Neuromorphic Computing Principles and Organization

Abderazek Ben Abdallah • Khanh N. Dang

Neuromorphic Computing Principles and Organization

Springer

Abderazek Ben Abdallah
School of Computer Science
University of Aizu
Aizu-Wakamatsu, Fukushima, Japan

Khanh N. Dang
School of Computer Science
University of Aizu
Aizu-Wakamatsu, Fukushima, Japan

ISBN 978-3-030-92524-6 ISBN 978-3-030-92525-3 (eBook)
https://doi.org/10.1007/978-3-030-92525-3

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

To my parents, my wife, Sonia, and my children, Tesnim and Beyram

ABA

Dedicated to my parents

KND

Preface

As the end of Moore's law seems closer than ever, computer scientists have been exploring to build machines as complex and efficient as our brain, dealing with power density and clock frequency challenges of the conventional architecture. Our brain works entirely differently compared to traditional von Neumann architecture. There are many secrets behind how the human brain works. We know that it distributes computation and memory among more than 100 billion biological neurons, and each of them is connected with thousands of others via synapses. Neurons communicate with each other through spikes (i.e., short electrical pulses). The brain is a powerful computation system that helps us survive, adapt, and predict, while consuming tens of watts.

Brain-inspired or neuromorphic computing is a biologically inspired approach created from highly connected neurons to model neuroscience theories and solve machine learning problems. The term neuromorphic was first introduced by Carver Mead in 1990, where it referred to very-large-scale integration (VLSI) with analog components to mimic biological neural systems. Such systems can be categorized into non-spiking and spiking approaches. First, the non-spiking approach is referred to as the implementation of traditional artificial neural networks (ANNs) which aims to improve the throughput over the power consumption (or acceleration purpose). In recent years, ANNs have shown a remarkable improvement in terms of accuracy for large-scale visual/auditory recognition and classification tasks. Notably, the convolution neural network (CNN) and recurrent neural network (RNN) have shown to be promising tools for a wide range of applications such as image, video, and speech. They are typically trained by using graphic processing units (GPUs) or on the cloud side. The state-of-the-art neural networks tend to increase their number of layers and size (i.e., deep learning). However, this leads to challenges for hardware systems in terms of computation, memory, and communication resources.

The neuromorphic computing systems promises to drastically improve the efficiency of critical computational tasks such as decision making and perception. Unlike the typical artificial neural networks (ANNs), where neurons fire at each propagation cycle, the neurons in a brain-inspired neural networks model, named spiking neural networks (SNNs), fire only when a membrane potential reaches

a specific value. Spiking neurons are only activated when sufficient signals are integrated from other neurons, which leads to sparse neural activities at the network level. Hence, the large spike sparsity and simple synaptic operations in the network enable SNNs to outperform ANNs in terms of energy efficiency.

This book stands independent and is organized into nine chapters. We have made every attempt to make each chapter self-contained. Chapter 1 introduces the neuromorphic computing system and explores the fundamental concepts of artificial neural networks. We first discuss biological neurons and the dynamics that are abstracted from them to model artificial neurons. Next, we discuss artificial neurons and how they have evolved in their representation of biological neuronal dynamics. Afterward, we discuss implementing these neural networks in terms of neuron models, storage technologies, inter-neuron communication networks, learning, and various design approaches.

Chapter 2 presents the fundamental design principle to build an efficient neuromorphic system in hardware. The challenges that need to be solved toward building in hardware a spiking neural network architecture (neuromorphic) with many synapses include building a small-sized massively parallel architecture with low-power consumption, efficient neuron coding scheme, and lightweight on-chip learning algorithm. The other major challenge is the on-chip communication and routing network, which allows data to be communicated between neurocores and off-chip data to be transferred to the cores. The constraints mentioned above make the deployment of such a brain-like IC a challenging on-chip interconnect problem.

Chapter 3 presents how learning in neuromorphic computing systems is conducted. Neuromorphic hardware's primary goal is to emulate brain-like neural networks to solve real-world problems. However, training on neuromorphic systems is challenging to the required non-local computations of gradient-based learning algorithms. Spiking neural networks gained popularity by incorporating learning. In these neural networks, there are two fundamental modes: Inference and learning. The learning phase, which minimizes a particular cost (loss) function, is a complex process of acquiring the parameters to output the correct inference results. In contrast, inference is computing the output values based on the given input and the network parameters.

To design a neuromorphic system on hardware, it is imperative to develop artificial neurons that mimic biological neurons and artificial synapses that emulate biological synapses. Recently, numerous efforts have been made to realize artificial synapses using post-CMOS devices, including resistive random access memory (ReRAM), ferroelectric field-effect transistor (FeFET), phase change memory devices, magnetoresistive random access memory (MRAM). A non-CMOS neuron based on emerging devices has also been investigated. Chapter 4 discusses the major emerging memory technologies that promise neuromorphic computing and highlight some recent significant progress on device studies. The advantages and challenges for each device technology are also discussed.

The brain connectivity is generally described at several levels of scale, including individual synaptic connections that link individual neurons at the microscale, networks connecting neuronal populations at the mesoscale, and brain regions linked

by fiber pathways at the macroscale. Since each neuron is connected to many others, high bandwidth is required. Moreover, since the spike times are used to encode information, very low communication latency is also needed. Chapter 5 presents the circuits and architectures used for communication in neuromorphic systems. In particular, the Network-on-Chip fabric is introduced for receiving and transmitting spikes following the Address Event Representation (AER) protocol and the memory accessing method. First, the chapter describes the interconnect method for inter-neurons communication. Second, the interconnect design principle is covered to help understand the overall concept of on-chip and off-chip communication. The remaining parts cover advanced on-chip interconnect technologies, including si-photonic three-dimensional interconnects and fault-tolerant routing algorithms.

To develop such emerging systems, designers use large-scale models on dedicated hardware platforms, such as FPGAs, GPUs, or ASICs. The designers need a long time to collect datasets, train, and design accelerators to keep the trained models private and reliable. However, with the growing complexity of neuromorphic systems, there are severe vulnerabilities in the hardware implementations. An attacker who does not know the details of structures and designs inside these accelerators can effectively reverse engineer the neural networks by leveraging various side-channel information. Moreover, as neuromorphic systems are complex and integrate large number of neurons and synapses, the fault probability is accumulated and can threaten system reliability. Chapter 6 covers the main threats of reliability, and discusses several recovery methods.

Chapter 7 presents the architecture and hardware design of a reconfigurable spiking neuromorphic system. The architecture implements a Multi-Layer Perceptron (MLP) that can be reconfigured to recover from faults with suitable methods that use an FPGA without being dependent on FPGA intellectual property (IP). This approach makes possible its implementation in application-specific integrated circuits (ASICs). Most spiking neuromorphic designs mainly focused on fixed functionality using available off-the-shelf components. Such an approach is lacking the flexibility to adapt to various computing environments. A reconfigurable design approach supports multiple target applications via dynamic reconfigurability, network topology independence, and network expandability.

Chapter 8 presents a real hardware-software design of a reliable three-dimensional digital neuromorphic processor geared explicitly toward the 3D-ICs biological brain's three-dimensional structure. The platform enables high integration density and slight spike delay of spiking networks and features a scalable design. R-NASH is a design based on the through-silicon-via (TSV) technology, facilitating spiking neural network implementation on clustered neurons based on network-on-chip (NoC). The system provides a memory interface with the host CPU, allowing for online training and inference of spiking neural networks. Moreover, R-NASH supports fault detection and recovery with graceful performance degradation.

Chapter 9 presents a comprehensive survey of the research of neuromorphic computing systems. First, the chapter gives the motivations of neuromorphic computing. Then, it describes significant research works in the field, which we

categorize as software emulation approach, digital hardware approach, and analog and mixed-signal hardware approaches. This chapter aims to provide an exhaustive review of the research conducted in neuromorphic computing and illuminates the gaps in the field where new research is needed.

The neuromorphic computing principles and organization book is an excellent resource for researchers, scientists, graduate students, and hardware-software engineers dealing with the ever-increasing demands on fault-tolerance, scalability, and low power consumption. It is also an excellent resource for teaching advanced undergraduate and graduate students about the fundamentals concepts, organization, and actual hardware-software design of reliable neuromorphic systems with learning and fault-tolerance capabilities.

Acknowledgments

This book took nearly 3 years to complete. It evolved from our research and education experiences in adaptive computing systems and neuromorphic computing architectures designs. The Neuromorphic computing paradigm created excellent opportunities to explore cognitive AI system performance and created many design challenges that designers must overcome. To advance the field of neuromorphic computing, the exploration of novel materials and devices will be the key to improve the power efficiency and scalability of state-of-the-art CMOS solutions. Thus, we must continue innovating new algorithms and techniques to solve these challenges. We must also educate computer science and computer engineering students in both neuromorphic computing and engineering. The authors wish to thank Mark Ogbodo, Zhishang Wang, and Wang Jiangkun from the Adaptive Systems Laboratory at the University of Aizu, and Vu Huy The for their valuable comments, help, and discussion.

Finally, this first version of this book was completed without describing the didactic materials pedagogically as expected in a textbook with exercises and their solutions at the end of each chapter. Hopefully, those goals will be completed in the second edition of this book after receiving insightful feedback from students, instructors, researchers, and practicing engineers. We truly appreciate it if you give us such feedback, allowing us to prepare a second edition for this fast-growing and emerging computing paradigm.

Aizu-Wakamatsu, Japan — Abderazek Ben Abdallah
Aizu-Wakamatsu, Japan — Khanh N. Dang

Contents

1 Introduction to Neuromorphic Computing Systems 1
1.1 Introduction 1
1.2 Design Challenges 5
1.3 Neural Networks 6
1.3.1 Artificial Neural Networks 6
1.3.2 Spiking Neural Networks 7
1.4 Learning in Spiking Neural Networks 8
1.5 Synapse Memory Technologies 9
1.6 Neurons Communication Network 10
1.7 Neuromorphic System Design Domains 10
1.8 Chapter Summary 11
References 11

2 Neuromorphic System Design Fundamentals 15
2.1 Introduction 15
2.1.1 Spiking Neural Networks 16
2.1.2 Neural Coding Schemes 18
2.2 Spiking Neuron Models 21
2.2.1 Hodgkin-Huxley Model 22
2.2.2 Izhikevich Model 23
2.2.3 Leaky Integrate and Fire Model 24
2.3 Learning Algorithms 25
2.3.1 Supervised Learning 25
2.3.2 Unsupervised Learning 26
2.4 Synapse Memory 27
2.4.1 SRAM 28
2.4.2 eDRAM 28
2.4.3 Memristor 30
2.5 Inter-Neuron Communication Schemes 30
2.5.1 AER—Address Event Representation 31
2.6 Neuromorphic Spike Routing 32

2.7 Chapter Summary 32
References 32

3 Learning in Neuromorphic Systems 37
3.1 Learning Methods 37
3.2 Conversion from ANN to SNN 38
3.2.1 Converted SNNs 38
3.2.2 Challenges of ANN Conversion 39
3.3 Supervised Learning 43
3.3.1 *Tempotron* 43
3.3.2 *ReSuMe* 44
3.3.3 *SpikeProp* Algorithm 45
3.3.4 Approximate Derivative Method (*ADM*) 48
3.4 Unsupervised Learning 48
3.4.1 Pair-Based STDP Learning Rule 49
3.4.2 Triplet STDP Learning Rule 50
3.4.3 Reward-Modulated STDP Learning 51
3.4.4 Other Variants of STDP Learning Rule 51
3.5 Chapter Summary 52
References 52

4 Emerging Memory Devices for Neuromorphic Systems 55
4.1 Introduction 55
4.2 Memory Technology 57
4.2.1 SRAM 58
4.2.2 eDRAM 60
4.2.3 STT-RAM 61
4.2.4 RRAM and Resistive Crossbar 62
4.2.5 Phase Change Memory 63
4.2.6 Other Memory Technologies 64
4.3 Memory Organization 65
4.4 Memory for Neuromorphic Systems 67
4.4.1 Neuron State Memory 67
4.4.2 Synapse Memory 68
4.5 Dynamic NVM Synapse 76
4.5.1 Learning Related NVM 76
4.5.2 Conductance Drift in NVM 76
4.6 Chapter Summary 77
References 77

5 Communication Networks for Neuromorphic Systems 79
5.1 Introduction 79
5.2 Neural Communication 80
5.3 Interconnect for Inter-Neural Communication 82
5.3.1 SpiNNaker 83
5.3.2 TrueNorth 85
5.3.3 Loihi 85

5.4 Interconnect Design Principles 86
5.4.1 OSI Model for Network-on-Chip 87
5.4.2 Network Topologies 88
5.4.3 Application Mapping 90
5.4.4 Communication Architecture 92
5.5 Advanced Interconnects Multicore Neuromorphic Systems 98
5.5.1 Three Dimensional On-chip Interconnect 100
5.5.2 Photonic On-chip Interconnect for High-Bandwidth Multicore SoCs 110
5.5.3 Network Interface 121
5.6 Chapter Summary 122
References 122

6 Fault-Tolerant Neuromorphic System Design 127
6.1 Introduction 127
6.1.1 Measure of Fault Tolerance 128
6.1.2 Type of Faults and Behavior 129
6.1.3 Impact of Faults on Neuromorphic System 131
6.2 Conventional Computing System Fault Tolerance 132
6.2.1 Hardware Approach 132
6.2.2 Information Redundancy 134
6.2.3 Software Approach 137
6.3 Fault-Tolerance for Neuromorphic Computing 137
6.3.1 Memory Protection 138
6.3.2 Communication Protection 138
6.3.3 Computation Protection 138
6.3.4 SNN Mapping for Tolerating Errors 139
6.3.5 Fault-Tolerant Remapping for Neuromorphic Computing 140
6.4 Chapter Summary 150
References 153

7 Reconfigurable Neuromorphic Computing System 155
7.1 Introduction 155
7.2 Fault-Tolerant Neural Networks 158
7.2.1 Learning-Based Approach 158
7.2.2 Architecture-Based Approach 158
7.2.3 Hybrid-Based Approach 160
7.3 Inter-Neuron Communication Network 160
7.4 Reconfigurable Neuromorphic System Building Blocks 163
7.4.1 Spiking Neuron Processing Core 164
7.4.2 Network Interface 167
7.4.3 Fault-Tolerant Multicast 3D Router 167

7.5 Fault-Tolerant Spike Routing Algorithm 169
7.5.1 Shortest Path K-means Multicast Spike Routing Algorithm 169
7.5.2 Fault-Tolerant K-means Multicast Spike Routing Algorithm 170
7.6 Mapping 173
7.7 Complexity Analysis 175
7.8 Chapter Summary 177
References 178

8 Case Study: Real Hardware-Software Design of 3D-NoC-Based Neuromorphic System 183
8.1 Introduction 183
8.2 R-NASH System 185
8.3 R-NASH Hardware 186
8.3.1 R-NASH Hardware Building Blocks 186
8.3.2 Spiking Neural Processing Core (SNPC) 187
8.3.3 Network Interface 188
8.3.4 Crossbar 189
8.3.5 Controlling 190
8.3.6 Inter-Neural Interconnect 191
8.4 R-NASH Learning 192
8.4.1 Off-chip Learning 192
8.4.2 Online Learning with STDP 192
8.5 R-NASH Initial Mapping 194
8.5.1 Genetic Algorithm 195
8.5.2 Selection 196
8.5.3 Crossover 196
8.5.4 Mutation 197
8.5.5 Genetic Algorithm for Neurons Mapping on R-NASH Hardware 197
8.6 R-NASH Run-Time Maintenance 199
8.6.1 Data Integrity Protection 199
8.6.2 Communication Protection 200
8.6.3 Fault-Tolerant Neurons Mapping Scheme 200
8.7 R-NASH Evaluation Results 203
8.7.1 Initial Mapping Evaluation 204
8.7.2 Fault-Tolerant Mapping 208
8.7.3 Hardware Complexity 209
8.7.4 System Validation 210
8.7.5 Unsupervised STDP 213
8.8 Chapter Summary 214
References 215

9 Survey of Neuromorphic Systems 217
9.1 Introduction 217
9.2 Software Emulation Approach 219
9.2.1 SpiNNaker 219
9.3 Digital Hardware Design Approach 227
9.3.1 IBM TrueNorth 228
9.3.2 Intel Loihi 233
9.4 Analog and Mixed-Signal Hardware Approach 234
9.4.1 NeuroGrid 234
9.5 Chapter Summary 238
References 238

Index 241

Acronyms

AAAI	American Association for Artificial Intelligence
ACK	Acknowledgment
ACM	Association for Computing Machinery
ADC	Analog to Digital Converter
ADM	Approximate Derivative Method
AER	Address Event Representation
AHB	Advanced High-performance Bus
AI	Artificial Intelligence
AMBA	Advanced Microcontroller Bus Architecture
AMD	Advanced Micro Devices
ANN	Artificial Neural Network
API	Application Programming Interface
ARM	Advanced RISC Machines
ASIC	Application-Specific Integrated Circuit
ASID	Anti-Counterfeiting, Security and Identification
BCM	Building Cube Method
BE	Best Effort
BL	Bit Line
BP	Backpropagation
BW	Buffer Writing
BWCCA	Broadband and Wireless Computing, Communication and Applications
CAD	Computer Aided Design
CAM	Content-Access-Memory
CASES	Conference on Compilers, Architecture and Synthesis for Embedded Systems
CD	Compact Disc
CICC	Custom Integrated Circuits Conference
CIFAR	Canadian Institute For Advanced Research
CLEO	Conference on Lasers and Electro-Optics
CMD	Command

CMOS	Complementary Metal-Oxide Semiconductor
CNN	Convolution Neural Network
CP	Configuration Packet
CPU	Central Processing Unit
CRC	Cyclic Redundancy Code
CS	Computer Science
CS	Chip Select
CSUR	Computing Surveys
CT	Crossbar Traversal
DAC	Digital-to-Analog Converter
DBN	Deep Belief Network
DMA	Direct Memory Access
DNN	Deep Neural Network
DOR	Dimension Order Routing
DRAM	Dynamic Random Access Memory
DTCM	Data Tightly Coupled Memory
DVD	Digital Versatile Disk
DWDM	Dense Wavelength Division Multiplexing
DWM	Domain Wall Memory
EC	Electronic Controller
ECC	Error Correction Code
ECN	Electronic Control Network
ECTC	Electronic Components and Technology Conference
EDRAM	Embedded Dynamic Random Access Memory
EMCSI	Electromagnetic Compatibility Signal/Power Integrity
ESD	ElectroStatic Discharge
ETE	End-To-End
FeFET	Ferroelectric Field-Effect Transistor
FIFO	First In, First Out
FPGA	Field Programmable Gate Array
FPNA	Field Programmable Neural Array
FT-PHENIC	Fault-Tolerant Photonic Network-on-Chip
FTMC-3DR	Fault-Tolerant Multicast 3D Routers
FTPP	Fault-Tolerant Photonic Path-configuration algorithm
FTSP-KMCR	Fault-Tolerant Shortest Path K-means-based MultiCast Routing algorithm
FTSPKMCR	Fault-Tolerant MultiCast Routing algorithm
GA	Genetic Algorithm
GPU	Graphic Processing Units
GPGPU	General Purpose Graphic Processing Units
GS	Greedy Search
GUI	Graphical User Interface
HAL	Hardware Abstraction Layer
HDD	Hard Disk Drive
HiPC	High Performance Computing

HPCA	High Performance Computer Architecture
HRS	High Resistive state
IBM	International Business Machines
IC	Integrated Circuit
ID	Identification
IEEE	Institute of Electrical and Electronics Engineers
IF	Integrate and Fire
ILP	Integer Linear Programming
IP	Intellectual Property
i-PACT	Innovations in Power and Advanced Computing Technologies
ISBN	International Standard Book Number
ISCA	International Symposium on Computer Architecture
ISCAS	International Symposium on Circuits and Systems
ISI	Inter-Spike-Interval
ITCM	Instruction Tightly Coupled Memory
JPEG	Joint Photographic Experts Group
KB	Kilobyte
KL	Kernighan-Lin
LA-XYZ	Look-Ahead XYZ
LIF	Leaky Integrate-and-Fire
LRS	Low-Resistive State
LRU	Least Recently Used
LTD	Long-Term Depression
LTP	Long-Term Potentiation
LUT	Look-Up-Table
MB	Mega Bytes
MCSoC	Multicore/Many-Core Systems-on-Chip
MFMC	Max-Flow Min-Cut
MIVs	Monolithic Intertier Vias
MJT	Multi Junction Technology
MLP	Multi-Layer Perceptron
MNIST	Modified National Institute of Standards and Technology database
MPI	Message Passing Interface
MR	Microring Resonator
MRAM	Magnetoresistive Random-Access Memory
MRCT	Micro Ring Configuration Table
MRPR	Microring fault-Resilient Photonic Router
MRST	Micro Ring State Table
MTBF	Mean Time Between Failures
MTJ	Magnetic Tunneling Junction
MTTF	Mean Time to Failures
MTTR	Mean Time to Repair
NACK	Negative Acknowledgment
NAND	Not and
NASH	Neuro-inspired ArchitectureS in Hardware

NEWS	North-East-West-South
NI	Network Interface
NP-hard	Non-deterministic Polynomial-time hard
NSEW	North South East West
NVM	Non-Volatile Memories
OE	Output Enable
OSI	Open Systems Interconnection
OSI	Open Systems Interconnection Model
PB	Path_Blocked
PCB	Process Control Block
PCM	Phase Change Memory
PCN	Photonic Communication Network
PE	Processing Element
PJ	Picojoule
PNoC	Photonic Network-on-Chip
PS	Photonic Switch
PSCP	Path-Setup-Control Packet
PSO	Particle Swarm Optimization
PSP	Post-Synaptic Potential
PV	Process Variation
RAM	Random-Access Memory
RBL	Resistance Between Layers
RC	Routing Calculation
RE	Read Enable
RELU	Rectified Linear Unit
RISC	Reduced Instruction Set Computing
RMP	Residual Membrane Potential
R-NASH	Reconfigurable Neuro-inspired ArchitectureS in Hardware
RNN	Recurrent Neural Network
ROM	Read-Only Memory
RPM	Randomized Partially Minimal
RRAM	Resistive Random-Access Memory
RSP	Rapid System Prototyping
RTL	Register-Transfer Level
SA	Switch Allocation
SAF	Store-And-Forward
SDRAM	Synchronous Dynamic Random Access Memory
SDSP	Spike Driven Synaptic Plasticity
SECDED	Single Error Correction, Double Error Detection
SEQ	Sequencer
SET	Single-Event Transients
SEU	Single-Event Upsets
SMC	Systems, Man and Cybernetics
SNN	Spiking Neural Network
SNPC	Spiking Neuron Processing Core

SoC	System On a Chip
SP	Shortest Path
SRAM	Static Random Access Memory
SRDS	Symposium on Reliable Distributed Systems
STA	Sciences and Techniques of Automatic Control and Computer Engineering
STDP	Spike Timing Synaptic Plasticity
STPD	Spike-Timing-Dependent-Plasticity
STT-RAM	Spin-Transfer Torque RAM
TDM	Time-Division Multiplexing
TDMA	Time-Division-Multiple-Access
TECS	Transactions on Embedded Computing Systems
TMR	Triple Modular Redundancy
TODAES	Transactions on Design Automation of Electronic Systems
TSVs	Through Silicon Vias
TTFS	Time-To-First-Spike
TV	Thermal Variations
UI	User Interface
USA	United States of America
USB	Universal Serial Bus
VCSEL	Vertically Cavity Surface Emitting Laser
VGG	Visual Geometry Group Network
VLSI	Very-Large-Scale Integration
VTS	VLSI Test Symposium
WDM	Wavelength Division Multiplexing
WE	Write Enable
WL	Word Line
WL	Worst Loss
WTA	Winner-Take-All
XOR	Exclusive OR