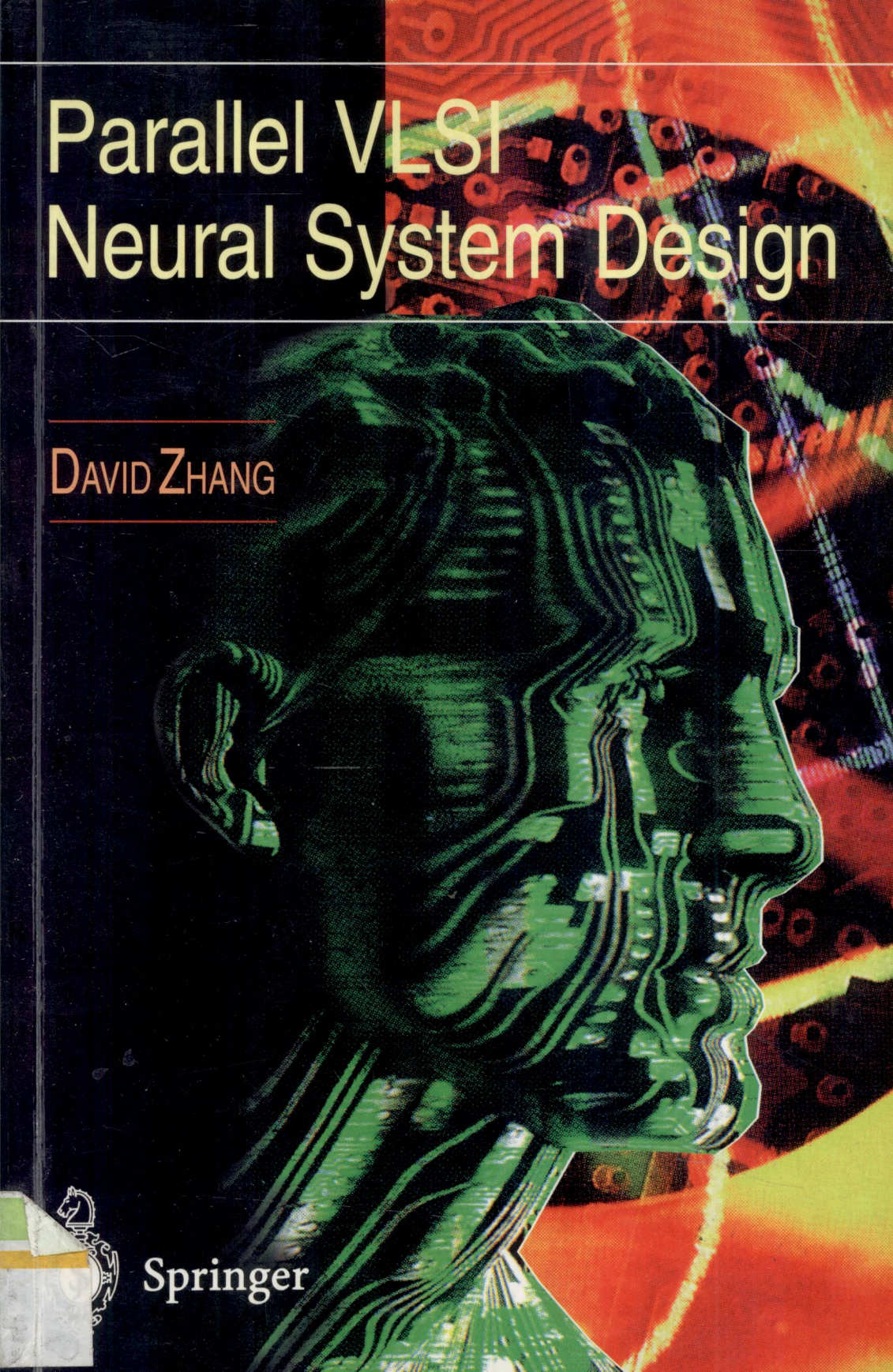
Parallel VLSI Neural System Design

DAVID ZHANG



Springer

Parallel VLSI Neural System Design

DAVID ZHANG

Springer

Dr. David Zhang
Department of Computer Science
Hong Kong Polytechnic University
Hung Hom Kowloon
Hong Kong

Library of Congress Cataloging-in-Publication Data

Parallel VLSI Neural System Design / David Zhang
p. cm.
Includes bibliographical references and index
ISBN 9813083301
1. Neural Networks (Computer science). 2.Parallel processing (Electronic computers). 3. Integrated circuits--Very large scale integration. I. David Zhang, 1949- . II. Title.
QA76.87.Z473 1998
006.3/2 --dc21 98-41790
CIP

ISBN 981-3083-30-1

Typesetting: Camera-ready by author
SPIN 10628591 5 4 3 2 1 0

Preface

Neural information processing has been emerging as a new field. This type of processing is an alternative form of computation that attempts to mimic the functionality of the biological human brain in solving demanding pattern recognition problems. However, researchers and engineers have long been fascinated by how efficient and how fast *artificial neural networks* (ANNs) are capable of performing such complex tasks as recognition. Such networks are capable of recognizing input data from *any of the five senses* with the necessary accuracy and speed to allow living creatures to survive. Machines which perform such complex tasks as recognition, with similar accuracy and speed, were difficult to be implemented until the technological advances of integrated circuits and VLSI systems. Since then, VLSI neural systems have witnessed an exponential growth and a new engineering discipline was born.

There were a number of excellent text and reference books on the subject, each dealing with one or two topics. This book attempts to present a parallel VLSI neural system design methodology for pattern recognition applications. The methodology emphasizes a coordination between model definition, architectural description, and hardware implementation. Depending on the different pattern recognition applications, the methodology provides appropriate ANN models suited to parallel/pipeline processing, mapping the models onto the corresponding VLSI architectures and finally hardware implementation.

A parallel ANN model is a basis of VLSI design because it directly reflects highly parallel, regular and modular VLSI architectures. Such three kinds of parallel ANN models, including an unsupervised learning model for fuzzy clustering, a supervised training model for pattern classification, and a neural-like network model for finite ring computing, are developed in Chapters 2, 3, and 4, respectively.

VLSI technology offers a highly advanced implementation medium both at the fabrication and the CAD level if efficient architectures can be provided. In Chapter 5, the mapping policies from ANNs to systolic arrays are given. Three typical VLSI architectures based on the effective matching policies are introduced in Chapters 6-8. They are a parallel architecture built by systolic arrays, a pipeline architecture based on window operation, and a simplified architecture using a priori knowledge.

The implementation of ANN architectures into silicon as special-purpose hardware is an important step in parallel VLSI neural system design. In Chapter 9, a computational block design for digital ANN based on dynamic pipelines is proposed. Chapter 10 presents a digital array compressor design based on C^2PL (complex complementary pass-transistor logic). In Chapter 11, a hybrid programmable ANN design using BiCMOS circuit building blocks is described. As an example of VLSI implementation, a finite ring ANN given in Chapter 4 is also implemented in Chapter 12.

The effectiveness of parallel VLSI neural system design methodology is illustrated by applying the designs to various pattern recognition applications, and analyzing the performances of the given systems.

This book is not a primer in ANN, in that a certain amount of prior knowledge, such as parallel processing and VLSI design, is assumed. It is my hope that this book will contribute to our understanding of this new and exciting discipline: Parallel VLSI Neural System Engineering.

David D. Zhang
The Hong Kong Polytechnic University

Acknowledgements

My sincere thanks goes to Professors M.I. Elmasry and M. Kamel at University of Waterloo, Ontario, Canada, for their support and advice throughout this research. I would like to thank Dr. Harold H. Szu for his useful comments on artificial neural networks. I would also like to express my gratitude to Professor G.A. Jullien and Professor W.C. Miller at University of Windsor, Ontario, Canada, for their previous supports and their encouragement to me to study VLSI design. Special thanks are due to my research assistant, Lei Huang, for his help in the preparation of this book. The support from the University Grants Committee in Hong Kong is appreciated.

Contents

1 VLSI Neural System Design Methodology

In this chapter, we begin by introducing the background for artificial neural networks (ANNs). Then, the existing ANN models, architectures and hardware implementation techniques are briefly reviewed. This leads to the definition of the research objective and provides insight into a parallel VLSI neural system design methodology.

1.1 Introduction

ANNs are massively parallel interconnected networks of simple (usually adaptive) nodes which are intended to interact with objects of the real world in the same way as biological nervous systems do [1].

The interest in these networks is due to the general opinion that they are able to perform some complicated and creative tasks, such as pattern recognition, similar to the way they are performed by human brains [2,10,35]. The implementations of these tasks by traditional computing methods have only reached relatively low performances in some limited aspects or environments. Nevertheless, as neural systems show some properties, like association, generalization, parallel searching, and adaptation to changes in the environment, which are analogous to human brain properties, they promise improved results.

The usage of ANNs for pattern recognition may be traced back to the perceptron models originated by Rosenblatt in 1950 [2]. The perceptron models used the concept of reward and punishment. In late 1960s, the progress in ANN models slowed down due to the limited capabilities of the early single layer perceptron models. In the mid-1970s and early 1980s, with the availability of enhanced computing power the progress in the development of ANN models accelerated. Researchers were able to model and test their theories about the functioning of the brain.

Today a number of well-developed theories and models of ANNs are available [3,34-36,40-47,55-64,198-201]. These networks consist of a large number of simple processing elements called nodes that represent the neurons. These nodes are interconnected by the synaptic connections. These models are capable of learning and making decisions; and are suitable for a variety of pattern recognition tasks [36,39].

Pattern recognition techniques can be grouped into two classes: supervised and unsupervised techniques. In supervised methods, certain number of samples is available for each category and these samples are used to train the classifier. In the case of unsupervised classification no training samples are available, and the network learns by detecting the similarity between the input patterns.

Now many ANN models and algorithms for pattern recognition applications are available. They include Back-Propagation (BP) learning, Competitive learning, Kohonen learning, Adaptive Resonance Theory, Neocognitron models, Hopfield Networks, and Boltzman machines [5-13]. Applications of ANNs include character recognition, human face identification, speech recognition, multispectral image analysis, and expert systems [14-18]. The BP learning is essentially of the supervised type and the network learns with the help of training sets. The BP networks have been successfully used for many pattern recognition problems [5,15-17].

Another important class of neural networks is self-organizing neural networks. The networks with competitive learning algorithms are self-organizing networks. Early models of competitive learning were developed by Malsburg in his study of visual cortex [19]. Rumelhart and Zipser have suggested an algorithm for competitive learning [9]. The main disadvantage of competitive learning is that the network forgets its earlier learning with new learning and the network may get set into an unstable state with the spurious input patterns. To overcome this drawback, Grossberg developed an adaptive resonance architecture [10-11] and Fukushima proposed the Neocognitron models [12]. Kohonen developed a learning paradigm for self-organizing networks known as Kohonen learning [6-7]. These algorithms can be used for a variety of tasks in pattern recognition.

ANNs consist of parallel distributed processing (PDP) models. The PDP models are well described in the work of Rumelhart and McCelland [4]. The functional synthesis of these models consists of establishing a relationship between the several inputs and one or more outputs. In ANN, the nodes are connected to each other by the synaptic connections or the links. There is an

associated synaptic strength or a weight with each connection. During the learning, the weights which represent the knowledge stored in the network are updated. The ANNs consist of two or several layers of nodes and each layer contains several nodes. The observed feature vector is presented to the input nodes. The input values may represent the probability that the discrete feature is present. Each possible decision or outcome can be represented by a node in the output layer.

1.2 Neural Network Models

ANNs are information processing systems. Due to the variety of fields of interest and applications, a broad range of ANN models has been emerged. In all these fields, the term "neural networks" is characterized by a combination of adaptive learning algorithms and parallel distributed implementations. Although ANNs are biologically motivated, their resemblance to the brain models is not straightforward. Since the basic concepts of biological neural networks provide a common ground for understanding the ANN models, we will first introduce basic biological neural networks in order to appreciate the differences and similarities with ANN.

1.2.1 Biological Neural Networks

As you know, the von Neumann computer performs poorly on certain tasks, such as pattern recognition that humans handle routinely. It is interesting to compare the human brain with a serial modern von Neumann computer from the information processing aspects as shown in Table 1.1. Although the neuron's switching time (a few milliseconds) is about a million times slower than modern computer elements, they have a thousand-fold greater connectivity than today's supercomputers. Furthermore, the brain is very powerfully efficient – it consumes less than 100 watts – by contrast a supercomputer may dissipate 10^5 watts.

The basic element of neural networks of a brain is a neuron. The neurons consist of four basic parts: cell body, synapses, axons, and dendrites. The cell body essentially sums the membrane potential provided by the synapses. The synapses provide an output. Axons are the connections between the neurons that carry charge, and the dendrites are the branch-like structures, which provide the sensory input to a cell body (See Fig.1.1).

In fact, the synapse represents the junction between an axon and a dendrite. How two or more neurons interact remains largely mysterious, and complexities of different neurons vary greatly. Generally speaking, a neuron sends its output to other neurons via its axon. An axon carries information through a series of action potentials, or waves of current, which depends on the neuron's voltage potential. More precisely, the membrane generates the action potential and propagates down the axon and its branches, where axonal insulators restore and amplify the signal as it propagates, until it arrives at a synaptic junction.

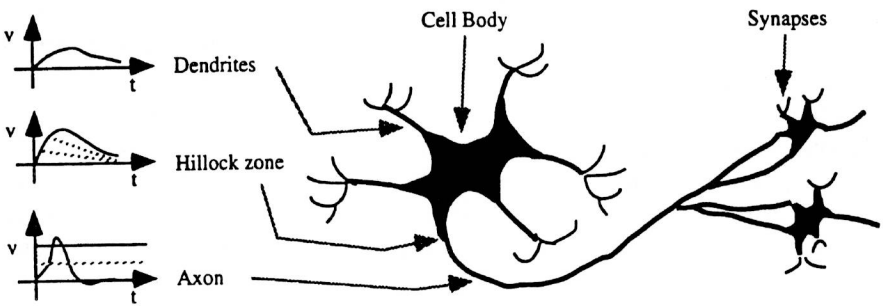


Fig.1.1 A prototype biological neuron

Table 1.1 Comparison between human brain and modern computer

	Human Brain	*Modern Computer*
Element	10^{11} neurons 10^{14} synapses	10^{9} transistors
Fanout	10^{3}	3
Implementation	Analog	Digital
Processing	passively parallel	largely serial
Switching Time	10^{-3} s	10^{-9} s
Power	100 watts	10^{5} watts

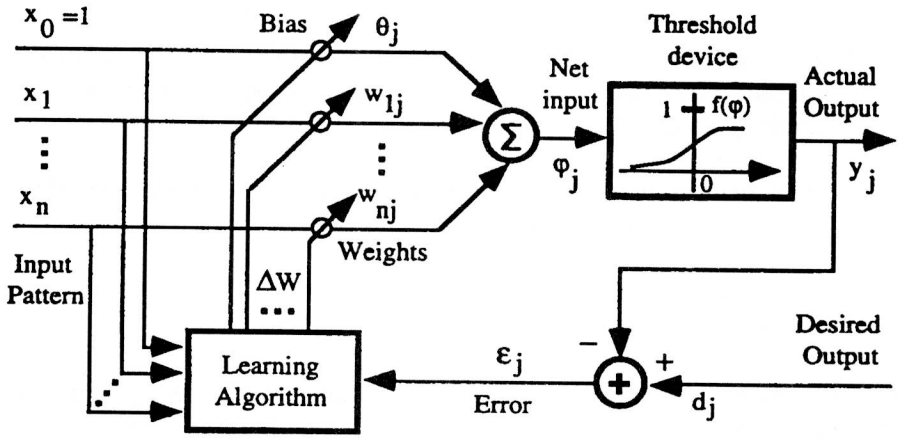


Fig.1.2 A typical artificial neuron model

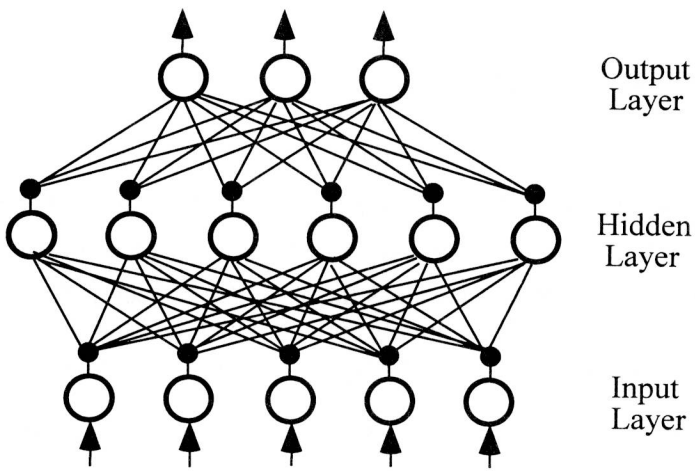


Fig.1.3 A three-layer feedforward network

There are several important features obtained from biological neural networks, which apply to all ANNs:

1) Each neuron acts independently of all others – each neuron's output relies only on its constantly available inputs from the abutting connections.

2) Each neuron relies only on local information – the information that is provided by the adjoining connections.

3) The large number of connections provides a large amount of redundancy and facilitates a distributed representation.

First two features allow neural networks to operate efficiently in parallel. The last one provides neural networks with inherent fault-tolerance.

1.2.2 Artificial Neural Networks

ANNs mimic the functioning of the neural networks of a brain. ANN consists of a large number of simple nodes. Each of the nodes is connected to another node(s) through a synaptic connection or a link [34].

Information processing takes place through the interaction between the nodes. Each node is associated with an activation value $\varphi_j(t)$. The activation value passes through an activation function $f(\varphi_j)$ to provide an actual output $y_j(t)$. These outputs pass through the unidirectional synaptic connections. There is an associated number, w_{ij}, called the weight or the connection strength, that determines the amount of effect node *i* can have on node *j*. For each node all the inputs are combined, and the total input, along with the current activation, determines the new activation value (Fig.1.2).

Usually ANNs consist of a number of layers and nodes in each layer. The most general model assumes the complete interconnections between all the nodes, and resolves the cases of the nonconnected nodes (*i, j*) by setting the weights $w_{ij} = 0$. A simple three-layer feedforward network is shown in Fig.1.3. The networks can be synchronous or asynchronous. The synchronous networks are controlled by clock pulses; whereas in asynchronous networks the nodes respond instantaneously to the incoming inputs.