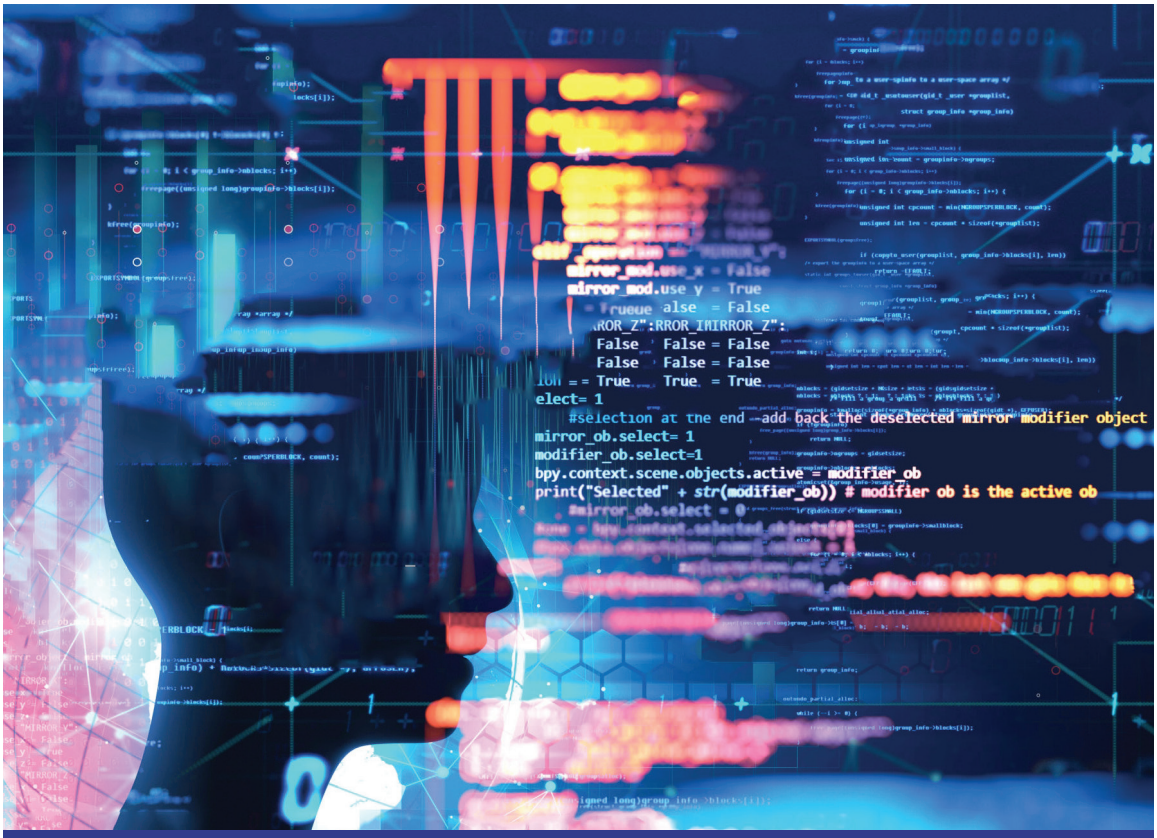


LEARNING IN ENERGY-EFFICIENT NEUROMORPHIC COMPUTING

ALGORITHM AND ARCHITECTURE CO-DESIGN

NAN ZHENG
PINAKI MAZUMDER

WILEY

Learning in Energy-Efficient Neuromorphic Computing
Algorithm and Architecture Co-Design

This edition first published

Registered Office(s)
John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, USA
John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, UK

Editorial Office
The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, UK

For details of our global editorial offices, customer services, and more information about Wiley products visit us at www.wiley.com.

Wiley also publishes its books in a variety of electronic formats and by print-on-demand. Some content that appears in standard print versions of this book may not be available in other formats.

Library of Congress Cataloging-in-Publication Data
Hardback ISBN: 9781119507383
[Typesetter: CiP data includes print ISBNs so when you add overwrite existing print ISBNs found on TV page]

Cover image: [Production Editor to insert]
Cover design by [Production Editor to insert]

Set in size of font and font name by Typesetter
[Typesetter: please leave line space here for printer to insert “Printed in [Country only]” – do not include in BPA files]

10 9 8 7 6 5 4 3 2 1

Table of Contents

Chapter 4 Operational Principles and Learning in SNNs

Chapter 5 Hardware Implementations of Spiking Neural Networks

Chapter 6 Conclusions

Appendix

PREFACE

Learning in Energy-Efficient Neuromorphic Computing: Algorithm and Architecture Co-Design

In 1987 when I was wrapping up my doctoral thesis at the University of Illinois, I had a rare opportunity to listen to Prof. John Hopfield of California Institute of Technology describing his groundbreaking research in neural networks to spellbound students in the Loomis Laboratory of Physics at Urbana-Champaign. He didactically described how to design and fabricate a recurrent neural network chip to rapidly solve the benchmark Traveling Salesman Problem (TSP), which is provably NP-complete in the sense that no physical computer could solve the problem in asymptotically bounded polynomial time as the number of cities in the TSP increases to a very large number.

This discovery of algorithmic hardware to solve intractable combinatorics problems was a major milestone in the field of neural networks as the prior art of perceptron-type feed-forward neural networks could merely classify a limited set of simple patterns. Though the founder of neural computing, Prof. Frank Rosenblatt of Cornel University had built Mark 1 Perceptron computer in late 1950's when the first waves of digital computers such as IBM 650 were just commercialized, subsequent advancements in neural hardware designs were stymied mainly because of lack of integration capability of large synaptic networks by using the then technology, comprising vacuum tubes, relays and passive components such as resistors, capacitors and inductors. Therefore, in 1985 when AT&T Bell Labs fabricated the first solid-state the proof-of-concept TSP chip by using MOS technology to verify Prof. John Hopfield's neural net architecture, it opened the vista for solving non-Boolean and brain-like computing on silicon.

Prof. John Hopfield's seminal work established that if the "objective function" of a combinatorial algorithm can be expressed in quadratic form, the synaptic links in a recurrent artificial neural network could be accordingly programmed to reduce (i.e., locally minimize) the value of the objective function through massive interactions between the constituent neurons. Hopfield's neural network consists of laterally connected neurons that can be randomly initialized and then the network can iteratively reduce the intrinsic Lyapunov energy function of the network to reach a local minima state. Notably, the Lyapunov function decreases in a monotone fashion under the dynamics of the recurrent neural networks where neurons are not provided with self-feedback[1].

---

[1] In the paper, titled: P. Mazumder and J. Yih, "A New Built-in Self-Repair Approach to VLSI Memory Yield Enhancement by Using Neural-Type Circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* Vol. 12, No. 1, Jan. 1993, pp. 124-136, we demonstrated that the quality of solutions achieved by the Hopfield network could be improved considerably by selectively providing self-feedback to allow neurons to escape the local minima. Such method is akin to hill climbing in a gradient descent search that often is trapped in a local minimum point. As self-feedback to neurons impair the stability of a Hopfield neural network, we did not apply any self-feedback to neurons until the network converged to a local minimum state. And, then we pulled out the network out of the local minimum by elevating the energy of the network through the hill climbing mechanism. By using this innovative technique, we showed in the above paper that yield of VLSI memories could be ameliorated by increasing the repair by about 25%.

Prof. Hopfield used a combination of four separate quadratic functions to represent the objective function of the TSP. The first part of the objective function ensures that the energy function minimizes if the traveling salesman traverses cities exactly once, the second part ensures that the traveling salesman visits all cities in the itinerary, the third part ensures that no two cities are visited simultaneously, and the fourth part of the quadratic function is designed to determine the shortest route connecting all cities in the TSP. Because of massive simultaneous interactions between neurons through the connecting synapses that are precisely adjusted to meet the constraints in the above quadratic functions, a simple recurrent neural network could rapidly generate a very good quality solution. However, unlike well-tested software procedures such as simulated annealing, dynamic programming, and the branch-and-bound algorithm, neural networks generally fail to find the best solution because of its simplistic connectionist structures.

Therefore, after listening to Prof. Hopfield's fascinating talk I harbored a mixed feeling about the potential benefit of his innovation. On one hand, I was thrilled to learn from his lecture how computationally hard algorithmic problems could be solved very fast by using simple neuromorphic CMOS circuits having very small hardware overhead. On the other hand, I thought that the TSP application that Prof. Hopfield selected to demonstrate the ability of neural networks to solve combinatorial optimization problems was not the right candidate, as software algorithms are well crafted to obtain nearly the best solution that the neural networks can hardly match. I started contemplating on developing self-healing VLSI chips where the power of neural-inspired self-repair algorithms could be used to automatically restructure faulty VLSI chips. Low overhead and ability to solve a problem concurrently through parallel interactions between neurons are two salient features that I thought could be elegantly deployed for automatically repairing VLSI chips by built-in neural net circuitry.

Soon after I joined the University of Michigan as an assistant professor, working with one of my doctoral students[2], at first I developed a CMOS analog neural net circuitry with asynchronous state updates, which lacked robustness due to process variation within a die. In order to improve the reliability of the self-repair circuitry, an MS student[3] and I designed a digital neural net circuitry with synchronous state updates. These neural circuits were designed to repair VLSI chips by formulating the repair problem in terms of finding the node cover, edge cover, or node pair matching in a bipartite graph. In our graph formalism, one set of vertices in the bipartite graph represented the faulty circuit elements, and the other set of vertices represented the spare circuit elements. In order to restructure a faulty VLSI chip into a fault-free operational chip, the spare circuit elements were automatically invoked through programmable switching elements after identifying the faulty elements through embedded built-in self-testing circuitry.

Most importantly, like the TSP problem, the 2-D array repair can be shown to be an NP-complete problem because the repair algorithm seeks the optimal number of spare rows

[2] P. Mazumder and J. Yih, "Fault-Diagnosis and Self-Repairing of Embedded Memories by Using Electronic Neural Network," *Proc. of IEEE 19th Fault-Tolerant Computing Symposium,* Chicago, Jun. 1989, pp. 270-277.
[3] M.D. Smith and P. Mazumder, "Analysis and Design of Hopfield-type Network for Built-in Self-Repair of Memories," *IEEE Transactions on Computers,* Vol. 45, No. 1, Jan. 1996, pp. 109-115.

and spare columns that can be assigned to bypass faulty components such as memory cells, word-line and bit-line drivers, and sense amplifier bands located inside the memory array. Therefore, simple digital circuits comprising counters and other blocks woefully fail to solve such intractable self-repair problems. Notably, one cannot use external digital computers to determine how to repair embedded arrays, as input and output pins of the VLSI chip cannot be deployed to access the fault patterns in the deeply embedded arrays.

In 1989 and 1992, I received two NSF grants to expand the neuromorphic self-healing design styles to a wider class of embedded VLSI modules such as memory array[4], processors array[5], programmable logic array, and so on[6]. However, this approach to improving VLSI chip yield by built-in self-testing and self-repair was a bit ahead of its time as the state-of-the-art microprocessors in early 1990's contained only a few hundred thousands of transistors as well as the sub-micron CMOS technology was relatively robust. Therefore, after developing the neural-net based self-healing VLSI chip design methodology for various types of embedded circuit blocks, I stopped working on CMOS neural networks. I was not particularly interested in pursing applications of neural networks for other types of engineering problems, as I wanted to remain focused in solving emerging problems in VLSI research.

On the other hand, in late 1980's there were mounting concerns among CMOS technology prognosticators about the impending red brick wall heralding the end of the shrinking era in CMOS. Therefore, to promote several types of emerging technologies that might push the frontier of VLSI technology, Defense Advanced Research Projects Agency (DARPA) in USA had initiated (around 1990) the Ultra Electronics: Ultra Dense, Ultra Fast Computing Components Research Program. Concurrently, the Ministry of International Trade & Industry (MITI) in Japan had launched the Quantum Functional Devices (QFD) Project. Early successes with a plethora of innovative non-CMOS technologies in both research programs led to the launching of National Nanotechnology Initiative (NNI), which is a U.S. Government research and development (R&D) initiative, involving 20 departments and independent agencies to bring about revolution in nanotechnology to impact the industry and society at large.

During the period of 1995 and 2010, my research group had at first focused on quantum physics based device and circuit modeling for quantum tunneling devices, and then we extensively worked on cellular neural network (CNN) circuits for image and video processing by using 1-D (double barrier resonant tunneling device), 2-D (self-assembled nanowire) and 3-D (quantum dot array) constrained quantum devices. Subsequently, we developed learning based neural network circuits by using resistive synaptic devices (commonly known as memristors) and CMOS neurons. We also developed analog voltage programmable nanocomputing architectures by hybridizing quantum tunneling and memristive devices in computing nodes of two-dimensional processing element (PE) ensemble. Our research on

[4] P. Mazumder and J. Yih, "Built-In Self-Repair Techniques for Yield Enhancement of Embedded Memories," *Proceedings of IEEE International Test Conference,* Sep. 1990, pp. 833-841.
[5] P. Mazumder and J. Yih, "Restructuring of Square Processor Arrays by Built-in Self-Repair Circuit," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 12, No. 9, Sept. 1993, pp. 1255-1265.
[6] P. Mazumder, "An integrated built-in self-testing and self-repair of VLSI/WSI hexagonal arrays," *Proceedings of IEEE International Test Conference,* Sep. 1992, pp. 968-977

nanoscale neuromorphic circuits will soon be published in our new book, titled: "Neuromorphic Circuits for Nanoscale Devices", *River Publishing*, UK, 2019.

After spending a little over a decade developing neuromorphic circuits with various types of emerging nanoscale electronic and spintronic devices, I decided to embark research on learning-based digital VLSI neuromporphic chips using nanaoscale CMOS technology in both sub-threshold and super-threshold modes of operation. My student and coauthor of this book, Dr. Nan Zheng conducted his doctoral dissertation work on architectures and algorithms for digital neural networks. We started our design from both machine learning and biological learning perspectives to design and fabricate energy-efficient VLSI chips using TSMC 65 nm CMOS technology.

We captured the actor-critic type reinforcement learning[7] and an example of temporal difference (TD) learning with off-line policy updating, called Q-learning[8] on VLSI chips from the machine learning perspectives. Further, we captured spiking correlation based synaptic plasticity commonly used in biological unsupervised learning applications. We also formulated hardware-friendly spike-timing-dependent plasticity (STDP) learning rules[9] which achieved a classification rates of 97.2% and 97.8% for the one-hidden-layer and two-hidden-layer neural networks, respectively on the Modified National Institute of Standards and Technology (MNIST) database benchmark. The hardware-friendly learning rule enabled both energy-efficient hardware design[10] as well as implementations that were robust to the process-voltage-temperature (PVT) variations associated with chip manufacturing[11]. We demonstrated that the hardware accelerator VLSI chip for actor-critic network solved some control-theoretic benchmark problems by emulating the adaptive dynamic programming (ADP), which is at the heart of reinforcement learning (RL) software program. However, compared with traditional software RL running on a general-purpose processor, the VLSI chip accelerator operating at 175 MHz achieves two orders of magnitude improvement in computational time while consuming merely 25 mW[12].

The chip layout diagrams included in the Preface contain sample of numerous digital neural network chips using CMOS technology that my research group has designed over the course of last 35 years. On the top row: Self-healing chip was designed in 1991 to repair faulty VLSI memory arrays automatically by running node-covering algorithms on a bipartite graph

---

[7] N. Zheng and P. Mazumder, "Hardware-Friendly Actor-Critic Reinforcement Learning Through Modulation of Spiking-Timing Dependent Plasticity," *IEEE Transactions on Computers*, Vol. 66, No. 2, Feb. 1 2017.

[8] I. Ebong and P. Mazumder, "Iterative Architecture for Value Iteration using Memristors," *IEEE Conference on Nanotechnology*, Toronto, Canada, 2014, pp. 967-970.

[9] N. Zheng and P. Mazumder, "Online Supervised Learning for Hardware-Based Multilayer Spiking Neural Networks Through the Modulation of Weight-Dependent Spike-Timing-Dependent Plasticity," in IEEE Transactions on Neural Networks and Learning Systems, vol. 29, no. 9, pp. 4287-4302, Sept. 2018.

[10] N. Zheng and P. Mazumder, "A Low-Power Hardware Architecture for On-Line Supervised Learning in Multi-Layer Spiking Neural Networks," 2018 IEEE International Symposium on Circuits and Systems (ISCAS), Florence, 2018, pp. 1-5.

[11] N. Zheng and P. Mazumder, "Learning in Memristor Crossbar-Based Spiking Neural Networks through Modulation of Weight Dependent Spike-Timing-Dependent Plasticity," in *IEEE Trans. on Nanotechnology,* 2018.

[12] N. Zheng and P. Mazumder, "A Scalable Low-Power Reconfigurable Accelerator for Action-Dependent Heuristic Dynamic Programming," in *IEEE Trans. on Circuits and Systems I: Regular Papers*, vol. PP, no. 99, pp. 1-12.

representing the set of faulty components and the available spare circuit elements. The spike-timing dependent plasticity (STDP) chip was designed in 2013 for controlling the motion of a virtual insect from an initial source to the selected destination by avoiding collisions while navigating through a set of arbitrary shaped blocked spaces. Deep learning chip described in the previous paragraph was designed in 2016.

On the bottom row is shown the reinforcement learning (RL) chip described in above paragraph and designed in 2016. Also, included on the bottom row are two ultra-low-power (ULP) CMOS chips, which were biased in subthreshold mode for wearable applications in health care. In one application, Kohonen's self-organizing map (SOM) neural networks was implemented to classify electro-cardiogram (ECG) waveforms, while the body-sensing network with wireless transceiver was designed to sense analog neuronal signals by using implantable multi-electrode sensor and provide the digitized data through built-in wake-up transceiver to doctors who could monitor the efficacy of drugs at the neuronal and synaptic levels in brain-related diseases such as schizophrenia, chronic depression, Alzheimer disease and so on.

Initially, when we decided to publish a monograph highlighting our work in the form of CMOS neuromorphic chips for brain-like computing, we wanted to aggregate various results of the cited papers in the Preface to compose the contents of the book. However, in the course of preparation of the manuscript, we modified our initial narrow goal, as it would be rather limiting to adopt the book in a regular course for teaching undergraduate and graduate students about the latest generation neural networks with learning capabilities.

Instead, we decided to write a comprehensive book on energy-efficient hardware design for neural networks with various types of learning capability by discussing expansive ongoing research in neural hardware. This is evidently a Herculean task warranting mulling through hundreds of archival sources of references and describing co-design and co-optimization methodologies for building hardware neural networks that can learn to perform various tasks. We attempted to provide a comprehensive perspective, from high-level algorithms to low-level implementation details by covering many fundamentals and essentials in neural networks (e.g., deep learning), as well as hardware implementation of neural networks. In a nutshell, the present version of the book has the following salient features:

- It includes cross-layer survey of hardware accelerators for neuromorphic algorithms,
- It covers the co-design of architecture and algorithms with emerging devices for much-improved computing efficiency, as well as
- It focuses on the co-design of algorithms and hardware, which is paramount for deploying emerging devices such as traditional memristors or diffusive memristors for neuromorphic computing.

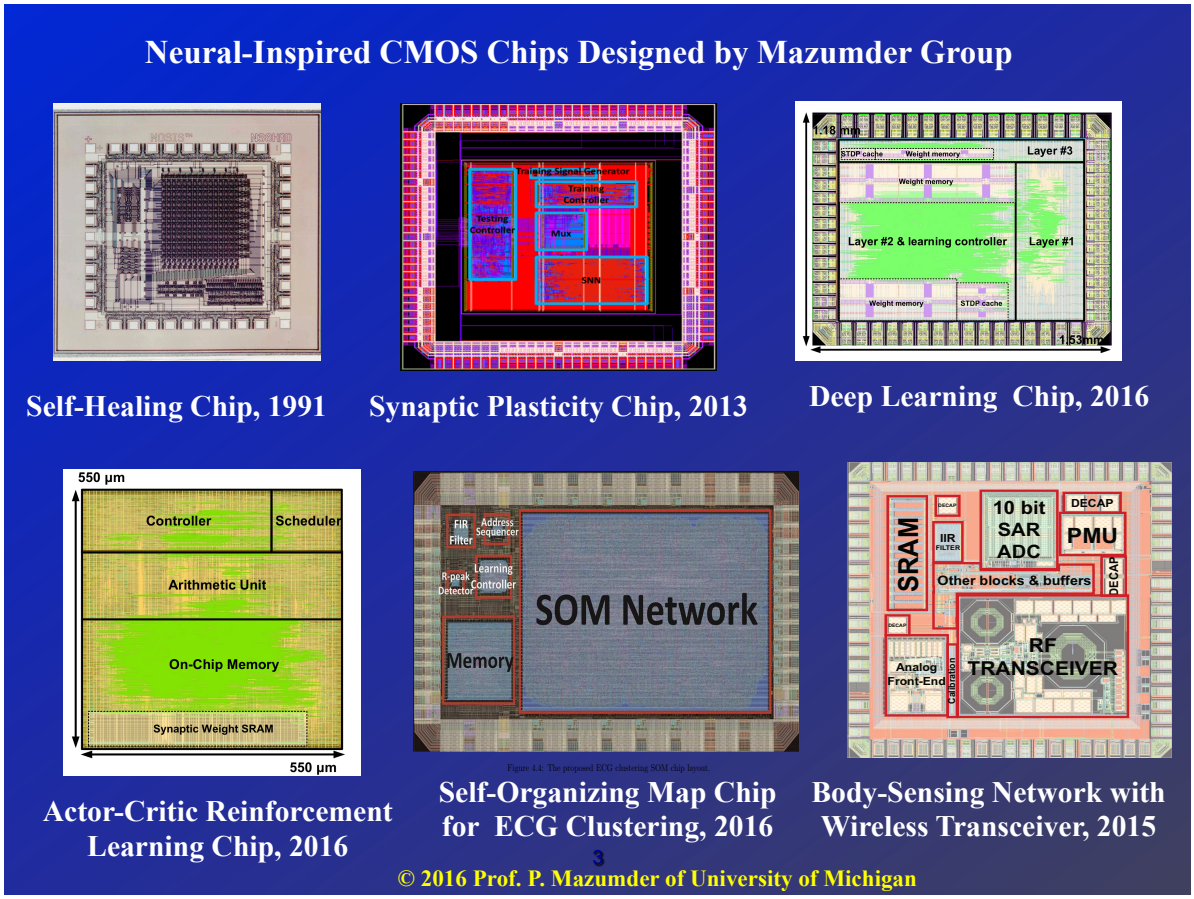
Finally, due to stringent time constraint to complete this book in conjunction with concomitant commitment to concurrently finish the complementary book ("Neuromorphic Circuits for Nanoscale Devices", *River Publishing*, UK, 2019), the present version of the book has been completed without describing the didactic materials pedagogically as expected in a textbook along with exercise problems at the end of each chapter. Hopefully, those goals will

be achieved in the next edition of the book after gathering valuable feedback from students, instructors, practicing engineers and other readers. I shall truly appreciate if you give me such guiding feedback, both positive and negative, which will enable me to prepare the Second Edition of the book. My contact information is included below for your convenience.

Pinaki Mazumder February 14th, 2019

Address:

4765 BBB Building
Division of Computer Science and Engineering
Department of Electrical Engineering and Computer Science
University of Michigan, Ann Arbor, MI 48109-2122
Ph: 734-763-2107
E-mail: mazum@eecs.umich.edu, pinakimazum@gmail.com
Website: http://www.eecs.umich.edu/~mazum



Acknowledgement

First, I would like to thank several of my senior colleagues who had encouraged me to carry on my research in neural computing during the past three decades after I published my first paper on self-healing of VLSI memories in 1989 by adopting the concept of the Hopfield network. Specifically, I would like to thank Prof. Leon O. Chua and Prof. Ernest S. Kuh of the University of California at Berkeley, Prof. Steve M. Kang, Prof. Kent W. Fuchs, and Prof. Janak H. Patel of the University of Illinois at Urbana-Champaign, Jacob A. Abraham of University of Texas at Austin, Prof. Supriyo Bandyopadhyay of Virginia Commonwealth University, Prof. Sudhakar M. Reddy of University of Iowa, and Prof. Tamas Roska and Prof. Csurgay Arpad of Technical University of Budapest, Hungary.

Second, I would like to thank several of my colleagues at the National Science Foundation where I served in the Directorate of Computer and Information Science and Engineering (CISE) as a Program Director of Emerging Models and Technologies program from January 2007 to December 2008, and then I served in the Engineering Directorate (ED) as a Program Director of Adaptive Intelligent Systems program from January 2008 to December 2009. Specifically, I would like to thank Dr. Robert Grafton and Dr. Sankar Basu of Computing and Communications Foundation Division of CISE, and Dr. Radhakrisnan Baheti, Dr. Paul Werbos and Dr. Jenshen Lin of Electrical Communications and Cyber Systems (ECCS) Division of ED for providing me with research funds over the past so many years to conduct research on learning based systems that enabled me to delve deep into CMOS chip design for brain-like computing. I had distinct pleasure interacting with Dr. Michael Roco during my stint at NSF and subsequently when I was invited to present our group's brain-like computing research in US-Korea Forums in Nanotechnology in 2016 at Seoul, Korea and in 2017 at Falls Church, Virginia, USA.

Third, I would like to thank Dr. Jih Shyr Yih, who was my first doctoral student and started working with me soon after I joined the University of Michigan. After taking a course with me, where I taught the memory repair algorithms, he had enthusiastically implemented the first self-healing VLSI chip using the analog Hopfield network. Next, I would like to acknowledge the contributions of Mr. Michael Smith, who implemented the digital self-healing chip as indicated above. My other doctoral students, Dr. W.H. Lee, Dr. I. Ebong, Dr. J. Kim, Dr. Y. Yalcin, and Ms. S. R. Li, had worked on RTD, quantum dots and memristors to build neuromorphic circuits. Their research works were included in a separate book. The bulk of this book is drawn from the doctoral dissertation manuscript of my student and coauthor of the book, Dr. N. Zheng. It was a joy to work with an assiduous student like him. I would like to also thank Dr. M. Erementchouk for scrutinizing the manuscript and providing some suggestions.

Finally, I would like to thank my wife, Sadhana, my son Bhaskar, my daughter Monika and their spouses Pankti Pathak and Thomas Parker, respectively for their understanding and support despite the fact that I spend most of my time with my research group.