USC–SIPI REPORT #282

Compact VLSI Array Processors Design for Multimedia Applications

by

Robert Chen-Hao Chang

April 1995

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.

I dedicate this dissertation to my parents,

Kun-Huang Chang

and

Yun-Fei Chang-Liu,

and my wife,

Ling-Chuan Lai.

Acknowledgments

I am deeply grateful to my research advisor, Professor Bing Sheu, for his generous guidance, encouragement, and support throughout these years of graduate study. I am also grateful to Professor Murray Gershenzon and Professor Wlodek Proskurowski for serving on my dissertation committee. I would like to thank Professor Sandeep Gupta and Dr. Wai-Chi Fang to serve as the other two members on my Ph.D. Qualifying Examination committee.

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 Robert A. Scholtz, Chairman of the Electrical Engineering - Systems Department; Professor Alvin Despain, Professor Alice C. Parker, Ms. Ramona Gordon, Ms. Anna Fong, Ms. Evelyn Jamora, and Ms. Gloria Halfacre in the Electrical Engineering Program, for providing me the great research environment for my Ph.D. study at the University of Southern California (USC). The support of several research organizations including the Signal and Image Processing Institute (SIPI), the Center for Neural Engineering (CNE), the Center for Photonic Technology (CPT), and the MOSIS Service of Information Science Institute (ISI) is appreciated. Very valuable interaction with Professor Ted Berger of Biomedical Engineering Department was highly appreciated.

Important discussions with graduated doctoral colleagues from VLSI Signal Processing Laboratory were truly valuable, including Dr. Wen-Jay Hsu and Dr. Sudhir

Gowda on circuit simulation, Dr. Sa H. Bang and Dr. Joongho Choi on VLSI array processing chips design, Dr. Josephine C.-F. Chang on digital VLSI design, and Dr. Oscal T.-C. Chen on image and information processing. Many thanks to Tony H. Wu, Eric Y. Chou, and Steve H. Jen for helping to obtain some simulation results. I also thank Richard H. Tsai and David C. Chen for managing the computing facility.

I whole-heartedly express my appreciation and gratitude for the love, inspiration, and encouragement of my father, Kun-Huang Chang, who passed away during my study. I am eternally grateful to my mother, Yun-Fei Chang-Liu, and my sisters, Mei-Chen, Li-Wen, and Mei-Ling, for their love and support. I am extremely thankful to my wife, Ling-Chuan Lai, for her love, understanding, and encouragement.

Contents

List Of Tables

List Of Figures

Abstract

With rapid advances of deep-submicron microelectronic technologies, a high-performance intelligent system with tens of millions of transistors can be integrated onto a single chip. The compact, high-computing power systems become feasible with significant progresses in the research and development of advanced computing architecture and array processing. Extensive studies of artificial and biological neural networks, which have inherent massively paralleled and distributed signal processing capabilities, have provided an excellent means to perform several complex functions in scientific and engineering applications such as image/pattern recognition, medical image, computer vision, path planning, and autonomous robots. Array processors based on cellular neural networks combine some features of fully interconnected neural networks with the nearest neighbor interactions and are especially well suited for very large-scale integration (VLSI) implementation. A 5×5 paralleled array processor chip has been designed and fabricated by using the 2-μm CMOS technology from the MOSIS Service. The prototype chip with digitally-programmable weights was constructed with many compact mixed-signal VLSI circuit components which were designed using the current-mode techniques. The low-voltage, low-power operation is supported with the current-mode scheme which scales well with the supply voltage. Measurement results of the VLSI computing cells are presented. Experimental results obtained from a custom-made circuit board are provided to illustrate the operation of the prototype chip. The software-hardware codesign methodology

is used to implement the high-performance intelligent microsystem which can be constructed by the array processor chips and software program. Neural networks with the hardware annealing method are very energy-efficient in solving many complex optimization problems. Demonstration of novel operation to achieve optimal solution at fast signal processing using standard IC parts is given. VLSI design of a variable-gain neuron circuit can be incorporated into the prototype chip to realize the optimal solution capability.

Chapter 1

Introduction

Rapid progresses in the research and development of array processing have made significant impacts in scientific, engineering, and daily-life applications. Many researchers have been contributing their efforts in this area from different levels including algorithms, architectures, microelectronic and micromechanical implementations. One of the key driving forces to build the intelligent microsystems is the continuous advances of the deep-submicron VLSI technology which has made it possible to integrate tens of millions of transistors onto a single micro-chip. Extensive studies of artificial and biological neural networks have provided excited evidence for high-performance information processing with the paralleled and distributed signal processing capabilities. It is important to combine the advanced computing architectures with the new enabling technologies to explore the capability of the intelligent microsystems. This dissertation is concerned with the mixed-signal VLSI design for the array processor to be used in future intelligent microelectronic systems.

1.1 Intelligent Microsystems

An intelligent microsystem is a high-performance smart machine which can accept different kinds of signals from the real world and provide output signals after internal processing. The machine would have the capability of learning, reasoning and making correct decision [1]. A microsystem can sense the video signals through a camera, a scanner or other image detecting devices, then process the information via various efficient algorithms for tasks such as image processing, vision processing or pattern recognition. It can also receive the audio signals through a microphone, then process the data through speech recognition, or speaker recognition. Other real-world signals, such as heat and pressure, can be detected by the microsensors. After the incoming signals are processed, the intelligent microsystem can send the output results to different output devices in different signal formats such as an image or text display, synthesized or translated speech signals and commands for robotics or microcontrollers.

The intelligent microsystems can be used to support the multi-media applications such as tele-conferencing and virtual-reality entertainment and education systems [2, 3]. By using the face recognition and speaker recognition techniques, reliable person identification could replace locks and keys in many instances [1]. It can also be utilized in many medical applications. For example, a surgery robot can accept commands from a doctor and perform the actual operation [4]. In addition, a microsystem equipped with image enhancement capability can help the doctors search the possible cancer cell locations on a X-ray film. These information processing tasks require lots of computation power so that the designs at the circuit and architectural levels with the advanced VLSI technology are very critical.

1.2 Capabilities of VLSI Technology

With rapid progress of submicron VLSI technologies during the past decade, tens of millions transistors can be integrated on a single silicon chip. For example, it would be possible to integrate 16 MB DRAM, 256 KB SRAM, and a microprocessor on one chip using a modified 256 MB DRAM technology which is augmented with the necessary metal layers. This integration capability can be increased by approximately two orders of magnitude before reaching the fundamental limits of the technology governed by laws of device physics.

Currently, the widely used technology for the microprocessor chips is 0.4 or 0.5 μm CMOS technologies with four or five layers of metal [5, 6, 7], while that for the memory chips is 0.25 μm CMOS technology [8, 9, 10]. The very competitive high-performance workstation and PC industries have resulted in rapid advancement of RISC and CISC microprocessors. Extraordinary improvement of semiconductor technology plays a major role in this advancement. In 1995, a microprocessor consisting 9.3 million transistors can process up to 64 instructions at one time with peak performance reaching 1.2 billion instructions per second from Digital Equipment Corp. [7]. Table 1.1 lists the recent development results on microprocessors [5, 6, 7, 11, 12, 13, 14].

There are two main reasons to implement the neural networks by using the advanced VLSI technology. First, the topology of neural networks is very regular and the number of well-defined operations involved in their learning algorithms is relatively small so that the design and layout of VLSI chips is greatly simplified. Second, the high functional density achievable in VLSI technology permits the implementation of a large number of identical, concurrently operating neurons on a

single silicon chip, which greatly facilitate the exploitation of inherent parallelism of neural networks.

1.3 Artificial Neural Networks

The straight line shown in Figure 1.1 indicates the performance trend of advanced VLSI systems [15]. As the performance of the VLSI systems being further enhanced with increasing in the integration level and in the operating frequency, applications of VLSI systems will be extremely expanded. For example, in the video area, the TV phone is the major application in the late 1980's. In the 1990's, the continuously strong demand of the consumer market has been pushing the video applications to the higher-performance multi-media computers, the digital HDTVs, and the super high definition TVs [15].

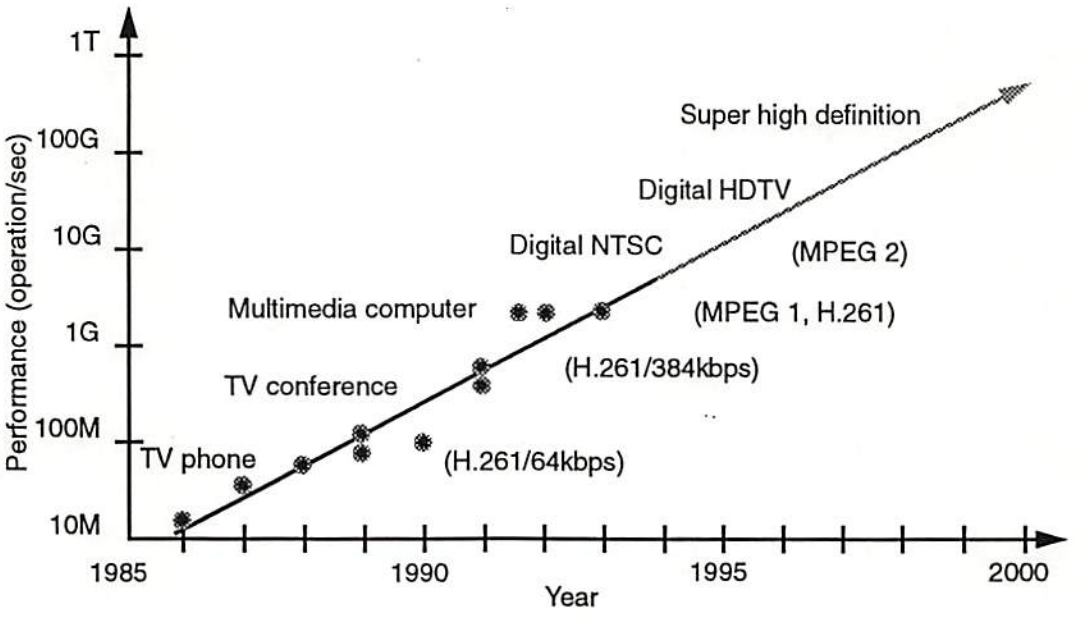


Figure 1.1: Performance trends of the VLSI systems.

Table 1.1: Summary of the microprocessors designed in the industry.

Yr	Characteristics	Technology	Speed	# of Transistors	Power (W)	Chip Size (mm^2)	Package	Maker
91	32b 80486	0.8-µm 3M CMOS	100MHz	1.2 M	8.0	6.8 x 11.8	-	Intel
92	64b dual-issue	0.75-µm 3M CMOS	200MHz	1.68 M	30 @ 3.3 V	13.9 x 16.8	431 PGA	Digital Equip-ment Corp.
92	superscaler	0.8-µm 3M BiCMOS	50MHz	3.0 M	9.0	15.98 x 15.98	293 PGA	SUN Microsystems
93	Pentium superscaler	0.8-µm BiCMOS	66MHz	3.1 M	-	-	-	Intel
93	32b Bipolar ECL	1.0-µm 5M Bipolar	300MHz	670 k	115 @ -5.2 V	15.4 x 12.6	504 PGA	Digital Equip-ment Corp.
94	32b RISC	0.5-µm 3M CMOS	80MHz	-	3 @ 3.3 V	7.39 x 11.47	240	IBM
94	64b RISC	0.55-µm 3M CMOS	140MHz	1.2 M	29 @ 4.4 V	14 x 15	540 PGA	Hewlett-Packard Co.
95	64b RISC quad-issue	0.5-µm 4M CMOS	300MHz	9.3 M	50 @ 3.3 V	16.5 x 18.1	499 IPGA	Digital Equip-ment Corp.
95	64b micro-processor	0.5-µm 4M CMOS	167MHz	5.2 M	30 @ 3.3 V	17.7 x 17.8	520 BGA	SUN Microsystems
95	64b super-scalar 4-issue	0.5-µm 4M CMOS	133MHz	6.88 M	30 @ 3.3 V	18.2 x 17.1	625 BGA	IBM

Since the computation requirement of system applications is growing from mega-bit-operation to multiple giga-bit-operation and tera-bit-operation, even the most powerful conventional microprocessors with giga-bit-operation capability can not afford to carry out these operations. The conventional Von Neumann computing approach consists of a single central processing unit and a main memory unit. It can sequentially execute the pre-stored instructions with a reasonable speed and accuracy for conventional data-processing applications. However, these digital computing machines, when packaged in a small physical size, can not perform computationally-intensive or ill-defined tasks with satisfactory performance in such areas as intelligent perceptron, including visionary and auditory signal processing, recognition, understanding, and logical reasoning where human being and even living animals can do a superb job [16]. Although modern digital signal processing (DSP) or reduced instruction-set computing (RISC) processors have a very high throughput rate up to giga bit operations per second, certain complex applications such as tracking of targets in real time and data fusion require more computational power than the existing RISC or DSP processors can provide.

Advances in biological-inspired neural engineering and VLSI design technology have provided an excellent means to perform several complex functions in scientific and engineering applications including pattern recognition, vision information processing, optimization [17, 18], adaptive control [19, 20], signal detection [21], data analysis [22], biomedical instrumentation [23], and so on. Studies of engineering neural network models were motivated by the investigation of human perceptron. The secret lies in the design optimization at various levels of computing and communication. Each neural network system consists of massively paralleled and distributed signal processors with every processor performing very simple operations.

Large computational capabilities of these systems are derived from collectively parallel processing and efficient data routing through well-structured interconnection networks.

The most fundamental feature of biological neural systems is their ability to carry out wide-scale collective computation by means of true physical parallelism. These systems also make a very efficient use of the hardware. Small-sized neural network IC modules with basic functionality have been successfully implemented using analog, digital, or hybrid analog-digital CMOS VLSI technologies. Since the human brain uses analog components in a massively parallel fashion, the analog VLSI implementation of neural networks could be an excellent way to build microelectronic systems that are similar to the biological systems. Analog circuits that compute sums of products can be built much more compactly than digital circuits. In addition, the large interconnectivity and the moderate precision required in neural network models present new opportunities for analog computing.

Analog implementation of artificial neural networks can be treated as one subclass of analog array processors [24]. The emerging field of regular analog processing arrays has attracted much interest worldwide by solving some real-life problems which are difficult or too time consuming for classical digital computers. Analog array processors combine some features of fully interconnected analog neural networks with the nearest neighbor interactions found in cellular automata and are especially well suited for VLSI implementation.

1.4 Existing Analog Neural Chips

Analog VLSI implementations of neural networks have been presented by many researchers in the academia and industry [25, 26, 27, 28, 29, 30, 31]. The Electrically

Trained Analog Neural Network (ETANN) from Intel Corp. was presented by Holler et al. [25]. The chip has 64 output neurons or threshold amplifiers with sigmoidal transfer function. Virtually, there are 128 neurons because the same 64 neurons can be used for both the middle and output layers. After an analog voltage signal which ranges from 0 V to 3.5 V is entered to one of the 64 inputs, it is presented to a synapse. The output signal of the synapse is a differential current proportional to the multiplication of the input voltage and a stored synapse weight value. The current sum of the inner products of 64 inputs and 64 components of a row of the synapse array is transmitted to the neuron which corresponds to that row. Besides, there are 16 internal fixed bias voltages connected to each neuron. Therefore, each neuron must receive a total sum of 80 synapse products. There are $80 \times 64 = 5{,}120$ synapses in the first-layer array and another 5,120 synapses in the second-layer array. The ETANN chip can achieve 2 billion connections per second by fully paralleled analog computation.

The ANNA chip was reported by Boser et al. of AT&T Bell Labs. [26]. The chip contains 8 neuron cells and 4,096 synapse weights. The synapses can be clustered into groups of size 64, 128, and 256. The internal processing is in analog format to reduce the power dissipation. However, the input/output interface is in digital format in order to simplify the system integration. The total number of transistors in this chip is 180K. It was fabricated in a single-poly, double-metal 0.9-μm CMOS technology for a 5-V power supply operation. Computation is performed with 3-bit accuracy for the neuron states and 6-bit accuracy for the synapse weights. The chip can perform over 2,000 multiplications and additions simultaneously. It has been used in an implementation of a neural network for optical character recognition.

Alspector et al. presented a cascadable learning chip using feedback connections and a local learning rule [27]. Their chip can perform a stochastic supervised learning algorithm similar to the Boltzmann machine and a mean-field theory learning algorithm. It consists of 32 neurons and 496 bidirectional adaptive synapses and can be programmed into full symmetry or some asymmetry configuration. There are on-chip biasing circuits and an on-chip 32 channel uncorrelated analog noise generator. A system, which is composed of the cascadable learning chip, data generators and analyzers, has been used to solve the parity and replication problem. The learning speed, which is limited by the system settling time, is about 100K patterns per second and is quite independent of the system size.

Carver Mead and his group at California Institute of Technology, Pasadena, CA, have done extensive work in the area of analog, biologically inspired neural network using silicon technologies [28]. It is possible to realize good non-linear characteristics by operating MOS transistors in the subthreshold region. Subthreshold operation consumes very low power. Although the operation speed of individual devices is slow due to the low current level in the subthreshold region, high performance results from massive parallelism can be achieved. Several basic components operated in the subthreshold region have been built such as transconductance amplifier. One of their famous achievements is the silicon retina chip which can generate results that are very close to those obtained from biological systems. Another example is the analog VLSI implementation of the distal portion of the vertebrate retina model on the silicon substrate which is a very fascinating design.

A reconfigurable analog VLSI neural network chip which contains 1,024 distributed neuron synapses was reported by Satyanarayana et al. [29]. It was fabricated by a 0.9-μm, double-metal, single-poly, n-well CMOS technology. The

distributed-neuron synapses are arranged in blocks of 16. In order to provide programmability of interconnections, switch matrices are interleaved between each of these blocks. The synaptic weights are stored in analog format on MOS capacitors and usable to a resolution of 1 % of their full scale value.

A self-learning neural network chip was developed by Arima et al. [30]. The chip integrates 336 neurons and 28,224 synapses with a 1.0-μm double-poly-Si double-metal CMOS technology. The operation speed can be higher than one tera connections per second. The chip is fully feedback connected and expandable by interconnecting multiple chips. The learning speed of the chip is 28×10^9 connection updates per second.

An analog neural network processor is reported by Sheu et al. [31]. The chip has 64 output neurons and 1,600 synapses uses a current-mode approach to perform analog multiplication and summation in parallel. Response times as low as 50 ns at a 1-pF load capacitance can be achieved by the two-stage winner-take-all circuit using the distributed biasing scheme and the dynamic steering circuit [32]. It is implemented in 2-μm CMOS technology. The chip can be used for self-organizing mapping and its computing capability is as high as 3.33 billion connections per second.

1.5 Organization of This Dissertation

The rest of this dissertation is organized as follows.

Chapter 2 introduces the basic theory and computation paradigm. Architecture of the processing element in the paralleled array processor is described. Applications of the array processor are discussed.

Chapter 3 covers the design of a 5×5 array processor chip which was fabricated in a 2-μm CMOS technology through the MOSIS Service. Design of important circuit building blocks using current-mode techniques is described in detail. Measurement results of the basic components and the prototype chip are presented in this chapter.

Chapter 4 presents the local-minimum problem which exists in the recurrent networks and the hardware annealing method which can be utilized to quickly search the global-minimum solution in a very effective way. VLSI circuit design of a variable-gain neuron cell for this purpose is described.

Chapter 5 summarizes the results of this dissertation.

Appendix A includes the CMOS technology files for the two different circuit-level models.

Appendix B Selected templates and its corresponding applications are presented.

Appendix C describes several fundamental design issues including physical layout, low-power design, optical input capability, and scalability of the array processors.

Appendix D includes software modules for compact software-hardware codesign systems.

Appendix E describes the design of a low-power transceiver chip for the dual-mode wireless communication systems.

Chapter 2

Overview of Theory and Architecture of Paralleled Array Processors

Paralleled array processors based on cellular neural networks (CNNs) [24, 33] are very useful in high-speed, real-time applications. CNNs are continuous- or discrete-time artificial neural networks that consist of the multi-dimensional array of processing elements. The processing elements are locally interconnected with their neighboring elements. By use of pre-determined templates, many complex scientific and engineering problems in signal/image processing and optimization could be solved by the networks. In this chapter, the basic theory and computation paradigm is reviewed, and the architecture of the processing element is described.

2.1 Basic Theory and Computation Paradigm

CNNs are continuous- or discrete-time networks with locally-interconnected processing elements which perform very simple synaptic operation. The original theory and architecture of cellular neural networks were proposed by Chua and Yang [24, 33] in 1988. The network could be an N-dimensional triangular-, rectangular-, or hexagonal-grid array [34]. The paralleled array processors considered here is based on a two-dimensional $n \times m$ rectangular-grid array network where n and m

are the number of rows and columns, respectively. Each element $C(i,j)$, $1 \leq i \leq n$, $1 \leq j \leq m$, in the array processor corresponds to a cell in the CNN and is interconnected with its neighbor elements $N_r(i,j)$, where $N_r(i,j)$ are defined as the elements $C(k,l)$, $1 \leq k \leq n$, $1 \leq l \leq m$, for which $|k-i| \leq r$ and $|l-j| \leq r$ and r is the distance between the element $C(i,j)$ to its farthest neighbor element. Figure 2.1 shows an $n \times m$ array processor with $r = 1$. The neighboring elements $N_1(i,j)$ of $C(i,j)$ are also indicated in the figure.

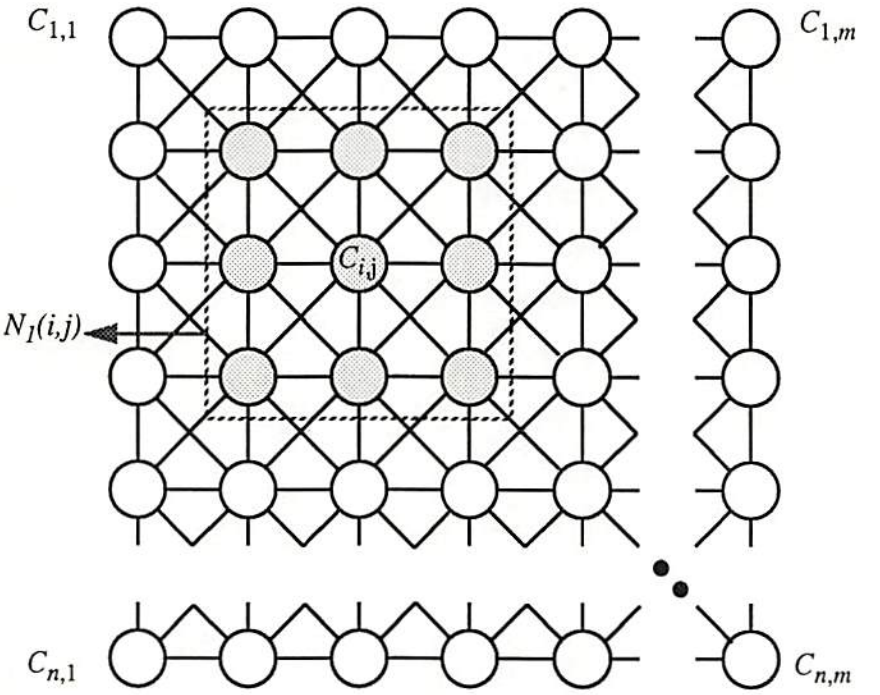


Figure 2.1: An $n \times m$ paralleled array processor.

Figure 2.2 shows the functional diagram of a processing element $C(i,j)$. The internal state signal and the output signal of the processing element are denoted by $v_x(i,j)$ and $v_y(i,j)$, respectively. The external input signal to the processing element is usually assumed to be constant over the network operation time interval $0 \leq t < T$ and it is denoted by $v_u(i,j)$. There are two kinds of weights for the processing element $C(i,j)$ to communicate with its neighboring $N_r(i,j)$, i.e., the feedback weights

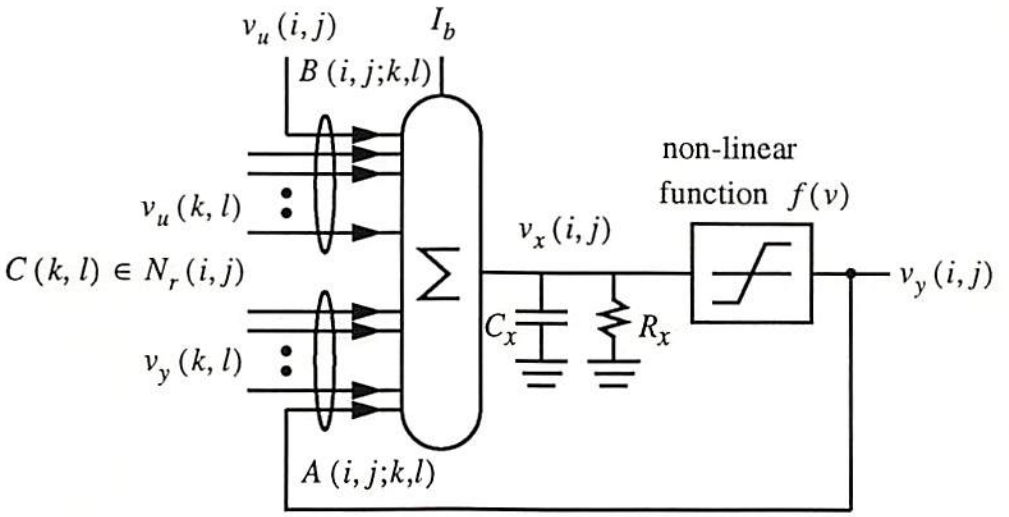


Figure 2.2: Functional block diagram of a processing element.

$A(k,l;i,j)$ & $A(i,j;k,l)$ and feedforward weights $B(k,l;i,j)$ & $B(i,j;k,l)$. The output signal of $C(k,l)$, $v_y(k,l)$, is connected to $C(i,j)$ through the feedback weight $A(i,j;k,l)$. On the other hand, the output signal of $C(i,j)$, $v_y(i,j)$ is connected to $C(k,l)$ through the feedback weight $A(k,l;i,j)$. Similarly, the input signal of $C(k,l)/C(i,j)$, $v_u(k,l)/v_u(i,j)$ is connected to $C(i,j)/C(k,l)$ through the feedforward weight $B(i,j;k,l)/B(k,l;i,j)$. Although the processing element $C(i,j)$ is only connected to its neighboring elements $C(k,l) \in N_r(i,j)$, it can indirectly communicate with all other elements in the whole array processor.

As shown in Fig. 2.2, the input signal to the state node of the processing element consists of the weighted sum of feedforward input signals, the weighted sum of feedback signals, and a constant bias term I_b. The bias term is used to adjust the threshold value of the neuron. An integration operation is performed at the state node through the equivalent resistance R_x and equivalent capacitance C_x. Assume the processing elements throughout the network have the same I_b, R_x, and C_x values and the weights A and B represent the transconductance values among

the elements. The state voltage $v_x(i,j)$ is established at the state node and satisfies a set of differential equations given as [24, 35]

$$\begin{aligned} C_x \frac{dv_x(i,j)}{dt} &= -\frac{1}{R_x} v_x(i,j) + \sum_{C(k,l)\in N_r(i,j)} A(i,j;k,l) v_y(k,l) \\ &+ \sum_{C(k,l)\in N_r(i,j)} B(i,j;k,l) v_x(k,l) + I_b;\ 1 \le i \le n, 1 \le j \le m. \end{aligned} \tag{2.1}$$

The output signal of the element is obtained by nonlinear transformation of the state voltage. It can be represented by $v_y(i,j) = f(v_x(i,j))$ where the nonlinear transfer function can be an appropriate non-decreasing function $y = f(x)$, provided that $f(0) = 0$, $f(+\infty) \to +1$ and $f(-\infty) \to -1$. A widely used nonlinear transfer function is the sigmoid functions as given by

$$y = f(x) = \frac{1 - e^{-\lambda x}}{1 + e^{-\lambda x}}. \tag{2.2}$$

where the parameter λ is proportional to the gain of the sigmoid function. A sigmoid function with $\lambda = 2$ is shown in Fig. 2.3. Even the steady-state outputs have to take

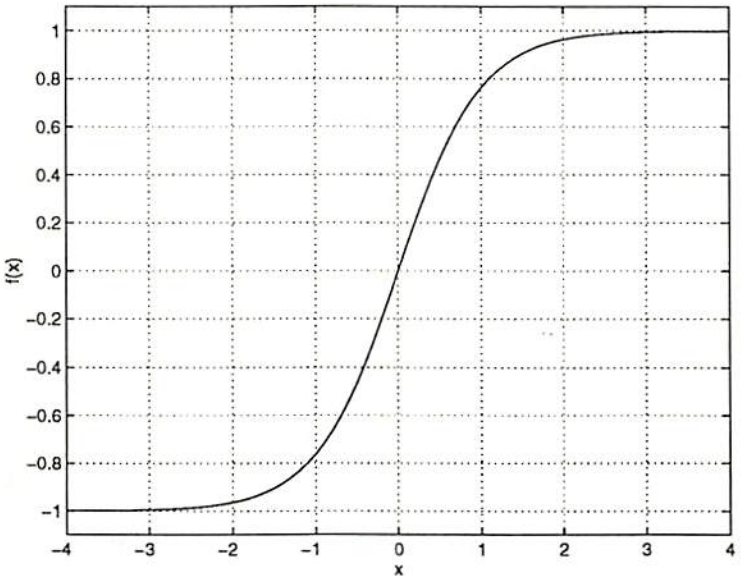


Figure 2.3: Sigmoid function.

the binary values for cellular neural networks, the gain of the cell doesn't have to

be large because the positive feedback factor in the network could be greater than one. Usually, a unity gain $df(x)/dx|_{x=0} = 1$ is used in the network. The sigmoid-like nonlinear function can be easily obtained by using microelectronic circuits such as operational amplifiers in the voltage-mode operation. However, the piecewise-linear function is easier for mathematical analysis and the microelectronic circuits in the current-mode operation. Therefore, the following architecture and circuit design are based on the piecewise-linear transfer function as given by

$$y = f(x) = \frac{1}{2}(|x+1| - |x-1|). \tag{2.3}$$

Figure 2.4 shows the transfer characteristic of the piecewise-linear function.

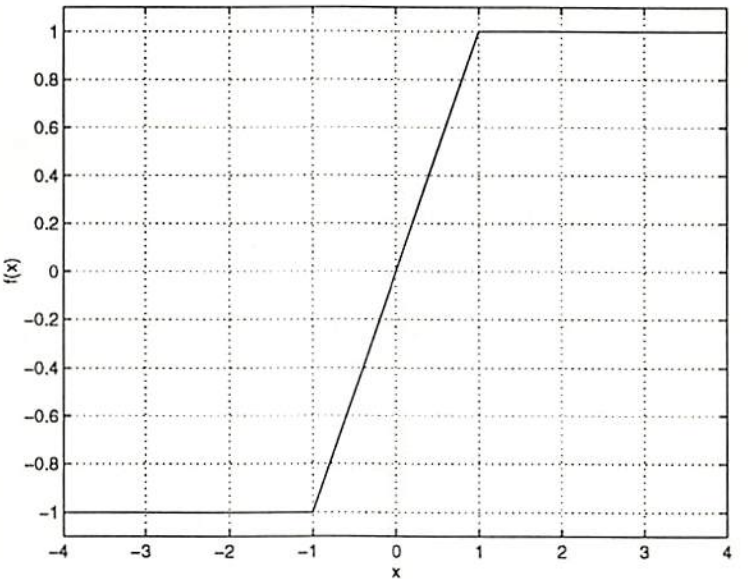


Figure 2.4: Piecewise-linear function.

The feedback and feedforward weights of the network do not depend on the positions of the elements in the array processors except at the edges because the elements located on the edge have fewer neighbors than those inside the network. Special boundary elements with time-invariant state variables could be added on the surrounding of the network [36, 37]. This property is a very attractive feature

of the network for the VLSI implementation of a large number of processors. The interconnection weights of the network can be represented by the $(2r+1) \times (2r+1)$ feedback and feedforward cloning templates [16, 35],

$$\mathbf{T}_A = \begin{bmatrix} a_{-r,-r} & a_{-r,-r+1} & \cdots & a_{-r,r} \\ a_{-r+1,-r} & a_{-r+1,-r+1} & \cdots & a_{-r+1,r} \\ \vdots & \vdots & \cdots & \vdots \\ a_{r,-r} & a_{r,-r+1} & \cdots & a_{r,r} \end{bmatrix} \tag{2.4}$$

$$\mathbf{T}_B = \begin{bmatrix} b_{-r,-r} & b_{-r,-r+1} & \cdots & b_{-r,r} \\ b_{-r+1,-r} & b_{-r+1,-r+1} & \cdots & b_{-r+1,r} \\ \vdots & \vdots & \cdots & \vdots \\ b_{r,-r} & b_{r,-r+1} & \cdots & b_{r,r} \end{bmatrix}. \tag{2.5}$$

By use of the vector and matric notations, (2.1) can be re-written as [16, 35]

$$C_x \frac{d\mathbf{x}}{dt} = -\frac{1}{R_x}\mathbf{x} + \mathbf{A}\mathbf{y} + \mathbf{B}\mathbf{u} + I_b\mathbf{w}, \tag{2.6}$$

where

$$\begin{aligned} N &= n \times m = \text{number of elements in the network}, \\ \mathbf{u}_{N\times 1} &= [u_1\ u_2\ \cdots\ u_N] = [\mathbf{v_{u1}}|\mathbf{v_{u2}}|\cdots|\mathbf{v_{un}}]^{\mathbf{T}}, \\ \mathbf{x}_{N\times 1} &= [x_1\ x_2\ \cdots\ x_N] = [\mathbf{v_{x1}}|\mathbf{v_{x2}}|\cdots|\mathbf{v_{xn}}]^{\mathbf{T}}, \\ \mathbf{y}_{N\times 1} &= [y_1\ y_2\ \cdots\ y_N] = [\mathbf{v_{y1}}|\mathbf{v_{y2}}|\cdots|\mathbf{v_{yn}}]^{\mathbf{T}}, \\ \mathbf{A}_{N\times N} &= toeplitz((\mathbf{A_0}|\mathbf{A_1}|\cdots|\mathbf{A_r}|0|\cdots), (\mathbf{A_0}|\mathbf{A_{-1}}|\cdots|\mathbf{A_{-r}}|0|\cdots)), \\ \mathbf{B}_{N\times N} &= toeplitz((\mathbf{B_0}|\mathbf{B_1}|\cdots|\mathbf{B_r}|0|\cdots), (\mathbf{B_0}|\mathbf{B_{-1}}|\cdots|\mathbf{B_{-r}}|0|\cdots)), \\ \mathbf{w}_{N\times 1} &= [1\ 1\cdots 1]^T. \end{aligned}$$

Here,

$$\begin{aligned} \mathbf{v_{uk}}_{1\times m} &= [v_u(k,1)\ v_u(k,2)\cdots v_u(k,n)], \\ \mathbf{v_{xk}}_{1\times m} &= [v_x(k,1)\ v_x(k,2)\cdots v_x(k,n)], \\ \mathbf{v_{yk}}_{1\times m} &= [v_y(k,1)\ v_y(k,2)\cdots v_y(k,n)], \\ \mathbf{A_{km}}_{\times m} &= toeplitz((a_{k,0}\ a_{k,1}\cdots a_{k,r}0\cdots), ((a_{k,0}\ a_{k,-1}\cdots a_{k,-r}0\cdots)), \\ \mathbf{B_{km}}_{\times m} &= toeplitz((b_{k,0}\ b_{k,1}\cdots b_{k,r}0\cdots), ((b_{k,0}\ b_{k,-1}\cdots b_{k,-r}0\cdots)). \end{aligned}$$

The Toeplitz matrix, $toeplitz(\mathbf{a}, \mathbf{b})$, is defined as the matrix with $\mathbf{a}$ in the first row and $\mathbf{b}$ in the first column. The output vector $\mathbf{y}$ is confined within the N-dimensional hypercube because $-1 \leq y_k \leq +1$, $\forall k$. Thus, $\mathbf{y} \in \mathbf{D}^N = \{\mathbf{y} \in \mathbf{R}^N : -1 \leq y_k \leq 1; k = 1, 2, \cdots, N\}$.

If $A(i,j;k,l) = A(k,l;i,j)$ and $B(i,j;k,l) = B(k,l;i,j)$, the cloning templates are called symmetric templates. In this case, $\mathbf{A}$ and $\mathbf{B}$ are symmetric matrices and the stability of the network is guaranteed [16, 35]. Actually, the symmetry of $\mathbf{A}$ is a sufficient condition for stability. The network always produces stable outputs in the steady state under the constraint conditions $|v_x(i,j)(0)| \leq 1$ and $|v_u(i,j)| \leq 1$, $\forall i, j$ [24]. Furthermore, if $A(i,j;i,j) > 1/R_x$, then the saturated outputs are guaranteed to take binary values.

All the internal states $v_x(i,j)$,$\forall t \geq 0$ in the cellular neural networks are bounded. The maximum state value $v_{x,max}$ can be determined by [24]

$$\begin{aligned} v_{x,max} &= 1 + R_x|I_b| \\ &+ R_x \max_{\substack{1 \leq i \leq n, \\ 1 \leq j \leq m}} \left(\sum_{C(k,l) \in N_r(i,j)} (|A(i,j;k,l)| + |B(i,j;k,l)|) \right). \end{aligned} \tag{2.7}$$

The terms in the right-hand side of (2.7) are contributed from the initial value, bias signal, feedback, and feedforward interconnections, respectively. In order to perform the summation and integration in the processing element, the operating range of the state node voltages must be at least $-v_{x,max} \leq v_x(i,j) \leq v_{x.max}$.

2.2 Architecture of a Processing Element

Local interconnection and simple synaptic operators are very desirable features of the cellular neural networks for VLSI implementation. Hardware implementation of the CNN has been studied by many researchers and progressive results have been

reported [38, 39, 36, 35, 40, 41, 42, 37, 43, 44, 45]. However, most of the reported chip implementations were confined to a fixed application for each chip. Therefore, a fully-programmable chip is still to be developed. The network considered in this design is a continuous-time, rectangular-grid CNN with $r = 1$ with digitally-programmable synapse weights. To be fully-programmable, eighteen synapse weights described by the feedback and feedforward cloning templates are to be supported. In selected applications, the cloning templates can be simplified to contain only six synapse weights, i.e.,

$$\mathbf{T}_A = \begin{bmatrix} a_2 & a_1 & a_2 \\ a_1 & a_0 & a_1 \\ a_2 & a_1 & a_2 \end{bmatrix} \tag{2.8}$$

$$\mathbf{T}_B = \begin{bmatrix} b_2 & b_1 & b_2 \\ b_1 & b_0 & b_1 \\ b_2 & b_1 & b_2 \end{bmatrix} \tag{2.9}$$

In my design, $b_1 = b_2 = 0$ was used so that only four programmable synapse weights a_0, a_1, a_2 and b_0 are supported.

Figure 2.5 shows the block diagram of one processing element consisting of core cell, synapse weights and input/output units. Four synapse circuits receive signals from the external input, self-feedback, and outputs of neighboring elements and multiply them with pre-stored template values. The resulting signals are sent to the core cell to perform summation, integration and nonlinear transformation. The output result is stored in a data latch and sent to the data bus R_Data by enabling the selection signal E_Sel. In each operation, the initial state $v_x(0)$ needs to be initialized to a value between -1 and +1. By use of the control signals ϕ and $\bar{\phi}$, the initial state $v_x(0)$ could share the same terminal with the external input signal. During the initialization operation, control signal ϕ is low and the initial state can be placed on the capacitor C_x. At the same time, the outputs of the synapse circuits are

forced into the high-impedance state to avoid possible erroneous operation induced by the closed loop with the parasitic capacitance at the state node.

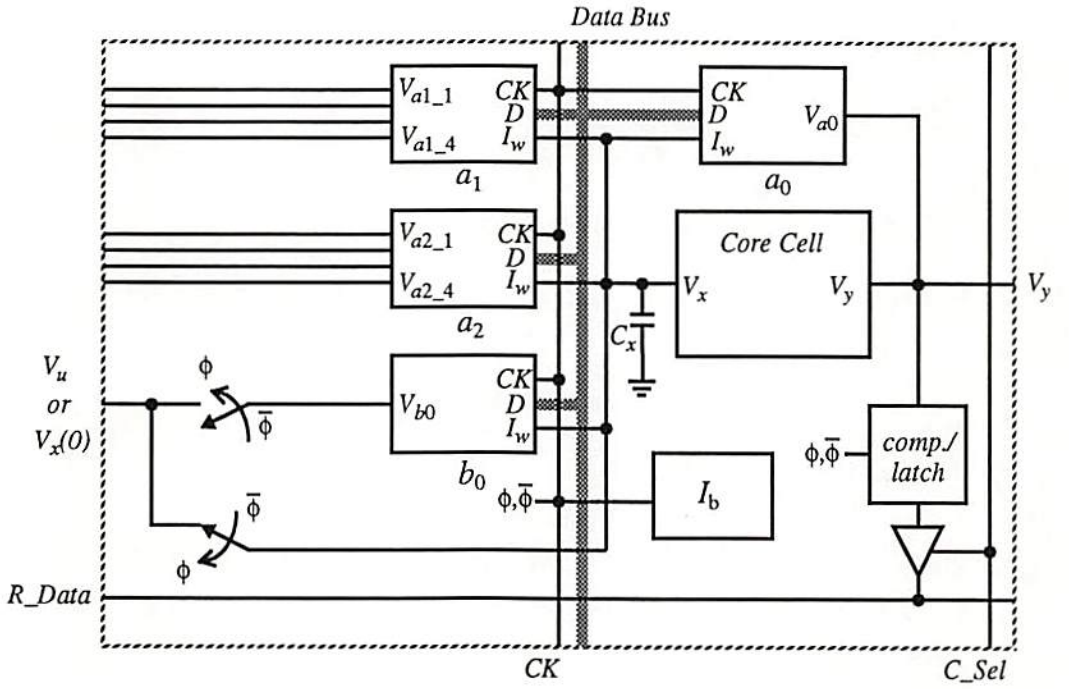


Figure 2.5: Block diagram of a single processing element.

An $n \times m$ rectangular-grid array processor can be constructed by using the processing element shown in Fig. 2.5 and appropriate interconnections with neighboring elements as shown in Fig. 2.6. Four 5-bit data registers are used to store the synapse weights a_0, a_1, a_2, and b_0. Those values can be transmitted to all the processing elements through the common control buses. In order to reduce the number of terminals for output signals, a multiplexing scheme can be used. Since a data bus *R_Data* is common to all the elements in a row, the element outputs can be read out column by column by activating the appropriate column selection signal *E_Sel*. The read operations can take place during the next network operation through the direct memory access (DMA) so that the network operation speed will not slow down in a

moderate-size array. Information on selected templates can be found in Appendix B.

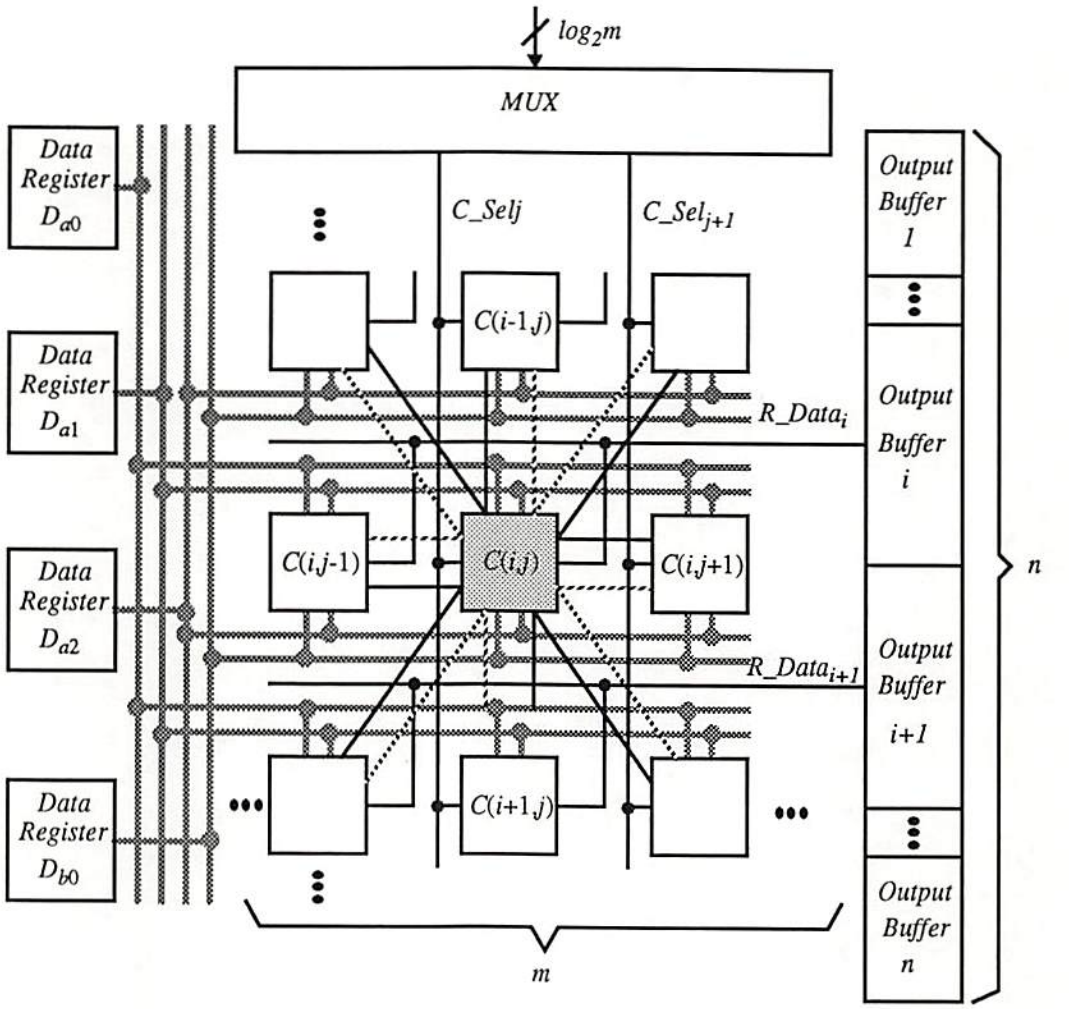


Figure 2.6: Architecture of an $n \times m$ processor array.

Chapter 3

Design of Paralleled Array Processors

To construct an intelligent microsystem, one important issue is to determine how to represent the signals. The representation of signal in a system could be the voltage, current, charge [46] or pulse-stream [47]. Most of the circuit designs are based on the voltage-mode technique because there exist many basic building circuit blocks for arithmetic operation by using the voltage to represent the signal [48, 49, 50, 51, 52]. However, quite a few implementations of neural networks by using the current-mode technique have been reported because the large number of currents can be summed at a single node [39, 36, 42]. Furthermore, power consumption is a very important design issue for an intelligent microsystem and the low-voltage, low-power operation is supported with the current-mode scheme in detailed circuit design. Therefore, the current-mode technique is used in my design of paralleled array processors.

In this chapter, the current-mode technique is used to design several VLSI circuit blocks for the paralleled array processors. A 5×5 processor-array chip was designed and fabricated in a 2-μm CMOS technology through MOSIS Service. Experimental results are provided to demonstrate the operation of the prototype chip.

3.1 Circuit Design of Basic Components

Figure 3.1 shows a detailed block diagram of a processing element based on the block diagram shown in Fig. 2.5. It consists of several basic building blocks including the current inverse circuit, the piecewise-linear circuit, the digitally-programmable synaptic weight circuit, voltage-to-current circuit, current comparator circuit, and bias current generation circuit. Detailed description of those circuits are in the following subsections.

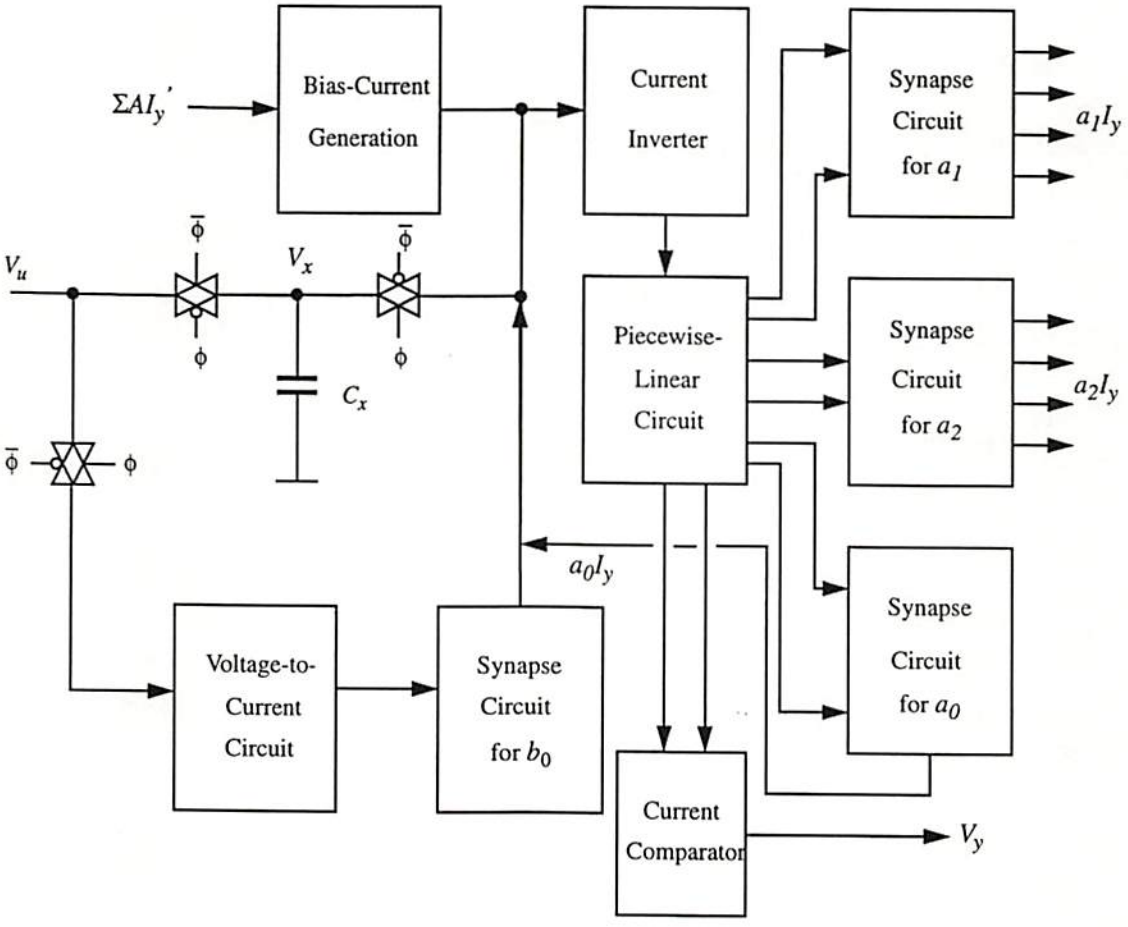


Figure 3.1: Detailed block diagram of a processing element using the current-mode technique.

3.1.1 Current Inverse Circuit

Figure 3.2 shows the circuit schematic diagram of the current inverse circuit [16, 35]. It reverses the direction of the input current and provides a constant input resistance whose value is independent of the input current. Linear I-V conversion is performed by transistors M_1 and M_2. Transistors M_1 and M_2 have the same device sizes and operate in the saturation region. If the currents flow through transistors M_1 and M_2 are I_1 and I_2, respectively, then $I_1 = I_{in} + I_2$. By use of current mirror operation, the currents flow through transistors M_3 and M_5 are also I_1 and $_2$, respectively, if the channel length modulation effect can be neglected. The output current at the output node is given by

$$I_{out} = I_2 - I_1 = -I_{in}. \tag{3.1}$$

Therefore, the circuit is a negative bidirectional current conveyer. To have a proper operation, the range of input I_{in} should be $|I_{in}| < \mu C_{OX}(W/2L)(V_B - 2V_{thn})^2$ [16, 35], where V_B is the gate voltage of M_2. The equivalent input resistance of the circuit can be expressed by [16, 35]

$$R_x = \frac{1}{\mu C_{OX}(W/L)(V_B - 2V_{thn})}. \tag{3.2}$$

Since the time constant of the core integrator is determined by the equivalent input resistance R_x and capacitance C_x, it is desirable for a processing element to have a controllable resistance R_x. This can be done by controlling the gate voltage V_B.

The transistor sizes of the current inverse circuit are listed in Table 3.1. Figure 3.3 shows the simulation results of the circuit using the SPICE-3f3 program [53] and the level-2 transistor model [54]. The input current is from -60 μA to 60 μA and the reference voltage V_B is 3.29 V. In the design, the supply voltages V_{DD} and V_{SS} are 5 V and 0 V, respectively. In order to compensate for the non-ideal effect of current

3.1.1 Current Inverse Circuit

Figure 3.2 shows the circuit schematic diagram of the current inverse circuit [16, 35]. It reverses the direction of the input current and provides a constant input resistance whose value is independent of the input current. Linear I-V conversion is performed by transistors M_1 and M_2. Transistors M_1 and M_2 have the same device sizes and operate in the saturation region. If the currents flow through transistors M_1 and M_2 are I_1 and I_2, respectively, then $I_1 = I_{in} + I_2$. By use of current mirror operation, the currents flow through transistors M_3 and M_5 are also I_1 and $_2$, respectively, if the channel length modulation effect can be neglected. The output current at the output node is given by

$$I_{out} = I_2 - I_1 = -I_{in}. \tag{3.1}$$

Therefore, the circuit is a negative bidirectional current conveyer. To have a proper operation, the range of input I_{in} should be $|I_{in}| < \mu C_{OX}(W/2L)(V_B - 2V_{thn})^2$ [16, 35], where V_B is the gate voltage of M_2. The equivalent input resistance of the circuit can be expressed by [16, 35]

$$R_x = \frac{1}{\mu C_{OX}(W/L)(V_B - 2V_{thn})}. \tag{3.2}$$

Since the time constant of the core integrator is determined by the equivalent input resistance R_x and capacitance C_x, it is desirable for a processing element to have a controllable resistance R_x. This can be done by controlling the gate voltage V_B.

The transistor sizes of the current inverse circuit are listed in Table 3.1. Figure 3.3 shows the simulation results of the circuit using the SPICE-3f3 program [53] and the level-2 transistor model [54]. The input current is from -60 μA to 60 μA and the reference voltage V_B is 3.29 V. In the design, the supply voltages V_{DD} and V_{SS} are 5 V and 0 V, respectively. In order to compensate for the non-ideal effect of current

Table 3.1: Transistor sizes of the current inverse circuit.

Transistor	W/L (μm / μm)
M_1	16/8
M_2	16/8
M_3	17/8
M_4	32/8
M_5	29/8

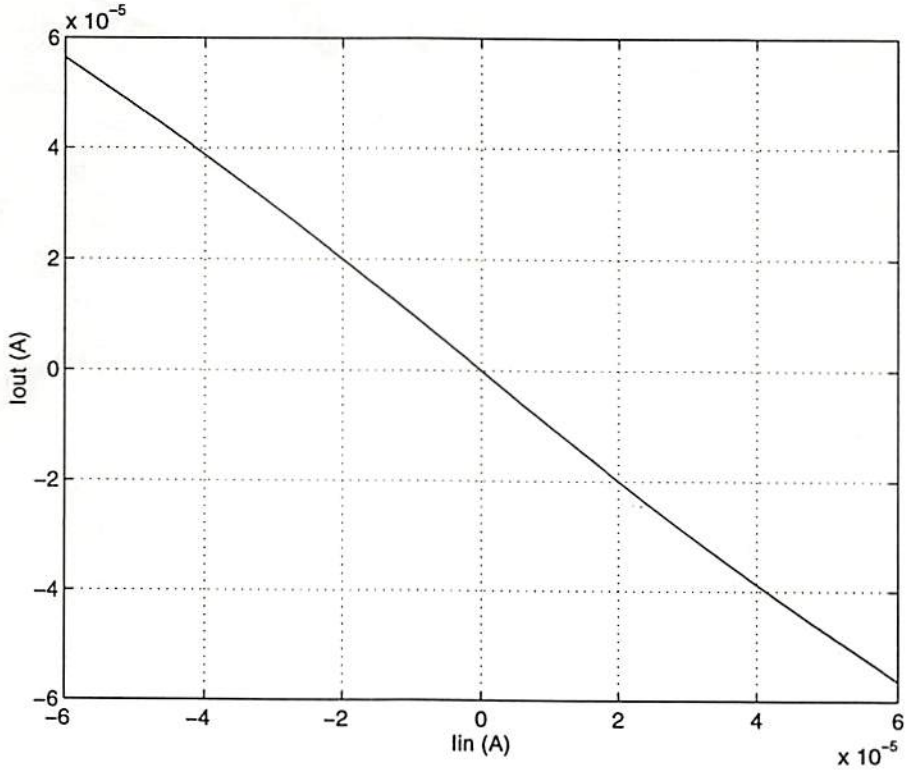


Figure 3.3: Simulation results of the current inverse circuit.

Table 3.1: Transistor sizes of the current inverse circuit.

Transistor	W/L (μm / μm)
M_1	16/8
M_2	16/8
M_3	17/8
M_4	32/8
M_5	29/8

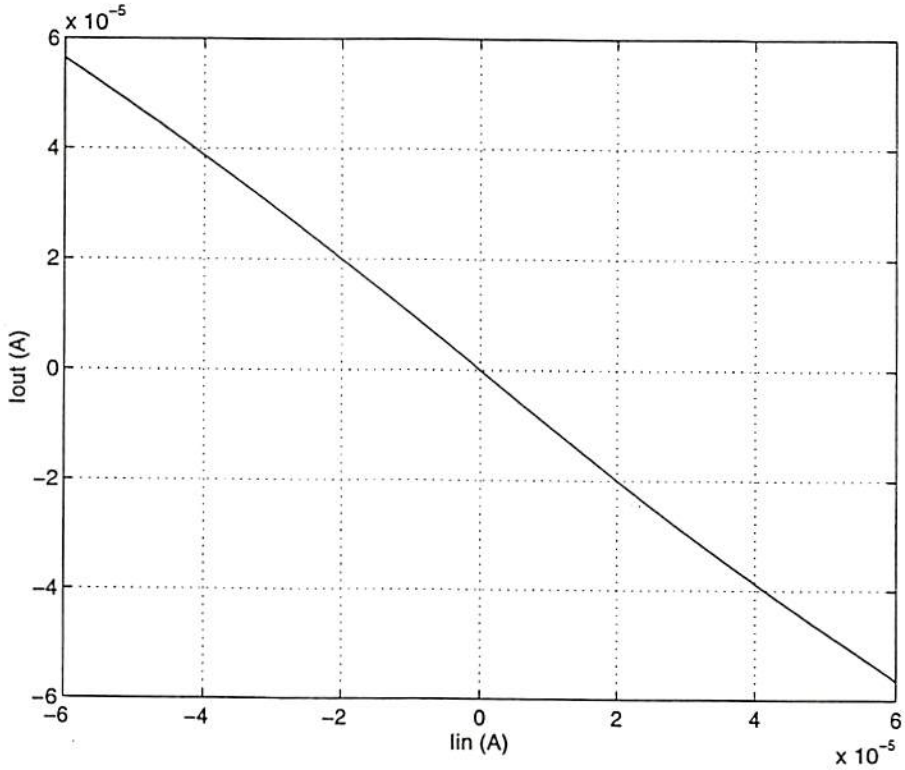


Figure 3.3: Simulation results of the current inverse circuit.

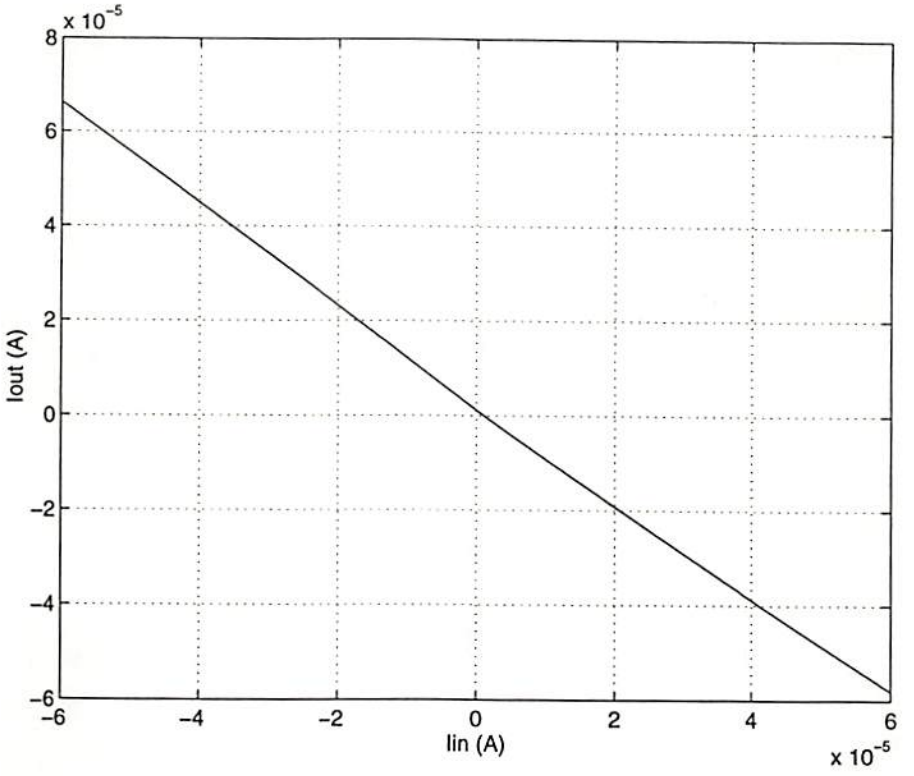


Figure 3.4: Simulation results of the current inverse circuit whose current mirrors have matched device sizes.

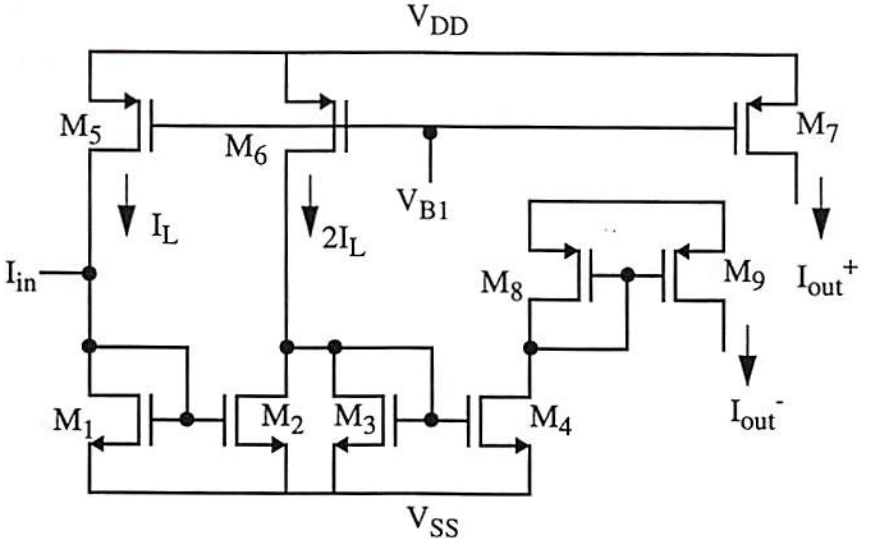


Figure 3.5: Circuit schematic diagram of the piecewise-linear circuit [16,35].

When the input current I_{in} is less than -I_L, no current flows through the transistors M_1 and M_2 so that transistor M_3 will sink all the current from transistor M_6. Thus, $I_{DS3} = I_{DS4} = I_{out}^- = 2I_L$ and the output current $I_{out}^+ - I_{out}^- = I_L - 2I_L = -I_L$. Notice that PMOS transistors M_8 and M_9 is a current mirror pair to change the current direction of I_{DS4}. When the input current I_{in} is greater than -I_L but less than I_L, $I_{DS1} = I_{DS2} = I_{in} + I_L$ and $I_{DS3} = I_{DS4} = 2I_L - (I_{in} + I_L) = I_L - I_{in}$. Therefore, the output current $I_{out}^+ - I_{out}^- = I_L - I_{DS4} = I_{in}$. When the input current I_{in} is greater than I_L, $I_{DS3} = I_{DS4} = 0$ and the output current $I_{out}^+ - I_{out}^- = I_L$.

The transistor sizes of the piecewise-linear circuit are listed in Table 3.2. Devices with a shorter channel length can be used. Figure 3.3 shows the SPICE3 simulation results of the circuit. In the design, the limiting current I_L is $10\mu A$ set by the reference voltage $V_{B1} = 3.7V$. Therefore, $10\mu A$ is used as the normalization factor for the network operation. Notice that if the device sizes of transistors M_2 and M_9 are $16\mu m/8\mu m$ and $32\mu m/8\mu m$, respectively, then the transfer curve of the circuit is asymmetry and doesn't pass the origin as shown in Fig. 3.7.

3.1.3 Digitally-Programmable Synaptic Weight Circuit

The schematic diagram of the digitally-programmable synaptic weight circuit is shown in Fig. 3.8 [16, 35]. This circuit is utilized to implement the cloning templates of the network. It consists of a double-MOS differential resistor [55] to construct the MSB bit and a binary-weighted current source array to construct the $(n-1)$ LSB bits. The circuit can perform four-quadrant multiplication. If the MSB bit is 1, then the positive input current I_{in}^+ flows through the upper branch while the negative input current I_{in}^- flows through the lower branch so that the polarity of the output current I_{out} is positive. If the MSB bit is 0, then I_{in}^+ and I_{in}^- are swapped so that the polarity of I_{out} is negative. The binary-weighted current source array is

Table 3.2: Transistor sizes of the piecewise-linear circuit.

Transistor	W/L (μm / μm)
M_1	16/8
M_2	17/8
M_3	16/8
M_4	16/8
M_5	32/8
M_6	64/8
M_7	32/8
M_8	32/8
M_9	27/8

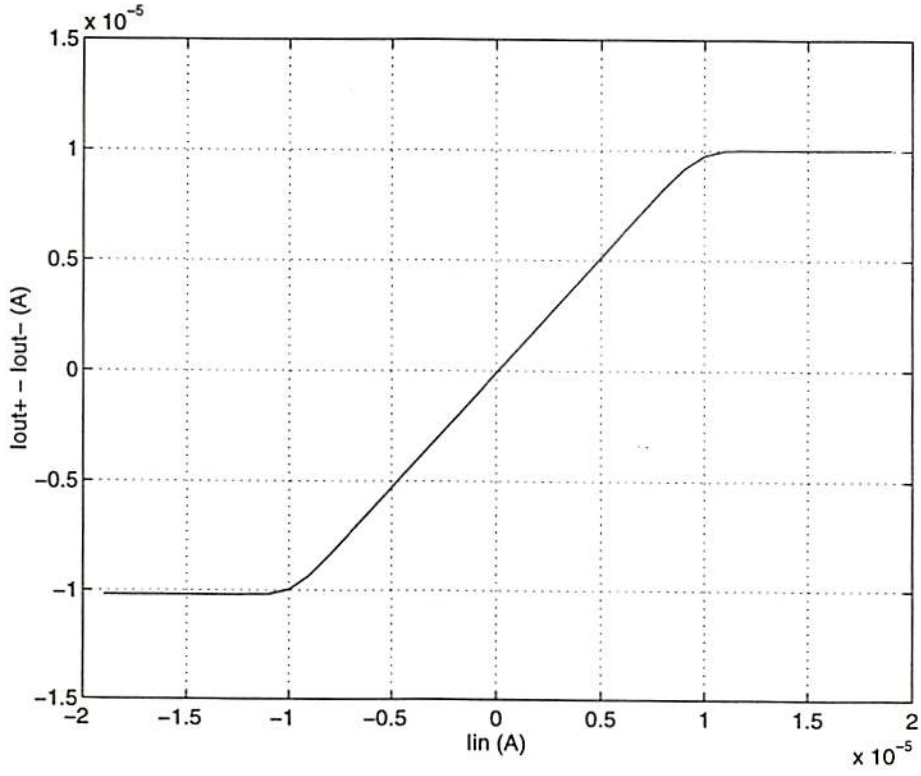


Figure 3.6: Simulation results of the piecewise-linear circuit.

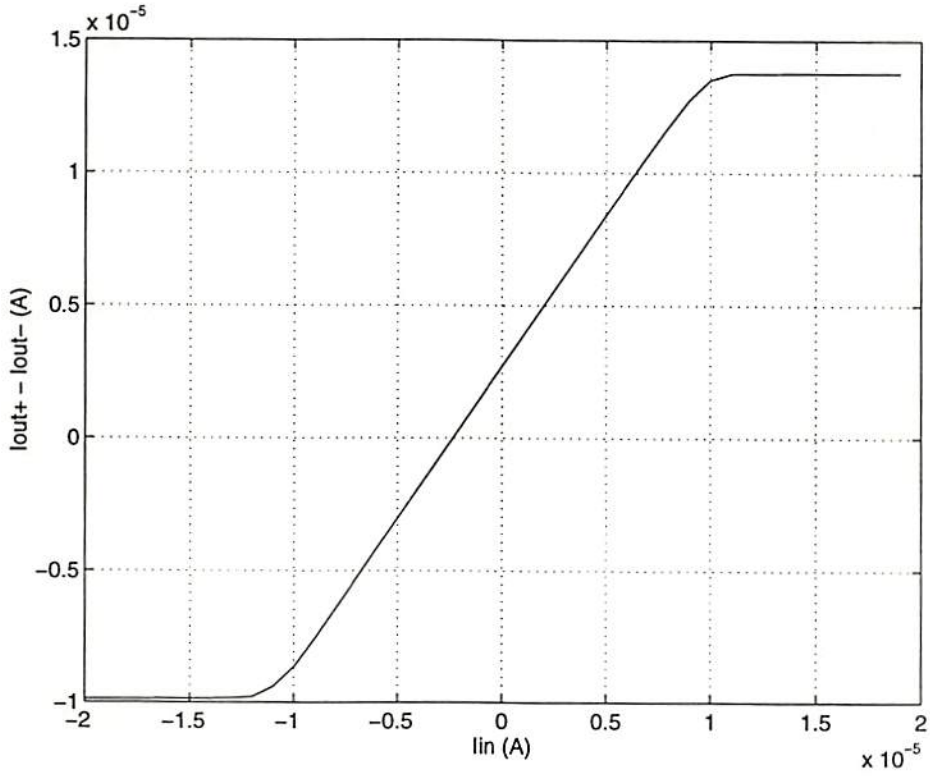


Figure 3.7: Simulation results of the piecewise-linear circuit with $(W/L)_2 = 16\mu m/8\mu m$ and $(W/L)_9 = 32\mu m/8\mu m$.

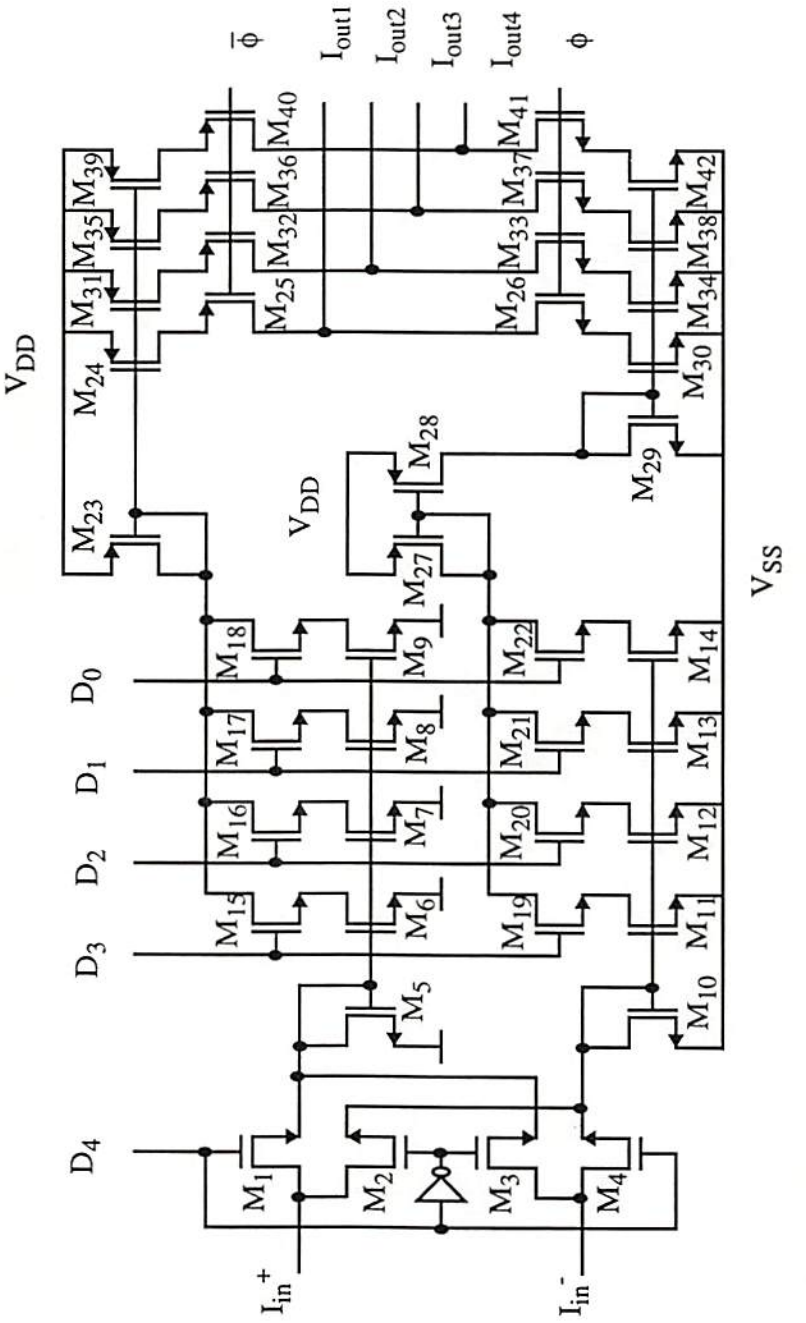


Figure 3.8: Circuit schematic of digitally-programmable synaptic weight circuit [16,35].

constructed by the transistors $M_5 - M_9$ and $M_{10} - M_{14}$. Whether $I_{DS6} - I_{DS9}$ and $I_{DS11} - I_{DS14}$ will contribute to the output current is controlled by the transmission gates $M_{15} - M_{22}$. In the design, $n = 5$ is chosen and the transistor sizes of the binary-weighted current source array are chosen such that $|I_{out}| \leq (4 - 2^{-2})|I_{in}|$, where $I_{in} = I_{in}^{+} - I_{in}^{-}$, in a step of $0.25I_{in}$. Since the current mirrors are used several times, it is very critical to match them as closely as possible through a careful simulation and layout design [35]. Besides, the cloning templates used in this design are given in (2.8) and (2.9) with $b_1 = b_2 = 0$. Hence, four digitally-programmable synaptic weight circuits are used to realize a_0, a_1, a_2, and b_0. The function of of the transistors $M_{27} - M_{42}$ is to generate four copies of output currents. Those transistors are needed to implement the synaptic weights a_1 and a_2. Since only one copy of output current is needed for a_0 and b_0, those transistors can be removed to simplify the circuit.

The transistor sizes of the digitally-programmable synaptic weight circuit are listed in Table 3.3. Figure 3.9 shows the SPICE3 simulation results of the piecewise-linear circuit combined with the synapse circuit. The synaptic weight values in the simulation are 2, 1, 0.5, and 0.25.

3.1.4 Conversion and Bias Current Generation Circuits

The external input voltage is converted to the current input by a voltage-to-current circuit. The circuit used in the design is a basic OTA circuit as shown in Fig. 3.10 [16, 35]. Since the input signal of the processing element is confined to ± 1 and the normalization current is $10\mu A$, the operating range of the circuit is designed around $\pm 10\mu A$ by carefully choosing the transistor sizes which are listed in Table 3.4. SPICE simulation results of the voltage-to-current circuit are shown in Fig. 3.11. In the

Table 3.3: Transistor sizes of the digitally-programmable synaptic weight circuit.

Transistor	W/L (μm / μm)
M_1, M_2, M_3, M_4	16/2
M_5, M_{10}	16/2
M_6, M_{11}	32/2
M_7, M_{12}	16/2
M_8, M_{13}	8/2
M_9, M_{14}	4/2
M_{15}, M_{16}, M_{17}, M_{18}	4/2
M_{19}, M_{20}, M_{21}, M_{22}	4/2
M_{23}	16/2
M_{24}, M_{31}, M_{35}, M_{39}	18/2
M_{25}, M_{32}, M_{36}, M_{40}	8/2
M_{26}, M_{33}, M_{37}, M_{41}	4/2
M_{27}, M_{28}	16/2
M_{29}	8/2
M_{30}, M_{34}, M_{38}, M_{42}	9/2

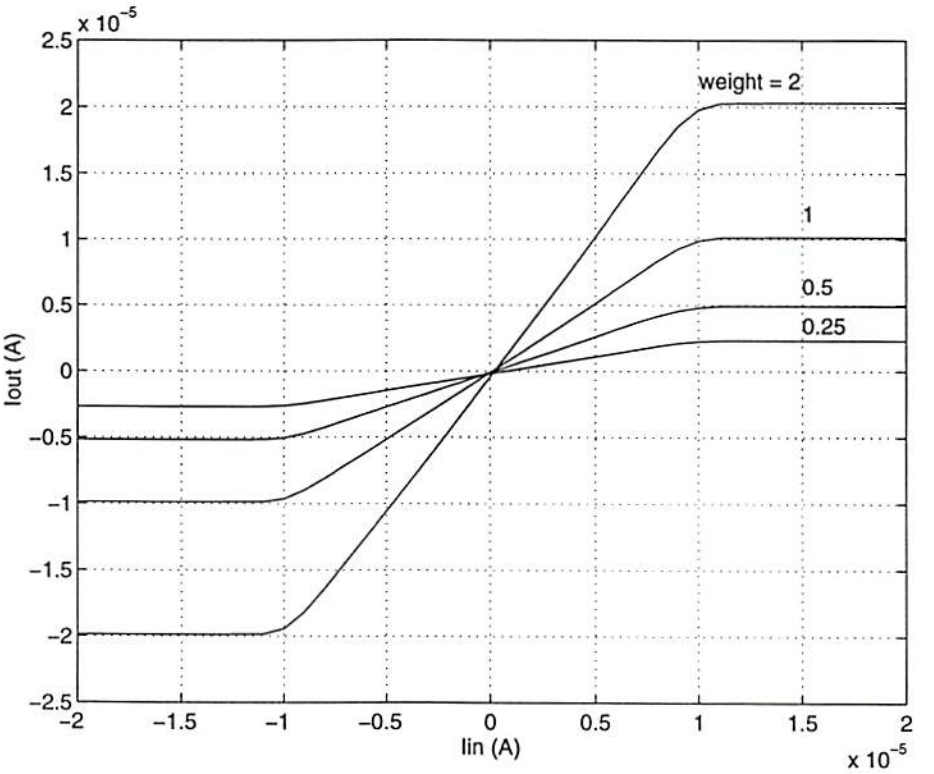


Figure 3.9: Simulation results of the digitally-programmable synaptic weight circuit.

simulation, $V_G = 2.5V$ and $V_{B2} = 2V$. To obtain input signals 1 and -1, the external input voltages should be 3.2 V and 1.8 V, respectively.

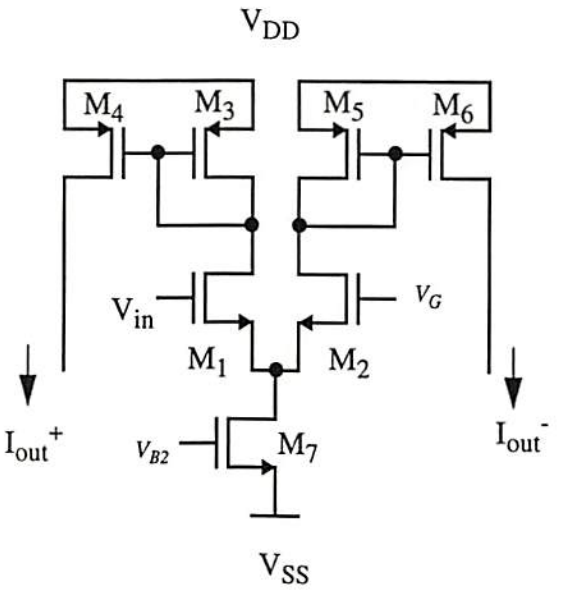


Figure 3.10: Circuit schematic diagram of the voltage-to-current circuit [16,35].

Table 3.4: Transistor sizes of the voltage-to-current circuit [16,35].

Transistor	W/L (μm / μm)
M_1	32/8
M_2	32/8
M_3	8/4
M_4	3/6
M_5	8/4
M_6	3/6
M_7	16/8

Figure 3.12 shows the schematic diagram of a simple current comparator which can be used to convert the output signal of the processing element from the current

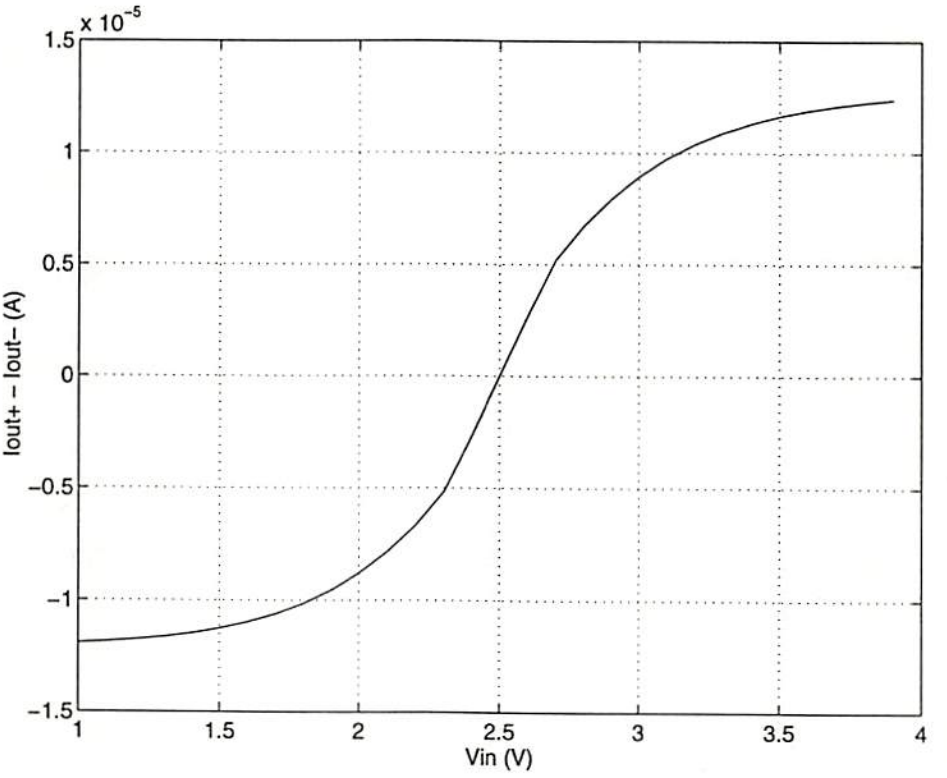


Figure 3.11: Simulation results of the voltage-to-current circuit.

to the voltage. It consists of a cascade of two inverters. The current I_y and reference current I_L are compared to produce an output voltage V_y. The transistor sizes of the current-to-voltage circuit are listed in Table 3.2. Figure 3.3 shows the SPICE3 simulation results of the circuit.

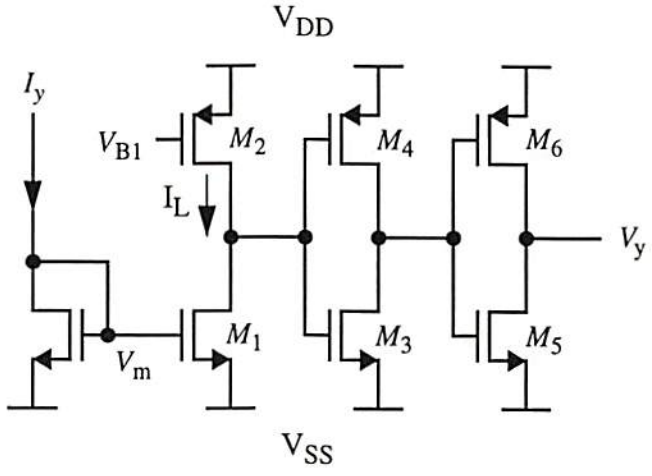


Figure 3.12: Circuit schematic diagram of the current-to-voltage circuit [16,35].

Table 3.5: Transistor sizes of the current-to-voltage circuit.

Transistor	W/L (μm / μm)
M_1	8/4
M_2	16/4
M_3	4/2
M_4	8/2
M_5	4/2
M_6	8/2

The bias current is needed to adjust the threshold value of the neuron. One pMOS and one nMOS transistors can be used to generate the bias current. Schematic diagram of the circuit is shown in Fig. 3.14. The bias current is provided by controlling

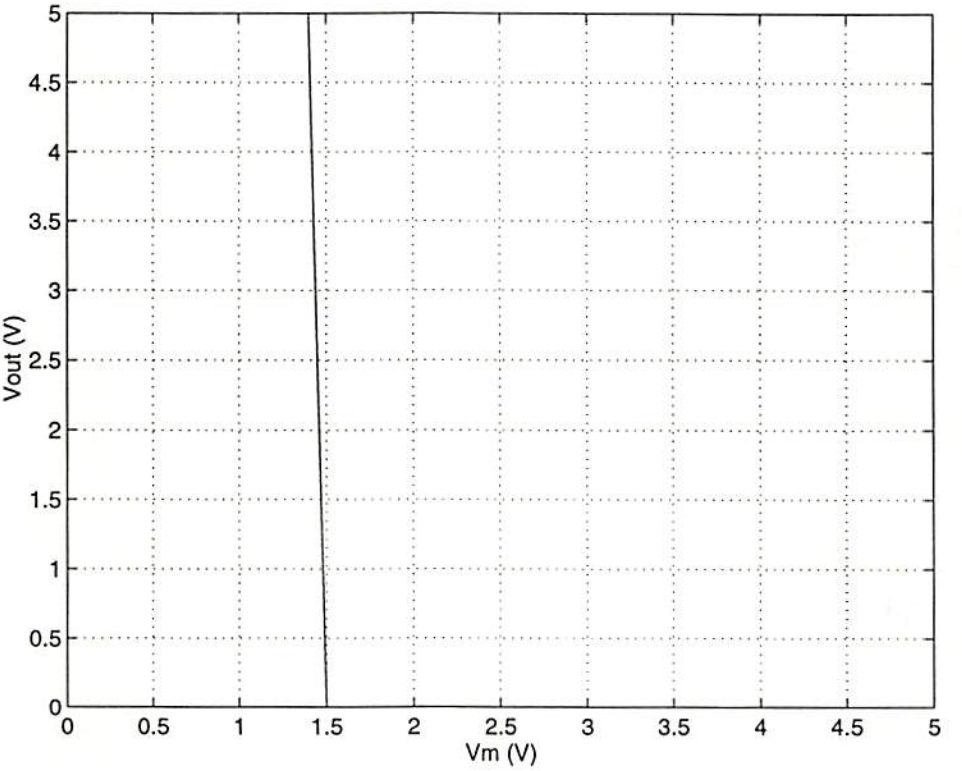


Figure 3.13: Simulation results of the current-to-voltage circuit.

the gate voltage V_I of the transistor M_1. The transistor sizes of the circuit are listed in Table 3.6. To obtain a bias value of -1, the control voltage V_I should be set to 1.7 V according to the SPICE simulation results shown in Fig. 3.15.

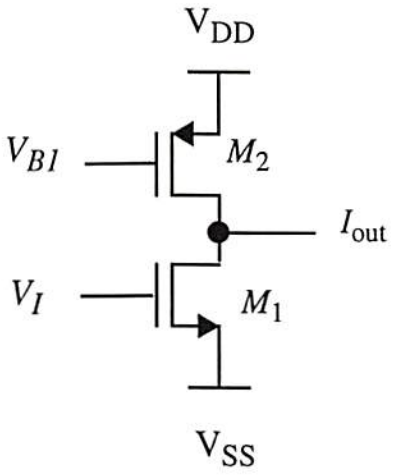


Figure 3.14: Circuit schematic diagram of the bias generation circuit [16,35].

Table 3.6: Transistor sizes of the bias generation circuit.

Transistor	W/L (μm / μm)
M_1	8/4
M_2	16/4

3.2 Measurement Results

The detailed circuit schematic diagram of a processing element is shown in Fig. 3.16 [16, 35] which is based on the basic circuit components discussed in the previous section and the block diagram of a processing element shown in Fig. 3.1. The physical layout of the processing element is shown in Fig. 3.17. It occupies an area of $470 \times 746\lambda^2$. The integration capacitor occupies an area of 20,000 λ^2 and the realized capacitance is about $10pF$ value in the given 2-μm CMOS technology. In fact, the

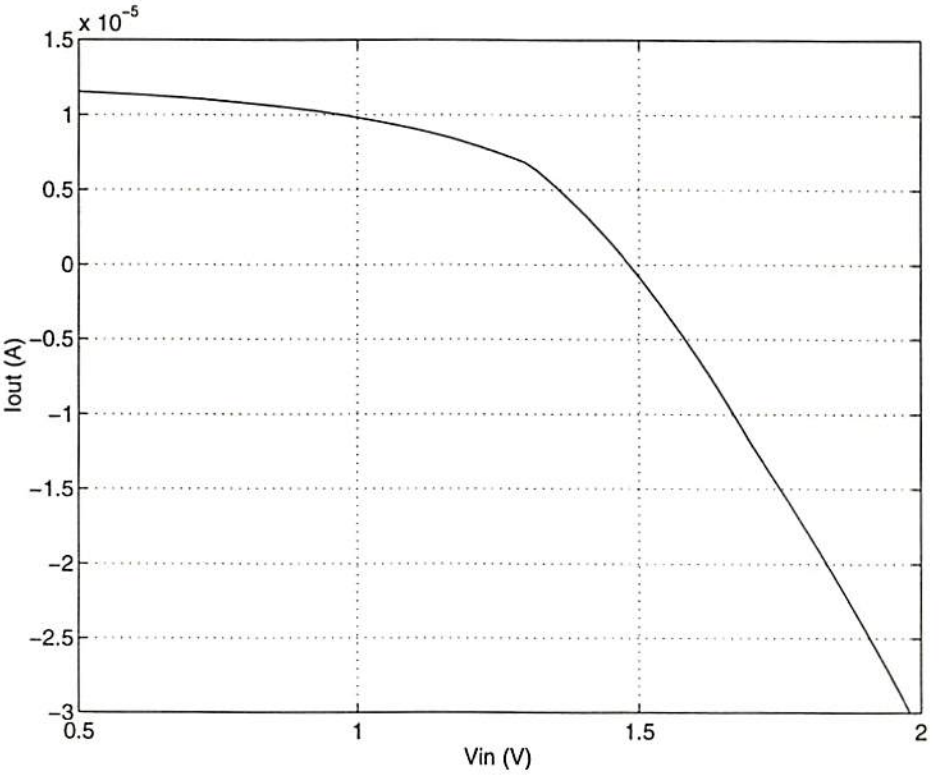


Figure 3.15: Simulation results of the bias generation circuit.

realized capacitor can be scaled down to about $1pF$ to reduce the chip area. In the scalable CMOS technology supported by the MOSIS Service at USC/Information Sciences Institute in Marina del Ray, CA [56, 57], one λ is equal to 1 μm for the 2-μm CMOS technology.

A prototype 5 $\times$ 5-processor array chip with digitally-programmable synaptic weight circuits was designed and fabricated in a 2-μm CMOS technology. The chip is in a 108-pin pin-grid-array package. In order to simplify testing, multiplexing scheme is not used to read out the output signals. The die photo of the prototype chip is shown in Fig. 3.18. The characteristics of the chip is summarized in Table 3.7. The chip consumes $24.7mW$.

Table 3.7: Characteristics of the prototype chip.

Fabrication technology	2.0-μm p-well CMOS
Number of transistors	11,250
Number of cells	5 x 5 = 25
Synapse weights storage	5-bit digital storage
Synapse weights connected/cell	10
Operating voltage	Single 5 V supply
Cell dimensions	276 μm x 746 μm
Cell array dimensions	1380 μm x 3730 μm
System dimensions	2689 μm x 3864 μm
Cell dissipation	989 μW
Cell array dissipation	24.7 mW

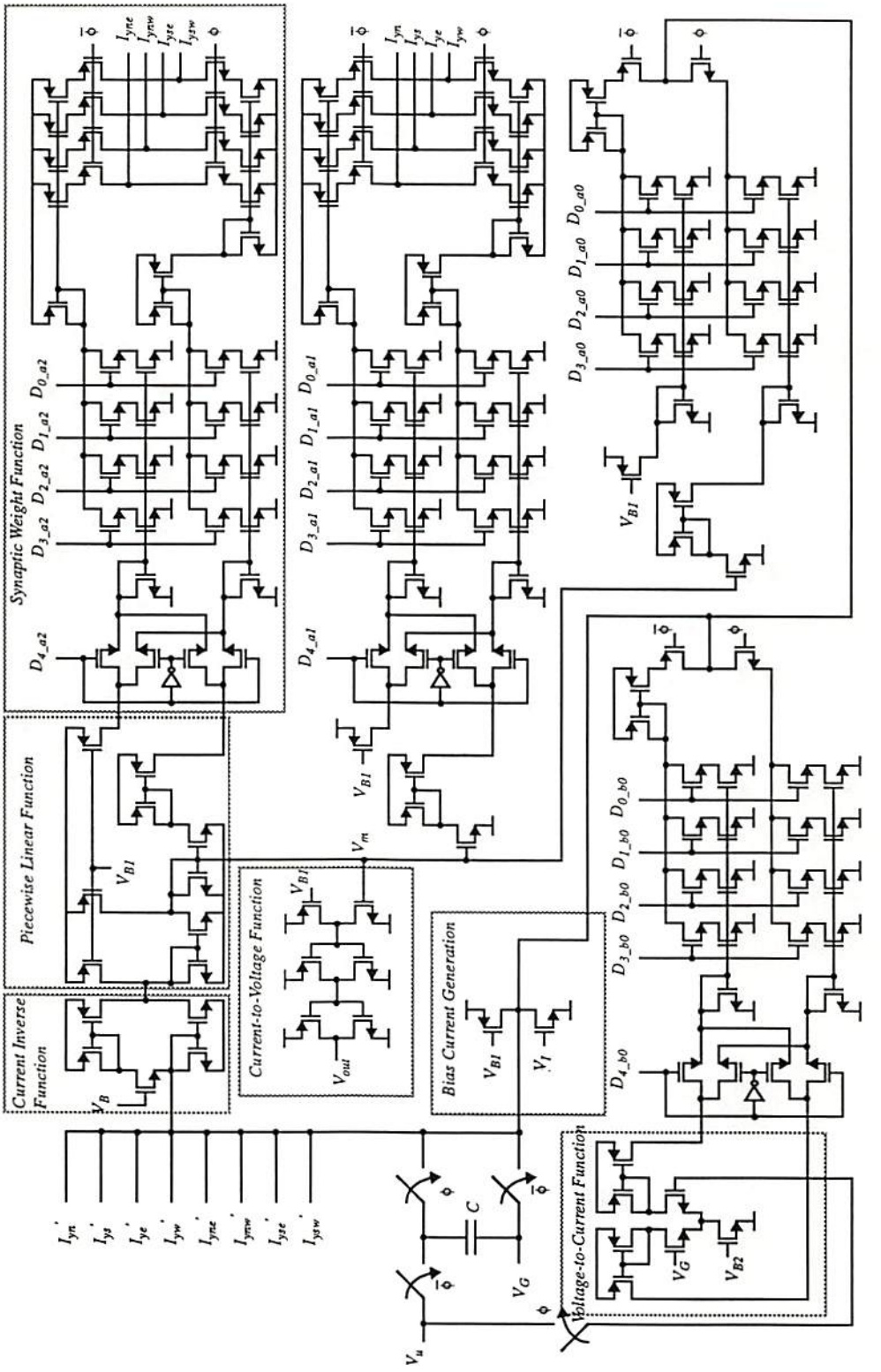


Figure 3.16: Circuit schematic diagram of a processing element [16,35].

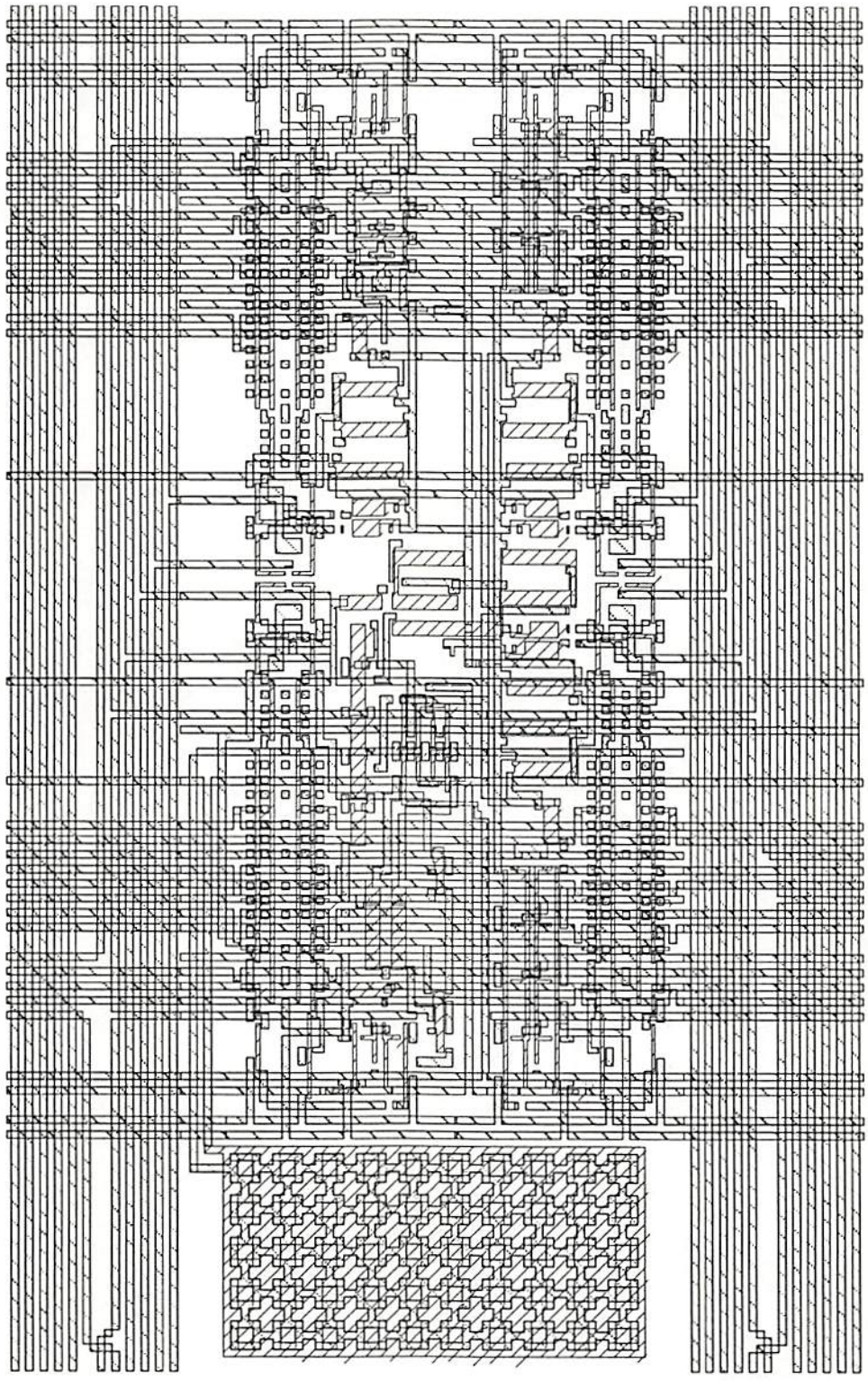


Figure 3.17: Physical layout of the processing element.

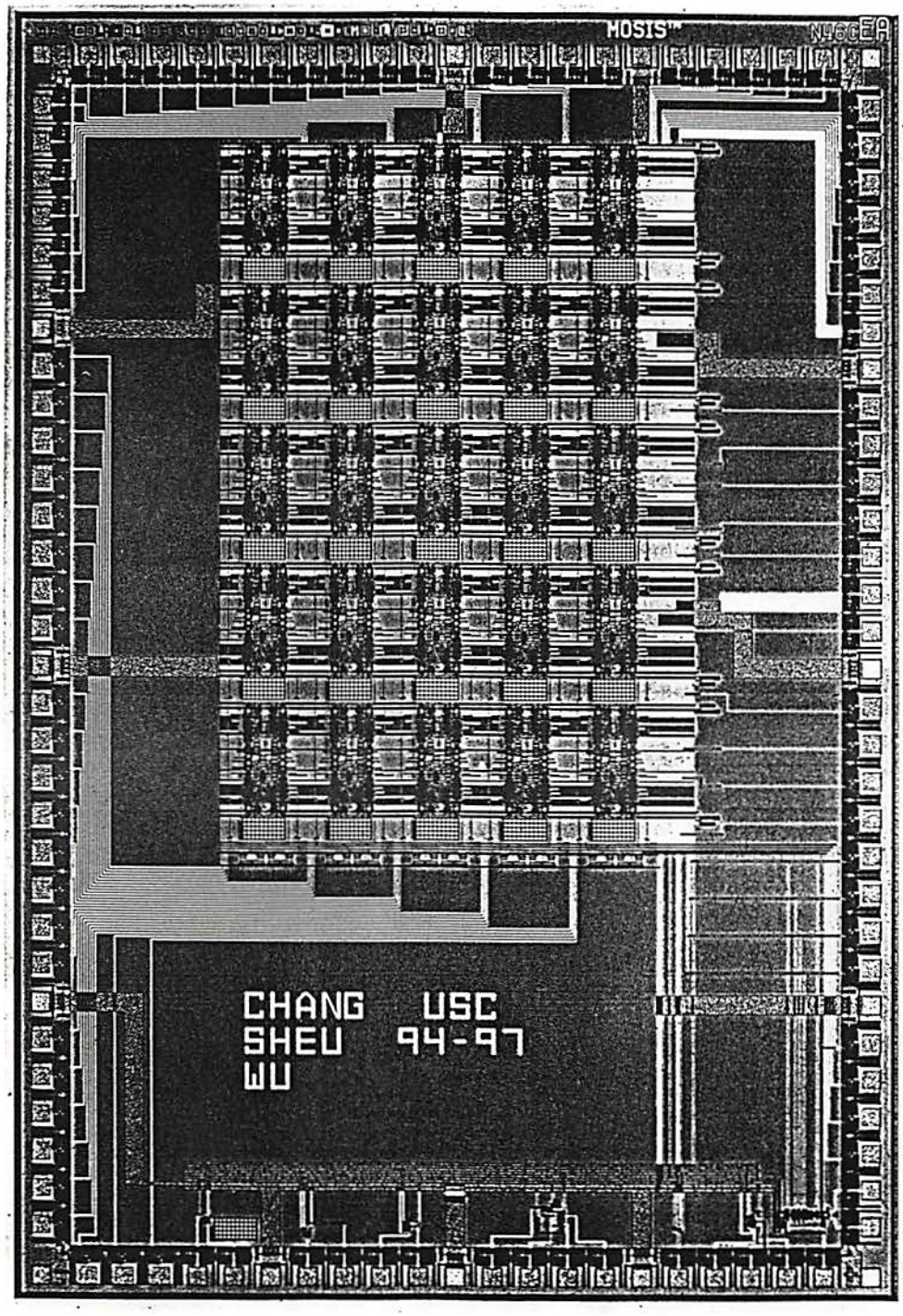


Figure 3.18: Die photo of the 5×5 CNN chip.

3.2.1 Basic Components

Several test structures are also included in the prototype chip. Figure 3.19 to Fig. 3.24 show the measurement results of the current inverse circuit, the piecewise-linear circuit, the digitally-programmable synaptic weight circuit, the voltage-to-current circuit, the current-to-voltage circuit, and the bias current generation circuit, respectively.

3.2.2 Circuit Board for the Prototype Chip

A circuit board is built to demonstrate the operation of this prototype chip. The block diagram of the circuit board is shown in Fig. 3.25. Hole filler and edge detection experiments were performed. The cloning templates used for hole filling operation are

$$\mathbf{T}_A = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \text{ and } \mathbf{T}_B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 3.75 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{3.3}$$

and the bias of the network is -1. The input pattern and the resulting output pattern are

$$\mathbf{V}_{u1} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}, \text{ and } \mathbf{V}_{y1} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}. \tag{3.4}$$

The output pattern corresponds to the LED display shown in Fig. 3.26. The input pattern of the hole filling operation can be changed to

$$\mathbf{V}_{u2} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{3.5}$$

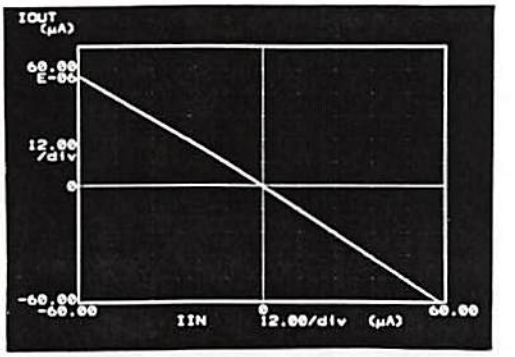


Figure 3.19: Measurement result of the current inverse circuit.

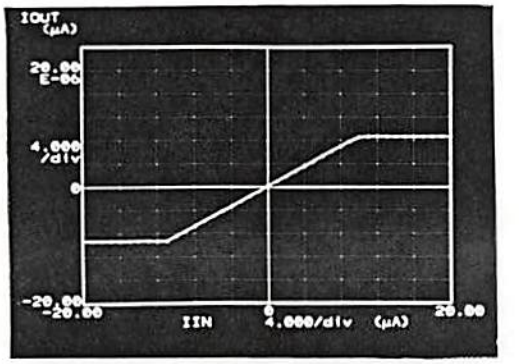


Figure 3.20: Measurement result of the piecewise-linear circuit.

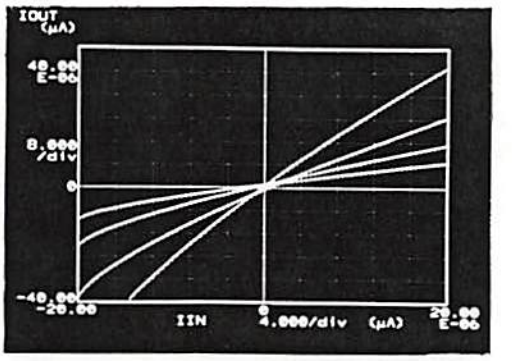


Figure 3.21: Measurement results of the synaptic weight circuit.

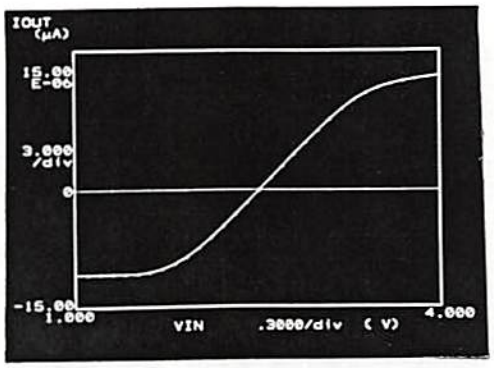


Figure 3.22: Measurement result of the voltage-to-current circuit.

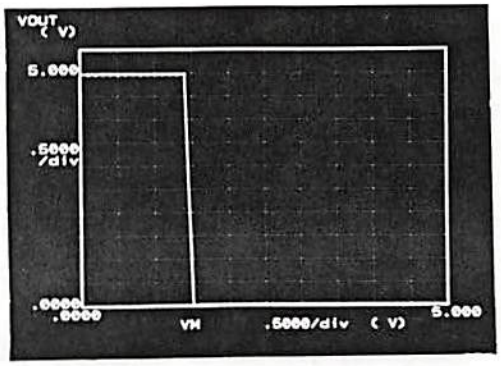


Figure 3.23: Measurement result of the current-to-voltage circuit.

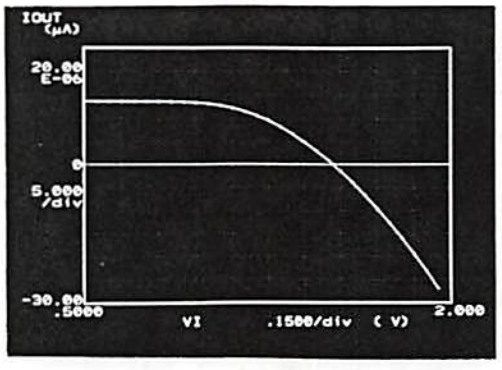


Figure 3.24: Measurement result of the bias current generation circuit.

Figure 3.25: Block diagram of the circuit board for the prototype chip.

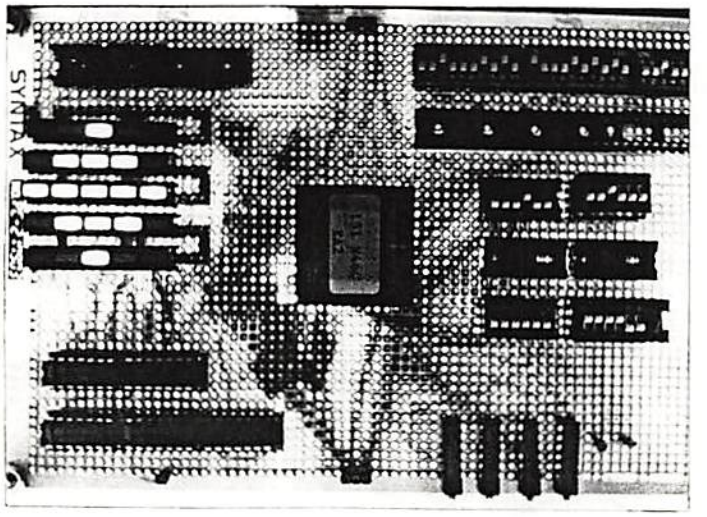


Figure 3.26: Hole filling operation of the prototype chip for input pattern V_{u1}.

The resulting output pattern is

$$\mathbf{V}_{y2} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.6}$$

which is shown in Fig. 3.27. Figures 3.28 and 3.29 show another two hole filling experiments with input and outputs patterns given as

$$\mathbf{V}_{u3} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \text{ and } \mathbf{V}_{y3} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{3.7}$$

and

$$\mathbf{V}_{u4} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \text{ and } \mathbf{V}_{y4} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{3.8}$$

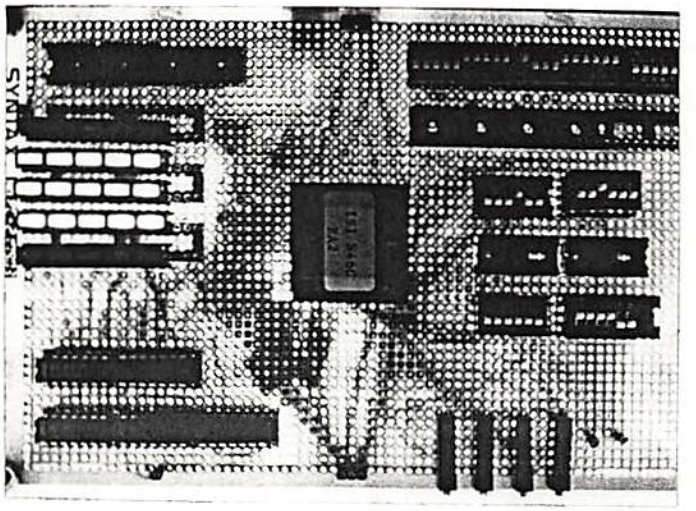


Figure 3.27: Hole filling operation of the prototype chip for input pattern V_{u2}.

Since the synaptic weight circuit of the prototype chip is digitally-programmable, the cloning template can be programmed to perform the edge detection operation. The cloning templates used for edge detection operation are

$$\mathbf{T}_A = \begin{bmatrix} 0 & -0.5 & 0 \\ -0.5 & 2 & -0.5 \\ 0 & -0.5 & 0 \end{bmatrix}, \text{ and } \mathbf{T}_B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{3.9}$$

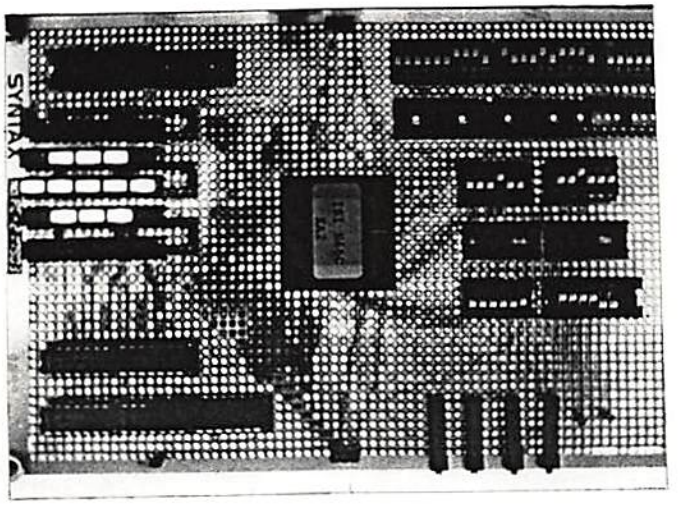


Figure 3.28: Hole filling operation of the prototype chip for input pattern V_{u3}.

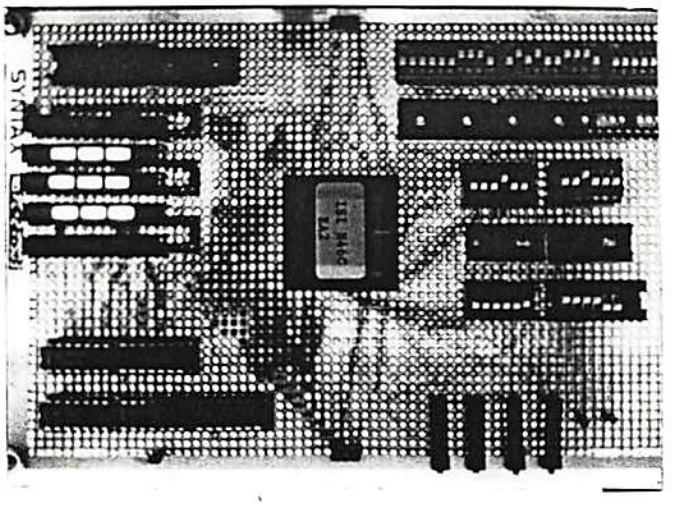


Figure 3.29: Hole filling operation of the prototype chip for input pattern V_{u4}.

and the bias of the network is -1.35. The input pattern and the resulting output patterns are

$$\mathbf{V}_u = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}, \text{ and } \mathbf{V}_y = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}. \tag{3.10}$$

The output pattern corresponds to the LED display shown in Fig. 3.30.

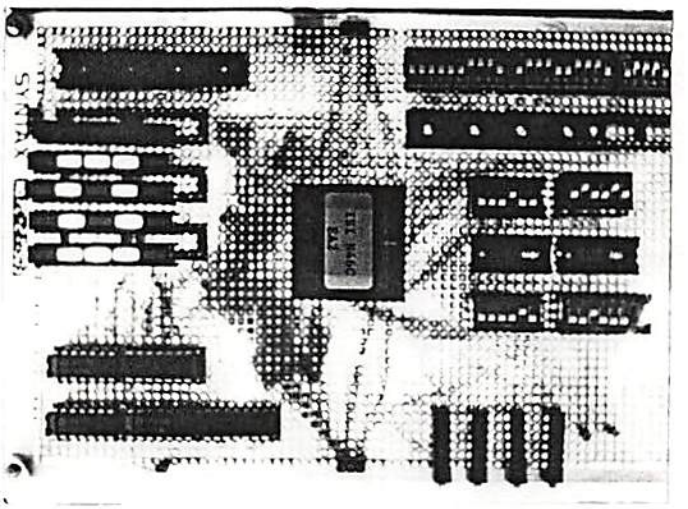


Figure 3.30: Edge Detection operation of the prototype chip.

3.3 Limitation on Higher Accuracy

There are several satisfactory measurement results obtained from the prototype chip. However, refinement for higher accuracy is still required. On problem is that the initial voltage is difficult to be programmed into the prototype chip. Since the initial voltage shares the same terminal with the external input voltage and only one clock

signal ϕ is used to control the switching, it is difficult to control the timing so that the network operation might arrive the steady-state before the actual external input voltage takes into operation. In addition, the operating range of the initial voltage is $\pm 0.25V$ which is very limited. It is desirable to enlarge the dynamic range. One way to alleviate the problem is to increase the normalization current. For example, the normalization current could be increased from $10\mu A$ to $40\mu A$ so that the dynamic range is increased to $\pm 0.5V$. However, the penalty is increasing the chip area and power consumption. More complex clocking scheme using two clock signals ϕ_1 and ϕ_2 could also be used to help the initialization operation. Clock signals ϕ_1 and ϕ_2 are non-overlapping clocks as shown in Fig. 3.31. The time t_{change} is determined by the speed of changing the terminal voltage from the initial voltage value to the input voltage value. If the analog multiplexer, CD4053B, is used, the propagation delay time from selection-in to signal-out is typically 300 ns.

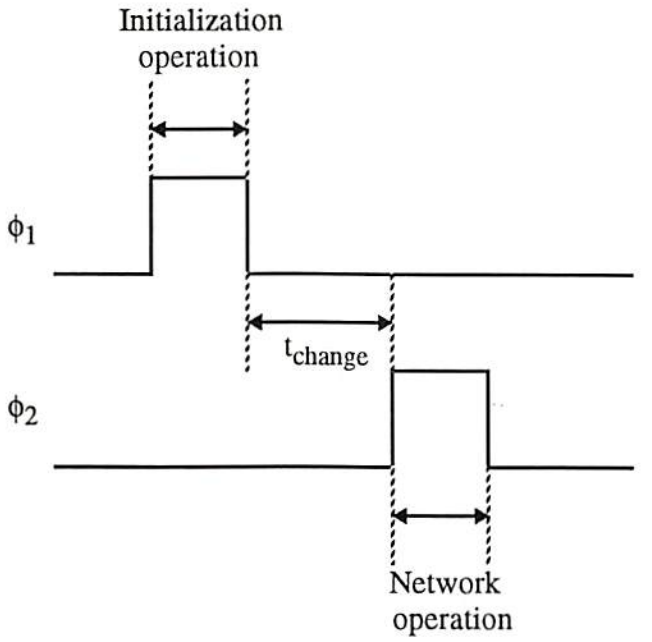


Figure 3.31: Initialization operation using two clock signals ϕ_1 and ϕ_2.

Another problem is matchness of the current mirrors. In addition to the careful layout design, the cascode current or the Wilson current source [58] could be used to improve the matchness and the linearity of the circuits.

Chapter 4

2-Neuron Network Case with Hardware Annealing

Optimization is an important subject in solving many scientific and engineering problems such as the traveling salesman problem (TSP) [59, 60, 61, 62], placement and routing of VLSI layout design [63, 64], and graph partition [65, 17]. Simulated annealing [66, 67, 68, 69] and Boltzmann machine [70, 71] are two of the popular approaches which are applicable to the combinatorial optimization problems. However, software execution of those algorithms requires a very large computation time and therefore consumes a lot of electric power because of the data flow of traditional digital computers. Recent advances in microelectronic technologies make possible the design of compact electronic neural network processors. A globally-interconnected Hopfield neural networks with hardware annealing capability [16, 72, 73] can be utilized in many difficult optimization problems such as TSP and analog-to-digital decision making. It is a paralleled, electronic version of the deterministic mean-field annealing method [74, 75] directly incorporated with the recurrent neural networks. Many optimization problems can be mapped onto a 2-dimensional grid array. The silicon retina chip from Mead's group is a very good example [28]. Paralleled array processors based on the cellular neural networks is quite versatile in solving many optimization problems.

where $\mathbf{M} = \mathbf{A} - (1/R_x)\mathbf{I}$ and $\mathbf{b} = \mathbf{Bu} + I_b\mathbf{w}$. If the matrix $\mathbf{A}$ is symmetric, then the stability of the network is guaranteed [35, 24]. Under the constraint conditions $|v_x(i,j)(0)| \leq 1$ and $|v_u(i,j)| \leq 1$, $\forall i,j$ [24, 33], the network with a symmetric $\mathbf{A}$ always produces a stable steady-state output at which the energy function (4.2) is locally minimized. Moreover, if $A(i,j;i,j) > 1/R_x$, then the saturated binary outputs are guaranteed.

The generalized energy function associated with the network possesses many local-minimum points which could trap the network into unwanted solutions when the network is used for optimization applications. To understand the local-minimum problem in a network, a simple two-neuron network as shown in Fig. 4.1 is considered.

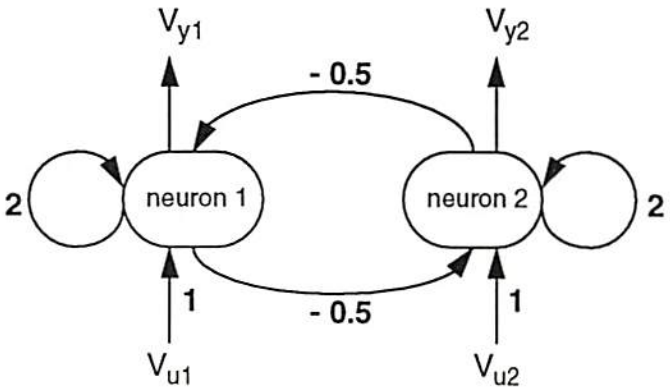


Figure 4.1: Block diagram of a 2-neuron CNN with $B_{11} = B_{22} = 1$ and $B_{12} = B_{21} = 0$.

Assume that the piecewise-linear transfer function is used, the integration capacitance C_x is normalized to a unity, the feedback weights are symmetric such that $A_{1,1} = A_{2,2} = A_0 > T_x = 1/R_x$, $A_{1,2} = A_{2,1} = A_1$, the feedforward weights

$B_{1,1} = B_{2,2} = 1$, and the network bias $I_b = 0$. Then, the generalized energy function of (4.2) can be simplified as [35],

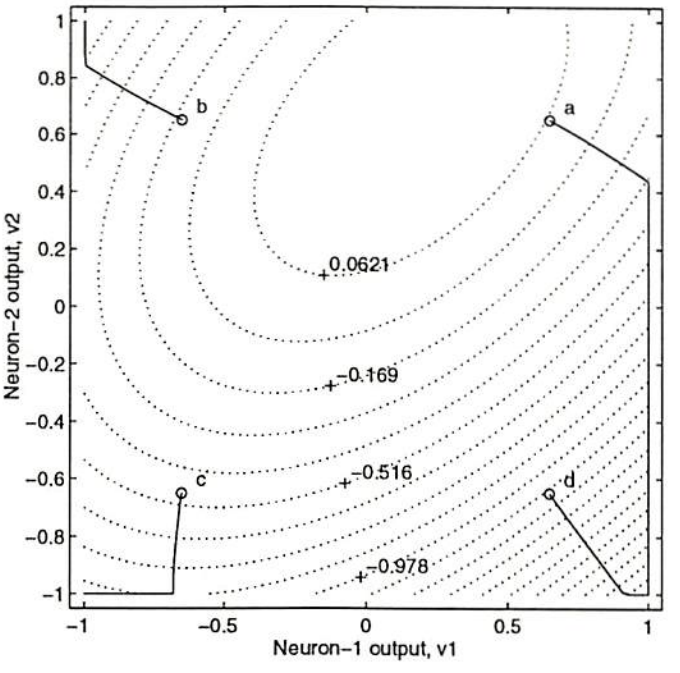
$$E = -\frac{1}{2}\begin{bmatrix} v_{y1} \\ v_{y2} \end{bmatrix}^T \begin{bmatrix} A_0 - T_x & A_1 \\ A_1 & A_0 - T_x \end{bmatrix} \begin{bmatrix} v_{y1} \\ v_{y2} \end{bmatrix} - \begin{bmatrix} v_{y1} \\ v_{y2} \end{bmatrix}^T \begin{bmatrix} v_{u1} \\ v_{u2} \end{bmatrix} \quad (4.3)$$

where $-1 \leq v_{y1}, v_{y2} \leq +1$. After solving the equation $\mathbf{Mx} = \lambda\mathbf{x}$, the eigenvectors and eigenvalues of $\mathbf{M}$ can be obtained as $\mathbf{x_1} = [1\ -1]^T$, $\mathbf{x_2} = [1\ 1]^T$, and $\lambda_1 = A_0 - A_1 - T_x$, $\lambda_2 = A_0 + A_1 - T_x$, respectively. If the eigenvalues λ's are positive and the bias is quite small, then the minima occur at all corners.

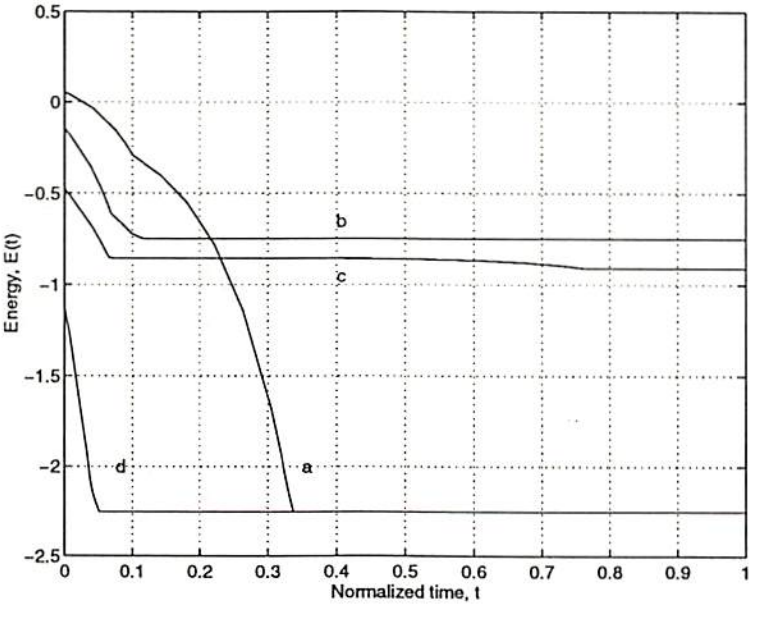
The lowest energy function value corresponds to the global-minimum while the others correspond to local minima. A Matlab program [35] was written to simulate the 2-neuron network with $A_0 = 2$ and $A_1 = -0.5$. The contour plots of the energy function and trajectories of output values with $[v_{u1}\ v_{u2}] = [0.2\ -0.6]$ is shown in Fig. 4.2(a). The global-minimum point is located at [1 -1] while the points [-1 -1] and [-1 1] are the local minima. The network responses located at initial points a, b, c, and d follow the individual trajectories when the network is allowed to evolve. The initial values at location a, b, c, and d are [0.7 0.7], [-0.7 0.7], [-0.7 -0.7], and [0.7 -0.7], respectively. The steady-state outputs for initial points a and d are both $[v_{y1}\ v_{y2}] = [1\ -1]$ which corresponds to the global-minimum. The final results for b and c are at the local minima. Figure 4.2(b) shows the plots of the energy function versus the network evolution time for the four cases. Notice that the energy value in the steady-state has no direct correspondence with the initial condition.

4.2 Review on Annealed Networks

The hardware annealing method can help the network escape from the local minima and quickly search for the global optimal solution. It is performed by controlling the



(a)



(b)

Figure 4.2: Multiple minima in concave energy function of two-neuron network. (a) Energy contour and trajectories of outputs for several initial conditions. (b) Corresponding energy functions $E(t)$ during network evolution.

gain of the neuron, which is assumed to be the same for all neurons in the network. After the state is initialized to $x = x(0)$, the initial gain at time t = 0 can be set to a very small, positive value such that $0 \leq g(0) \ll 1$. Therefore, the network can be linearized. It then increases continuously for $0 < t \leq T_a$ to the nominal gain of 1. The maximum gain $g_{max} = 1$ is maintained for $T_a < t \leq T$, during which the network is stabilized. When the hardware annealing is applied to the network by increasing the neuron gain $g(t)$, the transfer function can be described by $v_y(t) = f(g(t)v_x(t))$. Notice that the saturation level is still y=+1 or -1 and mainly the slope of $f(x)$ around $x = 0$ varies.

The gain of the processing element is controlled by a monotonically increasing function $g(t)$ such that

$$v_y = f(gv_x) = \begin{cases} +1, & \text{if } v_x > +1/g \\ gv_x, & \text{if } -1/g \leq v_x \leq +1/g \\ -1, & \text{if } v_x < -1/g \end{cases} \tag{4.4}$$

where $f(\cdot)$ is the piecewise-linear function.

In this case, the energy function can be written as [35]

$$E = -\frac{1}{2}\mathbf{y}^T\mathbf{M}_g\mathbf{y} - \mathbf{y}^T\mathbf{b}. \tag{4.5}$$

Since $\mathbf{M}_g$ is a real symmetric matrix, it can be diagonalized as $\mathbf{M}_g = \mathbf{A} - (T_x/g)\mathbf{I} = \mathbf{Q}\mathbf{\Lambda}_g\mathbf{Q}^T$, where $\mathbf{\Lambda}_g$ is the diagonal matrix of eigenvalues λ_k, $k = 1, 2, \cdots, N$, and $\mathbf{Q}$ is an $N \times N$ matrix whose columns are made of orthonormal set of eigenvectors $\mathbf{e}_k$'s. In an annealed network, the components of $\mathbf{\Lambda}_g$ are time-varying. Since $\mathbf{M}_g$ can be re-written as

$$\begin{aligned} \mathbf{M}_g &= \mathbf{A} - T_x\mathbf{I} - ((1-g)T_x/g)\mathbf{I} \\ &= \mathbf{M} - ((1-g)T_x/g)\mathbf{I}, \end{aligned} \tag{4.6}$$

the relationship between eigenvalues of the annealed and unannealed networks can be expressed as

$$\lambda_k = \lambda'_k - \frac{(1-g)T_x}{g}, \ \mathrm{k} = 1, 2 \cdots, N \tag{4.7}$$

where λ'_k's are the eigenvalues of $\mathbf{M}$ for the unannealed network. During hardware annealing operation, the eigenvalues λ_k's are varied from all negative initial values to the final values λ'_k's by increasing the neuron gain g. If the two-neuron network with $A_0 = 2, A_1 = -0.5, B_0 = 1, B_1 = 0$, and $T_x = 1$, the eigenvalues of $\mathbf{M}$ for the unannealed network are $\lambda'_1 = A_0 - A_1 - T_x = 1.5$ and $\lambda'_2 = A_0 + A_1 - T_x = 0.5$. The eigenvalues of $\mathbf{M}_g$ for the annealed network are $\lambda_1 = 1.5 - (1-g)/g$ and $\lambda_2 = 0.5 - (1-g)/g$. If the neuron gain g is increased from 0.1 to 1, λ_1 is varied from -7.5 to 1.5 and λ_2 is varied from -8.5 to 0.5. In the mean time, the generalized energy function (4.5) which is initially a convex function of $\mathbf{y}$, is gradually transformed into a concave function. Notice that the initial neuron gain g_0 must be chosen such that $\lambda_k(g_0)$ is less than zero for all k.

Consider the equilibrium $\mathbf{y} = \mathbf{y}_0$ at which the gradient $\bigtriangledown_y E = \partial E / \partial \mathbf{y}$ vanishes, and the steady-state output $\mathbf{y} = \mathbf{y}^*$ at which the energy E is globally minimized. The equilibrium $\mathbf{y}_0$ is not necessarily confined within the unit hypercube $\mathbf{D}^N$ where $\mathbf{D}^N = \{\mathbf{y} \in \mathbf{R}^N | \ -1 \leq y_k \leq +1, \ k = 1, 2, \cdots, N\}$, but can span a whole N-dimensional space, i.e., $\mathbf{y}_0 \in \mathbf{R}^N$. The saturated binary outputs are guaranteed in the steady-state for a shift-invariant, symmetric network with $A(i,j;i,j) > T_x, \forall i, j$. The minimal energy point should locate at a vertex or a multiple of vertices of $\mathbf{D}^N$.

If λ_k's are numbered in a decreasing order such that $\lambda^1_1 \geq \lambda^1_2 \geq \cdots \geq \lambda^1_N$, then the condition $A(i,j;i,j) > T_x, \forall i, j$, indicates that $\sum_{k=1}^{N} \lambda^1_k > 0$. Notice that there

may have repeated eigenvalues and the eigenvalues may be negative. Let the output $\mathbf{y}$ be expressed as a linear combination of $\mathbf{e}_k$'s, i.e.,

$$\mathbf{y} = \beta_1 \mathbf{e}_1 + \cdots + \beta_N \mathbf{e}_N, \tag{4.8}$$

where β_k, $1 \leq k \leq N$, are scalar constants. Thus,

$$\mathbf{Q}^T \mathbf{y} = \sum_k \mathbf{Q}^T (\beta_k \mathbf{e}_k) = [\beta_1 \ \ \beta_2 \ \ \cdots \ \ \beta_N]^T \equiv \beta \tag{4.9}$$

and (4.5) can be re-formulated as

$$\begin{aligned} E &= -\frac{1}{2}(\mathbf{Q}^T \mathbf{y})^T \mathbf{\Lambda} (\mathbf{Q}^T \mathbf{y}) - \mathbf{y}^T \mathbf{b} = -\frac{1}{2}\beta^T \mathbf{\Lambda} \beta - \sum_{k=1}^{N} \beta_k \mathbf{e}_k^T \mathbf{b} \\ &= \sum_{k=1}^{N} (-\frac{1}{2}\beta_k^2 \lambda_k - \beta_k \mathbf{e}_k^T \mathbf{b}) = \sum_{k=1}^{N} E_k. \end{aligned} \tag{4.10}$$

Therefore, the energy E can be decomposed into N separate terms by using a linear transformation $\mathbf{Q}^T = \mathbf{Q}^{-1}$ and the minimum value of E can be obtained by simultaneously minimizing E_k, $k = 1, 2, \cdots, N$, in $\mathbf{D}^N$ simultaneously. By solving $\partial E_k / \partial \beta_k = 0$ in (4.10), it can be seen that the network reaches its equilibrium $\mathbf{y}_0$ when

$$\beta_k = -\frac{\mathbf{e}_k^T \mathbf{b}}{\lambda_k} \equiv \beta_{0,k}, \quad \forall k. \tag{4.11}$$

The equilibrium point can also be found by solving $\nabla_y E = \mathbf{0}$ where

$$\nabla_y E = -(\mathbf{A} - \frac{T_x}{g}\mathbf{I})\mathbf{y} - \mathbf{b} = -\mathbf{M}_g \mathbf{y} - \mathbf{b}. \tag{4.12}$$

By using the symmetric property of $\mathbf{M}_g$ in the partial differentiation, the final result is given by

$$\begin{aligned} \mathbf{y}_0 &= -\mathbf{M}_g^{-1} \mathbf{b} = -\mathbf{Q} \mathbf{\Lambda}_g^{-1} \mathbf{Q}^T \mathbf{b} \\ &= \sum_{k=1}^{N} (-\frac{\mathbf{e}_k^T \mathbf{b}}{\lambda_k}) \mathbf{e}_k = \sum_{k=1}^{N} \beta_{o,k} \mathbf{e}_k. \end{aligned} \tag{4.13}$$

The minimal energy for $g = 1$ can be obtained by simultaneously minimizing the quadratic energy terms E_k's of (4.10) in $\mathbf{D}^N$.

The hardware annealing method is applied to the two-neuron network shown in Fig. 4.1. Until a saturation occurs in one or both neurons, the network is a linear time-varying system [35]. By diagonalizing The matrix $\mathbf{M}_g$ given in (4.5) can be diagonalized as $\mathbf{M}_g = \mathbf{Q}\mathbf{\Lambda}_g\mathbf{Q}^T$ where $\mathbf{\Lambda}_g = diag(\lambda_1, \lambda_2)$ and $\mathbf{Q} = [\mathbf{e}_1|\mathbf{e}_2]$ is the corresponding eigenvector matrix. The eigenvalues of the matrix are $\lambda = A_0 \pm A_1 - T_x/g$. The equilibrium $\mathbf{y}_0$ can be expressed as

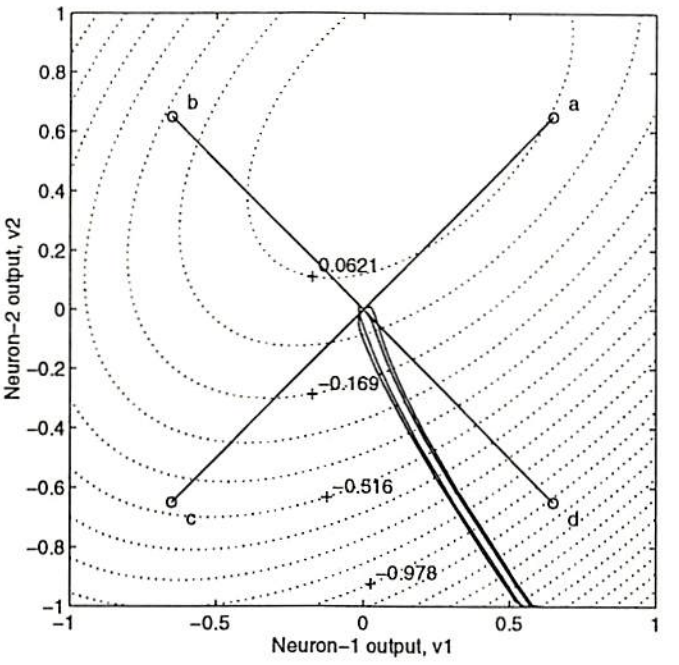
$$\mathbf{y}_0 = -\mathbf{M}_g^{-1}\mathbf{v}_u = -\frac{\mathbf{e}_1^T\mathbf{v}_u}{\lambda_1}\mathbf{e}_1 - \frac{\mathbf{e}_2^T\mathbf{v}_u}{\lambda_2}\mathbf{e}_2, \qquad \lambda_1, \lambda_2 \neq 0. \tag{4.14}$$

As λ_1 approaches zero, the first term of (4.14) is the dominating term and the evolution direction of $\mathbf{y}_0$ approaches the direction of the eigenvector $\mathbf{e}_1$. Notice that the global-minimum point is [+1 -1] which is toward the direction of $\mathbf{e}_1$.

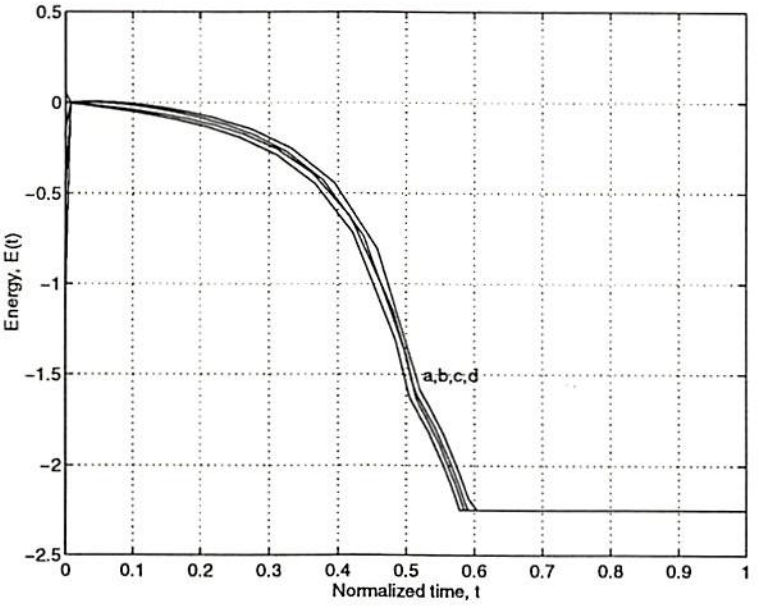
Matlab program simulation of the two-neuron network shown in Fig. 4.1 with hardware annealing capability is presented. The inter-neuron synapse weights are $A_{1,1} = A_{2,2} = 2$, $A_{1,2} = A_{2,1} = -0.5$ and $B_{1,1} = B_{2,2} = 1$. Figure 4.3 shows the trajectories of the outputs as the annealing process is applied to the network condition of Fig. 4.2. Regardless of initial state values, the network reaches the globally optimal point [+1 -1] which corresponds to the minimum energy state.

4.3 Measurement Results Using Standard Parts

A laboratory prototype of the two-neuron network was constructed using standard IC parts in the laboratory. A schematic diagram of a variable-gain neuron and fixed synaptic weights is shown in Fig. 4.4. The network condition of Fig. 4.2 and the initial location c are used. Figure 4.5 shows the network operation without annealing and the steady-state output value is [-1 -1] which corresponds to one of



(a)



(b)

Figure 4.3: Global optimization: Annealed network operation.

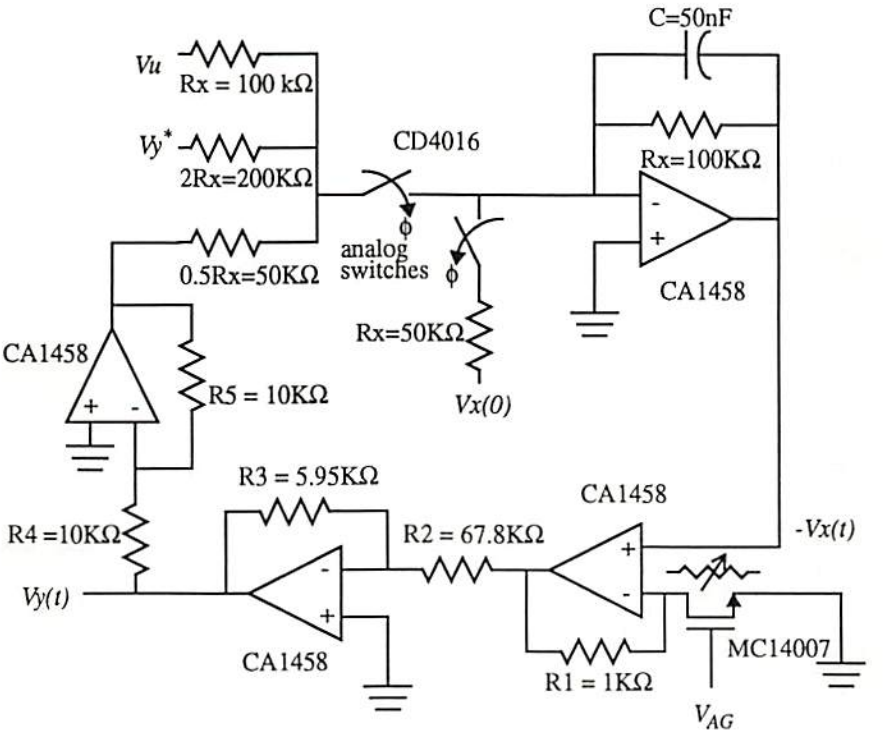


Figure 4.4: Board-level schematic diagram of a neuron and synapses for annealed CNN using standard IC parts.

the local-minimum point. Figure 4.6 shows the transfer curves of V_y versus -V_x with the gain-control voltages, V_{AG}, from 2 V to 10 V in step of 2 V. If the annealing method is applied by controlling the gain-control voltage, i.e., changing the neuron gain from a low-gain value to a high-gain value, the output value moves toward the origin first, then gradually settle down to the global-minimum point, which is shown in Fig. 4.7.

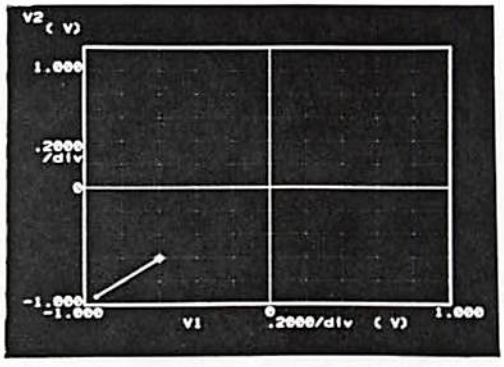


Figure 4.5: Measurement results of network operation without annealing.

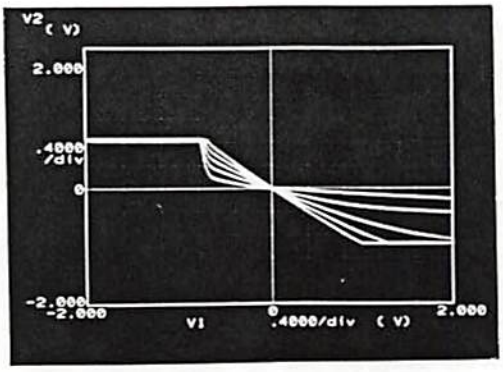


Figure 4.6: Measurement results of the transfer curves of V_y versus -V_x.

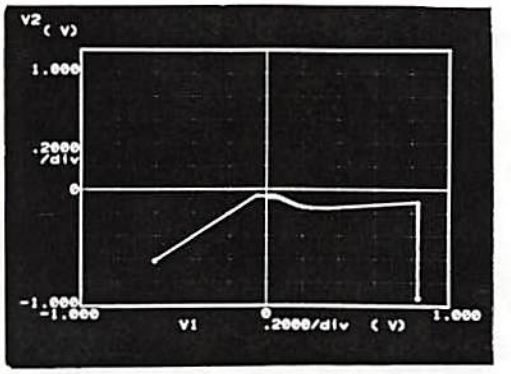


Figure 4.7: Measurement results of network operation with annealing.

4.4 A Variable-Gain Processing Element

4.4.1 Circuit Design

Microelectronic implementation of the networks with annealing capability can minimize power consumption by obtaining the optimal solution in a greatly reduced computation time. A variable-gain neuron cell consists of a summing circuit, an analog multiplier, and a nonlinear function circuit as shown in Figure 4.8. The hardware annealing method is operated by multiplying the state variable v_x with the neuron-gain control function g before the nonlinear function $f(\cdot)$ takes place. Therefore, an analog multiplier is inserted between the summing circuit and the nonlinearity circuit. A two-quadrant multiplier is adequate, because g is a positive function. The analog multiplier is a basic building block in many analog signal processing applications including artificial neural networks. However, special care must be taken in designing the circuit, because some analog multipliers often result in the post-multiplication $g \cdot f(v_x)$ in stead of the pre-multiplication $f(g \cdot v_x)$. In the post-multiplication case, the linear operation region and saturation levels are also

reduced as the gain decreases and the minimum gain g_{min} can not be made very small.

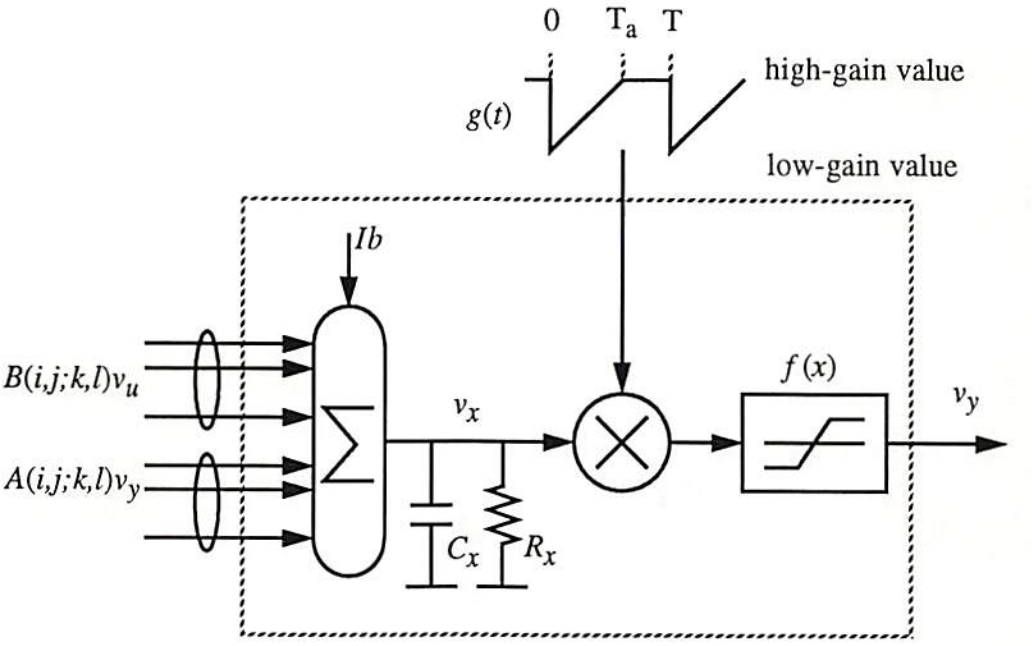


Figure 4.8: Block diagram of a variable-gain neuron cell for hardware annealing.

The complete schematic diagram of a variable-gain neuron cell is shown in Figure 4.9. One basic element in the circuit is the double-MOS differential resistor [55, 76] operating in triode region. It is used as a feedback resistor in the transimpedance amplifier as shown in Figure 4.9. The input/output relationship is

$$V_O - V_{ref} = \frac{I_2 - I_1}{2\mu C_{OX}(W/L)V_{GC}}, \tag{4.15}$$

where V_{ref} is the bias voltage which is close to the middle of the useful swing of output voltage V_O. For the negative feedback of an operational amplifier, the gain control voltage $V_{GC} = V_{DD} - V_C$ needs to be positive.

The equivalent resistance R_x of the transimpedance amplifier can be expressed as

$$R_x = \frac{1}{2\mu C_{OX}(W/L)V_{GC}}, \quad V_{GC} > 0. \tag{4.16}$$

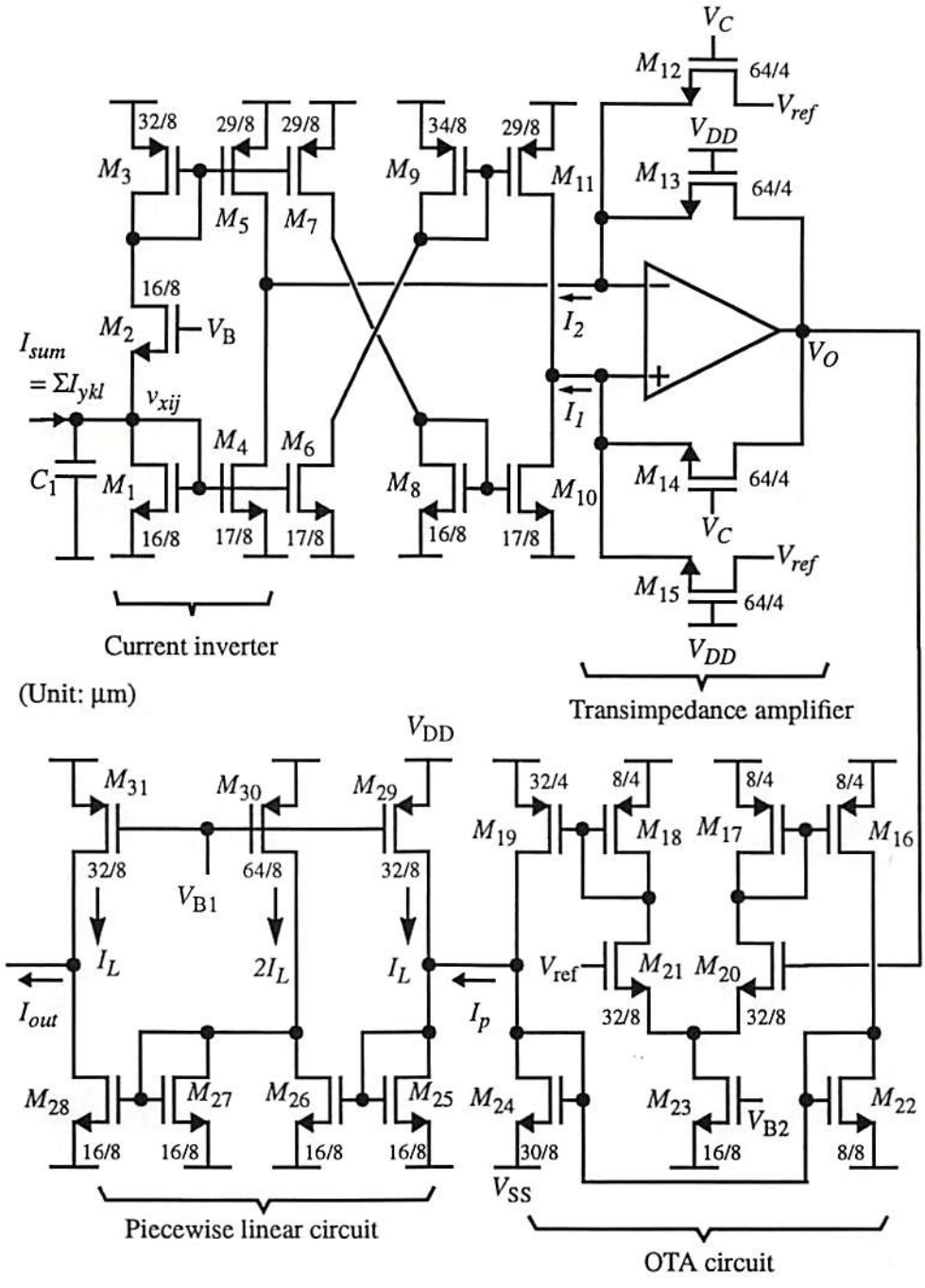


Figure 4.9: Complete schematic diagram of a variable-gain neuron cell.

It indicates that the value of the resistance R_x is inversely proportional to the voltage V_{GC}. Therefore, the summation of the weighted currents from the neighboring cells can not take place directly in the transimpedance amplifier because R_x varies as the annealing gain changes. A current inverter circuit which has a constant input equivalent resistance can be placed at the input stage of the multiplier to solve this problem.

The circuit for the nonlinear function $y = f(x)$ is accomplished by a piecewise-linear circuit which accepts current inputs. Thus, an OTA circuit is used to convert the voltage output of the transimpedance amplifier, V_O, into the current I_p. Detailed description and simulation results of the current inverter circuit, the OTA circuit, and the piecewise-linear circuit have been described in Chapter 3.

4.4.2 Simulation Results

SPICE-3 simulation results of a variable-gain neuron for several annealing gain values are shown in Fig. 4.10. In this example, V_{DD} is 5 V, $V_{SS} = 0$ V, and V_{ref} is 1.5 V. Bias voltages of V_B, V_{B1} and V_{B2} are set at 3.29 V, 3.77 V and 2.0 V, respectively. The normalization current is 10 μA. The neuron gain can be continuously changed from the low gain value 0.12 to the maximum gain $g_{max} = 1$ by controlling the voltage V_C from 2.95 V to 4.7 V. Simulation results demonstrate that this circuit can implement the annealing operation extremely well. A CNN chip with the variable-gain neuron circuit to support the optimal solution capability can be designed.

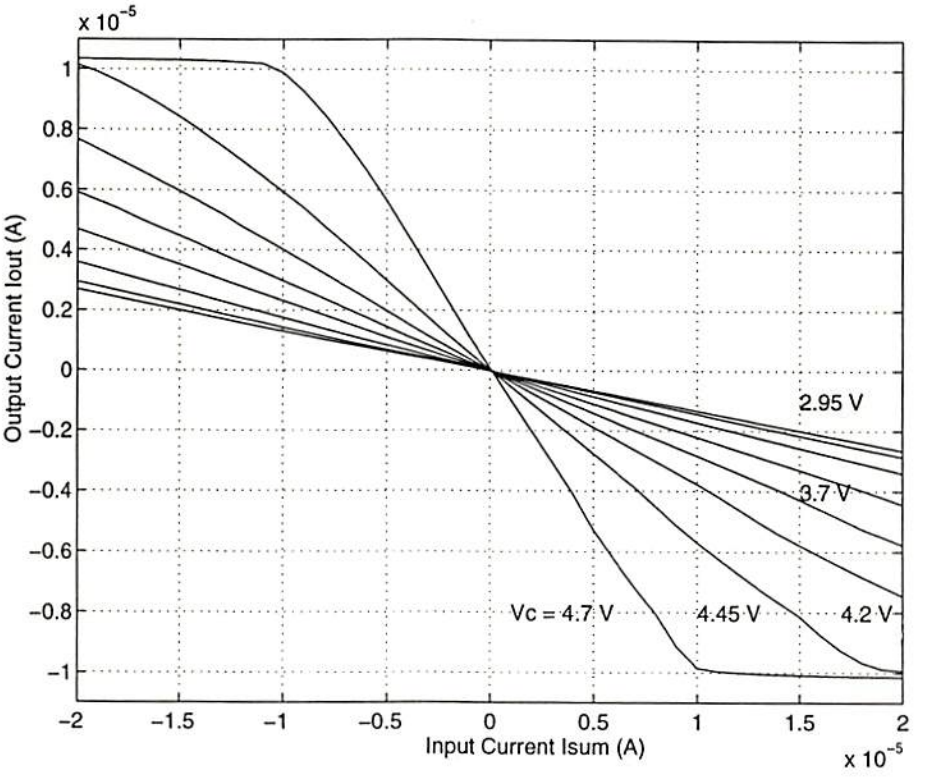


Figure 4.10: SPICE simulation results of the variable-gain neuron for several annealing gain values.

Chapter 5

Conclusions

In this dissertation research, a 5×5 paralleled array processor chip has been designed and fabricated in a 2-μm double-polysilicon, double-metal CMOS technology through the MOSIS Service. This prototype chip was constructed with many compact VLSI circuit components which were designed using the current-mode techniques. All building blocks were extensively simulated by the SPICE-3 program and successfully measured in the laboratory. The array processor chip can perform different functions by changing the cloning template content. A circuit board was built to demonstrate the operation of this chip. Experimental results of the hole filling and edge detection operations were presented.

A high-performance intelligent microsystem can be constructed by the array processor chips and software program. The system architecture is scalable in 2-dimension and can accommodate different numbers of processing elements in the array. A digital microprocessor or a personal computer can serve as the system controller to drive the custom-developed circuit board which contains the array processor chips and supporting hardware components. This whole system can be widely used for high-speed/high-performance information superhighway, and with possible significant applications in image/pattern recognition, computer vision, medical imaging, path planning, and autonomous robots.

The computing capability of the densely connected array processor is greatly enhanced by the hardware annealing technique which can quickly and effectively search for optimal solutions in many applications. The annealing operation was illustrated by an experimental circuit board which was built by the standard IC parts. VLSI design of a variable-gain computing element to support the annealing capability was described.

Reference List

[1] R. Kurzweil, *The Age of Intelligent Machines.* MIT Press: Cambridge, MA, 1990.

[2] J. A. Adam, "Interactive multi-media," *IEEE Spectrum Magazine*, vol. 30, pp. 22–23, Mar. 1993.

[3] H. Mizuno, "The fusion of home electronics with computer technologies," in *Tech. Digest IEEE Int'l Solid-State Circuits Conf.*, pp. 20–23, San Francisco, CA, Feb. 1994.

[4] W. S. Ng, B. L. Davies, R. D. Hibberd, A. G. Timoney, "Robotic surgery," *IEEE Engineering in Medicine and Biology*, pp. 120–125, Mar. 1993.

[5] D. Bearden *et al.*, "A 133 MHz 64b four-issue CMOS microprocessor," in *Tech. Digest IEEE Int'l Solid-State Circuits Conf.*, pp. 174–175, San Francisco, CA, Feb. 1995.

[6] A. Chamas *et al.*, "A 64b microprocessor with multimedia support," in *Tech. Digest IEEE Int'l Solid-State Circuits Conf.*, pp. 178–179, San Francisco, CA, Feb. 1995.

[7] W. J. Bowhill *et al.*, "A 300 MHz 64b quad-issue CMOS RISC microprocessor," in *Tech. Digest IEEE Int'l Solid-State Circuits Conf.*, pp. 182–183, San Francisco, CA, Feb. 1995.

[8] M. Asakura *et al.*, "A 34ns 256Mb DRAM with boosted sense-ground scheme," in *Tech. Digest IEEE Int'l Solid-State Circuits Conf.*, pp. 140–141, 324, San Francisco, CA, Feb. 1994.

[9] M. Nakamura *et al.*, "A 29ns 64Mb DRAM with hierarchical array architecture," in *Tech. Digest IEEE Int'l Solid-State Circuits Conf.*, pp. 246–247, San Francisco, CA, Feb. 1995.

[10] T. Sugibayashi *et al.*, "A 1Gb DRAM for file applications," in *Tech. Digest IEEE Int'l Solid-State Circuits Conf.*, pp. 254–255, San Francisco, CA, Feb. 1995.

[11] D. Dobberpuhl *et al.*, "A 200MHz 64b dual-issue CMOS microprocessor," in *Tech. Digest IEEE Int'l Solid-State Circuits Conf.*, pp. 106–107, 256, San Francisco, CA, Feb. 1992.

[12] D. Pham *et al.*, "A 3.0W 75SPECint92 85SPECfp92 superscalar RISC microprocessor," in *Tech. Digest IEEE Int'l Solid-State Circuits Conf.*, pp. 212–213, 341, San Francisco, CA, Feb. 1994.

[13] E. Rashid *et al.*, "A 64b microprocessor with multimedia support," in *Tech. Digest IEEE Int'l Solid-State Circuits Conf.*, pp. 178–179, San Francisco, CA, Feb. 1995.

[14] A. Saini, "Design of the Intel *Pentium*^{TM} processor," in *Proc. IEEE Int. Conf. Computer Design*, pp. 258–261, Cambridge, MA, Oct. 1993.

[15] H. I. H. Komiya, M. Yoshimoto, "Future technological and economic prospects for VLSI," *The Institute of Electronics, Information and Communication Engineers Trans. Electron*, vol. 76-C, pp. 15–30, Nov. 1993.

[16] B. J. Sheu, J. Choi, *Neural Information Processing and VLSI.* Kluwer Academic Publishers: Boston, MA, Jan. 1995.

[17] J. Yih, P. Mazumder, "A neural network design for circuit partitioning," *IEEE Trans. Computer-Aided Design*, vol. 9, pp. 1265–1271, Dec. 1990.

[18] S. T. Chakradhar *et al.*, "Toward massively parallel automatic test generation," *IEEE Trans. Computer-Aided Design*, vol. 9, pp. 981–994, Sep. 1990.

[19] S. Neusser *et al.*, "Neurocontrol for lateral vehicle guidance," *IEEE Micro Magazine*, vol. 13, pp. 57–66, Feb. 1993.

[20] A. Hiramatsu, "Integration of atm call admission control and link capacity control by distributed neural networks," *IEEE Jour. Selected Areas in Commun.*, vol. 9, pp. 1131–1138, Sep. 1991.

[21] D. W. Ruck *et al.*, "The multilayer perceptron as an approximation to a Bayes optimal discriminant function," *IEEE Trans. Neural Networks*, vol. 1, pp. 296–298, Dec. 1990.

[22] A. D. Culhane, M. C. Peckerar, C. R. K. Marrian, "A neural net approach to discrete Hartley and Fourier transforms," *IEEE Trans. Circuits and Systems*, vol. 36, pp. 695–703, May 1989.

[23] G. R. Gindi *et al.*, "Neural network and conventional classifiers for fluorescence-guided laser angioplasty," *IEEE Trans. on Biomedical Engineering*, vol. 38, pp. 246–252, Mar. 1991.

[24] L. O. Chua, L. Yang, "Cellular neural networks: theory," *IEEE Trans. Circuits and Systems*, vol. 35, pp. 1257–1272, Oct. 1988.

[25] M. Holler, S. Tam, H. Castro, R. Benson, "An electrically trainable artificial neural network (ETANN) with 10240 "floating gate" synapses," in *Proc. IEEE/INNS Int'l Joint Conf. Neural Networks*, vol. II, pp. 191–196, Washington, DC, Jun. 1989.

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

[27] J. Alspector, A. Jayakumar, S. Luna, "Experimental evaluation of learning in a neural microsystems," *Proc. of Neural Information Processing Systems*, vol. 4, pp. 871–878, 1991.

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

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

[30] Y. Arima *et al.*, "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.

[31] B. J. Sheu, J. Choi, C.-F. Chang, "An analog neural network processor for self-organizing mapping," in *Tech. Digest IEEE Int'l Solid-State Circuits Conf.*, pp. 136–137, 266, San Francisco, CA, Feb. 1992.

[32] J. Choi, 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.

[33] L. O. Chua, L. Yang, "Cellular neural networks: applications," *IEEE Trans. Circuits and Systems*, vol. 35, pp. 1273–1290, Oct. 1988.

[34] T. Roska, L. Chua, "Cellular neural networks with non-linear and delay-type template elements and non-uniform grids," *Int Jour. Circuit Theory and Applications*, vol. 20, pp. 469–481, 1992.

[35] S. H. Bang, "Performance optimization in cellular neural networks and associated VLSI architectures," Tech. Rep. USC-SIPI Report No. 268, Signal and Image Processing Institute, University of Southern California, University Park/MC-2564, Los Angeles, CA 90089, 1994.

[36] J. E. Varrientos, E. Sánchez-Sinencio, J. Ramirez-Angulo, "A current-mode cellular neural network implementation," *IEEE Trans. Circuits and Systems, II*, vol. 40, pp. 147–155, Mar. 1993.

[37] S. Espejo, *VLSI Design and Modeling of CNNs.* Ph.D. Dissertation, University of Sevilla, Spain, Apr. 1994.

[38] H. Halonen, V. Porra, T. Roska, L. O. Chua, "Programmable analog VLSI CNN chip with local digital logic," in *Proc. IEEE Int'l Symp. Circuits and Systems*, pp. 1291–1294, 1991.

[39] A. Rodríguez-Vázquez *et al.*, "Current-mode techniques for the implementation of continuous-and discrete-time cellular neural networks," *IEEE Trans. Circuits and Systems, II*, vol. 40, pp. 132–146, Mar. 1993.

[40] I. A. Baktir, M. A. Tan, "Analog CMOS implementation of cellular neural networks," *IEEE Trans. Circuits and Systems, II*, vol. 40, pp. 200–206, Mar. 1993.

[41] J. M. Cruz, L. O. Chua, "A CNN chip for connected component detection," *IEEE Trans. Circuits and Systems*, vol. 38, pp. 812–817, Jul. 1991.

[42] S. Espejo, A. Rodríguez-Vázquez, R. Domínguez-Castro, J. L. Huertas, E. Sánchez-Sinencio, "Smart-pixel cellular neural networks in analog current-mode CMOS technology," *IEEE Jour. Solid-State Circuits*, vol. 29, pp. 895–905, Aug. 1994.

[43] R. Domínguez-Castro, S. Espejo, A. Rodríguez-Vázquez, R. Carmona, "A cnn universal chip in cmos technology," in *Proc. IEEE Int'l Workshop on Cellular Neural Networks and their Applications*, Rome, Italy, Dec. 1994.

[44] P. Kinget, M. S. J. Steyaert, "A programmable analog cellular neural network cmos chip for high speed image processing," *IEEE Jour. Solid-State Circuits*, vol. 30, pp. 235–243, Mar. 1995.

[45] K. Halonen, V. Porra, T. Roska, L. O. Chua, "Programmable analogue VLSI CNN chip with local digital logic," *Int. J. Circuit Theory and Applicaitons*, vol. 20, pp. 573–582, Oct. 1992.

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

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

[48] P. K. Simpson, "Foundation of neural networks," in *Artificial Neural Networks: Paradigms, Applications, and Hardware Implementations* (E. Sánchez-Sinencio and C. Lau, eds.), pp. 29–37, IEEE Press: New York, NY, 1992.

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

[50] J. Choi, B. J. Sheu, "VLSI design of compact and high-precision analog neural network processors," in *Proc. IEEE/INNS Int'l Joint Conf. Neural Networks*, vol. II, pp. 637–641, Baltimore, MD, Jul. 1992.

[51] W.-C. Fang, B. J. Sheu, O. T.-C. Chen, 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.

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

[53] T. Quarles, A. R. Newton, D. O. Pederson, A. Sagiovanni-Vincentelli, Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, *SPICE3 Version 3f3 User's Manual*, May. 1993.

[54] A. Vladimirescu, S. Liu, "The simulation of MOS integrated circuits using SPICE2," Tech. Rep. Electron. Res. Lab. Memo ERL-M80/7, University of California, Berkeley, Oct. 1980.

[55] M. Ismail, S. V. Smith, R. G. Beale, "A new MOSFET-C universal filter structure for VLSI," *IEEE Jour. Solid-State Circuits*, vol. 23, pp. 183–194, Feb. 1988.

[56] C. Tomovich, "MOSIS–A gateway to silicon," *IEEE Circuits & Devices Magazine*, vol. 4, pp. 22–23, Mar. 1988.

[57] 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.

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

[59] G. S. P. G. V. Wilson, "On the stability of the traveling salesman problem algorithm of hopfield and tank," *Biol. Cybern.*, vol. 58, pp. 63–70, 1988.

[60] C. Peterson, "Parallel distributed approaches to combinatorial optimization: Benchmark studies on traveling salesman problem," *Neural Computation*, vol. 2, pp. 261–269, 1990.

[61] R. Cuykendall, R. Reese, "Scaling the neural TSP algorithm," *Biol. Cybern.*, vol. 60, pp. 365–371, 1989.

[62] E. Wacholder, J. Han, R. C. Mann, "A neural network algorithm for the multiple traveling salesmen problem," *Biol. Cybern.*, vol. 61, pp. 11–19, 1989.

[63] S. Wimer, I. Koren, "Analysis of strategies for constructive general block placement," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, pp. 15–30, Aug. 1988.

[64] P.-H. Shih, W.-S. Feng, "An application of neural networks on channel routing problem," *Parallel Computing*, vol. 17, pp. 229–240, 1991.

[65] A. Ushida, L. O. Chua, "A global optimization algorithm based on circuit partitioning technique," in *Proc. IEEE Int'l Symp. Circuits and Systems*, pp. 475–478, 1992.

[66] S. Kirkpatrick, C. D. Gelatt, Jr., M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671–680, May 1983.

[67] H. Szu, R. Hartley, "Fast simulated annealing," *Physics Letters A*, vol. 122, pp. 157–162, Jun. 1987.

[68] W. Jeffrey, R. Rosner, "Optimization algorithms: simulated annealing and neural network processing," *Astrophysical Journal*, vol. 310, pp. 473–481, Nov. 1986.

[69] C. R. Nassar, M. R. Soleymani, "Codebook design for trellis quantization using simulated annealing," *IEEE Trans. on Speech and Audio Processing*, vol. 1, pp. 400–404, Oct. 1993.

[70] J. H. M. Korst, E. H. L. Aarts, "Combinatorial optimization on a Boltzmann machine," *Jour. Parallel and Distributed Computing*, vol. 6, pp. 331–357, 1989.

[71] E. H. L. Aarts, J. H. M. Korst, *Simulated Annealing and Boltzmann Machine.* Wiley Chichster, 1988.

[72] B. W. Lee, B. J. Sheu, "Parallel hardware annealing for optimal solutions on electronic neural networks," *IEEE Trans. Neural Networks*, vol. 4, pp. 588–599, Jul. 1993.

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

[74] C. Peterson, J. R. Anderson, "A mean field learning algorithm for neural networks," *Complex Systems*, vol. 1, no. 5, pp. 995–1019, 1987.

[75] C. Peterson, "Mean field theory neural networks for feature recognition, content addressable memory and optimization," *Connection Science*, vol. 3, pp. 3–33, 1991.

[76] K. Bult, H. Wallinga, "A CMOS four-quadrant analog multiplier," *IEEE Jour. Solid-State Circuits*, vol. 21, pp. 430–435, June 1986.

Appendix A

SPICE Level-2 and BSIM_plus Models

The 5×5 array processor chip was designed and verified via extensive SPICE simulations. The SPICE (Simulation Program with Integrated Circuit Emphasis) program [A.1, A.2] has been widely used for circuit design and analysis since its introduction a decade ago. Device modeling plays an important role in VLSI circuit design Because of the computer-aided circuit analysis results are strongly depend on the models used, it is very critical to use accurate device models to simulate circuit. The SPICE-3f3 programs have provided several built-in MOS transistor models [A.3]-[A.7]. The Level-2 model is widely used for the analysis of analog circuits. Its expressions are derived from detailed device physics, However, when the circuits contain small-geometry transistors, its accuracy is quite limited. The Level-4 model, which is known as the BSIM model, is ideal for the simulation of submicron digital and analog circuits Besides, the BSIM_plus model has been developed at University of Southern California. It is a greatly enhanced version over the original BSIM model. The modeling equations for the experimental BSIM_plus model can be found in [A.8]-[A.10].

The detailed modeling equations for the Level-2 model can be found in [A.11, A.12]. Table A.1 lists the parameter set of the Level-2 model which was used to

simulate 80% of the circuits in this dissertation. These parameter values were extracted from a 2-μm double-polysilicon p-well CMOS process. The test wafers were fabricated by Orbit Semiconductor Inc. through the MOSIS Service.

Table A.1: Typical parameter values for the LEVEL-2 model.

parameters	symbols	n-channel	p-channel	unit
V_{tho}	VTO	0.932	-0.745	V
T_{ox}	TOX	3.18E-8	3.18E-8	m
X_j	XJ	2.0E-7	2.0E-7	m
N_{sub}	NSUB	3.2E+16	8.6E+15	cm^{-3}
Φ_f	PHI	0.6	0.6	V
λ	LAMBDA	0.026	0.046	V^{-1}
μ_0	U0	548.4	199	$cm^2/V-s$
μ_{crit}	UCRIT	41470	91570	V/cm
μ_{exp}	UEXP	0.169	0.327	-
γ	GAMMA	0.943	0.491	$V^{1/2}$
V_{max}	VMAX	54130	99990	m/s
L_d	LD	2.4E-7	2.3E-7	m
C_j	CJ	4.02E-4	2.04E-4	F/m^2
M_j	MJ	0.45	0.46	-
C_{jsw}	CJSW	5.97E-10	1.04E-10	F/m
M_{jsw}	MJSW	0.33	0.12	-
C_{GDO}	CGDO	3.91E-10	3.70E-10	F/m
C_{GSO}	CGSO	3.91E-10	3.70E-10	F/m
C_{GBO}	CGBO	4.14E-10	3.50E-10	F/m

Reference List

[A.1] L. W. Nagel, "SPICE2: A computer program to simulate semiconductor circuits," Electron. Res. Lab. Memo ERL-M520, University of California, Berkeley, May 1975.

[A.2] T. Quarles, A. R. Newton, D. O. Pederson, A. Sangiovanni-Vincentelli, *SPICE 3F3 User's Guide*, Dept. of EECS, U.C. Berkeley, CA, May 1993.

[A.3] A. Vladimirescu, S. Liu, "The simulation of MOS integrated circuits using SPICE2," Electron. Res. Lab. Memo ERL-M80/7, University of California, Berkeley, Oct. 1980.

[A.4] B. J. Sheu, D. L. Scharfetter, P. K. Ko, M.-C. Jeng, "BSIM: Berkeley short-channel IGFET model for MOS transistors," *IEEE Jour. of Solid- State Circuits*, vol. SC-22, no. 4, pp. 458-466, Aug. 1987.

[A.5] B. J. Sheu, W.-J. Hsu, P. K. Ko, "An MOS transistor charge model for VLSI design," *IEEE Trans. on Computer-Aided Design*, vol. CAD-7, no. 4, pp. 520-527, Apr. 1988.

[A.6] B. J. Sheu, "MOS transistor modeling and characterization for circuit simulation," Electron. Res. Lab. Memo ERL-M85/22, University of California, Berkeley, Oct. 1985.

[A.7] T. Sakurai, A. R. Newton, "A simple MOSFET model for circuit analysis," *IEEE Trans. on Electron Devices*, vol. 38, no. 4, pp. 887-894, Apr. 1991.

[A.8] S. M. Gowda, B. J. Sheu, "BSIM_plus: an advanced SPICE model for submicron MOS VLSI circuits," *IEEE Trans. on Computer-Aided Design*, vol. 13, no. 9, pp. 1166-1170, Sept. 1994.

[A.9] R. C. Chang, B. J. Sheu, "An analog MOS model for circuit simulation and benchmark test results," *IEEE Int'l Symposium on Circuits and Systems*, vol. I, pp. 311-314, London, England, May 1994.

[A.10] S. M. Gowda, *BSIM_plus: An Advanced MOS Transistor Model for VLSI Circuits*, Tech. Rep. #225, Signal and Image Processing Institute, USC, Los Angeles, CA, Nov. 1992.

[A.11] D. A. Divekar, *FET Modeling for Circuit Simulation*, Kluwer Academic: Boston, MA, 1988.

[A.12] P. Antognetti and G. Massobrio, *Semiconductor Device Modeling with SPICE*, McGraw-Hill: New York, 1988.

Appendix B

Selected Templates and Applications

The paralleled array processor can be used in many scientific and engineering applications by using different cloning templates [B.1]. A selected known templates for the network is listed in Table B.1 [B.2, B.3, B.4]. In the Table, "M" is the matrix of the input image pixel, "X" is the don't care term, I_b is the network bias, $v_x(0)$ is the initial state, v_u is the external input, $v_{x,s}(0)$ and $v_{u,s}$ are the state and input of border elements. Hole filling operation is to fill the holes of objects in the input image. The edges of the input image can be extracted by the edge detection function. Noise filtering function is to perform a neighborhood average of the pixel values. Connected component detection can reduce connected components in horizontal lines to a single pixel and project them toward the right with one-pixel separation. One pixel-wide vertical and diagonal line can be eliminated by the horizontal line detection operation. Notice that solid objects are preserved because they consist of multiple horizontal lines. Logic-AND operation is to execute pixel-wise logic-AND operation between two images. Isolated-pixel detection retains only every isolated pixel.

The processor array based on the cellular networks can process the gray-level input images and be used to perform feature extraction, motion detection & estimation, halftoning, including motion compensation, object counting & size estimation,

Table B.1: Selected templates for the network [B.2, B.3, B.4].

Application	TA	TB	$v_x(0)$	v_u	I_b	$v_{x,s}(0)$	$v_{u,s}$
Hole Filling	$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 0 \end{pmatrix}$	1	M	-1	-1	X
Laplacian Operator for Edge Detection	$\begin{pmatrix} 0 & -0.5 & 0 \\ -0.5 & 2 & -0.5 \\ 0 & -0.5 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$	M	M	-2.5	X	X
Edge Detector	$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} -0.25 & -0.25 & -0.25 \\ -0.25 & 2 & -0.25 \\ -0.25 & -0.25 & -0.25 \end{pmatrix}$	M	M	-2	X	-1
Noise Filtering	$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$	M	X	0	0	X
Connected Component Detection	$\begin{pmatrix} 0 & 0 & 0 \\ 1 & 2 & -1 \\ 0 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$	M	X	0	-1	X
Horizontal Line Detector	$\begin{pmatrix} 0 & 0 & 0 \\ 1 & 2 & 1 \\ 0 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$	M	X	0	0	X
AND	$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 1.5 & 0 \\ 0 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 1.5 & 0 \\ 0 & 0 & 0 \end{pmatrix}$	M1	M2	-1	X	X
Isolated Pixels Elimination	$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} -0.25 & -0.25 & -0.25 \\ -0.25 & 0.5 & -0.25 \\ -0.25 & -0.25 & -0.25 \end{pmatrix}$	M	M	-4	X	-1

path tracking, and collision avoidance. It also can be utilized to detect minima and maxima in a 3-D complex surface and detect area with gradients that exceed a given threshold.

Reference List

[B.1] T. Roska, L. Chua, "The CNN Universal Machine - An Analogic Array Computer," *IEEE Trans. Circuits and Systems, II*, vol. 40, pp. 163-173, Mar. 1993.

[B.2] S. Espejo, *VLSI Design and Modeling of CNNs*, Ph.D. Dissertation, University of Sevilla, Spain, Apr. 1994.

[B.3] B. J. Sheu, J. Choi, *Neural Information Processing and VLSI*, Kluwer Academic Publishers: Boston, MA, Jan. 1995.

[B.4] T. Roska, "CNN analogic (dual) software library," Internal Report DNS-1-1993, Computer and Automation Institute, Hungarian Academy of Science, Jan. 1993.

Appendix C

Related Design Issues

C.1 Layout Design Considerations

Careful physical layout is extremely important in the design of a high-performance paralleled array processors. In contrast with the traditional analog circuit architectures, the array processor is a massively interconnected network with a very large number of processing elements. Therefore, the size of each processing element should be minimized to reduce the total chip area or to integrate more elements in a given-size chip. Although the array processor is only locally connected, the massive interconnections still exist. For example, the interconnection occupies around 40% of the total area in our 5×5 array processor chip. In addition, due to the matrix configuration of array processors, additional constraints should be taken into consideration, such as the matching of X and Y dimensions around the matrix, horizontal and vertical signal feedthrough, and input/output terminal locations [C.1].

The dependence of circuit performance on physical layout is in general much more critical for analog circuits than that for digital circuits due to the sensitive nature of analog circuits [C.1]. Device mismatch effect can seriously degrade the circuit performance. By using the common centroid technique and the fully-differential architecture, the device mismatch effect can be alleviated. The effects of parasitics and

noise coupling on analog circuit layout can also lead to various kinds of performance degradations if proper cares were not taken.

Layout design of the current mirrors is very crucial to ensure the circuit linearity and the chip performance. In the digitally-programmable synapse weight circuit, the current mirror pairs with large current-gain should be carefully realized so that the programmability of the array processor can be achieved. Multi-section transistors can be used in the current mirror pair to eliminate the effects of process variation and guarantee the correct current gain.

C.2 Scalability of the Network

High-performance intelligent microsystems can greatly benefit from the scalable design, even though the multi-chip module technology can integrate many semiconductor dies onto a high-density interconnect substrates. Each array processor chip is to be developed in a modular format so that efficient data flow can be supported by the local and global interconnection. The required number of array processor chips and other supporting hardware components depends on the problem sizes.

The network size can be increased by connecting the identical processor chips in a two-dimensional grid array as shown in Fig. C.1. The dimension of the array processor chip is limited by the available chip area and pin counts. The original processor chip only requires the terminals for the external inputs and the final outputs. In order to connect those chips into a large-size network, the processing elements located at the edge of the processor chips must be able to transmit and receive the internal signals. The internal signals could be in voltage or current form which can be significantly degraded by the capacitive loading of the pins and the interconnection wires.

To reduce the complexity of the large-size networks for image processing applications, optical input capability must be equipped with each array processor chip. In addition, the output results should be transmitted in a time-multiplexing scheme. Each processor chip may need a chip-enable signal for a higher-level multiplexing configuration.

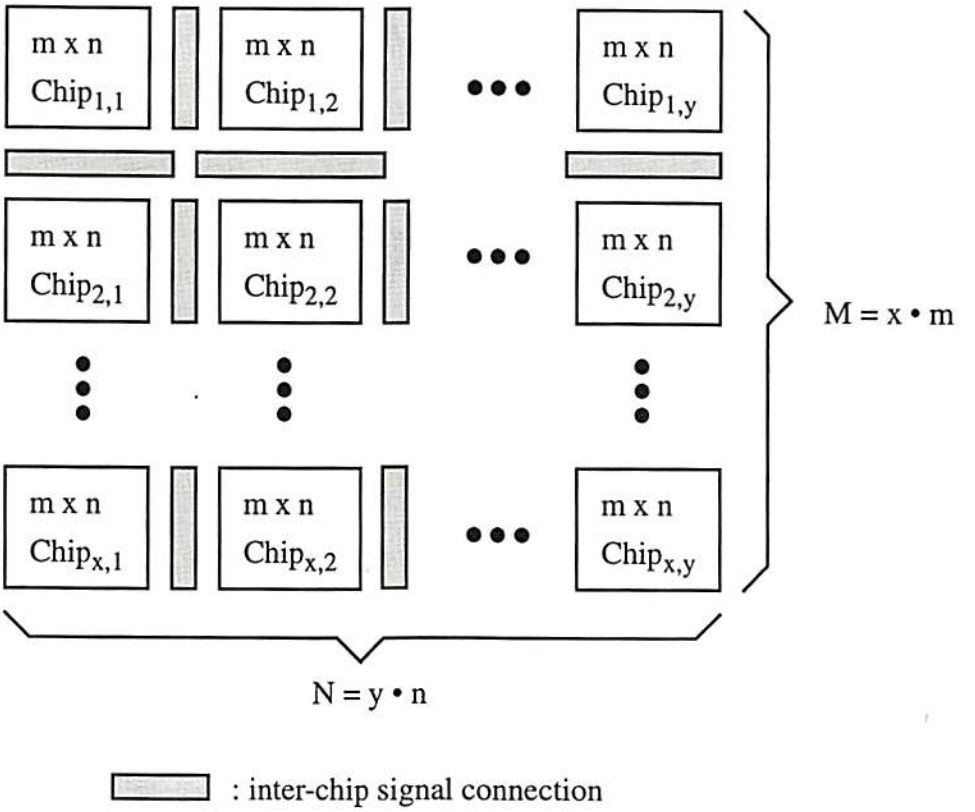


Figure C.1: Processor chips placed in a two-dimensional grid array.

C.3 Discussion on Desirable Features

C.3.1 Low-Power Circuits Design

Power consumption is a very important design issue for a microsystem. For many years, the speed, accuracy and silicon area were the major concerns as long as the heat generation of the systems was not too stringent. Due to the strong demand of

portable information systems, various hardware and software approaches have been used to reduce power consumption. Figure C.2 summaries some known approaches to reduce the power consumption at different design levels [C.2].

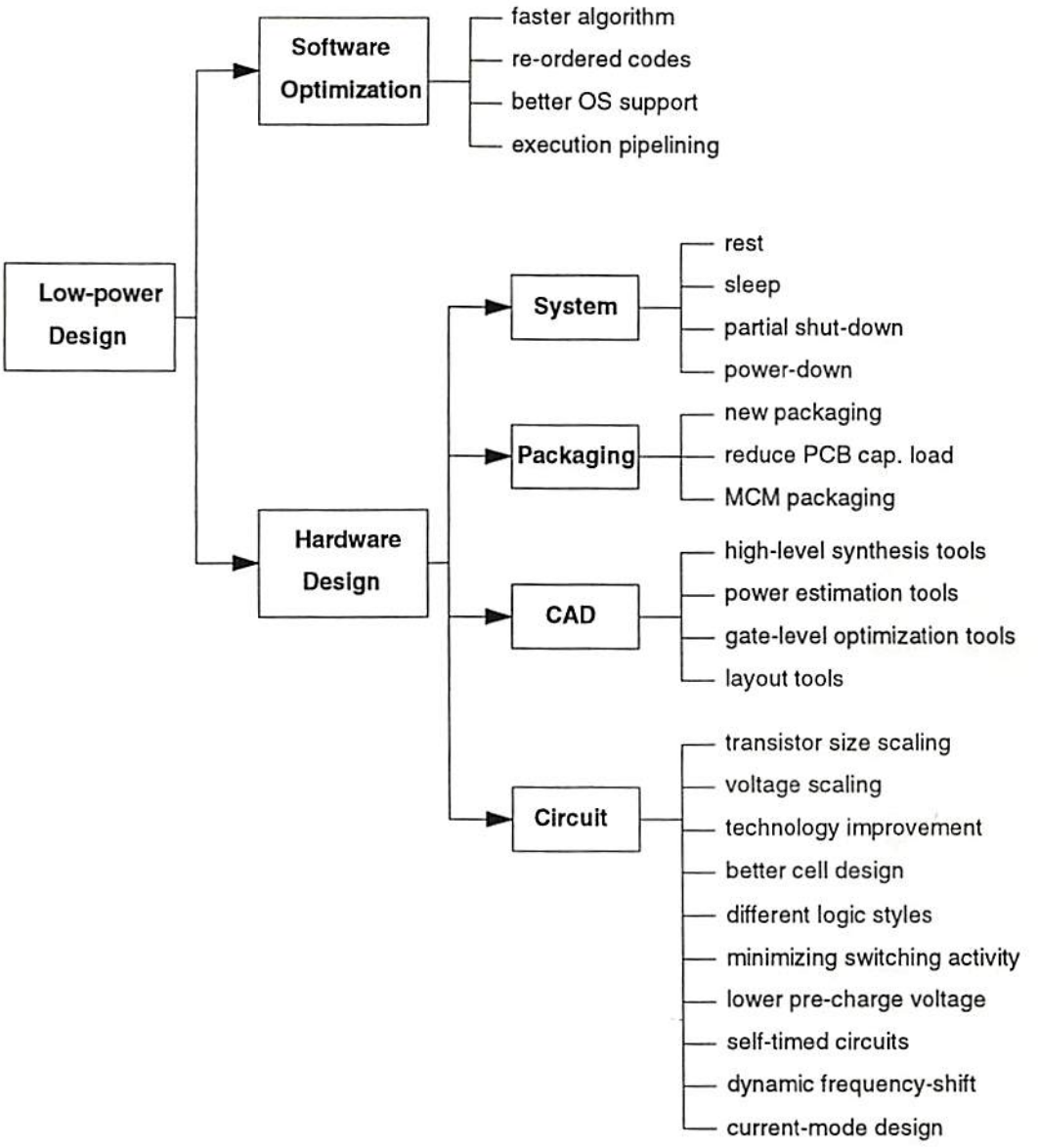


Figure C.2: Low-power design approaches [C.2].

Power-saving is not confined only to the technological issues. Software or hardware design methodologies also make a significant difference in many cases [C.3].

For example, careful software optimization can reduce power consumption by eliminating unnecessary codes and, thus, reduce the execution time and electric power consumption. This can be achieved by faster algorithms, re-ordered codes, better operating system support or execution pipelining. On the other hand, better CAD tools, such as high-level synthesis tools and power estimation tools, have been developed to assist VLSI designers to locate the major power consumption modes and to minimize the power consumption [C.4, C.5].

The hardware power-saving scheme can be classified into several categories. From a system point of view, overall power consumption can be greatly reduced by switching the system among the running, rest, sleep, or even shutdown modes. Many other system design techniques have also been utilized in the development of the newest PCs to meet the "energy-star" requirement. For example, minimizing the external capacitance load in PCB design, or using MCM technology for packaging can significantly reduce overall power consumption. At the circuit level, many design approaches have been recommended. The key factors are the feature size scaling due to technology advances [C.6] and signal voltage swing reduction [C.7]. Most of the results are focused on low-power digital circuits and systems design. In digital circuit design [C.8, C.9], power can be saved by minimizing switching activity factor, using dynamic logic circuits instead of static gates, lowering pre-charge voltage level to $V_{DD} - V_{th}$ instead of V_{DD}, using the pass-transistor logic configuration and truly single-phase clocking scheme, or even dynamic frequency-shift. Comparison between digital and analog circuits design is listed in Table C.1 [C.10]. Notice that the analog circuits design has special properties which are quite different from those of digital circuits design. Power reduction methods for digital circuits can not be applied to analog circuits without modification. So far, in the analog circuits and

systems, low-power optimization is mostly confined to modules, such as analog-to-digital converters [C.11]. In mixed-signal chips, common supply voltage can be used in digital and analog circuit blocks. Therefore, exploration of low-voltage/low-power design of analog circuits and mixed-signal chips is urgently necessary. Current-mode design can reduce the operating voltage to a very low value so that the circuit can consume very low power.

Table C.1: Comparison of digital and analog circuits design [C.10].

	Digital Circuit Design	Analog Circuit Design
Technology	CMOS prevailing	CMOS prefered, but Bipolar also used
Scalability (device size)	yes (can be scaled down to reduce the power consumption, and silicon area)	not necessary; "small is not always beautiful"
Portability	easy	hard
Device Types	transistor (mostly)	all types (transistor, capacitor, resistor, inductor)
Characteristics	switching	full range
Matching	average	essential
Temperature Effect	speed (mostly)	functionality (change the bias point)
Simulation	all various levels	spice (mostly)
Approaches	full custom, standard cell, gate array, FPGA.	full custom

C.3.2 Optical Input Capability

In many applications, the incoming signals processed by the paralleled array processor chip is a real image in the optical format. Thus, it is desirable to include smart-pixel configurations that can receive the optical input information by the photo-sensitive diodes or transistors [C.13, C.12]. The array processor with optical input capability can reduce the pin count of the chip without using the time-multiplexing scheme so that more processing elements or a larger array can be integrated on a single chip.

Photo-sensitive devices can be fabricated by the standard CMOS technology. The reverse-biased photodiodes can be formed between n^+-diffusion and the substrate; or between the well and the substrate in an n-well CMOS process [C.14, C.15]. For example, in a 1.6-μm CMOS technology, a 20 nA photocurrent was measured from a 100×100 μm^2 well-substrate diode [C.12]. If a vertical CMOS-BJT is utilized as a phototransistor, the photo-current, I_{Tr}, is (β+1) times that of the same-sized photodiode, where β is the transistor current-gain [C.12]. A typical measured value of the transistor current-gain is 38 [C.12]. In order to further increase the photo-current level, two bipolar transistors can be connected in a Darlington configuration and the resultant photo-current is (β+1) times of that of a single bipolar transistor. One measured photo-current for a 60×60-μm^2 Darlington-phototransistor is $18 \pm 2 \mu A$ and the corresponding dark current is around $250 \pm 10 pA$ [C.12]. Therefore, the achievable dynamic range is 100 dB.

Reference List

[C.1] D. J. Chen, B. J. Sheu, "Automatic layout generation for mixed analog-digital VLSI neural chips," in *Proc. IEEE Int. Conf. Computer Design*, pp. 29–32, Cambridge, MA, Oct. 1990.

[C.2] S. H. Jen, Dept. of EE/EP, USC, *priviate communication*, 1995.

[C.3] J. Snyder, J. McKie, B. Locanthi, "Low-power software for low-power people," in *IEEE Symp. on Low Power Electronics*, pp. 32–35, San Diego, CA, Oct. 1994.

[C.4] K. Keutzer, P. Vanbekbergen, "The impact of cad on the design of low power digital circuits," in *IEEE Symp. on Low Power Electronics*, pp. 42–45, San Diego, CA, Oct. 1994.

[C.5] P. Landman, J. Rabaey, "Power estimation for high level synthesis," in *Proc. 1993 EDAC Conference*, Paris, Feb. 1993.

[C.6] J. M. C. Stork, "Technology leverage for ultra low power information systems," in *IEEE Symp. on Low Power Electronics*, pp. 52–55, San Diego, CA, Oct. 1994.

[C.7] C. S. D. Liu, "Trading speed for low power by choice of supply and threshold voltages," *IEEE Jour. Solid-State Circuits*, pp. 10–17, Jan. 1993.

[C.8] A. Chandrakasan, S. Sheng, R. W. Brodersen, "Low-power CMOS digital design," *IEEE Jour. Solid-State Circuits*, vol. 27, pp. 473–484, Apr. 1992.

[C.9] M. Horowitz, "Low power digital design," in *IEEE Symp. on Low Power Electronics*, pp. 8–11, San Diego, CA, Oct. 1994.

[C.10] J. Tang, "Application, design and test of mixed-signal ICs," in *Proc. of Modern Engineering and Technology Symposium*, Chinese Institute of Engineers on Taiwan, pp. 19-28 Taipei, Dec. 1994.

[C.11] T. Cho, D. Cline, C. Conroy, P. Gray, "Design considerations for low-power high-speed CMOS analog/digital converters," in *IEEE Symp. on Low Power Electronics*, pp. 70–73, San Diego, CA, Oct. 1994.

[C.12] S. Espejo, A. Rodríguez-Vázquez, R. Domínguez-Castro, J. L. Huertas, E. Sánchez-Sinencio, "Smart-pixel cellular neural networks in analog current-mode CMOS technology," *IEEE Jour. Solid-State Circuits*, vol. 29, pp. 895–905, Aug. 1994.

[C.13] C. A. Mead, *Analog VLSI and Neural Systems.* Addison-Wesley Publishing Company: Reading, MA, 1989.

[C.14] A. H. Sayles, J. P. Uyemura, "An optoelectronic CMOS memory circuit for parallel detection and storage of optical data," *IEEE Jour. Solid-State Circuits*, vol. 26, pp. 1110–1115, Aug. 1991.

[C.15] A. Gruss, L. R. Carley, T. Kanade, "Integrated sensor and range-finding analog signal processor," *IEEE Jour. Solid-State Circuits*, vol. 26, pp. 184–191, Mar. 1991.

Appendix D

Software Modules for Compact Software-Hardware Codesign Systems

A compact high-performance computing architecture uses a hybrid analog-digital scheme to construct an intelligent supercomputing machine [D.1, D.2]. It is a cellular multidimensional array computer with distributed local memories, analogic (analog-and-logic) computing modules, as well as local and global communication and control units. The global stored-program is a key feature of the computing architecture for implementing the algorithms.

D.1 Software-Hardware Codesign

Fig. D.1 shows a software-hardware codesign scheme [D.2]. The C-like high-level language is written and translated into assembly language code using a compiler. Detailed template information can be included as a built-in or user-specific library. Some optimization techniques can also be applied during the compilation stage. For example, a superpipelining execution is possible by loading the next template coefficient data, executing a CNN operation, and producing the previous results at the same time.

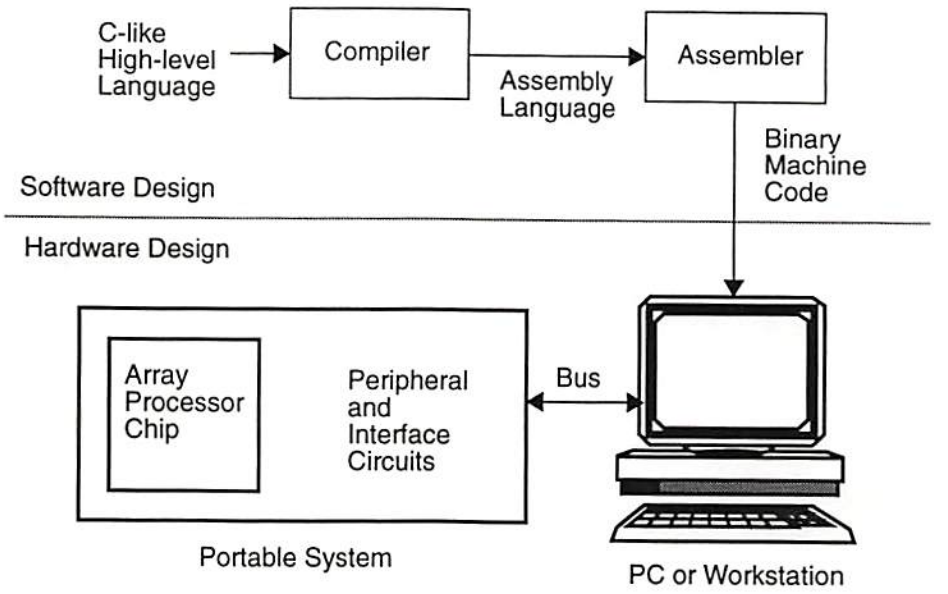


Figure D.1: A software-hardware codesign scheme.

An assembler can be used to translate the assembly language code generated in the previous step into binary machine code. An array processor chip can either attach directly and work as a co-processor inside the host machine, or communicate through a standard PC/workstation bus interfaces such as ISA bus or VME bus. Storage of the program and template data can be achieved by using either built-in RAMs or off-chip RAMs. Two different clock signals, one for analog processing and one for digital operation, are used.

D.2 Software Design Example

Detection of the intersection of vertical and diagonal edges from a gray-level image is used as an example for illustration purpose. The input images are first transformed into binary images and then edges are detected. Two more steps which can detect either the vertical edge and diagonal edge follow. Finally, a logic-AND operation is performed.

A sample C-like high-level language program is written as:

```
Program SAMPLE: Edge Detection
/* Declaration of variable */
        AIMG    M1;             /* gray-level image */
        DIMG    M2,M3,M4,M5,M6; /* binary value images */

/* AVERAGE,EDGE,VEDGE,DEDGE,LAND are pre-defined functions.*/
Begin
        INIT;
        LOAD(M1);
        M2 = AVERAGE(M1);
        M3 = EDGE(M2);
        M4 = VEDGE(M3);
        M5 = DEDGE(M3);
        M6 = LAND(M4,M5);
        OUT(M6);
End
```

The assembly code which was translated from the previous C-like program by the compiler is shown in Fig. D.2 [D.2]. Those words that are listed behind the ';' sign are the comments. Table D.1 lists the assembly instructions with a brief explanation.

The assembly language code is further translated into the machine code in order to be uploaded to array processor hardware. The code generated from the above program will look like the content of Fig. D.3 except those words after the '!' sign

```
BEGIN          ; PROGRAM START
; LOAD TEMPLATE INFOMATION
SELAPR 0       ; LOAD TEMPLATE 0 INFORMATION TO APR 0
LDAPR 0
SELAPR 1       ; LOAD TEMPLATE 1 INFORMATION TO APR 1
LDAPR 1
SELAPR 2       ; LOAD TEMPLATE 2 INFORMATION TO APR 2
LDAPR 2
SELAPR 3       ; LOAD TEMPLATE 3 INFORMATION TO APR 3
LDAPR 3
RESET          ; RESET LOCAL NUCLEU
INPUT          ; INPUT SAMPLE AND HOLD
; GET AVERAGE BLACK AND WHITE IMAGE FROM GRAYSCALE IMAGE
TEMP 0         ; SELECT TEMPLATE FROM APR 0
CNN            ; CNN TRANSITION
STO4 0         ; STORE THE OUTPUT IN LAM4(0)
FBACK 0        ; FEEBACK LAM4(0) TO THE INPUT/INITIAL STATE
; GET EDGE-DETECTED IMAGE
TEMP 1         ; SELECT TEMPLATE FROM APR 1
CNN            ; CNN TRANSITION
STO4 1         ; STORE THE OUTPUT IN LAM4(1)
FBACK 1        ; FEEBACK LAM4(1) TO THE INPUT/INITIAL STATE
; DETECT VERTICAL EDGE
TEMP 2         ; SELECT TEMPLATE FROM APR 2
CNN            ; CNN TRANSITION
STL 0          ; STORE LAOU TO LLM(0)
FBACK 1        ; FEEBACK LAM4(1) TO THE INPUT/INITIAL STATE
; DETECT DIAGONAL EDGE
TEMP 3         ; SELECT TEMPLATE FROM APR 3
CNN            ; CNN TRANSITION
STL 1          ; STORE LAOU TO LLM(1)
; AND OPERATION OF THE PREVIOUS TWO RESULTS
LLM 0          ; ACTIVATE THE LLM(0) TO THE INPUT OF LOGIC FUNCTION
LLM 1          ; ACTIVATE THE LLM(1) TO THE INPUT OF LOGIC FUNCTION
LDAND          ; LOAD THE AND FUCNTION TO THE LLU
LOUT           ; SEND THE LOGIC FUNCTION RESULT TO THE OUTPUT LINE
LDEA 0         ; DEACTIVATE LLM(0)
LDEA 1         ; DEACTIVATE LLM(1)
 END ; PROGRAM TERMINATE AND SENDS OUTPUT READY SIGNAL
```

Figure D.2: Translated assembly language of the sample program.

Table D.1: Assembly instruction Set.

Analogic Instruction		Mnemonic Instruction	Explanations	Equivalent SCR Configuration
First 4 bts	Last 4 bits			
0000	0000	RESET	reset local nucleus	SEL(S0)
0001	0000	INPUT	input sample and hold	SEL(S1)
0010	0000	CNN	start CNN transition	SEL(S2)
0011	X	STO4	store the output from LAOU to LAM4(X)	SEL(S3)
0100	Y	STL	store the output from LAOU to LLM(Y)	SEL(S4)
0101	X	FBACK	feedback LAM4(X) to the input/initial state	SEL(S5)
0110	0000	BEGIN	program begin	BEGIN
0111	Z	TEMP	select template from APR(Z)	SEL(TEM)
1000	W	LAND,LOR, LNOT	select the desired logic function	SEL(S6)
1001	Y	LLM	activate LLM(Y) to the input of the logic function	
1010	0000	LOUT	send the logic function result to the output line	
1011	Y	LDEA	deactivate LLM(Y) to the input of the logic function	
1100	0000		undefined	
1101	Z	SELAPR	select APR(Z) to load the CNN template info	External Setup
1110	0000	LDAPR	load template info to the activated APR	
1111	0000	END	program terminates and sends OUTPUT READY signal	END

[D.2]. The corresponding assembly program is also shown after the '!' sign. The SCR configuration which was suggested in [D.1] is also listed in Table D.1 [D.2].

```
0110    0000    ! BEGIN
1101    0000    ! SELAPR 0
1110    0000    ! LDAPR 0
1101    0001    ! SELAPR 1
1110    0001    ! LDAPR 1
1101    0002    ! SELAPR 2
1110    0002    ! LDAPR 2
1101    0003    ! SELAPR 3
1110    0003    ! LDAPR 3
0000    0000    ! RESET
0001    0000    ! INPUT
0111    0000    ! TEMP 0
0010    0000    ! CNN
0011    0000    ! STO4 0
0101    0000    ! FBACK 0
0111    0001    ! TEMP 1
0010    0000    ! CNN
0011    0001    ! STO4 1
0101    0001    ! FBACK 1
0111    0002    ! TEMP 2
0010    0000    ! CNN
0100    0000    ! STL 0
0101    0001    ! FBACK 1
0111    0003    ! TEMP 3
0010    0000    ! CNN
0100    0001    ! STL 1
1001    0000    ! LLM 0
1001    0001    ! LLM 1
1000    0000    ! LDAND
1010    0000    ! LOUT
1011    0000    ! LDEA 0
1011    0001    ! LDEA 1
1111    0000    ! END
```

Figure D.3: Translated machine code of the sample program.

Reference List

[D.1] T. Roska, L. Chua, "The CNN Universal Machine - An Analogic Array Computer," *IEEE Trans. Circuits and Systems, II*, vol. 40, pp. 163-173, Mar. 1993.

[D.2] B. J. Sheu, J. Choi, *Neural Information Processing and VLSI*, Kluwer Academic Publishers: Boston, MA, Jan. 1995.

Appendix E

A Compact Low-Power VLSI Transceiver for Wireless Communication

A 3-V CMOS VLSI for dual-mode wireless communication systems has been designed and fabricated using the MOSIS CMOS technology. By using mixed analog and digital CMOS design technologies, a single chip solution to baseband processing of data and supervisory audio tone signals in the analog transmission mode is possible. Key analog circuits include an anti-alias filter, two 5th-order low-pass filters, one 6th-order band-pass filter, an interpolator for sampling rate conversion, and two comparators. The digital modules perform data transmission and reception, error coding and decoding, as well as tone detection and regeneration. When implemented in the 2-μm CMOS technology from the MOSIS Service, the transceiver chip consumes less than 6 mW at receive-only mode. It is also quite suitable for battery-powered devices, such as portable terminals. Design technologies can be applied to future high-speed wireless transceiver design.

E.1 Hardware Architecture

For establishing high-capacity land mobile telephone and data transmission systems, economical and high-performance mobile radio units have been developed and used

in many cities over the world. New digital technologies have been employed to accommodate higher capacity by increasing the efficiency of frequency resources and the rapid trends in integrated data communication networks for computer and multimedia applications. In Europe, the pan-European Groupe Special Mobile (GSM) [E.1] system has been established for voice and data communications. The dual-mode, time-division multiple access (TDMA) [E.2] - [E.4] scheme has been adopted in the north America for the second-generation personal communication systems. By assigning multiple time slots in a single frequency channel, the channel capacity has increased by many-folds and the compatibility with conventional analog system is maintained. Both systems employ complex speech compression and sophisticated digital modulation methods to accommodate high-quality voice and data communications with the minimal frequency bandwidth. In such systems, low-power and high-performance VLSI's play an important role in transmitting, receiving signals and correcting data errors occurred during the transmission via multi-path channels.

The block diagram of a data transceiver and its function in wireless communications are shown in Figures E.1 and E.2. The modulation technique employed is a noncoherent binary frequency-shift keying (BFSK) in which the frequency of modulated carrier switches between two frequencies. Binary data is Manchester-encoded [E.2] for the purpose of embedding timing information into the data signal. The transceiver receives data streams from a FM discriminator, detects data using recovered data clock, and performs error detection and correction if it occurs. In transmit unit, the transceiver generates predefined streams of data by adding the parity bits of message and synchronization sequences. In addition, received supervisory audio tone (SAT) is regenerated by a phase-locked loop and re-transmitted. In the stand-by mode, only circuits for data reception are enabled to minimize power consumption.

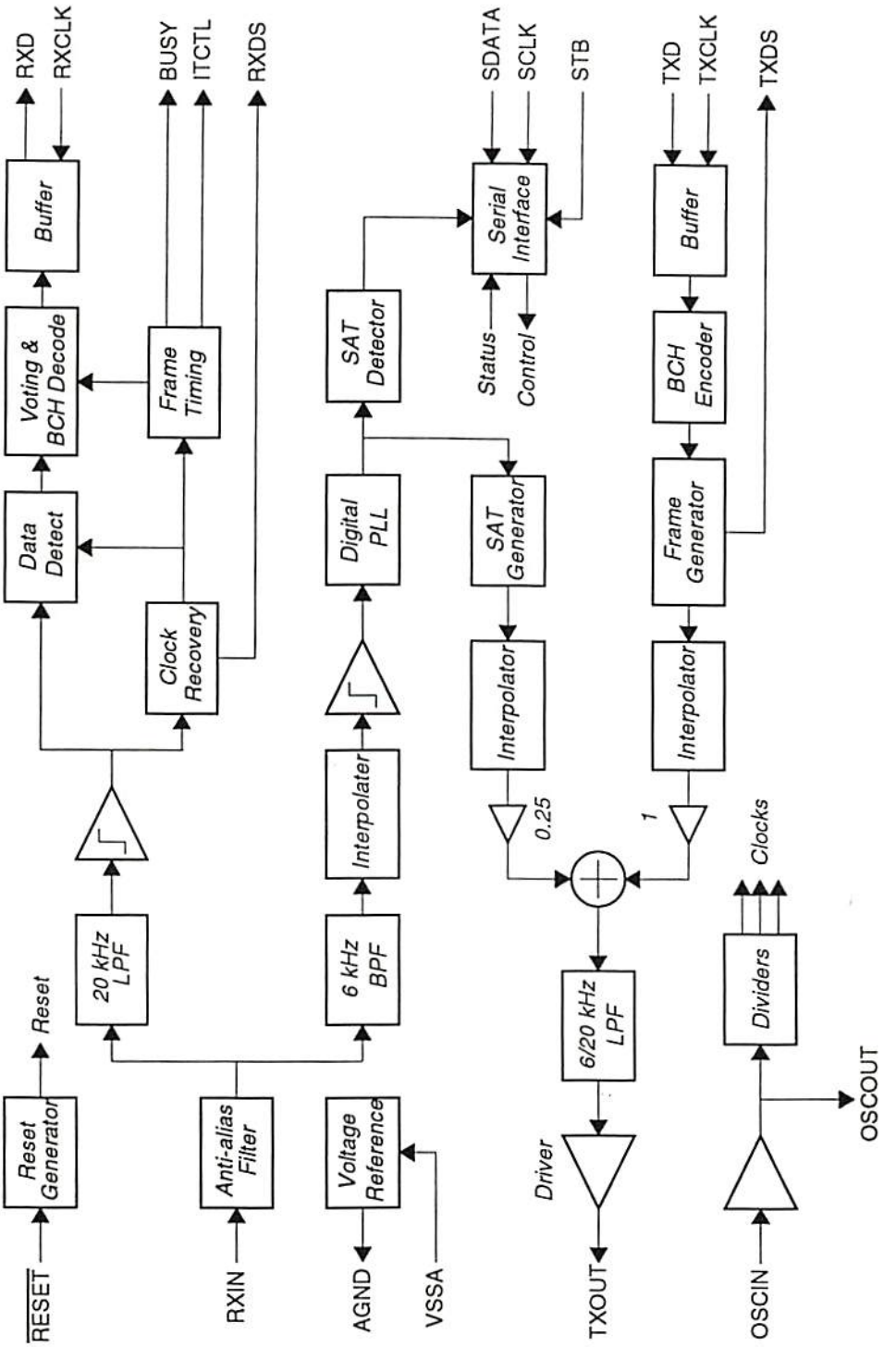


Figure E.1: Block diagram of Manchester-data transceiver.

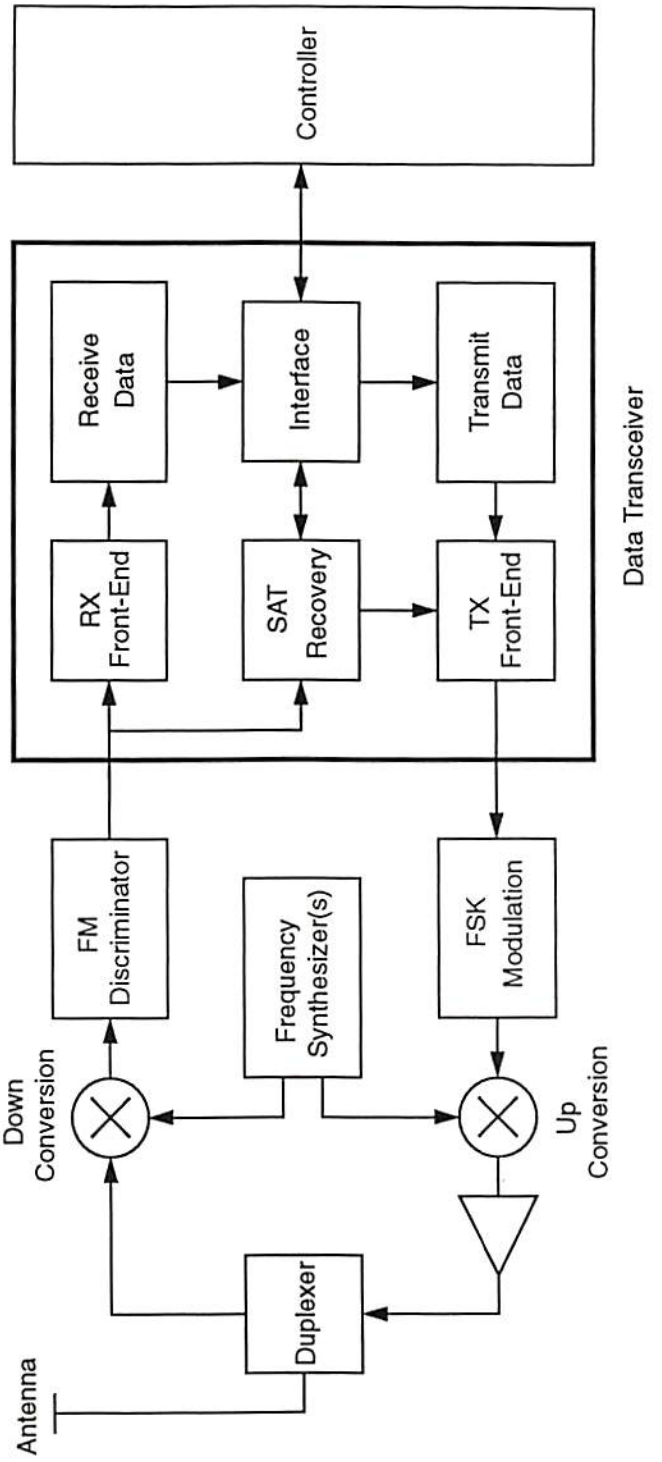


Figure E.2: Data transceiver VLSI chip in wireless communication systems.

E.1.1 Analog Front-End

The analog front-end for incoming signals contains an anti-aliasing filter, 20 kHz low-pass filter for data signal, and 6 kHz band-pass filter and interpolator for SAT signal. The anti-aliasing filter is a continuous-time active filter. All other analog circuits are based on the switched-capacitor (SC) technologies. The low-pass filter is a 5-th order Butterworth filter and is intended to remove all unwanted signal components above 20 kHz. Note that when a 10 kbps non-return-to-zero (NRZ) data stream is encoded by the Manchester encoding rule, i.e., low-to-high transition for logic-1 and high-to-low for logic-0, most of the signal energy is contained in the frequencies below 20 kHz. The output of the low-pass filter is then converted to NRZ stream by a comparator following the low-pass filter. The band-pass filter for SAT signal has a 200 Hz pass-band centered around 6 kHz. It removes voice-band signals as well as noise. Typically the N-path band-pass filter [E.5] is preferred in narrow band filtering applications. But the clock frequency requirement does not provide the compatibility with other building blocks. In this design, the filter is realized by a cascade of three biquads. To reduce the dynamic power dissipation in the filters, the clock frequency f_C chosen to be 300 kHz for both filters. When the filtered signal is translated to logic levels, the minimum duration that can be resolved by digital circuits is $1/f_C \approx 3.3\mu S$, for which it may be difficult to discriminate the frequencies of SAT signal in high accuracy. To improve the resolution, a linear interpolator driven by a 1.2 MHz clock is used to increase the sampling frequency by a factor of four. Please note that the interpolator can be implemented with a single operational amplifier instead of five amplifiers in the low-pass filter. A significant saving in the static power dissipation and the dynamic power associated with many capacitors in the filter has been achieved. The same comparator for processing the data signal is

also used to convert the received SAT signal, which is a single frequency tone plus noise, to square wave. The comparator acts like a zero-crossing detector or limiting device.

Analog postprocessing circuits for transmit data and SAT signals are also provided to minimize the number of external components. The Manchester-encoded transmit data from digital circuits contains high frequency components above 20 kHz. The transmit filter which has the same specifications as one in the receive section, removes these components for bandlimited FSK modulation. In the conversation mode, however, the clock frequency of the transmit low-pass filter is switched to 100 kHz, for which the cut-off frequency becomes one third of 20 kHz ($\approx 6.67kHz$). In this mode, the filter can effectively remove all harmonic components from the transmit SAT and retain the fundamental component, which is one of the 5970, 6000, 6030 Hz tones. In the summing amplifier which precedes the filter, two signals are properly scaled such that the modulation requirements are satisfied by using a single external device for level adjustment. In the stand-by mode, both the analog and digital transmit circuits are activated only during the transmission, which occurs intermittently.

E.1.2 Digital Transceiver Units

When the receiver operates asynchronously with the transmitter, the time instances at which the data is transmitted must be recovered before making decisions on data bits. As the Manchester-encoded data signal also conveys the clock information through level transitions, it can be readily regenerated using a digital phase-locked loop (DPLL). After the Manchester encoding, signal transitions occur between two consecutive data bits even when they have the same logic level. Therefore, these transitions must be removed first from the input before it is sent to the DPLL.

After the symbol timing is obtained from the clock recovery circuit, the Manchester decoding and data detection are performed. The data detection is based on the digital approximation of integrated-and-dump filter and may result in a sub-optimum performance over white Gaussian noise. Frame synchronization, an 11-bit Barker sequence [E.6], and other control bits are extracted afterward for use in the error correction unit. To cope with the deep signal fading, which occurs often during transmission in an urban area, two-fold error protection schemes, the majority voting and BCH coding, are employed in the system. After 5 repetitions of a data block is received, a bit-wise 3-out-of-5 majority voting is performed. Two consecutive repetitions are 8.8 mS apart. Therefore the information is not affected by a complete signal loss due to fading for a maximum period of 17.6 mS. In the BCH decoder stage, a single bit error can be corrected, based on a successive evaluation of the syndromes. If the error occurs in two or more bits, an uncorrectable error flag is sent outside the chip through the serial interface. In this design, the BCH decoder is capable of correcting many combinations of two-bit error patterns, thus improving the performance.

A message to be transmitted is pushed into the TX data buffer which is driven by the external serial data and clock. Upon the receipt of the first bit, the clock signals for analog and digital transmit circuits are activated. Notice that analog circuits must be stabilized well within the minimum duration of the transfer. When the transfer of the message is completed, the parity of the information is generated and concatenated to form a BCH-encoded message. While the message is being sent out for transmission, the information part is stored in a circular buffer as well to accommodate multiple transmissions for majority voting scheme. This arrangement frees up the TX data buffer for the next message. After the message is transmitted following two sequences for synchronization, it repeats ten more times. After

all necessary transmissions, the clock signals for transmit circuits are de-activated. In the conversation mode the analog circuits are turned on all the time for SAT transmission, and are not affected by the intermittent data transmissions.

The received noisy SAT signal is regenerated through a DPLL and transmitted back to the transmitting station for the channel monitoring purpose. Furthermore, at every 250 mS its frequency as well as the validity must be determined and compared with one contained in the received message. If two values do not match or it is not a valid SAT, the conversation shall be suspended temporarily. The DPLL is very similar to that for data timing recovery. However a second-order structure is needed. In a first-order DPLL, the output frequency will be floating over the whole tracking range when no valid signal is present. Thus it is likely that wrong decisions can be made. On the other hand, a second-order loop always pulls its output to one of two extreme frequencies in this case. The regenerated SAT signal is forwarded to analog circuits for re-transmission and to digital circuits for frequency discrimination.

E.2 Analog Front-End

Figure E.3 shows the block diagram of the analog front-end for the data transceiver chip. It consists of the receive module which interfaces the digital signal processing block through the comparators and the transmitter module which receives the output signals from the digital function block to generate the output waveforms.

E.2.1 Operational Amplifiers

Figure E.4 shows the circuit schematic diagram of the operational amplifier used throughout the design of the switched-capacitor circuits. By avoiding the cascode scheme of the transistors, a 2-stage amplifier is chosen for achieving large

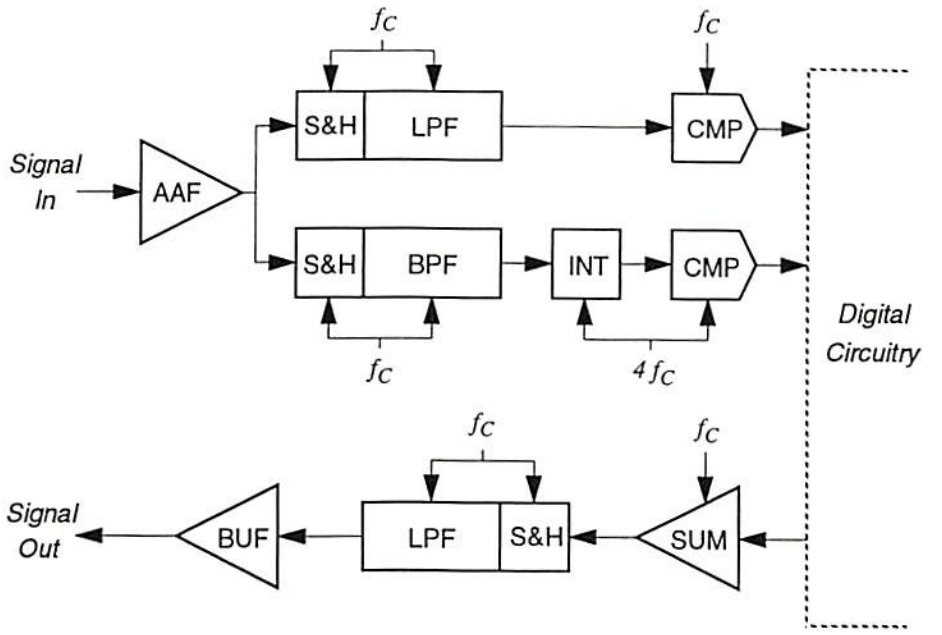


Figure E.3: Analog front-end.

input/output ranges with a single +3V power supply voltage. In order to reduce the standby current of the output stage, the class-AB output stage is used.

The simulated and measured results are summarized in Table E.1. SPICE simulation results show the greatly reduction of power dissipation. When the sampling frequency is 300 kHz and the 2-phase clocking scheme is used, the operational amplifier has to be settled down with 0.1% error during the clock pulse width of 1.67 μsec. The power dissipation of the operational amplifier is around 0.3 mW. Since the total load capacitance varies for each amplifier, the operational amplifiers are grouped into 3 types of different values of compensation capacitors (2.5 pF, 5 pF, and 10 pF) according to the load capacitance values to ensure operation stability.

E.2.2 Anti-aliasing Filter and Sample-and-Hold Circuit

The anti-aliasing filter is implemented with the active-RC approach. By using the passive resistance element which has a low sheet resistance value, a large resistor

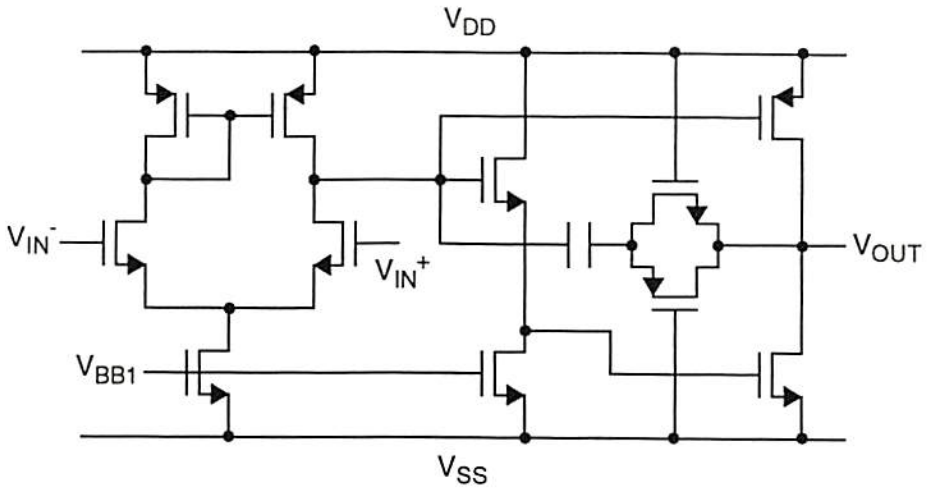


Figure E.4: 2-stage CMOS operational amplifier with class-AB output stage.

Table E.1: Simulated and measured characteristics of operational amplifier.

	Simulated Values	Measured Values	Unit
DC Gain	69	65	dB
Unity-Gain Frequency	12	12	MHz
Input Range	0.7-2.8	0.3-2.44	V
Output Range	0.1-2.9	0.14-2.66	V
Offset	0.069	6.2	mV
Slew Rate	22	20	V/μsec
Settling Time	150	170	nsec
Power	0.36	0.3	mW

Vdd=+3V, CL=5pF

is made with a significant amount of distributed capacitance associated with it. This distributed capacitance is quite big to obtain the additional values and can achieve higher-order attenuation well above the passband frequency. Figure E.5 shows the simulated and measured frequency characteristics of the second-order Rauch filter [E.7]. Changes in the frequency characteristics due to process variations are also plotted in the figure. When the desired sheet resistance is denoted by R_{poly}, it is assumed that the value varies from $R_{poly}/2$ to $2R_{poly}$. It is clear that the imperfect characteristic of the passive component is not critical for anti-aliasing low-pass operation.

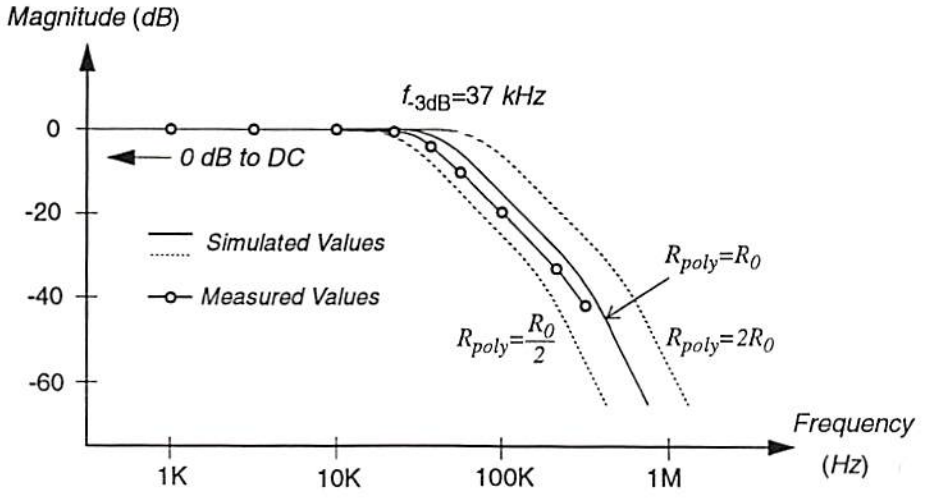


Figure E.5: Simulated and measured frequency characteristics of anti-aliasing filter.

For bilinear s-to-z transformation of the filters [E.8], the sample-and-hold circuit is employed. Two operational amplifiers are used in order to suppress their offset voltage, which appears at the output divided by the voltage gain of the operational amplifier. The sampling capacitance is chosen to be 5 pF with consideration of the speed and the noise performance requirement.

E.2.3 Switched-Capacitor Low-Pass and Band-Pass Filters

Figure E.6 shows the complete circuit schematic of the 5-th order Butterworth low-pass filter implemented with a switched-capacitor ladder circuit. Figure E.7 shows the 5-th order low-pass RLC filter prototype on which the design of the switched-capacitor is based. Since the passband ripple is not allowed, the Butterworth filter scheme is chosen. In order to achieve more exact sampled-data representation in z-domain, the bilinear s-to-z transformation is used. It requires four additional capacitors, which are calculated to be small values. The integrators are designed to be insensitive to parasitic capacitance. The sample-and-hold circuit is preceded by the filter. The sampling clock frequency is 300 kHz and the 3-dB cut-off frequency is 20 kHz. SWITCAP [E.9] simulation results of the frequency characteristics and the transient behavior are shown in Figure E.8.

The SAT signal from the anti-aliasing filter output passes through the band-pass filter in order to remove voice and unwanted noise signals. In our design, three biquad sections performing the band-pass filtering operation are cascaded to construct the 6-th order filter. Each biquad section has the same circuit topology, but has its own center frequency and the Q value so that the combined filter achieves the flat pass-band around the desired center frequency and the resultant Q factor. Figure E.9(a) shows the circuit schematic of a biquad SC section for the 6-th order band-pass filter. The SWITCAP simulation and measured results are shown in Figure E.9(b). The 3-dB bandwidth is equal to 200 Hz ($\pm 100Hz$) centered at 6 kHz.

E.2.4 Interpolator, Summing Amplifier, and Comparator

The circuit schematic of the switched-capacitor 1:4 linear interpolator is shown in Figure E.10. The frequency of two-phase clock (ϕ_1 and ϕ_2) is 1.2 MHz. The

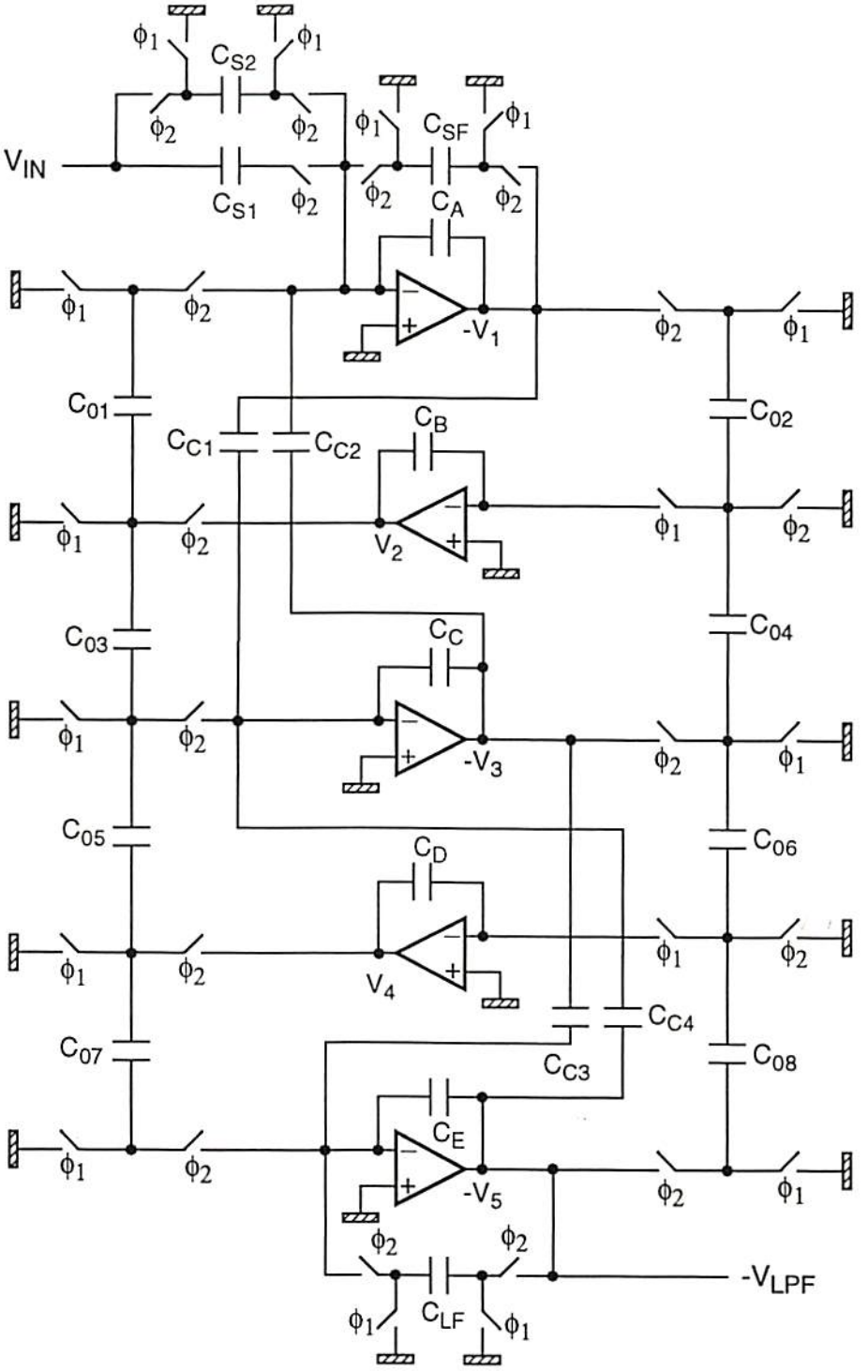


Figure E.6: Schematic diagram of a 5th-order low-pass filter.

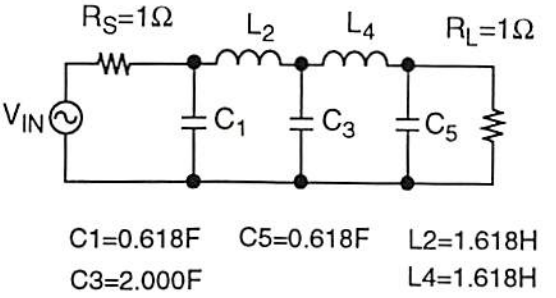


Figure E.7: LCR prototype filter of the 5th-order low-pass filter.

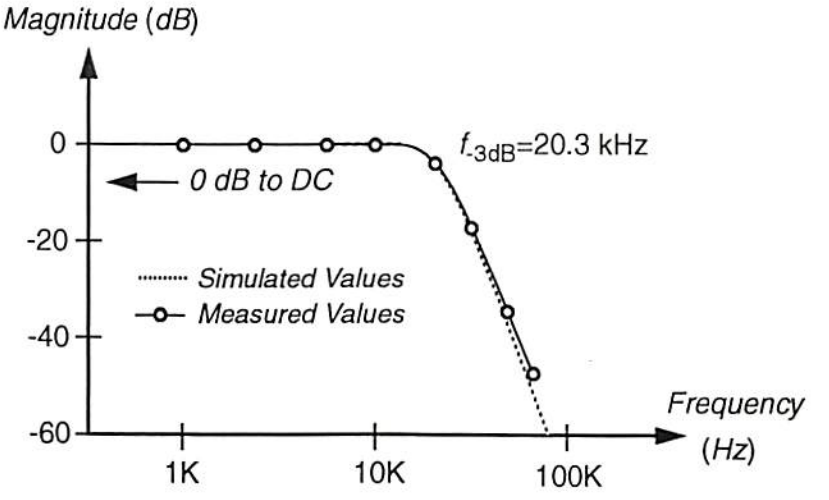
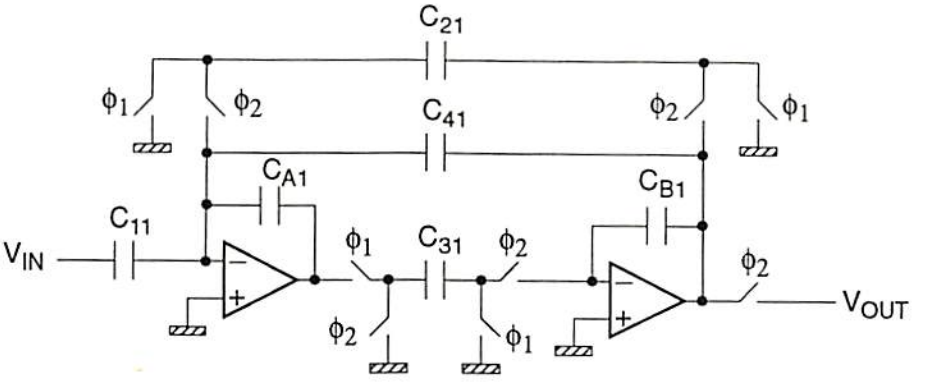
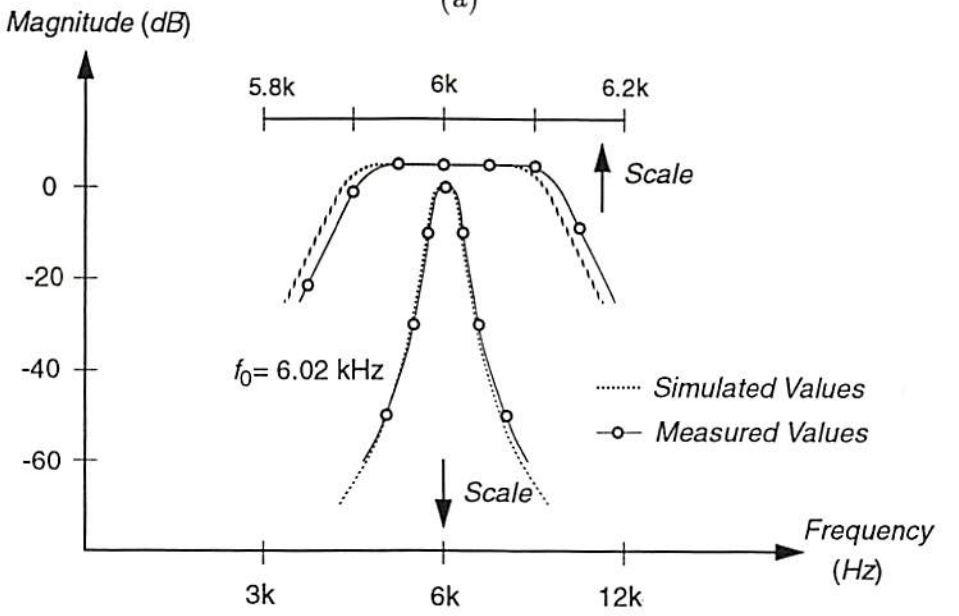


Figure E.8: Simulated and measured frequency characteristics of the 5th-order low-pass filter.



(a)



(b)

Figure E.9: 6th-order band-pass filter. (a) Biquad section of the filter. (b) Simulated and measured frequency response.

output value of the interpolator, V_{INT}, is sampled by ϕ_0. The frequency of ϕ_0 is 300 kHz. The difference between the applied input, V_{IN}, and the sampled output, V_{INT}, appears at the capacitor C_1. By choosing $C_2/C_1 = 4$, integration occurs during the next 4 cycles with each increment equal to one quarter of the provided difference.

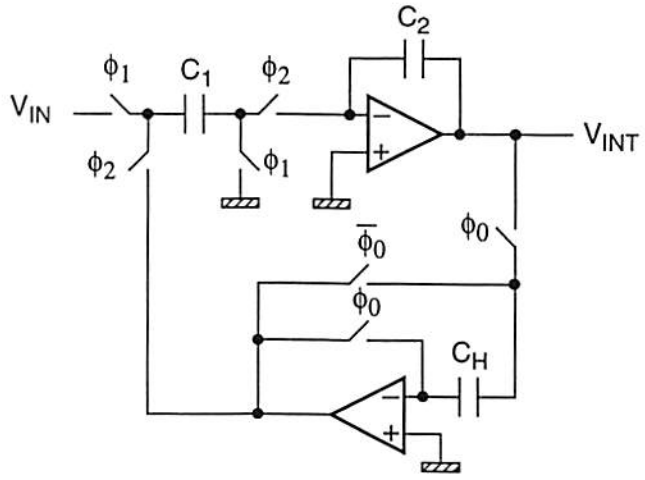


Figure E.10: Circuit schematic of 1-to-4 linear SC interpolator.

The voltage summing amplifier generates a step-type waveform with the transmit data and SAT signals. Since the output voltage is followed by the low-pass filter, its magnitude should be within the linear region of the operational amplifier. The V_{SS} to V_{DD} pulses of transmit data and SAT signals are converted to $(V_{DD} - V_{SS})/3$ to $2(V_{DD} - V_{SS})/3$ pulses by the reference voltage divider and the controlled switches. Since two voltage gains are required, the signal path is divided into two branches. One has a voltage gain of 1 and the other has a voltage gain of 1/4. When no input is activated, the analog ground voltage is applied to the circuit. The circuit is insensitive to the parasitic capacitance and the offset voltage of the operational amplifier is compensated.

Chopper inverter comparator [E.10] is employed at the final stage of the receive analog front-end to interface with the digital signal processing blocks. The operation is guaranteed at the supply voltage down to $V_{DD} = V_{tn} + |V_{tp}| + 2V_{nt}$ where V_{tn}, V_{tp} are

the threshold voltages of nMOS and pMOS transistors, respectively, and V_{nt} is the net control voltage of both nMOS and pMOS transistors. As shown in Figure E.3, it should operate at both the frequencies of $f_C = 300kHz$ and $4f_C = 1.2MHz$. The output of the comparator is stored in the D-flip flop which is edge-triggered by the delayed clock for obtaining the sufficient setup time.

E.3 Digital Circuits

E.3.1 Digital Phase-Locked Loop

Figure E.11 shows the digital data receiver with the clock recovery circuit using digital phase-locked loop (DPLL). The Manchester-encoded signal contains the 0-to-1 or 1-to-0 transitions, which occur at the middle point of each symbol. When this signal is applied to the DPLL circuit, the symbol clock can be recovered at the output. However, when two or more consecutive symbols have the same logic level, the signal also contains transitions between two symbols. The DPLL is preceded by the digital monostable circuit as shown in Figure E.11 for removing those transitions. The bit synchronization circuit is designed to detect the dotting sequence, which is contained in the data signal. The dotting sequence is repetition of the '10' pattern and provides a pure 5 kHz clock signal on a Manchester-encoded format. When it is detected, the DPLL is reset to the zero reference phase at the instance a signal transition occurs. This scheme can provide a fast initial operation of the DPLL without widening the bandwidth in the acquisition mode. The received signal is decoded to the non-return-to-zero (NRZ) signal through the Manchester decoder. The sum-and-reset circuit following the decoder approximates an analog integrate-and-dump filter, which has an optimum performance in white Gaussian noise environment.

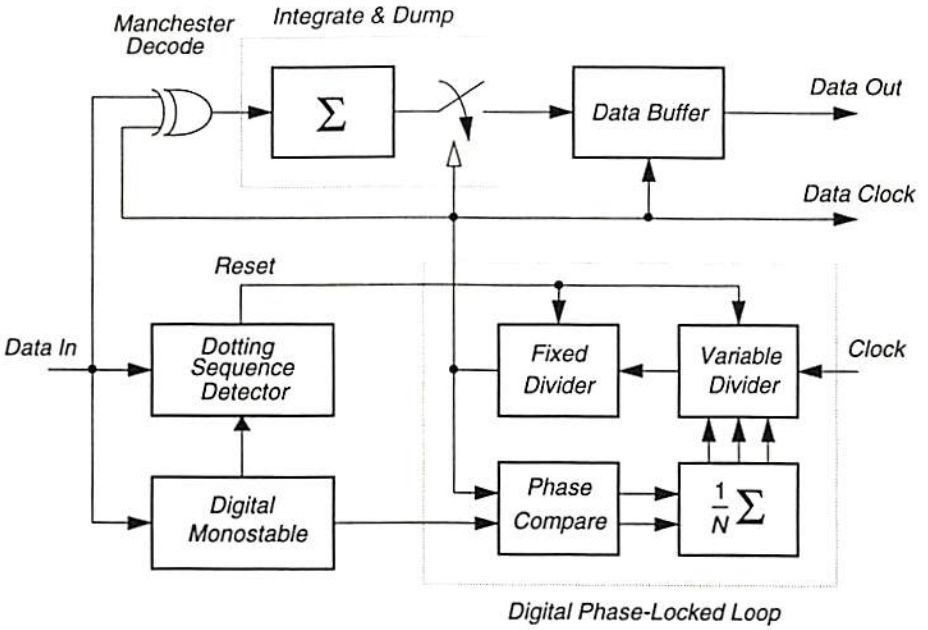


Figure E.11: All digital receiver for Manchester-encoded data.

The block diagram of the first-order DPLL circuit is shown in Figure E.12. Its tracking range is about 20 Hz at a center frequency of 10 kHz. Once it is initialized upon the detection of the dotting sequence, the reset line is disabled to prevent it from being re-initialized by successive pulses. A programmable up/down counter acts as a digital integrator and its output is attenuated by 4 in the combinational logic circuit. Attenuation in the feedback loop corresponds to reduction in the tracking range by the same factor as well as the reduction in noise level. The output of the combinational circuit is among -1, 0, +1, which correspond to 'advance', 'normal', and 'retard' in the diagram. If it is a nonzero value, then a single clock pulse is removed or added, to advance or retard the phase of recovered clock by a fixed amount. Please note that the up/down counter is preset to the zero state after any nonzero output (-1 or +1).

The acquisition time can be calculated as follows: Let $T_D, T_C(= 1/f_C)$, and N be the symbol timing interval to be recovered, master clock period, and the attenuation

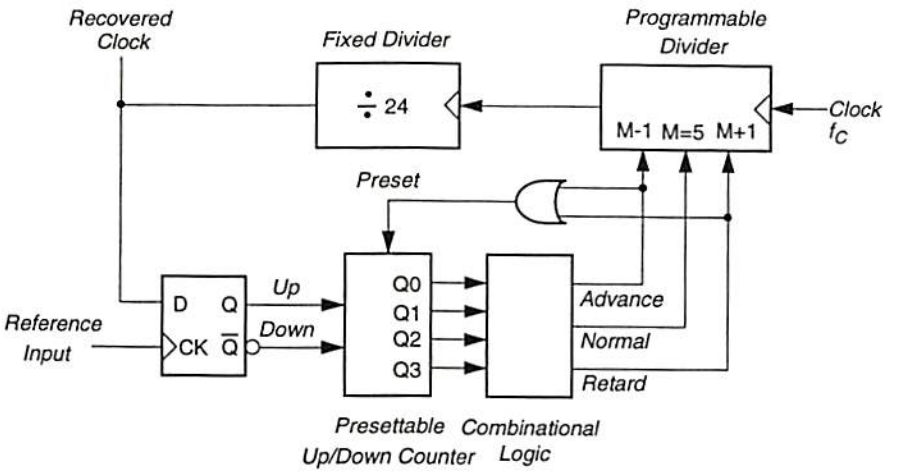


Figure E.12: First-order digital phase-locked loop for timing recovery.

factor, respectively. Assume that the initial phase error is $+\pi$ or -π, i.e., $+T_D/2$ or -$T_D/2$. Since it takes $N \cdot T_D$ seconds to retard or advance the output by T_C seconds, the time for the DPLL to acquire a complete synchronization is simply

$$T_{ACQ} = \frac{NT_D^2}{2T_C} = \frac{NT_D^2 f_C}{2} \qquad \text{(seconds)}. \tag{E.1}$$

When $f_C = 1/T_C = 1.2MHz$, $T_D = 100\mu S$, and $N = 4$, T_{ACQ} equals to 24 mS. If the input signal is noise-free, then the phase error lies in the range $-\pi/2 < \theta_E < +\pi/2$ and it provides correct decisions on the received data. The average acquisition time is one half of the maximum value determined by (E.1). Similar statements may apply when the DPLL begins to lose the synchronism due to noise or deep signal fading. Therefore, the DPLL is able to maintain the phase coherency for up to 12 mS during a deep fading which can cause a complete signal loss at the input. However, the input is purely random noise signal (positive or negative phase error with the same probability) during severe fading, the above-mentioned worst case phenomenon is almost unlikely to occur and the DPLL can withstand the fading of

much longer duration. The tracking range, defined as the range of frequency changes of recovered clock around the center frequency $1/T_D(Hz)$, is given as

$$f_{TR} = 2|\Delta f| = 2\left[\frac{1}{T_D} - \frac{1}{T_D + T_C/N}\right] \cong \frac{2}{NT_D^2 f_C} = \frac{1}{T_{ACQ}}. \quad \text{(E.2)}$$

For the values given above, $f_{TR} \approx 40Hz$, which means that the DPLL is able to track the input frequencies in the range $f = 10kHz \pm 20Hz$.

E.3.2 Majority Voting Circuit

In order to reduce data loss during transmission, 40-bit information is repeated five times, each being sandwiched between two repetitions of data. This scheme provides a good error protection for deep fading that spans up to $10 \sim 20\ mS$ long, and for additive random noise. In the receiver, four 40-bit buffers are required to hold received data temporarily. As the 5th repetition is received, a 3-out-of-5 majority voting is performed in bit-wise and the result is stored back to the first buffer. Each buffer consists of 40 static D-type flip-flops and is driven by a separate clock signal. During the 5th repetition interval, all buffers are clocked simultaneously. To reduce power consumption and required silicon area, chain-type D flip-flop is employed [E.11]. Slave (or master) latch shares its circuit between two adjacent master (or slave) latches in order to reduce the number of transistors almost by half. An additional transmission gate must be placed between two inverters so that one of them can switch its functions in each clock phase. Transistor sizes have to be carefully chosen to avoid the charge-sharing problem. The feedback switch is a single nMOS transistor with a small W/L ratio, such that it has a large time constant when passing logic-1 value.

E.3.3 Decoding of BCH Code

The Bose-Chaudhuri-Hocquenghem codes are a class of cyclic codes whose generator polynomials are chosen to make the minimum distance guaranteed by the lower bound large [E.12]. A (n, k)-BCH code has the following parameters

Block length	$: n = 2^m - 1$
Number of parity digits	$: n - k \leq mt$
Minimum distance	$: d \geq 2t + 1$

for any positive integers m and t. Clearly, this code is capable of correcting any combination of t or fewer errors in a block of $n = 2^m - 1$ digits. Decoding of a BCH code consists of the following steps;

Step 1) Calculate the syndrome $\mathbf{S} = [\mathbf{S}_1, \mathbf{S}_2, \cdots, \mathbf{S}_{2t}]$ from the received vector $\mathbf{r}(X) = r_0 + r_1 X + \cdots + r_{n-1} X^{n-1}$, where $\mathbf{S}_i = \mathbf{r}(\alpha^i), i = 1, 2, \cdots, 2t$, for a primitive element α of the Galois field $GF(2^m)$.

Step 2) Find the error location polynomial $\sigma(X)$ from $\mathbf{S}_1, \mathbf{S}_2, \cdots, \mathbf{S}_{2t}$.

Step 3) Determine the error location numbers $\beta_j, j = 1, 2, \cdots, t$, by finding the roots of $\sigma(X)$.

Step 2) is the most difficult part of decoding a BCH code and Berlekamp [E.13] has developed an iterative algorithm for finding the error location polynomial. From an implementation point of view, a programmable processor is quite suitable for the decoding algorithm. Thus with some sacrifice in error correctability, an error-trapping scheme is used for the decoding of a specific BCH code. An error-trapping decoder for cyclic codes is based on the following two theorems [E.12];

Theorem-1: If the syndrome of received vector $r(X)$ is taken to be the remainder after dividing $X^{n-k} r(X)$ by the generator polynomial $g(X)$, and all errors lie in

the highest-order $n-k$ symbols of $r(X)$, then the nonzero portion of error pattern appears in the corresponding positions of the syndrome.

Theorem-2: Let $s(X)$ denote the syndrome of $r(X)$. The syndrome of a cyclic shift of $r(X)$, that is, $X_r(X)$, is obtained by shifting the syndrome generator of $g(X)$ once with initial contents $s(X)$.

Once the syndrome of $X^{n-k}r(X)$ is obtained in the syndrome register, any error patterns with t or fewer errors confined within $n-k$ consecutive digits can be corrected by shifting the content of the register successively until the number of nonzero elements is equal to or less than t. If this situation does not happen until the k-th shift, an uncorrectable error has been detected. In a shortened version of BCH code by l digits, i.e., $(n-l, k-l)$-BCH code, it can be regarded as the original (n, k)-BCH code with l leading zeros. The first l zeros are always error-free and thus does not affect the syndrome calculations. In this case, the syndrome of $X^{n-k+1}r(X)$ is first calculated. The pre-multiplication by X^{n-k+1} can be accomplished by multiplying $r(X)$ by $f(X)$, which is the remainder after dividing X^{n-k+1} by the generator polynomial $g(X)$. If $X^{n-k+1}r(X) = s(X) + g(X)q_1(X)$ and $X^{n-k+1} = g(X)q_2(X) + f(X)$, then

$$\begin{aligned} f(X)r(X) &= [X^{n-k+1} + g(X)q_2(X)]r(X) \\ &= X^{n-k+1}r(X) + g(X)q_2(X)r(X) \\ &= [s(X) + g(X)q_1(X)] + g(X)q_2(X)r(X) \\ &= s(X) + g(X)q_3(X) \end{aligned} \tag{E.3}$$

Thus, $X^{n-k+1}r(X)$ and $f(X)r(X)$ generate the same remainder when divided by $g(X)$. The (40, 28)-BCH code is a shortened version of (63, 51)-BCH code with the following parameters;

a) with a minimum Hamming distance : 5

b) can detect 4 or fewer errors

c) can correct 2 or fewer random errors (t=2)

d) with a generator polynomial : $g(X) = 1 + X^3 + X^4 + X^5 + X^8 + X^{10} + X^{12}$.

Then $X^{n-k+1} = X^{35} = g(X)q_2(X) + f(X)$ where

$$q_2(X) = 1 + X + X^4 + X^7 + X^{10} + X^{11} + X^{12} + X^{14} + X^{16} + X^{17} + X^{21} + X^{23},$$
$$f(X) = 1 + X + X^3 + X^4 + X^6 + X^{10} + X^{11} \qquad \text{(E.4)}$$

The syndrome register that multiplies the received polynomial $r(X)$ by $f(X)$ and divides it by $g(X)$ is shown in Figure E.13. Received digits are shifted into both syndrome register which was set to zero initially, and the data buffer. After 40 shifts, the register holds $s(X)$, the syndrome of $X^{35}r(X)$. If the number of nonzero elements in $s(X)$ is less than or equal to 2, the feedback connection for division (SW2) is opened and the content of the syndrome register is added to that of the data buffer as they are shifted out of the register. Otherwise the content of the register with SW2 closed is shifted to the right by one bit to generate the syndrome of $X^{36}r(X)$. At the same time, the first bit in the data buffer is shifted out as a correct digit, because an error has not occurred at this position if it is correctable. The procedure repeats until the number of nonzero elements in the register is less than or equal to 2, which corresponds to correctable error patterns. If this situation does not happen until the 28th repetition, an uncorrectable error has been detected. Notice that, when a correctable error is detected, the error correction operation (SW2 opened, SW3 closed) continues until the final digit because the SR is cleared during at most $n - k = 12$ shifts and the tailing zeros have no effect. This error-trapping decoder is capable of correcting all one-digit errors and, two-digit errors

that are confined within any 12 consecutive digits. The complete BCH-decoder is shown in Figure E.13.

E.3.4 Threshold Logic

The threshold logic in the error-trapping decoder is a combinational logic circuit that generates logic-0 if the number of nonzero elements in the syndrome register is less than or equal to t, and logic-1 otherwise. In other words, if Q_i is the i-th output from the syndrome register, then

$$L_{th} = \begin{cases} 0 & \text{if } Q_{sum} = \sum_{k=0}^{n-k-1} Q_k \leq t, \\ 1 & \text{otherwise.} \end{cases} \tag{E.5}$$

Unless $t = 1$, the threshold logic can be very complex in order to accommodate all combinations of the event $Q_{sum} \leq t$. Figure E.14(a) shows a serial architecture that propagates local sums to the next stage. If $Q_{sum} > t$ at any stage, the remaining stages are 'don't care', and $L_{th} = 1$. Let $S_i^k, i = 1, 2, \cdots, t+1, k = 1, 2, \cdots, n-k$, denote the i-th local sum at the k-th stage. Then

$$S_i^k = \begin{cases} S_i^{k-1} & \text{if } Q_k = 0, \\ S_{i-1}^{k-1} & \text{if } Q_k = 1. \end{cases} \tag{E.6}$$

Here $S_0^k = 1$. If $S_i^j = 1, i = 1, 2, \cdots, t+1$, for some j, then $S_i^k = 1, k = j+1, \cdots, n-k$. Notice that if $S_{t+1}^k = 1$ for some $k = t+1, \cdots, n-k$, then $L_{th} = 1$. Thus $L_{th} = S_{t+1}^{n-k}$ and no additional decoding circuitry is required. The worst case propagation delay is $(n-k)T_m$ seconds, where T_m is the propagation delay in the multiplexer. If operating speed is the primary requirement in the circuit, the circuit shown in Figure E.14(b) may be used. This threshold circuit is based on the comparison of two currents set by the output of the syndrome register and reference. MOS transistors are sized such that current I_0 flows through each current source controlled by $Q_i, i = 1, 2, \cdots, n-k$ and the reference current is set to $(t + 0.5)I_0$. Thus

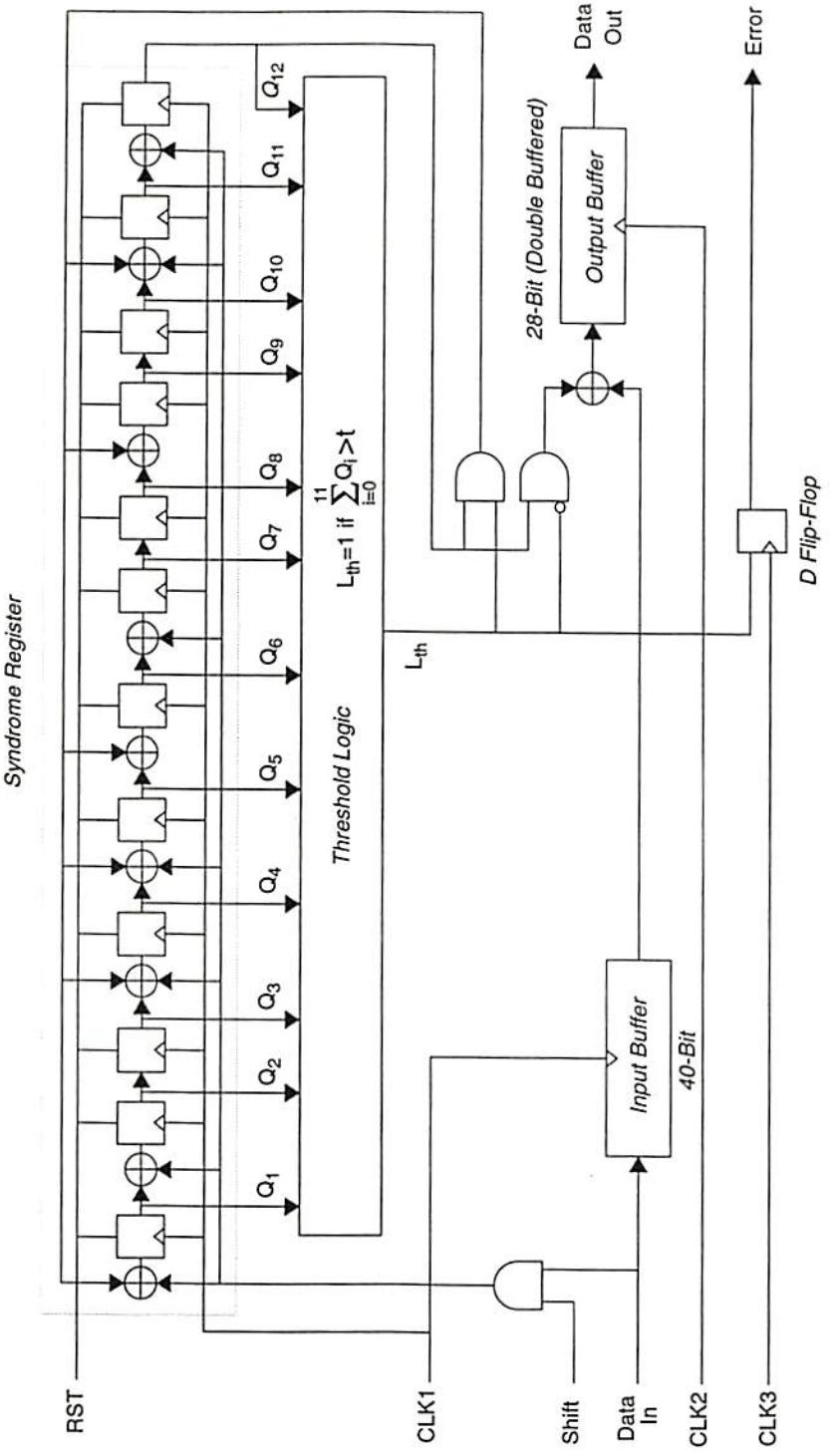


Figure E.13: Error-trapping decoder for (40,28)-BCH code.

the output of the current comparator is '0' if $Q_{sum} \leq t$, and '1' otherwise. Since all current sources are switched at the same time when $Q_i, i = 1, 2, \cdots, n-k$, is available, the propagation delay is now $T_I + T_C$ seconds, where T_I and T_C are the propagation delays in the current switch and current comparator, respectively. The complementary switches for current I_{dummy} are provided to improve the switching characteristics of current switches. For a large $(n-k)$ circuit, any mismatch in currents due to device parameter variations and geometrical distribution, may result in erroneous operation. In standard CMOS technologies, however, the accuracy of 8-bit $\sim$ 10-bit weighted-binary is readily achievable [E.14, E.15]. This corresponds to $n - k = 256 \sim 1024$.

E.4 Measurement Results

A low-power, 3V CMOS data transceiver chip for the dual-mode wireless mobile communication systems has been designed and fabricated using the 2-μm scalable CMOS technology from the MOSIS Service of USC/Information Sciences Institute at Marina Del Rey [E.16]. By using mixed analog/digital design technique, it provides a single-chip solution for the processing of data and supervisory audio tone signals. Key analog circuits include an anti-alias filter, two 5th-order low-pass filters, 6th-order band-pass filter, interpolator for sampling rate conversion, and comparators. Digital circuits performs data transmission and reception, error coding and decoding as well as tone detection and regeneration. In the SC filter blocks, the clocking switches are far separated from the analog signal components to avoid coupling from the digital switching noise. All capacitors are located above the ground-connected p-wells to absorb the undesired noise. A unit capacitance size is $20 \times 20\lambda^2$ which corresponds to 240 fF for the given technology. Table E.2 lists some measurement

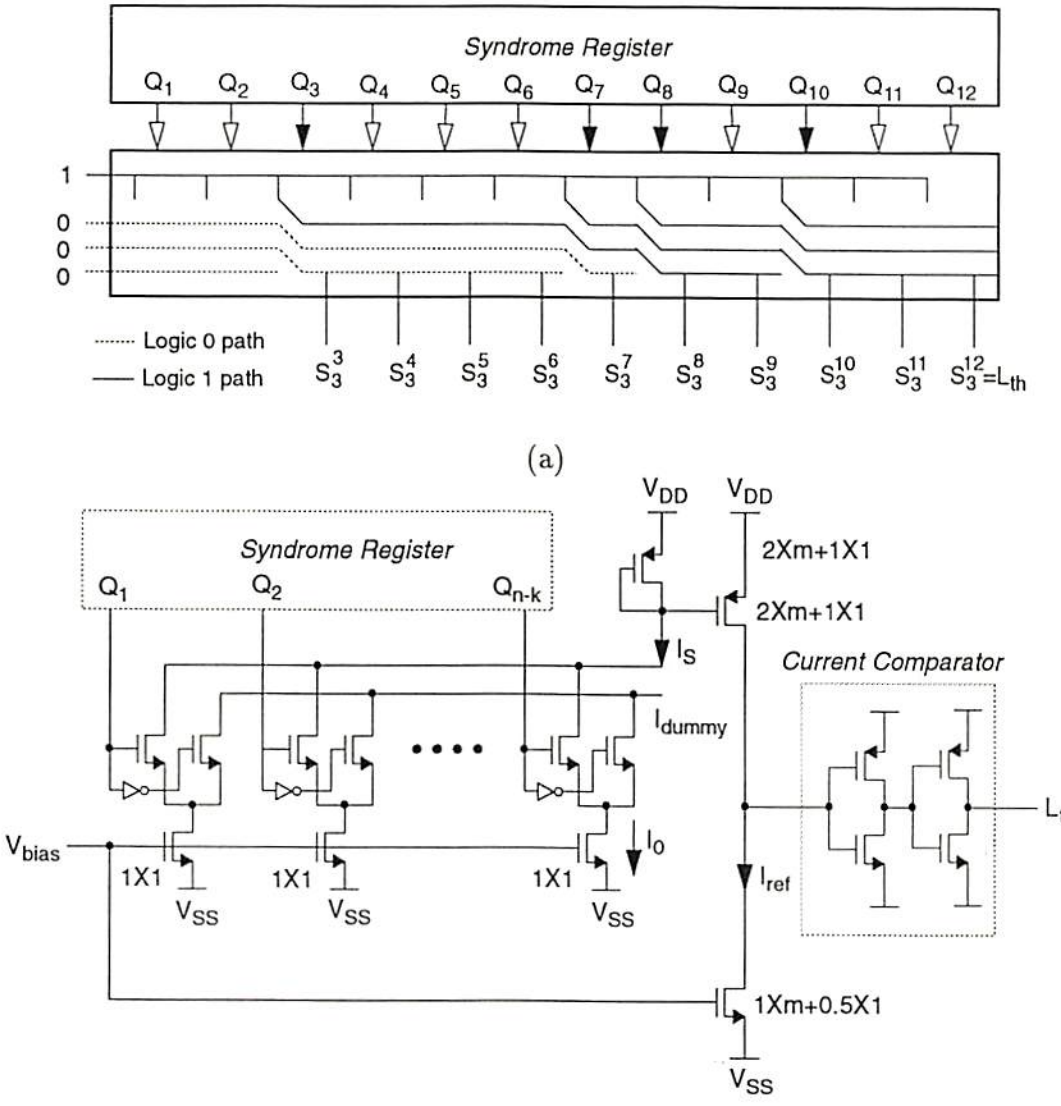


Figure E.14: Threshold logic circuits. (a) Propagation of partial sums (Q3 = Q7 = Q8 = Q10 = 1). (b) Current comparison.

results of the fabricated data transceiver chip. For each functional block, test vector was generated. The test vectors include typical patterns of signals for the verification of the chip. All tests were conducted with a bread-board containing the fabricated chip, EPROM which stores test vectors, and additional supporting logic gates from standard IC parts which generate the control timing. The designed chip has been successfully tested and it meets the specifications for the low-power wireless communication.

Table E.2: Measurement results.

Features	Results
Technology	2-μm CMOS Double-Poly, Double-Metal
Die Size	4.6 X 6.8 mm^2
Power Dissipation (receive only)	less than 6 mW
Power Dissipation (all active)	17.5 mW
Minimum Operating Supply Voltage	2.6 V

Reference List

[E.1] B. J. T. Mallinder, "An overview of the GSM system," *Proc. DCRC Conf., Hagen FRG*, pp. 1a/1-1a/13, Oct. 1988.

[E.2] G. A. Arredondo, J. C. Feggeler, and J. I. Smith, "Advanced Mobile Phone Service: Voice and data transmission," *Bell System Technology Journal*, Vol. 58, no. 1, pp. 97-122, Jan. 1979.

[E.3] The Electronic Industries Association, Cellular System Dual-Mode Mobile Station-Base Station Compatibility Standard, EIA/TIA/IS-54-B, Jan 1992.

[E.4] T. Habuak, et al, "A single-chip FM modem baseband CMOS LSI for land mobile telephone radio units," *IEEE J. Solid-State Circuits*, Vol. SC-20, no. 2, pp. 617-622, Apr. 1985.

[E.5] M. B. Ghaderi, J. A. Nossek, and G. C. Temes, "Narrow-band switched-capacitor band-pass filters," *IEEE Trans. on Circuits and Systems*, Vol. SC-29, no. 8, pp. 557-572, Aug. 1982.

[E.6] J. J. Spilker, Jr., *Digital Communications by Satellite*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1977.

[E.7] B.
K. Ahuja, "Implementation of active distributed RC anti-aliasing/smoothing filters," *IEEE J. of Solid-State Circuits*, Vol. SC-17, no. 6, pp. 1076-1080, Dec. 1982.

[E.8] R. Gregorian, G. C. Temes, *Analog MOS Integrated Circuits for Signal Processing*, John Wiley & Sons: New York, 1986.

[E.9] S. C. Fang, Y. P. Tsividis, O. Wing, "SWITCAP: A switched capacitor network analysis program," *IEEE Circuits and Systems Mag.*, Vol. 5, pp. 4-10, and 41-46, 1983.

[E.10] A. Matsuzawa, "Low voltage mixed analog/digital circuit design for portable equipment," *IEEE Symp. on VLSI Circuits*, pp. 49-54, Koyto, Japan, May 1993.

[E.11] N. Weste, K. Eshraghian, *Principles of CMOS VLSI Design: A Systems Perspective*, Addison-Wiley Publishing Co., Reading, MA, 1985.

[E.12] W. W. Peterson, E. J. Weldon, Jr., *Error-Correcting Codes*, MIT Press, Cambridge, MA, 1972.

[E.13] E. R. Berlekamp, *Algebraic Coding Theory*, McGraw-Hill, New York, 1968.

[E.14] T. Miki, et al., "An 80-MHz 8-bit CMOS D/A converter," *IEEE J. of Solid-State Circuits*, Vol. 21, no. 6, pp. 983-988, Dec. 1986.

[E.15] H. J. Schouwenaars, D. W. J. Groeneveld, H. A. H. Termeer, "A low-power stereo 16-bit CMOS D/A converter for digital audio," *IEEE J. Solid-State Circuits*, Vol. SC-23, no. 6, pp. 1290-1297, Dec. 1988.

[E.16] C. Tomovich, "MOSIS - A gateway to silicon," *IEEE Circuits Devices Mag.*, Vol. 4, no. 2, pp. 22-23, Mar. 1988.