USC–SIPI REPORT #246

Analog-Digital VLSI Neuroprocessors for Signal Processing and Communication

by

Joongho Choi

December 1993

Signal and Image Processing Institute
UNIVERSITY OF SOUTHERN CALIFORNIA
Department of Electrical Engineering-Systems
3740 McClintock Avenue, Room 404
Los Angeles, CA 90089-2564 U.S.A.

To My Parents,

Kyu Suk Choi

and

Hisuk Chun Choi,

in gratitude for their love

Acknowledgement

I would like to express the sincerest thanks to my research advisor Professor Bing J. Sheu, Director of VLSI Signal Processing Laboratory, for his guidance and support throughout the Ph. D. research work. I wish to extend my deep appreciation to Professor Murray Gershenzon, Associate Chairman of Materials Science and Engineering Department, and Professor Sandeep Gupta for serving as my dissertation committee members. I would also like to thank Professor B. Keith Jenkins and Professor Jay C.-C. Kuo for also serving as my qualifying examination committee members.

I am very grateful to Professor Leonard Silverman, Dean of the Engineering School; Professor Hans H. Kuehl, Chairman of the Electrical Engineering - Electrophysics Department; Professor Melvin A. Breuer, Chairman of the Electrical Engineering - Systems Department; and Ms. Ramona Gordon and Ms. Gloria Halfacre in the Electrical Engineering Program, for providing me with great research environment. This research work was conducted through connections with several research organizations at the University of Southern California including Center for Neural Engineering (CNE), Signal and Image Processing Institute (SIPI), Center for Photonic Technology (CPT), and MOSIS Service of Information Science Institute (ISI). I am grateful to Professor Michael Arbib, Director of CNE, and Professor Jerry M. Mendel, Director of SIPI. Information exchanges with Professor Ted Berger of Biomedical Engineering Department; Professor Ed

Blum of Biomedical Engineering and Mathematics Department; and Professor Leon O. Chua of Electrical Engineering and Computer Science Department at University of California, Berkeley are very beneficial.

I am thankful to Professor Stephen R. Forrest of Princeton University, formerly at USC, for his stimulus and valuable discussions in design of optoelectronic integrated circuits and applications to high-speed systems. I would like to thank Professor Wonchan Kim and the faculty of Electronics Engineering Department at Seoul National University, Seoul, Korea for their guidance in my Bachelor and Master degrees study.

It will be the great pleasure to remember the period time of my stay in the VLSI Signal Processing Laboratory at USC. I am very grateful to Dr. Bang W. Lee, currently at Samsung Electronics Co., for his pioneering work in the analog VLSI design and neural network implementation. I am deeply indebted to Mr. Sa H. Bang for his personal encouragement as well as technical interaction on study of telecommunication field. Valuable discussions with Dr. Wen-Jay Hsu, currently at Sun Microsystems, and Dr. Sudhir M. Gowda, currently at IBM T. J. Watson Research Center, give me helpful insights on the VLSI modeling and simulations. Cooperation with Dr. Wai-Chi Fang, currently at Jet Propulsion Laboratory, on image compression application of neural networks is very fruitful. Support and friendship of fellow graduate students have contributed to this work. Mr. Oscal T.-C. Chen provides the good computing facility. Discussions with Ms. Josephine C.-F. Chang on digital neuroprocessors and Mr. Chen -H. Chang on optical receivers were my great memories. I also thank Mr. Tony Wu and Mr. Vincent Wang for their help.

I would like to really thank my parents, Kyu Suk Choi and Hisuk C. Choi for their love and patience, the most valuable support I have ever had. I also thank my parents-in-law for their understanding. I am thankful to my sisters and brother-in-law, Yunsun, Yunhi, and Bryan S. Kim for their encouragements all these years. Finally, but not the least, this work could be made based on love and sacrifice of my wife, Ji Han R. Choi.

Contents

List Of Tables ix

List Of Figures x

1 Introduction 1

1.1 Integrated Information Processing System 1

1.2 Development of VLSI Technology 3

1.3 Artificial Neural Network Approach 4

1.4 Hardware Implementations 10
1.4.1 Analog (or Mixed), Digital, and Pulse Signals 13
1.4.2 General-Purpose or Application-Specific 14
1.4.3 Supervised or Unsupervised Learning 15
1.4.4 On-Chip Learning 15

2 Design of Analog Neural Network Processors 17

2.1 Analog Neural Computing 17

2.2 Various Synapse Storage Scheme 23
2.2.1 Dynamic Capacitor 25
2.2.2 Digital Memory 27
2.2.3 Floating-Gate Analog Memory 29
2.2.4 Summary 30

2.3 Applications with VLSI Neuroprocessors 33

2.4 Various Design Techniques 34

3 Design of General-Purpose Neural Networks 36

3.1 Programmable Synapses 38
3.1.1 Linear Multiplication Operation 38
3.1.2 Weight Value Storage 47

3.2 Input Neurons 51

3.3 Output Neuron 53
3.3.1 Linear Current-to-Voltage Conversion 53
3.3.2 Sigmoid Function Generation 55

3.4 System-Level Considerations 56
3.4.1 Required Refreshing Frequency 56
3.4.2 Scalability of Network Size 59

3.5 Experimental Results 61

4 Neuroprocessor for Self-Organization Mapping 73

4.1 Basic Architecture 74

4.2 Considerations of WTA Circuit 77

4.3 Basic WTA Circuit 80

4.4 Analysis and Design Considerations 83
4.4.1 Cascading of Stages 85
4.4.2 Distributed Biasing 85
4.4.3 Dynamic Current Steering 87

4.5 Experimental Results 90
4.5.1 The Winner-Take-All circuit 90
4.5.2 Self-Organization Neural Network 100

5 Analog VLSI Neuroprocessors for Communication 105

5.1 Background 105

5.2 System Architecture 107

5.3 Training Algorithm 110

5.4 VLSI Implementation 112

5.5 System Environment 117

5.6 Experimental Results and System Analysis 119

6 Conclusions and Further Works 126

A Gaussian Synapse Circuit 139

A.1 Gaussian Function Networks 140

A.2 Circuit Analysis 142

A.3 Programmability 144

B Nonideal Effects in WTA Circuit 154

B.1 Device Mismatches of Input Transistor 154

B.2 Parasitic Resistance 158

List Of Tables

1.1 Summary of the microprocessors designed in the 1990's. 5

1.2 Major artificial neural network models and their properties. 12

2.1 Performance comparison of three different approaches to achieve the 32 GCPS for neural computation. 24

3.1 Various general-purpose analog neural network processor chips. . . 39

3.2 Transistor sizes of the synapse cell. 44

4.1 Summary of various WTA circuit implementations. 78

4.2 Transistor sizes of the WTA cell shown in Fig. 4.3. 92

List Of Figures

1.1 Configuration of the integrated information processing system. . . 2

1.2 Number of transistors incorporated in the integrated logic chips. . 6

1.3 Trends in standard DRAM development. 7

1.4 Several configurations of the artificial neural network consisting of the neurons and the interconnections. (a) Single-layered network. (b) Recurrent network. (c) Multi-layered network. (d) Multi-layered network with feedback. 8

1.5 Various types of artificial neural networks. 11

2.1 Block diagram of the processors for neural computation. (a) Digital arithmetic unit. (b) Analog neural computing processor. 19

2.2 Conceptual diagram of the operational accuracy in the number of the bits with respect to the network size. 22

2.3 Circuit diagrams of the synapse cells with dynamic capacitor storage. (a) Using the modified Gilbert multiplier [36]. (b) Using the four matched transistors [38]. 26

2.4 Circuit diagrams of the synapse cells with the digital memory schemes. (a) Binary-weighted current mirrors are used for directly providing the output current [47]. (b) Binary-weighted current mirrors are used for producing the tail current of the differential pair [48]. 28

2.5 Circuit diagrams of the synapse cells with the floating-gate transistor. (a) The floating-gate transistor is used in the differential pair [53]. (b) The floating-gate transistor is used in the modified Gilbert multiplier [54]. 31

2.6 Summary of comparisons in the analog synapse cells with different storage scheme. (a) Functionality. (b) Programmability. (c) Data Storage. (d) Resource efficiency. 32

3.1 Block diagram of the multi-layered neural network. 37

3.2 Circuit schematics of the differential pair in (a) and the basic Gilbert multiplier in (b). 41

3.3 Circuit schematic of the synapse cell based on the wide-range Gilbert multiplier. 43

3.4 Simulated results of the DC transfer characteristics of the synaptic multiplication. The differential weight value W_{ji} increases from -2 V to 2 V in a step size of 0.5 V. 45

3.5 Simulated results of the linearity error and the total harmonic distortion of the synaptic multiplication. 46

3.6 Synapse weight programming scheme. 48

3.7 Synapse weight storage on the capacitance through the access switch transistor. 49

3.8 Synapse weight accuracy versus the charge retention time under different values of leakage current. 50

3.9 Input neuron. (a) Differential-to-differential input neuron. (b) Single-ended-to-differential input neuron which requires two additional resistors. (c) Circuit schematic diagram of the operational amplifier. 52

3.10 Circuit schematic of the output neuron. (a) Linear current-to-voltage converter. (b) Sigmoid function generation circuit. 54

3.11 Simulated results of the sigmoid function generation circuit with gain controllability. (a) DC analysis. (b) AC analysis. 57

3.12 Improved output neuron for reconfigurability. 58

3.13 Weight accuracy versus the number of synapses connected to one common signal line. 60

3.14 Partitioning the fixed-dimension chip into several layers. 62

3.15 Cascading the chips to increase the dimension of the network. (a) To increase the number of output neurons. (b) To Increase the number of input neurons. 63

3.16 Measured results on the multiplication operation of the synapse cell. (a) Synapse current versus input voltage. (b) Synapse current versus weight voltage. 65

3.17 Measured results of the synapse weight retention characteristics. . 66

3.18 Measured results of the output neuron. (a) Linear current-to-voltage conversion. (b) Sigmoid function generation. 67

3.19 Physical layout of a differential-input synapse cell. An additional capacitor is included to hold the synapse weight information. . . . 69

3.20 Measurement setup for system-level experiments. 70

3.21 The intentionally added offset term can be removed by the learning process. 72

4.1 Block diagram of a self-organization neural network. 75

4.2 Block diagram of a self-organization neuroprocessor chip. 76

4.3 Schematic diagram of the winner-take-all (WTA) circuit. 82

4.4 SPICE simulated results of the 2-input WTA circuit. (a) DC transfer curve. (b) Transient behavior of the winning output with C_L of 0.2 pF. 84

4.5 Calculation results on a 1000-input WTA with different number of the cells having the second largest inputs. (a) DC level of the winning output. (b) Response time of the winning output (C_L = 1.0pF). 86

4.6 Comparison of the fixed and distributed bias currents for a different number of competitive cells. 88

4.7 Current steering function to ensure only one output high. (a) The operation. (b) Circuit schematic. 89

4.8 SPICE simulation results of a 10-input WTA circuit with the dynamic current steering scheme. 91

4.9 Measured results of a 200-input WTA circuit. (a) DC characteristics of the output in the winner and another cell. (b) Transient behavior of the output in the winner. 94

4.10 Experimental results of a 200-input WTA with different number of the cells having the second largest inputs. (a) DC level of the winning output. (b) Response time of the winning output. 96

4.11 Measured results of the variation of the competition threshold for the winner. Fiver curves correspond to the output voltages of the winner with different positions of the cell having the second largest input, 2, 50, 100, 150, and 200 from the left to the right. 97

4.12 Experimental results of a 10-input WTA circuit with the dynamic current steering technique. (a) Experiment 1. (b) Experiment 2. . 98

4.13 Die photos of the WTA circuit fabricated in a 2.0-μm CMOS technology from MOSIS. (a) Enlarged die photo of the one WTA cell containing the cascaded stage. (The current steering circuit is not included in this die photo.) (b) Die photo of a 50-input WTA circuit. 99

4.14 Measured DC characteristics of the synapse cell computing the distortion or the distance measure using the linear multiplier. The square of the difference between the input and the weight value is calculated. 101

4.15 Simulated behavior of the WTA circuit for one benchmark problem shown performance improvements. The input values to the WTA circuit are obtained from the synapse matrix and the output neuron array. 102

4.16 Die photo of the self-organization neural network processor chip. There are 25 inputs neurons, 32 output neurons, and 800 synapses. 103

4.17 Block diagram of extending the dimension of the self-organization neural network processor. 104

5.1 Block diagram of the neural-based communication receiver and the inter-symbol interference (ISI) channel. 108

5.2 Block diagram of the VLSI neuroprocessor for digital communication receiver. 113

5.3 Circuit diagram of the input layer with the switched-capacitor delay line. 115

5.4 SWITCAP simulation results of the switched-capacitor analog delay circuits. (a) Node error voltages for the offset voltage of the operational amplifier being 10 mV. (b) Accumulative error voltages reflected to the output neuron. 116

5.5 Block diagram of the neural network processor system. 118

5.6 Measured waveforms of the input and the output voltages for the retrieving process. 120

5.7 Die photo of the 4-layered neural network processor chip. 122

5.8 Simulated results on the decision boundary of the neural-based receiver. (a) For the minimum-phase channel. The left portion and right portion are the decision regions for the -1 and +1 symbols, respectively. (b) For the nonminimum-phase channel. 123

5.9 Simulated results on the convergence rate of the neural-based receiver with different SNR's. (a) For the channel of $H(z) = 0.89443 + 0.44721z^{-1}$. (b) For the channel of $H(z) = 0.407 + 0.815z^{-1} + 0.407z^{-2}$. 124

5.10 Bit error rate (BER) performance. (a) For the channel of $H(z) = 0.89443 + 0.44721z^{-1}$. (b) For the channel of $H(z) = 0.44721 + 0.89443z^{-1}$. (c) For the channel of $H(z) = 0.407 + 0.815z^{-1} + 0.407z^{-2}$. 125

A.1 A portion of a complete neural network with Gaussian synapse characteristics. 141

A.2 Basic Gaussian synapse cell with single-ended input/weight values. (a) Circuit schematic. (b) SPICE simulation result. The solid line is the simulation results of the Gaussian synapse cell with the maximum magnitude of 9.14 μA, mean of 0, and standard deviation of 0.302. The dashed line is the ideal Gaussian curve. 145

A.3 Enhanced Gaussian synapse cell with differential input/weight values. (a) Circuit schematic. (b) SPICE simulation result. The solid line is the simulation results of the Gaussian synapse cell with the maximum magnitude of 13.91 μA, mean of 0, and standard deviation of 0.55. The dashed line is the ideal Gaussian curve. 146

A.4 Programmability of the enhanced Gaussian synapse cell. (a) Different amplitudes with I_X being 5 μA, 10 μA, and 20 μA. (b) Different mean values with V_W being -0.5 V, 0.0 V, and 0.5 V. (c) Different standard deviations with 0.7308, 0.5515, and 0.3768 produced by the input transistor W/L being 4 λ/ 4 λ, 8 λ / 4 λ, and 16 λ/ 4 λ. 149

A.5 An example network with four Gaussian synapse cells. 151

A.6 (a) DC characteristics of the example network. Synapse-I: mean of -1.6, standard deviation of 0.55, Synapse-II: mean of +0.2, standard deviation of 0.38, Synapse-III: mean of +1.3, standard deviation of 0.38, and Synapse-IV: mean of +2.5, standard deviation of 0.55. Output currents from four individual synapse cells are shown in the solid lines and the summed current is shown in the dashed line. (b) Speed response of the example Gaussian network. 152

B.1 Minimum input voltage difference between the winning input and the loser input. The mismatch error of a transconductance parameter β is 1 %. In the axis of *Vth variation*, ΔV_{th} changes from 0 V to 10 mV. In the axis of *common-mode input voltage*, V_C changes from 2 V to 4 V. 157

B.2 Simplified model of the proposed WTA circuit. 161

B.3 Signal and offset components due to the parasitic resistance in the common signal line versus the number of competing cells. 161

B.4 Method to extend the number of competing cells regardless of the parasitic resistor. 162

Abstract

Advances of computing systems and communication networks have made it possible to integrate the distributed information from a wide range of data fields. Integrated information processing systems process various data suchs as the image, voice, and text in order to support multi-media applications. High performance computation of data processing algorithms in real-time applications is indispensable for achieving these systems. The artificial neural network approach is one very promising method to enhance computational capabilities with rapid progresses of VLSI technologies. In order to take advantage of the fully massive parallelism of neural network computing, computations should be efficiently realized in hardware-software codesign with a neurocomputer or neuroprocessors. The analog neural computing approach with the assistance of digital control signals provides efficient implementations of high-performance artificial neural network processors with optimization on the operation speed, silicon area, and power consumption. Based upon this design methodology, key building blocks such as synapse cells and neuron cells are designed. With an industrial-level submicron VLSI technology, a fully-connected general-purpose neural chip can perform more than 30 giga-connections-per-second (GCPS). A neuroprocessor for self-organization mapping has been fabricated and evaluated, which is aimed for pattern recognition, data compression, and other signal processing applications. The high-precision winner-take-all circuit, which is a key element of the competitive learning, is designed with

performance-improving techniques such as cascading, distributed biasing, and dynamic current steering. An application-specific neuroprocessor chip has also been developed for the receiver in wireless communication. It is based on a four-layered neural network. System-level analysis and evaluation have been conducted. The accomplished research has paved an important foundation toward the construction of full-scale engineering neural systems in compact electronic hardware for scientific and biomedical applications.

Preface

The organization of this dissertation consists of the following chapters.

Chapter 1 describes implementation of the integrated information processing system which requires a significant amount of computational capabilities. Development of VLSI technologies and artificial neural network approaches have helped to process computationally-extensive algorithms. Hardware implementation of artificial neural network models, neuroprocessors, can be achieved based on careful consideration of various design issues.

In *Chapter 2*, various hardware implementations of artificial neural networks are reviewed. The analog signal approach is compared with the simple conventional digital computing system. Several techniques to implement analog neural networks are presented in terms of storing reliable synapse weight values. Advantages and drawbacks of each method are summarized. Special design techniques and applications for analog neural networks are introduced.

The detailed design of the proposed neural network is presented in *Chapter 3*. Input/output neurons and synapse cells are basic building blocks of the analog neuroprocessors. Design and operations of each building block are described and some methods are provided to overcome the problems occurred in practical situations. The general-purpose neural network which can support programmability and reconfigurability are described.

In *Chapter 4*, the self-organizing neural network is presented for pattern recognition and data compression. Implementation of the competitive learning algorithm must heavily depend on winner-take-all (WTA) operation. Design of the WTA circuit, which determines the performance of the network, is described in detail with several performance improvement schemes. Experimental results show high-accuracy operations with a fast response time and feasibility to extend the number of competing cells over than 1000.

Chapter 5 presents one example of system implementation of neural network hardware. The receiver for digital communication is designed to handle the inter-symbol interference (ISI) and white Gaussing noise. The structure of such a neuroprocessor is optimized to a four-layered network with the switched-capacitor analog delay line.

Finally, the summary of the foregoing research is concluded in *Chapter 6*. Full system-level evaluation and suggestions for the future work are included.

In *Appendix A*, the synapse cell performing the Gaussian function between the input and synapse weight values are described. Gaussian synapse cells are used to build the radial-basis function (RBF) neural networks in order to reduce the convergence time compared with the conventional back-propagation neural network.

Appendix B presents the nonideal effects which degrade the performance of the proposed WTA circuit - device mismatches of input transistors and parasitic resistance on the common signal line. Methods to overcome the problems are also suggested.

Chapter 1

Introduction

1.1 Integrated Information Processing System

Advances of computer systems and interconnection networks make it possible to process or manipulate information from widely distributed sources in a wide range of data fields. Different kinds of media such as the image, voice, and text are integrated among the information processing systems supporting the multi-media capability [1]. Such a powerful multi-media machine can be used in many places and will help people in their business, education, and personal lives [2]. The real-world signals are interfaced with such a system through the video and audio channels as shown in Fig. 1.1. Various sensors receive the information as the inputs. The pictorial data are acquired through the camera, image sensor, motion detector, and so on. The acoustic data are obtained by the microphone, the sound locator, and so on. These incoming data are processed in the main information processing systems. The pictorial data are processed by various computations such as image processing algorithms, pattern recognitions, or motion detection. The acoustic data are processed through speech recognition, speech synthesis, or sound recognition. The processed information appears as the system output in the

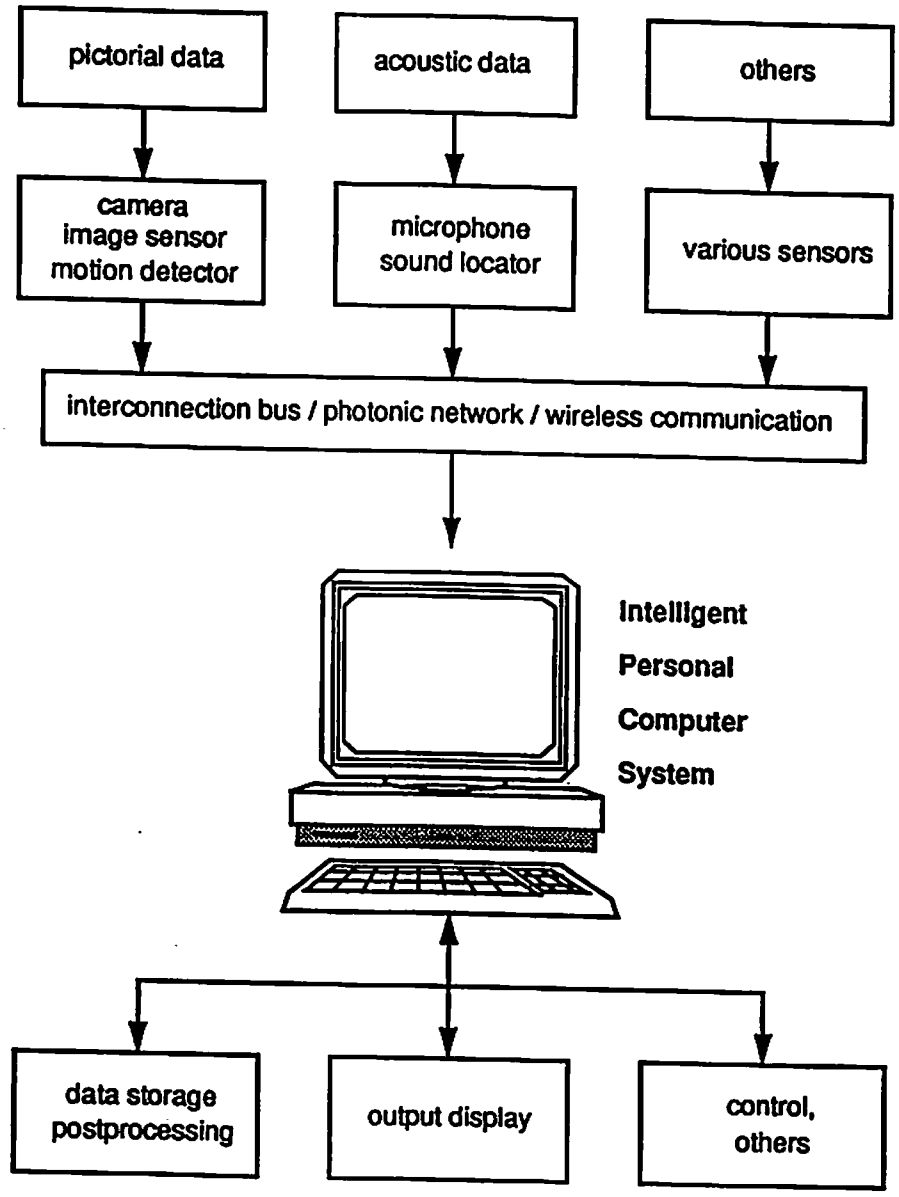


Figure 1.1: Configuration of the integrated information processing system.

forms of displayed images, sound and/or speech signals, and the physical control of movable parts.

High-speed execution of the data processing algorithms are indispensable for achieving the above-mentioned information processing system in real-time application. In the case of a high definition television (HDTV) [3], more than one billion operations per second is needed for the vast amount of video data. The recent U.S. government report in 1992 showed that the intensive computational power is strongly required for future information processing applications. A high-performance system is anticipated to be able to support tera operations-per-second in 1996 [4]. Such a high computational speed is only is possible through further advances of the very large scale integration (VLSI) technologies. Some algorithm-specific multiprocessor chips can be developed to fully exploit the inherent computational powers of various information processing architectures.

1.2 Development of VLSI Technology

Rapid advances of VLSI technologies have made it possible to integrate several million transistors on a single chip. The state-of-the art device feature size for the memory chip in the year of 1993 is around 0.25 μm, and that for the microprocessor chip is around 0.45 μm. The expected feature size for the future VLSI technology will be 0.1 μm - 0.15 μm in the year of 2000 [5]. The extensive use of VLSI circuits can greatly reduce the size of electronic systems because the time-consuming algorithms can be efficiently executed in hardware to enhance the data throughput. In addition, the performance and reliability of microelectronic systems can be improved. Future development of VLSI hardware will be inspired by many innovative research results in data processing architectures.

Performance of microprocessors has been continuously increased for the pase decade. Results from industrial development in microprocessors are summarized in Table 1.1. Sub-micron CMOS or BiCMOS technologies enable the fabrication of IC chips with die size being larger than 1 cm^2, and running at the speed of several hundred mega-floating-operations-per-second (MFLOPS) in 64-bit data representation. The increased number of transistors in the monolithically integrated signal-processing chips is shown in Fig. 1.2 [5]. A highly-integrated chip will be expected to incorporate as many as 50 million transistors in the year of 2000. The density for dynamic random access memories (DRAM's) has quadrupled every three years since their advent about 20 years ago [6]. Trend for the DRAM development is shown in Fig. 1.3. In 1993, a 256-Mb DRAM from a 0.25-μm CMOS technology can run at the access time of 30 $nsec$.

1.3 Artificial Neural Network Approach

The artificial neural network is one of many approaches to enhance the computational capabilities in high-speed information processing. Hardware accelerators such as the SAIC Sigma-1 [7] and the HNC ANZA [8] have enabled the simulations to be carried out over 20 times faster than the regular engineering workstations.

Unlike the conventional approaches to increase the computational power of a single processor, the artificial neural network uses a large collection of simple processing elements which are highly interconnected as shown in Fig. 1.4.

Inspired by the physiology of the human brain and the study of the brains of living animals, these simple processing elements can perform mathematical algorithms to carry out the information processing through their responses to

Table 1.1: Summary of the microprocessors designed in the 1990's.

YR.	CHARACTERISTICS	SPEED	TECHNO-LOGY	CHIP SIZE [mm^2]	#TRAN-SISTORS	POWER [W]	PACK-AGE	MAKER
90	32b 68040	50(in), 25(out) MHz	1.0-mm 2M CMOS	14.4 x 15.5	1.2 M	-	179 PGA	Motorola
90	64b RISC	40 MHz, 40 MIPS/20 MFLOPS	0.8-µm 2M CMOS	14.85 x 15.13	1.0 M	4.0	238 PGA	Matsushita/ Solbourne
90	32b data, 200b instr. VLIW	50 MHz	1.5-µm 2M CMOS	8.0 x 10.0	77 K	-	224 pads	Philips
91	32b 80486	100 MHz	0.8-µm 3M CMOS	6.8 x 11.8	1.2 M	8.0	-	Intel
91	32b super pipelined, dsta-driven	50 MFLOPS	0.8-µm 2P 2M CMOS	14.65 x 14.65	700 K	4.0	281 PGA	Mitsubishi/ Osaka Univ.
91	64b fl. pt. coproce-ssor for RISC	65 MHz 33.2 MFLOPS	0.8-µm 2M CMOS	12.75 x 13.16	640 K	2.3	-	Texas Inst./ Hewlett-Packard
91	64b superscalar with DSP enhancement	50(in), 25(out) MHz 100 MIPS	0.8-µm 2M CMOS	13.0 x 13.0	1.1 M	-	223 CPGA	National Semi-conductor Corp.
92	macropipelined CISC	100 MHz 400MIPs, 200MFLOPS	0.75-µm 3M CMOS	16.2 x 14.6	1.3 M	16 - 18	339 THPGA	Digital Equip-ment Corp.
92	64b dual-issue	200 MHz	0.75-µm 3M CMOS	13.9 x 16.8	1.68 M	30.0 @ 3.3 V	431 PGA	Digital Equip-ment Corp.
92	superscaler	50 MHz	0.8-µm 3M BiCMOS	15.98 x 15.98	3.0 M	9.0	283 PGA	SUN Microsystems
92	single-chip supercomputer	70 MHz 289 MFLOPS	0.5-µm 3M CMOS	15.75 x 16.0	1.5 M	5.0 @ 3.3 V	-	Fujitsu
92	superscalar	250 MHz 1,000 MIPS	0.3-µm 3M CMOS	8.1 x 8.0	1.02 M	-	-	Hitachi
93	32b Bipolar ECL microprocessor	300 MHz	1.0-mm 5M Bipolar	15.4 x 12.8	670 k	115 @-5.2 V	504 PGA	Digital Equip-ment Corp.

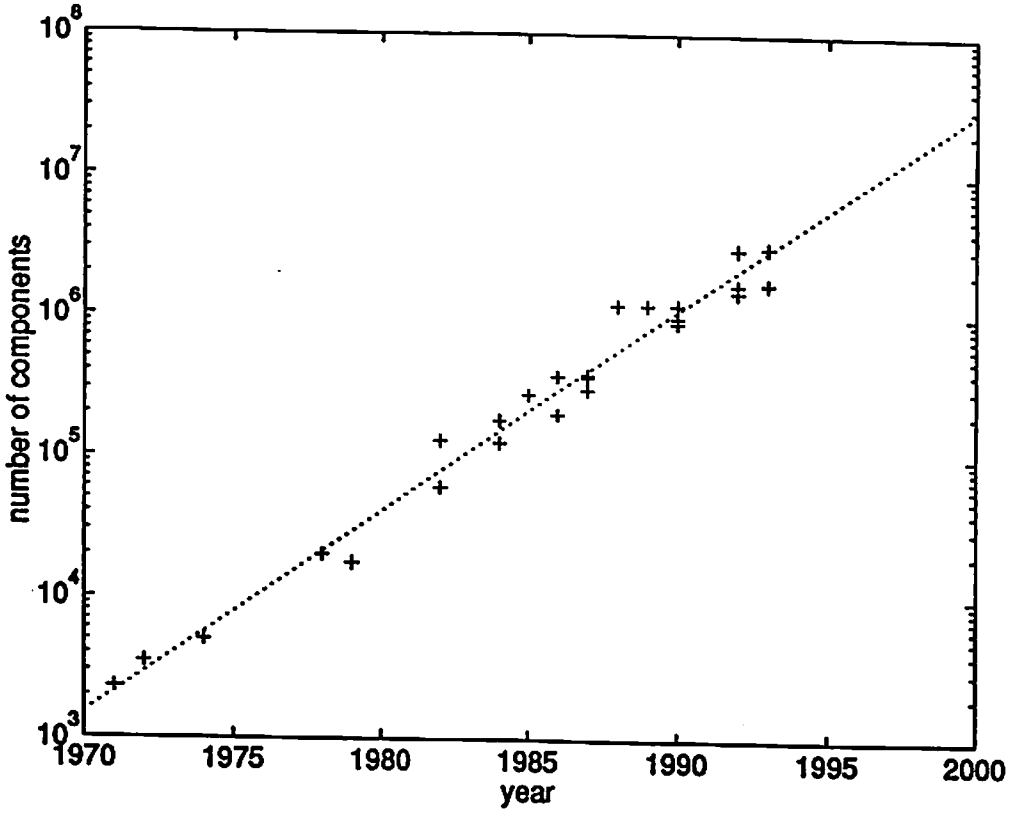


Figure 1.2: Number of transistors incorporated in the integrated logic chips.

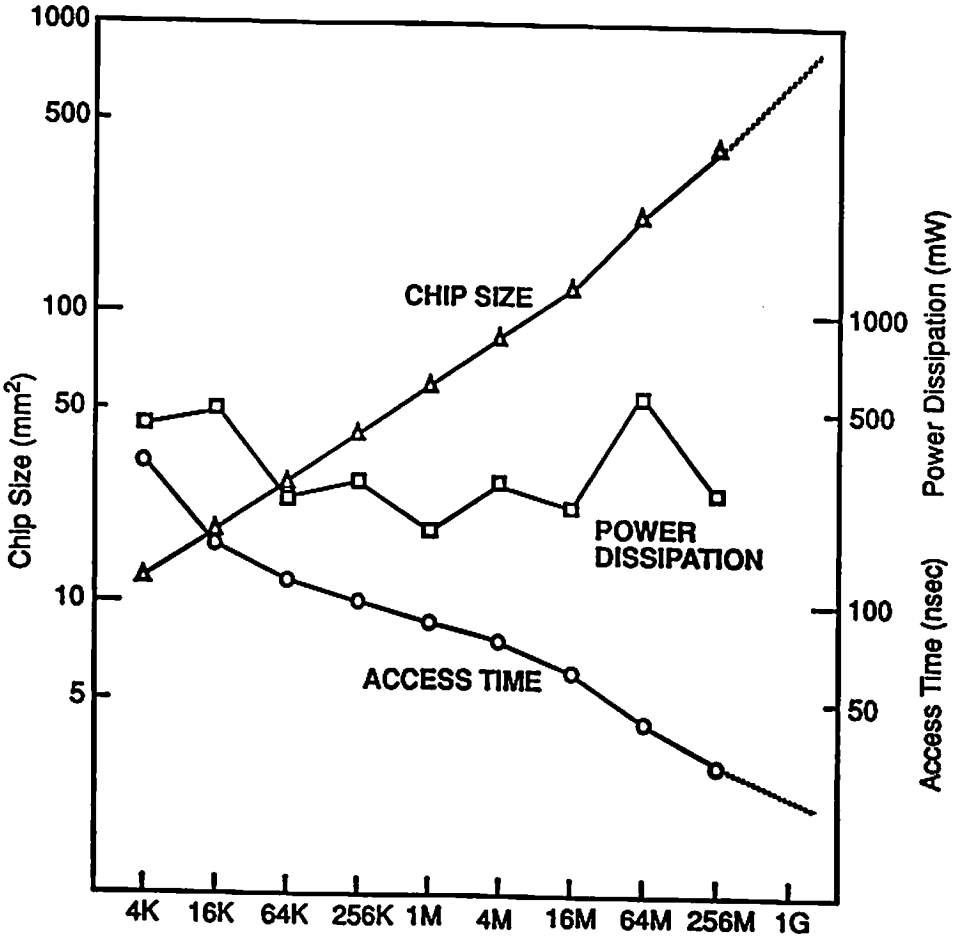


Figure 1.3: Trends in standard DRAM development.

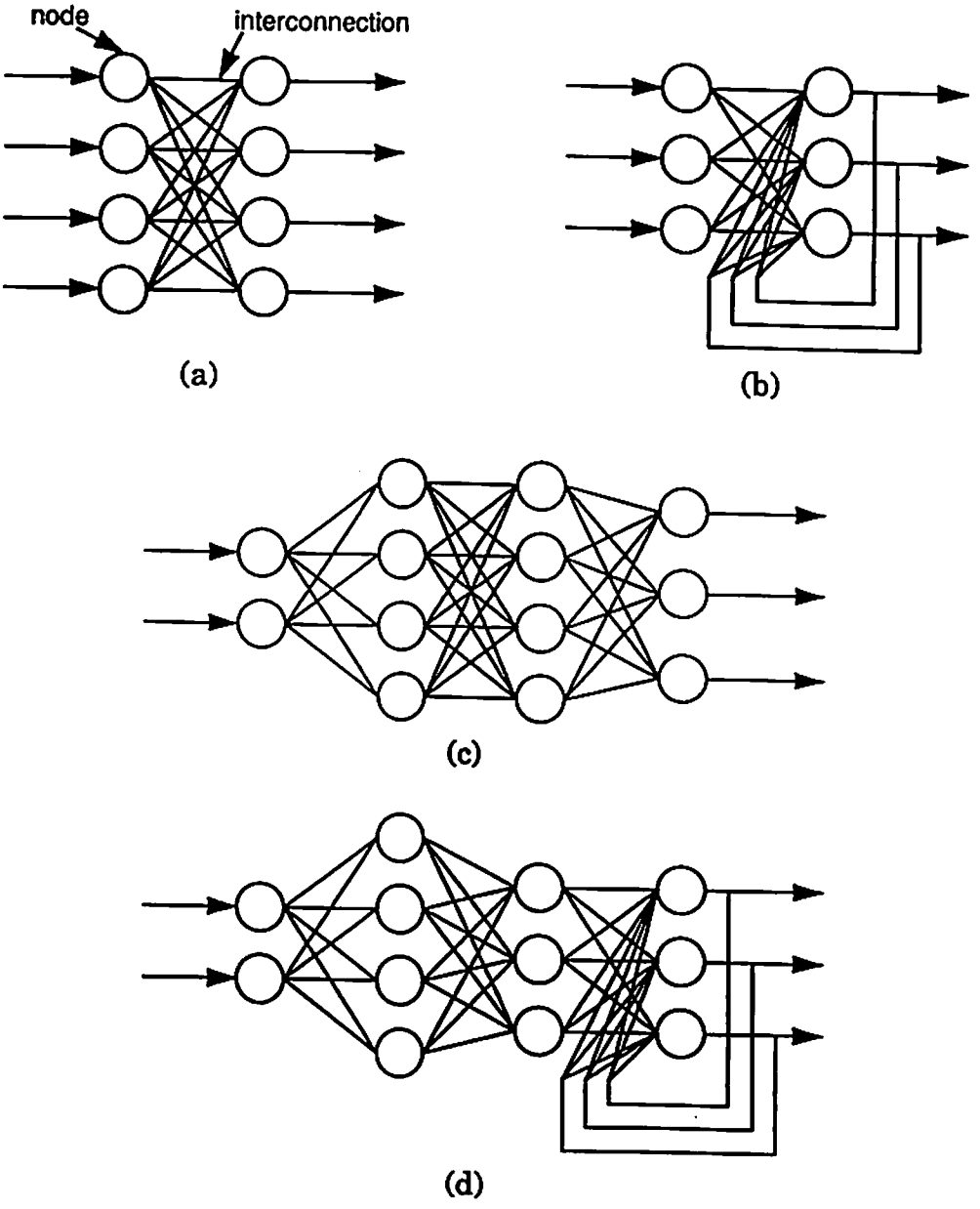


Figure 1.4: Several configurations of the artificial neural network consisting of the neurons and the interconnections. (a) Single-layered network. (b) Recurrent network. (c) Multi-layered network. (d) Multi-layered network with feedback.

stimuli. Artificial neural networks have demonstrated the ability to provide superior, powerful solutions to problems that have challenged conventional computing approaches for the past many years.

There are several situations where neural networks are advantageous [9]:

- Only a few decisions are required from a massive amount of data to be processed.
- Nonlinear mapping must be automatically acquired.
- A nearly optimal solution to a combinatorial optimization problem is required very quickly.

Based upon these advantages, the artificial neural networks are suitable for the following applications:

- **Classification:** an input pattern is applied to the network, and the representative class appears as the output.
- **Pattern Matching:** an input pattern is passed to the network, and the network produces the corresponding output pattern.
- **Pattern Completion:** after the incomplete input pattern is applied, the network produces the output pattern that includes the missing portions of the input pattern.
- **Noise Removal:** the network receives a noise-corrupted input data and produces the clearer version of the output with removal of the noise.
- **Optimization:** an input pattern representing the initial values for a specific optimization problems is presented to the network, and the network produces a set of variables that represents a solution to the problem.

- Control: an input pattern represents the current state of a controller and the desired response for the controller, the proper sequence of commands is obtained as the network output which will produces the desired response.

According to the network topology and the types of the learning and retrieving processes, many different classes of artificial neural networks exist [10, 11, 12] as shown in Fig. 1.5. Some of the key network is summarized in Table 1.2. The artificial neural network is used in various scientific and engineering applications such as the machine vision, speech and pattern recognition, robotics, control, telecommunication, financial analysis, and expert systems.

1.4 Hardware Implementations

To implement a high-speed artificial neural network system, it is essential to develop the neurocomputing processors. It is specially optimized for the artificial neural algorithms. Such systems employ parallel processing architecture in order to increase the throughput.

Some challenging design factors for implementing the neural system are [13]:

- to work for one or several neural algorithms;
- to contain the maximum number of neurons, which can be monolithically integrated;
- to store programmable weights reliably;
- to support the learning schemes;
- to occupy small-area neurons and synapses;

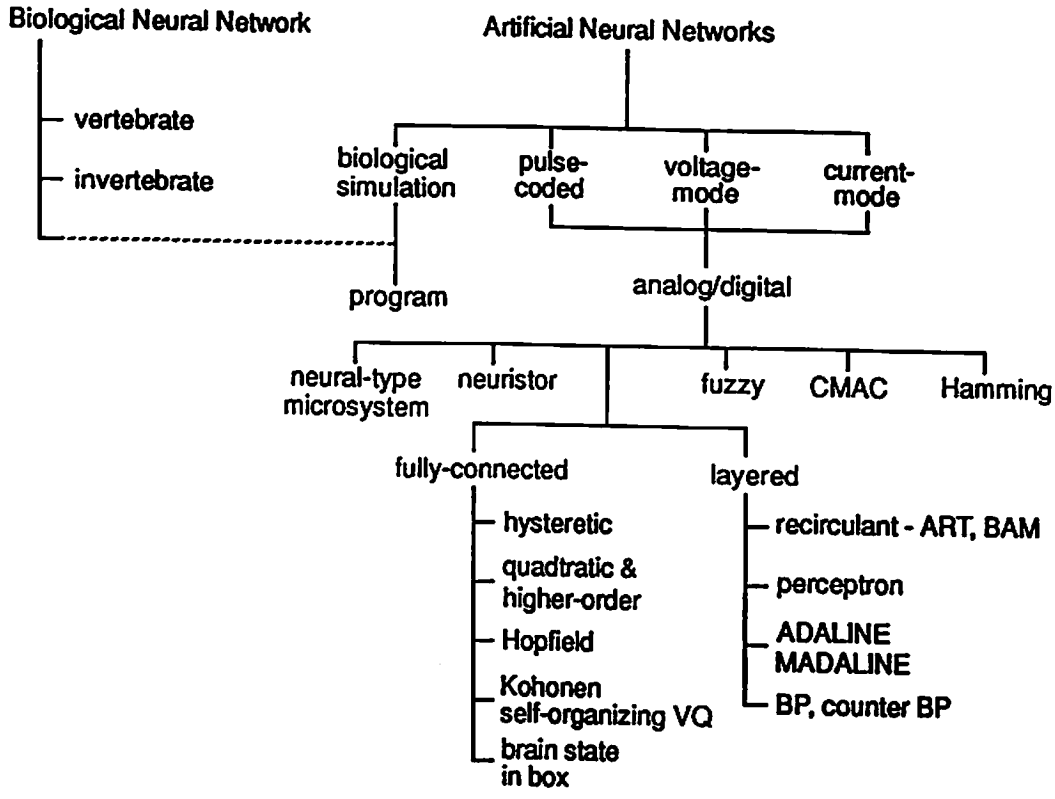


Figure 1.5: Various types of artificial neural networks.

Table 1.2: Major artificial neural network models and their properties.

neural models	strengths	limitations	primary applications
perceptron	simple structure	cannot recognize complex patterns	typed-character recognition
Hopfield	simple structure	weights must be pre-set	retrieval of data/ images from fragmants
multi-layer percetron with delta rule	more genral than the perceptron	cannot recognize complex patterns	pattern recognition
back-propagation	most popular, work well, and is simple to be learned	required for large volume of examples in advance	wide range: speech synthesis to loan application scoring
Boltzmann machine	simple network using noise function to reach global minimum	long training time	pattern recognition for radar/sonar
self-organizing map	batter performance than many algorithmic techniques	extensive learning	mapping one geometrical region onto another
cellular neural network	scalability to large network size	no formal known method to adapt weights	2-dimensional pattern processing

- to consume a low power;
- to be stable, reproducible, and extendible so that larger systems can be built by direct interconnecting neural network building blocks;
- to be affordable in manufacturing cost.

1.4.1 Analog (or Mixed), Digital, and Pulse Signals

According to the types of signals to be processed in neurocomputing, neural network processors are classified as using the analog (or mixed-signal) data representation, digital representation, and pulse-modulated data representation.

The analog or mixed-signal neural network processors have been widely envisioned and preferred for the nature of analog signals in efficiently processing the neurocomputing functions. Various circuit techniques are used for analog multipliers as synapse cells and the summation is normally done in the current signal format by the Kirchoff's current law. Since the power dissipation and the silicon area of the constituting elements is small, a large number of such components can be easily integrated. The accuracy of synapse weights, however, is limited to a moderate value due to the imperfect analog devices.

Digital neurocomputers [14, 15, 16, 17, 18, 19, 20, 21] are based upon the multiprocessor configuration. Each processing element consists of a digital multiplier, digital adder, and the digital memory. Many processing elements are connected through the multiple bus and the data transfer among them is efficiently controlled. The network is fully programmable and reconfigurable and it supports the on-line learning algorithms. The operational accuracy is proportional to the usage of the hardware resource.

The last signal representation in the neural network processors is the pulse trains in the pulse-modulated neural networks [22, 23, 24, 25, 26, 27, 28]. The operation is based upon the probabilistic properties and the number of the pulses to be fired is the key information to be processed. While the input and output signals are pulse signals, the intermediate operations can be in analog representation in the forms of electrical charges and the capacitor.

1.4.2 General-Purpose or Application-Specific

Designs of neural processor chips are directed into two approaches for general-purpose and the application-specific neural network applications. The general-purpose neuroprocessor chips are created for various applications. The design process basically follows the conventional standard-cell approach. Each building block is well characterized and made in high-precision which can be fit into the various applications. A maximum number of the building components is put together as long as the hardware resource allows. The constructed system must be controlled for reconfigurability of the network topology and programmability of the network operation.

In the application-specific neural networks, the configuration of the network is optimally customized by the designated purpose. The network topology is determined before the design and the synapse values are fixed in usual. Trade-off factors among the accuracy, the speed, the power dissipation, and the resource amount should be considered for optimal design. Some special circuitry can be included before the input layer and/or after the final output layer of the neural network.

1.4.3 Supervised or Unsupervised Learning

The strongest advantage of the neural network is its ability of learning according to the response to the applied stimuli and the changing environments. Thus, learning makes the neural system be powerful in the adaptive signal processing applications and be robust with respect to the undesired faults. Both the supervised learning and the unsupervised learning are supported by implementation of VLSI neural network processors.

In the supervised learning [29], a pair of the input and the desired response of the network is appled at each time when the learning algorithm is processed. The difference between the desired response and the actual output is used as the error signal for the learning algorithm. Back-propagation learning is popularly used in the various scientific and engineering applications due to its well-established characteristics. The neural network hardware implemented for the general purpose can usually support this supervised learning.

In the unsupervised learning [12], the desired responses of the neural network are unknown in advance. The synapse updating process evolves itself spontaneously according to the computed strengths between the neuron nodes. Several self-organizing neural networks have been implemented which support the competitive learning.

1.4.4 On-Chip Learning

When the learning algorithms are implemented with the companion DSP board or the dedicated DSP processor chip, the genuine operations of the weighted-summation and nonlinear transformation are used in the neural network processor.

The processor performs only retrieving or recall process. Since the implementations of the learning rules are fully programmable, various learnings can be used in this approach. In addition, the required operational accuracy is not restricted due to the limited hardware resource.

On the other hand, the circuitry for learning can also be implemented on the same chip for the on-chip learning. Both feedforward operations and feedback operations are included. Most digital neurocomputing processor supports the on-chip learning [30]. A lot of researches have also been done in the analog signal representation for their simplicity of implementations. In the case of the analog neural network implementation, due to the lack of data transfers between the chip and the off-chip supporting devices, the complicated interface such as the analog-to-digital conversion and the digital-to-analog conversion is not needed. Since the accuracy, however, is limited by the non-ideal effects of the chip due to the fabrication, some learning algorithms which require very high resolution cannot be realized.

Chapter 2

Design of Analog Neural Network Processors

2.1 Analog Neural Computing

In order to exploit all of the speed and fault-tolerance advantages inherent in the full-parallelism of neural networks, we need to implement them on a fully parallel architecture. Like any other processing system, a neural network consists of basic building elements for the computation and communication. Except inr the pulse-stream signal representation approach, the data having the multi-level values are processed in the analog domain or in the digital domain. The choice between analog and digital processing of these individual elements can be made independently for the different subsystems, with the purpose of optimization in terms of silicon area, speed, accuracy, and power consumption for the whole network.

The basic neural computation consists of multiplications between the input signals and the synapse weights and summations of these products. The operational behaviors of this computation are discussed with respect to the analog neural networks and digital arithmetic units. Several design issues such as the

computational time and accuracy as well as the consumed silicon resource are considered.

Figure 2.1(a) and (b) show the typical weighted summations performed in the digital arithmetic unit and the analog neural network, respectively. In the analog signal scheme, a large number of the multipliers are distributed forming the matrix of the synapse cells. Each synapse cell stores the weight value and multiplies this value with the applied input producing the output current. For each neuron, a simple current-summing node is used for the summation according to the Kirchhoff's current law.

In a typical digital processor, the processing element (PE) consists of a parallel multiplier, an adder used as the accumulator, and the digital memory. The multiplier fetches the operands from the input data buffer for the input signals and from the memory for the synapse weighted values. In each operation cycle, one multiplier and one accumulator are used to produce the weighted summation. Extensively iterative uses of these devices are needed in order to process the large amount of data for neural computations because a limited number of processing elements can be integrated in a signle chip.

For comparisons on both approaches, a single layer is considered which consists of an array of M input neurons, an array of N output neurons, and a matrix of M-by-N fully connected synapses. The following definitions are used,

- Nm_A: the number of the available analog multipliers per system (chip or board);

- Nm_D: the number of the available digital multipliers per system (chip or board);

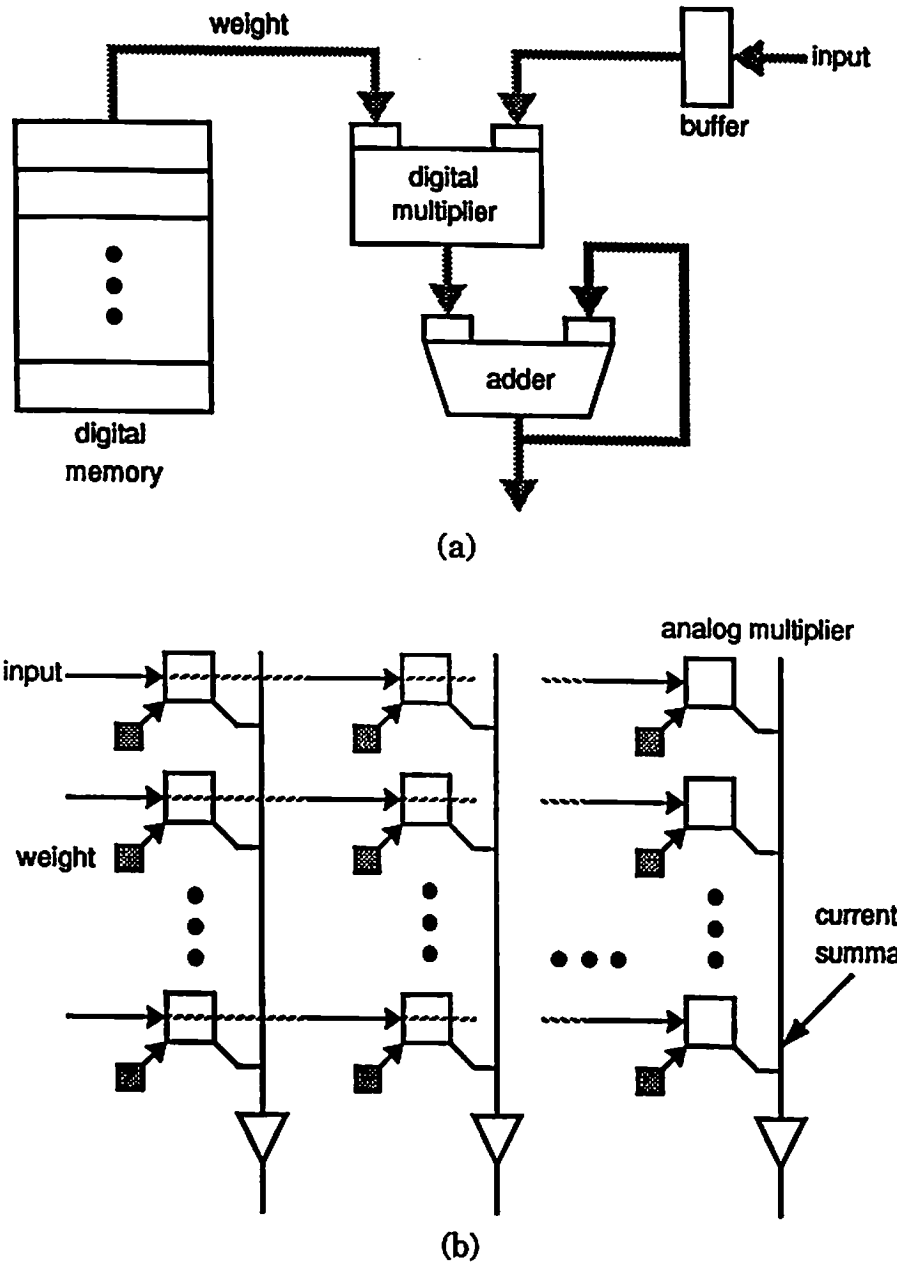


Figure 2.1: Block diagram of the processors for neural computation. (a) Digital arithmetic unit. (b) Analog neural computing processor.

- Na_A: the number of the available analog adders per system (chip or board);
- Na_D: the number of the available digital adders per system (chip or board);
- Tm_A: the operational time of the available analog multiplier;
- Tm_D: the operational time of the available digital multiplier;
- Ta_A: the operational time of the available analog adder;
- Ta_D: the operational time of the available digital adder.

The speed-up factor of analog implementation over digital implementation is defined as the ratio of the total execution times in both approaches as follows,

$$F_{speed} \equiv \frac{Texec_D}{Texec_A} \simeq \frac{\max(\frac{M \cdot N}{Nm_D}, \frac{M \cdot N}{Na_D})}{\max(\frac{M \cdot N}{Nm_A}, \frac{N}{Na_A})}. \tag{2.1}$$

Considering the normal condition, F_{speed} can be simplified to,

$$F_{speed} = \left(\frac{Nm_A}{Nm_D}\right)\left(\frac{Tm_D}{Tm_A}\right). \tag{2.2}$$

The high speed-up factor of the analog implementation approach mainly results from the fact that a significantly large number of components can be integrated. For example, when Nm_D and Nm_A are 2 and 1000, respectively, the speed-up factor of the analog implementation over the digital implementation reaches 25 if the clocking frequency of the digital processor is 200 MHz and that of the analog processor is 10 MHz.

Power disspation is one of the important factors which makes the analog signal approach be preferable in neural network implementation. Power consumed in one digital gate such as an inverter can be expressed as,

$$P_{gate} = C_L(V_{DD} - V_{SS})^2 f_{clk}, \tag{2.3}$$

where C_L is the load capacitance and f_{clk} is the operating clock frequency of this gate [31]. When $C_L = 100fF, V_{DD} = 5V, V_{SS} = 0V$, and $f_{clk} = 100MHz$, the power dissipation per gate is 0.5 mW. In one implementation example with an 8-bit fully-parallel multiplier and an 8-bit the Manchester carry-chain adder, the numbers of the gates are about 350, and 150 respectively. Thus the total power dissipation required for the basic weighted summation is 250 mW. On the other hand, the total power dissipation of the combination of the input neuron, the synapse cell, and the output neuron is about 50 mW. Large power consumption makes the digital VLSI neural network undesirable for portable applications.

The main drawbacks of analog neural networks are the lack of very accurate computing devices and programming flexibility. While the operational accuracy of the digital network is linearly proportional to the word-length of the network, there are limitations on the precision which an analog computing block can achieve as shown in Fig. 2.2. The analog neural network is best utilized for processing the feedforword signals in the retrieving phase. Since the learning algorithms requires a high-resolution computation, the analog neural network cannot easily support the on-chip learning in the pure analog signal domain. The analog network is inferior to the digital one in terms of the flexibilities such as the configurability, controllability and the programmability.

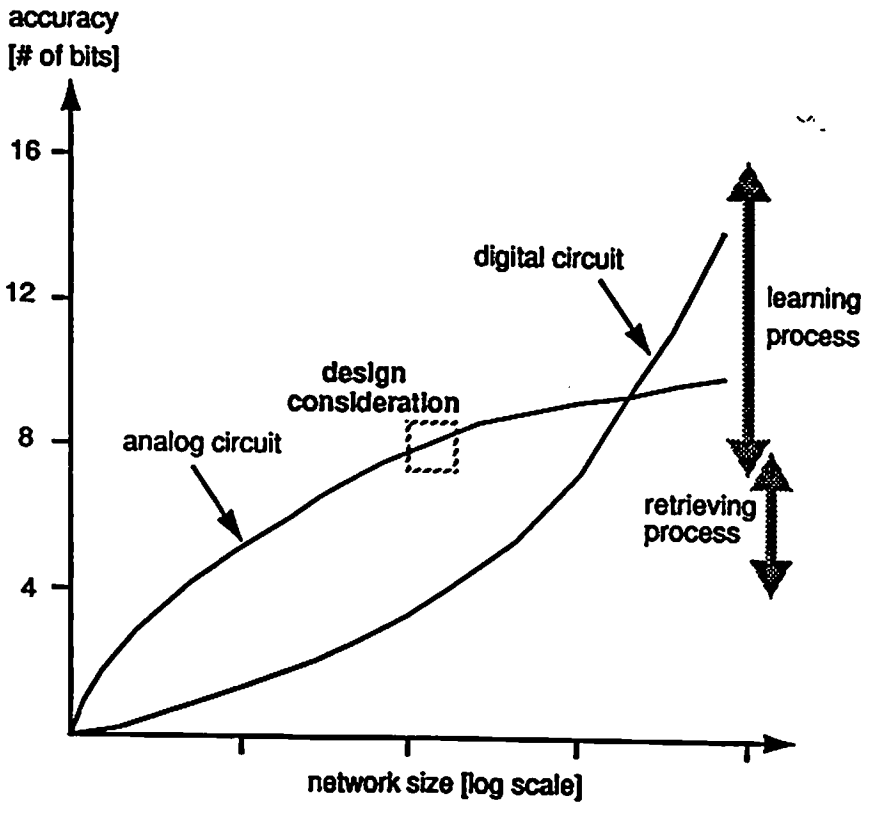


Figure 2.2: Conceptual diagram of the operational accuracy in the number of the bits with respect to the network size.

The analog neural network chip is not suitable for an independent system. It is ideal for serving as a co-processor to handle the computationally intensive tasks. Learning and controlling schemes are processed in the digital processor or the host computer, of which outputs are interfaced with the analog neural network chip through the data conversion modules. On the other hand, the digital neural network can be realized in digital signal processor chips or executed on the computer.

Three different approaches are compared for realizing the neural computing algorithms which requires 32 giga-connections-per-second (GCPS). Performances of the advanced reduced-instruction set computer (RISC) microprocessor [32], the digital signal processor [33] and a high-performance analog VLSI neural network processor are studied. Summary of the comparison is listed in Table 2.1. Here, the performance of the analog neural processor in a 0.8-μm technology is estimated from the results of prototyping building blocks fabricated in a 2.0-μm CMOS technology.

2.2 Various Synapse Storage Scheme

There have been many researches on the storage of synaptic weight values in the analog neural networks. In this section, comparison of several main storage schemes which are quite promising for implementing analog neural networks are considered. The desired features of the synapse cells are to provide multiplication and a reliable, long-term storage of the synapse value.

Table 2.1: Performance comparison of three different approaches to achieve the 32 GCPS for neural computation.

	microprocessor	digital signal processor	analog neural processor*
characteristics	dual-issue microprocessor chip	vector-pipeline arch. DSP chip	g-p analog neural net. processor chip
technology	0.75-μm 3M CMOS	0.8-μm 2M CMOS	0.8-μm 2M CMOS
operational speed	200 MHz 400 MIPS 200 MFLOPS	60 MHz 60 MIPS 2 GOPS	2.5 -5 MHz 32 GCPS
accuracy	64 bits	16 bits (fixed)	7-8 bits
chip size	16.8 x 13.9 mm^2	12.38 x 12.90 mm^2	10 x10 mm^2
power dissipation	30 W @ 3.3 V	2.4 W @ 5 V	5 W @ +/-5 V
# of transistors	1.68 M	930 K	160 K
Target Connection	32 GCPS		
# of chips required	60	12	1
power consumption	1800 W	28.8 W	5W
total silicon area	14011 mm^2	1916 mm^2	100 mm^2
# of transistors	101M	11.2 M	160 K

* The specification of the analog neural processor in a 0.8-μm technology are estimated from the fabricated chip in a 2.0-μm CMOS technology.

2.2.1 Dynamic Capacitor

The analog synapse cell with the capacitance for weight storage has the main advantage of full functionality over other approaches [34, 35, 36, 37]. The mature designs of analog multipliers have been used with capacitors which are easily implementable in the CMOS technology. The use of capacitance to store synaptic weights make the synapse cell very compact. The synapse cell provides four-quadrant multiplication with a large operational range.

Figure 2.3 shows two examples of the synapse cell using the capacitors for weight storage. Figure 2.3(a) shows the circuit schematic of a modified Gilbert multiplier [36]. The differential output current is used for the synapse output current. The circuit is capable of four-quadrant multiplication between the input signals and the synapse values. In Fig. 2.3(b), four matched transistors biased in the triode region are used to cancel the nonlinear term of $(1/2)V_{DS}^2$ in the drain-current expression [38]. The output currents are obtained from the difference of two current branches.

The voltage on the capacitor is continuously decayed due to the leakage current through the reverse-biased *pn*-junctions between the diffusion area and the substrate of the pass transistors. By using the fully differential scheme as shown in Fig. 2.3, the effect of the leakage current can be eliminated to the first-order. In order to maintain the weight value for a long term, the refreshing scheme should be included. This refreshing scheme increases the circuit complexity for the on-chip method [39, 40] or the interface complexity for the off-chip method [34, 41, 42, 43].

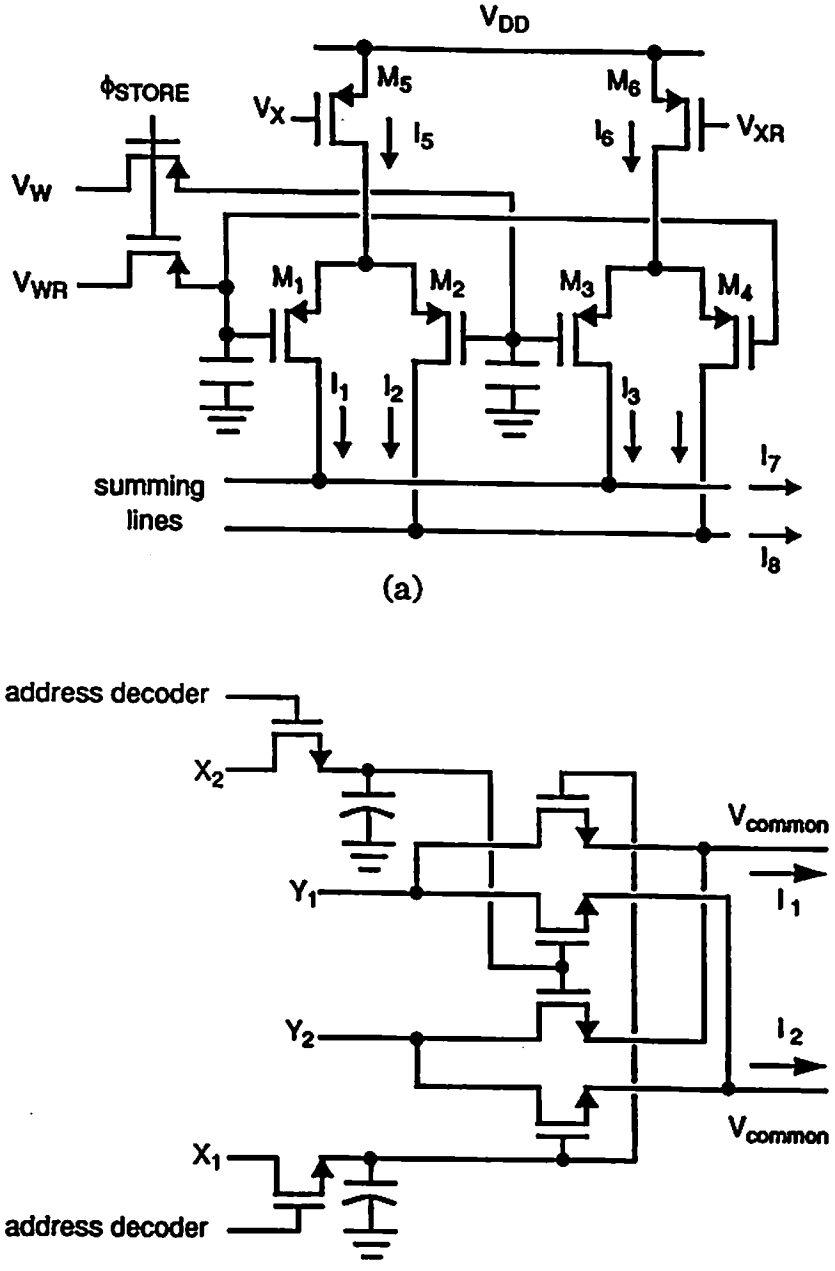


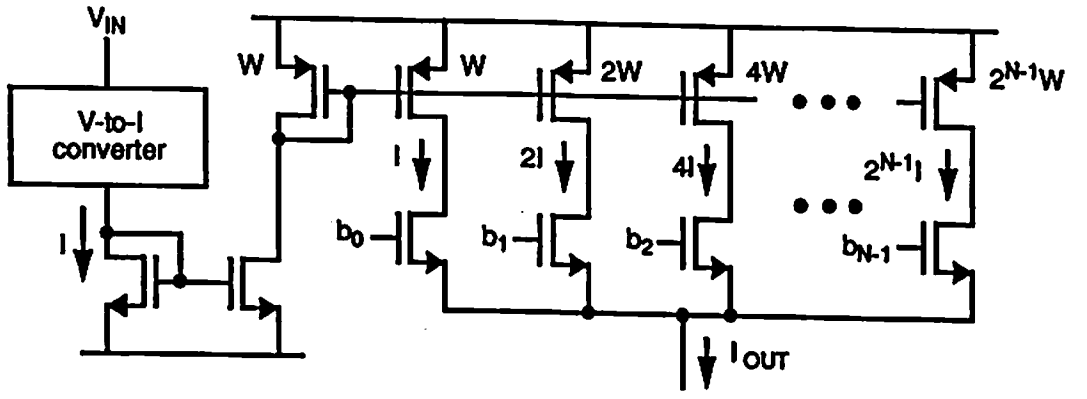
Figure 2.3: Circuit diagrams of the synapse cells with dynamic capacitor storage. (a) Using the modified Gilbert multiplier [36]. (b) Using the four matched transistors [38].

2.2.2 Digital Memory

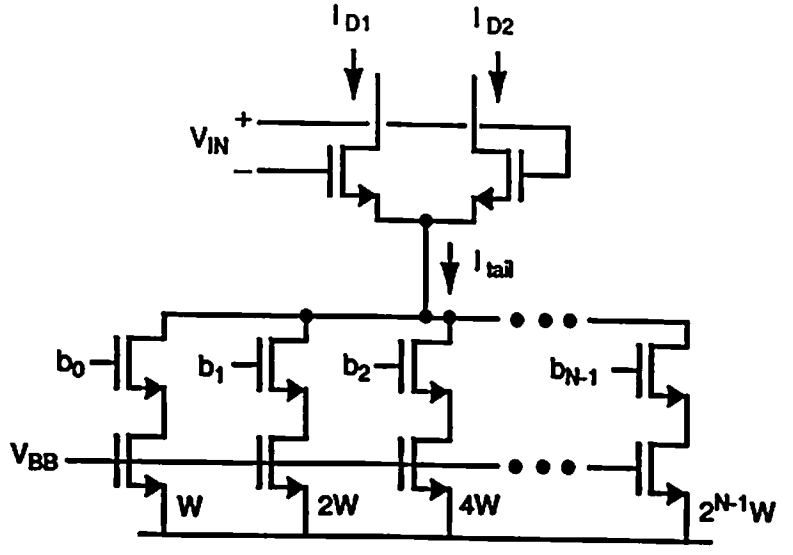
The synapse value can be stored in the internal D-flip flops to represent the weight value by the digital signals. These synapse cells are again divided according to the number of the bits used in the memory. The single-bit synapse cell includes the one D-flip flop and relevant switches generating only ON-current or OFF-current for weighting operations [44, 45]. The multi-bit synapse cell consists of multiple D-flip flops and a digital-to-analog converter so that the analog values can be provided as the synapse output in proportional to the stored digital weights [46, 47, 48, 49].

In the multi-bit synapse cell, the digital-to-analog converters can be constructed with passive elements such as resistors and capacitors or the active devices such as transistors. The active devices are widely used since their performance are better than the passive counterparts in terms of the occupied area and accuracy. By using the binary-scaled current mirror, the CMOS transistor circuitry easily provides the binary-weighted currents as shown in Fig. 2.4(a) [47]. The contribution from each binary current branch is controlled by the associated switch which is turned on or off according to the content of the memory. The above-mentioned current sources (or sinks) can be also used to provide the tail current in the differential pair as shown in Fig. 2.4(b) [48].

Main advantages are that treatment of the synapse weight is quite easy because the synaptic representation is directly in a digital signal format. The digital output from a main processor is directly applied to the network without digital-to-analog conversion. D-flip flop is usually used due to simple operation without refreshing and some simple manipulations are made for on-chip learning [49].



(a)



(b)

Figure 2.4: Circuit diagrams of the synapse cells with the digital memory schemes. (a) Binary-weighted current mirrors are used for directly providing the output current [47]. (b) Binary-weighted current mirrors are used for producing the tail current of the differential pair [48].

The occupied area, however, becomes quite large since the size of the current mirror transistor increases by 2^N, where N is the number of bits. In addition, the unit transistor size should be large to minimize the device mismatch effects across the entire chip. In the current mirror synapse cell, the area of the synapse cell is doubled to provide the positive and negative currents. The synapse cell using the binary-weighted tail current has severe functional limitations. The linear operation range is proportional to the amount of the weight value so that the early saturation can occur for a large input signal which will reduce the achievable accuracy of the enitre network. Additionally, the multiplying operations are performed only in two quadrants.

2.2.3 Floating-Gate Analog Memory

The charge in the capacitance formed by the MOS transistor is decayed due to the leakage current through the diffusion-substrate junction. In addition, the dynamic capacitor memory and the digital flip-flop will suffer from data loss when the power of the system is turned off so that there might be complicated up-loading and down-loading operation in setting up the network through non-volatile medium such as hard-disks. This problem can be avoided by using the floating-gate analog memory [50, 51, 52, 53, 54, 55]. The main advantage of the floating-gate storage technique is its capability of very long data-retention time, which can be longer than 10 years, at the room temperature [56].

By varying the pulse width and the pulse numbers, the threshold voltage of the transistor can be modified. This is an effective method for programming the synapse values. Two examples to illustrate this method are shown in Fig. 2.5. In Fig. 2.5(a), the threshold voltage of one transistor in the differential pair is

changed so that the current difference appears as the synapse output current [53]. In Fig. 2.5(b), the threshold voltage of a transistor providing tail current in one differential pair is different from that of the other differential current [54].

Main drawbacks of this method are the requirement of special silicon fabrication technology and the difficulty of precise programming. The floating gates can be implemented only in the double-polysilicon technology or specialized EEPROM technology. Since programming is done through controlling various parameters of the programming pulses, some complex pulse-generator is necessary. In addition, the magnitude of the programming pulse is usually much larger than that of the power supply voltage. The electrical programming of synapse weight values is mainly a stochastic process. Therefore the achieved results cannot be very precise and is typically in the range of 5-6 bits accuracy.

2.2.4 Summary

Summary of the three approaches is shown in Fig. 2.6, where comparison is done in four aspects. The **functionality** criterion determines how good the synapse cell performs multiplication in terms of the linearity and operating range. The limitation due to poor functionality restricts the designed neural network to only specific applications. The weight storage schemes are compared in the **data storage** criterion in terms of maintenance of the data for a long term. The **programmability** criterion determines how easily the content of the synapse cell can be updated. The amount of resources needed for the synapse cell such as the occupied area, power dissipation, and fabrication technology limitation are compared in **resource efficiency** criterion.

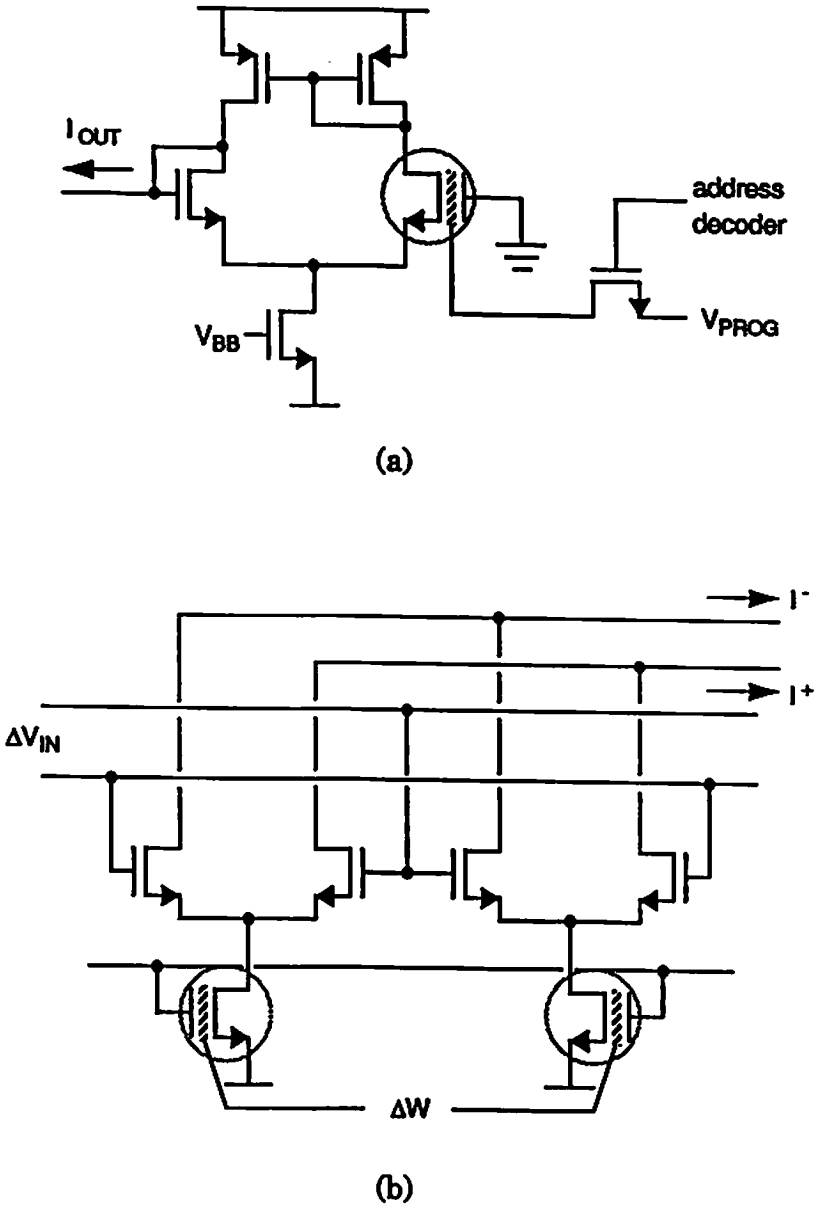


Figure 2.5: Circuit diagrams of the synapse cells with the floating-gate transistor. (a) The floating-gate transistor is used in the differential pair [53]. (b) The floating-gate transistor is used in the modified Gilbert multiplier [54].

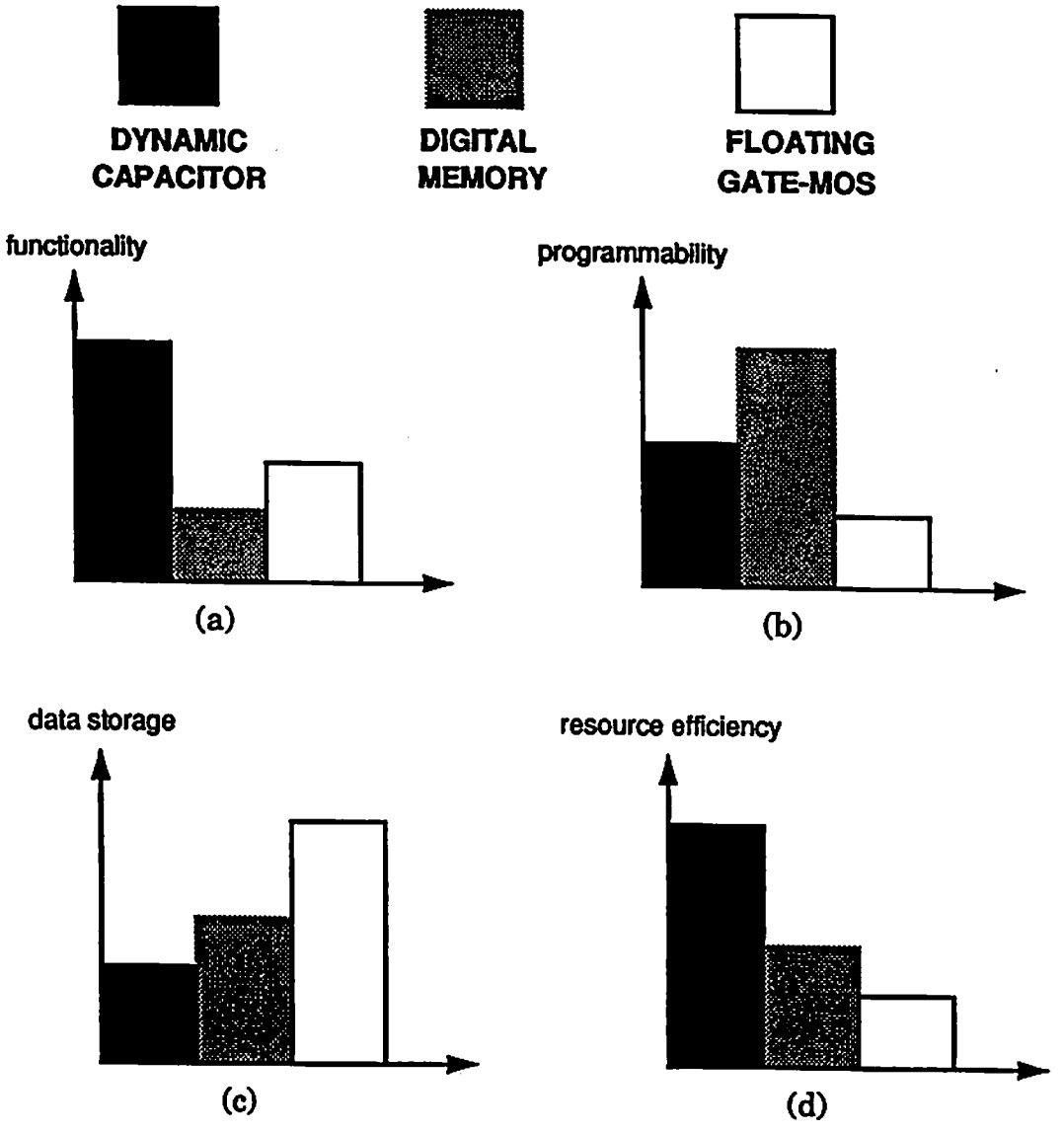


Figure 2.6: Summary of comparisons in the analog synapse cells with different storage scheme. (a) Functionality. (b) Programmability. (c) Data Storage. (d) Resource efficiency.

In a synapse cell with the dynamic capacitance, programmability of the analog weight value is well supported by the efficient use of the analog-to-digital converter which can be easily controlled by the entire system. The issue of continuously decaying weight values on the dynamic capacitances can be overcome by the additional refreshing scheme which is implemented by the on-chip or off-chip methods. Since refreshing is the learning process without modifying the weight values, it is naturally fit into the neural network hardware supporting the general learning process. The synapse cell with the dynamic capacitor have been more attractive for their full functionality. In the following chapters, the modified Gilbert multiplier with the MOS capacitor will be described for efficiently constructing the analog neurocomputing processor in various applications. The detailed operations and limitations of this type of synapse cells will be presented. In addition, the input neuron and the output neuron are discussed which is optimally combined with the synapse matrix.

2.3 Applications with VLSI Neuroprocessors

The neural networks, especially the analog implementations are well suited for various scientific and engineering applications. The general-purpose neural network processors can be reconfigured and programmed to solve specific problems. On the other hand, the architecture of the application-specific analog neural network processors can be fixed during the hardware design phase.

A biologically-inspired model is used in order to build the efficient neural processor for sensory data. For the front-end visual reception function, several silicon retina chips have been implemented: the basic silicon retina model [57], the adaptive retina [58], the contrast-sensitive silicon retina [59], and the silicon

retina with correlation-based and velocity-tuned pixels [60]. The front-end visual processing is followed by early vision processors such as the edge detection chip [61], optical motion sensor [62], object position and orientation IC [63], and the analog VLSI chip for figure-ground segregation [64]. The front-end auditory data is processed through the electronic cochlea [65]. Several silicon chips have been developed for further auditory processing and computer peripherals [66, 67]. In addition, microelectronic tactile sensing has been reported in [68].

Various analog neural processors have been implemented to process the wide-range of information such as the image, audio, and text data. Several neural-based image processing algorithms have been mapped onto the hardware [69, 70, 71, 72, 73]. Some hardware implementations were conducted for recognizing the text data [74, 75, 76]. Artificial neural networks have been implemented for processing the speech data as reported in [77, 78] and for control applications [79]. In addition, the analog neural network have been implemented for signal processing [80, 81] and communication fields [42, 82].

2.4 Various Design Techniques

Various circuit design techniques have been developed to improve the hardware performance. In the charge-manipulation circuits, the charges on the capacitor, rather than the voltage, represent the synaptic and neuronal signals. Electric charges can be efficiently manipulated with a quite small power dissipation. In the self-learning neural network chip [83], the charge pumps are used to update the synapse values which are the charges stored on the capacitors. In the capacitive synapse matrix, charges are processed throughout the network as the neuron

signals [84, 85]. In the νMOS technology [86], one MOS transistor operates as a functional unit to perform a weighted summation.

The switched-capacitor (SC) networks have been widely used for performing the neural computations [87, 88, 89, 90]. The well-developed SC integrators are used for summation of the weighted charges or currents in the capacitor. In addition, simple arithmetic manipulations are achieved by arranging the configuration of capacitors and interconnections.

Several different technologies have been adopted for implementing the analog neural network processors. The charge-coupled device (CCD) processors have been used especially for low power consumption and easy data manipulation as reported in [91, 92]. The field programmable gate array (FPGA) technique has been also chosen for implementing analog circuits [93]. The BiCMOS technology was extensively used to build the synapse matrix because the bipolar transistors have better matching property [41] and larger dynamic range [94].

Chapter 3

Design of General-Purpose Neural Networks

The generic neural operation consists of a weighted summation and nonlinear function, which are summarized as,

$$v_j = f_s\left(\sum_i w_{j,i} v_i\right) \tag{3.1}$$

where v_j is the j^{th} neuron output voltage. Here v_i is the output voltage of teh i^{th} neuron from the previous layer in a multi-layer network of from the same layer in a recursive network. The synapse value between the i^{th} and the j^{th} neurons is represented by $w_{j,i}$ and f_s is the nonlinear transfer function for the output neuron, which is usually a sigmoid function. This operation describes the feedforward or the retrieving process in the artificial neural network operations. Figure 3.1 shows the block diagram of a multi-layered neuroprocessor chip. It consists of an array of input neurons which function as drivers, an array of output neurons, and the synapse matrix. In order to support the feedback or learning process, the weight value should be programmable. It is to be efficiently updated and reliably maintained. Calculation of the new synapse weight values can be done by the companion digital signal processor chip or the host processor outside the chip or by the dedicated on-chip learning circuitry or DSP module.

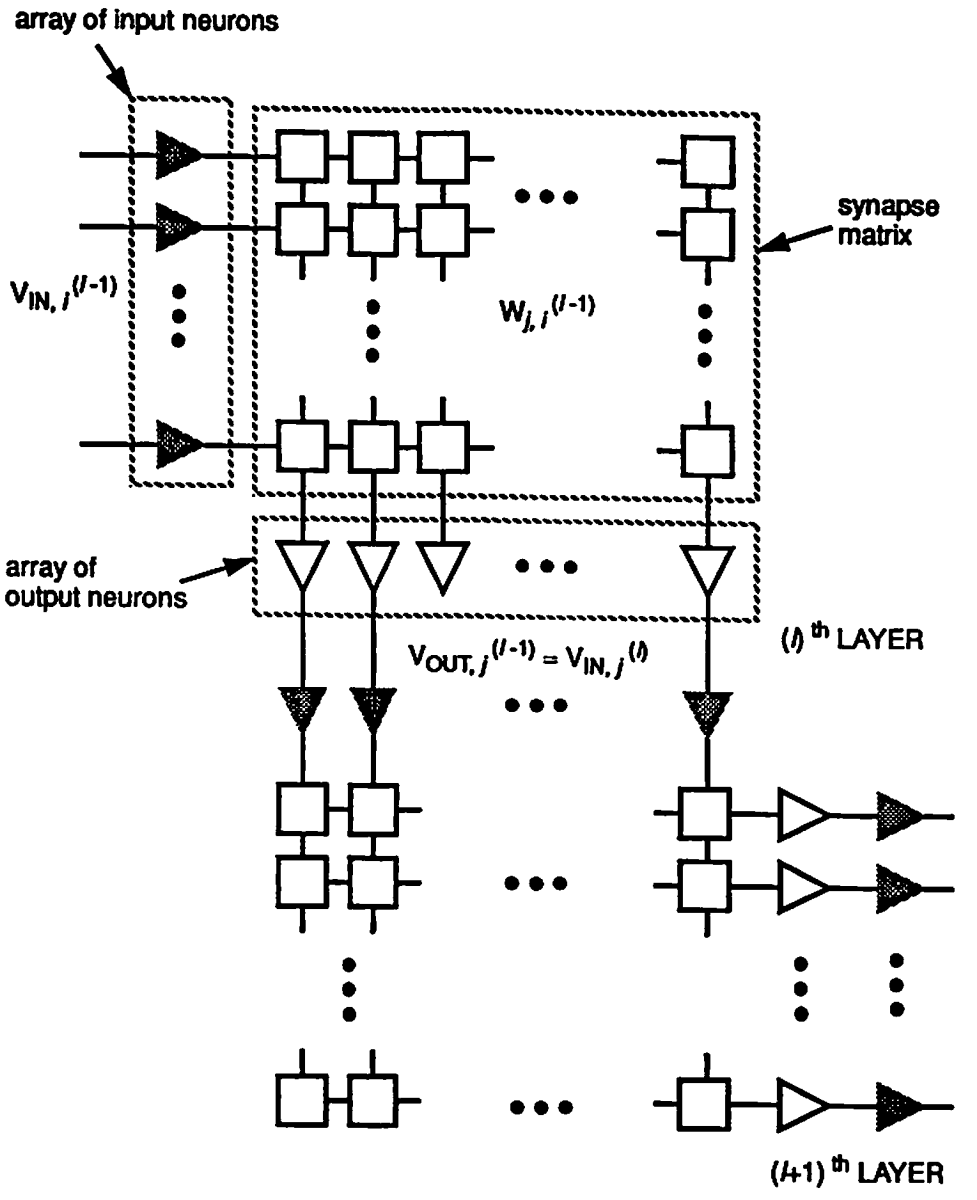


Figure 3.1: Block diagram of the multi-layered neural network.

There have been many researches on analog VLSI neural network implementations using various design technologies. Significant numbers of synapses and neurons are included in order to achieve a high computational throughput by using fully massive parallelism. Some major analog VLSI neuroprocessor chips and their properties are listed in Table 3.1 [54, 41, 95, 48, 83, 96, 97, 34, 47, 98, 99].

In this chapter, design methods for the synapse and neuron circuits will be presented in detail. In addition, solutions to effectively address physical design constraints will be discussed and system-level design issues such as the timing scheme and the network-dimension scaling methods will be outlined. The experimental results will be given.

3.1 Programmable Synapses

Since the number of synapse cells is dominant over the number of neurons, characteristics of the analog multiplier used to realize the synapse cell determines the overall accuracy, silicon area, and power consumption of the neuroprocessor chip. Several design issues should be considered for performance optimization of the synapse cell.

3.1.1 Linear Multiplication Operation

In order to understand the operation of the analog multiplier circuit, the differential pair shown in Fig. 3.2(a) should be considered as a basic building block. The output currents can be expressed as,

$$I_1 = \frac{\beta}{2}\left(\frac{V_{in}}{2} + \frac{1}{2}\sqrt{\frac{2I_{SS}}{\beta/2} - V_{in}^2}\right)^2, \tag{3.2}$$

Table 3.1: Various general-purpose analog neural network processor chips.

yr.	maker	char.	tech.	net.	size	speed	accuracy	config.	purpose
'89	Intel	floating-gate 15 yr. lifetime	EEPROM	10,240 syn.	128 ipns 64 opns	1010 low-prc. mults/sec	7 - 8 bits 4 bits[1]	single layer +feedback	general-purpose
'90	Matsu-shita	BiCMOS dyn. capacitor	2.2-µm BiCMOS	768 syn.	64 neurons	FF: 10ns 108 mults/sec	multiplier error: 5%	3-layered PDP	BP learn-ing
'90	USC	compactness dyn. capacitor	2.0-µm CMOS	4,096 syn.	64 ipns 64 opns	0.2 sec for refreshing	8 bits	single layer fully-connect	general-purpose
'90	NCSU	digital memory	1.25-µm CMOS	6,561 syn.[2]	81 nrn.[2]		6 bits	single layer fully-connect	general-purpose
'91	Mitsu-bishi	self-learning dyn. capacitor	1.0-µm CMOS	10,240 syns	25 ipns 100opns	5 ms/pattern	1 bit / capacitor	fully feed-back conn.	self-orga-nization
'91	Lock-heed	board with dis-crete elements	Lockheed prgm. res.	2,048 syn.	256 nrn.	writing time: 1 us/resistor	5 bits	single layer fully-connect	pattern recog.
'91	AT&T Bell Lab.	dyn. capacitor prog. topology	0.9-µm CMO	4096 syn.	16-256 nrn.	5 GCPS 20 MHz	6 bits/syn. 3 bits/nrn.	single, multi-layer,Hopfield	character recog.
'92	Columbia Univ.	reconfigurable dyn. capacitor	0.9-mm CMOS	1,024 syn.	1,024 nrn.	74 msec for whole refresh	operation error: 1%	1, 2, 3 layers maxnet	general-purpose
'92	Univ. of Penn.	chip-modules digital memory	2.0-mm CMOS	2,466 syn.[3]	72 nrn.[3]	1011 FLOPS	6 bits	distributed modules	neural computer
'92	Toshiba	on-chip learn digital memory	0.8-mm CMOS	576 syn.[4]	24 nrn.[4]	36 GOPS / 480 neurons	8 bits	single layer fully-connect	BP,Hebb. learning
'92	USC	wide op. range dyn. capacitor	2.0-mm CMOS	960 syn	30 ipns 32 opns	FF: 400ns 2.4 GCPS	8 bits	single layer fully-connect	general-purpose

1. in the case when there is no learning for long time.
2. estimated values from the prototype chips.
3. 100 chips are used.
4. Two chips contains the synapse matrix and neuron array, respectively.

and

$$I_2 = \frac{\beta}{2}\left(\frac{V_{in}}{2} - \frac{1}{2}\sqrt{\frac{2I_{SS}}{\beta/2} - V_{in}^2}\right)^2, \tag{3.3}$$

where β is the transconductance parameter of the transistor. The differential output current is expressed as,

$$\Delta I = I_1 - I_2 = \frac{\beta}{2}V_{in}\sqrt{\frac{2I_{SS}}{\beta/2} - V_{in}^2}. \tag{3.4}$$

For a small differential input voltage V_{in}, (3.4) can be approximated as,

$$\Delta I \simeq \frac{\beta}{2}V_{in}\sqrt{\frac{2I_{SS}}{\beta/2}} = \sqrt{\beta I_{SS}}V_{in}. \tag{3.5}$$

In order to implement a linear multiplication between the input voltage and the stored synapse weight voltage, the Gilbert multiplier circuit can be used. Figure 3.2(b) shows the circuit schematic of the Gilbert multiplier core. All transistors operates in the saturation region. The output current is obtained from the difference of two currents, I^+ nd I^-,

$$I_{out} = I^+ - I^- = (I_3 + I_5) - (I_4 + I_6) = (I_3 - I_4) - (I_6 - I_5). \tag{3.6}$$

By substituting (3.5) into (3.6), we can obtain

$$I_{out} \simeq \sqrt{\beta_u I_1}(V_3 - V_4) - \sqrt{\beta_u I_2}(V_3 - V_4) = \sqrt{\beta_u}(\sqrt{I_1} - \sqrt{I_2})(V_3 - V_4), \tag{3.7}$$

where $\beta_u = \beta_3 = \beta_4 = \beta_5 = \beta_6$. From (3.2) and (3.3), (3.7) can be refomulated as,

$$I_{out} = \sqrt{\frac{\beta_u \beta_l}{2}}(V_1 - V_2)(V_3 - V_4), \tag{3.8}$$

where $\beta_l = \beta_1 = \beta_2$.

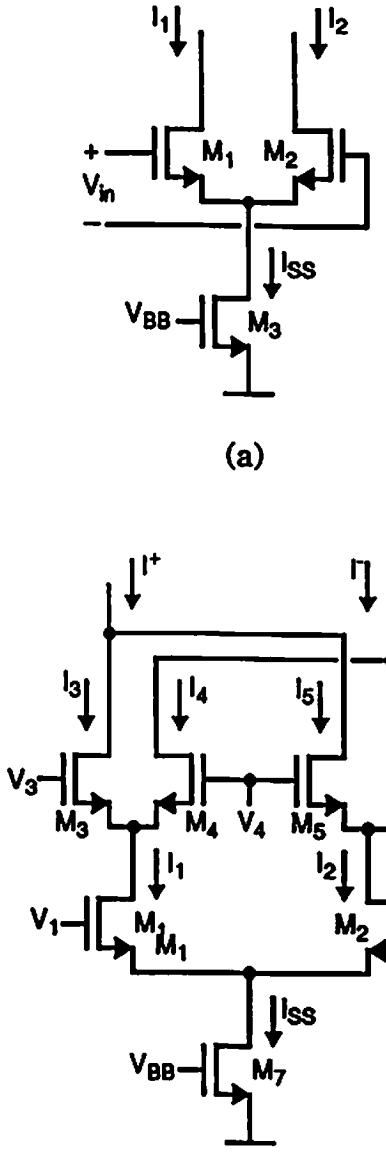


Figure 3.2: Circuit schematics of the differential pair in (a) and the basic Gilbert multiplier in (b).

Figure 3.3 shows the complete circuit schematic of the multiplier used for the synapse cell [99]. Two upper differential pairs of Fig. 3.2(b) are converted to the PMOS differential pair in order to achieve large operational ranges by reducing the number of transistors stacked between two power supply lines. The input voltage $V_{IN,i}$ is applied to the NMOS differential pair of M_1-M_2, while the synapse weight value $W_{j,i}$ is applied to the PMOS pairs of $M_6 - M_7$ and $M_9 - M_{10}$. The differential input and weight values ensure the balanced-operations of the positive and negative signals as well as can achieve the common-mode rejection. The differential output current is converted into the single-ended current through the cascode current-mirror stage consisting of transistors M_{12} through M_{21}. Based on (3.8), the synapse output current can be determined as,

$$I_{j,i} = K\sqrt{\frac{1}{2}\beta_P\beta_N}(V^+_{IN,i} - V^-_{IN,i})(W^+_{j,i} - W^-_{j,i}) = G_m V_{IN,i} W_{j,i}, \qquad (3.9)$$

where K is the current gain from transistor $M_{12(14)}$ to transistor $M_{13(15)}$. Here β_P and β_N are transconductance parameters of a PMOS transistor and an NMOS transistor in the differential pairs, respectively. Table 3.2 lists the sizes of all transistors in a prototype design.

Figure 3.4 shows the SPICE-3 [100] circuit simulation results on the DC characteristics of the analog multiplier for differential weights of -2.0 V to +2.0 V in a step size of 0.5 V. The integral errors and total harmonic distortions for various weight values are plotted in Fig. 3.5. The figure shows that the available operation range with the 1 error is $\pm$ 2.0 V both for the input and the weight voltages.

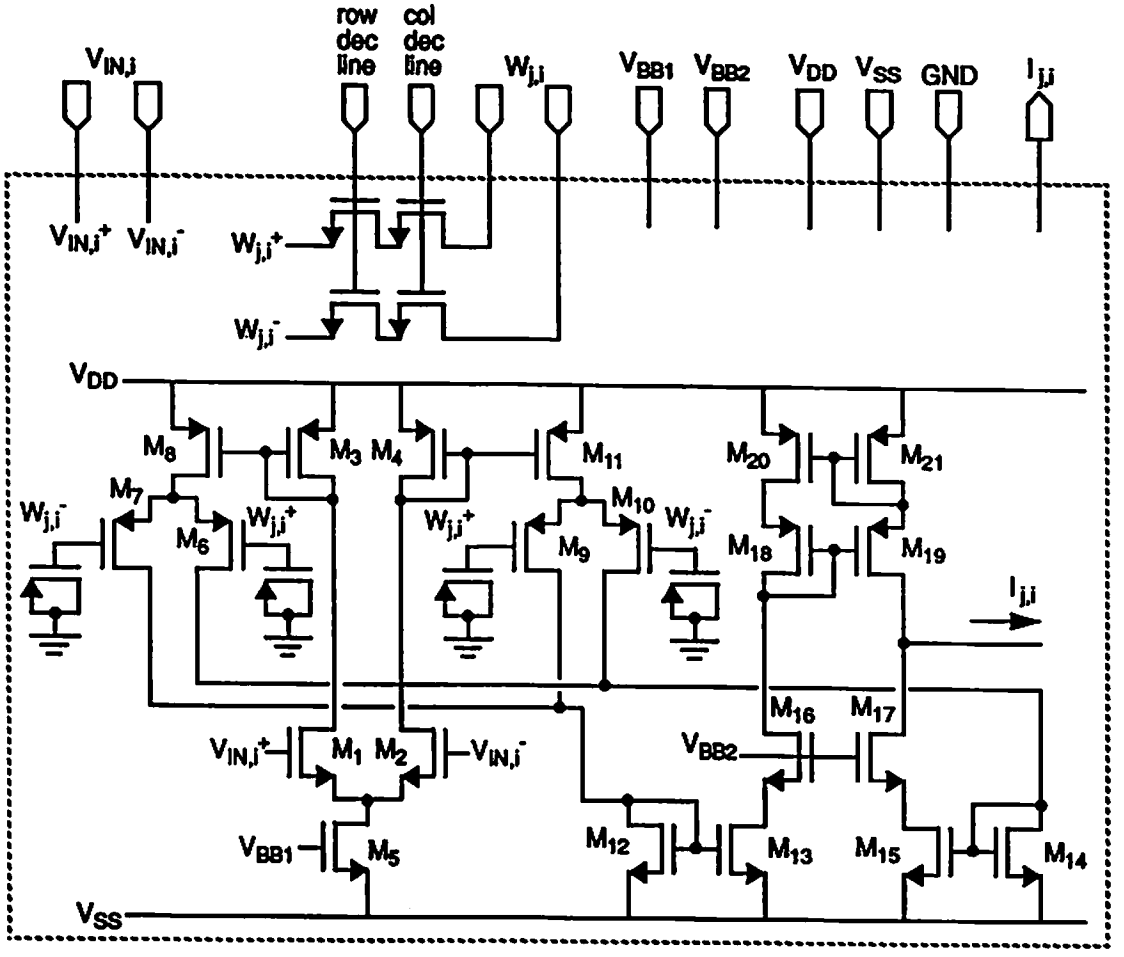


Figure 3.3: Circuit schematic of the synapse cell based on the wide-range Gilbert multiplier.

Table 3.2: Transistor sizes of the synapse cell.

Transistor	W/L [µm / µm]
M_1 , M_2	4 / 40
M_3 , M_4	20 / 2
M_5	30 / 2
M_6 , M_7, M_9 , M_{10}	4 / 30
M_8 , M_{11}	20 / 2
M_{12} , M_{14}	8 / 4
M_{13} , M_{15}	24 / 4
M_{16} , M_{17}	60 / 2
M_{18} , M_{19}	16 / 4
M_{20} , M_{21}	14 / 6

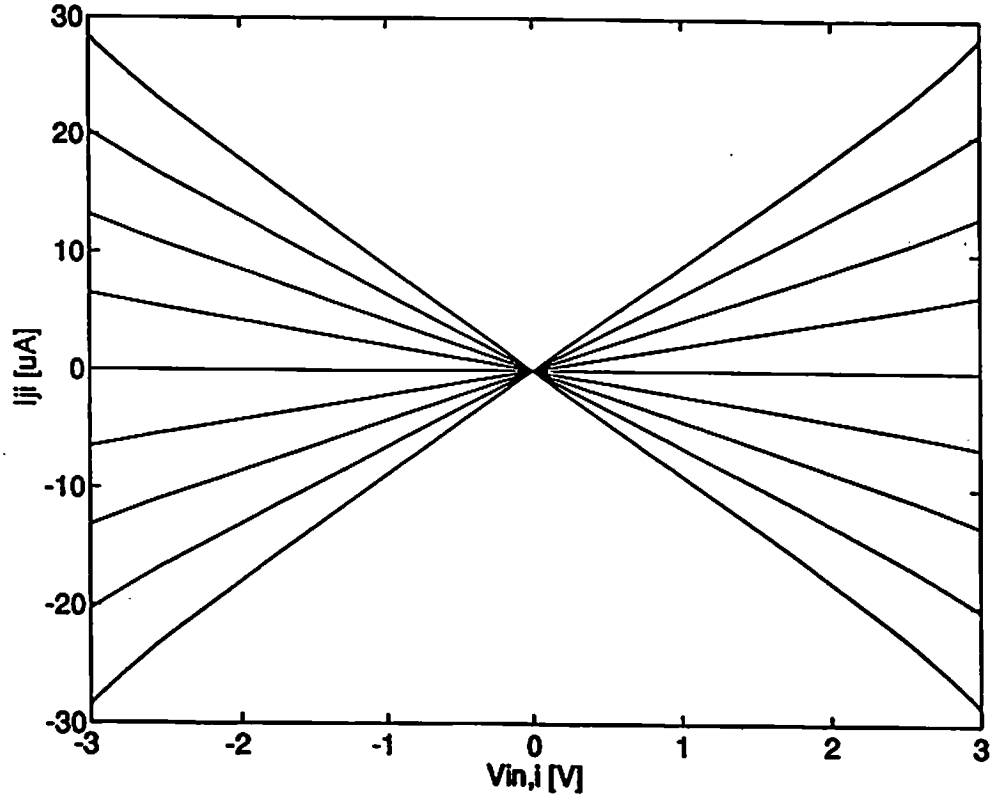


Figure 3.4: Simulated results of the DC transfer characteristics of the synaptic multiplication. The differential weight value W_{ji} increases from -2 V to 2 V in a step size of 0.5 V.

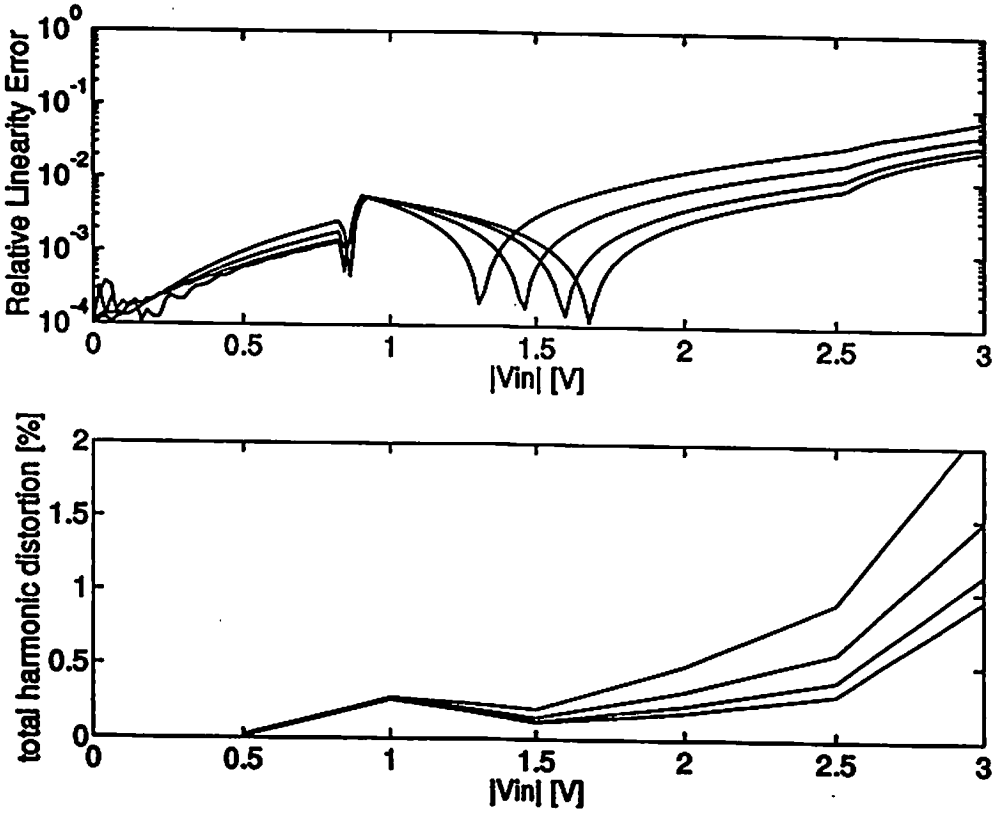


Figure 3.5: Simulated results of the linearity error and the total harmonic distortion of the synaptic multiplication.

3.1.2 Weight Value Storage

The learning algorithm can be performed in the host computer or the companion digital signal processing (DSP) chip or plug-in board and the calculated synapse values are stored in the digital memory. Each synapse value is converted into the analog voltage and loaded into the capacitor of the synapse cell. This synapse value is dynamically stored on the capacitance which are contributed by MOS transistor and possibly augmented by the additional capacitor. Address decoders are used to direct the common signal line to the desired synapse site as shown in Fig. 3.6. Two types of errors can degrade the operational accuracy: the error voltage due to the switching transient and data loss due to the leakage current.

Although the neural computation is performed in the continuous-time fashion, the physical weight modification is processed in the sampled-data format which might result in the charge feedthrough problem. The synapse value is sampled-and-held by the access switches. During the switch-off transient period, charges remaining in the channel and the overlap capacitance between the gate and the source/drain terminals of the MOS transistor make the actual stored weight voltage deviate from the desired value [101, 102]. In addition to using the minimum-sized pass transistors as the switches, the differential weight signal scheme circumvents the charge feedthrough problem by rejecting the common-mode error. If necessary, a dummy transistor controlled by a complementary clock signal can be added.

The voltage stored on the capacitor is decayed due to the leakage current flowing through the reversed-biased PN junction between the diffusion region and the substrate of the acess switch transistor as shown in Fig. 3.7. Periodic refresh is required to maintain the accurate synapse weight values [34, 41, 42, 43]. Figure 3.8

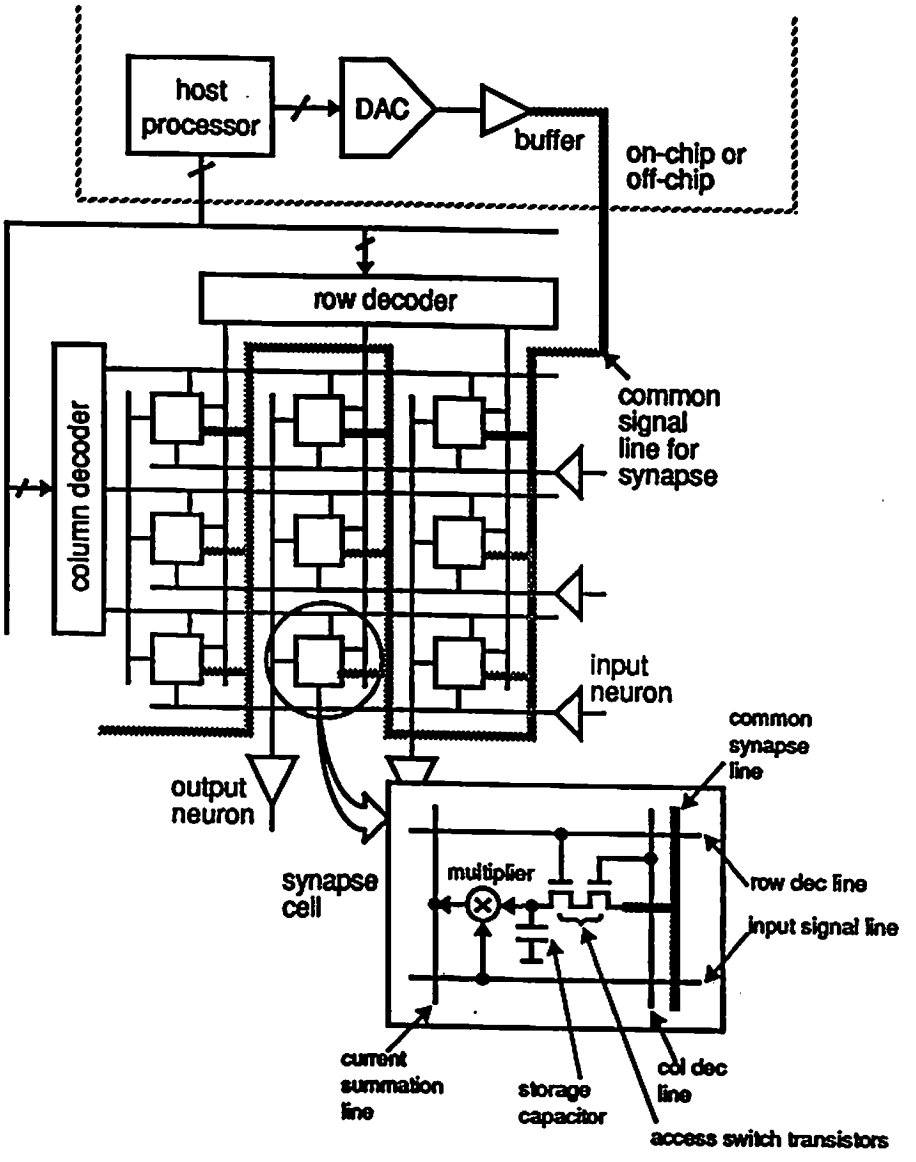


Figure 3.6: Synapse weight programming scheme.

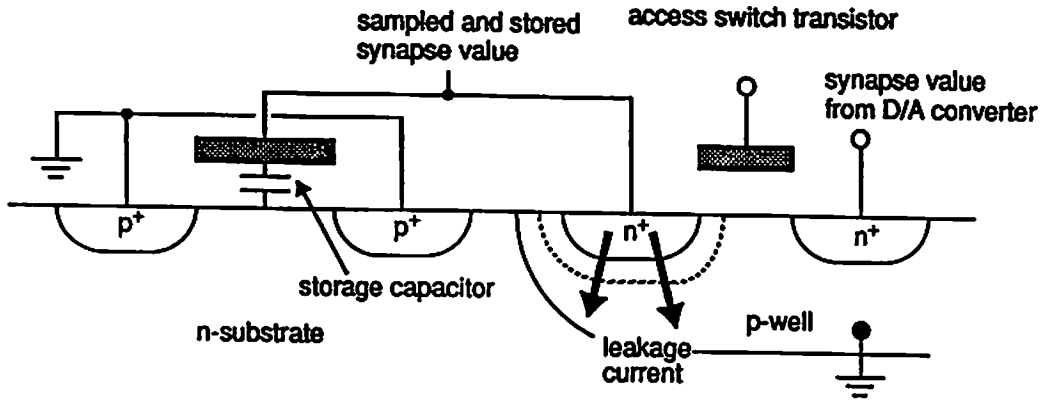


Figure 3.7: Synapse weight storage on the capacitance through the access switch transistor.

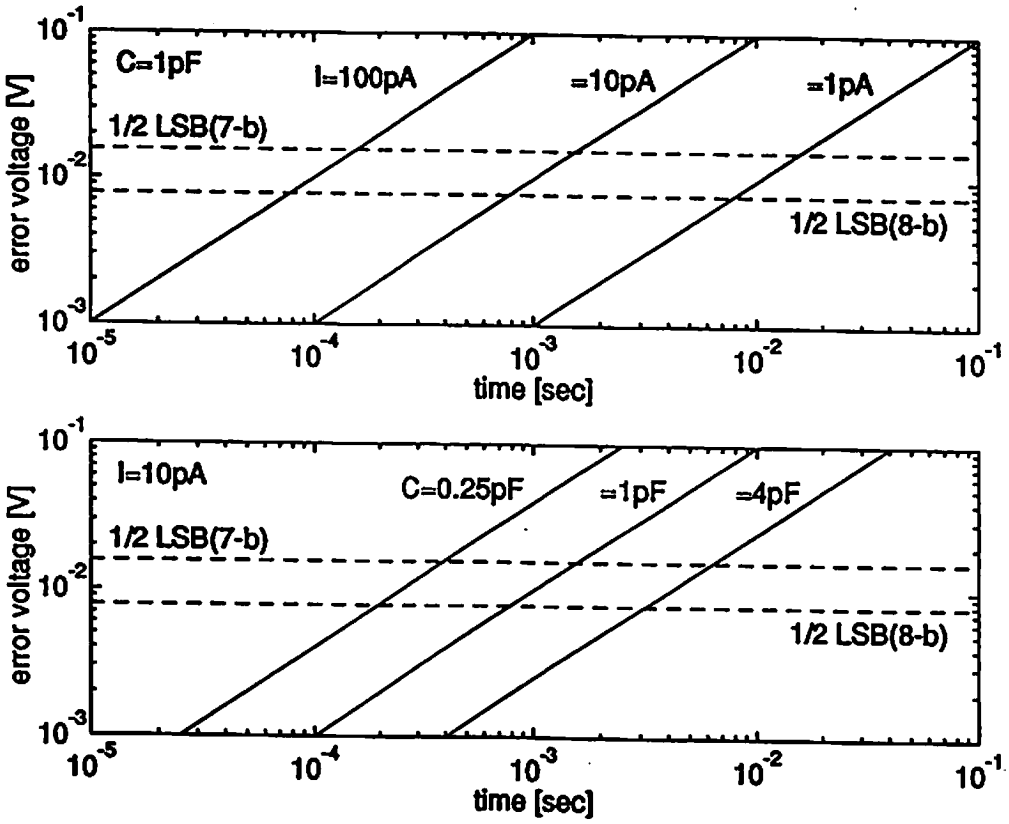


Figure 3.8: Synapse weight accuracy versus the charge retention time under different values of leakage current.

shows the calculation results of the data loss for several magnitudes of the leakage current. Please note that refreshing in every 1 *msec* is sufficient to maintain the 8-bit accuracy for a given leakage current of 10 *pA*.

3.2 Input Neurons

The input neuron buffers the input signal and provides the high-speed driving capability for a large capacitive load. The proposed input neuron consists of an operational amplifier configured as the unity-gain buffer. Since there is a large number of synapse cells to be driven by one input neuron, the equivalent load capacitance to the input neuron might be quite large. Thus, a fast settling response of the input neuron needs careful design. It is desirable to let the input neuron occupy a compact silicon area and dissipate very low power. Figure 3.9 shows the circuit schematic diagram of the input neuron which provides the differential input to the synapse cells. In Fig. 3.9(a), the input voltage is applied in the differential format and two amplifiers are needed. In Fig. 3.9(b), the input voltage is applied in the single-ended format in order to reduce the number of the I/O pins. Two amplifiers and two resistors are needed. The circuit diagram of the operational amplifier used in the input neuron is shown. In order to reduce the power consumption in the output branch while maintaining the speed, the class-AB output stage is used [103].

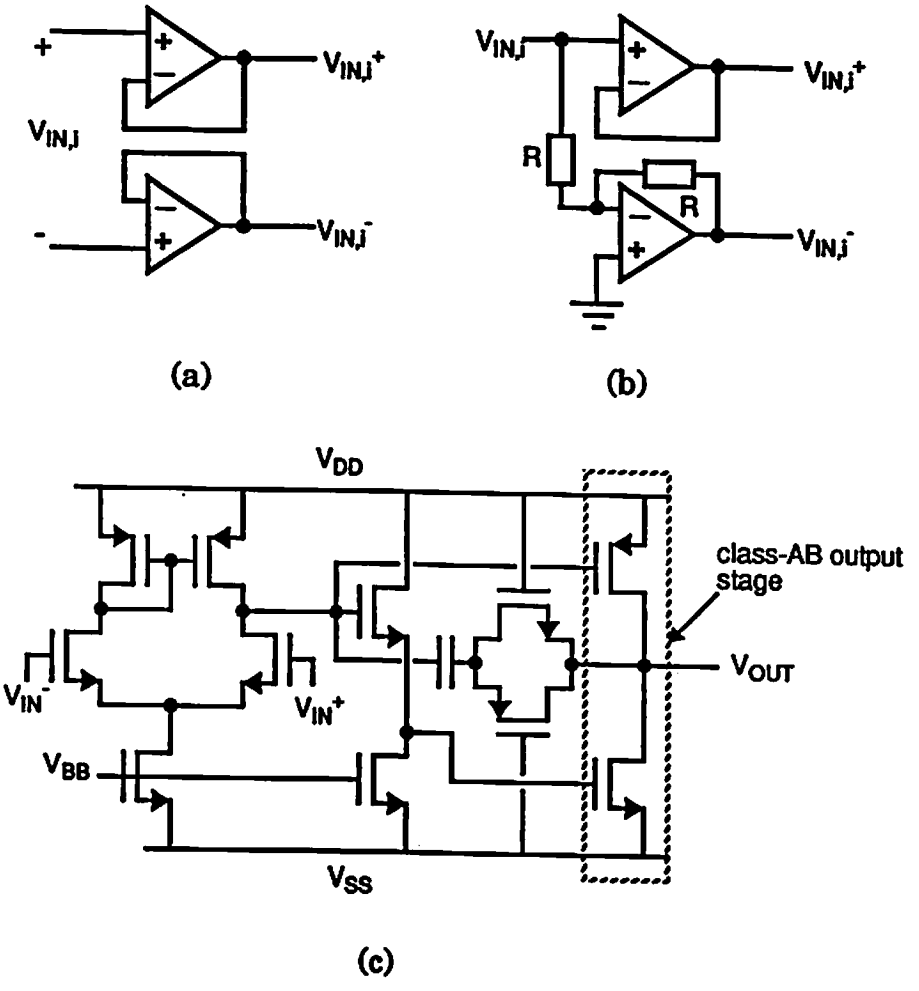


Figure 3.9: Input neuron. (a) Differential-to-differential input neuron. (b) Single-ended-to-differential input neuron which requires two additional resistors. (c) Circuit schematic diagram of the operational amplifier.

3.3 Output Neuron

The output neuron converts the summed current into the voltage and performs one of several types of functions such as the thresholding, linear amplification, and sigmoid function.

3.3.1 Linear Current-to-Voltage Conversion

In Fig. 3.10, the detailed circuit schematic of the output neuron is shown [99]. Summation of the weighted products is naturally done by hard-wiring according to the Kirchoff's current law. Current-to-voltage conversion is performed by the transresistance amplifier consisting of an operational amplifier and a feedback resistor. Since the output impedance of the synapse cell is finite, the synapse current is dependent on the output voltage, which is the input node voltage of the output neuron. In the proposed design, this node is connected to the virtual ground of the operational amplifier to eliminate the the synapse current variation.

Since the current summation results from a large array of synapse cells, the transresistance amplifier should have a sufficient capability to handle a large magnitude of the current for proper linear conversion. Thus, the operational amplifier includes the source follower as an output stage as shown in Fig. 3.10(a). In addition, six active transistors are used to implement the feedback resistor, which can achieve higher accuracy in the wide operational range and less silicon area [104]. The low-precision passive resistor is not recommended. The summed current is converted into the voltage according to the following expression,

$$V_{LIN,j} = R_{eq} \cdot \sum_{i=1}^{M} I_{j,i} = \frac{\sum_{i=1}^{M} I_{j,i}}{2\beta_R(V_{RF} - V_{thp})}, \tag{3.10}$$

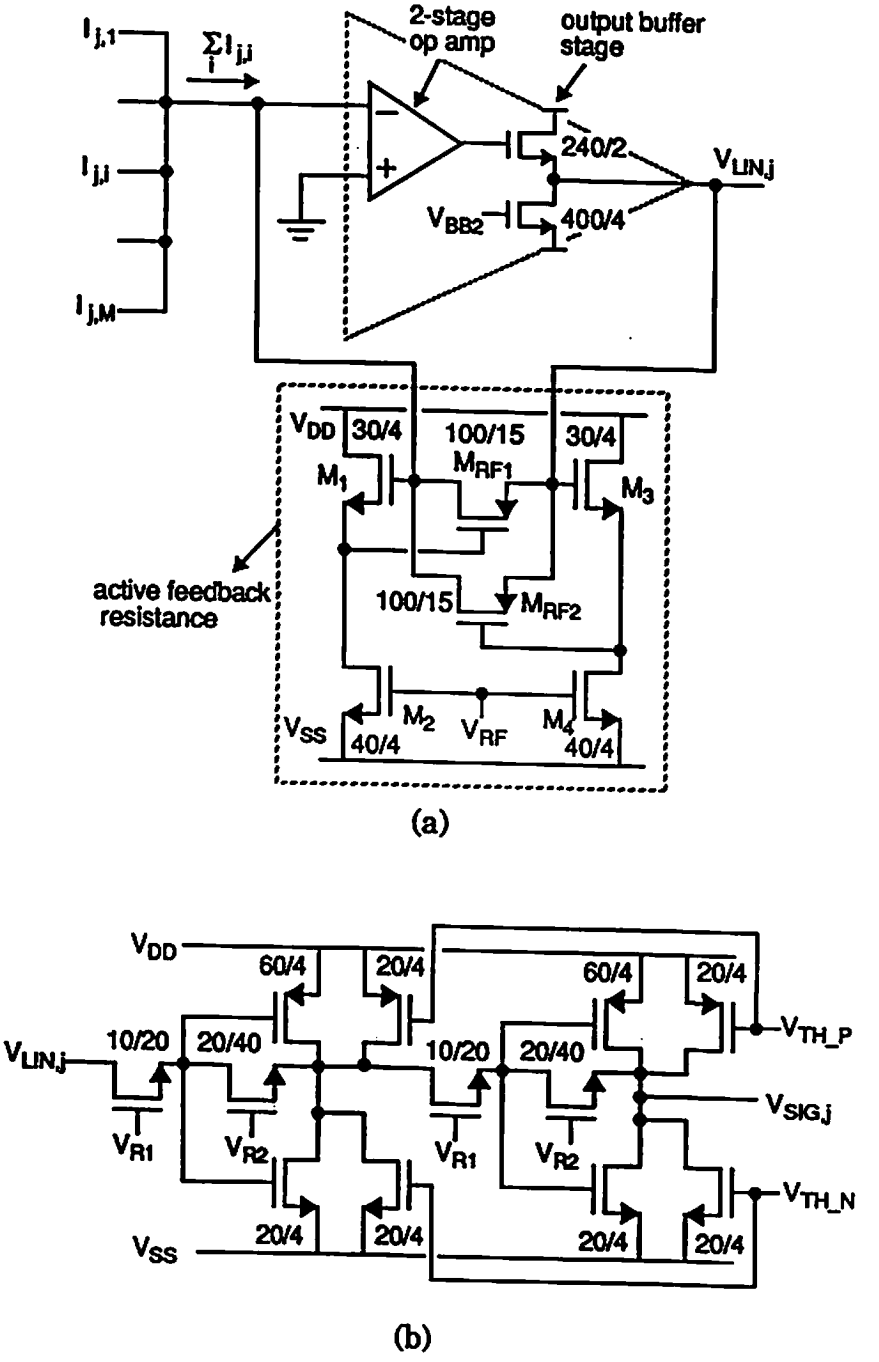


Figure 3.10: Circuit schematic of the output neuron. (a) Linear current-to-voltage converter. (b) Sigmoid function generation circuit.

where V_{RF} is the control voltage to tune the equivalent feedback resistance value R_{eq}. Here β_R and V_{thp} are the transconductance parameter and the threshold voltage of transistors $M_{R1(2)}$, respectively.

3.3.2 Sigmoid Function Generation

Sigmoid function generation is required for performing the back-propagation learning algorithm. The sigmoid function is approximately realized by cascading simple inverting amplifiers with the input and feedback resistors as shown in Fig. 3.10(b). The voltage gain of the sigmoid function is determined as,

$$\frac{V_{SIG,j}}{V_{LIN,j}} = \left(\frac{\frac{R_{eq,2}}{R_{eq,1}}}{1 + \frac{1}{A}(1 + \frac{R_{eq,2}}{R_{eq,1}})} \right)^2, \tag{3.11}$$

where A is the voltage gain of the inverting amplifier. Since the resistors are realized by transistors biased in the triode region, their equivalent resistance values can be controlled,

$$R_{eq,1(2)} = \frac{1}{\beta(V_{R1(2)} - V_{th})}, \tag{3.12}$$

where β, V_{th} are transconductance parameter and the effective threshold voltage of the transistors, respectively. The control voltages, V_{TH_P} and V_{TH_N} are used to reduce the offset voltage of the sigmoid function generator due to process-induced nonideal device characteristics. Their values can be set during the chip-initializing phase. The entire gain of the output neuron are controlled by the gate voltages of the PMOS transistors of the feedback resistor in the transresistance amplifier (V_{RF} in (3.10)) and the voltages of the input and the feedback resistors in the sigmoid function generator (V_{R1} and V_{R2} in (3.12)). Figure 3.11 shows the SPICE

circuit simulation results on the DC characteristics of the output neuron with various voltage gains.

In order to provide programmability and flexibility, the output neuron can be reconfigured into several operational modes. This can be done by multiplexing several operations with switches. Figure 3.12 shows the detailed circuit schematic of the improved output neuron. By controlling the interconnection switches, 4 different operations can be achieved.

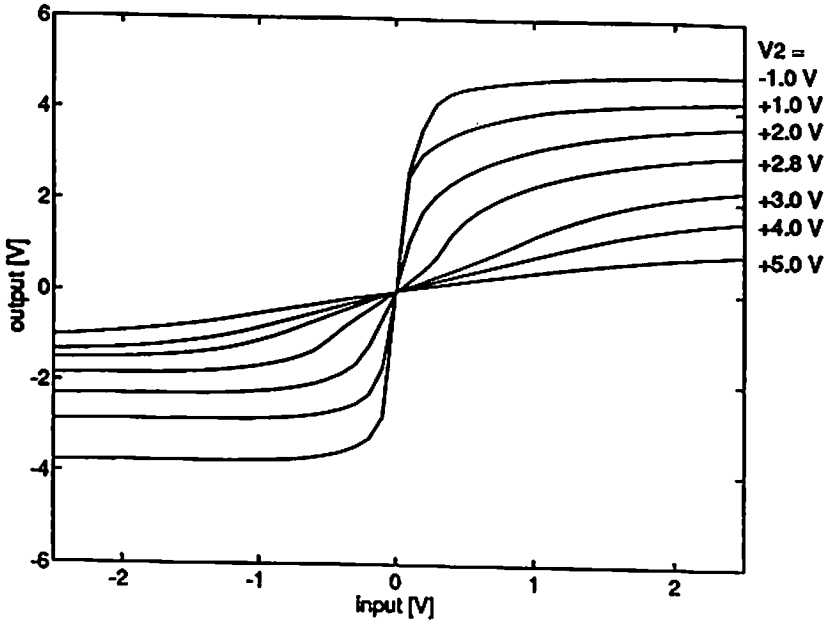
3.4 System-Level Considerations

3.4.1 Required Refreshing Frequency

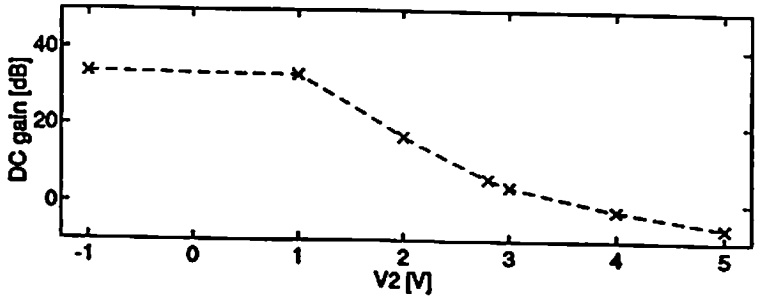
As mentioned earlier, the analog synapse weight value stored at the capacitor could gradually lose its initial value due to the leakage current. The refreshing process is required so that the analog value can be maintained within the allowable error imposed by the system specification. In the proposed design, the digital synapse weight information in the system memory is periodically converted by the digital-to-analog converter and written into the analog memory site. The amount of time to write the computed weight value into the synapse storage, T_{write}, consists of the following items [42],

$$T_{write} = T_{data} + T_{DAC} + 6R_{on}C_W, \tag{3.13}$$

where T_{data} is the amount of time for fetching the digital data from the system memory, T_{DAC} is the digital-to-analog conversion time, R_{on} is the ON-resistance of the switch transistors, and C_W is the effective storage capacitance. If there

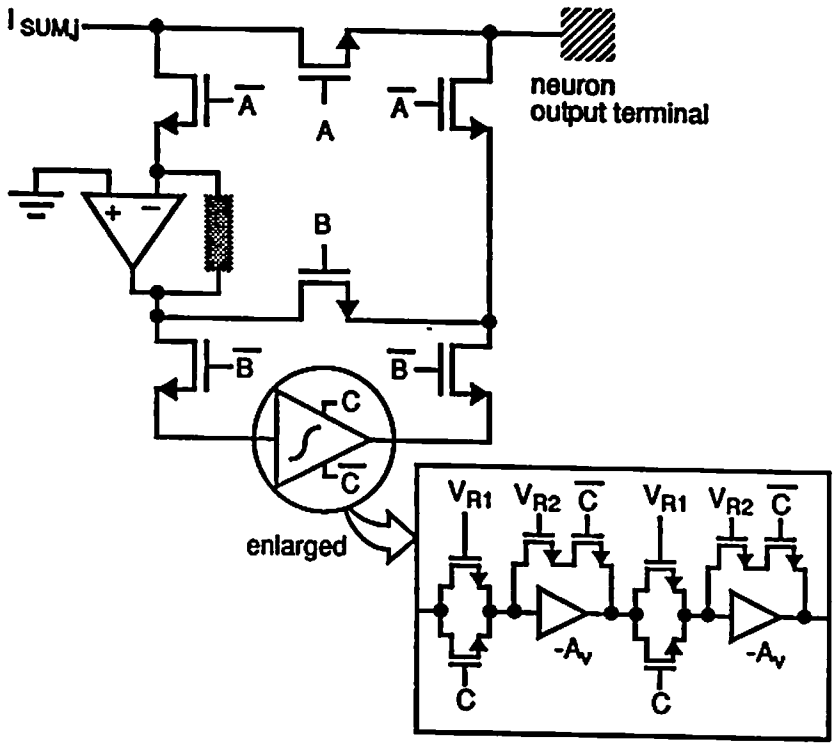


(a)



(b)

Figure 3.11: Simulated results of the sigmoid function generation circuit with gain controllability. (a) DC analysis. (b) AC analysis.



X	Y	A	B	C	Operations
0	0	1	0	0	ready for increasing the number of the input neurons
0	1	0	1	0	linear unit (slope controllable)
1	0	0	0	1	threshold unit (threshold controllable)
1	1	0	0	0	sigmoid uint (gain controllable)

Figure 3.12: Improved output neuron for reconfigurability.

are M synapse cells in the entire system, then the total time for updating or refreshing the network is,

$$T_{update} = M \cdot T_{write}. \tag{3.14}$$

The amount of time for the synapse value to experience a change equivalent to the 1/2-bit resolution, ΔT, can be determined as,

$$\Delta T = (\frac{C_W}{I_L})(\frac{V_F}{2^{N+1}}), \tag{3.15}$$

where I_L is the leakage current, V_F is the full-dynamic range of the synapse weight value, and N is the number of bits to represent the data. From (3.14) and (3.15), the timing requirement of the system for a reliable refreshing scheme is determined as,

$$\Delta T > T_{update}. \tag{3.16}$$

Figure 3.13 shows the numerical examples of the size and accuracy relationship of the synapse matrix for different values of the storage capacitances.

3.4.2 Scalability of Network Size

The proposed general-purpose neural network is programmable not only in terms of synapse weight values but also in the light of the network size scalability. The chip is based upon the one layer of an artificial neural network consisting of the input neuron array, the synapse matrix, and the output neuron array. Due to the constraint of finite chip area, the number of components must be restricted, so the size of a network is fixed. In this section, the methods to implement the multi-layered neural network by chip-partitioning and to implement a larger size network by chip-cascading are presented.

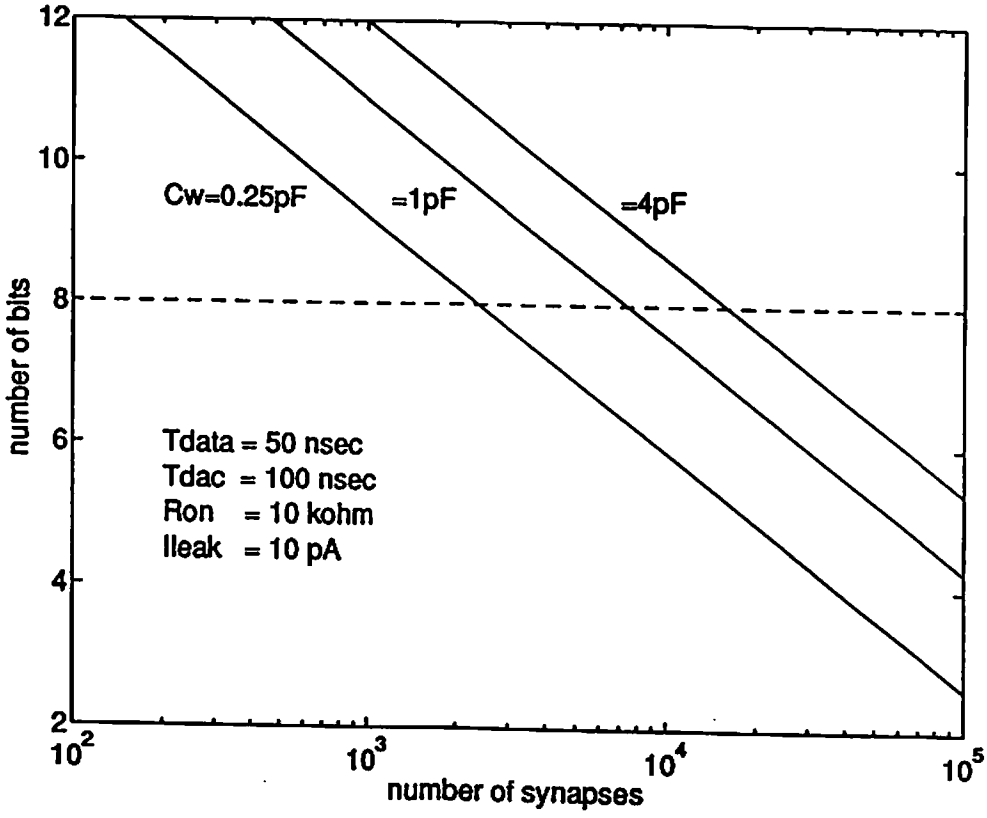


Figure 3.13: Weight accuracy versus the number of synapses connected to one common signal line.

Figure 3.14 shows one example of chip-partitioning to implement a four-layered perceptron. Only diagonal portion of the synapse matrix is effectively used and the other area shown in the shaded region is not used in order to avoid conflicting use of available input/output neurons. The output voltage of an output neuron in one layer can be externally fed into the input of the input neuron in the other layer because the input and output neurons are designed to be directly compatible.

The network size can be increased by cascading the identical neuroprocessor chips as shown in Figure 3.15. In order to increase the number of the output neurons, the inputs of multiple chips are connected to the common input signal bus. In order to increase the number of the input neurons, each current summing node is connected and the array of the output neuron in one chip is activated to produce the output voltage. The above two ways can be combined to implement neural networks of any sizes.

3.5 Experimental Results

A prototyping general-purpose neural network chip consisting of an array of the input neurons, an array of the output neurons, and the synapse matrix performs the neural computation of one layer. The chip was fabricated in a 2-μm double-polysilicon CMOS technology from the MOSIS Service of USC/Information Science Institute at Marina del Ray, CA, 90292, USA [105, 106].

Figure 3.16 shows the measured DC characteristics of a synapse cell performing linear multiplication. One cell can be arbitrarily accessed by the row/column address decoders from the synapse matrix. The differential synapse values in Fig. 3.16(a) and the input voltage in Fig. 3.16(b) range from -2.0 V to +2.0 V

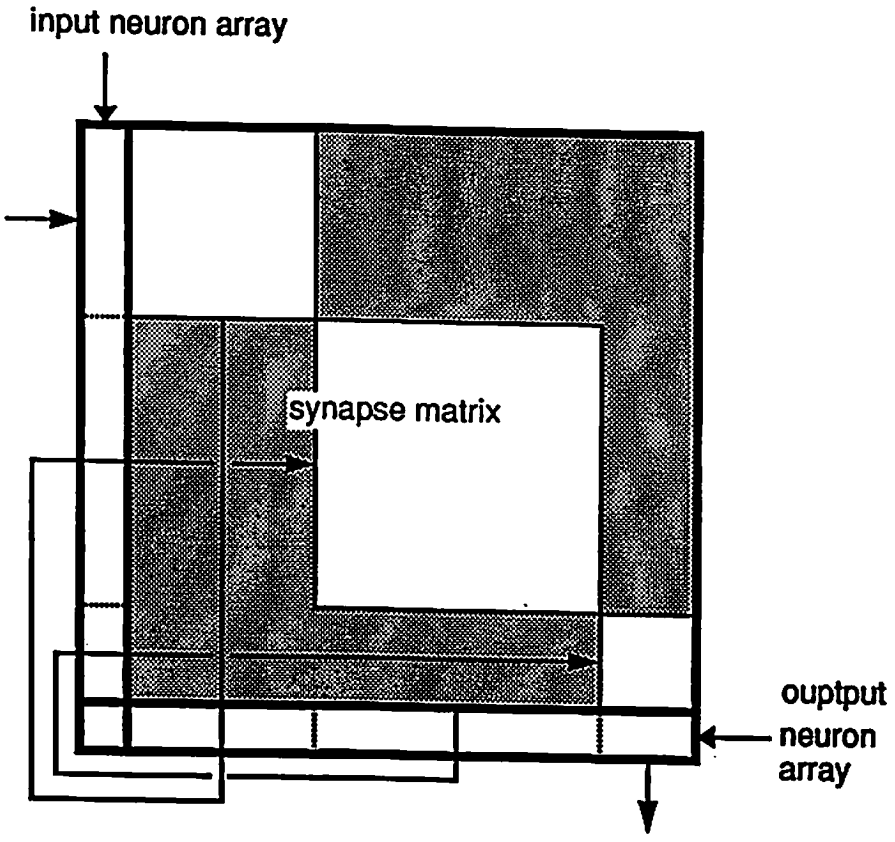


Figure 3.14: Partitioning the fixed-dimension chip into several layers.

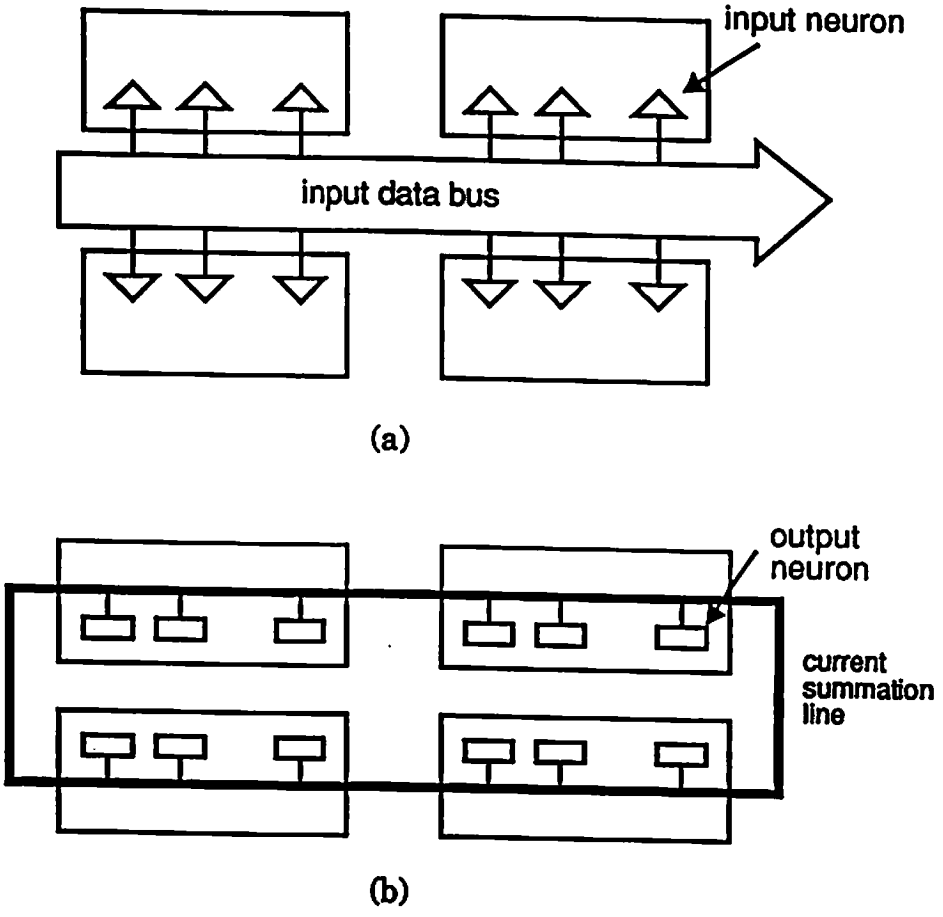


Figure 3.15: Cascading the chips to increase the dimension of the network. (a) To increase the number of output neurons. (b) To Increase the number of input neurons.

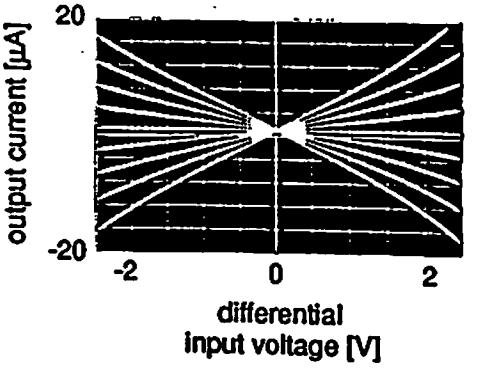
in a step size of 0.5 V. The operation ranges for the input signal and the synapse weight value corresponding to a linearity error of less than 1 % is -1.75 V to +1.75 V. The small offset currents can be easily compensated through modification of synapse values and/or the use of an additional synapse weight connecting to a fixed input voltage for a output neuron.

Figure 3.17 shows the measured dynamic synapse weight value changes as a function of time due to the leakage current. The rate of the output current change is about 12.5 nA/sec. For the conductance value of the synapse cell of 5 $\mu A/V$, the voltage change rate is 2.5 mV/sec, which corresponds to the time elapse of about 4 sec for the 1-bit resolution change of the synapse weight value in 8-bit operations.

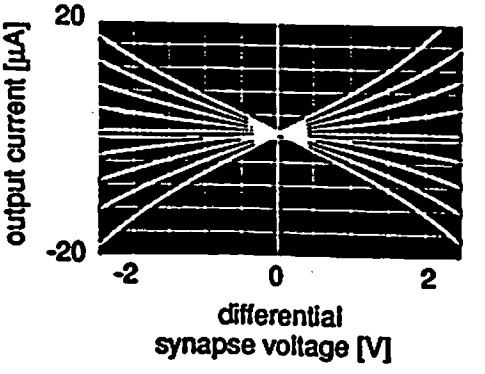
The measured dynamic range of the input neuron is from -3.0 V to +3.0 V, which is sufficient to drive the synapse cell. The settling time with 1 % error is about 300 $nsec$ for the capacitive load of 7 pF.

Measured result of the linear current-to-voltage converter as a part of the output neuron is shown in Fig. 3.18(a). This converter can handle the input current level of more than 250 μA. In the measurement, the active feedback resistance was set to 4 $k\Omega$. The measured sigmoid function with various voltage gains of 6.5, 2.8, and 0.8 are shown in Fig. 3.18(b). The achievable maximum voltage gain is around 2,000.

The physical layout of one synapse cell is shown in Fig. 3.19. It occupies an area of 124 x 186 λ^2. The weight-storage capacitor shown in the wide solid line occupies an area of 1,046 λ^2 which is implemented by an efficient use of the silicon area. The realized capacitance is around 0.95 pF value in the given 2-μm CMOS



(a)



(b)

Figure 3.16: Measured results on the multiplication operation of the synapse cell. (a) Synapse current versus input voltage. (b) Synapse current versus weight voltage.

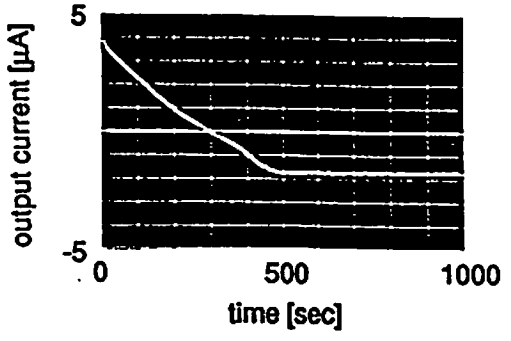
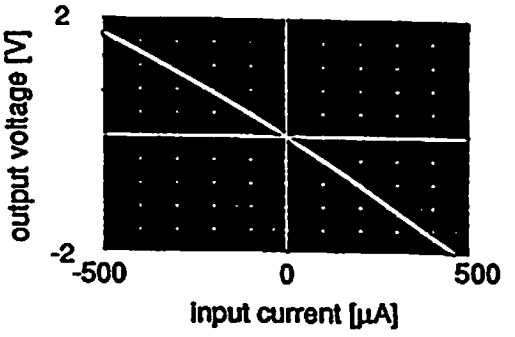
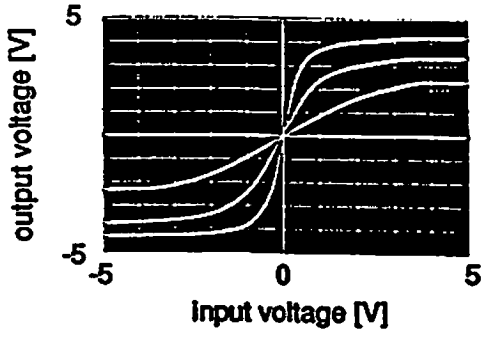


Figure 3.17: Measured results of the synapse weight retention characteristics.



(a)



(b)

Figure 3.18: Measured results of the output neuron. (a) Linear current-to-voltage conversion. (b) Sigmoid function generation.

technology. In the scalable CMOS design supported by the MOSIS Service, one λ corresponds to one micron fro the 2-μm CMOS technology.

System-Level Measurement

The system-level experiments were conducted with the fabricated synapse cells in our laboratory in order to understand the feasibility of neuroprocessing system implementation. Figure 3.20 shows the schematic diagram of the measurement setup, which consists of four synapse cells and a linear current-to-voltage converter as a single output neuron. Here, the sigmoid function is not included in the simple learning. The host processor executes the learning algorithms. An analog-to-digital converter is used to interface the output voltage of the network. Update of the synapse values is performed sequentially by the digital-to-analog converter and the address control signals. An additional digital-to-analog converter is used for determining the analog input voltages.

The primary goal of the learning experiment is to compensate the bias which was intentionally added so that the desired output voltage will return to zero. The bias current is provided by the constant voltage source V_{bias} and the resistor R_{bias}. The generalized *Delta Rule* can be expressed as [11],

$$w_{j,i}(n+1) = w_{j,i}(n) + \Delta w_{j,i}(n), \tag{3.17}$$

and

$$\Delta w_{j,i}(n) = \eta \cdot [t_j(n) - o_j(n)] \cdot i_i(n), \tag{3.18}$$

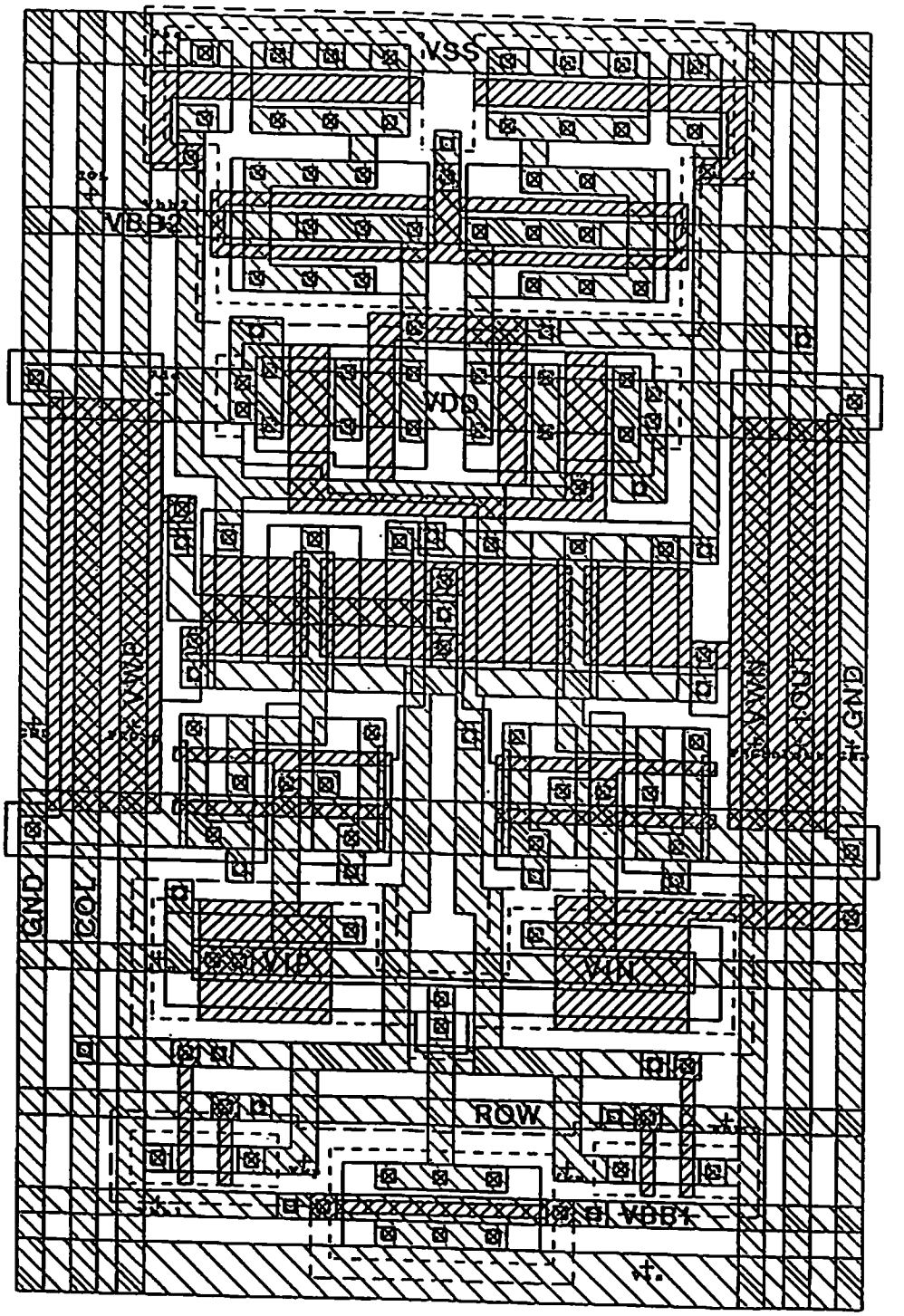


Figure 3.19: Physical layout of a differential-input synapse cell. An additional capacitor is included to hold the synapse weight information.

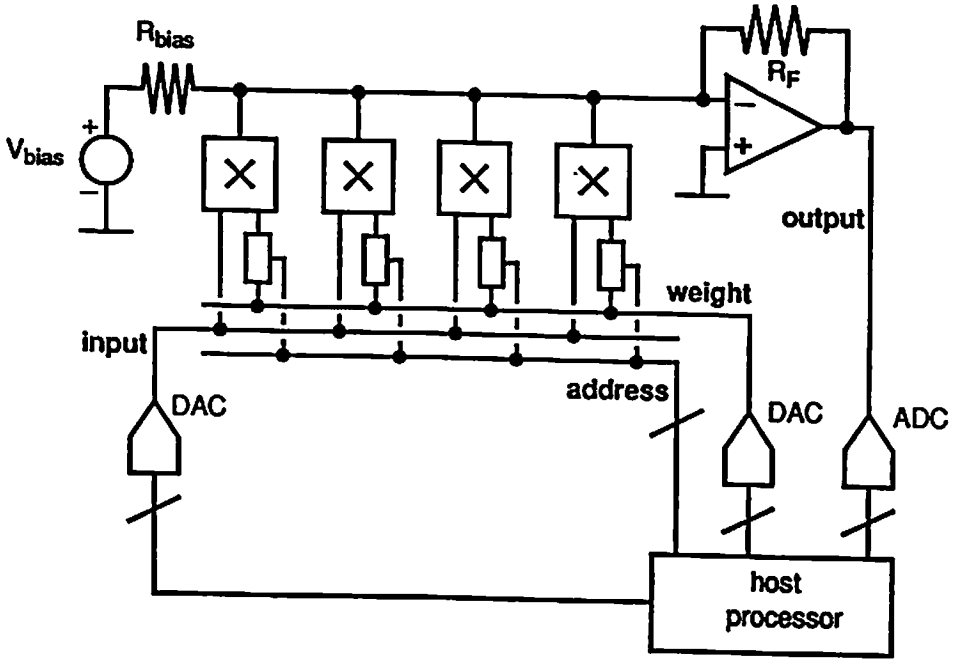


Figure 3.20: Measurement setup for system-level experiments.

where η is the learning rate constant and $t_j(n)$ is the desired output value at the n^{th} iteration. The i^{th} input and j^{th} output at the n^{th} iteration are $i_i(n)$ and $o_j(n)$, respectively. In our experiment, (3.18) can be simplified to

$$\Delta w_i(n) = -\eta \cdot o(n) \cdot i_i(n) = \eta \cdot o(n), \tag{3.19}$$

because the desired value is zero and the input is fixed to -1 V.

Figure 3.21 shows the measurement results. In the initialization phase (marked A), four weight values are set to be very small random numbers. Thus, the summation current results mainly from the bias term as follows,

$$I_{sum} = \sum_{i=0}^{3} G_M w_{j,i} i_i + \frac{V_{bias}}{R_{bias}} = \sum_{i=0}^{3} 4.8 \cdot 10^{-6}(0)(-1) + \frac{-3.013}{52.4 \cdot 10^3} = -57.5\mu A,$$

and the output voltage is

$$o(initial) = -R_F \cdot I_{sum} = -19.5 \cdot 10^3 \cdot (-57.5 \cdot 10^{-6}) = 1.12V.$$

After the learning proceeds for a sufficiently long period as shown in the figure marked by B, the weight values are updated in order to compensate the effet of the added bias as follows,

$$I_{sum} = \sum_{j=0}^{3} 4.8 \cdot 10^{-6}(-3)(-1) + \frac{-3.013}{52.4 \cdot 10^3} = 0.1\mu A,$$

and the output voltage is

$$o(final) = -19.5 \cdot 10^3 \cdot 0.1 \cdot 10^{-6} = -1.95mV.$$

The output value obtained after the learning can be significantly close to the desired output one.

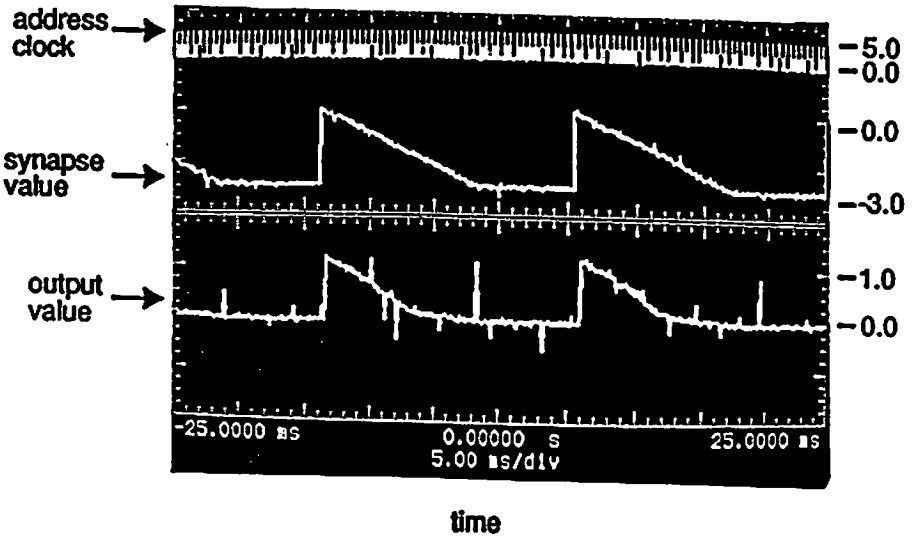


Figure 3.21: The intentionally added offset term can be removed by the learning process.

Chapter 4

Neuroprocessor for Self-Organization Mapping

There are several types of unsupervised learning schemes: the Hebbian learning, the competitive learning, the differential Hebbian learning, and the differential competitive learning. A self-organizing network using competitive learning has the desirable property of effectively producing spatially organized representation of various features of the input signals. Competitive learning depends on competition between output nodes of the neural network. In the competitive learning layer of a self-organized neural network, the winner-take-all (WTA) operation is executed as lateral inhibition operation. In this operation, the node with the largest output value is selected as the winning node and it inhibits all other nodes.

There have been several VLSI implementations of the Hamming network and Kohonen self-organization mapping algorithms. In [107], the modular CMOS design of a Hamming network was described. In [108], the charge-based Hamming network was designed for high-density and low power consumption. In [109], the basic study of implementing a Kohonen mapping network was outlined. In [39], the Kohonen neuron was designed based on a multi-level storage technique. In

[37], the analog neural network processor for self-organizing mapping was presented. The on-chip learning implementations of the Kohonen neural network were reported in [110, 111]

4.1 Basic Architecture

A self-organization neural network mainly consist of two layers as shown in Fig. 4.1. The first layer contains an array of input neurons and the second layer is the competitive layer performing the winner-take-all operation. These two layers can reside on two separate chips as a chip set or the latter circuitry can be combined with the existing general-purpose neural network module in a standard cell approach.

The block diagram of thee proposed self-organization neural network processor is shown in Fig. 4.2. The synapse matrix computes the distortion or the distance-measure between the applied input and the stored weight values [37]. The $(j,i)^{th}$-synapse cell produces the current,

$$I_{j,i} = G_m(U_i - W_{j,i})^2, \tag{4.1}$$

where U_i, $W_{j,i}$ are the input voltage and the stored synapse value, respectively. The above distance measure is computed by the multiplier in the voltage scheme with $V^+_{in,i} = W^+_{j,i} = U_i$ and $V^-_{in,i} = W^-_{j,i} = W_{j,i}$ from (3.9) of the previous chapter. In the j^{th}-output neuron, all currents from the corresponding synapse array are

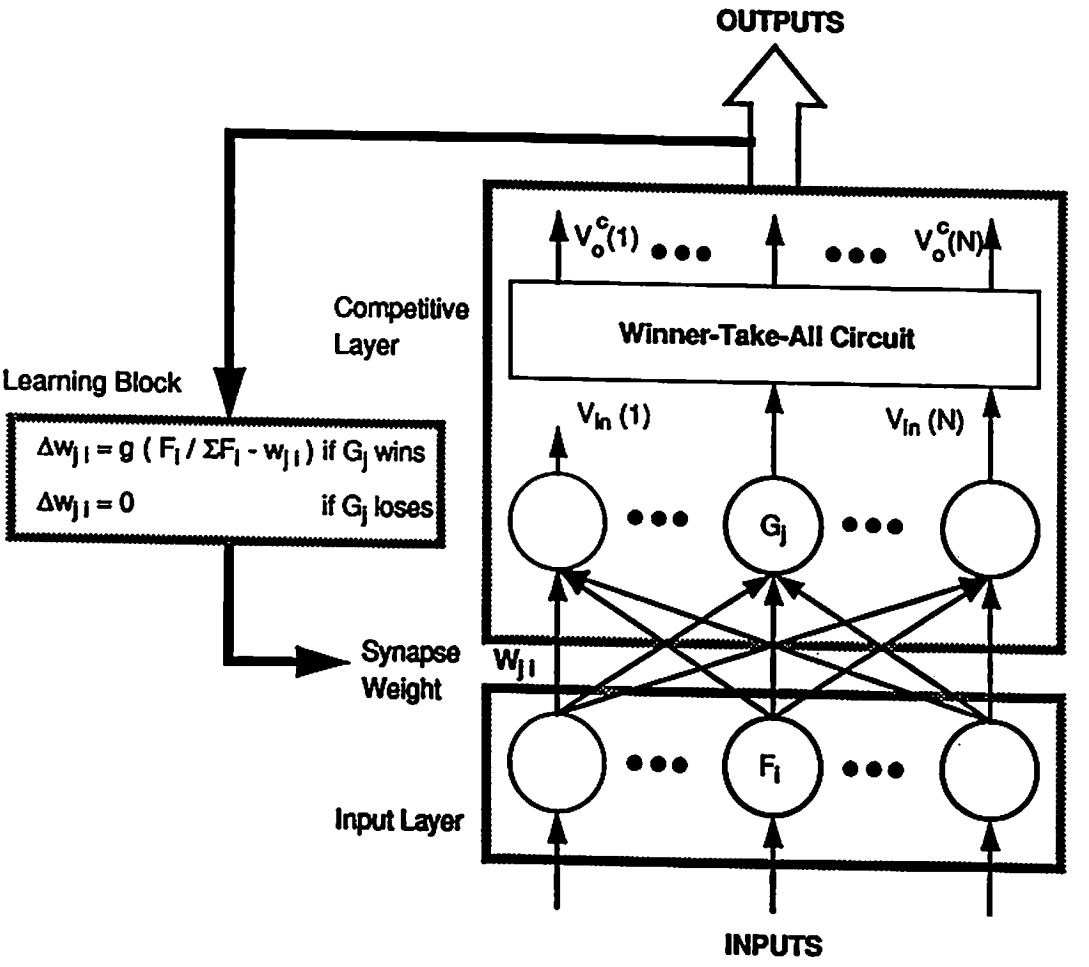


Figure 4.1: Block diagram of a self-organization neural network.

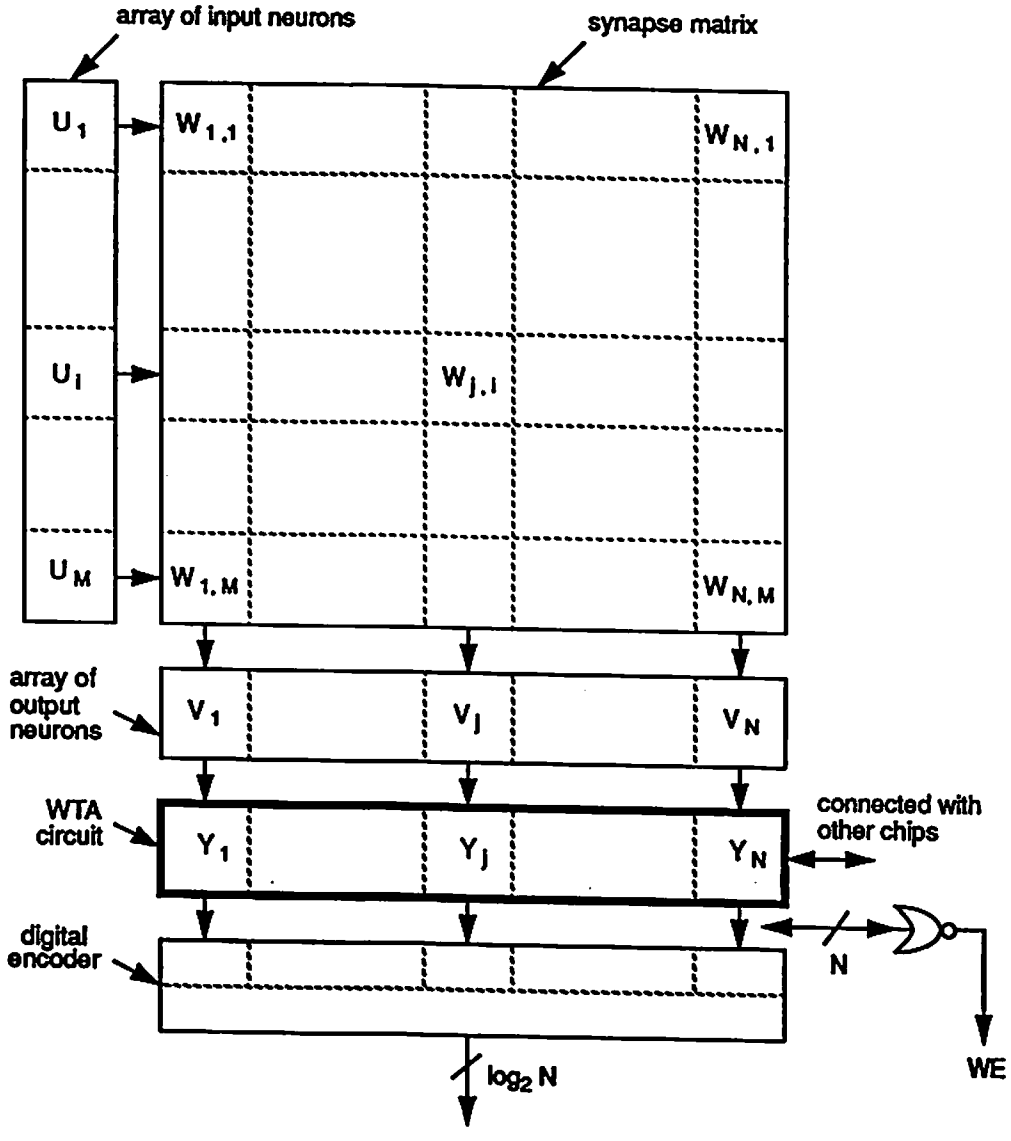


Figure 4.2: Block diagram of a self-organization neuroprocessor chip.

summed. The linear current-to-voltage conversion and sigmoid function generation are performed on the summing current as follows,

$$V_j = (-1) \cdot f[\alpha \sum_{i=i}^{M} (U_i - W_{j,i})^2] \quad for\ j = 1 \cdots N, \tag{4.2}$$

where α is the proportional constant and $f(\cdot)$ is the sigmoid function.

This output voltage appears as the input to the winner-take-all circuit. The output voltage of the WTA circuit is defined as,

$$Y_j = \begin{cases} logic-1 & \text{if } V_j > V_k \text{ for all } k \\ logic-0 & \text{otherwise.} \end{cases} \tag{4.3}$$

Thus, the minimum distortion or distance-measure is selected as the output of the self-organization network. The digital encoder of the final stage is used to reduce the required pin numbers. Since the WTA circuit is the key building block for a self-organization neural network, the main emphasis is placed on the design of a high-performance WTA circuit.

4.2 Considerations of WTA Circuit

In order to make the WTA circuit be useful for large-scale problems, several design considerations are carefully addressed:

- high accuracy,
- high operation speed, and
- compactness.

Several WTA circuits have been reported in various technologies as listed in Table 4.1 [112, 113, 114, 87, 107, 115, 116, 117].

Table 4.1: Summary of various WTA circuit implementations.

	MIT [88]	Caltech[89]	Johns Hopk. Univ. [91]	T. U. of Nova Scotia [91]	Texas A&M Univ. [92]	Tohoku Univ. [93]	Ohio Umv. [93]	USC [92]
characteristics	RS+ comparators	2 (or 4) -transistor cell	2-transistor cell	SC network	2 transistor cell	logic circuits + neuron-MOSFET	switches + diff. amp	multi-input source-coupled circuit
operation	neighbor selection logic circuit	common current steering operation	common current steering operation	full conn. + integrating C's	common current steering operation	comparison of local signal & global ramp signal	comparison of local & global feedback signal	cascading dyn. biasing current steer.
bias	strong	weak	weak	strong	strong	strong	strong	strong
I/O format	V/V	I, V/V	I/I	V/V	I/V	V/V	V/V	V/V
speed	fast	very slow	very slow	medium	fast	very fast	very fast	very fast
compatibility	medium	< medium	< medium	good	< medium	> medimu	> medium	good
accuracy	medium	medium	medium	good	high	medium	high	high
power dis.	large	very small	very small	very large	> small	very small	large	> small
silicon area	very large	very small	very small	very large	very small	small	large	> small
dimension	O(N)	O(N)	O(N)	$O(N^2)$	O(N)	O(N)	O(N)	O(N)
purpose	self-organization network	cooperative stereo matching	current-mode signal processing	SC associative memory	Hamming network	associative memory	self-organization network	self-organization network

The WTA circuits built with transistors biased in the subthreshold region [113, 114], are quite promising and consume an extremely small amount of electrical power. The approach is suitable for implementing biologically inspired artificial neural systems where billions of transistors can be integrated on a single substrate within the next decade due to efficient use of electrical power. However, it has some significant limitations such as low operation speed, small dynamic range, and poor noise immunity. The current signal input scheme [107, 114] prevents the network from being well compatible with the available general-purpose neural network chip of which outputs are in the voltage format. They usually does not provide fully binary output voltages so that additional circuitry is required to interface with digital processors for post data processing. Finally, several circuits can not operate in the continuous-time mode. The switched-capacitor version [87] must use the clocking signals. The circuit in [115] should use the global ramp signal and the operation of the circuit in [116] should be divided into two phases.

In the proposed WTA circuit, all transistors are biased in the strong inversion region to achieve high-speed operation. Its input is the voltage of the output neuron and the fully-binary values are produced as the output of the WTA circuit. Since the operation is performed in the continuous-time mode, there is no need to use the global clock or the comparison signal. This circuit can be easily interfaced to a digital processor for efficient learning. It can also be arranged to process more than 1,024 inputs in practical scientific and industrial applications. Several key design techniques such as cascading, distributed biasing, and dynamic current steering are used to enhance the circuit performance.

4.3 Basic WTA Circuit

A detailed schematic diagram of the WTA circuit is shown in Fig. 4.3. All transistors operate in the saturation region. Each WTA cell consists of two branches. The first branch (M_1, M_2, and M_5) converts an input voltage into the cell current as,

$$I_C^{(j)} = \frac{\beta_1}{2}(V_{in}^{(j)} - V_{CM} - V_{th,1})^2 \quad for\ j = 1 \cdots N, \tag{4.4}$$

where β_i and $V_{th,i}$ are the transconductance parameter and the effective threshold voltage of transistor M_i, respectively. These currents are compared and redistributed along the common signal line V_{CM}. In the second branch (M_3 and M_4), the current in each cell is converted into the output voltage as,

$$V_{out}^{(j)} = \frac{1}{\lambda_4}\left[\frac{2mI_C^{(j)}}{\beta_4(V_{BB2} - V_{SS} - V_{th,4})^2} - 1\right] + V_{SS}, \tag{4.5}$$

where λ_i is the channel-length modulation parameter and m is the current gain between transistors M_2 and M_3. V_{CM} is the common node voltage to which all source terminals of the input transistors M_1's are tied. Since the source terminals are at the same potential for all the cells, the current flowing through each cell is related to the square of the input voltage. Thus, the strongest input can bring the largest amount of current out of the total bias current. This largest current is converted and amplified to produce the largest voltage as the output of the winning node. If the input voltage differences are sufficiently big, the winner output is saturated at the positive supply value. while the other outputs are saturated at the negative power supply value. Through the use of a common voltage node, the total bias current is provided by the transistor M_5 in every cell.

As the number of inputs increases, the WTA circuit can be extended linearly by simply abutting the common signal node through cells.

To facilitate the analysis, consider P different input voltage groups. The i^{th}-group i has n_i elements and its input is V_i. The current flowing through each cell in the i^{th}-group is

$$I_i = \frac{\beta_1}{2}(V_i - V_{CM} - V_{th})^2. \tag{4.6}$$

The bias current from transistor M_5 in each cell is

$$I_B = \frac{\beta_5}{2}(V_{BB1} - V_{SS} - V_{th})^2[1 + \lambda_n(V_{CM} - V_{SS})]. \tag{4.7}$$

The total current is distributed as,

$$\sum_{i=1}^{P} n_i \cdot I_i = N \cdot I_B, \tag{4.8}$$

where

$$N = \sum_{i=1}^{P} n_i.$$

Here N is the total number of the competitive cells. From (4.6) to (4.8), the common node voltage V_{CM} is determined and the output voltage of each cell is uniquely decided according to the applied input. When the input voltage value to the j^{th}-group is the largest, the number of cells in the j^{th}-group should be one ($i.e\ n_j = 1$). The current flowing in this cell, I_j, should be larger than the current flowing through a single cell in any other groups in order to ensure the proper winner-take-all function. This largest current is designed to make the output node saturated to the positive power supply voltage.

In a simple case of two cells only, this winner-take-all circuit is degenerated to a simple differential amplifier with differential voltage outputs. In Fig. 4.4(a),

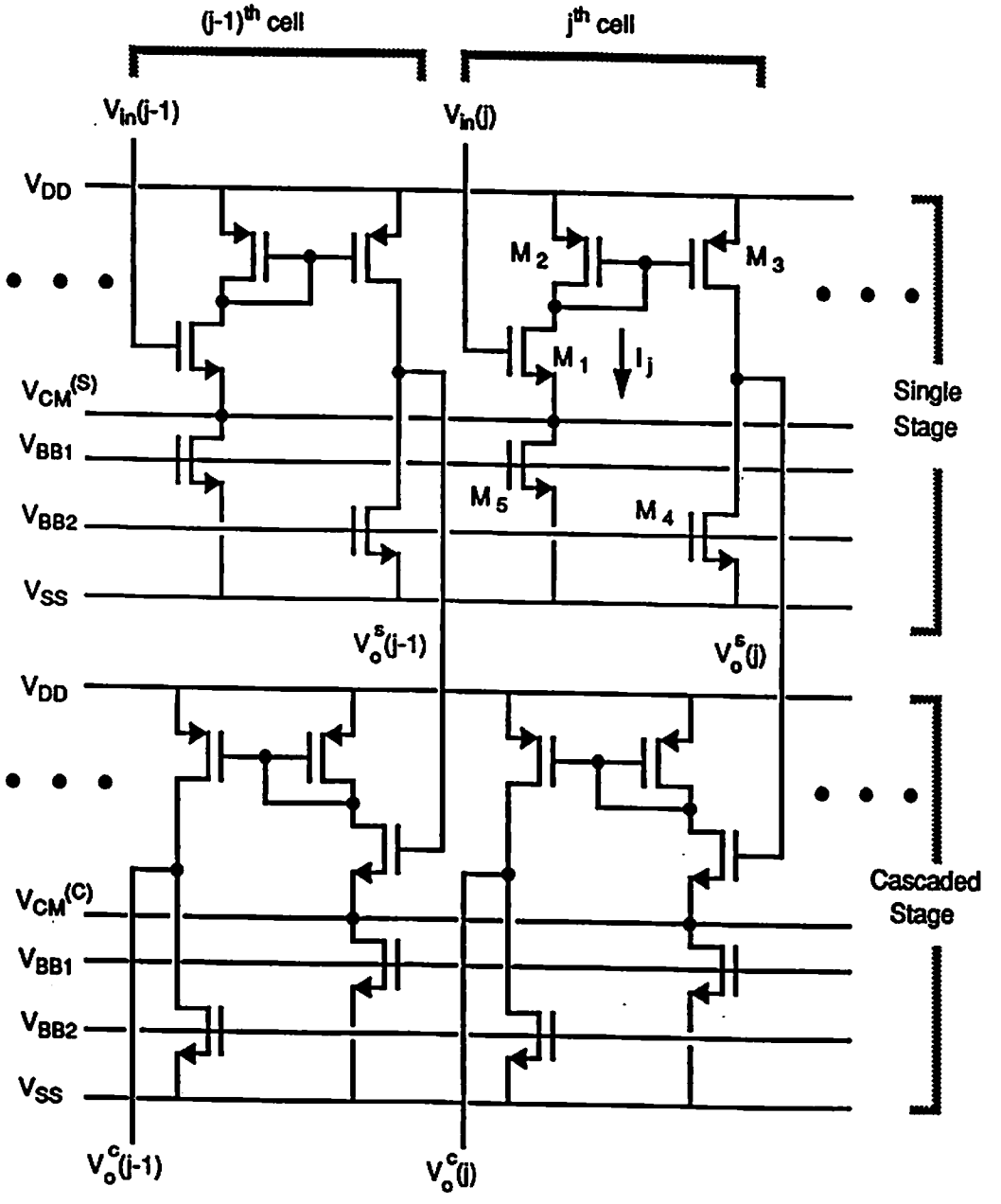


Figure 4.3: Schematic diagram of the winner-take-all (WTA) circuit.

the simulated DC characteristics of a two-cell WTA circuit are shown. A single power supply voltage of 5 V is used. One input is set to 2.5 V, and the other input is increased linearly from 2 V to 3 V. To obtain the fully binary output values for the winning and losing nodes, the required input voltage difference is found to be at least 100 mV for the single stage configuration. The simulated transient response is shown in Fig. 4.4(b). The response time of the single stage is 60 $nsec$ with a capacitance load of 0.2 pF at each cell. Please note that the circuit performance can be enhanced by cascading the identical stages as shown in Fig. 4.4, which will be described in detail in the following section.

4.4 Analysis and Design Considerations

To effectively illustrate the behavior of this WTA circuit in a large-scale network, three groups of input voltages are considered: the winning voltage V_W, the second largest input voltage V_M, and the smallest voltage V_L. The number of cells in each group is 1, M, and $N-M-1$, respectively. From (4.6), the current flowing through a single cell in each group can be expressed as,

$$I_W = \frac{\beta_1}{2}(V_W - V_{CM} - V_{th})^2, \tag{4.9}$$

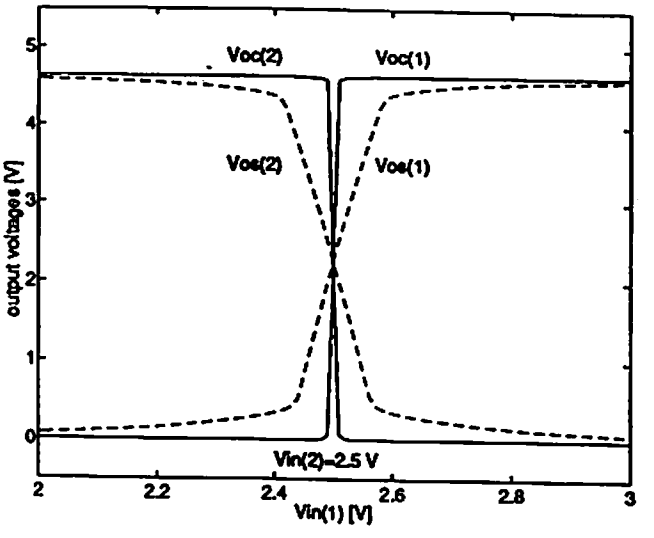
$$I_M = \frac{\beta_1}{2}(V_M - V_{CM} - V_{th})^2, \tag{4.10}$$

and

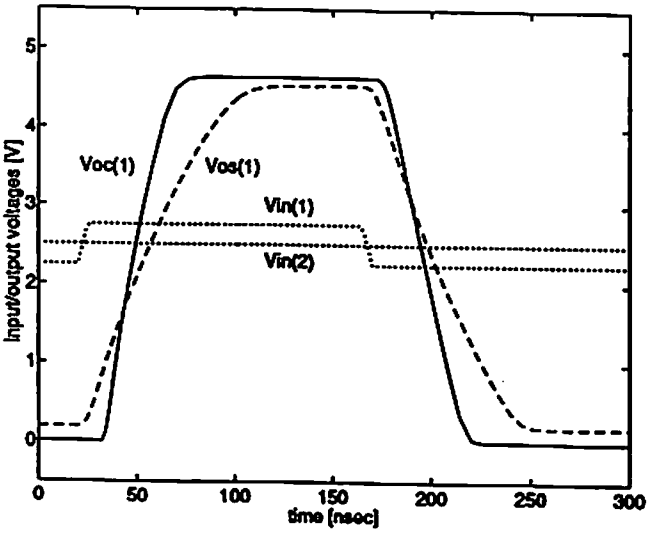
$$I_L = \frac{\beta_1}{2}(V_L - V_{CM} - V_{th})^2, \tag{4.11}$$

respectively. The total current is obtained from (4.8) as,

$$I_{total} = 1 \cdot I_W + M \cdot I_M + (N - M - 1) \cdot I_L = N \cdot I_B. \tag{4.12}$$



(a)



(b)

Figure 4.4: SPICE simulated results of the 2-input WTA circuit. (a) DC transfer curve. (b) Transient behavior of the winning output with C_L of 0.2 pF.

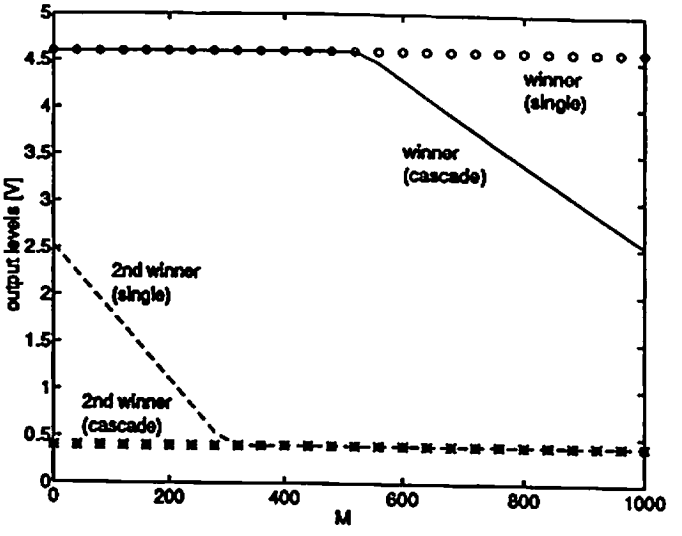
In Fig. 4.5, calculated results for a 1000-input WTA circuit are shown. The voltage levels of V_W, V_M, V_L are set to 2.525 V, 2.500 V, and 2.475 V, respectively. As the number of the second largest input, M, increases, a larger portion of the total bias current flows through this group and the current flowing through the winner cell decreases. This results in reduction of output DC level and increases the rising time of the winning output.

4.4.1 Cascading of Stages

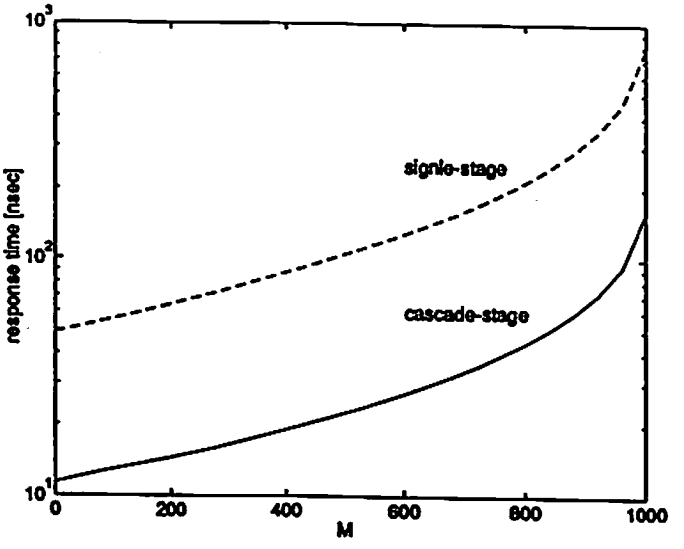
Performance improvement due to cascading of identical single stage is apparent as shown in Fig. 4.4 for a 2-input network. Cascading configuration makes the entire voltage gain of the cell drastically increase so that the transition region between the winner and losers is greatly narrowed. The operation speed is improved since a large load capacitance can be efficiently driven by the stronger output signal. For a large-scale network as discussed in Fig. 4.5, due to the loss of the available current in the winner cell, the output levels of the winner and losers cannot be fully binary (0 and 5 V) although the winning output is still larger than the rest. The corresponding response time becomes quite long since a less amount of charging current is provided for transistor M_3 in the winner cell. By using the cascading configuration, the output voltage level of the winner can be maintained to be saturated toward the positive supply voltage and the operation speed can be greatly increased as shown in Fig. 4.5.

4.4.2 Distributed Biasing

In Fig. 4.3, the total bias current is provided by the transistor M_5 from each cell in the distributed manner instead of having a fixed amount of tail current source.



(a)



(b)

Figure 4.5: Calculation results on a 1000-input WTA with different number of the cells having the second largest inputs. (a) DC level of the winning output. (b) Response time of the winning output ($C_L = 1.0pF$).

Thus, the total bias current is proportional to the number of competitive circuit cells. This approach makes the circuit response time be quite independent of the number of cells. In Fig. 4.6, the simulated response time of the winning output is shown. Here the winning input is 2.6 V and all other inputs are set to 2.5 V. In the fixed bias current case, the response time of the winning output increases rapidly as the number of the cells increases. On the other hand, in the case of the distributed biasing, the response time is almost constant because the available charging current is increased proportionally.

4.4.3 Dynamic Current Steering

If input voltages to multiple cells have very large values as compared to the values to the rest of the cells, then most of the total biasing current is consumed by these cells. In such a case, more than one output voltage might saturate at the positive power supply value. This results from the fact that the currents of these cells are large enough to make output values saturated though they are still smaller than the winning current as shown in Fig. 4.7(a). Here, the critical current, I_θ, is used to describe the value which the current flowing in the cell should exceed in order that the output can be interpreted as a logic-high value. This value is determined from the circuit parameters as shown in (4.5). The circuit schematic shown in Fig. 4.7(b) is proposed to ensure that only the winning output will have the logic-high value by dynamically adjusting the current levels. The multi-input source-coupled configuration consists of transistors M_F's in the cells and the shared M_{F0} transistor. The input to this transistor is the output of the corresponding WTA cell. As the number of competing inputs increases, the circuit can still be easily extended. The drain terminal of transistor M_B in each cell is

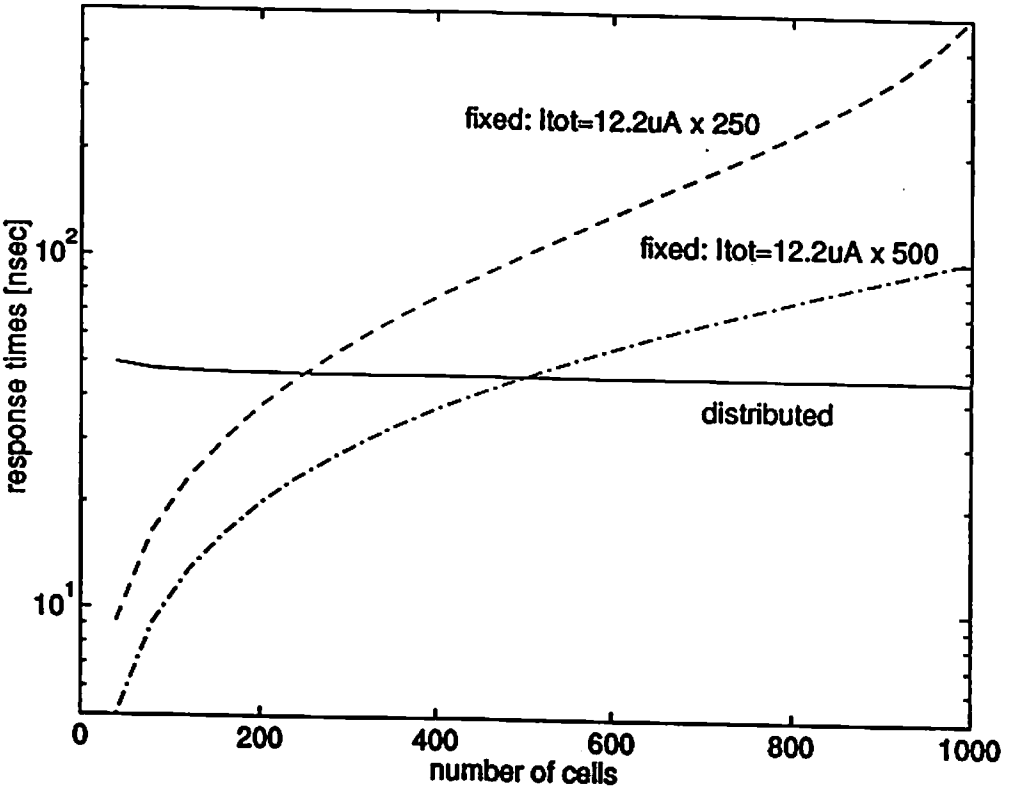


Figure 4.6: Comparison of the fixed and distributed bias currents for a different number of competitive cells.

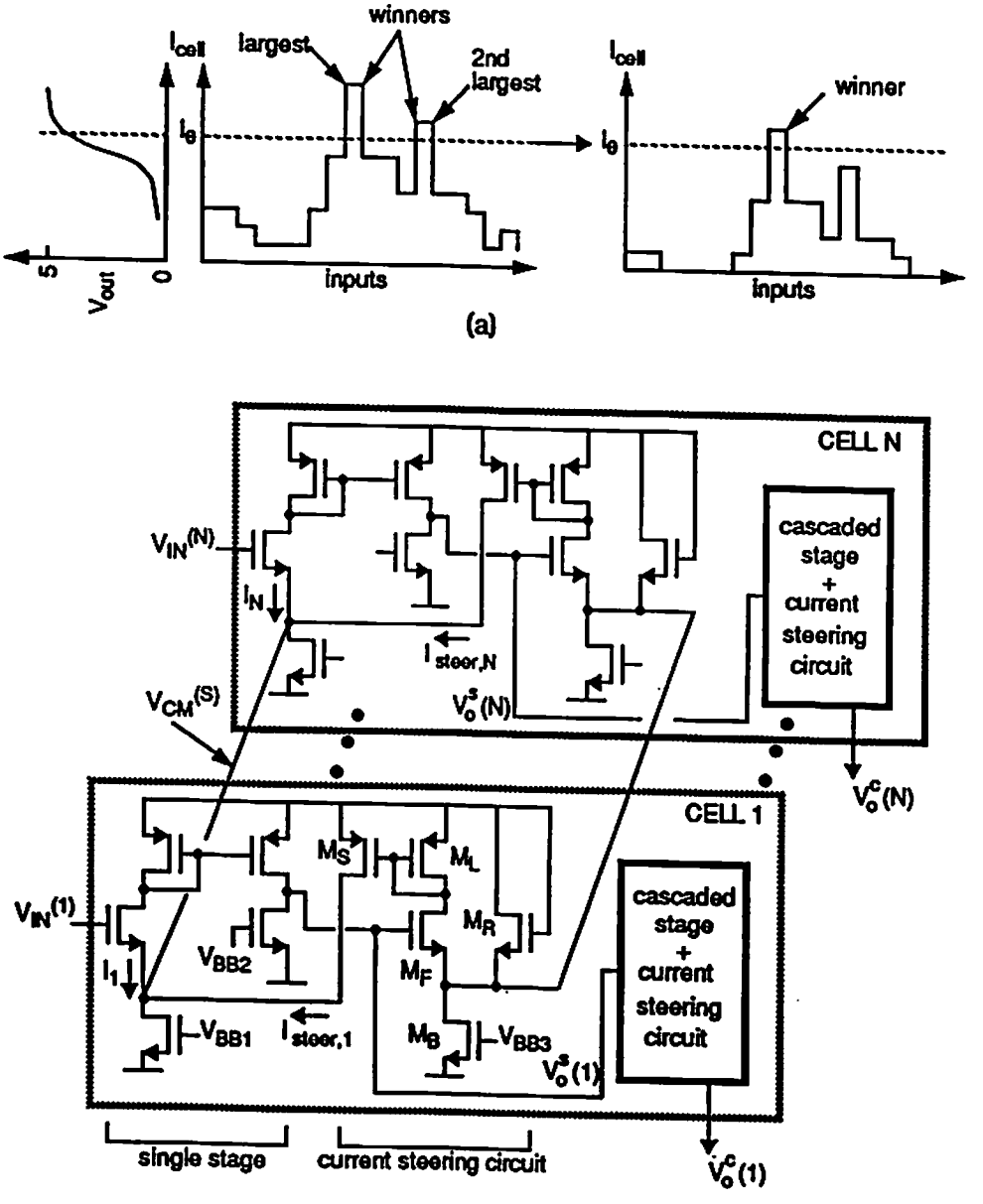


Figure 4.7: Current steering function to ensure only one output high. (a) The operation. (b) Circuit schematic.

tied together to provide the necessary bias current. When the number of outputs with high voltage values is more than one, the currents can flow in transistors M_F and M_S in each cell so that the current in each WTA cell is decreased by the same amount. This steering current is strongly dependent on the number of high outputs. The operation continues until only one output has a high voltage value and the currents of other cells are below the threshold to make the corresponding output voltage values low as shown in Fig. 4.7(a). In this condition, the multi-input differential pair is reduced to 2-input one and all tail source current flows through transistor M_{F0}, since its size is much larger than the M_F's. The amount of steering current in each cell is determined from Fig. 4.7(a) as

$$I_C^{(2nd)} - I_\theta < \Delta I_{steering} < I_C^{(W)} - I_\theta. \tag{4.13}$$

Fig. 4.8 shows the simulation results of the operation of the dynamic current steering circuit for a 10-input WTA circuit. After the current steering technique is applied, large steering current flows since several outputs are high. This reduces the current levels of all outputs. Once all cell currents corresponding to the secondary inputs are below the threshold current, the steering current decreases so that only the winning output rises up to the positive supply voltage. The transistor sizes in one complete WTA cell are listed in Table 4.2.

4.5 Experimental Results

4.5.1 The Winner-Take-All circuit

Each experimental prototype chip consists of 50 cells in a standard TinyChip package provided by MOSIS Service at USC/Information Sciences Institute in

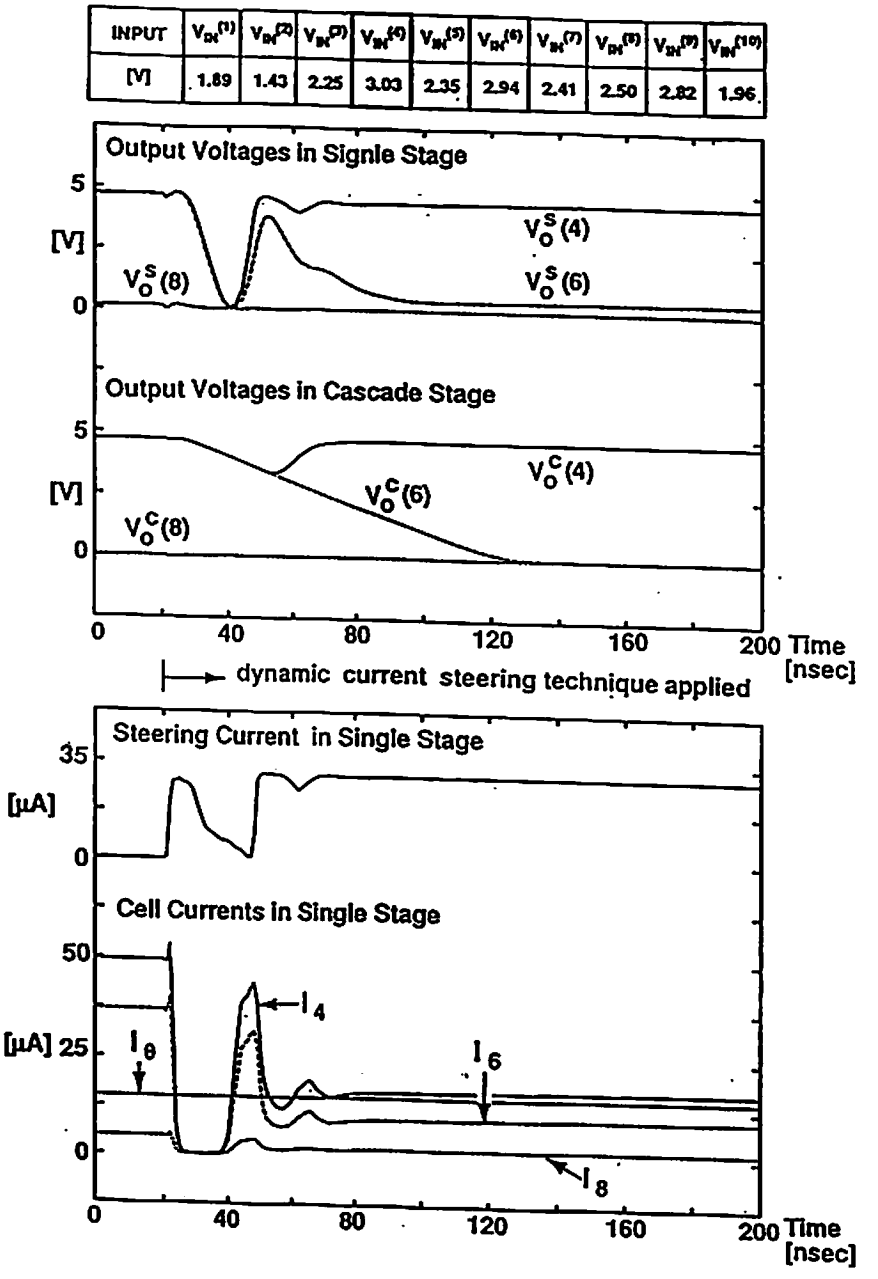


Figure 4.8: SPICE simulation results of a 10-input WTA circuit with the dynamic current steering scheme.

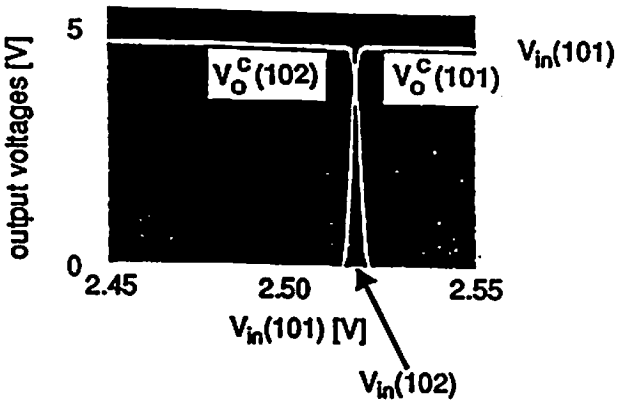
Table 4.2: Transistor sizes of the WTA cell shown in Fig. 4.3.

Transistor	W/L [µm / µm]
M_1	8 / 2
M_2 , M_4	8 / 4
M_3 , M_5	16 / 4
M_F	8 / 2
M_R	4 / 2
M_L	6 / 4
M_S	24 / 4
M_B	12 / 4

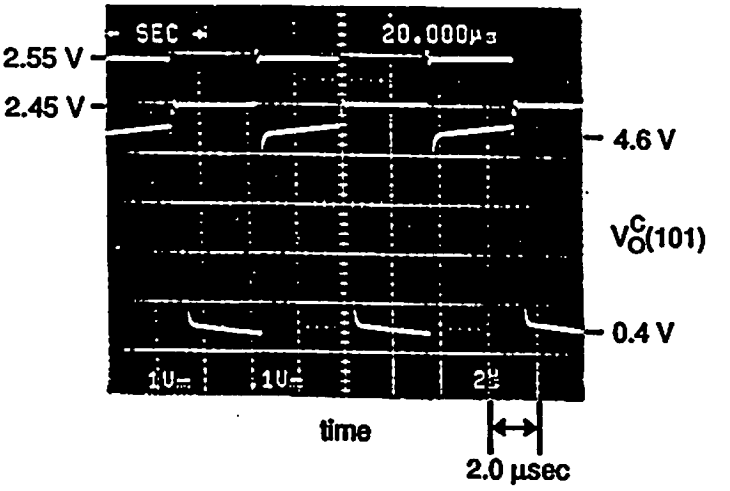
Marina del Ray, CA [105, 106]. Four chips are used to construct the 200-input WTA circuit in the experiment. *Cell* 1 to *Cell* 50 are in *Chip* 1. *Cell* 51 to *Cell* 100 are in *Chip* 2, and so on. This extension can be done by directly connecting the common signal pin from each chip. In the prototype design, the cascade configuration of winner-take-all cells are used. In Fig. 4.9(a), measurement results of the DC characteristics are shown. Here, the input to *Cell* 101 is increased linearly from 2.50 V to 2.55 V along the x-axis. The input of *Cell* 102 is set to 2.52 V, and the other inputs are set to 2.50 V. As V_{in}^{101} exceeds V_{in}^{102}, the corresponding outputs are reversed as the new winner and the loser emerge. All other outputs are near the negative supply voltage. In Fig. 4.9(b), the output waveform of the winner (*Cell* 101) is shown. The input to the winner is 0.1 V_{p-p} pulse around the center of 2.5 V and the load capacitance is 7 pF. The rise time and the fall time are 202 *nsec* and 400 *nsec*, respectively.

In Fig. 4.10, the output behavior of Cell 101 as the winner is shown. The input is set to 2.58 V. The second largest inputs, and the other inputs are set to 2.50 V, 2.42 V, respectively. The output levels and the response time against the number of the second largest inputs are shown in Fig. 4.10(a) and (b), respectively. The solid lines show the calculated results. All output levels of the winner are above 90 % of the full operation range. The operation speed can be significantly increased when the circuit is integrated with other blocks in the VLSI chip because the internal load capacitance will be much less than 7 pF of the measurement setup.

Fig. 4.11 shows the variation of the competition threshold for a winner across the chips. Five curves correspond to the outputs of *Cell* 1 (*Chip* 1) with different positions of the second largest input (set to be 2.52 V) at *Cell* 2 (*Chip* 1), *Cell* 50 (*Chip* 1), *Cell* 100 (*Chip* 2), *Cell* 150 (*Chip* 3), or *Cell* 200 (*Chip* 4), respectively.



(a)



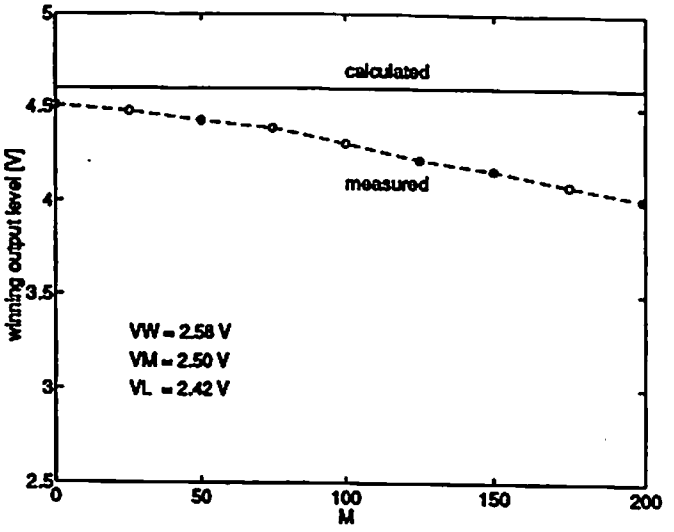
(b)

Figure 4.9: Measured results of a 200-input WTA circuit. (a) DC characteristics of the output in the winner and another cell. (b) Transient behavior of the output in the winner.

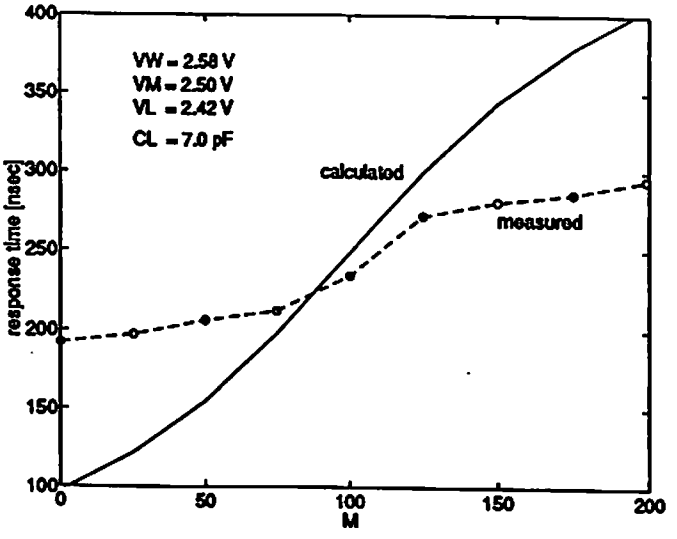
The other inputs are set to 2.50 V. The variation range is shown to be less than 15 mV. Apparently, intra-chip variation has been found to be less than the inter-chip variation.

Figure 4.12 shows the measured operation of the 10-input WTA circuit with the current steering circuit. In Fig. 4.12(a), $V_{in}(1)$ is the square wave between 3.4 V and 3.9 V, $V_{in}(2)$ is set to 3.5 V. All other inputs, $V_{in}(3)$ to $V_{in}(10)$, are 2.0 V. When the $V_{in}(1)$ pulse goes up to 3.9 V, the winning input is $V_{in}(1)$ and the corresponding output rises to 5 V. On the other hand, when $V_{in}(1)$ goes down to 3.4 V, the winning input is $V_{in}(2)$ and the corresponding outputs are flipped. In Fig.. 4.12(b), the winning input $V_{in}(1)$ is set to 3.75 V, the second largest input $V_{in}(2)$ is set to 3.50 V, and all other inputs are 2.0 V. The upper curve is the bias control signal so that the dynamic current steering circuit can operate for $V_{BBC} = 1.5V$ and can be disabled for $V_{BBC} = 0.0V$. Only one output is ensured to have a high output value corresponding to the winner with the dynamic current steering circuit.

Figure 4.13 shows the die photo of a 50-input WTA circuit and an enlarged picture of one cell, which consists of the cascaded stages without the current steering circuit. Each cascaded competitive cell occupies 58 μm x 96 μm in a scalable 2-μm CMOS technology. The power dissipation per each cell is 120 μW at a 5 V supply voltage. Additional silicon area and power consumption are required to support the various performance-enhancing schemes. Silicon area and power consumption are doubled for the cascading configuration because two identical stages are used. The dynamic current steering scheme requires an additional 25 % of the silicon area.



(a)



(b)

Figure 4.10: Experimental results of a 200-input WTA with different number of the cells having the second largest inputs. (a) DC level of the winning output. (b) Response time of the winning output.

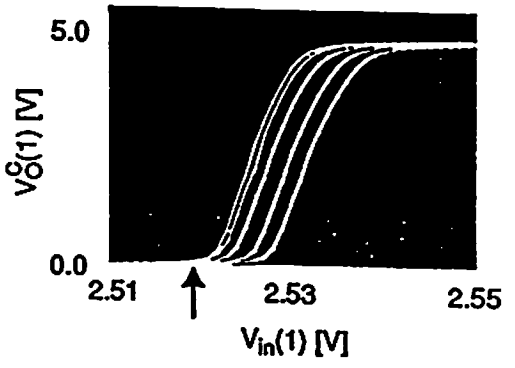


Figure 4.11: Measured results of the variation of the competition threshold for the winner. Fiver curves correspond to the output voltages of the winner with different positions of the cell having the second largest input, 2, 50, 100, 150, and 200 from the left to the right.

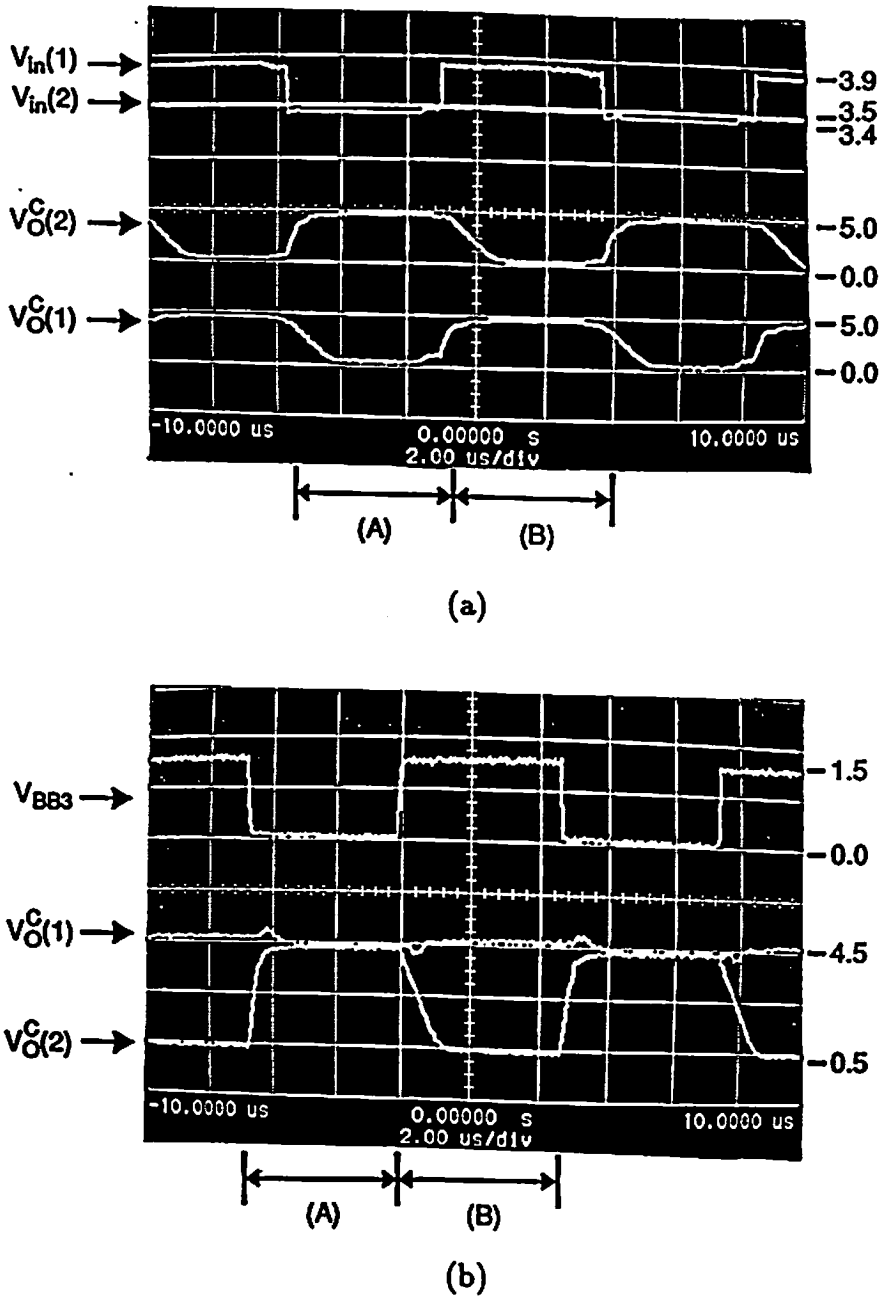
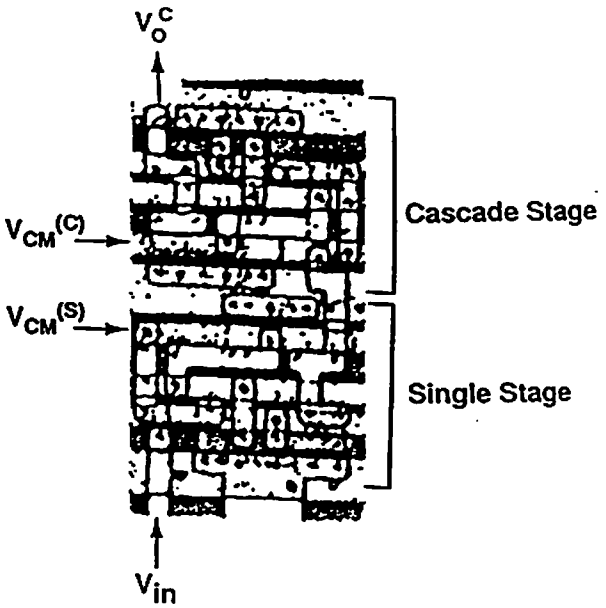
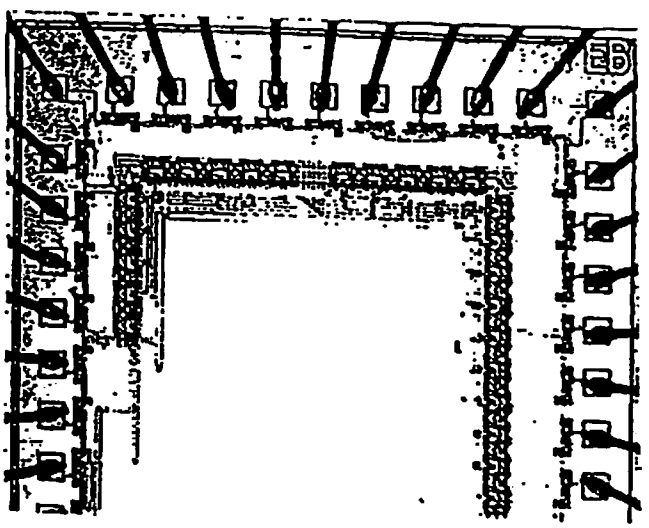


Figure 4.12: Experimental results of a 10-input WTA circuit with the dynamic current steering technique. (a) Experiment 1. (b) Experiment 2.



(a)



(b)

Figure 4.13: Die photos of the WTA circuit fabricated in a 2.0-μm CMOS technology from MOSIS. (a) Enlarged die photo of the one WTA cell containing the cascaded stage. (The current steering circuit is not included in this die photo.) (b) Die photo of a 50-input WTA circuit.

4.5.2 Self-Organization Neural Network

Figure 4.14 shows the measured result of the linear multiplier performing the distance-measure calculation. The multiple curves correspond to the weight voltages of -1.0 V, 0.0V, and +1.0V from the left to the right.

Figure 4.15 shows the simulated behavior on one benchmark problem for the network with a 64-input winner-take-all circuit. The input pattern is obtained from the distance-measuring network in [71]. Since the differences between the winning input and others are not large enough, the intermediate output levels are produced by the single stage configuration. Cascading makes the outputs as fully binary logical values and the dynamic current steering technique ensures only one winner.

The physical layout of a self-organization neural network processor chip is shown in Fig. 4.16. The die size is 4.6 x 6.8 mm^2 in the standard SmallChip frame provided by MOSIS Service [105, 106]. The number of input neurons is 25 and the number of output neurons is 32. There are 800 fully-connected synapses between them. Thirty-two WTA outputs are digital values and encoded by the 32-to-5 digital encoder for efficient interfacing outside the chip.

The network size within the one chip is limited by the area of the building blocks. In the proposed design, the lateral interaction is done only in the common signal lines among the winner-take-all cells, which makes the easy scaling of the network size without additional complicated devices. Figure 4.17 shows the system architecture combining 4 chips and the relevant supporting sub-system environment.

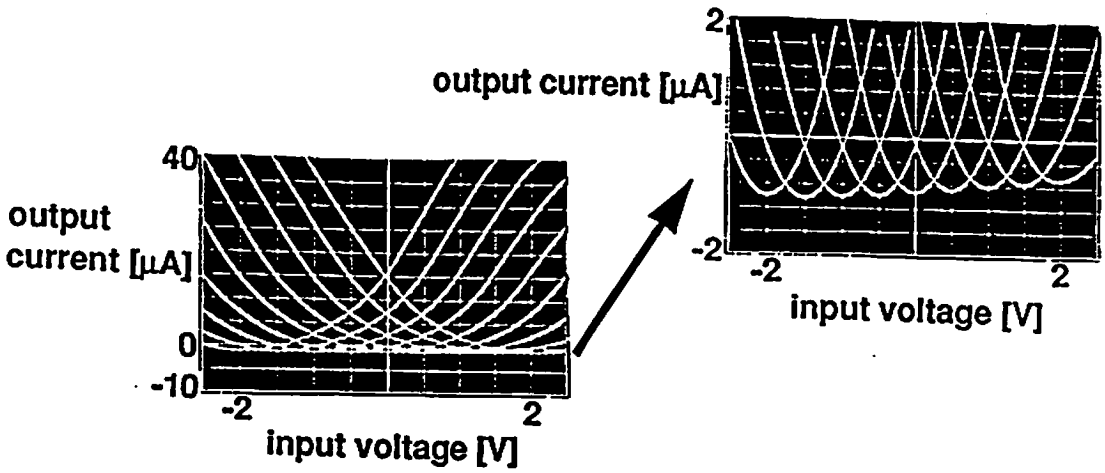


Figure 4.14: Measured DC characteristics of the synapse cell computing the distortion or the distance measure using the linear multiplier. The square of the difference between the input and the weight value is calculated.

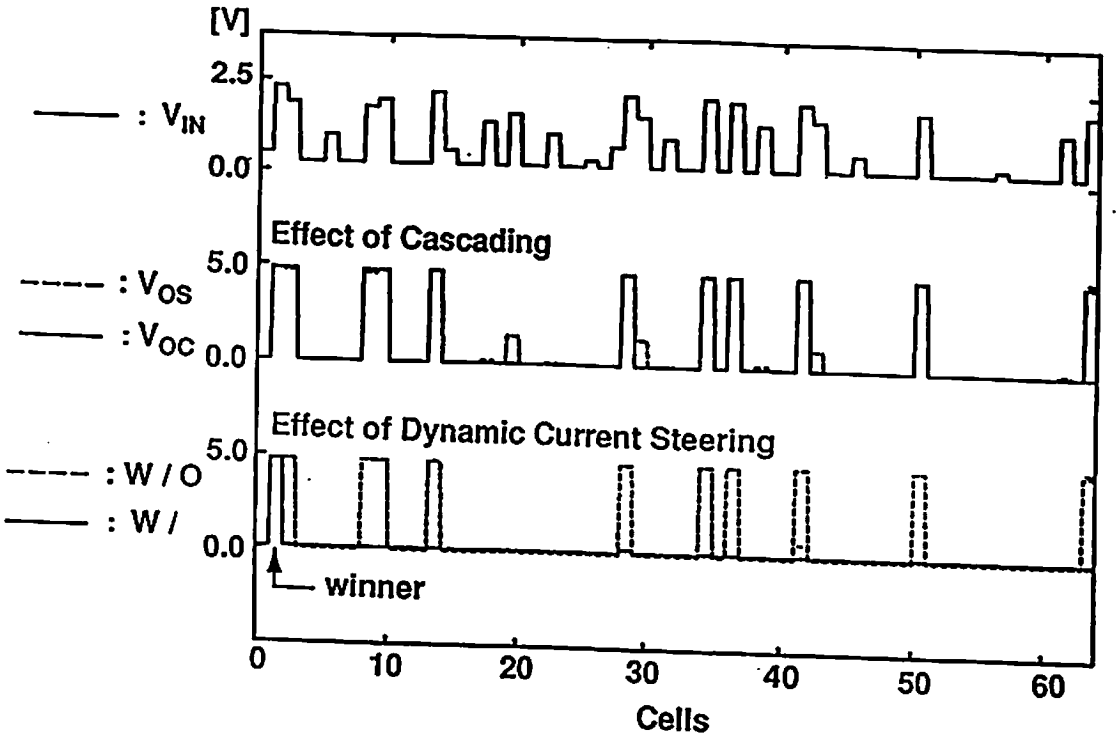


Figure 4.15: Simulated behavior of the WTA circuit for one benchmark problem shown performance improvements. The input values to the WTA circuit are obtained from the synapse matrix and the output neuron array.

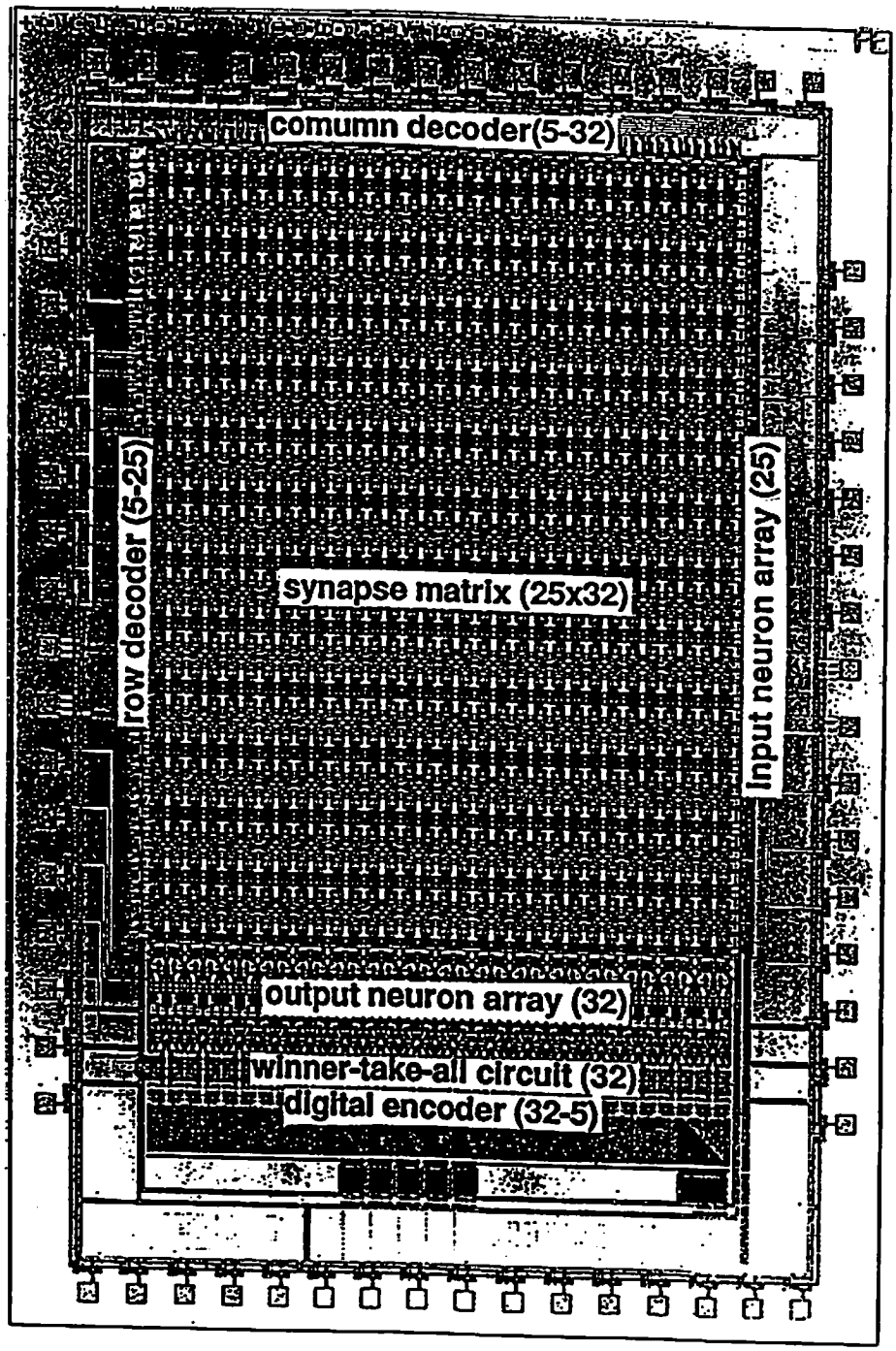


Figure 4.16: Die photo of the self-organization neural network processor chip. There are 25 inputs neurons, 32 output neurons, and 800 synapses.

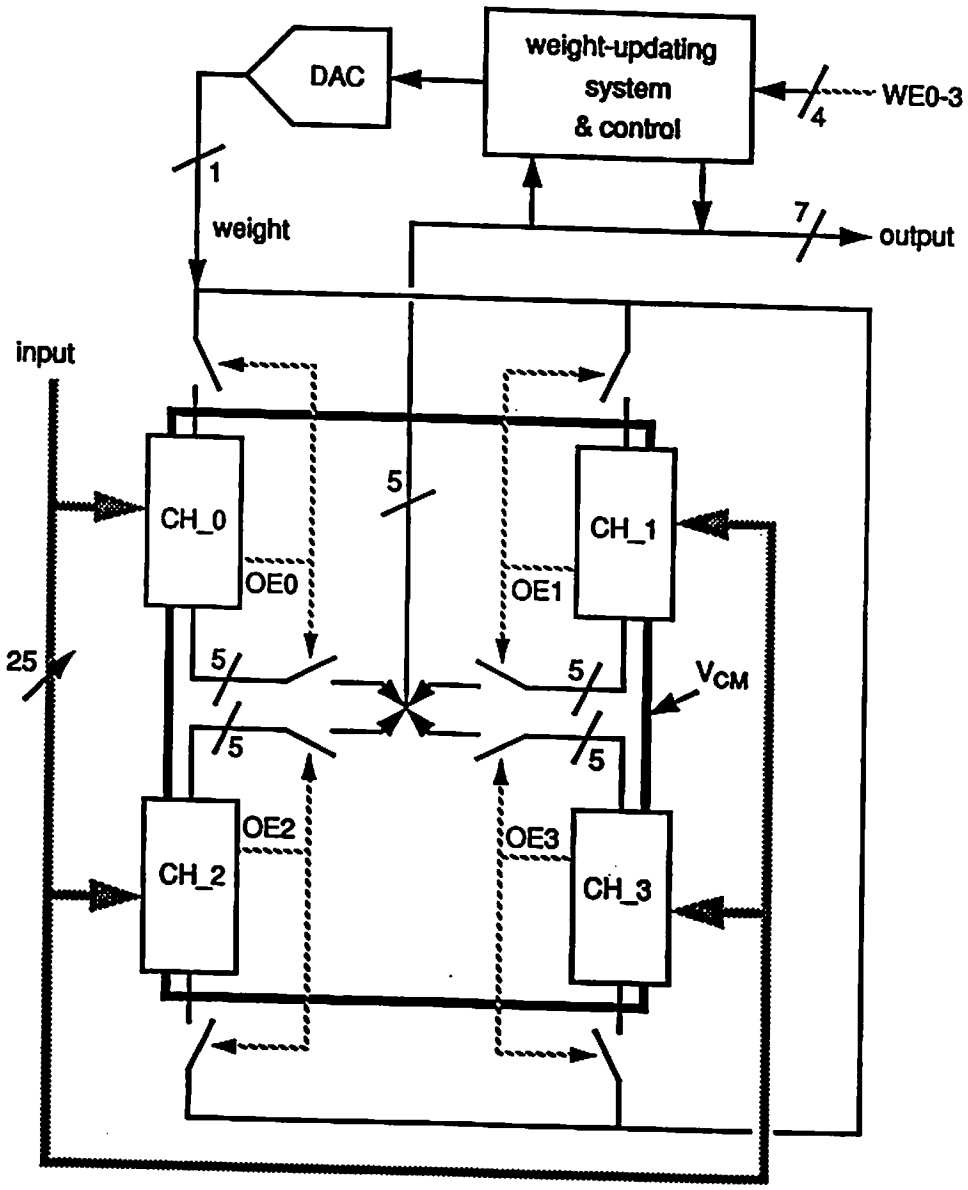


Figure 4.17: Block diagram of extending the dimension of the self-organization neural network processor.

Chapter 5

Analog VLSI Neuroprocessor for Communication

5.1 Background

Rapid integration of communication, computer, and multi-media technologies has become the key driving force toward global information networking for scientific and engineering applications. A growth in data communication has created a strong need for building adaptive filters to overcome problems inherent in the transmission channels. Traditionally the adaptive filter is cascaded with an unknown linear channel which has an associated transfer function. The purpose of the equalizer is to approximate the inverse of the transfer function in order to compensate the undesired channel response which may vary in time in most mobile communications.

The performance of digital communication systems over band-limited, multi-path fading channels is largely determined by the ability to overcome channel impairments introduced during signal propagation. Various compensation techniques [118] such as maximum-likelihood sequence estimation (MLSE) and the decision-feedback equalizers (DFE) were developed and extensively used. The

MLSE method which is generally implemented by the Viterbi algorithm, can result in an optimum data detection in terms of the error rate performance. This optimum performance is achieved with the exact knowledge of channel characteristics. Since the channel response is also estimated from received signals, any error in the estimated channel response may cause performance degradation. The complexity increases exponentially as the number of channel states increases. Since it also requires a large memory size, the MLSE method severely limits the practical applications. On the other hand, the DFE method is used in practical communication systems due to their relatively simple architecture and computation complexity. The performance, however, is not optimal and may be degraded severely when an incorrect estimate is fed back to the equalizer.

In this chapter, an analog VLSI neural network processor for channel equalization is described [42]. The performance of the proposed neural network receiver is superior to those of conventional approaches [119]. Prior estimation of the channel characteristics is not required. Furthermore, unlike most of the other detection algorithms, the noise statistics for the proposed receiver needs not to be a white Gaussian.

Each programmable synapse cell is made of a compact and wide-range analog multiplier circuit. The arrays of the input neurons and the output neurons are optimized for large current-driving capability. The floor-planning of the processor chip makes good used of the silicon area. The neural network processor chip performs the feedforward operation and a companion DSP board executes the back-propagation algorithms. This approach is appropriate for quick system-level demonstration. In addition, the synapse weight is stored permanently and precisely in the digital memory with system back-up on the hard-disc or magnetic

tape, which makes the proposed learning system be non-volatile in contrast to the purely analog on-chip learning scheme, which will lose the weight information if the electric power is turned off.

The chip was fabricated in a 2-μm double-polysilicon CMOS technology from the MOSIS Service. The well-characterized building blocks for analog neural network processors are optimized for the proposed communication receiver. Such an approach makes the design and improvement of VLSI hardware more quickly.

5.2 System Architecture

The block diagram for the neural-based equalizer is shown in Fig. 5.1. In order to accommodate the possible rapid changes in channel characteristics, network training is performed by the extended Kalman filtering (EKF) algorithm which has been widely used for the identification of nonlinear dynamic systems. Although the system behavior is quite sensitive to the initial conditions of the states, it provides a good solution to improve the slow convergence speed, and the imperfect operation of the conventional back-propagation training methods for a network with moderate complexity. The advantage of this scheme over the conventional equalizer of the MLSE method lies on the fact that the channel estimator and in principle, the matched-filter are not needed because they are already embedded in the training operations. The EKF training algorithm is executed on a companion digital signal processor and is interfaced to the analog VLSI neural network processor through the control circuitry.

The neural network processor consists of a 4-layered feedforward network and is built with compact analog circuit cells. The input signal is first sampled at a rate greater than or equal to the data symbol rate. As the sampled input signal

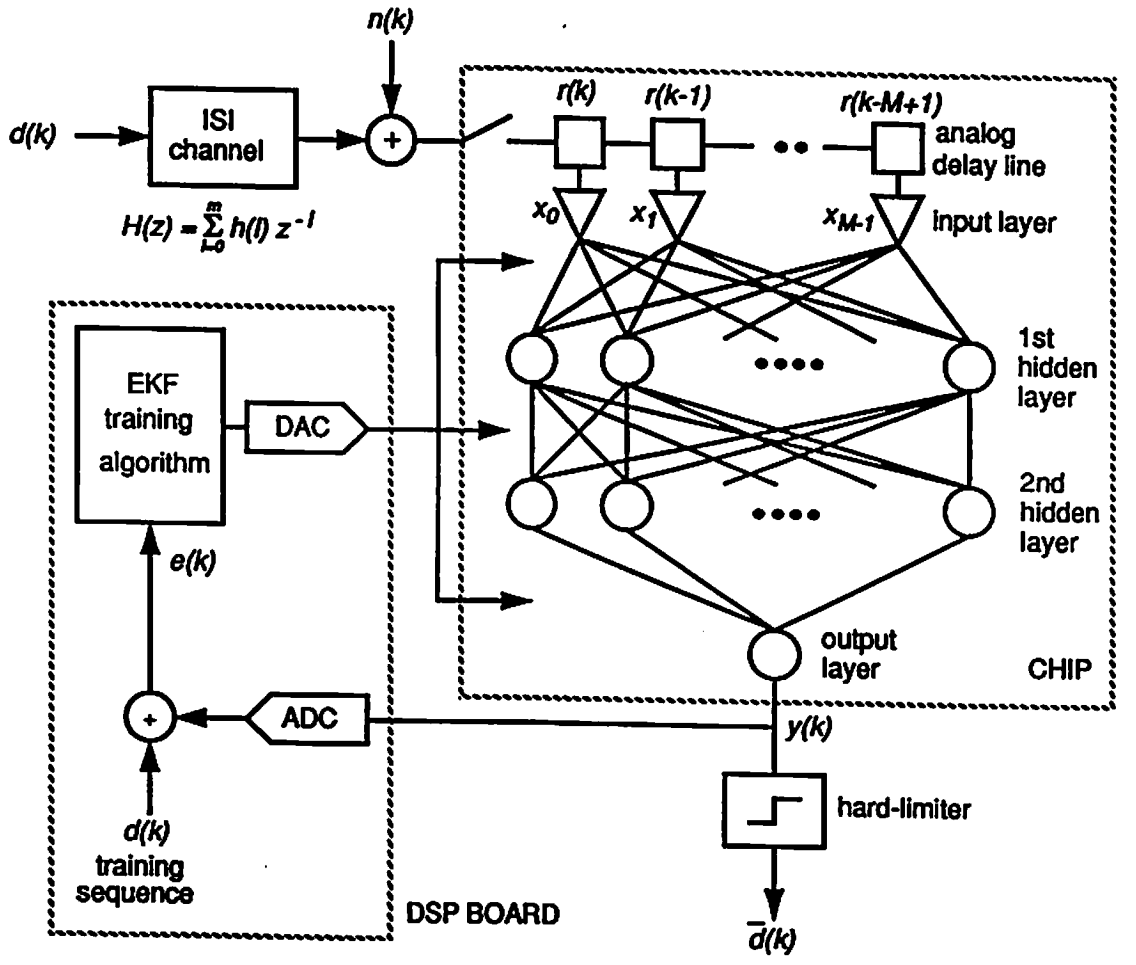


Figure 5.1: Block diagram of the neural-based communication receiver and the inter-symbol interference (ISI) channel.

$x(k)$ propagates into the analog delay line, its delayed-versions are applied to the network in parallel. Here, the analog delay line serves as the input layer. The network has two hidden layers and one output layer. Since we deal with binary pulse-amplitude modulation (PAM) communication systems, the output layer contains only one node.

In the analysis, we assume for simplicity that the symbol timing is perfect so that the residual inter-symbol interference due to transmitter and receiver Nyquist base-band filters is negligible. Also the frequency offset between the received and the local oscillator signals is assumed to be zero. By using the discrete-time model, the received signal $r(k)$ is a convolution of the transmitted signal $d(k)$ and the channel response $h(k)$, plus additive white Gaussian noise $n(k)$ with a zero mean and the variance δ^2,

$$r(k) = d(k) * h(k) + n(k) = \sum_{l=0}^{m} h(l)d(k-l) + n(k). \tag{5.1}$$

Then the input to the network becomes

$$x_i = r(k-i) \quad for\ i = 0 \cdots M-1, \tag{5.2}$$

where M is the number of the input nodes in the input layer. This number is to be greater than or equal to the maximum delay-spread of the signal through the channel, normalized by the symbol duration T_S. The number of neurons in the hidden layers can be judiciously chosen and has to be supported by the analog VLSI processor chip.

The complexity of the EKF algorithm is a quadratic function of the total number of nodes. For a network with M input nodes, it can perform equalization

up to M-ray channel models. In comparison, the complexity of a DFE with standard recursive least square (RLS) algorithm is proportional to N^2, where N is the number of tabs in the equalizer. The MSLE receiver requires the operations proportional to P^Q for the P-ary system, where Q is the number of states to be estimated.

5.3 Training Algorithm

The generalized-delta rule is a simple training algorithm in which the weights W_k at time k are updated in order to decrease the average of squared errors

$$E = \sum_{p \in \Omega} E_p = \frac{1}{2} \sum_{p \in \Omega} (d_p - o_p)^2, \tag{5.3}$$

where Ω is a set of training sequences, d_p and o_p are the desired and actual outputs, respectively.

Here, we assumed that the channel characteristics are totally unknown and no channel estimation mechanism is proveded. The lack of *a priori* information on the channel implies that a exact decision boundary is not available for the network training and thus the correspondence between the network input and the output does not exist in general. In the real-world communication systems, the preamble sequence which is normally used for synchronization and estimation of the channel response at the receiver with stored replica, may corresponds to the training sequence. Duration of two consecutive preamble sequences is long enough such that they enclose a packet of data, but short enough compared to the rate of changes in channel characteristics, which thus can be regarded as constant values during the interval. Therefore, in the present application, learning

is performed every time a new training sequence is received and the obtained weights are effective until the next update. If the received data following the training sequence is buffered, the trainings need not to be real-time.

In our design, the extended Kalman filtering algorithm is used to provide faster network training [120]. In each iteration, the EKF algorithm approximates the optimum estimate by expanding a nonlinear system function into the Taylor series around the normal operating point and discarding the higher-order terms. For the given nonlinear system model,

$$d(k) = o(k) + z(k) = f[W(k), x(k), k] + z(k), \quad k > 0, \tag{5.4}$$

where W is the state of the network, x is the input, and z is the observation noise. An iterative method of updating the state can be expressed as,

$$K(k) = P(k-1)H^T(k)[R(k) + H(k)P(k)H^T(k)]^{-1}, \tag{5.5}$$

$$P(k) = P(k-1) - K(k)H(k)P(k-1), \tag{5.6}$$

$$e(k) = d(k) - o(k)' \tag{5.7}$$

and

$$\hat{W}(k) = \hat{W}(k-1) + e(k)K(k). \tag{5.8}$$

Here K is the Kalman gain, $R = E(zz^T)$ is the covariance of the observation noise, and H is the partial derivative of $f(\cdot)$ with respect to W, evaluated at the point $\hat{W}$, *i.e.*,

$$H^T(k) = \left[\frac{\partial f(W(k), \cdot, \cdot)}{\partial W(k)}\right]_{W(k)=\hat{W}(k)}, \tag{5.9}$$

for given $d_1, D_2, \cdots, d_k$. The nonlinearity and its dynamic behavior are denoted by $f(\cdot, \cdot, k)$.

5.4 VLSI Implementation

Figure 5.2 shows the block diagram of the prototyping analog VLSI neural network processor chip. This VLSI processor chip contains four layers: the input layer, two hidden layers, and the output layer. The numbers of neural units in the input layer, two hidden layers, and the output layer are 8, 12, 12, and 1, respectively. The input layer is constructed by an array of input neurons incorporating the switched-capacitor analog delay circuits. The hidden layers consist of the combination of the output neurons for the present layer and the input neurons for the next layer. The output layer contains only one output neuron. The whole network includes a total of 252 synapse cells, which are fully connected between the corresponding adjacent layers.

The building components such as the synapse cell, the input neuron and the output neuron were described in detail in the previous chapter. In monolithical integration of the multi-layered network into one micro chip, the input signals are used in the single-ended format instead of the differential approach in order to reduce the number of interconnection lines and to save the silicon area. The slightly degraded performance can be circumvented during the network learning process.

The input neuron in the front-end portion of the network is optimized for the communication receiver. The input voltage to a given input neuron is the delayed version of the voltage to the preceding input neuron and an analog delay circuit is used [121]. Figure 5.3 shows the schematic diagram of the proposed

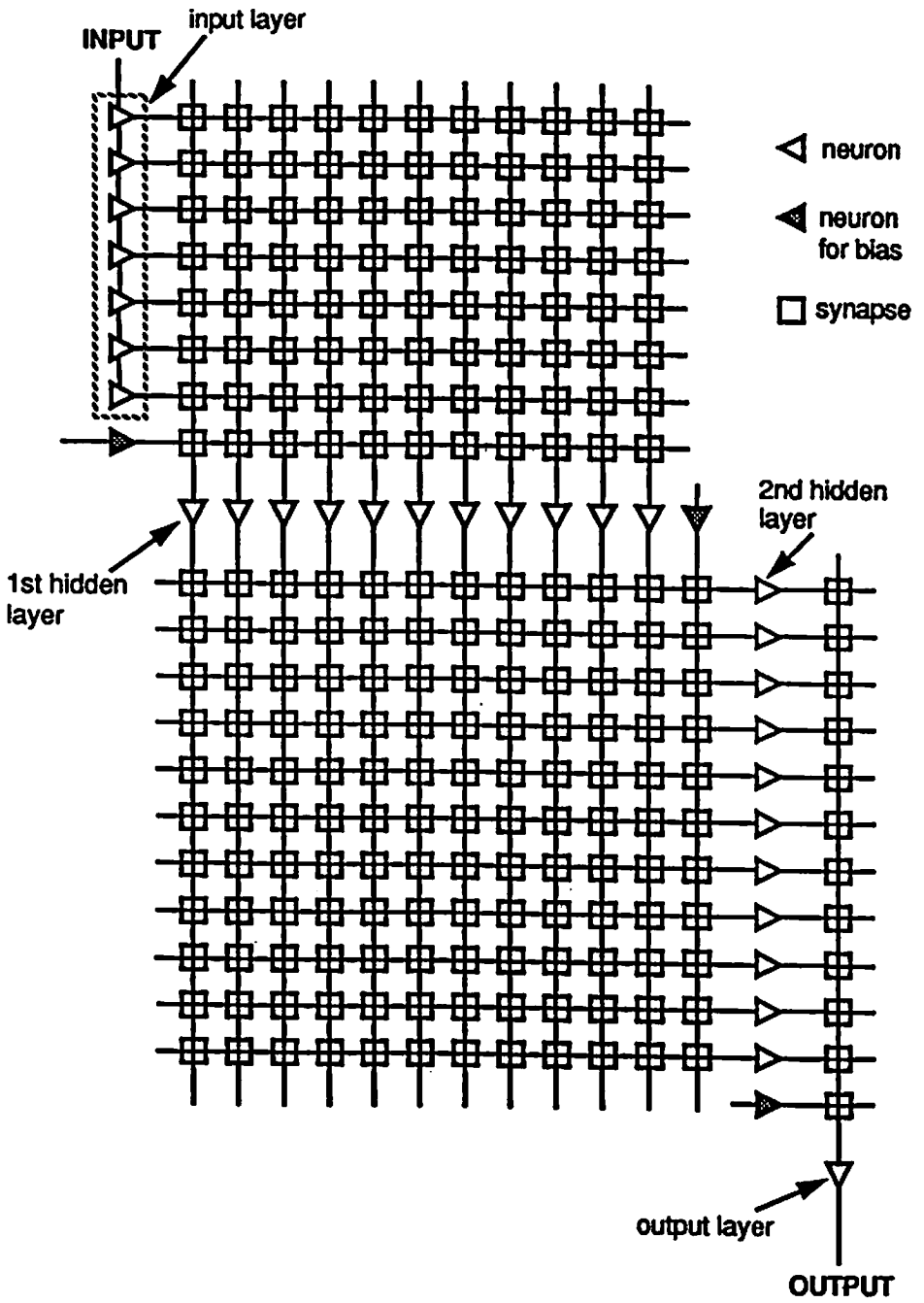


Figure 5.2: Block diagram of the VLSI neuroprocessor for digital communication receiver.

first layer. which is implemented by the switched-capacitor analog delay circuit scheme. Each sample-and-hold (S-H) circuit is connected in parallel and an array of switches operates as the rotating connector. The unity-gain amplifier in each S-H circuit functions as an input neuron for the first layer. The parallel connection in the delay chain avoids accumulation of the offset voltages from the unity-gain amplifiers, which could become very significant in the cascaded version of S-H circuits. The input operational amplifier buffers the input signal and establishes the feedback loop so that any error due to the finite amplifier gain and the offset voltage in the input neuron is divided by the voltage gain of this input amplifier. Performances on the offset voltages of the operational amplifier are simulated using the SWITCAP switched-capacitor network simulator [122] and shown in Fig. 5.4. A DC voltage of 1 V is applied and the offset voltage is assumed to be 10 mV. Apparently, the cascaded version suffers from the accumulation of the offset voltages along the delay line. In the proposed parallel version, the improved feedback configuration makes all nodes having the same offset voltage of the operational-amplifier building block. Since the error voltages due to the offset of the operational amplifier are accumulated in the output neuron, their effects are critical as shown in Fig. 5.4(b).

Prior to the input operational amplifier, a D/A converter converts the digital input signal into the analog value. In addition, one A/D converter is dedicated to receiving the output data from the network. All required clocking and control signals are generated by the digital finite state-machine circuitry built in the same chip. They are globally synchronized with the external control clock which comes from the DSP chip.

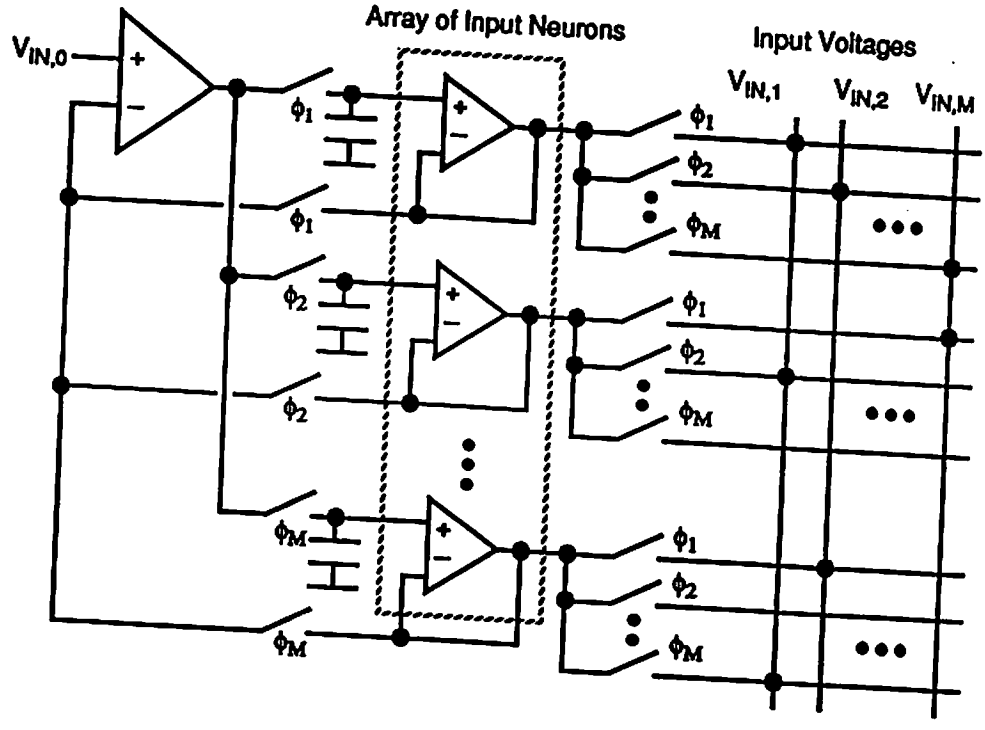
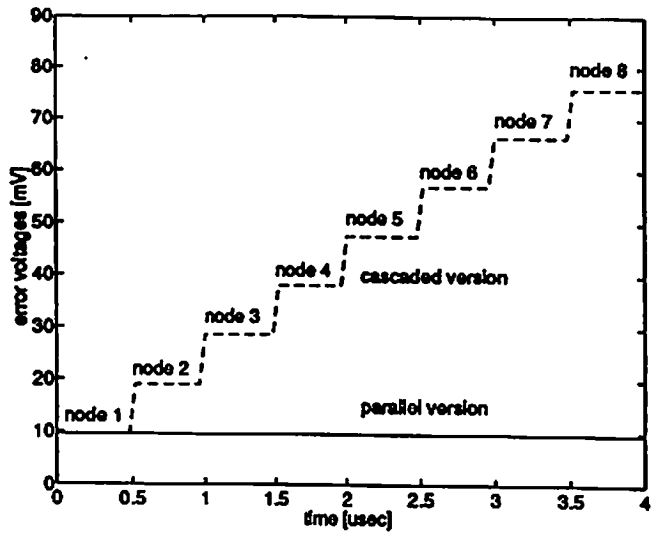
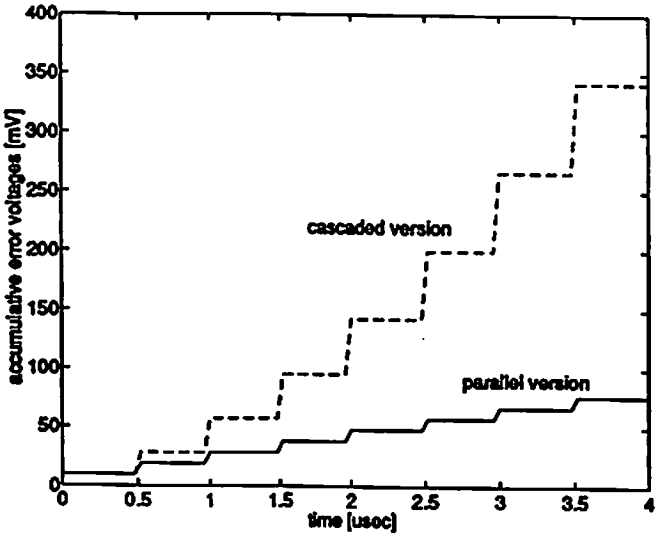


Figure 5.3: Circuit diagram of the input layer with the switched-capacitor delay line.



(a)



(b)

Figure 5.4: SWITCAP simulation results of the switched-capacitor analog delay circuits. (a) Node error voltages for the offset voltage of the operational amplifier being 10 mV. (b) Accumulative error voltages reflected to the output neuron.

5.5 System Environment

The fabricated chip was mounted on the custom-made circuit board. Standard IC's were used to support the efficient manipulation of control signals. This board is interfaced to an IBM/PC-AT through the DSP56000ADS digital signal processing board and the DSP56ADC16 data conversion board from Motorola Inc. [123]. The host computer is also used for displaying the output results. Figure 5.5 shows the interfacing block diagram for forward and learning processing operations.

In the learning experiments, the DSP board controls the entire operations by providing the control and the clock signals for global synchronization. It executes the learning algorithm. The companion interface board is dedicateed to data conversion from the digital memory to the analog storage in the neural network chip. The digital weight signals calculated in the DSP board are converted into the analog values by an embedded digital-to-analog converter and into the differential signals by additional operational amplifiers. Another digital-to-analog conversion block is used for the input signal voltage from the communication channel. The analog-to-digital conversion block sends the output data from the neural network chip to the DSP board. They consist of the data converters, the analog scaler, the level-shifter, and the digital data latches. The control circuitry on the custom-made board produces the required clocking signals by frequency-dividing of the global clock and generates the synchronized addresses for storing analog synapse weight values on the neural network chip.

The clock frequency for updating one synapse cell is 48 kHz when the serial interface is used. The corresponding refresh cycle for the entire network is 252/(48·

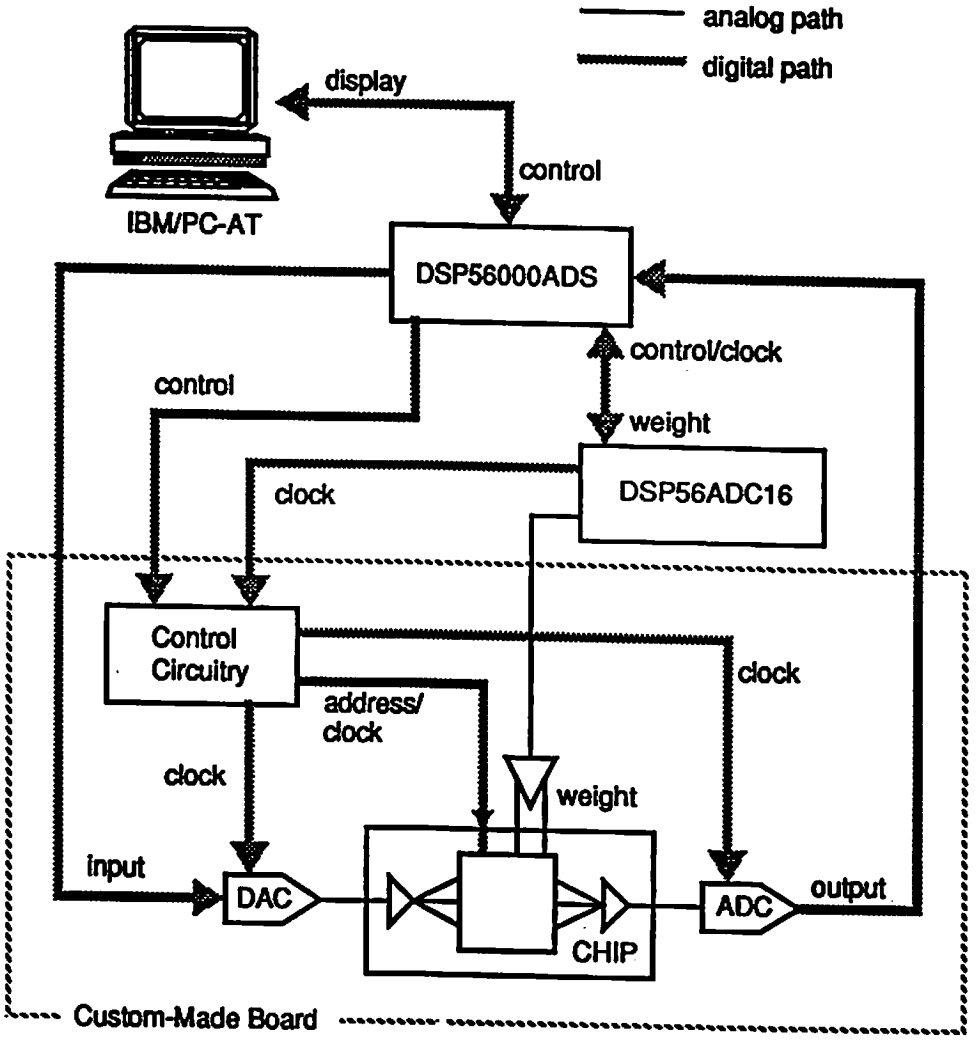


Figure 5.5: Block diagram of the neural network processor system.

10^3) = 5.25 *msec*. This value is long enough to prevent the synapse value from changing due to the leakage current. Notice that the clock rate can be significantly increased up to 2-3 MHz by using the advanced DSP chips instead of the low-cost general-purpose DSP board. The input/output sampling clock rate of the network is 1.024 MHz.

5.6 Experimental Results and System Analysis

A neural network processor chip was specially designed for the specific 4-layered perceptron network. The prototype chips was fabricated in a 2-μm double-polysilicon CMOS technology from the MOSIS Service of USC/Information Sciences Institute at Marina del Rey, CA [105, 106].

Figure 5.6 shows the measured transient characteristics of the 4-layered neural network. The input signal was applied to the input neuron of the first layer and the output waveform was monitored from the output neuron of the final layer. The response time of the neural network processor chip is 0.66 μsec. In the experiment, the AD7524 chip was used in the preprocessing step to perform the digital-to-analog conversion for the input signal. The TLC5502 chip was used in the postprocessing step to perform the analog-to-digital conversion [124]. Their response times are 100 *nsec* and 50 *nsec*, respectively. The total operation time for the entire feedforward path is 810 *nsec*, which is well suited for the designated operating frequency of 1.024 MHz.

Figure 5.7 shows the die photo of the prototyping 4-layered perceptron chip for the communication receiver applications. This neural network processor chip occupies an area of 4.6 x 6.8 mm^2. In our design, all output nodes of the hidden

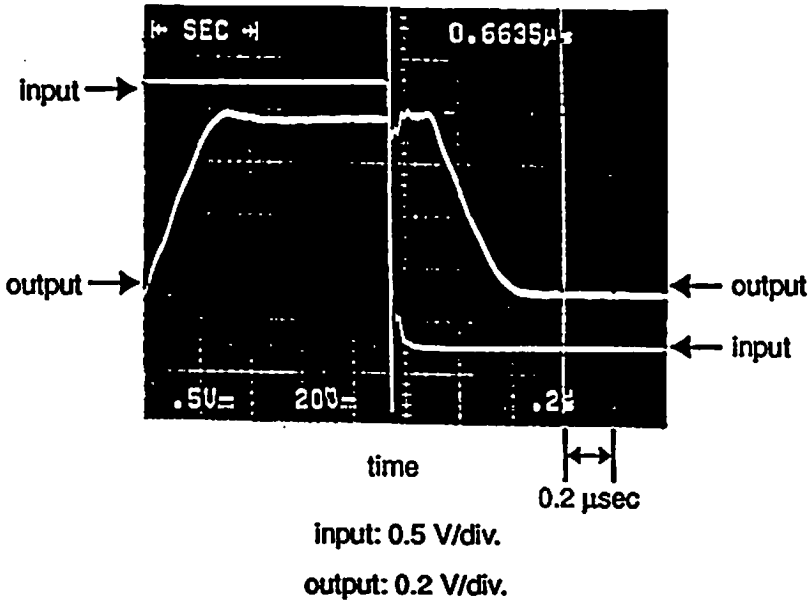


Figure 5.6: Measured waveforms of the input and the output voltages for the retrieving process.

layers and the output layer can be accessed externally for easy debugging and monitoring of the network operation.

System analyses were performed for both the minimum-phase and nonminimum-phase channels [125]. Figure 5.8(a) shows a decision boundary of the network for the channel with the minimum-phase transfer function of $H(z) = 0.89443 + 0.44721z^{-1}$ and a signal-to-nose ratio (SNR) of 6 dB. As the number of training symbols increases, the decision boundary closely approximates that of the optimum receiver. The equalizer is trained against the additive noise as well as the multi-path fading. Figure 5.8(b) shows a decision boundary of the equalizer when the channel has a transfer function of $H(z) = 0.44721 + 0.89443z^{-1}$, which is a typical nonminimum-phase characteristics. Here the signal-to-noise ratio is 12 dB.

In Fig. 5.9, the convergence speed is plotted for several signal-to-noise ratio values. The mean-squared-error of the network output was monitored in the experiment. Figure 5.9 shows the results for a channel with the transfer function of $H(z) = 0.89443 + 0.44721z^{-1}$ and Fig. 5.9(b) shows the results for a channel with the transfer function of $H(z) = 0.407 + 0.815z^{-1} + 0.407z^{-2}$. Notice that the equalizers settle to their normal operating values within 300 iterations for the minimum-phase channel and 700 iterations for nonminimum-phase channel.

Figure 5.10 shows the measured error rate as a function of the signal-to-noise ratio for different channels. In the experiments, the neural network receiver was trained with 2,500 symbols and the error rate was averaged over the execution of 10,000 symbols.

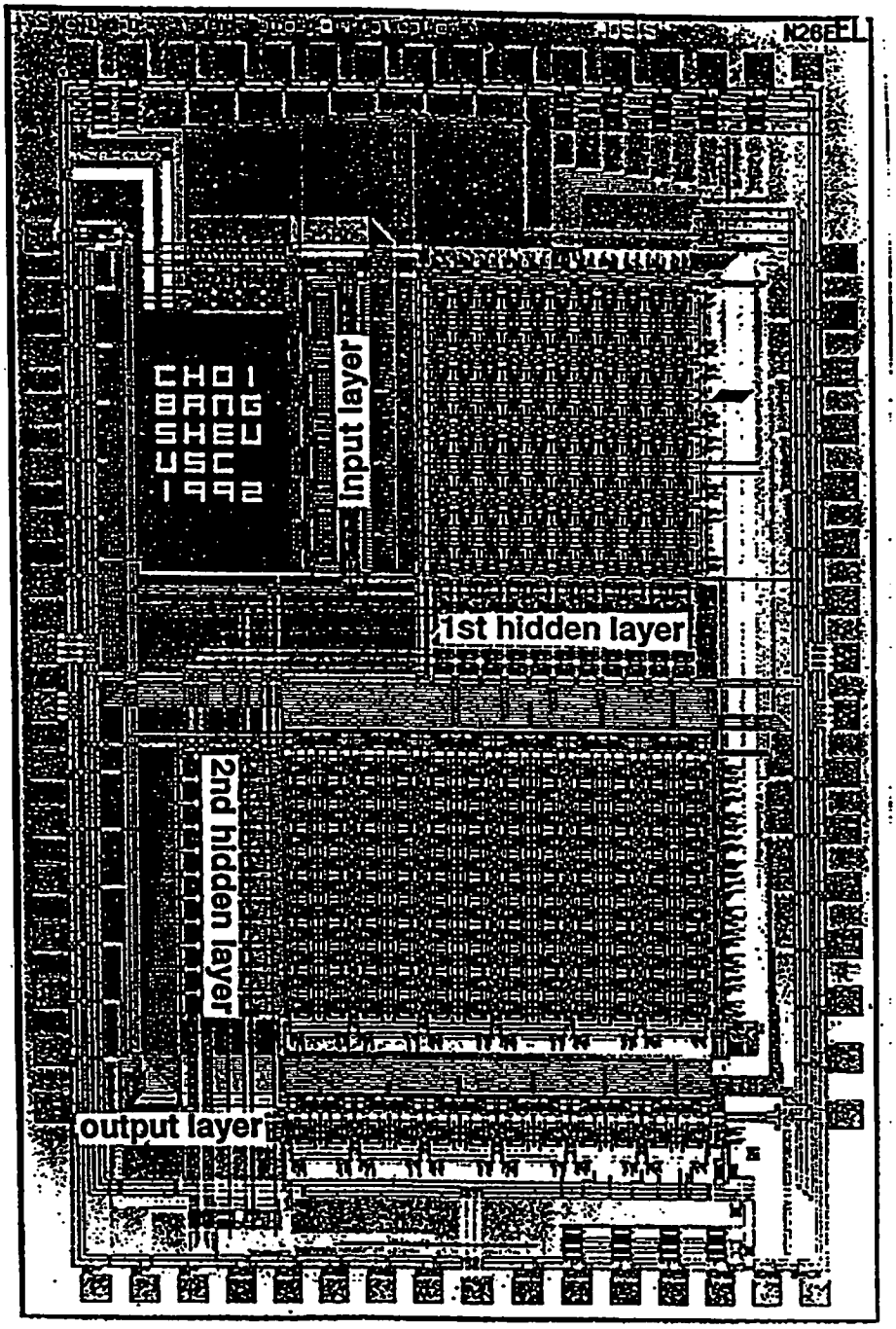
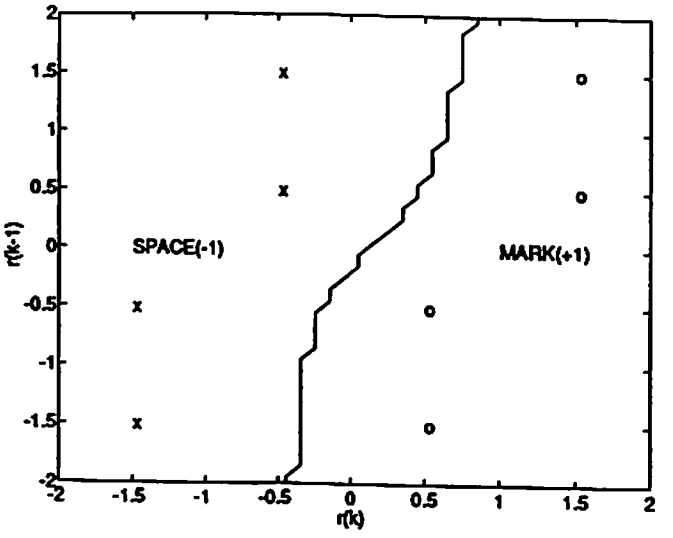
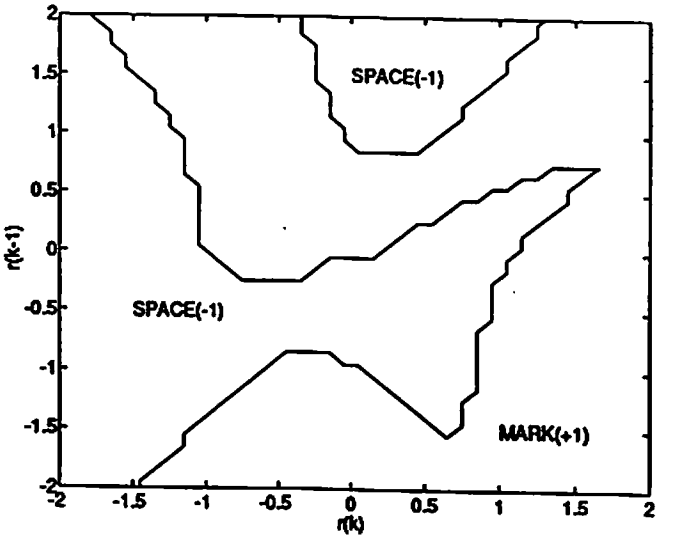


Figure 5.7: Die photo of the 4-layered neural network processor chip.

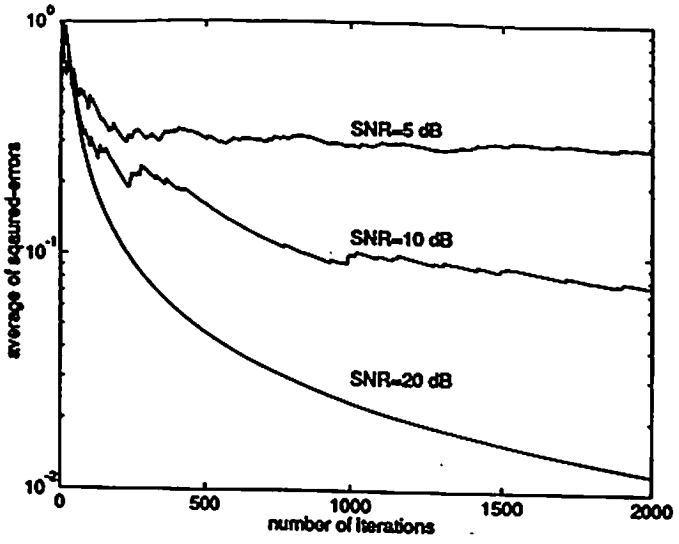


(a)

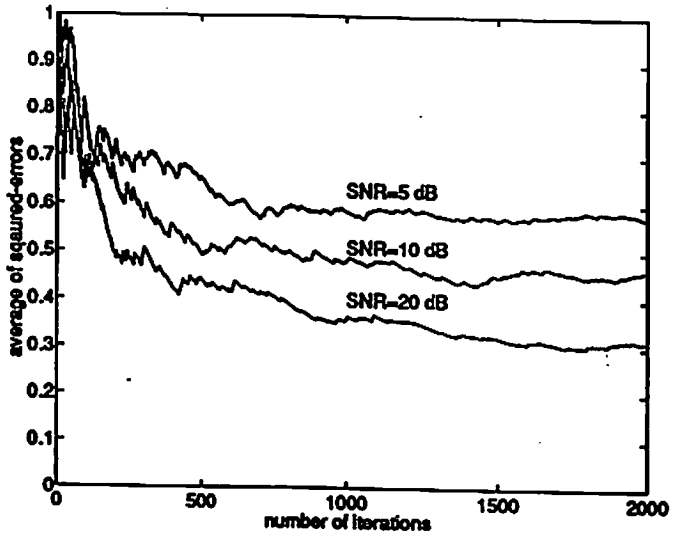


(b)

Figure 5.8: Simulated results on the decision boundary of the neural-based receiver. (a) For the minimum-phase channel. The left portion and right portion are the decision regions for the -1 and +1 symbols, respectively. (b) For the nonminimum-phase channel.



(a)



(b)

Figure 5.9: Simulated results on the convergence rate of the neural-based receiver with different SNR's. (a) For the channel of $H(z) = 0.89443 + 0.44721z^{-1}$. (b) For the channel of $H(z) = 0.407 + 0.815z^{-1} + 0.407z^{-2}$.

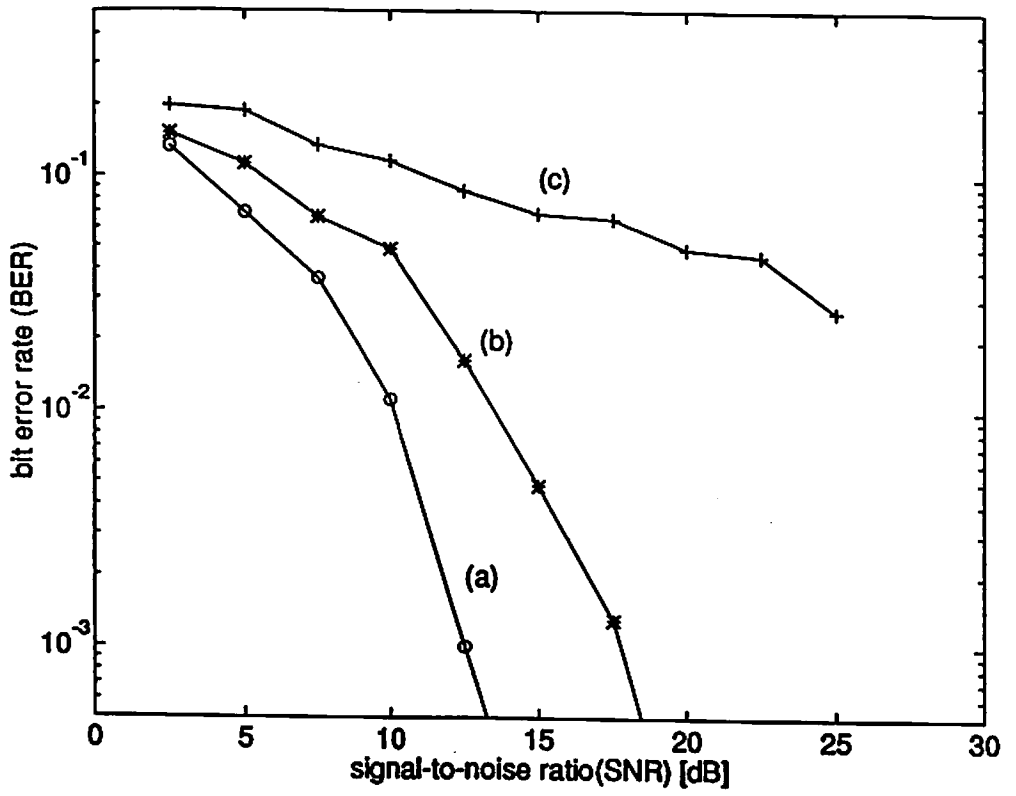


Figure 5.10: Bit error rate (BER) performance. (a) For the channel of $H(z) = 0.89443 + 0.44721z^{-1}$. (b) For the channel of $H(z) = 0.44721 + 0.89443z^{-1}$. (c) For the channel of $H(z) = 0.407 + 0.815z^{-1} + 0.407z^{-2}$.

Chapter 6

Conclusions and Further Works

In this dissertation research, many analog-digital neural computing chips have been designed and fabricated for the scientific and engineering applications such as signal processing and communication. Several versions of the prototype chips and the complete chips were fabricated in a 2.0-μm double-polysilicon, double-metal CMOS technology through the MOSIS Service.

The building blocks were designed for the purpose of optimum use in a large network construction. The programmable synapse circuit uses the wide-range modified Gilbert multiplier with the dynamic capacitors which are inherent in the CMOS technology. The output neuron consists of the transimpedance amplifier and the sigmoid function generation. Several different kinds of operations can be achieved by reconfiguration signals in addition to the modifiable gain. The input neuron is the unity-gain buffer which is obtained from the conventional operational amplifier. All building blocks and the subsystems with a small dimension of the network were successfully measured in the laboratory.

The self-organizing neural network was implemented in an efficient hardware. The multipliers are used for representing the distances between the applied input and the stored data. The high-precision winner-take-all (WTA) circuit, which

is the key element, was designed and fabricated. Several performance-improving techniques such as cascading, distributed biasing, current steering are employed. Successful experimental results support the possible extension of the network size up to at least 1,000 competition cells.

The general-purpose analog neural network is customized to the application in digital communication. The microchip includes a four-layered network and the switched-capacitor (SC) analog delay line. Selected measured results on the components and the entire chips were presented. The system-level simulations have been performed on the minimum-phase channel and the nonminimum-phase channel.

For future works, the microchips incorporating the complete system can be explored. In order to investigate the neural network chips for the system-level applications, the supporting measurement scheme as well as an improved version of the microchips, should be implemented. It can consist of the designed chip, the customized board for signal interference, and the host computer processing the learning algorithms. All basic operational behaviors such as checking the input-output waveform and programming the synapses throughout the chip, which were already done, will be controlled and displayed by the host-computer control in the user-interface mode. Finally, the analog-digital neuroprocessor systems can be applied to the study of models from biological systems toward the better lives for the human beings.

Reference List

[1] D. C. Nagel, "Platforms for multimedia: the impact on consumer electronics," *Tech. Digest IEEE Inter. Solid-State Circuits Conf.*, pp. 20–23, San Francisco, CA, Feb. 1993.

[2] J. Clark and A. Yuille, *Data Fusion for Snesory Information Processing Systems.* Kluwer Academic Publishers: Boston, MA, 1990.

[3] K. B. Benson and D. G. Fink, *HDTV: Advanced Television for the 1990s.* McGraw-Hill Publishing Company, Inc.: New York, NY, 1991.

[4] National Science Foundation, Washington D.C., *Grand challenges: High performance computing and communications, The FY 1992 U.S. Research and Development Program*, 1991.

[5] H. Komiya, "Future technological and economic prospects for VLSI," *Tech. Digest IEEE Inter. Solid-State Circuits Conf.*, pp. 16–19, San Francisco, CA, Feb. 1993.

[6] K. Itoh, "Trends in megabit DRAM circuit design," *IEEE Jour. Solid-State Circuits*, vol. 25, pp. 778–789, June 1990.

[7] Neurocomputers, vol. 2, *DELTA/SIGMA/ANSim*, 1988.

[8] R. Hecht-Nielsen, "Neural-computing: picking the human brain," *IEEE Spectrum*, vol. 25, pp. 36–41, Mar. 1988.

[9] P. K. Simpson, "Foundations of neural networks," in *Artificial Neural Networks: Paradigms, Applications, and Hardware Implementations* (E. Sánchez-Sinencio and C. Lau, eds.), IEEE Press: Piscataway, NJ, 1992.

[10] R. Lippmann, "An introduction to computing with neural nets," *IEEE Acoustic, Speech, and Signal Processing Magazine*, pp. 4–22, Apr. 1987.

[11] D. E. Rumelhart, J. L. McClelland, and the PDP Research Group, *Parallel Distributed Processing*, vol. 1. The MIT Press: Cambridge, MA, 1989.

[12] B. Kosko, *Neural Networks and Fuzzy Systems.* Prentice-Hall, Englewood Cliffs, NJ, 1992.

[13] D. R. Collins and P. A. Penz, "Considerations for neural network hardware implementations," *Proc. IEEE Inter. Symp. Circuits and Systems*, pp. 834–837, Portland, OR, May 1989.

[14] M. Griffin, G. Tahara, K. Knorpp, R. Pinkham, B. Riley, D. Hammerstrom, and E. Means, "An 11 million transistor digital neural network execution engine," *Tech. Digest IEEE Inter. Solid-State Circuits Conf.*, pp. 180–181, San Francisco, CA, Feb. 1991.

[15] R. Mason, W. Robertson, and D. Pincock, "An hierarchical VLSI neural network architecture," *IEEE Jour. Solid-State Circuits*, vol. 27, pp. 106–108, Jan. 1992.

[16] M. S. Melton, T. Phan, D. S. Reeve, and D. E. Van den Bout, "The TInMANN VLSI chip," *IEEE Trans. Neural Networks*, vol. 3, pp. 375–384, May 1992.

[17] N. Mauduit, M. Duranton, J. Gobert, and J.-A. Sirat, "Lneuro 1.0: A piece of hardware LEGO for building neural network systems," *IEEE Trans. Neural Networks*, vol. 3, pp. 414–422, May 1992.

[18] C.-F. Chang and B. J. Sheu, "Digital VLSI multiprocessor design for neurocomputers," *Proc. IEEE/INNS Inter. Joint Conf. Neural Networks*, vol. II, pp. 1–6, Baltimore, MD, June 1992.

[19] K. Uchimura, O. Saito, and Y. Amemiya, "A high-speed digital neural network chip with low-power chain-reduction architecture," *IEEE Jour. Solid-State Circuits*, vol. 27, pp. 1862–1867, Dec. 1992.

[20] M. Yasunaga, N. Masuda, M. Yagyu, M. Asai, K. Shibata, M. Ooyama, M. Yamada, T. Sakaguchi, and M. Hashimoto, "A self-learning digital neural network using wafer-scale LSI," *IEEE Jour. Solid-State Circuits*, vol. 28, pp. 106–114, Feb. 1993.

[21] T. Watanabe, K. Kimura, M. Aoki, T. Sakata, and K. Itho, "A single 1.5-V digital chip for a 10^6-synapse neural network," *IEEE Trans. Neural Networks*, vol. 4, pp. 387–393, May 1993.

[22] J. E. Tomberg and K. K. K. Kaski, "Pulse-density modulation technique in VLSI implementations of neural network algorithms," *IEEE Jour. Solid-State Circuits*, vol. 26, pp. 1277–1286, Oct. 1990.

[23] G. Erten and R. M. Goodman, "A digital neural network architecture using random pulse trains," *Proc. IEEE/INNS Inter. Joint Conf. Neural Networks*, vol. I, pp. 190–105, Baltimore, MD, June 1992.

[24] J. A. Dickson, R. D. Mcleod, and H. C. Card, "Stochastic arithmetic implementations of neural networks with in situ learing," *Proc. IEEE/INNS Inter. Joint Conf. Neural Networks*, vol. II, pp. 711–716, San Francisco, CA, Mar. 1993.

[25] A. F. Murray, "Pulse arithmetic in VLSI neural networks," *IEEE Micro Magazine*, pp. 64–74, Dec. 1989.

[26] A. Hamilton, A. F. Murray, D. J. Baxter, S. Churcher, H. M. Reekie, and L. Tarassenko, "Integrated pulse stream neural networks: results, issues, and pointer," *IEEE Trans. Neural Networks*, vol. 3, pp. 385–393, May 1992.

[27] G. Moon, M. E. Zaghloul, and R. W. Newcomb, "VLSI implementation of synaptic weighting and summing in pulse coded neural-type cells," *IEEE Trans. Neural Networks*, vol. 3, pp. 394–403, May 1992.

[28] J. Donald and L. Akers, "An adaptive neural processing node," *IEEE Trans. Neural Networks*, vol. 4, pp. 413–426, May 1993.

[29] D. R. Hush and B. F. Horne, "Progress in supervised neural networks," *IEEE Signal Processing Magazine*, pp. 8–39, Jan. 1993.

[30] D. Hammerstrom, "A VLSI architecture for high-performance, low-cost, on-chip learning," *Proc. IEEE/INNS Inter. Joint Conf. Neural Networks*, vol. 2, pp. 537–543, 1990.

[31] N. Weste and K. Eshraghian, *Principles of CMOS VLSI Design: A Systems Perspective*. Addison-Wesley Publishing Company: Reading, MA, 1985.

[32] R. T. Witek, "A 200-MHz 64-n dual-issue CMOS microprocessor," *IEEE Jour. Solid-State Circuits*, vol. 27, pp. 1555–1567, Nov. 1992.

[33] M. Toyokura, K. Okamoto, H. Kodama, A. Ohtani, T. Araki, and K. Aono, "A video digital signal processor with a vector-pipeline architecture," *Tech. Digest IEEE Inter. Solid-State Circuits Conf.*, pp. San Francisco, CA, Feb., 1992.

[34] S. Satyanarayana, Y. P. Tsividis, and H. P. Graf, "A reconfigurable VLSI neural network," *IEEE Jour. Solid-State Circuits*, vol. 27, pp. 67–81, Jan. 1992.

[35] B. Lont and W. Guggenbühl, "Analog CMOS implementation of a multilayer perceptron with nonlinear synapses," *IEEE Trans. Neural Networks*, vol. 3, pp. 457–465, May 1992.

[36] F. J. Kub, K. K. Moon, J. A. Mack, and F. M. Long, "Programmable analog vector-matrix multipliers," *IEEE Jour. Solid-State Circuits*, vol. 25, pp. 207–214, Feb. 1990.

[37] B. J. Sheu, J. Choi, and C.-F. Chang, "An analog neural network processor for self-organizing mapping," *Tech. Digest IEEE Inter. Solid-State Circuits Conf.*, pp. 136–137, San Francisco, CA, Feb. 1992.

[38] N. I. Khachab and M. Ismail, "A nonlinear CMOS analog cell for VLSI signal and information processing," *IEEE Jour. Solid-State Circuits*, vol. 26, pp. 1689–1699, Nov. 1991.

[39] B. Hochet, V. Peiris, S. Abdo, and M. J. Declercq, "Implementation of a learning Kohonen neuron based on a new multilevel storage scheme," *IEEE Jour. Solid-State Circuits*, vol. 26, pp. 262–267, Mar. 1991.

[40] Y. Arima, M. Murasaki, T. Yamada, A. Maeda, and H. Shinohara, "A refreshable analog VLSI neural network chip with 400 neurons and 40k synapses," *Tech. Digest IEEE Inter. Solid-State Circuits Conf.*, pp. 132–133, San Francisco, CA, Feb. 1992.

[41] T. Morishita, Y. Tamura, and T. Otsuki, "A BiCMOS analog neural network with dynamically updated weights," *Tech. Digest IEEE Inter. Solid-State Circuits Conf.*, pp. 142–143, San Francisco, CA, Feb. 1990.

[42] J. Choi, S. H. Bang, and B. J. Sheu, "A programmable analog VLSI neural network processor for communication receiver," *IEEE Trans. Neural Networks*, vol. 4, pp. 484–495, May 1993.

[43] B. W. Lee and B. J. Sheu, *Hardware Annealing in Analog VLSI Neurocomputing*. Kluwer Academic Publishers: Boston, MA, 1991.

[44] H. P. Graf and D. Henderson, "A reconfigurable CMOS neural network," *Tech. Digest IEEE Inter. Solid-State Circuits Conf.*, pp. 144–145, San Francisco, CA, Feb. 1990.

[45] M. Verleysen, B. Sirletti, A. M. Vandemeulebroecke, and P. G. A. Jespers, "Neural networks for high-storage content-addressable memory: VLSI circuit and learning algorithm," *IEEE Jour. Solid-State Circuits*, vol. 24, pp. 562–569, June 1989.

[46] J. I. Raffel, "Electronic implementation of neuromorphic systems," *Proc. IEEE Custom Integrated Circuits Conf.*, pp. 10.1.1–10.1.7, Rochester, NY, May 1988.

[47] J. Van der Spiegel, P. Mueller, D. Blackman, P. Chance, C. Donham, R. Etienne-Cummings, and P. Kinget, "An analog neural computer with modular architecture for real-time dynamic computations," *IEEE Jour. Solid-State Circuits*, vol. 27, pp. 82–91, Jan. 1992.

[48] P. W. Hollis and J. J. Paulos, "Artificial neural networks using MOS analog multiplier," *IEEE Jour. Solid-State Circuits*, vol. 25, pp. 849–855, June 1990.

[49] A. Jayakumar and J. Alspector, "A cascadable neural network chip set with on-chip learning using noise and gain annealing," *Proc. IEEE Custom Integrated Circuits Conf.*, pp. 19.5.1–19.5.4, Boston, MA, May 1992.

[50] D. A. Durfee and F. S. Shoucair, "Comparison of floating gate neural network memory cells in standard VLSI CMOS technology," *IEEE Trans. Neural Networks*, vol. 3, pp. 347–353, May 1992.

[51] C.-K. Sin, A. Kramer, V. Hu, R. R. Chu, and P. K. Ko, "EEPROM as an analog storage device, with particular applications in neural networks," *IEEE Trans. Electro Devices*, vol. 39, pp. 1410–1419, June 1992.

[52] T. Blyth, S. Khan, and R. Simko, "A non-volatile analog storage device using EEPROM technology," *Tech. Digest IEEE Inter. Solid-State Circuits Conf.*, pp. 192–193, San Francisco, CA, Feb. 1991.

[53] B. W. Lee, H. Yang, and B. J. Sheu, "Analog floating-gate synapses for general-purpose VLSI neural computation," *IEEE Trans. Circuits and Systems*, vol. 38, pp. 654–658, June 1991.

[54] M. Holler, S. Tam, H. Castro, and R. Benson, "An electrically trainable artificial neural network (ETANN) with 10240 floating gate synapses," *Proc. IEEE/INNS Inter. Joint Conf. Neural Networks*, vol. 2, pp. 191–196, 1989.

[55] T. H. Borgstrom, M. Ismail, and S. B. Bibyk, "Programmable current-mode neural network for implementation in analogue MOS VLSI," *IEE Proceeding, Part G*, vol. 37, pp. 175–184, Apr. 1990.

[56] H. A. Castro, S. M. Tam, and M. A. Holler, "Implementation and performance of an analog nonvolatile neural network," *Jour. Analog Integrated Circuits and Signal Processing*, vol. 4, pp. 97–113, Sep. 1993.

[57] C. A. Mead, *Analog VLSI and Neural Systems.* Addison-Wesley Publishing Company: Reading, MA, 1989.

[58] C. Mead, "Adaptive retina," in *Analog VLSI Implementations of Neural Systems* (C. Mead and M. Ismail, eds.), pp. 239–246, Norwell, MA: Kluwer Academic Publishers: Boston, MA, 1989.

[59] K. Boahen and A. Andreou, "A contrast sensitive silicon retina with reciprocal synapses," in *Advances in Neural Information Processing Systems, Vol. 4* (J. E. Moody, S. J. Hanson, and R. P. Lippmann, eds.), pp. 764–774, San Mateo, CA: Morgan Kaufmann, 1990.

[60] T. Delbrück, "Silicon retina with correlation-based, velocity-tuned pixels," *IEEE Trans. Neural Networks*, vol. 4, pp. 529–541, May 1993.

[61] W. Bair and C. Koch, "An analog VLSI chip for finding edges from zero-crossing," in *Neural Information Processing Systems, vol. 2* (R. P. Lippmann, J. E. Moody, and D. S. Touretzky, eds.), pp. 399–405, Palo Alto, CA: Morgan Kaufmann, 1991.

[62] J. Tanner and C. A. Mead, "An integrated analog optical motion sensor," in *VLSI Signal Processing II* (S.-Y. Kung, R. E. Owen, and J. G. Nash, eds.), pp. 59–76, New York, NY: IEEE Press, 1987.

[63] D. L. Standley, "An object position and orientation IC with embedded images," *IEEE Jour. Solid-State Circuits*, vol. 26, pp. 1853–1859, Dec. 1991.

[64] J. Luo, C. Koch, and B. Mathur, "Figure-ground segregation using an analog VLSI chip," *IEEE Micro*, vol. 12, pp. 46–57, Dec. 1992.

[65] R. F. Lyon and C. A. Mead, "An analog electronic cochlea," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 36, pp. 1119–1134, July 1988.

[66] C. A. Mead, X. Arreguit, and J. P. Lazzaro, "Analog VLSI models of binaural hearing," *IEEE Trans. Neural Networks*, vol. 2, pp. 230–236, Mar. 1991.

[67] J. Lazzaro, J. Wawrzynek, M. Mahowald, M. Sivilotti, and D. Gillespie, "Silicon auditory processors as computer peripherals," *IEEE Trans. Neural Networks*, vol. 4, pp. 523–528, May 1993.

[68] A. G. Andreou, "Electronic receptors for tactile/haptic sensing," in *Advances in Neural Information Processing Systems, Vol. 1* (D. S. Touretzky, ed.), pp. 785–793, San Mateo, CA: Morgan Kaufmann, 1989.

[69] H. Kobayashi, J. L. White, and A. A. Abidi, "An active resistor network for Gaussian filtering of images," *IEEE Jour. Solid-State Circuits*, vol. 26, pp. 738–748, May 1991.

[70] A. Moore, J. Allmand, and R. M. Goodman, "A real-time neural system for color constancy," *IEEE Trans. Neural Networks*, vol. 2, pp. 237–247, Mar. 1991.

[71] W.-C. Fang, B. J. Sheu, O. T.-C. Chen, and J. Choi, "A VLSI neural processor for image data compression using self-organization neural networks," *IEEE Trans. Neural Networks*, vol. 3, pp. 506–518, May 1992.

[72] J.-C. Lee, B. J. Sheu, J. Choi, and R. Chellappa, "A mixed-signal VLSI neuroprocessor for image restoration," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 2, pp. 319–324, Sep. 1992.

[73] J.-C. Lee, B. J. Sheu, W.-C. Fang, and R. Chellappa, "VLSI neuroprocessors for video motion detection," *IEEE Trans. Neural Networks*, vol. 4, pp. 178–191, Mar. 1993.

[74] J. S. Dener, "Neural network recognizer for hand-written zip code digits," in *Neural Information Processing Systems, vol. 2* (D. S. Touretzky, ed.), pp. 323–331, Palo Alto, CA: Morgan Kaufmann, 1989.

[75] E. Säckinger, B. E. Boser, J. Bromley, Y. LeCun, and L. D. Jakel, "Application of the anna network chip to high-speed character recognition," *IEEE Trans. Neural Networks*, vol. 3, pp. 498–505, May 1992.

[76] F. Lisa, J. C. anf C. Pérez-Vicente, N. Avellana, and E. Valderrama, "Two-bit weights are enough to solve vehicle license recognition problem," *Proc. IEEE Inter. Joint Conf. Neural Networks*, pp. 1242–1246, San Francisco, CA, Apr. 1993.

[77] W. Liu, A. G. Andreou, and M. H. Goldstein, "Voice-speech representation by an analog silicon model of the auditory periphery," *IEEE Trans. Neural Networks*, vol. 3, pp. 477–487, May 1992.

[78] Y. Horio, S. Nakamura, J. Miyasaka, and H. Tasaka, "Speech recognition network with SC neuron-like components," *Proc. IEEE Inter. Symp. Circuits and Systems*, pp. 495–498, 1988.

[79] S. P. DeWeerth, L. Nielsen, C. A. Mead, and K. J. Åanstrõm, "A simple neuron servo," *IEEE Trans. Neural Networks*, vol. 2, pp. 248–2511, Mar. 1991.

[80] G. Avitabile, M. Forti, S. Manetti, and M. Marini, "A new nonsymmetrical neural network with applications to signal processig," *IEEE Trans. Circuits and Systems*, vol. 38, pp. 202–209, Feb. 1991.

[81] J. Brauch, S. M. Tam, M. A. Holler, and A. L. Shmurun, "Analog VLSI neural network for impact signal processing," *IEEE Micro*, pp. 34–45, Dec. 1992.

[82] J. Meador, "Pulse-coded communication in VLSI neural networks," *Neural Network*, vol. 3, no. 4, pp. 147–148, 1989.

[83] Y. Arima, K. Mashito, K. Okada, T. Yamada, A. Maeda, H. Notani, H. Kondou, and S. Kayano, "A 336-neuron 28k-synapse, self-learning neural network chip with branch-neuron-unit architecture," *IEEE Jour. Solid-State Circuits*, vol. 27, pp. 1637–1644, Nov. 1991.

[84] U. Cilingiroglŭ, "A pulrely capacitive synaptic matrix for fixed-weight neural networks," *IEEE Trans. Circuits and Systems*, vol. 38, pp. 210–217, Feb. 1991.

[85] D. B. Scharwtz, R. E. Howard, and W. E. Hubbard, "A programmable analog neural network chip," *IEEE Jour. Solid-State Circuits*, vol. 24, pp. 313–319, Apr. 1989.

[86] T. Shibata and T. Ohmi, "A functional MOS transistor feuturing gate-level weighted sum and threshold operations," *IEEE Trans. Electron Devices*, vol. 39, pp. 1444–1455, June 1992.

[87] B. J. Maunday and E. I. El-Masry, "Feedforward associative memory switched-capacitor artificial neural networks," *Jour. Analog Integrated Circuits and Signal Processing*, vol. 1, pp. 321–338, Dec. 1991.

[88] A. Cichocki and R. Unbehauen, "Switched-capapcitor artificial neural networks for non-linear optimization with constraints," *Proc. IEEE Inter. Symp. Circuits and Systems*, vol. 3, pp. 2809–2812, 1990.

[89] J. E. Hansen, J. K. Skelton, and D. J. Allstot, "A time-multiplexed switched-capacitor circuit for neural network applications," *Proc. IEEE Inter. Symp. Circuits and Systems*, vol. 3, pp. 2177–2180, 1989.

[90] A. Rodríquez-Vázquez, R. Domínguez-Castro, A. Rueda, J. L. Huerstas, and E. Sánchez-Sinencio, "Nonlinear switched-capacitor neural networks for optimization problem," *IEEE Trans. Circuits and Systems*, vol. 37, pp. 384–389, Mar. 1990.

[91] A. M. Chiang and M. L. Chuang, "A CCD programmable image processor and its neural network applications," *IEEE Jour. Solid-State Circuits*, vol. 26, no. 12, pp. 1894–1901, 1991.

[92] A. J. Agranat, C. F. Neugebauer, R. D. Nelson, and Y. Ariv, "The CCD processor: a neural network integrated circuit with 65536 programmable analog synapses," *IEEE Trans. Circuits and Systems*, vol. 37, pp. 1073–1075, Aug. 1990.

[93] E. K. F. Lee and P. G. Gulak, "A CMOS field-programmable analog array," *IEEE Jour. Solid-State Circuits*, vol. 26, pp. 1860–1867, Dec. 1991.

[94] P. W. Hollis and J. J. Paulos, "An analog BiCMOS Hopfield neuron," *Jour. Analog Integrated Circuits and Signal Processing*, vol. 2, pp. 273–280, Nov. 1992.

[95] B. W. Lee and B. J. Sheu, "A compact and general-purpose neural chip with electrically programmable synapses," *Proc. IEEE Custom Integrated Circuits Conf.*, pp. 26.6.1–26.6.4, Boston, Ma, May 1990.

[96] W. A. Fisher, R. J. Fujimoto, and R. C. Smithson, "A programmable analog neural network processor," *IEEE Trans. Neural Networks*, vol. 2, pp. 222–229, Mar. 1991.

[97] B. Boser, E. Säckinger, J. Bromley, Y. LeCun, and L. D. Jackel, "An analog neural network processor with programmable topology," *IEEE Jour. Solid-State Circuits*, vol. 26, pp. 2017–2025, Dec. 1991.

[98] T. Shima, T. Kimura, Y. Kamatani, T. Itakura, Y. Fujita, and T. Iida, "Neuro chips with on-chip back-propagation and/or Hebbian learning," *IEEE Jour. Solid-State Circuits*, vol. 27, pp. 1868–1876, Dec. 1992.

[99] J. Choi and B. J. Sheu, "VLSI design of compact and high-precision analog neural network processors," *Proc. IEEE/INNS Inter. Joint Conf. Neural Networks*, vol. 2, pp. 637–641, Balitmore, MD, July 1992.

[100] B. Johnson and T. Quarles and A. R .Newton and D. O. Pederson and A. Sagiovanni-Vincentelli, Department of Electrical Engineering and Computer Science, University of California, Berkeley, *SPICE3 Version 3E1 Users Guide*, Apr. 1991.

[101] B. J. Sheu and C. Hu, "Switched-induced error voltage on a switched capacitor," *IEEE Jour. Solid-State Circuits*, vol. 19, pp. 519–525, Aug. 1984.

[102] C. Eichenberger and W. Guggenbühl, "Dummy transistor compensation of analog MOS switches," *IEEE Jour. Solid-State Circuits*, vol. 24, pp. 1143–1146, Aug. 1989.

[103] R. Gregorian and G. C. Temes, *Analog MOS Integrated Circuits for Signal Processing*. John Wiley and Sons: New York, NY, 1986.

[104] M. Banu and Y. Tsividis, "Floating voltage-controlled resistors in CMOS technology," *Electronic Lett.*, vol. 18, pp. 678–679, 1982.

[105] C. Tomovich, "MOSIS-a gate way to silicon," *IEEE Circuit and Device Mag.*, vol. 4, pp. 22–23, Mar. 1988.

[106] G. Lewicki, "Foresight: a fast turn-around and low cost ASIC prototyping alternative," *Proc. IEEE ASIC Seminar and Exhibit*, pp. p.6–8.1/8.2, Rochester, NY, Sep. 1990.

[107] M. E. Robinson, H. Yoneda, and E. Sánchez-Sinencio, "A modular CMOS design of a Hamming network," *IEEE Trans. Neural Networks*, vol. 3, pp. 444–456, May 1992.

[108] Y. He, U. Cilingirogŭlu, and E. Sáchez-Sinencio, "A high density and low power charge-based Hamming network," *IEEE Trans. VLSI Systems*, vol. 1, pp. 56–62, 1993.

[109] J. Mann and S. Gilbert, "An analog self-organizing neural network," in *Neural Information Processing Systems, vol. 1* (D. S. Touretzky, ed.), pp. 739–747, Palo Alto, CA: Morgan Kaufmann, 1989.

[110] D. Macq, M. Verleysen, P. Jespers, and J. Legat, "Analog implementation of a Kohonen map with on-chip learning," *IEEE Trans. Neural Networks*, vol. 4, pp. 456–461, May 1993.

[111] Y. He and U. Cilingirogŭlu, "A charged-based on-chip adaptation Kohonen neural network," *IEEE Trans. Neural Networks*, vol. 4, pp. 462–469, May 1993.

[112] J. Mann, R. Lippmann, B. Berger, and J. Raffel, "A self-organizing neural net chip," *Proc. IEEE Custom Integrated Circuits Conf.*, pp. 10.3.1–10.3.5, Rochester, NY, May 1988.

[113] M. A. Mahowald and T. Delbrück, "Cooperative stero matching using static and dynamic image features," in *Analog VLSI Implementation of Neural Systems* (C. Mead and M. Ismail, eds.), pp. 213–238, Kluwer Academic Publishers: Boston, MA, 1989.

[114] A. G. ANdreou, K. A. Boahen, P. O. Pouliquen, A. Pavasovíc, R. E. Jenkins, and K. Strohbehn, "Current-mode subthreshold MOS circuits for analog VLSI systems," *IEEE Trans. Neural Networks*, vol. 2, pp. 205–213, Mar. 1991.

[115] T. Yamashita, T. Shibata, and T. Ohmi, "Neuron MOS winner-take-all circuit and its application to associative memory," *Tech. Digest IEEE Inter. Solid-State Circuits Conf.*, pp. 236–237, San Francisco, CA, Feb. 1993.

[116] X. Fang and J. Starzyk, "A novel winner-take-all circuit," *Proc. World Congress on Neural Networks*, vol. 4, pp. 689–692, Portland, OR, July 1993.

[117] J. Choi and B. J. Sheu, "A high-precision VLSI winner-take-all circuit for self-organizing neural networks," *IEEE Jour. Solid-State Circuits*, vol. 28, pp. 576–584, May 1993.

[118] J. G. Proakis, *Digital Communications.* McGraw-Hill Publishing Company, Inc.: New York, NY, 1983.

[119] S. H. Bang and B. J. Sheu, "A neural-based digital communication receiver for inter-symbol interference and white Gaussian noise channel," *Proc. IEEE Inter. Symp. Circuits and Systems*, vol. 6, pp. 2933–2936, San Diego, CA, May 1992.

[120] S. Singhal and L. Wu, "Training multi-layer perceptrons with the extended kalman algorithm," in *Advances in Neural Information Processing Systems, vol. 1* (D. S. Touretzky, ed.), New York, NY: Morgan Kaufmann, 1990.

[121] Y. S. Lee and K. W. Martin, "A switched-capacitor realization of multiple FIR filters on a single chip," *IEEE Jour. Solid-State Circuits*, vol. 23, pp. 536–542, Apr. 1988.

[122] S. C. Fang, Y. Tsividis, and O. Wing, "SWITCAP: a switched capacitor network analysis program," *IEEE Circuits and Systems Mag.*, vol. 5, pp. 4–10, and 41–46, 1983.

[123] Motorola Inc., *DSP56000ADS Application Development System User's Manual and DSP56ADC16 Evaluation Board User's Manual*, 1991.

[124] Texas Instruments, vol. 2, *Data Book: Linear Circuits - Data Acquisition and Conversion*, 1989.

[125] G. J. Gibson, S. Siu, S. Chen, C. F. N. Cowan, and P. M. Grant, "The application of nonlinear architectures to adaptive channel equalization," *IEEE Proc. Inter. Conf. Communications*, pp. 649–653, 1990.

Appendix A

Gaussian Synapse Circuit

The conventional error back-propagation network usually requires a quite long convergence time for correct weight adjustment. The sigmoid function of a conventional multi-layer network gives a smooth response over a wide range of input values. In contrast, the Gaussian function responds significantly only to local regions of the space of input values. Backward propagation training is more efficient in neural networks based on Gaussian functions, Radial Basis Function (RBF) networks, than those based on sigmoid functions in the hidden layers. Up to two or three orders of magnitude speed-up in training has been reported in applications for pattern recognition such as phoneme classification by using Gaussian function neural networks [A1, A2].

In this chapter, the design of compact analog VLSI circuits for Gaussian function neural networks is presented. The proposed circuit is biased in the strong-inversion region and optimized for several design issues such as high precision, high operation speed, and area compactness to make it suitable for scalable neural network implementations.

A.1 Gaussian Function Networks

Figure A.1 shows the portion of a complete Gaussian function neural network. Input neurons are fully connected to output neurons through the synapse matrix. Input neurons can belong to the input layer or a hidden layer of the complete neural networks, while the output neurons can belong to another hidden layer or the output layer of the complete network. The resultant operation of input neurons, synapses, and output neurons can be expressed as

$$Y_j = \sum_{i=1}^{M} A_{ji} e^{-\frac{(X_i - W_{ji})^2}{2\sigma_{ji}^2}}, \quad for\ j = 1 \cdots N, \tag{A.1}$$

where W_{ji} is a weight value of the synapse cell which connects the i^{th} input neuron and the j^{th} output neuron. The input neurons layer has M neurons and the output neurons layer has N neurons. Each synapse between an input neuron and an output neuron can perform the Gaussian function with the mean value being a weight value. Changing the mean value W_{ji} means to increase or decrease connection strength of a input neuron X_i to an output neuron Y_j [A3]. The σ_{ji} determines the standard deviation value of the Gaussian function characteristic.

In conventional back-propagation networks, linear multiplications between the input and synapse values, $W_{ji} \cdot X_i$, are used and easily implemented by the analog multipliers. In self-organization networks, squared-difference functions between the input and the synapse values, $(X_i - W_{ji})^2$, can be implemented by the differential-input Gilbert multipliers in the analog neuroprocessor design [A4]. On the other hand, in a Gaussian function network, each synapse needs to compute the exponential nonlinearity, $e^{-\frac{(X_i - W_{ji})^2}{2\sigma_{ji}^2}}$. The exponential nonlinearity was computed in simulations on digital computers of the conventional approach. One

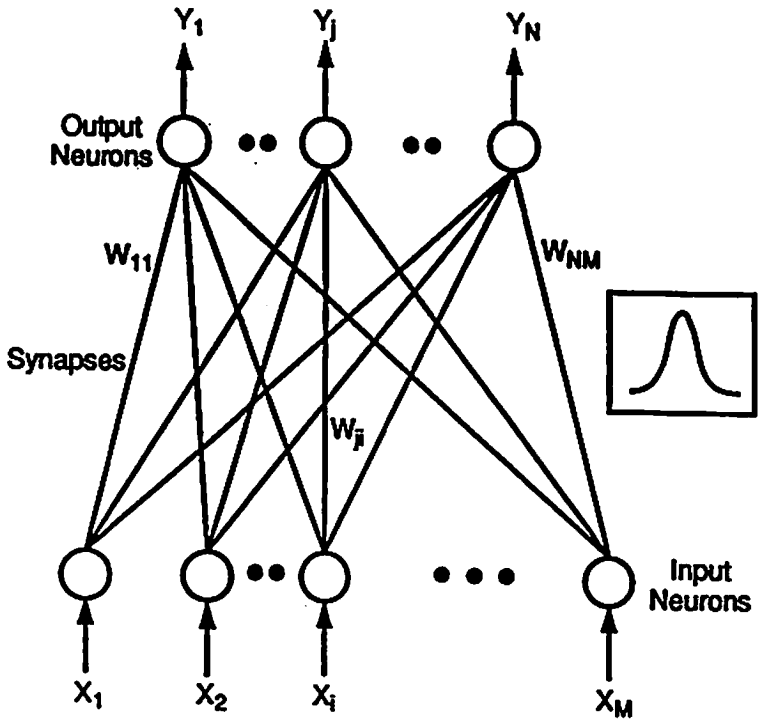


Figure A.1: A portion of a complete neural network with Gaussian synapse characteristics.

hardware implementation of analog Gaussian function computing with transistors biased in the subthreshold region was reported [A5]. In the subthreshold region, the drain current of an MOS transistor has an exponential dependence on the gate bias so that the exponential nonlinearity can be easily achieved. The subthreshold-region VLSI circuits are suitable for implementation of biologically-inspired artificial neural systems [A6]. Millions of MOS transistors can be integrated on a single silicon chip because of extremely low power consumption in each transistor. In the strong-inversion region, however, MOS transistors have a power-law dependence of the drain current on the gate voltage. Since the strong-inversion operation of MOS devices provides features of high current driving, large dynamic range, and high noise immunity, high-speed analog VLSI neural network processors can be built with MOS transistors biased in the strong-inversion region for engineering applications.

A.2 Circuit Analysis

Figure A.2(a) shows the circuit schematic diagram and transistor sizes of a basic synapse cell with single-ended input data. The Gaussian function synapse cell consists of the MOS differential pair and several arithmetic computational units operating in the current-mode configuration. The power-law of a drain current on the gate voltage in the MOS transistor biased in the strong-inversion region makes an implementation of the Gaussian function to be a piecewise approximation. Transistors with non-minimum channel lengths are used to avoid the channel-length modulation effect. The input voltage is applied to the gate terminal of one transistor in the differential pair and the weight value is stored on the total gate

capacitance of the other transistor. The two currents in the differential pair can be expressed as [A7],

$$I_1 = I_x - \frac{\beta}{4}(V_{in} - V_w)\sqrt{\frac{4I_x}{\beta/2} - (V_{in} - V_w)^2}, \tag{A.2}$$

and

$$I_2 = I_x + \frac{\beta}{4}(V_{in} - V_w)\sqrt{\frac{4I_x}{\beta/2} - (V_{in} - V_w)^2}, \tag{A.3}$$

with the differential input voltage in a finite region of

$$|V_{in} - V_w| \leq \sqrt{\frac{2I_x}{\beta/2}}. \tag{A.4}$$

Here I_x is the tail current of the differential pair, and $\beta = \mu \cdot C_{ox} \cdot \frac{W}{L}$ is the transconductance value of transistors M_1 and M_2. The output current of this synapse cell can be determined by

$$I_{out} = A \cdot I_x - (I_6 + I_9), \tag{A.5}$$

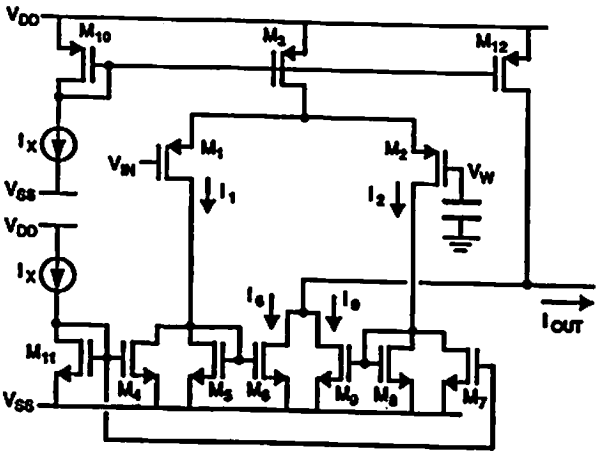
where A is the drain current ratio of transistor M_{12} to M_{10}. When $V_{in} - V_w < 0$, then $I_1 > I_x$ and $I_2 < I_x$. In this case, $I_6 = I_1 - B \cdot I_x$ and $I_9 = 0$. Here, B is the drain current ration of transistor $M_4(M_7)$ to M_{11}. On the other hand, when $V_{in} - V_w > 0$, then $I_1 < I_x$ and $I_2 > I_x$. In this case, $I_6 = 0$ and $I_9 = I_2 - B \cdot I_x$. Then the input voltage V_{in} is comparable to the synapse weight value V_w, transistors M_5, M_6, M_8, and M_9 nearly turn off and the output current is mainly contributed by a transistor M_{12}. Current gain values A and B can be chosen to better approximate the ideal Gaussian curve. Their typical values are quite close to one. The SPICE-3 [A8] circuit simulation result with weight value

being zero is shown in Fig. A.2(b). The simulated output current closely matches the ideal Gaussian function curve within the operational range.

An enhanced synapse cell with differential input/weight has also been developed. The circuit schematic diagram and transistor sizes are shown in Fig. A.3(a). Figure A.3(b) shows the comparison of the simulated output current curve of this enhanced synapse cell and the ideal Gaussian function curve. A better approximation than the basic synapse cell has been achieved due to symmetric handling of the positive and negative signals. Both the symmetric input and the synapse voltages are obtained with reference to the analog ground by inverting voltage amplifiers, which consist of operational amplifiers with the input and feedback resistors. The enhanced synapse cell approximates the Gaussian function with an accuracy better than 98 % over the input voltage range of $\pm$ 3 V in the ideal case when device imperfections suchs as mismatch, offset and so one are not considered. Device mismatch induced by fabrication process will cause some degradation to this accuracy. In fact, the usable output signal range of the enhanced synapse cell is almost doubled due to the use of differential circuit architecture. However, the area of the enhanced synapse cell is approximately twice of that of the basic synapse cell. The required silicon area for the basic synapse cell is 125 x 69 λ^2, and that for the enhanced cell is 146 x 99 λ^2 for the CMOS scalable design rule from MOSIS Service of USC/Information Sciences Institute at Marina del Rey, CA [A9].

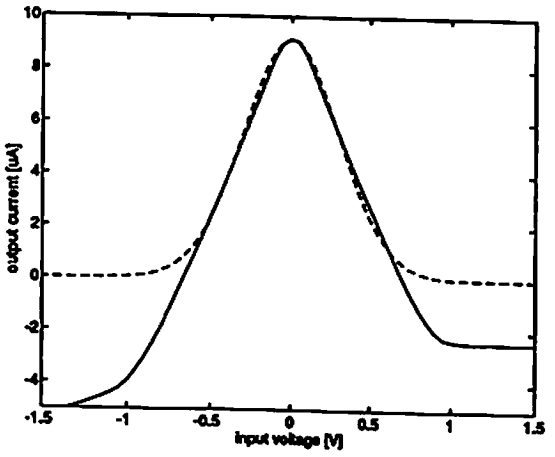
A.3 Programmability

A great power of artificial neural networks results from their ability to adapt to the changing environment. Therefore, good programmability is of fundamental



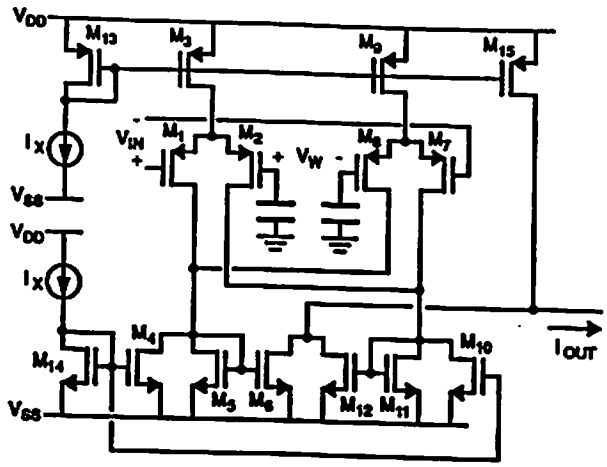
transistor	$M_{1,2}$	M_3	$M_{4,5,6,7,8,9}$	M_{10}	M_{11}	M_{12}
size (λ/Λ)	8/4	10/4	8/8	5/4	7/8	4/4

(a)



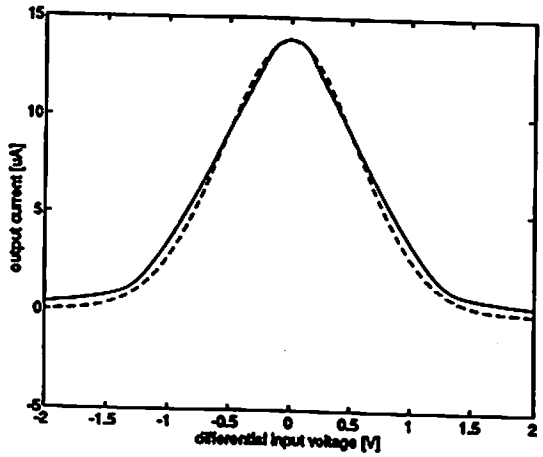
(b)

Figure A.2: Basic Gaussian synapse cell with single-ended input/weight values. (a) Circuit schematic. (b) SPICE simulation result. The solid line is the simulation results of the Gaussian synapse cell with the maximum magnitude of 9.14 μA, mean of 0, and standard deviation of 0.302. The dashed line is the ideal Gaussian curve.



transistor	$M_{1,2,7,8}$	$M_{3,9}$	$M_{4,10}$	$M_{5,6,11,12}$	M_{13}	M_{14}	M_{15}
size (λ/λ)	8/4	25/4	19/8	16/8	25/4	16/8	29/4

(a)



(b)

Figure A.3: Enhanced Gaussian synapse cell with differential input/weight values. (a) Circuit schematic. (b) SPICE simulation result. The solid line is the simulation results of the Gaussian synapse cell with the maximum magnitude of 13.91 μA, mean of 0, and standard deviation of 0.55. The dashed line is the ideal Gaussian curve.

importance in designing circuit building blocks for VLSI neural network processors. Three values are to be programmed in a Gaussian function: the maximum magnitude A_{ji}, the mean value W_{ji}, and the standard deviation σ_{ji} in (A.1).

Maximum magnitude

As seen from the previous section, the output current is controlled with respect to the tail current I_x of the input differential pair. By changing this reference current, the magnitude of the Gaussian output can be adjusted.

Mean value

The mean value is stored on the gate capacitance of an MOS transistor in the differential pair. Due to the possible leakage through the reverse-biased *pn*-junctions, periodic refreshing is necessary to keep the accurate synapse value. If the EEPROM device is used, the mean value can be stored permanently at the room temperature. Each mean value of a synapse cell can be accessed by an address decoder.

Standard deviation

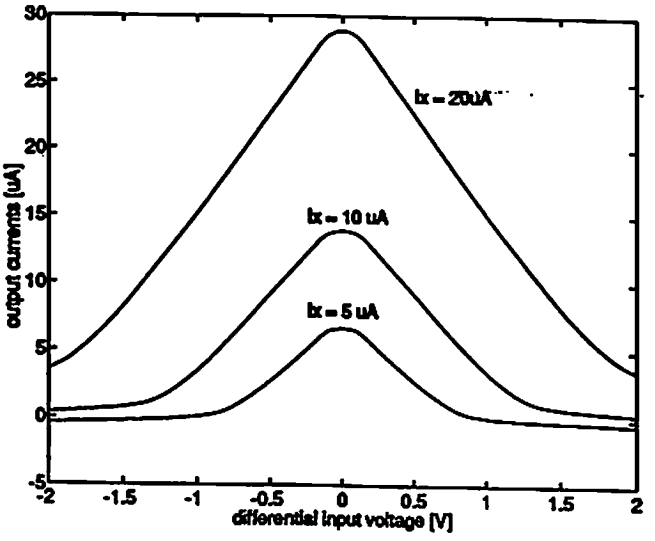
The standard deviation of the Gaussian function can be changed according to the constraints given by (A.4). For a fixed value of I_x, the shape of the output current curve can be varied by changing the W/L ratio of input transistors of the differential pair. In the differential pair, transistors with different sizes can be connected together through MOS switches which are controlled by the data stored in a local D-flip flop [A10]. By combining these programmable data, various sizes of input transistors in the differential pair bring the corresponding standard deviation of the Gaussian function.

The simulated results on the programmability of the proposed enhanced Gaussian synapse cell are shown in Fig. A.4 for different values of the amplitude, mean,

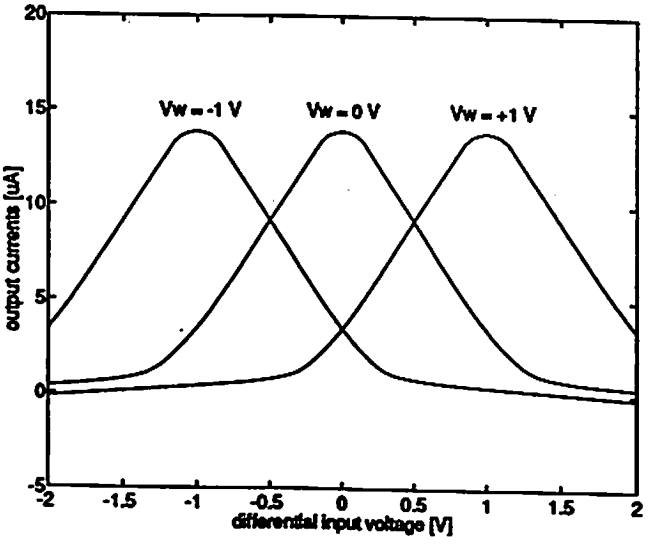
and standard deviation. In Fig. A.4(a), three Gaussian curves are created by setting the reference current to 5 μA, 10 μA, and 20 μA, respectively. Here, the input voltage is set to be equal to the weight value. In Fig. A.4(b), the weight value is changed to -0.5 V, 0.0 V, and 0.5 V, respectively. Here, the reference current is kept to a constant value of 10 μA. In Fig. A.4(c), the W/L ratio of transistors in the differential pair is changed to 1, 2, and 4, respectively.

Figure A.5 shows an example network for demonstrating the performance of the proposed Gaussian function network. Input neurons consist of unity-gain amplifiers as data buffers. The same input voltage value is applied to the four input neurons. A linear resistor in the output neuron converts the summed current into the output voltage. A minimum value of this feedback resistor is determined by the allowable output voltage which can be differentiated from the noise. When the number of synapses increases, the summed current may drastically increase because all current are unipolar. Thus, a proper value of the maximum feedback resistor value should be determined from the network size. Since the inverting input of the output neuron is virtually grounded, the contribution from one synapse cell current is independent of the output resistance of the synapse cell.

Figure A.6 shows the SPICE simulation results of four Gaussian synapses as shown in Fig. A.5. Here, four synapse values are set to be -1.6 V, 0.2 V, 1.3 V, and 2.5 V. In Fig. A.6(a), DC characteristics of four synapses are shown. The output current of each Gaussian cell is shown in solid lines and their summed current in the dashed line as the input voltage changes from -2.5 V to 2.5 V. Typical response time less than 100 $nsec$ is achieved for the internal capacitive load as shown in Fig. A.6(b).



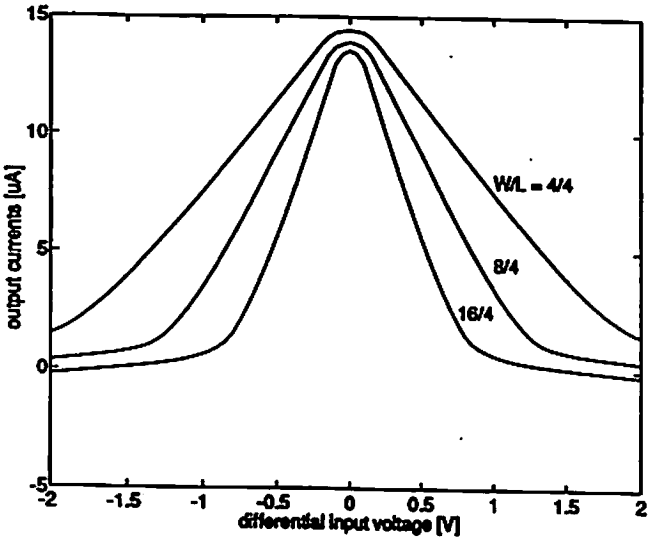
(a)



(b)

(continued on the next page)

(continued from the previous page)



(c)

Figure A.4: Programmability of the enhanced Gaussian synapse cell. (a) Different amplitudes with I_X being 5 μA, 10 μA, and 20 μA. (b) Different mean values with V_W being -0.5 V, 0.0 V, and 0.5 V. (c) Different standard deviations with 0.7308, 0.5515, and 0.3768 produced by the input transistor W/L ratios being 4 λ/ 4 λ, 8 λ / 4 λ, and 16 λ/ 4 λ.

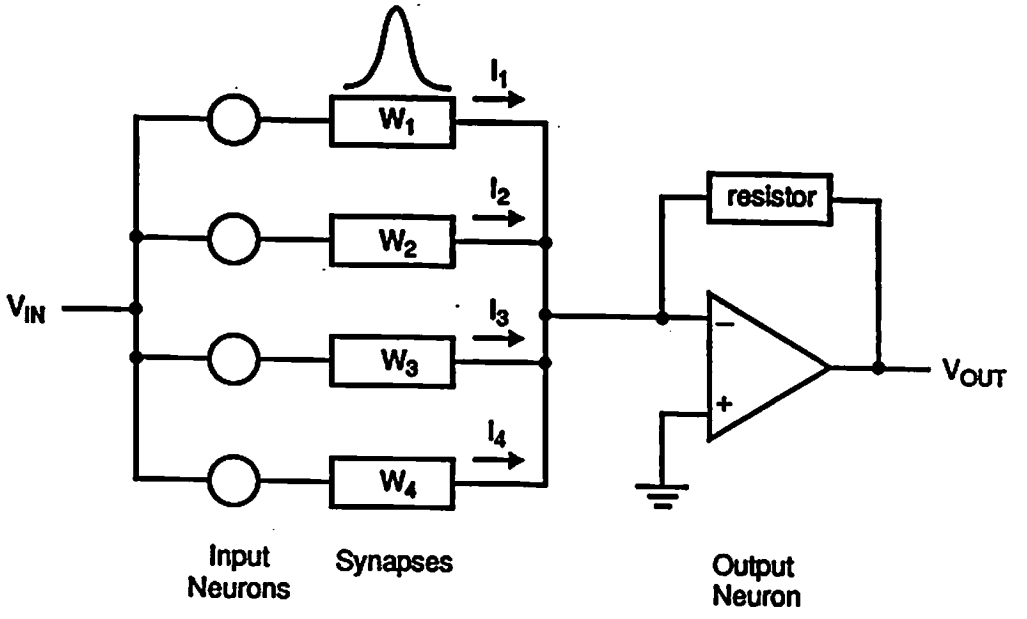
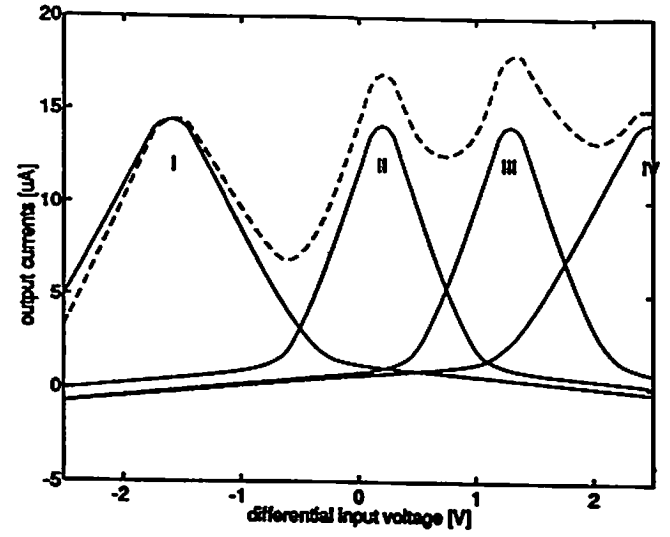
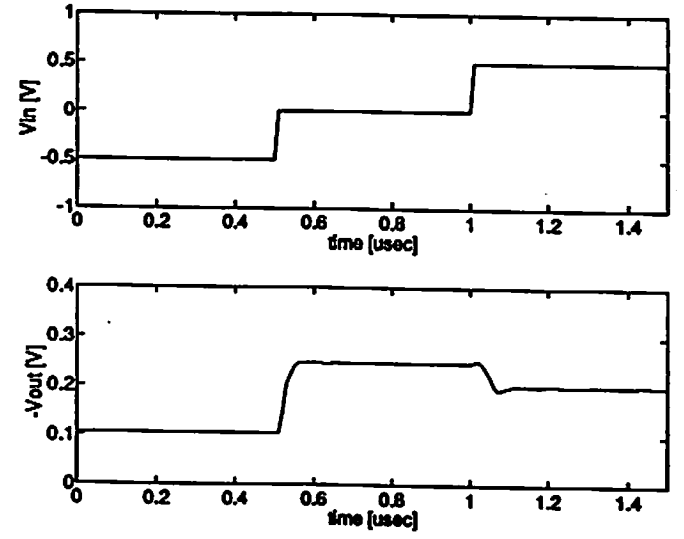


Figure A.5: An example network with four Gaussian synapse cells.



(a)



(b)

Figure A.6: (a) DC characteristics of the example network. Synapse-I: mean of -1.6, standard deviation of 0.55, Synapse-II: mean of +0.2, standard deviation of 0.38, Synapse-III: mean of +1.3, standard deviation of 0.38, and Synapse-IV: mean of +2.5, standard deviation of 0.55. Output currents from four individual synapse cells are shown in the solid lines and the summed current is shown in the dashed line. (b) Speed response of the example Gaussian network.

Reference List

[A1] J. Platt, "A resource-allocating neural network for function interpolation," *Neural Computation*, vol. 3, no. 2, pp. 213-225, Summer 1991.

[A2] A. L. Dajani, M. Kamel, and M. I. Elmasry, "Single layer potential function neural network for unsupervised learning," *Proc. IEEE/INNS Inter. Joint Conf. Neural Networks*, vol. II, pp. 273-278, San Diego, CA, June 1990.

[A3] T. Poggio and F. Girosi, "Networks for approximation and learning," *IEEE Proceedings*, vol. 78, no. 9, pp. 1481-1497, Sep. 1990.

[A4] B. J. Sheu, J. Choi and C.-F. Chang, "An analog neural network processor for self-organizing mapping," *Tech. Digest IEEE Inter. Solid-State Circuits Conf.*, pp. 136-137, San Francisco, CA, Feb. 1992.

[A5] S. S. Watkins and P. M. Chau, "A radial basis function neurocomputer implemented with analog VLSI circuits," *Proc. IEEE/INNS Inter. Joint Conf. Neural Networks*, vol. II, pp. 607-612, Baltimore, MD, June 1992.

[A6] C. A. Mead, *Analog VLSI and Neural Systems*, Addison-Wesley Publishing Company, 1989.

[A7] P. R. Gray and R. G. Meyer, *Analysis and Design of Analog Integrated Circuits, 2nd Ed.*, John Wiley & Sons, 1993.

[A8] B. Johnson, T. Quarles, A. R. Newton, D. O. Pederson, and A. Sangiovanni-Vincentelli, *SPICE3 Version 3E1 Users Guide*, Department of EECS, University of California, Berkeley, Apr. 1991.

[A9] C. Tomovich, "MOSIS-A gateway to silicon," *IEEE Circuits and Devices Magazine*, vol. 4, no. 2, pp. 22-23, Mar. 1988.

[A10] P. W. Hollis and J. J. Paulos, "Artificial neural networks using MOS analog multipliers," *IEEE Jour. of Solid-State Circuits*, vol. 25, no. 3, pp. 849-855, June 1990.

Appendix B

Nonideal Effects in WTA Circuit

B.1 Device Mismatches of Input Transistor

Since the proposed winner-take-all circuit is based on the multi-input source-coupled amplifiers, device mismatches of the input transistors M_1 in Fig. 4.3 mostly influence the operation to decide which cell is a winner. Common device mismatches resulting from the fabrication process are the threshold voltage change and device geometry variation. The drain current expression for an input transistor M_1 of each cell is given by,

$$I_j = \frac{\beta_j}{2}(V_j - V_{CM} - V_{th,j})^2 \quad for\ j = 1 \cdots N, \tag{B.1}$$

where $\beta_j = \mu C_{ox}\frac{W}{L}$ and $V_{th,j}$ are the transconductance parameter and the effective threshold voltage of an input transistor in the j^{th} cell, respectively.

Consider the drain currents given in (B.1) of the winning cell (I_W) and the losing cell (I_L). These currents can be expressed as,

$$I_W = \frac{\beta_W}{2}(V_W - V_{CM} - V_{th,W})^2 \tag{B.2}$$

and

$$I_L = \frac{\beta_L}{2}(V_L - V_{CM} - V_{th,L})^2, \tag{B.3}$$

where V_W and V_L are the winning input voltage and losing input voltage, respectively, so that V_W should be sufficiently larger than V_L. The minimum requirement in order to achieve the proper winner-decision operation should be

$$I_W > I_L. \tag{B.4}$$

The worst case occurs when

$$\beta_W < \beta_L \ \ and \ \ V_{th,W} > V_{th,L}. \tag{B.5}$$

The differential-mode components and the common-mode components for the β and V_{th} values can be defined as follows,

$$\Delta\beta = \beta_L - \beta_W, \tag{B.6}$$

$$\beta_C = \frac{\beta_L + \beta_W}{2}, \tag{B.7}$$

$$\Delta V_{th} = V_{th.W} - V_{th.L}, \tag{B.8}$$

and

$$V_{th,C} = \frac{V_{th,W} + V_{th,L}}{2}. \tag{B.9}$$

Based upon these equations, (B.2) and (B.3) can be changed into the following equations, respectively,

$$I_W = \frac{1}{2}(\beta_C - \frac{\Delta\beta}{2})\left(V_W - V_{CM} - V_{th.C} - \frac{\Delta V_{th}}{2}\right)^2 \tag{B.10}$$

and

$$I_L = \frac{1}{2}(\beta_C + \frac{\Delta\beta}{2})\left(V_L - V_{CM} - V_{th,C} + \frac{\Delta V_{th}}{2}\right)^2. \tag{B.11}$$

The differential-mode and common-mode components of the input voltages can be defined as,

$$\Delta V = V_W - V_L \tag{B.12}$$

and

$$V_C = V_W + V_L - 2(V_{CM} + V_{th}). \tag{B.13}$$

Then, the differential current of $I_W - I_L$ can be expressed as,

$$\begin{aligned} I_W - I_L &= \frac{1}{2}\left(\beta_C - \frac{\Delta\beta}{2}\right)\left(V_C + \frac{\Delta V}{2} - \frac{\Delta V_{th}}{2}\right)^2 \\ &\quad -\frac{1}{2}\left(\beta_C + \frac{\Delta\beta}{2}\right)\left(V_C - \frac{\Delta V}{2} + \frac{\Delta V_{th}}{2}\right)^2 \\ &= \frac{1}{2}\beta_C(\Delta V - \Delta V_{th})(2V_C) - \frac{\Delta\beta}{4}\left(2V_C^2 + \frac{\Delta V^2}{2} + \frac{\Delta V_{th}^2}{2} - \Delta V \Delta V_{th}\right) \\ &\simeq \beta(\Delta V - \Delta V_{th})V_C - \frac{\Delta\beta}{2}V_C^2, \end{aligned} \tag{B.14}$$

by neglecting the higher-order terms of differential-mode components. According to (B.4), the minimum voltage difference between the winning input and loser input values can be expressed as,

$$\Delta V > \Delta V_{th} + \frac{\Delta\beta}{2\beta_C}V_C. \tag{B.15}$$

Figure B.1 shows the calculation results of the minimum input voltage difference of (B.15) for β error of 1 %.

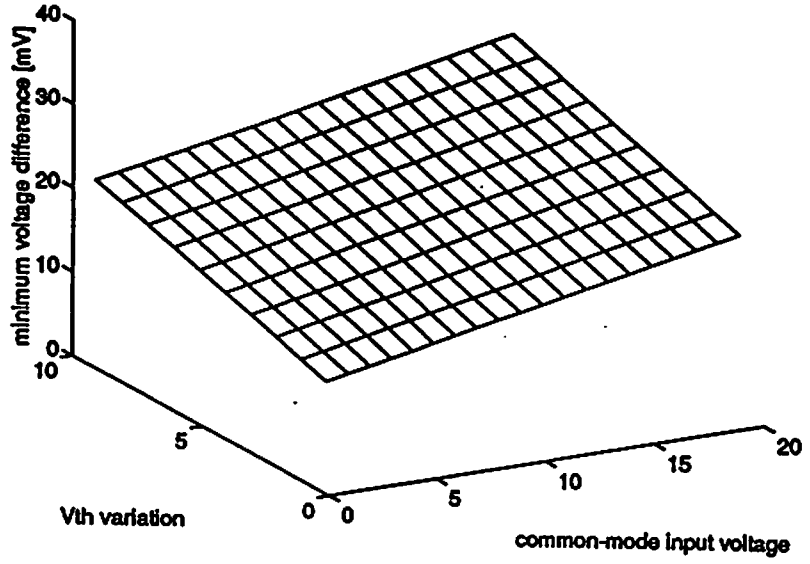


Figure B.1: Minimum input voltage difference between the winning input and the loser input. The mismatch error of a transconductance parameter β is 1 %. In the axis of *Vth variation*, ΔV_{th} changes from 0 V to 10 mV. In the axis of *common-mode input voltage*, V_C changes from 2 V to 4 V.

B.2 Parasitic Resistance

In the proposed WTA circuit, one of main sources to restrict the number of cells to be connected side by side is a parasitic resistance along the common signal line. Through this line, all input currents are redistributed and compared one another. In general, the common signal line is made of the metal, of which sheet resistance is very small. However, when the number of cells are significantly large, the length of this line should be so long that the resistance value between two far ends on this line cannot be ignored. The voltage drop across the common signal line causes a gate-to-source voltage of each input transistor to be different from the applied input voltage subtracted by the potential of the common sign line.

To analyze the effect of this finite resistance value along the common signal line on the number of the cells, simple model is introduced in Figure B.2, where each cell is represented by the equivalent current source flowing the cell current. The current flowing through each cell is I_B, the bias current of each cell, for a state of equilibrium when all input voltages are same. Thus the different input voltage applied to the cell results in deviation from the equilibrium value I_B such as,

$$I_j = I_B + \Delta I_j \tag{B.16}$$

with the condition of

$$\sum_{j=1}^{N} \Delta I_j = 0, \tag{B.17}$$

where N is the number of the competing cell. If the largest input voltage is applied to the $cell-1$ as the winning input, then the difference of the input voltages between this cell and $cell-j$ is expressed as,

$$V_{in}^{(1)} - V_{in}^{(j)} = \sqrt{\frac{2}{\beta}}\left(\sqrt{I_B + \Delta I_1} - \sqrt{I_B + \Delta I_j}\right) + R_{unit} \cdot \sum_{k=1}^{j}(j-k)\Delta I_k, \quad \text{(B.18)}$$

where R_{unit} is the unit resistor value of the common signal line between two adjacent cells. The first term in (B.18) is a voltage difference for the different current assuming the perfect device mismatch and no parasitic components. The second term in (B.18) is the voltage drop along the common signal line from $cell-1$ to $cell-j$ due to the parasitic resistor R. For the proper WTA operation, the magnitude of the first term must be larger than that of the second term.

For illustration purpose, the following condition is assumed: The winning input is applied to the $cell-1$ and the other input voltages are applied to the next cells in the descending order so that $V_{in}^{(1)} > V_{in}^{(2)} > V_{in}^{(3)} > \cdots > V_{in}^{(N)}$, and the difference of input voltages in the adjacent cell is identical so that $V_{in}^{(j)} - V_{in}^{(j+1)}$ are identical for all j. From these assumption, the following condition for the currents can be derived as,

$$I_j = I_B + \left(\frac{N-1}{2} - (j-1)\right) \cdot \Delta I, \quad \text{(B.19)}$$

where ΔI is the constant differential current value. Here, the winning cell current I_1 is $I_B + \frac{N-1}{2} \cdot \Delta I$, the middle cell current $I_{(N+1)/2}$ is I_B, and the last cell current I_N is $I_B - \frac{N-1}{2} \cdot \Delta I$. Based upon this condition, two components in (B.18) are shown in Figure B.3 for ΔI of 80 nA. From the fabrication results, because the sheet resistance value of the metal line is 0.026 Ω per square and there are 20 squares in the common signal between the adjacent cells, the unit resistor R_{unit}

value is 0.52 Ω. Figure B.3 shows that more than 400 cells can be connected in series and Figure B.4 shows the method to increase the number of competing cells over 400.

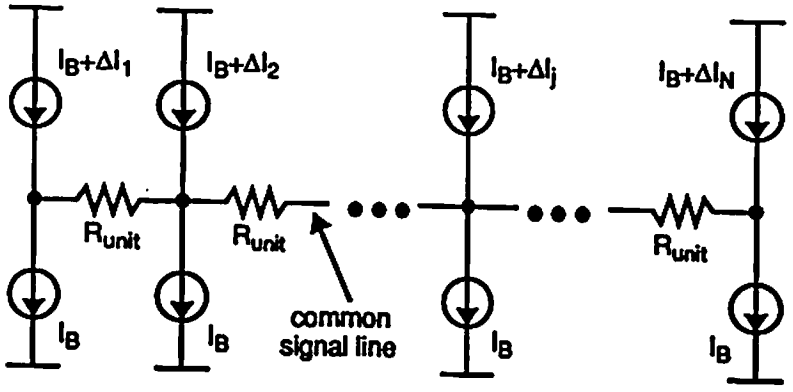


Figure B.2: Simplified model of the proposed WTA circuit.

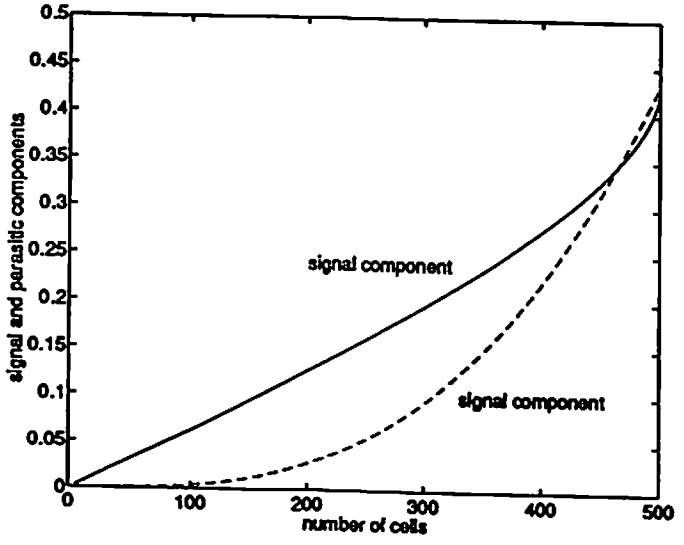


Figure B.3: Signal and offset components due to the parasitic resistance in the common signal line versus the number of competing cells.

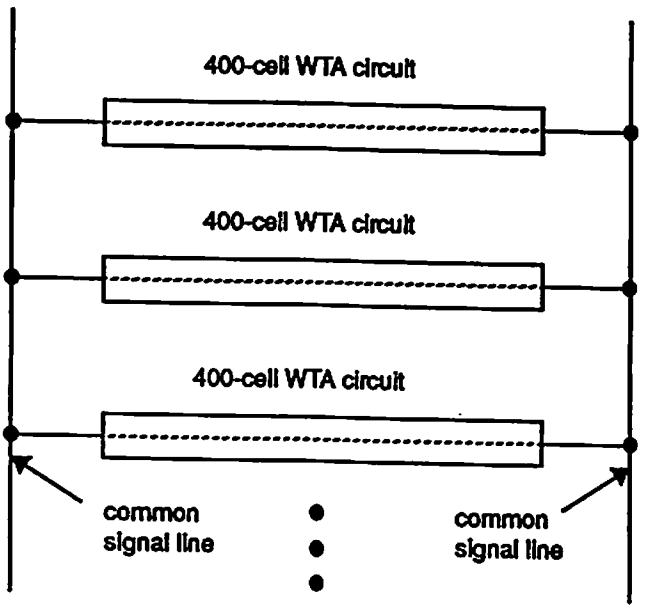


Figure B.4: Method to extend the number of competing cells regardless of the parasitic resistor.