VLSI DESIGN OF A NEUROHARDWARE PROCESSOR IMPLEMENTING THE KOHONEN NEURAL NETWORK ALGORITHM

AVINASH RAJAH

UNIVERSITI TEKNOLOGI MALAYSIA

VLSI DESIGN OF A NEUROHARDWARE PROCESSOR IMPLEMENTING THE KOHONEN NEURAL NETWORK ALGORITHM

AVINASH RAJAH

A thesis submitted in fulfilment of the
requirements for the award of the degree of
Master of Engineering (Electrical)

Faculty of Electrical Engineering
Universiti Teknologi Malaysia

DECEMBER 2005

ABSTRACT

As artificial neural networks continue to gain popularity in the domain of pattern recognition, there have been growing demands for these models to be executed at high-speeds. Thus, to cater to this need, the VLSI design and implementation of a neurohardware for high-speed pattern recognition is proposed in this research. The UTM-Neuroprocessor implements the Kohonen Neural Network for pattern classification. High-speed pattern classification by the neural paradigm is achieved through massively parallel execution based on the neuron-parallel processing approach. For proof of concept purposes, a 10x10 UTM-Neuroprocessor, which implements a 10x10 Kohonen network, was developed in this work. The design and rapid FPGA prototyping of the neuroprocessor was achieved using VHDL and the Altera Nios embedded system development kit. The FPGA-based prototype of the 10x10 UTM-Neuroprocessor is able to function at a frequency of 100 MHz and delivers performances up to 5.079 GCPS and 2.285 GCUPS. Software components, including a VB-based GUI, were also developed to allow execution of pattern recognition applications on the UTM-Neuroprocessor. For efficient VLSI implementation of the UTM-Neuroprocessor, the combined FPGA-VLSI approach was proposed. Correspondingly, the VLSI design of a 2x2 array computation engine, termed the Array_2x2 microchip, was developed in the AMI 0.5µm process technology and fabricated at the Europractice IC foundry. The fabricated Array_2x2 microchip can be applied to produce a 2x2 UTM-Neuroprocessor, in the combined FPGA-VLSI implementation approach. The design consumes an area of 16.9 mm^2 on silicon and is encapsulated in 84-pin PGA package. SPICE simulations of the Array_2x2 design proved functionality at an operating frequency of 90 MHz. The microchip is able to deliver performances of up to 169.41 MCPS and 75.78 MCUPS MCUPS for a 2x2 UTM-Neuroprocessor.

ABSTRAK

Pertumbuhan dalam penggunaan teknologi rangkaian saraf (neural networks) dalam pelbagai bidang aplikasi telah mewujudkan keperluan untuk perkakasan mikroelektronik canggih yang mampu melaksanakan rangkaian saraf pada kelajuan tinggi. Oleh demikian, implementasi VLSI bagi sebuah pemproses yang melaksanakan rangkaian saraf Kohonen pada kelajuan tinggi untuk aplikasi pengecaman corak, telah dicadangkan dalam kajian ini. Pengecaman corak pada kelajuan tinggi oleh UTM-Neuroprocessor direalisasikan menggunakan kaedah pemprosesan selari. Bagi tujuan pemprototaipan, rekabentuk sebuah UTM-Neuroprocessor, yang melaksanakan rangkaian Kohonen 10x10, telah dibangunkan. Pembangunan rekabentuk 10x10 UTM-Neuroprocessor telah menggunakan VHDL dan kit pembangunan sistem terbenam Altera Nios. Rekabentuk yang dibangunkan telah diprototaip segera pada FPGA dan mampu mencapai kelajuan setinggi 100 MHz. Beberapa aturcara perisian juga telah dibangunkan bersama, untuk membolehkan pemprosesan aplikasi pengecaman corak pada 10x10 UTM-Neuroprocessor. Menggunakan aturcara yang telah dibangunkan, beberapa aplikasi pengecaman corak dunia sebenar telah dilaksanakan. UTM-Neuroprocessor telah didapati mampu menawarkan kuasa pemprosesan setinggi 5.079 GCPS dan 2.285 GCUPS untuk aplikasi-aplikasi tersebut. Bagi implementasi VLSI UTM-Neuroprocessor, kaedah FPGA-VLSI tergabung telah dicadangkan. Berdasarkan cadangan tersebut, sebuah mikrocip yang melaksanakan rangkaian Kohonen 2x2 telah direkabentuk dengan proses teknologi AMI 0.5µm. Mikrochip tersebut telah difabrikasi di Europractice, Belgium dan boleh digunakan untuk membangunkan sebuah 2x2 UTM-Neuroprocessor dalam kaedah implementasi FPGA-VLSI tergabung. Simulasi SPICE telah membuktikan kefungsian rekabentuk mikrocip tersebut pada 90 MHz. Pada kelajuan ini, Array_2x2 membolehkan kuasa pemprosesan setinggi 169.41 MCPS dan 75.78 MCUPS dicapai oleh sebuah 2x2 UTM-Neuroprocessor.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ASIC	–	Application Specific Integrated Circuit
CAD	–	Computer Aided Design
CMOS	–	Complementary Metal Oxide Semiconductor
CPU	–	Central Processing Unit
DMA	–	Direct Memory Access
EDA	–	Electronic Design Automation
FPGA	–	Field Programmable Gate Array
FSM	–	Finite State Machine
GUI	–	Graphical User Interface
HDL	–	Hardware Description Language
I/O	–	Input / Output
IP	–	Intellectual Property
LE	–	Logic Element
MCPS	–	Millions of Connections Per Second
MCUPS	–	Millions of Connections Update Per Second
NN	–	Neural Network
PC	–	Personal Computer
RAM	–	Random Access Memory
SIMD	–	Single Instruction Multiple Data
SoC	–	System On Chip
SOM	–	Self-Organizing Map
SRAM	–	Static Random Access Memory
VB	–	Visual Basics
VHDL	–	VHSIC Hardware Description Language
VHSIC	–	Very High Speed Integrated Circuit
VLSI	–	Very High Scale Integrated Circuit

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

This thesis proposes the VLSI design and implementation of a neural network hardware, or neurohardware for pattern recognition. The aim is to produce a neurohardware that executes the Kohonen Neural Network at massive parallelism for high-speed pattern classification and serves as a comprehensive computing platform for pattern recognition applications. In this first chapter, the background of the domain is discussed, providing the rationale and focus points behind the research.

1. 1 Background and Motivation

Although conventional logic based computing has been successful in many applications, it has not been effective in solving a variety of critical and complex problems. At the same time, these perplexing problems are solved trivially and routinely in real-time by human beings. It is this intriguing predicament that has led to the study of information processing by the human brain and subsequently the emergence of artificial neural networks. Artificial neural networks, or simply neural networks, attempt to mimic the computational power of the mammalian brain by massively interconnecting very simple computational units called neurons (Misra 1997). The adoption of this similar design philosophy provides neural networks with the ability to learn and solve tasks challenging to conventional computing.

Neural networks have found a wide range of applications, with the majority associated with the pattern recognition domain. The pattern classification attribute of neural networks have been instrumental in the following examples of pattern recognition applications; predictive and preventive maintenance, condition monitoring, character recognition, speech synthesis, intelligence based medical diagnosis and intrusion detection and predictive penetration services in computer networks.

As artificial neural networks gain popularity for pattern recognition in a variety of application domains, it is critical that these models are able to be executed speedily and generate results in real-time (Lindsey 1998). Although a number of implementations of neural networks are available on conventional general purpose machines, most of these implementations require an inordinate amount of time to train or run neural networks and are not able to provide real-time response, especially when the network sizes are large. Large network sizes are often required in real world applications and execution performances by these machines are simply unacceptable. This drawback is apparently due to the fact that general purpose computers are traditionally based on the von-Neumann architecture which is sequential in nature (Schoenauer 1998). Artificial neural networks on the other hand have a parallel structure by conception.

The most obvious solution to this problem is to accelerate the execution artificial neural network algorithms is through simulation on dedicated parallel hardware. The massively parallel and distributed processing brand of neural networks suggest massively parallel hardware as an obvious implementation choice to obtain appropriate algorithm-architecture matching and high execution speeds. When implemented in parallel hardware, neural networks can take full advantage of their inherent parallelism and run in orders of magnitude faster than software simulations on sequential machines. Parallel processing with multiple simple processing elements working together can provide tremendous speed-ups in neural network and produce real-time responses and fast learning phases.

Consequently, a new breed of hardware, termed neurohardware, have emerged. Neurohardware is typically defined as dedicated hardware designed to

implement neural algorithms and take full advantage of the inherent parallelism in neural networks through parallel processing. At present, a wide spectrum of neurohardware implementations that primarily differ in terms of performance-space compromise, degree of parallelism and system architecture approaches are available. However, with the continuous burgeoning of neural paradigms and increasing applicability of neural networks in real-time based applications, there is a great demand and market for massively parallel and dedicated neurohardware.

1.2 Problem Statement

Neurohardware providing parallel execution platforms for neural networks can adopt two different architectural directions in doing so; general-purpose architectures that emulate a wide range of neural network models, and special-purpose architectures dedicated to a specific neural paradigm (Ruckert 2002). Dedicated implementations are able to be built at a low hardware cost to execute the algorithm more quickly and efficiently compared to general-purpose architectures. The speed achieved by special-purpose architectures is unattainable by general-purpose architectures (Liao 2001). Therefore, special-purpose architectures would be viable for neurohardware targeting high-speed execution of specific neural paradigms.

Neural network can be effectively grouped into three categories that are distinguishable by their learning approaches; supervised, reinforcement and unsupervised (Cheang 2003). Unsupervised learning possesses a number of advantages over the other types of learning, which includes faster training and execution for large networks. This brand of networks would be suitable for pattern recognition problems, given their ability to detect structures and relations in data that are not so apparent. One such neural paradigm that has been successful in pattern classification and recognition applications is the Kohonen neural network (NN) algorithm. The Kohonen NN is a proven algorithm and could be easily mapped onto hardware than other neural paradigms (Glesner and Pochmuller 1994).

In developing the ASIC implementation of neurohardware, two main stream technologies can be considered. FPGAs have evolved tremendously under the current advancements of VLSI process technologies. The flexibility and reconfigurability of FPGAs advantageously support parameterized designs and rapid prototyping of advanced hardware architectures. VLSI implementations on the other hand are able to guarantee higher compaction and speed for hardware designs, compared to FPGAs. However, both technologies can be jointly utilized and advantageously exploited for implementation of neurohardware.

1.3 Objectives

From the discussion in the preceding sections, the objectives of the work presented in this thesis are as follows:

1) To design a neurohardware that provides high-speed pattern classification. The Kohonen NN algorithm is to be implemented, in a massively parallel and dedicated manner, to deliver high-speed pattern classification.

2) To design a neurohardware that serves as a comprehensive computing platform for pattern recognition applications, based on the Kohonen NN algorithm.

3) To propose a viable VLSI implementation approach for the designed neurohardware and to develop a prototype for demonstration of real-world pattern recognition applications.

1.4 Scope of Work

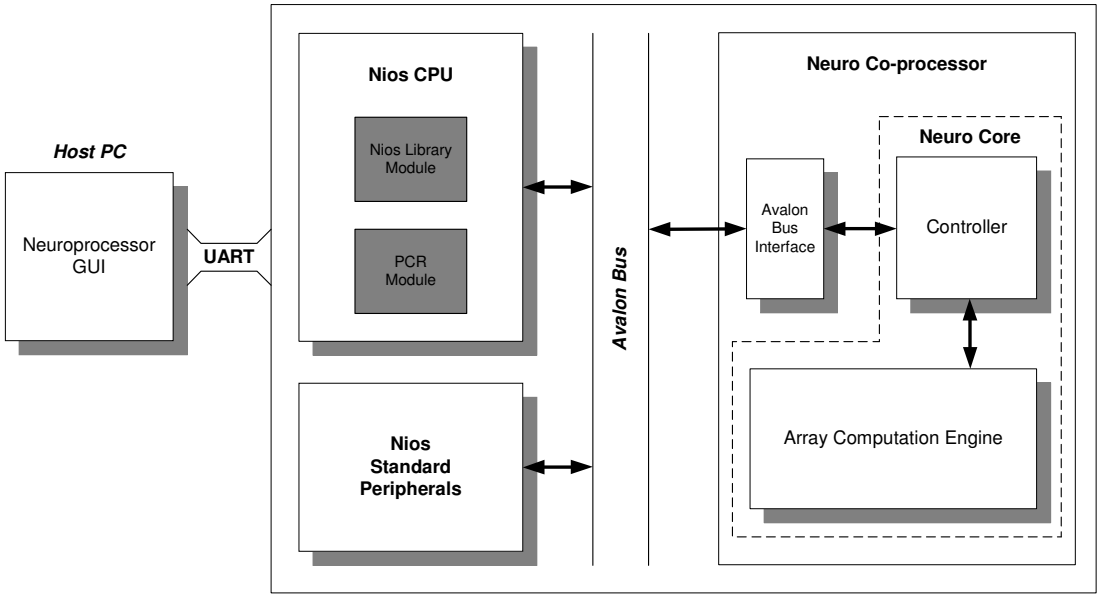

Figure 1.1 : Top-Level Block Diagram of UTM-Neuroprocessor

Based on the outlined objectives, the neurohardware design illustrated by Figure 1.1, the UTM-Neuroprocessor, is proposed in this work. The scope of work involved in producing the proposed neurohardware is as follows:

1) Comprehensive study of the Kohonen NN algorithm and determining necessary algorithmic modifications for efficient hardware implementation of the algorithm.

2) Design and FPGA prototyping of the UTM-Neuroprocessor, using the Altera Nios embedded system development kit. The Neuro co-processor that executes the Kohonen NN at massive parallelism is designed using VHDL.

3) Development of the software components of the UTM-Neuroprocessor. The embedded software component, the PCR module, is developed in C while the GUI program is developed using Visual Basics.

4) VLSI implementation and fabrication of the array computation engine in the AMI 0.5 µm process technology, to realize implementation of the UTM-Neuroprocessor in the proposed combined FPGA-VLSI approach.

5) Demonstration of real-world pattern recognition datasets on the UTM-Neuroprocessor. Datasets from selected application domains are used to verify the classification speed and viability of the neuroprocessor as a computing platform for pattern recognition applications.

1.5 Research Contributions

1) A systematic study and modification of the Kohonen NN algorithm for efficient hardware implementation.

2) A comprehensive design and prototyping flow for FPGA-based embedded systems using the Altera Nios development kit.

3) A viable ASIC design methodology for the UTM-ECAD VLSI Laboratory, based on the AMI 0.5µm process technology. The design methodology incorporates industry standard EDA tools and is applicable from design entry stages to tape-out.

1.6 Organization of Thesis

The work in this thesis is presented conveniently over eight chapters. The first chapter outlines the motivations and objectives of the thesis and subsequently presents the scope of work involved in meeting the research objectives.

The second chapter provides brief summaries of the literature reviewed prior to engaging in the mentioned scope of work. Review of literature on previously attempted efforts assists in achieving the research objectives.

Chapter three presents the VLSI design methods and tools adopted in producing the FPGA prototyping and ASIC implementation of the neurohardware in this work.

Chapter four provides a detailed discussion on the implemented Kohonen NN algorithm and outlines the necessary algorithmic modifications. Based on the modifications, the architectural and design specifications of the neurohardware is ascertained.

Chapter five delivers a description of the top-level architecture and behaviour of the UTM-Neuroprocessor. Subsequently, the focus is shifted to the design of the hardware module which implements the Kohonen NN algorithm at massive parallelism for the neuroprocessor, is detailed elaborately in the chapter.

Chapter six dwells into the embedded system design and prototyping of the UTM-Neuroprocessor using the Altera Nios development kit. The software components of the neuroprocessor are also discussed in the chapter.

Chapter seven focuses on the VLSI implementation of the array computation engine, for implementation of the UTM-Neuroprocessor in the combined FPGA-VLSI approach. The chapter also presents ASIC design and fabrication of a prototype design of the computation engine, termed the Array_2x2 microchip, in the AMI 0.5 µm process technology.

Chapter eight provides details the application demonstration work on the UTM-Neuroprocessor, using real-world pattern recognition datasets. Performance evaluation and benchmarking of the neuroprocessor against previous works are also reported by the chapter.

In the final chapter of the thesis, the research work is summarized and deliverables of the research are stated. Potential extensions and improvements to the design are also given.

1.7 Summary

In this chapter, a brief introduction was given to the background and motivation. The need for neurohardware that executed neural networks at massive parallelism for high-speed pattern classification was identified. Correspondingly, several objectives were outlined to meet this need in the research. The UTM-Neuroprocessor was proposed to fulfill the objectives of the research. The UTM-Neuroprocessor executes the Kohonen Neural Network at massive parallelism for high-speed pattern classification and serves as a comprehensive computing platform for pattern recognition applications. The following chapter will discuss some literature relevant to producing the proposed neurohardware and covers previous works accomplished on the design of neurohardware for the similar objectives.

REFERENCES

Ahmad, R., Bambang, S.S., Rahman, W. and Rais, A. (1999). "Development of Optimized Digital CMOS Standard Cell Library". Proceeding of World Engineering Congress (WEC99). 159-161.

Altera Corporation. (2003a). "Nios Hardware Development Tutorial". Altera Corporation.

Altera Corporation. (2003b). "Introduction to Quartus II". Altera Corporation.

Altera Corporation. (2003c). "Sopc Builder Data Sheet". Altera Corporation.

Altera Corporation. (2003d). "Nios Software Development Tutorial". Altera Corporation.

Altera Corporation (2003e). "Nios Development Board: Reference Manual, Stratic Professional Edition". Altera Corporation.

Altera Corporation. (2003f). "Stratix Device Handbook: Volume 1". Altera Corporation.

Asral B.J, Ahmad, R., Rais, A. (1999). "Standard Cell Library Development." Proceeding of IEEE International Conference on Microelectronics (ICM). 161-163.

Beale, R. and Jackson, T. (1990). "Neural Computing: An Introduction". Adam Hilger, IOP Publishing Ltd.

Brown, S., and Vranesic, Z. (2000). "Fundamentals of Digital Logic with VHDL Design". Singapore: McGraw_Hill.

Burr, J.B. (1992). "Digital Parallel Implementations of Neural Networks". Prentice Hall. 223-281.

Carpinelli, J. D. (2001). "Computer Systems, Organization & Architecture". USA: Addison Wesley.

Cheang, CH. (2003). "A Digital Neurohardware Implementation of Kohonen Neural Network for Pattern Recognition". Universiti Teknologi Malaysia.

Fausett, L. (1994). "Fundamentals of Neural Networks". Prentice-Hall Inc.

Fischer, T., Eppler, W., Gemmeke, H., Kock, G. and Becher, T. (1997). "The SAND Neurochip and its Embedding in the MiND System". 7th International Conference on Artificial Neural Networks.

Fisher, R.A. (1936). "The Use of Multiple Measurements in Taxonomic Problems". Annual Eugenics, 7, Part II, 179-188.

Geus, A. J. (1989). "Logic synthesis speeds ASIC Design". IEEE Spectrum. August 1989. 27-31.

Glesner, M. and Pochmuller, W. (1994). "Neurocomputers – An overview of neural networks in VLSI". Chapman and Hall.

Haykin, S. (1994). "Neural Networks: A Comprehensive Foundation". Macmillan College Publishing Company.

Honkela, T., Kaski, S., Lagus, K. and Kohonen, T. (1997). "Websom --- self-organizing maps of document collections". Workshop on Self-Organizing Maps.

Ienne, P. (1995). “Digital Hardware Architectures for Neural Networks”. SPEEDUP Journal, Vol. 9, No. 1.

Kohonen, T. (1995). “Self-Organizing Maps”. Springer-Verlag.

Kohonen, T. (1988). “The neural phonetic typewriter”. Computer.

Kohonen, T., Torkkola, K., Shozakai, M.,.Kangas, J. and Venta, O. (1988). “Phonetic Typewriter for Finnish and Japanese”. Proceedings of the IEEE 1988 International Conference on Acoustics, Speech and Signal Processing.

Karnik, T. (2000). “Microprocessor Layout Method”. in Chen Wai-Kai. “The VLSI Handbook”. USA: CRC Press. 62.1-62.28.

Kurup, P. and Abbasi, T. (1997). “Logic Synthesis Using Synopsys”. USA: Kluwer Academic Publishers.

Lindsey, Clark S. (1998). “Neural Networks in Hardware: Architectures, Products and Applications”. www.particle.kth.se/~lindsey.

Manavendra Misra (1997). “Parallel Environments for Implementing Neural Networks”. Neural Computing Surveys. Vol 1. 48-60.

Martin-Del-Brio, B., Medrano-Marques, N. and Hernandez-Sanchez, S. “A Low-Cost Neuroprocessor Board for Emulating the SOFM Neural Model”. 5th IEEE International Conference on Electronics, Circuits and Systems.

Melton, M.S., Tan Phan, Reeves, D.S. and Van den Bout, D.E. (1992). “The TInMANN VLSI Chip”. IEEE Transactions on Neural Networks.

Moerland, P.D. and Fiesler, E. (1996). “Hardware-Friendly Learning Algorithms for Neural Networks: An Overview”. Fifth International Conference on Microelectronics for Neural Networks and Fuzzy Systems.

Moerland, P. and Fiesler, E. (1997). “Neural Network Adaptations to Hardware Implementations”. Handbook of Neural Computation E1.2. 1-13.

Mohamed Khalil and Koay, K.H. (1999). “VHDL Module Generator: A Rapid-prototyping Design Entry Tool for Digital ASICs”. Jurnal Teknologi.

Muroga, S. (2000). “Cell-Library Design Approach”, in Chen Wai-Kai. “The VLSI Handbook”. USA: CRC Press. 45.1-45.3.

Murphy, P.M. and Aha, D.W. (1994). “UCI Repository of machine learning databases”. University of California, Department of Information and Computer Science. (http://www.ics.uci.edu/~mlearn/MLRepository.html).

Patterson, D.W. (1996). “Artificial Neural Networks: Theory and Applications”. Prentice Hall.

Peter van der Putten. (1996). “Utilizing the Topology Preserving Property of Self-Organizing Maps for Classification”. Utrecht University: Masters thesis.

Rabaey, J. (2003). “Digital Integrated Circuits: A Design Perspective”. Usa: Prentice-Hall. 2nd Edition.

Ruping, S., Ruckert, U and Goser, K. (1993). “Hardware Design For SOFM with Binary Input Vectors”. “New Trends in Neural Computation, Lecture Notes in Computer Science”. Springer Verlag, Berlin.

Ruping, S., Ruckert, U and Goser, K. (1994). “A Chip for Self Organizing Feature Maps”. Fourth International Conference on Microelectronics for Neural Networks and Fuzzy Systems.

Ruping, S. and Ruckert U. (1996). “A Scalable Processor Array for SOFM”. Fifth International Conference on Microelectronics for Neural Networks and Fuzzy Systems.

Ruping, S., Porrmann, M. and Ruckert, U. (1997). “A High Performance SOFM Hardware System”. International Work-Conference on Artificial and Natural Neural Networks.

Ruping, S., Porrmann, M. and Ruckert, U. (1999) “SOM Hardware with Acceleration Module for Graphical Representation of the Learning Process”. Seventh International Conference on Microelectronics for Neural, Fuzzy and Bio-Inspired Systems.

Salapura, V. (1994). “Neural Networks using bit stream arithmetic: A space efficient implementation”. IEEE International Symposium on Circuits and Systems, London.

Schoenauer, T. et al. (1998). “Digital Neurohardware: Principles and perspectives”. Neuronal Networks in Applications.

Schoenauer, T., Jahnke, A., Roth, U. and Klar, H. (1998). “Digital Neurohardware: Principles and Perspectives”. Neural Networks in Applications.

Skapura, D.M. (1996). “Building Neural Networks”. ACM Press.

Speckmann, H., Thole, P. and Rosenstiel, W. (1992). “Hardware Implementation of Kohonen's Self Organising Feature Map”. International Conference of Neural Networks.

Speckmann, H., Thole, P. and Rosenstiel, W. (1993). “Hardware Synthesis for Neural Networks from a Behavioral Description With VHDL”. International Joint Conference on Neural Networks.

Silvaco International (1998). “Cell Characterization with .Modif Statement in SmartSpice”. Simulation Standard. Volume 9. 12-13.

Synopsys Online Documentation. (2000a). “Design Compiler User Guide”. Synopsys Inc., USA.

Synopsys Online Documentation. (2000b). “Design Compiler Tutorial”. Synopsys Inc., USA.

Synopsys Online Documentation. (2000c). “Library Compiler User Guide”. Synopsys Inc., USA.

Tanner Research, Inc. (2001a). “L-Edit User Guide”. Tanner Research Inc., USA.

Tanner Research, Inc. (2001b). “T-Spice Pro User Guide”. Tanner Research Inc., USA.

Vesanto, J. (2000). “Neural Network Tool for Data Mining: SOM Toolbox”. Symposium for Tool Environments and Development Methods for Intelligent Systems.

Visa, A., Iivarinen, J., Valkealahti, K. and Simula, O. (1995). “Neural Network Based Classifier”. International Conference on Artificial Neural Networks.

Weste, N.H.E. and Eshraghian, K. (1992). “Principles of CMOS VLSI Design – A Systems Perspective (Second Edition)”. Addison-Wesley Publishing Company.

Wolberg, W. H. and Mangasarian, O. L. (1990) "Cancer diagnosis via linear programming". SIAM News, Volume 23, Number 5.

Xiang Fang, Thole, P., Goppert, J. and Rosenstiel, W. (1996). “A Hardware Supported System for A Special Online Application of Self-Organising Maps”. International Conference on Neural Networks.

Yihua Liao. (2001). “Neural Networks in Hardware: A Survey”. Department of Computer Science, University of California: ECS250A Project.