

# **USCIPI REPORT #209**

**An Analog Edge Detection Chip and  
A Digital Neural Multiprocessor Chip  
for Machine Vision Processing**

**by**

**Hiroto Okada**

**June 1992**

**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.**

## Acknowledgment

I would like to express my sincere thanks to Professor Bing Sheu, who gave me this great opportunity to pursue research on artificial neural networks and vision processing, his continuous encouragement, and support throughout my M.S.-degree study. I would also like to extend my gratitude towards Professor Zhen Zhang for serving on my master thesis committee.

I gratefully acknowledge Professor Ramakant Nevatia, Director of the IRIS, who taught me on the machine vision processing. I am very grateful to Professor Hans Kuehl, Chairman of EE-Electrophysics Department; Professor Melvin A. Breuer, Chairman of EE-Systems Department; and Ms. Ramona Gordon, Senior Administrative Assistant in EE-Electrophysics Department, for providing me with the great research environment.

Interactions with Ph. D students in the VLSI Signal Processing Laboratory have been very useful. Valuable discussions with Mr. Joongho Choi on analog VLSI circuits has been most useful. I also thank Ms. Chia-Fen Chang for giving a lot of information of the digital multiprocessor design. Mr. Sudhir Gowda and Mr. Tzyh-Chiang Chen deserve my gratitude for advising on the computer simulation and laboratory measurement. Ms. Min Chen and Mr. Chen-Hao Chang help to manage the computing facility.

I also wish to thank many people in NKK Corporation to provide support for my study in the U.S.A., especially, Dr. Kazuo Kunioka, Mr. Kiyokastu Arai and Mr. Eiichi Makabe for their continuous encouragement and support.

Finally, I would like to thank my wife Ikuko Okada, and my parents, Ichiro and Misako Okada for their encouragement as well as the sacrifice through all these years.

## Table of Contents

<b>Chapter 1: Introduction</b> .....	1
<b>Chapter 2: Overview of Vision Processing and Neural Networks</b> .....	4
2.1 Overview of Vision Processing .....	4
2.2 Overview of Artificial Neural Networks .....	7
2.3 VLSI Implementation of Neural Networks .....	8
<b>Chapter 3: Compact System for Neural-Based Early Vision Processing</b> .....	12
<b>Chapter 4: The Analog Edge Detection Chip</b> .....	15
4.1 Edge Detection Algorithm .....	15
4.2 Hardware Implementation .....	16
4.2.1 Embedded Photosensor .....	16
4.2.2 Processing Cell .....	17
4.2.3 Sense-amplifier and I/O circuit .....	18
<b>Chapter 5: The Digital Multiprocessor Array</b> .....	20
5.1 System Architecture .....	20
5.2 Algorithm Mapping .....	21
5.2.1 Back-Propagation Neural Networks .....	21
5.2.2 Self-Organization Neural Networks .....	23
5.3 Processing Element Design .....	24
<b>Chapter 6: Conclusion</b> .....	26
<b>References</b> .....	27
<b>Figures</b> .....	30
<b>Tables</b> .....	45
Appendix I Publication by Hiroto Okada .....	46
Appendix II (Publication from "VLSI Signal Processing Laboratory")..	47

## List of Figures

- Fig. 1.1 System configuration for neural-based vision processing using an analog edge detection chip and a digital multi-processor chip.
- Fig. 2.1 Multiple level structure of vision processing.
- Fig. 2.2 Typical neural processing element model.
- Fig. 2.3 VLSI neuron model  
(a) analog implementation. (b) digital implementation.
- Fig. 3.1 System configuration for neural-based early vision processing using an analog edge detection chip and a digital multiprocessor array chip.
- Fig. 4.1 Early vision processing.
- Fig. 4.2 The 3 x 3 window.
- Fig. 4.3 Stencil of the convolution weight (3 x 3 Laplacian operator).
- Fig. 4.4 Computer simulation results of edge detection using the Laplacian operator.  
(a) girl image: Upper: Original image Bottom: Edge image  
(b) baboon: Upper: Original image Bottom: Edge image
- Fig. 4.5 Block diagram of the edge detection chip. The 50 x 66 cells, a decoder, 48 sense-amplifier, buffer registers, multiplexer, and output registers are included.
- Fig. 4.6 Photosensor using parasitic bipolar transistor in a p-well CMOS technology.
- Fig. 4.7 Current mirror circuit.
- Fig. 4.8 Circuit schematic of the basic cell.
- Fig. 4.9 Current mismatch improved circuit schematic.
- Fig. 4.10 Simulated characteristics of the basic cell and the cascoded cell.

**Fig. 4.11** SPICE simulation results of 8 x 8 array of cells.  
(a) vertical edge line.  
(b) diagonal edge line.

**Fig. 4.12** Microphotograph of the cell.

**Fig. 4.13** Schematic diagram of the sense-amplifier.

**Fig. 4.14** Detailed circuit schematic diagram of the operational amplifier.

**Fig. 5.1** Ring systolic array architecture.

**Fig. 5.2** Mesh-connected systolic array architecture.

**Fig. 5.3** Mapping of back-propagation algorithm onto the multi PE chip.

**Fig. 5.4** Block diagram of the digital processor.

**Fig. 5.5** Floor-plan of the single processor.

**Fig. 5.6** Requirements from the Grand Challenges on high-performance computing.

### **List of Tables**

**Table 2.1** Complexity of Algorithms in Early Vision Processing

**Table 2.2** Major Neural Network Models and Properties

## **Abstract**

A combination of an analog front-end chip and a digital neural multiprocessor array chip can play an important role in exploring and exploiting vision processing which demands tremendous parallel communications and computations. The analog front-end chip which integrates a photosensor array and pre-processing elements, and performs edge detection on an image captured in the photosensor array is presented. The nature of analog processing obtained in the analog edge detection chip can provide high parallelism for the whole system, eliminating sequential procedure caused by A/D conversion which is required in a digital processor. The photosensor are realized using parasitic bipolar transistors in a digital CMOS process. The edge detection are done by convolution using the 3 x 3 Laplacian operator. The edge detection chip consisting 50 x 66 cells can be implemented in a chip of 7.9 mm x 9.2 mm, using the MOSIS 2.0- $\mu$ m CMOS p-well technology. A digital multiprocessor array chip is also described. The digital multiprocessor array chip can construct 1-dimensional ring-connected systolic and 2-dimensional mesh-connected systolic array, which can effectively perform representative neural network algorithms such as the back-propagation network, and competitive learning. A systolic array, where communication and computation are well balanced, is a suitable architecture for neuro-computer because its features such as regularity and modularity are desirable for VLSI implementation. The digital multiprocessor array chip includes four-bidirectional buses connecting the four nearest cells and 256 words of 8-bit data cache for supporting the systolic array operation. Twenty processing elements have been implemented in an 19.3 mm x 20.9 mm chip, using the 0.5- $\mu$ m CMOS technology provided by TRW, Inc. and achieve 2 billion connection updates per second.

## **Chapter 1**

### **Introduction**

Machine vision processing is one of the most challenging tasks in intelligence systems, because in which massive computations, communications, and complex algorithms are required. In vision processing, the input data are usually in the analog format which is sensory-data sampled by an image sensor, and are represented as picture elements (pixels) of image matrices. (In a video rate high-resolution signal, 1024 x 1024 pixels per frame are produced.) These data are digitalized by A/D converters, and are further processed by digital processors. The primary goal of machine vision processing is to obtain the descriptive information of objects in the input image, where the output data are no longer of the form of pixels, but of symbols that correspond to knowledge models previously stored in the database. Machine vision processing consists of different kinds of computation which requires a unique representation of the data and different types of communication. To implement vision processing in a compact hardware system, the system should provide massive and fast parallel computation and communication capabilities, and many specialized processing elements.

Besides the hardware requirements, constructing software for vision processing may be an intractable task. When we construct a vision processing system using conventional microprocessors or digital signal processors, no matter whether the system is a single processor system or a multiple-processor system, software must be virtually perfect if it is to work properly. In the early development of machine vision processing, however, it might be that the efficient algorithms or knowledge rules are not well known. Some of the existing algorithms are too expensive, or difficult to develop the software implementation.

A new approach to the machine vision processing which does not require complicated algorithms or rules and can significantly reduce the complexity of software

that must be developed has recently become available [1]. Such an approach is based on the neurocomputing. Neurocomputing is the technological discipline concerning with parallel, distributed, and adaptive information processing. The primary information processing structures in neurocomputing are artificial neural networks. By definition, artificial neural networks provide massively parallel computation capability and adaptability or the learning capability which often eliminates the obligation of understanding of the algorithms or rules to be strictly specified in a conventionally programmed system. This salient feature makes artificial neural networks and neurocomputing very attractive to the information processing community, especially for machine vision processing. Figure 1.1 shows the schematic diagram of a neural-based system for advanced information processing. Artificial neural networks can be used to implement specialized pre-processors and post-processors, which have efficient interface with the real world. For example, consider the auto-vehicle navigation system. We can employ the neural network systems for a pre-processor and a post-processor. The neural network pre-processor can recognize the condition of the road by using pattern recognition and/or motion detection. The symbolic host computer can make decisions for the vehicle. For instance, to avoid an obstacle ahead the vehicle can use the information from the neural network processor. According to the decisions made by the host computer, the neural network post-processor at the action point which controls the wheels, the breaks and the engine, can find an appropriate degree for the wheels to around, the right timing to apply the break, and so on, to control the vehicle to avoid the obstacle.

There are two major implementation technologies of artificial neural networks: electronic and optical methods. The optical neural networks have a strong potential with more parallelism for communication and computation than the electronic ones. Due to the unique nature of optics, at present, the implementation technologies need further development for large-scale and real-world applications. Electronic method consisting of analog, digital, and mixed-mode scheme can use the state-of-the-art



very large scale integration (VLSI) technologies that have advanced rapidly and continuously. It is largely due to the advent of VLSI technologies that has made the neurocomputing practically possible. The neurocomputers constructed by VLSI technologies can provide key platforms to explore the machine vision processing.

In the following chapter, we will overview the studies on artificial vision processing and neural networks. Chapter 3 presents a compact system configuration for early vision processing, by using an analog edge detection chip and a digital neural-based multiprocessor chip. In Chapter 4, the analog edge detection chip is described in detail. Chapter 5 describes the digital neural multiprocessor array chip. Finally, Chapter 6 gives concluding remarks.

## Chapter 2

### Overview of Vision Processing and Artificial Neural Networks

#### 2.1 Overview of Vision processing

The primary goal of machine vision processing is interpretation and/or recognition of 3-dimensional (sometimes 2-dimensional) objects mapped onto the 2-dimensional images. For high-level interpretation, the principal unit of information is symbolic descriptions of an object or a set of image events extracted from the image. The description includes relationships both to features of object itself, i.e. other 2-dimensional symbolic tokens extracted from the sensory data, such as lines, regions, and surfaces and to other objects in the 3-dimensional scene [2]. High-level interpretation can only take place after a certain amount of low-level processing has been processed. The major low-level processing includes the classical image processing techniques such as edge detection, segmentation. At the low-level processing, the primary unit of information is the pixel, consisting of the color or intensity values of the image, and possibly range data for the visible surface element associated with each pixel. There is no simple computational transformation of the pixel-based image into symbolic description. It is known that many levels of representation and many stages of processing must take place to interpret a scene. The intermediate-level processing bridges the low-level processing and the high-level processing. At the intermediate-level, the basic unit of information is a description of an image event extracted from the image data. Figure 2.1 shows multi-level structure of machine vision processing.

**Low-level Processing:** Low-level processing consists mainly of operations on pixels and local neighborhoods of the pixels. Examples of the processing belong to this level are edge detection, edge linking, contour tracing, linear approximation, and segmentation. Some of these tasks are iconic, i.e. the input and output are based on

pixels. Since signal processing is pretty local, it can be easily achieved in a parallel machine. At the later stages of the low-level processing, functions such as linear approximation and segmentation, will produce symbolic results. The results are in a list format such as the list of edges forming the contours, or the list of straight lines that approximate the contours [3].

**Intermediate-level Processing:** This level is viewed as a bridge between the sensory data to the knowledge world. The information of the image, obtained in the low-level processing is reorganized to be suitable for recognition of the object in the high-level processing. Grouping, splitting, and labeling processes are involved. These processes are quite global. The data representation at this stage consists of 2-dimensional image events such as regions and lines, or 3-dimensional line segments. The 3-dimensional motion and depth information are key factors to perform matching the objects with the 3-dimensional objects. Pattern matching, such as Kanji character and fingerprint identification, is included in this level.

**High-level Processing:** High-level processing generates object descriptions which represent information about the 3-dimensional world in a representation that can be used to form symbolic identification of image events. The result of the high-level processing is symbolic representation of a particular object in an image which is matched with a knowledge model that was previously stored in the database. There are two methods to perform matching between the object and a knowledge model. One is data-directed, i.e. the bottom-up, method. It derives depth data from the image and use it in matching. The other one is knowledge-based, i.e. the top-down, method. In this method, prior knowledge of an object is used to verify interpretation of ambiguous image data. A symbolic LISP machine is a desirable processor for this level processing.

To implement vision processing that has the above-mentioned features in a hardware system, the system should provide [2] (1) the ability to process both pixel data and symbolic data, (2) fast and massive computation capability for pixel-level data, and (3) transformation capability of the data; from analog sensory data into digital image data, and from the image data into a set of meaningful symbols.

Consider the case when high-resolution video signals are input of some vision system using digital processor(s) in order to understand computation and communication complexity in early vision processing (low- and intermediate- level processing). There are 30 frames per second to be processed and each frame consists of 1024 x 1024 pixels. It leads the fact that 30 million processing operations and communications per second are required to process one operation on one pixel in the input images. Of course, even simple processing such as edge detection algorithm includes around ten instructions in one process. This example shows that early vision processing is quite computationally expensive. Table 2.1 [3] lists the complexity of some representative early vision processing algorithms. Notice that the number of integer and floating-point operations that takes place in the system for each algorithm, as mentioned before, are bigger than the complexity listed in Table 2.1 by one to three orders of magnitudes.

Various researchers have presented parallel implementations of vision processing ranging from individual modules for edge detection to the entire process of the machine vision processing. J. Little et. al from Massachusetts Institute of Technology use the Connection Machine for edge detection and the Hough transform [4]. S. Levitan et. al. from University of Massachusetts construct a parallel system to implement the entire process on the Image Understanding Architecture [2]. The other examples of parallel implementation can be found in [5,6,7]. Many of them employ the single-instruction multiple-data (SIMD) structure or multiple-instruction multiple-data (MIMD) structure, or combination of these structures such as SIMD-and-SIMD, SIMD-and-MIMD.

## 2.2 Overview of Artificial Neural Networks

An artificial neural network uses a parallel and distributed information processing structure consisting of processing elements, which are interconnected through the connection wires. The schematic diagram of a typical processing element inspired by the biological model in neurocomputing is shown in Fig. 2.2. The input signals  $x_1, x_2, \dots, x_n$  are provided to the processing element. The output signal is determined by

$$Z_{out} = f \left( w_0 + \sum_{i=0}^n w_{ji} x_i \right) \quad (1)$$

where  $w_{ji}$  is the synapse interconnection weight, and  $f(\cdot)$  denotes the transfer function of the neuron. The output signals can branch into as many collateral connections to other neural units.

The artificial neural network can be characterized by (1) Data transformation structures (Networks) and (2) Learning rules. The data transformation networks can be divided into two classes. One is the single-layer network, called associative network. In this class, linear associate networks which are associated with the Hebb's learning law [8] and the Widrow learning law [9] and recursive associate network such as the Hopfield network [10] are included. The other data transformation network is multi-layer data transformation structures. The back-propagation neural network [11], self-organization map [12], and counterpropagation network [13] are included. Table 2.2 shows popular networks and their properties [14].

The learning rules are strongly associated with the network structures. Learning is the process of self-adapting of the weights associated with the connections. The first popular leaning rule, named Hebb's learning rule [15], which shows that the neural networks can learn for a certain function, was presented in 1957. According to the Hebb's rule, if an input and output are activated at the same time, the weighting between the input and output is increased. This type of learning is called unsupervised learning, which does not require reference data, targeted output data. On the other hand, supervised learning requires the desired output data as a reference.

The simple delta rule is applied to adjust the weights, using the error between the desired output and actual network output. The back-propagation learning is the supervised learning. The root-mean-square error at the output layer propagates backward through the network and is used to update the weights between the layers. Many derivatives from the simple delta rule and the back-propagation network learning are used for efficient learning rules. The counterpropagation learning, which can be applied only to three-layer networks, combines the competitive learning [12] and the delta learning rule. Another type of learning rules is called reinforcement learning [16], that resides between the unsupervised learning and supervised learning. In this learning, an external observer guides the network, giving a response as to whether the network output is good or not. The learning rule of a Boltzmann machine [17] is based on the stochastic process, which is a simulated annealing technique.

### 2.3 VLSI Implementation of Neural Networks

Neural computing demands a tremendous quantity of computation and communication. Therefore, multiprocessors, array processors, or massively parallel processors are most appealing architectures for the neurocomputer design. For hardware implementation, we need to map the neural network onto array architectures. There are two kinds of approaches for the mapping: (1) the direct design approach, (2) indirect design approach. From the viewpoint of implementation technologies, there exist the analog implementation, the digital implementation, and the mixed-mode implementation. Figure 2.3 shows simple examples of implementations of the neural processing element shown in Fig. 2.2 in an analog and a digital approach, respectively. In Fig. 2.3 (a), the weights,  $w_{ji}$  are realized with resistors. The transfer function can be realized using an operational amplifier. The digital processor has a pipeline architecture which has been originally developed in the digital signal processor chip for signal processing applications. Loading two data from two registers to the multiplier, multiplying them and accumulating the products are executed in parallel by this pipelined

architecture.

**(1) Direct design:**

In the direct design, the neural computing architecture directly imitates the structure of artificial neural networks. Consequently, a large number of communications in the connectionist nets are supported directly by hardware, which tends to become a limiting factor of the maximum size of neural networks. Most existing dedicated neural network processors emphasize single chip implementation instead of the complete system construction.

Analog circuits can process more than 1 bit per transistor and provide very high speed processing, which compares favorably to digital circuits [18]. For real-time early vision processing, dedicated analog processing chips offer the most appealing alternative. The asynchronous updating properties of analog devices can provide extremely high speed computation which are qualitatively different from those of any digital computer [19]. The analog circuits will occupy an essential portion in electrical systems because the integration of analog sensors and neural-based pre-processing and post-processing is vital to many real-world applications.

Although the analog circuits are more attractive for the biological type neural networks, its suitability for the connectionist type network is very limited. Compared with digital circuits, analog circuits are more sensitive to noise, crosstalk, temperature effects, power supply variations and so on. Since the higher the precision the more chip area will be required, analog precision is usually limited to no more than 8 bits.

Many of the early work on the analog neural network implementation can be found in the text book [20] written by C. of Mead, California Institutes of Technology.

**(2) Indirect design:**

By indirect mapping, the original neural network structure can be converted into multiple processor architecture. A desirable architecture would have local interconnections only and the computations are performed in the pipeline fashion. To

achieve such a design, matrix-based mapping methodology is the most mature and effective tools for neural and other information processing applications. The tool can accommodate several key models, such as Hebbian, delta, competitive, and back-propagation learning rules in single-layer feedback networks and multi-layer feed-forward networks. This approach matches well with the design principle of VLSI system, which exploits highly regular, parallel, and pipeline architecture, and reduces the communication complexity.

One of the major concern in the indirect design approach is how to map a given algorithm onto the hardware array systematically and efficiently. Mapping strategies would define efficiency of the processing, complexity of communication of the data, and the number of iterations of computation. Especially communication with external memory is one of the most time-consuming tasks in the whole process. Including initialization of the local memory embedded in each processor, strategy of load and reload data between the local memory and the external memory should be carefully designed.

The following processor networks are the most common architectures in multiprocessor systems:

- (1) Common Bus Connection,
- (2) Linear Array,
- (3) Ring Array,
- (4) 2-dimensional Mesh Array, and
- (5) Hypercube Array.

From the viewpoint of control methodology, there are four categories in multiprocessor architecture;

- (1) Bus-oriented architecture,
- (2) SIMD architecture,
- (3) MIMD architecture, and
- (4) Pipeline architecture.



Bus-oriented architecture has been widely used for many years in the conventional multiprocessor architectures. This architecture uses the common bus (single-bus, multiple-bus) connection. Although high-speed bus can make the system very effective, the number of processors in a system will be limited to a certain number due to limitation of bandwidth of the bus. The MARK III and the MARK IV systems from TRW Inc. employ this architecture.

An SIMD machine is a parallel array of arithmetic processors with local memory, which has control buses and data buses. Instructions are broadcast from a host such that all processing elements executes the same instruction at the same time. One prominent example of the SIMD architecture is the Connection Machine by Thinking Machine, Inc.

In an MIMD machine, each processor can execute different instructions at the same time, communicating either using a shared memory or a message passing scheme. This architecture can offer very great architectural flexibility. Consequently, control of data and instruction become more complex and software to be developed will be complicated and difficult to design.

Typical pipeline array architectures are the systolic arrays and wavefront arrays. A systolic array system is a network of processors that rhythmically compute and pass data through the system [21]. Every processor regularly pumps data in and out, each time performing some short computation, in order that a regular flow of data is kept in the network [22]. Wavefront array is an asynchronous, data-driven computation array.

Examples of the pipeline array architecture are the WARP hardware from Carnegie-Mellon University, and the advanced WARP chip, iWARP, from Intel Inc.

## Chapter 3

### Compact System for Neural-Based Early Vision Processing

In the last chapter, we reviewed the machine vision processing and artificial neural networks. In those discussions, we found that a combination of an analog front-end chip which integrates a sensor and pre-processing elements and a digital neural-based multiprocessor array which provide massively parallel communication and computation can achieve extremely high efficiency as the pre-processing system for machine vision processing. A dedicated analog chip which integrates photosensors and associated circuitry can provide a compact and efficient interface between the real world and the signal processing system. A digital neural-based multiprocessor supports most essential features in the early vision processing, such as passively parallel computation, efficient communication, and simplification of complexity of the algorithms to be developed. The combination of these two chips can keep high parallelism of processing through the whole system.

In conventional early vision processing systems, the input image are sampled by an image sensor which can be based on solid-state CCD or MOS technologies, and represented in electrical signal which can be either the analog voltage or current values. The electrical signal usually follows some industrial standard such as the PAL format for video in order to simplify the interfacing to various systems, or it may simply be a raster scan readout scheme [23]. Analog-to-digital conversion and image normalization are conducted on the input signal before the main processing is performed. In real-time or high-speed processing, however, the A/D conversion stage is crucial. Impracticality of integration of an A/D converter and each signal generator in the image sensor requires the sequential digitalization of the signals generated. While parallel execution of different parts of the algorithm can be done using many processors, the sequential nature of data input will limit the degree of parallelism of the digital system to a level lower than that is achievable by the system with parallel

data generation and input. Only the dedicated analog processor that integrates image sensors and processing elements can eliminate the explicit A/D converter without leading to impractically low pixel density [24].

For the multiprocessor architecture, we choose the systolic array configuration, because its excellent properties such as modularity, regularity, local interconnection, and pipelining, are well suited to VLSI implementation of artificial neural networks. The systolic array architectures have the following key advantage [21]:

1. The exploitation of pipelining is very natural in regular and locally connected networks. It yields high throughput and simultaneously saves the cost associated with the communication.
2. It provides a good balance between the computation and communication, which is most critical to the effectiveness of array computing.
3. In order to support most of the connectionist models, the VLSI array architecture appears to be most viable.

Figure 3.1 shows the system configuration for neural-based early vision processing by using the analog edge detection chip and the digital neural multiprocessor chip [25]. The edge detection chip works as an image sensor as well as the pre-processor for edge detection. The output data are in the binary format, i.e. the output of a pixel which has an edge is with the logic-1 value, while the pixel which has no edge is with the logic-0 value. The output data are transferred in a group of 8 bits to the SRAM memory for buffering data. The binary format reduces the bandwidth of the communication bus and the required capacity of the SRAM. Compared with the conventional approach of having the input image signal converted by an A/D converter and is not processed for edge detection, the 8-bit bus to transfer the data and 8-bit cells in the SRAM to store are required for only one pixel. In our design, while, the 8-bit bus can carry 8-pixel data and one bit cell in the SRAM is enough to store one pixel data.

The digital multiprocessor array performs vision processing using neural network algorithm in systolic array fashion. Necessary data for computation, such as initial weight value and the target values, are loaded from the second SRAM module, while the input image data is fed from the first SRAM module. The second SRAM module also stores and re-loads the intermediate results which for multiprocessor array. The system controller can be the SPARC station which controls the procedures of the analog and digital chips and the transformation of data between the chips.

In the next two chapters, detailed design of the analog edge detection chip and the digital neural multiprocessor chip are described.

## Chapter 4

### The Analog Edge Detection Chip

Searching for an alternative analog processing array that does not require A/D conversion has attracted considerable attention [26, 27, 28, 29, 24, 30]. While some analog processing arrays are inspired by their biological counterparts [26, 28], the others are analog implementations of known algorithms that do not resemble any processing technique used in the biological neural system [29, 24, 30]. In both cases, analog hardware cannot be conveniently programmed as the digital processors, and is mostly hardwired. The circuit is constructed according to Kirchhoff's laws and the terminal characteristics of the components embody the desired algorithm. Although such circuit synthesis is guided by experience, and ingenuity, and the approach is ad hoc and limited in its generality. When successfully executed, it may offer a saving in power and enhancement in speed by orders of magnitude over the digital approach [31].

Edge detection is the first essential step in early vision processing as shown in the functional block diagram of Fig. 4.1. A number of algorithms for edge detection are available in the literature. Representative algorithms among them are the Sobel filter, the Median filter, and the Canny algorithm [32].

First the edge detection algorithm used in our chip is described. Then hardware implementation is detailed.

#### 4.1 Edge Detection Algorithm

Implementation of processing elements embedding with photo-sensors requires simple algorithms in order to increase pixels density. We choose the discrete Laplacian operator with a 3 x 3 window. Consider a 3 x 3 window of pixels shown in Fig. 4.2. The Laplacian can be approximated as follows [33]:

$$\frac{\partial^2 E}{\partial x^2} = \frac{1}{\epsilon^2} ( E_{i-1,j} - 2E_{i,j} + E_{i+1,j} ), \quad (1)$$

$$\frac{\partial^2 E}{\partial y^2} = \frac{1}{\epsilon^2} ( E_{i,j-1} - 2E_{i,j} + E_{i,j+1} ), \quad (2)$$

where  $E(x, y)$  represents the image brightness,  $x$  and  $y$  are coordinates, and  $\epsilon$  is the spacing between picture cell centers. Therefore,

$$\frac{\partial^2 E}{\partial x^2} + \frac{\partial^2 E}{\partial y^2} = \frac{1}{\epsilon^2} ( E_{i-1,j} + E_{i,j-1} + E_{i+1,j} + E_{i,j+1} - 4E_{i,j} ). \quad (3)$$

The result is zero in the region of constant brightness.

Such approximation to differential operators are used in the finite difference solution of partial difference equations. This convolutional weight can be represented by the stencil shown in Fig. 4.3. A stencil is a pattern of weights used in computing a convolution, which shows the spatial relationships between the places where the weights are applied. The stencil of Fig. 4.3 shows the Laplacian operation is equivalent to convolution of the sequence of functions that feature a central depression surrounded by a positive wall.

Figures 4.4 shows computer simulation results of edge detection using the 3 x 3 Laplacian operator. The input images are with 256 x 256 pixels and quantized in 8-bit gray levels.

## 4.2 Hardware Implementation

Our edge detection chip consists of 50 x 66 mesh-connected cells, 48 sense-amplifiers, a column decoder, and multiplexor, and output registers. Figure 4.5 shows the block diagram of the edge detection chip. Each cell has a photosensor and a processing element which computes the convolution.

### 4.2.1 Embedded Photosensor

An image focused on the chip surface is sampled by a matrix of embedded photosensors. The photosensor generates image current depending on intensity of the input light and supplies it into the processing element within the same cell. The

photosensor is economically fabricated by parasitic vertical bipolar transistors in a standard CMOS process, whose collector current is proportional to the light intensity incident on the collector junction along the p-well boundary, as shown in Fig. 4.6. In the room light environment, the output currents ranging from nano-amperes to tens of nano-amperes are obtained, while hundreds pico-ampere currents are observed as the dark currents.

#### 4.2.2 Processing Cell

The edge detection algorithm described in section 4.1 can be implemented using current mirror circuit shown in Fig. 4.7. The current induced by the photosensor is duplicated by current mirror circuits formed with transistors M1-M5 in two directions. Basic cell schematic is shown in Fig 4.8, which is originally proposed by C. Chong et. al. [30]. Since the photo-current may be of a few hundred pico-amperes to tens of nano-ampere, current mismatch is very large especially in the current duplicated by p-MOS transistors. To minimize current mismatch, cascoded current mirror circuits [34] are used. The circuit schematic of the improved cell is shown in Fig. 4.9. Figure 4.10 shows SPICE simulation results of the output of the current mirror circuit. The photo-current is mirrored in the current mirror circuit consisting of transistors Mn1 - Mn6, Mp1 and Mp2. Transistors Mp3 - Mp10 distribute the duplicated currents to the four nearest-neighborhood cells. The W/L ratio of transistors Mn5 and Mn6 is designed to be four times to generate a subtracting current. At the drain of Mn5, currents from the four adjacent cells are merged to performed the convolution of the Laplacian operation. The output current is given by,

$$I_{out} = I_{i-1,j} + I_{i+1,j} + I_{i,j-1} + I_{i,j+1} - 4I_{i,j} \quad (4)$$

where  $I_{i,j}$  is the current generated in the cell  $(i, j)$ .

The presence of an edge is defined in the cell on the brighter side. In this case, the condition for the detection of an image edge is

$$I_{out} \geq I_{th} \quad (5)$$

where  $I_{th}$  is a threshold value. In a simple case, it will be zero. No edge is present if

$$I_{out} < I_{th} \quad (6)$$

SPICE simulations have been done on the 8 x 8 sub-array, which are shown in Fig. 4.11. The produced photocurrents are 100 pA in the dark side and 10 nA in the bright side. The results show that edge detection is properly performed by choosing an appropriate threshold value. The cell size is 140  $\mu\text{m}$  x 140  $\mu\text{m}$  in a 2- $\mu\text{m}$  CMOS technology. Figure 4.12 shows a selective layout of cell array and decoder circuit in the edge detection chip.

#### 4.2.3 Sense-amplifier and I/O circuit

To reduce delay time caused by the small magnitude of the currents, the two-stage current sense-amplifiers are used. Figure 4.13 shows the schematic of the amplifier. The first stage amplifier, S0, is a bidirectional current sense-amplifier which converts the output current from the cell to a voltage values, and magnifies it by the value of the resistor R (> 5,000  $\Omega$ ). The second sense-amplifier, S1, operates as a comparator. The representation of the output data is a binary format. The threshold voltage is set by external calibration at the positive input of S1. For the moderate output currents of tens of nano-amperes, the time constant of a cell output column line of one picofarad capacitance is about one microsecond [35]. Figure 4.14 shows the detailed circuit schematic of the sense-amplifiers used in S0 and S1. Transistor  $M_{13}$  and  $M_{14}$  form an improved cascode stage to increase the voltage gain and  $M_{24}$  operates as a resistor for proper frequency compensation. The amplifier voltage gain can be 100 dB.

The procedure to output the edge data is similar to that used in static RAM. To keep high parallelism in the chip, a sense-amplifier is implemented in each column. In the block diagram of the chip shown in Fig. 4.5, the row decoder selects one row.



The currents are magnified according to the Laplacian weights and distributed into adjacent cells using analog current mirror circuits. The output signals of cells in the row arrive at the sense-amplifiers in the column. The sense-amplifiers generate the output edge data. This process is performed with all column data but the most outside columns simultaneously, which supports high parallelism. The most outside cells in four sides will experience the fringe effect, because there are no neighboring cells outside the cell array. The input from the lacking cells are connected to ground. Therefore the number of effective data are 48 by 64. The data are stored in a two-stage 48 bit buffer register and arranged in 6 groups of 8 bits. The multiplexor chooses one group from the six groups at one clock. The output data are sent in a group of 8 bits to the buffer SRAM, which reduces the required capacity of the SRAM because one output data occupies one bit instead of eight bits in the 8-bit gray representation usually used in a digital processing input. The data can be read out in 30  $\mu$ -seconds. Most of the requested time are due to the response time of the sense-amplifier. Since the time to output the data from the sense-amplifier is in the order on tens of nano-seconds, all data in the same row are read out in about 30  $\mu$ -seconds. This is valid for the case when there are 256 columns in a row, which shows that the analog approach achieves quite high-speed processing, including reading out the data.

The edge detection chip consisting 50 x 66 cells array can be implemented in a 7.9 mm x 9.2 mm, using the MOSIS 2.0  $\mu$ m p-well CMOS technology. The power supply voltages are 5 V, -5V and 0V. The total power dissipation of the chip is 2 W when a generated current in a cell is 10 nano-amperes.

## **Chapter 5**

### **The Digital Multiprocessor Array**

This chapter presents the digital multiprocessor design for machine vision processing and artificial neural networks, which has been under development in the University of Southern California [36]. The high-level architecture is similar to the Touchstone Delta System from Intel Corp. [37] and the Systolic Cellular Array Processors system from Hughes Co. [38]. The design can support the 1-dimensional systolic ring connection and 2-dimensional mesh-connected configuration.

#### **5.1 System Architecture**

##### **(1) Ring Systolic Array**

The overall system architecture of a ring systolic array used in our system is shown in Fig. 5.1. The host computer serves as the interface between the user and the systolic array. It provides the problem-specific parameters, such as input patterns, initial values, convergence-controlling parameters. The controller specifies and monitors the executions in each PE and also performs loading and receiving data for each PE. The control line is designed in a broadcast fashion since all commands which are performed in all the PE's are the same during each clock cycle.

##### **(2) Mesh-Connected Array**

A mesh-connected architecture requires two-way communication between each PE and all of its nearest neighbors, as shown in Fig. 5.2. The I/O operations occur on the right-most and left-most columns. The top and bottom rows inside the processor array are connected in a wrap-around fashion. The controller is used to control all of the operations in mesh-connected processors. The system memory is a two-port memory which is shared by the processor array. It receives data and instructions from the host computer when the host call is received by the array controller. It also

loads data into the host computer following system call. Data are only transmitted to this system memory from the right-most and left-most columns or vice versa.

## 5.2 Algorithm Mapping

Various algorithms for early vision processing and neural network applications can be mapped onto the multiprocessor array. For example, the mapping of the back-propagation neural network and the competitive learning in a self-organization network onto the mesh-connected PE array are described.

### 5.2.1. Back-propagation Neural Networks

#### (1) The Feedforward Phase

The calculations during the feedforward operation in an artificial neural network are mainly matrix-vector multiplication:

$$S_i(l + 1) = \sum_{j=1}^{m_l} w_{ij}(l + 1)a_j(l) \quad (1)$$

$$a_i(l + 1) = f_i(S_i(l + 1), \theta_i(l + 1)) \quad (2)$$

where  $l$  represents the layer number. The table-look-up method is used for dealing with the nonlinear transfer function.

#### (2) The Back-propagation Phase

The data propagate from the output layer back to the input layer. The weight updating rules are the following:

$$\Delta w_{ji}(l + 1) = \eta \delta_j(l + 1)a_i(l) \quad (3)$$

$$w_{ji}^{new}(l + 1) = w_{ji}^{old}(l + 1) - \Delta w_{ji}(l + 1) \quad (4)$$

The calculations for  $\delta_j$ 's are different in the output layer and in the hidden layers.

For the output layer:

$$\delta_j(l) = (t_j(l) - a_j(l))f'_j(S_j(l)) \quad (5)$$

For each hidden layer:

$$e_j(l) = \sum_{k=1}^N \delta_k(l+1)w_{kj}(l+1) \quad (6)$$

$$\delta_j(l) = e_j(l)f'_j(S_j(l)). \quad (7)$$

Equations (5) and (7) are executed by one column of PEs and (3), (4), and (6) are executed by the whole matrix of PEs.  $\delta_j$  is the error term in the present layer and  $\delta_k$  is that in the next layer. Each processing element has the same value of  $\eta$  which is the updating rate in the data cache. Figure 5.3 shows the mapping of the back-propagated phase into the PE matrix. The right block is for updating weights between the output layer and the hidden layer. The left block is for updating weights between the hidden layer and the input layer. For a back-propagated neural network with more than one hidden layer, the left block can be repeated by the number of the hidden layers. In this case, the left block represents operations for updating weights between two hidden layers.

The host machine receives  $t_j - a_j$  and calculates  $\delta_j$ 's according to (5). At the right block,  $a_j$ 's propagate leftward and  $\delta_j$ 's propagate upward. After receiving  $a_i$  and  $\delta_j$ , the PE calculates  $\Delta w_{ji}$  and updates  $w_{ji}$  by using (4). Notice that the  $\delta_j$ 's at this block are also  $\delta_k$ 's for the left block. Therefore, before  $\delta_j$  moves upward, the PE also calculates  $\delta_k w_{kj}$  and accumulates it with the  $\delta_k w_{kj}$  from the right PE and moves it leftward. Finally, the  $e_j$ 's are obtained at the leftmost column of PE matrix and is sent through the delta function generator for calculating  $\delta_j$ 's of the hidden layer.

At the left block,  $\delta_j$ 's move leftward and  $a_i$ 's move upward. If there is only one hidden layer, each PE only performs (3) and (4). On the other hand, if there are more than one hidden layer, each processing element has to perform (3), (4) and (6), as the processing elements in the right block do, for preparing the  $\delta_j$ 's for the following hidden-layer weight adjustments.

### 5.2.2. Self-Organization Neural Networks

Vector quantization has emerged in recent years as a powerful technique that can provide large reduction in bit rates while preserving the essential signal characteristics. The Kohonen self-organization feature map and the frequency-sensitive learning have been shown to be quite effective for vector quantization codebook design [39]. The mapping of the self-organization learning algorithm onto the multiprocessor chip is described as follows:

- (1) Input vectors  $x_k$ ,  $k = 1, 2, \dots, n$ , to  $n$  rows of processors are applied to the processors.
- (2) Weight vectors, which are also called code vectors,  $c_i$ 's moves downward to perform distortion calculation  $E_i = d(x_i, c_i)$  for all output neural units. An index pointer  $p_k$  indicating the current winner also passes to the right processor.
- (3) After calculations in the  $k$ -th row have been completed,  $p_k(m)$  indicated the output unit in the first row with the smallest distortion. Label this unit as the winner and its weight vector as  $c_i^*$ .
- (4) Adjust the selected weight vector

$$c_i^*(t + 1) = c_i^*(t) + \epsilon(t)[x_k(t) - c_i^*(t)]$$

and write it back into the system memory, where  $t$  is the training time index.

- (5) Repeat steps (3) to (4) for all other rows.

Notice that the adjusted weight vector  $c_i^*(t + 1)$  will not be applied to the on-going processing in the other rows. It only affects the next training cycle. The value selected for  $\epsilon(t)$  does not depend on the magnitude of the data. The training rule moves the weight toward the training vector by some fractional amounts,  $\epsilon(t)$ . Typically,  $0 < \epsilon(t) < 1$  and the  $\epsilon(t)$  value decreases as training progresses.

### 5.3 Processing Element Design

Figure 5.4 shows the building blocks of one processing element. After the initialization phase, the initial parameters will be down-loaded into the system memory. One on-chip data cache memory with 256 words of 8 bit data is included in each PE to accelerate data processing. The memory management unit determines memory allocation policies which include block replacement schemes, fetch and main memory update strategies. The memory management unit is shared by all PEs and is inside the controller. Four I/O ports available for four-directional communication.

The detailed floor plan for a single PE is shown in Fig. 5.5. The data bus is 8 bits and address bus is 16 bits. Addresses from the external address bus are passed through the peripheral address bus to both the data cache and the next processor. High level language from the host computer are first compiled into macroinstructions which are sent to the controller. The controller decodes these macroinstructions into microinstructions and broadcast them to all PEs via the 43 Mcode lines shown in Fig. 5.4. The register bus is separated into four buses, REG1, REG2, CPU1, and CPU2. A multiplexor is used to multiplex these four buses into the internal data bus. When a data is to be sent from the data cache to the register file, or vice versa, CPU1 bus is used.

The execution unit in each PE includes a 8-bit multiplier and a 20-bit adder. The two operands of both multiplier and adder can come from the CPU1 and CPU2 buses, or the REG1 and REG2 buses depending on the addressing mode. One operand is from the CPU1 or REG1 bus only, the other operand is from the CPU2 or REG2 bus only. The 16-bit output of the multiplier is sent to the register file through the REG1 bus only while the 20-bit result of the adder is sent to the register file through the REG2 bus only.

The system clock is synchronized for the processor array. The global bus clock cycle and the internal bus clock cycle are faster than 100 MHz and 200 MHz, respectively. Each PE requires a total of 65 I/O lines for one-direction ring connected

communication.

A two-level microprogramming is used to speedup the network response because most operations are repetitive and commonly used. For a particular program, a systolic procedure is first implemented by a high level language in the host computer, then is sent to the array controller. In order to increase the processor speed, the internal bus is segmented into multiple local bus lines. Therefore, more steps can be overlapped because they do not occupy the same bus. The speed of the processor is limited by the multiplier. The Wallace multiplier is used in our design. Since the Wallace structure can be separated into 4 x 4 multiplier blocks and Wallace tree blocks, a pipeline procedure can be added. A fault detection and recovery module in the controller is used for detecting the faulty processor, correcting and recovering from errors by spare processors.

A prototype ring-connected 20-PE chip is designed on an area of 19.3 mm x 20.9 mm using the 0.5- $\mu$ m CMOS technology from the VHSIC (Very high speed integrated circuits) program of TRW, Inc. An simulated speed is 2 billion calculations per second. The fabricated chip will be available in the Fall 1992 semester for hardware testing and system integration.

Figure 5.6 shows the relative performance of our processor array and the performance requirements of the Grand Challenges on High-Performance Computing [40,41].

## **Chapter 6**

### **Conclusion**

Machine vision processing is one of the most computationally intractable domains of artificial intelligence. In this thesis, an attempt to implement the early vision processing system using the analog edge detection chip and the digital neural multiprocessor array chip has been shown to be quite effective. The analog intelligent sensor can provide much more parallelism than any other digital implementations and can reduce the hardware size. The artificial neural network approach to the early vision processing is rather exciting, because the neural networks and the vision processing have similar properties, such as massive parallel computation and communication.

An analog edge detection chip is presented in Chapter 4. It accommodates 50 x 66 pixels on a silicon area of 7.9 mm x 9.2 mm in a 2- $\mu$ m CMOS technology. The chip can provide binary format output which are suitable for digital processing to the digital neural multiprocessor described in Chapter 5. The digital neural multiprocessor consists of a 20-ring-connected processing elements, which is implemented in silicon area of 19.3 mm x 20.9 mm in a 0.5- $\mu$ m CMOS technology. The computational capability is 2 billion calculations per second.

Finally, we believe that the mixed-mode neural network based system for early vision processing presented in this thesis is very important one step to the construction of machine the vision processing system.



## REFERENCES

- [1] R. Hecht-Nielsen, *Neurocomputing*, Addison-Wesley: MA, 1989.
- [2] S. P. Levitan, C.C. Weems, A. R. Hanson, E. M. Riseman, "The UMass image understanding architecture," *Parallel Computer Vision*, Academic Press, Inc.: Orland, FL, pp.215-248, 1987
- [3] C. Reinhart, R. Nevatia, "Parallel linear feature extraction," *The DARPA Image Understanding Workshop*, Feb. 1992.
- [4] J. Little, G. Brelloch, and T. Cass "Parallel algorithms for computer vision on the Connection Machine," *Proceeding of the DARPA Image Understanding Workshop*, page 628-638, Feb. 1987.
- [5] R. Vaillant, R. Deriche, O. Faugeras, "3D vision on the parallel machine CAPITAN," *International Workshop on Industrial Application of Machine Intelligence and Vision*, pp.326-331, Apr. 1989.
- [6] D. B. Shu, J. G. Nash, M. M. Eshaghian, and K. Kim, "Straight-line detection on a gated-connection," *Prod. of the Tenth Intr. Conf. on Pattern Recognition*, pp. 456 - 461, Jun. 1990.
- [7] S. Y. Lee, J. K. Aggarwal, "Parallel 2-D convolution on a mesh connected array processor," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-9(9): pp. 590 -594, Jul. 1987.
- [8] J. A. Anderson, "A simple neural network generating an interactive memory," *Mathematical Bioscience*, vol. 14, pp. 197 -220, 1972.
- [9] B. Widrow, Bernard, Hoff, and Marcian, "Adaptive switching circuits," *1960 IRE WESCON Convention Record*, Part 4, pp. 96 104, Aug. 1960.
- [10] J.J. Hopfield, "Neural network and physical system with emergent collective computational abilities," *Proc. Natl. Acad., Sci. U.S.A.*, vol. 79, pp. 2554 - 2558, Apr. 1982.
- [11] D. E. Rumelhart, J. L. McClelland, *Parallel Distributed Processing: Explorations in the Micostructure of Congnition, I & II*, MIT Press: Cambrigde MA, 1986.
- [12] T. Kohonen, *Self-Organization and Associate Memory*, 2 nd ED., New York: Springer-Verlag, 1987.
- [13] R. Hect-Nielsen, "Counter-propagation networks," *Proc. of IEEE First Intr. Conf. on Neural Network*, vol. II, pp. 19 -32, San Diego, CA, 1987.

- [14] B. W. Lee, B. J. Sheu, *Hardware Annealing in Analog VLSI Neurocomputing*, Kluwer Academic: MA., 1991.
- [15] D. Hebb, *The Organization and Behavior*, New York: Wiley, 1949.
- [16] A. H. Klopf, "A drive-reinforcement model of single neuron function: an alternative to the Hebbian neural model," *Proc. of Conf. on NEural Networks for Comuting*, pp. 265 - 270. Snowbird, UT, Apr. 1986.
- [17] G. E. Hinton, T. J. Sejnowski, "A learning algorithm for Boltzman machine," *Cognitive Science*, vol. 9, pp. 147 - 169, 1985.
- [18] S. Y. Kung, W. Chou, "Mapping neural network onto VLSI array processors," *Digital Implemenation of Neural Network*, pp. 3 - 49, 1992.
- [19] J. J. Hopfield, D. W. Tank, "Neural computation of decision in optcimization problbem," *Biological Cybernetics*, 52: pp. 141 - 152, 1985.
- [20] C. Mead, *Analog VLSI and neural systems*. Addison-Wesley: MA, 1989.
- [21] S. Y. Kung, *VLSI Array Processor*, Prentice-Hall, Inc.: 1988.
- [22] H. T. Kung, C.E. Leiserson, "Systolic arrays (for VLSI)", *Sparse Matrix Symposium*, pp. 256 - 282, SIAM, 1978.
- [23] S. Anderson, W. Bruce, P. Denyer, D.Renshaw, and G. Wang, "A single chip sensor & image processor for fingerprint verification," *IEEE Custom Integrated Circuits Conference*, San Diego, U.S.A. May, 1991.
- [24] C. P. Chong, C. A. Salama, and K. C. Smith, "Image-Motion detection using analog VLSI," *IEEE Jour. of Solid-State Circuits*, vol. 27, no. 1, pp. 93 - 96, Jan. 1992.
- [25] H. Okada, B. J. Sheu, and C. F. Chang, "An analog edge detection chip and a digital multiprocessor chip for neural-based vision processing," *IEEE Intr. Conf. System Engineering*, Kobe, Japan, Sep. 1992.
- [26] C. Mead, M. Ismail, *Analog VLSI implementation of Neural systems*. Kluwer Academic: Boston, 1989.
- [27] A. G. Andreou, "Current-mode subthreshold MOS circuits for analog VLSI neural systems," *IEEE Trans. Neural Networks*, vol. 2, pp. 205 -213, 1991.
- [28] C. Mead, X. Arreguit, and J. Lazzaro, "Analog VLSI model of binaural hearing," *IEEE Trans. Neural Network*, vol. 2, pp. 230 - 236, 1991.
- [29] H. Kobayashi, J. L. White, and A. A. Abidi, "An Active Resistor Network for Gaussian Filtering of Images," *IEEE Jour. of Solid-State Circuits*, vol. 26, no. 5, May. 1991.

- [30] Chu P. Chong, C. Andre T. Salama, Kenneth C. Smith, "Real-time edge detection and image segmentation," *Journal of Analog Integrated Circuits & Signal Processing*, Kluwer Academic Publishers, 1992.
- [31] E. A. Vittzo, "Future of analog in the VLSI enviroment," *Proc. ISCAS*, New Orleans, LA, pp. 1372 -1375, May. 1990.
- [32] J. Canny, "A computaional approach to edge detection," *IEEE Tran. of Pattern Analysis and Machine Intelligence*, vol. PAMI-8, No. 6, pp. 679 - 698, Nov. 1986.
- [33] B. K. P. Horn, *Robot Vision*, The MIT Press: Cambridge, MA, 1986.
- [34] P. E. Allen, D. R. Holberg, "CMOS Analog Circuit Design," Holt, Rinehart and Winston, Inc: New York, NY, 1987
- [35] C. A. Mead, T. Delbruck, "Scanners for visualizing activity of analog VLSI circuitry," *Analog Integrated Circuits and Signal Processing*, Kluwer Academic: MA., 1991.
- [36] C. Chang, B. Sheu, and H. Okada, "Design of a multiprocessor DSP chip for flexible information processing," *IEEE Inter. Conf. on Acoustic, Speech, and Signal Processing*, San Francisco, CA, Mar. 1992.
- [37] *IEEE Spectrum Magazine*, Jan. 1991.
- [38] K. W. Przytula, W. Lin, and V. Kumar, "Partitioned implementation of neural networks on mesh connected array processors," *VLSI Signal Processing IV*, pp. 106 - 115, IEEE Pres: Piscataway, NJ, 1991.
- [39] T. Kohonen, "The self-organization map," *IEEE Proceeding*, pp. 1464 - 1480, Sep. 1990.
- [40] D. Dobberpuhl, et. al. "A 200 MHz 64b dual-issue CMOS microprocessor," *Intr. Solid-State Circuits Conf.*, pp. 106 - 107, Feb. 1992.
- [41] U. S. Government Committee, *Grand Challenges: High Performance Computing and Communications*, National Science Foundation, 1991.

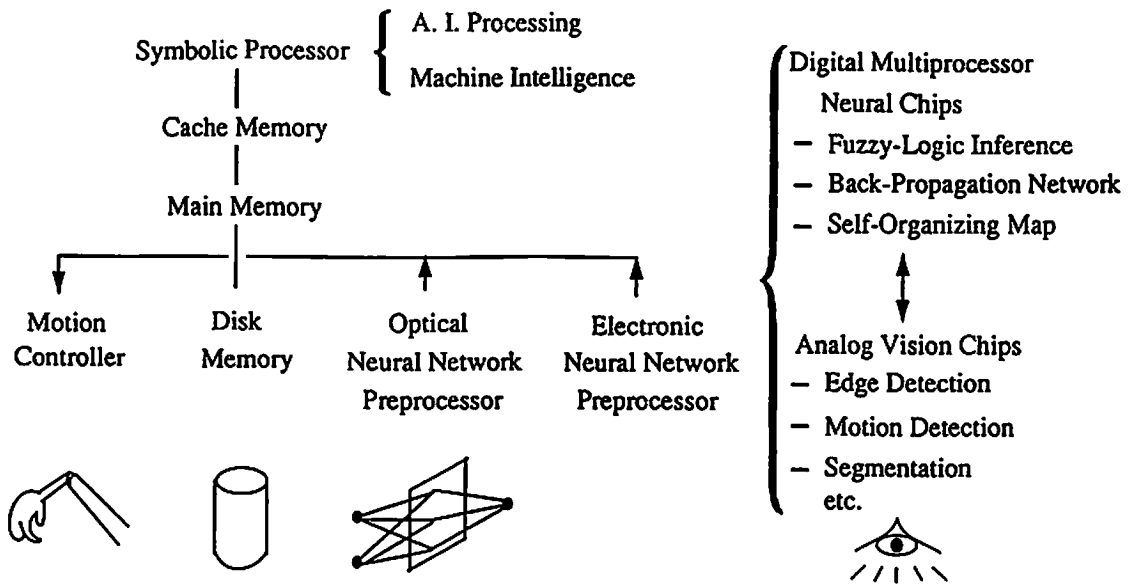


Fig. 1.1 System configuration for neural-based vision processing using an analog edge detection chip and a digital multi-processor chip.

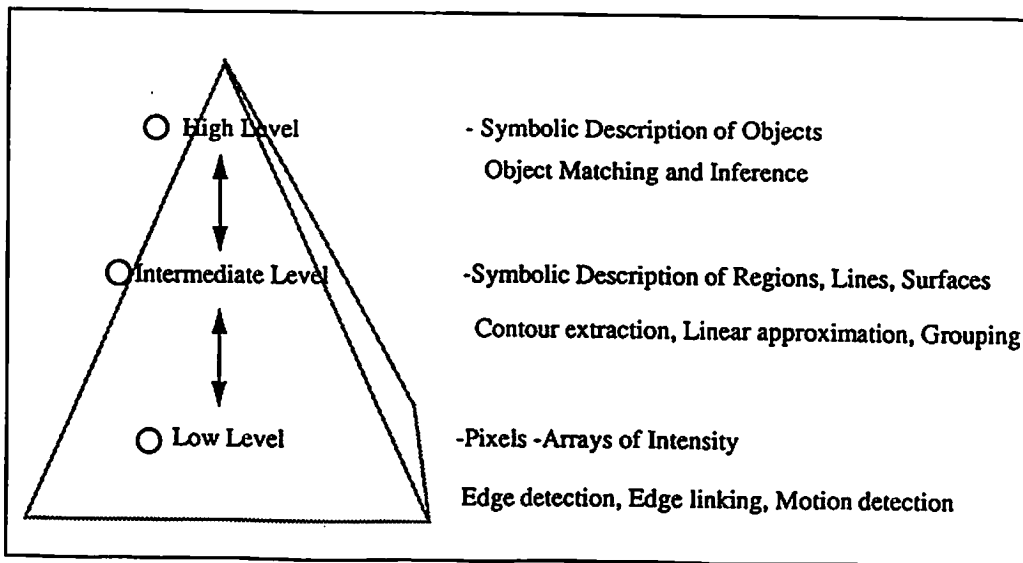


Fig. 2.1 Multiplr level structure of vision processing.

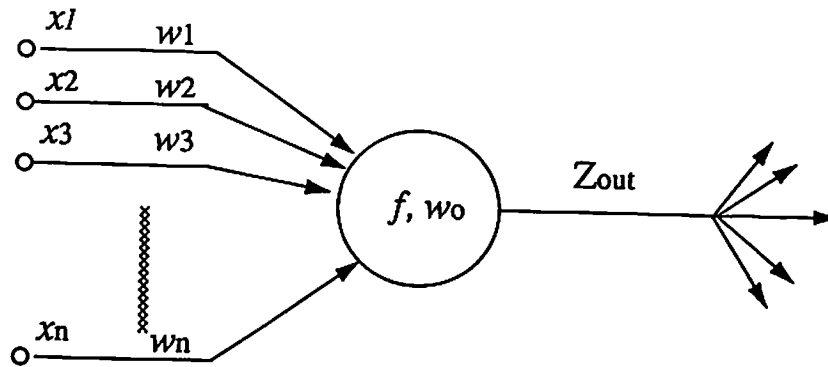
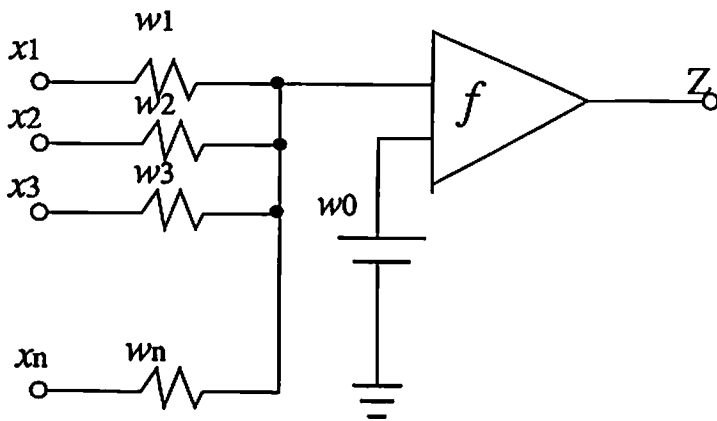
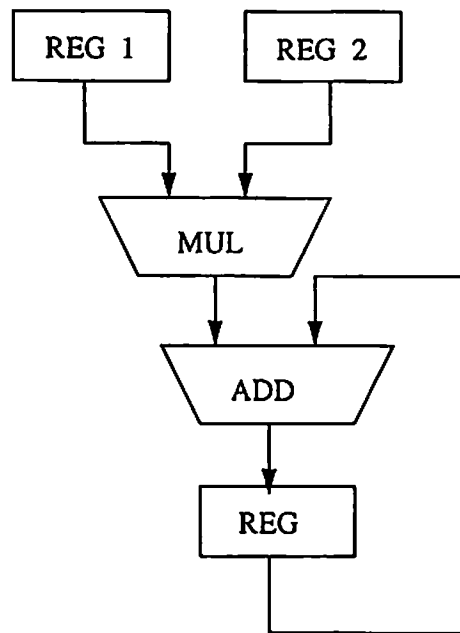


Fig. 2.2 Typical neural processing element model.



(a) analog implementataion.



(b) digital implementation.

Fig. 2.3 VLSI neuron model  
 (a) analog implementataion. (b) digital implementation.

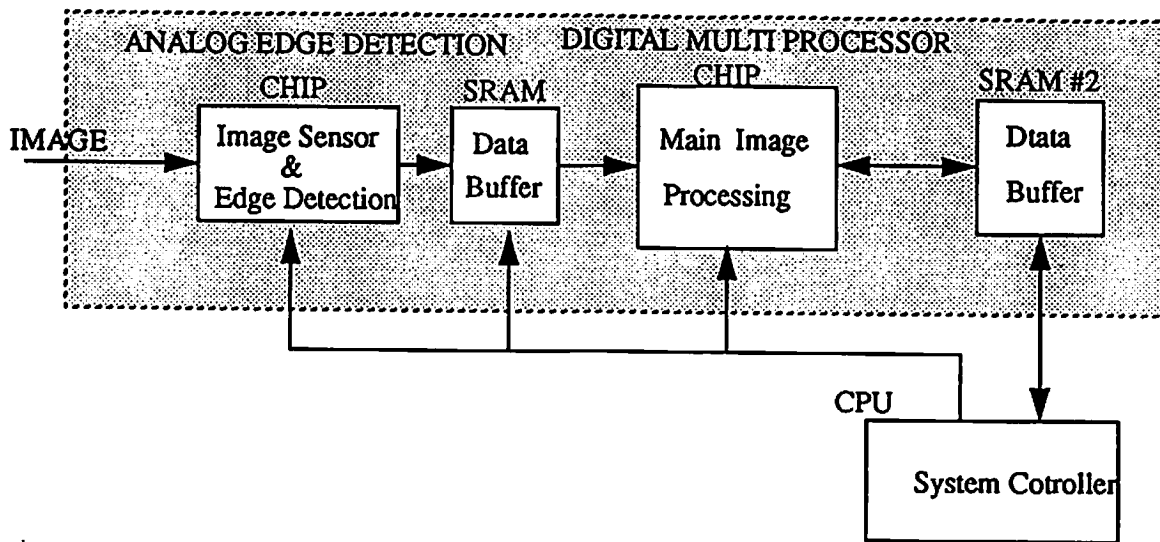


Fig. 3.1 System configuration for neural-based early vision processing using an analog edge detection chip and a digital multiprocessor array chip.

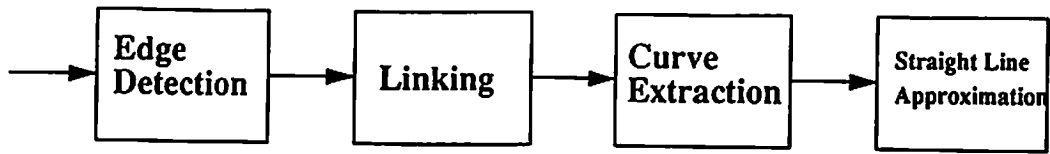


Fig. 4.1 Early vision processing.

$C(i-1,j+1)$	$C(i,j+1)$	$C(i+1,j+1)$
$C(i-1,j)$	$C(i,j)$	$C(i+1,j)$
$C(i-1,j-1)$	$C(i,j-1)$	$C(i+1,j-1)$

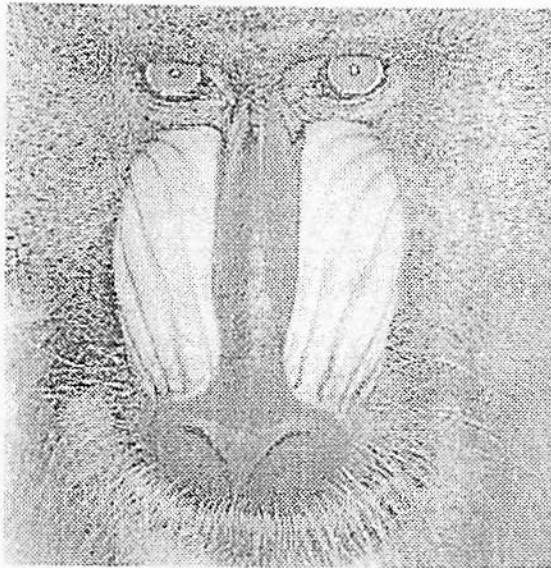
Fig. 4.2 The 3 x 3 window.

0	1	0
1	-4	1
0	1	0

Fig. 4.3 Stencil of the convolution weight (3 x 3 Laplacian operator).



(a) girl image:



(b) baboom:

Fig. 4.4 Computer simulation results of edge detection using the Laplacian operator.  
(a) girl image: Upper: Original image Bottom: Edge image  
(b) baboom: Upper: Original image Bottom: Edge image



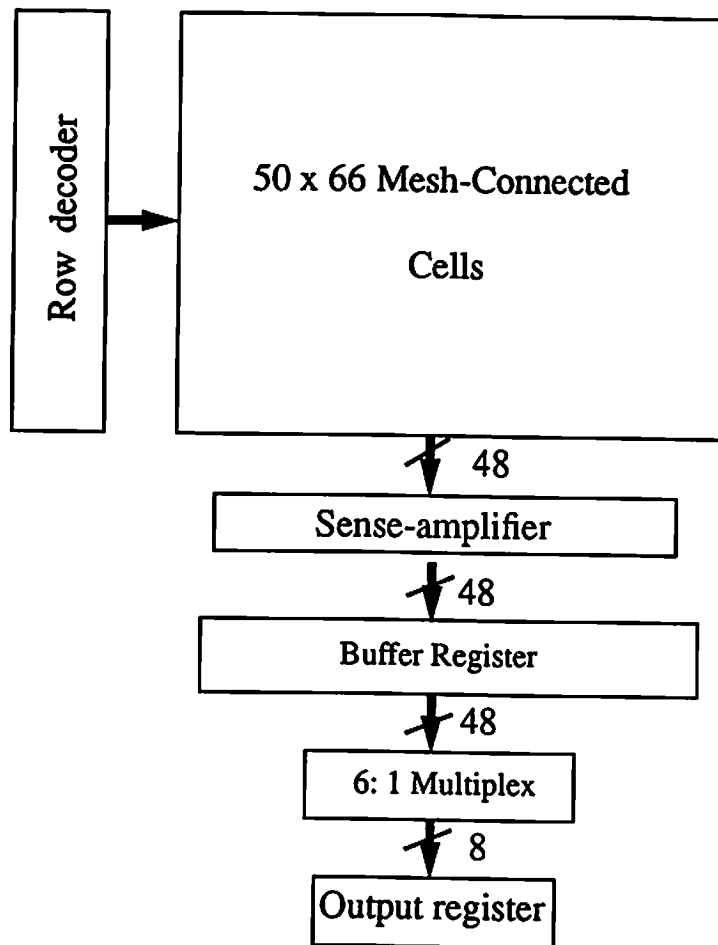


Fig. 4.5 Block diagram of the edge detection chip. The 50 x 66 cells, a decoder, 48 sense-amplifier, buffer registers, multiplexer, and output registers are included.

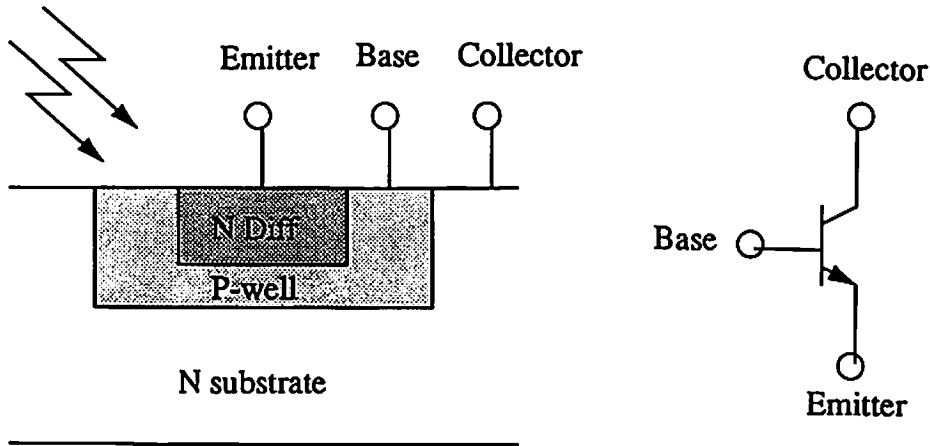


Fig. 4.6 Photosensor using parasitic bipolar transistor in a p-well CMOS technology.

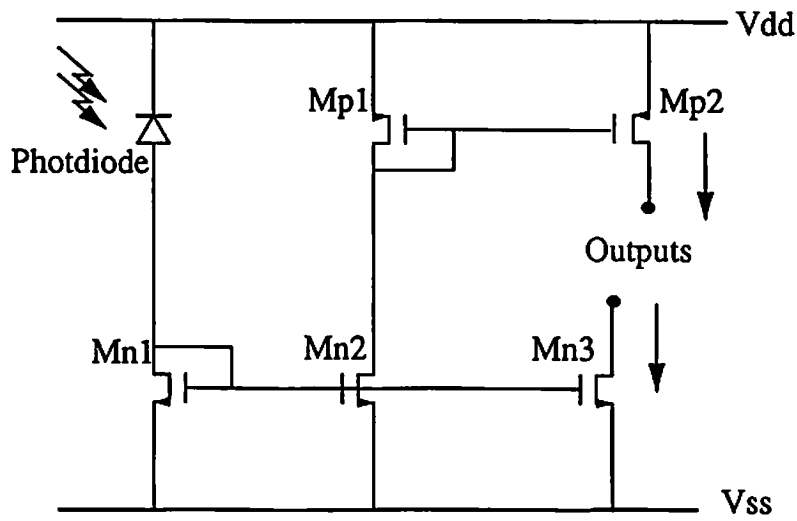


Fig. 4.7 Current mirror circuit.

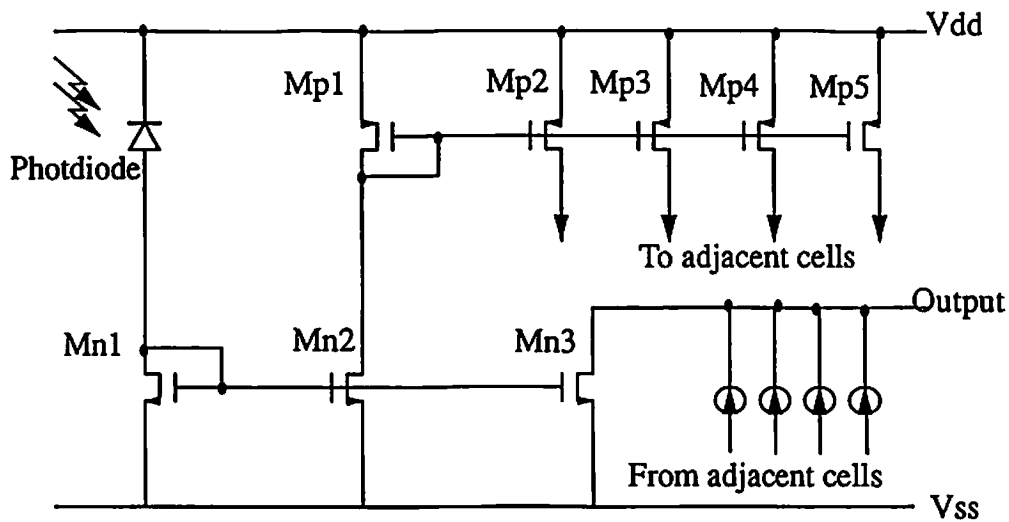


Fig. 4.8 Circuit schematic of the basic cell.

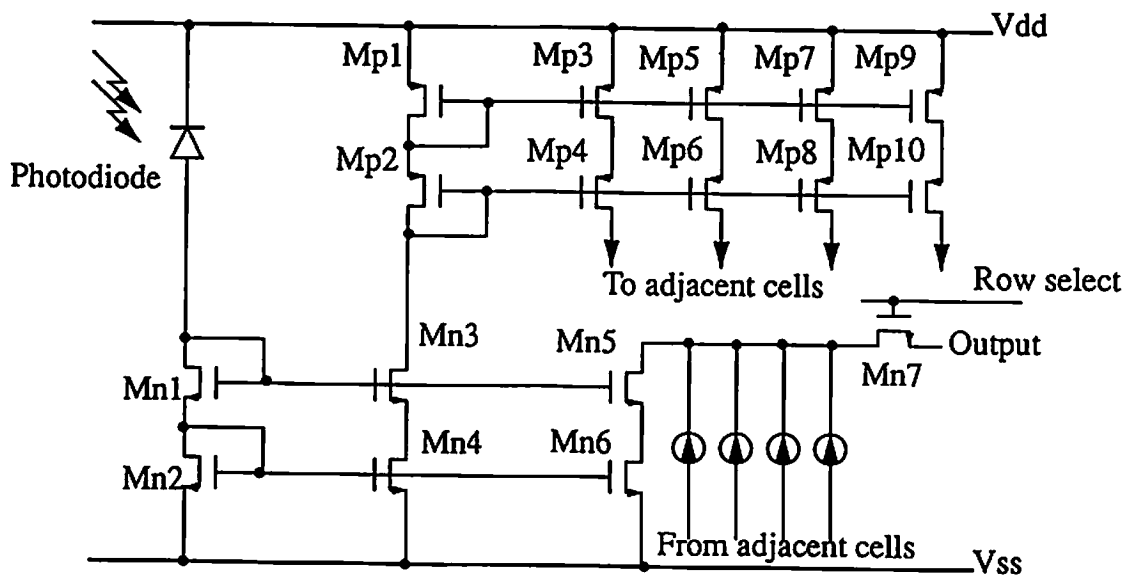


Fig. 4.9 Current mismatch improved circuit schematic.

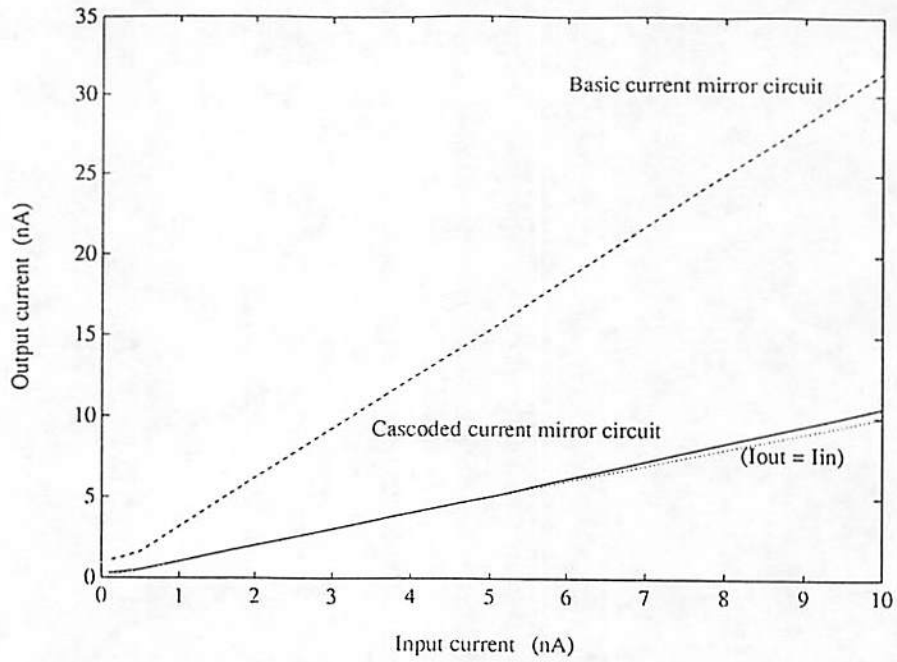


Fig. 4.10 Simulated characteristics of the basic cell and the cascoded cell.

20.1	9.65	9.65	9.65	9.65	9.65	15.9	11.9	26.7	16.7	16.7	16.7	16.7	16.7	16.7	16.7	26.7
9.65	-0.85	-0.85	-0.85	-0.85	5.39	-3.33	-2.62	16.7	6.63	6.63	16.4	16.4	16.4	6.63	16.7	
9.65	-0.85	-0.85	-0.85	5.39	-3.33	-7.36	2.65	16.7	6.63	16.4	-19.6	-9.78	-19.6	16.4	16.7	
9.65	-0.85	-0.85	5.39	-3.33	-7.36	-2.09	2.65	16.7	6.63	16.4	-9.78	0	-9.78	16.4	2.65	
9.65	-0.85	5.39	-3.33	-7.36	-2.09	-2.09	2.65	16.7	6.63	16.4	-9.78	0	-9.78	16.4	2.65	
9.65	5.39	-3.33	-7.36	-2.09	-2.09	-2.09	2.65	16.7	6.63	16.4	-19.6	-9.78	-19.6	16.4	2.65	
15.9	-3.33	-7.36	-2.09	-2.09	-2.09	-2.09	2.65	16.7	6.63	6.63	16.4	16.4	16.4	6.63	2.65	
11.9	-2.62	2.65	2.65	2.65	2.65	2.65	7.39	26.7	16.7	16.7	16.7	16.7	16.7	16.7	26.7	

The numbers in cells represent the output currents (nA)

:  $I_{ph} = 10\text{nA}$     
  :  $I_{ph} = 10\text{pA}$     
  : Edge cell    
  $I_{ph}$  : The current generated by photosensor

Fig. 4.11 SPICE simulation results of 8 x 8 array of cells.

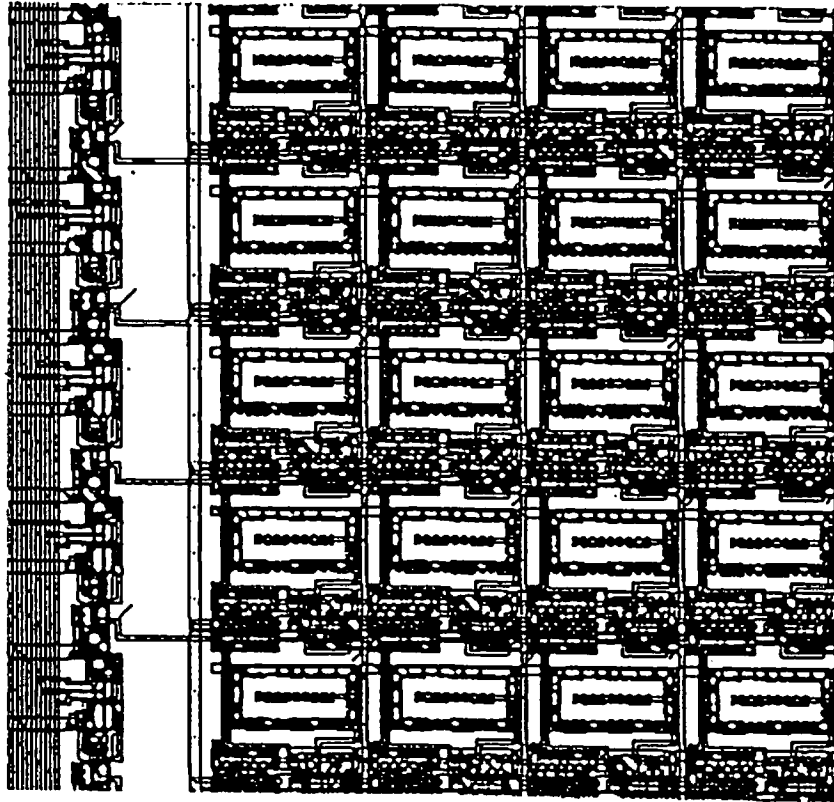


Fig. 4.12 Microphotograph of the cell.

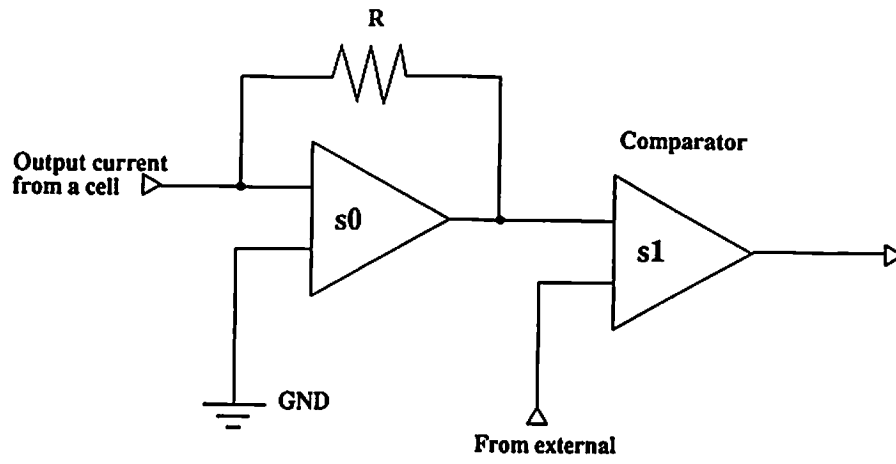
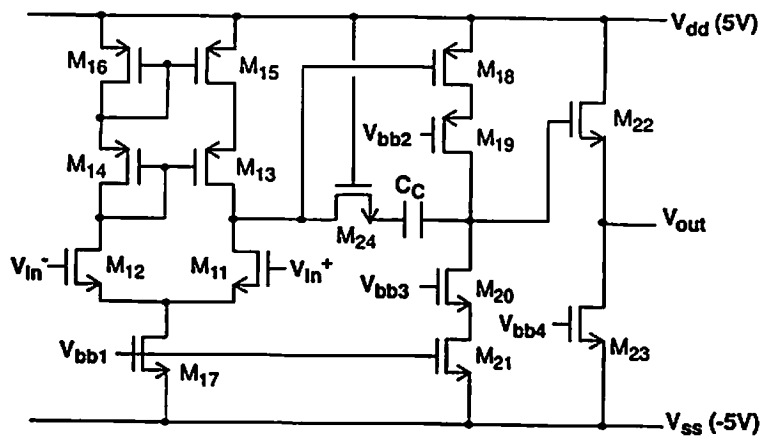


Fig. 4.13 Schematic diagram of the sense-amplifier.



Transistor	M <sub>11</sub>	M <sub>12</sub>	M <sub>13</sub>	M <sub>14</sub>	M <sub>15</sub>	M <sub>16</sub>	M <sub>17</sub>	M <sub>18</sub>
Size in um	16/2	16/2	6/2	6/2	20/4	20/4	40/4	46/4
Transistor	M <sub>19</sub>	M <sub>20</sub>	M <sub>21</sub>	M <sub>22</sub>	M <sub>23</sub>	M <sub>24</sub>	C <sub>C</sub>	
Size in um	90/2	26/2	52/4	600/2	400/4	50/2	1.5pF	

Fig. 4.14 Detailed circuit schematic diagram of the operational amplifier.

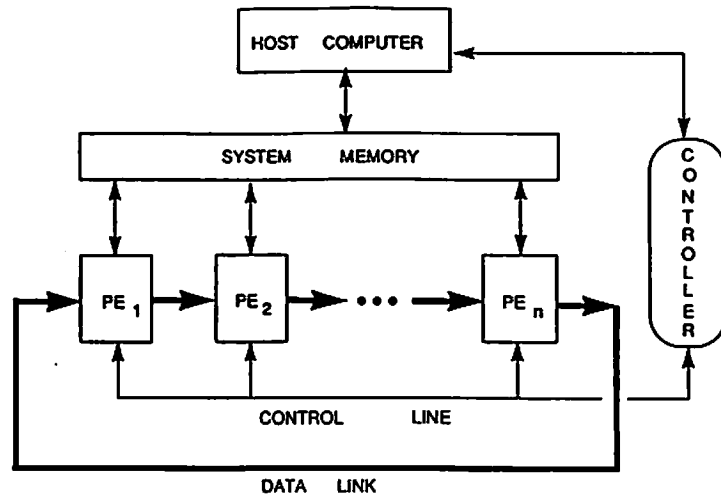


Fig. 5.1 Ring systolic array architecture.

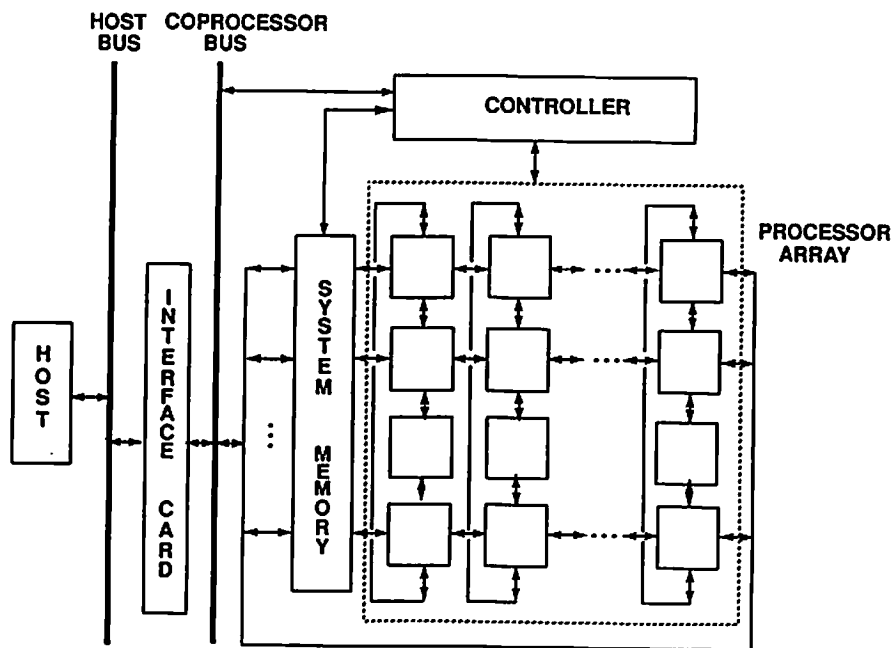


Fig. 5.2 Mesh-connected systolic array architecture.

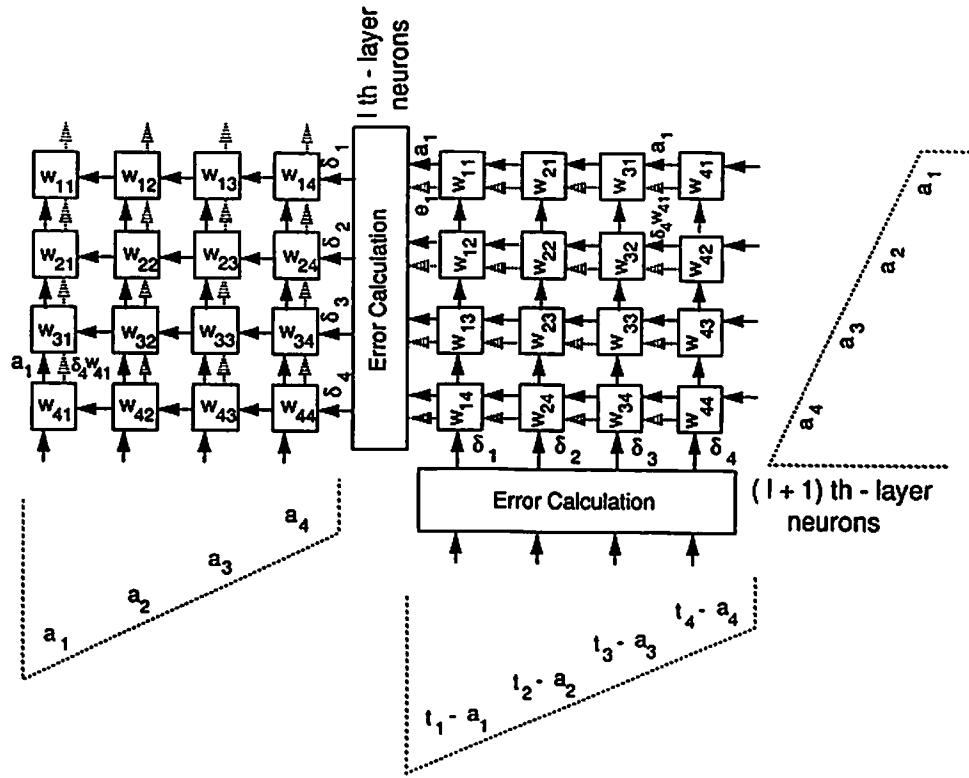


Fig. 5.3 Mapping of back-propagation algorithm onto the multi PE chip.

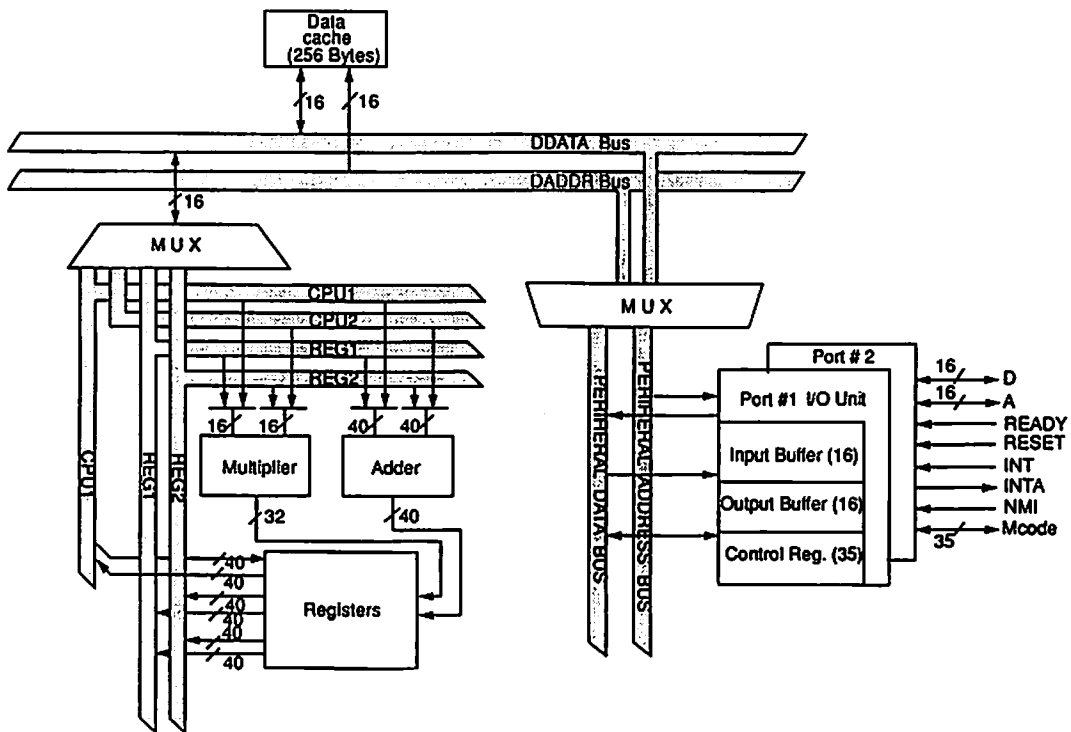


Fig. 5.4 Block diagram of the digital processor.



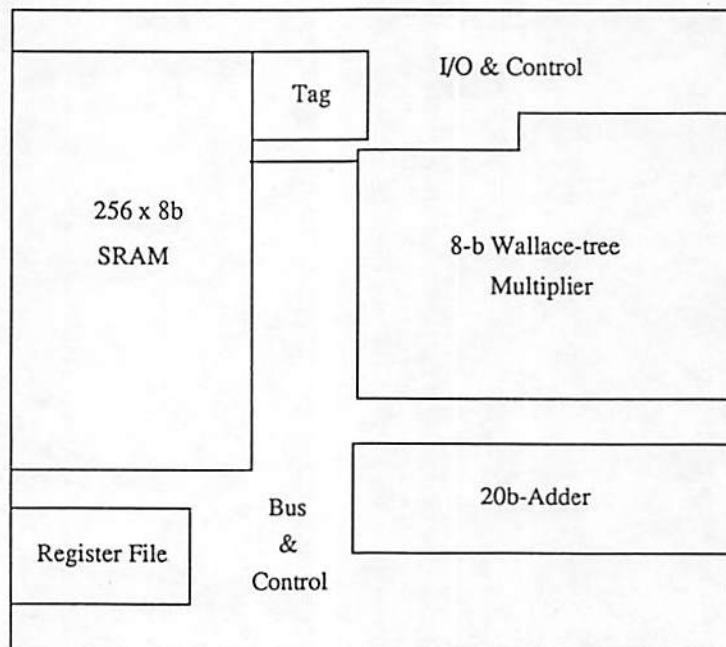
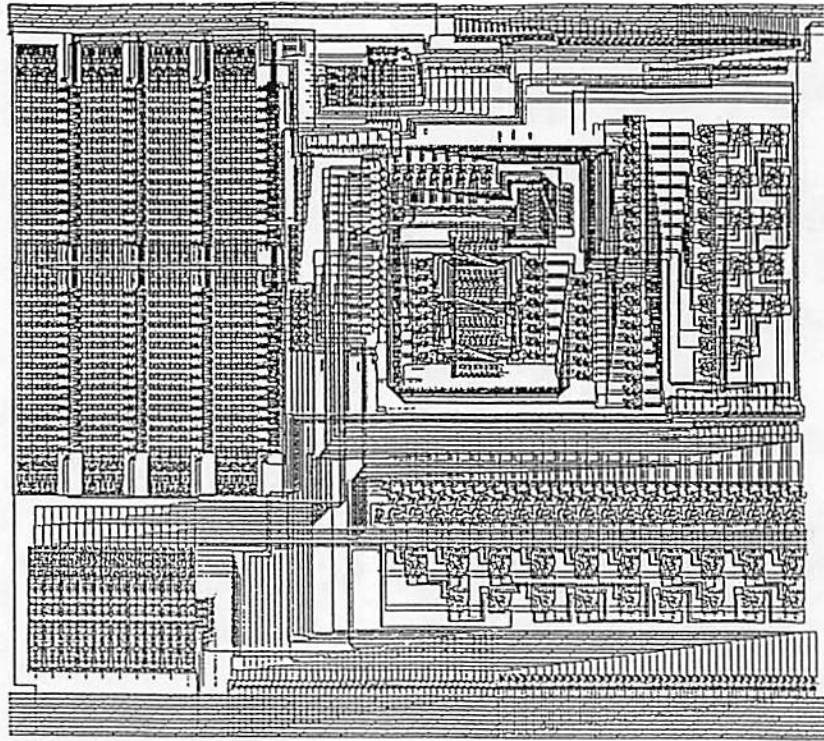


Fig. 5.5 Floor-plan of the single processor.

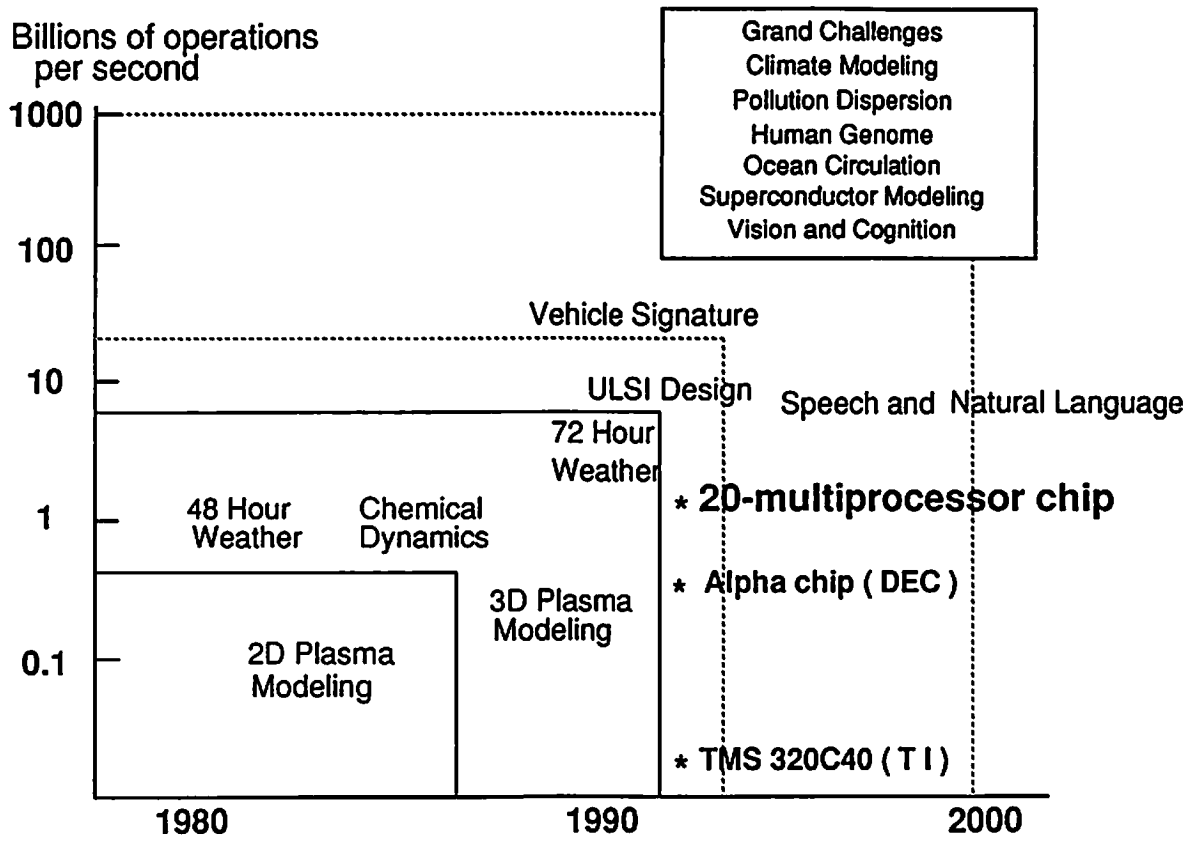


Fig. 5.6 Requirements from the Grand Challenges on high-performance computing.

**Table 2.1. Complexity of Algorithms in Early Vision Processing**

Algorithm	Complexity
Edge Detection	$O(i^2)$
Edge Linking	$O(i^2)$
Contour Extraction	$O(i^2)$
Linear Approximation	$O(nl)$

where  $i$ :  $i \times i$  pixels image,  $l$ : average contour length  
 $n$ : number of contours

**Table 2.2. Major Neural Network Models and Properties**

Neural model	Primary applications	Strengths	Limitations
Perceptron	Typed-character recognition	Oldest neural network	Cannot recognize complex patterns
Hopfield	Retrieval of data/images from fragments	Large-scale integration	Does not learn, weights must be set
Multilayer Perceptron/Delta Rule	Pattern recognition	Simple network, more general than the perceptron	cannot recognize complex patterns
Back Propagation	Wide range: speech synthesis to loan - application scoring	Most popular, work well, and is simple to learn	Supervised training with abundant examples
Boltzmann Machine	Pattern recognition for radar/sonar	Simple network that uses noise function to reach global minimum	Long training time
Self-Organizing Map	Mapping one geometrical region onto another	Better performance than many algorithmic techniques	Extensive learning

## **Appendix 1:**

### **Publications**

#### **Confereces:**

- [1] C. Chang, B. Sheu, and H. Okada, "Design of a multiprocessor DSP chip for flexible information processing," *IEEE International Conference on Acoustic, Speech, and Signal Processing*, San Francisco, CA, Mar. 1992.
- [2] H. Okada, B.J. Sheu, "An Analog Edge Detection Chip," *USC-SIPI '92 Anual Report*, Marina Del Ray, CA, Apr. 1992.
- [3] H. Okada, B. J. Sheu, and C. F. Chang, "An analog edge detection chip and a digital multiprocessor chip for neural-based vision processing," *IEEE International Conference of System Engineering*, Kobe, Japan, Sep. 1992.
- [4] J. C. Lee, H. Okada, B. J. Sheu, R. Chellapa, "A mixed-signal VLSI of competitive neural network for video motion detection," *IEEE/INNS International Joint Conference on Neural Networks*, Beijing, China, Nov. 1992. [Submitted]

#### **Paper:**

- [1] H. Okada, "A study of an application of the back-propagation network for surface reconstruction," *USC-SIPI Report #180 VLSI Neurocomputers: EE599 Term Projects/ 1990 - 1991*, May 1991.

## **7.0 Publications From "VLSI Signal Processing Laboratory"**

### **7.1 Refereed Journal Publications**

- 7.1.1 B. J. Sheu, C. Hu, "Modeling the switch-induced error voltage on a switched-capacitor," *IEEE Trans. on Circuits and Systems*, vol. CAS-30, no. 12, 911-913, Dec. 1983.
- 7.1.2 B. J. Sheu, D. L. Scharfetter, C. Hu, D. O. Pederson, "A compact IGFET charge model," *IEEE Trans. on Circuits and Systems*, vol. CAS-31, no. 8, 745-748, Aug. 1984.
- 7.1.3 B. J. Sheu, C. Hu, "Switch-induced error voltage on a switched capacitor," *IEEE Jour. of Solid-State Circuits*, vol. SC-19, no. 4, pp. 519-525, Aug. 1984.
- 7.1.4 B. J. Sheu, C. Hu, P. K. Ko, F.-C. Hsu, "Source-and-drain series resistance of LDD MOSFET's," *IEEE Elec. Device Letters*, vol. EDL-5, no. 9, pp. 365-367, Sept. 1984.
- 7.1.5 B. J. Sheu, P. K. Ko, "A capacitance method to determine channel lengths for conventional and LDD MOSFETs," *IEEE Elec. Device Letters*, vol. EDL-5, no. 11, pp. 491-493, Nov. 1984.
- 7.1.6 B. J. Sheu, P. K. Ko, "A simple method to determine channel widths for conventional and LDD MOSFETs," *IEEE Elec. Device Letters*, vol. EDL-5, no. 11, pp. 485-486, Nov. 1984.
- 7.1.7 B. J. Sheu, P. K. Ko, "Short-channel effects on MOS transistor capacitances," *IEEE Trans. on Circuits and Systems*, vol. CAS-33, no. 10, pp. 1030-1032, Oct. 1986.
- 7.1.8 B. J. Sheu, J.-H. Shieh, M. Patil, "Modeling charge injection in MOS analog switches," *IEEE Trans. on Circuits and Systems*, vol. CAS-34, no. 2, pp. 214-216, Feb. 1987.
- 7.1.9 J.-H. Shieh, M. Patil, B. J. Sheu, "Measurement and analysis of charge injection in MOS analog switches," *IEEE Jour. of Solid-State Circuits*, vol. SC-22, no. 2, pp. 277-281, Apr. 1987.
- 7.1.10 B. J. Sheu, P. K. Ko, "Measurement and modeling of short-channel MOS transistor gate capacitances," *IEEE Jour. of Solid-State Circuits*, vol. SC-22, no. 3, pp. 464-472, Jun. 1987.
- 7.1.11 M. C. Hsu, B. J. Sheu, "Inverse-geometry dependence of MOS transistor electrical parameters," *IEEE Trans. on Computer-Aided Design*, vol. CAD-6, no. 4, pp. 582-585, July 1987.
- 7.1.12 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.
- 7.1.13 B. J. Sheu, A. H. Fung, Y.-N. Lai, "A knowledge-based approach to analog integrated circuit design" *IEEE Trans. on Circuits and Systems*, vol. CAS-35, no. 2, pp. 256-258, Feb. 1988.
- 7.1.14 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.
- 7.1.15 C.-C. Shih, B. J. Sheu, H. M. Le, "Characterization of GaAs MESFET gate capacitances," *IEEE Jour. of Solid-State Circuits*, vol. SC-23, no. 3, Jun. 1988.
- 7.1.16 C.-P. Wan, B. J. Sheu, S.-L. Lu, "Device and circuit simulation interface for an integrated VLSI design environment," *IEEE Trans. on Computer- Aided Design*, vol.

- CAD-7, no. 7, pp. 998-1004, Sept. 1988.
- 7.1.17 A. H. Fung, B. W. Lee, B. J. Sheu, "Self-reconstructing technique for expert system-based analog IC designs," *IEEE Trans. on Circuits and Systems*, vol. 36, no. 2, pp. 318-321, Feb. 1989.
- 7.1.18 B. J. Sheu, W.-J. Hsu, B. W. Lee, "An integrated-circuit reliability circuit simulator - RELY," *IEEE Jour. of Solid-State Circuits*, vol. SC-24, no. 2, pp. 473-477, Apr. 1989.
- 7.1.19 B. W. Lee, B. J. Sheu, "Design of a neural-based A/D converter using Hopfield network," *IEEE Jour. of Solid-State Circuits*, vol. SC-24, no. 4, pp. 1129-1135, Aug. 1989.
- 7.1.20 C.-P. Wan, B. J. Sheu, "Temperature dependence modeling for MOS VLSI circuit simulation," *IEEE Trans. on Computer-Aided Design*, vol. 8, no. 10, pp. 1065-1073, Oct. 1989.
- 7.1.21 W.-J. Hsu, B. J. Sheu, V. C. Tyree, "Computer-aided VLSI circuit reliability assurance," *International Journal of Modeling and Simulation (by IASTED, Calgary, Canada)*, vol. 9, no. 4, pp. 118-123, 1989.
- 7.1.22 B. W. Lee, B. J. Sheu, "A high slew-rate CMOS amplifier for analog signal processing," *IEEE Jour. of Solid-State Circuits*, vol. 25, no. 3, pp. 885-889, June 1990.
- 7.1.23 B. J. Sheu, J.-C. Lee, A. H. Fung, "Flexible architecture approach to knowledge-based analog IC design," *IEE Proceedings*, vol. 137, Part G, no. 4, pp. 266-274, Aug. 1990.
- 7.1.24 C.-P. Wan, B. J. Sheu, "A new method to determine geometric dependences of sub-micron MOS transistor parameters," *IEE Proceedings*, vol. 137, Part G, no. 4, pp. 275-278, Aug. 1990.
- 7.1.25 B. W. Lee, B. J. Sheu, "Hardware annealing in electronic neural networks," *IEEE Trans. on Circuits and Systems*, vol. 38, no. 1, pp. 134-137, Jan. 1991.
- 7.1.26 B. W. Lee, B. J. Sheu, "Modified Hopfield neural networks for retrieving the optimal solution," *IEEE Trans. on Neural Networks*, vol. 2, no. 1, pp. 137-142, Jan. 1991.
- 7.1.27 W.-J. Hsu, B. J. Sheu, S. M. Gowda, "Reliable VLSI circuit design using simulation techniques," *IEEE Jour. of Solid-State Circuits*, vol. 26, no. 3, pp. 452-457, Mar. 1991.
- 7.1.28 B. W. Lee, H. Yang, B. J. Sheu, "Analog floating-gate synapses for general-purpose VLSI neural computation," *IEEE Transactions on Circuits and Systems*, vol. 38, no. 6, pp. 654-658, June 1991.
- 7.1.29 B. W. Lee, B. J. Sheu, "A high-speed CMOS amplifier with dynamic frequency compensation," *Jour. of Semicustom ICs*, Elsevier Science Publishers, Oxford, England, vol. 8, no. 3, pp. 42-47, 1991.
- 7.1.30 D. J. Chen, B. J. Sheu, "Automatic layout synthesis of analog ICs using circuit recognition and constraint analysis techniques," *Journal of Analog Circuits and Signal Processing*, Kluwer Academic Publishers, vol. 1, no. 1, pp. 75-87, Sept. 1991.
- 7.1.31 W.-J. Hsu, S. M. Gowda, B. J. Sheu, "Integrated-circuit reliability with hot-carrier effects emphasis," *Journal of Analog Circuits and Signal Processing*, Kluwer Academic Publishers, vol. 1, no. 3, pp. 231-245, Nov. 1991.
- 7.1.32 H. Yang, B. Sheu, J.-C. Lee, "A non-volatile analog neural memory using floating-gate MOS transistors," *Journal of Analog Circuits and Signal Processing*, Kluwer Academic Publishers, vol. 2, no. 1, pp. 19-25, Feb. 1992.

- 7.1.33 W.-J. Hsu, B. J. Sheu, S. M. Gowda, C.-G. Hwang, "Advanced integrated-circuit reliability simulation including dynamic stress effects," *IEEE Jour. of Solid-State Circuits*, vol. 27, no. 3, pp. 247-257, Mar. 1992.
- 7.1.34 D. J. Chen, B. Sheu, "A generalized approach to automatic custom layout of analog ICs," *IEE Proceedings*, vol. 139, Part G, Mar. 1992.
- 7.1.35 S. M. Gowda, B. J. Sheu, "Explicit geometry dependence of MOS transistor parameters by the pseudo-boundary method," *Jour. of Analog ICs and Signal Processing*, Kluwer Academic Publishers, vol. 2, no. 2, pp. 105-115, Apr. 1992.
- 7.1.36 J.-C. Lee, B. J. Sheu, R. Chellappa, "A VLSI neuroprocessor chip for digital image restoration using analog systolic architecture," *Jour. of VLSI Signal Processing*, vol. 4, Kluwer Academic Publishers, May 1992.
- 7.1.37 W.-C. Fang, B. J. Sheu, O. T.-C. Chen, J. Choi, "A VLSI neural processor for image data compression using self-organizing networks," *IEEE Trans. on Neural Networks*, vol. 3, no. 3, pp. 506-518, May 1992.
- 7.1.38 J.-C. Lee, B. J. Sheu, J. Choi, R. Chellappa, "A mixed-signal VLSI neuroprocessor for image restoration," *Trans. on Circuits and Systems for Video Technology*, (Express Letter), Sept. 1992.
- 7.1.39 J.-C. Lee, B. J. Sheu, R. Chellappa, "A mixed-signal VLSI competitive neuroprocessor for video motion detection," *Journal of VLSI Signal Processing*, Aug. 1992.
- 7.1.40 O. T.-C. Chen, B. J. Sheu, W.-C. Fang, "Image compression on a VLSI neural-based vector quantizer," *Jour. of Information Processing and Management*, vol. 28, no. 5, Pergamon Press Ltd.: New York, NY, 1992.
- 7.1.41 B. W. Lee, B. J. Sheu, "General-purpose neural chips with electrically programmable synapses and gain-adjustable neurons," *IEEE Jour. of Solid-State Circuits*, accepted in 1992.
- 7.1.42 J.-C. Lee, B. J. Sheu, W.-C. Fang, R. Chellappa, "VLSI neuroprocessors for video motion detection," *IEEE Trans. on Neural Networks*, accepted, 1992.
- 7.1.43 B. W. Lee, B. J. Sheu, "Paralleled Hardware Annealing for Optimal Solutions on Electronic Neural Networks," *IEEE Trans. on Neural Networks*, accepted, 1992.
- 7.1.44 J. Choi, B. J. Sheu, "A high-precision winner-take-all circuit for self-organizing neural networks," *IEEE Jour. of Solid-State Circuits*, accepted, 1992.

## 7.2 Refereed Conference Publications

- 7.2.1 B. J. Sheu, P. K. Ko, F.-C. Hsu, "Characterization of intrinsic capacitances of small-geometry MOSFET's," *IEEE 1984 Symposium on VLSI Technology*, Tech. Dig., pp. 80-81, Sept. 1984.
- 7.2.2 B. J. Sheu, P. K. Ko, "An analytical model for intrinsic capacitances of short-channel MOSFETs," *IEEE IEDM Technical Digest*, pp. 300-303, 1984.
- 7.2.3 B. J. Sheu, P. K. Ko, C. Hu, "On the high-frequency and high-speed characteristics of MOS VLSI circuits," *IEEE 2nd Symposium on VLSI Technology, Systems, and Applications*, Tech. Dig., pp. 231-234, Taipei, Taiwan, May 1985.
- 7.2.4 B. J. Sheu, D. L. Scharfetter, P. K. Ko, Tak K. Young, "BSIM, an IC process-oriented MOSFET model and the associated characterization system," *IEEE ISCAS Proceedings*, pp. 433-436, Kyoto, Japan, June 1985.
- 7.2.5 B. J. Sheu, C.-P. Wan, C.-C. Shih, W.-J. Hsu, M. C. Hsu, "Determination of Process-Dependent SPICE Parameters for Application-Specific ICs," *IEEE 1988 International Conference on Microelectronic Test Structures*, Long Beach, CA, pp. 73-78, Feb. 1988.

- 7.2.6 W.-J. Hsu, C.-C. Shih, B. J. Sheu, "RELY: A reliability simulator for VLSI circuits," IEEE Custom Integrated Circuits Conference Proceedings, pp. 27.4.1-4, Rochester, NY, May 1988.
- 7.2.7 B. W. Lee, B. J. Sheu, "An investigation on local minimum of Hopfield network for optimization circuits," IEEE International Conference on Neural Networks Proceedings, vol. I, pp. 45-51, San Diego, CA, July 1988.
- 7.2.8 A. H. Fung, D. J. Chen, Y.-N. Lai, B. J. Sheu, "Knowledge-based analog circuit synthesis with flexible architectures," IEEE International Conference on Computer Design Proceedings, pp. 48-51, Rye Brook, NY, Oct. 1988.
- 7.2.9 B. J. Sheu, W.-J. Hsu, V. C. Tyree, "Computer-aided VLSI circuit reliability assurance," International Association of Science and Technology for Development (IASTED) Symposium on Quality Control and Reliability Proceedings, Los Angeles, CA, Dec. 1988.
- 7.2.10 C.-P. Wan, B. J. Sheu, "Algorithms and modeling techniques for circuit simulation in computer integrated manufacturing," IEEE International Symposium on Circuits and Systems Proceedings, pp. 983-896, Portland, OR, May 1989.
- 7.2.11 C.-P. Wan, B. J. Sheu, "SPICE temperature modeling for VLSI design using the level-4 model," Int. Symposium on VLSI Technology, Systems, and Applications Proceedings, pp. 111-115, Taipei, Taiwan, May 1989.
- 7.2.12 B. J. Sheu, W.-J. Hsu, V. C. Tyree, "Modeling requirements for computer-aided VLSI circuit reliability assessment," Eighth Biennial University/Government/Industry Microelectronics Symposium, pp. 199-204, Westborough, Massachusetts, June 1989.
- 7.2.13 B. W. Lee, B. J. Sheu, "Electronic neural circuits with simulated annealing," IEEE International Joint Conf. on Neural Networks, vol. II, p. 615, Washington, D.C., June 1989.
- 7.2.14 B. W. Lee, B. J. Sheu, "CMOS amplifier design with enhanced slew rate and power supply rejection," IEEE 32nd Midwest Symposium on Circuits and Systems, pp. 435-438, Urbana, Illinois, Aug. 1989.
- 7.2.15 D. J. Chen, J.-C. Lee, B. J. Sheu, "SLAM: A smart analog module layout generator for mixed analog-digital VLSI design," IEEE International Conference on Computer Design, pp. 24-27, Cambridge, Massachusetts, Oct. 1989.
- 7.2.16 W.-J. Hsu, B. J. Sheu, V. C. Tyree, "Digital and analog integrated-circuit design with built-in reliability," IEEE International Conference on Computer Design, pp. 496-499, Cambridge, Massachusetts, Oct. 1989.
- 7.2.17 S. M. Gowda, B. W. Lee, B. J. Sheu, "An improved neural network approach to the traveling salesman problem," IEEE Proceedings of 4th TENCON Conference, pp. 28.1.1-28.1.4, Bombay, India, Nov. 1989.
- 7.2.18 J.-C. Lee, S. M. Gowda, B. J. Sheu, "Fully automated layout generators for high-performance analog VLSI modules," IEEE Proceedings of 4th TENCON Conference, pp. 44.1.1-44.1.4, Bombay, India, Nov. 1989.
- 7.2.19 B. J. Sheu, W.-J. Hsu, V. C. Tyree, "Reliability assurance of application-specific microelectronic circuits," Annual Reliability and Maintainability Symposium, pp. 90RM-124:1-8, Los Angeles, CA, Jan. 1990.
- 7.2.20 B. W. Lee, B. J. Sheu, "Combinatorial optimization using competitive-Hopfield neural network," IEEE IJCNN-90-WAH DC Conference, pp. II-627-630, Washington DC, Jan. 1990.
- 7.2.21 C.-P. Wan, H. Yang, B. J. Sheu, "A new method to determine effective channel widths of MOS transistors for VLSI device design," IEEE Proceedings of International Conference on Microelectronic Test Structures, pp. 217-220, San Diego, CA,



Mar. 1990.

- 7.2.22 W.-J. Hsu, B. J. Sheu, "VLSI Circuit design with built-in reliability using simulation techniques," IEEE Custom Integrated Circuits Conference Proceedings, pp. 19.3.1-19.3.4, Boston, MA, May 1990.
- 7.2.23 B. W. Lee, B. J. Sheu, "A compact and general-purpose neural chip with electrically programmable synapses," IEEE Custom Integrated Circuits Conference Proceedings, pp. 26.6.1-26.6.4, Boston, MA, May 1990.
- 7.2.24 B. W. Lee, B. J. Sheu, "A high-speed CMOS amplifier with dynamic frequency compensation," IEEE Custom Integrated Circuits Conference Proceedings, pp. 8.4.1-8.4.4, Boston, MA, May 1990.
- 7.2.25 B. W. Lee, J.-C. Lee, B. J. Sheu, "VLSI image processors using analog programmable synapses and neurons" IEEE Int. Joint Conf. on Neural Networks, vol. II, pp. 575-580, San Diego, CA, June 1990.
- 7.2.26 D. J. Chen, B. J. Sheu, "Automated layout generation for mixed analog-digital VLSI neural chips," IEEE Int. Conf. on Computer Design, pp. 29-32, Cambridge, MA, Sept. 1990.
- 7.2.27 W.-C. Fang, B. J. Sheu, J.-C. Lee, "Real-time computing of optical flow using adaptive VLSI neuroprocessors," IEEE Int. Conf. on Computer Design, pp. 122-125, Cambridge, MA, Sept. 1990.
- 7.2.28 J.-C. Lee, B. J. Sheu, "Parallel digital image restoration using adaptive VLSI neural chips," IEEE Int. Conf. on Computer Design, pp. 126-129, Cambridge, MA, Sept. 1990.
- 7.2.29 J.-C. Lee, B. J. Sheu, "Analog VLSI neuroprocessors for early vision processing," Chapter 31 in VLSI Signal Processing IV, pp. 319-328, Editors: H. S. Moscovitz, K. Yao, R. Jain, IEEE Press: New York, 1991.
- 7.2.30 W.-C. Fang, C.-Y. Chang, B. J. Sheu, "Systolic tree-searched vector quantizer for real-time image compression," Chapter 34 in VLSI Signal Processing IV, pp. 352-361, Editors: H. S. Moscovitz, K. Yao, R. Jain, IEEE Press: New York, 1991.
- 7.2.31 W.-C. Fang, B. J. Sheu, "Adaptive VLSI data compression," in Advanced Research in VLSI Conference, editor, C. H. Sequin, pp. 371-386, MIT Press: Cambridge, MA, 1991.
- 7.2.32 W.-C. Fang, B. J. Sheu, T.-C. Chen, "A neural network based VLSI vector quantizer for real-time image compression," IEEE & NASA Data Compression Conference, pp. 342-251, Snowbird, Utah, Apr. 1991.
- 7.2.33 W.-C. Fang, B. J. Sheu, "A neuroprocessor for real-time image flow computing," IEEE Int. Conf. on Acoustic, Speech and Signal Processing, pp. 2413-2416, Toronto, Ontario, Canada, May 1991.
- 7.2.34 W.-C. Fang, B. J. Sheu, "Real-time high-ratio image compression using adaptive VLSI neuroprocessors," IEEE Int. Conf. on Acoustic, Speech and Signal Processing, pp. 1173-1176, Toronto, Ontario, Canada, May 1991.
- 7.2.35 D. Chen, B. J. Sheu, "Automatic custom layout of analog ICs using constraint-based module generation," IEEE Custom Integrated Circuits Conference, pp. 5.5.1-4, San Diego, CA, May 1991.
- 7.2.36 C.-F. Chang, B. J. Sheu, W.-C. Fang, J. Choi, "A trainable analog neural chip for image compression," IEEE Custom Integrated Circuits Conference, pp. 16.1.1-4, San Diego, CA, May 1991.
- 7.2.37 W.-J. Hsu, S. M. Gowda, B. J. Sheu, "Integrated-circuit reliability simulation including dynamic stress effects," IEEE Custom Integrated Circuits Conference, pp.

- 23.2.1-4, San Diego, CA, May 1991.
- 7.2.38 S. M. Gowda, B. J. Sheu, J. S. Cable, "An accurate MOS transistor model for sub-micron VLSI circuits - BSIM\_plus," IEEE Custom Integrated Circuits Conference, pp. 23.2.1-4, San Diego, CA, May 1991.
- 7.2.39 J. Choi, B. J. Sheu, S. M. Gowda, "Analog VLSI neural network implementations of hardware annealing and winner-take-all functions," 34th Midwest Symposium on Circuits and Systems (invited paper), VLSI Analog Signal Processor Networks Session, Monterey, CA, May 1991.
- 7.2.40 J.-C. Lee, B. J. Sheu, C.-F. Chang, R. Chellappa, "Multiprocessor-based video motion detection using adaptive neural systems," International Symposium on VLSI technology, Systems, & Applications, pp. 74-78, Taipei, Taiwan, May 22-24, 1991.
- 7.2.41 B. J. Sheu, "VLSI neurocomputing with analog programmable chips and digital systolic chips," IEEE International Symposium on Circuits and Systems (invited paper), pp. 1267-1270, Singapore, June 1991.
- 7.2.42 B. J. Sheu, C.-F. Chang, T.-H. Chen, O. T.-C. Chen, "Neural-based analog trainable vector quantizer and digital systolic processors," IEEE International Symposium on Circuits and Systems (invited paper), pp. 1380-1383, Singapore, June 1991.
- 7.2.43 W.-C. Fang, B. J. Sheu, O. T.-C. Chen, "A real-time VLSI neuroprocessor for adaptive image compression based upon frequency-sensitive competitive learning," IEEE/INNS Inter. Joint Conference on Neural Networks, vol. I, pp. 429-436, Seattle, WA, July 1991.
- 7.2.44 B. J. Sheu, B. W. Lee, C.-F. Chang, "Hardware annealing for fast retrieval of optimal solutions in Hopfield neural networks," IEEE/INNS Inter. Joint Conference on Neural Networks, vol. II, pp. 327-332, Seattle, WA, July 1991.
- 7.2.45 C.-F. Chang, B. J. Sheu, "Design of a digital VLSI neuroprocessor for signal and image processing," IEEE-SP Proc. on Neural Networks for Signal Processing, pp. 606-615, Princeton, NJ, Sept. 1991.
- 7.2.46 J. Choi, B. J. Sheu, S. Forrest, "A GaAs receiver module for optoelectronic computing and interconnection," IEEE Inter. Conference on Computer Design, pp. 494-497, Cambridge, MA, Oct. 1991.
- 7.2.47 W.-J. Hsu, B. J. Sheu, S. M. Gowda, "Testing of analog array-processor neural chips," IEEE Inter. Conference on Computer Design, pp. 118-121, Cambridge, MA, Oct. 1991.
- 7.2.48 B. J. Sheu, J. Choi, C.-F. Chang, "An analog neural network processor for self-organizing mapping," IEEE International Solid-State Circuits Conference, pp. 136-137, 266, San Francisco, CA, Feb. 1992.
- 7.2.49 C.-F. Chang, B. J. Sheu, H. Okada, "Design of a multiprocessor DSP chip for flexible information processing," IEEE Int. Conf. on Acoustic, Speech and Signal Processing, vol. V, pp. 637-640, San Francisco, CA, Mar. 1992.
- 7.2.50 O. T.-C. Chen, B. J. Sheu, W.-C. Fang, "Adaptive vector quantizer for image compression using self-organization approach," IEEE Int. Conf. on Acoustic, Speech and Signal Processing, vol. II, pp. 385-388, San Francisco, CA, Mar. 1992.
- 7.2.51 S. H. Bang, B. J. Sheu, "A multi-chip module for hand-held digital cellular mobile telephone," IEEE Multi-Chip Module Conference, pp. 115-118, Santa Cruz, CA, Mar. 1992.
- 7.2.52 O. T.-C. Chen, Z. Zhang, B. J. Sheu, "An adaptive high-speed lossy data compression," Data Compression Conference, pp. 349-358, (Editors) J. Storer, M. Cohn, IEEE Computer Society Press: Los Alamitos, CA, Mar. 1992.

- 7.2.53 S. H. Bang, B. J. Sheu, "A neural-based digital communication receiver for inter-symbol interference (ISI) and white Gaussian noise channels," Proc. of IEEE International Symposium on Circuits and Systems, San Diego, CA, May 1992.
- 7.2.54 S. M. Gowda, B. J. Sheu, J. Choi, "Testing of programmable analog neural network processors," IEEE Custom Integrated Circuits Conference, pp. 17.1.1-4, Boston, MA, May 1992.
- 7.2.55 J. Choi, B. J. Sheu, "VLSI design of compact and high-precision analog neural network processors," IEEE/INNS Inter. Joint Conference on Neural Networks, Baltimore, MD, June 1992.
- 7.2.56 C.-F. Chang, B. J. Sheu, "Digital VLSI multiprocessor design for neurocomputers," IEEE/INNS Inter. Joint Conference on Neural Networks, Baltimore, MD, June 1992.
- 7.2.57 S. H. Bang, B. J. Sheu, "Neural network communication receiver based on the non-linear filtering," IEEE/INNS Inter. Joint Conference on Neural Networks, Baltimore, MD, June 1992.
- 7.2.58 H. Okada, B. J. Sheu, C.-F. Chang, "An analog VLSI edge detection chip and digital multiprocessor chip for neural-based vision processing," IEEE International Conference on Systems Engineering (invited paper), Kobe, Japan, Sept. 1992.
- 7.2.59 O. T.-C. Chen, Z. Zhang, B. J. Sheu "An adaptive high-speed vector quantization," in VLSI Signal Processing V, Editors: K. Yao, R. Jain, I. Rabaey, IEEE Press: New York, 1993.
- 7.2.60 W.-C. Fang, B. J. Sheu, "High-speed VLSI pipelined processor design for lossless image data compression," in VLSI Signal Processing V, Editors: K. Yao, R. Jain, I. Rabaey, IEEE Press: New York, 1993.

### 7.3 Books and Book Chapters

- 7.3.1 B. W. Lee, B. J. Sheu, *Hardware Annealing in Analog VLSI Neurocomputing*, 234 pp., Kluwer Academic Publishers: Boston, MA, 1991.
- 7.3.2 B. W. Lee, B. J. Sheu, *Design and Analysis of VLSI Neural Networks*, Chapter 8 in *Neural Networks for Signal Processing*, pp. 229-286, B. Kosko, editor, Prentice-Hall: Englewood Cliffs, NJ, 1992.
- 7.3.3 B. J. Sheu, W.-C. Fang, *Image and Video Compression with VLSI Processors*, in progress, Kluwer Academic Publishers: Boston, MA.