	5.0	4.489	0.4931	0.011	0.0069
	1.0	4.5	-	-	5.0	0.0	5.0	0.25	-	-
	4.5	1.0	-	-	5.0	0.0	5.0	0.25	-	-
	4.5	4.5	-	-	0.0	0.0	0.0	0.0	-	-
Column 15	1.0	1.0	-	-	5.0	5.0	0.4930	4.489	-	-
	1.0	4.5	0.5	4.5	5.0	5.0	0.4931	4.489	0.0069	0.011
	4.5	1.0	-	-	0.0	0.0	0.0	0.0	-	-
	4.5	4.5	-	-	5.0	5.0	0.4920	4.488	-	-
Column 16	1.0	1.0	-	-	5.0	5.0	0.4930	4.489	-	-
	1.0	4.5	-	-	0.0	0.0	0.0	0.0	-	-
	4.5	1.0	0.5	4.5	5.0	5.0	0.4931	4.490	0.0069	0.010
	4.5	4.5	-	-	5.0	5.0	0.4920	4.488	-	-

Table 6.1.2 (a) Transient PSPICE simulation of a 2x2x2 feedforward MOS circuit with the sequential learning circuit #1 (units are in volts). τ is a target and $\bar{y}$ is an actual output. The initial conditions are given by

($\underline{T}_{11}(0)$ $\underline{T}_{12}(0)$ $\underline{T}_{21}(0)$ $\underline{T}_{22}(0)$ $\bar{T}_{11}(0)$ $\bar{T}_{12}(0)$ $\bar{T}_{21}(0)$ $\bar{T}_{22}(0)$ $\underline{\theta}_1(0)$ $\underline{\theta}_2(0)$ $\bar{\theta}_1(0)$ $\bar{\theta}_2(0)$) = (0.5 -0.5 -0.5 0.5 0.5 0.5 0.5 0.5 -0.5 -0.5 0.5 0.5)

($\underline{w}_{11}(0)$ $\underline{w}_{12}(0)$ $\underline{w}_{21}(0)$ $\underline{w}_{22}(0)$ $\bar{w}_{11}(0)$ $\bar{w}_{12}(0)$ $\bar{w}_{21}(0)$ $\bar{w}_{22}(0)$ $\underline{\Theta}_1(0)$ $\underline{\Theta}_2(0)$ $\bar{\Theta}_1(0)$ $\bar{\Theta}_2(0)$) = (1.0 -1.0 -1.0 1.0 1.0 1.0 1.0 1.0 -1.0 -1.0 1.0 1.0).

	Column #1	Column #2	Column #3	Column #4	Column #5
x_1	4.5	1.0	1.0	4.5	4.5
x_2	4.5	1.0	4.5	1.0	4.5
τ	0.5	0.5	4.5	4.5	0.5
$\bar{y}_1$	0.5098	0.4928	4.510	4.510	0.5098
$\bar{y}_2$	0.5098	0.4928	4.510	4.510	0.5098
$\tau-\bar{y}_1$	-0.0098	0.0072	-0.01	-0.01	-0.0098
$\tau-\bar{y}_2$	-0.0098	0.0072	-0.01	-0.01	-0.0098
$\underline{T}_{11}$	5.473E-04	-3.002E-04	-3.410E-04	4.575E-04	5.428E-04
$\underline{T}_{12}$	5.473E-04	-3.002E-04	4.488E-04	-3.476E-04	5.428E-04
$\underline{T}_{21}$	5.473E-04	-3.002E-04	-3.410E-04	4.575E-04	5.428E-04
$\underline{T}_{22}$	5.473E-04	-3.002E-04	4.488E-04	-3.476E-04	5.428E-04
$\bar{T}_{11}$	3.410E-03	3.411E-03	3.410E-03	3.410E-03	3.410E-03
$\bar{T}_{12}$	3.410E-03	3.411E-03	3.410E-03	3.410E-03	3.410E-03
$\bar{T}_{21}$	3.410E-03	3.411E-03	3.410E-03	3.410E-03	3.410E-03
$\bar{T}_{22}$	3.410E-03	3.411E-03	3.410E-03	3.410E-03	3.410E-03
$\underline{\theta}_1$	5.473E-04	3.951E-04	4.488E-04	4.575E-04	5.428E-04
$\underline{\theta}_2$	5.473E-04	3.951E-04	4.488E-04	4.575E-04	5.428E-04
$\bar{\theta}_1$	-3.771E-03	2.748E-03	-2.723E-03	-2.723E-03	-3.739E-03
$\bar{\theta}_2$	-3.771E-03	2.748E-03	-2.723E-03	-2.723E-03	-3.739E-03
$\underline{w}_{11}$	-1	-1	-1	-1	-1
$\underline{w}_{12}$	-1	-1	-1	-1	-1
$\underline{w}_{21}$	-1	-1	-1	-1	-1
$\underline{w}_{22}$	-1	-1	-1	-1	-1
$\bar{w}_{11}$	-0.1396	0.1394	-0.1663	-0.1663	-0.1396
$\bar{w}_{12}$	-0.1396	0.1394	-0.1663	-0.1663	-0.1396
$\bar{w}_{21}$	-0.1396	0.1394	-0.1663	-0.1663	-0.1396
$\bar{w}_{22}$	-0.1396	0.1394	-0.1663	-0.1663	-0.1396
$\underline{\Theta}_1$	-1	-1	-1	-1	-1
$\underline{\Theta}_2$	-1	-1	-1	-1	-1
$\bar{\Theta}_1$	-1	-1	-1	-1	-1
$\bar{\Theta}_2$	-1	-1	-1	-1	-1

Table 6.1.2(b) DC Analysis of a 2x2x2 feedforward ANN circuit with the convergent weights Table 6.1.2(a). τ is a target, y_j is an output of the hidden layer, and $\bar{y}$ is an output of the output layer.

	x_1	x_2	τ_1	τ_2	y_1	y_2	$\bar{y}_1$	$\bar{y}_2$	$\tau_1 - \bar{y}_1$	$\tau_2 - \bar{y}_2$
Column #1,5	1.0	1.0	-	-	5.0	5.0	0.0	0.0	-	-
	1.0	4.5	-	-	0.0	0.0	0.5107	0.5107	-	-
	4.5	1.0	-	-	0.0	0.0	0.5107	0.5107	-	-
	4.5	4.5	0.5	0.5	0.0	0.0	0.5107	0.5107	-0.0107	-0.0107
Column #2	1.0	1.0	0.5	0.5	5.0	5.0	0.4931	0.4931	0.0069	0.0069
	1.0	4.5	-	-	0.0	0.0	0.0	0.0	-	-
	4.5	1.0	-	-	0.0	0.0	0.0	0.0	-	-
	4.5	4.5	-	-	0.0	0.0	0.0	0.0	-	-
Column #3,4	1.0	1.0	-	-	5.0	5.0	0.0	0.0	-	-
	1.0	4.5	4.5	4.5	0.0	0.0	4.511	4.511	-0.011	-0.011
	4.5	1.0	4.5	4.5	0.0	0.0	4.511	4.511	-0.011	-0.011
	4.5	4.5	-	-	0.0	0.0	4.511	4.511	-	-

Another modification of the modified learning rule results in the following update law:

When the neuron k is in the output layer,

$$\dot{\bar{w}}_{kj} = \eta\ (t_k - \bar{y}_k)\ \underline{y}_j - \alpha_{ji}\bar{w}_{ji} \tag{6.1.1}$$

and when the neuron j is in any hidden layer,

$$\dot{\underline{w}}_{ji} = \eta \sum_m \bar{w}_{mj}\ (t_m - \bar{y}_m)\ y_i - \alpha_{ji}\underline{w}_{ji}, \tag{6.1.2}$$

where $\bar{y}_k$ is the output of the neuron k in the output layer, $\underline{y}_j$ is the output of the neuron j in a hidden layer, and y_i is the output of neuron i in a previous layer. If the previous layer is the input layer, then y_i is the sigmoidal output to the external input x_i.

6.2. The Implementation of the Module Chip

One module chip is designed for the use of any layer since the learning circuitry of the output layer is the as same as that of the hidden layer. The block diagram of n x m module chip is shown in Figure 6.2.1, where n is the number of inputs and m is the number of outputs. Its block representation is shown in the box depicted on the left corner at the bottom of Figure 6.2.1, where x is an n-dimensional input vector, y is an m-dimensional output vector, e is an m-dimensional error vector from the next higher layer, and $\bar{e}$ is an n-dimensional back-propagated error vector to the previous lower layer. N denotes the nonlinearity of each neuron, $w_i^r = [w_{i1} \cdots w_{in}]$, $w_i^c = [w_{1j} \cdots w_{mj}]$, n-D represents n-dimensional vector multiplier, 1-DI represents 1-dimensional multiplier with an integrating capacitor, and MUX denotes a 2x1 analog multiplexer. z_i is the output of MUX whose inputs are y_i and 0.

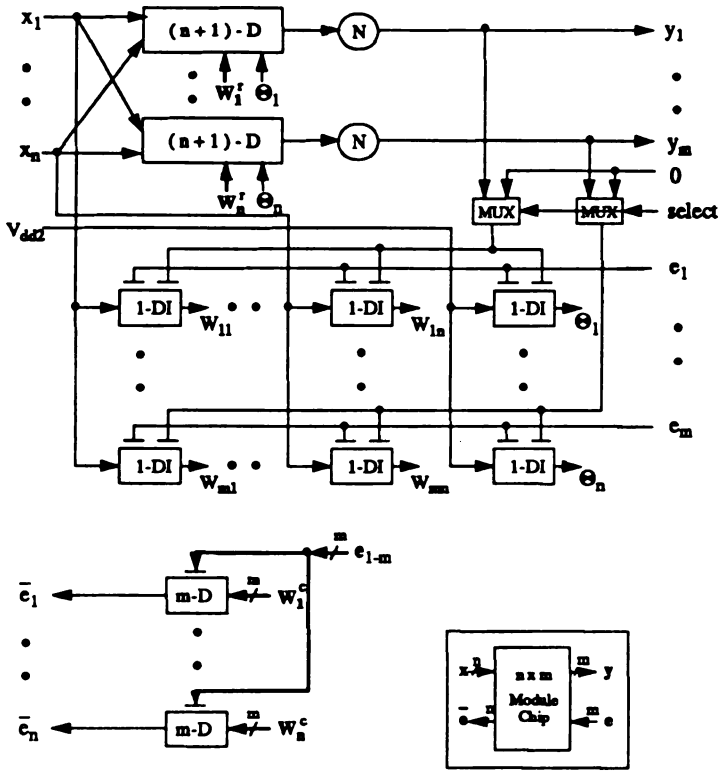


Figure 6.2.1 The block diagram of the n x m module chip

The module chip consists of two subcircuits: the feedforward subcircuit and the learning subcircuit. The feedforward subcircuit generates m outputs, y_i's, and these outputs are applied to the inputs of the next higher layer or used to modify the connection weights, w_{ij}'s, and the threshold weights, θ_i's, in the learning subcircuit. These modified weights are used to generate back-propagated error signals for the previous lower layer and also applied to the feedforward subcircuit.

The circuit implementation of the nxm module chip results in the following equations:

$$y_i = S_n(k_{n+1}(\sum_j w_{ij}x_j + \Theta_i)) \tag{6.2.1}$$

$$RC\dot{w}_{kj} = k_1 (e_k - z_k)x_j - w_{kj} \tag{6.2.2}$$

$$\bar{e}_i = k_m \sum_m w_{mi}(e_i - z_i), \tag{6.2.3}$$

where $z_i = y_i$ when a module chip is used in the output layer and $z_i = 0$ when a module chip is used in a hidden layer.

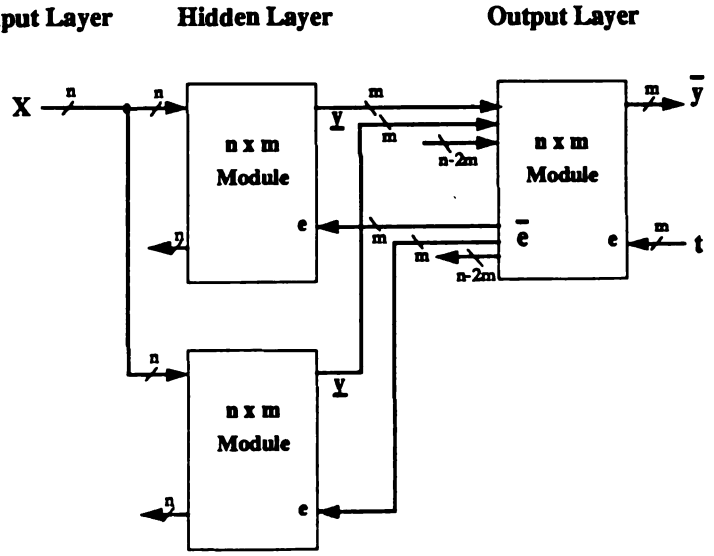


Figure 6.2.2 nx 2m x m two-layer feedforward ANNs with learning

Figure 6.2.2 illustrates a simple example of how this module chip can be used to expand vertically and horizontally. Three nxm module chips are used to implement an n x $2m$ x m two-layer feedforward ANNs with learning. Generally, if $n = km$, then an n x km x m fully-connected two-layer feedforward ANNs with learning can be implemented using $(k+1)$ nxm module chips.

The input layer is composed of just the input voltage nodes, x. The module chips in the middle form a hidden layer. The outputs of the hidden layer are applied to the inputs of the output layer. In the output layer, the desired target vector is supplied through the error vector, e. Then a back-propagated error vector is generated and applied to the error vector node, e, of the module chips in the hidden layer. n back-propagated error signals are represented by $\bar{e}_i = \sum_{j=1}^{m} w_{ji}\ (\tau_i - \bar{y}_j)$, $i = 1, ..., n$.

If there is no lower layer, then the $\bar{e}$ vector is not propagated backward any further. Otherwise, it is propagated through the error vector, e, of the module chip in that lower layer.

The module chip has four different-type signals: *i/o* signals, tuning signals, control signals, and voltage sources. There are four *i/o* signals, x, y, e, and $\bar{e}$.

There are five tuning signals, namely, V_{c_2}, V_{y_2}, V_{bw}, V_{bvf}, and V_{bcop}. V_{c_2} represents the tuning voltage for a multiplier and γ V_{c_2}'s are required for γ different number of multipliers. For example, since the 4x1 module chip consists of two different kinds of multipliers, a 5-D multiplier and 1-D multipliers, $\gamma = 2$. V_{bw} is a bias for the w_shifter, V_{bvf} is a bias for a voltage follower, and V_{bcop} is a bias for a CMOS operational amplifier. The value of these tuning signals is determined by the PSPICE simulation after (MOSIS) fabrication process of a module chip is completed and the PSPICE parameters are provided (by MOSIS).

There are three control signals: *freeze*, *select*, and *desel*. *freeze* is set to +5 for the on-chip learning mode and −5 for the off-chip learning mode. *select* is set to +5 when a module chip is used as the output layer and −5 for the hidden layer. *desel* is set to +5 to turn on the analog multiplexer and −5 for turning it off.

There are five different voltage sources (VS): +2.5 voltage source (Vdd_2), -2.5 voltage source (Vss_2), and 0 voltage source (*GND*), +5 voltage source (*Vdd*), and -5 voltage source (*Vss*).

Two different kinds of pins are reserved for interface circuitry, namely, *r−w* and *address*. *p r−w* pins are used as the paths to read or write the synaptic weights from the off-chip by selecting *address* in sequential order. These extra signals are summarized in Table 6.2.1.

Table 6.2.1 The list of the extra signals for the nxm module chip

Type	Signal	Use
tuning	V_{c_2}	tuning voltage for multipliers
	V_{w_2}	common w_2 input for multipliers
	V_{bw}	bias voltage for the w_shifter
	V_{bvf}	bias voltage for a voltage follower
	V_{bcop}	bias voltage for a CMOS op-amp
control	*freeze*	+5v (the on-chip learning), -5v (the off-chip learning)
	select	+5v (the output layer), -5v (a hidden layer)
	desel	+5v (MUX on), -5v (MUX off)
voltage source	Vdd	+5v
	Vss	-5v
	GND	0v
	Vdd2	+2.5v
	Vss2	-2.5v

The design of the module chip is critically dependent upon the number of pins of the standard chip and its chip area. Table 6.2.2 shows the four MOSIS standard chip sizes for 2 μm process. Table 6.2.3 tabulates the number of pins used for nxm module chip.

The module chip can be connected vertically and horizontally to increase the number of outputs of each layer. However, the number of inputs of each layer can not be increased for fully-connected ANN circuits. Accordingly, the module chip is designed so that n is at least three times bigger than m. Table 6.2.4 summarizes the nxm module chip vs. the MOSIS standard chip when the only limitation is given by

the number of pins.

Table 6.2.2 The four MOSIS standard chip sizes for 2 microns process

Chip	Die Size(μm)	Max. Project Size(mm)	Area Normalization	Price (Quantities)
Tiny	2540 x 2667	2.22 x 2.25	1	$500 (4)
Small	4826 x 7112	4.6 x 6.8	6.26	$2,500 (12)
Medium	7112 x 7112	6.9 x 6.8	9.39	$5,400 (24)
Large	8218 x 9779	7.9 x 9.2	14.55	$10,900 (32)

Table 6.2.3 The number of pins used for the nxm module chip

Signals	x	y	e	$\bar{e}$	tuning	control	address	r-w	V.S.	Total
# of pins	n	m	m	n	γ+4	3	a	p	5	$2n + 2m + 10 + \gamma + a + p$

Table 6.2.4 The nxm module chip vs. the MOSIS standard chip

Chip	40 Tiny	40	64	84	108	132
n x m	4 x 1	4 x 2	12 x 4	21 x 6	30 x 10	40 x 10

6.3. The Operation of the Module Chip

In order to provide flexibility for learning, this module chip is designed with two learning features: on-chip learning and off-chip learning. The on-chip learning is executed by using on-chip learning circuits which realize the modified learning rule [9]. The off-chip learning is executed by using off-line personal computer (PC).

Given any learning rule, a software program on the PC will compute the learning process and produce the appropriate weights. These final weights are represented as analog voltage values and are written onto the analog storage devices by the control of the PC. The analog storage devices we use are simply capacitors.

A capacitor takes a large chip-area when it is implemented using MOS technology (section 3.4). A MOS capacitor is designed with a small capacitance for our analog storage device. However, a small capacitance MOS capacitor is not a good analog storage device since it loses its stored information due to its leakage currents. Therefore, each stored weight will need to be refreshed regularly in order not to dissipate its stored value. In order to decide the refreshing interval, first, we have to specify the allowable loss rate of the weight information, say 1% or 4%. Then determine the decay rate, or specifically decay time, of 1% or 4% of the desired weight, called t_{decay}. We need to refresh each weight with a time interval shorter than the decay time, t_{decay}.

Large scale ANNs have a huge number of weights. We have to specify how many weights can be refreshed within the allowable refreshing time. By assuming appropriate refreshing time, t_{rf}, we can estimate the number of weights to be refreshed within the allowable fall time it takes the voltage level stored in the capacitor to decrease by about 1% to 4% from its original rate t_{decay}. The block diagram of the interfacing circuitry is shown in Figure 6.3.1. There are some interfacing elements to generate time delay such as digital-analog-converters (DAC), multiplexers, sample-and-hold (S/H) circuits, and so on. Also there are on-chip time-delay elements such as analog multiplexer, CMOS switches, and so on. Let's denote the critical time delay as t_{cd}. Then the maximum number of weights can be approximated by

$$(\#\ of\ weights)_{\max} = \frac{t_{decay}}{t_{rf}},$$

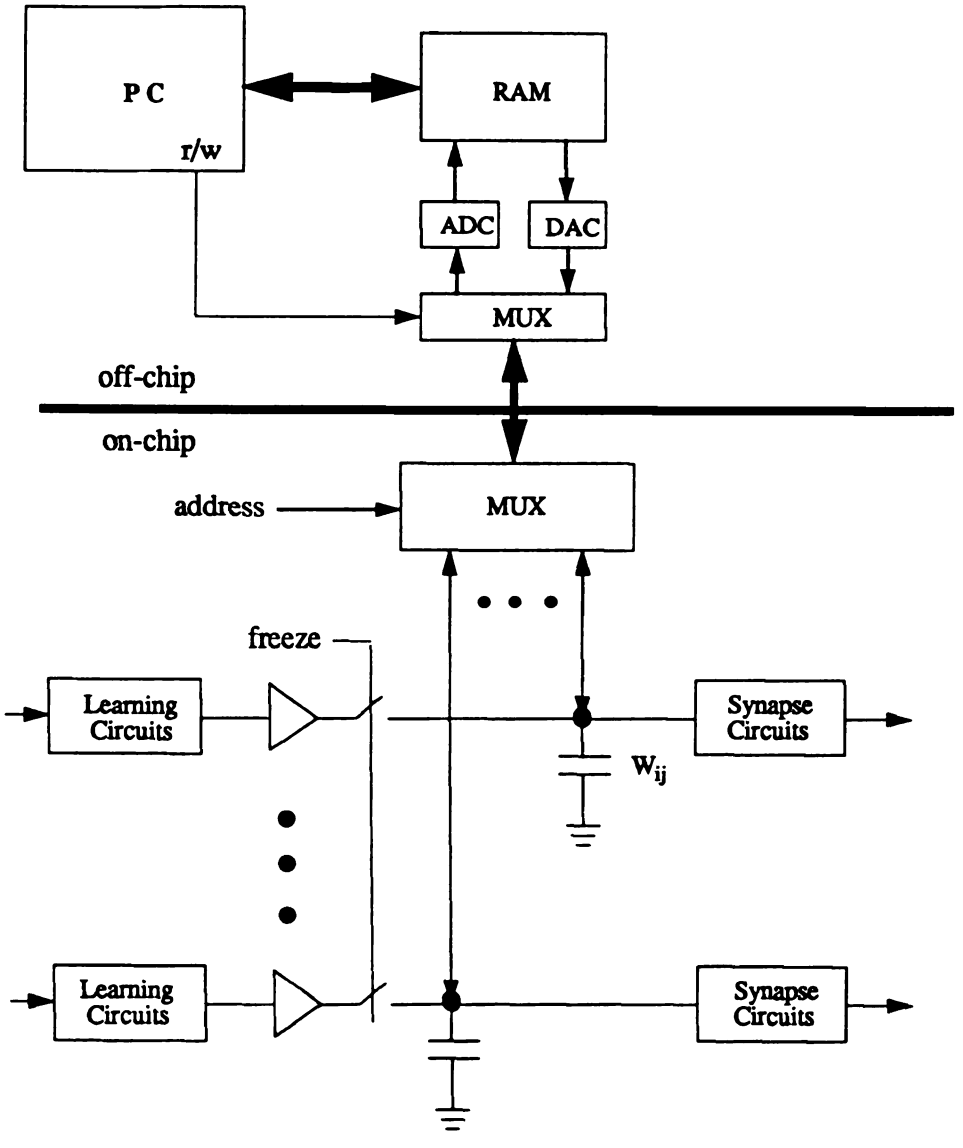


Figure 6.3.1 The block diagram of interface circuitry

where $t_{rf} > t_{cd}$. We can refresh $(\#\ of\ weights)_{max}$ weights within the allowable loss of weight information.

Under the control of a PC, the operation of the module chip proceeds as follows:

To initiate the operation of the module chip,

1. Set all the tuning signals properly: V_{c_2}'s, V_{w_2}, V_{bw}, V_{bvf}, and V_{bcop}.
2. *select* is set to +5v for the use of the output layer and set to -5v for the use of the hidden layer.
3. Decide or specify the refreshing time.

To perform the learning process using the on-chip learning circuits,

1. Set *freeze* = 5v and *desel* = -5v (the on-chip learning mode).
2. Apply an input vector x and its corresponding target vector t.
3. After convergence, set *desel* = 5v (the analog MUX is on).
4. Read the steady state values by setting *address* in sequential order through the interface circuitry and store them in the off-chip digital memory.

To refresh or write each weight which is achieved from the on-chip learning process or from the off-chip learning process,

1. Set *freeze* = -5v and *desel* = +5v (the off-chip learning mode).
2. Write each weight by setting *address* in sequential order through the interfacing circuitry.
3. Refresh each weight every refreshing interval.

6.4. Implementation of Module Chips for Large Scale Feedforward ANNs with Learning

A modular design has been described for a large scale implementation of feedforward ANNs with learning. Two implementations are designed using the MAGIC VLSI editor : 4x1 module chip on a 40-pin MOSIS Tinychip and 9x3 module chip on a 64-pin MOSIS small chip. The module chips can be connected vertically and horizontally to realize a large scale feedforward ANNs with optionally using on-chip learning circuit or off-chip learning capability.

6.4.1. The 4x1 Module Chip on a 40-pin TINYCHIP

A 4x1 module chip is designed on a 40-pin Tinychip with analog pads and has been sent for fabrication via MOSIS using 2 μm CMOS n-well process. There are 612 MOS transistors on the 2220x2250 μm^2 project area. Figure 6.4.1 shows the layout on the 40-pin Tinychip pad frame. Since the chip area of the Tinychip is small (see Table 6.1(b) in Section 6.0), the value of n and m are limited by the chip area and not by the number of pins of the chip.

The 4x1 module chip contains four inputs, one output, one error signal, and four back-propagation errors. There are five weights : four synaptic weights and one threshold weight. These weights can be read and written through one $r-w$ pin using a three-bit addressing analog multiplexer which is controlled by a PC. The refreshing time can be chosen up to 1 msec for 1% loss of weight value because of small number of weights to refresh.

The 40-pin Tinychip consists of 34 I/O pins available for a designer and 6 pins for Vdd and Vss, which are fixed by MOSIS. 26 out of the 34 pins are used for the

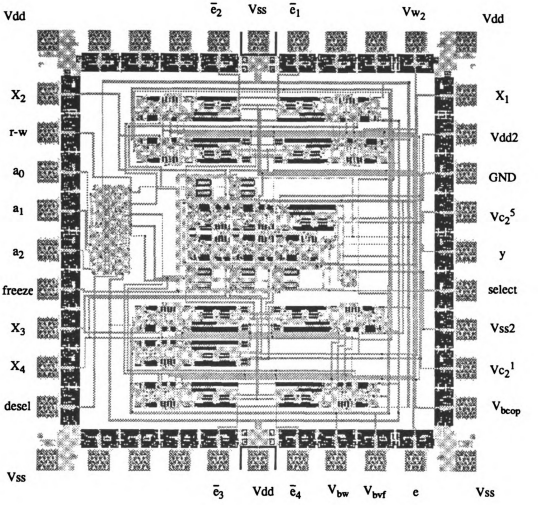


Figure 6.4.1 The 4x1 module chip on a 40-pin Tinychip

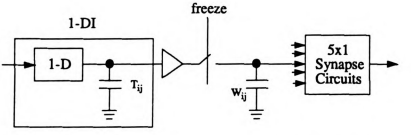


Figure 6.4.2 The circuit for each weight of the 4x1 module chip

4x1 module chip. Table 6.4.1 summarizes the number of pins used. Table 6.4.2 lists what kind of and how many cells are used for the 4x1 module chip, where 1-DI is the 1-dimensional multiplier with an integrating capacitor, 1-D is the 1-dimensional multiplier, and DI is a double inverter for a nonlinearity and a cell body. Figure 6.4.2 shows the circuit for each weight of the 4x1 module chip. Table 6.4.3 tabulates the pin-assignment of the 4x1 module chip.

The operation of the 4x1 module chip, with the proper setting of tuning signals, is as follows:

To perform the learning process using the on-chip learning circuits,

1. Set *freeze* = 5v and *desel* = -5v (the on-chip learning mode).
2. Apply an input vector x and its corresponding target t.
3. After convergence, *desel* = 5v and read the steady state values by setting the address ($a_2\ a_1\ a_0$) in sequential order.

To refresh or write each weight (which is achieved from the on-chip learning process or the off-chip learning process),

1. Set freeze = -5v and desel = +5v (the off-chip learning mode).
2. Write each weight by setting the address ($a_2\ a_1\ a_0$) in sequential order.
3. Refresh each weight every refreshing interval.

Table 6.4.1 The number of pins used for the 4x1 module chip

Signal	x	y	e	$\bar{e}$	tuning	control	address	$r-w$	V.S.	Total
# of pins	4	1	1	4	6	3	3	1	3	26

Table 6.4.2 The cells used for the 4x1 module chip

Cells	5-D	1-DI	1-D	DI	1x8 MUX	1x2 MUX
# of cells	1	5	4	6	1	1

Table 6.4.3 Pin-assignment of the 4x1 module chip

Pin #	1	2	3	4	5	6	7	8	9	10
Signal	$V_{c_2}^{5}$	GND	Vdd2	X_1	Vdd	V_{w_2}	*	*	$\bar{e}_1$	Vss
Pin #	11	12	13	14	15	16	17	18	19	20
Signal	$\bar{e}_2$	*	*	*	Vdd	X_2	r-w	a_0	a_1	a_2
Pin #	21	22	23	24	25	26	27	28	29	30
Signal	freeze	X_3	X_4	desel	Vss	*	*	*	$\bar{e}_3$	Vdd
Pin #	31	32	33	34	35	36	37	38	39	40
Signal	$\bar{e}_4$	V_{bw}	V_{bwf}	e	Vss	V_{bcap}	$V_{c_2}^{1}$	Vss2	select	y

(* means "do not use.")

6.4.2. The 9x3 Module Chip on a 64-pin MOSIS Small Standard Chip

A 9x3 module chip is designed on a 64-pin MOSIS small standard chip with analog pads using 2 μm CMOS technology. The project area of the MOSIS small standard chip is 4600x6800 μm^2, which is 6.26 times larger than that of the Tinychip. The 9x3 module chip is designed with 2,973 MOS transistors for the purpose of future fabrication.

The 9x3 module chip contains nine inputs, three outputs, three error signals, and nine back-propagation errors. There are thirty weights: twenty-seven synaptic weights and three threshold weights. These weights can be read or written through two pins using four-bit addresses by the control of a personal computer. The refreshing time can be chosen up to 0.2 msec for 1% loss of weight value because of the relatively small number of weights to be refreshed.

The layout of this chip is shown in Figure 6.4.3. 44 out of the 64 pins are used for the 9x3 module chip. Table 6.4.4 summarizes the number of pins used and Table 6.4.5 lists what kind of and how many cells are used for the 9x3 module chip. Table 6.4.6 tabulates the pin-assignment of this module chip.

The 9x3 module chip has two different circuits from 4x1 module chip. One difference is that *desel* is not used for the enable signal of the analog multiplexer in order to reduce the chip-area and one pin. Instead (a_3 a_2 a_1 a_0) is set to (5v 5v 5v 5v) to turn off the analog multiplexer, i.e., when this address is selected, the other outputs of the analog multiplexer are off and the corresponding output is on but not connected (or used). Accordingly, this address plays a role of *desel*.

The other is that only one capacitor is used for each weight while the 4x1 module chip has two capacitors for each weight (See Figure 6.4.4 and Figure 6.4.2). The PSPICE simulation has been performed by modifying a 2x2x2 feedforward ANN

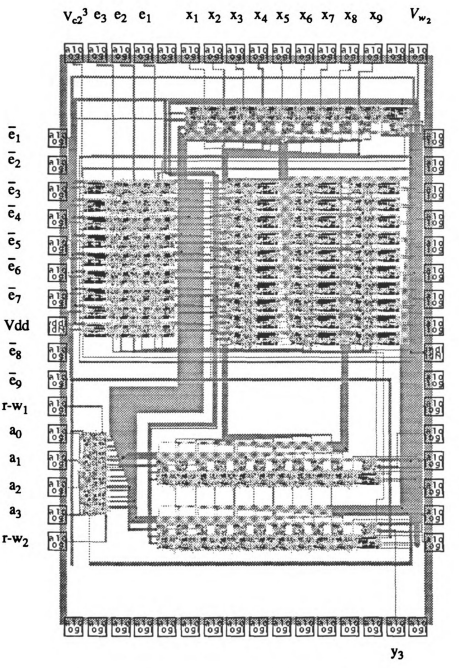


Figure 6.4.3 The 9x3 module chip on a 64-pin MOSIS small pad frame

Table 6.4.4 The number of pins used for the 9x3 module chip

Signal	x	y	e	$\bar{e}$	tuning	control	address	$r-w$	V.S.	Total
# of pins	9	3	3	9	7	2	4	2	5	44

Table 6.4.5 The cells used for the 9x3 module chip

Cells	10-D	1-D	3-D	DI	2x2[4] MUX	1x2 MUX
# of cells	3	30	9	33	1	3

Table 6.4.6 Pin-assignment of the 9x3 module chip

Pin #	1	2	3	4	5	6	7	8	9	10
Signal	*	-2.5v	GND	V_{bcop}	V_{bw}	V_{brf}	$V_{c_2}^{1}$	y_1	V_{w_2}	*
Pin #	11	12	13	14	15	16	17	18	19	20
Signal	x_9	x_8	x_7	x_6	x_5	x_4	x_3	x_2	x_1	*
Pin #	21	22	23	24	25	26	27	28	29	30
Signal	e_1	e_2	e_3	$V_{c_2}^{3}$	$\bar{e}_1$	$\bar{e}_2$	e_3	$V_{c_2}^{3}$	$\bar{e}_1$	$\bar{e}_2$
Pin #	31	32	33	34	35	36	37	38	39	40
Signal	$\bar{e}_3$	$\bar{e}_4$	$\bar{e}_5$	$\bar{e}_6$	$\bar{e}_7$	Vdd	$\bar{e}_8$	$\bar{e}_9$	$rw1$	a_0
Pin #	41	42	43	44	45	46	47	48	49	50
Signal	*	*	*	*	*	*	*	*	*	*
Pin #	51	52	53	54	55	56	57	58	59	60
Signal	*	*	*	*	y_3	*	+2.5v	*select*	*freeze*	$V_{c_2}^{10}$
Pin #	61	62	63	64	-	-	-	-	-	-
Signal	y_2	*	*	Vss	-	-	-	-	-	-

(* means "do not use.")

with learning in Figure 6.1.1 with the circuit in Figure 6.4.4. The results show that the circuit of the 9x3 module chip converges to the same weight values as the circuit of the 4x1 module chip does.

The operation of the 9x3 module chip (with the proper setting of tuning signals) is as follows:

To perform the learning process using the on-chip learning circuits,

1. Set freeze = 5v and, set (a_3 a_2 a_1 a_0)=(5v 5v 5v 5v) (the on-chip learning mode).
2. Apply an input vector x and its corresponding target t.
3. After convergence, set *freeze* = 5v and read the steady state values by setting (a_3 a_2 a_1 a_0) in sequential addressing order.

To refresh or write each weight (which is achieved from the on-chip learning process or the off-chip learning process),

1. Set *freeze* = -5v (the off-chip learning mode).
2. Write each weight by setting (a_3 a_2 a_1 a_0) in sequential addressing order.
3. Refresh each weight every refreshing interval.

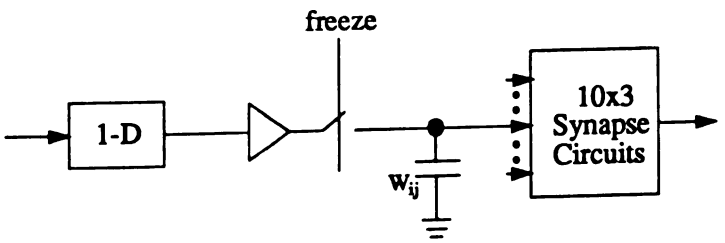


Figure 6.4.4 The circuit for each weight of the 9x3 module chip

CHAPTER 7

SUMMARY

A feedforward neural network with a supervised learning rule is implemented using standard CMOS technology. In order to implement a programmable synapse cell, we employ a simple tunable all-MOS analog multiplier, voltage shifters/attenuators, voltage followers, and MOS capacitors.

This multiplier uses voltage signals for its inputs and its output and executes the multiplication between analog inputs and analog synaptic weights. Employing this multiplier, we can implement large scale ANNs with learning using fewer MOS transistors than are required by implementations using the so-called Gilbert multiplier. Voltage shifters/attenuators adjust applied input voltages to meet the input operating range of the multiplier, and they increase the input operating range of the multiplier cell.

In order to test the performance of this scalar/vector analog multiplier, an 11-dimensional vector multiplier is designed on a 40-pin MOSIS Tinychip with analog pads, and it has been sent for fabrication via MOSIS using 2 μm CMOS n-well process on Oct. 3, 1990. After receiving this chip, scalar to a maximum of an 11-dimensional vector multiplier will be tested and compared to the results obtained from the PSPICE simulation.

A modified error back-propagation learning rule [43] is used in order to realize on-chip learning circuits using standard CMOS technology. Using this modified learning rule, sigmoid-derivative circuits are not necessarily required for the implementation of feedforward ANNs with learning.

A prototype 2x2x1 feedforward ANN with learning has been implemented and simulated using the circuit simulator in order to prove that the learning scheme is successfully implementable. The PSPICE simulation has been done by applying several test vectors which are formed of a given input vector and its corresponding desired pattern. During the learning process the connection weights converge to their steady states that achieve the learning of the given pattern [29].

Two 2x2x1 feedforward ANNs with sequential learning circuits and two 2x2x1 feedforward ANN with simultaneous learning circuits are implemented and simulated using the PSPICE circuit simulator. The results show that the former circuits may not learn all the desired targets concurrently but that the latter can easily learn all the desired targets. That is, a feedforward ANN with simultaneous learning circuit has achieved all the weights, which are an equilibrium point, in order to achieve an acceptable minimum value of the total squared error function, E. On the other hand, a feedforward ANN with sequential learning circuit forces all the weights to converge to an equilibrium point where only the squared error function for the desired target p, E_p, has an acceptable minimum value. The two equilibrium points obtained are expected to be different. That is, the final weights of the feedforward ANN with simultaneous learning circuit may not be a minimum for the feedforward ANN with sequential learning circuit. Therefore, feedforward ANNs with sequential learning circuits may ignore the previous learned input-output patterns.

A modular design is described for a large-scale implementation of feedforward ANNs with learning. Two implementations are designed using the MAGIC VLSI

editor : a 4x1 module chip on the MOSIS Tinychip and a 9x3 module chip on a MOSIS small standard chip. The module chips can be connected vertically and horizontally to realize a large scale feedforward ANNs with learning.

In order to provide flexibility in the learning process, this module chip is designed with two learning features : on-chip learning and off-chip learning. The on-chip learning is executed by using on-chip learning circuits which realize the modified learning rule [43]. The off-chip learning is executed by using any choice of a learning rule as a software on a personal computer (PC).

The modified weights are represented as analog voltages and are written onto the analog storage devices by the control of the PC. A MOS capacitor is employed as a simple analog storage device over a short interval of time. Due to leakage currents, however, the MOS capacitors are required to be refreshed regularly in order not to lose the stored weight values. This refreshing process is executed periodically through interface circuitry under the control of the PC.

A 4x1 module chip is designed on a 40-pin MOSIS Tinychip with analog pads using the MAGIC VLSI editor and was sent for fabrication via MOSIS using 2 μm CMOS n-well process on Oct. 31, 1990. After receiving the 4x1 module chip, it will be tested by forming two-layer feedforward ANNs with learning. If the test results are successful, then the 9x3 module chip design will be finalized on a 64-pin MOSIS small standard chip with analog pads and sent for fabrication.

APPENDICES

APPENDIX A

SPICE PARAMETERS

APPENDIX A.1

The SPICE parameters from MOSIS 3 micron CMOS process on June 1987.

	NMOS TRANSISTOR	PMOS TRANSISTOR
LEVEL	2	2
TPG		-1
LD	280.000000E-09	280.000000E-09
VTO	.827125	-.894654
KP	32.866490E-06	15.264520E-06
GAMMA	1.3596	.879003
PHI	.6	.6
LAMBDA	.01605	.047087
RSH	25	95
CJ	320.000000E-06	200.000000E-06
CJSW	900.000000E-12	450.000000E-12
MJ	0.5	0.5
MJSW	0.33	0.33
CGSO	520.000000E-12	400.000000E-12
CGDO	520.000000E-12	400.000000E-12
NSUB	10.000000E+15	112.108800E+12
NSS	0	0
NFS	1.234795E+12	878.861700E+09
TOX	50.000000E-09	50.000000E-09
XJ	400.000000E-09	400.000000E-09
UO	200	100
UCRIT	999.000000E+03	16.376500E+03
UEXP	1.001000E-03	.153441
VMAX	100.000000E+03	100.000000E+03
NEFF	.01001	.01001
DELTA	1.2405	1.93831

APPENDIX A.2

The SPICE parameters from MOSIS 2 micron CMOS process on June 1989

	NMOS TRANSISTOR	PMOS TRANSISTOR
LEVEL	2	2
TPG		-1
LD	212.340000E-09	250.000000E-09
VTO	.783736	-.807
KP	54.600000E-06	21.300000E-06
GAMMA	.5262	.5644
PHI	.6	.6
LAMBDA	.035333	.056595
RSH	33.4	121.6
CJ	113.400000E-06	254.000000E-06
CJSW	477.000000E-12	331.000000E-12
MJ	.708	.553
MJSW	.253	.352
CGSO	266.956500E-12	314.303100E-12
CGDO	266.956500E-12	314.303100E-12
CGBO	425.025500E-12	457.437700E-12
NSUB	5.860000E+15	6.740000E+15
NSS	1.000000E+12	1.000000E+12
NFS	954.427000E+09	100.000000E+09
TOX	41.200000E-09	41.200000E-09
XJ	150.000000E-09	50.000000E-09
UO	651	253.997
UCRIT	30.664600E+03	16.929200E+03
UEXP	.177364	.2458
VMAX	57.874100E+03	37.028200E+03
NEFF		1.001
DELTA	1.99612	1.001368E-06

APPENDIX A.3

The SPICE parameters from MOSIS 2 micron CMOS process on Oct. 3, 1990

	NMOS TRANSISTOR	PMOS TRANSISTOR
LEVEL	2	2
TPG		-1
LD	250.000000E-09	209.610000E-09
VTO	.771327	-.78821
KP	56.060000E-06	22.300000E-06
GAMMA	.53	.5486
PHI	.6	.6
LAMBDA	.030894	.049817
RSH	29.6	90.7
CJ	113.650000E-06	251.360000E-06
CJSW	531.870000E-12	294.470000E-12
MJ	.6862	.5471
MJSW	.2651	.3188
CGSO	322.924900E-12	270.753200E-12
CGDO	322.924900E-12	270.753200E-12
CGBO	672.642900E-12	583.962500E-12
NSUB	6.257042E+15	6.722092E+15
NSS	10.000000E+09	10.000000E+09
NFS	4.834708E+12	100.000000E+09
TOX	40.100000E-09	40.100000E-09
XJ	250.000000E-09	250.000000E-09
UO	650.996	259
UCRIT	41.931500E+03	12.594500E+03
UEXP	.153507	.182346
VMAX	67.660800E+03	41.222600E+03
NEFF		1.001
DELTA	.472546	.844423

APPENDIX B

PSPICE INPUT FILES

APPENDIX B.1

PSPICE input file for the frequency response of the CMOS op-amp

```
.MODEL N NMOS LEVEL=2.00000 LD=0.212340U TOX=412.000E-10
+NSUB=5.860000E+15 VTO=0.783736 KP=5.460000E-05
+GAMMA=0.5262 PHI=0.600000 UO=651.000 UEXP=0.177364
+UCRIT=30664.6 DELTA=1.99612 VMAX=57874.1 XJ=0.1500000U
+LAMBDA=3.533329E-02 NFS=9.544270E+11 NEFF=1
+NSS=1.000000E+12 TPG=1.00000 RSH=33.400 CGSO=2.669565E-10
+CGDO=2.669565E-10 CGBO=4.250255E-10 CJ=1.13400E-4
+MJ=0.70800 CJSW=4.77000E-10 MJSW=0.253000 PB=0.80000

.MODEL P PMOS LEVEL=2.00000 LD=0.250000U TOX=412.000E-10
+NSUB=6.740E+15 VTO=-0.807 KP=2.13000E-05
+GAMMA=0.5644 PHI=0.60 UO=253.997 UEXP=0.2458
+UCRIT=16929.2 DELTA=1.001368E-06 VMAX=37028.2 XJ=0.0500U
+LAMBDA=5.659491E-02 NFS=1.000E+11 NEFF=1.001
+NSS=1.000000E+12 TPG=-1.00000 RSH=121.600 CGSO=3.143031E-10
+CGDO=3.143031E-10 CGBO=4.574377E-10 CJ=2.5400E-4 MJ=0.55300
+CJSW=3.3100E-10 MJSW=0.35200 PB=0.80

V6 6 0 0.0
V5 5 0 0 AC 1.0
VBIAS 70 0 3.2
VDD 90 0  5.0
VSS 80 0 -5.0

M1 2 6 1 90 P L=4U W=16U
M2 3 5 1 90 P L=4U W=16U
M3 2 2 80 80 N L=16U W=4U
M4 3 2 80 80 N L=16U W=4U
M5 1 70 90 90 P L=10U W=4U
M6 8 3 80 80 N L=4U W=240U
M7 8 4 90 90 P L=4U W=200U
M8 4 2 80 80 N L=4U W=4U
M9 4 4 90 90 P L=4U W=4U
MC1 8 3 8 90 P L=15U W=120U

.AC DEC 25 1 100MEG
.OPTION LIMPTS=10000
.PROBE
.END
```

APPENDIX B.2

PSPICE input file for the 1-D multiplier cell

```
VY1  1  0
VY2  2  0   0.0
VX1  6  0
EX2  7  0  0 6 1
VC2  60 0  4.959

XVW1   1 2 3 5 VSW
XVF1   3 4 4   COP1
XVF2   5 55 55 COP1
XVX1   6 7 8 9 VSWX
XMUL1  4 55 8 9 11 12 10 60 MUL1

.SUBCKT VSW 1 3 20 40
VSS 80  0 -5.0
VDD 90  0  5.0
V70 70  0  3.0

M1  90 1  2  80 N L=22U W=11U
M3  2  1  80 80 N L=4U W=10U
M5  80 2  20 90 P L=4U W=4U
M7  20 70 90 90 P L=4U W=4U
M2  90 3  4  80 N L=22U W=11U
M4  4  3  80 80 N L=4U W=10U
M6  80 4  40 90 P L=4U W=4U
M8  40 70 90 90 P L=4U W=4U
.ENDS VSW

.SUBCKT VSWX 1 3 2 4
VSS 80  0 -5.0
VDD 90  0  5.0

M1  80 1  2  90 P L=16U W=8U
M3  2  1  90 90 P L=4U W=20U
M2  80 3  4  90 P L=16U W=8U
M4  4  3  90 90 P L=4U W=20U
.ENDS VSWX

.SUBCKT MUL1 11 12 13 14 5 6 8 60
VSS 80 0 -5.0
VC1 50 0  5.0

MG1X 11 14 6 80 N L=24U W=4U
MD1X 12 13 6 80 N L=24U W=4U
MG2X 12 14 5 80 N L=24U W=4U
MD2X 11 13 5 80 N L=24U W=4U
MR61 6  60 0 80 N L=48U W=4U
MR62 6  50 8 80 N L=48U W=4U
MR51 5  60 8 80 N L=48U W=4U
MR52 5  50 0 80 N L=48U W=4U
XCOP1 5 6 8 COP
.ENDS MUL1
```

```
.SUBCKT COP1 5 6 8
VBIAS 70 0 3.2
VDD 90 0  5.0
VSS 80 0 -5.0

M1 2 6 1 90 P L=4U W=8U
M2 3 5 1 90 P L=4U W=8U
M3 2 2 80 80 N L=16U W=4U
M4 3 2 80 80 N L=16U W=4U
M5 1 70 90 90 P L=16U W=4U
M6 8 3 80 80 N L=4U W=60U
M7 8 4 90 90 P L=4U W=47U
M8 4 2 80 80 N L=4U W=4U
M9 4 4 90 90 P L=4U W=4U
MC1 8 3 8 90 P L=20U W=10U
.ENDS COP1

.SUBCKT COP 5 6 8
VBIAS 70 0 3.2
VDD 90 0  5.0
VSS 80 0 -5.0

M1 2 6 1 90 P L=4U W=16U
M2 3 5 1 90 P L=4U W=16U
M3 2 2 80 80 N L=16U W=4U
M4 3 2 80 80 N L=16U W=4U
M5 1 70 90 90 P L=10U W=4U
M6 8 3 80 80 N L=4U W=240U
M7 8 4 90 90 P L=4U W=200U
M8 4 2 80 80 N L=4U W=4U
M9 4 4 90 90 P L=4U W=4U
MC1 8 3 8 90 P L=40U W=40U
.ENDS COP

.MODEL N NMOS LEVEL=2.00000 LD=0.212340U TOX=412.000E-10
+NSUB=5.860000E+15 VTO=0.783736 KP=5.460000E-05
+GAMMA=0.5262 PHI=0.600000 UO=651.000 UEXP=0.177364
+UCRIT=30664.6 DELTA=1.99612 VMAX=57874.1 XJ=0.1500000U
+LAMBDA=3.533329E-02 NFS=9.544270E+11 NEFF=1
+NSS=1.000000E+12 TPG=1.00000 RSH=33.400 CGSO=2.669565E-10
+CGDO=2.669565E-10 CGBO=4.250255E-10 CJ=1.13400E-4
+MJ=0.70800 CJSW=4.77000E-10 MJSW=0.253000 PB=0.80000

.MODEL P PMOS LEVEL=2.00000 LD=0.250000U TOX=412.000E-10
+NSUB=6.740E+15 VTO=-0.807 KP=2.13000E-05
+GAMMA=0.5644 PHI=0.60 UO=253.997 UEXP=0.2458
+UCRIT=16929.2 DELTA=1.001368E-06 VMAX=37028.2 XJ=0.0500U
+LAMBDA=5.659491E-02 NFS=1.000E+11 NEFF=1.001
+NSS=1.000000E+12 TPG=-1.00000 RSH=121.600 CGSO=3.143031E-10
+CGDO=3.143031E-10 CGBO=4.574377E-10 CJ=2.5400E-4 MJ=0.55300
+CJSW=3.3100E-10 MJSW=0.35200 PB=0.80

.PROBE
.DC VY1 -2.5  2.5  0.1 VX1 -2.5 2.5 2.5
.OPTIONS LIMPTS=100000
.PRINT DC V(3) V(4) V(4,55) V(10)
.END
```

APPENDIX B.3

PSPICE input file for the X-nonlinearity of the 1-D cell

```
VY1  1  0   2.5
VY2  2  0   0.0
VX1  6  0   PWL(0 -2.5 10M 2.5 20M -2.5 30M 2.5 40M -2.5 50M 2.5)
EX2  7  0   0 6 1
VC2  60 0   4.959

XVW1   1 2 3 5 VSW
XVF1   3 4 4   COP1
XVF2   5 55 55 COP1
XVX1   6 7 8 9 VSWX
XMUL1  4 55 8 9 11 12 10 60 MUL1

.SUBCKT VSW 1 3 20 40
VSS 80  0 -5.0
VDD 90  0  5.0
V70 70  0  3.0

M1  90 1  2  80 N L=22U W=11U
M3  2  1  80 80 N L=4U W=10U
M5  80 2  20 90 P L=4U W=4U
M7  20 70 90 90 P L=4U W=4U
M2  90 3  4  80 N L=22U W=11U
M4  4  3  80 80 N L=4U W=10U
M6  80 4  40 90 P L=4U W=4U
M8  40 70 90 90 P L=4U W=4U
.ENDS VSW

.SUBCKT VSWX 1 3 2 4
VSS 80  0 -5.0
VDD 90  0  5.0

M1  80 1  2  90 P L=16U W=8U
M3  2  1  90 90 P L=4U W=20U
M2  80 3  4  90 P L=16U W=8U
M4  4  3  90 90 P L=4U W=20U
.ENDS VSWX

.SUBCKT MUL1 11 12 13 14 5 6 8 60
VSS 80 0 -5.0
VC1 50 0  5.0

MG1X 11 14 6 80 N L=24U W=4U
MD1X 12 13 6 80 N L=24U W=4U
MG2X 12 14 5 80 N L=24U W=4U
MD2X 11 13 5 80 N L=24U W=4U
MR61 6  60 0 80 N L=48U W=4U
MR62 6  50 8 80 N L=48U W=4U
MR51 5  60 8 80 N L=48U W=4U
MR52 5  50 0 80 N L=48U W=4U
XCOP1 5 6 8 COP
.ENDS MUL1
```

```
.SUBCKT COP1 5 6 8
VBIAS 70 0 3.2
VDD 90 0  5.0
VSS 80 0 -5.0

M1 2 6 1 90 P L=4U W=8U
M2 3 5 1 90 P L=4U W=8U
M3 2 2 80 80 N L=16U W=4U
M4 3 2 80 80 N L=16U W=4U
M5 1 70 90 90 P L=16U W=4U
M6 8 3 80 80 N L=4U W=60U
M7 8 4 90 90 P L=4U W=47U
M8 4 2 80 80 N L=4U W=4U
M9 4 4 90 90 P L=4U W=4U
MC1 8 3 8 90 P L=20U W=10U
.ENDS COP1

.SUBCKT COP 5 6 8
VBIAS 70 0 3.2
VDD 90 0  5.0
VSS 80 0 -5.0

M1 2 6 1 90 P L=4U W=16U
M2 3 5 1 90 P L=4U W=16U
M3 2 2 80 80 N L=16U W=4U
M4 3 2 80 80 N L=16U W=4U
M5 1 70 90 90 P L=10U W=4U
M6 8 3 80 80 N L=4U W=240U
M7 8 4 90 90 P L=4U W=200U
M8 4 2 80 80 N L=4U W=4U
M9 4 4 90 90 P L=4U W=4U
MC1 8 3 8 90 P L=40U W=40U
.ENDS COP

.MODEL N NMOS LEVEL=2.00000 LD=0.212340U TOX=412.000E-10
+NSUB=5.860000E+15 VTO=0.783736 KP=5.460000E-05
+GAMMA=0.5262 PHI=0.600000 UO=651.000 UEXP=0.177364
+UCRIT=30664.6 DELTA=1.99612 VMAX=57874.1 XJ=0.1500000U
+LAMBDA=3.533329E-02 NFS=9.544270E+11 NEFF=1
+NSS=1.000000E+12 TPG=1.00000 RSH=33.400 CGSO=2.669565E-10
+CGDO=2.669565E-10 CGBO=4.250255E-10 CJ=1.13400E-4
+MJ=0.70800 CJSW=4.77000E-10 MJSW=0.253000 PB=0.80000

.MODEL P PMOS LEVEL=2.00000 LD=0.250000U TOX=412.000E-10
+NSUB=6.740E+15 VTO=-0.807 KP=2.13000E-05
+GAMMA=0.5644 PHI=0.60 UO=253.997 UEXP=0.2458
+UCRIT=16929.2 DELTA=1.001368E-06 VMAX=37028.2 XJ=0.0500U
+LAMBDA=5.659491E-02 NFS=1.000E+11 NEFF=1.001
+NSS=1.000000E+12 TPG=-1.00000 RSH=121.600 CGSO=3.143031E-10
+CGDO=3.143031E-10 CGBO=4.574377E-10 CJ=2.5400E-4 MJ=0.55300
+CJSW=3.3100E-10 MJSW=0.35200 PB=0.80

.PROBE
.TRAN 100U 55M
.OPTIONS LIMPTS=100000
.END
```

APPENDIX B.4

PSPICE input file for the W-nonlinearity of the 1-D cell

```
VY1   1   0    PWL(0 -2.5 10M 2.5 20M -2.5 30M 2.5 40M -2.5 50M 2.5)
VY2   2   0    0.0
VX1   6   0    2.5
EX2   7   0    0 6 1
VC2   60  0    4.959

XVW1    1 2 3 5 VSW
XVF1    3 4 4    COP1
XVF2    5 55 55 COP1
XVX1    6 7 8 9 VSWX
XMUL1   4 55 8 9 11 12 10 60 MUL1

.SUBCKT VSW 1 3 20 40
VSS 80   0 -5.0
VDD 90   0  5.0
V70 70   0  3.0

M1   90 1   2   80 N L=22U W=11U
M3   2  1   80  80 N L=4U W=10U
M5   80 2   20  90 P L=4U W=4U
M7   20 70  90  90 P L=4U W=4U
M2   90 3   4   80 N L=22U W=11U
M4   4  3   80  80 N L=4U W=10U
M6   80 4   40  90 P L=4U W=4U
M8   40 70  90  90 P L=4U W=4U
.ENDS VSW

.SUBCKT VSWX 1 3 2 4
VSS 80   0 -5.0
VDD 90   0  5.0

M1   80 1   2   90 P L=16U W=8U
M3   2  1   90  90 P L=4U W=20U
M2   80 3   4   90 P L=16U W=8U
M4   4  3   90  90 P L=4U W=20U
.ENDS VSWX

.SUBCKT MUL1 11 12 13 14 5 6 8 60
VSS 80 0 -5.0
VC1 50 0  5.0

MG1X 11 14 6 80 N L=24U W=4U
MD1X 12 13 6 80 N L=24U W=4U
MG2X 12 14 5 80 N L=24U W=4U
MD2X 11 13 5 80 N L=24U W=4U
MR61 6  60 0 80 N L=48U W=4U
MR62 6  50 8 80 N L=48U W=4U
MR51 5  60 8 80 N L=48U W=4U
MR52 5  50 0 80 N L=48U W=4U
XCOP1 5 6 8 COP
.ENDS MUL1
```

```
.SUBCKT COP1 5 6 8
VBIAS 70 0 3.2
VDD 90 0  5.0
VSS 80 0 -5.0

M1 2 6 1 90 P L=4U W=8U
M2 3 5 1 90 P L=4U W=8U
M3 2 2 80 80 N L=16U W=4U
M4 3 2 80 80 N L=16U W=4U
M5 1 70 90 90 P L=16U W=4U
M6 8 3 80 80 N L=4U W=60U
M7 8 4 90 90 P L=4U W=47U
M8 4 2 80 80 N L=4U W=4U
M9 4 4 90 90 P L=4U W=4U
MC1 8 3 8 90 P L=20U W=10U
.ENDS COP1

.SUBCKT COP 5 6 8
VBIAS 70 0 3.2
VDD 90 0  5.0
VSS 80 0 -5.0

M1 2 6 1 90 P L=4U W=16U
M2 3 5 1 90 P L=4U W=16U
M3 2 2 80 80 N L=16U W=4U
M4 3 2 80 80 N L=16U W=4U
M5 1 70 90 90 P L=10U W=4U
M6 8 3 80 80 N L=4U W=240U
M7 8 4 90 90 P L=4U W=200U
M8 4 2 80 80 N L=4U W=4U
M9 4 4 90 90 P L=4U W=4U
MC1 8 3 8 90 P L=40U W=40U
.ENDS COP

.MODEL N NMOS LEVEL=2.00000 LD=0.212340U TOX=412.000E-10
+NSUB=5.860000E+15 VTO=0.783736 KP=5.460000E-05
+GAMMA=0.5262 PHI=0.600000 UO=651.000 UEXP=0.177364
+UCRIT=30664.6 DELTA=1.99612 VMAX=57874.1 XJ=0.1500000U
+LAMBDA=3.533329E-02 NFS=9.544270E+11 NEFF=1
+NSS=1.000000E+12 TPG=1.00000 RSH=33.400 CGSO=2.669565E-10
+CGDO=2.669565E-10 CGBO=4.250255E-10 CJ=1.13400E-4
+MJ=0.70800 CJSW=4.77000E-10 MJSW=0.253000 PB=0.80000

.MODEL P PMOS LEVEL=2.00000 LD=0.250000U TOX=412.000E-10
+NSUB=6.740E+15 VTO=-0.807 KP=2.13000E-05
+GAMMA=0.5644 PHI=0.60 UO=253.997 UEXP=0.2458
+UCRIT=16929.2 DELTA=1.001368E-06 VMAX=37028.2 XJ=0.0500U
+LAMBDA=5.659491E-02 NFS=1.000E+11 NEFF=1.001
+NSS=1.000000E+12 TPG=-1.00000 RSH=121.600 CGSO=3.143031E-10
+CGDO=3.143031E-10 CGBO=4.574377E-10 CJ=2.5400E-4 MJ=0.55300
+CJSW=3.3100E-10 MJSW=0.35200 PB=0.80

.PROBE
.TRAN 100U 55M
.OPTIONS LIMPTS=100000
.END
```

APPENDIX B.5

PSPICE input file for the step response of the 1-D cell

```
VY1   1   0   2.5
VY2   2   0   0
VX1   6   0   PULSE(-2.5 2.5 0.1N 0.1N 0.1N 1M 2M)
EX2   7   0   0 6 1
VC2   60   0   4.959

XVW1    1 2 3 5 VSW
XVF1    3 4 4   COP1
XVF2    5 55 55 COP1
XVX1    6 7 8 9 VSWX
XMUL1   4 55 8 9 11 12 10 60 MUL1

.SUBCKT VSW 1 3 20 40
VSS 80   0 -5.0
VDD 90   0   5.0
V70 70   0   3.0

M1   90 1   2   80 N L=22U W=11U
M3   2   1   80 80 N L=4U W=10U
M5   80 2   20 90 P L=4U W=4U
M7   20 70 90 90 P L=4U W=4U
M2   90 3   4   80 N L=22U W=11U
M4   4   3   80 80 N L=4U W=10U
M6   80 4   40 90 P L=4U W=4U
M8   40 70 90 90 P L=4U W=4U
.ENDS VSW

.SUBCKT VSWX 1 3 2 4
VSS 80   0 -5.0
VDD 90   0   5.0

M1   80 1   2   90 P L=16U W=8U
M3   2   1   90 90 P L=4U W=20U
M2   80 3   4   90 P L=16U W=8U
M4   4   3   90 90 P L=4U W=20U
.ENDS VSWX

.SUBCKT MUL1 11 12 13 14 5 6 8 60
VSS 80 0 -5.0
VC1 50 0   5.0

MG1X 11 14 6 80 N L=24U W=4U
MD1X 12 13 6 80 N L=24U W=4U
MG2X 12 14 5 80 N L=24U W=4U
MD2X 11 13 5 80 N L=24U W=4U
MR61 6   60 0 80 N L=48U W=4U
MR62 6   50 8 80 N L=48U W=4U
MR51 5   60 8 80 N L=48U W=4U
MR52 5   50 0 80 N L=48U W=4U
XCOP1 5 6 8 COP
.ENDS MUL1
```

```
.SUBCKT COP1 5 6 8
VBIAS 70 0 3.2
VDD 90 0  5.0
VSS 80 0 -5.0

M1 2 6 1 90 P L=4U W=8U
M2 3 5 1 90 P L=4U W=8U
M3 2 2 80 80 N L=16U W=4U
M4 3 2 80 80 N L=16U W=4U
M5 1 70 90 90 P L=16U W=4U
M6 8 3 80 80 N L=4U W=60U
M7 8 4 90 90 P L=4U W=47U
M8 4 2 80 80 N L=4U W=4U
M9 4 4 90 90 P L=4U W=4U
MC1 8 3 8 90 P L=20U W=10U
.ENDS COP1

.SUBCKT COP 5 6 8
VBIAS 70 0 3.2
VDD 90 0  5.0
VSS 80 0 -5.0

M1 2 6 1 90 P L=4U W=16U
M2 3 5 1 90 P L=4U W=16U
M3 2 2 80 80 N L=16U W=4U
M4 3 2 80 80 N L=16U W=4U
M5 1 70 90 90 P L=10U W=4U
M6 8 3 80 80 N L=4U W=240U
M7 8 4 90 90 P L=4U W=200U
M8 4 2 80 80 N L=4U W=4U
M9 4 4 90 90 P L=4U W=4U
MC1 8 3 8 90 P L=40U W=40U
.ENDS COP

.MODEL N NMOS LEVEL=2.00000 LD=0.212340U TOX=412.000E-10
+NSUB=5.860000E+15 VTO=0.783736 KP=5.460000E-05
+GAMMA=0.5262 PHI=0.600000 UO=651.000 UEXP=0.177364
+UCRIT=30664.6 DELTA=1.99612 VMAX=57874.1 XJ=0.1500000U
+LAMBDA=3.533329E-02 NFS=9.544270E+11 NEFF=1
+NSS=1.000000E+12 TPG=1.00000 RSH=33.400 CGSO=2.669565E-10
+CGDO=2.669565E-10 CGBO=4.250255E-10 CJ=1.13400E-4
+MJ=0.70800 CJSW=4.77000E-10 MJSW=0.253000 PB=0.80000

.MODEL P PMOS LEVEL=2.00000 LD=0.250000U TOX=412.000E-10
+NSUB=6.740E+15 VTO=-0.807 KP=2.13000E-05
+GAMMA=0.5644 PHI=0.60 UO=253.997 UEXP=0.2458
+UCRIT=16929.2 DELTA=1.001368E-06 VMAX=37028.2 XJ=0.0500U
+LAMBDA=5.659491E-02 NFS=1.000E+11 NEFF=1.001
+NSS=1.000000E+12 TPG=-1.00000 RSH=121.600 CGSO=3.143031E-10
+CGDO=3.143031E-10 CGBO=4.574377E-10 CJ=2.5400E-4 MJ=0.55300
+CJSW=3.3100E-10 MJSW=0.35200 PB=0.80

.PROBE
.TRAN 10US 6MS
.OPTIONS LIMPTS=100000
.END
```

APPENDIX B.6

PSPICE input file for the 11-D cell

```
VY1   1   0
VY2   2   0   0.0
VX1   6   0
EX2   7   0   0 6 1
VC2   60  0   4.549

XVW1    1 2 3  5  VSW
XVW2    1 2 31 51 VSW
XVW3    1 2 32 52 VSW
XVW4    1 2 33 53 VSW
XVW5    1 2 34 54 VSW
XVW6    1 2 35 55 VSW
XVW7    1 2 36 56 VSW
XVW8    1 2 37 57 VSW
XVW9    1 2 38 58 VSW
XVW10   1 2 39 59 VSW
XVW11   1 2 41 61 VSW
XVX1    6 7 8 9 VSX
XMUL1   3 5 8 9 11 12 MULF
XMUL2   31 51 8 9 11 12 MULF
XMUL3   32 52 8 9 11 12 MULF
XMUL4   33 53 8 9 11 12 MULF
XMUL5   34 54 8 9 11 12 MULF
XMUL6   35 55 8 9 11 12 MULF
XMUL7   36 56 8 9 11 12 MULF
XMUL8   37 57 8 9 11 12 MULF
XMUL9   38 58 8 9 11 12 MULF
XMUL10 39 59 8 9 11 12 MULF
XMUL11 41 61 8 9 11 12 MULF
XMULB   11 12 10 60 MULB

.SUBCKT VSW 1 3 21 41
VSS 80   0 -5.0
VDD 90   0  5.0
V70 70   0  3.0

M1   90 1  2   80 N L=22U W=11U
M3   2  1  80 80 N L=4U W=10U
M5   80 2  20 90 P L=4U W=4U
M7   20 70 90 90 P L=4U W=4U
M2   90 3  4   80 N L=22U W=11U
M4   4  3  80 80 N L=4U W=10U
M6   80 4  40 90 P L=4U W=4U
M8   40 70 90 90 P L=4U W=4U
XVF20 20 21 21 COP1
XVF40 40 41 41 COP1
.ENDS VSW
```

```
.SUBCKT VSX 1 3 2 4
VSS 80  0 -5.0
VDD 90  0  5.0

M1  80 1  2  90 P L=16U W=8U
M3  2  1  90 90 P L=4U W=20U
M2  80 3  4  90 P L=16U W=8U
M4  4  3  90 90 P L=4U W=20U
.ENDS VSX

.SUBCKT MULF 11 12 13 14 5 6
VSS 80 0 -5.0
MG1X 11 14 6 80 N L=24U W=4U
MD1X 12 13 6 80 N L=24U W=4U
MG2X 12 14 5 80 N L=24U W=4U
MD2X 11 13 5 80 N L=24U W=4U
.ENDS MULF

.SUBCKT MULB 5 6 8 60
VSS 80 0 -5.0
VC1 50 0  5.0
MR61 6  60 0 80 N L=48U W=4U
MR62 6  50 8 80 N L=48U W=4U
MR51 5  60 8 80 N L=48U W=4U
MR52 5  50 0 80 N L=48U W=4U
XCOP1 5 6 8 COP
.ENDS MULB

.SUBCKT COP1 5 6 8
VBIAS 70 0 3.2
VDD 90 0  5.0
VSS 80 0 -5.0

M1 2 6 1 90 P L=4U W=8U
M2 3 5 1 90 P L=4U W=8U
M3 2 2 80 80 N L=16U W=4U
M4 3 2 80 80 N L=16U W=4U
M5 1 70 90 90 P L=16U W=4U
M6 8 3 80 80 N L=4U W=60U
M7 8 4 90 90 P L=4U W=47U
M8 4 2 80 80 N L=4U W=4U
M9 4 4 90 90 P L=4U W=4U
MC1 8 3 8 90 P L=20U W=10U
.ENDS COP1

.SUBCKT COP 5 6 8
VBIAS 70 0 3.2
VDD 90 0  5.0
VSS 80 0 -5.0

M1 2 6 1 90 P L=4U W=16U
M2 3 5 1 90 P L=4U W=16U
M3 2 2 80 80 N L=16U W=4U
M4 3 2 80 80 N L=16U W=4U
M5 1 70 90 90 P L=10U W=4U
M6 8 3 80 80 N L=4U W=240U
M7 8 4 90 90 P L=4U W=200U
M8 4 2 80 80 N L=4U W=4U
M9 4 4 90 90 P L=4U W=4U
MC1 8 3 8 90 P L=40U W=40U
.ENDS COP
```

```
.MODEL N NMOS LEVEL=2.00000 LD=0.212340U TOX=412.000E-10
+NSUB=5.860000E+15 VTO=0.783736 KP=5.460000E-05
+GAMMA=0.5262 PHI=0.600000 UO=651.000 UEXP=0.177364
+UCRIT=30664.6 DELTA=1.99612 VMAX=57874.1 XJ=0.1500000U
+LAMBDA=3.533329E-02 NFS=9.544270E+11 NEFF=1
+NSS=1.000000E+12 TPG=1.00000 RSH=33.400 CGSO=2.669565E-10
+CGDO=2.669565E-10 CGBO=4.250255E-10 CJ=1.13400E-4
+MJ=0.70800 CJSW=4.77000E-10 MJSW=0.253000 PB=0.80000

.MODEL P PMOS LEVEL=2.00000 LD=0.250000U TOX=412.000E-10
+NSUB=6.740E+15 VTO=-0.807 KP=2.13000E-05
+GAMMA=0.5644 PHI=0.60 UO=253.997 UEXP=0.2458
+UCRIT=16929.2 DELTA=1.001368E-06 VMAX=37028.2 XJ=0.0500U
+LAMBDA=5.659491E-02 NFS=1.000E+11 NEFF=1.001
+NSS=1.000000E+12 TPG=-1.00000 RSH=121.600 CGSO=3.143031E-10
+CGDO=3.143031E-10 CGBO=4.574377E-10 CJ=2.5400E-4 MJ=0.55300
+CJSW=3.3100E-10 MJSW=0.35200 PB=0.80

.PROBE
.DC VY1 -2.5  2.5  0.1 VX1 -2.5 2.5 2.5
.OPTIONS LIMPTS=100000
.PRINT DC V(3,5) V(8,9) V(11) V(11,12) V(10)
.END
```

APPENDIX B.7

PSPICE input file for the sigmoid-derivative cell

```
VY1  1   0
V111 111 0  2.5
V117 117 0  2.5
VC2  60  0  4.99

XDS1   1  101  NEU

XVW1   101 0   103 105 VSW
XVF1   103 104 104 COP
XVF2   105 106 106 COP
XVX1   0   101 108 109 VSX

XVW2   111 0   113 115 VSW
XVF3   113 114 114 COP
XVF4   115 116 116 COP
XVX2   117 0   118 119 VSX

XMUL1  104 106 108 109 11 12 MUL1
XMUL2  114 116 118 119 11 12 MUL1

XFDR1  11 12 10 60 FDR

.SUBCKT NEU 1 3
VSS1 80 0 -5.0
VDD1 90 0  5.0
VSS2 60 0 -2.5
VDD2 70 0  2.5
M1 2 1 60 80 N L=2U W=2U
M2 2 1 70 90 P L=2U W=4U
M3 3 2 60 80 N L=2U W=2U
M4 3 2 70 90 P L=2U W=4U
.ENDS NEU

.SUBCKT VSX 1 3 2 4
VSS 80  0 -5.0
VDD 90  0  5.0
M1  80 1  2  90 P L=16U W=8U
M3  2  1  90 90 P L=4U W=20U
M2  80 3  4  90 P L=16U W=8U
M4  4  3  90 90 P L=4U W=20U
.ENDS VSX

.SUBCKT VSW 1 3 20 40
VSS 80  0 -5.0
VDD 90  0  5.0
V70 70  0  3.
M1  90 1  2  80 N L=22U W=11U
M3  2  1  80 80 N L=4U W=10U
M5  80 2  20 90 P L=4U W=4U
M7  20 70 90 90 P L=4U W=4U
```

```
M2  90 3  4  80 N L=22U W=11U
M4  4  3  80 80 N L=4U W=10U
M6  80 4  40 90 P L=4U W=4U
M8  40 70 90 90 P L=4U W=4U
.ENDS VSW

.SUBCKT MUL1 11 12 13 14 5 6
VSS 80  0 -5.0
MG1X 11 14 6 80 N L=24U W=4U
MD1X 12 13 6 80 N L=24U W=4U
MG2X 12 14 5 80 N L=24U W=4U
MD2X 11 13 5 80 N L=24U W=4U
.ENDS MUL1

.SUBCKT FDR 5 6 8 60
VSS 80  0 -5.0
VC1 50  0  5.0
MR61 6  60 0 80 N L=44U W=4U
MR62 6  50 8 80 N L=44U W=4U
MR51 5  60 8 80 N L=44U W=4U
MR52 5  50 0 80 N L=44U W=4U
XCOP1 5 6 8 COP
.ENDS FDR

.SUBCKT COP 5 6 10
VBIAS 70 0 -3.5
VDD 90 0  5.0
VSS 80 0 -5.0

M1 7 5 1 80 N L=4U W=24U
M2 8 6 1 80 N L=4U W=24U
M3 3 2 7 80 N L=24U W=4U
M4 2 2 8 80 N L=24U W=4U
M5 3 2 90 90 P L=24U W=4U
M6 2 2 90 90 P L=24U W=4U
M7 1 70 80 80 N L=16U W=4U
M8 10 70 80 80 N L=4U W=200U
M9 10 3  90 90 P L=4U W=280U
MR 3  80 9  90 P L=4U W=16U
MCN 10 9 10 80 N L=40U W=40U
MCP 10 9 10 90 P L=40U W=40U
.ENDS COP

.MODEL N NMOS LEVEL=2.00000 LD=0.212340U TOX=412.000E-10
+NSUB=5.860000E+15 VTO=0.783736 KP=5.460000E-05
+GAMMA=0.5262 PHI=0.600000 UO=651.000 UEXP=0.177364
+UCRIT=30664.6 DELTA=1.99612 VMAX=57874.1 XJ=0.1500000U
+LAMBDA=3.533329E-02 NFS=9.544270E+11 NEFF=1
+NSS=1.000000E+12 TPG=1.00000 RSH=33.400 CGSO=2.669565E-10
+CGDO=2.669565E-10 CGBO=4.250255E-10 CJ=1.13400E-4
+MJ=0.70800 CJSW=4.77000E-10 MJSW=0.253000 PB=0.80000

.MODEL P PMOS LEVEL=2.00000 LD=0.250000U TOX=412.000E-10
+NSUB=6.740E+15 VTO=-0.807 KP=2.13000E-05
+GAMMA=0.5644 PHI=0.60 UO=253.997 UEXP=0.2458
+UCRIT=16929.2 DELTA=1.001368E-06 VMAX=37028.2 XJ=0.0500U
+LAMBDA=5.659491E-02 NFS=1.000E+11 NEFF=1.001
+NSS=1.000000E+12 TPG=-1.00000 RSH=121.600 CGSO=3.143031E-10
+CGDO=3.143031E-10 CGBO=4.574377E-10 CJ=2.5400E-4 MJ=0.55300
+CJSW=3.3100E-10 MJSW=0.35200 PB=0.80
```

```
.PROBE
.DC VY1 -0.05 0.05 0.0001
.OPTIONS LIMPTS=100000
.PRINT DC V(101) V(10)
.END
```

APPENDIX B.8

PSPICE input file for the Table 5.2.1

```
VH1 1   0 4.5
VL1 2   0 0.5
VL2 100 0 1.0
VRR 120 0 2.5

XMH1 3 1 4 2 5 MUL2
XNH1 5 9 101 NEU1

XMH2 6 1 7 2 8 MUL2
XNH2 8 10 103 NEU1

XMO1 11 9 12 10 13 MUL2
XNO1 13 14 NEU2

XL11 101 100 14 102 11 MUL11
XL12 103 100 14 104 12 MUL11
XL21 11  100 14 105 MUL12
XL24 12  100 14 106 MUL12
XL23 105 1  120 108 3  MUL11
XL22 105 2  120 107 4  MUL11
XL26 106 1  120 110 6  MUL11
XL25 106 2  120 109 7  MUL11

.SUBCKT MUL2 1 3 11 13 8
VSS  80 0 -5.0
V2   2  0 -3.0
V4   4  0  2.5
MG1X 1  4  6 80 N L=40U W=4U
MD1X 2  3  6 80 N L=40U W=4U
MG2X 2  4  5 80 N L=40U W=4U
MD2X 1  3  5 80 N L=40U W=4U
MG1  11 4  6 80 N L=40U W=4U
MD1  2  13 6 80 N L=40U W=4U
MG2  2  4  5 80 N L=40U W=4U
MD2  11 13 5 80 N L=40U W=4U
EM1 8 0 5 6 1E08
RI 5 6 100MEG
R1 8 6 140K
R2 5 0 140K
.ENDS MUL2

.SUBCKT NEU1 1 5 7
VDD  90 0  5.0
VR2 2 0 2.5
EVS1 3 2 1 0 1
MN1 4 3 0  0  N L=10U W=4U
MP1 4 3 90 90 P L=4U W=10U
MN2 5 4 0  0  N L=4U W=4U
MP2 5 4 90 90 P L=10U W=10U
VR  6 0 -4.0
EVS2 7 6 5 0 0.4
.ENDS NEU1
```

```
.SUBCKT NEU2 1 5
VDD  90 0  5.0
VR2 2 0 2.5
EVS 3 2 1 0 1
MN1 4 3 0  0  N L=4U W=10U
MP1 4 3 90 90 P L=4U W=4U
MN2 5 4 0  0  N L=4U W=10U
MP2 5 4 90 90 P L=4U W=4U
.ENDS NEU2
.SUBCKT MUL11 1 3 4 8 10
VSS 80 0 -5.0
VYC2 2 0 -3.0
MG1X 1 4 6 80 N L=40U W=4U
MD1X 2 3 6 80 N L=40U W=4U
MG2X 2 4 5 80 N L=40U W=4U
MD2X 1 3 5 80 N L=40U W=4U
RI 5 6 100MEG
EM1 7 0 5 6 1E08
R1 7 6 100K
R2 5 0 100K
RR 7 8 100
CC 8 0 1PF
EVS 10 2 8 0 1
.ENDS MUL11

.SUBCKT MUL12 1 3 4 10
VSS 80 0 -5.0
VYC2 2 0 -3.0
MG1X 1 4 6 80 N L=40U W=4U
MD1X 2 3 6 80 N L=40U W=4U
MG2X 2 4 5 80 N L=40U W=4U
MD2X 1 3 5 80 N L=40U W=4U
RI 5 6 100MEG
EM1 7 0 5 6 1E08
R1 7 6 100K
R2 5 0 100K
EVS 10 2 7 0 1
.ENDS MUL12

.MODEL N NMOS LEVEL=2.00000 LD=0.280000U TOX=500.000E-10
+NSUB=1.000000E+16 VTO=0.827125 KP=3.286649E-05
+GAMMA=1.35960 PHI=0.600000 UO=200.000 UEXP=1.001000E-03
+UCRIT=999000. DELTA=1.24050 VMAX=100000. XJ=0.400000U
+LAMBDA=1.604983E-02 NFS=1.234795E+12 NEFF=1.001000E-02
+NSS=0.000000E+00 TPG=1.00000 RSH=25 CGSO=5.2E-10
+CGDO=5.2E-10 CJ=3.2E-4 MJ=0.5 CJSW=9E-10 MJSW=0.33

.MODEL P PMOS LEVEL=2.00000 LD=0.280000U TOX=500.000E-10
+NSUB=1.121088E+14 VTO=-0.894654 KP=1.526452E-05
+GAMMA=0.879003 PHI=0.600000 UO=100.000 UEXP=0.153441
+UCRIT=16376.5 DELTA=1.93831 VMAX=100000. XJ=0.400000U
+LAMBDA=4.708659E-02 NFS=8.788617E+11 NEFF=1.001000E-02
+NSS=0.000000E+00 TPG=-1.00000 RSH=95 CGSO=4E-10
+CGDO=4E-10 CJ=2E-4 MJ=0.5 CJSW=4.5E-10 MJSW=0.33

.IC V(108)=-0.5 V(107)=0.6 V(110)=-0.5 V(109)=0.5
.IC V(102)=0.7  V(104)=0.5
.PROBE
.TRAN 0.1ns 1ns
.END
```

APPENDIX B.9

PSPICE input file for the dc analysis of the Table 5.2.1

```
VX1 1   0
VX2 2   0
VW11 3   0   -4.0
VW12 4   0   -4.0
VW21 6   0   -4.0
VW22 7   0   -4.0
VW31 11  0   -2.8603
VW32 12  0   -2.8609
VT1  16  0   -4.0
VT2  17  0   -4.0
VT11 18  0   -4.0

XFF1 1 2 3 4 6 7 9 10 11 12 14 16 17 18 FFXR

.SUBCKT FFXR 1 2 3 4 6 7 9 10 11 12 14 16 17 18
XMH1 3 1 4 2 16 5 MUL3
XNH1 5 9 NEU1
XMH2 6 1 7 2 17 8 MUL3
XNH2 8 10 NEU1
XMO1 11 9 12 10 18 13 MUL3
XNO1 13 14 NEU1
.ENDS FFXR

.SUBCKT MUL3 1 3 11 13 21 8
VSS  80 0 -5.0
V2   2  0 -3.0
V4   4  0  2.5
V23 23  0  3.5

MG11 1   4   6 80 N L=400U W=4U
MD11 2   3   6 80 N L=400U W=4U
MG12 2   4   5 80 N L=400U W=4U
MD12 1   3   5 80 N L=400U W=4U

MG21 11 4   6 80 N L=400U W=4U
MD21 2  13  6 80 N L=400U W=4U
MG22 2  4   5 80 N L=400U W=4U
MD22 11 13  5 80 N L=400U W=4U

MG31 21 4   6 80 N L=400U W=4U
MD31 2  23  6 80 N L=400U W=4U
MG32 2  4   5 80 N L=400U W=4U
MD32 21 23  5 80 N L=400U W=4U

EM1 8 0 5 6 1E08
RI 5 6 100MEG
R1 8 6 400K
R2 5 0 400K
.ENDS MUL3
```

```
.SUBCKT NEU1 1 5
VDD  90 0  5.0
VR2 2 0 2.5
EVS1 3 2 1 0 1
MN1 4 3 0  0  N L=2U W=2U
MP1 4 3 90 90 P L=2U W=4U
MN2 5 4 0  0  N L=2U W=2U
MP2 5 4 90 90 P L=2U W=4U
.ENDS NEU1

.MODEL N NMOS LEVEL=2.00000 LD=0.280000U TOX=500.000E-10
+NSUB=1.000000E+16 VTO=0.827125 KP=3.286649E-05
+GAMMA=1.35960 PHI=0.600000 UO=200.000 UEXP=1.001000E-03
+UCRIT=999000. DELTA=1.24050 VMAX=100000. XJ=0.400000U
+LAMBDA=1.604983E-02 NFS=1.234795E+12 NEFF=1.001000E-02
+NSS=0.000000E+00 TPG=1.00000 RSH=25 CGSO=5.2E-10
+CGDO=5.2E-10 CJ=3.2E-4 MJ=0.5 CJSW=9E-10 MJSW=0.33

.MODEL P PMOS LEVEL=2.00000 LD=0.280000U TOX=500.000E-10
+NSUB=1.121088E+14 VTO=-0.894654 KP=1.526452E-05
+GAMMA=0.879003 PHI=0.600000 UO=100.000 UEXP=0.153441
+UCRIT=16376.5 DELTA=1.93831 VMAX=100000. XJ=0.400000U
+LAMBDA=4.708659E-02 NFS=8.788617E+11 NEFF=1.001000E-02
+NSS=0.000000E+00 TPG=-1.00000 RSH=95 CGSO=4E-10
+CGDO=4E-10 CJ=2E-4 MJ=0.5 CJSW=4.5E-10 MJSW=0.33

.PRINT DC V(2) V(9) V(10) V(14)
.PROBE
.OPTIONS ITL4=40 RELTOL=0.01 ITL5=0
.DC VX1 0.5 4.5 4.0 VX2 0.5 4.5 4.0
.END
```

APPENDIX B.10

PSPICE input file for Figure 5.2.2

```
VX1   1   0  PULSE(0.5 4.5 0.02ns 0 0 0.01ns 0.04ns)
VX2   2   0  PULSE(0.5 4.5 0.01ns 0 0 0.01ns 0.02ns)
VT  100   0  PULSE(1.0 4.5 0.01ns 0 0 0.02ns 0.04ns)

VRR 120 0 2.5

XMH1 3 1 4 2 5 MUL2
XNH1 5 9 101 NEU1

XMH2 6 1 7 2 8 MUL2
XNH2 8 10 103 NEU1

XMO1 11 9 12 10 13 MUL2
XNO1 13 14 NEU2

XL11 101 100 14 102 11 MUL11
XL12 103 100 14 104 12 MUL11
XL21 11  100 14 105 MUL12
XL24 12  100 14 106 MUL12
XL23 105 1  120 108 3  MUL11
XL22 105 2  120 107 4  MUL11
XL26 106 1  120 110 6  MUL11
XL25 106 2  120 109 7  MUL11

.SUBCKT MUL2 1 3 11 13 8
VSS  80 0 -5.0
V2   2  0 -3.0
V4   4  0  2.5
MG1X 1  4  6 80 N L=40U W=4U
MD1X 2  3  6 80 N L=40U W=4U
MG2X 2  4  5 80 N L=40U W=4U
MD2X 1  3  5 80 N L=40U W=4U
MG1  11 4  6 80 N L=40U W=4U
MD1  2  13 6 80 N L=40U W=4U
MG2  2  4  5 80 N L=40U W=4U
MD2  11 13 5 80 N L=40U W=4U
EM1 8 0 5 6 1E08
RI 5 6 100MEG
R1 8 6 140K
R2 5 0 140K
.ENDS MUL2

.SUBCKT NEU1 1 5 7
VDD  90 0  5.0
VR2 2 0 2.5
EVS1 3 2 1 0 1
MN1 4 3 0  0  N L=10U W=4U
MP1 4 3 90 90 P L=4U W=10U
MN2 5 4 0  0  N L=4U W=4U
MP2 5 4 90 90 P L=10U W=10U
VR  6 0 -4.0
EVS2 7 6 5 0 0.4
.ENDS NEU1
```

```
.SUBCKT NEU2 1 5
VDD  90 0  5.0
VR2 2 0 2.5
EVS 3 2 1 0 1
MN1 4 3 0  0  N L=4U W=10U
MP1 4 3 90 90 P L=4U W=4U
MN2 5 4 0  0  N L=4U W=10U
MP2 5 4 90 90 P L=4U W=4U
.ENDS NEU2
.SUBCKT MUL11 1 3 4 8 10
VSS 80 0 -5.0
VYC2 2 0 -3.0
MG1X 1 4 6 80 N L=40U W=4U
MD1X 2 3 6 80 N L=40U W=4U
MG2X 2 4 5 80 N L=40U W=4U
MD2X 1 3 5 80 N L=40U W=4U
RI 5 6 100MEG
EM1 7 0 5 6 1E08
R1 7 6 100K
R2 5 0 100K
RR 7 8 100
CC 8 0 1PF
EVS 10 2 8 0 1
.ENDS MUL11

.SUBCKT MUL12 1 3 4 10
VSS 80 0 -5.0
VYC2 2 0 -3.0
MG1X 1 4 6 80 N L=40U W=4U
MD1X 2 3 6 80 N L=40U W=4U
MG2X 2 4 5 80 N L=40U W=4U
MD2X 1 3 5 80 N L=40U W=4U
RI 5 6 100MEG
EM1 7 0 5 6 1E08
R1 7 6 100K
R2 5 0 100K
EVS 10 2 7 0 1
.ENDS MUL12

.MODEL N NMOS LEVEL=2.00000 LD=0.280000U TOX=500.000E-10
+NSUB=1.000000E+16 VTO=0.827125 KP=3.286649E-05
+GAMMA=1.35960 PHI=0.600000 UO=200.000 UEXP=1.001000E-03
+UCRIT=999000. DELTA=1.24050 VMAX=100000. XJ=0.400000U
+LAMBDA=1.604983E-02 NFS=1.234795E+12 NEFF=1.001000E-02
+NSS=0.000000E+00 TPG=1.00000 RSH=25 CGSO=5.2E-10
+CGDO=5.2E-10 CJ=3.2E-4 MJ=0.5 CJSW=9E-10 MJSW=0.33

.MODEL P PMOS LEVEL=2.00000 LD=0.280000U TOX=500.000E-10
+NSUB=1.121088E+14 VTO=-0.894654 KP=1.526452E-05
+GAMMA=0.879003 PHI=0.600000 UO=100.000 UEXP=0.153441
+UCRIT=16376.5 DELTA=1.93831 VMAX=100000. XJ=0.400000U
+LAMBDA=4.708659E-02 NFS=8.788617E+11 NEFF=1.001000E-02
+NSS=0.000000E+00 TPG=-1.00000 RSH=95 CGSO=4E-10
+CGDO=4E-10 CJ=2E-4 MJ=0.5 CJSW=4.5E-10 MJSW=0.33

.IC V(108)=-0.5 V(107)=0.6 V(110)=-0.5 V(109)=0.5
.IC V(102)=0.7 V(104)=0.5
.PROBE
.TRAN 0.1ns 1ns
.END
```

APPENDIX B.11

PSPICE input file for Table 5.2.3

```
VH1 301 0 4.5
VL1 302 0 4.5
VTT 303 0 0.5
VR1 304 0 4.5

***FF Circuits***

XFF1 301 302 3 4 5 6 11 12 1 2 13 8 9 7 FFXR

***Learning Circuits***

XL11 11 303 13 121 101 1 MUL11
XL12 12 303 13 122 102 2 MUL11
XLT1 304 303 13 127 107 7 MUL11

XC11 1 303 13 131 MUL13
XC12 2 303 13 132 MUL13

XI11  131 301 123 103 3 MUL12
XI12  131 302 124 104 4 MUL12
XI21  132 301 125 105 5 MUL12
XI22  132 302 126 106 6 MUL12

XTH12 131 304 128 108 8 MUL12
XTH22 132 304 129 109 9 MUL12

.SUBCKT FFXR 1 2 3 4 6 7 9 10 11 12 14 16 17 18
XMH1 3 1 4 2 16 5 MUL3
XNH1 5 9 NEU1
XMH2 6 1 7 2 17 8 MUL3
XNH2 8 10 NEU1
XMO1 11 9 12 10 18 13 MUL3
XNO1 13 14 NEU1
.ENDS FFXR

.SUBCKT MUL3 1 3 11 13 21 8
VSS  80 0 -5.0
V2   2  0 -3.0
V4   4  0  2.5
V23 23  0  3.5

MG11 1  4  6 80 N L=400U W=4U
MD11 2  3  6 80 N L=400U W=4U
MG12 2  4  5 80 N L=400U W=4U
MD12 1  3  5 80 N L=400U W=4U

MG21 11 4  6 80 N L=400U W=4U
MD21 2  13 6 80 N L=400U W=4U
MG22 2  4  5 80 N L=400U W=4U
MD22 11 13 5 80 N L=400U W=4U
```

```
MG31 21 4  6 80 N L=400U W=4U
MD31 2  23 6 80 N L=400U W=4U
MG32 2  4  5 80 N L=400U W=4U
MD32 21 23 5 80 N L=400U W=4U

EM1 8 0 5 6 1E08
RI 5 6 100MEG
R1 8 6 400K
R2 5 0 400K
.ENDS MUL3

.SUBCKT NEU1 1 5
VDD  90 0  5.0
VR2 2 0 2.5
EVS1 3 2 1 0 1
MN1 4 3 0  0  N L=2U W=2U
MP1 4 3 90 90 P L=2U W=4U
MN2 5 4 0  0  N L=2U W=2U
MP2 5 4 90 90 P L=2U W=4U
.ENDS NEU1

.SUBCKT MUL12 11 3 8 10 12
VSS 80 0 -5.0
VSS1 50 0 -1.0
VDD1 60 0  1.0
VYC2 2 0 -3.0
VXC2 4 0 2.5
EVX1 1 2 11 0 1
MG1X 1 4 6 80 N L=40U W=4U
MD1X 2 3 6 80 N L=40U W=4U
MG2X 2 4 5 80 N L=40U W=4U
MD2X 1 3 5 80 N L=40U W=4U
RI 5 6 100MEG
EM1 7 0 5 6 1E08
R1 7 6 100K
R2 5 0 100K
ROU 7 8 100
C8  8 0 1PF

MN1 9 8 50 50 N L=10U W=4U
MP1 9 8 60 60 P L=4U W=10U
MN2 10 9 50 50 N L=10U W=4U
MP2 10 9 60 60 P L=4U W=10U
EVS 12 2 10 0 1
.ENDS MUL12

.SUBCKT MUL11 101 3 4 8 10 12
VSS 80 0 -5.0
VSS1 50 0 -1.0
VDD1 60 0  1.0
VYC2 2  0 -3.0
VCC1 20 0 -4.0
EVX1 1 20 101 0 0.4

MG11X 1 4 6 80 N L=40U W=4U
MD11X 2 3 6 80 N L=40U W=4U
MG12X 2 4 5 80 N L=40U W=4U
MD12X 1 3 5 80 N L=40U W=4U
```

```
R1 7 6 100K
R2 5 0 100K
EVOP 7 0 5 6 1E08
RIN  5 6 100MEG
ROU 7 8 100
C8  8 0 1PF

MN1 9 8 50 50 N L=10U W=4U
MP1 9 8 60 60 P L=4U W=10U
MN2 10 9 50 50 N L=10U W=4U
MP2 10 9 60 60 P L=4U W=10U
EVS 12 2 10 0 1
.ENDS MUL11

.SUBCKT MUL13 1 3 4 7
VSS 80 0 -5.0
VYC2 2  0 -3.0

MG11X 1 4 6 80 N L=40U W=4U
MD11X 2 3 6 80 N L=40U W=4U
MG12X 2 4 5 80 N L=40U W=4U
MD12X 1 3 5 80 N L=40U W=4U

R1 7 6 100K
R2 5 0 100K
EVOP 7 0 5 6 1E08
RIN  5 6 100MEG
.ENDS MUL13

.MODEL N NMOS LEVEL=2.00000 LD=0.280000U TOX=500.000E-10
+NSUB=1.000000E+16 VTO=0.827125 KP=3.286649E-05
+GAMMA=1.35960 PHI=0.600000 UO=200.000 UEXP=1.001000E-03
+UCRIT=999000. DELTA=1.24050 VMAX=100000. XJ=0.400000U
+LAMBDA=1.604983E-02 NFS=1.234795E+12 NEFF=1.001000E-02
+NSS=0.000000E+00 TPG=1.00000 RSH=25 CGSO=5.2E-10
+CGDO=5.2E-10 CJ=3.2E-4 MJ=0.5 CJSW=9E-10 MJSW=0.33

.MODEL P PMOS LEVEL=2.00000 LD=0.280000U TOX=500.000E-10
+NSUB=1.121088E+14 VTO=-0.894654 KP=1.526452E-05
+GAMMA=0.879003 PHI=0.600000 UO=100.000 UEXP=0.153441
+UCRIT=16376.5 DELTA=1.93831 VMAX=100000. XJ=0.400000U
+LAMBDA=4.708659E-02 NFS=8.788617E+11 NEFF=1.001000E-02
+NSS=0.000000E+00 TPG=-1.00000 RSH=95 CGSO=4E-10
+CGDO=4E-10 CJ=2E-4 MJ=0.5 CJSW=4.5E-10 MJSW=0.33

.IC V(123)=3.412E-03 V(124)=3.412E-03 V(125)=2.516E-04 V(126)=3.407E-04
.IC V(121)=3.410E-03 V(122)=3.411E-03
.IC V(128)=1.215E-03 V(129)=9.873E-05 V(127)=-4.714E-03

.print tran v(11) v(12) v(13) v(1)
.print tran v(2) v(3) v(4) v(5) v(6)
.print tran v(108) v(109) v(107) v(102)
.print tran v(103) v(104) v(105) v(106) v(101)
.print tran v(123) v(124) v(125) v(126) v(121)
.print tran v(128) v(129) v(127) v(122)
.PROBE
.OPTIONS ITL4=40 RELTOL=0.01 ITL5=0
.TRAN 0.2ns 3.0ns
.END
```

APPENDIX B.12

PSPICE input file for Table 5.3.1

```
VH1 301 0 4.5
VL1 302 0 1.0
VL2 303 0 1.0
***FF Circuits***

XFF1 302 302 3 4 5 6 11 12 1 2 13 8 9 7 FFXR
XFF2 301 302 3 4 5 6 21 22 1 2 23 8 9 7 FFXR
XFF3 302 301 3 4 5 6 31 32 1 2 33 8 9 7 FFXR
XFF4 301 301 3 4 5 6 41 42 1 2 43 8 9 7 FFXR

***Learning Circuits***

XL11 11 21 31 41 303 301 301 303 13 23 33 43 121 101 1 MULP
XL12 12 22 32 42 303 301 301 303 13 23 33 43 122 102 2 MULP
XLT1 301 301 301 301 303 301 301 303 13 23 33 43 127 107 7 MULP

XL111 302 301 302 301 303 301 301 303 13 23 33 43 113 MULP2
XI11  113 101 123 103 3 MUL12
XL112 302 302 301 301 303 301 301 303 13 23 33 43 114 MULP2
XI12  114 101 124 104 4 MUL12
XL121 302 301 302 301 303 301 301 303 13 23 33 43 115 MULP2
XI21  115 102 125 105 5 MUL12
XL122 302 302 301 301 303 301 301 303 13 23 33 43 116 MULP2
XI22  116 102 126 106 6 MUL12

XLT11 301 301 301 301 303 301 301 303 13 23 33 43 118 MULP2
XTH11 118 101 128 108 8 MUL12
XLT22 301 301 301 301 303 301 301 303 13 23 33 43 119 MULP2
XTH22 119 102 129 109 9 MUL12

.SUBCKT FFXR 1 2 3 4 6 7 9 10 11 12 14 16 17 18
XMH1 3 1 4 2 16 5 MUL3
XNH1 5 9 NEU1
XMH2 6 1 7 2 17 8 MUL3
XNH2 8 10 NEU1
XMO1 11 9 12 10 18 13 MUL3
XNO1 13 14 NEU1
.ENDS FFXR

.SUBCKT MUL3 1 3 11 13 21 8
VSS  80 0 -5.0
V2   2  0 -3.0
V4   4  0  2.5
V23 23  0  3.5

MG11 1  4  6 80 N L=400U W=4U
MD11 2  3  6 80 N L=400U W=4U
MG12 2  4  5 80 N L=400U W=4U
MD12 1  3  5 80 N L=400U W=4U
```

```
MG21 11 4   6 80 N L=400U W=4U
MD21 2  13 6 80 N L=400U W=4U
MG22 2  4   5 80 N L=400U W=4U
MD22 11 13 5 80 N L=400U W=4U

MG31 21 4   6 80 N L=400U W=4U
MD31 2  23 6 80 N L=400U W=4U
MG32 2  4   5 80 N L=400U W=4U
MD32 21 23 5 80 N L=400U W=4U

EM1 8 0 5 6 1E08
RI 5 6 100MEG
R1 8 6 400K
R2 5 0 400K
.ENDS MUL3

.SUBCKT NEU1 1 5
VDD  90 0  5.0
VR2 2 0 2.5
EVS1 3 2 1 0 1
MN1 4 3 0  0  N L=2U W=2U
MP1 4 3 90 90 P L=2U W=4U
MN2 5 4 0  0  N L=2U W=2U
MP2 5 4 90 90 P L=2U W=4U
.ENDS NEU1

.SUBCKT MUL12 11 13 8 10 12
VSS 80 0 -5.0
VSS1 50 0 -1.0
VDD1 60 0  1.0
VYC2 2 0 -3.0
VXC2 4 0 2.5
EVX1 1 2 11 0 1
EVY1 3 4 13 0 2.5
MG1X 1 4 6 80 N L=40U W=4U
MD1X 2 3 6 80 N L=40U W=4U
MG2X 2 4 5 80 N L=40U W=4U
MD2X 1 3 5 80 N L=40U W=4U
RI 5 6 100MEG
EM1 7 0 5 6 1E08
R1 7 6 100K
R2 5 0 100K
*EVA1 77 0 7 0 50
ROU 7 8 100
C8  8 0 1PF

MN1 9 8 50 50 N L=10U W=4U
MP1 9 8 60 60 P L=4U W=10U
MN2 10 9 50 50 N L=10U W=4U
MP2 10 9 60 60 P L=4U W=10U
EVS 12 2 10 0 1
.ENDS MUL12

.SUBCKT MULP 101 111 121 131 3 13 23 33 4 14 24 34 8 10 12
VSS 80 0 -5.0
VSS1 50 0 -1.0
VDD1 60 0  1.0
VYC2 2  0 -3.0
VCC1 20 0 -4.0
```

```
EVX1 1 20 101 0 0.4
EVX11 11 20 111 0 0.4
EVX21 21 20 121 0 0.4
EVX31 31 20 131 0 0.4

MG11X 1 4 6 80 N L=400U W=4U
MD11X 2 3 6 80 N L=400U W=4U
MG12X 2 4 5 80 N L=400U W=4U
MD12X 1 3 5 80 N L=400U W=4U

MG21X 11 14 6 80 N L=400U W=4U
MD21X 2 13 6 80  N L=400U W=4U
MG22X 2 14 5 80  N L=400U W=4U
MD22X 11 13 5 80 N L=400U W=4U

MG31X 21 24 6 80 N L=400U W=4U
MD31X 2 23 6 80  N L=400U W=4U
MG32X 2 24 5 80  N L=400U W=4U
MD32X 21 23 5 80 N L=400U W=4U
MG41X 31 34 6 80 N L=400U W=4U
MD41X 2 33 6 80  N L=400U W=4U
MG42X 2 34 5 80  N L=400U W=4U
MD42X 31 33 5 80 N L=400U W=4U

R1 7 6 300K
R2 5 0 300K
EVOP 7 0 5 6 1E08
RIN  5 6 100MEG
*EVA1 77 0 7 0 50
ROU 7 8 100
C8  8 0 1PF

MN1 9 8 50 50 N L=10U W=4U
MP1 9 8 60 60 P L=4U W=10U
MN2 10 9 50 50 N L=10U W=4U
MP2 10 9 60 60 P L=4U W=10U
EVS 12 2 10 0 1
.ENDS MULP

.SUBCKT MULP2 101 111 121 131 3 13 23 33 4 14 24 34 7
VSS 80 0 -5.0
VYC2 2  0 -3.0
VCC1 20 0 -4.0
EVX1 1 20 101 0 0.4
EVX11 11 20 111 0 0.4
EVX21 21 20 121 0 0.4
EVX31 31 20 131 0 0.4

MG11X 1 4 6 80 N L=400U W=4U
MD11X 2 3 6 80 N L=400U W=4U
MG12X 2 4 5 80 N L=400U W=4U
MD12X 1 3 5 80 N L=400U W=4U

MG21X 11 14 6 80 N L=400U W=4U
MD21X 2 13 6 80  N L=400U W=4U
MG22X 2 14 5 80  N L=400U W=4U
MD22X 11 13 5 80 N L=400U W=4U
```

```
MG31X 21 24 6 80 N L=400U W=4U
MD31X 2 23 6 80  N L=400U W=4U
MG32X 2 24 5 80  N L=400U W=4U
MD32X 21 23 5 80 N L=400U W=4U

MG41X 31 34 6 80 N L=400U W=4U
MD41X 2 33 6 80  N L=400U W=4U
MG42X 2 34 5 80  N L=400U W=4U
MD42X 31 33 5 80 N L=400U W=4U

R1 7 6 300K
R2 5 0 300K
EVOP 7 0 5 6 1E08
RIN  5 6 100MEG
.ENDS MULP2

.MODEL N NMOS LEVEL=2.00000 LD=0.280000U TOX=500.000E-10
+NSUB=1.000000E+16 VTO=0.827125 KP=3.286649E-05
+GAMMA=1.35960 PHI=0.600000 UO=200.000 UEXP=1.001000E-03
+UCRIT=999000. DELTA=1.24050 VMAX=100000. XJ=0.400000U
+LAMBDA=1.604983E-02 NFS=1.234795E+12 NEFF=1.001000E-02
+NSS=0.000000E+00 TPG=1.00000 RSH=25 CGSO=5.2E-10
+CGDO=5.2E-10 CJ=3.2E-4 MJ=0.5 CJSW=9E-10 MJSW=0.33

.MODEL P PMOS LEVEL=2.00000 LD=0.280000U TOX=500.000E-10
+NSUB=1.121088E+14 VTO=-0.894654 KP=1.526452E-05
+GAMMA=0.879003 PHI=0.600000 UO=100.000 UEXP=0.153441
+UCRIT=16376.5 DELTA=1.93831 VMAX=100000. XJ=0.400000U
+LAMBDA=4.708659E-02 NFS=8.788617E+11 NEFF=1.001000E-02
+NSS=0.000000E+00 TPG=-1.00000 RSH=95 CGSO=4E-10
+CGDO=4E-10 CJ=2E-4 MJ=0.5 CJSW=4.5E-10 MJSW=0.33

.IC V(123)=0.01 V(124)=-0.01 V(125)=-0.01 V(126)=0.01
.IC V(121)=0.01 V(122)=0.01  V(128)=-0.01 V(129)=-0.01 V(127)=0.01

.print tran v(113) v(114) v(115) v(116)
.print tran v(11) v(21) v(31) v(41)
.print tran v(12) v(22) v(32) v(42)
.print tran v(123) v(124) v(125) v(126) v(121)
.print tran v(128) v(129) v(127) v(122)
.print tran v(103) v(104) v(105) v(106) v(101)
.print tran v(108) v(109) v(107) v(102)
.print tran v(13) v(23) v(33) v(43)
.print tran v(118) v(119)
.PROBE
.OPTIONS ITL4=40 RELTOL=0.01 ITL5=0
.TRAN 0.4ns 8.0ns
.END
```

APPENDIX B.13

PSPICE input file for Table 5.3.2

```
VH1 301 0 4.5
VL1 302 0 1.0
VL2 303 0 1.0
***FF Circuits***

XFF1 302 302 3 4 5 6 11 12 1 2 13 8 9 7 FFXR
XFF2 301 302 3 4 5 6 21 22 1 2 23 8 9 7 FFXR
XFF3 302 301 3 4 5 6 31 32 1 2 33 8 9 7 FFXR
XFF4 301 301 3 4 5 6 41 42 1 2 43 8 9 7 FFXR

***Learning Circuits***

XL11 11 21 31 41 303 301 301 303 13 23 33 43 121 101 1 MULP
XL12 12 22 32 42 303 301 301 303 13 23 33 43 122 102 2 MULP
XLT1 301 301 301 301 303 301 301 303 13 23 33 43 127 107 7 MULP

XL111 302 301 302 301 303 301 301 303 13 23 33 43 113 MULP2
XI11  113 101 123 103 3 MUL12
XL112 302 302 301 301 303 301 301 303 13 23 33 43 114 MULP2
XI12  114 101 124 104 4 MUL12
XL121 302 301 302 301 303 301 301 303 13 23 33 43 115 MULP2
XI21  115 102 125 105 5 MUL12
XL122 302 302 301 301 303 301 301 303 13 23 33 43 116 MULP2
XI22  116 102 126 106 6 MUL12

XLT11 301 301 301 301 303 301 301 303 13 23 33 43 118 MULP2
XTH11 118 101 128 108 8 MUL12
XLT22 301 301 301 301 303 301 301 303 13 23 33 43 119 MULP2
XTH22 119 102 129 109 9 MUL12

.SUBCKT FFXR 1 2 3 4 6 7 9 10 11 12 14 16 17 18
XMH1 3 1 4 2 16 5 MUL3
XNH1 5 9 NEU1
XMH2 6 1 7 2 17 8 MUL3
XNH2 8 10 NEU1
XMO1 11 9 12 10 18 13 MUL3
XNO1 13 14 NEU1
.ENDS FFXR

.SUBCKT MUL3 1 3 11 13 21 8
VSS  80 0 -5.0
V2   2  0 -3.0
V4   4  0  2.5
V23 23  0  3.5

MG11 1  4  6 80 N L=400U W=4U
MD11 2  3  6 80 N L=400U W=4U
MG12 2  4  5 80 N L=400U W=4U
MD12 1  3  5 80 N L=400U W=4U

MG21 11 4  6 80 N L=400U W=4U
MD21 2  13 6 80 N L=400U W=4U
MG22 2  4  5 80 N L=400U W=4U
MD22 11 13 5 80 N L=400U W=4U
```

```
MG31 21 4   6 80 N L=400U W=4U
MD31 2  23 6 80 N L=400U W=4U
MG32 2  4   5 80 N L=400U W=4U
MD32 21 23 5 80 N L=400U W=4U

EM1 8 0 5 6 1E08
RI 5 6 100MEG
R1 8 6 400K
R2 5 0 400K
.ENDS MUL3

.SUBCKT NEU1 1 5
VDD  90 0  5.0
VR2 2 0 2.5
EVS1 3 2 1 0 1
MN1 4 3 0  0  N L=2U W=2U
MP1 4 3 90 90 P L=2U W=4U
MN2 5 4 0  0  N L=2U W=2U
MP2 5 4 90 90 P L=2U W=4U
.ENDS NEU1

.SUBCKT MUL12 11 13 8 10 12
VSS 80 0 -5.0
VSS1 50 0 -1.0
VDD1 60 0  1.0
VYC2 2 0 -3.0
VXC2 4 0 2.5
EVX1 1 2 11 0 1
EVY1 3 4 13 0 2.5
MG1X 1 4 6 80 N L=40U W=4U
MD1X 2 3 6 80 N L=40U W=4U
MG2X 2 4 5 80 N L=40U W=4U
MD2X 1 3 5 80 N L=40U W=4U
RI 5 6 100MEG
EM1 7 0 5 6 1E08
R1 7 6 100K
R2 5 0 100K
*EVA1 77 0 7 0 50
ROU 7 8 100
C8  8 0 1PF

MN1 9 8 50 50 N L=10U W=4U
MP1 9 8 60 60 P L=4U W=10U
MN2 10 9 50 50 N L=10U W=4U
MP2 10 9 60 60 P L=4U W=10U
EVS 12 2 10 0 1
.ENDS MUL12

.SUBCKT MULP 101 111 121 131 3 13 23 33 4 14 24 34 8 10 12
VSS 80 0 -5.0
VSS1 50 0 -1.0
VDD1 60 0  1.0
VYC2 2  0 -3.0
VCC1 20 0 -4.0
EVX1 1 20 101 0 0.4
EVX11 11 20 111 0 0.4
EVX21 21 20 121 0 0.4
EVX31 31 20 131 0 0.4
```

```
MG11X 1 4 6 80 N L=400U W=4U
MD11X 2 3 6 80 N L=400U W=4U
MG12X 2 4 5 80 N L=400U W=4U
MD12X 1 3 5 80 N L=400U W=4U

MG21X 11 14 6 80 N L=400U W=4U
MD21X 2 13 6 80  N L=400U W=4U
MG22X 2 14 5 80  N L=400U W=4U
MD22X 11 13 5 80 N L=400U W=4U

MG31X 21 24 6 80 N L=400U W=4U
MD31X 2 23 6 80  N L=400U W=4U
MG32X 2 24 5 80  N L=400U W=4U
MD32X 21 23 5 80 N L=400U W=4U

MG41X 31 34 6 80 N L=400U W=4U
MD41X 2 33 6 80  N L=400U W=4U
MG42X 2 34 5 80  N L=400U W=4U
MD42X 31 33 5 80 N L=400U W=4U

R1 7 6 300K
R2 5 0 300K
EVOP 7 0 5 6 1E08
RIN  5 6 100MEG
*EVA1 77 0 7 0 50
ROU 7 8 100
C8  8 0 1PF

MN1 9 8 50 50 N L=10U W=4U
MP1 9 8 60 60 P L=4U W=10U
MN2 10 9 50 50 N L=10U W=4U
MP2 10 9 60 60 P L=4U W=10U
EVS 12 2 10 0 1
.ENDS MULP

.SUBCKT MULP2 101 111 121 131 3 13 23 33 4 14 24 34 7
VSS 80 0 -5.0
VYC2 2  0 -3.0
VCC1 20 0 -4.0
EVX1 1 20 101 0 0.4
EVX11 11 20 111 0 0.4
EVX21 21 20 121 0 0.4
EVX31 31 20 131 0 0.4

MG11X 1 4 6 80 N L=400U W=4U
MD11X 2 3 6 80 N L=400U W=4U
MG12X 2 4 5 80 N L=400U W=4U
MD12X 1 3 5 80 N L=400U W=4U

MG21X 11 14 6 80 N L=400U W=4U
MD21X 2 13 6 80  N L=400U W=4U
MG22X 2 14 5 80  N L=400U W=4U
MD22X 11 13 5 80 N L=400U W=4U

MG31X 21 24 6 80 N L=400U W=4U
MD31X 2 23 6 80  N L=400U W=4U
MG32X 2 24 5 80  N L=400U W=4U
MD32X 21 23 5 80 N L=400U W=4U
```

```
MG41X 31 34 6 80 N L=400U W=4U
MD41X 2 33 6 80  N L=400U W=4U
MG42X 2 34 5 80  N L=400U W=4U
MD42X 31 33 5 80 N L=400U W=4U

R1 7 6 300K
R2 5 0 300K
EVOP 7 0 5 6 1E08
RIN  5 6 100MEG
.ENDS MULP2

.MODEL N NMOS LEVEL=2.00000 LD=0.280000U TOX=500.000E-10
+NSUB=1.000000E+16 VTO=0.827125 KP=3.286649E-05
+GAMMA=1.35960 PHI=0.600000 UO=200.000 UEXP=1.001000E-03
+UCRIT=999000. DELTA=1.24050 VMAX=100000. XJ=0.400000U
+LAMBDA=1.604983E-02 NFS=1.234795E+12 NEFF=1.001000E-02
+NSS=0.000000E+00 TPG=1.00000 RSH=25 CGSO=5.2E-10
+CGDO=5.2E-10 CJ=3.2E-4 MJ=0.5 CJSW=9E-10 MJSW=0.33

.MODEL P PMOS LEVEL=2.00000 LD=0.280000U TOX=500.000E-10
+NSUB=1.121088E+14 VTO=-0.894654 KP=1.526452E-05
+GAMMA=0.879003 PHI=0.600000 UO=100.000 UEXP=0.153441
+UCRIT=16376.5 DELTA=1.93831 VMAX=100000. XJ=0.400000U
+LAMBDA=4.708659E-02 NFS=8.788617E+11 NEFF=1.001000E-02
+NSS=0.000000E+00 TPG=-1.00000 RSH=95 CGSO=4E-10
+CGDO=4E-10 CJ=2E-4 MJ=0.5 CJSW=4.5E-10 MJSW=0.33

.IC V(123)=0.01 V(124)=-0.01 V(125)=-0.01 V(126)=0.01
.IC V(121)=0.01 V(122)=0.01  V(128)=-0.01 V(129)=-0.01 V(127)=0.01

.print tran v(113) v(114) v(115) v(116)
.print tran v(11) v(21) v(31) v(41)
.print tran v(12) v(22) v(32) v(42)
.print tran v(123) v(124) v(125) v(126) v(121)
.print tran v(128) v(129) v(127) v(122)
.print tran v(103) v(104) v(105) v(106) v(101)
.print tran v(108) v(109) v(107) v(102)
.print tran v(13) v(23) v(33) v(43)
.print tran v(118) v(119)
*.PROBE
.OPTIONS ITL4=40 RELTOL=0.01 ITL5=0
.TRAN 0.4ns 8.0ns
.END
```

APPENDIX B.14

PSPICE input file for Table 6.1.1(a)

```
VH1 301 0 4.5
VL1 302 0 4.5
VTT 303 0 0.5
VR1 304 0 4.5

***FF Circuits***

XFF1 301 302 3 4 5 6 13 14 1 2 11 12 15 16 8 9 7 10 FFXR

***Learning Circuits***

XL11 13 303 15 121 101 1 MUL11
XL12 14 303 15 122 102 2 MUL11
XL21 13 303 16 131 111 11 MUL11
XL22 14 303 16 132 112 12 MUL11

XLT1 304 303 15 127 107 7 MUL11
XLT2 304 303 16 137 117 10 MUL11

XC11 1 303 15 2 303 16 141 MUL2
XC12 11 303 15 12 303 16 142 MUL2

XI11  141 301 123 103 3 MUL12
XI12  141 302 124 104 4 MUL12
XI21  142 301 125 105 5 MUL12
XI22  142 302 126 106 6 MUL12

XTH12 141 304 128 108 8 MUL12
XTH22 142 304 129 109 9 MUL12

.SUBCKT FFXR 1 2 3 4 6 7 9 10 11 12 15 16 14 18 21 22 23 24
XMH1 3 1 4 2 21 5 MUL3
XNH1 5 9 NEU1
XMH2 6 1 7 2 22 8 MUL3
XNH2 8 10 NEU1
XMO1 11 9 12 10 23 13 MUL3
XNO1 13 14 NEU1
XMO2 15 9 16 10 24 17 MUL3
XNO2 17 18 NEU1
.ENDS FFXR

.SUBCKT MUL3 1 3 11 13 21 8
VSS  80 0 -5.0
V2   2  0 -3.0
V4   4  0  2.5
V23 23  0  3.5

MG11 1  4  6 80 N L=400U W=4U
MD11 2  3  6 80 N L=400U W=4U
MG12 2  4  5 80 N L=400U W=4U
MD12 1  3  5 80 N L=400U W=4U
```

```
MG21 11 4   6 80 N L=400U W=4U
MD21 2  13 6 80 N L=400U W=4U
MG22 2  4   5 80 N L=400U W=4U
MD22 11 13 5 80 N L=400U W=4U

MG31 21 4   6 80 N L=400U W=4U
MD31 2  23 6 80 N L=400U W=4U
MG32 2  4   5 80 N L=400U W=4U
MD32 21 23 5 80 N L=400U W=4U

EM1 8 0 5 6 1E08
RI 5 6 100MEG
R1 8 6 400K
R2 5 0 400K
.ENDS MUL3

.SUBCKT MUL2 1 3 4 11 13 14 8
VSS  80 0 -5.0
V2   2  0 -3.0

MG11 1  4   6 80 N L=40U W=4U
MD11 2  3   6 80 N L=40U W=4U
MG12 2  4   5 80 N L=40U W=4U
MD12 1  3   5 80 N L=40U W=4U

MG21 11 4   6 80 N L=40U W=4U
MD21 2  13 6 80 N L=40U W=4U
MG22 2  4   5 80 N L=40U W=4U
MD22 11 13 5 80 N L=40U W=4U

EM1 8 0 5 6 1E08
RI 5 6 100MEG
R1 8 6 60K
R2 5 0 60K
.ENDS MUL2

.SUBCKT NEU1 1 5
VDD  90 0  5.0
VR2 2 0 2.5
EVS1 3 2 1 0 1
MN1 4 3 0  0  N L=2U W=2U
MP1 4 3 90 90 P L=2U W=4U
MN2 5 4 0  0  N L=2U W=2U
MP2 5 4 90 90 P L=2U W=4U
.ENDS NEU1

.SUBCKT MUL12 11 3 8 10 12
VSS 80 0 -5.0
VSS1 50 0 -1.0
VDD1 60 0  1.0
VYC2 2 0 -3.0
VXC2 4 0 2.5
EVX1 1 2 11 0 1
MG1X 1 4 6 80 N L=40U W=4U
MD1X 2 3 6 80 N L=40U W=4U
MG2X 2 4 5 80 N L=40U W=4U
MD2X 1 3 5 80 N L=40U W=4U
```

```
RI 5 6 100MEG
EM1 7 0 5 6 1E08
R1 7 6 100K
R2 5 0 100K
*EVA1 77 0 7 0 50
ROU 7 8 100
C8  8 0 1PF

MN1 9 8 50 50 N L=10U W=4U
MP1 9 8 60 60 P L=4U W=10U
MN2 10 9 50 50 N L=10U W=4U
MP2 10 9 60 60 P L=4U W=10U
EVS 12 2 10 0 1
.ENDS MUL12

.SUBCKT MUL11 101 3 4 8 10 12
VSS 80 0 -5.0
VSS1 50 0 -1.0
VDD1 60 0  1.0
VYC2 2  0 -3.0
VCC1 20 0 -4.0
EVX1 1 20 101 0 0.4

MG11X 1 4 6 80 N L=40U W=4U
MD11X 2 3 6 80 N L=40U W=4U
MG12X 2 4 5 80 N L=40U W=4U
MD12X 1 3 5 80 N L=40U W=4U

R1 7 6 100K
R2 5 0 100K
EVOP 7 0 5 6 1E08
RIN  5 6 100MEG
*EVA1 77 0 7 0 50
ROU 7 8 100
C8  8 0 1PF

MN1 9 8 50 50 N L=10U W=4U
MP1 9 8 60 60 P L=4U W=10U
MN2 10 9 50 50 N L=10U W=4U
MP2 10 9 60 60 P L=4U W=10U
EVS 12 2 10 0 1
.ENDS MUL11

.SUBCKT MUL13 1 3 4 7
VSS 80 0 -5.0
VYC2 2  0 -3.0

MG11X 1 4 6 80 N L=40U W=4U
MD11X 2 3 6 80 N L=40U W=4U
MG12X 2 4 5 80 N L=40U W=4U
MD12X 1 3 5 80 N L=40U W=4U

R1 7 6 100K
R2 5 0 100K
EVOP 7 0 5 6 1E08
RIN  5 6 100MEG
.ENDS MUL13
```

```
.MODEL N NMOS LEVEL=2.00000 LD=0.280000U TOX=500.000E-10
+NSUB=1.000000E+16 VTO=0.827125 KP=3.286649E-05
+GAMMA=1.35960 PHI=0.600000 UO=200.000 UEXP=1.001000E-03
+UCRIT=999000. DELTA=1.24050 VMAX=100000. XJ=0.400000U
+LAMBDA=1.604983E-02 NFS=1.234795E+12 NEFF=1.001000E-02
+NSS=0.000000E+00 TPG=1.00000 RSH=25 CGSO=5.2E-10
+CGDO=5.2E-10 CJ=3.2E-4 MJ=0.5 CJSW=9E-10 MJSW=0.33

.MODEL P PMOS LEVEL=2.00000 LD=0.280000U TOX=500.000E-10
+NSUB=1.121088E+14 VTO=-0.894654 KP=1.526452E-05
+GAMMA=0.879003 PHI=0.600000 UO=100.000 UEXP=0.153441
+UCRIT=16376.5 DELTA=1.93831 VMAX=100000. XJ=0.400000U
+LAMBDA=4.708659E-02 NFS=8.788617E+11 NEFF=1.001000E-02
+NSS=0.000000E+00 TPG=-1.00000 RSH=95 CGSO=4E-10
+CGDO=4E-10 CJ=2E-4 MJ=0.5 CJSW=4.5E-10 MJSW=0.33

.IC V(123)=0.5 V(124)=-0.5 V(125)=-0.5 V(126)=0.5
.IC V(121)=0.5 V(122)=0.5  V(131)=0.5  V(132)=0.5
.IC V(128)=-0.5 V(129)=-0.5 V(127)=0.5 V(137)=0.5

.print tran v(13) v(14) v(15) v(16)
.print tran v(1) v(2) v(11) v(12)
.print tran v(3) v(4) v(5) v(6)
.print tran v(8) v(9) v(7) v(10)
.print tran v(101) v(102) v(111) v(112)
.print tran v(103) v(104) v(105) v(106)
.print tran v(121) v(122) v(131) v(132)
.print tran v(123) v(124) v(125) v(126)
.print tran v(108) v(109) v(107) v(117)
.print tran v(128) v(129) v(127) v(137)
.PROBE
.OPTIONS ITL4=40 RELTOL=0.01 ITL5=0
.TRAN 0.2ns 3.0ns
.END
```

APPENDIX B.15

PSPICE input file for Table 6.1.1(c)

```
V1     1 0
V2     2 0

VWH11 3  0 -4.0
VWH12 4  0 -4.0
VWH21 6  0 -4.0
VWH22 7  0 -4.0
VWO11 11 0 -3.1396
VWO12 12 0 -3.1396
VWO21 15 0 -3.1396
VWO22 16 0 -3.1396
VTH1  21 0 -4.0
VTH2  22 0 -4.0
VTO1  23 0 -4.0
VTO2  24 0 -4.0

XMH1 3 1 4 2 21 5 MUL3
XNH1 5 9 NEU1
XMH2 6 1 7 2 22 8 MUL3
XNH2 8 10 NEU1
XMO1 11 9 12 10 23 13 MUL3
XNO1 13 14 NEU1
XMO2 15 9 16 10 24 17 MUL3
XNO2 17 18 NEU1

.SUBCKT MUL3 1 3 11 13 21 8
VSS  80 0 -5.0
V2   2  0 -3.0
V4   4  0  2.5
V23 23  0  3.5

MG11 1  4  6 80 N L=400U W=4U
MD11 2  3  6 80 N L=400U W=4U
MG12 2  4  5 80 N L=400U W=4U
MD12 1  3  5 80 N L=400U W=4U

MG21 11 4  6 80 N L=400U W=4U
MD21 2  13 6 80 N L=400U W=4U
MG22 2  4  5 80 N L=400U W=4U
MD22 11 13 5 80 N L=400U W=4U

MG31 21 4  6 80 N L=400U W=4U
MD31 2  23 6 80 N L=400U W=4U
MG32 2  4  5 80 N L=400U W=4U
MD32 21 23 5 80 N L=400U W=4U

EM1 8 0 5 6 1E08
RI 5 6 100MEG
R1 8 6 400K
R2 5 0 400K
.ENDS MUL3
```

```
.SUBCKT NEU1 1 5
VDD  90 0  5.0
VR2 2 0 2.5
EVS1 3 2 1 0 1
MN1 4 3 0  0  N L=2U W=2U
MP1 4 3 90 90 P L=2U W=4U
MN2 5 4 0  0  N L=2U W=2U
MP2 5 4 90 90 P L=2U W=4U
.ENDS NEU1

.MODEL N NMOS LEVEL=2.00000 LD=0.280000U TOX=500.000E-10
+NSUB=1.000000E+16 VTO=0.827125 KP=3.286649E-05
+GAMMA=1.35960 PHI=0.600000 UO=200.000 UEXP=1.001000E-03
+UCRIT=999000. DELTA=1.24050 VMAX=100000. XJ=0.400000U
+LAMBDA=1.604983E-02 NFS=1.234795E+12 NEFF=1.001000E-02
+NSS=0.000000E+00 TPG=1.00000 RSH=25 CGSO=5.2E-10
+CGDO=5.2E-10 CJ=3.2E-4 MJ=0.5 CJSW=9E-10 MJSW=0.33
.MODEL P PMOS LEVEL=2.00000 LD=0.280000U TOX=500.000E-10
+NSUB=1.121088E+14 VTO=-0.894654 KP=1.526452E-05
+GAMMA=0.879003 PHI=0.600000 UO=100.000 UEXP=0.153441
+UCRIT=16376.5 DELTA=1.93831 VMAX=100000. XJ=0.400000U
+LAMBDA=4.708659E-02 NFS=8.788617E+11 NEFF=1.001000E-02
+NSS=0.000000E+00 TPG=-1.00000 RSH=95 CGSO=4E-10
+CGDO=4E-10 CJ=2E-4 MJ=0.5 CJSW=4.5E-10 MJSW=0.33

.print dc v(9) v(10) v(14) v(18)
.PROBE
.OPTIONS ITL4=40 RELTOL=0.01 ITL5=0
.DC V1 1.0 4.5 3.5 V2 1.0 4.5 3.5
.END
```

LIST OF REFERENCES

[1] W. Hubbard, J. S. Denker, H. P. Graf, R. E. Howard, L. D. Jackel, B. Straughn, and D. Schwartz, "Electronic Neural Networks", AIP Neural Networks for Computing, pp. 227-234, 1986.

[2] M. Sivilotti, M. Emerling, and C. Mead, "A Novel Associative Memory Implemented Using Collective Computation", 1985 Chapel Hill Conference on VLSI, pp. 329-342.

[3] M. Sivilotti, M. Emerling, and C. Mead, "VLSI Architecture for implementation of Neural Networks", AIP Neural Networks for Computing, pp. 408-413, 1986.

[4] J. P. Sage, K. Thompson, and R. S. Withers, "An Artificial Neural Network Integrated Circuit Based on MNOS/CCD Principles", AIP Neural Networks for Computing, pp. 381-385, 1986.

[5] A. P. Thakoor, A. Moopen, J. Lambe, and S. K. Khanna, "Electronic hardware implementations of neural networks", Applied Optics, Vol. 26, pp. 5085-5092.

[6] F. Kub, I. Mack, K. Moon, C. Yao, and J. Modolo, "Programmable analog synapses for microelectronic neural networks using a hybrid digital-analog approach", poster presented at IEEE ICNN, San Diego, July 24-27, 1988.

[7] F. Kub, K. Moon, I. Mack, and F. M. Long, "Programmable Analog Vector-Matrix Multipliers", IEEE J. Solid-State Circuits, SC-25, pp. 207-214, Feb. 1990.

[8] S. Eberhardt, T. Duong, and A. Thakoor, "Design of Parallel Hardware Neural Network Systems from Custom Analog VLSI 'Building Block' Chips", IJCNN 1989, pp. II-183 - II-190.

[9] B. Furman and A. A. Abidi, "An Analog CMOS Backward Error-Propagation LSI", Asilomar Conf. on Signal, Control, and Systems, pp. 645-648, 1988.

[10] H. P. Graf and P. deVegvar, "A CMOS Implementation of a Neural Network Model", Proc. Stanford Confer. Advanced Research in VLSI, MIT Press, pp. 351-367, 1987.

[11] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities", Proc. Natl. Acad. Sci. U.S.A., vol. 79, pp. 2554-2558, 1982.

[12] J. J. Hopfield, "Neurons with graded response have collective computational properties like those of two-state neurons", Proc. Natl. Acad. Sci. U.S.A., vol. 81, pp. 3088-3092, 1984.

[13] J. J. Hopfield and D. W. Tank, "Simple neural optimization networks: an A/D converter, a signal decision circuit, and a linear programming circuit", IEEE Trans. on Circuits and Systems, vol. CAS-33, no.5, May 1986, pp. 533-541.

[14] M. Banu and Y. Tsividis, "Fully-integrated active-RC filters in MOS technology", IEEE J. Solid-State Circuits, SC-18, pp. 644-651, Dec. 1983.

[15] Y. Tsividis, M. Banu, and J. Khoury, "Continuous-time MOSFET-C Filters in VLSI", IEEE Trans. on Circuits and Systems, vol. CAS-33, pp. 125-140, Feb. 1986.

[16] Z. Czarnul, "Modification of the Banu-Tsividis continuous-time integrator structure", IEEE Trans. on Circuits and Systems, vol. CAS-33, pp. 714-716, July 1986.

[17] B.-S. Song, "CMOS RF Circuits for Data Communications Applications", IEEE J. Solid-State Circuits, SC-21, no. 2, pp. 310-317, April 1986.

[18] P. R. Gray and R. G. Meyer, "MOS operational amplifier design - A tutorial overview", IEEE J. Solid-State Circuits, SC-17, pp. 969-982, Dec. 1982.

[19] B. Gilbert, "A high-performance monolithic multiplier using active feedback", IEEE J. Solid-State Circuits, SC-9, pp. 364-373, Dec. 1974.

[20] D. Soo and R. G. Meyer, "A four-quadrant NMOS analog multiplier", IEEE J. Solid-State Circuits, SC-17, pp. 1174-1178, Dec. 1982.

[21] J. N. Babanezhad and G. C. Temes, "A 20-V Four-Quadrant CMOS Analog Multiplier", IEEE J. Solid-State Circuits, SC-20, pp. 1158-1168, Dec. 1985.

[22] P. J. Ryan and D. G. Haigh, "Novel Fully Differential MOS Transconductor for Integrated Continuous-time filters", Elect. Lett., pp. 742-743, June 1987.

[23] M. Ismail, "Four-transistor continuous-time MOS transconductor", Electron. Lett., vol. 23, no. 20, pp. 1099-1100, Sept. 1987.

[24] N. I. Khachab and M. Ismail," Novel Continuous-Time All MOS Four Quadrant Multipliers", IEEE Int. Symp. Circuits and Systems, May 1987, pp. 762-765.

[25] M. VanHorn and R. L. Geiger, "A CMOS OTA for Voltage-Controlled Analog Signal Processing", Proc. 28th Midwest Symposium on Circuits and Systems, Louisville, KY, pp. 596-599, Aug. 1985.

[26] S.-C. Qin and R. L. Geiger, "A ±5-V CMOS Analog Multiplier", IEEE J. Solid-State Circuits, SC-22, no. 6, pp. 1143-1146, Dec. 1987.

[27] F. M. A. Salam, N. Khachab, M. Ismail, and Y. Wang, "An Analog MOS Implementation of The Synaptic Weights For Feedback Neural Nets," IEEE Int. Symp. on Circuits and Systems, Portland, Oregon, CA, May 1989, pp. 1223-1226.

[28] F. M. A. Salam, M. R. Choi, and Y. Wang, "An Analog MOS Implementation of Synaptic Weights for Feedforward/Feedback Neural Nets", Proc. of 32nd Midwest Symposium on Circuits and Systems, Champaign, Illinois, August, 1989.

[29] F. M. A. Salam and M. R. Choi, "An All-MOS Analog Feedforward Neural Circuit With Learning", IEEE Int. Symp. on Circuits and Systems, May 1990, pp. 2508-2511.

[30] C. Mead, Analog VLSI and Neural Systems, Addison Wesley, 1989.

[31] D. E. Rumelhart, J. L. McClelland, and the PDP Research Group Eds., "Parallel distributed processing-Explorations in the microstructure of cognition", vol. 1, Foundations. Cambridge, MA: MIT Press, 1986.

[32] F. C. Hoppensteadt, An Introduction to the mathematics of neurons, Cambridge University Press, 1986.

[33] P. E. Allen and D. R. Holberg, CMOS Analog Circuit Design, HRW, 1987.

[34] R. Gregorian and G. C. Temes, Analog MOS Integrated Circuits For Signal Processing, John Wiley & Sons, 1986.

[35] P. R. Gray and R. G. Meyer, Analysis and Design of Analog Integrated Circuits, John Wiley & Sons, 1984.

[36] Y. Tsividis, Operation and Modeling of the MOS Transistor, McGraw-Hill, 1986.

[37] Y. Tsividis and P. Antognetti, Eds., Design of MOS VLSI Circuits for Telecommunications, Prentice-Hall, 1985.

[38] R. L. Geiger, P. E. Allen, and N. R. Strader, VLSI Design Techniques for Analog and Digital Circuits, McGraw-Hill, 1990.

[39] C. Mead and L. Conway, Introduction to VLSI Systems, Addison Wesley, 1980.

[40] N. Weste and K. Eshraghian, Principles of CMOS VLSI Design, Addison Wesley, 1985.

[41] F. M. A. Salam, M. R. Choi, Y. Wang, and B. Javidi, "On the Design of Neural Nets", 25th Annual Allerton Conference on Communication, Control, and Computing, Sep. 30 - Oct. 2, 1987.

[42] F. M. A. Salam, "A Formulation For The Design of Neural Processors", The Proceedings of the 2nd IEEE Annual International Conference on Neural

Networks (ICNN), San Diego, CA, July 24-27, 1988, pp. I-173-180.

[43] F. M. A. Salam, "A Modified Learning Rule for Feedforward Artificial Neural Nets For Analog Implementation," Memorandum No. MSU/EE/S 90/02, Department of Electrical Engineering, Michigan State University, East Lansing, MI 48824-1226, 26 January 1990.

[44] F. M. A. Salam, "A Model of Neural Circuits For Programmable VLSI Implementation", 1989 IEEE International Symposium on Circuits and Systems (ISCAS), Portland, Oregon, CA, May 9-11, 1989.

[45] F. M. A. Salam and Y. Wang, "Some Properties of Dynamic Feedback Neural Nets", in the session on Neural Networks and Control Systems, the 27th IEEE Conference on Decision and Control, December 1988, pp. 337-342.

[46] R. P. Lippmann, "An Introduction to Computing with Neural Nets", IEEE ASSP Magazine, April 1987, pp. 4-24.

[47] D. H. Sheingold, Nonlinear Circuits Handbook, Analog Devices, 1976.

[48] J. Darnell, I. Lodish, and D. Baltimore, Molecular Cell Biology, Scientific Books, 1986.

[49] F. M. A. Salam, S. Bai, and J. Hou, "Dynamic of Feedback Neural Nets with Unsupervised Learning", 1990 IEEE International Joint Conference on Neural Networks (IJCNN), San Diego, CA, June 17-21, 1990, pp. II-239-244.

[50] F. M. A. Salam and S. Bai, "A Feedback Neural Network with Supervised Learning", 1990 IEEE International Joint Conference on Neural Networks (IJCNN), San Diego, CA, June 17-21, 1990, pp. III-263-268.

[51] Linear Products Databook, Analog Devices Inc., 1988.