WOODHEAD PUBLISHING SERIES IN ELECTRONIC AND OPTICAL MATERIALS

MEMRISTIVE DEVICES FOR BRAIN-INSPIRED COMPUTING

FROM MATERIALS, DEVICES, AND CIRCUITS TO APPLICATIONS - COMPUTATIONAL MEMORY, DEEP LEARNING, AND SPIKING NEURAL NETWORKS

Edited by
SABINA SPIGA, ABU SEBASTIAN,
DAMIEN QUERLIOZ, BIPIN RAJENDRAN

ELSEVIER

Memristive Devices for Brain-Inspired Computing

From Materials, Devices, and Circuits to Applications — Computational Memory, Deep Learning, and Spiking Neural Networks

Woodhead Publishing Series in Electronic and Optical Materials

Memristive Devices for Brain-Inspired Computing

From Materials, Devices, and Circuits to Applications — Computational Memory, Deep Learning, and Spiking Neural Networks

Edited by

Sabina Spiga

Abu Sebastian

Damien Querlioz

Bipin Rajendran

ELSEVIER

WP
WOODHEAD
PUBLISHING
An imprint of Elsevier

Woodhead Publishing is an imprint of Elsevier
The Officers' Mess Business Centre, Royston Road, Duxford, CB22 4QH, United Kingdom
50 Hampshire Street, 5th Floor, Cambridge, MA 02139, United States
The Boulevard, Langford Lane, Kidlington, OX5 1GB, United Kingdom

Notices

Knowledge and best practice in this field are constantly changing. As new research and experience broaden our understanding, changes in research methods, professional practices, or medical treatment may become necessary.

Practitioners and researchers must always rely on their own experience and knowledge in evaluating and using any information, methods, compounds, or experiments described herein. In using such information or methods they should be mindful of their own safety and the safety of others, including parties for whom they have a professional responsibility.

To the fullest extent of the law, neither the Publisher nor the authors, contributors, or editors, assume any liability for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions, or ideas contained in the material herein.

British Library Cataloguing-in-Publication Data
A catalogue record for this book is available from the British Library

Library of Congress Cataloging-in-Publication Data
A catalog record for this book is available from the Library of Congress

ISBN: 978-0-08-102782-0 (print)

ISBN: 978-0-08-102787-5 (online)

Publisher: Matthew Deans
Acquisitions Editor: Kayla Dos Santos
Editorial Project Manager: Mariana Henriques
Production Project Manager: Sojan P. Pazhayattil
Cover Designer: Alan Studholme

Typeset by MPS Limited, Chennai, India

Contents

List of contributors

Stefano Ambrogio IBM Research– Almaden, 650 Harry Road, San Jose, CA, United States

Rotem Ben-Hur Technion – Israel Institute of Technology, Haifa, Israel

Stephanie Bohaichuk Stanford University, Stanford, CA, United States

Stefano Brivio CNR—IMM, Unit of Agrate Brianza, Agrate Brianza, Italy

Geoffrey W. Burr IBM Research– Almaden, 650 Harry Road, San Jose, CA, United States

Meng-Fan Chang Electrical Engineering Department, National Tsing Hua University, Hsinchu, Taiwan, ROC

Solomon Amsalu Chekol Center for Single Atom-based Semiconductor Device and Department of Materials Science and Engineering, Pohang University of Science and Technology (POSTECH), Pohang-si, South Korea

Thomas Dalgaty University of Grenoble Alpes, CEA, LETI, Grenoble, France

Chunmeng Dou Electrical Engineering Department, National Tsing Hua University, Hsinchu, Taiwan, ROC

Ahmed Eltawil Department of Electrical Engineering and Computer Science, University of California—Irvine, Irvine, CA, United States

Mohammed E. Fouda Department of Electrical Engineering and Computer Science, University of California—Irvine, Irvine, CA, United States

Bin Gao Institute of Microelectronics, Tsinghua University, Beijing, P.R. China

Julie Grollier Unité Mixte de Physique, CNRS, Thales, Université Paris-Saclay, 91767 Palaiseau, France

Ameer Haj-Ali Technion – Israel Institute of Technology, Haifa, Israel

Liza Herrera Diez Center for Nanosciences and Nanotechnology, CNRS, Université Paris-Saclay, Palaiseau, France

Hyunsang Hwang Center for Single Atom-based Semiconductor Device and Department of Materials Science and Engineering, Pohang University of Science and Technology (POSTECH), Pohang-si, South Korea

Daniele Ielmini Department of Electronics, Information and Bioengineering, Polytechnic University of Milan and IU.NET, Milan, Italy

Giacomo Indiveri Institute of Neuroinformatics, University of Zurich and ETH Zurich, Zürich, Switzerland

Suhas Kumar Hewlett Packard Labs, Palo Alto, CA, United States

Fadi Kurdahi Department of Electrical Engineering and Computer Science, University of California—Irvine, Irvine, CA, United States

Shahar Kvatinsky Technion – Israel Institute of Technology, Haifa, Israel

Raphaël Laurent HawAI.tech S.A.S., Grenoble, France

Manuel Le Gallo IBM Research – Zurich, Rüschlikon, Switzerland

Haitong Li Department of Electrical Engineering, Stanford University, Stanford, CA, United States

Seokjae Lim Center for Single Atom-based Semiconductor Device and Department of Materials Science and Engineering, Pohang University of Science and Technology (POSTECH), Pohang-si, South Korea

Bernabé Linares-Barranco Instituto de Microelectrónica de Sevilla IMSE-CNM, CSIC and Universidad de Sevilla, Sevilla, Spain

Nicolas Locatelli Center for Nanosciences and Nanotechnology, CNRS, Université Paris-Saclay, Palaiseau, France

Wei D. Lu Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI, United States

Charles Mackin IBM Research– Almaden, 650 Harry Road, San Jose, CA, United States

Stephan Menzel Peter-Grünberg-Institut 7, Forschungszentrum Jülich GmbH, Juelich, Germany

Rivu Midya Department of Electrical and Computer Engineering, University of Massachusetts, Amherst, MA, United States

Thomas Mikolajick NaMLab gGmbH, Dresden, Germany; Institute for Semiconductors and Microsystems, TU Dresden, Dresden, Germany

Valerio Milo Department of Electronics, Information and Bioengineering, Polytechnic University of Milan and IU.NET, Milan, Italy

Subhasish Mitra Department of Electrical Engineering, Stanford University, Stanford, CA, United States

Alice Mizrahi Unité Mixte de Physique, CNRS, Thales, Université Paris-Saclay, 91767 Palaiseau, France

Pritish Narayanan IBM Research– Almaden, 650 Harry Road, San Jose, CA, United States

Emre Neftci Department of Cognitive Sciences and Department of Computer Science, University of California—Irvine, Irvine, CA, United States

Jaehyuk Park Center for Single Atom-based Semiconductor Device and Department of Materials Science and Engineering, Pohang University of Science and Technology (POSTECH), Pohang-si, South Korea

Melika Payvand Institute of Neuroinformatics, University of Zurich and ETH Zurich, Zürich, Switzerland

Damien Querlioz Centre for Nanoscience and Nanotechnology, Universite Paris-Saclay, Palaiseau, France

Jan M. Rabaey Department of Electrical Engineering and Computer Sciences, UC Berkeley, Berkeley, CA, United States

Abbas Rahimi Department of Information Technology and Electrical Engineering, ETH Zurich, Zurich, Switzerland; Department of Electrical Engineering and Computer Sciences, UC Berkeley, Berkeley, CA, United States

Bipin Rajendran Department of Engineering, King's College London, London, United Kingdom

Ronny Ronen Technion – Israel Institute of Technology, Haifa, Israel

Abu Sebastian IBM Research – Zurich, Rüschlikon, Switzerland

Robert M. Shelby IBM Research– Almaden, 650 Harry Road, San Jose, CA, United States

Max M. Shulaker Electrical Engineering and Computer Science Department, MIT, Cambridge, MA, United States

Jeonghwan Song Center for Single Atom-based Semiconductor Device and Department of Materials Science and Engineering, Pohang University of Science and Technology (POSTECH), Pohang-si, South Korea

Sabina Spiga CNR–IMM, Agrate Brianza, Italy

Hsinyu Tsai IBM Research– Almaden, 650 Harry Road, San Jose, CA, United States

Elisa Vianello University of Grenoble Alpes, CEA, LETI, Grenoble, France

Nimrod Wald Technion – Israel Institute of Technology, Haifa, Israel

Zhongrui Wang Department of Electrical and Computer Engineering, University of Massachusetts, Amherst, MA, United States

H.-S. Philip Wong Department of Electrical Engineering, Stanford University, Stanford, CA, United States

Huaqiang Wu Institute of Microelectronics, Tsinghua University, Beijing, P.R. China

Tony F. Wu Department of Electrical Engineering, Stanford University, Stanford, CA, United States

J. Joshua Yang Department of Electrical and Computer Engineering, University of Massachusetts, Amherst, MA, United States

Jongmyung Yoo Center for Single Atom-based Semiconductor Device and Department of Materials Science and Engineering, Pohang University of Science and Technology (POSTECH), Pohang-si, South Korea

Ying Zhou Institute of Microelectronics, Tsinghua University, Beijing, P.R. China

Mohammed A. Zidan Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI, United States

Preface

To process the ever-increasing amounts of data, computing technology has relied on the principles of Dennard and Moore's laws to scale up the performance of conventional von Neumann machines. As these strategies break down due to technological limits, a radical departure from the processor-memory dichotomy is needed to circumvent the limitations of today's computers. Brain-inspired computing is a key non-von Neumann approach that is being actively researched.

Resistive memory devices, also known as memristive devices, could play an essential role in brain-inspired computing. The first part of the book (Chapters 1−5) will introduce the current state of the art of memristive devices and their role in brain-inspired computing. The materials systems and device characteristics of the various memory technologies (such as phase-change memory, resistive random access memory, magnetoresistive random access memory, and ferroelectric random access memory) will be presented and discussed in view of applications beyond information storage and toward brain-inspired computing.

Computing systems that are inspired by the way the brain works could have varying levels of similarity to the computational principles of the brain. At a basic level, a key attribute is the collocation of memory and processing. If data are stored in memristive devices, then their physical attributes and array-level organization can be exploited to achieve in-place computation, which we refer to as in-memory computing. Memristive devices, when organized within a computational memory unit, can be used to perform a range of tasks from logical and arithmetic operations to stochastic computing. The second part of the book (Chapters 6−11) will give an overview of these applications.

Recently, deep artificial neural networks (deep learning) have shown remarkable human-like performance in tasks such as image processing and voice recognition. These network architectures are loosely inspired by the brain but are still implemented in conventional von Neumann machines comprising large CPU-GPU clusters. To address the associated computational inefficiency, non-von Neumann coprocessors based on memristive devices are being actively researched. The third part of the book (Chapters 12−14) provides a detailed account of how memristive devices are exploited for deep learning acceleration.

Despite our ability to train deep neural networks with brute-force optimization, the computational principles of neural networks remain poorly understood. Hence significant research is aimed at unraveling the principles of computation or information processing in large biological neuronal networks. It is widely believed that because of the added temporal dimension, spiking neural networks (SNNs) are computationally more powerful. The fourth part of the book (Chapters 15−19) provides an overview of how memristive devices could efficiently implement these novel spike-based algorithms.

This book targets a broad and interdisciplinary audience working in the field of brain-inspired computing from materials scientists, physicists, and electrical engineers to computer scientists. We hope it will be a valuable and timely resource to researchers from both academia and industry and at various levels of expertise, including masters and PhD students.

We would like to express our sincere appreciation to the authors who provided excellent contributions and have been willing to go through multiple iterations in order to improve the quality and consistency within the book. We would also like to thank the colleagues at our respective institutions, CNR-IMM-Agrate Brianza, IBM Research—Zurich, Université Paris-Saclay/CNRS, and King's College London, for all the support during the preparation of this book. In particular we would like to thank Anne-Marie Cromack from IBM Research—Zurich for editorial assistance.

Sabina Spiga[1], Abu Sebastian[2], Damien Querlioz[3] and Bipin Rajendran[4]

[1]CNR−IMM, Agrate Brianza, Italy, [2]IBM Research − Zurich, Rüschlikon, Switzerland, [3]Centre for Nanoscience and Nanotechnology, Universite Paris-Saclay, Palaiseau, France, [4]Department of Engineering, King's College London, London, United Kingdom

May 5, 2020

Part I

Memristive devices for brain–inspired computing

Chapter 1

Role of resistive memory devices in brain-inspired computing

Sabina Spiga[1], Abu Sebastian[2], Damien Querlioz[3] and Bipin Rajendran[4]
[1]CNR–IMM, Agrate Brianza, Italy, [2]IBM Research – Zurich, Rüschlikon, Switzerland,
[3]Centre for Nanoscience and Nanotechnology, Universite Paris-Saclay, Palaiseau, France,
[4]Department of Engineering, King's College London, London, United Kingdom

1.1 Introduction

This chapter introduces various types of resistive memory devices (also named memristive devices) of current interest for brain-inspired computing. These memristive device technologies include a broad class of two- or three-terminal devices whose resistance can be modified upon electrical stimuli. The resistance changes can last for short- or long-time scales, leading to a volatile or nonvolatile memory effect, respectively. Memristive devices are based on a large variety of physical mechanisms, such as redox reactions and ion migration, phase transitions, spin-polarized tunneling, and ferroelectric polarization. The switching geometry can involve a volume, interfacial, or confined 1D filamentary regions [1–8].

Although these technologies have been mainly developed as nonvolatile memory devices for storage applications, recently, they have been receiving increasing interest for brain-inspired computing, and many exciting developments are underway in this direction [1,9–23]. Today we are facing a revolution driven by the increasing amount of data generated each day, which need to be stored, classified, and processed, leading to the paradigm of data-centric-computing. On the other hand current computing systems are inherently limited in energy efficiency and data bandwidth by the physically separated memory and processing units (von Neumann bottleneck), as well as by the latency mismatch between the memory and processing units (memory wall) [9,10,13]. Memristive devices have the potential to meet the considerable demand for new devices that enable energy-efficient and area-efficient

Memristive Devices for Brain-Inspired Computing. DOI: https://doi.org/10.1016/B978-0-08-102782-0.00001-0

information processing that transcends von Neumann computing. In the following sections we describe the leading memristive technologies and their current potential for various applications.

1.2 Type of resistive memory devices

Fig. 1.1 shows a classification of the most representative and mature resistive switching memory technologies (RRAM, PCM, MRAM, and FeRAM) based on their underlying physical mechanism, location of the switching region, and their current–voltage or resistance–voltage behavior.

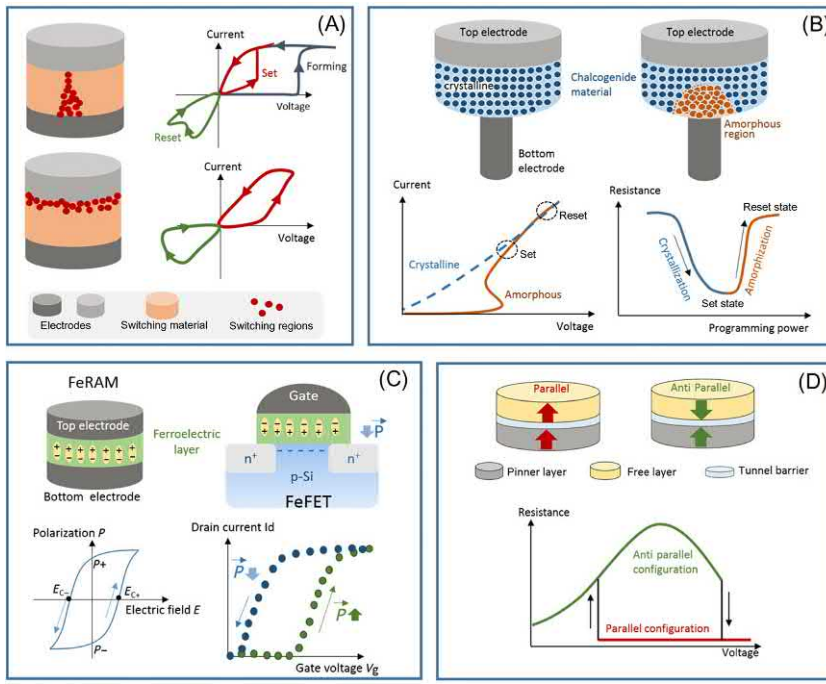


FIGURE 1.1 (A) Sketch of resistive random access memory (RRAM) devices featuring filamentary (top) and interfacial switching (bottom). The corresponding representative current–voltage characteristics are reported on the right, indicating bipolar switching from a high- to low-resistance state (set) and vice versa (reset). For filamentary systems an initial high-voltage forming is required. (B, Top) Schematic drawing of phase-change memory (PCM) cell in the crystalline (LRS) and amorphous state (HRS) of the chalcogenide material. (Bottom) Current–voltage and resistance-programming power characteristics. (C) Ferroelectric random access memory (FeRAM) and its polarization–electric field hysteresis (left), and ferroelectric field effect transistor (FeFET), with corresponding drain current versus gate voltage characteristics (right). (D) Sketch of a magnetic tunnel junction (MTJ) (top) and evolution of the resistance versus applied voltage (bottom) for the parallel and antiparallel configurations of the magnetization orientation of the pinned and free layers. The MTJ is the key element for a magnetic random access memory (MRAM).

Resistive random access memory (RRAM) shown in Fig. 1.1A is based on a two-terminal structure where a switching medium is sandwiched among two electrodes, whose resistance can be switched reversibly among two or more states by applying external electrical stimuli [1−3,24−28]. Many materials can be used for the switching medium, including inorganic and organics ones (transition metal oxides, perovskites, chalcogenides, polymers, etc.), and the resistive switching process typically involves the creation and rupture of conductive filaments (CF) shorting the two electrodes. These types of devices are also named filamentary RRAMs, the low-resistance state (LRS) occurs when a CF bridges the two metal layers (Fig. 1.1A-top), while the high-resistance state (HRS) is achieved by partial dissolution of the filament. The devices based on oxygen ion migration effects and subsequent valence change of the metals in metal oxides are named valence-change memory (VCM) or anion-based memory, and the resulting CF is formed by a localized concentration of defects. Examples of VCM RRAM are related to material systems based on HfO_x, TaO_x, TiO_x, SiO_x, WO_x, Al_2O_3, or other transition metal oxides in combination with TiN, TaN, Ti, Ta, and Pt electrodes [3,14,26,28]. There is another class of devices where the CF is formed by the metal ion movement from the active electrode (often Ag or Cu) into the switching medium, which are named as electrochemical metallization memory (ECM), cation-based memory, or even Conductive Bridge Random Access Memories (CBRAM) [4,14,26,27]. These types of RRAMs are fabricated by using an active electrode (Ag, Cu, Co, etc.) in combination with an inert electrode (TaN, W, Pt, etc.), while the switching medium is based on solid electrolytes such as GeSe, GeS, Ag_xS, Cu_xS, or even oxides such as SiO_2. For filamentary RRAMs, VCM or CBRAM, an initial electroforming step is necessary to establish the first filament formation; after that forming step, the cell can be repeatedly switched between the LRS and HRS (cycling endurance) up to 10^4-10^{12} cycles, depending on the materials systems and if the measurements are done at a single device or array level. RRAM is considered a nonvolatile memory since the two states can be retained for a long time (retention) up to many years. Incidentally it is worth noting that similar device structures, especially ECM, can be further engineered to achieve a threshold switching behavior suitable for selector application (more details in Chapter 5, RRAM-Based Coprocessors for Deep Learning) or even short-term retention to be exploited in spiking neural networks (SNNs, Chapter 17, Synaptic Realizations Based on Memristive Devices). An additional class of RRAM relies on uniform interfacial switching (Fig. 1.1A, bottom), in which the conductance scales with the junction area of the device, and the mechanism is related to a homogenous oxygen ion movement through the oxides leading to the modulation of the electrode/oxide interfacial Schottky barrier [2,23−26]. This RRAM type exhibits an analog switching nature, and the initial electroforming is not necessary. This uniform switching is often reported in complex oxides and perovskite, such as Bismuth Ferrite

($BiFeO_3$—BFO) and Praseodymium Calcium manganite ($Pr_{1-x}Ca_xMgO_3$ − PCMO) [2,24]. The class of RRAM devices is extensively discussed in Chapter 2, Resistive Switching Memories.

Phase-change memory (PCM) as shown in Fig. 1.1B is based on the reversible phase transitions in chalcogenide glass materials, such as thermally activated amorphous−crystalline transitions [29−32]. This class of materials involves ternary or quaternary chalcogenide glasses that frequently, but not always, involve Ge, Te, and/or Sb in various compositions following a Ge−Sb−Te ternary phase diagram. $Ge_2Sb_2Te_5$ is the most studied alloy, often with an additional fourth dopant element such as In and As. The mechanism is related to a Joule heating-induced, rapid, and reversible transition from a highly resistive amorphous phase (HRS) to a relatively high conductive polycrystalline one (LRS). To switch the PCM cells from the LRS (crystalline material) to the HRS (amorphous materials), it is necessary to heat the materials above the melting point (typically around 600°C), followed by a rapid quench of the molten phase. The reverse process from HRS to LRS is achieved by holding the temperature of the chalcogenide materials above the crystallization temperature (typically around $\approx$400°C) for a time long enough to achieve crystallization. By engineering the materials composition, it is possible to vary the crystallization and melting point temperature of the chalcogenide materials so that the PCM cell can be adapted to various applications. Moreover various cell structures have been proposed, and one critical aspect is to have a bottom electrode that can confine heat and current. The material of the bottom electrode, also called the heater, has to be carefully chosen to satisfy these requirements. More details can be found in Chapter 3, Phase-Change Memory, and in a recent book dedicated to PCM [32]. To summarize, PCM is a nonvolatile device exhibiting stable LRS and HRS with an endurance of up to 10^8−10^9 cycles and by using programming pulses of the same polarity. Recently PCM and RRAM are shown to control a multilevel or analog operation, which enables many applications toward in-memory computing, analog computing and neural networks as described in the following paragraph and in the Parts II-IV of this book.

Ferroelectric memory (FeRAM) as shown in Fig. 1.1C relies on the polarization switching in a ferroelectric material, such as a perovskite material or most recently doped-HfO_2 and $HfZrO_x$ [6,33−37]. Ferroelectric materials are characterized by two stable polarization states that can be reversibly switched from one to another by applying an external electrical field larger than the coercive field E_C, that is, the field corresponding to a zero ferroelectric polarization. As shown in Fig. 1.1C-left, the typical characteristic that can be measured in a ferroelectric material is the hysteresis curve of the polarization P as a function of the electrical field E. The field-driven switching mechanism of two stable polarization states has been exploited for nonvolatile memories in capacitor-based FeRAM or ferroelectric field effect transistor (FeFET). Capacitor-based FeRAM relies on a ferroelectric

capacitor, where a ferroelectric layer is sandwiched between two electrodes, usually fabricated in the back-end-of-line of the CMOS process and with the bottom electrode connected to the drain selector transistor in the 1T-1C configuration. Other more complex cell structures, still based on capacitor-based ferroelectric memories, are also proposed, including the use of 3D capacitors. As a second choice, the ferroelectric material can be implemented as a part of the gate dielectric of a FeFET [35]. In this option the polarization change of the material leads to a shift of the $I-V$ curve of the transistor that can be used to determine the state of the ferroelectric polarization in a nondestructive way. Recently HfO_2-based FeFET has gained considerable interest, and it is currently proposed as a possible key element to implement both a synaptic or neuronal functionality (see Chapter 4, Magnetic and Ferroelectric Memories, for more details) [37,38].

Magnetoresistive random access memories (MRAMs) as shown in Fig. 1.1D are based on the tunnel magnetoresistance phenomena in a magnetic tunnel junction (MTJ) [7,8,39–43]. An MTJ consists of two ferromagnetic metal layers (for instance, CoFeB) that sandwich a very thin (1–2-nm) insulator acting as a tunneling barrier, typically MgO. One of the two ferromagnetic layers is the reference one with a pinned magnetization orientation (*pinned layer*), whereas the other magnetic layer (*free layer*) can be switched between two opposite orientations by applying voltage pulses of opposite polarity. Depending on whether the ferromagnetic polarizations of the pinned and free layers are parallel or antiparallel, the LRS (parallel configuration) and HRS (antiparallel configuration) are obtained due to the tunnel magnetoresistive effect [40]. Currently various types of MRAM cells have been developed based on the used switching mechanisms, for instance, spin transfer torque (STT), voltage-controlled magnetic anisotropy (VCMA), and spin–orbit torque (SOT) [41–43]. Spin transfer torque MRAM (STT-MRAM) is currently the most established MRAM technology in terms of CMOS integration and density. In STT-MRAM, the magnetization direction of the free layer can be switched to the parallel configuration by a spin-polarized current: conduction electrons are first spin-polarized by the pinned layer. They then rotate the magnetic polarization of the free layer due to the conservation of magnetic momentum. Although STT-MRAMs have a larger footprint than PCM and RRAM, MRAM brings the advantage of a very well-known and consolidated switching mechanism, switching speed (down to ns and ps) and outstanding endurance ($> 10^{14}$).

To summarize, we have described the leading and most mature resistive switching technologies of current interest for brain-inspired computing. A comparison, and more details on their properties in term of programming speed, endurance, cells size, memory windows, control of the multilevel operation, can be found in many review papers [2,3,9,12–15,18,22], and they are also discussed in the following chapters and sections of this book.

That being said, it is worth noting that these characteristics are subject to continuous improvement with the potential for novel outcomes thanks to intensive research efforts worldwide. Now we introduce the novel applications beyond storage envisaged for the above-described devices.

1.3 Resistive memory devices for brain-inspired computing

All the resistive switching memories exhibit exciting properties that can be exploited for novel computation schemes. The capability of these devices to retain the information for an extended time (years), that is, their nonvolatility, fast switching speed, low switching energy, and cycling switching endurance are the main characteristics exploited for storage applications [2,3,6,7,28,29,32]. For instance, STT-MRAM or other MRAMs can be used to replace the volatile DRAM and SRAM due to their fast switching speed with the advantage of nonvolatility. Resistive switching memories such as RRAM, PCM, and FeRAM (partially) can be considered as storage-class memories (SCM), filling the gap in access time to the information between FLASH-based storage and DRAM. This addition can help to reduce the latency associated with accessing data from the memory units.

All these resistive switching technologies exhibit many other appealing features related to their rich dynamics and to device size and process integration aspects, which make them very interesting for a broad range of applications where memory is put at the core of computation, including analog computing, in-memory logic, near-memory and in-memory computing, machine learning, neuromorphic computing, and various types of proposed neural networks [1,9–23,44–49]. As a common feature we can mention the small footprint of memristive devices since most of these are two-terminal ones, and because of the switching mechanism that allows their scaling down to the nanometer scale, low process temperature fabrication (down to $<400^{\circ}C$), compatibility with CMOS integration, stackability on multilayer to increase the density. All these properties can enable the use of these devices in complex circuits and systems, and the high device density decreases the cost of computing systems and is thus an important benefit for all new types of computing applications that demand a large number of devices.

One of the proposed computing applications for memristive device is the in-memory logic, which exploits the nonvolatile binary storage capability of memristive devices to map Boolean logic states (1 and 0) to the resistance of the LRS and HRS. Stateful logic can be realized in a memory array and has been demonstrated for RRAM and STT-MRAM. [50–52] The low switching energy, fast switching speed, and cycling switching endurance of memristive

devices are already an advantage for storage but are also a benefit for almost all types of computing and for applications that demand frequent device programming and fast access to data, such as in-memory logic.

Additional intrinsic features of the memristive devices that can be further exploited for computing are (1) multilevel state or analog operation, (2) stochasticity and intrinsic variability, and (3) rich dynamics of the devices including the possibility to engineer their retention toward volatile devices, exhibiting a fast conductance decay after programming. In particular the possibility to control the programming and storage, not only of two states but of several (multilevel) and up to a continuum (analog) of conductance or resistance values, opens the possibility of many applications [1,9,47,49]. The number of states that can be stored on a memristive device is also referred to bit precision. There is a large variety of application domains that memristive devices can address by combining their mentioned properties, from the low precision (2 bit) stochastic computing and security (random generation number, unclonable functions) to an intermediate−high precision type of applications such as reservoir computing, associative memory, deep learning, sparse coding, and scientific computing. In the framework of low-precision type of applications, the stochasticity associated with the switching behavior in memristive devices has been exploited for cybersecurity. MRAM and RRAM have been proposed as key elements to implement a low-cost and easily accessible true random number generator (TRNG) and physically unclonable functions (PUF) [53−58]. In an MRAM, the MTJ switching is inherently stochastic due to the thermal fluctuations of the free layer, and the write voltage and duration can be used to tune the switching probability. In filamentary RRAM, the switching mechanism is intrinsically stochastic, and the devices show cycle-to-cycle variation leading to distribution on LRS and HRS. Another example is the random telegraph noise (RTN) resulting from metastable defect fluctuations near the CF [58].

Moving to the applications that require an intermediate−high precision (usually at least 3−4 bits or more), analog in-memory computing uses memristive devices crossbars as multiplier−accumulators to compute a vector−matrix multiplication in a single computing cycle using Ohm's law for multiplications and Kirchhoff 's current law for accumulations in parallel (additional information are reported in Chapter 3, Phase-Change Memory, and Part II of the Book). Moreover memristive devices, such as PCM and RRAM, and FeRAM to some extent, have also shown an accumulative behavior for which the conductance of devices can be progressively increased or decreased by a train of pulses. This nonvolatile accumulative behavior, despite its stochastic nature, can also be exploited for training deep neural networks.

Finally memristive devices are of considerable interest for SNNs [1,11,14,22,59−77], which target the implementation at the hardware level of biological functions of the human brain and encode information with

spikes (see Chapter 15, Memristive Devices for Spiking Neural Networks, for more information). SNNs are one of the most advanced approachs to design brain-inspired circuits that can perform computational tasks with superior power efficiency to conventional computers. Although it is not generally true that a single memristive device can implement at hardware level all the desired functionalities reproducing the synaptic or neural dynamics, memristive devices can enable the fabrication of small circuit blocks for synapses and neurons, bringing the additional advantage, with respect to standard CMOS solutions, of nonvolatility and overall smaller size. Besides most of the memristive device technologies are integrated into the back-end-of-line, thus enabling their integration with a wide range of front-end CMOS technologies. As an example of the hardware implementation of synapses, the analog control of conductance states by the train of pulses in RRAM, PCM, and FeRAM devices has been used to reproduce the synaptic plasticity of biological synapses [22, 46–48, 59–63, 66–68, 74–77]. The conductance of these electronic synapses can be updated by different biological inspired learning rules relying on the frequency and/or time difference of pre and postsynaptic pulses (more detailed in Chapter 2, Resistive Switching Memories, and 17, Synaptic Realizations Based on Memristive Devices). Moreover the rich memristive device dynamics, for instance, the fast- or slow- conductance decay observed in some devices, can be exploited to emulate specific synaptic and neurodynamics functions, such as short-term plasticity, or to build more complex spatiotemporal information processing capabilities. Finally MRAM, PCM, and FeFET have been exploited to reproduce the neural functionality taking advantage of stochastic switching and accumulative behavior to reproduce the integrate-and-fire neural dynamics [38,59,70,78,79] (more detailed in Chapter 4, Magnetic and Ferroelectric Memories, and Chapter 16, Neuronal Realizations Based on Memristive Devices).

1.4 Conclusions and perspectives

In summary in this chapter we introduced various technologies for memristive devices including their related physical mechanisms, the basic operation principles of such devices, and the current application domains taking advantage of their characteristics. The following chapters of Part I of this book provide an in-depth discussion of the leading technologies, while the application of these technologies for computational memories, deep learning, and spiking neural networks are discussed in Parts II–IV of the book. Chapter 2, Resistive Switching Memories, introduces memristive devices based on redox reactions and electrochemical phenomena in oxides (RRAM). The chapter reviews both filamentary and interfacial switching devices, which rely on different materials systems and mechanisms, as well as devices showing negative differential resistance, and currently investigated RRAM

applications. Chapter 3, Phase-Change Memory, introduces the memristive devices that rely on phase change in chalcogenide-based systems upon electrical stimuli. It reviews the current state-of-the-art of PCM-based technologies, including electrical transport, structural dynamics (crystallization and structural relaxation), integrations aspects, and target application for PCM in the field of brain-inspired computing. Chapter 4, Magnetic and Ferroelectric Memories, introduces memristive devices based on purely electronic effects, relying on ferromagnetic (spintronic) or ferroelectric materials, such as MRAM, STT-MRAM, and FeRAM. Some of these devices, such as spin−torque magnetic random access memory, have already reached industrial maturity. Finally Chapter 5, Selector Devices for Emerging Memories, reviews the current state-of-the-art of the proposed selectors for resistive memory by describing material systems and device characteristics and with a view on 3D integration strategies.

It is worth noting that the reviewed memristive devices are today at a high level of maturity and have already shown the ability to be integrated into scaled CMOS-circuits and systems at a very large-scale level of integration, together with suitable selector devices. They have also shown a great potential to go beyond pure memory device application for storage, and then they can be used as key elements for computational memory, neural networks, and logic-in-memory operation, to overcome the von Neumann bottleneck, as discussed in Parts II−IV of this book. Finally other emerging devices and materials systems, not reviewed in this introductory chapter, are also of increasing interest for future novel computation schemes. A nonexhaustive list can include threshold switching devices, metal−insulator transition (MIT) devices, 2D materials RRAM, photonic memory devices, organic material for flexible memristive systems, topological insulators, and skyrmions [10,72,73,80−89]. Therefore new types of memristive devices can potentially enable hitherto challenging computations far more efficiently than current methods in the future.

References

[1] Z. Wang, H. Wu, G.W. Burr, C.S. Hwang, K.L. Wang, Q. Xia, et al., Resistive switching materials for information processing, Nat. Rev. Mater 5 (2020) 173−195.

[2] S. Slesazeck, T. Mikolajick, Nanoscale resistive switching memory devices: a review, Nanotechnology 30 (2019) 352003.

[3] D. Wouters, R. Waser, M. Wuttig, Phase-change and redox-based resistive switching memories, Proc. IEEE. 103 (8) (2015) 1274−1288.

[4] D.S. Jeong, et al., Emerging memories: resistive switching mechanisms and current status, Rep. Prog. Phys. 75 (2012) 076502.

[5] X. Zhu, S.H. Lee, W.D. Lu, Nanoionic resistive-switching devices, Adv. Electron. Mater. 5 (2019) 1900184.

[6] T. Mikolajick, U. Schroeder, S. Slesazeck, The past, the present, and the future of ferroelectric memories, IEEE Trans. Electron Devices 67 (4) (2020) 1434−1443.

[7] S. Bhatti, R. Sbiaa, A. Hirohata, H. Ohno, S. Fukami, N. Piramanayagam, Spintronics based random access memory: a review, Mater. Today 20 (Issue 9) (2017) 530–548.
[8] S. Ikeda, K. Miura, H. Yamamoto, K. Mizunuma, H.D. Gan, M. Endo, et al., A perpendicular-anisotropy CoFeB–MgO magnetic tunnel junction, Nat. Mater. 9 (2010) 721–724.
[9] A. Sebastian, M. Le Gallo, R. Khaddam-Aljameh, E. Eleftheriou, Memory devices and applications for in-memory computing, Nat. Nanotechnol. (2020).
[10] S.H. Sung, D.H. Kim, T.J. Kim, I.-S. Kang, K.J. Lee, Unconventional inorganic-based memristive devices for advanced intelligent systems, Adv. Mater. Technol. 4 (2019) 1900080.
[11] T. Zhang, K. Yang, X. Xu, Y. Cai, Y. Yang, R. Huang, Memristive devices and networks for brain-inspired computing, Phys. Status Solidi 13 (2019). 1900029.
[12] D. Ielmini, S. Ambrogio, Emerging neuromorphic devices, Nanotechnology 31 (2020) 092001.
[13] D. Ielmini, H.-S.P. Wong, In-memory computing with resistive switching devices, Nat. Electron. 1 (2018) 333–343.
[14] R. Dittmann, J.P. Strachan, Redox-based memristive devices for new computing paradigm, APL Mater. 7 (2019) 110903.
[15] N.K. Upadhyay, H. Jiang, Z. Wang, S. Asapu, Q. Xia, J.J. Yang, Emerging memory devices for neuromorphic computing, Adv. Mater. Technol. 4 (2019) 1800589.
[16] G.W. Burr, et al., Neuromorphic computing using non-volatile memory, Adv. Phys. X 2 (2016) 89–124.
[17] S. Yu, Neuro-inspired computing with emerging nonvolatile memorys, Proc. IEEE 106 (2018) 260–285.
[18] J.J. Yang, D.B. Strukov, D.R. Stewart, Memristive devices for computing, Nat. Nanotechnol 8 (2013) 13–24.
[19] C.-H. Kim, S. Lim, S.Y. Woo, W.-M. Kang, Y.-T. Seo, S.-T. Lee, et al., Emerging memory technologies for neuromorphic computing, Nanotechnology 30 (2019). 032001 (33pp).
[20] H. Jeong, L. Shi, Memristor devices for neural networks, J. Phys. D Appl. Phys 52 (2019) 023003.
[21] M.A. Zidan, J.P. Strachan, W.D. Lu, The future of electronics based on memristive systems, Nat. Electron 1 (2018) 22–29.
[22] V. Milo, G. Malavena, C.M. Compagnoni, D. Ielmini, Memristive and CMOS devices for neuromorphic computing, Materials 13 (2020) 166.
[23] W. Zhang, B. Gao, J. Tang, X. Li, W. Wu, H. Quian, et al., Analog-type resistive switching devices for neuromorphic computing, Phys. Status Solidi 13 (2019). 1900204.
[24] A. Sawa, Resistive switching in transition metal oxides, Mater. Today 11 (2008) 28–36.
[25] R. Waser, M. Aono, Nanoionics-based resistive switching memories, Nat. Mater. 6 (2007) 833–840.
[26] R. Waser, R. Dittmann, G. Staikov, K. Szot, Redox-based resistive switching memories – nanoionic mechanisms, prospects, and challenges, Adv Mater. 21 (2009) 2632–2663.
[27] I. Valov, R. Waser, J.R. Jameson, M.N. Kozicki, Electrochemical metallization memories—fundamentals, applications, prospects, Nanotechnology 22 (2011) 254003.
[28] Daniele Ielmini, Rainer Waser (Eds.), Resistive Switching: From Fundamentals of Nanoionic Redox Processes to Memristive Device Applications, Wiley, 2016. 784 p. (ISBN: 978-3-527-33417-9).

[29] M. Wuttig, N. Yamada, Phase-change materials for rewriteable data storage, Nat. Mater. 6 (2007) 824–832.
[30] G.W. Burr, et al., Recent progress in phase-change memory technology, IEEE J. Emerg. Sel. Top. Circuits Syst 6 (2016) 146–162.
[31] S. Raoux, W. Wełnic, D. Ielmini, Phase change materials and their application to nonvolatile memories, Chem. Rev. 110 (2010) 240–267.
[32] Phase Change Memory Device Physics, Reliability and Applications. Andrea Redaelli editor, Springer 2018.
[33] T. Boescke, J. Mueller, D. Braeuhaus, U. Schroeder, U. Boettger, Ferroelectricity in hafnium oxide thin films, Appl. Phys. Lett. 99 (2011) 102903.
[34] T. Mikolajick, S. Slesazeck, M.H. Park, U. Schroeder, Ferroelectric hafnium oxide for ferroelectric random-access memories and ferroelectric field-effect transistors, MRS Bull. 43 (2018) 340–346.
[35] B.-E. Park, H. Ishiwara, M. Okuyama, S. Sakai, Ferroelectric-Gate Field Effect Transistor Memories: Device Physics and Applications, Springer, Dordrecht, The Netherlands, 2016.
[36] M.H. Park, Y.H. Lee, H.J. Kim, Y.J. Kim, T. Moon, K.D. Kim, et al., Understanding the formation of the metastable ferroelectric phase in hafnia–zirconia solid solution thin films, Nanoscale 10 (2018) 716–725.
[37] H. Mulaosmanovic, J. Ocker, S. Müller, M. Noack, J. Müller, P. Polakowski, et al., Novel ferroelectric FET based synapse for neuromorphic systems, Symp. VLSI Technol. (2017) T176–T177.
[38] H. Mulaosmanovic, E. Chicca, M. Bertele, T. Mikolajick, S. Slesazeck, Mimicking biological neurons with a nanoscale ferroelectric transistor, Nanoscale 10 (2018) 21755.
[39] A. Brataas, A.D. Kent, H. Ohno, Current-induced torques in magnetic materials, Nat. Mater. 11 (2012) 372–381.
[40] C. Chappert, A. Fert, F. Nguyen Van Dau, The emergence of spin electronics in data storage, Nat. Mater. 6 (2007) 813–823.
[41] A.V. Khvalkovskiy, D. Apalkov, S. Watts, R. Chepulskii, R.S. Beach, A. Ong, et al., Basic principles of STT-MRAM cell operation in memory arrays, J. Phys. D: Appl. Phys. 46 (2013) 139601.
[42] A. Manchon, H.C. Koo, J. Nitta, S.M. Frolov, R.A. Duine, New perspectives for Rashba spin–orbit coupling, Nat. Mater. 14 (2015) 871–882.
[43] C.-G. Duan, et al., Surface magnetoelectric effect in ferromagnetic metal films, Phys. Rev. Lett. 101 (2008) 137201.
[44] A. Sebastian, M. Le Gallo, G.W. Burr, S. Kim, M. BrighSky, E. Eleftheriou, Tutorial: Brain-inspired computing using phase-change memory devices, J. Appl. Phys. 124 (2018) 111101.
[45] Y. Zhang, Z. Wang, J. Zhu, Y. Yang, M. Rao, W. Song, et al., Brain-inspired computing with memristors: Challenges in devices, circuits, and systems, Appl. Phys. Rev. 7 (2020) 011308.
[46] J. d Valle, J.G. Ramirez, M.J. Rozenberg, I.K. Schuller, Challenges in materials and devices for resistive-switching-based neuromorphic computing, J. Appl. Phys. 24 (2018) 211101.
[47] D. Ielmini, Brain-inspired computing with resistive switching memory (RRAM): Devices, synapses and neural networks, Microelectron. Eng. 190 (2018) 44–53.

[48] M. Suri, D. Querlioz, O. Bichler, G. Palma, E. Vianello, D. Vuillaume, et al., Bio-inspired stochastic computing using binary CBRAM synapses, IEEE Trans. Electron Devices 60 (2013) 2402.
[49] C. Li, C.E. Graves, X. Sheng, D. Miller, M. Foltin, G. Pedretti, et al., Analog content-addressable memories with memristors, Nat. Commun. 11 (2020) 1638.
[50] J. Borghetti, et al., Memristive switches enable stateful logic operations via material implication, Nature 464 (2010) 873.
[51] S. Kvatinsky, et al., MAGIC-memristor-aided logic, IEEE Trans. Circuits Syst. II Express Briefs 61 (2014) 895–899.
[52] H. Mahmoudi, T. Windbacher, V. Sverdlov, S. Selberherr, Implication logic gates using spin-transfer-torque-operated magnetic tunnel junctions for intrinsic logic-in-memory, Solid State Electron 84 (2013) 191–197.
[53] R. Carboni, D. Ielmini, Stochastic memory devices for security and computing, Adv. Electron. Mater. 5 (2019) 1900198.
[54] H. Jiang, et al., A novel true random number generator based on a stochastic diffusive memristor, Nat. Commun. 8 (2017) 882.
[55] W.H. Choi, et al., A magnetic tunnel junction based true random number generator with conditional perturb and real-time output probability tracking, Proc. The International Electron Devices Meeting 12–5, IEEE, 2014.
[56] H. Nili, et al., Hardware-intrinsic security primitives enabled by analogue state and non-linear conductance variations in integrated memristors, Nat. Electron. 1 (2018) 197.
[57] R. Carboni, et al., Random number generation by differential read of stochastic switching in spin-transfer torque memory, IEEE Electron Device Lett. 39 (2018) 951–954.
[58] S. Ambrogio, et al., Statistical fluctuations in HfO_x resistive-switching memory (RRAM): Part II—Random telegraph noise, IEEE Trans. Electron Devices 61 (2014) 2920–2927.
[59] J. Grollier, D. Querlioz, K.Y. Camsari, et al., Neuromorphic spintronics, Nat. Electron. (2020). Available from: https://doi.org/10.1038/s41928-019-0360-9.
[60] W. Wang, G. Pedretti, V. Milo, R. Carboni, A. Calderoni, N. Ramaswamy, et al., Learning of spatiotemporal patterns in a spiking neural network with resistive switching synapses, Sci. Adv 4 (2018) eaat4752.
[61] T. Dalgaty, M. Payvand, F. Moro, D.R.B. Ly, F. Pebay-Peyroula, J. Casas, et al., Hybrid neuromorphic circuits exploiting non-conventional properties of RRAM for massively parallel local plasticity mechanisms, APL Mater. 7 (2019) 081125.
[62] V. Milo, C. Zambelli, P. Olivo, E. Pérez, M.K. Mahadevaiah, O.G. Ossorio, et al., Multilevel HfO_2-based RRAM devices for low-power neuromorphic networks, APL Mater. 7 (2019) 081120.
[63] J. Frascaroli, S. Brivio, E. Covi, S. Spiga, Evidence of soft bound behaviour in analogue memristive devices for neuromorphic computing, Sci. Rep. 8 (1) (2018) 7178.
[64] S. Battistoni, V. Erokhin, S. Iannotta, Frequency driven organic memristive devices for neuromorphic short term and long term plasticity, Org. Electron. 65 (2019) 434–438.
[65] S. La Barbera, D.R.B. Ly, G. Navarro, N. Castellani, O. Cueto, G. Bourgeois, et al., Narrow heater bottom electrode-based phase change memory as a bidirectional artificial synapse, Adv Electron Mater 4 (9) (2018) 1800223.
[66] S. Brivio, D. Conti, M.V. Nair, J. Frascaroli, E. Covi, C. Ricciardi, et al., Extended memory lifetime in spiking neural networks employing memristive synapses with nonlinear conductance dynamics, Nanotechnology 30 (1) (2019) 015102.

[67] E. Covi, S. Brivio, A. Serb, T. Prodromakis, M. Fanciulli, S. Spiga, Analog memristive synapse in spiking networks implementing unsupervised learning, Front. Neurosci. 10 (2016) 482.
[68] G. Indiveri, B. Linares-Barranco, R. Legenstein, G. Deligeorgis, T. Prodromakis, Integration of nanoscale memristor synapses in neuromorphic computing architectures, Nanotechnology 24 (2013) 384010.
[69] T. Tuma, A. Pantazi, M. Le Gallo, A. Sebastian, E. Eleftheriou, Stochastic phase-change neurons, Nat. Nanotechnol. 11 (2016) 693–699.
[70] M. Jerry, A. Parihar, B. Grisafe, A. Raychowdhury and S. Datta, Ultra-low power probabilistic IMT neurons for stochastic sampling machines, 2017 Symposium on VLSI Technology, Kyoto, 2017, pp. T186-T187.
[71] D. Kuzum, S. Yu, H.-S.P. Wong, Synaptic electronics: materials, devices and applications, Nanotechnology 24 (2013) 382001.
[72] S. Battistoni, C. Peruzzi, A. Verna, S.L. Marasso, M. Cocuzza, V. Erokhin, et al., Synaptic response in organic electrochemical transistor gated by a graphene electrode, Flex. Print. Electron. 4 (2019) 044002.
[73] E. Juzekaeva, A. Nasretdinov, S. Battistoni, T. Berzina, S. Iannotta, R. Khazipov, et al., Coupling cortical neurons through electronic memristive synapse, Adv. Mater. Technol. 4 (2019) 1800350.
[74] I. Boybat, M. Le Gallo, S.R. Nandakumar, T. Moraitis, T. Parnell, T. Tuma, et al., Neuromorphic computing with multi-memristive synapses, Nat. Commun. 9 (2018) 2514.
[75] S. Boyn, et al., Learning through ferroelectric domain dynamicsin solid-state synapses, Nat. Commun. 8 (2016) 14736.
[76] M.-K. Kim, J.-S. Lee, Ferroelectric analog synaptic transistors, Nano Lett. 19 (2019) 2044–2050.
[77] M. Halter, L. Bégon-Lours, V. Bragaglia, M. Sousa, B.J. Offrein, S. Abel, et al., Back-end, cmos-compatible ferroelectric field-effect transistor for synaptic weights, ACS Appl. Mater. Interfaces 12 (15) (2020) 17725–17732.
[78] M. Wu et al., Extremely Compact Integrate-and-Fire STT-MRAM Neuron: A Pathway toward All-Spin Artificial Deep Neural Network, In: 2019 Symposium on VLSI Technology, Kyoto, Japan, 2019, pp. T34-T35.
[79] A. Pantazi, S. Wozniak, T. Tuma, E. Eleftheriou, All-memristive neuromorphic computing with level-tuned neurons, Nanotechnology 27 (2016) 355205.
[80] C. Ríos, et al., Integrated all-photonic non-volatile multi-level memory, Nat. Photon 9 (2015) 725.
[81] M. Wang, et al., Robust memristors based on layered two- dimensional materials, Nat. Electron. 1 (2018) 130–136.
[82] Y. Fan, et al., Magnetization switching through giant spin–orbit torque in a magnetically doped topological insulator heterostructure, Nat. Mater. 13 (2014) 699–704.
[83] M. Wuttig, H. Bhaskaran, T. Taubner, Phase-change materials for non-volatile photonic applications, Nat. Photon 11 (2017) 465.
[84] Maheswari Sivan, et al., All WSe_2 1T1R resistive RAM cell for future monolithic 3D embedded memory integration, Nat. Commun. 10 (2019) 520.
[85] L. Zhang, T. Gong, H. Wang, Z. Guoband, H. Zhan, Memristive devices based on emerging two-dimensional materials beyond graphene, Nanoscale 11 (2019) 12413.
[86] J. d Valle, Y. Kalcheim, J. Trastoy, A. Charnukha, D.N. Basov, I.K. Schuller, Electrically induced multiple metal-insulator transitions in oxide nanodevices, Phys. Rev. Appl. 8 (2017) 054041.

[87] W. Kang et al., Magnetic skyrmions for future potential memory and logic applications: Alternative information carriers, In: 2018 Design, Automation & Test in Europe Conference & Exhibition (DATE), Dresden, 2018, pp. 119–124.

[88] K. Yue, Y. Liu, R.K. Lake, A.C. Parker, A brain-plausible neuromorphic on-the-fly learning system implemented with magnetic domain wall analog memristors, Sci. Adv. 5 (4) (2019) eaau8170.

[89] C.-Y. Wang, C. Wang, F. Meng, P. Wang, S. Wang, S.-J. Liang, et al., 2D layered materials for memristive and neuromorphic applications, Adv. Electron. Mater. 6 (2020) 1901107.

Chapter 2

Resistive switching memories

Stefano Brivio[1] and Stephan Menzel[2]
[1]CNR—IMM, Unit of Agrate Brianza, Agrate Brianza, Italy, [2]Peter-Grünberg-Institut 7, Forschungszentrum Jülich GmbH, Juelich, Germany

2.1 Introduction

In this chapter, the resistive switching memory technology is reviewed. Resistive switching random access memories (RRAMs), named also memristive devices, are metal/insulator/metal (MIM) structures displaying volatile or nonvolatile resistance changes, when a sufficiently strong electric stimulus is applied to their two terminals. A nonvolatile resistance change—that is, persistent in time with no applied voltage—always involves ionic movement and local redox reactions, which justifies the name *redox-based devices* used in the following. Indeed, the insulator layer of the devices is usually a mixed ionic−electronic conductor (MIEC), as a transition metals oxide or a chalcogenide [1], which supports both electron flow and ion migration. Devices showing volatile resistance changes can base their operation on temperature- or electric field−driven insulator-to-metal transitions (IMTs) or threshold switching events. The details of the programming operation and the physics of the switching are addressed in Section 2.2. Section 2.3 correlates the explained physical concepts with the expected performance for the various nonvolatile memory technologies. Section 2.4 deals with advanced concepts, functionalities, programming operations enabling various kinds of unconventional in-memory computing schemes (described in Chapters 6−11), implementations of deep learning accelerators (Chapters 12−14) and of spiking neural networks and computation based on interacting oscillators (Chapter 15−19). Finally, some conclusions and perspectives are drawn in Section 2.5.

2.2 Basic concepts of the physics of resistive switching

Fig. 2.1A shows a pictorial quasi-static $I-V$ characteristic of a bipolar binary RRAM device and defines the conventional employed terminology. For a binary device, the resistance of a MIM stack can be switched under

Memristive Devices for Brain-Inspired Computing. DOI: https://doi.org/10.1016/B978-0-08-102782-0.00002-2

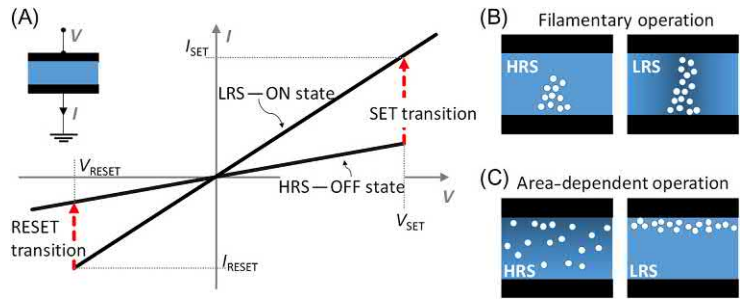


FIGURE 2.1 (A) Basic switching $I-V$ characteristic for a bipolar operation; the memristive switching phenomenon can interest a filamentary region (B) or the entire device section (C).

suitable voltage application from a low resistance state (LRS) or ON state to a high resistance state (HRS) or OFF state, and vice versa. The transition from the HRS to the LRS is called SET transition, whereas the reverse transition is called RESET. In the bipolar operation mode, the device switches to the LRS with one voltage polarity. To reset the device, the opposite voltage polarity needs to be applied. The voltage at which the SET/RESET transition sets in is called SET/RESET voltage, V_{SET}/V_{RESET}. Likewise, the highest absolute current during SET/RESET is called SET/RESET current, I_{SET}/I_{RESET}, as shown in Fig. 2.1A. RRAM devices exist in which the SET and RESET occur independently to the applied voltage polarity. Such devices are called unipolar or nonpolar devices, and they usually exhibit highly variable, relatively high power operation, and, most importantly, low endurance (i.e., a low maximum number of SET/RESET cycles before device irreversible failure) [2]. For these reasons, nonpolar devices are not considered relevant for computing application, in which extended endurance is fundamental. Therefore, they are not addressed in this chapter, but extended information can be found in Refs. [1,2].

The spatial extent of the active switching area can be used to categorize RRAM devices. Indeed, as illustrated in Fig. 2.1B and C, the switching can occur locally in a conductive filament (CF) region (filamentary switching) or in the whole cross section of the device (area-dependent or homogenous switching) [3]. In principle, both modes can coexist in the same device stack [4].

The information stored in the RRAM device is read at a low voltage, which is supposed not to produce any switching event. As the resistance states can be nonlinear, the read current can be much lower than the write current. In principle, RRAM devices can be programmed in more than two resistance states. This enables multilevel (multibit) or analog operation. The ratio between the maximum and minimum programmable resistance is called memory window.

In nonvolatile RRAMs, the driving force for the resistance change is based on the motion of ionic defects and local redox reactions. Three different mechanisms are identified: the electrochemical metallization mechanism (ECM)—for devices named conductive bridge random access memories (CBRAMs) or Atomic Switches; the valence change mechanism (VCM)—for device named OxRAMs; and the thermochemical mechanism (TCM)—also named fuse/antifuse mechanism [1]. ECM and VCM devices typically show a bipolar switching mode, whereas TCM devices show a unipolar operation mode, which will not be addressed here for the reasons explained earlier. ECM (VCM) relies on the motion of metal cations (oxygen defects), and thus the corresponding devices are also referred to as cationic (anionic) [1].

In the following subsections, we discuss the physics of cation-based ECM cells (Section 2.2.1) and anion-based VCM cells (Section 2.2.2). In addition, Section 2.2.3 deals with negative differential resistance (NDR) behavior due to volatile switching.

2.2.1 Resistive switching based on cation migration

The MIM stack of an ECM cell consists of one chemically active Ag or Cu electrode, an ion-conducting layer, and an inert counter electrode [1]. The ion-conducting layer is a primary solid electrolyte, a secondary solid electrolyte, or an untypical solid electrolyte. The first group comprises compounds of the active ionic species, for example, AgI [5], $RbAg_4I_5$ [6], Ag_2S [7], or Cu_2S [8]. Secondary electrolytes are well-known ion conductors for Ag or Cu cations, for example, GeS_x [9,10] or $GeSe_x$ [11,12]. In untypical solid electrolytes, the required anionic counter charge is provided externally. For example, residual water in SiO_2 or Ta_2O_5 can provide OH^- ions by cathodic reactions under applied bias [13,14]. It has been shown that porous untypical solid electrolytes are beneficial for fast device operation [15]. In addition, the catalytic activity for water splitting of the inert counter electrode can influence the switching dynamics [16,17].

The physical and electrochemical processes occurring in an ECM cell are illustrated in Fig. 2.2 along with a typical $I-V$ characteristic [18]. To set the device, a positive voltage is applied to the chemically active electrode Ag in Fig. 2.2A. The electrode is oxidized and Ag cations are injected into the solid electrolyte layer (Fig. 2.2B). The cations migrate within the electric field to the cathode where they are reduced to metallic Ag. After the formation of a stable nucleus, a Ag CF grows toward the anode by further electroreduction (Fig. 2.2C). When the CF tip approaches the anode, electron tunneling sets in and the device resistance decreases (Fig. 2.2D). Depending on the applied current compliance, a tunneling gap remains or a galvanic contact is established, which may grow laterally in size [19]. This latter configuration corresponds to the device LRS and always supports a metallic-like

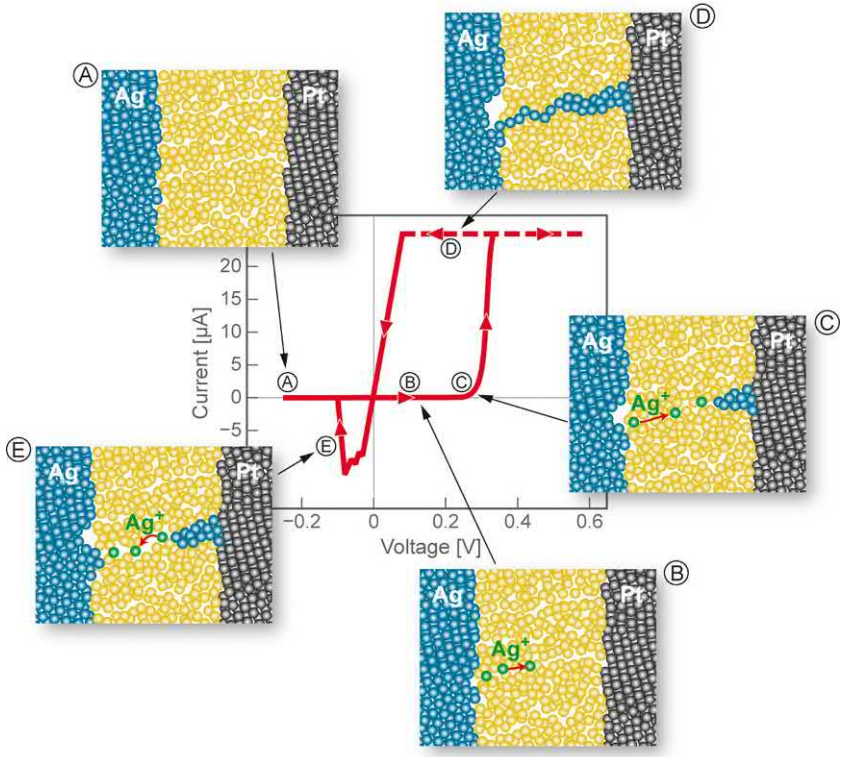


FIGURE 2.2 Sketch of the SET (A)–(D) and RESET (E) operations of an ECM cell. In the figure, the voltage is applied to the Ag electrode and Pt electrode is always grounded. *Reproduced with permission from I. Valov, R. Waser, J.R. Jameson and M.N. Kozicki, Electrochemical metallization memories—fundamentals, applications, prospects, Nanotechnology. 22 (25), 2011, 254003.*

conduction. To reset the device, a negative voltage is applied to the Ag electrode. If a tunneling gap remains after the SET operation, all electrochemical and ionic processes are reversed and the CF dissolves (Fig. 2.2E). If the Ag CF bridges the complete solid electrolyte, the CF needs to dissolve first laterally until a gap opens [20–22]. Then, the CF dissolution proceeds as in the previous case.

The described filamentary growth mode is the conventional one. Based on the ratio between reduction/oxidation rate and migration rate, the growth mode can change [23]. The conventional just described growth mode occurs if the reduction/oxidation rate is the limiting process. In contrast, the CF may grow from the anode to the cathode, if the migration rate is the limiting process. The latter growth mode is dominating in gap-type atomic switches [24].

In ECM cells, the SET and RESET transitions are both very abrupt. One reason for this behavior is that the electron tunneling process between the

CF tip and the counter electrode depends exponentially on the inverse tunneling distance. This is also the reason for the high resistance window of ECM devices. Thus a small change in the tunneling gap leads to a high current jump. The opening/closure of the tunneling gap can be achieved by removing/adding only a small amount of atoms, at least if the CF diameter is small. If a galvanic contact is achieved during SET, the removal of single atoms from the CF induces only a minor change of the resistance [25]. In this case, a more gradual RESET process can be expected. This conclusion is supported by scalpel AFM tomography of $Cu/Al_2O_3/TiN$ devices [22]. After an abrupt RESET transition, a broken CF was found. In contrast, a CF with a narrow constriction was found after a gradual RESET transition.

The fast resistance change during SET must be contained by limiting the current flow through the RRAM device by a series transistor kept in saturation regime. In particular, the LRS resistance can be tuned by varying the gate voltage on the transistor [11,12]. This fact is usually pictorially ascribed to an increase CF size for higher current compliance. In case the formed CF is too big to be dissolved, for example, in case of no applied current limitation, the device fails.

It should be mentioned that the metallic CF is not always stable in the insulating layer, especially if it is very thin. Depending on the interfacial energy, the CF may dissolve into small nanospheres, deteriorating the device low resistive state [26,27]. Volatile switching ECM devices programmed at very low compliance currents have been proposed exploiting this instability [28–30].

From an operative point of view, it must be mentioned that, in some cases, the first SET operation in ECM devices may require a higher voltage (and power) than the following. This operation is called electroforming and constitutes an initialization procedure that produces the first filament. The electroforming step can be necessary only in case the corresponding cations are not already present in the ion-conducting (electronic insulator) layer, that is, only in case the ion-conducting layer is a secondary solid electrolyte or an untypical solid electrolyte.

2.2.2 Resistive switching based on anion migration

In this subsection, we discuss the switching mechanism based on the migration of oxygen defects (anion migration), which can lead to bipolar filamentary, area-dependent switching, and complementary switching operations.

2.2.2.1 Filamentary bipolar switching

Bipolar filamentary resistive switching based on anion migration was found in many different oxide materials, for example, HfO_2, Ta_2O_5, TiO_2, and

$SrTiO_3$ [1]. As switching layer, also bilayers of different oxides or of the same oxide with different stoichiometry are reported [31–33]. The switching layer is sandwiched between two metal electrodes. Typically, one of the metal electrodes consists of a high work function, chemically inert material. An easily oxidizable metal with a low work function is often chosen as a second electrode material: the ohmic electrode. Systems with symmetric electrodes often show unipolar switching or complementary switching, as discussed below [34,35].

In the as-fabricated state, bipolar filamentary VCM cells are highly insulating. An electroforming step is required to enable repetitive switching between resistance states that are less resistive than the initial resistance state. In this first forming step, the active filamentary region is formed, in which the switching takes place [36,37]. It has been reported that several filamentary regions can be formed in the oxide layer [38]. In this case, the active CF might change during cycling causing cycle-to-cycle variability [39]. The forming operation is characterized by a quite abrupt current increase indicating some kind of runaway effect. Indeed, it has been proposed that local instabilities first lead to a volatile current increase, which then triggers the chemical processes inducing a permanent resistance change [40,41].

After the electroforming process, a filamentary region with a high concentration of oxygen vacancies, $V_O^{\bullet\bullet}$, is created [1,36]. In a simplified picture, the SET and RESET processes can be understood in terms of a redistribution of the oxygen defects in this filamentary region close to one of the electrodes, as illustrated in Fig. 2.3 [42]. The electrode represented in the figure is the high work function inert electrode. In the HRS, the region close to this electrode has a low $V_O^{\bullet\bullet}$ concentration (Fig. 2.3A). Thus the electrostatic barrier at the metal/oxide interface limits the current injection. In addition, the $V_O^{\bullet\bullet}$ concentration determines the local conductivity, as oxygen vacancies act as donors. By applying a negative voltage to the inert electrode, oxygen vacancies drift toward it (Fig. 2.3B). Therefore, the $V_O^{\bullet\bullet}$ concentration close to this interface increases. Consequently, the electrostatic barrier is lowered and its thickness is reduced allowing easy current injection. In addition, the local conductivity close to the electrode increases enabling a high current transport through the filamentary region (Fig. 2.3C). For low-ohmic LRS, obtained through high SET currents, the $I–V$ curve shows a symmetric, quite linear behavior. At lower current levels, the $I–V$ characteristic becomes more asymmetric and nonlinear. When a positive potential is applied to the electrode, the oxygen vacancies move away from the interface (Fig. 2.3D). The electrostatic barrier is restored, and the CF partially dissolves (Fig. 2.3A). Still, the conduction in the HRS state is filamentary. In other cases, in the HRS state, the barrier between the electrode and the filamentary region with low defect concentration does not limit the conduction through the device; the current is rather limited by the electronic

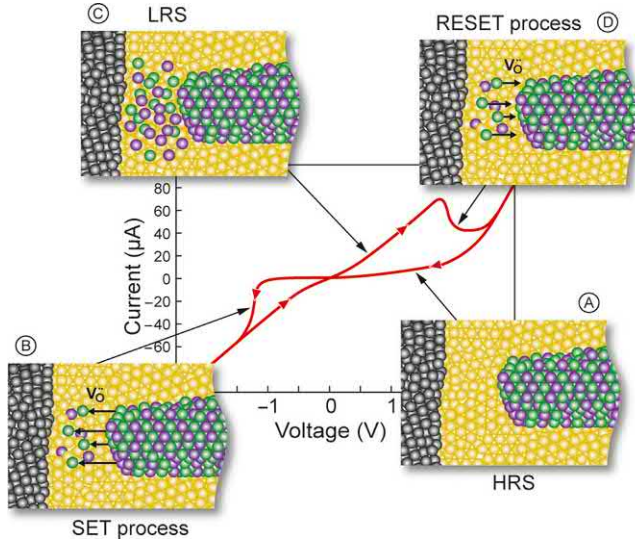


FIGURE 2.3 Illustration of the SET and RESET processes of a filamentary switching VCM cell. The *green spheres* represent the oxygen vacancies, $V_O^{\bullet\bullet}$, the *yellow* and *purple spheres* represent the cations of the binary metal oxide in their standard oxidation state and the reduced state, respectively. The *gray spheres* on the left resemble the inert electrode. *Reproduced with permission from R. Waser, R. Bruchhaus, S. Menzel. Redox-based resistive switching memories, 2012. Copyright 2012, Wiley-VCH.*

conduction through defects, as in the case of trap-assisted tunneling [43,44]. In this case, the SET operation produces the accumulation of oxygen vacancies close to the electrode providing states within the band gap and favoring the electronic conduction. The RESET causes the dispersion of the same oxygen vacancies, thus inhibiting the conduction [43,44].

The switching with the described polarities for the SET and RESET transitions is referred to as counter-eightwise switching [4]. The SET transition in filamentary switching VCM cells is quite abrupt whereas the RESET transition is rather gradual. This behavior is the result of the interplay between the dynamics of ion migration, Joule heating effects, and the sensitivity of the current change with respect to a change in the atomic configuration. If a constant voltage is applied, a gradual current increase followed by an abrupt current jump is observed [45]. The asymmetric dynamics of SET and RESET transitions can be explained with the following mechanism. During SET, the ionic defects move, and, due to the low sensitivity of the current to a change in the atomic configuration, the current increases only slightly. This increase in current leads to an increase of the dissipated power, and thus the temperature increases in the CF. The increased temperature accelerates the ionic processes, which leads to a further current increase. This positive feedback mechanism leads to a thermal runaway and to the observed

current jump [45]. The RESET transition, in contrast, is more gradual. During RESET, the current decreases, and, thus the local temperature decreases. The ionic processes slow down (negative feedback). In addition, the ionic drift gives rise to a concentration gradient. Thus ion diffusion sets in, which counteracts the ionic drift. Eventually, a balance between drift and diffusion is established [46]. The combination of the described processes leads to the observed gradual RESET switching behavior. The gradual RESET process allows programming a targeted resistance values by controlling the RESET voltage amplitude and/or the programming time [47–49]. It should be noted that an additional series resistance may lead to a more gradual SET operation, but at the same time to a more abrupt RESET operation. This change in behavior is related to a voltage divider effect, which changes the positive and negative feedbacks during SET and RESET transition [33]. Recently, it was shown that an oxygen exchange process at the ohmic electrode influences the switching dynamics [50]. Using W, for example, as ohmic electrode material in a W/Ta_2O_5/Pt device, allowed programming a larger resistance window compared with a Ta/Ta_2O_5/Pt stack. Thus not only the oxide material, but also the interface to the electrode needs to be optimized to obtain the desired properties.

Oxygen exchange reactions can also occur at the inert electrode [51,52]. If that mechanism dominates, the switching polarity changes to a so-called eightwise switching mode, that is, the SET/RESET transition occurs with a positive/negative voltage applied to the inert electrode. Moreover, both mechanisms of ionic movement and of oxygen exchange can interact leading to a more gradual transition until one of the mechanisms dominates. Thus eightwise and counter-eightwise switching can coexist in the same device as frequently reported [4,53,54].

2.2.2.2 *Complementary switching*

A special switching mode is the so-called complementary resistive switching (CRS) or complementary switching. In a CRS device, there are two different switching locations. While the oxygen vacancies move during SET toward one of the two electrodes, they deplete at the opposite one [37]. Therefore, if one interface switches on, the opposite one switches off. The mechanism of CRS is illustrated in Fig. 2.4. In the figure, the sketches of the MIM stack report also the concentration of defects through the insulating layer. It is shown that defects are removed from one interface or from the opposite one depending on the applied voltage, thus producing a SET and a RESET transition in sequence and current peaks for both polarities. By stopping the voltage at either one of the current peaks or by applying current compliance a SET operation, the RESET transition is avoided and a conventional bipolar operation can be obtained [34]. Another interesting aspect of CRS is the rather gradual nature of the SET transition. The RESET at one interface

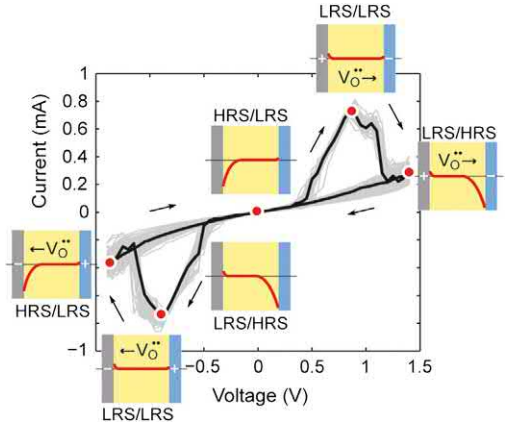


FIGURE 2.4 Illustration of the stages of complementary switching in a VCM cell. The insets show the change of the oxygen defect profile (*red solid lines*) in the oxide material (*yellow* background). The metal electrodes are shown in *gray* and *blue*. *Reproduced with permission from A. Schönhals, R. Waser, S. Menzel, V. Rana, 3-Bit read scheme for single layer Ta_2O_5ReRAM, in: 2014 14th Annual Non-Volatile Memory Technology Symposium (NVMTS), 2014, pp. 1–4 [55]. Copyright 2014, Institute of Electrical and Electronic Engineers.*

mitigates the positive feedback due to the SET transition at the opposite interface [56]. The CRS as described here occurs in a single VCM device. Usually, CRS manifests in case of a VCM symmetric stack with identical electrodes or with electrically equivalent top and bottom metal/oxide interfaces. However, CRS was first demonstrated by artificially realizing two alternative switching locations through the antiserial connection of two VCM or ECM devices [57]. Furthermore, CRS has been exploited to demonstrate up to eight distinct logic states [35,55,58].

2.2.2.3 Area-dependent switching

In principle, the redistribution of ionic defects described for bipolar filamentary switching can also occur homogenously on the complete electrode area. This fact is described for example for GaO_x [59]. To obtain reliable homogenous switching, however, any positive feedback mechanism needs to be eliminated as it would favor the formation of filaments. Thus, already the forming process should be very smooth. The reported examples for homogenous switching typically show an initially conducting film [60–63]. Instead of a hard electroforming step at high voltages, only some initialization cycles are required to obtain stable resistive switching. During switching, positive feedbacks need to be avoided, too. This can be accomplished by a proper device design. Typically, area-dependent switching systems consist of a bilayer of one conducting oxide and an oxide acting as a barrier for conduction [60–63]. The switching mechanism of area-dependent switching is

illustrated along with a typical $I-V$ characteristic in Fig. 2.5 [60]. In the shown Pt/PCMO/YSZ/SRO stack, the PCMO layer serves as a conducting oxide whereas the yttria-stabilized zirconia is the barrier oxide. In the as-deposited state, YSZ is not homogenous and not fully oxidized (Fig. 2.5A). This results in a high conductance. In the first positive sweep (Fig. 2.5A and B), oxygen ions move from the PCMO into the YSZ, which very likely homogenizes. This process is an initialization step, rather than an electro-forming process, that brings the device into a less conductive state. In the LRS, the thickness of the YSZ barrier oxide defines the resistance state. Decreasing the barrier thickness leads to a current increase. If a positive voltage is applied to the Pt/YSZ interface, an oxygen exchange reaction occurs at the YSZ/PCMO interface. Oxygen ions are extracted from PCMO and go to interstitial sites of YSZ (Fig. 2.5C). The remaining oxygen vacancies within PCMO are easily compensated with electrons of the conducting oxide. Due to the negative charging of the oxide barrier layer with O_i, the bands of the barrier bend upwards. This arrangement increases the effective barrier height for electron tunneling, and the device switches to the HRS. By reversing the polarity, the oxygen interstitials are driven back to the PCMO layer, where they recombine with the oxygen vacancies. The tunnel barrier decreases again, and the device is set to the LRS (Fig. 2.5D). The currents in

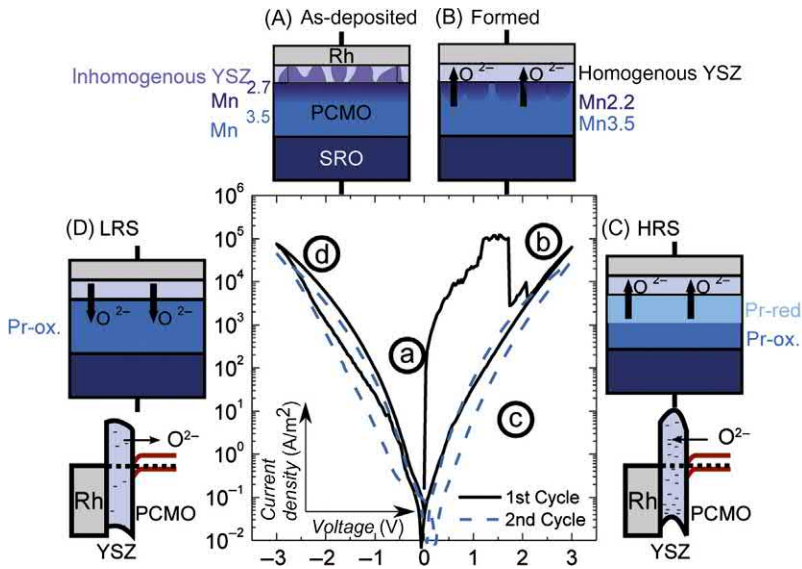


FIGURE 2.5 Sketch of the initialization process (A) and (B) and the reversible switching process (C) and (D) in an area-dependent switching Rh/YSZ/PCMO/SRO device. *Reprinted from B. Arndt, F. Borgatti, F. Offi, M. Phillips, P. Parreira, T. Meiners, et al., Spectroscopic indications of tunnel barrier charging as the switching mechanism in memristive devices, Adv. Funct. Mater.;27 (45) (2017) 1702282 under CC-BY 4.0 license.*

LRS and HRS are highly nonlinear for high-applied voltages, as the electrons tunnel through a triangular barrier. Both SET and RESET transitions are rather gradual. This fact is ascribed to two factors. First, the current density flowing in the device is low, so that Joule heating is not expected and a thermal runaway process is eliminated. In addition, oxygen ions need to be moved into/out of the tunnel oxide over the whole device area to modify the electron transport. Thus the resistance change is not too sensitive to a small change in the $V_O^{\bullet\bullet}$ configuration. Moreover, the bilayer structure may prevent an additional positive feedback effect. Indeed, during SET, the resistance of the barrier oxide decreases and the voltage divider between conduction oxide and tunnel oxide changes (at least at high voltages). This slows down the SET process. As the thickness of the tunnel oxide is fixed, the electric field stays rather constant during switching. In contrast, a thinning of the tunneling gap during SET would enhance the electric field, leading to a positive feedback loop.

2.2.3 Negative differential resistance devices

The NDR phenomenon is reported for MIM stacks employing different oxide layers, for example, VO_x [64–66], NbO_x [67,68], TaO_x [69], or TiO_x [70]. NDR can be observed in an $I-V$ characteristic measured in the current mode. It is characterized by a voltage snap-back when a certain current is reached as shown in Fig. 2.6A and B [68,71]. If the device is operated in voltage-driven mode, a hysteric threshold switching appears. In fact, the opening of the hysteresis is a measure of the magnitude of the NDR effect.

In the literature, two different mechanisms for this volatile switching process have been described: the IMT and the field-triggered thermal runaway. For VO_2 [64–66], NbO_2 [67,68], or Ti_4O_7 [70], it was shown that the electrical resistance can change due to the temperature-induced phase transition between an insulating and a metallic phase. At low temperatures, the material is in an insulating phase. At a certain temperature, the material undergoes a phase transition to a well-conducting phase, evidenced by an abrupt jump in the current. For this IMT, Joule heating is necessary. Thus a filamentary path is required to reach the desired temperature in current/voltage ranges compatible with electronic devices. When the voltage is reduced the temperature decreases, the material turns back to its insulating state.

For NbO_2-based devices, the NDR operation was initially ascribed to an IMT-based operation [72]. The NbO_2 IMT temperature, however, is very high (>1000K). Nevertheless, abrupt threshold switching was observed at comparably low currents. This discrepancy led to a second model for the threshold switching characteristic [71,73,74]. The basis of this model is a highly field-dependent current transport mechanism such as Poole–Frenkel or electric field–driven polaron hopping. The transport mechanism is based on the field-dependent lowering of an activation barrier. By increasing the

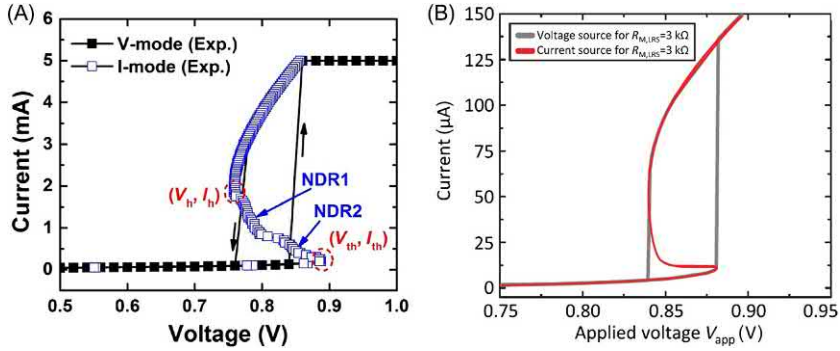


FIGURE 2.6 (A) Experimental $I-V$ curve featuring NDR in a NbO_x-based device when measured in the current-driven mode (empty squares); hysteresis is obtained in the voltage-driven mode (filled squares). (B) Simulated NDR and hysteretic behavior in an NbO_x-based device on the basis of an electric field–driven thermal runaway model. *Reprinted with permission from (A) X. Liu, S. Li, S.K. Nandi, D.K. Venkatachalam, R.G. Elliman, Threshold switching and electrical self-oscillation in niobium oxide films, J. Appl. Phys.; 120 (12) (2016) 124102; Copyright 2016, American Institute of Physics. (B) Reprinted with permission from C. Funck, S. Menzel, N. Aslam, H. Zhang, A. Hardtdegen, R. Waser, et al., Multidimensional simulation of threshold switching in NbO_2 based on an electric field triggered thermal runaway model, Adv. Electron. Mater.; 2 (7) (2016) 1600169; Copyright 2016, Wiley-VCH.*

electric field, the electronic current increases nonlinearly and Joule heating sets in. Due to the activation barrier, the electronic transport is also highly temperature-dependent. As soon as Joule heating sets in, the current increases even more, which, in turn, leads to more Joule heating. This positive feedback ends in a thermal runaway process, which characterizes the observed abrupt current jump. It was also shown that this highly nonlinear process leads to the formation of filamentary regions [75]. Moreover, it was demonstrated that the two described mechanisms for NDR can coexist in one device [76].

The volatile digital transition between a high and a LRSs is typically harnessed to design two terminal selector devices as explained in detail in Chapter 5, Selectors for resistive memory devices.

2.2.4 Switching features related to physical processes

In this section, we discuss some general switching features, which are relevant for all types of redox-based memristive switching. ECM cells, as well as, VCM cells show a highly nonlinear *switching kinetics*. If the voltage is increased by a few hundreds of millivolts, the switching time reduces by orders of magnitude. As discussed in Ref. [77], the switching time is limited by the slowest ionic process involved. The relevant processes are electrocrystallization/nucleation of a new phase, ion migration and electron-transfer reactions (oxidation/reduction) occurring at the metal/insulator interfaces. In

Fig. 2.7, the SET switching time is plotted as a function of the applied voltage on a logarithmic scale for various ECM and VCM cells, as compiled in Ref. [77]. For ECM cells, different slopes appear in this plot, each of them indicating a different limiting process. It has been shown that the SET switching kinetics of ECM cells are limited by electro-crystallization at low voltages (I), electron-transfer reactions at higher voltages (II), and finally by a combination of ion migration and electron-transfer reactions (III) [5,16]. The latter two mechanisms also determine the switching kinetics of the RESET process [78]. From Fig. 2.7A and B, three different groups of ECM cells can be identified. The fastest switching is obtained in the primary solid electrolytes, followed by secondary solid electrolytes and untypical solid electrolytes. These groups differ in their ability to host the Ag/Cu cations. As the ionic current density is proportional to the concentration of the moving ionic species, materials with a higher solubility of Ag/Cu cations switch

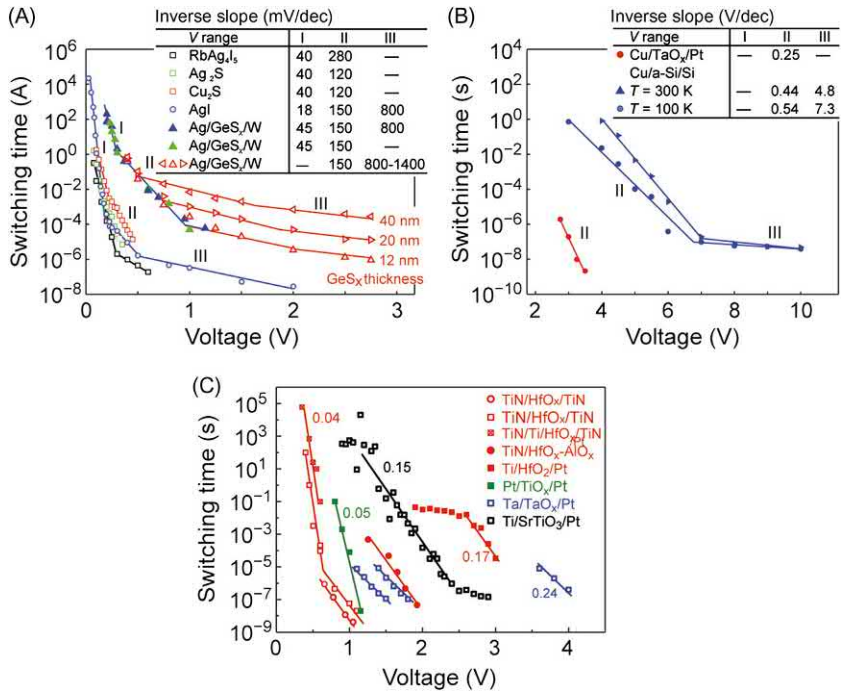


FIGURE 2.7 Switching kinetics of ECM cells showing different slopes in the voltage regimes "low" (I), "medium" (II), and "high" (III) for (A) primary solid electrolytes Ag_2S, Cu_2S, $RbAg_4I_5$, and AgI as well as secondary electrolytes Ag-GeS_x and (B) untypical solid electrolytes device Cu/TaO_x/Pt and Cu/a−Si/Si. (C) Switching kinetics data for VCM cells of hafnium oxide (*red*), titanium oxide (*green*), tantalum oxide (*blue*), and strontium titanate (*black*). The specific inverse slopes $\Delta V/\mathrm{dec}(t)$ in the $\log(t)-V$ diagram are given as numbers. *Reprinted with permission from S. Menzel, U. Böttger, M. Wimmer, M. Salinga, Physics of the switching kinetics in resistive memories, Adv. Funct. Mater.; 25 (2015) 6306−6325; Copyright 2015, Wiley-VCH.*

faster. As mentioned above, the HRS current in ECM cells is too low to enable Joule heating. Thus, the SET switching dynamics are only accelerated by the applied voltage/electric field. During RESET process, the currents are higher, but Ag and Cu are very good heat conductors. Thus Joule heating may assist the RESET process only at very high current levels ($> 100\ \mu A$).

For VCM cells, mainly one slope is observed (cf. Fig. 2.2C) for each dataset, indicating one dominating ionic process. It has been demonstrated that, in filamentary switching VCM cells, the origin of the observed nonlinearity between applied voltage and switching time (mind that the graphs report a logarithmic vertical scale) is the temperature-assisted migration of the ionic defects [79–81]. It was theoretically shown that ion hopping cannot account for a high nonlinearity in the switching kinetics if only accelerated by electric field [80]. The observed high nonlinearity can be achieved only in combination with Joule heating. For area-dependent switching devices, Joule heating effects are less relevant. As a consequence, a flatter slope and higher switching voltage are reported for area-dependent switching systems [82,83].

The switching kinetics curves reported in Fig. 2.7 can be interpreted in the following manner. Given a pulse with a certain voltage level, the minimum pulse duration sufficient to complete a switching event is the one defined by the switching kinetics curve at the corresponding voltage value. Typically, moving from this condition toward higher voltages and/or longer pulse width (strong programming conditions), more reliable switching operations are obtained. On the contrary, if lower voltages or shorted pulse widths are chosen (weak programming conditions), the switching operation can result stochastic, partial, or even absent, as it will be discussed in Section 2.4.2.

Two universal switching laws of redox-based RRAM devices are direct consequences of the nonlinear switching kinetics [79,84,85]. First, the programmed resistance R during SET scales linearly with the current compliance as shown in Fig. 2.8A [42]. When the current compliance sets in, the voltage drop on the RRAM device reduces, if its resistance decreases further. According to Fig. 2.7, the decrease of the voltage leads to an exponential reduction in switching time, for example, by reducing the drift velocity of the ionic defects. Thus the switching process virtually stops in the timescale of the experiment.

The second universal law is that the maximum RESET current scales linearly with the programming SET current, as shown in Fig. 2.8B [42]. The origin of the second universal law is linked to the nonlinear switching kinetics too. The voltage magnitude at which the SET transition stops is the one at which the RESET transitions starts. At this voltage, the driving force is high enough to trigger the ionic processes, and, thus, the RESET process itself. The universal RESET law holds as long as the symmetry/asymmetry of the current with respect to the voltage polarity is constant over all

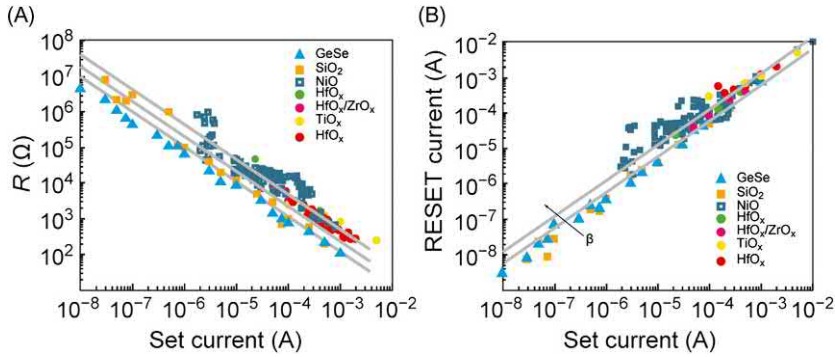


FIGURE 2.8 Universal SET (A) and RESET characteristics (B) of redox-based memristive devices. Experimental data are displayed for a Ag:GeSe ECM-system, a Cu:SiO_2 ECM-system, a NiO TCM-system, a HfO_x VCM-system (*green dots*), a bilayer HfO_x/ZrO_x VCM-system, a TiO_x VCM-system and a HfO_x VCM-system (*red dots*). *Reprinted with permission from R. Waser, R. Bruchhaus, S. Menzel., Redox-based resistive switching memories, 2012; Copyright 2012, Wiley-VCH.*

resistance regimes. If the symmetry/asymmetry changes, a deviation from this law appears [84].

The above observations may be interpreted in a phenomenological fashion. In particular, the LRS resistance scales either with CF size or with the extension of a tunneling gap establishing between the CF apex and the metal electrode, in different resistance ranges, respectively. Indeed, these two quantities can be controlled through the SET current.

An additional general feature differentiating filamentary and area-dependent RRAM devices is variability. Indeed, filamentary devices are affected by large cycle-to-cycle and device-to-device variability, because the resistance of the devices is dominated by the atomic configuration of a few defects defining the CF [86–89]. A minimal different alignment of defects results in a large resistance variation. On the contrary, for area-dependent switching, the entire cross section of the RRAM devices is involved in the current flow. Indeed, it can be imagined that the effects of the change in atomic configuration are averaged out over the entire device area. The variability in the resistance state results in a variability in the transition voltages and currents and vice versa. In addition to such variability related to the programming operation, reading noise is present in filamentary devices, in particular random telegraph noise (RTN). RTN consists of random current fluctuations measured at low reading currents. In nanoscale devices, the trap and the release of electronic charges by few defect sites affect the values of the flowing current due to Coulomb shielding. This is relevant for filamentary devices, in which defects surrounding the CF can trap and de-trap charges. This partially inhibits and restores the current flowing through the CF. The conduction can be dramatically suppressed in

very thin CF, that is, for very high LRS resistances obtained for low current compliances [86–89].

2.3 Resistance switching technology: performances and industrial-level prototypes

The main historical application of RRAM devices is in the nonvolatile memory field. In this section, we define the performance characteristics of RRAM devices that are relevant for memory operations. Therefore, we consider only the switching phenomena leading to nonvolatile resistance transitions and corresponding devices. The major application of volatile switches, that is, two terminal selector devices, are discussed in Chapter 5, Selectors for resistive memory devices. In this section, we describe the correlation between the physics of the switching of the classes of devices defined in Section 2.2 and their expected performance figures of merit and possible trade-offs among them. We report the best results in the literature with particular reference to array-level or industrial-level demonstrations and products.

Generally speaking, a computing architecture is composed by a memory hierarchy ranging from the two extreme cases of computing (SRAM and DRAM) and storage memories (Solid State Disk and Hard Disk Driver) [90], as illustrated in Fig. 2.9. Computing memory is a highly volatile ($\sim$1–10 ns retention), high speed ($\sim$10–100 ns access time), relatively large (6–100F^2, where F is the feature size), and expensive memory ($\sim$10–1000 \$/GB), with virtually unlimited endurance and extremely low energy ($\sim$10 fJ per switching operation). Computing memory is in close connection with the computing processing unit. Storage memories are nonvolatile ($\sim$10 years retention times), compact (6F^2), cheap ($\sim$0.1–1 \$/GB), relatively low speed ($\sim$0.1–1 ms latency), low endurance ($\sim 10^6$ cycles),

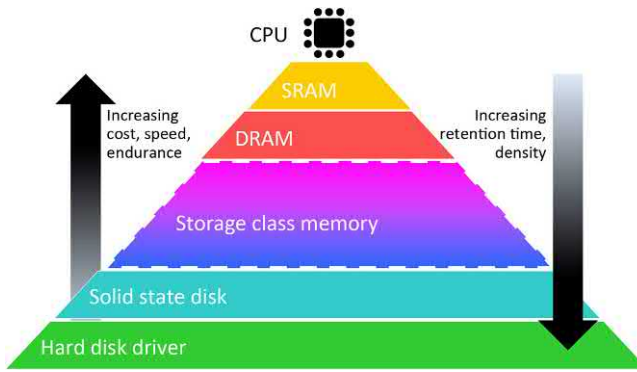


FIGURE 2.9 Memory hierarchy in a standard computing architecture. The storage class memory is the missing element in nowadays architecture.

and high power (100 pJ−10 nJ) memories, which are connected to the processor through a bus. A latency gap opens between the extreme cases of (computing) memory and storage, which limits the efficiency of nowadays computing. Some solutions are now trying to fill such a gap by harnessing the versatility of RRAMs and other resistive memory technologies. For instance, the 3D XPoint technology by Intel and Micron, which is supposed to be composed of phase change elements, is already at the production level [91]. The memory chip aimed at filling the latency gap are known as storage class memories (SCMs). In general, in the development of emerging devices, the performance parameters must be optimized, depending on the addressed level within the memory hierarchy (Fig. 2.9). In particular, for storage devices, *retention* and *scalability* are more important than *endurance*, read/write *speed*, and extremely small *energy consumption*, which, on the contrary, are crucial for computing memory. Conversely, resistance state *variability* and programming *variability* must be contained whichever level of the memory hierarchy is targeted. In addition, *multibit operation* of a single device constitutes an alternative way to improve the information density and, therefore, is an opportunity for storage devices.

Filamentary devices, cationic or anionic, unipolar or bipolar, enable scalability down to the nanometer scale. Single devices down to $10 \times 10\,\text{nm}^2$ [92] and $1 \times 3\,\text{nm}^2$ [93] have been produced and successfully operated. Also at array-level, aggressive memory chip scaling has been demonstrated [94−96].

The great issue of filamentary devices lies in their variability. Indeed, LRS and HRS resistances show a large cycle-to-cycle and device-to-device variability. The resistance distributions may partially overlap over large device arrays, thus producing error bits. This issue becomes more problematic if the average resistance window is small. Program and verify techniques, which slow down the bit writing, can mitigate the variability issue [97−100]. The variability is due to the inherently stochastic processes of CF formation and dissolution. The smaller the CF, the more the randomness of the involved processes becomes evident [101,102]. It is immediate that increasing the compliance current, that is, increasing the CF size or the defect density, reduces the ON state variability and enlarges the resistance window, but, on the other side, it dramatically raises the energy consumption [101−103]. Therefore, a general compromise between performances and reliability has to be found [104]. Material engineering has been widely explored to mitigate this compromise. For instance, for both ECM and bipolar filamentary VCM devices, uniformity can be improved in two major manners. The engineering of defects inside the switching layer of RRAMs through doping improves uniformity, because doping is supposed to establish fixed preferential paths for CF formation [95,105−108]. Alternatively, interstitial layers at the metal/oxide interface act as nonswitching barriers for conduction and reduce the impact of variability and RTN [105,107]. Interestingly,

the use of a HfO_2/Al_2O_3 bilayer for uniformity and endurance improvement to 10^6 cycles has been recently verified on a 16 kb 1T-1RRAM array [109].

As already stated, a large resistance window can accommodate resistance drift under zero bias. Thus the state retention is supposed to be longer. In other cases, a trade-off has been evidenced between retention time and resistance window [110–112]. As already pointed out, the amplitude of the resistance window scales with the energy consumption. Enlarging the resistance window, however, has the drawback of degrading the endurance [110,113]. This fact can be naively interpreted as follows: a high resistance window requires the migration of a large amount of defect or a migration over relatively long distances and a correspondingly large voltage/thermal stress during programming. In this manner, the device switching efficiency is degraded more rapidly than in a device with a lower resistance window. However, endurance was extended to 10^{10} cycles by Chen et al. [114] by a careful balancing of the SET/RESET pulse conditions which also increase the resistance window in single Hf/HfO_2-based 1 T1R devices. An impressive endurance record up to 10^{12} cycles has been measured on single filamentary VCM devices based on Ta oxides bilayers with a resistance window of one order of magnitude. A collection of data from the literature in Ref. [113] shows a clear inverse proportionality between resistance window and maximum number of achievable cycles in ECM devices.

Switching speed in filamentary devices can be as fast as few hundreds of picosecond provided a suitably high voltage is applied [115], according to the nonlinear switching kinetics [77]. This high voltage, however, might be incompatible with ultra-scaled CMOS technology nodes [90]. As stated earlier, however, the latency of a memory is not governed only by the device switching times but also by the correction codes in programming and reading operations, which must be related to device variability.

For what concerns very large scale integration, some aspects must be considered. Filamentary devices require an individual electroforming after fabrication and a compliance current during SET operation. Thus 1 transistor—1 ReRAM (1T-1R) circuits, as memory unit element, are typically used. In this circuit, the transistor acts also as selector of individual devices in the array. In view of ultimate scaling and possible 3D integration, alternative two-electrode back-end compatible selectors have been proposed, for example, as threshold switches and nonlinear elements as bidirectional diodes [90]. Selector devices will be dealt with in Chapter 5, Selectors for resistive memory devices. From the integration point of view, forming free and self-compliant single VCM devices would be beneficial. Such devices, however, show often high device-to-device variability, LRS resistance variability, and power consumption [56,112,116,117].

For what concerns area-dependent devices, the trade-offs between performance parameters have not been clarified in as much details as for filamentary ones. In contrast to filamentary devices, area-dependent devices are

characterized by better uniformity [118] and reduced RTN [119], because the electric conduction involves the entire cross section of the devices [120,121]. In comparison to filamentary devices, therefore, low power operation can be reliably achieved in combination with relatively low resistance windows of the order of 10 without sacrificing the uniformity, as in the case of $TiN/Al_2O_3/TiO_2/TiN$ [61] and $Pt/Ta_2O_5/HfO_{2-x}/Ti$ devices [120]. Endurance usually reaches values of 10^5-10^6 cycles [122–124], but latest works report cycling up to 10^{12} in TaO_x/TiO_2 devices [125]. Retention times do not reach values as high as for filamentary devices [119,124,126]. Area-dependent devices do not require electroforming and neither current limitation during SET, which constitutes a huge technological advantage over filamentary devices. Interestingly, area-dependent devices usually show nonlinear $I-V$ curves [124,125,127], which can be exploited for the self-selection of memory element in passive crossbar arrays as in a prototype 64 Mb array fabricated in the 130 nm node [128]. One obstacle for the industrialization of area-dependent resistive memory is the extensive use of nonfab-friendly materials, such as $Pr_{1-x}Ca_xMnO_3$, $SrTiO_3$ switching media, and Pt electrodes.

The just described trade-offs among device performance parameters evidence the complexity, as well as, the opportunities of RRAM devices, which indeed has already seen some industrial-level prototypes and (few) commercialization, for example, the 4 Mbit array mounted in a microcomputer for wearable devices by Panasonic and Fujitsu [129,130]; the ECM-based chip by Adesto designed for Internet-of-Things applications [131]; the 16 Gbit RRAM chip in 27 nm technology node designed by Sony/Micron for FLASH replacement [132]; or the various RRAM solutions for stand-alone and System-on-Chip applications by Crossbar Inc. [133]. As demonstrated by these latter examples, in the field of conventional memory devices, RRAM are likely to play a role as storage elements mainly in system-on-chip devices for Internet-of-Things applications. In any case, still recently, RRAM devices have been considered for FLASH replacement in stand-alone memory devices [120] and for SCM applications [125], also according to SanDisk [134] and as recently announced by Western Digital [135].

2.4 Advanced functionalities and programming schemes

The classification of resistive switches into the category of *memristive devices* has been driving the research toward the investigation of novel or advanced functionalities that could make those devices a key-enabling technology for in-memory, deep learning, or brain-inspired computing schemes. Indeed, such advanced functionalities had not been investigated with such a strength for memory applications. For instance, neural accelerators applications renewed the interest toward multilevel operation. In the recent literature, many studies demonstrated that RRAMs can actively implement some

computational tasks locally and collectively. For instance, RRAM devices display a plastic change of resistance, that is, an integrative response to sequences of voltage spikes and/or sensitive to their relative timing. In addition, the phenomenon of negative differential resistance has been recently spotlighted for the realization of oscillators in view of the emulation of collective neural dynamics. Sections 2.4.1–2.4.4 deal with all these novel or advanced functionalities on a device level and in relation with the physics of the involved switching mechanisms. System-level studies will be examined in depth in Chapters 6–19.

2.4.1 Multilevel operation

As described in Section 2.2, the resistance of nonvolatile RRAM devices depends in a phenomenological point of view either on the width or on the conductivity of a CF or on the height or the width of a metal/insulator barrier for electronic conduction. These quantities can be modulated to some extent by tuning the driving voltage or the compliance current during SET operations, giving rise to multiple resistance states.

In filamentary devices (either VCM and ECM type), the formation of a CF, which occurs in an abrupt way, can be interrupted deliberately at precise resistance values through an external current limitation during SET (cf. Fig. 2.10A [136]). In VCM devices, the dissolution of the CF occurs gradually. In this case, the maximum applied voltage, as well as the time interval of the voltage stimulation, governs the HRS resistance value (see the tuning of the RESET operation for negative voltages in Fig. 2.10A [136]). Conversely, in ECM devices, the dissolution of the CF usually occurs in an abrupt fashion, which prevents the programming of more than two resistance states with a RESET operation. Generally, in area-dependent RRAM devices, both SET and RESET processes occur gradually as a function of the applied ramped voltage, and they can be both parceled out through the modulation of the maximum applied voltage or the time interval of voltage application.

From the conventional memory application point of view, *resistance bits*, or *resistance levels*, must be precisely distinguishable. An overlap in the distributions of the corresponding resistance values due to cycle-to-cycle, device-to-device variability, reading noise or retention degradation would result in a read failure. As a consequence, multibit operation can be conveniently operated in devices that display high resistance window, low cycle-to-cycle, device-to-device variability, and low reading noise (including RTN).

For filamentary devices, multilevel operation requires a hard compromise between power consumption, variability, and resistance window. Indeed, to enlarge the resistance window, low LRS resistances, and consequently high switching energies must be accepted. As an example, Figs. 2.10B and C report the cumulative distributions of resistance levels obtained from 16 kb

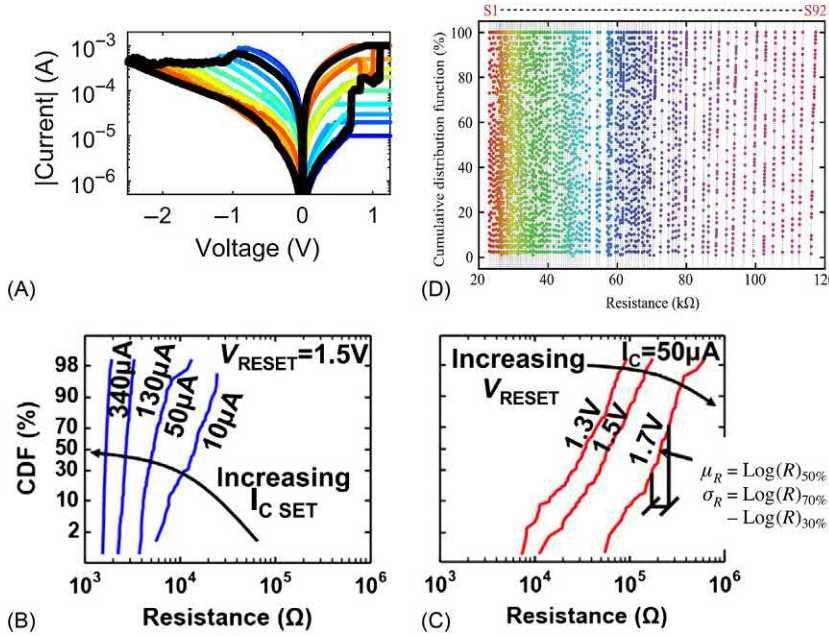


FIGURE 2.10 (A) Representative multilevel programming of a filamentary HfO_2-based memristor obtained by limiting the current during the SET operation at progressively increasing values (positive voltages) and by tuning the RESET stop voltage (negative voltages). Representative distributions of resistance values belonging to distinct memory states programmed through the tuning of the compliance current (B) and RESET stop voltage (C) in a 16-kbit array of 1T-1 HfO_2-based RRAM. (D) Resistance cumulative distribution of 92 levels obtained through a program and verify technique on a single TiO_2-based device. *Rearranged with permission from (A) S. Brivio, S. Spiga., Stochastic circuit breaker network model for bipolar resistance switching memories, J. Comput. Electron.; 16 (4) (2017) 1154−1166; Copyright 2018, Springer. (B and C) Reprinted with permission from D. Garbin, E. Vianello, Q. Rafhay, M. Azzaz, P. Candelier, B. DeSalvo, et al., Resistive memory variability: a simplified trap-assisted tunneling model, Solid-State Electron.; 115 (2016) 126−132; Copyright 2016, Elsevier. (D) Reprinted under CC-BY license from S. Stathopoulos, A. Khiat, M. Trapatseli, S. Cortese, A. Serb, I. Valov, et al., Multibit memory operation of metal-oxide bi-layer memristors, Sci. Rep.; 7 (1) (2017) 17532.*

filamentary RRAM arrays by tuning the current compliance [panel (B)] and the reset stop voltage [panel (C)] [103]. Only for relatively high compliance currents (130−340 μA), the overlap between the distributions is definitely reduced. As said, the scientific results are rapidly making progress, though. Stathoupoulos et al. [137] were able to demonstrate 92 levels in a single TiO_2-based device within a resistance window as small as roughly 60. The device is probably filamentary because an initial forming step is required. The resistance level distributions reported in Fig. 2.10D were obtained from consecutive reading over a period of 8 hours after programming, demonstrating impressive low noise features. The result was obtained by a program and

verify algorithm, which is usually adopted to overcome the variability of filamentary devices [97–100]. An interesting result on a small scale array is reported by Li et al. [138] who demonstrated 64 levels (6 bits) in a 128 × 64 array of 1T-1R devices with a Ta/HfO_2/Pd structure and an available resistance window slightly lower than a factor 10.

Area-dependent devices suffer less than filamentary ones from intrinsic variability and RTN. Furthermore, the power consumption can be reduced without severe constraints on the resistance window, as they can achieve very high OFF resistances [120]. Both features facilitate the multilevel operation of area-dependent devices.

The shift of the target application from memory to alternative computing paradigms allows a certain re-thinking of the device specifications, although they have not been identified with precision yet. Device unreliability, for example, can be partially tolerated in neural systems or for stochastic or approximate computing schemes. Furthermore, for memory applications, the resistance of each individual device must be read with high precision in a short time, which poses a limit to the minimum current to be acquired and, hence, to the maximum device OFF resistance. This limitation, together with the need of high ON resistances for power economy, limits the resistance window and hinders a multilevel operation. In contrast, in neural architectures working in real time, the current from one line of devices is usually summed up and integrated over millisecond timescales, which enables extremely high HRS [139]. On the downside, in-memory computation, and computational schemes relying on vector-matrix multiplication usually require the parallel programming of several devices, which may prevent verifying the successful programming of individual cells and applying error correction codes [140].

2.4.2 Implementation of plasticity in resistive switching random access memories devices

Plasticity is the functionality of biological synapses that has been demonstrated in RRAM devices, feeding the expectations toward spiking neural networks comprising memristive synapses (Chapters 15–19). Plasticity is considered the property of the device to change its own resistance as a function of time in response to a repeated stimulation. Biological plasticity can be emulated in RRAM devices by two ideally different operations. The first one corresponds to the *analog* operation, which is intended to be the *progressive or gradual change of resistance* over a continuum of resistance values. It is driven by sequences of identical spikes and results from the switching dynamics of the devices. The second operation exploits the device *stochasticity*, that is, the nondeterministic, binary switching between two resistance states. It is worth pointing out that there is an overlap between analog and stochastic operation as they both rely on the change of the atomic

configuration. Nevertheless, the two following subsections are dedicated to these two ideally distinct programming operations.

2.4.2.1 Plasticity by analog switching dynamics

An example of analog plasticity is reported in Fig. 2.11. Specifically, Figs. 2.11A and B show the current and conductance increase and decrease as a function of trains of identical positive and negative pulses, respectively [141,142]. SET and RESET are usually called potentiation and depression operations for this synaptic electronic application. The resistance evolution as reported in Fig. 2.11A and B is intended as plasticity because the *i*th spike brings the device to a resistance value (R_i) that is a function of the previous resistance value, that is, $R_i = f(R_{i-1})$. In plastic operation, the history of the stimulation of the device determines its final resistance value. In contrast, in standard memory operation, one pulse is desired to bring the device into a

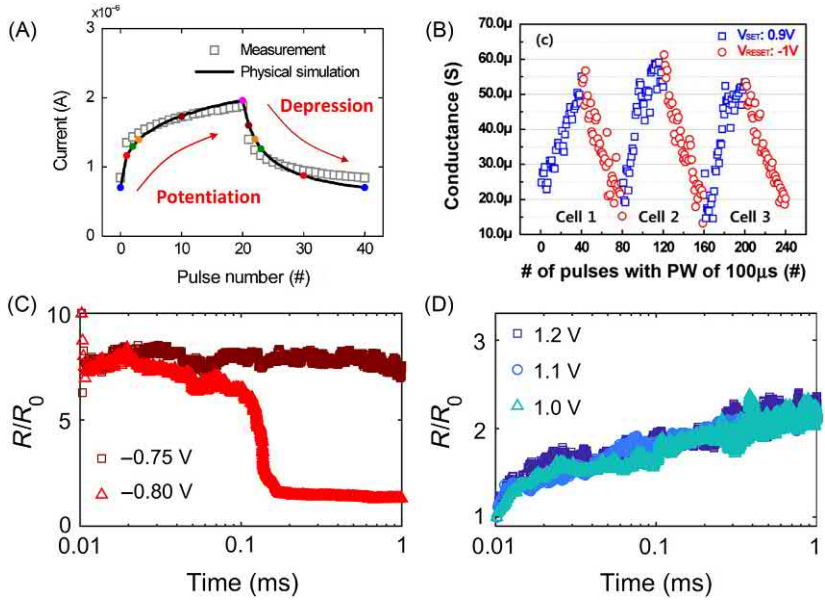


FIGURE 2.11 (A) Analog current dynamics of a TaO_x/TiO_2-based filamentary memristor stimulated by train of identical pulses for potentiation and depression (right panel). (B) Three potentiation and depression cycles performed through trains of identical pulses for a device showing linear and symmetric evolution [142]. Resistance read as a function of time while applying potentiation pulses [− 0.75 and −0.8 V, panel (C)] and depression pulses [1.0−1.2 V, panel (D)]. *(A) Reprinted under CC-BY license from Y.-F. Wang, Y.-C. Lin, I.-T. Wang, T.-P. Lin, T.-H. Hou, Characterization and modeling of nonfilamentary Ta/TaOx/TiO2/Ti analog synaptic device, Sci. Rep.; 5 (2015). (B)−(D) Adapted with permission from E. Covi, S. Brivio, J. Frascaroli, M. Fanciulli, S. Spiga, (Invited) Analog HfO2-RRAM switches for neural networks, ECS Trans.; 75 (32) (2017) 85−94; Copyright 2017.*

precise resistance value independently from the previous programming history.

The modulation of the conductance of a device through train of identical pulses is the result of its switching dynamics. Figs. 2.11C and D show the influence of the applied voltage on the dynamics of HfO_2-based device potentiation and depression, respectively [143]. In the figures, the current is monitored while applying 1-ms-long voltage pulses with varying amplitude. During depression (Fig. 2.11D), the current shows a gradual evolution for all applied voltages with characteristic times of fraction of milliseconds. In contrast, a gradual evolution for potentiation is only observed at voltages lower than −0.8 V (Fig. 2.11C). At −0.8 V, the device first switches gradually but then jumps abruptly to the LRS (cf. Section 2.2). The resulting total resistance change of one order of magnitude is close to the maximum resistance window of this device [143]. Therefore, a −0.8-V-high and 1-ms-long pulse corresponds to a *strong programming* condition in this case, according to the definition given in relation to the RRAM switching kinetics in Section 2.2.4. In contrast, changing the resistance by only a fraction of the maximum resistance window can be considered *weak programming* condition. Only under weak programming conditions, a plastic device response is achieved, as if the switching dynamics in Figs. 2.11C and D is sampled through short pulses.

In such reasoning, we are considering that the delivery of a train of N identical pulses with pulse width Δt_w and voltage V_p gives the same effect as one single pulse with the same voltage V_p and pulse width of $N\Delta t_w$. This approximation appears reasonable in some devices and under specific programming conditions [143], even though recent studies have proven the existence of second-order effects, according to which the resistance evolution depends on the spike rate and spike timing [144–146]. Section 2.4.3 is dedicated to the description of these second-order effects.

Analog plastic operation has been first demonstrated in filamentary VCM devices only in the RESET operation, for example, in HfO_2/TiO_x-based devices [47,147]. In principle, the SET process of filamentary devices is hardly parceled out into many steps, because of the positive feedback (see Section 2.2) that drives sharp transitions. However, some authors demonstrated analog resistance evolution in both SET and RESET operations in HfO_2-based RRAM devices with interfacial defective interlayers at the HfO_2–TiN interface [148–151]. Such interlayers may introduce a CRS operation that mitigates the positive feedback in the CF formation process (see Section 2.2) [149]. This mechanism leads to smooth resistance dynamics during SET operation similar to the one in Fig. 2.11C [143]. In other cases, an additional layer is used as a tunable series resistance, again mitigating the positive feedback [142]. Investigated materials comprise standard filamentary switching oxides such as HfO_2 and TaO_x in combination with interfacial or additional layers made of TiO_x, AlO_x, and

TaO_x [141,142,148–152]. Very recently, analog plasticity has been demonstrated also in 1T-1R configuration [153].

Area-dependent devices are expected to outperform filamentary ones in the analog plasticity, because they usually display gradual SET and RESET transitions when programmed through voltage ramps. Furthermore, the typical uniformity of area-dependent devices helps in defining sharp distributions of multiple resistance levels. The main oxides studied in the recent literature and resulting in an area-dependent analog switching comprise TaO_x, TiO_x, Al_2O_3, SiO_x, $Pr_{1-x}Ca_xMnO_3$ [137,141,152,154–156]. Noteworthy is the use of amorphous silicon, which results in extremely high resistance range and low power consumption, at the cost of relatively high programming voltages [126].

In recent times, the research has concentrated on optimizing the resistance evolution toward two aspects: linear resistance update ($\Delta R = R_i - R_{i-1} = const.$) and symmetry of the resistance evolution for SET and RESET processes. These properties are considered critical features for the learning performances of neural systems, especially those employing back-propagation algorithms (Chapters 12–14) [140,142,157,158]. Moved by these results, many works appeared in the literature proposing the engineering of material combinations and programming conditions in view of improved linearity and symmetry (e.g., see Fig. 2.11B) often at the further expense of resistance window and switching uniformity [142,148,152,155,156,159]. The achievement of linear and symmetric dynamics in SET and RESET transitions is expected to be tough in filamentary VCM devices, as SET and RESET are self-accelerated and self-decelerated processes, respectively [47,149]. It has been recently demonstrated, however, that in bio-inspired systems with unsupervised or semi-supervised learning schemes, nonlinear and asymmetric synaptic plasticity characteristics improve the network performances [160,161].

2.4.2.2 *Plasticity by stochastic switching*

Stochastic switching is a prerogative of filamentary devices, rather than area-dependent ones. Indeed, CF formation and disruption is driven by stochastic processes at the nanoscale, for example, defect generation or hopping from site to site, each of which individually produces a macroscopic resistance change. Therefore, the resistance transition in an ideal binary device, displaying only two levels with well separated distributions, occurs with a certain probability when programming with weak pulses, according to the definition given in Section 2.2.4 in relation to the RRAM switching kinetics. In several recent works [47,162–164], it has been proposed to exploit this kind of operation for a different implementation of plasticity for neural applications. Indeed, a train of identical weak programming pulses drive the resistance change after a certain number of pulses on average. Thus a history-dependent

and plastic operation on average is realized. An example of nondeterministic switching of a Pt/HfO_x/TiO_x/HfO_x/TiO_x/TiN device is reported in Fig. 2.12. Fig. 2.12A shows that different trials display the resistance transition from HRS to LRS after a different pulse number. An endurance test at weak programming conditions allows determining the average SET switching probability as shown in Fig. 2.12B [165]. In the reported case, the authors adopted a HfO_x/TiO_x/HfO_x/TiO_x multilayer device to increase the resistance window. In this manner, they found stochastic SET transitions between almost ideally binary states and gradual analog RESET dynamics [165]. Endurance tests performed on 1T-1R vertical TiN/HfO_2/Ti/TiN RRAMs revealed that a correlation exists among the HRS and LRS values over tens of cycles and, thus, that the variability in the resistance states is not purely stochastic in the investigated system [167,168]. An alternative manner to analyze the stochastic device operation is to monitor the time required by a specific voltage value to

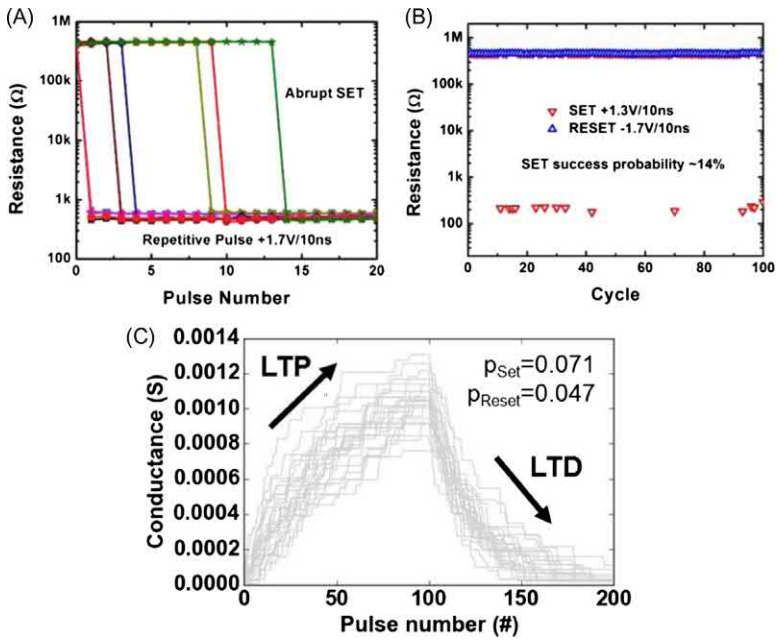


FIGURE 2.12 (A) Example of stochastic operation of a binary RRAM under stimulation with trains of identical weak pulses; (B) result of an endurance test with weak SET pulses. (C) Example of analog operation obtained with a parallel of 20 binary devices programmed in a stochastic manner. *(A and B) Reprinted under the CC-BY license from S. Yu, B. Gao, Z. Fang, H. Yu, J. Kang, H-SP, Wong, Stochastic learning in oxide binary synaptic device for neuromorphic computing, Front. Neurosci.; 7 (2013) 186. (C) Reprinted under the CC-BY license from T. Werner, E. Vianello, O. Bichler, D. Garbin, D. Cattaert, B. Yvert, et al., Spiking neural networks based on OxRAM synapses for real-time unsupervised spike sorting, Front. Neurosci.; 10 (2016).*

drive the resistance transition. With this methodology, Gaba et al. found that wait times collected in an ensemble of programming trials follow a Poisson distribution in Ag/(amorphous Si)/Si devices [164], which indicates that the switching events are independent and completely uncorrelated, differently from Ref. [167].

The parallel arrangement of binary devices programmed in a stochastic way has been proposed by some authors in order to endow the overall resulting circuit with the ability to go over many conductance values (given by the total resistance of all the device resistances in parallel, see Fig. 2.12C [166]) [162,163,166]. Bill and Legenstein [169] named this solution *compound synapse*. Obviously, the acquired property of gradual synaptic weight change comes at the expenses of the increase of the area occupancy.

2.4.2.3 Implementation of plasticity: assessment and practical issues

In the previous sections, we described two paths for emulating plasticity. Both are realized by weak programming pulses. The employment of weak programming conditions leads to a number of technological issues. The first one concerns the width of the *programming window*. For instance, Frascaroli et al. [148]. show that a limited voltage window of few hundred of millivolts is available for reliable analog operation in filamentary HfO_2 devices for a wide range of pulse widths [45]. A narrow programming window is available also for the stochastic operations, as can be appreciated, for example, in works by Suri et al. [170] dealing with ECM devices and by Yu et al. dealing with filamentary VCM devices [165]. Especially for VCM devices, weak programing conditions only exploit a small part of the *resistance window*, which, in turn, can be achieved in the same device with strong programming conditions. In particular, for analog devices, a trade-off has been identified between the analog character, the number of levels (intended in a relaxed manner), and the resistance window [148,159]. The decrease of the resistance window with decreasing the strength of the programming conditions has been verified also by Garbin et al. for the stochastic operation of HfO_2 devices [103] and by Nishi in Ta_2O_5 devices [168]. A limited resistance window poses severe issues in case of highly scaled technologies, in which crossbar lines assume high resistance values that can prevent the sensing of the small resistance variation of the RRAM devices. As a matter of fact, the most gradual resistance evolutions are recorded for a maximum resistance window in the range of factor 2−5 [148,154]. Only few works report of gradual resistance change over 1 order of magnitude for RESET [47] and SET operations [149,151]. In addition, the strength of the programming condition (weak/strong) may affect the data retention. This fact is rarely investigated in the literature in relation to analog plasticity. Interestingly, Zhao et al. [171]. monitored the resistance levels of a 1-kb 1T-1R array employing

Al-doped HfO_2 devices programmed with trains of identical pulses at elevated temperatures for hundreds of seconds, proving a reasonable retention. On the other side, the use of weak programming conditions are expected to bring the advantage of reducing the *power consumption* [47,172] and of extending the *device endurance* [164]. A quantitative evidence of these facts, however, has not been reported so far to the best of authors' knowledge.

Apart from these general considerations about the use of weak programming conditions, a metric for the assessment of the performance of the analog operation, as well as, for the stochastic operation of RRAM devices has not been established yet. As mentioned above, analog devices have been optimized either toward achieving *linear* and, sometimes, *symmetric dynamics* at the expenses of resistance window [142,152] or toward maximum *resolutions* (number of levels) at the expenses of linearity [154]. In particular, some reports claim for high number of levels with distinct distributions. For instance, Park et al. demonstrated the repeatability of 64 distinct resistance levels programmed with trains of identical pulses upon 30 cycles in an area-dependent $Mo/MoO_x/TiO_x/TiN$ device [154]. However, a clear and well-established definition of the concept of "levels" in an analog device does not exist. It must be mentioned that the analog dynamics are affected by large variability and by RTN, which both limit the ultimate resolution of the devices. Variability is an additional direct consequence of weak programming conditions. RTN is intrinsic to the device reading operation. Furthermore, it has been recently shown that weak programming pulses can stimulate random resistance variation around a constant resistance value [89].

For what concerns stochastic programming operations, an explicit comparison of the device performances is lacking in the literature. It is expected, though, that high precision neural networks rely on stochastic memristive devices with switching probabilities as low as 0.001 [169]. The switching probabilities reported in the literature, however, only get close to values as low as 0.1 [164,165,170].

2.4.3 Rate and timing computing with resistive switching random access memories devices

The previous sections deal with implementations for storing memory bits and for keeping track of occurring events (plasticity). Advanced functionality is investigated to dress the RRAM devices with the concepts of local computing. One example is the local coding of the rate and of the timing properties of the occurring events that stimulate the RRAM device itself. The processing of the incoming events based on their rate and timing is believed to be a fundamental cognitive function and, therefore, has important applications in bio-inspired neuromorphic computing (Chapters 15–19). Among such cognitive functions, in the recent literature, a primary role is played by

spike-timing-dependent plasticity (STDP). In STDP, the timing occurrence of spikes on two neurons (pre- and postneurons) leads to a weight change of the synapse connecting the neurons [173]. If the preneuron contributes to the firing of the postneuron, on average the preneuron fires always before the postneuron. If the postneuron firing is completely independent of the preneurons, their firing is uncorrelated in time. STDP prescribes a weight increase if the synapse is stimulated by the preneuron first and the postneuron afterwards, within a certain time window. A weight decrease occurs in the opposite case. Therefore, STDP is a realization of local computing with timing, and it has been demonstrated to be also sensitive on the spike rate in biological synapses [173,174]. In the following, we describe the recently proposed device-level STDP implementations. A thorough description learning systems based on STDP and involving memory devices can be found in Chapter 17, Synaptic realizations based on memristive devices, and Chapter 18, Neuromorphic coprocessors and experimental demonstrations. In several literature reports, STDP has been implemented in plastic RRAM devices, either stochastic [162,163,165,175] or analog [47,143,172,176,177], by delivering long pseudo-triangular-shaped overlapping pulses to the two device terminals, as shown in the top panel of Fig. 2.13A [141]. The overlapping of such shaped pulses results in voltage drop on the device that depends on the relative pulse arrival timing and realizes a spike timing−dependent resistance change, see bottom panel of Fig. 2.13A [141]. In the stochastic version of STDP, the probability of switching assumes increasing finite values with increasing the delay-time window [162,163,165,175]. Such pairwise STDP implementation is based on a clever programming of plasticity. On the downside, such implementations require programming pulses as long as the timescales which STDP is designed to be sensitive to. Furthermore, it does not take into account pulse rates because it relies only on spike couples [161]. On the other hand, pioneering works have demonstrated that advanced cognitive functions are intrinsic to single devices, as we will describe in the following.

To explain the effect of the rate or of the timing of programming pulses delivered to a RRAM device, let us first consider that a programming voltage spike, when isolated from previous or successive spikes, is able to drive a certain resistance change ΔR_{alone}. It has been experimentally demonstrated that a preliminary conditional spike can prepare the ground for a subsequent programming spike so that the resistance change driven by the paired pulses (ΔR_{pair}) is larger than the one of a single pulse, ΔR_{alone}. This effect is named pulse pair facilitation (PPF). In general, for this effect to take place, the conditional pulse and the programming pulse must be at a minimum time separation. The closer they are, the larger the difference $\Delta R_{\text{pair}} - \Delta R_{\text{alone}}$. This effect realizes a resistance change that depends on the spike timing or equivalently on the spike rate. As said, some pioneering works have experimentally observed PPF and attributed them to diverse physical mechanisms,

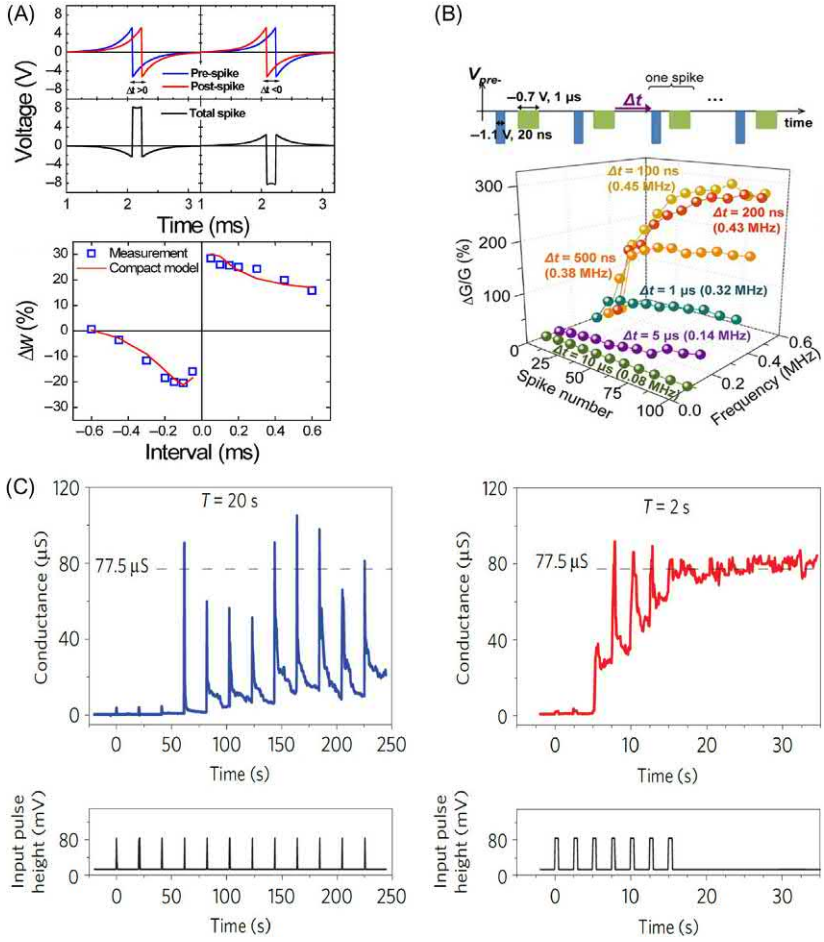


FIGURE 2.13 (A) STDP obtained through temporal overlapping of couple of pulses as in the top panel, the weight change as a function of the delay among pulses is shown in the bottom panel. (B) PPF obtained with preheating low voltage and a programming high voltage spikes as reported in the top panel: the closer the two spikes, the more and the faster the conductance changes as shown in the bottom panel. (C) Short-term memory (STM) driven by low frequency spike train (left panel) and gradual transition from STM to long-term memory for high frequency spike train (right panel). *(A) Reprinted under the CC-BY license (A) Y.-F. Wang, Y.-C. Lin, I.-T. Wang, T.-P. Lin, T-H. Hou, Characterization and modeling of nonfilamentary Ta/TaOx/TiO2/Ti analog synaptic device, Sci. Rep.; 5 (2015). (B) Reprinted (adapted) with permission from S. Kim, C. Du, P. Sheridan, W. Ma, S. Choi, W.D. Lu, Experimental demonstration of a second-order memristor and its ability to biorealistically implement synaptic plasticity, Nano Lett.; 15 (3) (2015) 2203−2211; Copyright 2015, American Chemical Society. (C) Reprinted with permission from T. Ohno, T. Hasegawa, T. Tsuruoka, K. Terabe, J.K. Gimzewski, M. Aono, Short-term plasticity and long-term potentiation mimicked in single inorganic synapses, Nat. Mater.; 10 (8) (2011) 591−595; Copyright 2011, Springer Nature.*

depending on the employed material stack. Kim et al. [144]. observed PPF for a filamentary Pd/Ta_2O_{5-x}/TaO_y/Pt device, as shown in Fig. 2.13B. The authors propose that one conditional spike (−1.1 V, 20 ns pulse in Fig. 2.13B) raises the temperature of the device, which slowly decays back to room temperature with a characteristic time of the order of 1 μs. A following spike (−0.7 V, 1 μs pulse in Fig. 2.13B) takes advantage of the preheating effect if it occurs within roughly 1 μs after the previous spike. The PPF is ascribed to a temperature effect because it does not depend on the voltage polarity of the conditional heating pulse [144]. Du et al. reported PPF in WO_x-based devices and ascribed it to the internal ion dynamics that govern the resistance switching [145]. The PPF effect lasts for a few milliseconds time, which compares well with the PPF in biological synapses. In this case, the short-term dynamics is sensitive to the voltage polarity of the preconditioning pulses, which allows the implementation of STDP with nonoverlapping identical pulses. In all these examples, the PPF is governed by a short-term dynamics internal to the device. Differently, the PPF effect has been realized by harnessing the CF instability of a Ag/Ag_2S/nanogap/scanning tunneling microscope tip structure [178]. In the nanogap region, the voltage application produced the growth of a silver CF, which dissolves in a second timescale. This phenomenon is identified also as short-term memory (STM). The repetition of voltage pulses with a sufficiently high rate brings the CF in a stable (long-term) state. Indeed, Fig. 2.13C shows that a pulse repetition period of 20 second produces a slight conductance change whereas a shorter period of 2 second brings the device to a high conductance value. The transition from an STM resistance state, due to CF instabilities, to a long-term resistance modification has been reported by other authors for both ECM [179] and VCM devices [146,180,181].

2.4.4 Oscillatory systems

In the previous subsections, we discussed about the research aimed at localizing some computational tasks in individual devices. In this subsection, we describe research trends aimed at building nonlinear elements which can be used to implement neuronal units [67] (please refer to Chapter 16: Neuronal realizations based on memristive devices for details) or to implement alternative computing architectures such as chaos computing [182]. The series connection of an element featuring NDR (see Section 2.2.3) with a fixed resistor and a capacitor displays an oscillatory behavior when stimulated by a fixed voltage [183−185]. Let us consider the representative $I-V$ curve of an NDR device operated in voltage mode (see Section 2.2.3) and the circuit reported in Fig. 2.14A. With an initial $V_{DD} = 0$ V, the memristor assumes an HRS value larger than that of the load resistor (R_{load}). In this case, the applied voltage V_{DD} drops mostly over the NDR device. Therefore, a V_{DD} value larger than the turn ON voltage, $V_{TH,ON}$, produces the switch of the NDR

device to the LRS state with $R \ll R_{load}$, which causes most of the voltage to drop on the load resistor in a time interval governed by the capacitor C and the $I-V$ hysteresis. The release of the voltage from the device, down to values below the turn OFF voltage, $V_{TH,OFF}$, lets it switch back to the HRS state, which completes an entire oscillation cycle. The process continues in this way and the circuit oscillates indefinitely. The circuit scheme in Fig. 2.14A is named Pearson–Anson relaxation oscillator after the scientists who first applied it to a Neon discharge lamp [187]. Further, the use of an external discrete capacitor may be avoided in the presence of parasitic capacitances that play the same role [68,183].

Although the literature about oscillator devices is still very recent and deals with proof of concepts for device functionalities rather than assessment of device performances, we try to evidence few properties that can be relevant for an efficient implementation of oscillator devices and oscillator-based computing systems. First, for an oscillation to occur, the HRS and LRS values of the switching element have to be much larger and much lower

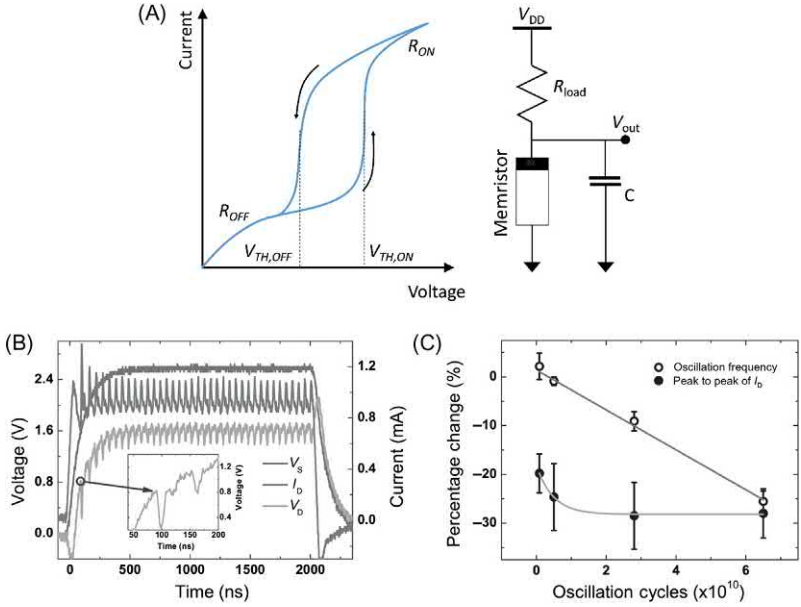


FIGURE 2.14 (A) Sketch of the $I-V$ curve as a function of applied voltage for a device showing NDR when programmed in current mode and oscillator circuit; (B) experimental data for voltage applied to the oscillator, current through the NDR device and output voltage; (C) endurance degradation of oscillator frequency and endurance of a NbO_x-based oscillator over 10^6 cycles. *(B and C) Reprinted with permission from S. Li, X. Liu, S.K. Nandi, D.K. Venkatachalam, R.G. Elliman, High-endurance megahertz electrical self-oscillation in Ti/NbOx bilayer structures, Appl. Phys. Lett.; 106 (21) (2015) 212902 [186]; Copyright 2015, American Institute of Physics.*

than the resistance value of the load resistance, respectively. This fact requires the switching element to ensure a high resistance ratio, which is usually obtained at the expense of power consumption. Indeed, most of the oscillator devices reported in the literature operate in the milliampere range [68,188], which is hardly compatible with electronic systems comprising a huge number of such devices. For what concerns the practical realization of VLSI-compatible oscillator-based system, the compact implementation of the load resistor may be an issue. Interesting is the solution proposed by Sharma et al. [183], who demonstrate the oscillation of a compact 1T-1R system without additional discrete capacitor and operating at relatively low currents. One of the most relevant performance parameters for the exploitation of RRAM device oscillators is the switching endurance (or more precisely the oscillation endurance), which is required to be extremely expanded to support oscillatory computing. Indeed, it has already been the subject of some investigations, which reported endurances up to 10^6 cycles with some oscillation amplitude and frequency degradation, as shown in Fig. 2.14C [186]. Furthermore, for high performance computing, high oscillation frequencies are required: values as high as 500 MHz have been reported for filamentary devices. These values are influenced by the electrical parasitics of the oscillator circuitry [69].

2.5 Conclusions and perspectives

In this chapter, we introduced various technologies belonging to the class of RRAM memories, comprising, filamentary and area-dependent VCM, ECM, and devices showing negative differential resistance. We discussed the basic operation principles of such devices and described their electric operation and conventional programming methodologies and performances, especially for storage applications.

Furthermore, we reviewed the main routes toward the implementation of unconventional RRAM functionalities toward all the advanced computing schemes that will be treated in Chapters 6−19, comprising multilevel capability, plasticity, timing-based and rate-based computing and oscillatory systems. Despite these routes are still wide open field of investigations, we presented qualitative discussions about the physical mechanisms involved in the implementation of such unconventional functionalities, which allowed us to discuss the general limitations, opportunities, trade-offs, and open issues. The major open point regards the plastic operation. Concerning analog implementation, values programming window and conductance window are currently extremely small and problematic for real applications and no much optimization work in this direction is present in the literature. The stochastic implementation of plasticity is less investigated than the analog one, but the same problems of limited programming and conductance window already appear in certain publications.

The highlighted open issues require scientific research at the edge between material science and device technology. Furthermore, now more than ever, a cross-fertilization between device and system-level research is increasing in importance, especially for the front-line functionality of timing-, rate-, and oscillator-based computing.

References

[1] R. Waser, R. Dittmann, G. Staikov, K. Szot, Redox-based resistive switching memories – nanoionic mechanisms, prospects, and challenges, Adv. Mater. 21 (25–26) (2009) 2632–2663.

[2] D. Ielmini, R. Bruchhaus, R. Waser, Thermochemical resistive switching: materials, mechanisms, and scaling projections, Phase Transit. 84 (7) (2011) 570–602.

[3] A. Sawa, Resistive switching in transition metal oxides, Mater. Today. 11 (6) (2008) 28–36.

[4] R. Muenstermann, T. Menke, R. Dittmann, R. Waser, Coexistence of filamentary and homogeneous resistive switching in Fe-doped $SrTiO_3$ thin-film memristive devices, Adv. Mater. 22 (43) (2010) 4819–4822.

[5] S. Menzel, S. Tappertzhofen, R. Waser, I. Valov, Switching kinetics of electrochemical metallization memory cells, Phys. Chem. Chem Phys. 15 (18) (2013) 6945–6952.

[6] I. Valov, I. Sapezanskaia, A. Nayak, T. Tsuruoka, T. Bredow, T. Hasegawa, et al., Atomically controlled electrochemical nucleation at superionic solid electrolyte surfaces, Nat. Mater. 11 (6) (2012) 530–535.

[7] A. Nayak, T. Tamura, T. Tsuruoka, K. Terabe, S. Hosaka, T. Hasegawa, et al., Rate-limiting processes determining the switching time in a Ag_2S atomic switch, J. Phys. Chem. Lett. 1 (3) (2010) 604–608.

[8] A. Nayak, T. Tsuruoka, K. Terabe, T. Hasegawa, M. Aono, Switching kinetics of a Cu_2S-based gap-type atomic switch, Nanotechnology. 22 (23) (2011) 235201.

[9] J.R. Jameson, N. Gilbert, F. Koushan, J. Saenz, J. Wang, S. Hollmer, et al., One-dimensional model of the programming kinetics of conductive-bridge memory cells, Appl. Phys. Lett. 99 (6) (2011) 063506.

[10] M.N. Kozicki, M. Balakrishnan, C. Gopalan, C. Ratnakumar, M. Mitkova, Programmable metallization cell memory based on Ag-Ge-S and Cu-Ge-S solid electrolytes, Non-Volatile Memory Technology Symposium, 2005, pp. 7–89.

[11] M. Kund, G. Beitel, C. Pinnow, T. Rohr, J. Schumann, R. Symanczyk, et al., Conductive bridging RAM (CBRAM): an emerging non-volatile memory technology scalable to sub 20nm, in: IEEE International Electron Devices Meeting, 2005 IEDM Technical Digest, 2005, pp. 754–757.

[12] C. Schindler, M. Meier, R. Waser, M.N. Kozicki, Resistive switching in Ag-Ge-Se with extremely low write currents, in: 2007 Non-Volatile Memory Technology Symposium, Albuquerque, NM: IEEE, 2007, pp. 82–85.

[13] I. Valov, T. Tsuruoka, Effects of moisture and redox reactions in VCM and ECM resistive switching memories, J. Phys. Appl. Phys. 51 (41) (2018) 413001.

[14] T. Tsuruoka, I. Valov, S. Tappertzhofen, J. van den Hurk, T. Hasegawa, R. Waser, et al., Redox reactions at Cu, Ag/Ta_2O_5 interfaces and the effects of Ta_2O_5 film density on the forming process in atomic switch structures, Adv. Funct. Mater. 25 (40) (2015) 6374–6381.

[15] C. Mannequin, T. Tsuruoka, T. Hasegawa, M. Aono, Composition of thin Ta_2O_5 films deposited by different methods and the effect of humidity on their resistive switching behavior, Jpn. J. Appl. Phys. 55 (6S1) (2016) 06GG08.
[16] M. Lübben, S. Menzel, S.G. Park, M. Yang, R. Waser, I. Valov, SET kinetics of electrochemical metallization cells: influence of counter-electrodes in SiO_2/Ag based systems, Nanotechnology. 28 (13) (2017) 135205.
[17] S. Tappertzhofen, R. Waser, I. Valov, Impact of the counter-electrode material on redox processes in resistive switching memories, ChemElectroChem. 1 (8) (2014) 1287–1292.
[18] I. Valov, R. Waser, J.R. Jameson, M.N. Kozicki, Electrochemical metallization memories—fundamentals, applications, prospects, Nanotechnology. 22 (25) (2011) 254003.
[19] S. Menzel, U. Böttger, R. Waser, Simulation of multilevel switching in electrochemical metallization memory cells, J. Appl. Phys. 111 (1) (2012) 014501.
[20] S. Menzel, Comprehensive modeling of electrochemical metallization memory cells, J. Comput. Electron. 16 (4) (2017) 1017–1037.
[21] S. Menzel, I., Valov, R. Waser, N. Adler, J. Hurk van den, S. Tappertzhofen, Simulation of polarity independent RESET in electrochemical metallization memory cells, in: 5th IEEE International Memory Workshop, 2013, pp. 92–95.
[22] U. Celano, L. Goux, A. Belmonte, K. Opsomer, R. Degraeve, C. Detavernier, et al., Understanding the dual nature of the filament dissolution in conductive bridging devices, J. Phys. Chem. Lett. 6 (10) (2015) 1919–1924.
[23] Y. Yang, P. Gao, L. Li, X. Pan, S. Tappertzhofen, S. Choi, et al., Electrochemical dynamics of nanoscale metallic inclusions in dielectrics, Nat. Commun. 5 (2014).
[24] K. Terabe, T. Hasegawa, T. Nakayama, M. Aono, Quantized conductance atomic switch, Nature. 433 (7021) (2005) 47–50.
[25] H.J. Kim, K.J. Yoon, T.H. Park, H.J. Kim, Y.J. Kwon, X.L. Shao, et al., Filament shape dependent reset behavior governed by the interplay between the electric field and thermal effects in the Pt/TiO_2/Cu electrochemical metallization device, Adv. Electron. Mater. 3 (2) (2017) 1600404.
[26] C.-P. Hsiung, H.-W. Liao, J.-Y. Gan, T.-B. Wu, J.-C. Hwang, F. Chen, et al., Formation and instability of silver nanofilament in Ag-based programmable metallization cells, ACS Nano 4 (9) (2010) 5414–5420.
[27] R. Midya, Z. Wang, J. Zhang, S.E. Savel'ev, C. Li, M. Rao, et al., Anatomy of Ag/Hafnia-based selectors with 10^{10} nonlinearity, Adv. Mater. 29 (12) (2017) 1604457.
[28] W. Chen, H.J. Barnaby, M.N. Kozicki, Volatile and non-volatile switching in Cu-SiO_2 programmable metallization cells, IEEE Electron. Device Lett. 37 (5) (2016) 580–583.
[29] Z. Wang, M. Rao, R. Midya, S. Joshi, H. Jiang, P. Lin, et al., Threshold switching of Ag or Cu in dielectrics: materials, mechanism, and applications, Adv. Funct. Mater. 28 (6) (2017) 1704862.
[30] Gopalakrishnan, R.S. Shenoy, C.T. Rettner, K. Virwani, D.S. Bethune, R.M. Shelby, et al., Highly-scalable novel access device based on mixed ionic electronic conduction (MIEC) materials for high density phase change memory (PCM) arrays, in: 2010 Symposium on VLSI Technology, 2010, pp. 205–206.
[31] Yu, Y. Wu, Y. Chai, J. Provine, H-P. Wong, Characterization of switching parameters and multilevel capability in HfO_x/AlO_x bi-layer RRAM devices, in: Proceedings of 2011 International Symposium on VLSI Technology, Systems and Applications, 2011, pp. 1–2.
[32] M.-J. Lee, C.B. Lee, D. Lee, S.R. Lee, M. Chang, J.H. Hur, et al., A fast, high-endurance and scalable non-volatile memory device made from asymmetric Ta_2O_{5-x}/TaO_{2-x} bilayer structures, Nat. Mater. 10 (8) (2011) 625–630.

[33] A. Hardtdegen, C.L. Torre, F. Cüppers, S. Menzel, R. Waser, S. Hoffmann-Eifert, Improved switching stability and the effect of an internal series resistor in HfO_2/TiO_xBilayer ReRAM cells, IEEE Trans. Electron. Devices. 65 (8) (2018) 3229−3236.

[34] F. Nardi, S. Balatti, S. Larentis, D.C. Gilmer, D. Ielmini, Complementary switching in oxide-based bipolar resistive-switching random memory, IEEE Trans. Electron. Devices. 60 (1) (2013) 70−77.

[35] A. Schönhals, J. Mohr, D.J. Wouters, R. Waser, S. Menzel, 3-bit resistive RAM write-read scheme based on complementary switching mechanism, IEEE Electron. Device Lett. 38 (4) (2017) 449−452.

[36] J.J. Yang, F. Miao, M.D. Pickett, D.A.A. Ohlberg, D.R. Stewart, C.N. Lau, et al., The mechanism of electroforming of metal oxide memristive switches, Nanotechnology. 20 (21) (2009) 215201.

[37] A. Marchewka, R. Waser, S. Menzel, Physical modeling of the electroforming process in resistive-switching devices, in: 2017 International Conference on Simulation of Semiconductor Processes and Devices (SISPAD), 2017, pp. 133−136.

[38] G.-S. Park, Y.B. Kim, S.Y. Park, X.S. Li, S. Heo, M.-J. Lee, et al., *In situ* observation of filamentary conducting channels in an asymmetric Ta_2O_{5-x}/TaO_{2-x} bilayer structure, Nat. Commun. 4 (2013) 2382.

[39] C. Baeumer, R. Valenta, C. Schmitz, A. Locatelli, T.O. Menteş, S.P. Rogers, et al., Subfilamentary networks cause cycle-to-cycle variability in memristive devices, ACS Nano. 11 (7) (2017) 6921−6929.

[40] D.K. Gala, A.A. Sharma, D. Li, J.M. Goodwill, J.A. Bain, M. Skowronski, Low temperature electroformation of TaO_x-based resistive switching devices, APL. Mater. 4 (1) (2016) 016101.

[41] A.A. Sharma, I.V. Karpov, R. Kotlyar, J. Kwon, M. Skowronski, J.A. Bain, Dynamics of electroforming in binary metal oxide-based resistive switching memory, J. Appl. Phys. 118 (11) (2015) 114903.

[42] R. Waser, R. Bruchhaus, S. Menzel, Redox-based resistive switching memories, in: Rainer Waser (Ed.), Nanoelectronics and Information Technology, third ed. Weinheim: Wiley-VCH, 2012 (Chapter 30).

[43] F.M. Puglisi, L. Larcher, A. Padovani, P. Pavan, A complete statistical investigation of RTN in HfO_2-based RRAM in high resistive state, IEEE Trans. Electron. Devices. 62 (8) (2015) 2606−2613.

[44] L. Vandelli, A. Padovani, L. Larcher, G. Bersuker, Microscopic modeling of electrical stress-induced breakdown in poly-crystalline hafnium oxide dielectrics, IEEE Trans. Electron. Devices. 60 (5) (2013) 1754−1762.

[45] K. Fleck, C. La Torre, N. Aslam, S. Hoffmann-Eifert, U. Böttger, S. Menzel, Uniting gradual and abrupt SET processes in resistive switching oxides, Phys. Rev. Appl. 6 (6) (2016).

[46] A. Marchewka, B. Roesgen, K. Skaja, H. Du, C.-L. Jia, J. Mayer, et al., Nanoionic resistive switching memories: on the physical nature of the dynamic reset process, Adv. Electron. Mater. 2 (2015) 1500233.

[47] S. Yu, B. Gao, Z. Fang, H. Yu, J. Kang, Wong, et al., A low energy oxide-based electronic synaptic device for neuromorphic visual systems with tolerance to device variation, Adv. Mater. 25 (12) (2013) 1774−1779.

[48] F. Nardi, S. Larentis, S. Balatti, D.C. Gilmer, D. Ielmini, Resistive switching by voltage-driven ion migration in bipolar RRAM—part I: experimental study, IEEE Trans. Electron. Devices. 59 (9) (2012) 2461−2467.

[49] L. Goux, Y.-Y. Chen, L. Pantisano, X.-P. Wang, G. Groeseneken, M. Jurczak, et al., On the gradual unipolar and bipolar resistive switching of TiN\ HfO_2\Pt memory systems, Electrochem. Solid-State Lett. 13 (6) (2010) G54–G56.

[50] W. Kim, S. Menzel, D.J. Wouters, Y. Guo, J. Robertson, B. Roesgen, et al., Impact of oxygen exchange reaction at the ohmic interface in Ta_2O_5-based ReRAM devices, Nanoscale. 8 (41) (2016) 17774–17781.

[51] D. Cooper, C. Baeumer, N. Bernier, A. Marchewka, C. La Torre, R.E. Dunin-Borkowski, et al., Anomalous resistance hysteresis in oxide ReRAM: oxygen evolution and reincorporation revealed by in situ TEM, Adv. Mater. 29 (23) (2017) 1700212.

[52] A. Schönhals, C.M.M. Rosário, S. Hoffmann-Eifert, R. Waser, S. Menzel, D.J. Wouters, Role of the electrode material on the RESET limitation in oxide ReRAM devices, Adv. Electron. Mater. 4 (2) (2018) 1700243.

[53] H. Zhang, S. Yoo, S. Menzel, C. Funck, F. Cüppers, D.J. Wouters, et al., Understanding the coexistence of two bipolar resistive switching modes with opposite polarity in Pt/TiO_2/Ti/Pt nanosized ReRAM devices, ACS Appl. Mater. Interfaces. 10 (35) (2018) 29766–29778.

[54] K. Shibuya, R. Dittmann, S. Mi, R. Waser, Impact of defect distribution on resistive switching characteristics of Sr_2TiO_4 thin films, Adv. Mater. 22 (3) (2010) 411–414.

[55] Schönhals A, Waser R, Menzel S, Rana V. 3-bit read scheme for single layer Ta2O5ReRAM. In: 2014 14th Annual Non-Volatile Memory Technology Symposium (NVMTS). 2014, pp. 1–4.

[56] S. Brivio, J. Frascaroli, S. Spiga, Role of metal-oxide interfaces in the multiple resistance switching regimes of Pt/HfO_2/TiN devices, Appl. Phys. Lett. 107 (2) (2015) 023504.

[57] E. Linn, R. Rosezin, C. Kügeler, R. Waser, Complementary resistive switches for passive nanocrossbar memories, Nat. Mater. 9 (5) (2010) 403–406.

[58] S. Balatti, S. Larentis, D.C. Gilmer, D. Ielmini, Multiple memory states in resistive switching devices through controlled size and orientation of the conductive filament, Adv. Mater. 25 (10) (2013) 1474–1478.

[59] Y. Aoki, C. Wiemann, V. Feyer, H.-S. Kim, C.M. Schneider, H. Ill-Yoo, et al., Bulk mixed ion electron conduction in amorphous gallium oxide causes memristive behaviour, Nat. Commun. 5 (2014) 3473.

[60] B. Arndt, F. Borgatti, F. Offi, M. Phillips, P. Parreira, T. Meiners, et al., Spectroscopic indications of tunnel barrier charging as the switching mechanism in memristive devices, Adv. Funct. Mater. 27 (45) (2017) 1702282.

[61] Govoreanu B, Redolfi A, Zhang L, Adelmann C, Popovici M, Clima S, et al. Vacancy-modulated conductive oxide resistive RAM (VMCO-RRAM): An area-scalable switching current, self-compliant, highly nonlinear and wide on/off-window resistive switching cell. In: Electron Devices Meeting (IEDM), 2013 IEEE International, pp. 10.2.1–10.2.4.

[62] S. Park, S. Jung, M. Siddik, M. Jo, J. Lee, J. Park, et al., Memristive switching behavior in $Pr_{0.7}Ca_{0.3}MnO_3$ by incorporating an oxygen-deficient layer, Phys. Status Solidi RRL – Rapid Res. Lett. 5 (10–11) (2011) 409–411.

[63] W.R. Acevedo, C. Ferreyra, M.J. Sánchez, C. Acha, R. Gay, D. Rubi, Concurrent ionic migration and electronic effects at the memristive TiO_x/$La_{1/3}Ca_{2/3}MnO_{3-x}$interface, J. Phys. Appl. Phys. 51 (12) (2018) 125304.

[64] T. Driscoll, H.-T. Kim, B.-G. Chae, M. Di Ventra, D.N. Basov, Phase-transition driven memristive system, Appl. Phys. Lett. 95 (4) (2009) 043503.

[65] S.H. Chang, S.B. Lee, D.Y. Jeon, S.J. Park, G.T. Kim, S.M. Yang, et al., Oxide double-layer nanocrossbar for ultrahigh-density bipolar resistive memory, Adv. Mater. 23 (35) (2011) 4063–4067.
[66] M. Son, J. Lee, J. Park, J. Shin, G. Choi, S. Jung, et al., Excellent selector characteristics of nanoscale VO_2 for high-density bipolar ReRAM applications, IEEE Electron. Device Lett. 32 (11) (2011) 1579–1581.
[67] M.D. Pickett, G. Medeiros-Ribeiro, R.S. Williams, A scalable neuristor built with Mott memristors, Nat. Mater. 12 (2) (2013) 114–117.
[68] X. Liu, S. Li, S.K. Nandi, D.K. Venkatachalam, R.G. Elliman, Threshold switching and electrical self-oscillation in niobium oxide films, J. Appl. Phys. 120 (12) (2016) 124102.
[69] A.A. Sharma, Y. Li, M. Skowronski, J.A. Bain, J.A. Weldon, High-frequency TaO_x-based compact oscillators, IEEE Trans. Electron. Devices. 62 (11) (2015) 3857–3862.
[70] M.D. Pickett, J. Borghetti, J.J. Yang, G. Medeiros-Ribeiro, R.S. Williams, Coexistence of memristance and negative differential resistance in a nanoscale metal-oxide-metal system, Adv. Mater. 23 (15) (2011) 1730–1733.
[71] C. Funck, S. Menzel, N. Aslam, H. Zhang, A. Hardtdegen, R. Waser, et al., Multidimensional simulation of threshold switching in NbO_2 based on an electric field triggered thermal runaway model, Adv. Electron. Mater. 2 (7) (2016) 1600169.
[72] F.A. Chudnovskii, L.L. Odynets, A.L. Pergament, G.B. Stefanovich, Electroforming and switching in oxides of transition metals: the role of metal–insulator transition in the switching mechanism, J. Solid. State Chem. 122 (1) (1996) 95–99.
[73] S. Slesazeck, H. Mähne, H. Wylezich, A. Wachowiak, J. Radhakrishnan, A. Ascoli, et al., Physical model of threshold switching in NbO_2 based memristors, RSC Adv. 5 (124) (2015) 102318–102322.
[74] G.A. Gibson, S. Musunuru, J. Zhang, K. Vandenberghe, J. Lee, C.-C. Hsieh, et al., An accurate locally active memristor model for S-type negative differential resistance in NbO_x, Appl. Phys. Lett. 108 (2) (2016) 023505.
[75] J.M. Goodwill, A.A. Sharma, D. Li, J.A. Bain, M. Skowronski, Electro-thermal model of threshold switching in TaO_x-based devices, ACS Appl. Mater. Interfaces. 9 (13) (2017) 11704–11710.
[76] S. Kumar, Z. Wang, N. Davila, N. Kumari, K.J. Norris, X. Huang, et al., Physical origins of current and temperature controlled negative differential resistances in NbO_2, Nat. Commun. 8 (1) (2017) 658.
[77] S. Menzel, U. Böttger, M. Wimmer, M. Salinga, Physics of the switching kinetics in resistive memories, Adv. Funct. Mater. 25 (2015) 6306–6325.
[78] J. van den Hurk, E. Linn, H. Zhang, R. Waser, I. Valov, Volatile resistance states in electrochemical metallization cells enabling non-destructive readout of complementary resistive switches, Nanotechnology. 25 (42) (2014) 425202.
[79] D. Ielmini, Modeling the universal set/reset characteristics of bipolar RRAM by field- and temperature-driven filament growth, IEEE Trans. Electron. Devices. 58 (12) (2011) 4309–4317.
[80] S. Menzel, M. Waters, A. Marchewka, U. Böttger, R. Dittmann, R. Waser, Origin of the ultra-nonlinear switching kinetics in oxide-based resistive switches, Adv. Funct. Mater. 21 (23) (2011) 4487–4492.
[81] Y. Nishi, S. Menzel, K. Fleck, U. Böttger, R. Waser, Origin of the SET kinetics of the resistive switching in tantalum oxide thin films, IEEE Electron. Device Lett. 35 (2) (2014) 259–261.

[82] Govoreanu B, Crotti D, Subhechha S, Zhang L, Chen YY, Clima S, et al. A-VMCO: a novel forming-free, self-rectifying, analog memory cell with low-current operation, nonfilamentary switching and excellent variability. In: 2015 Symposium on VLSI Technology (VLSI Technology). 2015, pp. T132−T133.

[83] N. Du, N. Manjunath, Y. Li, S. Menzel, E. Linn, R. Waser, et al., Field-driven hopping transport of oxygen vacancies in memristive oxide switches with interface-mediated resistive switching, Phys. Rev. Appl. 10 (2018) 5.

[84] S. Menzel, R. Waser, Analytical analysis of the generic SET and RESET characteristics of electrochemical metallization memory cells, Nanoscale. 5 (22) (2013) 11003−11010.

[85] Ielmini D, Menzel S. Universal Switching Behavior, in: Daniele Ielmini, Stephan Menzel (eds.), Resistive Switching. Weinheim: Wiley-VCH. (2016).

[86] S. Choi, Y. Yang, W. Lu, Random telegraph noise and resistance switching analysis of oxide based resistive memory, Nanoscale. 6 (1) (2014) 400−404.

[87] F.M. Puglisi, P. Pavan, L. Larcher, Random telegraph noise in HfO_x resistive random access memory: from physics to compact modeling, in: 2016 IEEE International Reliability Physics Symposium (IRPS), 2016, pp. MY-8-1−MY-8-5.

[88] S. Ambrogio, S. Balatti, V. McCaffrey, D. Wang, D. Ielmini, Impact of low-frequency noise on read distributions of resistive switching memory (RRAM), in: 2014 IEEE International Electron Devices Meeting (IEDM), pp. 14.4.1−14.4.4.

[89] S. Brivio, J. Frascaroli, E. Covi, S. Spiga, Stimulated ionic telegraph noise in filamentary memristive devices, Sci. Rep. 9 (1) (2019) 6310. 16 April.

[90] 2015 ITRS 2.0 OFFICIAL PUBLICATION - Beyond CMOS [Internet]. Dropbox. Available from: <https://www.dropbox.com/sh/3jfh5fq634b5yqu/AADYT8V2Nj5bX6C5q764kUg4a?dl = 0>.

[91] Intel 3D XPoint Memory Die Removed from Intel Optane™ PCM (Phase Change Memory) [Internet]. Available from: <http://www.techinsights.com/about-techinsights/overview/blog/intel-3D-xpoint-memory-die-removed-from-intel-optane-pcm/>.

[92] B. Govoreanu, G.S. Kar, Y.Y. Chen, V. Paraschiv, S. Kubicek, A. Fantini, et al., $10x10nm^2$ Hf/HfO_x crossbar resistive RAM with excellent performance, reliability and low-energy operation, in: 2011 International Electron Devices Meeting, 2011, pp. 31.6.1−31.6.4.

[93] K.-S. Li, C. Ho, M.-T. Lee, M.-C. Chen, C.-L. Hsu, J.M. Lu, et al., Utilizing Sub-5 nm sidewall electrode technology for atomic-scale resistive memory fabrication, in: 2014 Symposium on VLSI Technology (VLSI-Technology): Digest of Technical Papers, 2014, pp. 1−2.

[94] X. Ma, H. Wu, D. Wu, H. Qian, A 16 Mb RRAM test chip based on analog power system with tunable write pulses, in: 2015 15th Non-Volatile Memory Technology Symposium (NVMTS), 2015, pp. 1−3.

[95] D. Jana, S. Roy, R. Panja, M. Dutta, S.Z. Rahaman, R. Mahapatra, et al., Conductive-bridging random access memory: challenges and opportunity for 3D architecture, Nanoscale Res. Lett. 10 (1) (2015) 188.

[96] T.Y. Liu, T.H. Yan, R. Scheuerlein, Y. Chen, J.K. Lee, G. Balakrishnan, et al., A 130.7-mm^2 2-layer 32-Gb ReRAM Memory device in 24-nm technology, IEEE J. Solid-State Circuits 49 (1) (2014) 140−153.

[97] A. Fantini, G. Gorine, R. Degraeve, L. Goux, C.Y. Chen, A. Redolfi, et al. Intrinsic program instability in HfO_2 RRAM and consequences on program algorithms, in: 2015 IEEE International Electron Devices Meeting (IEDM), 2015, pp. 7.5.1−7.5.4.

[98] F.M. Puglisi, C. Wenger, P. Pavan, A novel program-verify algorithm for multi-bit operation in HfO_2 RRAM, IEEE Electron. Device Lett. 36 (10) (2015) 1030−1032.

[99] K. Higuchi, T.O. Iwasaki, K. Takeuchi, Investigation of verify-programming methods to achieve 10 million cycles for 50 nm HfO_2 ReRAM, in: 2012 4th IEEE International Memory Workshop. Milan: IEEE, 2012, pp. 1−4.

[100] J.J. Ryu, B.K. Park, T.-M. Chung, Y.K. Lee, G.H. Kim, Optimized method for low-energy and highly reliable multibit operation in a HfO_2-based resistive switching device, Adv. Electron. Mater. 0 (0) (2018) 1800261.

[101] A. Grossi, E. Nowak, C. Zambelli, C. Pellissier, S. Bernasconi, G. Cibrario, et al., Fundamental variability limits of filament-based RRAM, in: 2016 IEEE International Electron Devices Meeting (IEDM), 2016, pp. 4.7.1−4.7.4.

[102] R. Degraeve, A. Fantini, N. Raghavan, L. Goux, S. Clima, B. Govoreanu, et al., Causes and consequences of the stochastic aspect of filamentary RRAM, Microelectron. Eng. 147 (2015) 171−175.

[103] D. Garbin, E. Vianello, Q. Rafhay, M. Azzaz, P. Candelier, B. DeSalvo, et al., Resistive memory variability: a simplified trap-assisted tunneling model, Solid-State Electron. 115 (2016) 126−132.

[104] N. Raghavan, Performance and reliability trade-offs for high-κ RRAM, Microelectron. Reliab. 54 (9) (2014) 2253−2257.

[105] S. Brivio, J. Frascaroli, S. Spiga, Role of Al doping in the filament disruption in HfO_2 resistance switches, Nanotechnology. 28 (39) (2017) 395202.

[106] J. Frascaroli, F.G. Volpe, S. Brivio, S. Spiga, Effect of Al doping on the retention behavior of HfO_2 resistive switching memories, Microelectron. Eng. 147 (2015) 104−107.

[107] B. Magyari-Köpe, D. Duncan, L. Zhao, Y. Nishi, Doping technology for RRAM - Opportunities and challenges. In: 2016 International Symposium on VLSI Technology, Systems and Application (VLSI-TSA). (2016) 1−2.

[108] N. Raeis-Hosseini, S. Lim, H. Hwang, J. Rho, Reliable $Ge_2Sb_2Te_5$-integrated high-density nanoscale conductive bridge random access memory using facile nitrogen-doping strategy. Adv Electron Mater. 4 (2018) 1800360.

[109] M. Azzaz, A. Benoist, E. Vianello, D. Garbin, E. Jalaguier, C. Cagli, et al., Improvement of performances HfO_2-based RRAM from elementary cell to 16kb demonstrator by introduction of thin layer of Al_2O_3, Solid-State Electron. 125 (2016) 182−188.

[110] C. Nail, G. Molas, P. Blaise, G. Piccolboni, B. Sklenard, C. Cagli, et al., Understanding RRAM endurance, retention and window margin trade-off using experimental results and simulations, in: 2016 IEEE International Electron Devices Meeting (IEDM), 2016, pp. 4.5.1−4.5.4.

[111] Y.Y. Chen, M. Komura, R. Degraeve, B. Govoreanu, L. Goux, A. Fantini, et al., Improvement of data retention in HfO_2/Hf 1T1R RRAM cell under low operating current, in: Electron Devices Meeting (IEDM), 2013 IEEE International, 2013, pp. 10.1.1−10.1.4.

[112] M. Yu, Y. Cai, Z. Wang, Y. Fang, Y. Liu, Z. Yu, et al., Novel vertical 3D structure of TaO_x-based RRAM with self-localized switching region by sidewall electrode oxidation, Sci. Rep. 6 (2016) 1.

[113] G. Molas, J. Guy, M. Barci, F. Longnos, G. Palma, E. Vianello, et al., Conductive bridge RAM (CBRAM): functionality, reliability and applications, in: 2015 International Conference on Solid State Devices and Materials, 2015, pp. 1142−1143.

[114] Y.Y. Chen, B. Govoreanu, L. Goux, R. Degraeve, A. Fantini, G.S. Kar, et al., Balancing SET/RESET pulse for 10^{10} endurance in HfO_2/Hf 1T1R bipolar RRAM, IEEE Trans. Electron. Devices. 59 (12) (2012) 3243−3249.

[115] A.C. Torrezan, J.P. Strachan, G. Medeiros-Ribeiro, R.S. Williams, Sub-nanosecond switching of a tantalum oxide memristor, Nanotechnology. 22 (48) (2011) 485203.

[116] Z. Fang, H.Y. Yu, X. Li, N. Singh, G.Q. Lo, D.L. Kwong, Multilayer-based forming-free RRAM devices with excellent uniformity, IEEE Electron. Device Lett. 32 (4) (2011) 566–568.

[117] H.Y. Lee, P.S. Chen, T.Y. Wu, Y.S. Chen, C.C. Wang, P.J. Tzeng, et al., Low power and high speed bipolar switching with a thin reactive Ti buffer layer in robust HfO_2 based RRAM, in: 2008 IEEE International Electron Devices Meeting, 2008, pp. 1–4.

[118] S. Subhechha, B. Govoreanu, Y. Chen, S. Clima, K.D. Meyer, J.V. Houdt, et al., Extensive reliability investigation of a-VMCO nonfilamentary RRAM: relaxation, retention and key differences to filamentary switching, in: 2016 IEEE International Reliability Physics Symposium (IRPS), 2016, pp. 6C-2-1–6C-2–5.

[119] J. Ma, Z. Chai, W.D. Zhang, J.F. Zhang, Z. Ji, B. Benbakhti, et al., Investigation of pre-existing and generated defects in nonfilamentary a-Si/TiO_2 RRAM and their impacts on RTN amplitude distribution, IEEE Trans. Electron. Devices. 65 (3) (2018) 970–977.

[120] J.H. Yoon, K.M. Kim, S.J. Song, J.Y. Seok, K.J. Yoon, D.E. Kwon, et al., Pt/Ta_2O_5/HfO_{2-x}/Ti resistive switching memory competing with multilevel NAND flash, Adv. Mater. 27 (25) (2015) 3811–3816.

[121] J.H. Yoon, S.J. Song, I.-H. Yoo, J.Y. Seok, K.J. Yoon, D.E. Kwon, et al., Highly uniform, electroforming-free, and self-rectifying resistive memory in the Pt/Ta_2O_5/HfO_{2-x}/TiN structure, Adv. Funct. Mater. 24 (32) (2014) 5086–5095.

[122] H. Choi, J. Yi, S. Hwang, S. Lee, S. Song, S. Lee, et al., The effect of tunnel barrier at resistive switching device for low power memory applications, in: 2011 3rd IEEE International Memory Workshop (IMW), 2011, pp. 1–4.

[123] M. Jo, D. Seong, S. Kim, J. Lee, W. Lee, J. Park, et al., Novel cross-point resistive switching memory with self-formed schottky barrier, in: 2010 Symposium on VLSI Technology, 2010, pp. 53–54.

[124] S, Jung, M, Siddik, W. Lee, J. Park, X. Liu, J. Woo, et al., Thermally-assisted Ti/$Pr_{0.7}Ca_{0.3}MnO_3$ReRAM with excellent switching speed and retention characteristics, in: 2011 International Electron Devices Meeting, 2011, pp. 3.6.1–3.6.4.

[125] C.-W. Hsu, I.-T. Wang, C.-L. Lo, M.-C. Chiang, W.-Y. Jang, C.-H. Lin, et al., Self-rectifying bipolar TaO_x/TiO_2 RRAM with superior endurance over 10^{12} cycles for 3D high-density storage-class memory, in: 2013 Symposium on VLSI Technology (VLSIT), 2013, pp. T166–T167.

[126] S. Subhechha, R. Degraeve, P. Roussel, L. Goux, S. Clima, K.D. Meyer, et al., Kinetic defect distribution approach for modeling the transient, endurance and retention of a-VMCO RRAM, in: 2017 IEEE International Reliability Physics Symposium (IRPS), 2017, pp. 5A-5.1–5A-5.6.

[127] S. Park, M.K. Yang, H. Ju, D. Seong, J.M. Lee, E. Kim, et al., A non-linear ReRAM cell with sub-1μA ultralow operating current for high density vertical resistive memory (VRRAM), in: 2012 International Electron Devices Meeting, 2012, pp. 20.8.1–20.8.4.

[128] C.J. Chevallier, C.H. Siau, S.F. Lim, S.R. Namala, M. Matsuoka, B.L. Bateman, et al., A 0.13μm 64Mb multi-layered conductive metal-oxide memory, in: 2010 IEEE International Solid-State Circuits Conference - (ISSCC), 2010, pp. 260–261.

[129] Panasonic Starts World's First Mass Production of ReRAM Mounted Microcomputers | Headquarters News | Panasonic Newsroom Global [Internet]. Available from: <https://news.panasonic.com/global/press/data/2013/07/en130730-2/en130730-2.html>.

[130] Fujitsu Semiconductor Launches World's Largest Density 4 Mbit ReRAM Product for Mass Production : FUJITSU SEMICONDUCTOR [Internet]. Available from: <http://www.fujitsu.com/jp/group/fsl/en/resources/news/press-releases/2016/1026.html>.

[131] CBRAM | Adesto Technologies [Internet]. Available from: <https://www.adestotech.com/about-us/cbram/>.

[132] R. Fackenthal, M. Kitagawa, W. Otsuka, K. Prall, D. Mills, K. Tsutsui, et al., 19.7 A 16Gb ReRAM with 200MB/s write and 1GB/s read in 27nm technology, in: 2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2014, pp. 338−339.

[133] ReRAM Memory Overview | Crossbar [Internet]. Crossbar Website. Available from: <https://www.crossbar-inc.com/en/technology/reram-overview/>.

[134] Y. Chen, C. Petti, ReRAM technology evolution for storage class memory application, in: 2016 46th European Solid-State Device Research Conference (ESSDERC), 2016, pp. 432−435.

[135] A. Shilov, Western Digital to Use 3D ReRAM as Storage Class Memory for Special-Purpose SSDs [Internet]. Available from: <https://www.anandtech.com/show/10562/western-digital-to-use-3d-reram-as-storage-class-memory-for-specialpurpose-ssds>.

[136] S. Brivio, S. Spiga, Stochastic circuit breaker network model for bipolar resistance switching memories, J. Comput. Electron. 16 (4) (2017) 1154−1166.

[137] S. Stathopoulos, A. Khiat, M. Trapatseli, S. Cortese, A. Serb, I. Valov, et al., Multibit memory operation of metal-oxide bi-layer memristors, Sci. Rep. 7 (1) (2017) 17532.

[138] C. Li, M. Hu, Y. Li, H. Jiang, N. Ge, E. Montgomery, et al., Analogue signal and image processing with large memristor crossbars, Nat. Electron. 1 (1) (2018) 52−59.

[139] G. Indiveri, B. Linares-Barranco, R. Legenstein, G. Deligeorgis, T. Prodromakis, Integration of nanoscale memristor synapses in neuromorphic computing architectures, Nanotechnology. 24 (38) (2013) 384010.

[140] S. Agarwal, S.J. Plimpton, D.R. Hughart, A.H. Hsia, I. Richter, J.A. Cox, et al., Resistive memory device requirements for a neural algorithm accelerator, in: Neural Networks (IJCNN), 2016 International Joint Conference on [Internet]. IEEE, 2016, pp. 929−938. Available from: <http://ieeexplore.ieee.org/abstract/document/7727298/>.

[141] Y.-F. Wang, Y.-C. Lin, I.-T. Wang, T.-P. Lin, T.-H. Hou, Characterization and modeling of nonfilamentary Ta/TaO_x/TiO_2/Ti analog synaptic device, Sci. Rep. (2015) 5.

[142] J. Woo, K. Moon, J. Song, S. Lee, M. Kwak, J. Park, et al., Improved synaptic behavior under identical pulses using AlO_x/HfO_2 bilayer RRAM array for neuromorphic systems, IEEE Electron. Device Lett. 37 (8) (2016) 994−997.

[143] E. Covi, S. Brivio, J. Frascaroli, M. Fanciulli, S. Spiga, Invited) Analog HfO_2-RRAM switches for neural networks, ECS Trans. 75 (32) (2017) 85−94.

[144] S. Kim, C. Du, P. Sheridan, W. Ma, S. Choi, W.D. Lu, Experimental demonstration of a second-order memristor and its ability to biorealistically implement synaptic plasticity, Nano Lett. 15 (3) (2015) 2203−2211.

[145] C. Du, W. Ma, T. Chang, P. Sheridan, W.D. Lu, Biorealistic implementation of synaptic functions with oxide memristors through internal ionic dynamics, Adv. Funct. Mater. 25 (27) (2015) 4290−4299.

[146] Z.-H. Tan, R. Yang, K. Terabe, X.-B. Yin, X.-D. Zhang, X. Guo, Synaptic metaplasticity realized in oxide memristive devices, Adv. Mater. 28 (2) (2016) 377−384.

[147] L. Zhao, H.-Y. Chen, S.-C. Wu, Z. Jiang, S. Yu, T.-H. Hou, et al., Multi-level control of conductive nano-filament evolution in HfO_2 ReRAM by pulse-train operations, Nanoscale. 6 (11) (2014) 5698.

[148] J. Frascaroli, S. Brivio, E. Covi, S. Spiga, Evidence of soft bound behaviour in analogue memristive devices for neuromorphic computing, Sci. Rep. 8 (1) (2018) 7178.

[149] S. Brivio, E. Covi, A. Serb, T. Prodromakis, M. Fanciulli, S. Spiga, Gradual set dynamics in HfO_2-based memristor driven by sub-threshold voltage pulses, in: 2015 International Conference on Memristive Systems (MEMRISYS), 2015, pp. 1–2.

[150] Y. Matveyev, K. Egorov, A. Markeev, A. Zenkevich, Resistive switching and synaptic properties of fully atomic layer deposition grown TiN/HfO_2/TiN devices, J. Appl. Phys. 117 (4) (2015) 044901.

[151] S. Brivio, E. Covi, A. Serb, T. Prodromakis, M. Fanciulli, S. Spiga, Experimental study of gradual/abrupt dynamics of HfO_2-based memristive devices, Appl. Phys. Lett. 109 (13) (2016) 133504.

[152] Z. Wang, M. Yin, T. Zhang, Y. Cai, Y. Wang, Y. Yang, et al., Engineering incremental resistive switching in TaO_x based memristors for brain-inspired computing, Nanoscale. 8 (29) (2016) 14015–14022.

[153] P. Yao, H. Wu, B. Gao, S.B. Eryilmaz, X. Huang, W. Zhang, et al., Face classification using electronic synapses, Nat. Commun. 8 (2017) 15199.

[154] J. Park, M. Kwak, K. Moon, J. Woo, D. Lee, H. Hwang, TiO_x-Based RRAM synapse with 64-levels of conductance and symmetric conductance change by adopting a hybrid pulse scheme for neuromorphic computing, IEEE Electron. Device Lett. 37 (12) (2016) 1559–1562.

[155] K. Moon, A. Fumarola, S. Sidler, J. Jang, P. Narayanan, R.M. Shelby, et al., Bidirectional non-filamentary RRAM as an analog neuromorphic synapse, Part I: Al/Mo/$Pr_{0.7}Ca_{0.3}MnO_3$ material improvements and device measurements, IEEE J. Electron. Devices Soc. 6 (2018) 146–155.

[156] J.W. Jang, S. Park, G.W. Burr, H. Hwang, Y.H. Jeong, Optimization of conductance change in $Pr_{1-x}Ca_xMnO_3$-based synaptic devices for neuromorphic systems, IEEE Electron. Device Lett. 36 (5) (2015) 457–459.

[157] P.-Y. Chen, B. Lin, I.-T. Wang, Hou T.-H., J. Ye, S. Vrudhula, et al., Mitigating effects of non-ideal synaptic device characteristics for on-chip learning, in: 2015 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), IEEE, 2015, pp. 194–199.

[158] A. Fumarola, S. Sidler, K. Moon, J. Jang, R.M. Shelby, P. Narayanan, et al., Bidirectional non-filamentary RRAM as an analog neuromorphic synapse, part II: impact of Al/Mo/$Pr_{0.7}Ca_{0.3}MnO_3$ device characteristics on neural network training accuracy, IEEE J. Electron. Devices Soc. 6 (2018) 169–178.

[159] D. Lee, K. Moon, J. Park, S. Park, H. Hwang, Trade-off between number of conductance states and variability of conductance change in $Pr_{0.7}Ca_{0.3}MnO_3$-based synapse device, Appl. Phys. Lett. 106 (11) (2015) 113701.

[160] S. La Barbera, D.R.B. Ly, G. Navarro, N. Castellani, O. Cueto, G. Bourgeois, et al., Narrow heater bottom electrode-based phase change memory as a bidirectional artificial synapse, Adv. Electron. Mater. 4 (9) (2018) 1800223.

[161] S. Brivio, D. Conti, M.V. Nair, J. Frascaroli, E. Covi, C. Ricciardi, et al., Extended memory lifetime in spiking neural networks employing memristive synapses with nonlinear conductance dynamics, Nanotechnology. 30 (1) (2019) 015102.

[162] D. Garbin, E. Vianello, O. Bichler, Q. Rafhay, C. Gamrat, G. Ghibaudo, et al., HfO_2-based OxRAM devices as Synapses for convolutional neural networks, IEEE Trans. Electron. Devices. 62 (8) (2015) 2494–2501.

[163] M. Suri, D. Querlioz, O. Bichler, G. Palma, E. Vianello, D. Vuillaume, et al., Bio-inspired stochastic computing using binary CBRAM synapses, IEEE Trans. Electron. Devices. 60 (7) (2013) 2402−2409.
[164] S. Gaba, P. Sheridan, J. Zhou, S. Choi, W. Lu, Stochastic memristive devices for computing and neuromorphic applications, Nanoscale. 5 (13) (2013) 5872−5878.
[165] S. Yu, B. Gao, Z. Fang, H. Yu, J. Kang, H.-S.P. Wong, Stochastic learning in oxide binary synaptic device for neuromorphic computing, Front. Neurosci. 7 (2013) 186.
[166] T. Werner, E. Vianello, O. Bichler, D. Garbin, D. Cattaert, B. Yvert, et al., Spiking neural networks based on OxRAM synapses for real-time unsupervised spike sorting, Front. Neurosci. 10 (2016) 474.
[167] G. Piccolboni, G. Molas, D. Garbin, E. Vianello, O. Cueto, C. Cagli, et al., Investigation of cycle-to-cycle variability in HfO_2-based OxRAM, IEEE Electron. Device Lett. 37 (6) (2016) 721−723.
[168] Y. Nishi, U. Böttger, R. Waser, S. Menzel, Crossover from deterministic to stochastic nature of resistive-switching statistics in a tantalum oxide thin film, IEEE Trans. Electron. Devices. 65 (10) (2018) 4320−4325.
[169] J. Bill, R. Legenstein, A compound memristive synapse model for statistical learning through STDP in spiking neural networks, Front. Neurosci. 8 (2014) 412.
[170] M. Suri, O. Bichler, D. Querlioz, G. Palma, E. Vianello, D. Vuillaume, et al., CBRAM devices as binary synapses for low-power stochastic neuromorphic systems: Auditory (Cochlea) and visual (Retina) cognitive processing applications, in: 2012 International Electron Devices Meeting, 2012, pp. 10.3.1−10.3.4.
[171] M. Zhao, H. Wu, B. Gao, Q. Zhang, W. Wu, S. Wang, et al., Investigation of statistical retention of filamentary analog RRAM for neuromophic computing, in: 2017 IEEE International Electron Devices Meeting (IEDM), 2017, pp. 39.4.1−39.4.4.
[172] I.-T. Wang, Y.-C. Lin, Y.-F. Wang, C.-W. Hsu, T.-H. Hou, 3D synaptic architecture with ultralow sub-10 fJ energy per spike for neuromorphic computation, in: 2014 IEEE International Electron Devices Meeting (IEDM), 2014, pp. 28.5.1−28.5.4.
[173] G. Bi, M. Poo, Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type, J. Neurosci. 18 (24) (1998) 10464−10472.
[174] P.J. Sjöström, G.G. Turrigiano, S.B. Nelson, Rate, timing, and cooperativity jointly determine cortical synaptic plasticity, Neuron. 32 (6) (2001) 1149−1164.
[175] S. Ambrogio, S. Balatti, V. Milo, R. Carboni, Z.Q. Wang, A. Calderoni, et al., Neuromorphic learning and recognition with one-transistor-one-resistor synapses and bistable metal oxide RRAM, IEEE Trans. Electron. Devices. 63 (4) (2016) 1508−1515.
[176] E. Covi, S. Brivio, A. Serb, T. Prodromakis, M. Fanciulli, S. Spiga, Analog memristive synapse in spiking networks implementing unsupervised learning, Front. Neurosci. 10 (2016) 482.
[177] M. Prezioso, F. Merrikh Bayat, B. Hoskins, K. Likharev, D. Strukov, Self-adaptive spike-time-dependent plasticity of metal-oxide memristors, Sci. Rep. 6 (2016) 21331.
[178] T. Ohno, T. Hasegawa, T. Tsuruoka, K. Terabe, J.K. Gimzewski, M. Aono, Short-term plasticity and long-term potentiation mimicked in single inorganic synapses, Nat. Mater. 10 (8) (2011) 591−595.
[179] S. La Barbera, D. Vuillaume, F. Alibart, Filamentary switching: synaptic plasticity through device volatility, ACS Nano. 9 (1) (2015) 941−949.
[180] T. Chang, S.-H. Jo, W. Lu, Short-term memory to long-term memory transition in a nanoscale memristor, ACS Nano. 5 (9) (2011) 7669−7676.

[181] R. Berdan, E. Vasilaki, A. Khiat, G. Indiveri, A. Serb, T. Prodromakis, Emulating short-term synaptic dynamics with memristive devices, Sci. Rep. 6 (2016) 18639.

[182] S. Kumar, J.P. Strachan, R.S. Williams, Chaotic dynamics in nanoscale NbO_2 Mott memristors for analogue computing, Nature. 548 (7667) (2017) 318−321.

[183] A.A. Sharma, T.C. Jackson, M. Schulaker, C. Kuo, C. Augustine, J.A. Bain, et al., High performance, integrated 1T1R oxide-based oscillator: Stack engineering for low-power operation in neural network applications, in: 2015 Symposium on VLSI Technology (VLSI Technology), 2015, pp. T186−T187.

[184] A. Ascoli, S. Slesazeck, H. Mähne, R. Tetzlaff, T. Mikolajick, Nonlinear dynamics of a locally-active memristor, IEEE Trans. Circuits Syst. Regul. Pap. 62 (4) (2015) 1165−1174.

[185] S.K. Nandi, S. Li, X. Liu, R.G. Elliman, Temperature dependent frequency tuning of NbO_x relaxation oscillators, Appl. Phys. Lett. 111 (20) (2017) 202901.

[186] S. Li, X. Liu, S.K. Nandi, D.K. Venkatachalam, R.G. Elliman, High-endurance megahertz electrical self-oscillation in Ti/NbO_x bilayer structures, Appl. Phys. Lett. 106 (21) (2015) 212902.

[187] S.O. Pearson, H.S.G. Anson, The neon tube as a means of producing intermittent currents, Proc. Phys. Soc. Lond. 34 (1) (1921) 204.

[188] P.Y. Chen, J. Seo, Y. Cao, S. Yu. Compact oscillation neuron exploiting metal-insulator-transition for neuromorphic computing, in: 2016 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), 2016, pp. 1−6.

Chapter 3

Phase-change memory

Manuel Le Gallo and Abu Sebastian
IBM Research – Zurich, Rüschlikon, Switzerland

3.1 Introduction

3.1.1 Historical overview of phase-change memory

Phase-change memory (PCM) exploits the behavior of so-called phase-change materials that can be switched reversibly between amorphous and crystalline phases of different electrical resistivity. The amorphous phase tends to have high electrical resistivity, whereas the crystalline phase exhibits a low resistivity, sometimes three or four orders of magnitude lower. This large resistance contrast is used to store information in PCM (the high-resistance state can represent a logical "0" while the low-resistance state can represent a logical "1"). Thus a PCM device consists essentially of a layer of phase-change material sandwiched between two metal electrodes.

In the mid-1950s, the semiconducting properties of chalcogenide-based glasses were discovered by Kolomiets and Goryunova at the Ioffe Physical-Technical Institute [1]. Thereafter in 1968 Stanford R. Ovshinsky of Energy Conversion Devices observed a fast reversible switching effect in the $Si_{12}Te_{48}As_{30}Ge_{10}$ (STAG) composition [2]. He also observed, for the first time, a memory effect when slightly changing the STAG material composition, whereby the retention of the low-resistance state obtained after switching was maintained even in the absence of voltage [2]. Ovshinsky noted possible commercial applications of these materials as the active region of electronic switches and memory cells [3]. In 1970 a 256-bit array of amorphous semiconductor memory cells was developed by R. G. Neale, D. L. Nelson and Gordon E. Moore [4].

Further attempts to develop reliable PCM cells from the 1970s up to early 2000s encountered significant difficulties due to device degradation and instability of operation, and thus the interest in making electrical memory cells with phase-change materials gradually decreased. However phase-change materials became widely used since the 1990s in optical memory devices and still currently serve as the information storage medium in CDs,

Memristive Devices for Brain-Inspired Computing. DOI: https://doi.org/10.1016/B978-0-08-102782-0.00003-4

DVDs, and Blu-Ray disks [5]. In optical memory, the phase-change material is heated with a laser source and it is the contrast in optical reflectivity between the amorphous and crystalline phases that is used to store information.

The research results and success of optical storage with phase-change materials led to a renewed interest in PCM in the early 2000s. Companies such as Intel, Samsung, STMicroelectronics and SKHynix licensed the technology from Ovonyx (who owned the proprietary PCM technology originally invented by Ovshinsky; it was acquired by Micron in 2012) and started building their own PCM chips of various size, up to 8 Gb [6]. The first PCM product consisting of 128-Mbit memories in a 90-nm process was introduced in 2008 by Numonyx [7], a memory company launched by Intel and STMicroelectronics that was acquired by Micron in 2010 [8]. A 45-nm 1-Gbit PCM chip supplied to Nokia for inclusion in mobile phones was introduced by Micron in 2012 but withdrawn in 2014 [8]. The latest key technological development in PCM is the recent announcement of 3D Xpoint memory by Intel and Micron in July 2015, for which it is widely believed that a phase-change alloy is used as the storage part of the memory element [9]. This technology has been first released in 2018 under the brand Intel Optane and is currently available as a low-latency low-capacity nonvolatile memory (16–64GB) that can be used to accelerate existing storage [10].

3.1.2 Applications of phase-change memory

3.1.2.1 Memory technology

The memory hierarchy of conventional computing architectures is designed to bridge the performance gap between the fast central processing units (CPU) and the slower memory and storage technologies. A technology classified as storage is nonvolatile (i.e., the stored data will be retained when the power supply is turned off) and low-cost, but has much slower access times than the CPU operations (Fig. 3.1). Storage technologies include NOR and NAND Flash, magnetic hard disk drive (HDD), and tape. Memory technologies on the other hand are volatile (the data are lost when the power supply is turned off) and more expensive than storage, but have much smaller access times. Memory technologies include the static random access memory (SRAM) used in the CPU caches and off-chip dynamic random access memory (DRAM).

The use of PCM as potential DRAM replacement as part of the main memory system has been investigated in a wide variety of works for more than 10 years as of now [12–15]. At the time when the first investigations were performed (around 2009), DRAM had fallen behind NAND Flash and standard complementary metal-oxide semiconductor logic technologies in terms of scaling to the 45 nm technology node and preparation for the 32 nm

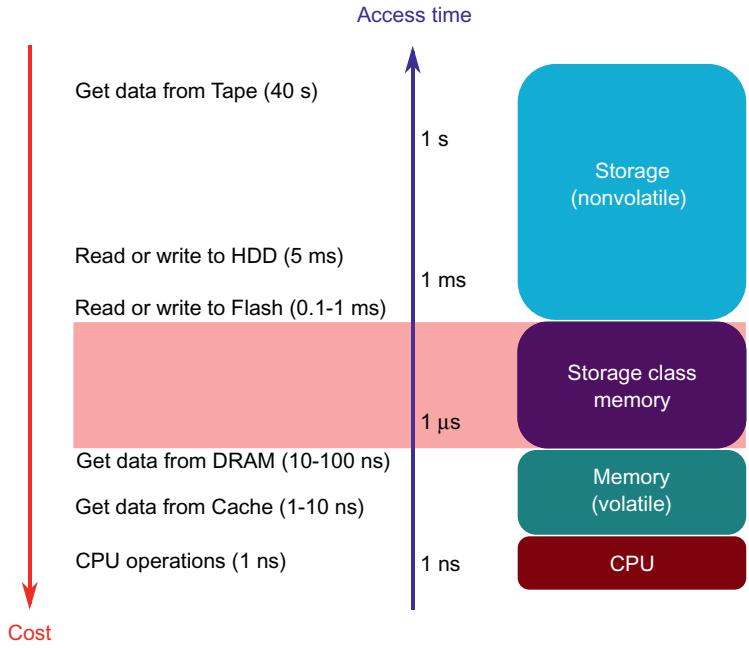


FIGURE 3.1 Access times for various memory and storage technologies. Small amounts of expensive high-performance volatile memory sits near the CPU whereas vast amounts of low-cost yet slow storage is used to stock data. Currently there exists a gap in access times of about three orders of magnitude between memory and storage, which may be potentially filled by a so-called "storage class memory". *Modified from G. Burr, M. Breitwisch, M. Franceschini, D. Garetto, K. Gopalakrishnan, B. Jackson, et al., "Phase change memory technology," J. Vac. Sci. Technol. B, 28, p. 223, 2010.*

node [11]. However PCM had already been demonstrated to scale down to 20 nm node [16]. Hence PCM could compete favorably in terms of forward scaling for increasing main memory density and capacity due to challenges in making DRAM capacitors small and yet being able to store charge reliably. Various studies conclude that in case PCM can be made at a higher density than DRAM, various architectural reorganizations of the main memory system could make PCM a viable alternative to DRAM despite the lower latency and finite endurance. Moreover the nonvolatility of PCM could be exploited in the main memory and would avoid to rewrite after each read access, which is unavoidable with DRAM [11]. However at the time of writing, DDR4 DRAM technology has been scaled down to 10 nm-class node, which denotes a process technology node somewhere between 10 and 19 nm [17]. Due to the recent advances in integrating DRAM in smaller nodes, it is currently unclear whether PCM will be able to displace such a stable and reliable technology.

Another potential application of PCM as a conventional memory technology is its use as so-called storage class memory (SCM). As shown in Fig. 3.1, there is currently a gap of three orders of magnitude between the access times of DRAM and Flash. SCM aims at bridging this performance–cost gap between memory and storage, which could be made possible with PCM. SCM would blur the traditional boundaries between storage and memory by combining the benefits of a solid-state memory, such as high performance and robustness, with the archival capabilities and low cost of conventional hard disk magnetic storage [11]. One variant of SCM could act as a fast solid-state drive (SSD) with better native endurance and write access times than the Flash-based SSDs. Access times on the order of 1 μs would be acceptable, but low cost via high density would be most important [11]. Another variant could have access times on the order of 100 ns with low-power and cost constraints, which would be fast enough to enable it to be connected to the usual memory controller [11]. SCM would likely not be as fast as DRAM, but its nonvolatility could allow the amount of DRAM required to maintain a high bandwidth to be greatly reduced. In this way the power consumption and hopefully the cost of the overall system would be reduced [11].

3.1.2.2 Non-von Neumann computing

An additional key emerging application area for resistive memory devices such as PCM is that of non-von Neumann computing. In this novel computing paradigm, memory elements are not only used to store information but also execute computational tasks with collocated memory and processing at considerable speeds. For this a low-power, multistate, programmable and nonvolatile nanoscale memory device is needed. Resistive memory devices (or *memristive devices*) that remember the history of the current that previously flowed through them, are promising candidates for this application. Memristive devices include PCM but also other emerging nonvolatile memories such as resistive random access memory, conductive bridge random access memory, or magnetic random access memory [18]. A significant implication of this concept is that the clear-cut distinction between memory and computing is blurred-out, which may lead to entirely new computational models and algorithms that would take advantage of non-von Neumann architectures.

Two non-von Neumann computing paradigms using memristive devices have currently emerged. In one approach memristive devices are used for implementing neuromorphic computing systems. The aim is to perform machine learning tasks using a neural network system whereby the neurons and/or synapses consisting the neural network are implemented with memristive devices. Another fascinating paradigm is that of in-memory computing, whereby the physical attributes and state dynamics of memristive devices are

used to perform logic operations or for analog computing, without being tied to a neural network framework.

The feasibility to program single PCM devices to a wide range of different states (inherent to the working principle of PCM) is promising for non-von Neumann computing applications and has been in fact exploited in all experimental demonstrations using PCM to date. Another key property of PCM is that the amorphous region can be progressively crystallized by the application of repetitive electrical pulses. This accumulation property (the PCM in fact *integrates* the electric current flowing through it) is essential for emulating synaptic dynamics and can also be used to implement some arithmetic operations. These promising characteristics indicate that PCM could potentially play a key role as the central element in a non-von Neumann computing system.

3.2 Essentials of phase-change memory

A fundamental property of a memory device is that it must allow the storage and retrieval of data. PCM records data by causing a phase-change material inside the memory device to switch from a crystalline (ordered) phase to an amorphous (disordered) phase and vice versa. This transformation is accompanied by a strong change of electrical and optical properties. The amorphous phase has a high resistivity and low optical reflectivity, whereas the crystalline phase has a low resistivity and a high optical reflectivity. The contrast in optical properties of phase-change materials has been widely employed to enable optical data storage devices such as DVDs and Blu-Ray discs. For electrical data storage with PCM, however, it is the contrast in resistivity between the two phases which is used to store information. Thus a WRITE operation in PCM consists in switching between the amorphous and crystalline states via the application of an electrical pulse. A READ operation typically consists in reading the electrical resistance of the PCM device, which then allows to know whether it is in the amorphous (high-resistance, logical “0”) or crystalline (low-resistance, logical “1”) state.

After the discovery of the memory effect, it soon became clear that it is associated with a material transition from an amorphous phase to a crystalline phase [3]. The amorphous phase is a thermodynamically unstable glass but the crystallization time at room temperature is very long. However by heating the amorphous material to a high enough temperature but below the melting temperature, it will rapidly crystallize. To transform the material back to the amorphous phase, it needs to be heated above its melting temperature and then rapidly cooled down. This rapid cooldown will “freeze” the atomic structure into a disordered state. In PCM the heat is produced by the passage of an electric current through the phase-change material (Joule heating effect). The electrical pulse used to switch the device to the high-resistance amorphous state is referred to as RESET pulse, and the pulse used

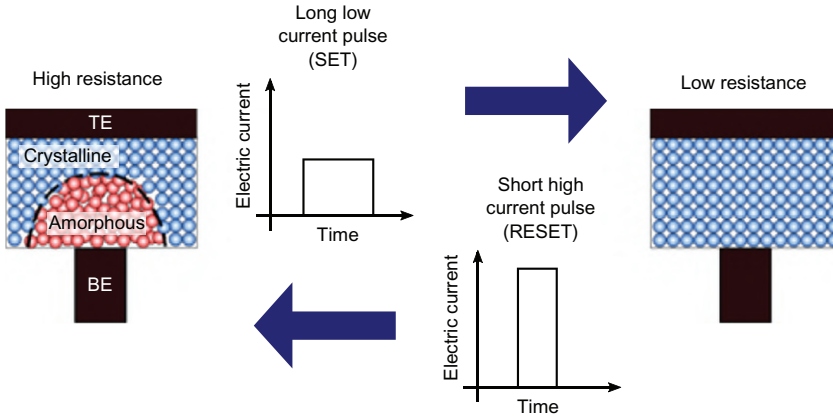


FIGURE 3.2 Operation principle of PCM. The mushroom-type PCM device displayed consists of a layer of phase-change material sandwiched between a top electrode (TE) and a narrower bottom electrode (BE). A long low current pulse (SET) is applied to bring the PCM device to the low-resistance crystalline state. A short high current pulse (RESET) is applied to bring the PCM device to the high-resistance amorphous state. *Modified from M. Wuttig and N. Yamada, "Phase-change materials for rewriteable data storage," Nat. Mater., 6 (11), 2007, 824–832.*

to switch the device back to the low-resistance crystalline state is referred to as SET pulse (Fig. 3.2).

A ternary phase diagram of the most commonly used phase-change alloys for both PCM and optical storage is shown in Fig. 3.3. In contrast to the strong glass-forming chalcogenide-based alloys used in the 1970s such as STAG, commonly used alloys nowadays lie along the $GeTe-Sb_2Te_3$ line, which show much faster recrystallization [5]. A phase-change material in this family frequently used in commercial products for both optical storage and PCM is $Ge_2Sb_2Te_5$ (GST). A second family of doped Sb_2Te alloys such as $Ag_5In_5Sb_{60}Te_{30}$ (AIST) is also often used.

Different types of memory cell designs are possible in order to build PCM devices based on such alloys. A typical PCM cell is designed such that the volume of phase-change material that must be melted and quenched to the amorphous state to completely block the current path through the device is minimized. In this way, current needed to WRITE the device is minimized, making the memory cell more efficient. In general PCM cell structures tend to fall in two categories: *contact-minimized* cells that control the cross-section by the size of one of the electrodes and *volume-minimized* or *confined* cells that minimize the volume of phase-change material itself within the cell. The most common contact-minimized cell design is the "mushroom" cell depicted in Fig. 3.2, in which the bottom electrode contact (often denoted "heater") is the smallest element in the cell. It is well known that confined cells generally achieve lower WRITE currents than contact-minimized cells for a given cross-sectional area, therefore significant

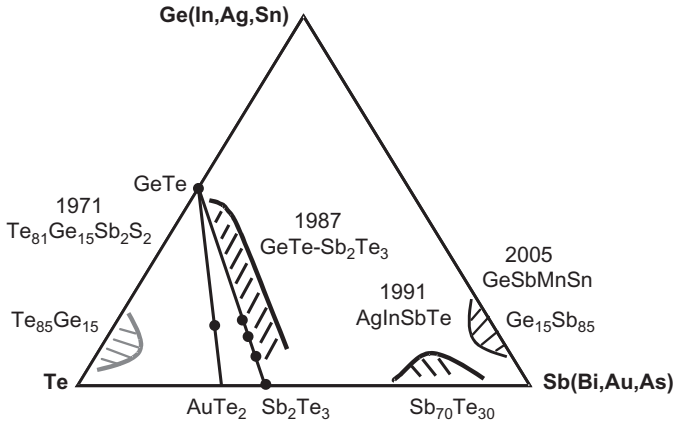


FIGURE 3.3 Most commonly used phase-change materials for PCM and optical storage. *Reprinted from M. Wuttig and N. Yamada, "Phase-change materials for rewriteable data storage," Nat. Mater., 6 (11) 2007, 824–832.*

research efforts have explored a variety of such cell structures. A common design is the "pillar" cell where a stack of phase-change material and top electrode material is patterned into sublithographic islands on a large bottom electrode [11]. Another similar design is the "pore" cell where a sublithographic hole is formed in an insulating material on top of the bottom electrode which is filled with phase-change material [11]. A different confined cell approach is the "bridge" cell, which consists of a narrow line of ultrathin phase-change material bridging two electrodes [19]. Other extensions of these concepts include the μ-trench cell and dash confined cell [11]. Another orthogonal type of memory cell design is interfacial PCM (iPCM), which uses a superlattice phase-change material stack formed by alternating two crystalline layers with different compositions [20]. It has been proposed that this superlattice stack switches between high- and low-resistance states without melting the material [21].

The key requirements for a PCM device to be used for electrical data storage is a high endurance (typically $>10^8$ SET/RESET cycles before failure), low RESET current ($\leq$200 μA highly desirable), fast SET speed ($\leq$100 ns), high retention (typically 10 years at 85 °C, but there are different requirements for embedded memories), good scalability (<45 nm node), and low intra- and inter-cell variability. While a single PCM device can be designed to easily meet one of the above constraints, the challenge is to build an array of devices that will meet all the above requirements. Individual PCM devices have demonstrated $>10^{12}$ endurance cycles, <10 μA RESET current, ~25 ns SET speed, projected 10 years retention at 210 °C, and sub-20 nm node scalability [22–24]. We refer the reader to recent reviews for the most up-to-date advances in PCM technology [24–26].

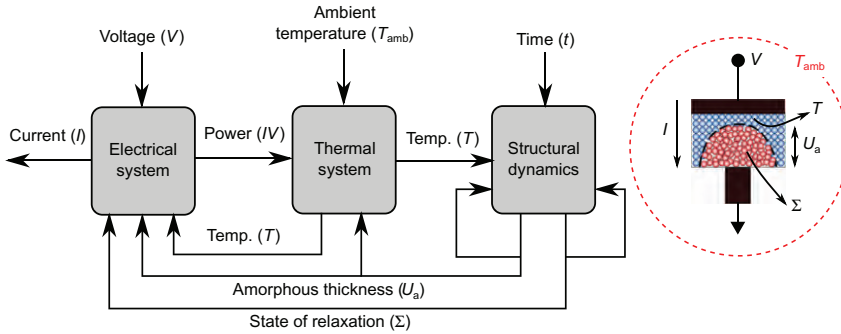


FIGURE 3.4 Block diagram of interconnections between electrical, thermal, and structural dynamics in a PCM device.

A PCM device has a rich body of dynamics which result from an intricate feedback interconnection of electrical, thermal, and structural dynamics. A block diagram illustrating the currently established dynamics in a PCM device is shown in Fig. 3.4. Electrical transport exhibits a strong voltage and temperature dependence. The output current I is influenced by the applied voltage V, the amorphous thickness u_a, which is used a measure of the size of the amorphous region, the temperature distribution within the device T, which is a function of three-dimensional coordinates, and the state of relaxation of the amorphous phase denoted Σ. The thermal system comprises all nanoscale thermal transport properties of the PCM device as well as significant thermoelectric effects [27]. The temperature distribution T in a PCM device is influenced by the electrical input power IV, the amorphous thickness u_a, and the ambient temperature T_{amb}. Lastly structural dynamics encompass what relates to crystallization/amorphization dynamics as well as structural relaxation. Crystallization is influenced by the amorphous thickness u_a, the temperature T, the time t and the state of relaxation Σ (through the viscosity). The state of relaxation Σ is mostly influenced by time t and temperature T with some possible dependence on u_a (a different u_a implies a different glass, which may lead to different relaxation properties).

3.3 A detailed description of the write operation

3.3.1 SET/RESET operation

The principles of crystallization and amorphization underlying the WRITE operation of PCM are illustrated in Fig. 3.5. To amorphize the phase-change material inside the PCM device (RESET), a high voltage or current pulse with sharp edges is applied. The resulting power dissipation must be high enough such that through Joule heating, the temperature within the PCM device reaches values above the melting temperature, T_{melt}, of the phase-

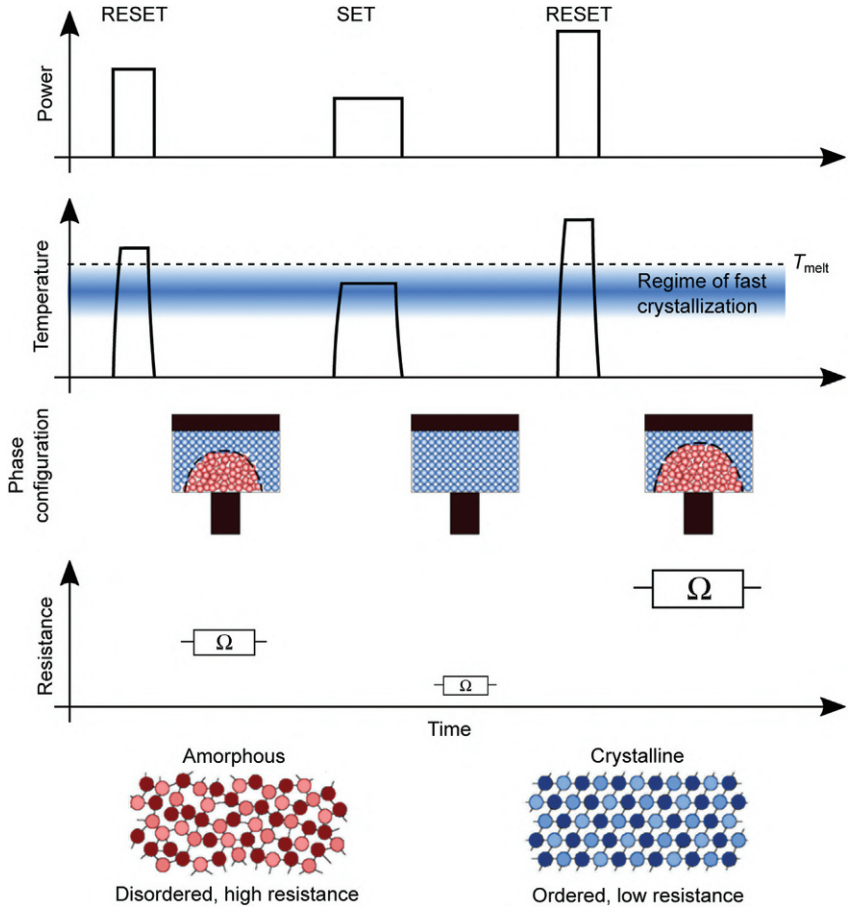


FIGURE 3.5 Principles of WRITE operation in PCM. A RESET brings the PCM device to a high-resistance state via amorphization of the phase-change material by heating above the melting temperature T_{melt} and subsequent rapid cooling of the material. A SET brings the PCM device to a low-resistance state via crystallization of a previously amorphous region. The size of the amorphous region can be modulated by changing the pulse power amplitude. *Modified from M. Salinga and M. Wuttig, "Phase-change memories on a diet,"* Science*, 332 (6029) 2011, 543–544.*

change material. The induced melting destroys any periodic atomic arrangement that was previously created. Once the phase-change material is molten, it must be rapidly cooled down (or *quenched*) in order to "freeze" the atomic structure into a disordered state. If the regime of fast crystallization (Fig. 3.5) is rapidly bypassed by fast quenching, the atomic mobility at temperatures below this regime becomes so small that the atoms cannot rearrange and find their most energetically favorable configuration during

cooldown, and thus are frozen into a nonequilibrium (or "glassy") amorphous state. This process is commonly referred to as *glass transition* and leads to the creation of the amorphous (high-resistance or RESET) state. The amorphization process can be as fast as a few tens of picoseconds, thanks to the fast melting kinetics of PCM [29], with the phase-change material typically melted to temperatures greater than $\sim$1000 K [30].

To switch from the amorphous to the crystalline state (SET), a voltage or current pulse is applied to bring the temperature within the PCM device to a temperature inside the regime of fast crystallization. Moreover the length of the pulse has to be long enough so that complete crystallization of any previously created amorphous region occurs. This process leads to the creation of a crystalline (low-resistance or SET) state. The crystallization process typically takes much longer than the amorphization process, typically tens to hundreds of nanoseconds, and the crystallization is realized at temperatures typically above $\sim$500–600 K but below T_{melt} [30].

The crystallization speed of PCM depends on the volume of initially amorphous material that is going to be crystallized and the crystallization kinetics of the phase-change material used, which are highly temperature dependent. The crystallization kinetics of PCM at elevated temperatures can be either nucleation or growth driven, and has been (and continues to be) a subject of intense research [30–35]. Nucleation is a stochastic process in which a crystalline nucleus eventually reaches a critical size beyond which it is stable, such that it can grow rather than dissolve. The build-up of the critical size nucleus requires an incubation time. The critical size depends on the temperature and is determined by the bulk free-energy difference between amorphous and crystalline phases (reduces the critical size when it increases) and the interfacial energy density between amorphous and crystalline phases (increases the critical size when it increases). Crystal growth occurs when the nucleus reaches the critical size, and is a deterministic process. The crystal growth velocity is highly temperature dependent and determined by the free-energy difference between liquid and crystalline phases (increases growth velocity when it increases) and the viscosity (decreases growth velocity when it increases). In conventional conditions such as those for optical disks, it has been shown that crystallization in AIST is growth driven (slow nucleation), and in GST it is nucleation driven (fast nucleation) [30]. However it has been argued that in nanoscale PCM devices, the role of nucleation may be less important and crystallization may be governed mostly by crystal growth [34,36]. This is because after RESET, a large population of nuclei already exists in the melt-quenched amorphous phase [36], and an amorphouscrystalline interface is present (Fig. 3.5). Therefore substantial crystal growth of the existing nuclei and at the amorphous–crystalline interface may be dominant over additional nucleation, even in nucleation-driven materials.

3.3.2 Switching process

In order for the above crystallization/amorphization scheme to be of practical use for electrical data storage with PCM, the ability to rapidly increase strongly the temperature within the device independent of the resistance state is needed. In optical storage, this is easily achieved by heating the phase-change material with a laser source of sufficient power regardless of the state of the material. In PCM a key property which enables fast substantial power dissipation by the application of a relatively low voltage pulse whose amplitude is mostly independent of the resistance state is a highly nonlinear current/voltage ($I-V$) characteristic. Typical $I-V$ characteristics of the amorphous and crystalline states are represented in Fig. 3.6. While the crystalline state has a fairly ohmic behavior at low voltages, the variation of the current with applied voltage in the amorphous state is highly nonlinear. In

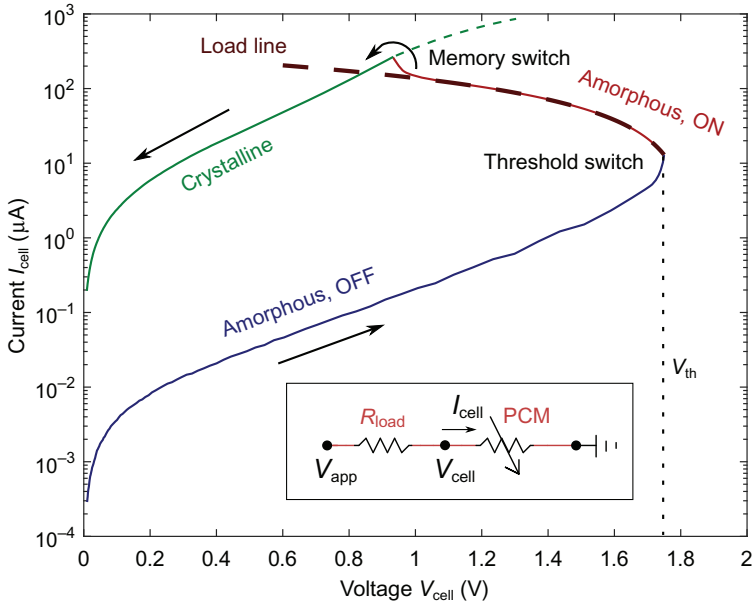


FIGURE 3.6 Quasi-static switching $I-V$ characteristic of a PCM device initially in the amorphous state measured in voltage mode (see inset). A triangular voltage ramp is applied to the PCM device in series with a load resistor $R_{load} \sim 5k\Omega$, and the voltage drop across R_{load} is subtracted from the applied voltage to obtain the PCM $I-V$ characteristic. Upon reaching V_{th}, threshold switching occurs and the current quickly increases, leading to a voltage snapback. The measured negative-differential resistance is that of R_{load} because the device resistance drops below R_{load} upon threshold switching. Memory switching (total crystallization) occurs when the amorphous ON state $I-V$ characteristic merges with that of the crystalline state. The dashed green line shows the continuation of the $I-V$ characteristic starting from the crystalline state when applying higher voltages, for which the phase-change material gets heated up to high temperatures and eventually melts.

the so-called amorphous OFF state (or subthreshold regime), the current shows an ohmic, exponential, and super-exponential behavior with increasing applied voltage. Beyond a certain voltage V_{th} called *threshold switching voltage*, the conductivity of the amorphous phase increases rapidly via a feedback-driven mechanism resulting in a negative-differential resistance (voltage *snapback*). If the device current is measured in voltage mode as shown in Fig. 3.6, the observed negative-differential resistance will typically be that of the load resistor R_{load} used in series to the PCM device to limit the current, because the PCM resistance typically decreases below R_{load} upon threshold switching. When PCM is operated in an array, the negative-differential resistance will be controlled by the nonlinear selector device or transistor in series with the PCM. The state reached upon threshold switching is typically called amorphous ON state, because the amorphous phase has not yet crystallized. Once sufficient current passes through the PCM device in the amorphous ON state for a sufficiently long time, memory switching (total crystallization) occurs and the amorphous ON state $I-V$ characteristic merges with that of the crystalline state.

The origin of the threshold switching mechanism in PCM is a long standing debate which is still not yet resolved despite the fact that the phenomenon was first observed more than 50 years ago by Ovshinsky [2,37,38]. A large number of models have been proposed to explain threshold switching in PCM [1], which can be broadly classified as either thermal (i.e., the switching is associated with an electrothermal instability occurring in the device) [39–45] or purely electronic [46–55].

Thermally initiated switching will occur when the temperature increase within the device due to Joule heating induces a significant conductivity increase due to thermal activation of carriers. A positive feedback loop will be established, resulting, as the conductivity increases, in increased power dissipation in the device which in turn will lead to a further increase of the conductivity. This can trigger the onset of an instability in this highly nonlinear feedback system, leading to a negative-differential $I-V$ characteristics. This electrothermal instability was the first mechanism proposed to explain threshold switching in phase-change materials [56], but it was mostly discarded in the 1980s in favor of an electronic excitation mechanism [48]. However thermally initiated switching was recently reconsidered when dealing with nanoscale PCM devices, in which self-heating effects were shown to play a significant role [44,45,57].

Other purely electronic mechanisms were proposed in the 1970–1980s to explain threshold switching in semiconducting glasses. The most notable ones are the double-injection model by Mott [46] and Henisch et al. [47] and the generation-recombination model of Adler et al. [48]. Most of the experimental work at that time was done on thin films, typically with large thermal time constants, and a debate over thermal versus electrical origin of threshold switching was settled mostly in favor of the latter [48]. In

the past 10 years, these electronic models have been revived and modified to explain data in nanometric PCM devices [49,50]. Moreover new models have been developed to explain threshold switching via a wide variety of different mechanisms, such as tunneling between trap states [51], energy gain via carrier temperature increase [52,53], field-induced nucleation [54,58], or quantum percolation [55].

The fact that so far no unique mechanism has been proven to quantitatively capture all commonly observed features of threshold switching in a unified way more than 50 years after the phenomenon was first reported in phase-change materials suggesting that many different mechanisms play a role. Depending on the device structures, functional materials, or switching conditions, some mechanisms might be more prominent than others, and understanding how they interact will likely yield significant insight. Further research in decoupling the thermal effects from the purely electronic ones in experiments is likely needed in order to make progress in this direction.

3.3.3 Multilevel operation

A key property of a PCM device is that the size of the amorphous region can be altered in an almost completely analog manner by the application of suitable electrical pulses. This is a consequence of the inhomogeneous temperature distribution within the PCM device. In the mushroom-type PCM device depicted in Fig. 3.5, the highest temperature reached through Joule heating resulting from an electrical pulse is typically close to the bottom electrode. Therefore by applying a RESET pulse which dissipates more power, a bigger amorphous region is created because T_{melt} is reached further away from the bottom electrode. This bigger amorphous region will result in a higher resistance of the PCM device. By exploiting this property one can therefore code more than 1 bit of information in a single PCM device because a continuum of resistance states can be achieved, each of which can represent a certain bitstream (e.g., “11” and “10”). One can also change the width of the pulse (for SET) or the length of its trailing edge to program multiple resistance levels. The map between PCM resistance and programming power is typically referred to as *programming curve*. One such typical programming curve obtained with a mushroom-type PCM device initially in the RESET state is shown in Fig. 3.7. The left part of the programming curve is unidirectional as it involves mostly amorphous-to-crystalline phase transition (e.g., it is not possible to have crystalline-to-amorphous phase transition in this part of the curve). The right part of the programming curve is mostly bidirectional, with the melt-quench process dominating the phase transition (e.g., both crystalline-to-amorphous and amorphous-to-crystalline phase transitions can be realized in this part of the curve). Reliable multilevel storage with PCM has been demonstrated for up to 3-bit (8 levels) per memory cell [59].

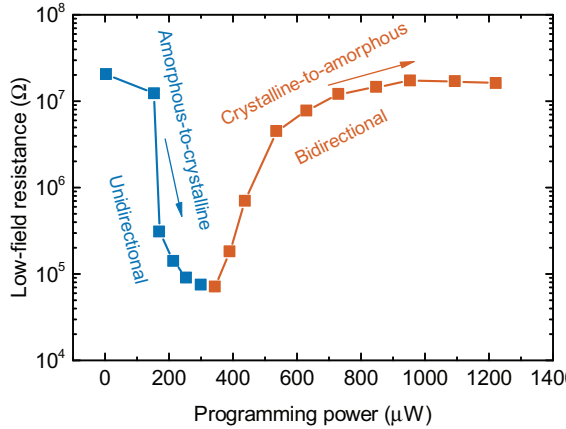


FIGURE 3.7 Low-field resistance as a function of the applied programming power (programming curve) for a PCM device initially in the RESET state. Box pulses of increasing power amplitude with 7.5 ns edges and 200 ns width are applied. In the left part of the programming curve, the initially created amorphous region progressively crystallizes until the low-field resistance reaches a minimum value. In the right part of the programming curve, an amorphous region of increasing size is formed, resulting in an increase of the resistance with increasing programming power.

3.4 A detailed description of the read operation

The READ operation in PCM typically consists in reading the resistance of the PCM device by the application of a low voltage pulse. The READ voltage has to be lower than the threshold switching voltage so that it does not perturb the state of the device. Typical $I-V$ characteristics of three different resistance states are shown in Fig. 3.8A, which indicate that the low-field resistance increases and the slope of $\log(I)$ vs. V decreases with increasing size of the amorphous region. However a key challenge while retrieving the stored information is the resistance variations with time and temperature. These resistance variations are caused mostly by the phase-change material in the amorphous phase. Typical low-field resistance measurements for different resistance states at room temperature are shown in Fig. 3.8B. It can be observed that the resistance increases over time, which is typically referred to as *resistance drift*. The resistance drift makes it difficult to detect the different resistance states of PCM over time reliably. The significant fluctuations of the resistance over time for the higher resistance states are also observed. This noise is arising from the amorphous phase and is another key challenge for multilevel storage. In the following sections, the PCM resistance dependence on voltage and temperature, resistance drift, and noise are described.

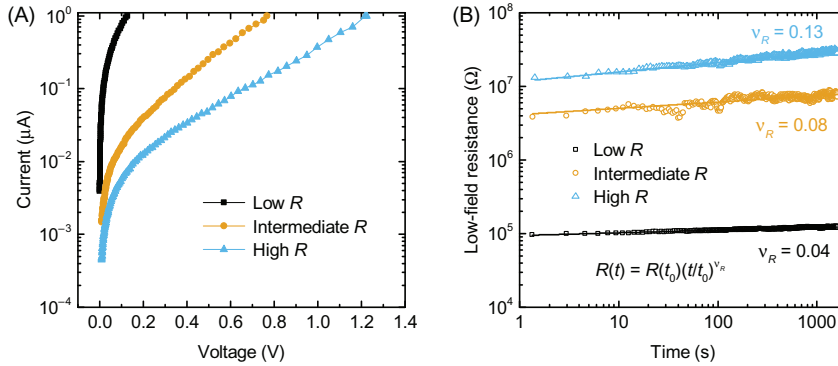


FIGURE 3.8 (A) $I-V$ characteristics of three different resistance states (low, intermediate, and high). The low-field resistance increases and the slope of $\log(I)$ versus V decreases with increasing size of the amorphous region. (B) Resistance as a function of time for three different resistance states (low, intermediate, and high).

3.4.1 Subthreshold electrical transport: voltage and temperature dependence

In disordered materials, electrical transport occurs either via localized states through quantum-mechanical tunneling or via extended states dominated by trapping and release events (trap-limited band transport or multiple-trapping) [60]. In several amorphous phase-change materials, it has been shown that multiple-trapping can successfully describe the low-field conductivity measurements at temperatures above approximately 200 K, whereas at lower temperatures tunneling in localized states dominates transport [61,62]. This is mainly motivated by the fact that in most of the commonly used amorphous phase-change materials, the activation energy for conduction at room temperature and above is close to half of the optical bandgap [63,64]. In PCM operated at room temperature and above, the low-field resistance R measured in the ohmic region of the $I-V$ characteristic can be thus described by

$$R = R_0 \exp\left(\frac{E_a}{k_B T}\right), \tag{3.1}$$

where E_a is the activation energy for conduction (the energy distance between the Fermi level and the mobility edge), k_B is the Boltzmann constant, and T is the temperature. E_a is typically in the range of 0.2−0.4 eV for amorphous PCM and has a slight dependence on the temperature because of the temperature dependence of the optical bandgap of phase-change materials [62,65].

To explain the variation of conductivity with the electric field in the multiple-trapping picture for disordered materials, the Poole−Frenkel effect

is commonly used [66–68]. Thismodel is based on thermal emission from ionizable defect centers that are assumed to create a Coulomb potential. The ionization energy is then lowered upon the electric field by $\beta F^{1/2}$ with $\beta = e^2/\sqrt{e\pi\varepsilon_r\varepsilon_0}$, where F is the applied electric field, e the electronic charge, ε_0 the vacuum permittivity, and ε_r the relative high-frequency dielectric constant. The conductivity is expected to follow a law (Poole–Frenkel) of the form $\sigma_{\mathrm{PF}}(F) = \sigma_0^{PF}\exp(\beta F^{1/2}/k_BT)$. When the defect centers are close to each other so that there is significant overlap between the Coulomb potentials, it has been shown by using a two-center Coulomb potential that the ionization energy lowering upon field is $eFs/2$ [67]. The conductivity is then expected to follow a law (Poole) of the form $\sigma_{\mathrm{P}}(F) = \sigma_0^{P}\exp(eFs/2k_BT)$, where s is the distance between the two centers.

One of the first studies of electrical transport in nanoscale PCM devices was by Ielmini and Zhang where they mostly observed an ohmic regime at low fields and Poole-type behavior at higher fields [51]. However experimental measurements since then clearly showed the existence of three distinct regimes, an ohmic regime, a Poole regime, and a Poole–Frenkel regime [69,70]. The ohmic regime occurs at very low fields and the transition from Poole to Poole–Frenkel conduction occurs at high fields [71]. A unified model based on multiple-trapping transport together with 3D Poole–Frenkel emission from a two-center Coulomb potential was shown to capture experimental data both in as-deposited phase-change material thin films and nanoscale PCM devices over a wide range of temperatures and applied voltages [72,73]. Experimental measurements of the resistance versus applied voltage of a PCM device in the RESET state at different ambient temperatures along with a simulation using the model of [72] are shown in Fig. 3.9.

3.4.2 Resistance drift

At constant ambient temperature, the low-field resistance of PCM typically exhibits a temporal dependence characterized by

$$R(t) = R(t_0)(t/t_0)^{\nu_R}, \tag{3.2}$$

where $R(t_0)$ is the resistance measured at time t_0. The drift exponent ν_R, which has a typical value of 0.1 for the RESET state, exhibits significant inter and intradevice variability. This drift variability is arguably the most significant challenge for multilevel storage in PCM.

Resistance drift in PCM devices has been explained as a consequence of spontaneous structural relaxation of the amorphous phase-change material [74–78]. This structural relaxation is a direct consequence of the amorphization process described in Section 3.3.1. When the molten phase-change material is quenched rapidly, the atomic configurations are frozen into a highly stressed glass state. Over time the atomic configuration of this state will

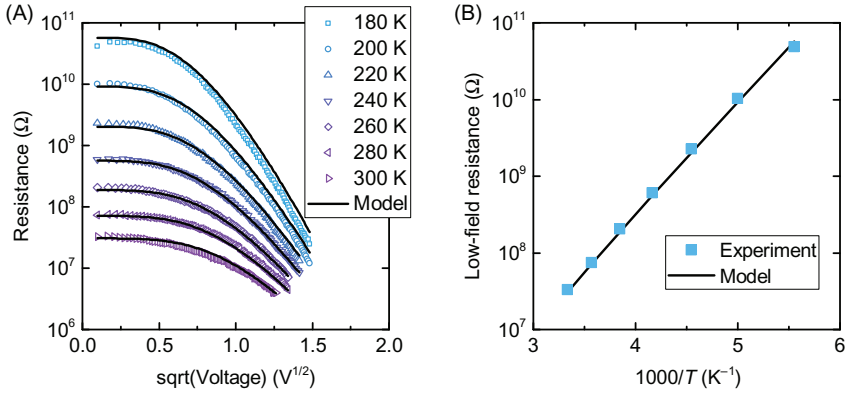


FIGURE 3.9 (A) Resistance of a PCM device in the RESET state as a function of voltage at different ambient temperatures. The device was RESET and annealed at room temperature for 1000 s prior to measurement. (B) Low-field resistance R of the device as a function of the temperature. R follows the Arrhenius-type behavior of Eq. (3.1). *Adapted from M. Le Gallo, M. Kaes, A. Sebastian, and D. Krebs, "Subthreshold electrical transport in amorphous phase-change materials," N. J. Phys., 17 (9) 2015, 093035.*

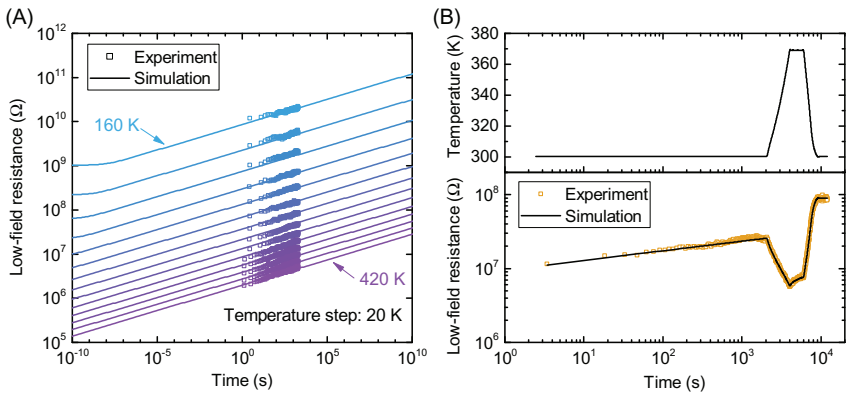


FIGURE 3.10 (A) Low-field resistance drift of a PCM device in the RESET state at various ambient temperatures. (B) Low-field resistance drift of a PCM device in the RESET state (bottom) upon application of a time-varying temperature profile (top). *Adapted from M. Le Gallo, D. Krebs, F. Zipoli, M. Salinga, and A. Sebastian, "Collective structural relaxation in phase-change memory devices,"* Adv. Electron. Mater., *4 (9) 2018, 1700627.*

relax toward an energetically more favorable "ideal glass" configuration. The observed increase in resistance has been shown to be a consequence of the atomic rearrangements resulting from this evolution [79–82].

Experimental measurements of constant temperature low-field resistance drift over a wide range of temperatures are shown in Fig. 3.10A. After setting the temperature to a certain value, the PCM device is RESET and the

evolution of the low-field resistance R is monitored. It can be observed that the slope of $\log(R)$ versus $\log(t)$ is temperature independent in the experimentally accessible range of time [79,83,84]. However when the ambient temperature is varied during the resistance measurement, reversible as well as irreversible effects of temperature on electrical transport occur upon drift, because structural relaxation is accelerated at higher temperatures [79,85,86]. An experiment showing the low-field resistance variation of a PCM device after RESET during the application of a time-varying temperature profile is presented in Fig. 3.10B. When the temperature increases above room temperature, the relaxation is accelerated. Therefore when the device is brought back to room temperature, its resistance becomes higher than if it would have stayed at room temperature for the entire duration of the experiment, and it stops increasing because of the preceding annealing at higher temperature.

Several approaches have been tried in order to counter the effect of resistance drift to retrieve the stored information in a PCM device. One approach is to take advantage of the nonlinearity of the $I-V$ characteristic of the amorphous state (Fig. 3.8A) and obtain a better measure of the phase configuration, which is drift-invariant, by measuring the resistance in the high-field regime [87]. As seen in Fig. 3.8A, the slope of $\log(I)$ vs. V at high V can be used as a measure of the size of the amorphous region [72] and depends only weakly on drift compared to the low-field resistance. In the absence of a priori knowledge of the programmed state, the only way to explore the high-field regime of every programmed state is by applying a varying read voltage and then detecting the voltage or time at which a certain current threshold (I_t) is reached. This voltage or time value (typically referred to as the M-metric) is used as the measure of the programmed state. It has been shown that the effect of drift can be significantly mitigated by using the M-metric [87,88].

Another fascinating approach to eliminate the effect of resistance drift upon READ is building a so-called projected PCM device [89]. This device comprises a carefully designed segment consisting of a noninsulating material (projection segment) that is parallel to the phase-change segment. The resistance of this projection segment is judiciously chosen such that it has only a marginal influence on the WRITE operation, but a significant influence on the READ operation. This is indeed possible because of the highly nonlinear nature of the electrical transport of the amorphous phase. The idea is that during WRITE the current flows through the phase-change segment because the resistance of the amorphous ON state is lower than the resistance of the projection segment. However, during READ, the current flows through the projection segment because it has a lower resistance than the amorphous OFF state. In this way, information retrieval is completely decoupled from information storage, and all the undesirable properties of the amorphous phase such as resistance drift, temperature dependence, and noise are hidden

upon READ. This approach has been shown to reduce the drift exponent by almost two orders of magnitude, and practically eliminate the READ current noise and temperature dependence [89].

3.4.3 Noise

The most commonly observed type of noise in PCM is referred to as $1/f$ noise (or flicker noise), which is a type of noise frequently observed in electronic devices [90]. The $1/f$ noise is characterized by a power spectral density inversely proportional to the frequency of the signal. $1/f$ noise in nanoscale PCM was first measured in [91], where the normalized current spectral density S_I/I^2 of the amorphous state was reported to be two orders of magnitude higher than the crystalline state in GST. Later measurements showed that S_I/I^2 remains relatively constant with respect to the applied voltage for low enough voltages [92]. Experimental spectra of S_I/I^2 for crystalline and amorphous states measured in nanoscale GeTe line cells are shown in Fig. 3.11. A $1/f$ frequency dependence is observed for the amorphous state from 1 Hz to 100 kHz, and S_I/I^2 is roughly 10^5 times lower for the crystalline state. Besides $1/f$ noise, random telegraph noise (RTN) is also typically observed in intermediate resistance states in PCM [94]. The current makes sharp transitions between two levels at random times and the fluctuation amplitude can sometimes be quite large, resulting often in a larger normalized variance than current signals of the RESET state.

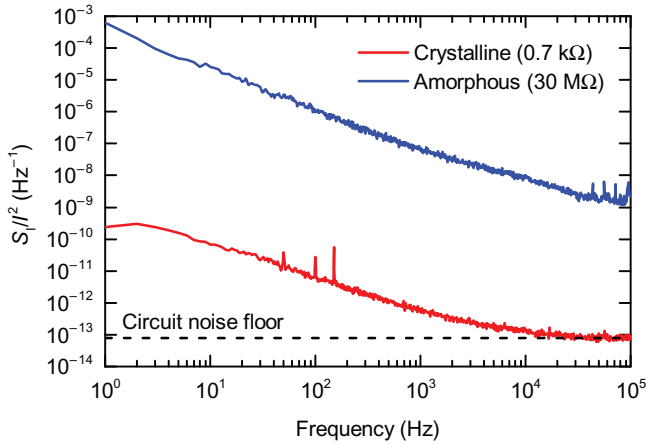


FIGURE 3.11 Normalized current spectral density S_I/I^2 of amorphous and crystalline states in a nanoscale GeTe line cell. The applied voltage on the device is 59 mV. *Adapted from I. Giannopoulos, A. Sebastian, M.L. Gallo, V.P. Jonnalagadda, M. Sousa, M.N. Boon, et al., "8-bit precision in-memory multiplication with projected phase-change memory," in* IEEE International Electron Devices Meeting (IEDM), *Dec 2018, pp. 27.7.1–27.7.4.*

No unique model has been established for explaining the origin of $1/f$ noise in PCM. Few models have been proposed mainly based on the concept of double-well potentials (DWPs) [95,96], in which either atoms or electrons switch between two energy minima separated by a potential barrier W, creating fluctuations. The general approach to arrive at a spectrum $S(f) \propto 1/f$ is to assume that there are many fluctuation events, each with relaxation time $\tau = \tau_0 \exp(W/k_B T)$, where τ_0^{-1} is the attempt frequency to surpass the barrier W. If it is then assumed that W is distributed uniformly, this approach yields a $1/f$ spectrum [95]. So in principle, any system that has local bistable configurations with an exponentially broad distribution of relaxation times would exhibit $1/f$ noise. The source of $1/f$ noise in the bulk electrical resistance can be related to charge carrier mobility or concentration fluctuations due to transitions in the DWPs [95]. In order to elucidate the precise underlying mechanisms in PCM, measurements of the temperature dependence of $1/f$ noise and its high-field non-Ohmic regime, in particular, would be required [95].

3.5 Key enablers for brain-inspired computing

3.5.1 Multilevel storage

The first key property of PCM that enables brain-inspired computing is multilevel storage, that is the ability to store a continuum of conductance values in a single PCM device as discussed in Section 3.3. This enables the ability to perform an analog in-memory dot-product, or more generally a matrix–vector multiplication, where the matrix elements are stored as the conductance values of PCM devices organized in a crossbar configuration. As shown in Fig. 3.12, to perform $Ax = b$, the elements of A should be mapped linearly to the conductance values of PCM devices, and the x values are encoded into the amplitudes or durations of read voltage pulses applied

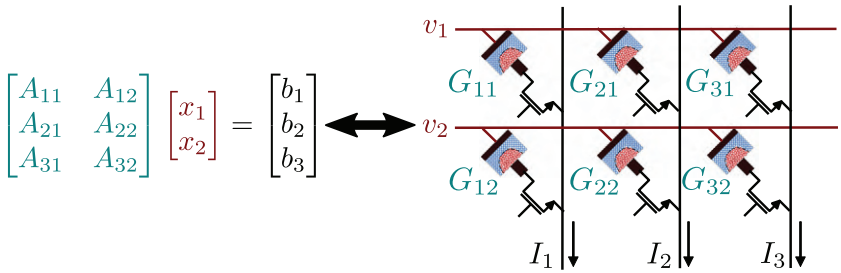


FIGURE 3.12 Crossbar array of phase-change memory (PCM) devices used to perform an analog matrix–vector multiplication $Ax = b$. Matrix A is mapped to the conductance values G of PCM devices organized in a crossbar configuration. Input vector x is mapped to read voltages v applied on the rows of the crossbar. The result vector b is deciphered from the column currents I.

along the rows. The positive and negative elements of A could be coded on separate devices together with a subtraction circuit, and negative vector elements could be applied as negative voltages. The resulting currents along the columns will be proportional to the result b. If inputs are encoded into durations, the result b is the total charge (e.g., current integrated over time). The multiplication operation is performed at every crosspoint by Ohm's law, with current summation along rows or columns performed by Kirchhoff's current law. Thus these multiplication−accumulate operations can be performed in parallel at the location of data with locally analog computing, avoiding the time and energy of moving the matrix data. The same crossbar configuration can be used to perform a matrix−vector multiplication with the transpose of A. For this the input voltage has to be applied to the column lines and the resulting current has to be measured along the rows.

Mapping of the matrix elements to the conductance values of the PCM devices can be achieved via iterative programming using the right side of the programming curve (crystalline-to-amorphous transition, see Fig. 3.7). In iterative programming, after each programming pulse, a verify step is performed, and the value of the device conductance programmed in the previous iteration is read. The programming current applied to the PCM device in the subsequent iteration is adapted according to the sign of the value of the error between the target level and the read value of the device conductance. The algorithm runs until the programmed conductance reaches a value within a predefined margin from the target value. Fig. 3.13 shows experimental results of the iterative programming of five representative conductance levels of 5,000 devices from a prototype multilevel PCM chip fabricated in the 90 nm technology node [97]. Using a conductance margin of 1.74μS, the algorithm converged on 100% of the devices under test in less than 20 iterations (see Fig. 3.13a), and the resulting conductance distributions 50μs after programming have an average standard deviation of 1.4μS (see Fig. 3.13b). However, due to the effect of drift (see Section 3.4), the conductance values decrease away from the as-programmed conductance over time (see Fig. 3.13c). This implies that additional compensations schemes to cope with the effect of drift may need to be used in conjunction with the analog computation from the crossbar array to perform accurate matrix−vector multiplications over time with PCM [97].

Fig. 3.14 shows the experimental result of a matrix−vector multiplication using the prototype PCM chip, where each matrix element is programmed on four PCM devices averaged using iterative programming. Matrix A is a 256x256 Gaussian matrix coded in a PCM chip and x is a 256-long Gaussian vector applied as voltages to the devices. It can be seen that the matrix−vector multiplication has a precision comparable to that of a fixed-point implementation where the matrix and vector elements are quantized to 4 bits. This precision is mostly determined by the conductance fluctuations discussed in Section 3.4.

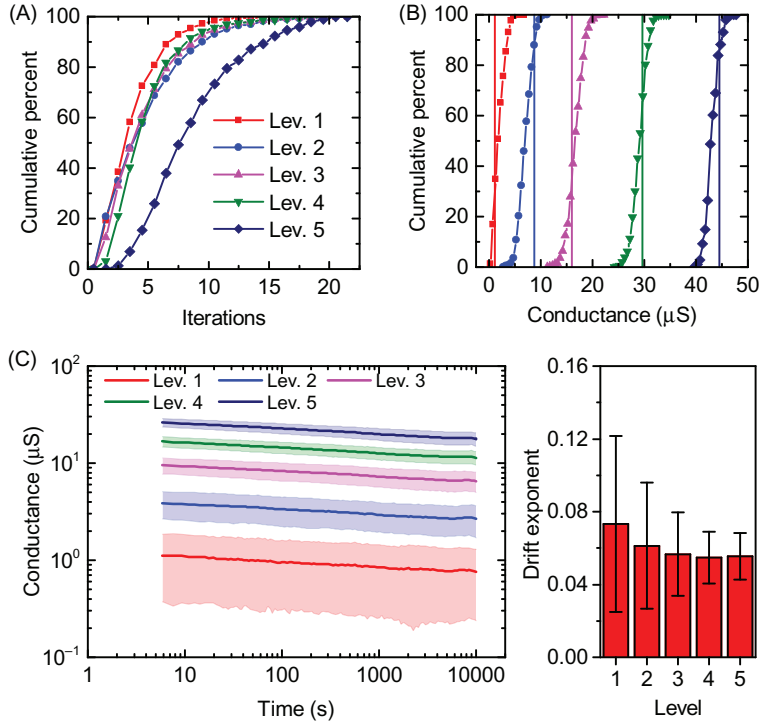


FIGURE 3.13 Iterative programming of five representative conductance levels (vertical lines in B) on 5000 devices of a PCM chip. (A) Number of iterations needed for convergence of the iterative programming algorithm. (B) Conductance distributions at approximately 50 μs after programming. (C) Evolution of the mean conductance values of the five programmed levels versus time; filled areas represent the standard deviation for each level; the plot on the right shows the calculated drift exponent ν of the five levels computed from $G(t) = G(t_0)(t/t_0)^{-\nu}$. *Adapted from M.L. Gallo, A. Sebastian, G. Cherubini, H. Giefers and E. Eleftheriou, Compressed sensing with approximate message passing using in-memory computing, IEEE Trans. Electron. Devices 65, Oct 2018, 4304–4312.*

3.5.2 Accumulative behavior

The second key property that enables brain-inspired computing is the accumulative behavior arising from the crystallization dynamics. By the successive application of SET pulses with the same amplitude, one can induce progressive reduction in the size of the amorphous region (and hence the device resistance). This accumulation property (the PCM in fact integrates the electric current flowing through it) is essential for emulating synaptic dynamics [98] and can also be used to implement some arithmetic operations [99].

Fig. 3.15 shows experimental measurements of this accumulation property across an array of PCM devices [100]. In the experiment, 10,000 devices

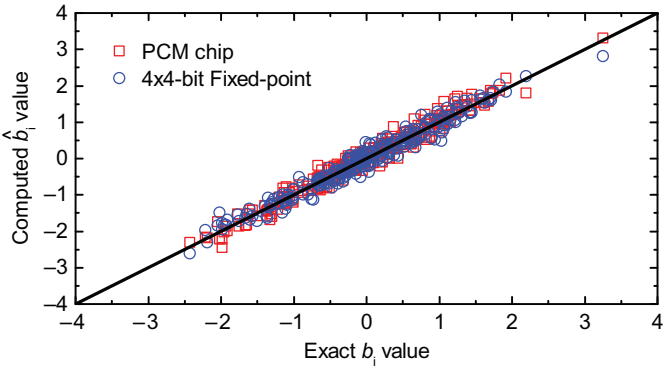


FIGURE 3.14 Comparison of the precision in the computation of $Ax = b$ by the experimental PCM chip and 4×4-bit multiplications. A is a 256×256 Gaussian matrix coded in the PCM chip, x is a 256-long Gaussian vector applied as voltages, and b_i is the i-th element of b. *Adapted from M.L. Gallo, A. Sebastian, G. Cherubini, H. Giefers, and E. Eleftheriou, "Compressed sensing with approximate message passing using in-memory computing,"* IEEE Trans. Electron. Devices*, 65, Oct 2018, pp. 4304–4312.*

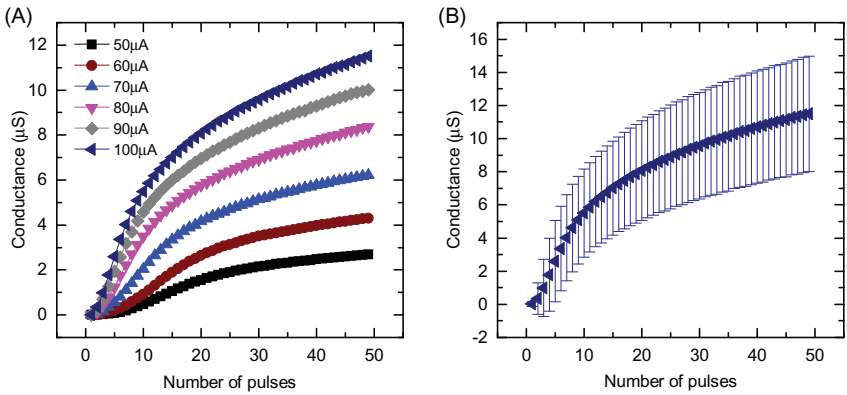


FIGURE 3.15 (A) The mean accumulation curve of 10,000 devices showing the map between the device conductance and the number of pulses. The devices achieve a higher conductance value with increasing SET current and also with increasing number of pulses. (B) The mean and standard deviation associated with the accumulation curve corresponding to the SET current of 100 μA. *Adapted from A. Sebastian, T. Tuma, N. Papandreou, M. Le Gallo, L. Kull, T. Parnell, et al., "Temporal correlation detection using computational phase-change memory,"* Nat. Commun.*, 8, 2017, p. 1115.*

were arbitrarily chosen and were first RESET by applying a rectangular current pulse of 1 μs duration and 440 μA amplitude. After RESET, a sequence of SET pulses of 50 ns duration were applied to all devices, and the resulting device conductance values were monitored after the application of each pulse. The map between the device conductance and the number of pulses is

referred as accumulation curve. The accumulation curves corresponding to different SET currents are shown in Fig. 3.15A. These results clearly show that the mean conductance increases monotonically with increasing SET current (in the range from 50 and 100 μA) and with increasing number of SET pulses. Moreover the increase in conductance as a function of the number of SET pulses is highly nonlinear, owing to the nonlinear temperature dependence of the PCM crystallization dynamics and the inhomogeneous temperature distribution in the device upon the application of the SET pulses [34]. From Fig. 3.15B, it can also be seen that a significant variability is associated with the evolution of the device conductance values. This variability arises from interdevice and intradevice variabilities (see Section 3.5.3). Besides the variability arising from the crystallization process, additional fluctuations in conductance also arise from $1/f$ noise and drift variability.

Although the accumulation property in PCM is certainly not ideal due to the high nonlinearity and stochasticity of the conductance response, it has been nonetheless applied successfully for simple arithmetic operations such as finding the factors of numbers [101], or more involved tasks such as detecting temporal correlations in binary data streams [100]. The accumulation property can also be used for the task of training artificial neural networks, where the synaptic weights are represented by the conductance values of PCM devices organized in crossbar arrays [102,103]. The synaptic weights can be updated in situ by applying suitable SET pulses to PCM devices organized in a differential configuration. For instance, one PCM device can encode the positive part of a synaptic weight, another PCM device the negative part, and the weight can be increased (decreased) by applying SET pulses to the positive (negative) device [104]. Alternatively a bidirectional PCM synaptic device has been recently demonstrated, by narrowing down the bottom electrode along one direction to four nanometers combined with a special device initialization procedure to enable partial amorphization with fast identical RESET pulses [105]. However further work in reducing the nonlinearity and stochasticity of the conductance response will be needed to achieve accurate neural network training with this approach using PCM.

3.5.3 Inter and intradevice randomness

A third property of PCM that can be harnessed for brain-inspired computing is the inter and intradevice randomness. Here we focus on the stochasticity of the PCM switching process, namely, the threshold switching and the crystallization process. The native switching stochasticity of PCM can be exploited, in particular, for population coding in spiking neural networks [106], or for random number generation for stochastic computing or cryptography [107,108]. The essential property used in all these applications is the fact that, when applying a particular pulse to an array of PCM devices, the

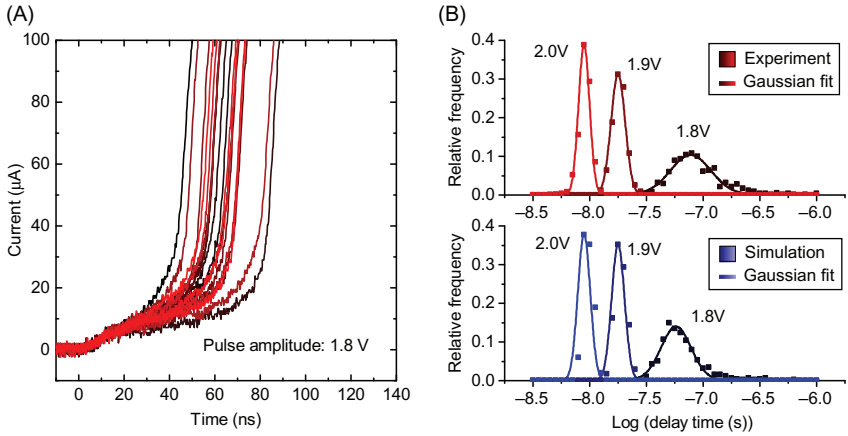


FIGURE 3.16 (A) Experimentally measured current traces for a delay time experiment with voltage pulse amplitude of 1.8 V repeated 20 times. The PCM device is RESET in between each delay time measurement. (B) Experimental and simulated delay time distributions from 500 measurements for three different applied voltages. *Adapted from M. Le Gallo, T. Tuma, F. Zipoli, A. Sebastian and E. Eleftheriou, Inherent stochasticity in phase change memory devices, Proceedings of the European Solid-State Device Research Conference (ESSDERC), 2016, IEEE, 373–376.*

devices will switch with a certain probability P ($0 \leq P \leq 1$), and P can be modulated by changing either the pulse amplitude or the pulse width.

Fig. 3.16A shows representative measurements of the stochasticity of the threshold switching delay time on a single PCM device [107]. This delay time represents the time it takes for the current to rise steeply after the application of a voltage pulse. Therefore, the PCM will switch only if the width of the applied voltage pulse is greater than the delay time. In the experiment, after each RESET operation, a voltage pulse with amplitude slightly above the steady-state threshold switching voltage is applied to the PCM device. It can be clearly seen that each experiment results in a different current trace and thus a different delay time. For a more detailed characterization of this randomness, we obtained delay time measurements 500 times for three pulse amplitudes of 1.8, 1.9 and 2 V. The results are shown in Fig. 3.16B. The delay time random variable was found to follow roughly a log-normal distribution in all three cases. A simulation using the model described in [45] was able to capture well the experimentally measured distributions by introducing a small (0.5%) randomness in the amorphous thickness and activation energy of the device after RESET. It indicates that the stochasticity observed in the threshold switching process can be explained by variations in the atomic configurations of the amorphous phase created upon each RESET process. In order to use the threshold switching stochasticity in practice for probabilistic computing, one can tune both the pulse width and pulse amplitude to make the device switch with a given probability p.

A second source of stochasticity in the PCM switching process arises from the crystallization process. In the nanoscale mushroom-type PCM device depicted in Fig. 3.2, the crystallization mechanism is assumed to be mainly dominated by crystal growth due to the large amorphous–crystalline interface area and small volume of the amorphous region. Moreover for large enough pulse amplitudes, the temperature distribution reached within the device when a voltage pulse applied also favors crystal growth at the amorphous–crystalline interface. Although crystal growth is a deterministic process, small variations in the atomic configurations of the amorphous volume created upon RESET can lead to variations in the effective amorphous thickness initially created. This in turn leads to a stochastic behavior of the crystallization time of a PCM device. An experiment that measures the stochasticity in the PCM crystallization time is shown in Fig. 3.17 [107]. In the experiment, a PCM device is first RESET and then a sequence of SET pulses are applied to the device. The amplitude of the crystallizing pulses is substantially larger than the threshold switching voltage to avoid any delay time stochasticity as well as to provide sufficient current to induce Joule heating and crystal growth. After the application of each crystallizing pulse, the low-field electrical resistance is measured. Experimentally measured traces of the resistance as a function of the number of crystallizing pulses for a constant pulse width of 50 ns are shown in Fig. 3.17A. It can be seen that the resistance decreases incrementally upon the application of the pulses until it reaches its lowest value when the whole amorphous region has

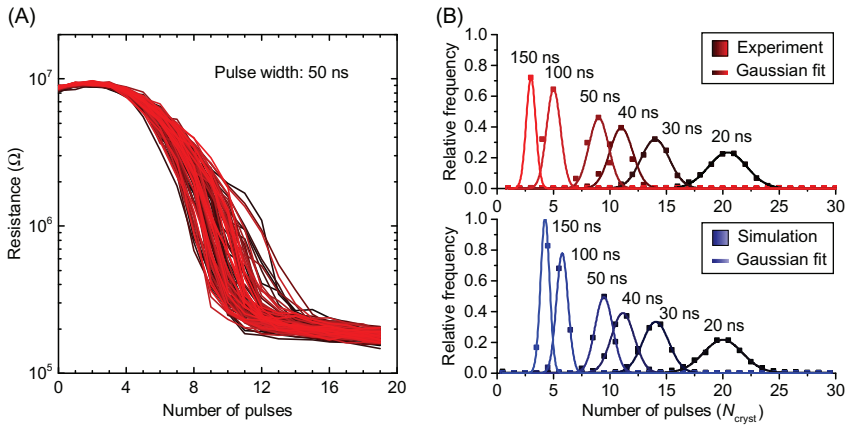


FIGURE 3.17 **(A)**, Experimentally measured resistance as a function of the number of crystallization pulses applied for a fixed pulse width of 50 ns. The experiment was repeated 100 times and the device was RESET in between each experiment. **(B)**, Experimental and simulated N_{cryst} distributions from 1000 measurements for six different pulse widths. *Adapted from M. Le Gallo, T. Tuma, F. Zipoli, A. Sebastian, and E. Eleftheriou, "Inherent stochasticity in phase change memory devices," in* Proceedings of the European Solid-State Device Research Conference (ESSDERC), *IEEE, 2016, pp. 373–376.*

crystallized. Moreover the resistance trajectories are different in each experiment, leading to a randomness in the total number of pulses needed to fully crystallize. In Fig. 3.17B we report the distributions of the number of pulses to crystallize N_{cryst} for different pulse widths. As for threshold switching, a simulation was able to capture well the experimentally measured distributions by introducing a 0.5% randomness in the amorphous thickness, using the model presented in Ref. [34] to capture the crystallization dynamics in the PCM device. Based on these distributions, the number of pulses N_{cryst} can therefore be adapted such that the device will switch with a given probability P for a certain pulse width.

An important question that arises when using the PCM switching stochasticity for practical applications across an array is how much the interdevice randomness is compared with the intradevice randomness investigated in the above paragraphs. An experiment that was designed to shed light into this matter is shown in Fig. 3.18 [103]. To capture the intradevice randomness, a single PCM device was first RESET, and then a train of SET pulses of 100 μA current and 50 ns width was applied to it. The change in device conductance between the application of the third and fourth pulse was measured. This experiment was repeated 1000 times on the same device and the resulting distribution of conductance change is reported in Fig. 3.18A. To capture the interdevice randomness, 1000 PCM devices were subject to the same RESET and pulse train, and the change in device conductance between the application of the third and fourth pulse was measured across the 1000 devices. The resulting distribution of conductance change is reported in Fig. 3.18B. It can be observed that the mean and standard deviation of the distributions resulting from the two experiments are similar. It indicates that, at least in this particular experiment, the intra and interdevice randomness

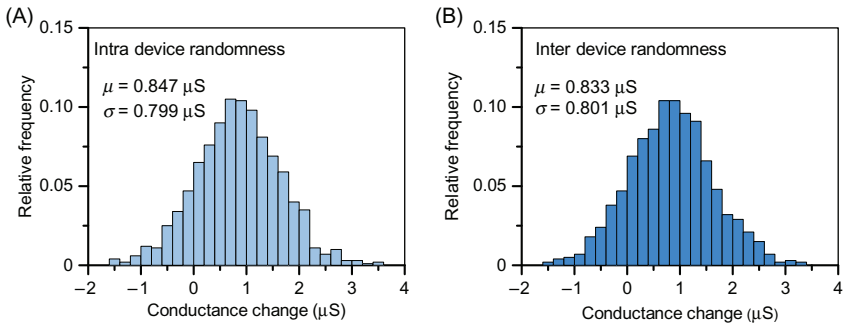


FIGURE 3.18 (A) Representative distribution of the conductance change induced by a single pulse applied on the same PCM device 1000 times. (B) Representative distribution of the conductance change induced by a single pulse on 1000 devices. The negative conductance changes are attributed to drift variability. *Adapted from I. Boybat, M. Le Gallo, S.R. Nandakumar, T. Moraitis, T. Parnell, T. Tuma, et al., "Neuromorphic computing with multi-memristive synapses,"* Nat. Commun., *9, 2018, p. 2514.*

are comparable. Thus it means that the intradevice stochasticity is dominating over the interdevice stochasticity when performing such experiments on these specific PCM devices. Nonetheless we expect that the intra and interdevice randomness highly depends on the technology node and fabrication process, thus different experiments on different device technologies may result in diverse conclusions regarding the role of intra and interdevice variations.

3.6 Outlook

PCM is arguably the most mature resistive memory technology as of today, because the materials have been extensively studied and mass produced, for example, in DVDs and Blu-Ray disks, and it has already appeared as a digital memory product on the market (Intel Optane). Its attractive properties such as multilevel storage, fast read/write latency, nonvolatility, good cycling endurance, and good scalability make it an ideal candidate to be envisaged for applications in novel computing paradigms. However there are also numerous roadblocks associated with using PCM devices for computational purposes. One key challenge applicable to almost all the applications in brain-inspired computing is the variation in conductance values arising from $1/f$ noise as well as structural relaxation of the melt-quenched amorphous phase. There are also temperature-induced conductance variations. One promising research avenue toward addressing this challenge is that of projected PCM [89]. Another challenge is the limited endurance of PCM devices, which, while being relatively high ($\sim 10^9 - 10^{12}$), may not be adequate for certain computational applications. The nonlinearity and stochasticity associated with the accumulative behavior are key challenges, in particular, for applications involving in situ supervised learning with backpropagation. Multi-PCM architectures could partially address these challenges [103]. However more research in terms of device geometries and randomness associated with crystal growth is required. Alternatively one may attempt to adapt the learning algorithms in order to take advantage of the device nonidealities, rather than trying to overcome them. In fact, it has been shown that the nonlinearity and stochasticity can actually be beneficial in applications involving unsupervised learning, as they can help to stabilize and regularize the network during learning [103,105,109,110]. Nonetheless system-level studies show that even with today's PCM technology including all its nonidealities, higher performance could be achieved compared with general purpose computing approaches for certain computational tasks [97,111]. Therefore the application of PCM for brain-inspired computing remains to be a potentially attractive solution for building energy-efficient and highly parallel non-von Neumann computing systems.

References

[1] N. Bogoslovskiy, K. Tsendin, Physics of switching and memory effects in chalcogenide glassy semiconductors, Semiconductors 46 (5) (2012) 559–590.

[2] S.R. Ovshinsky, Reversible electrical switching phenomena in disordered structures, Phys. Rev. Lett. 21 (20) (1968) 1450–1453.

[3] S. Ovshinsky, An introduction to ovonic research, J. Non-Crystalline Solids 2 (1970) 99–106.

[4] R. Neale, J.A. Aseltine, The application of amorphous materials to computer memories, IEEE Trans. Electron. Devices 20 (2) (1973) 195–205.

[5] M. Wuttig, N. Yamada, Phase-change materials for rewriteable data storage, Nat. Mater. 6 (11) (2007) 824–832.

[6] Y. Choi, I. Song, M.-H. Park, H. Chung, S. Chang, B. Cho, et al., A 20nm 1.8 V 8Gb PRAM with 40MB/s program bandwidth, International Solid-State Circuits Conference Digest of Technical Papers (ISSCC), IEEE, 2012, pp. 46–48.

[7] Intel, "Intel, STMicroelectronics deliver industry's first phase change memory prototypes." https://phys.org/news/2008-02-intel-stmicroelectronics-industryphase-memory.html, 2008.

[8] P. Clarke, "Exclusive: Micron drops phase-change memory – for now." http://electronics360.globalspec.com/article/3931/exclusive-micron-dropsphase-change-memory-for-now, 2014.

[9] P. Clarke, "Patent search supports view 3D XPoint based on phase-change." http://www.eetimes.com/author.asp?section_id = 36&doc_id = 1327313, 2015.

[10] B. Tallis, "Intel announces optane memory M15: 3D XPoint On M.2 PCIe 3.0 x4." https://www.anandtech.com/show/14437/intel-announces-optanememory-m15-3d-xpoint-on-m2-pcie-30-x4, 2019.

[11] G. Burr, M. Breitwisch, M. Franceschini, D. Garetto, K. Gopalakrishnan, B. Jackson, et al., Phase change memory technology, J. Vac. Sci. Technol. B 28 (2010) 223.

[12] B.C. Lee, E. Ipek, O. Mutlu, and D. Burger, "Architecting phase change memory as a scalable dram alternative," in *International Symposium on Computer Architecture (ISCA)*, June 2009, 2–13.

[13] M.K. Qureshi, V. Srinivasan, J.A. Rivers, Scalable high performance main memory system using phase-change memory technology, SIGARCH Computer Architecture N. 37 (3) (2009) 24–33.

[14] A.P. Ferreira, M. Zhou, S. Bock, B. Childers, R. Melhem, D. Mosse, Increasing PCM main memory lifetime, Proceedings of the Conference on Design, Automation and Test in Europe (DATE), European Design and Automation Association, 2010, pp. 914–919.

[15] S. Lee, H. Bahn, S.H. Noh, CLOCK-DWF: A write-history-aware page replacement algorithm for hybrid PCM and DRAM memory architectures, IEEE Trans. Computers 63 (9) (2014) 2187–2200.

[16] S. Raoux, G.W. Burr, M.J. Breitwisch, C.T. Rettner, Y.C. Chen, R.M. Shelby, et al., Phase-change random access memory: a scalable technology, IBM J. Res. Dev. 52 (4-5) (2008) 465–479.

[17] "Samsung now mass producing industry's first 2nd-generation, 10-nanometer class DRAM." https://news.samsung.com/global/samsung-now-mass-producingindustrys-first-2nd-generation-10-nanometer-class-dram, 2017.

[18] H.-S.P. Wong, S. Salahuddin, Memory leads the way to better computing, Nat. Nanotechnol. 10 (3) (2015) 191–194.

[19] Y. Chen, C. Rettner, S. Raoux, G. Burr, S. Chen, R. Shelby, et al., Ultra-thin phase-change bridge memory device using gesb, 2006 International Electron Devices Meeting, IEEE, 2006, pp. 1−4.
[20] R. Simpson, P. Fons, A. Kolobov, T. Fukaya, M. Krbal, T. Yagi, et al., Interfacial phase-change memory, Nat. Nanotechnol. 6 (8) (2011) 501.
[21] J. Tominaga, R. Simpson, P. Fons, A. Kolobov, The first principle computer simulation and real device characteristics of superlattice phase-change memory, 2010 International Electron Devices Meeting, IEEE, 2010, pp. 22−23.
[22] I.S. Kim, S.L. Cho, D.H. Im, E.H. Cho, D.H. Kim, G.H. Oh, et al., High performance PRAM cell scalable to sub-20nm technology with below 4F2 cell size, extendable to DRAM applications, 2010 Symposium on VLSI Technology, IEEE, 2010, pp. 203−204.
[23] J. Liang, R.G.D. Jeyasingh, H.Y. Chen, H.S.P. Wong, A 1.4 μA reset current phase change memory cell with integrated carbon nanotube electrodes for cross-point memory application, 2011 Symposium on VLSI Technology, IEEE, 2011, pp. 100−101.
[24] G.W. Burr, M.J. Brightsky, A. Sebastian, H.-Y. Cheng, J.-Y. Wu, S. Kim, et al., Recent progress in phase-change memory technology, IEEE J. Emerg. Sel. Top. Circuits Syst. 6 (2) (2016) 146−162.
[25] S.W. Fong, C.M. Neumann, H.P. Wong, Phase-change memory−Towards a storage-class memory, IEEE Trans. Electron. Devices 64 (11) (2017) 4374−4385.
[26] Q. Zheng, Y. Wang, J. Zhu, Nanoscale phase-change materials and devices, J. Phys. D: Appl. Phys. 50 (May 2017) 243002.
[27] A. Athmanathan, D. Krebs, A. Sebastian, M. Le Gallo, H. Pozidis, E. Eleftheriou, A finiteelement thermoelectric model for phase-change memory devices, International Conference on Simulation of Semiconductor Processes and Devices (SISPAD), IEEE, 2015, pp. 289−292.
[28] M. Salinga, M. Wuttig, Phase-change memories on a diet, Science 332 (6029) (2011) 543−544.
[29] K. Sonoda, A. Sakai, M. Moniwa, K. Ishikawa, O. Tsuchiya, Y. Inoue, A compact model of phase-change memory based on rate equations of crystallization and amorphization, IEEE Trans. Electron. Devices 55 (7) (2008) 1672−1681.
[30] W. Zhang, R. Mazzarello, M. Wuttig, E. Ma, Designing crystallization in phase-change materials for universal memory and neuro-inspired computing, Nat. Rev. Mater. 4 (2019) 150−168.
[31] J. Orava, A.L. Greer, B. Gholipour, D.W. Hewak, C.E. Smith, Characterization of supercooled liquid ge2sb2te5 and its crystallization by ultrafast-heating calorimetry, Nat. Mater. 11 (4) (2012) 279−283.
[32] G.W. Burr, P. Tchoulfian, T. Topuria, C. Nyffeler, K. Virwani, A. Padilla, et al., Observation and modeling of polycrystalline grain formation in $Ge_2Sb_2Te_5$, J. Appl. Phys. 111 (10) (2012), pp. 104308−104308.
[33] M. Salinga, E. Carria, A. Kaldenbach, M. Bornhofft, J. Benke, J. Mayer, et al., Measurement of crystal growth velocity in a melt-quenched phase-change material, Nat. Commun. 4 (2371) (2013).
[34] A. Sebastian, M. Le Gallo, D. Krebs, Crystal growth within a phase change memory cell, Nat. Commun. 5 (4314) (2014).
[35] J. Orava, D.W. Hewak, A.L. Greer, Fragile-to-strong crossover in supercooled liquid ag-in-sbte studied by ultrafast calorimetry, Adv. Funct. Mater. 25 (30) (2015) 4851−4858.
[36] B.-S. Lee, R.M. Shelby, S. Raoux, C.T. Retter, G.W. Burr, S.N. Bogle, et al., Nanoscale nuclei in phase change materials: origin of different crystallization mechanisms of ge2sb2te5 and aginsbte, J. Appl. Phys. 115 (6) (2014) 063506.

[37] S. Menzel, U. Böttger, M. Wimmer, M. Salinga, Physics of the switching kinetics in resistive memories, Adv. Funct. Mater. 25 (40) (2015) 6306–6325.
[38] A. Pigozzo (Ed.), Phase Change Memory, Springer International Publishing, 2018.
[39] K.W. Böer, S.R. Ovshinsky, Electrothermal initiation of an electronic switching mechanism in semiconducting glasses, J. Appl. Phys. 41 (6) (1970) 2675–2681.
[40] A. Warren, J. Male, Field-enhanced conductivity effects in thin chalcogenide-glass switches, Electron. Lett. 6 (18) (1970) 567–569.
[41] D.M. Kroll, Theory of electrical instabilities of mixed electronic and thermal origin, Phys. Rev. B 9 (4) (1974) 1669.
[42] M. Shaw, Thermal instability—the precursor to switching in inhomogeneous thin films, IEEE Trans. Electron. Devices 26 (11) (1979) 1766–1771.
[43] K.D. Tsendin, The changing of initial state in a strong electric field and memory effect in chalcogenides, J. Optoelectron. Adv. Mater. 9 (10) (2007) 3035–3038.
[44] K. Tsendin, Electro-thermal theory of the switching and memory effects in chalcogenide glassy semiconductors, Phys. Status Solidi (b) 246 (8) (2009) 1831–1836.
[45] M. Le Gallo, A. Athmanathan, D. Krebs, A. Sebastian, Evidence for thermally assisted threshold switching behavior in nanoscale phase-change memory cells, J. Appl. Phys. 119 (2) (2016) 025704.
[46] N.F. Mott, Conduction in non-crystalline systems: VII. non-ohmic behaviour and switching, Philos. Mag. 24 (190) (1971) 911–934.
[47] H. Henisch, E. Fagen, S. Ovshinsky, A qualitative theory of electrical switching processes in monostable amorphous structures, J. Non-Crystalline Solids 4 (1970) 538–547.
[48] D. Adler, M.S. Shur, M. Silver, S.R. Ovshinsky, Threshold switching in chalcogenide-glass thin-films, J. Appl. Phys. 51 (6) (1980) 3289–3309.
[49] A. Pirovano, A.L. Lacaita, A. Benvenuti, F. Pellizzer, R. Bez, Electronic switching in phasechange memories, IEEE Trans. Electron. Devices 51 (3) (2004) 452–459.
[50] A. Redaelli, A. Pirovano, A. Benvenuti, A.L. Lacaita, Threshold switching and phase transition numerical models for phase change memory simulations, J. Appl. Phys. 103 (11) (2008) 111101.
[51] D. Ielmini, Y. Zhang, Analytical model for subthreshold conduction and threshold switching in chalcogenide-based memory devices, J. Appl. Phys. 102 (5) (2007) 054517.
[52] D. Ielmini, Threshold switching mechanism by high-field energy gain in the hopping transport of chalcogenide glasses, Phys. Rev. B 78 (Jul 2008) 035308.
[53] C. Jacoboni, E. Piccinini, F. Buscemi, A. Cappelli, Hot-electron conduction in ovonic materials, Solid-State Electron. 84 (0) (2013) 90–95.
[54] V.G. Karpov, Y.A. Kryukov, S.D. Savransky, I.V. Karpov, Nucleation switching in phase change memory, Appl. Phys. Lett. 90 (12) (2007).
[55] J. Liu, Microscopic origin of electron transport properties and ultrascalability of amorphous phase change material germanium telluride, IEEE Trans. Electron. Devices 64 (5) (2017) 2207–2215.
[56] D. Eaton, Electrical conduction anomaly of semiconducting glasses in the system As-Te-I, J. Am. Ceram. Soc. 47 (11) (1964) 554–558.
[57] N. Bogoslovskiy, K. Tsendin, Dynamics of the current filament formation and its steady-state characteristics in chalcogenide based PCM, Solid-State Electron. 129 (2017) 10–15.
[58] J.A.V. Diosdado, P. Ashwin, K.I. Kohary, C.D. Wright, Threshold switching via electric field induced crystallization in phase-change memory devices, Appl. Phys. Lett. 100 (25) (2012) 253105.

[59] M. Stanisavljevic, H. Pozidis, A. Athmanathan, N. Papandreou, T. Mittelholzer, and E. Eleftheriou, "Demonstration of reliable triple-level-cell (TLC) phase-change memory," in *IEEE 8th International Memory Workshop (IMW)*, pp. 1−4, May 2016.

[60] N.F. Mott, E.A. Davis, Electronic Processes in Non-crystalline Materials, Oxford University Press, 2012.

[61] D. Krebs, T. Bachmann, P. Jonnalagadda, L. Dellmann, S. Raoux, Changes in electrical transport and density of states of phase change materials upon resistance drift, N. J. Phys. 16 (4) (2014) 043015.

[62] J.L.M. Oosthoek, D. Krebs, M. Salinga, D.J. Gravesteijn, G.A.M. Hurkx, B.J. Kooi, The influence of resistance drift on measurements of the activation energy of conduction for phase-change material in random access memory line cells, J. Appl. Phys. 112 (8) (2012) 084506.

[63] S.K. Bahl, K.L. Chopra, Amorphous versus crystalline gete films. iii. electrical properties and band structure, J. Appl. Phys. 41 (5) (1970) 2196−2212.

[64] P.C.G. Jost, *Charge Transport in Phase Change Materials*. PhD thesis, RWTH Aachen, 2013.

[65] J. Luckas, S. Kremers, D. Krebs, M. Salinga, M. Wuttig, C. Longeaud, The influence of a temperature dependent bandgap on the energy scale of modulated photocurrent experiments, J. Appl. Phys. 110 (1) (2011), pp. −.

[66] J.L. Hartke, The three-dimensional poole-frenkel effect, J. Appl. Phys. 39 (10) (1968) 4871−4873.

[67] R.M. Hill, Poole-Frenkel conduction in amorphous solids, Philos. Mag. 23 (181) (1971) 59−86.

[68] M. Ieda, G. Sawa, S. Kato, A consideration of poole-frenkel effect on electric conduction in insulators, J. Appl. Phys. 42 (10) (1971) 3737−3740.

[69] Y.H. Shih, M.H. Lee, M. Breitwisch, R. Cheek, J.Y. Wu, B. Rajendran, et al., Understanding amorphous states of phase-change memory using frenkel-poole model, Electron Devices Meeting (IEDM), 2009 IEEE International, IEEE, 2009, pp. 1−4.

[70] A. Calderoni, M. Ferro, D. Ielmini, P. Fantini, A unified hopping model for subthreshold current of phase-change memories in amorphous state, IEEE Electron. Device Letters 31 (9) (2010) 1023−1025.

[71] G.B. Beneventi, L. Guarino, M. Ferro, P. Fantini, Three-dimensional poole-frenkel analytical model for carrier transport in amorphous chalcogenides, J. Appl. Phys. 113 (2013) 044506.

[72] M. Le Gallo, M. Kaes, A. Sebastian, D. Krebs, Subthreshold electrical transport in amorphous phase-change materials, N. J. Phys. 17 (9) (2015) 093035.

[73] M. Kaes, M. Le Gallo, A. Sebastian, M. Salinga, D. Krebs, High-field electrical transport in amorphous phase-change materials, J. Appl. Phys. 118 (13) (2015) 135707.

[74] I. Karpov, M. Mitra, D. Kau, G. Spadini, Y. Kryukov, V. Karpov, Fundamental drift of parameters in chalcogenide phase change memory, J. Appl. Phys. 102 (12) (2007) 124503.

[75] M. Boniardi, A. Redaelli, A. Pirovano, I. Tortorelli, D. Ielmini, F. Pellizzer, A physics-based model of electrical conduction decrease with time in amorphous $Ge_2Sb_2Te_5$, J. Appl. Phys. 105 (8) (2009) 084506.

[76] D. Ielmini, D. Sharma, S. Lavizzari, A. Lacaita, Reliability impact of chalcogenide-structure relaxation in phase-change memory (PCM) cells, part I: Experimental study, IEEE Trans. Electron. Devices 56 (5) (2009) 1070−1077.

[77] M. Rizzi, A. Spessot, P. Fantini, D. Ielmini, Role of mechanical stress in the resistance drift of $Ge_2Sb_2Te_5$ films and phase change memories, Appl. Phys. Lett. 99 (22) (2011) 223513.

[78] P. Fantini, S. Brazzelli, E. Cazzini, A. Mani, Band gap widening with time induced by structural relaxation in amorphous $Ge_2Sb_2Te_5$ films, Appl. Phys. Lett. 100 (1) (2012) 013505.
[79] M. Le Gallo, D. Krebs, F. Zipoli, M. Salinga, A. Sebastian, Collective structural relaxation in phase-change memory devices, Adv. Electron. Mater. 4 (9) (2018) 1700627.
[80] J.Y. Raty, W. Zhang, J. Luckas, C. Chen, R. Mazzarello, C. Bichara, et al., Aging mechanisms in amorphous phase-change materials, Nat. Commun. 6 (7467) (2015).
[81] S. Gabardi, S. Caravati, G. Sosso, J. Behler, M. Bernasconi, Microscopic origin of resistance drift in the amorphous state of the phase-change compound GeTe, Phys. Rev. B 92 (5) (2015) 054201.
[82] F. Zipoli, D. Krebs, A. Curioni, Structural origin of resistance drift in amorphous GeTe, Phys. Rev. B 93 (11) (2016) 115201.
[83] D. Krebs, R.M. Schmidt, J. Klomfass, J. Luckas, G. Bruns, C. Schlockermann, et al., Impact of dose changes on resistance drift and threshold switching in amorphous phase change materials, J. Non-Crystalline Solids 358 (17) (2012) 2412–2415.
[84] M. Boniardi, D. Ielmini, Physical origin of the resistance drift exponent in amorphous phase change materials, Appl. Phys. Lett. 98 (24) (2011) 243506.
[85] A. Sebastian, D. Krebs, M. Le Gallo, H. Pozidis, and E. Eleftheriou, "A collective relaxation model for resistance drift in phase change memory cells," in *Proc. IRPS*, pp. MY.5.1–MY.5.6, 2015.
[86] M.L. Gallo, A. Sebastian, D. Krebs, M. Stanisavljevic, and E. Eleftheriou, "The complete time/temperature dependence of I-V drift in PCM devices," in *Proc. IEEE International Reliability Physics Symposium (IRPS)*, pp. MY–1–1–MY–1–6, 2016.
[87] A. Sebastian, N. Papandreou, A. Pantazi, H. Pozidis, E. Eleftheriou, Non-resistance-based cellstate metric for phase-change memory, J. Appl. Phys. 110 (8) (2011) 084505.
[88] M. Stanisavljevic, A. Athmanathan, N. Papandreou, H. Pozidis, and E. Eleftheriou, "Phase-change memory: Feasibility of reliable multilevel-cell storage and retention at elevated temperatures," *Proc. IRPS*, pp. 5B.6.1–5B.6.6, 2015.
[89] W.W. Koelmans, A. Sebastian, V.P. Jonnalagadda, D. Krebs, L. Dellmann, E. Eleftheriou, Projected phase-change memory devices, Nat. Commun. 6 (8181) (2015).
[90] S. Kogan, Electronic noise and fluctuations in solids, Cambridge University Press, 2008.
[91] P. Fantini, A. Pirovano, D. Ventrice, A. Redaelli, Experimental investigation of transport properties in chalcogenide materials through 1/ f noise measurements, Appl. Phys. Lett. 88 (26) (2006) 263506.
[92] P. Fantini, G.B. Beneventi, A. Calderoni, L. Larcher, P. Pavan, F. Pellizzer, Characterization and modelling of low-frequency noise in pcm devices, IEEE International Electron Devices Meeting (IEDM), IEEE, 2008, pp. 1–4.
[93] I. Giannopoulos, A. Sebastian, M.L. Gallo, V.P. Jonnalagadda, M. Sousa, M.N. Boon, et al., "8-bit precision in-memory multiplication with projected phase-change memory," in *IEEE International Electron Devices Meeting (IEDM)*, pp. 27.7.1–27.7.4, Dec 2018.
[94] D. Fugazza, D. Ielmini, S. Lavizzari, and A. Lacaita, "Distributed-poole-frenkel modeling of anomalous resistance scaling and fluctuations in phase-change memory (pcm) devices," in *Electron Devices Meeting (IEDM), 2009 IEEE International*, pp. 1–4, 2009.
[95] M. Nardone, V.I. Kozub, I.V. Karpov, V.G. Karpov, Possible mechanisms for 1/ f noise in chalcogenide glasses: A theoretical description, Phys. Rev. B 79 (2009) 165206.
[96] G. Betti Beneventi, A. Calderoni, P. Fantini, L. Larcher, P. Pavan, Analytical model for lowfrequency noise in amorphous chalcogenide-based phase-change memory devices, J. Appl. Phys. 106 (5) (2009) 054506.

[97] M.L. Gallo, A. Sebastian, G. Cherubini, H. Giefers, E. Eleftheriou, Compressed sensing with approximate message passing using in-memory computing, IEEE Trans. Electron. Devices 65 (Oct 2018) 4304–4312.

[98] T. Tuma, M. Le Gallo, A. Sebastian, E. Eleftheriou, Detecting correlations using phase-change neurons and synapses, IEEE Electron. Device Lett. 37 (9) (2016) 1238–1241.

[99] C.D. Wright, Y. Liu, K.I. Kohary, M.M. Aziz, R.J. Hicken, Arithmetic and biologically inspired computing using phase-change materials, Adv. Mater. 23 (30) (2011) 3408–3413.

[100] A. Sebastian, T. Tuma, N. Papandreou, M. Le Gallo, L. Kull, T. Parnell, et al., Temporal correlation detection using computational phase-change memory, Nat. Commun. 8 (2017) 1115.

[101] P. Hosseini, A. Sebastian, N. Papandreou, C.D. Wright, H. Bhaskaran, Accumulation-based computing using phase-change memories with FET access devices, IEEE Electron. Device Lett. 36 (9) (2015) 975–977.

[102] G.W. Burr, R.M. Shelby, S. Sidler, C. Di Nolfo, J. Jang, I. Boybat, et al., Experimental demonstration and tolerancing of a large-scale neural network (165 000 synapses) using phase-change memory as the synaptic weight element, IEEE Trans. Electron. Devices 62 (11) (2015) 3498–3507.

[103] I. Boybat, M. Le Gallo, S.R. Nandakumar, T. Moraitis, T. Parnell, T. Tuma, et al., Neuromorphic computing with multi-memristive synapses, Nat. Commun. 9 (2018) 2514.

[104] M. Suri, O. Bichler, D. Querlioz, O. Cueto, L. Perniola, V. Sousa, et al., "Phase change memory as synapse for ultra-dense neuromorphic systems: Application to complex visual pattern extraction," in *2011 International Electron Devices Meeting*, pp. 4.4.1–4.4.4, Dec 2011.

[105] S. La Barbera, D.R.B. Ly, G. Navarro, N. Castellani, O. Cueto, G. Bourgeois, et al., Narrow heater bottom electrode-based phase change memory as a bidirectional artificial synapse, Adv. Electron. Mater. 4 (9) (2018) 1800223.

[106] T. Tuma, A. Pantazi, M. Le Gallo, A. Sebastian, E. Eleftheriou, Stochastic phase-change neurons, Nat. Nanotechnol. 11 (2016) 693–699.

[107] M. Le Gallo, T. Tuma, F. Zipoli, A. Sebastian, E. Eleftheriou, Inherent stochasticity in phasechange memory devices, Proceedings of the European Solid-State Device Research Conference (ESSDERC), IEEE, 2016, pp. 373–376.

[108] S. Gaba, P. Sheridan, J. Zhou, S. Choi, W. Lu, Stochastic memristive devices for computing and neuromorphic applications, Nanoscale 5 (13) (2013) 5872–5878.

[109] R. Gütig, R. Aharonov, S. Rotter, H. Sompolinsky, Learning input correlations through nonlinear temporally asymmetric Hebbian plasticity, J. Neurosci. 23 (9) (2003) 3697–3714.

[110] E.O. Neftci, B.U. Pedroni, S. Joshi, M. Al-Shedivat, G. Cauwenberghs, Stochastic synapses enable efficient brain-inspired learning machines, Front. Neurosci. 10 (2016) 241.

[111] M. Le Gallo, A. Sebastian, R. Mathis, M. Manica, H. Giefers, T. Tuma, et al., Mixed-precision in-memory computing, Nat. Electron. 1 (4) (2018) 246.

Chapter 4

Magnetic and ferroelectric memories

Nicolas Locatelli[1], Liza Herrera Diez[1] and Thomas Mikolajick[2,3]
[1]Center for Nanosciences and Nanotechnology, CNRS, Université Paris-Saclay, Palaiseau, France, [2]NaMLab gGmbH, Dresden, Germany, [3]Institute for Semiconductors and Microsystems, TU Dresden, Dresden, Germany

4.1 Magnetic memories

4.1.1 "Spintronics" at a glance

The crosslink between magnetic properties and other intrinsic properties of condensed matter is a concept that does not cease to be of great interest to the scientific community given its large potential for practical applications. The most prominent example is "spintronics," a short form for spin electronics, where the spin and charge degrees of freedom of the electron are intimately related and therefore integrate the usual mechanism for control and detection in magnetism and electronics. Spintronics is a prolific field of research that has many times transcended the limits of the basic science laboratories into widely spread technological applications. Most of these are related to the discovery of the giant magnetoresistance (GMR) [1,2], which links the alignment of magnetic moments in neighboring layers to the electrical resistance of the whole stack, a major scientific milestone that has been recognized with the Nobel Prize in Physics awarded to Albert Fert and Peter Grünberg in 2007. This scientific achievement was rapidly followed by a commercial breakthrough in information technology at the level of data storage and retrieval that continues to develop at an everincreasing speed. A large fraction of these promising novel technologies under development are oriented toward the realization of nonvolatile memories based on the low-power spintronics nanodevices such as the magnetic random-access memory (MRAM). In the following sections, the key functionalities of magnetic memories, storing, reading, and writing of information, will be discussed in detail in the context of the physical processes involved, materials, and device design.

Memristive Devices for Brain-Inspired Computing. DOI: https://doi.org/10.1016/B978-0-08-102782-0.00004-6

4.1.2 Storing information

Information storage technologies are based on the concept of binary encoding of information in the form of a collection of '1' and '0' bits. In terms of hardware, this needs a physical system in which two stable and unambiguous states can be produced and reliably maintained over time. In magnetic memories, two well-defined magnetic states can be evidenced as the two directions, and the magnetic moment of a piece of ferromagnetic metal can take along one axis in space: "1" and "0" can simply be represented by the "up" and "down" directions, respectively, of the magnetic moment respective to the axis. An electrical analog can be found in the presence or absence of electrical charges in a transistor memory cell and is the basis of the widespread solid-state drives. Currently, traditional magnetic recording—on magnetic tapes—remains unbeaten in terms of storage capacity and production cost [3]. In this chapter, we introduce the challenges inherent to the production of addressable magnetic devices needed for the development of random-access memories.

At the heart of the concept of storage, there is the ability to guarantee the outstanding stability of the magnetic state against external perturbations present in the everyday life, such as temperature and electromagnetic perturbations. To achieve the required long-lasting stability, magnetic materials have to be carefully engineered, which leads us to the very heart of magnetic interactions in solid-state matter.

4.1.2.1 Ferromagnetism

Different magnetic interactions exist that allow for the manipulation of the magnetic states in different directions in space [4]. *Paramagnetism* allows for a material with unpaired electron spins to align them with an external magnetic field applied along a given direction, producing a sizeable net magnetic moment in the material. However, on removal of the external stimulus, all trace of collective alignment of electrons spin gives in to thermal excitations, canceling out the net magnetic moment. This fundamental limitation to reliable storage of the magnetic states is overcome in *ferromagnetic* systems, where nonvolatility is one of its defining characteristics. This feature is intimately linked to the electronic structure of the material: the existence of a strong exchange interaction between neighboring spins favors collinear alignment. In this way, a nonzero net magnetic moment always survives in ferromagnetic materials even in the absence of an external stimulus such as a magnetic field, which can be extremely robust to external perturbations.

Fig. 4.1 shows an illustration of the electronic density of states (DOS) for two transition metals: (1) a nonmagnetic metal (Cu) and (2) a ferromagnetic metal (Co) in the absence of an external magnetic field. While the DOS is equal for both electron spins in the nonmagnetic metal, we observe a strong asymmetry in the case of the ferromagnetic metal: 3d electrons, responsible

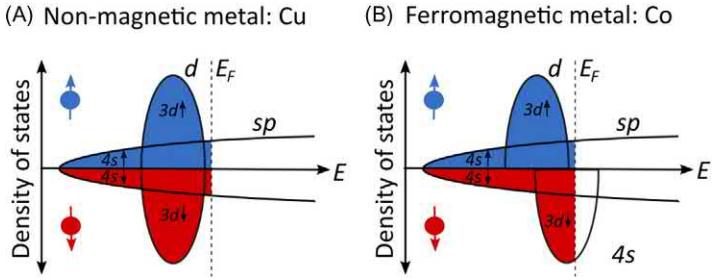


FIGURE 4.1 Electronic density of states for two transition metals: (A) a nonmagnetic metal [copper (Cu)] and (B) a ferromagnetic metal [cobalt (Co)] in the absence of an external magnetic field. 3d bands are equally filled in Cu, but there are more electrons with "up" spin than "down" spin in Co.

for the magnetic properties, have lower energy when their spin is aligned with the local magnetization. Notably, a high asymmetry exists at the Fermi energy E_F and will affect electrons involved in the transport processes. This is known as *spin polarization* of the ferromagnetic metal.

In the following sections, if not stated otherwise, we will consider magnetic nanostructures, with typical thickness close to 1 nm and lateral dimensions of tens of nanometers in which unpaired spins all point in the same direction, forming a so-called *single magnetic domain*.

4.1.2.2 Magnetic anisotropy and magnetic materials

The spin asymmetry described earlier goes hand in hand with another defining characteristic of ferromagnetism, a strong link of magnetic properties to the structure of the material through the so-called *spin-orbit coupling*. In this way, the energetics of the magnetic system depends on the orientation of the spins with respect to the lattice axes. This results in *magnetic anisotropy*, which defines energy minima for the alignment of spins along given directions dictated by the lattice geometry, called *easy axes*. Magnetic anisotropy induced by the crystalline structure can appear in many flavors, like planar biaxial or perpendicular uniaxial. This adds up with other sources of anisotropy, such as shape anisotropy or surface anisotropy [5]. The combination of composition and structure of the magnetic material along with shape engineering can define a single axis along which the magnetization will preferentially align. This easy axis can be either "in-plane" or "out-of-plane" with respect to the sample surface as shown in Fig. 4.2.

Reliability issues have an important connection to intrinsic material properties like magnetic anisotropy, but extrinsic factors like size are also of great importance. As mentioned earlier, ferromagnetic states are very robust in terms of thermal stability *in the bulk*; however, the nanostructuring needed for technological applications introduces additional challenges. In these

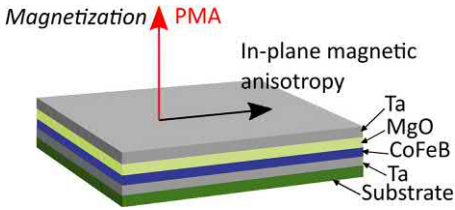


FIGURE 4.2 A typical CoFeB/MgO multilayer structure showing the directions of the magnetization for in-plane and perpendicular magnetic anisotropy (PMA).

conditions, thermal stability (Δ) can be addressed by considering the energy needed to flip the spins away from the easy axis with respect to thermal energy, $\Delta = K_u V/(k_B T)$, where K_u is the magnetic anisotropy constant (accounting for every source of anisotropy), V is the volume of the magnetic nanostructure, T is the temperature, and k_B is the Boltzmann constant.

As a reference, the barrier height, $K_u V$ separating the two stable magnetic states, needed to ensure that all the bits in a 1 MB magnetic memory are stable for 10 years at working temperature should verify $K_u V > 56.2\ k_B T$. Therefore, the cost of miniaturization is commonly afforded by working with materials exhibiting strong magnetic anisotropy.

While initial technologies were developed with materials having in-plane magnetic anisotropy, materials with PMA became most important for developing technological applications and are at the heart of present and novel magnetic storage technologies, as they only can support the continual decrease of devices dimensions without suffering critical loss of performances [6]. PMA develops strongly in ultra-thin magnetic films (typically below 1 nm) with a high surface-to-volume ratio. The existence of a symmetry breaking at the surface/interface of the magnetic material is the key to PMA: when in a stack, the last atomic layers on each side are in contact with different materials (in contrast with the symmetric environment inside the solid). Therefore, important efforts in material science are dedicated to interface engineering in magnetic multilayer stacks. High PMA materials can be found within the Co alloys/multilayers family including other ferromagnetic materials like Fe and Ni or nonmagnetic materials like Pt or Pd [5]. In particular, CoFeB alloys in contact with MgO insulating barriers (see Fig. 4.2 for a typical multilayer structure) are among the most intensely investigated materials in relation to memory technologies [7,8]. They share the high PMA with Pt/Co and Pd/Co alloys, but have much lower defect density and achieve better performance in terms of electrical control of the magnetization as will be discussed in the following.

4.1.3 Reading information

To be able to read the stored "0" and "1" bits of information, a reliable scheme needs to be in place to distinguish between the two states and make the information available for processing. Here is where spintronics shines. In

spintronics devices, magnetic states can be picked up as electrical signals, namely, as two distinct resistance states. The archetypical device is a stack consisting of two ferromagnetic layers separated by a nonmagnetic interlayer. In this section, we introduce how the relative orientation of the two ferromagnetic layers [parallel or antiparallel (AP)] gives rise to two different resistance states. Once again, a close look at the electronic structure and the device design is needed to understand how this outstanding physical effect is integrated into the already highly developed concept of MRAM memories.

4.1.3.1 Electronic transport in magnetic structures

To study transport in structures involving magnetic materials, one needs to consider not simply electrons conduction, but the sum of "up" electrons conduction and "down" electrons conduction. In a fair approximation, these two conduction channels can be considered fully independent as the probability of spin-mixing, the event of an electron flipping its spin along its path, can be assumed to be low, even at the working temperature. In this context, the resistance of a ferromagnetic structure can be evaluated as the resistance resulting from these two parallel channels.

In ferromagnetic transition metals, *s*, *p*, and *d* electrons contribute to electrical conduction, while only *d* electrons contribute to the magnetic properties. The most significant contribution to the electrical transport comes from the *s* and *p* electrons, which have a higher mobility, while *d* electrons are more localized. This is due to the shape of the electronic band structure: the *d* bands are much narrower than *sp* bands (see Fig. 4.1) and therefore imply a higher effective mass and in turn a lower velocity. Resistivity in metals is related to scattering events due to impurities, interaction with a local potential, phonons, etc. In ferromagnetic metals, *s* conduction electrons at the Fermi level will dominantly scatter on local *d* potentials. Due to spin asymmetry, these *d* potentials are more numerous at the Fermi level in one of the two channels, the "down" electrons in the case of the cartoon shown in Fig. 4.1B, resulting in stronger scattering in this channel and, in turn, higher resistivity.

4.1.3.2 Spin-valve structure and the giant magnetoresistance

Let us now consider a fully metallic multilayer structure, a so-called *spin-valve*, composed of a nonmagnetic layer sandwiched between two ferromagnetic layers and where electrons flow across the full stack. Fig. 4.3 shows the schematics of the two cases that can occur with respect to the alignment of the magnetic moment in the two ferromagnetic layers, namely, parallel (*P*) and *AP*.

For an electron pointing in a given direction, being aligned with the magnetization of both layers, with neither or with only one will result in experiencing different resistivities along its path. The global resistance when

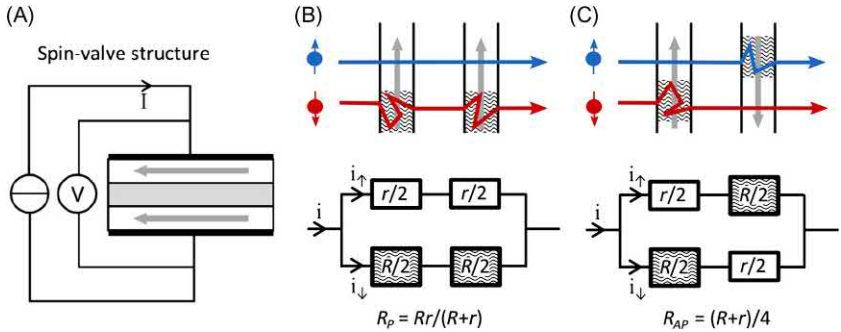


FIGURE 4.3 (A) The spin valve. Spin-dependent electrical transport through a multilayer stack in the (B) parallel and (C) antiparallel state. The graphic representation of the equivalent circuits leads to the expression for the corresponding resistances.

the magnetization vectors of the spin-valve layers are parallel to each other is then found to be lower than the resistance when they are AP (see Fig. 4.3). This Nobel prize winning discovery is known as the GMR [1,2,9].

The amplitude of the phenomenon is measured by the *GMR ratio*, expressed as $GMR = (R_{AP} - R_P)/R_P$. While it could only reach a few percent, the discovery of GMR and the further improvement of the spin-valve structures enabled the development of magnetic field sensors with stronger sensitivity, including those that were used in the reading heads of magnetic hard disk drives, allowing for a leap forward in the evolution of data density in hard disk drives [10].

4.1.3.3 Tunneling magnetoresistance

Starting in 1975, the metallic intermediate layer started being replaced by an insulating layer. The resulting device is called a *magnetic tunnel junction* (MTJ), referring to the classical tunnel junction where usual metallic electrodes are now ferromagnetic metals. In contrast to spin valves, in MTJs, the *AP-P* resistance difference is not due to spin-dependent scattering but due to spin-dependent tunneling across the insulating barrier. As before, we can work in the fair assumption that the electron spin is conserved during tunneling [11]. As a direct consequence, the total conductance of the stack will be the sum of the conductances in both the spin-up and spin-down channels.

As emerges from the study of the classical quantum tunneling effect, the electron tunneling probability under low voltages is proportional to a transmission coefficient, associated with the evanescent wave function inside the barrier and with the DOS at the Fermi level E_F in each electrode surrounding the barrier (see Fig. 4.4). Considering that the transmission coefficient is the same for every electron, the conductance of each channel is then proportional to the product of the corresponding DOS at E_F in each electrode (see Fig. 4.5). In a first approximation and for noncrystalline barriers, every

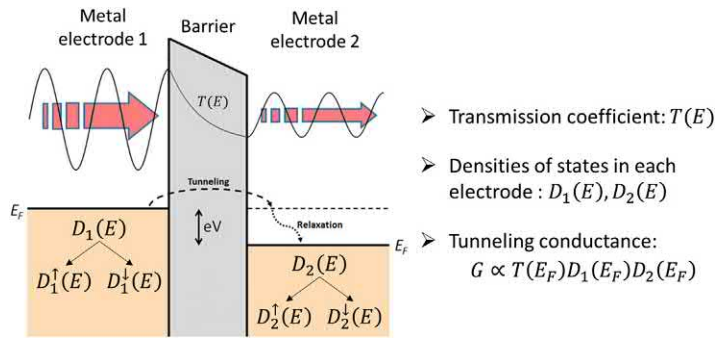


FIGURE 4.4 A tunnel barrier between two conductive electrodes. The electron tunneling probability under low voltages is proportional to a transmission coefficient, associated with the evanescent wave function inside the barrier and with the DOS at the Fermi level in each electrode surrounding the barrier.

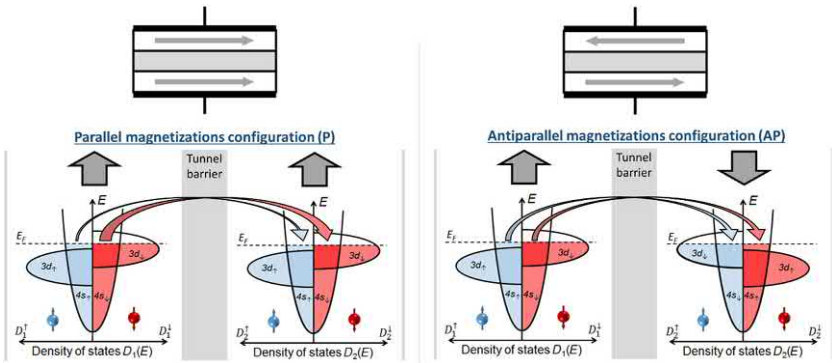


FIGURE 4.5 Spin-dependent tunneling conductance. The conductance of each channel, associated with each spin direction, is proportional to the product of the corresponding DOS in each electrode. The total conductance in the parallel (P) configuration becomes higher than the one in the antiparallel (AP) configuration.

electron, *s*, *p*, or *d*, contributes equally to the process. Considering the spin asymmetry previously described, the global conductance of the MTJ takes different values in the *AP* or *P* states. In this context, the degree of spin asymmetry in the electrodes, the *spin polarization*, is of extreme importance to achieve a significant difference of resistance between the two states. This phenomenon is known as *tunnel magnetoresistance* (TMR) and is characterized by the TMR ratio, $TMR = (R_{AP} - R_P)/R_P$ [12,13].

TMR was first observed in 1975 at low temperatures [12], but it was not until 1995 that a large room temperature TMR ratio was demonstrated [13]. In a first generation of devices, the insulating barriers were mainly made of

amorphous aluminum oxide, largely limiting the TMR to a few tens of percent. The nature and crystalline structure of the insulating barrier was later found to be crucial for the enhancement of the spin-filtering effect [14]. When a lattice match exists between the barrier and the electrodes, electrons from different orbitals will exhibit different transmission coefficients. Therefore, a careful choice of the barrier and the electrodes can strongly limit the tunneling to electrons to orbitals with very high spin asymmetry [14–16]. For instance, considering the illustration of Fig. 4.5, s electrons are not spin polarized, while d electrons are 100% polarized. If only d electrons were to contribute to the tunneling current, the TMR ratio would be virtually infinite. This feature has allowed for a remarkable improvement in TMR values that can currently reach a few hundred percent at room temperature like in crystalline CoFeB/MgO/CoFeB structures [17].

4.1.3.4 Device design

As explained in the previous sections, the design of a magnetic memory cell is based on two magnetic electrodes separated by an insulating barrier. "0" and "1" are encoded in the form of parallel and AP states of two neighboring magnetic layers that translate their magnetic information into two distinct values of resistance that can be detected and processed in an electrical circuit. Therefore, storing new data involves switching the magnetization of only one electrode, the *free electrode*, while the second one should remain unperturbed, acting as a *reference electrode*.

To fix the magnetization direction in the reference electrode, an increase in thickness (volume) compared with the free layer can be used to achieve a higher stability of the magnetization. In practice, the pinning of the reference electrode is achieved through its coupling with a neighboring antiferromagnetic layer. The strength of this coupling is chosen to be much higher than the energy needed for the "free" layer to switch, allowing for an independent manipulation of the magnetic state of the free layer without perturbing the alignment of the pinned layer. As shown in Fig. 4.6, applying a field

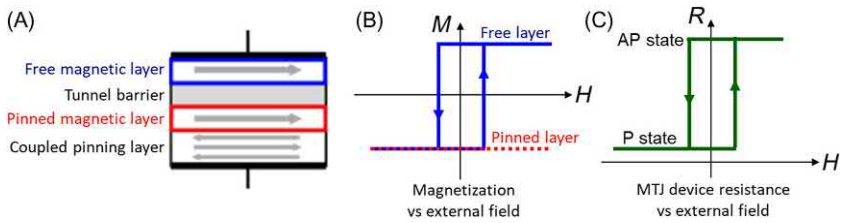


FIGURE 4.6 (A) Classical structure of an MTJ device and (B) magnetization switching and (C) TMR response induced by the application of an external magnetic field. On switching of the free-layer magnetization, the resistance of the device shows two different values depending if the magnetizations are parallel (P, low resistance) or antiparallel (AP, high resistance). Bits of information "0"and "1" are encoded in the direction of the free layer's magnetization.

consecutively in positive and negative directions will reverse the magnetization of the free layer, while leaving the pinned layer unperturbed. The junction switches between P and AP states, and the resistance level follows. Eventually, at high enough fields—bigger than its so-called coercitive field—the reference layer can also be switched.

4.1.4 Writing information

While magnetic fields are the most intuitive way of reversing magnetic states, they can easily perturb neighboring bits in a densely packed memory structure. In addition, switching fields can greatly increase as structures become smaller. This section is dedicated to *locally* reversing the magnetic states in a MTJ using a powerful spintronics tool, the spin-transfer torque (STT) effect, discovered in 1996 by Slonczewski [18] and Berger [19]. STT allows to switch the magnetization of the free layer of a spin-valve or an MTJ by running an electrical current through the structure and is at the heart of the STT-MRAM technology.

4.1.4.1 Acting on the magnetization by current flow: spin transfer

An electrical current flowing through a magnetic structure can get spin-polarized, for example, when conduction electrons flow through a ferromagnet, a majority will orient their spin along the direction of the local magnetic moment. They take an imprint of the first layer they cross and then transfer it to the second layer. In the MTJ structure where two magnetic layers are involved, a transfer of spin momentum, or a transfer of spin-polarization, takes place when the current flows from one magnetic layer to the next.

Let us consider the case of a "positive" current, flowing from the free layer into the pinned layer—electrons flow from the pinned layer into the free layer—in an MTJ with AP configuration as shown in Fig. 4.7A. The majority of the electrons tunneling exiting from the pinned layer and arriving into the free layer are polarized "up," according to the direction of the magnetization: they arrive opposed to the magnetization of the free layer. As the electrons flow into it, they reorient their spin to align with the new magnetic environment ("down"). But the interaction between conduction electrons and local magnetization is mutual, and the local magnetization is also pushed toward the direction of incoming spins. At low currents, this effect is negligible, and the magnetization stays put. When the current increases, the spin transfer starts to have a noticeable effect on the local magnetization, destabilizing it. Eventually, when a critical current is reached, incoming "up" electrons will be numerous enough to reverse the local magnetization and align it "up": the *AP* state switches into *P*. This effect is expressed as a torque, the *STT*, which acts on the free layer's magnetization opposing the natural

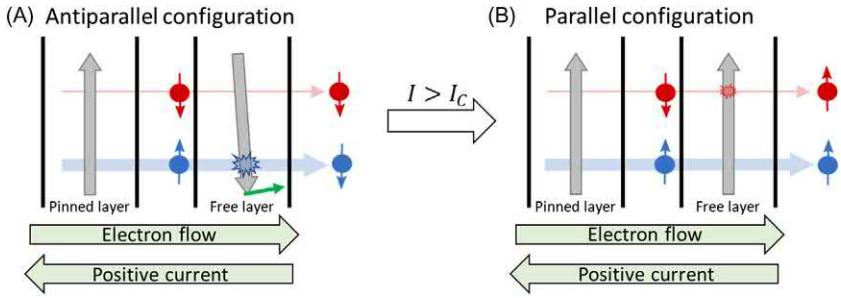


FIGURE 4.7 Spin-transfer torque in an MTJ device, in the case when a positive current is applied. When in AP configuration (A), the majority of electrons impinging on the free layer have "up" spins. Their spin flips as they cross the free layer and a spin transfer to the local magnetization occurs, resulting in a torque (green arrow) that destabilizes the magnetization direction. If the current goes beyond the critical current, due to the spin-transfer torque, the magnetization of the free layer will become unstable and will eventually switch, leaving the junction in its P state. Once in the P state (B), the positive current enhances the stability of the parallel configuration.

damping torque that keeps the magnetization steady [20–22]. After switching occurred, and following the same reasoning, the same sign of current will stabilize the magnetization in the *P* state. On the other hand, when the current flows in the opposite direction, "negative," electrons reflected on the pinned layer will in turn exert the STT on the free layer, which is now able to switch the magnetization from the *P* state to the *AP* state.

4.1.4.2 *Electrical control of magnetic states*

The previous paragraph describes the main origin of the spin-transfer effect, from which we extract that if a positive current can "write" a *P* state in the MTJ, then a negative current can "write" an *AP* state. A *critical threshold current* is needed for the STT switching to occur [23,24], and its reduction is the key aspect in the development of energy-efficient MRAM technologies [15,25]. Fig. 4.8 illustrates the hysteretic behavior. Critical currents are expected to have different values when switching from P and AP states, but correspond to sensibly identical critical voltages.

This critical current scale up with the value of the anisotropy constant, an energetic cost linked to stability, as well as other material parameters such as the Gilbert damping, characterizing the ability of the magnetic system to relax into its equilibrium state. Materials with low damping parameters can be switched with lower critical currents, which is crucial for good STT performance. The spin polarization of the current is also of great importance, critical currents reduce as polarization increases. In conclusion, it is of outmost importance to consider the choice and combination of materials to engineer magnetic junctions with high stability, yet efficient spin-transfer effect to reduce the energetic cost of the writing operation.

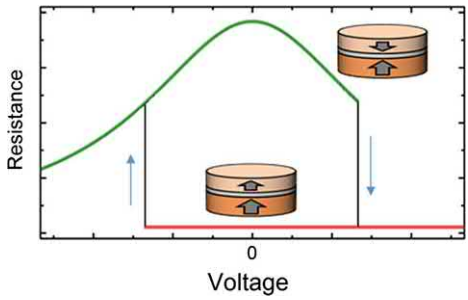


FIGURE 4.8 Evolution of the resistance of a magnetic tunnel junction (MTJ) versus applied voltage. When a positive voltage is applied beyond the critical value, the free layer (top) switches leaving the MTJ in a P state where the resistance is low. Identically, a negative voltage switches the junction to the AP state (high resistance). The high resistance state shows a strong dependence with voltage.

In addition, it appears that the critical current value scales with the size of the device, making this effect very powerful in the context of continuous decrease of the devices' size for smaller nodes [26]. Typical critical current densities are around 10^6-10^7 A/cm^2. Demonstration of the STT-mediated switching in a MTJ has already been obtained for devices down to 11 nm diameter [27]. In addition, it is to be noted that the STT switching is initiated by thermal fluctuations and is in consequence, a stochastic process [23,28]. The switching delay with respect to the application of a voltage to the junction is indeed a random variable, whose spreading around mean value becomes smaller as the current is higher than the critical current. The deterministic switching illustrated in Fig. 4.8 corresponds to quasistatic variations of the current and does not feature the stochastic effects.

4.1.4.3 Magnetic domains and domain walls

Up to this moment, only single-domain nanomagnets have been considered, but as the size of the magnet increases, stray fields at the edges become a significant source of perturbation in a homogeneously magnetized layer. In these cases, the system minimizes these so-called demagnetizing fields by spontaneously breaking the uniform magnetic state ("up" or "down") into a patchwork of regions with "up" and "down" magnetization, which are called *magnetic domains*. At the transition between two "up" and "down" regions, a continuous rotation of the spins takes place forming what is known as a "domain wall" (DW) (see Fig. 4.9A). The width of these DWs is given by an interplay between the exchange energy, favoring the collinear alignment of spins (wide DWs), and magnetic anisotropy, which favors the alignment along the easy axis (narrow DWs) [29]. As discussed earlier, PMA materials show high anisotropies and therefore present narrow walls typically in the range of tens of nanometers.

Magnetic DWs are intensely investigated in the spintronics community since they are at the basis of a promising prototype of magnetic memory, the "racetrack" [30]. In this device, the information is stored in the form of "up" and "down" magnetic domains separated by DWs along a magnetic

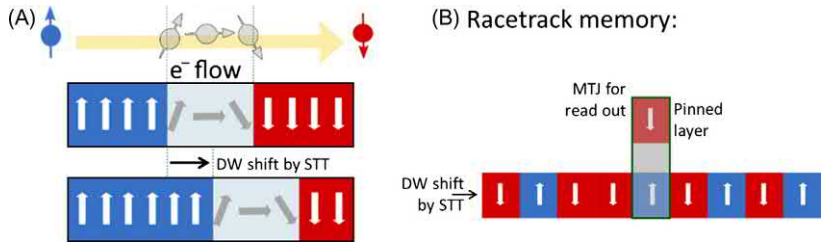


FIGURE 4.9 (A) A ferromagnetic film with two magnetic domains and a current flowing in the plane: when the current flows across the domain wall, its motion is induced by the spin-transfer torque. (B) Racetrack memory.

nanowire. This is a fundamental difference with respect to MRAM where magnetic bits are individually stored in separated devices (MTJs).

The same principle described for STT in MTJs applies for magnetic DWs. When the current flows from one domain to the other, as the electrons pass through this noncollinear spin arrangement, their spin rotates to align with the local spins inside the DW, and by doing so, they induce a torque. The addition of torques inside the DW results in the motion of the magnetic DW as a whole. This is depicted in Fig. 4.9A. A spin-polarized current flowing perpendicular to the magnetic layer plane can, in some cases, also induce DW motion. Note that in any case, the direction of the motion is once again dependent on the sign of the current. In case of two domains coexisting in a magnetic layer, the motion of the DW will result in the extension of one of the domains (see Fig. 4.9A) depending on the sign of the current. In the case of a racetrack, with several consecutive domains, the current-induced DW motion will act on the multiple DWs located along the current path, hence moving them collectively (see Fig. 4.9B).

For the reading operation in the racetrack, a single MTJ is used to read the entire collection of bits in the nanowire. Each bit is brought to the MTJ by moving the entire ensemble of DWs in the nanowire (an extended free layer) so as to align it with the position of the MTJ. A small current flowing through the MTJ and a read of the resistance will give the state of the chosen domain. This process is depicted in Fig. 4.9B. The racetrack is regarded as a highly promising concept in view of achieving a "universal memory" with high storage density and fast operation. As in the case of STT in MTJs, a low critical current to move the DW is one of its main technological challenges.

4.1.5 Latest developments

4.1.5.1 Voltage control of magnetic anisotropy

As mentioned earlier, high magnetic anisotropy assures the thermal stability of the system, but at the same time, it renders it less susceptible to STT,

which increases critical currents. One imaginable solution to this problem could be to have a device in which the anisotropy can be switched temporarily to a "low" value for the writing operation to reduce the critical current but be brought back to a "high" value immediately after for durable storage.

Electrical gating of magnetic anisotropy in ferromagnetic metals has been first observed in 2007 [31], and since then, it has not stopped to develop at an everincreasing speed. It is particularly efficient in materials with PMA, where a change in the occupation of the electronic levels at the interface with an oxide can have a large impact in magnetic anisotropy. This technology has already proven successful in assisting switching in MTJs [32], and it has a key role in the control of the motion of magnetic DWs. Strong pinning/depinning of DWs and nucleation of magnetic domains can be induced by a gate voltage in technologically relevant ferromagnetic layers [33–35], which is viewed as a potential improvement to the racetrack design and also constitutes a promising concept for magnetic DW logic circuit design [36].

4.1.5.2 Pure spin currents

As previously discussed, a fully polarized current is needed to maximize the efficiency of the spin-STT to switch the magnetization. In materials with strong spin-orbit coupling, the flow of a current results in the deflection of electrons with different spins in opposite directions, even in the absence of any external magnetic field, an effect called the spin-hall effect (SHE) in analogy with the well-known Hall effect [37,38]. Electrons with "up" and "down" spins then flow in opposite directions, and a magnetic layer placed on the top of such material could then experience a flow of "pure spin current," exerting a STT particularly efficient to switch its magnetization [39]. This innovative approach is being intensively investigated since it has the potential of providing a path toward a critical reduction in power consumption with respect to STT.

4.2 Ferroelectric memories

4.2.1 Ferroelectric materials

4.2.1.1 Ferroelectricity

Ferroelectricity is the property to have a spontaneous electric polarization that can be reversed by the application of an external electric field [40]. This property is only observed in special materials with a noncentrosymmetric crystal structure that allows dipoles to be formed and switched within the crystal. As a result, a ferroelectric material has two stable polarization states that can be switched between each other. Pyroelectricity as the property of a material to change its polarization with temperature and piezoelectricity as the property of a crystal to change its polarization with an applied mechanical stress are also based on the crystal structure but do not require the

polarization to be switched by an electrical field. Therefore, all ferroelectric materials are also pyroelectric and piezoelectric. As a consequence, ferroelectricity can be considered as an extremely valuable material property that allows a number of interesting applications. However, the complex crystal structure required often limits its practical use in cases where the benefits are not counteracted by the complexity of the material integration. Fig. 4.10A indicates the ferroelectric materials as a subgroup of pyroelectric and piezoelectric materials inside the big class of dielectric materials. The switchable polarization gives rise to a hysteresis curve of the polarization versus electrical field behavior as shown in Fig. 4.10B. If we identify the two remnant polarization points at zero applied electric field as "0" and "1," it becomes clear that such material will be an ideal candidate for a binary nonvolatile memory. It does not only have two stable polarization points, but switching between them is done purely by applying an electrical field and only the current to move the involved ions in the crystal from the one stable lattice position to the other one needs to be supplied. In contrast, most other nonvolatile memories use switching mechanisms that involve a current and some sort of inefficiency when switching the device [41].

4.2.1.2 *Perovskite-based ferroelectric materials*

Perovskites [42] having the general formula ABO_3, where A and B are metal cations [42], are the most important materials that can show ferroelectricity. A very popular model system is $BaTiO_3$ (BTO). Probably, the most

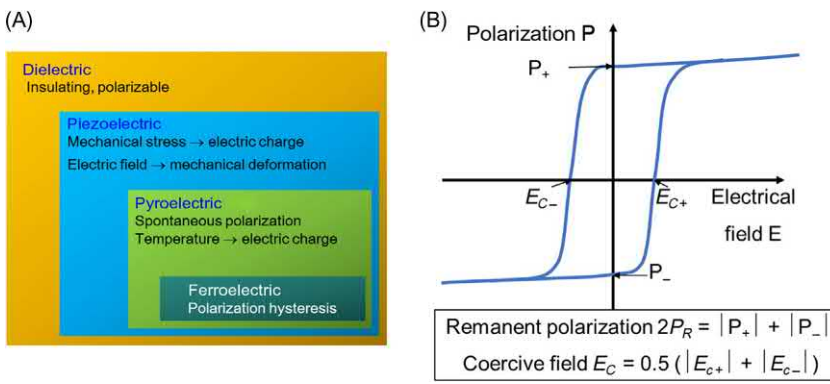


FIGURE 4.10 (A) Ferroelectric materials as a subgroup of dielectric, piezoelectric, and pyroelectric materials. A ferroelectric material will also be piezo and pyroelectric. (B) Typical ferroelectric hysteresis of the polarization as a function of the electrical field. The remanent polarization for the positive (P_{R+}) and negative (P_{R-}) polarization state, and the positive coercive field E_{C+} and the negative coercive field E_{C-} are important parameters to characterize the curve. If the curve is nearly symmetric with respect to field and polarization, the double remanent polarization $2P_R$ and the average coercive field E_C are often used to characterize the ferroelectric behavior of a material.

frequently applied material system is $PbZr_xTi_{1-x}O_3$ (PZT), were the B-site in the perovskite can be occupied either by a Zr or a Ti cation [43]. The general crystal structure of a perovskite together with the concrete configurations for BTO and PZT are shown in Fig. 4.11A. BTO and PZT were also the materials that were initially used to explore ferroelectricity in memory devices. Already back in the 1950s, first experiments with BTO crystals having electrodes on the front and back were made to realize a much cheaper and better scalable alternative to magnetic core memories [44]. However, the nonideal steep hysteresis leads to disturb problems when a pure cross-point arrangement is used, and therefore, this effort was discontinued. In the late 1980s, semiconductor technology was developed to a level that made it possible to add an MOS transistor to the ferroelectric capacitor and by this block disturbs [45]. Due to its superior properties, PZT was integrated into the back end of line to form 1 transistor and 1 capacitor memory cells similar to a dynamic random access memory (DRAM). In the first realizations, Pt electrodes were used together with the PZT material to form the capacitor. In such a configuration, the material shows a pronounced fatigue effect during field cycling that limits the application field the memory can be used in. One solution to that problem was the even more complicated material $SrBi_2Ta_2O_9$ (SBT) [46]. This so-called layered perovskite has Bi_2O_2 layers in between pseudo-perovskite layers. This material showed fatigue free behavior on platinum electrodes and on top of that has a lower coercive field

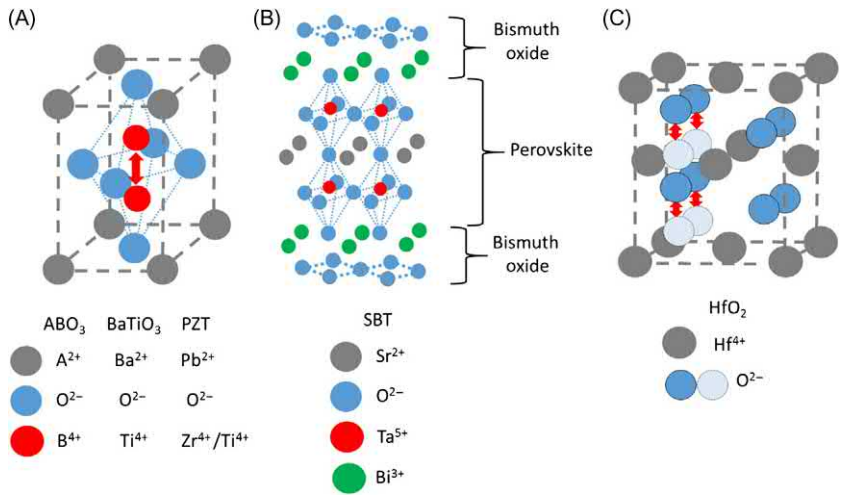


FIGURE 4.11 Crystal structure of (A) a perovskite with $BaTiO_3$ and $PbZr_xTi_{1-x}O_3$ as examples and (B) the layered perovskite $SrBi_2Ta_2O_9$ and (C) a fluoride type ferroelectric based on HfO_2. In the perovskite and layered perovskite materials, a central cation is responsible for the polarization switching, while the oxygen anions are responsible for the polarization switching in the fluoride ferroelectric.

(see Fig. 4.10B) compared with PZT, which translate to a lower switching voltage of the ferroelectric capacitor at the same film thickness. The typical crystal structure of SBT is illustrated in Fig. 4.11B. Using both materials PZT and SBT, low volume products reached the market in the 1990s, and there was a high expectation for a fast-further development that would lead to competitive products in the standalone memory arena. However, compared with typical materials like Si, Al, Cu, or binary oxides like SiO_2, Si_3N_4, or Al_2O_3 that are typically used in semiconductor production processes, the ferroelectric materials have a very complex structure as shown in Fig. 4.11A and B. The complicated crystal structure results in a number of drawbacks that turned out to hinder the scaling of the ferroelectric memories at the same pace as competing solutions. The disadvantages are as follows:

- To create the needed crystal structure, a rather high thermal budget is required that is significantly higher than other elements used in the back end of line. Here, SBT is even worse than PZT since it needs 650°C–700°C [47] annealing temperature compared with about 550°C–600°C for PZT [48].
- At the interface toward the electrodes, dead layers are formed that are nonferroelectric. These dead layers hinder the scaling of the film thickness. Therefore, a film thickness above 50 nm is required [49].
- The complex oxides can be rather easily reduced. A typical back end of line process in a semiconductor manufacturing environment, however, contains a high amount of reducing hydrogen that will degrade the perovskite structure and requires some protecting layers [50].
- Such multicomponent oxides are nontrivial to deposit by chemical vapor deposition processes. Although the deposition itself was demonstrated for both PZT and SBT for scaled structures, it was not possible to create the correct crystal phase at the sidewalls of a three-dimensional capacitor [51]. Therefore, the path toward a 3D integration is at least questionable.

As a result, ferroelectric memories are established on the market since a quarter of a decade, but scaling progresses are significantly slower compared with conventional memories like DRAM or Flash. Today, the most advanced ferroelectric memories are based on a 130-nm technology node [52], while DRAM uses sub 20-nm ground rules. Moreover, even many years of intense research did not lead to a groundbreaking advancement but only small improvements. This situation has drastically reduced the effort the industry is spending on solving the aforementioned issues. Therefore, there is essentially no hope that perovskite-based ferroelectric memories will leave the niche role they are currently in.

4.2.1.3 *Fluoride structure ferroelectric materials*

In 2011, Boeske et al. reported on the observation of ferroelectricity in silicon doped Hafnium oxide [53]. This unexpected discovery needed to be

verified by the scientific community. The effect was confirmed using different dopants [54] and different fabrication methods like atomic layer deposition [54], physical vapor deposition [55,56], or chemical solution deposition [57]. Although there is still no general accepted model of how the metastable orthorhombic phase is stabilized in detail [58], it is now well established that it is indeed the orthorhombic $Pca2_1$ phase that is responsible for the ferroelectric hysteresis [59]. Moreover, significant knowledge has been gathered on how the ferroelectric parameters can be controlled while fabricating the films [60]. In Fig. 4.11C, the schematic of an orthorhombic HfO_2 crystal is drawn showing the possible movement of the oxygen ions leading to the polarization switching. Note that the oxygen anion is assumed to be responsible for the ferroelectric switching in doped hafnium oxide. This is in contrast to perovskite-based ferroelectrics were one of the cations (in case of PZT, this is either Ti^{4+} or Zr^{4+}) moves during polarization switching. Since the hafnium oxide crystallizes in a fluoride crystal structure, it has become popular to call this class of ferroelectrics as also fluoride structure ferroelectrics. When comparing the fluoride structure ferroelectric materials with conventional perovskite-based ferroelectrics, then we can find a remanent polarization that is comparable but a coercive field that is typical above 1 MV/cm and is therefore about one order of magnitude larger than the one observed in PZT of about 100 kV/cm and SBT of about 50 kV/cm [61]. This striking difference will have important implications for different memory applications discussed in the following.

4.2.2 Capacitor-based ferroelectric memories

4.2.2.1 Ferroelectric random-access memory based on a one transistor–one capacitor cell

Up till now, the most successful way of integrating a ferroelectric material into a memory cell is the one transistor–one capacitor (1T-1C) approach that is comparable with a DRAM where the capacitor dielectric is replaced by the ferroelectric, and the plate-line is actively driven by a voltage to switch the ferroelectric material between the two polarization states (see Fig. 4.12A and B). This concept is referred to as a ferroelectric random-access memory (FeRAM). As mentioned in Section 4.2.1.2, the 1T-1C concept is already on the market but limited to niche applications caused by the inability to fabricate three-dimensional ferroelectric capacitors. With aluminum doped hafnium oxide, it was shown in 2013 that a three-dimensional capacitor will give almost the full area enhancement expected from the three-dimensional structure [62,63]. However, now the large coercive field becomes the limiting issue. First, the high coercive filed means a higher operating voltage. This can be partially compensated by the significantly thinner film, but a net disadvantage remains. But even more severe, the high

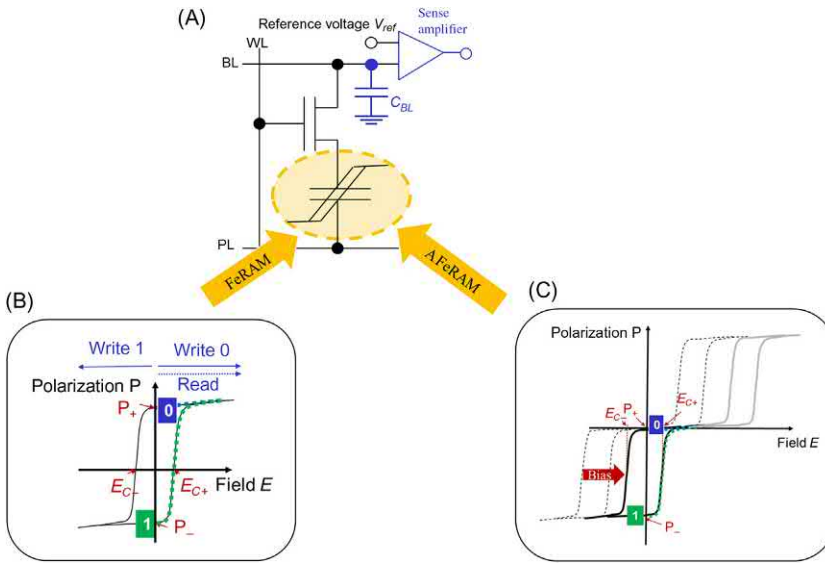


FIGURE 4.12 One Transistor–one capacitor ferroelectric random-access memory (FeRAM) cell. (A) The schematic of the cell. The storage capacitor can be realized in two ways. (B) A classical ferroelectric capacitor as indicated by the hysteresis is the most common realization. (C) An alternative is to use the antiferroelectric double hysteresis loop with an additional built-in bias.

field applied during switching limits the field cycling endurance [64]. Although significant progress in understanding and improving the issue has been achieved, the current endurance performance is still not at the level required for a nonvolatile random-access memory and is still inferior to the performance seen from optimized PZT devices although the basic behavior shows some striking similarities when comparing with the behavior of non-optimized PZT layers [65]. Moreover, since the physical origin of the field cycling stability and the imprint seem to be connected [66], we can also expect some more work to remain in imprint optimization, which was so far given much lower attention compared with the field cycling.

4.2.2.2 *Antiferroelectric random-access memory*

When using dopants smaller than Hf [54], also an antiferroelectric hysteresis as shown by the dotted lines in Fig. 4.12C, is observable. This can even be obtained in pure ZrO_2 [67]. The antiferroelectric variant shows much better field cycling stability compared with the ferroelectric variant [68]. However, in the antiferroelectric hysteresis, there is no remanent polarization. Therefore, it was proposed by Pesic et al. to use a built-in bias an shift the hysteresis curve to obtain a remanent polarization again [69,70]. This concept is called an antiferroelectric random-access memory. The resulting

hysteresis curve is schematically drawn by the bold black lines in Fig. 4.12C. Indeed, this variant can deliver better cycling endurance and promising retention and imprint performance [71]. The built-in bias can be either generated by using electrodes with different work functions or by using a stacked dielectric that can produce inherent fixed charges or dipoles [72]. Although the data that are available so far is all based on simple capacitor test structures, they show that the concepts could be a promising alternative for the future. Note that even ZrO_2, as it is used in DRAM memories today in the storage capacitor, can produce an antiferroelectric hysteresis that can be explored in this concept [69].

4.2.3 Transistor-based ferroelectric memories

When integrating a ferroelectric into the gate stack of metal insulator semiconductor transistor, the switching of the dipole will shift the V_T of the device as indicated in Fig. 4.13A–C. The resulting device is called a ferroelectric field effect transistor (FeFET). This variant has two inherent advantages when compared with the capacitor-based realization of a memory:

- Unlike the case of a capacitor, where the polarization needs to be switched in a predefined direction during reading, the read operation is nondestructive [73].
- The transistor has inherent amplification, and therefore, the signal is proportional to the charge per area and not the absolute charge. Therefore, scaling does not require to keep the capacitor area constant.

This would make the transistor-based architecture favorable. However, when integrating the ferroelectric into the gate stack, new difficulties arise. Traditional perovskite-based ferroelectrics are not stable on a silicon surface, and interface layers are required. But even more severe, this interface layer together with the unavoidable depletion layer in the silicon constitutes a series capacitor to the ferroelectric that will generate a depolarization during data storage. This depolarization field hindered the success of FeFETs using classical perovskite-based ferroelectrics [74]. Moreover, it is important to note that the memory window that is generated by polarization switching is proportional to the coercive field of the ferroelectric multiplied by its thickness [61]. At typical coercive fields for a perovskite material range from 50 to 100 kV/cm; this means that the ferroelectric needs to be several hundreds of nanometer thick to generate a reasonable sized memory window [62]. At this point, the rather high coercive field of hafnium oxide-based ferroelectrics turns out to be an advantage, as film thicknesses of around 10 nm will produce a memory window above 1 V. Moreover, the permittivity of hafnium oxide is in the range of 20–30 and therefore much lower compared with that of typical perovskites, which can have values of a few hundred. As a consequence, the depolarization field becomes manageable and nonvolatile

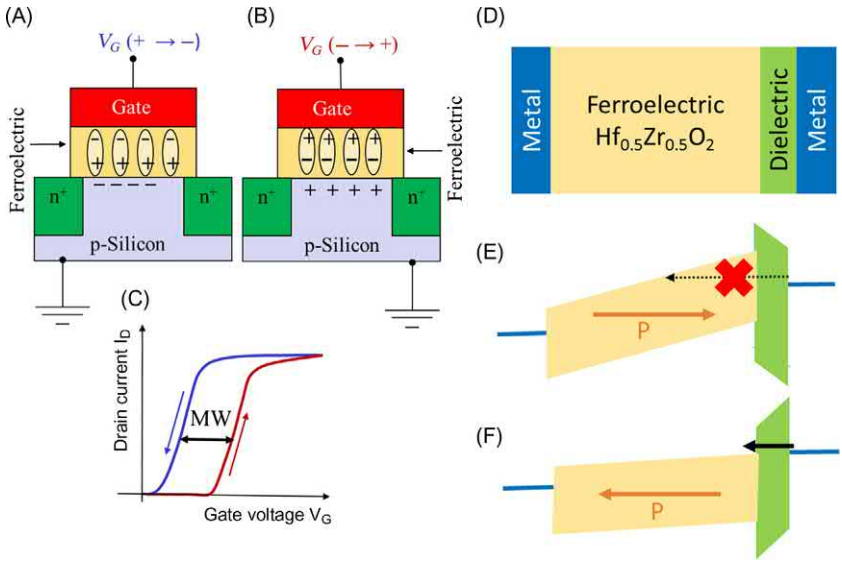


FIGURE 4.13 Ferroelectric field effect transistor (FeFET) and ferroelectric tunneling junction (FTJ). (A) and (B) An FeFET in the two possible polarization states. (C) The resulting transfer characteristic that now includes the hysteresis of the ferroelectric material. The memory window (MW) is a measure for the signal that can be generated out of the ferroelectric FE when used as a memory cell. (D) A two layer ferroelectric tunnel junction together with the band diagrams of such a structure for the two possible polarization states in (E) and (F). Tunneling through the thin dielectric barrier is enabled if the polarization points toward the dielectric (E). In the case where the polarization points away from the dielectric (F), the ferroelectric layer itself will be the part of the barrier significantly reducing the tunneling current.

retention can be achieved [75]. This feature very quickly has inspired research into this direction, which resulted in first devices integrated into a 28-nm technology in 2011 [76] and fully integrated arrays including periphery transistors in 28 and 22 nm fully depleted silicon on insulator (FDSOI) technology in 2016 [77] and 2017 [78], respectively.

4.2.4 Ferroelectric tunneling junctions

For some applications like synapses in neuromorphic computing systems, a two terminal resistive switch may be beneficial. A number of possibilities exist. First, a classical resistive switching based on the valence change mechanism can be introduced also in a ferroelectric capacitor based on doped hafnium oxide after a forming pulse is applied [79]. Interestingly, after a deep reset, the same structure that was used for resistive switching can be used as a ferroelectric capacitor again [79]. However, since the resistive switching mechanism in this case is not related to the ferroelectric polarization change, this variant will not be discussed here. Since a few years, the possibility to

alter the conductivity in ferroelectric DWs and use this as a nonvolatile memory device has also been investigated [80]. However, this effect is still in the basic research stage, and also so far only materials that are not compatible with an easy complementary metal-oxide-semiconductor (CMOS) integration are used. Therefore, this effect will also not be covered here.

The main objective of this section is FTJs. In 1971, this concept was proposed [81]. However, to fabricate such a device, a very thin high-quality ferroelectric layer was required. Therefore, it was not till the 2000s that such a device could be experimentally demonstrated [82]. Since then, quite intense research is going on in this field [83]. It is not surprising that also for ferroelectric hafnium oxide such a device is the agenda. Although it was shown that hafnium oxide-based ferroelectrics can be scaled down to 3 nm and beyond [84,85], these films show inferior remanent polarization, and due to their polycrystalline nature, high leakage of currents occurs. Therefore, another approach was taken, where a ferroelectric was placed in a series with a very thin dielectric tunneling layer. Fig. 4.13D shows the structure, and Fig. 4.13E and F illustrates the working principle of such a double-layer FTJ device using the band diagrams. In the "on" state, the tunneling will go through the thin tunneling layer only, while in the "off" state, the ferroelectric layer will be in the tunneling path and results in a much thicker barrier. A first demonstration of such a device using $Hf_xZr_{1-x}O_2$ was given in 2016 [86]. Recently, a larger parameter variation was done to evaluate the design constrains for such a device [87]. However, the basic shortcoming of such devices is the low read current that can be obtained. One solution can be obtained by amplifying the current using an additional MOSFET [88]. However, this comes at the expense of adding two additional transistors to the cell, and therefore, the advantage of having a compact cell produced in the back end of line is lost. Improvements of the read current can be obtained by optimizing the stack compositions and thicknesses. However, it remains unclear if a high enough read current can be obtained to make the FTJ a real contender as a resistive switching memory.

4.3 Memories beyond the Von Neumann architectures

As processors become faster and faster, the effort for accessing the memory that is separated from the computing unit limits both performance and energy efficiency of computing systems. This is referred to as the von Neumann bottleneck. With the possibility to integrate the ferroelectric and ferromagnetic memory devices into standard CMOS processes, possibilities that go beyond the realization of memory arrays become possible.

The logic-in-memory approach described in Section 4.3.1 is already a step toward overcoming this von Neumann bottleneck. It allows for the reconfiguration of logic gates inside the computing architecture by leveraging the two stable states of the nonvolatile memory elements. By this, it

strongly reduces the number of necessary memory accesses for computing. Neuromorphic computing, described in Section 4.3.2, is a more radical approach that implements architectures inspired from the basic structure of the brain: connections of artificial neurons and artificial synapses forming an artificial neural network. Section 4.3.3 presents potential applications leveraging the stochastic behavior of the devices.

4.3.1 Logic-in-memory

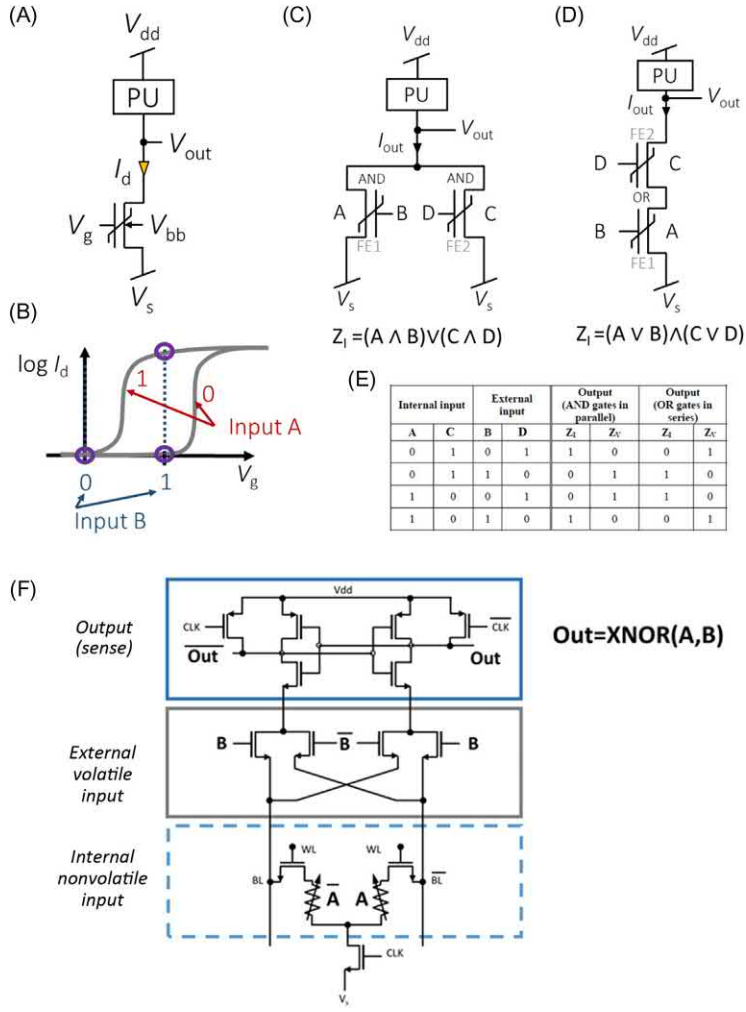
The possibility to distribute the nonvolatile memory devices in the CMOS circuit opens up the possibility to merge the signal processing and the storage. By integrating the nonvolatile memory devices directly into the logic circuits, it strongly reduces the number of necessary memory accesses for computing, and power supply can be cut off during the standby mode. In addition, the circuit area can be reduced when the memory devices are fabricated on top of the CMOS circuits and do not occupy extra area.

Therefore, the hybrid logic-in-memory architectures could provide a way to realize ultra-low power consumption and high-performance computing capability for the next-generation processor. Moreover, some computing system paradigms, such as brain-inspired computing, are expected to be realized by using such hybrid architectures.

4.3.1.1 Ferroelectric field effect transistor-based logic-in-memory

As discussed in section 4.2.3, ferroelectrics realize FeFET that is very similar to a CMOS device [77,78] that can be packed very closely together [89]. When a ferroelectric transistor is combined with a load element in an inverter type structure, NOR and NAND functionality can be obtained by using the variable stored in the ferroelectric FET as one input and the signal applied to the ferroelectric gate as the second input. This is illustrated in Fig. 4.14A and B.

Moreover, it is possible to switch between the NOR and NAND functions of such a circuit by either applying a source voltage to the ferroelectric FET or using a device fabricated in an FDSOI technology and utilizing the back-bias effect [91]. Following the same principle, more complex gates can be constructed by connecting the ferroelectric FETs in parallel or serial connections [92]. Fig. 4.14 shows the (A, C, and D) schematic, (B) current voltage characteristics and the truth table of the NAND/NOR gate and of the gates realized by connecting two FeFETs in (C) parallel or (D) serial connection, respectively. It becomes clear that with such an approach, one can realize calculations between variables stored in the FeFET that only need to be changed from time to time and fast changing variables at the input of the FeFET. A simple and straight forward application would be a content



Internal input		External input		Output (AND gates in parallel)		Output (OR gates in series)	
A	C	B	D	Z_I	Z_V	Z_I	Z_V
0	1	0	1	1	0	0	1
0	1	1	0	0	1	1	0
1	0	0	1	0	1	1	0
1	0	1	0	1	0	0	1

FIGURE 4.14 (A–E) Logic-in-memory concept based on the FeFET transistor. (A) A reconfigurable NAND/NOR gate can be achieved by connecting the FeFET device to a load transistor and taking the internal stored value as one input and the external signal as the second input. (B) This principle is illustrated using the transfer characteristic of the FeFET and the transistor current as the output signal. The reconfiguration can be done by either applying a source voltage to the FeFET or realizing the device in FDSOI technology and utilizing the back-bias effect to shift the IV characteristics. If two or more devices are connected in parallel as shown in (C) or serial connection as shown in (D), more complex functions can be realized. The truth table in (E) shows the output for both the case that the current is used as the output signal (Z_I) and the case that the voltage is used as the output signal (Z_V). (F) The logic-in-memory concept using MTJs. Internal output A is stored in two complementary MTJs, and external input B is applied on four transistor gates. This circuit implements an exclusive NOR operation, and the external input transistors can be arranged to implement any logic function. W. Zhao, M. Moreau, E. Deng, Y. Zhang, J.-M. Portal, J.-O. Klein, et al., Synchronous non-volatile logic gate design based on resistive switching memories, IEEE Trans. Circuits Syst. I: Regul. Pap. 61 (2) (2013) 443–454 [90].

addressable memory. Note that the parallel connection of the FeFETs mimics an AND type memory array, and the serial connection of FeFETs mimics an OR type memory array. Therefore, such concepts can be either distributed in the logic circuits or can be realized in an array fashion making use of regular and simple to implement structures [92].

This logic-in-memory approach also faces several challenges: the switching latency of the nonvolatile device is much larger than that of the conventional CMOS transistors, resulting in relatively lower computing frequency, as well as the limited endurance of ferroelectric junctions.

4.3.1.2 Comparison with the integration of magnetic devices

The concepts described for ferroelectric logic-in-memory are still valid for magnetic equivalent. However, it is instructive to point out differences between the two technologies.

The main advantage of employing spintronic devices in the logic-in-memory architecture is that MTJs can feature outstanding switching endurance, even when the devices are scaled down. However, the main difference between ferroelectric and ferromagnetic junctions comes from the resistance ratio between the two stable states: in MTJs, the resistance in AP state only reaches a couple of times the resistance in the P state. Sensitive circuitry is required to handle the small read-signal margin, namely, a sense-amplifier circuit. Details of such circuit and its application to magnetic memory can be found in Ref. [90]. It is also possible to enhance the sense amplifier to provide them with a logic-in-memory feature similar to what can be achieved with FeFETs. For example, the circuit shown in Fig. 4.14F naturally performs an exclusive NOR operation between the state of MTJs and an external voltage input. This circuit can be adapted to provide any logic function [90]. Other approaches for logic-in-memory architectures that integrate MTJs can be seen in the literature [93–96]. Recent developments like voltage control of MTJs also shed new lights on these concepts [97].

4.3.2 Perspectives for neuromorphic computing: brain-inspired architectures

For neuromorphic computing, two basic elements are required, namely, neurons that generate signal spikes based on an input signal and synapses that play the memory part, connecting neurons based on the history of spikes that had been applied to them. Both elements can be realized using conventional CMOS circuits. However, a pure CMOS realization has two drawbacks. First a significant number of devices are necessary that consume significant silicon real estate, which in turn limits the possible complexity such circuits could have. Second, the synaptic weights need to be stored in SRAM or

EEPROM/Flash memory cells limiting the power efficiency and flexibility of such approaches [98].

To realize synapses and neurons, two enabling properties can be explored in nonvolatile memory cells: first, an analog switching that allows to realize many different states of the memory cell and second, the ability to accumulate input signals before a response is achieved. The analog switching mechanism is a topic that is intensively researched in resistive switching devices [99,100] based on oxygen ion movement [101], charge trapping [102], or metal ion movement [103], and in phase change memory cells [104], both the analog switching and the accumulative switching have been demonstrated. In magnetic systems, analog and cumulative switching have been demonstrated using the sequential motion of a magnetic DW in a nanodevice [105] as will be described in the following.

4.3.2.1 Magnetic synapse and neuron

A key aspect of realizing a magnetic synapse is the possibility of creating a multistate analog switching between "0" and "1" in magnetic devices. The first work that considered the use of magnetic devices in the framework of building neuromorphic engines proposed the use of several binary MTJs connected in parallel to emulate the synapse function [106].

In terms of device design, the motion of magnetic DWs provides the possibility of conducting the switching of the magnetic state in an MTJ in several intermediate steps, contrary to the single-step switching that was discussed in section 4.1.4. This can be realized by allowing a DW to propagate inside the free layer though STT and controlling its position by applying current pulses (see Fig. 4.15A). The length of each displacement step is then determined by the amplitude and duration of the pulses and by the random distribution of pinning sites in the device, where the DW can actually stop. The resistance of the device is then determined by the proportion of the device that is in the *P* state and in the *AP* state, as the width of the DW can

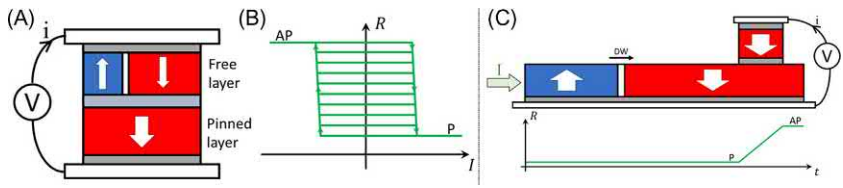


FIGURE 4.15 (A) A magnetic synapse realized by domain wall propagation in an MTJ device and (B) several levels of resistance achieved by moving the domain wall in the free layer with the spin-transfer torque. The resistance value is an image of the portion of the device being in the antiparallel configuration. (C) A proposal for a magnetic neuron based on domain wall propagation in an extended free layer. The resistance level remains unchanged until the DW reaches the read junction. Here, different currents are used for DW motion through the spin transfer effect and for reading.

be neglected. This allows the device to show resistances between the minimum, fully *P*, and the maximum, fully *AP* values [105] (Fig. 4.15B). It is to be noted that the stochastic nature of the DW pinning process makes the precise control of the DW position very complex, but that this feature can also be particularly interesting for neuromorphic applications.

This concept certainly is a first important step toward the development of analog bio-inspired hardware. In addition, voltage control of anisotropy could be incorporated into similar devices to add interesting low-power consuming new functionalities. Local gates could be used to predefine pinning sites and, in this way, allow for a reprogrammable control of the number of intermediate states and their pinning potential in view of enhancing the device's bio-inspired memory capabilities.

The expected behavior of a mimic neuron is to "integrate and fire" under voltage or current pulses originating from any external stimulus. The intrinsic physics of magnetization dynamics in two-state MTJs does not involve cumulative effects that would allow to design a neuron with a single binary junction. Some proposals have then been made of spintronics device-based neuromorphic circuits combining nonvolatile synapses based on the MTJ devices and MOS neurons [107]. But DW dynamics can again provide a solution to realize a magnetic neuron [108,109]. The device illustrated in Fig. 4.15C can be compared with the synapse, except that the ferromagnetic layer containing the DW is more extended than the junction stack. The magnetizations on the two sides of the junction are initially parallel, and the resistance is low. Cumulative pulses of the current then move the DW toward the junction, so that eventually the magnetization in the extended film becomes AP to the reference direction: the resistance switches to the high level. Of course, one would note that this concept description eludes the fact that once ejected, a new DW needs to be nucleated to start a new cycle. Other proposals for neuronal realizations based on magnetic devices can be found in Chapter 16, Neuronal realizations based on memristive devices.

The development of magnetic memory technologies has largely focused in the last decades, both at an academic level and an industrial level, on obtaining two reliable and programmable binary states. Nowadays, the memristive capabilities of the magnetic devices are being explored at an academic level [110–112] to pave the way to the new neuromorphic technologies to come.

4.3.2.2 Ferroelectric synapse and neuron

The ability to scale down FeFETs to dimensions that are in the range of typical domain sizes has led to considerable new insights into the switching processes involved in ferroelectric devices [113]. While large area devices can show continuous switching behavior, scaled devices shows a very abrupt switching in a limited number of steps [114]. This behavior is illustrated in

Fig. 4.16 using a large $W = L = 500$ nm and a small $W = 80$ nm and $L = 30$ nm device fabricated in the same technology process. Moreover, the switching in small devices has an accumulative character: if pulses with an amplitude below the coercive voltage are applied, a certain number of pulses are required to switch the device. When this number of pulses is reached, the device switches abruptly in an "integrate and fire" like behavior [115]. This type of behavior can be seen in Fig. 4.18B and C.

For realizing a synapse, the analog switching behavior is one of the most important features. The so far realized ferroelectric synapse circuits therefore normally rely on rather large devices [116–119]. The synapse device proposed in Ref. [117] is shown in Fig. 4.17. By using a transistor with $W = L = 500$ nm and the pulsing scheme shown in Fig. 4.17B, the spike time-dependent plasticity curve depicted in Fig. 4.17C was obtained. The drawback of this approach is the mentioned fact that once the devices will be so small that single-device switching is observed, the number of achievable states will be limited. In the future, several approaches can be

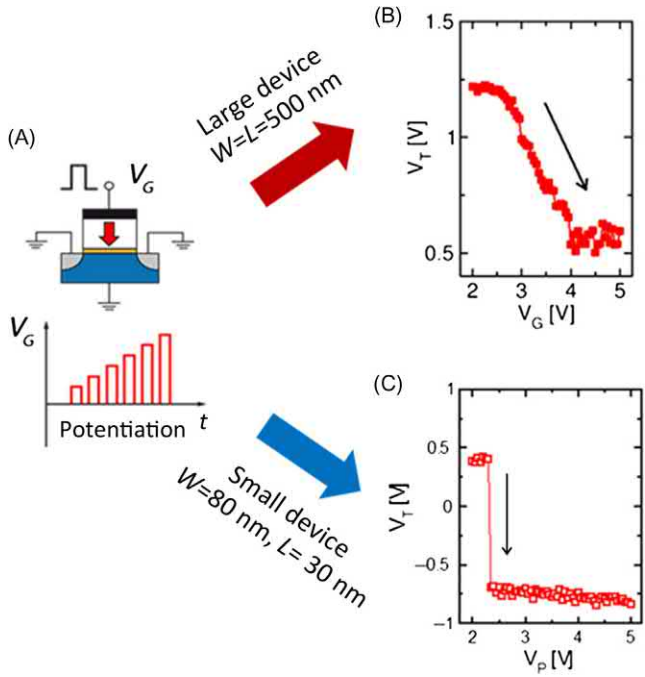


FIGURE 4.16 (A) FeFET as a basic building block in neuromorphic computing. Voltage pulses with rising amplitudes applied to a ferroelectric field effect transistor. (B) Response of a large device with $W = L = 500$ nm to a pulse rain as shown in (A) and (C) response of a scaled device with $W = 80$ nm and $L = 20$ nm to a pulse rain as shown in (A). The large device switches continuously, while the scaled device does not show a response to the first pulses but switches spontaneously after a certain number of pluses is reached.

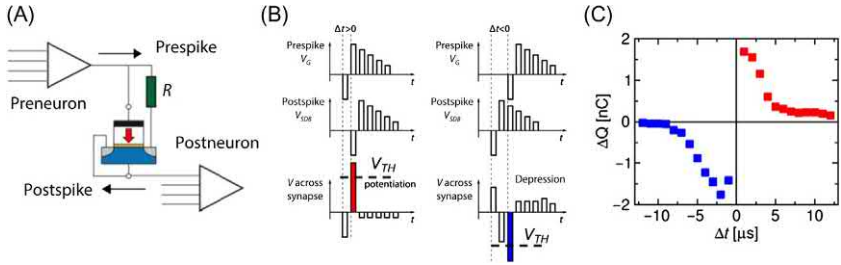


FIGURE 4.17 (A) Simple two terminal synapse circuit constructed from an FeFET and a resistor. (B) Pulse trains used for presynaptic and postsynaptic pulses and (C) spike-time-dependent plasticity obtained from a synapse as illustrated in (A) and the pulse trains shown in (B) when using a device with $W = L = 500$ nm.

considered to cope with this issue. First, grain and domain size need to be minimized by the fabrication procedure. Second, the results described so far were all obtained using planar devices. However, a FinFET would have a considerably larger area due to the three-dimensional structures and true 3D devices, similar to the ones used in 3D NAND devices, which would again allow for more area efficient solutions. A modified version of a 3D array would be possibly a very good fit toward a highly complex neuromorphic circuit. As an alternative to the FeFET, a synapse can also be realized using a FTJ as already demonstrated using a classical BFO-based FTJ in Ref. [120] and hafnium zirconium oxide in Ref. [121]. In these cases, we need to consider that the current now is flowing parallel to the dipole orientation and not perpendicular to the dipole as in the FeFET case. Therefore, the behavior of a device having only a few domains will not be the same. Since very little is known about such small FTJ devices so far, more research is needed to evaluate the potential of such an approach.

When looking at a possible implementation of a neuron, the features recently discovered in scaled-down devices can open an interesting opportunity. In Fig. 4.18B and C, the switching behavior of a scaled-down FeFET having $W = 80$ nm and $L = 30$ nm under the excitation by a series of pulses below the critical voltage (in the range of the coercive voltage) where the device switches already with the first pulse is shown. Note that a nearly symmetrical behavior for both polarities is observed. This symmetry is typically not found in resistive switching devices like phase change memory cells [122]. It becomes clear that the device shows an integrating behavior. It does not respond to the pulses until a certain amount of pulses is reached, and then it switches abruptly. This behavior can mimic the "integrate and fire" behavior of a neuron. For realizing the leaky behavior of a neuron, an additional voltage in the opposite state is applied between the pulses [123]. Another important feature is illustrated in Fig. 4.18D and E. When a pulse train, which includes a reset pulse and a refractory period on top of the pulse

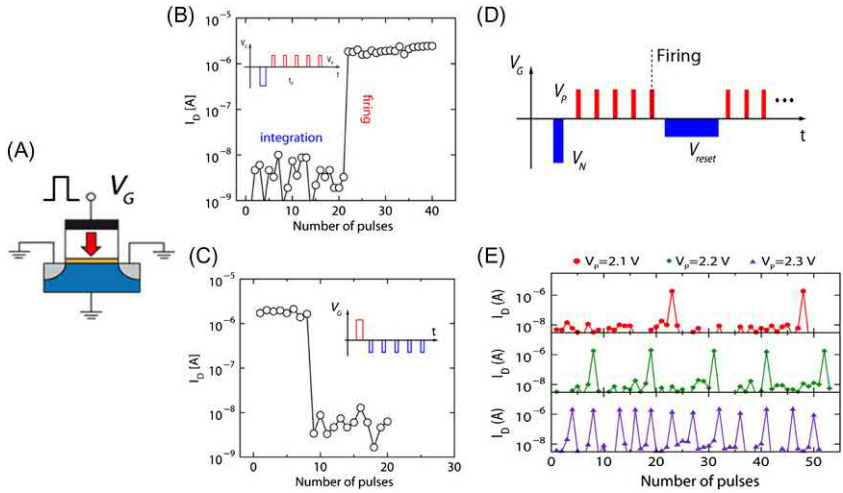


FIGURE 4.18 (A) Small FeFET device ($W = 80$ nm, $L = 30$ nm; see Fig. 4.16). The device shows an "integrate and fire" behavior as illustrated in (B) for the switching form the off state to the on state and (C) for the switching of the off state to the one state. When using the pulse trains illustrated in (D) with different amplitudes for the pulses VP, the spiking behavior shown in (E) is obtained.

train as indicated in Fig. 4.18D, is applied and the amplitude is changed, it results in the behavior as shown in Fig. 4.18E. Here, the activity of pulses at the output is a function of the amplitude of the incoming pulses. Although these data are only a first step, they already show a promising path toward realizing simpler neuron circuitry because the capacitor normally needed for the integration process can be omitted. Moreover, the possibility to use the ferroelectric hafnium oxide both for an important constitute of the neuron and for the synapses either realized by FeFETs or FTJ allows to use maximum synergy between both required elements.

4.3.3 Leveraging stochastic switching: random number generation, approximate computing

Imperfections of the devices are not always a flaw. For instance, some concepts propose to make use of the imperfections of the junctions switching process like true random number generation. This was achieved for ferroelectric junctions [124] and magnetic junctions [125,126] and could have applications for stochastic computing [127] among others. Another concept, approximate computing, relies on the ability of many systems and applications to tolerate some loss of quality or optimality in the computed result. By relaxing the need for fully precise or completely deterministic operations, approximate computing techniques allow substantially improved energy

efficiency [127–129]. An example involving MTJs programmed nondeterministically can be found in Ref. [130]. Further information and examples will be more extensively developed in Chapter 11.

4.3.4 Summary and outlook

Both magnetic and ferroelectric materials have in common that an important state variable shows a stable hysteretic behavior. Such a behavior makes them very well suited for semiconductor memories. However, limitations in integration and scaling have hindered their widespread success so far. Both in magnetic and ferroelectric devices, material and structural advances have gone a long way to solve the limitations.

In the field of magnetic nanostructures, the combination of the magnetoresistance effect and the spin-transfer effect creates a promising candidate for low-power nonvolatile memory device. The strong industrial interest for the development of magnetic random-access memories (MRAM) has fueled the research and development for new materials and novel structures allowing the scalability of the magnetic junctions while ensuring storing stability and low-power writing ability. In the magnetic nanotechnology front, the quest continues to reduce switching currents in memory prototypes. However, on the path toward industrialization, spintronics has unveiled a number of new physical effects that may not be exploited for the optimization of magnetic memories. In the light of recent developments of neuromorphic applications, it seems that spintronics has already much to offer to neuromorphic applications by simply reinventing exiting technologies and capitalizing well-known effects. Notably, the combination of magnetoresistance effects and the STT effect allows designing new devices based on the nucleation on motion of DWs inside magnetic devices. It is also exciting to imagine that the search of neuromorphic functionalities beyond the existing spintronics concepts will lead to the discovery of new magneto-electric phenomena and in turn to an ultimate feedback into the magnetic memory community.

Ferroelectric switching is also very well suited for nonvolatile memory applications due to the pure electric field–driven switching combined with an energy barrier high enough for nonvolatile retention. However, the scalability was limited by the complexity of commonly known ferroelectric materials. With the discovery of ferroelectricity in hafnium and zirconium oxide, a CMOS compatible path to integrating ferroelectrics into electronics has appeared. To make a memory cell out of the ferroelectric switching, three different possibilities exist. The material can be integrated into a capacitor that is written and read via transistor, an approach known as ferroelectric random-access memory that is established in the market place using traditional ferroelectric materials since a quarter of a century now. The ferroelectricity in hafnium oxide immediately solves the scaling issue of this approach. However, currently, the cycling endurance is still a major issue to

realize such a device. Recently, a variant using an antiferroelectric hysteresis combined with a built-in bias field called antiferroelectric RAM was proposed to solve the cycling issue. This shows promising results on capacitor level, but integrated devices are not shown so far. In the 1 T FeFET, the integration has already advanced to the 22-nm node with flash like performance and some improvements in speed and flexibility. So here we can see an advanced development stage already. Finally, the FTJ is still in a basic stage with some major challenges to be solved. Beyond the pure memory application, ferroelectrics show an interesting potential for logic-in-memory and neuromorphic applications. The integration as storage directly into the CMOS circuit can be a game changer for logic-in-memory applications, while features like continuous, abrupt, and accumulative switching may be used as building blocks to realize devices for neuromorphic computing. Therefore, we expect to see even more research and first applications utilizing ferroelectric switching in hafnium and zirconium oxide in the future. FeFETs are also a very hot topic as a candidate to realize a next-generation energy-efficient switch by operating the ferroelectric in the negative capacitance region and in turn to achieve voltage amplification [131]. However, this approach is beyond the scope of this chapter, and interested readers are referred to specific overviews on the issue [132].

In summary, both magnetic and ferroelectric hysteresis are features that are well suited for data storage and have been exploited for this application for quite a while. With the trend toward non-von Neumann computation and the growing demand for neuromorphic computing solutions, both magnetic and ferroelectric materials can extend their application field. The possibility to create analog memristive devices opens up new possibilities for their applications. Moreover, a number of "undesired" physical processes, like stochasticity, inherent to the switching processes have been sources of noise and trouble to achieve the best performances of device prototypes and can now be reconsidered in a different light and even become strong advantages.

References

[1] M.N. Baibich, J.M. Broto, A. Fert, F. Dau, F. Nguyen Van Petroff, P. Etienne, et al., Giant magnetoresistance of (001)Fe/(001)Cr magnetic superlattices, Phys. Rev. Lett. 61 (1988) 2472.

[2] G. Binasch, P. Grünberg, F. Saurenbach, W. Zinn, Enhanced magnetoresistance in layered magnetic structures with antiferromagnetic interlayer exchange, Phys. Rev. B 39 (1989). 4828(R).

[3] S. Furrer, M.A. Lantz, P. Reininger, A. Pantazi, H.E. Rothuizen, R.D. Cideciyan, et al., 201 Gb/in2Recording areal density on sputtered magnetic tape, IEEE Trans. Magnetics 54 (2) (2018) 1–8.

[4] A. Campbell, A. Fert, Ferromagnetic Materials, E. P. Wolfarth, North Holland, Amsterdam, 1982.

[5] M.T. Johnsony, P.J.H. Bloemenzx, F.J.A. den Broedery, J.J. de Vriesz, Magnetic anisotropy in metallic multilayers, Rep. Prog. Phys. 59 (1996) 1409–1458.

[6] H. Sato, M. Yamanouchi, S. Ikeda, S. Fukami, F. Matsukura, H. Ohno, Perpendicular-anisotropy CoFeB-MgO magnetic tunnel junctions with a MgO/CoFeB/Ta/CoFeB/MgO recording structure, Appl. Phys. Lett. 101 (2012) 022414.
[7] S.S. Ikeda, K. Miura, H. Yamamoto, K. Mizunuma, H.D. Gan, M. Endo, et al., A perpendicular-anisotropy CoFeB–MgO magnetic tunnel junction, Nat. Mater. 9 (2010) 721–724.
[8] C. Burrowes, N. Vernier, J.-P. Adam, L.H. Diez, K. Garcia, I. Barisic, et al., Low depinning fields in Ta-CoFeB-MgO ultrathin films with perpendicular magnetic anisotropy, Appl. Phys. Lett. 103 (2013) 182401.
[9] A. Barthélémy, A. Fert, F. Petroff, Giant magnetoresistance of magnetic multilayers, Handbook of Magnetic Materials, 12, Elsevier, 1999.
[10] E. Grochowski, R. Hoyt, Future trends in hard disk drives, IEEE Trans. Magnetics 32 (3) (1996) 1850–1854.
[11] R. Meservey, P.M. Tedrow, Spin-polarized electron tunneling, Phys. Rep. 238 (1994) 173–243.
[12] M. Julliere, Tunneling between ferromagnetic films, Phys. Lett. A 54 (1975) 225–226.
[13] J.S. Moodera, L.R. Kinder, T.M. Wong, R. Meservey, Large magnetoresistance at room temperature in ferromagnetic thin film tunnel junctions, Phys. Rev. Lett. 74 (1995) 3273.
[14] W.H. Butler, X.-G. Zhang, T.C. Schulthes, J.M. MacLaren, Spin-dependent tunneling conductance of Fe|MgO|Fe sandwiches, Phys. Rev. B 63 (2001) 054416.
[15] A.V. Khvalkovskiy, D. Apalkov, S. Watts, R. Chepulskii, R.S. Beach, A. Ong, et al., Basic principles of STT-MRAM cell operation in memory arrays, J. Phys. D. App. Phys 46 (2013) 139601.
[16] S. Yuasa, D.D. Djayaprawira, Giant tunnel magnetoresistance in magnetic tunnel junctions with a crystalline MgO(001) barrier, J. Phys. D. Appl. Phys. (2007) 337–354.
[17] D.D. Djayaprawira, K. Tsunekawa, M. Nagai, H. Maehara, S. Yamagata, N. Watanabe, et al., 230% room-temperature magnetoresistance in CoFeB/MgO/CoFeB magnetic tunnel junctions, Appl. Phys. Lett. 86 (2005) 092502.
[18] J.C. Slonczewski, Current-driven excitation of magnetic multilayers, JMMM 159 (1996) L1–L7.
[19] L. Berger, Emission of spin waves by a magnetic multilayer traversed by a current, Phys. Rev. B 54 (1996) 9353.
[20] M. Stiles, J. Miltat, Spin DYNamics in Confined Magnetic Structures III, Springer, 2006.
[21] D.C. Ralph, M.D. Stiles, Spin transfer torques, J. Magnetism Magnetic Mater. 320 (2008) 1190–1216.
[22] J. Sun, D. Ralph, Magnetoresistance and spin-transfer torque in magnetic tunnel junctions, J. Magnetism Magnetic Mater. 320 (7) (2008) 1227–1237.
[23] J.Z. Sun, T.S. Kuan, J.A. Katine, R.H. Koch, Spin angular momentum transfer in a current-perpendicular spin-valve nanomagnet, in: SPIE Proceedings Vol. 5359: Quantum Sensing and Nanophotonic Devices, 2004.
[24] Z. Diao, Z. Li, S. Wang, Y. Ding, A. Panchula, E. Chen, et al., Spin-transfer torque switching in magnetic tunnel junctions and spin-transfer torque random access memory, J. Physics Condens. Matter 19 (16) (2007).
[25] T. Devolder, P. Crozat, J.-V. Kim, C. Chappert, Magnetization switching by spin torque using subnanosecond current pulses assisted by hard axis magnetic fields, Appl. Phys. Lett. 88 (2006) 152502.
[26] J.Z. Sun, P.L. Trouilloud, M.J. Gajek, J. Nowak, R.P. Robertazzi, G. Hu, et al., Size dependence of spin-torque induced magnetic switching in CoFeB-based perpendicular magnetization tunnel junctions, J. Appl. Phys. 111 (07C711) (2012).

[27] H. Sato, E.C.I. Enobio, M. Yamanouchi, S. Ikeda, S. Fukami, S. Kanai, et al., Properties of magnetic tunnel junctions with a MgO/CoFeB/Ta/CoFeB/MgO recording structure down to junction diameter of 11nm, Appl. Phys. Lett. 105 (062403) (2014).
[28] A.F. Vincent, N. Locatelli, J.-O. Klein, W.S. Zhao, S. Galdin-Retailleau, D. Querlioz, Analytical macrospin modeling of the stochastic switching time of spin-transfer torque devices, IEEE Trans. Electron. Devices 62 (2015) 164–170.
[29] A. Hubert, R. Schäfer, Magnetic Domains: The Analysis of Magnetic Microstructures, Springer, 2009.
[30] S.S.P. Parkin, M. Hayashi, L. Thomas, Magnetic domain-wall racetrack memory, Science 320 (2008) 190.
[31] M. Weisheit, S. Faehler, A. Marty, Y. Souche, C. Poinsignon, D. Givord, Electric field-induced modification of magnetism in thin-film, Science 315 (2007) 349–351.
[32] W.-G. Wang, M. Li, S. Hageman, C.L. Chien, Electric-field-assisted switching in magnetic tunnel junctions, Nat. Mat. 11 (2012) 64–68.
[33] A. Bernand-Mantel, L.H. Diez, L. Ranno, S. Pizzini, J. Vogel, D. Givord, et al., Electric-field control of domain wall nucleation and pinning in a metallic ferromagnet, Appl. Phys. Lett. 102 (2012) 122406.
[34] Y.T. Liu, S. Ono, G. Agnus, J.-P. Adam, S. Jaiswal, J. Langer, et al., Electric field controlled domain wall dynamics and magnetic easy axis switching in CoFeB/MgO films, J. Appl. Phys. 122 (2017) 133907.
[35] U. Bauer, S. Emori, G.S.D. Beach, Voltage-controlled domain wall traps in ferromagnetic nanowires, Nat. Nanotechnol. 8 (2013) 411–416.
[36] D.A. Allwood, G. Xiong, C.C. Faulkner, D. Atkinson, D. Petit, R.P. Cowburn, Magnetic domain-wall logic, Science 309 (5741) (2005) 1688–1692.
[37] S. Murakami, N. Nagaosa, S.-C. Zhang, Dissipationless quantum spin current at room temperature, Science 301 (2003) 5638.
[38] M. Dyakonov, V. Perel, Posibility of orienting electrons spins with current, Phys. Lett. A 35 (1971) 459–460.
[39] L. Liu, C.-F. Pai, Y. Li, H.W. Tseng, D.C. Ralph, R.A. Buhrman, Spin-torque switching with the giant spin hall effect of tantalum, Science 336 (2012) 555–558.
[40] K.M. Rabe, C.H. Ahn, J.-M. Triscone, Physics of Ferroelectrics, Springer, Berlin, Heidelberg, 2007.
[41] K. Prall, Benchmarking and metrics for emerging memory, in: IEEE International Memory Workshop (IMW), Monterey, CA, 2017, pp. 1–5.
[42] R.E. Cohen, Origin of ferroelectricity in perovskite oxides, Nature 358 (1992) 136–138.
[43] J. Scott, Switching kinetics of lead zirconate titanate submicron thin-film memories, J. Appl. Phys. 64 (1998) 787.
[44] J.R. Anderson, Ferroelectric materials as storage elements for digital computers and switching systems, Trans. Amer. Inst. Elect. Engrs. 71, Part. I: Commun. Electron. (1953) 395–401.
[45] D. Bondurant, Ferroelectronic RAM memory family for critical data storage, Ferroelectrics 112 (1990) 273–282.
[46] C.A.-P. de Araujo, J.D. Cuchiaro, L.D. McMillan, M.C. Scott, J. Scott, Fatigue-free ferroelectric capacitors with platinum electrodes, Nature 374 (1995) 627–629.
[47] M. Mört, G. Schindler, W. Hartner, I. Kasko, M.J. Kastner, T. Mikolajick, et al., Low temperature process and thin SBT films for ferroelectric memory devices, Ferroelectrics 30 (1-4) (2000) 235–244.
[48] J.S. Zhao, et al., Metallorganic CVD of high-quality PZT thin films, J. Electrochem. Soc. (2004) C283–C291.

[49] T. Oikawa, H. Morioka, A. Nagai, H. Funakubo, Thickness scaling of polycrystalline Pb (Zr, Ti)O3 films downto 35nm prepared by metalorganic chemical vapor depositionhaving good ferroelectric properties, Appl. Phys. Lett. 85 (10) (2004) 1754−1756.

[50] W. Hartner, G. Schindler, P. Bosk, Z. Gabric, M. Kastner, G. Beitel, et al., Integration of H2 barriers for ferroelectric memories based on SrBi2Ta2O9 (SBT), Integr. Ferroelectr. 31 (1-4) (2000) 273−284.

[51] J.-M. Koo, B.-S. Seo, S. Kim, S. Shin, J.-H. Lee, H. Baik, et al., Fabrication of 3D trench PZT capacitors for 256Mbit FRAM device application, IEDM Techn. Dig. (2005) 340−343.

[52] H. McAdams, R. Acklin, T. Blake, X.-H. Du, J. Eliason, J. Fong, et al., A 64-Mb embedded FRAM utilizing a 130-nm 5LM Cu/FSG logic process, IEEE J. Solid-St. Circ. 39 (2004) 667−677.

[53] T. Boescke, J. Mueller, D. Braeuhaus, U. Schroeder, U. Boettger, Ferroelectricity in hafnium oxide thin films, Appl. Phys. Lett. 99 (2011) 102903.

[54] U. Schroeder, E. Yurchuk, J. Müller, D. Martin, T. Schenk, P. Polakowski, et al., Impact of different dopants on the switching properties of ferroelectric hafniumoxide, Japanese J. Appl. Phys. 53 (8S1) (2014). 08LE02-1.

[55] T. Olsen, U. Schröder, S. Müller, A. Krause, D. Martin, A. Singh, et al., Co-sputtering yttrium into hafnium oxide thin films to produce ferroelectric properties, Appl. Phys. Lett. 101 (8) (2012) 082905.

[56] L. Xu, S. Shibayama, K. Izukashi, T. Nishimura, T. Yajima, S. Migita, et al., General relationship for cation and anion doping effects on ferroelectric HfO formation, in: IEEE International Electron Devices Meeting (IEDM), San Francisco, CA, 2016, pp. 25.2.1−25.2.4.

[57] S. Starschich, U. Boettger, An extensive study of the influence of dopants on the ferroelectric properties of HfO 2, J. Mater. Chem. C. 5 (2017) 333−338.

[58] M.H. Park, Y.H. Lee, H.J. Kim, Y.J. Kim, T. Moon, K.D. Kim, et al., Understanding the formation of the metastable ferroelectric phase in hafnia−zirconia solid solution thin films, Nanoscale 10 (2018) 716−725.

[59] X. Sang, E.D. Grimley, T. Schenk, U. Schroeder, J.M. LeBeau, On the structural origins of ferroelectricity in HfO2 thin films, Appl. Phys. Lett. 106 (2015) 162905.

[60] T. Mittmann, F.P. Fengler, C. Richter, M.-H. Park, T. Mikolajick, U. Schroeder, Optimizing process conditions for improved $Hf_{1-x}Zr_xO_2$ ferroelectric capacitor performance, Microelectronic Eng. 178 (2017) 48−51.

[61] T. Mikolajick, S. Slesazeck, M.H. Park, U. Schroeder, Ferroelectric hafnium oxide for ferroelectric random-access memories and ferroelectric field-effect transistors, MRS Bull. 43 (5) (2018) 340−346.

[62] J. Muller, T. Boscke, S. Muller, E.Y.P. Polakowski, J. Paul, D. Martin, et al., Ferroelectric hafnium oxide: a CMOS-compatible and highly scalable approach to future ferroelectric memories, in: IEEE International Electron Devices Meeting (IEDM), 2013, pp. 10.8.1−10.8.4.

[63] P. Polakowski, S. Riedel, W. Weinreich, M. Rudolf, J. Sundqvist, K. Seidel, et al., Ferroelectric deep trench capacitors based on Al:HfO2 for 3D nonvolatile memory applications, in: IEEE 6th International Memory Workshop (IMW), 2014, pp. 1−4.

[64] U. Schroeder, M. Pešic, T. Schenk, H. Mulaosmanovic, S. Slesazeck, J. Ocker, et al., Impact of field cycling on HfO2 based non-volatile memory devices, in: 46th European Solid-State Device Research Conference (ESSDERC), 2016, pp. 364−368.

[65] F.P. Fengler, M. Pešić, S. Starschich, T. Schneller, C. Künneth, U. Böttger, et al., Domain pinning: comparison of Hafnia and PZT based ferroelectrics, Adv. Electron. Mater. 3 (4) (2017) 1600505.

[66] F. Fengler, M. Hoffmann, S. Slesazeck, T. Mikolajick, U. Schroeder, On the relationship between field cycling and imprint in ferroelectric Hf0.5Zr0.5O2, J. Appl. Phys. 123 (20) (2018) 204101.
[67] J. Müller, T.S. Böscke, U. Schröder, S. Mueller, D. Bräuhaus, U. Böttger, et al., Ferroelectricity in simple binary ZrO2 and HfO2, Nano Lett. 12 (8) (2012) 4318–4323.
[68] X. Liu, D. Zhou, Y. Guan, et al., Endurance properties of silicon-doped hafnium oxide ferroelectric and antiferroelectric-like thin films: a comparative study and prediction, Acta Materialia 154 (1) (2018) 190–198.
[69] M. Pešić, M. Hoffmann, C. Richter, T. Mikolajick, U. Schroeder, Nonvolatile random access memory and energy storage based on antiferroelectric like hysteresis in ZrO2, Adv. Funct. Mater. 26 (41) (2016) 7486–7494.
[70] M. Pesic, S. Knebel, M. Hoffmann, C. Richter, T. Mikolajick, U. Schroeder, How to make DRAM non-volatile? Anti-ferroelectrics: a new paradigm for universal memories, IEEE International Electron Devices Meeting (IEDM), 2016, pp. 11.6.1–11.6.4.
[71] M. Pešić, U. Schroeder, S. Slesazeck, T. Mikolajick, Comparative study of reliability of ferroelectric and anti-ferroelectric memories, IEEE Trans. Device Mater. Reliab. 18 (2) (2018) 154–162.
[72] M. Pešić, T. Li, V.D. Lecce, M. Hoffmann, M. Materano, C. Richter, et al., Built-in bias generation in anti-ferroelectric stacks: methods and device applications, IEEE J. Electron. Devices Soc. 6 (2018) 1019–1025.
[73] T. Mikolajick, Ferroelectric nonvolatile memories, Ref. Module Mater. Sci. Mater. Eng. (2016) 1–5.
[74] T. Ma, J.-P. Han, Why is nonvolatile ferroelectric memory field-effect transistor still elusive, IEEE Electron. Device Lett. 23 (2002) 386–388.
[75] N. Gong, T. Ma, Why is FE–HfO2 more suitable than PZT or SBT for scaled nonvolatile 1-T memory cell? A retention perspective, IEEE Electron. Device Lett. 37 (9) (2016) 1123–1126.
[76] J. Müller, J. Müller, E. Yurchuk, T. Schlösser, J. Paul, R. Hoffmann, et al., Ferroelectricity in HfO2 enables nonvolatile data storage in 28 nm HKMG, Proc. IEEE Symposia VLSI Technol. (2012) 25–26.
[77] M. Trentzsch, S. Flachowsky, R. Richter, J. Paul, B. Reimer, D. Utess, et al., A 28nm HKMG super low power embedded NVM technology based on ferroelectric FETs, in: IEEE International Electron Devices Meeting (IEDM), 2016, pp. 11.5.1–11.5.4.
[78] S. Dünkel, M. Trentzsch, R. Richter, P. Moll, C. Fuchs, O. Gehring, et al., A FeFET based super-low-power ultra-fast embedded NVM technology for 22nm FDSOI and beyond, in: IEEE International Electron Devices Meeting (IEDM), 2017, pp. 19.7.1–19.7.4.
[79] B. Max, M. Pešić, S. Slesazeck, T. Mikolajick, Interplay between ferroelectric and resistive switching in doped crystalline HfO2, J. Appl. Phys. 123 (2018) 134102.
[80] J. Jiang, Z.L. Bai, Z.H. Chen, L. He, D.W. Zhang, Q.H. Zhang, et al., Temporary formation of highly conducting domain walls for non-destructive read-out of ferroelectric domain-wall resistance switching memories, Nat. Mater. 17 (2018) 49–56.
[81] L. Esaki, R.B. Laibowitz, P.J. Stiles, Polar switch, IBM Tech. Discl. Bull. 13 (1971) 2161.
[82] E.Y. Tsymbal, H. Kohlstedt, Tunneling across a ferroelectric, Sci. 14 313 (5784) (2006) 181–183.
[83] V. Garcia, M. Bibes, Ferroelectric tunnel junctions for information storage and processing, Nat. Commun. 5 (2014) 4289.
[84] A. Chernikova, M. Kozodaev, A. Markeev, D. Negrov, M. Spiridonov, S. Zarubin, et al., Ultrathin Hf0.5Zr0.5O2 ferroelectric films on Si, ACS Appl. Mater. Interfaces 8 (11) (2016) 7232–7237.

[85] X. Tian, S. Shibayama, T. Nishimura, T. Yajima, S. Migita, A. Toriumi, Evolution of ferroelectric HfO2 in ultrathin region down to 3 nm, Appl. Phys. Lett. 112 (2018) 102902.
[86] S. Fujii, Y. Kamimuta, T. Ino, Y. Nakasaki, R. Takaishi, M. Saitoh, First demonstration and performance improvement of ferroelectric HfO2-based resistive switch with low operation current and intrinsic diode property, IEEE Symposium VLSI Technol. (2016) 1−2.
[87] B. Max, M. Hoffmann, S. Slesazeck, T. Mikolajick, Ferroelectric tunnel junctions based on ferroelectric-Dielectric Hf0.5Zr0.5.O2/ Al2O3 capacitor stacks, in: 48th European Solid-State Device Research Conference (ESSDERC), 2018, pp. 142−145.
[88] S. Slesazeck, V. Havel, E. Breyer, H. Mulaosmanovic, M. Hoffmann, B. Max, et al., Uniting the trinity of ferroelectric HfO2 memory devices in a single memory cell, in: IEEE International Memory Workshop (IMW), 2019, pp. 1−4.
[89] S.B. et al., Embedded FeFETs as a low power and non-volatile beyond-von-Neumann memory solution, in: Nonvolatile Memory Technology Symposium (NVMTS), 2018, pp. 28−29.
[90] W. Zhao, M. Moreau, E. Deng, Y. Zhang, J.-M. Portal, J.-O. Klein, et al., Synchronous non-volatile logic gate design based on resistive switching memories, IEEE Trans. Circuits Syst. I: Regul. Pap. 61 (2) (2013) 443−454.
[91] E.T. Breyer, H. Mulaosmanovic, T. Mikolajick, S. Slesazeck, Reconfigurable NAND/ NOR logic gates in 28 nm HKMG and 22 nm FD-SOI FeFET technology, in: IEEE International Electron Devices Meeting (IEDM), 2017, pp. 28.5.1−28.5.4.
[92] E.T. Breyer, H. Mulaosmanovic, S. Slesazeck, T. Mikolajick, Demonstration of versatile nonvolatile logic gates in 28nm HKMG FeFET technology, in: IEEE International Symposium on Circuits and Systems (ISCAS), 2018, pp. 1−5.
[93] T. Hanyu, others, Standby-power-free integrated circuits using MTJ-based VLSI computing, Proc. IEEE 104 (2016) 1844−1863.
[94] W. Kang, E. Deng, Z. Wang, W. Zhao, Spintronic logic-in-memory paradigms and implementations, Applications of Emerging Memory Technology, Springer, Singapore, 2019, pp. 215−229.
[95] S. Matsunaga, J. Hayakawa, S. Ikeda, K. Miura, H. Hasegawa, T. Endoh, et al., Fabrication of a nonvolatile full adder based on logic-in-memory architecture using magnetic tunnel junctions, Appl. Phys. Express 1 (2008) 091301.
[96] S. Jain, A. Ranjan, K. Roy, A. Raghunathan, Computing in memory with spin-transfer torque magnetic RAM, IEEE Trans. VLSI Syst. 26 (2018) 470−483.
[97] L. Wang, W. Kang, F. Ebrahimi, X. Li, Y. Huang, C. Zhao, et al., Voltage-controlled magnetic tunnel junctions for processing-in-memory implementation, IEEE Electron. Device Lett. 39 (2018) 440−443.
[98] V. Saxena, X. Wu, I. Srivastava, K. Zhu, Towards neuromorphic learning machines using emerging memory devices with brain-like energy efficiency, J. Low. Power Electron. Appl. 8 (4) (2018) 34.
[99] D. Ielmini, Brain-inspired computing with resistive switching memory (RRAM): devices, synapses and neural networks, Microelectronic Eng. 190 (2018) 44−53.
[100] E. Vianello, T. Werner, G. Piccolboni, D. Garbin, O. Bichler, G. Molas, et al., Binary OxRAM/CBRAM memories for efficient implementations of embedded neuromorphic circuits, in: S. Yu (Ed.), Neuro-Inspired Computing Using Resistive Synaptic Devices, *Springer*, *Cham*, 2017, pp. 253−269.
[101] T. You, N. Du, S. Slesazeck, T. Mikolajick, G. Li, D. Bürger, et al., Bipolar electric-field enhanced trapping and detrapping of mobile donors in BiFeO3 memristors, ACS Appl. Mater. & interfaces 6 (22) (2014) 19758-−119765.

[102] H. Mähne, H. Wylezich, F. Hanzig, S. Slesazeck, D. Rafaja, T. Mikolajick, Analog resistive switching behavior of Al/Nb2O5/Al device, Semiconductor Sci. Technol. 29 (10) (2014) 104002.
[103] R. Symanczyk, M. Balakrishnan, C. Gopalan, T. Happ, M. Kozicki, M. Kund, et al., Electrical characterization of solid state ionic memory elements, in: Proceedings of the non-volatile memory technology symposium (NVMTS), 2003, pp. 1–17.
[104] A. Sebastian, M.L. Gallo, G.W. Burr, S. Kim, M. BrightSky, E. Eleftheriou, Tutorial: brain-inspired computing using phase-change memory devices, J. Appl. Phys. 124 (2018) 111101.
[105] S. Lequeux, J. Sampaio, V. Cros, K. Yakushiji, A. Fukushima, R. Matsumoto, et al., A magnetic synapse: multilevel spin-torque memristor with perpendicular anisotropy, Sc. Rep. 6 (2016) 31510.
[106] D. Querlioz, O. Bichler, A.F. Vincent, C. Gamrat, Bioinspired programming of memory devices for implementing an inference engine, Proc. IEEE 103 (2015) 1398–1416.
[107] Y. M. a. T. Endoh, A novel neuron circuit with nonvolatile synapses based on magnetic-tunnel-junction for high-speed pattern learning and recognition, in: Proc. Asia-Pacific Workshop Fundam. Appl. Adv. Semicond. Devices, vols. 4B-1, 2015, pp. 273–228.
[108] J. Grollier, D. Querlioz, M.D. Stiles, Spintronic nanodevices for bioinspired computing, Proc. IEEE 104 (2016) 2024–2039.
[109] M. Sharad, C. Augustine, G. Panagopoulos, K. Roy, Spin-based neuron model with domain-wall magnets as synapse, IEEE Trans. Nanotechnol. 11 (2012) 843–853.
[110] X. Wang, Y. Chen, H. Xi, H. Li, D. Dimitrov, Spintronic memristor through spin-torque-induced magnetization motion, IEEE Electron. Device Lett. 30 (2009) 294–297.
[111] J. Münchenberger, G. Reiss, A. Thomas, A memristor based on current-induced domain-wall motion in a nanostructured giant magnetoresistance device, J. Appl. Phys. 111 (2012) 07D303.
[112] N. Locatelli, V. Cros, J. Grollier, Spin-torque building blocks, Nat. Mater. 13 (2014) 11–20.
[113] H. Mulaosmanovic, J. Ocker, S. Mueller, U. Schroeder, J. Mueller, P. Polakowski, et al., Switching kinetics in nanoscale hafnium oxide based ferroelectric field-effect transistors, ACS Appl. Mater. Interfaces 9 (4) (2017) 3792–3798.
[114] H. Mulaosmanovic, S. Slesazeck, J. Ocker, M. Pesic, S. Muller, S. Flachowsky, et al., Evidence of single domain switching in hafnium oxide based FeFETs: Enabler for multi-level FeFET memory cells, in: IEEE International Electron Devices Meeting (IEDM), 2015, pp. 26.8.1–26.8.3.
[115] H. Mulaosmanovic, T. Mikolajick, S. Slesazeck, Accumulative polarization reversal in nanoscale ferroelectric transistors, ACS Appl. Mater. Interfaces 10 (28) (2018) 23997–24002.
[116] Y. Kaneko, Y. Nishitani, M. Ueda, A. Tsujimura, Neural network based on a three-terminal ferroelectric memristor to enable on-chip pattern recognition, Symposium VLSI Technol. (2013) T238–T239.
[117] H. Mulaosmanovic, J. Ocker, S. Müller, M. Noack, J. Müller, P. Polakowski, et al., Novel ferroelectric FET based synapse for neuromorphic systems, Symposium VLSI Technol. (2017) T176–T177.
[118] S. Oh, T. Kim, M. Kwak, J. Song, J. Woo, S. Jeon, et al., HfZrOx-based ferroelectric synapse device with 32 levels of conductance states for, IEEE Electron. Device Lett. 38 (6) (2017) 732–735.
[119] M. Jerry, S. Dutta, A. Kazemi, K. Ni, J. Zhang, P.-Y. Chen, et al., A ferroelectric field effect transistor based synaptic, J. Phys. D Appl. Phys 51 (2018) (2018) 43400–434001.

[120] S. Boyn, J. Grollier, G. Lecerf, B. Xu, N. Locatelli, S. Fusil, et al., Learning through ferroelectric domain dynamics in solid-state synapses, Nat. Commun. 8 (2017) 14736.
[121] L. Chen, T.-Y. Wang, Y.-W. Dai, M.-Y. Cha, H. Zhu, Q.-Q. Sun, et al., Ultra-low power Hf0.5Zr0.5O2 based ferroelectric tunnel junction synapses for hardware neural network applications, Nanoscale 10 (2018) 15826−15833.
[122] S. Slesazeck, T. Mikolajick, Nanoscale resistive switching memory devices: a review, Nanotechnology 30 (35) (2019) 352003.
[123] H.M.E. Chicca, M. Bertele, T. Mikolajick, S. Slesazeck, Mimicking biological neurons with a nanoscale ferroelectric transistor, Nanoscale (2018).
[124] H. Mulaosmanovic, T. Mikolajick, S. Slesazeck, Random number generation based on ferroelectric switching, IEEE Electron. Device Lett. 39 (1) (2018) 135−138.
[125] D. Vodenicarevic, N. Locatelli, A. Mizrahi, J. S. Friedman, A. F. Vincent, M. Romera, et al., Low-energy truly random number generation with superparamagnetic tunnel junctions for unconventional computing, Phys. Rev. Appl. 8 (2017) 054045.
[126] D. Vodenicarevic, N. Locatelli, A. Mizrahi, T. Hirtzlin, J.S. Friedman, J. Grollier, et al., Circuit-level evaluation of the generation of truly random bits with superparamagnetic tunnel junctions, IEEE Int. Symposium Circuits Syst. (ISCAS) (2018) 1−4.
[127] J. Han, M. Orshansky, Approximate computing: an emerging paradigm for energy-efficient design, in: 18TH IEEE European Test Symposium (ETS), 2013, pp. 1−6.
[128] K. Lingamneni, A. Palem, Ten years of building broken chips: the physics and engineering of inexact computing, ACM Trans. Embedded Comput. Syst. (TECS) - Spec. Sect. Probabilistic Embedded Comput. 12 (2013).
[129] S. Manipatruni, D.E. Nikonov, I.A. Young, Beyond CMOS computing with spin and polarization, Nat. Phys. 14 (2018) 338−343.
[130] N. Locatelli, A.F. Vincent, D. Querlioz, Use of magnetoresistive random-access memory as approximate memory for training neural networks, arXiv:1810.10836.
[131] S. Salahuddin, Datta, Use of negative capacitance to provide voltage amplification for low power nanoscale devices, Nano Lett. 8 (2) (2008) 405−410.
[132] M. Hoffmann, S.S.T. Mikolajick, C.S. Hwang, Negative capacitance in fluorite-type ferroelectrics, in: U. Schroeder, C.S. Hwang, H. Funakubo (Eds.), Ferroelectricity in Doped Hafnium Oxide—Materials, Properties and Device, Woodhead Publishing, 2019.

Chapter 5

Selector devices for emerging memories

Solomon Amsalu Chekol, Jeonghwan Song, Jaehyuk Park, Jongmyung Yoo, Seokjae Lim and Hyunsang Hwang

Center for Single Atom-based Semiconductor Device and Department of Materials Science and Engineering, Pohang University of Science and Technology (POSTECH), Pohang-si, South Korea

5.1 Introduction

Following the technological limitations of device scaling, new memories, such as resistive switching memory (RRAM) [1], phase-change memory (PCM) [2], ferroelectric RAM (FeRAM) [3], and spin-torque-transfer magnetic RAM (STT-MRAM) [4], have been emerging as potential replacements for existing memory devices and novel computation schemes beyond von Neumann and brain-inspired computing. Further details on the operational principle of RRAM, PCM, FeRAM, and STT-MRAM are reported, respectively, in previous Chapters 2–4. These emerging memory devices, also recently named memristive devices, ought to be capable of implementing extremely high-density memory when integrated into a crossbar array with an effective cross section of $\sim 4F^2$, where F is the minimum feature size [5]. The recent announcement of Intel's 3D nonvolatile memory, the X-Point, is an example of this concept [6]. This memory, which is based on a combination of a PCM with an ovonic threshold switching (OTS) selector element, is much faster and more scalable than the conventional Flash memory. The operation of such large arrays involves a voltage drop on specifically selected cells without disturbing any other devices. When voltage is applied to the line edges of the row and column of the array containing the target cell, a net voltage drop in the selected cell is expected. Depending on the amount of the net voltage drop, successful programming, erasing, or reading of the selected cell can be achieved. Crossbar arrays consisting of numerous memristive devices, referred to as 1R, suffer from undesired data reading failures and high power consumption due to the nonzero net current running through the unselected cells as shown in Fig. 5.1B.

Memristive Devices for Brain-Inspired Computing. DOI: https://doi.org/10.1016/B978-0-08-102782-0.00005-8

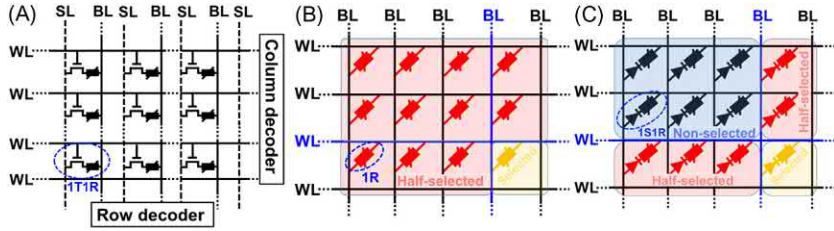


FIGURE 5.1 (A) Memory cell array with discretely connected transistors (1T1R). (B) A cross-point array containing only resistors (1R). When a bias is applied in a single row and column, the whole cells in the array will be affected. (C) Array of a one-selector and one-resistor (1S1R) structure showing selected cell, affected cells in the same row and column (half-selected), and others (nonselected cells).

To tackle such problem, different device structures have been introduced. One method is the use of a transistor (T) with an adjacent memory device (R), from now on referred to as 1T1R (Fig. 5.1A). While transistors can effectively block the unnecessary leakage current, the scalability and achieving of a highly dense array become problematic due to scaling limits of the transistor.

Therefore access devices called selectors that have a simple two-terminal structure are essential to suppress such leakage current by adding discretely at each cross-point in a one-selector one-resistor architecture, referred in the following as 1S1R (Fig. 5.1C) [7]. The selector device in the following also named 1S is turned on when a voltage above the threshold value is applied, leading to an abrupt resistance decrease, and thus supplies sufficient current to program/erase the selected memory cell without affecting other cells in the array. However, in the low-voltage regime, the selector retains its OFF state, which effectively prevents the sneak path current.

The issue of sneak path current becomes more troublesome for applications on novel computation schemes beyond the von Neumann architecture and brain-inspired computing systems. Mostly, these devices work in an analog fashion with many tightly separated memory states in a single device. Reading and writing of such memory states require precise control of applied voltage to the selected cells in a cross-point array. Therefore selector devices are very important in minimizing leakage currents through the nonselected cells to deliver the right voltage to the selected cell.

Nevertheless, an ideal selector device has to fulfill some requirements, such as extremely low OFF current (I_{OFF}), high enough ON current (I_{ON}), fast switching speeds, infinite cycling endurance, excellent device uniformity, large voltage margins (voltage window separating the ON and OFF states), compatible operating voltage conditions with memristive devices, low-temperature fabrication processes, high thermal stability, and 3D stackable two-terminal device structure. CMOS transistors are the best candidates

to fulfill most of the required electrical characteristics, but three-terminal devices are not suitable for ultra-high density crossbar array architectures. Therefore several types of new selector devices, such as the tunnel barrier-type device, mixed ionic−electronic conduction (MIEC) device, OTS, insulator−metal transition (IMT), and conductive bridging RAM (CBRAM)-type selector device [8], have been suggested to fulfill the aforementioned criteria. Among them, this chapter introduces threshold-type selector devices, such as IMT, OTS, and CBRAM type, and discusses directions for the future development of selector devices.

5.2 Insulator−metal transition selector

IMT is a unique phenomenon observed in certain transition oxides (Fig. 5.2A) in which the materials show a large conductivity change as the result of a phase transition [9]. The phase transition can be triggered by different mechanisms, such as thermal [11], optical [12], and electrical [13]. The most common approach to induce IMT is by temperature, that is, heating and cooling the material. These materials normally show a highly insulating state at low temperature because of their distorted structure. However, above a certain critical temperature, the resistance drops and there will be an abrupt transition from an insulative to metallic state. The reversible fast transition from a conductive state to an insulative state makes IMT materials ideal for selector device applications.

Recently, there have been interests on IMT devices for selector device application in crossbar arrays thanks to their simple metal/insulator/metal structure, bidirectional threshold characteristics, and fast transition speed

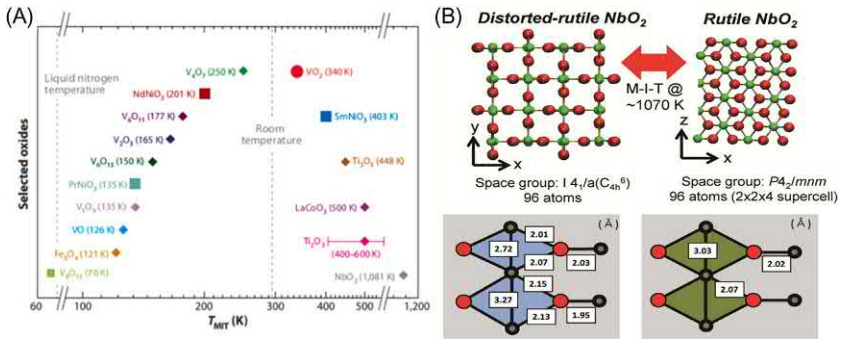


FIGURE 5.2 (A) The figure shows a number of selected IMT oxides with respect to their transition temperature. (B) Schematic diagram showing the unit cell of distorted-rutile NbO_2 (left) and rutile NbO_2 (right). *Adapted with permission from (A) J. Wei, Z. Wang, W. Chen and D.H. Cobden, New aspects of the metal-insulator transition in single-domain vanadium dioxide nanobeams, Nat. Nanotechnol. 4 (2009) 420−424; (B) E. Cha, J. Woo, D. Lee et al., Nanoscale (<10 nm) 3D vertical ReRAM and NbO_2 threshold selector with TiN electrode, in: Proc. IEEE Int. Electron Devices Meeting, December 2013, pp. 10.5.1−10.5.4 [10].*

(Fig. 5.3) [13,14]. VO_2 [13] and NbO_2 [14] are the two most studied materials for IMT-based selector device.

When the applied voltage exceeds a certain threshold voltage (V_{th}) (see Fig. 5.3B), the device changes to the metallic state from an insulating state. The switching mechanism of IMT is usually interpreted as a result of Joule heating due to current flow in the device during the application of an electric field. When the internal temperature from Joule heating exceeds 340K and 1080K for VO_2 and NbO_2, respectively, these devices exhibit a transition to a metallic state. Even though VO_2 has been proved to show good IMT characteristics, its feasibility as a selector is limited because of its low transition temperature ($\sim$340K, Fig. 5.2A) [15]. On the other hand, NbO_2 has a relatively high transition temperature (1080K, Fig. 5.2A), which makes it suitable for selector application [16,17].

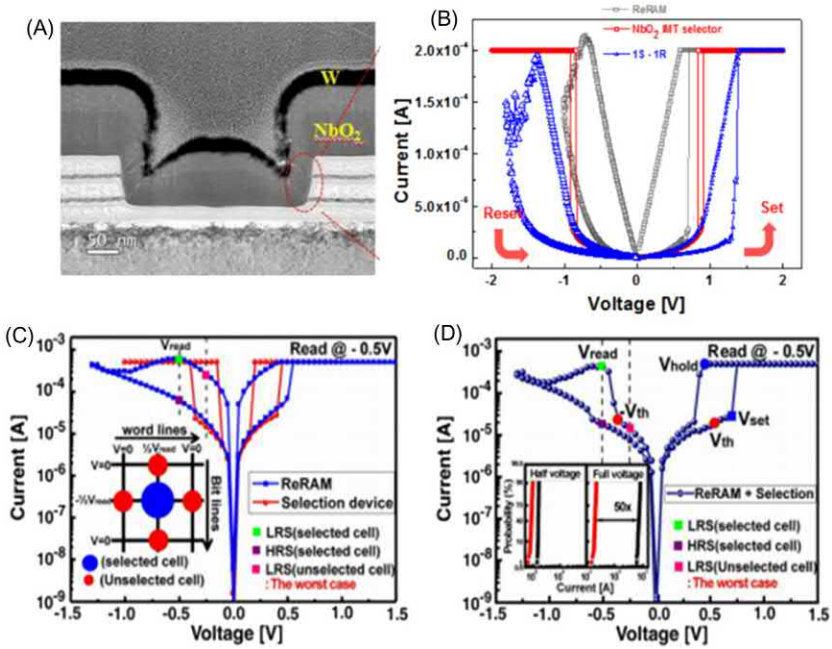


FIGURE 5.3 (A) Cross-sectional TEM image of NbO_2 based selector having a W electrode and (B) typical $I-V$ characteristics of a selector (NbO_2), a Ta_2O_5-based RRAM and one-selector one-ReRAM (1S1R) integrated device. (C) $I-V$ characteristics of ZrO_x/HfO_x-based RRAM and $Pt/VO_2/Pt$ selector devices. (D) $I-V$ characteristics of the integrated (1S1R) device. *Adapted with permission from (A and B) E. Cha, J. Woo, D. Lee, S. Lee, J. Song, Y. Koo, et al., Nanoscale (<10 nm) 3D vertical ReRAM and NbO_2 threshold selector with TiN electrode, in: 2013 IEEE International Electron Devices Meeting, Washington, DC, 2013, pp. 10.5.1–10.5.4; (C and D) M. Son, J. Lee, J. Park, J. Shin, G. Chio, S. Jung, et al., Excellent selector characteristics of nanoscale VO_2 for high-density bipolar ReRAM applications, Electron. Device Lett. 32 (11) (2011) 1579–1581.*

NbO_2-based selectors show interesting characteristics, such as highly uniform switching, fast operating speed (<50 ns), and high thermal stability (>430K). In addition, scalability and tunability of different switching parameters, such as forming voltage (V_F), V_{th}, and I_{OFF}, can be regulated by careful controlling of film thickness and device area. Cha et al. studied the scalability and tunability of IMT devices using TiN/NbO_2/W device [18]. According to the report, both V_{th} and V_f can be reduced by scaling the thickness of the NbO_2 film and the I_{OFF} can be reduced by reducing the device size as shown in Fig. 5.4.

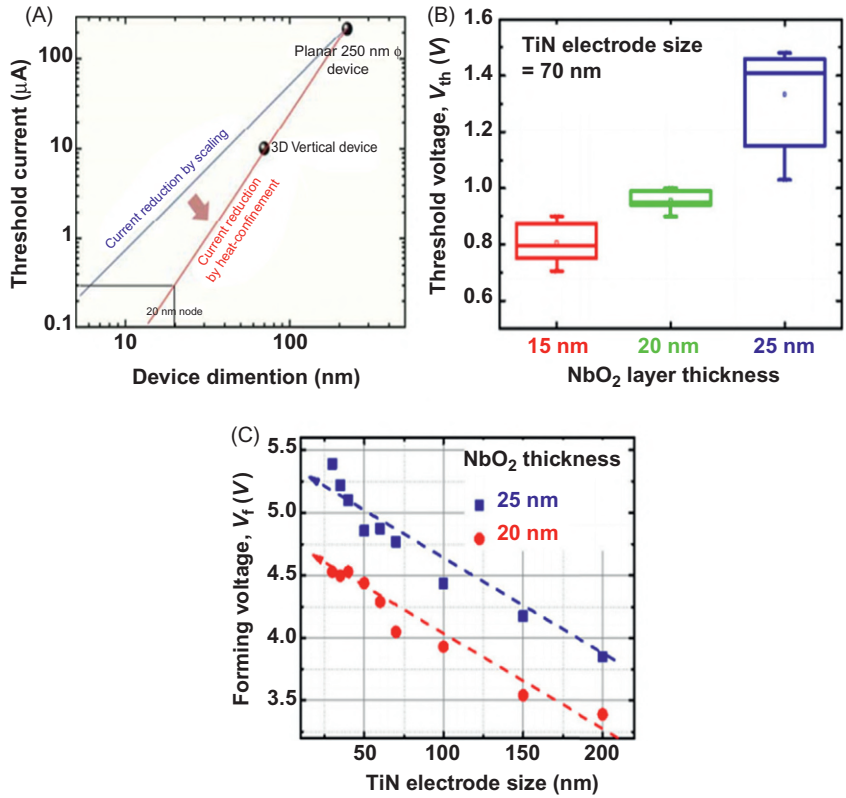


FIGURE 5.4 V_{th} dependence of TiN/NbO_2/W device on device dimension (A) and on NbO_2 film thickness (B), respectively. V_f tendency according to film thickness and device dimension (C). *Adapted with permission from (A) E. Cha, J. Woo, D. Lee et al., Nanoscale (<10 nm) 3D vertical ReRAM and NbO_2 threshold selector with TiN electrode, in: Proc. IEEE Int. Electron Devices Meeting, December 2013, pp. 10.5.1–10.5.4; E. Cha, J. Park, J. Woo, D. Lee, A. Prakash and H. Hwang, Comprehensive scaling study of NbO2 insulator-metal-transition selector for cross point array application, Appl. Phys. Lett. 108 (15) (2016), no. 153502; (B) E. Cha, J. Woo, D. Lee et al., Nanoscale (<10 nm) 3D vertical ReRAM and NbO_2 threshold selector with TiN electrode, in: Proc. IEEE Int. Electron Devices Meeting, December 2013, pp. 10.5.1–10.5.4.*

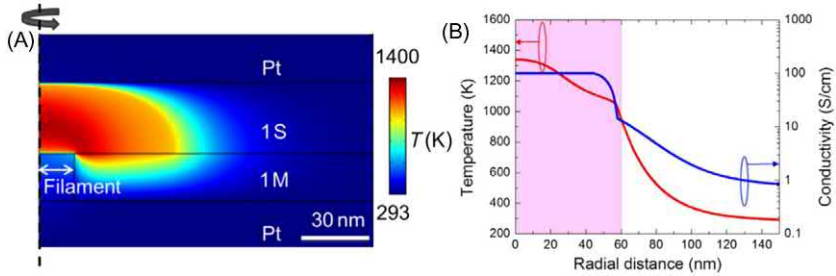


FIGURE 5.5 (A) The temperature distribution in the memory (1M) and NbO_2 selector (1S) at threshold voltage and (B) corresponding temperature and electrical conductivity value in the middle of the NbO_2 film. The simulation shows that the temperature at the center is reaching 1300K. *Adapted with permission from S.K. Nandi, X. Liu, D.K. Venkatachalam and R.G. Elliman, Threshold current reduction for the metal–insulator transition in NbO2 − x selector devices: the effect of ReRAM integration, J. Phys. D: Appl. Phys. 48 (19) (2015).*

Developing a unified model for the IMT mechanism is an effective strategy to improve selector performance, and there have been many studies on the IMT switching mechanism of NbO_2 device [19–22]. The initial attempt tried to explain the phenomenon as a temperature-controlled transition, which said Joule heating induced high temperature in the system, and once the temperature reached transition value (1080K), the device showed metallic characteristics. People who support this claim present simulations of the temperature distribution at the filament as shown in Fig. 5.5 [19]. On the other hand, others argue that the Joule heating model conflicts with the fact that the transition temperature of NbO_2 is much higher than the temperature that can be induced from Joule heating. Instead, they interpret the transition phenomenon using the thermal runaway model [20–22]. Funck et al. did a thorough simulation based on the thermal runaway model along with Poole–Frenkel mechanism, and they found out that the IMT phenomenon in NbO_2 can be started at a much lower temperature ($\sim$320K), rather than the original transition temperature (1080K) as shown in Fig. 5.6 [20]. This can be explained as the combined effect of field-induced barrier lowering and Joule heating effect as shown (Fig. 5.7A).

The negative differential resistance (NDR) behavior (also see Chapter 2: Resistive switching memories, for further details on NDR) of NbO_2 has been debatable as to whether it is caused by a relatively low-temperature non-linear transport mechanism or a high-temperature Mott transition. Recently, Kumar et al. employed a spectromicroscopic characterization technique to measure the actual temperature and chemical distributions during the transition to explain the NDR behavior of NbO_2 [23]. They observed that the NbO_2 film normally exhibits two NDR, current-controlled NDR1 and temperature-controlled NDR2, as shown in Fig. 5.7. The first NDR1 is caused by a Joule heating thermal runaway in a highly nonlinear conduction at a low temperature

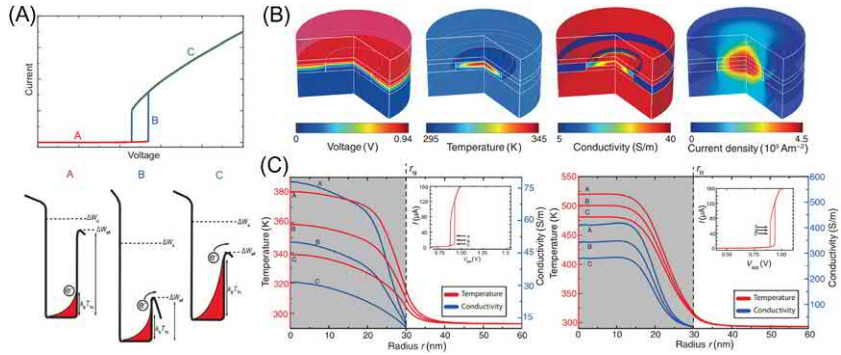


FIGURE 5.6 (A) Schematic diagram of the energy barrier to transit to a metallic state from an insulating state of the NbO_2 device. (B) A thermal simulation of a NbO_2 device during operation shows that the transition can occur at a much lower temperature (~320K) than the transition temperature of the NbO_2 device (~1070K) under an electric field. (C) Radial temperature and conductivity development has taken in the middle of the threshold region for the points A, B, and C in the inset $I-V$ curve during turn-on (left) and turn-off (right). *Adapted with permission from C. Funck, S. Menzel, N. Aslam, H. Zhang, A. Hardtdegen, R. Waser, et al., Multidimensional simulation of threshold switching in NbO2 based on an electric field triggered thermal runaway model, Adv. Elec. Mater. 2 (7) (2016), no. 201600169.*

(~400K). Then, the device exhibits a temperature-controlled NDR2 at a high temperature (~1000K) (Fig. 5.7). Even though both are Joule heating driven and involve temperature as a variable, the physical mechanism is different and occurs at a different temperature.

Besides studying the switching mechanism, device performance improvements using different approaches have been made by several researchers [24–28]. However, despite all the effort, the I_{OFF} of the IMT selector remains relatively high compared with other selector candidates, such as CBRAM-type selectors and OTS selectors. This creates difficulties in utilizing NbO_2 as a selector device in a cross-point array. Researchers have employed different physical analysis tools, such as conductive atomic force microscopy (C-AFM) analysis, to reveal the origin of the leakage current of IMT devices [24,25].

As shown in Fig. 5.8, C-AFM results reveal that structural defects, such as local nonstoichiometric regions, in polycrystalline thin-film and defects along grain boundaries are mainly responsible for the observed high leakage current in the NbO_2 film [24,25]. These defects can generate conduction sub-bands between the conduction band and the valence band of IMT materials (NbO_2, VO_2) and thus lead to high leakage current. Furthermore, interface defects between the electrode and IMT material can induce leakage current because defects can pin the Schottky barrier height [26]. Therefore defective regions and paths in the film should be passivated to further reduce the leakage current. One of the approaches is the insertion of dielectric material as a barrier between the electrodes and IMT film, which can effectively eliminate both interfacial and bulk defects [24,26,27].

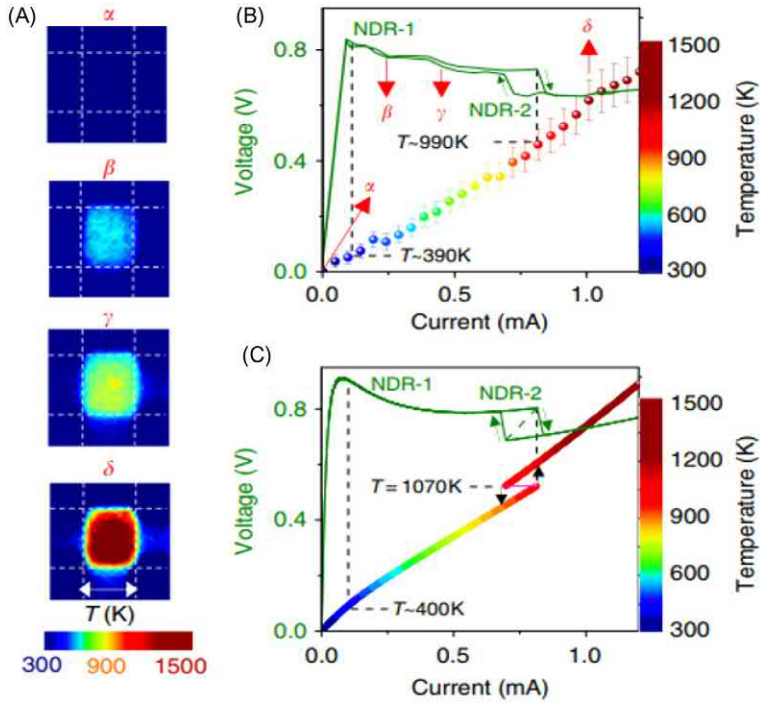


FIGURE 5.7 (A) Steady-state temperature maps as a function of applied current. (B and C) Experimental and simulated $I-V$ characteristics and the corresponding average temperature within the cross-point area (in A) at different current levels, respectively. *Adapted with permission from S. Kumar, Z. Wang, N. Davila, N. Kumari, K.J. Norris, X. Huang, et al., Physical origins of current and temperature controlled negative differential resistances in NbO_2, Nat. Commun. 8 (2017), no. 658.*

Liu et al. adopted a thin HfO_x layer between the bottom electrode and NbO_2 IMT layer, which can form a narrow conductive filament during the forming process. As a result, suppression of defects and thus reduction of the leakage current is achieved as shown in Fig. 5.9C. Similarly, Kang and Son introduced a thin TiO_2 dielectric film as a tunneling layer in between NbO_2 and Pt layer and observed leakage reduction [28]. Park et al. also inserted a NiO_y barrier layer in between both sides of the electrode and NbO_2 IMT layer ($W/NiO_y/NbO_2/NiO_y/W$) to form the perfect Schottky barrier height (Fig. 5.9B), which can reduce the I_{OFF} of NbO_2-based IMT device (Fig. 5.9A).

As a result of this device engineering, the NiO_y-inserted selector exhibits excellent characteristics, such as high I_{ON}/I_{OFF} ratio (>5400), fast transition speed (<2 ns), short delay time (<40 ns), and high operating temperature (>453K) as shown in Fig. 5.10. Also, the 1S1R characteristics of this selector were investigated by the same group using a HfO_x-based RRAM device

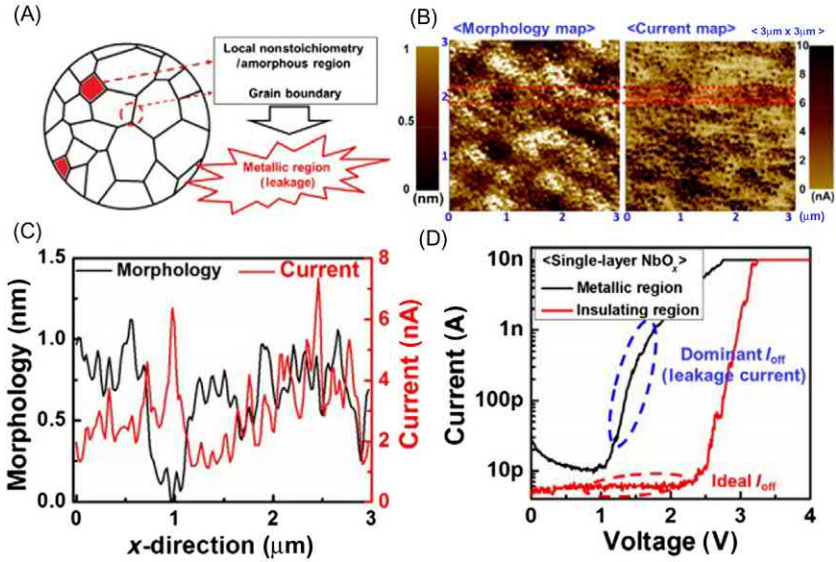


FIGURE 5.8 (A) Schematic diagram showing potential sources of leakage current in the NbO_2 system. (B) Surface morphology and current map of the deposited NbO_2 film. (C and D) Line and point profiles (red box in B) show that the morphologically shrink position which corresponding to defects that generate the leakage current. *Adapted with permission from J. Park, E. Cha, D. Lee, S. Lee, J. Song, J. Park, et al., Improved threshold switching characteristics of multi-layer NbO_x for 3-D selector application, Microelectronic Eng. 147 (2015) 318–320.*

(Fig. 5.10B–D). Thanks to the excellent characteristics of the NbO_2 selector, a significantly improved read-out margin (up to 2^9 word lines) can be achieved in a large crossbar memory array (Fig. 5.10D) [26].

5.3 Ovonic threshold switching

The threshold switching (TS) phenomenon observed in amorphous chalcogenide alloys as a response of external field is generally defined as OTS. This phenomenon was first observed by Ovshinsky in 1968 (Fig. 5.11) [29]. Since then, chalcogenides have been studied comprehensively for selector application by different research groups. The highly insulative state at low external bias makes OTS a suitable material for the selector to effectively suppress the leakage current in a crossbar array. Once the external electric field exceeds the threshold value of the material (threshold voltage (V_{th})), it exhibits instant, abrupt, and volatile transition to ON state and the resistance drops to an extremely low value (<1 kΩ) [30], capable of supplying enough current to successfully operate the adjacent memory device. There are many reports on OTS showing an initial soft breakdown of the OTS layer by applying relatively higher voltage during the first cycle, which is referred to

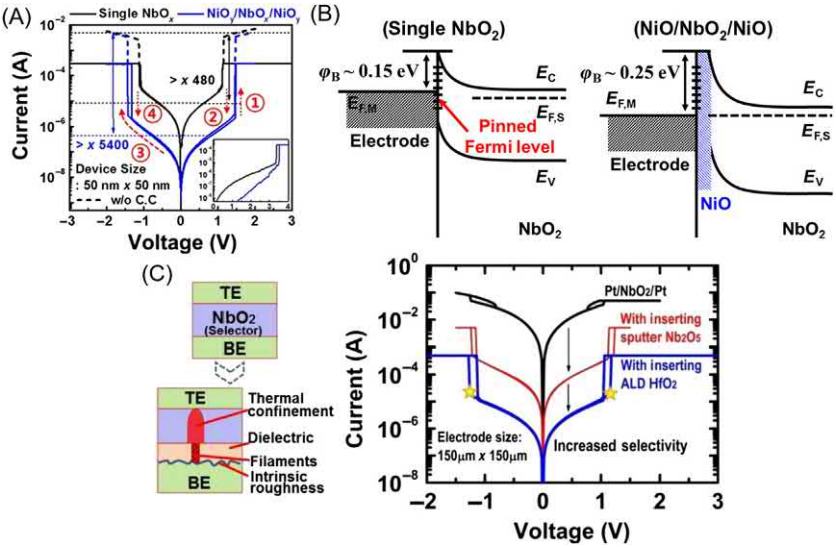


FIGURE 5.9 (A) Reduction of I_{OFF} of NbO_2 device by using a NiO_y barrier layer (B) schematic illustration explaining the effect of NiO_y layer. (C) The $I-V$ characteristics of the barrier inserted Pt/HfO_2/NbO_2/Pt device. *Adapted with permission from (A and B) J. Park, T. Hadamek, A.B. Posadas, E. Cha, A.A. Demkov and H. Hwang, Multi-layered $NiO_y/NbO_x/NiO_y$ fast drift-free threshold switch with high I_{on}/I_{off} ratio for selector application, Sci. Rep. 7 (2017) 4068; (C) X. Liu, S.K. Nandi, D.K. Venkatachalam, K. Belay, S. Song and R.G. Elliman, Reduced threshold current in NbO_2 selector by engineering device structure, IEEE Elec. Dev. Lett. 35 (10) (2014) 1055–1057.*

as a forming step [31,32]. However, there are other claims for the existence of forming less OTS devices [30]. Despite these claims, the elementary mechanism of forming and the reason why some OTS materials exhibit forming less behavior are still remain unclear.

OTS selectors are a potential candidate for selector application in cross-bar memory array owing to their favorable electrical switching mechanism, which satisfies most of the selector requirements, such as field-dependent and abrupt transition (switching slope <1 mV/dec) [30], ultrafast operating speed (delay time <8 ns and transition time <2 ns) [31], nondestructive and repeatable ($>8\times10^{12}$ cycles [33] and $>10^{11}$ cycles [32]), and thermal stability ($>450°C$) [34]. In addition, OTS devices show electrically stable ON and OFF states under an electrical stress test, making them a robust potential candidate for selector application. Under more than 10^3 s of either constant current (300 μA) [32] or constant voltage (1.2 V_{th}) [35] stress, the OTS devices retained their performances securely, as shown in Fig. 5.12.

One of the distinguishing factors of OTS among other types of selectors is that the switching is a primarily electronic process, suggesting that OTS has a

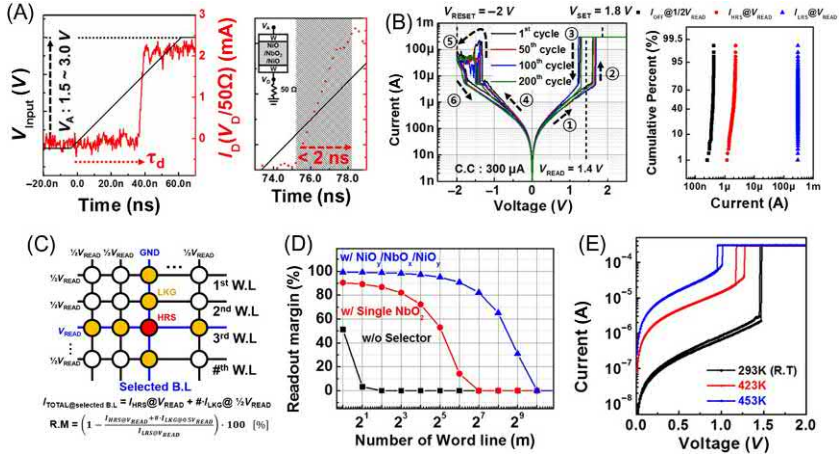


FIGURE 5.10 (A) The delay time (<40 ns) and transition speed (<2 ns) of W/NiO_y/NbO_2/NiO_y/W selector device. (B) DC $I-V$ characteristics of the 1S1R integrated structure. (C) Schematic illustration and (D) calculated read-out margin of cross-point array based on the 1S1R result. (E) The W/NiO_y/NbO_2/NiO_y/W maintain threshold characteristics at over 453K. *Adapted with permission from J. Park, T. Hadamek, A.B. Posadas, E. Cha, A.A. Demkov and H. Hwang, Multi-layered NiO_y/NbO_x/NiO_y fast drift-free threshold switch with high I_{on}/I_{off} ratio for selector application, Sci. Rep. 7 (2017) 4068.*

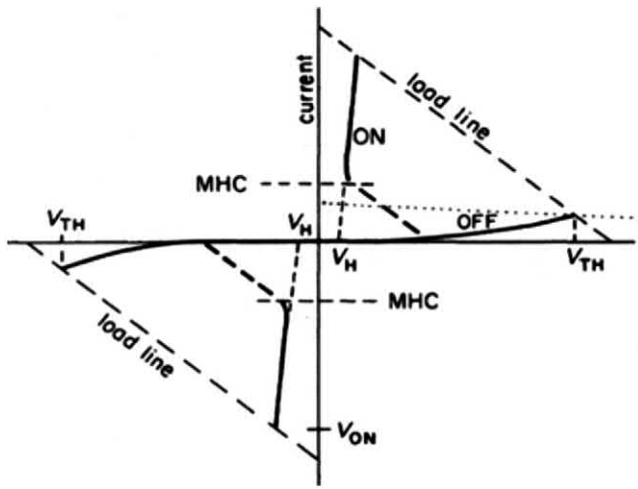


FIGURE 5.11 Typical current−voltage characteristics of OTS from Ovshinsky. *Adapted with permission from S.R. Ovshinsky, Reversible electrical switching phenomena in disordered structures, Phys. Rev. Lett. 21 (20) (1968) 1450−1453.*

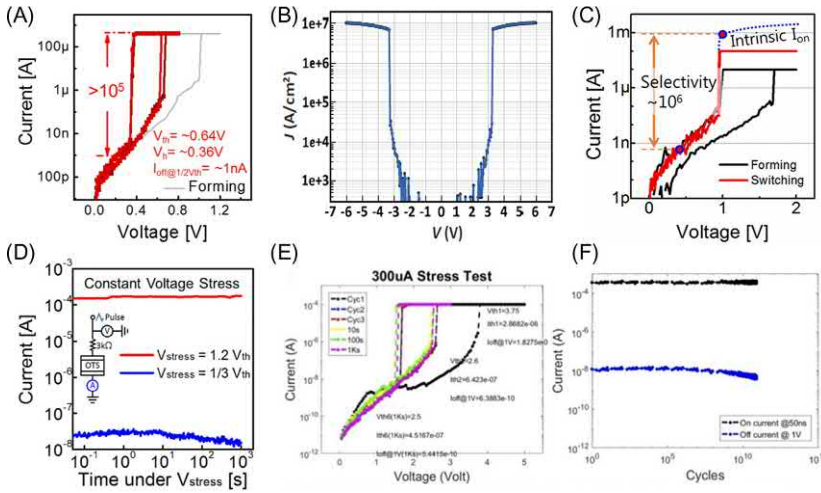


FIGURE 5.12 Reported performance of OTS selector devices, high selectivity, stable ON and OFF states under electrical stresses and ultra-high endurance. (A) A W/CTe/W, (B) BC-based OTS, (C and D) W/SiTe/W, and (E and F) TiN/TeAsGeSi/W OTS devices. Some materials require a higher voltage than the V_{th} to activate the OTS during the first sweep, a process called forming, as shown in (C) and (E). *Adapted with permission from (A–D) Y. Koo, K. Baek, and H. Hwang, Te-based amorphous binary OTS device with excellent selector characteristics for X-point memory applications, in: Symp. VLSI Tech. Dig., 2016, pp. T86–T87; S.A. Chekol, J. Yoo, J. Park, J. Song, C. Sung and H. Hwang, A C–Te-based binary OTS device exhibiting excellent performance and high thermal stability for selector application, Nanotechnology 29 (34) (2018); Y. Koo, S. Lee, S. Park, M. Yang, and H. Hwang, Simple binary ovonic threshold switching material SiTe and its excellent selector performance for high-density memory array application, IEEE Elec. Dev. Lett. 38 (5) (2017) 568–571; S. Yasuda, K. Ohba, T. Mizuguchi, H. Sei, M. Shimuta, K. Aratani, et al., A cross point Cu-ReRAM with a novel OTS selector for storage class memory applications, in: Symp. VLSI Tech. Dig., June 2017, pp. T30–T31, respectively; (E and F) H.Y. Cheng, W.C. Chien, I.T. Kuo, E.K. Lai1, Y. Zhu, J.L. Jordan-Sweet, et al., An ultra high endurance and thermally stable selector based on TeAsGeSiSe chalcogenides compatible with BEOL IC integration for cross-point PCM, in: Proc. IEEE Int. Electron Devices Meeting, December 2017, pp. 2.2.1–2.2.4.*

high potential for extremely fast switching operation [36]. Transition time less than 2 ns is already reported by different groups (Fig. 5.13) [30,31,34].

Besides the observed promising performance, the basic switching mechanism of OTS remains under debate. Following the original publication by Ovshinsky [29], detailed characteristics of threshold switches have been widely investigated. As a result, several theoretical models have been proposed to explain the switching mechanism, such as thermally induced instability by Warren [38], Shockley–Read–Hall recombination with impact ionization by Adler et al. (Fig. 5.14A) [39] and by Pirovano et al. (Fig. 5.14B) [40], polaron destabilization by Emin (Fig. 5.14D) [41], nucleation theory by Nardone (Fig. 5.14E) [42], and thermally assisted hopping

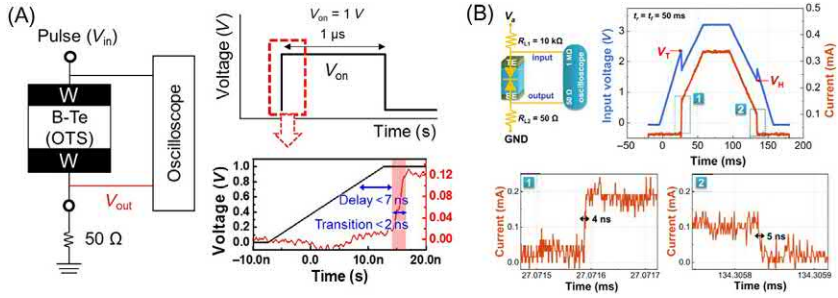


FIGURE 5.13 Superfast switching speed of B-Te based OTS (A) and AsSiTe based OTS (B) showing an abrupt transition from OFF-to-ON-state and fast falling from ON-to-OFF-state. *Adapted with permission from (A and B) J. Yoo, Y. Koo, S.A. Chekol, J. Park, J. Song, and H. Hwang, Te-based binary OTS selectors with excellent selectivity (>105), endurance (>108) and thermal stability (>450°C), in: Symp. on VLSI Tech. Dig., 2018; S. Kim, H. Kim, and S. Choi, Intrinsic threshold switching responses in AsTeSi thin film, J. Alloy. Compd. 667 (2016) 91–95, respectively.*

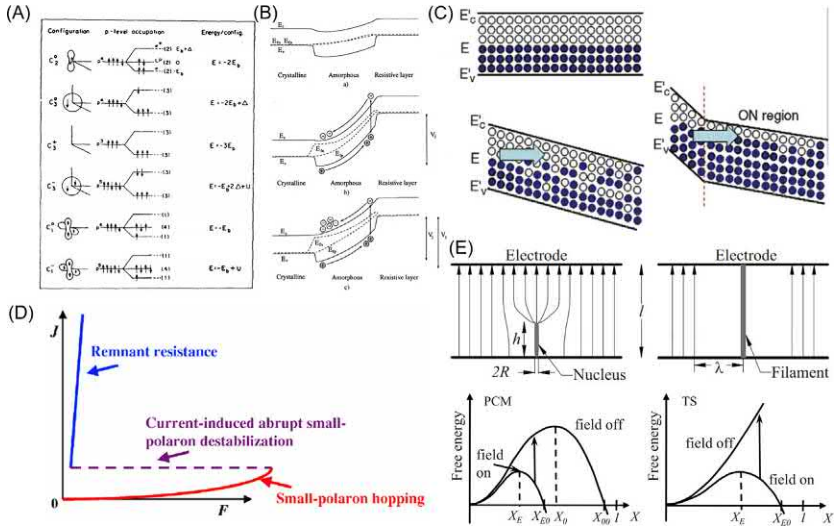


FIGURE 5.14 Some of the proposed switching mechanisms of OTS. (A and B) Shockley–Read–Hall recombination with impact ionization by Adler and Pirovano, (C) thermally assisted hopping model by Ielmini, (D) polaron destabilization by Emin, and (E) nucleation theory by Nardone. *Adapted with permission from (A–E) D. Adler, M.S. Shur, M. Silver, and S.R. Ovshinsky, Threshold switching in chalcogenide-glass thin films, J. Appl. Phys. 51 (6) (1980) 3289–3309; A. Pirovano, A.L. Lacaita, A. Benvenuti, F. Pellizzer, and R. Bez, Electronic switching in phase-change memories, IEEE Trans. Electron. Devices 51 (3) (2004) 452–459; D. Ielmini, Threshold switching mechanism by high-field energy gain in the hopping transport of chalcogenide glasses, Phys. Rev. B 78 (3) (2008) 035308; D. Emin, Current-driven threshold switching of a small polaron semiconductor to a metastable conductor, Phys. Rev. B 74 (3) (2006) 035206; M. Nardone, V. G. Karpov, D. C. S. Jackson and I. V. Karpov, A unified model of nucleation switching, Appl. Phys. Lett. 94 (10), 2009, 103509, respectively.*

model by Ielmini (Fig. 5.14C) [43]. Most of the models have already been proven to partially explain the OTS phenomenon. However, none of the proposed mechanisms can fully account for the observed characteristics and a unified model that represents the physical switching mechanism of OTS is still unavailable.

Realization of large crossbar memory arrays requires a selector device that has good electrical performance and thermal stability. Repeatable and stable switching, high uniformity, low OFF-current density, high ON-current density, high selectivity, and fast switching speed are some of the requirements that a selector device should meet. However, it is difficult to fulfill all the requirements at the same time in a simple material system. Especially, owing to the low glass transition temperature of chalcogen elements, chalcogenide-based selectors suffer from poor thermal stability, which is a crucial requirement to pass the back-end-of-line process. As a result, doping of different materials into the chalcogenide was proposed to improve certain electrical characteristics and thermal stability [32,37,44,45]. However, while doping can improve selector device characteristics, as the number of constituents increases and the material becomes more complex, it leads to difficulty in composition control and fabrication process.

In general, simple material systems are preferable to avoid such fabrication-related issues. Koo et al. introduced Te-based binary OTS materials, such as GeTe, ZnTe, and SiTe, that exhibit good performance and reasonable thermal stability [30], although further studies and improvements are required. There have been efforts to improve the performance of OTS, resulting in reports of numerous material combinations, giving more freedom of choice on material selection for OTS selector devices (Fig. 5.15). While having Te or Se as a core element, the other components can be chosen from various elements, such as Ge, Si, P, As, Sb, Bi, Zn, N, B, or C, which can enhance the tunability of device parameters, such as V_{th}, V_f, and V_h.

Although some of the reported selectors have shown good performance, their thermal stability is still not high enough, with tellurium-based binary OTS selectors, despite their outstanding electrical performance. Integration of selectors in crossbar array requires the thermal stability of at least 400°C to survive the back-end-of-line process. Several researchers tried various approaches to improve the performance and thermal stability of OTS devices (Fig. 5.16) [31,32,34,47] and recently reported binary telluride, which has thermal stability above 450°C using C and B as a constituting element [34]. According to the report, incorporation of small atomic size elements, such as C and B can effectively suppress the segregation of tellurides in the OTS film, which is the main reason for the thermal instability of tellurium-based OTS selectors. The suggested devices also showed excellent performances, such as high selectivity ($>10^5$), fast operation speed (<10 ns), low leakage current (~ 1 nA), and good cyclic endurance ($>10^8$).

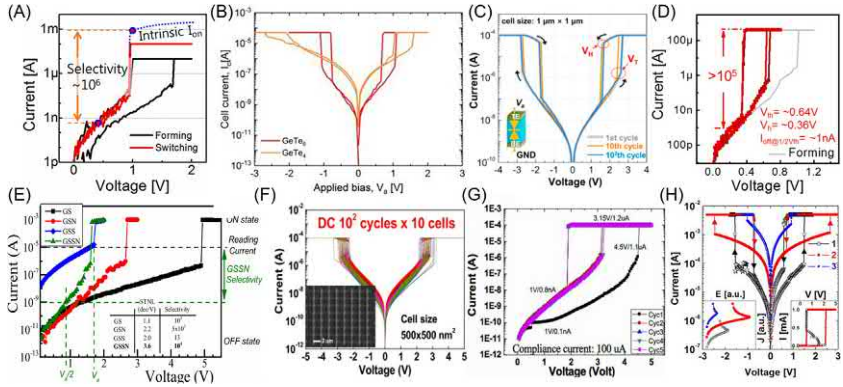


FIGURE 5.15 The vast choice of OTS material systems enhances tunability of different device properties. (A) W/SiTe/W, (B) W/GeTe/W, (C) Pt/AsTeSi/Pt, (D) W/CTe/W, (E) TiN/GeSeSbN/W, (F) Ti/AsTeGeSiN/Ti, (G) TiN/TeAsGeSiSe/W and (H) Pt/SiAsTe/Pt. *Adapted with permission from (A–H) A. Velea K. Opsomer, W. Devulder J. Dumortier, J. Fan, C. Detavernier, et al., Te-based chalcogenide materials for selector applications, Sci. Rep. 7 (2017), no. 8103; Y. Koo, K. Baek, and H. Hwang, Te-based amorphous binary OTS device with excellent selector characteristics for X-point memory applications, in: Symp. VLSI Tech. Dig., 2016, pp. T86–T87; S.A. Chekol, J. Yoo, J. Park, J. Song, C. Sung and H. Hwang, A C–Te-based binary OTS device exhibiting excellent performance and high thermal stability for selector application, Nanotechnology 29 (34) (2018); H.Y. Cheng, W.C. Chien, I.T. Kuo, E.K. Lai1, Y. Zhu, J. L. Jordan-Sweet, et al., An ultra high endurance and thermally stable selector based on TeAsGeSiSe chalcogenides compatible with BEOL IC integration for cross-point PCM, in: Proc. IEEE Int. Electron Devices Meeting, December 2017, pp. 2.2.1–2.2.4; S. Kim, H. Kim, and S. Choi, Intrinsic threshold switching responses in AsTeSi thin film, J. Alloy. Compd. 667 (2016) 91–95; M. Lee, D. Lee, H. Kim, H. Choi, J. Park, H. Kim, et al., Highly-scalable threshold switching select device based on chalcogenide glasses for 3D nanoscaled memory arrays, in: Proc. IEEE Int. Electron Devices Meeting, December 2012, pp. 2.6.1–2.6.3; A. Verdy, G. Navarro, V. Sousa, P. Noé, M. Bernard, F. Fillot, et al., Improved electrical performance thanks to Sb and N doping in Se-rich GeSe-based OTS selector devices, in: IEEE International Memory Workshop (IMW), May 2017; J. Lee, G. Kim, Y. Ahn, J. Park, S. Ryu, C. Hwang, et al., Threshold switching in Si-As-Te thin film for the selector device of crossbar resistive memory, Appl. Phys. Lett. 100 (2012) 123505, respectively [7,30–32,37,44–46].*

5.4 CBRAM-type selector

CBRAM-type TS devices are another class of emerging selector devices in which the threshold characteristic is obtained from the instability of the conductive filament in conventional CBRAM. The small and unstable filament can be formed when CBRAM is programmed under a relatively low compliance current during the first applied bias. This process of making a weak and permanent filament is called forming. As a result, spontaneous self-rupturing of the unstable filament can occur when the external bias is removed, and this programming and self-rupturing process can be used as a TS device. The operation mechanism of such TS device is quite similar to that of the conventional CBRAM-type memories

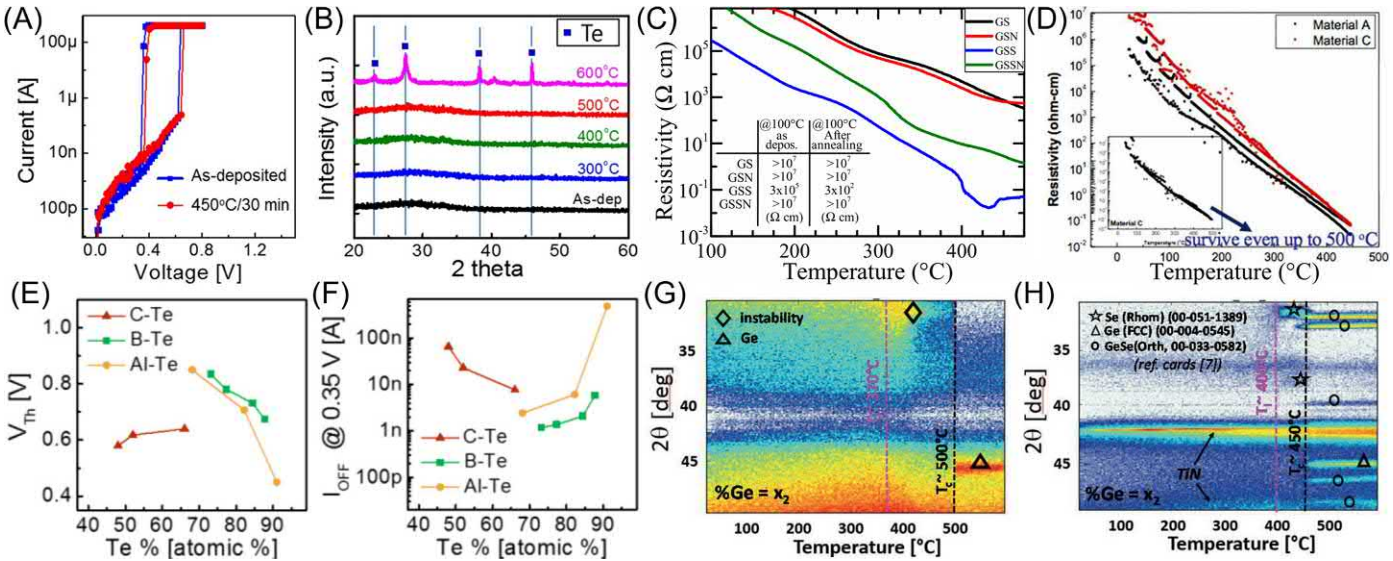


FIGURE 5.16 Thermal stability of OTS selector devices and studies to explain the device performance degradation at high temperature. (A and B) A CTe-based binary OTS, (C) Sb- and N-doped GeSe-based OTS device, (D) TeAsGeSiSe-based OTS, (E and F) CTe-, BTe-, and AlTe-based OTS, and (G and H) XRD thermal images taken from a GeSe film showing the beginning of the crystallization temperatures. *Adapted with permission from (A and B) S.A. Chekol, J. Yoo, J. Park, J. Song, C. Sung and H. Hwang, A C–Te-based binary OTS device exhibiting excellent performance and high thermal stability for selector application, Nanotechnology 29 (34) (2018); (C) A. Verdy, G. Navarro, V. Sousa, P. Noé, M. Bernard, F. Fillot, et al., Improved electrical performance thanks to Sb and N doping in Se-rich GeSe-based OTS selector devices, in: IEEE International Memory Workshop (IMW), May 2017; (D) H.Y. Cheng, W.C. Chien, I. T. Kuo, E.K. Lai1, Y. Zhu, J.L. Jordan-Sweet, et al., An ultra high endurance and thermally stable selector based on TeAsGeSiSe chalcogenides compatible with BEOL IC integration for cross-point PCM, in: Proc. IEEE Int. Electron Devices Meeting, December 2017, pp. 2.2.1–2.2.4; (E and F) J. Yoo, Y. Koo, S.A. Chekol, J. Park, J. Song, and H. Hwang, Te-based binary OTS selectors with excellent selectivity ($>10^5$), endurance ($>10^8$) and thermal stability ($>450°C$), in: Symp. on VLSI Tech. Dig., 2018; (G and H) B. Govoreanu, G.L. Donadio, K. Opsomer, W. Devulder, V.V. Afanas'ev, T. Witters, et al., Thermally stable integrated Se-based OTS selectors with >20 MA/cm^2 current drive, >3.103 half-bias nonlinearity, tunable threshold voltage and excellent endurance, in: Symp. VLSI Tech. Dig., June 2017, pp. T92–T93.*

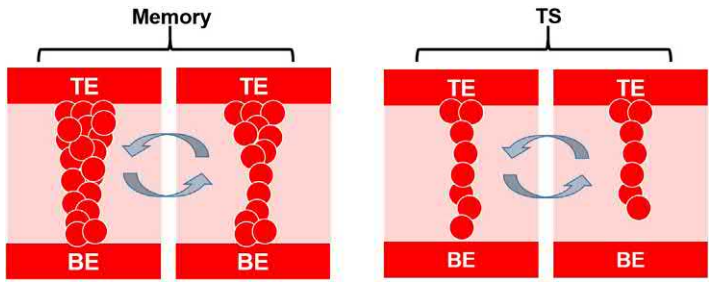


FIGURE 5.17 Schematic illustration of CBRAM-type memory (left) and threshold switching (TS) device (right). As depicted in the figure, the TS phenomenon takes place by the rupture of weak filament after the removal of applied bias.

(refer back to Chapter 2: Resistive switching memories). While the formed filament in CBRAM-type memory is strong and nonvolatile, the filament in TS devices is weak and unstable (Fig. 5.17).

Accordingly, several material combinations containing at least one active metal, such as Cu or Ag, have been reported as a selector device [48–61]. Song et al. reported an Ag-based ($Ag/TiO_2/Pt$) threshold switch as a selector device in a crossbar array [56]. Fig. 5.18A and B shows the cross-sectional TEM image and typical $I-V$ curve, respectively. The $Ag/TiO_2/Pt$ device exhibits bidirectional TS characteristics with high selectivity ($\sim 10^7$) and an abrupt transition (<5 mV/dec). However, when the compliance current becomes higher than 100 μA, the volatile behavior disappears and the device shows memory switching characteristics. Later, the Ag-based threshold device was reported by the same group using a-Si as an electrolyte (Ag/a-Si/Pt) [57]. Intrinsic defects, such as Si-dangling bonds (in this case), can impede the diffusion of atoms and slow down the filament dissolution speed. Hydrogen doping is introduced in [57] to passivate such defective regions, and thus a faster dissolution of the metal filament can be achieved by enhancing diffusivity of the Ag atoms. In addition, the removal of trap sites (leakage paths) can also lower the leakage current in the a-Si film of the device as shown in Fig. 5.18C and D.

Bricalli et al. reported an Ag/SiO_x-based selector device that exhibits a stable bidirectional switching [61]. The reported device shows volatile characteristics with a compliance current below 80 μA and a low leakage current of ~ 1 pA, as shown in Fig. 5.19A. However, a higher compliance current above 80 μA leads to nonvolatile switching due to the formation of a relatively stronger filament as shown in Fig. 5.19B. Feasibility and characterization of 1S1R are also investigated by serially connecting the $Ag/SiO_x/C$ threshold switch with a discrete $Ti/SiO_x/C$ RRAM device. As shown in the $I-V$ characteristics of the 1S1R device (Fig. 5.19C), the leakage current was significantly suppressed at the half-bias region.

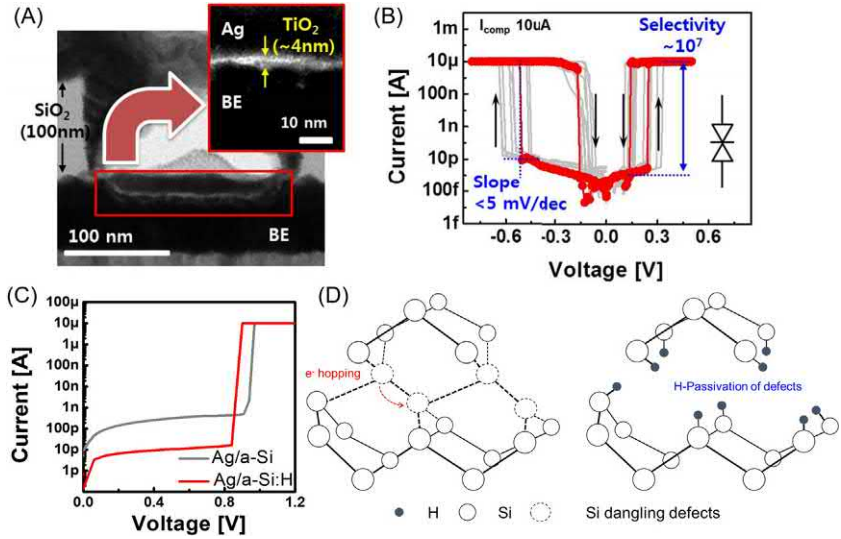


FIGURE 5.18 (A) Cross-sectional TEM image and (B) $I-V$ characteristics of Ag/TiO_2/Pt device, (C) typical $I-V$ characteristics of Ag/a-Si/Pt and Ag/a-Si:H/Pt selector devices, and (D) defect passivation through hydrogen doping. *Adapted with permission from (A and B) J. Song, J. Woo, A. Prakash, D. Lee, H. Hwang, Threshold selector with high selectivity and steep slope for cross-point memory array, IEEE Electron. Device Lett. 36 (7) (2015) 681–683; (C and D) S. Lim, J. Yoo, J. Song, J. Woo, J. Park, H. Hwang, Excellent threshold switching device (I_{off} ~ 1 pA) with atom-scale metal filament for steep slope (<5 mV/dec), ultra-low voltage (V_{dd} = 0.25 V) FET applications, in: 2016 IEEE Int. Electron. Devices Meet. (IEDM), 2016, pp. 34.7.1–37.7.4.*

Midya et al. reported an Ag/Hafnium oxide-based (Pd/Ag/HfO_x/Ag/Pd) selector device that can withstand an ON state current of 100 μA [59], as well as showing excellent performance such as high selectivity 10^{10}, steep turn-on slope <1 mV/dec, and high endurance beyond 10^8 cycles. Furthermore, OFF-to-ON and ON-to-OFF speeds of 75 and 250 ns, respectively, are obtained from the same device. They also demonstrated the vertical integration of Pd/Ag/HfO_x/Ag/Pd threshold switch on top of a Pd/Ta_2O_5/TaO_x/Pd RRAM device. A cross-sectional image of the 1S1R integrated device and typical $I-V$ characteristics of the selector, RRAM, and integrated (1S1R) devices are shown in Fig. 5.20. Successful suppression of leakage current at the half-bias region is achieved as a result of the selector device (Fig. 5.20D).

As shown above, CBRAM-type selector devices have shown excellent properties, such as ultra-low leakage current and abrupt transition. Since the formation and rupture of an Ag or Cu filament occur without causing electrical breakdown of the switching layer, the low leakage current can be maintained under cycling.

However, CBRAM-type selector devices lose volatile behavior and exhibit memory characteristics after electrical pulse above certain current

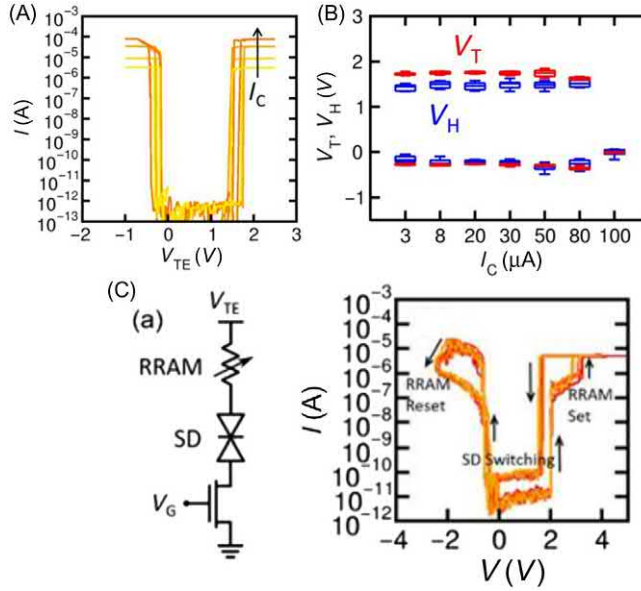


FIGURE 5.19 (A) $I-V$ characteristics of an Ag/SiO_x-based (Ag/SiO_x/C) selector device at different compliance current. (B) Data showing the change from TS to memory behavior above 80 μA. (C) Schematic diagram and $I-V$ curve of the 1S1R device consisting of an Ag/SiO_x/C selector and a Ti/SiO_x/C ReRAM. *Adapted with permission from A. Bricalli, E. Ambrosi, M. Laudato, M. Maestro, R. Rodriguez, and D. Ielmini, SiO_x-based resistive switching memory (RRAM) for crossbar storage/select elements with high on/off ratio, in: Proc. IEEE IEDM, December 2016, pp. 4.3.1–4.3.4.*

compliance, due to the formation of a relatively thick filament. The maximum ON current of Ag/TiO_2/Pt [56], Pd/Ag/HfO_x/Ag/Pd [59], and Ag/SiO_x/C [61] selector devices are 10, 100, and 80 μA, respectively, suggesting that further improvement in ON current is needed.

Furthermore, thermal stability is very important for selectors to pass the thermal budget during the back-end-of-line process. However, due to the excess diffusion of ions at high temperature, CBRAM-type selectors suffer from performance degradation at high temperature treatments [62]. To minimize such degradation, Song et al. adopted a TiN barrier that can control the in-diffusion of ions [63]. By optimizing the thickness of the TiN liner in AgTe/TiN/TiO_2/Pt structure, they showed a thermally stable and electrically reliable selector as shown in Fig. 5.21.

The switching speed of CBRAM-type selector devices is generally slower, compared with other electronic-based selectors, such as IMT and OTS devices, because of the involvement of ion motion [64]. Owing to the voltage-time relation in CBRAM-based devices, the turn-on (delay and transition) speed can be boosted by increasing applied voltage and as a result, a

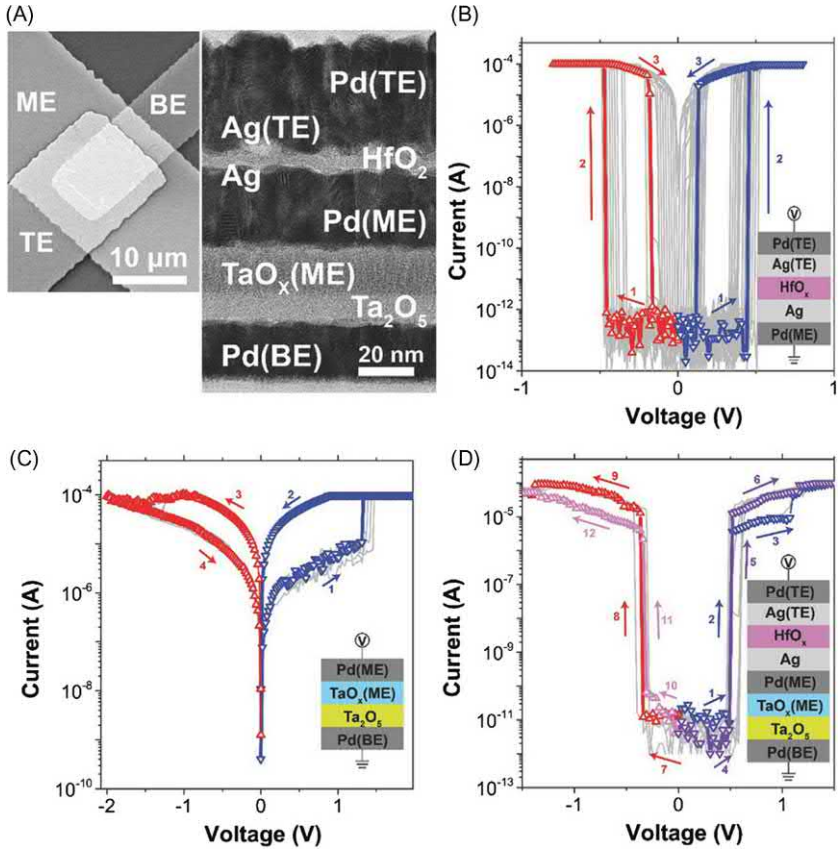


FIGURE 5.20 (A) Cross-sectional image of the fabricated 1S1R device. (B) *I–V* characteristics of the Pd/Ag/HfO_x/Ag/Pd selector (C) RRAM and (D) an integrated 1S1R device, respectively. *Adapted with permission from R. Midya, Z. Wang, J. Zhang, S.E. Savel'ev, C. Li, M. Rao, et al., Anatomy of Ag/hafnia-based selectors with 1010 nonlinearity, Adv. Mater. 29 (12) (2017) 1604457.*

delay time of less than 75 ns [59] and a transition speed of less than 10 ns [56] have already been reported. However, the turn-off (relaxation) speed is much slower than the turn-on speed because of the self-rupturing mechanism. The transition from ON-state to OFF-state takes place by breaking off the filament without any external bias.

Song et al. proposed a Te-doped Ag top electrode to accelerate the filament dissolution speed while maintaining the selector characteristics under a high ON-current condition [65]. The role of Te is to facilitate the extraction of Ag ions from the switching layer as Ag tends to form Ag–Te alloys, which are thermodynamically more favorable phases [66,67]. Therefore improved volatility characteristics can be achieved even at a higher operation current by using moderate Te content (35% Te) that can act as an additional

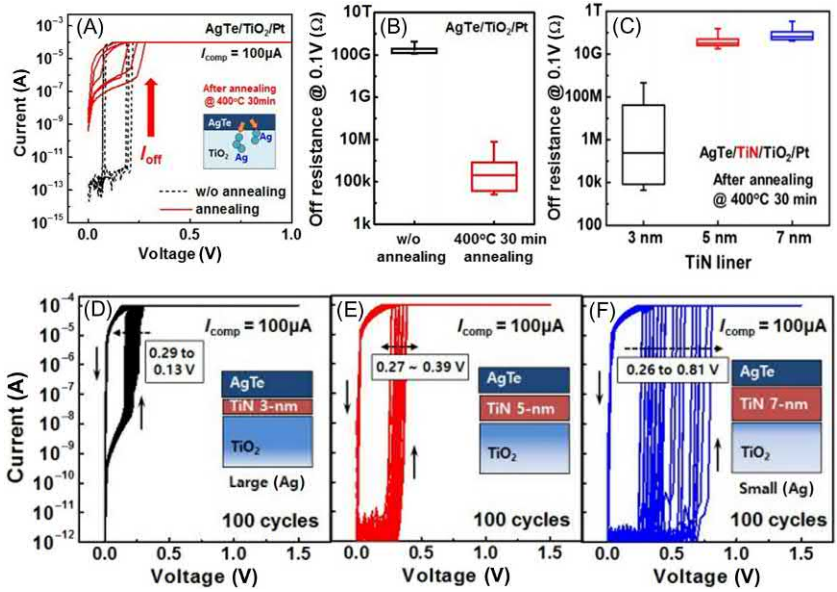


FIGURE 5.21 (A) Annealing effect on the $I-V$ characteristics of AgTe/TiO_2/Pt device without any liner. (B) OFF-state resistance distribution of as-fabricated and annealed AgTe/TiO_2/Pt TS device. (C) OFF-state resistance distribution after annealing with TiN liner (D–F) optimization of the TiN liner thickness. *Adapted with permission from J. Song, J. Woo, J. Yoo, S.A. Chekol, S. Lim, C. Sung, et al., Effects of liner thickness on the reliability of AgTe/TiO2-based threshold switching devices, IEEE Trans. Electron. Devices 64 (11) (2017) 4763–4767.*

driving force for fast filament dissolution (Fig. 5.22B). However, higher Te content (71%) results in significantly reduced ON current due to the higher extraction of Ag from the filament. Therefore a turn-off (relaxation) speed of 100 ns, which is 10 times faster than that of selector device with a pure Ag top electrode (Ag/TiO_2/Pt) is obtained.

As mentioned before, one of the concerns in CBRAM-type TS is the existence of a trade-off between the filament stability and ON-current. To have volatile characteristics, the device should have an unstable filament that can rupture back when the applied voltage is removed. However, as the compliance current increases, a strong and more stable filament will be formed and, as a result, the device loses its threshold characteristics. To overcome this problem, Zhao et al. used defective graphene (DG) layer between the active electrode and the switching layer (Ag/defective grain/SiO_2/Pt) to control the filament stability as shown in Fig. 5.23A [68]. As a result, they obtained a selector with a very high ON-current of 500 μA, large on/off ratio of $>10^8$, and fast switching speed of $<0.1/1$ μs (Fig. 5.23B and C). By controlling the size and concentration of the discrete DG, cation injecting path to the switching layer can be modulated and this leads to the formation of discrete tiny conductive filaments that can be easily self-ruptured. The

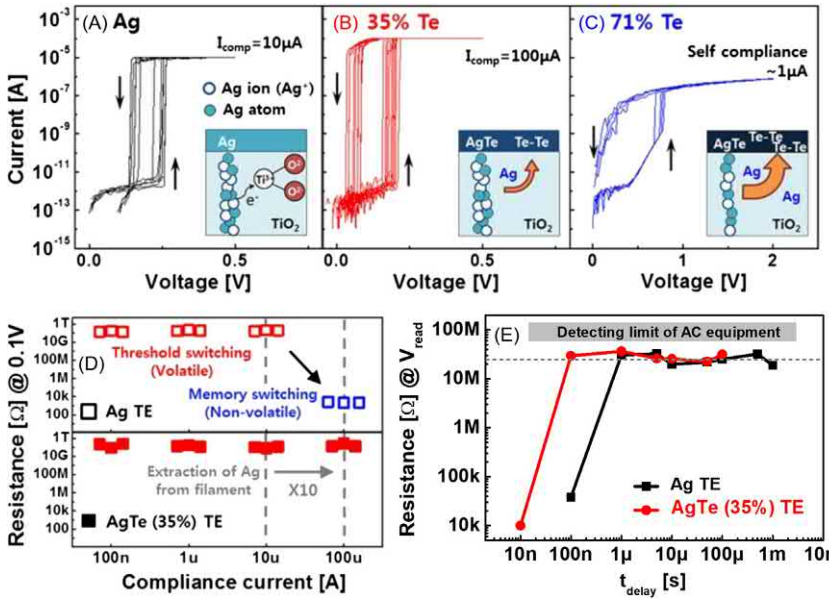


FIGURE 5.22 *I–V* characteristics of Ag-based TS device with (A) Ag, (B) AgTe (35%Te), and (C) AgTe (71%Te) top electrodes. (D) Read resistance following turn-on operation with increasing compliance current and (E) measured relaxation speed of selector devices with Ag and AgTe (35%) top electrode. *Adapted with permission from J. Song, J. Park, K. Moon, J. Woo, S. Lim, J. Yoo, et al., Monolithic integration of AgTe/TiO_2 based threshold switching device with TiN liner for steep slope field-effect transistors, in: Proc. IEEE IEDM, December 2016, pp. 25.3.1–25.3.4.*

functionality of the selector is also evaluated by serially connecting a Cu-based memory in an 1S1R configuration and successful suppression of leakage current at half-bias region is achieved (Fig. 5.23D).

To summarize, several types of material combinations are reported as a TS device and various attempts have been made to improve the performance of the observed selector characteristics, such as utilizing doped top electrode, introducing barrier layer, and controlling the filament size. However, certain characteristics still need further improvement to realize their functionality in a crossbar array, which can be provided through an in-depth study of the switching mechanism. The turn-on operation is the same as that of the conventional CBRAM device. The transition from OFF-to-ON occurs when a sufficient positive voltage is applied to the active electrode. However, the turn-off mechanism has not yet been established, even though various attempts have been made to explain it.

Hsiung et al. investigated the volatile characteristics of Ag/TiO_2/Pt-based TS and observed the rupture of the filament into a chain of Ag nanoparticles (Fig. 5.24A–C) [55]. This phenomenon is explained by using the Rayleigh

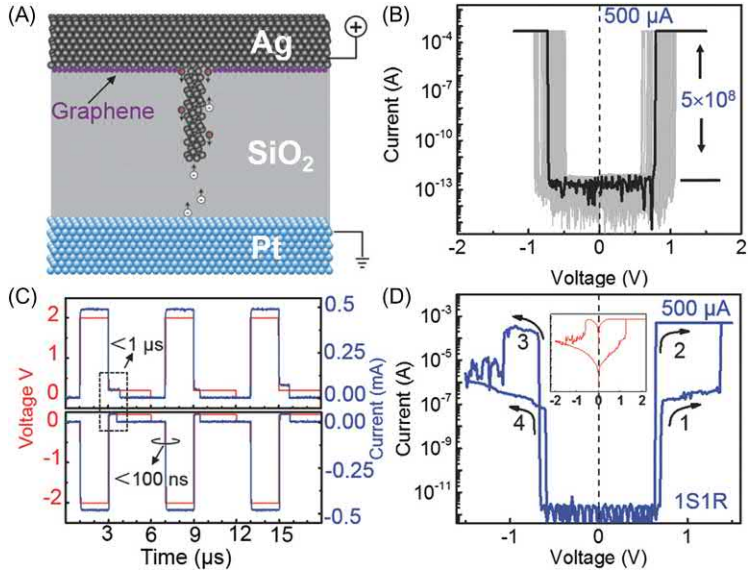


FIGURE 5.23 (A) Schematic representation of cation migration, (B) $I-V$ characteristics and (C) switching speed of an Ag/DG/SiO_2/Pt selector device, (D) 1S1R $I-V$ curve of serially connected Ag/DG/SiO_2/Pt selector device and Cu/HfO_2/Pt memory. *Adapted with permission from X. Zhao, J. Ma, X. Xiao, Q. Liu, L. Shao, D. Chen, et al., Breaking the current-retention dilemma in cation-based resistive switching devices utilizing graphene with controlled defects, Adv. Mater. (2018) 1705193.*

instability theorem, which states that large curvatures result in a high chemical potential of a surface atom. On the other hand, Ag particles can be formed spontaneously, when the bias is released after forming, to reduce the interfacial energy between the Ag and dielectric material, as observed by Wang et al. in Au/SiO_xN_y:Ag/Au device [54]. The filament dissolution through the diffusion process is shown in Fig. 5.24D.

Furthermore, various mechanisms, such as ion migration induced mechanical stress [49], steric repulsion [57], electromotive force [48], and tunnel barrier modulation [60], have been proposed. However, these mechanisms only qualitatively explain why the filaments are broken in each material and therefore quantitative analysis including concrete physical evidence and support is still required to figure out which parameters mainly determine the selector properties.

5.5 Conclusion

This chapter discussed the necessity of a selector device for crossbar arrays and presented a detailed review of threshold-type selector devices, which have a potential of tackling the inevitable problem of crossbar arrays. Crossbar arrays consisting of numerous memristive devices suffer from undesired data

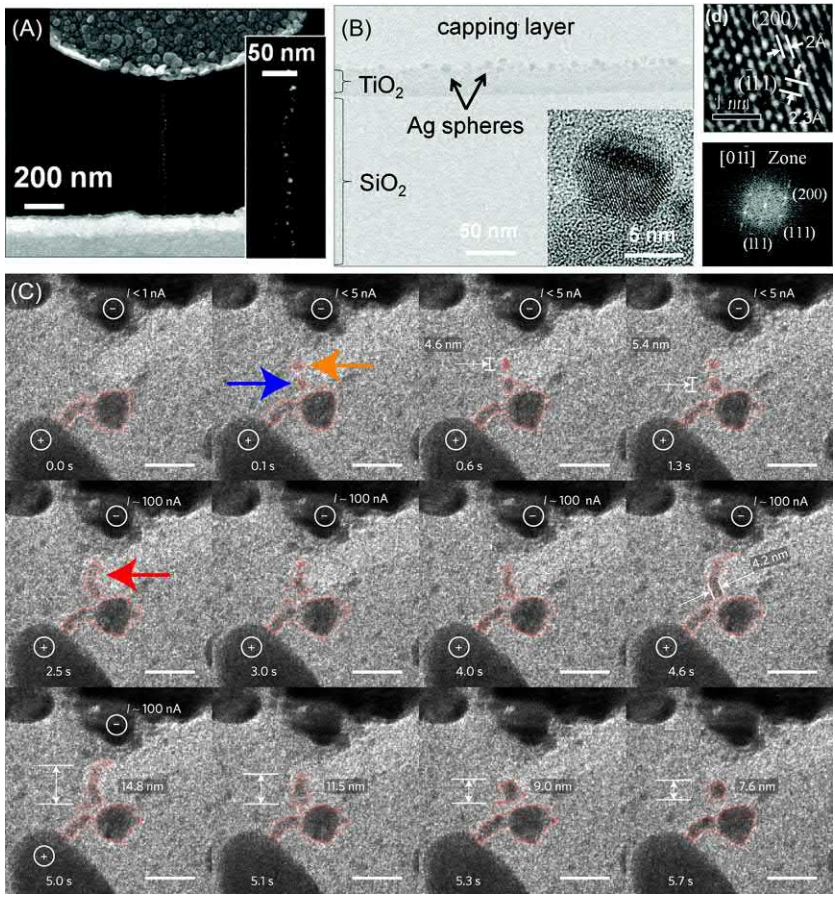


FIGURE 5.24 (A) SEM image of Ag/TiO_2/Pt structure after forming and (B) cross-sectional TEM image showing Ag nanospheres. (C) In situ TEM observation of interfacial energy-driven diffusion of Ag ions in Au/SiO_xN_y:Ag/Au system. *Adapted with permission from (A and B) C.-P. Hsiung, H.-W. Liao, J.-Y. Gan, T.-B. Wu, J.-C. Hwang, F. Chen, et al., Formation and instability of silver nanofilament in Ag-based programmable metallization cells, ACS Nano 4 (9) (2010) 5414–5420; (C) Z. Wang, S. Joshi, S.E. Saveliev, H. Jiang, R. Midya, P. Lin, et al., Memristors with diffusive dynamics as synaptic emulators for neuromorphic computing, Nat. Mater. 16 (1) (2017) 101–108.*

disturbance and power consumption due to the nonzero net voltage drop in the unselected cells. Therefore to minimize such undesirable problems without affecting the intrinsic operating characteristics of the memory, a selector device should be integrated to selectively access the target cell without affecting the rest. However, selectors should also meet necessary requirements in both electrical characteristics (low OFF current, high ON current density, fast switching speeds, infinite cycling endurance, large voltage margins, and

TABLE 5.1 Selector requirements comparison of IMT, CBRAM type, and OTS.

Device type	IMT	CBRAM-type	OTS
On-current density			
Off-current density			
Selectivity			
Bidirectional operation			
Process compatibility			
Switching speed			

compatible operating voltage conditions with memristive devices) and the fabrication process (low temperature fabrication processes, high thermal stability, and 3D stackable two-terminal device structure).

IMT-based selectors have been extensively researched due to their unique bidirectional, fast, and uniform switching characteristics. However, despite all the efforts, the high OFF current problem hinders the realization of selector application.

Recently, OTS selectors emerged as a potential candidate owing to their field-driven abrupt and fast switching characteristics. Especially, binary OTS selectors become more interesting due to their simple material system and easy fabrication. However, further enhancement of reliability related issues such as cell-to-cell or cycle-to-cycle uniformity and thermal stability must be reached to have a fully functional device for crossbar arrays.

There is no doubt that CBRAM-type TS devices have great potential for selector application. However, the slow turn-off speed and loss of selector characteristics at high operating current conditions are the drawbacks. Furthermore, because of the filamentary nature of the switching mechanism, the formation of filaments is randomized. As a result, cycle-to-cycle or cell-to-cell variation of threshold voltage, hold voltage, and other parameters can occur. Further research on the reliability, switching uniformity, and turn-off speed is needed.

Performance comparison of the three threshold-type selectors is shown in Table 5.1. Due to the limited selector characteristics, it is difficult to implement a high-density crossbar array device using the current selector devices. To solve the bottleneck of von Neumann computing for pattern recognition applications, neuromorphic computing based on deep neural network was proposed. To implement hardware neuromorphic computing, we need to develop high-density analog synapse array devices for matrix multiplication. To maximize the pattern recognition accuracy of the hardware neural network, we need to minimize disturbance and the sneak path current of synapse array devices by adopting an ideal selector device. Developing an ideal selector with an insight on the switching mechanism is essential technology for both high-density crossbar memory applications and synapse array for hardware neural network applications.

References

[1] R. Waser, R. Dittmann, G. Staikov, K. Szot, Redox-based resistive switching memories—nanoionic mechanisms, prospects, and challenges, Adv. Mater. 21 (2009) 2632–2663.

[2] D. Kau, S. Tephen Tang, I.V. Karpov, R. Dodge, B. Klehn, J.A. Kalb, et al., A stackable cross point phase change memory, in: Proc. IEEE Int. Electron Devices Meeting, December 2009, pp. 27.1.1–27.1.4.

[3] K. Kim, Y.J. Song, Current and Future High Density FRAM Technology, Taylor and Francis, 2004, pp. 1058–4587.

[4] M. Hosomi, H. Yamagishi, T. Yamamoto, K. Bessho, Y. Higo, K. Yamane, et al. A novel nonvolatile memory with spin torque transfer magnetization switching: spin-ram, in: IEEE International Electron Devices Meeting, 2005. IEDM Technical Digest., Washington, DC, 2005, pp. 459–462.
[5] E.P.G. Wright, Electric connecting device, U.S. patent 2 667 542, September 25, 1954.
[6] J. Choe, Intel 3D Xpoint memory die removed from Intel Optane, Tech. Insights, 18 May 2017.
[7] A. Velea, K. Opsomer, W. Devulder, J. Dumortier, J. Fan, C. Detavernier, et al., Te-based chalcogenide materials for selector applications, Sci. Rep. 7 (2017). no. 8103.
[8] G.W. Burr, R.S. Shenoy, K. Virwani, P. Narayanan, A. Padilla, B. Kurdi, et al., Access devices for 3D crosspoint memory, J. Vac. Sci. Technol. 32 (4) (2014). Art. no. 040802.
[9] J. Wei, Z. Wang, W. Chen, D.H. Cobden, New aspects of the metal-insulator transition in single-domain vanadium dioxide nanobeams, Nat. Nanotechnol. 4 (2009) 420–424.
[10] E. Cha, J. Woo, D. Lee, et al., Nanoscale (<10 nm) 3D vertical ReRAM and NbO_2 threshold selector with TiN electrode, in: Proc. IEEE Int. Electron Devices Meeting, December 2013, pp. 10.5.1–10.5.4.
[11] F.J. Morin, Oxides which show a metal-to-insulator transition at the neel temperature, Phys. Rev. Lett. 3 (1) (1959) 34.
[12] A. Cavalleri, M. Rini, H.H.W. Chong, S. Fourmaux, T.E. Glover, P.A. Heimann, et al., Band-selective measurements of electron dynamics in VO_2 using femtosecond near-edge X-ray absorption, Phys. Rev. Lett. 95 (6) (2005). no. 067405.
[13] M. Son, J. Lee, J. Park, J. Shin, G. Chio, S. Jung, et al., Excellent selector characteristics of nanoscale VO_2 for high-density bipolar ReRAM applications, Electron. Device Lett. 32 (11) (2011) 1579–1581.
[14] E. Cha, J. Woo, D. Lee, S. Lee, J. Song, Y. Koo, et al., Nanoscale (<10 nm) 3D vertical ReRAM and NbO_2 threshold selector with TiN electrode, in: 2013 IEEE International Electron Devices Meeting, Washington, DC, 2013, pp. 10.5.1–10.5.4.
[15] W.A. Vitale, E.A. Casu, A. Biswas, T. Rosca, C. Alper, A. Krammer, et al., A steep-slope transistor combining phase-change and band-to-band-tunneling to achieve a sub-unity body factor, Sci. Rep. 7 (355) (2017).
[16] X. Liu, S.M. Sadaf, M. Son, J. Shin, J. Park, J. Lee, et al., Diode-less bilayer oxide (WO_x-NbO_x) device for cross-point resistive memory applications, Nanotechnology 22 (47) (2011). no. 175702.
[17] S. Kim, X. Liu, J. Park, S. Jung, W. Lee, J. Woo, et al., Ultrathin (<10 nm) Nb2O5/NbO2 hybrid memory with both memory and selector characteristics for high density 3D vertically stackable RRAM applications, in: Symposium VLSI Technol., 2012, p. T18.3.
[18] E. Cha, J. Park, J. Woo, D. Lee, A. Prakash, H. Hwang, Comprehensive scaling study of NbO_2 insulator-metal-transition selector for cross point array application, Appl. Phys. Lett. 108 (15) (2016). no. 153502.
[19] S.K. Nandi, X. Liu, D.K. Venkatachalam, R.G. Elliman, Threshold current reduction for the metal–insulator transition in NbO_{2-x} selector devices: the effect of ReRAM integration, J. Phys. D: Appl. Phys. 48 (19) (2015).
[20] C. Funck, S. Menzel, N. Aslam, H. Zhang, A. Hardtdegen, R. Waser, et al., Multidimensional simulation of threshold switching in NbO_2 based on an electric field triggered thermal runaway model, Adv. Elec. Mater 2 (7) (2016). no. 201600169.
[21] S. Slesazeck, H. Mähne, H. Wylezich, A. Wachowiak, J. Radhakrishnan, A. Ascoli, et al., Physical model of threshold switching in NbO_2 based memristors, RSC Adv. 5 (124) (2015) 102318–102322.

[22] G.A. Gibson, S. Musunuru, J. Zhang, K. Vandenberghe, J. Lee, C.C. Hsieh, et al., An accurate locally active memristor model for S-type negative differential resistance in NbO_x, Appl. Phys. Lett. 108 (2) (2016). no. 023505.

[23] S. Kumar, Z. Wang, N. Davila, N. Kumari, K.J. Norris, X. Huang, et al., Physical origins of current and temperature controlled negative differential resistances in NbO_2, Nat. Commun. 8 (2017). no. 658.

[24] J. Park, E. Cha, D. Lee, S. Lee, J. Song, J. Park, et al., Improved threshold switching characteristics of multi-layer NbO_x for 3-D selector application, Microelectronic Eng. 147 (2015) 318–320.

[25] G. Bersukera, J. Yum, L. Vandelli, A. Padovani, L. Larcher, V. Iglesias, et al., Grain boundary-driven leakage path formation in HfO_2 dielectrics, Solid. State Electron. 65–66 (2011) 146–150.

[26] J. Park, T. Hadamek, A.B. Posadas, E. Cha, A.A. Demkov, H. Hwang, Multi-layered $NiO_y/NbO_x/NiO_y$ fast drift-free threshold switch with high Ion/Ioff ratio for selector application, Sci. Rep. 7 (4068) (2017).

[27] X. Liu, S.K. Nandi, D.K. Venkatachalam, K. Belay, S. Song, R.G. Elliman, Reduced threshold current in NbO_2 selector by engineering device structure, IEEE Elec. Dev. Lett. 35 (10) (2014) 1055–1057.

[28] M. Kang, J. Son, Off-state current reduction in NbO_2-based selector device by using TiO_2 tunneling barrier as an oxygen scavenger, Appl. Phys. Lett. 109 (2016) 202101.

[29] S.R. Ovshinsky, Reversible electrical switching phenomena in disordered structures, Phys. Rev. Lett. 21 (20) (1968) 1450–1453.

[30] Y. Koo, K. Baek, H. Hwang, Te-based amorphous binary OTS device with excellent selector characteristics for X-point memory applications, Symp. VLSI Tech. Dig. (2016) T86–T87.

[31] S.A. Chekol, J. Yoo, J. Park, J. Song, C. Sung, H. Hwang, A C–Te-based binary OTS device exhibiting excellent performance and high thermal stability for selector application, Nanotechnology 29 (34) (2018).

[32] H.Y. Cheng, W.C. Chien, I.T. Kuo, E.K. Lai1, Y. Zhu, J.L. Jordan-Sweet, et al., An ultra high endurance and thermally stable selector based on TeAsGeSiSe chalcogenides compatible with BEOL IC integration for cross-point PCM, in: Proc. IEEE Int. Electron Devices Meeting, December 2017, pp. 2.2.1–2.2.4.

[33] S.R. Ovshinsky, An introduction to Ovonic research, J. Non Crystalline Solid. 2 (1970) 99–106.

[34] J. Yoo, Y. Koo, S.A. Chekol, J. Park, J. Song, H. Hwang, Te-based binary OTS selectors with excellent selectivity (>105), endurance (>108) and thermal stability ($>450°C$), in: Symp. on VLSI Tech. Dig., 2018.

[35] Y. Koo, S. Lee, S. Park, M. Yang, H. Hwang, Simple binary ovonic threshold switching material SiTe and its excellent selector performance for high-density memory array application, IEEE Elec. Dev. Lett. 38 (5) (2017) 568–571.

[36] S. Yasuda, K. Ohba, T. Mizuguchi, H. Sei, M. Shimuta, K. Aratani, et al., A cross point Cu-ReRAM with a novel OTS selector for storage class memory applications, in: Symp. VLSI Tech. Dig., June 2017, pp. T30–T31.

[37] S. Kim, H. Kim, S. Choi, Intrinsic threshold switching responses in AsTeSi thin film, J. Alloy. Compd. 667 (2016) 91–95.

[38] A.C. Warren, Reversible thermal breakdown as a switching mechanism in chalcogenide glasses, IEEE Trans. Electron. Devices 20 (2) (1973) 123–131.

[39] D. Adler, M.S. Shur, M. Silver, S.R. Ovshinsky, Threshold switching in chalcogenide-glass thin films, J. Appl. Phys. 51 (6) (Jun. 1980) 3289–3309.

[40] A. Pirovano, A.L. Lacaita, A. Benvenuti, F. Pellizzer, R. Bez, Electronic switching in phase-change memories, IEEE Trans. Electron. Devices 51 (3) (2004) 452–459.
[41] D. Emin, Current-driven threshold switching of a small polaron semiconductor to a metastable conductor, Phys. Rev. B 74 (3) (2006) 035206.
[42] M. Nardone, V.G. Karpov, D.C.S. Jackson, I.V. Karpov, A unified model of nucleation switching, Appl. Phys. Lett. 94 (10) (2009) 103509.
[43] D. Ielmini, Threshold switching mechanism by high-field energy gain in the hopping transport of chalcogenide glasses, Phys. Rev. B 78 (3) (2008) 035308.
[44] M. Lee, D. Lee, H. Kim, H. Choi, J. Park, H. Kim, et al., Highly-scalable threshold switching select device based on chalcogenide glasses for 3D nanoscaled memory arrays, in: Proc. IEEE Int. Electron Devices Meeting, December 2012, pp. 2.6.1–2.6.3.
[45] A. Verdy, G. Navarro, V. Sousa, P. Noé, M. Bernard, F. Fillot, et al., Improved electrical performance thanks to Sb and N doping in Se-rich GeSe-based OTS selector devices, in: IEEE International Memory Workshop (IMW), May 2017.
[46] J. Lee, G. Kim, Y. Ahn, J. Park, S. Ryu, C. Hwang, et al., Threshold switching in Si-As-Te thin film for the selector device of crossbar resistive memory, Appl. Phys. Lett. 100 (2012) 123505.
[47] B. Govoreanu, G.L. Donadio, K. Opsomer, W. Devulder, V.V. Afanas'ev, T. Witters, et al., Thermally stable integrated Se-based OTS selectors with >20 MA/cm^2 current drive, >3.103 half-bias nonlinearity, tunable threshold voltage and excellent endurance, in: Symp. VLSI Tech. Dig., June 2017, pp. T92–T93.
[48] J.V.D. Hurk, E. Linn, H. Zhang, R. Waser, I. Valov, Volatile resistance states in electrochemical metallization cells enabling non-destructive readout of complementary resistive switches, Nanotechnology 25 (42) (2014) 425202.
[49] S. Ambrogio, S. Balatti, S. Choi, D. Ielmini, Impact of the mechanical stress on switching characteristics of electrochemical resistive memory, Adv. Mater. 26 (23) (2014) 3885–3892.
[50] J. Song, A. Prakash, D. Lee, J. Woo, E. Cha, S. Lee, et al., Bidirectional threshold switching in engineered multilayer (Cu_2O/Ag:Cu_2O/Cu_2O) stack for cross-point selector application, Appl. Phys. Lett. 107 (11) (Sep. 2015) 113504.
[51] Q. Luo, X. Xu, H. Liu, H. Lv, T. Gong, S. Long, et al., Cu BEOL compatible selector with high selectivity (>107), extremely low off-current (~pA) and high endurance (>1010), in: Proc. IEEE IEDM, December 2015, pp. 10.4.1–10.4.4.
[52] W. Chen, H.J. Barnaby, M.N. Kozicki, Volatile and non-volatile switching in Cu-SiO_2 programmable metallization cells, IEEE Electron. Device Lett. 37 (5) (2016) 580–583.
[53] J. Woo, D. Lee, E. Cha, S. Lee, S. Park, H. Hwang, Control of Cu conductive filament in complementary atom switch for cross-point selector device application, IEEE Electron. Device Lett. 35 (1) (2014) 60–62.
[54] Z. Wang, S. Joshi, S.E. Saveliev, H. Jiang, R. Midya, P. Lin, et al., Memristors with diffusive dynamics as synaptic emulators for neuromorphic computing, Nat. Mater. 16 (1) (2017) 101–108.
[55] C.-P. Hsiung, H.-W. Liao, J.-Y. Gan, T.-B. Wu, J.-C. Hwang, F. Chen, et al., Formation and instability of silver nanofilament in Ag-based programmable metallization cells, ACS Nano 4 (9) (2010) 5414–5420.
[56] J. Song, J. Woo, A. Prakash, D. Lee, H. Hwang, Threshold selector with high selectivity and steep slope for cross-point memory array, IEEE Electron. Device Lett. 36 (7) (2015) 681–683.
[57] S. Lim, J. Yoo, J. Song, J. Woo, J. Park, H. Hwang, Excellent threshold switching device (I_{off}~1 pA) with atom-scale metal filament for steep slope (<5 mV/dec), ultra-low

voltage ($V_{dd} = 0.25$ V) FET applications, in: 2016 IEEE Int. Electron. Devices Meet. (IEDM), 2016, pp. 34.7.1−37.7.4.

[58] N. Shukla, B. Grisafe, R.K. Ghosh, N. Jao, A. Aziz, J. Frougier, et al., Ag/HfO_2 based threshold switch with extreme non-linearity for unipolar cross-point memory and steep-slope Phase-FETs, in: Proc. IEEE IEDM, December 2016, pp. 34.6.1−34.6.4.

[59] R. Midya, Z. Wang, J. Zhang, S.E. Savel'ev, C. Li, M. Rao, et al., Anatomy of Ag/hafnia-based selectors with 10^{10} nonlinearity, Adv. Mater. 29 (12) (2017) 1604457.

[60] H. Sun, Q. Liu, C. Li, S. Long, H. Lv, C. Bi, et al., Direct observation of conversion between threshold switching and memory switching induced by conductive filament morphology, Adv. Funct. Mater. 24 (36) (2014) 5679−5686.

[61] A. Bricalli, E. Ambrosi, M. Laudato, M. Maestro, R. Rodriguez, D. Ielmini, SiO_x-based resistive switching memory (RRAM) for crossbar storage/select elements with high on/off ratio, in: Proc. IEEE IEDM, December 2016, pp. 4.3.1−4.3.4.

[62] S.K. Bhagat, N.D. Theodore, T.L. Alford, Thermal stability of tungsten−titanium diffusion barriers for silver metallization, Thin Solid Films 516 (2008) 7451−7457.

[63] J. Song, J. Woo, J. Yoo, S.A. Chekol, S. Lim, C. Sung, et al., Effects of liner thickness on the reliability of AgTe/TiO_2-based threshold switching devices, IEEE Trans. Electron. Devices 64 (11) (2017) 4763−4767.

[64] I. Valov, R. Waser, J.R. Jameson, M.N. Kozicki, Electrochemical metallization memories—fundamentals, applications, prospects, Nanotechnology 22 (25) (2011) 254003.

[65] J. Song, J. Park, K. Moon, J. Woo, S. Lim, J. Yoo, et al., Monolithic integration of AgTe/TiO_2 based threshold switching device with TiN liner for steep slope field-effect transistors, in: Proc. IEEE IEDM, December 2016, pp. 25.3.1−25.3.4.

[66] W. Devulder, K. Opsomer, J. Meersschaut, D. Deduytsche, M. Jurczak, L. Goux, et al., Combinatorial study of Ag−Te thin films and their application as cation supply layer in CBRAM cells, ACS Combinatorial Sci. 17 (5) (2015) 334−430.

[67] L. Goux, K. Opsomer, R. Degraeve, R. Müller, C. Detavernier, D.J. Wouters, et al., Influence of the Cu-Te composition and microstructure on the resistive switching of Cu-Te/Al_2O_3/Si cells, Appl. Phys. Lett. 99 (5) (2011) 053502.

[68] X. Zhao, J. Ma, X. Xiao, Q. Liu, L. Shao, D. Chen, et al., Breaking the current-retention dilemma in cation-based resistive switching devices utilizing graphene with controlled defects, Adv. Mater. (2018) 1705193.

Part II

Computational memory

Chapter 6

Memristive devices as computational memory

Abu Sebastian[1], Damien Querlioz[2], Bipin Rajendran[3] and Sabina Spiga[4]

[1]IBM Research – Zurich, Rüschlikon, Switzerland, [2]Centre for Nanoscience and Nanotechnology, Universite Paris-Saclay, Palaiseau, France, [3]Department of Engineering, King's College London, London, England, [4]CNR-IMM, Agrate Brianza, Italy

6.1 Introduction

Today's computing systems are based on the von Neumann architecture that dates back to the 1940s. Memory and processing units are physically separated and large amounts of data need to be shuttled back and forth between them during the execution of various computational tasks. The latency and energy associated with accessing data from the memory units is a key performance bottleneck for a range of applications, in particular for the increasingly prominent artificial intelligence-related workloads [1]. Even at the relatively old 45 nm complementary metal oxide semiconductor (CMOS) node, the cost of an on-chip SRAM access is over 2 orders of magnitude higher than that of multiplying two 8-bit numbers [2]. The energy cost associated with moving data is clearly a key challenge for severely energy-constrained mobile and edge computing. However, it is also a challenge for high-performance computing in a cloud environment given that the computing systems are severely power limited due to cooling constraints. The current approaches such as using hundreds of processors in parallel [3] or application-specific processors [4] are not likely to fully overcome the challenge of data movement. It is getting increasingly clear that novel architectures need to be explored where memory and processing are better collocated.

6.2 In-memory computing

In-memory computing is one such non-von Neumann approach where certain computational tasks are performed in place in the memory itself organized as

Memristive Devices for Brain-Inspired Computing. DOI: https://doi.org/10.1016/B978-0-08-102782-0.00006-X

a computational memory unit [5–7]. As schematically illustrated in Fig. 6.1, in-memory computing obviates the need to move data into a processing unit. Computing is performed by exploiting the physical attributes of the memory devices, their array-level organization, the peripheral circuitry, and the control logic. In this paradigm the memory is not just a place to store information but is an active participant in the computational task. Besides reducing latency and energy cost associated with data movement, in-memory computing also has the potential to improve the computational time complexity associated with certain computational tasks due to the massive parallelism afforded by a dense array of millions of nanoscale memory devices performing analog computation. By introducing physical coupling between the memory devices there is also a potential for further reduction in computational time complexity [8,9].

Naturally, memory devices are central to in-memory computing. Traditionally, information is stored in terms of the presence or absence of charge such as in dynamic random access memory (DRAM), static random access memory (SRAM), and Flash memory [10]. However, there is also an emerging class of memory devices where information is stored in terms of differences in the atomic arrangements, orientation of ferromagnetic, or ferroelectric material layers. In these devices such differences manifest as a change of resistance and the devices are thus termed resistive memory devices [11]. They are also often referred to as memristive devices due to their relation to the memristor concept proposed by Leon Chua [12]. Memristive devices such as redox-based resistive random access memory (ReRAM), phase-change memory (PCM), and magnetoresistive random access memory (MRAM) reviewed extensively in Part I of the book, are particularly well suited for in-memory computing.

It is essential to understand the key physical attributes that enable in-memory computing using memristive devices. First of all, the ability to store

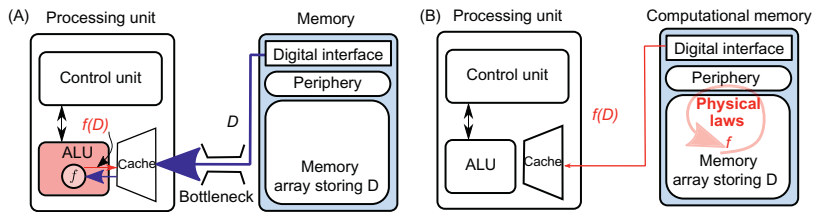


FIGURE 6.1 (A) In a conventional computing system, when an operation f is performed on data D, D has to be moved into a processing unit. This incurs significant latency and energy cost and creates the well-known von Neumann bottleneck. (B) With in-memory computing, $f(D)$ is performed within a computational memory unit by exploiting the physical attributes of the memory devices. This obviates the need to move D to the processing unit. *Adapted from A. Sebastian, T. Tuma, N. Papandreou, M. Le Gallo, L. Kull, T. Parnell, et al., Temporal correlation detection using computational phase-change memory, Nat. Commun. 8 (2017) 1115.*

two levels of resistance/conductance values in a nonvolatile manner and to reversibly switch from one level to the other (binary storage capability) can be exploited for computing. Fig. 6.2A shows the resistance values achieved upon repeated switching of a representative PCM device between low-resistance SET states and high-resistance RESET states. Due to the SET and RESET states, resistance could serve as an additional logic state variable. In conventional CMOS, voltage serves as the single logic state variable. The

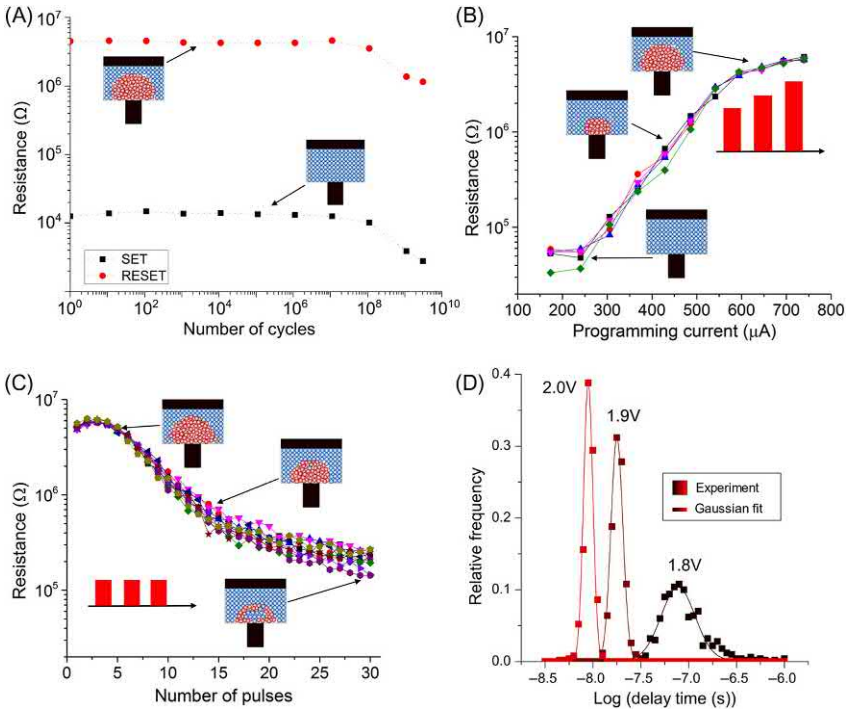


FIGURE 6.2 Experimental measurements on PCM devices are used to illustrate the key physical attributes of memristive devices that facilitate in-memory computing. (A) Binary storage capability whereby the devices can be switched between high- and low-resistance values in a repeatable manner. (B) Multilevel storage capability whereby the devices can be programmed to a continuum of resistance values by the application of appropriate programming pulses. (C) The accumulative behavior whereby the resistance of a device can be progressively decreased by the successive application of identical programming pulses. (D) Intrinsic stochasticity associated with the switching behavior. Experimentally measured delay time distributions from 500 measurements for three different applied voltages. *Adapted from (A) A. Sebastian, M. Le Gallo, E. Eleftheriou, Computational phase-change memory: beyond von neumann computing, J. Phys. D: Appl. Phys. 52 (2019) 443002 [13]. (B and C) A. Sebastian, M. Le Gallo, G.W. Burr, S. Kim, M. BrightSky, E. Eleftheriou, Brain-inspired computing using phase-change memory devices, J. Appl. Phys. 124 (2018) 111101 [14]. (D) M. Le Gallo, T. Tuma, F. Zipoli, A. Sebastian, E. Eleftheriou, Inherent stochasticity in phase-change memory devices, in: 46th European Solid-State Device Research Conference (ESSDERC), IEEE (2016), pp. 373–376 [15].*

input signals are processed as voltage signals and are output as voltage signals. By combining CMOS circuitry with memristive devices it is possible to exploit the additional resistance state variable. For example, the RESET state could indicate logic "0" and the SET state could denote logic "1". This enables logical operations that rely on the interaction between the voltage and resistance state variables and could enable the seamless integration of processing and storage. This is the essential idea behind memristive logic, which is an active area of research [16–18]. The first two chapters in this part illustrate how memristive logic can be applied to problems such as image processing and machine learning. Chapter 7, Memristor-based in-memory logic and its application in image processing, by Haj-Ali et al. presents an overview of memristor-based logic techniques and presents a potential memristive Memory Processing Unit (mMPU). The efficacy of mMPU in performing various image processing tasks is shown. In Chapter 8, Hyperdimensional computing nanosystem: In-memory computing using monolithic 3D integration of RRAM and CNFET by Rahimi et al., presents the applicability of memristive logic in brain-inspired hyperdimensional (HD) computing. At its very core HD computing is about manipulating and comparing large binary vectors (dimensions of approximately 10,000) [19]. Hence, the operations related to HD computing are particularly well suited to memristive logic.

Certain memristive devices can be programmed to not just two levels but a continuum of resistance or conductance values (analog storage capability). For example, Fig. 6.2B shows a continuum of resistance levels in a PCM device achieved by the application of programming pulses with varying amplitude. The device is first programmed to the fully crystalline state, after which RESET pulses are applied with progressively increasing amplitude. The device resistance is measured after the application of each RESET pulse. On account of this property, it is possible to program a memristive device to a certain desired resistance value through iterative programming by applying several pulses in a closed-loop manner [20]. A very useful in-memory computing primitive enabled by the analog storage capability is matrix–vector multiplication [21,22]. The physical laws that are exploited to perform this operation are Ohm's law and Kirchhoff's current summation laws. For example to perform the operation $Ax = b$, the elements of A are mapped linearly to the conductance values of memristive devices organized in a crossbar configuration. The x values are mapped linearly to the amplitudes of read voltages and are applied to the crossbar along the rows. The result of the computation, b, will be proportional to the resulting current measured along the columns of the array. Chapter 9, Vector multiplications using memristive devices and applications thereof, by Zidan and Lu presents a comprehensive review of applications enabled by this computing primitive.

In-memory computing is also enabled by dynamical properties such as the accumulative behavior. For example, in many memristive devices it is

possible to progressively reduce the device resistance by the successive application of SET pulses with the same amplitude. And in certain cases it is possible to progressively increase the resistance by the successive application of RESET pulses. Experimental measurement of this accumulative behavior in a PCM device is shown in Fig. 6.2C. Besides the accumulative behavior, there are additional nonlinear dynamics exhibited by memristive devices such as those based on Mott insulator-metal transition [23]. Chapter 10, Computing with device dynamics, by Bohaichuk and Kumar presents a few examples of the applicability of these dynamical properties in in-memory computing. Note that the accumulative property is also central to applications that involve the training of artificial neural networks that will be covered in depth in the latter two parts of the book.

Yet another physical attribute that can be exploited for in-memory computing is the intrinsic stochasticity associated with the switching behavior in memristive devices [24]. For example, ReRAM, PCM, and MRAM all exhibit significant stochasticity associated with the write voltage and its duration. In an MRAM, the MTJ switching is inherently stochastic due to the thermal fluctuations affecting the free layer and the write voltage and duration can be used to tune the switching probability [25]. In ReRAM, if the write voltage is comparable to the SET voltage, then the SET transition takes place after a certain time delay. The delay time exhibits significant cycle-to-cycle statistical variations [26]. This behavior is also observed in PCM devices and is attributed to the threshold switching dynamics as well as the variability associated with the RESET states [15,27]. In both ReRAM and PCM the delay time distribution can be tuned due to the dependence of the delay time on the write voltage. PCM exhibits additional stochasticity associated with crystallization time that is attributed to the small variations in the atomic configurations of the amorphous volume created upon the preceding RESET [15,28]. Chapter 11, Exploiting the stochasticity of memristive devices for computing by Mizrahi et al., presents a comprehensive overview of how the stochasticity exhibited by memristive devices can be exploited for computing.

6.3 Future outlook

In this book we highlight the outstanding potential of memristive devices for in-memory computing. However, it should be noted that in parallel there has also been significant recent advances in the use of charge-based memory (SRAM and DRAM) for in-memory computing [29–31]. Compared to these memory devices a key advantage of memristive devices is the potential to be scaled to dimensions of a few nanometers [32–35]. Most of the memristive devices are also suitable for back-end-of-line integration, thus enabling their integration with a wide range of front-end CMOS technologies. Another key advantage is the nonvolatility of these devices that would obviate the need for

computing systems to be constantly connected to a power supply. However, there are also challenges that need to be overcome. The significant intradevice and interdevice variability associated with the SET and RESET states is a key challenge for applications where memristive devices are used for logical operations. For applications that rely on analog storage capability, a significant challenge is programming variability that captures the inaccuracies associated with programming an array of devices to desired conductance values. In ReRAM this variability is mostly attributed to the stochastic nature of filamentary switching and one prominent approach to counter this is that of establishing preferential paths for CF formation [36,37]. Representing single computational elements by using multiple memory devices is another promising approach [38]. Yet another challenge is the temporal and temperature-induced variations of the programmed conductance values. The resistance "drift" in PCM devices that is attributed to the intrinsic structural relaxation of the amorphous phase is an example. The concept of projected phase change memory is a promising approach toward tackling "drift" [39,40]. There are also several challenges to be tackled at the peripheral circuit level for in-memory computing, such as the finite resistance of the crossbar wires. This can lead to parasitic voltage drops on the devices during readout when a high current is flowing through them (*IR* drop) and can create computational errors.

The requirements that the memristive devices need to fulfill when employed for computational memory are heavily application dependent. For memristive logic, high cycling endurance ($>10^{12}$ cycles) and low device-to-device variability of the SET/RESET resistance values are critical. For computational tasks involving read-only operations, such as matrix−vector multiplication, the conductance states must remain relatively unchanged during execution. It is also desirable to have a gradual analog-type switching characteristic for programming a continuum of resistance values in a single device. A linear and symmetric accumulative behavior is also required in applications where the device conductance needs to be incrementally updated, such as in neural network training [41]. For stochastic computing applications, random device variability is not problematic, but graceful device degradation is highly desirable [15].

To conclude, in-memory computing using memristive devices is poised to have a significant impact on improving the energy/area efficiency and the latency compared to conventional computing systems with physically separated processing and memory units. In spite of some of the challenges computational memory based on memristive devices could usher in a new era of non-von Neumann accelerators/coprocessors.

References

[1] O. Mutlu, S. Ghose, J. Gómez-Luna, R. Ausavarungnirun, Processing data where it makes sense: Enabling in-memory computation, Microprocessors Microsyst. 67 (2019) 28–41.

[2] M. Horowitz, Computing's energy problem (and what we can do about it), in: Proceedings of the International Solid-state Circuits Conference (ISSCC), IEEE, pp. 10–14.
[3] S.W. Keckler, W.J. Dally, B. Khailany, M. Garland, D. Glasco, GPUs and the future of parallel computing, IEEE Micro 31 (2011) 7–17.
[4] N.P. Jouppi, C. Young, N. Patil, D. Patterson, G. Agrawal, R. Bajwa, et al., In-datacenter performance analysis of a tensor processing unit, in: Proceedings of the International Symposium on Computer Architecture (ISCA), IEEE, pp. 1–12.
[5] A. Sebastian, T. Tuma, N. Papandreou, M. Le Gallo, L. Kull, T. Parnell, et al., Temporal correlation detection using computational phase-change memory, Nat. Commun. 8 (2017) 1115.
[6] J.J. Yang, D.B. Strukov, D.R. Stewart, Memristive devices for computing, Nat. Nanotechnol. 8 (2013) 13.
[7] D. Ielmini, H.-S.P. Wong, In-memory computing with resistive switching devices, Nat. Electron. 1 (2018) 333.
[8] M. Di Ventra, Y.V. Pershin, The parallel approach, Nat. Phys. 9 (2013) 200.
[9] Z. Sun, G. Pedretti, E. Ambrosi, A. Bricalli, W. Wang, D. Ielmini, Solving matrix equations in one step with cross-point resistive arrays, Proc. Natl Acad. Sci. U. S. A. 116 (2019) 4123–4128.
[10] V.V. Zhirnov, M.J. Marinella, Emerging Nanoelectronic Devices, Wiley Online Library.
[11] H.-S.P. Wong, S. Salahuddin, Memory leads the way to better computing, Nat. Nanotechnol. 10 (2015) 191.
[12] L. Chua, Resistance switching memories are memristors, Appl. Phys. A 102 (2011) 765–783.
[13] A. Sebastian, M. Le Gallo, E. Eleftheriou, Computational phase-change memory: beyond von neumann computing, J. Phys. D: Appl. Phys. 52 (2019) 443002.
[14] A. Sebastian, M. Le Gallo, G.W. Burr, S. Kim, M. BrightSky, E. Eleftheriou, Brain-inspired computing using phase-change memory devices, J. Appl. Phys. 124 (2018) 111101.
[15] M. Le Gallo, T. Tuma, F. Zipoli, A. Sebastian, E. Eleftheriou, Inherent stochasticity in phase-change memory devices, 46th European Solid-State Device Research Conference (ESSDERC), IEEE, 2016, pp. 373–376.
[16] J. Borghetti, G.S. Snider, P.J. Kuekes, J.J. Yang, D.R. Stewart, R.S. Williams, Memristive switches enable stateful logic operations via material implication, Nature 464 (2010) 873.
[17] I. Vourkas, G.C. Sirakoulis, Emerging memristor-based logic circuit design approaches: A review, IEEE Circuits Syst. Mag. 16 (2016) 15–30.
[18] S. Kvatinsky, D. Belousov, S. Liman, G. Satat, N. Wald, E.G. Friedman, et al., MAGIC-memristor-aided logic, IEEE Transactions on Circuits and Systems II: Express Briefs 61 (2014) 895–899.
[19] A. Rahimi, S. Datta, D. Kleyko, E.P. Frady, B. Olshausen, P. Kanerva, et al., High-dimensional computing as a nanoscalable paradigm, IEEE Transactions on Circuits and Systems I: Regular Papers 64 (2017) 2508–2521.
[20] N. Papandreou, H. Pozidis, A. Pantazi, A. Sebastian, M. Breitwisch, C. Lam, et al., Programming algorithms for multilevel phase-change memory, in: Proceedings of the International Symposium on Circuits and Systems (ISCAS), IEEE, pp. 329–332.
[21] G.W. Burr, et al., Neuromorphic computing using non-volatile memory, Adv. Physics: X 2 (2017) 89–124.
[22] M.A. Zidan, J.P. Strachan, W.D. Lu, The future of electronics based on memristive systems, Nat. Electron. 1 (2018) 22.

[23] S. Kumar, J.P. Strachan, R.S. Williams, Chaotic dynamics in nanoscale NbO_2 Mott memristors for analogue computing, Nature 548 (2017) 318.
[24] R. Carboni, D. Ielmini, Stochastic memory devices for security and computing, Adv. Electron. Mater. (2019) 1900198.
[25] A. Mizrahi, T. Hirtzlin, A. Fukushima, H. Kubota, S. Yuasa, J. Grollier, et al., Neural-like computing with populations of superparamagnetic basis functions, Nat. Commun. 9 (2018) 1533.
[26] S.H. Jo, K.-H. Kim, W. Lu, Programmable resistance switching in nanoscale two-terminal devices, Nano Lett. 9 (2008) 496–500.
[27] M. Le Gallo, A. Athmanathan, D. Krebs, A. Sebastian, Evidence for thermally assisted threshold switching behavior in nanoscale phase-change memory cells, J. Appl. Phys. 119 (2016) 025704.
[28] T. Tuma, A. Pantazi, M. Le Gallo, A. Sebastian, E. Eleftheriou, Stochastic phase-change neurons, Nat. Nanotechnol. 11 (2016) 693–699.
[29] V. Seshadri, D. Lee, T. Mullins, H. Hassan, A. Boroumand, J. Kim, et al., Ambit: In-memory accelerator for bulk bitwise operations using commodity dram technology, in: Proceedings of the International Symposium on Microarchitecture (MICRO), IEEE, pp. 273–287.
[30] S. Aga, S. Jeloka, A. Subramaniyan, S. Narayanasamy, D. Blaauw, R. Das, Compute caches, in: Proceedings of the International Symposium on High Performance Computer Architecture (HPCA), IEEE, pp. 481–492.
[31] N. Verma, H. Jia, H. Valavi, Y. Tang, M. Ozatay, L.-Y. Chen, et al., In-memory computing: Advances and prospects, IEEE Solid-State Circuits Mag. 11 (2019) 43–55.
[32] F. Xiong, A.D. Liao, D. Estrada, E. Pop, Low-power switching of phase-change materials with carbon nanotube electrodes, Science 332 (2011) 568–570.
[33] K.-S. Li, C. Ho, M.-T. Lee, M.-C. Chen, C.-L. Hsu, J. Lu, et al., Utilizing sub-5 nm sidewall electrode technology for atomic-scale resistive memory fabrication, in: Proceedings of the Symposium on VLSI Technology, IEEE, pp. 1–2.
[34] M. Salinga, B. Kersting, I. Ronneberger, V.P. Jonnalagadda, X.T. Vu, M. Le Gallo, et al., Monatomic phase change memory, Nat. Mater. 1 (2018).
[35] S. Pi, C. Li, H. Jiang, W. Xia, H. Xin, J.J. Yang, et al., Memristor crossbar arrays with 6-nm half-pitch and 2-nm critical dimension, Nat. Nanotechnol. 14 (2019) 35.
[36] S. Brivio, J. Frascaroli, S. Spiga, Role of Al doping in the filament disruption in HfO_2 resistance switches, Nanotechnology 28 (2017) 395202.
[37] S. Choi, S.H. Tan, Z. Li, Y. Kim, C. Choi, P.-Y. Chen, et al., SiGe epitaxial memory for neuromorphic computing with reproducible high performance based on engineered dislocations, Nat. Mater. 17 (2018) 335.
[38] I. Boybat, M. Le Gallo, S. Nandakumar, T. Moraitis, T. Parnell, T. Tuma, et al., Neuromorphic computing with multi-memristive synapses, Nat. Commun. 9 (2018) 2514.
[39] W.W. Koelmans, A. Sebastian, V.P. Jonnalagadda, D. Krebs, L. Dellmann, E. Eleftheriou, Projected phase-change memory devices, Nat. Commun. 6 (2015) 8181.
[40] I. Giannopoulos, A. Sebastian, M. Le Gallo, V. Jonnalagadda, M. Sousa, M. Boon, et al., 8-bit precision in-memory multiplication with projected phase-change memory, in: Proceedings of the International Electron Devices Meeting (IEDM), IEEE, pp. 27–27.
[41] S. Yu, Neuro-inspired computing with emerging nonvolatile memory, Proc. IEEE 106 (2018) 260–285.

Chapter 7

Memristor-based in-memory logic and its application in image processing

Ameer Haj-Ali, Ronny Ronen, Rotem Ben-Hur, Nimrod Wald and Shahar Kvatinsky
Technion – Israel Institute of Technology, Haifa, Israel

7.1 Introduction

A leading cause of inefficient execution in modern von Neumann-based systems has been the separation of the memory from the processing space. Moving data between these spaces incurs high energy and performance overhead. For example, moving data to an off-chip DRAM suffers two orders of magnitude longer delay and consumes three orders of magnitude more energy than the computation itself [1]. This challenge is often called the *memory wall* or the von Neumann bottleneck.

Many works have proposed overcoming this challenge in standard complementary metal–oxide–semiconductor (CMOS) technologies by physically bringing the memory and processing units closer or fabricating them on the same die [2–5]. These in-memory computing (IMC) solutions significantly reduced but did not completely eliminate data movement overhead as data still had to be transferred between the processing and memory units (on the same side). Furthermore, inadequate technology prevented the widespread adoption of IMC. With the emergence of memristive technologies such as Resistive Random Access Memory (RRAM) [6], the IMC concept has been reincarnated. Due to their high switching speed, high endurance, nonvolatility, low operating power, tight integration with CMOS, and ability to both store and perform different logic operations, many memristor-based logic techniques have been proposed to leverage these advantages [7–27] and mitigate the memory wall.

There are various approaches for computing with memristors [28]. For example, memristors can be used as the memory cells that store data; by reading the data and based on the sensed current that unfolds the resistances

Memristive Devices for Brain-Inspired Computing. DOI: https://doi.org/10.1016/B978-0-08-102782-0.00007-1

of the memristors, it is possible to perform different logic operations [29] or sum of products [30–32]. This is very similar to in-memory-periphery logic with conventional memory technologies such as charge sharing in DRAM (e.g., Ambit [5]) and SRAM (e.g., Compute Caches [33]). The memristive crossbar structure in the memory can also be leveraged to perform different logic operations where the output is the final stored data in the memory cell and the inputs are the applied voltages across these cells [10,16,22] or the resistance of the input memristors [34,35].

IMC with memristors has many benefits. To demonstrate these benefits, we conduct a case study on a recently proposed memristor-based logic technique called Memristor Aided loGIC (MAGIC) [34]. It is a promising logic-in-memory approach for executing in-memory computations. It enables the execution of NOR and NOT operations within a memristive memory array, where the inputs and outputs of logic gates at different stages of the computations are represented by the resistance of specific memory cells. Storing the data in RRAM as resistance allows information to be stored and processed using the same cells, with no need for conversion, sensing or moving of data. Much of the recent research on MAGIC and similar techniques has been conducted to exploit this advantage [35–41]. An important feature of MAGIC is its ability to execute multiple gates simultaneously, when their inputs and outputs are located in the same row/column. MAGIC will thus greatly boost the performance of applications that need to execute the same instruction on multiple data in parallel (SIMD).

The advantages of MAGIC can be leveraged to build a memristive Memory Processing Unit (mMPU) where the computation is done directly in the memory cells, consequently overcoming the von Neumann bottleneck. Fig. 7.1 shows the computation in the mMPU as compared with that in the von Neumann architectures. The mMPU consists of a standard RRAM with the required modifications in the controller and peripheral circuits that enable the support of MAGIC operations. Hence, the advantages of a memristive crossbar array, such as density and nonvolatility, are maintained. Furthermore,

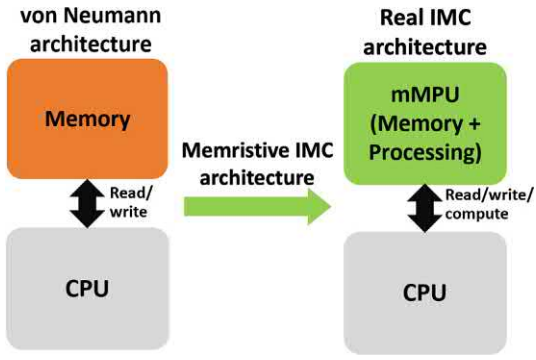


FIGURE 7.1 The computation scheme in the von Neumann machines (on the left) where there is high memory operations traffic (bandwidth bottleneck) versus that of the memristive Memory Processing Unit (mMPU, on the right) in which the memory cells are used to perform logic operations and store the data, thus significantly reducing the memory traffic.

because the mMPU can function as a standard memory, it is completely backward compatible with von Neumann architecture and can operate as a hybrid memory-processing unit or standard memory.

This chapter starts with a brief overview of different memristor-based logic techniques. Then, we present our case study, which uses the example of MAGIC to show the advantages of memristor-based IMC. We further show how data intensive applications that are based on SIMD operations could benefit greatly from computation in the mMPU. To that end, the case study focuses on digital image processing. Image processing is often used to enhance images or extract useful information from them [30,31,42–48]. Since many pixels are processed similarly in parallel, requiring many data transfers and parallel execution, these applications would benefit naturally from MAGIC-based execution in the mMPU. We propose several different algorithms to support image processing within the mMPU. Finally, we evaluate the mMPU and compare it with two other memristor-based IMC architectures: APIM [49] and Pinatubo [29].

7.2 Memristor-based logic

Due to their high switching speed, low operating power, scalability, and high endurance [13], memristors are considered to be attractive candidates to replace conventional memory and storage technologies (e.g., DRAM and Flash). Memristive technologies have also been explored for additional applications such as analog circuits [12,19,21], neuromorphic circuits [7,8,17,23], RF circuits [50], and logic circuits. Different approaches for performing logical operations with memristors have been suggested.

Some memristive logic techniques proposed that processing can be performed near the memory, similar to CMOS-based IMC, where memristors are used only as memory cells, exploiting their density and tight integration with CMOS periphery circuits [9,11,20,27]. In some other logic families, memristors are integrated with CMOS logic structures as configurable switches or as logic gates [14,18,24,25]. In these logic families, the logical values are represented by voltage levels. These techniques cannot be used to perform computation within the memory cells without explicitly reading the data and transforming it from resistance to voltage. To enable computation close to the memory, some of them require dedicated memristive circuits, such as conversion circuits.

Several logic families that use the structure of a memristive memory crossbar array to perform logical operations have been proposed. Linn et al. [10,16,22,51] introduced execution of logic operations on a single memory cell, where the inputs are the applied voltages across the two terminals of the memory cell and the output is the final stored resistance in the memory cell. This was demonstrated for a single memristor and for Complementary Resistive Switches [52]. Because the inputs and outputs are represented

differently (inputs as voltages and resistances as outputs), the outputs must be converted to voltages to be used as the inputs of the next gate.

Similarly, a modified memory array has been proposed to perform logical operations within an Akers array [15], where the inputs are the stored resistive data within the array and the output is voltage. Using this technique, any logic function can be performed in a constant time but with $O(N^2)$ devices where N is the number of inputs. However, supporting these operations is not feasible in pure memristive arrays since four transistors must be added to each memristive cell to support the logic operations. While both techniques can be executed within a memristive memory, they have to perceive (by sensing or reading) either the input or output and transform data from voltage to resistance and vice versa.

Another technique for performing logical operations in a crossbar structure is the fast Boolean logic circuit (FBLC) [26], where any Boolean function could be executed within a constant number of steps. Nevertheless, such operations require the use of disabled memristors (permanently in the high resistance state). Consequently, each computing element is able to perform only one specific logic operation, pre-determined according to the Boolean logic function. This technique thus offers limited programmability and can actually be viewed as adding fixed function processing units near the memory array by sacrificing memory cells.

Another approach is called *stateful logic*, where the logical state of the logic gates is represented solely by the resistance and the inputs and outputs are, respectively, the state of the memristors before and at the end of the computation. The logical operation is based on the application of voltages across the memristors. The result is written to the output based on the initially stored values in the input memristors. By applying a sequence of voltages, more complicated operations could be performed. A few stateful logic families have been proposed that are compatible with memristive crossbar arrays, for example, MAGIC [34,40] and IMPLY [35,53–57]. MAGIC could be used to perform multi-input NOR gates and IMPLY could be used to perform multi-input logical material implication gates. The main differences between these techniques are the different voltages applied across the bitlines and wordlines of the memory array that construct different logic gate structures and the required periphery elements (e.g., the need for an external resistor in each bitline in the IMPLY logic gate [35]).

Reuben et al. [28] provided a taxonomy to characterize and classify different memristor-based logic techniques and proposed a framework for evaluating and comparing them. The techniques are classified by how the input and output are represented (statefulness), the location of the computation with respect to the memristive memory array (proximity of computation), and the possible functionality (flexibility). This classification for different memristor-based logic techniques is summarized in Table 7.1.

TABLE 7.1 The classification of the different logic techniques for memory technologies based on the taxonomy in Ref. [28].

Technique	Statefulness	Proximity	Flexibility	Technology
IMPLY [53]	✓	In-memory-array	✓	Memristor
MAGIC [34]	✓	In-memory-array	✓	Memristor
MRL [14]	×	Out-of-memory	×	Memristor
FBLC [26]	✓	Out-of-memory	×	Memristor
MAJ [51]	×	In-memory-periphery	✓	Memristor
Akers [15]	×	In-memory-periphery	×	Memristor
APIM [49]	×	In-memory-periphery	✓	Memristor
Pinatubo [29]	×	In-memory-periphery	✓	Memristor
Ambit [5]	×	In-memory-periphery	✓	DRAM
Compute Caches [33]	×	In-memory-periphery	✓	SRAM

A memristive logic family is *stateful* if the Boolean variable is represented only as the state of the memristor (i.e., resistance) and computation is performed by manipulating this state. For example, Pinatubo is not stateful. To compute in Pinatubo [29], the data are sensed as voltage in the periphery, where dedicated circuits perform the logic gates, and the result is written back to the memristive memory array. On the other hand, MAGIC is stateful because performing logic on the input resistances directly updates the resistance of the output based on the logical states (resistances) of these inputs.

The *proximity of computation* is divided into three categories: in-memory-array,[1] in-memory-periphery[1] and out-of-memory. (1) If the computation is done directly in the memory cells without any data movement or conversion, then it is considered in-memory-array (e.g., IMPLY and MAGIC [34,35]); otherwise it could be (2) in-memory-periphery if the data must be moved out of the memory array during the computation (e.g., if the data needs to be read during the computation for conversion or when part of the computation is performed by using the peripheral circuits as the case in MAJ

1. In this chapter we use the terms in-memory-array and in-memory-periphery. However, Reuben et al. [28], referred to in-memory-array as in-memory and in-memory-periphery as near memory. While the names differ, the principles are identical.

Akers, APIM, Pinatubo, Ambit, and Compute Caches [5,15,29,33,49,51]), or (3) out-of-memory if the entire computation is performed outside of the memory array, such as FBLC and MRL [14,26].

Flexibility is achieved if a variety of operations can be executed using the same computing elements. In other words, a logic family has to provide a basic operation (or a set thereof), which is functionally complete. For example, Ambit [5] performs in-memory-periphery logic using OR, AND, and NOT gates in the periphery. These gates form a functionally complete set and thus Ambit is flexible.

7.2.1 Memristor Aided loGIC (MAGIC)

The focus of this chapter is MAGIC [34]. In MAGIC, only a single voltage is used to perform a NOR logic operation, and there are separate input and output memristors. Fig. 7.2 shows a schematic of a MAGIC NOR gate and how it is performed over column vectors within a memristive memory. First, the output(s) is initialized to R_{ON} (logical "1") by applying V_{SET}. Then, a voltage pulse, V_G, is applied. Due to the polarity of the output, its resistance could either remain constant if the voltage drop on it is not high enough or switch to R_{OFF} (logical "0") if the voltage drop is sufficient to cause switching. If both inputs are in the R_{OFF} state, then the voltage drop on the output will be very low due to the voltage divider being between R_{ON} and $R_{OFF}/2$ ($R_{OFF}/2 >> R_{ON}$). In any other case, at least one of the inputs will be in the R_{ON} state and thus the voltage on the output will be at least $V_G/2$, which is sufficient to switch it to R_{OFF}. This behavior is equivalent to a NOR gate.

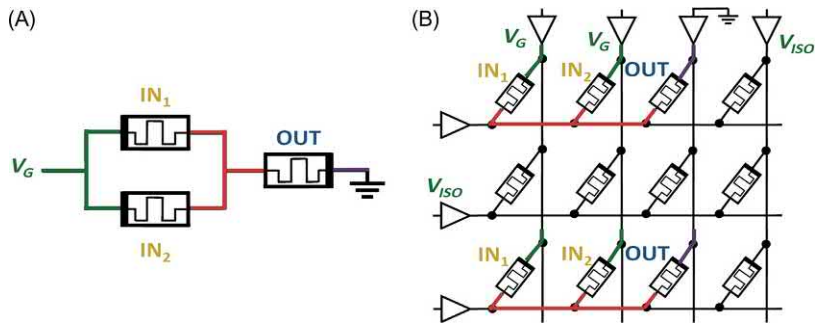


FIGURE 7.2 Schematic of a (A) MAGIC NOR gate and a (B) MAGIC NOR gate within a memristive memory array. IN_1 and IN_2 are the input memristors and OUT is the output memristor. The inputs are the initial resistances (states) of the input memristors and the output is the resistance (state) of the output memristor at the end of the computation. To perform the NOR operation, a single voltage V_G is applied to the bitline of the inputs, ground is applied to the bitline of the output memristors, and V_{ISO} (the isolation voltage) is applied to unselected bitlines and wordlines.

Note that a single input NOR is basically a NOT gate, and that the same principle can be adapted to any desired number of inputs.

Since NOR is functionally complete, a MAGIC NOR operation is sufficient to execute any Boolean operation. As a pure stateful logic family, MAGIC enables real in-memory-array processing since data need not be read/sensed during the computation; the data are processed using only memory cells chosen by a memory controller, thus eliminating the data transfer. MAGIC NOR can enable the execution of any function by dividing it into a sequence of MAGIC NOR operations. These basic NOR operations will be executed one after the other using the memory cells as computation elements. An additional important advantage of MAGIC is its ability to perform logic operations in parallel on sets of data. Due to the structure of the crossbar array, applying the operating voltage V_G on any two selected columns and grounding a third column will result in NOR operations being performed on all selected rows (isolation of rows is possible if desired). This allows massive parallelism within the memory, which is independent of the data size. Due to the symmetry of memristive crossbar arrays, performing NOR operations on row vectors is feasible in a similar manner. The advantages of MAGIC make it an attractive basis for computation in a memristive Memory Processing Unit (mMPU), where the computation is performed directly in the memory cells, thus mitigating the von Neumann bottleneck. The mMPU is presented in Section 7.3.

7.2.2 Digital image processing

Digital image processing is a method for analyzing and manipulating digitized images to get an enhanced image or to extract useful information from it [30,31,42–48,58,59]. One of the most common operations for image manipulation is convolution, where a kernel is slided over an image and its values are multiplied by the corresponding pixel values of the image. Numerous different kernels exist, whose size and values determine the exact operation. Image and video processing have become extremely important applications in many fields such as medical imaging, image recognition, computational photography, autonomous vehicles, and others [42,45–48].

Among the prominent image processing tasks are the Hadamard product [43]. In the Hadamard product, two images of the same size are multiplied element-by-element, producing an image of the same size. Pixel (i,j) of the output image when performing the Hadamard product of two images, $image_1$ and $image_2$, is calculated by

$$out_{(i,j)} = image1_{(i,j)} \cdot image2_{(i,j)}. \tag{7.1}$$

2D convolution between a kernel and an image yields a filtered image [44]. In the convolution process, the value of each pixel of the output image is determined by the values of its neighbor pixels in the input image, and by

the weights of the kernel. Convolution with different size and value of kernels leads to diverse processing operations, such as blurring, sharpening, edge detection and more [60]. Pixel(i,j) of the output image when convoluted with a kernel w of size $P \times P$ is calculated by

$$out_{(i,j)} = \sum_{n=0}^{P-1} \sum_{m=0}^{P-1} in_{(i-n-\lfloor \frac{P}{2} \rfloor, j-m-\lfloor \frac{P}{2} \rfloor)} \cdot w_{(n,m)}. \tag{7.2}$$

7.2.3 Previous attempts to accelerate image processing with memristors

Previous attempts to accelerate image processing tasks with memristors [30–32] have relied on analog based computation using the accumulation of currents in analog-to-digital converters (ADCs) to perform sum of products. The input multipliers are represented as voltage while the multiplicands are stored as resistances in multi-level cells (MLC) [61–64]. While these approaches are efficient, they suffer from limited precision and reliability when compared to digital computation since the number of levels in MLC is limited and ADCs have limited accuracy. Therefore digital computation is often necessary when precision is a concern. For example, digital image convolution is used to implement demosaicing of Bayer color arrays [48,65], which is used in most digital camera image sensors. Additionally since these approaches are restricted to predefined tasks that are based on the sum of products operation, they cannot be programmable. By contrast, since MAGIC NOR is functionally complete, any task could be mapped to a sequence of MAGIC NOR operations. Furthermore applying different voltage levels requires large digital-to-analog converters, and sensing different current levels requires huge ADCs. This complex and considerable area overhead significantly restricts the area efficiency of the memory.

In-memory-periphery approaches with memristors for digital image processing include Pinatubo [29] and APIM [49]. These approaches use standard CMOS logic in the periphery to compute. Pinatubo uses the periphery to perform bit-wise operations while APIM uses the periphery to generate the partial products. Therefore, Pinatubo and APIM move the data serially to and from the periphery for every operation, which the mMPU avoids with the MAGIC-based execution. APIM and Pinatubo are compared with the mMPU in Section 7.5.

7.3 The memristive Memory Processing Unit

The mMPU [36,66] is a standard RRAM memory with a few modifications that enable the support of MAGIC-based IMC instructions. In other words, the mMPU functions as a standard memory that supports memory operations

(such as, read and write) with additional IMC capabilities, and thus it is backward compatible with the von Neumann computing scheme. The mMPU architecture is shown in Fig. 7.3. To support IMC instructions, the memory controller, the memory protocol, and the peripheral circuits (such as, voltage drivers and row/column decoders) must be modified to support MAGIC instructions. The mapping of data is also modified to maintain persistence and coherence. Note however that the memory crossbar structure itself is not modified.

The mMPU CMOS controller is a finite state machine that supports standard and IMC memory instructions by generating the necessary control signals. The controller receives the commands from the CPU and performs the decoded instruction. The IMC instructions are translated to a pre-synthesized and optimized sequence of MAGIC NOR gates. To execute different applications in-memory-array using MAGIC, algorithms that translate these applications to an optimized sequence of MAGIC NOR/NOT gates must be developed [67]. In Fig. 7.4, an example for performing bitwise OR between two n-bit vectors (A and B) using MAGIC is shown. In the first cycle, columns three and four are initialized to R_{ON} simultaneously by applying V_{SET}. In the second cycle, V_G is applied to the first two columns and the third column is grounded so that the outputs in the third column now store $NOR(A, B)$. Finally, in the third cycle, V_G is applied to column three and the fourth column is grounded so that the outputs in the fourth column now store $OR(A, B)$.

The optimized algorithms to perform sequences of MAGIC NOR and NOT operations could be generated manually or automatically. In Ref. [39], we proposed optimized, manual algorithms for performing Fixed-Point (FiP) multiplication using MAGIC and further extended them in Ref. [38] to perform different image processing tasks. For automatically generated algorithms, we proposed SIMPLE [37]: an automatic synthesis tool that receives any Boolean function automatically and generates the equivalent, optimal sequence of MAGIC NOR operations. The operation is converted to a NOR CMOS-based netlist which is mapped to a sequence of MAGIC NOR gates by solving an

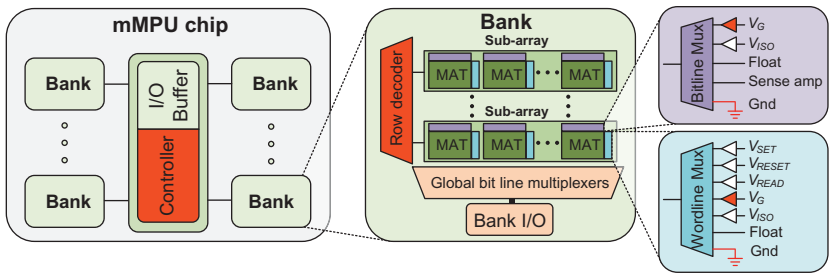


FIGURE 7.3 The mMPU chip architecture. The only modifications to conventional RRAM chip architecture (shown in red) are in the controller and peripheral circuits.

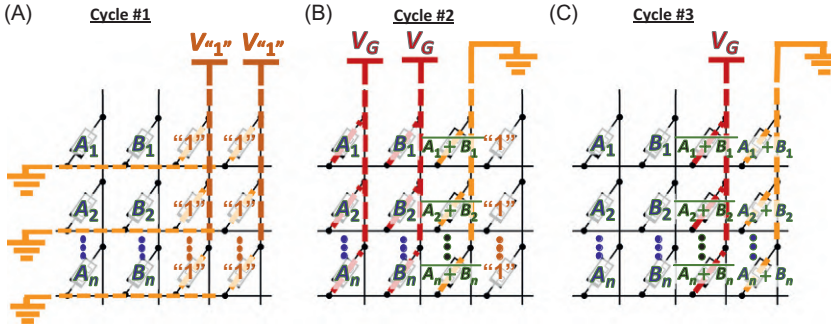


FIGURE 7.4 An example of an algorithm that performs bitwise OR operations between two n-bit vectors *i.e.*, A and B. (A) In the first cycle, the memristors in columns #3 and #4 are initialized to logical "1" (R_{ON}) so they can be used to perform MAGIC gates. (B) In the second cycle, V_G is applied to the memristors in columns #1 and #2 and column #3 is grounded so that a MAGIC NOR gate is performed, resulting in $(A + B)'$ in the memristors of column #3. (C) Finally, in the third cycle, V_G is applied to the memristors in column #3 and column #4 is grounded so that a MAGIC NOT gate is performed, resulting in $A + B$ in the memristors of column #4.

optimization problem. Such a tool will serve as the basis for the mMPU controller design, with the manual mapping left for specific tasks.

Due to the parallel nature of MAGIC, the tasks that benefit most from execution in the mMPU consist of simple SIMD operations. Each operation will be performed in a single row. For example, to add two vectors of 512 elements each in a MAT of size 512×512, each row in the MAT will store two elements, one from each vector, and all the elements from each vector will share the same columns so that all the elements will be added simultaneously. This execution scheme substantially improves the throughput (number of executions per cycle). Translating applications such as image processing [30,31,42–48] for execution in the mMPU will thus be very attractive as they are based on simple instructions (add/multiply) that are executed similarly on all the inputs and the demand for data movement (which the mMPU avoids) only increases as image resolution becomes higher.

7.3.1 Challenges of the memristive Memory Processing Unit

When processing multiple elements simultaneously in the mMPU, data alignment is a challenge as the physical addresses of these elements have to share the same wordlines/bitlines to be executed simultaneously. Therefore, two operands that need to be processed but are stored in different wordlines/bitlines must be first aligned, that is, copied to addresses that share the same wordlines/bitlines in the same MAT. Such alignment could be achieved using driver hints indicating which operands should be mapped to the same wordlines/bitlines. However, if such a mapping is not achieved, we proposed

multiple techniques to organize the data after it has been stored, based on the addresses of the inputs and outputs [41]. If the operands are stored in the same MAT but in different wordlines/bitlines, then MAGIC NOT gates could be used to align them. However, aligning operands that are stored in different MATs requires one operand to be read to the bank I/O via the sense amplifiers, transferring it to the bank I/O of the destination bank via the internal bus of the chip if the destination bank is different from the current bank, and writing it to the desired MAT in an address that shares the same wordlines/bitlines with the other operand.

Power and endurance limitations are additional important challenges. The execution using MAGIC results in periodic writing to the output memristors, which wears them out, decreases the memory lifetime, and consumes power. In Refs. [38,41], we investigated the impact of these challenges. Alleviating their impact is possible but might limit the performance or require technological improvements.

7.4 Performing image processing in the memristive Memory Processing Unit

7.4.1 Fixed-Point multiplication

The FiP multiplication is used in most digital image processing applications [30,31]. It is very similar to standard integer multiplication with an implied decimal point, which permits fractional results. Therefore it is implemented by generating the partial products and accumulating them [68].

7.4.1.1 Performing Fixed-Point multiplicating using MAGIC

To perform FiP multiplication inside the acceptably sized memristive memory arrays in the mMPU, we have proposed the full precision FiP multiplication (FPFiPM) algorithm. The algorithm generates the partial products and adds them. This algorithm reuses the memristors that store data no longer needed for future computation, significantly reducing the area overhead. The additions of partial products are performed serially; therefore, each adder could be implemented in the same area and the partial products could be generated one after the other and stored in the same memristors. Fig. 7.5 shows this computing scheme. Note that memristor reuse requires initializing the memristors to R_{ON} before each computation to perform MAGIC [34]. Further details and possible optimizations are available in Ref. [38].

7.4.2 MAGIC-based algorithms for image processing

To perform the Hadamard product between two images, the images are stored adjacent to each other inside the MAT. Then every two equivalent FiP vectors are multiplied (element-wise). To perform image convolution

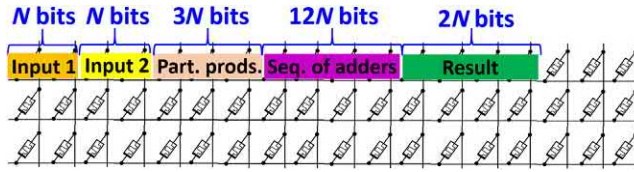


FIGURE 7.5 The proposed FPFiPM algorithm computing scheme. Only $3N$ memristors are used to store all the partial products and $12N$ memristors for all the adders [40]. After each partial product generation and addition, the $15N$ memristors are initialized to R_{ON} and used again in the next computation.

between an image of size $H \times W$ and a kernel of size $P \times P$, the image is duplicated P times, and the duplicates are stored one below the other next to H duplicates of the kernel. The convoluted image is obtained after performing H multiply-accumulate operations. The latency and area for all the proposed algorithms are summarized in Table 7.2. More details on the proposed algorithms are given in Ref. [38].

Due to the limited size of the MATs, some images might not fit in a single MAT. Therefore, the images should be split into multiple MATs, where each split needs to be processed independently using the same algorithm. Since these splits could be processed simultaneously, the parallelism will be further improved and the performance will be better than having a sufficiently large array that could fit the entire image. For 8-bit precision and an array of size 512×512, the area used for storing the inputs and performing the computations limits the width of the image to 8 in image convolution, and to 12 in the Hadamard product. While the height of the image in Hadamard product is 512, the height in image convolution with a $P \times P$ kernel is limited to $\left\lfloor \frac{512}{P} \right\rfloor$ due to the P required duplicates of the original input image. The maximum split size for each algorithm is listed in Table 7.3.

The image processing algorithms assume that the data are organized in a specific way that optimizes the parallelism. This might be the case in a system dedicated to image processing. However, if this is not the case the techniques proposed in Section 7.3 are used to align the data using MAGIC NOT gates. The data organization overheads are summarized in Table 7.4.

7.5 Evaluation

7.5.1 Methodology

We evaluate the image processing algorithms and compare them with two other memristive logic techniques: APIM and Pinatubo. To evaluate the performance of the image processing tasks, we built a cycle-accurate, functional simulator in MATLAB that accurately performs the logical flow of each algorithm. The simulator considers the array size, the precision (N), the dimensions of the images, the locations and the worst case organization of the inputs.

TABLE 7.2 Expressions for latency and area of the proposed algorithms.

Algorithm	Latency (cycles)	Area (#*rows* × #*columns*)
Hadamard product	$W(13N^2 - 16N + 6)$	$(4NW + 16N - 5) \times H$
Image convolution	$WP(13N^2 + 32N - 4) - W(46N - 10) + \#colors \cdot H(P-1)$	$(P(H + P - 1)\#colors) \times (5WN + PN + 21N - 5)$

N is the number of precision bits in each number. H and W are the height and width of the images respectively ($H \times W$), and P is the size of the kernel in convolutions ($P \times P$). #*colors* is 3 for RGB and 1 for gray-scale images.

TABLE 7.3 The maximum split size for each algorithm in a single 512×512 array with 8-bit precision.

Split size (dimensions)	
Hadamard product	**Image convolution**
512×12	$\left\lfloor \frac{512}{P} \right\rfloor \times 8$

TABLE 7.4 The data organization latency overhead for each algorithm.

Latency (cycles)	
Hadamard product	**Image convolution**
$MAX(2H, H + WN)$	$2P \cdot (H + P - 1) \cdot \#colors$

APIM performs image processing by using an in-memory-periphery approach where the periphery is used to speed up the generation of partial products and then applying MAGIC NOR-based fast carry-save adders on the intermediate results. The latency of this adder is 285, 551, and 1057 MAGIC cycles for 8, 16, and 32-bit numbers [38]. Pinatubo is in-memory-periphery architecture, where the data are stored in the memristive array but computed in the periphery using dedicated CMOS logic, which supports bit-wise logic operations, such as, OR, AND, XOR, and NOT. We use algorithms similar to ours in Pinatubo to perform image processing by replacing the MAGIC NOR gates with the optimal combination of XOR, AND, and OR gates.

The size of each MAT is 512×512. To model the memristor we use the VTEAM [69] model, where the device parameters fit the HfOX-based bipolar memristor [70]. The parameters for MAGIC NOR gates and read/write operations are extracted from SPICE simulations. The ratio of read, write and MAGIC latencies is, respectively, 1:2.5:3.25 where the read latency is 8.9 ns [29]. For an apples-to-apples comparison, we exclude the performance and energy overhead of the CMOS logic in the APIM and Pinatubo periphery. Note that the latency and energy of APIM and Pinatubo are higher in practice (when the CMOS logic overheads are included); our results are conservative in that they give APIM and Pinatubo an advantage.

For the image processing tasks, we use the CIFAR-10 [71] image classification benchmark dataset, a test set of 10,000 images, where instances are 32×32 color (RGB) images representing airplanes, automobiles, birds, cats, deer, dogs, frogs, horses, ships, and trucks. For image convolution we run a layer of 3×3 filters used for sharpening and edge detection on the dataset. The filters are slided over the images and their values are multiplied by the corresponding pixel values. For the Hadamard product we perform element-wise multiplications between the images and 32×32 matrices used for pattern recognition and lossy compression algorithms such as JPEG.

7.5.2 Performance

Fig. 7.6 shows the speedup of the mMPU for the Hadamard product and image convolution as compared with APIM and Pinatubo. More detailed results are available in Refs. [38,66]. For the Hadamard product and image convolution, the mMPU improves the performance by $35\times$ and $195\times$ over Pinatubo and APIM, on average, respectively, thanks to the parallelism MAGIC enables. Moreover, APIM performs complex computations in the periphery, rendering its computation serial and Pinatubo's bandwidth of moving the data to and from the periphery is limited due to the bulky sense amplifiers in RRAM, which consequently limits overall system performance. Furthermore, the logic added by APIM and Pinatubo to every single memory array complicates the periphery and lowers the capacity of the memory.

7.5.3 Energy

Fig. 7.7 shows the energy efficiency improvement of the mMPU for the Hadamard product and image convolution over APIM and Pinatubo. The effective energy overhead per logic gate performed decreases as more gates are performed simultaneously due to the dominant energy consumption in the parasitic resistances and sneak paths [72,73]. In MAGIC, all the rows execute MAGIC NOR gate simultaneously, and thus the energy cost per gate is lower than that in Pinatubo and APIM. Furthermore, the mMPU provides high parallelism and eliminates the data movement allowing it to improve

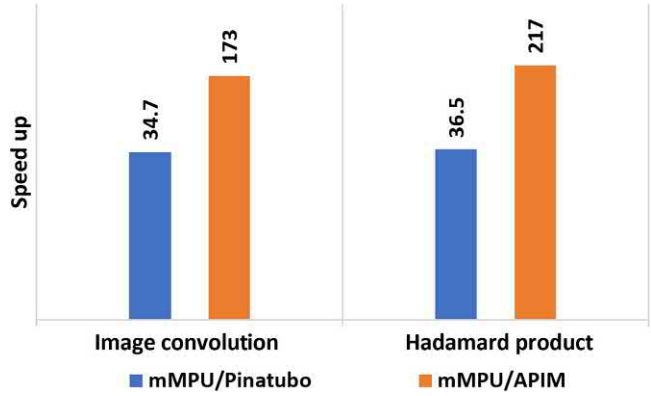


FIGURE 7.6 Speedup of the mMPU when compared with APIM and Pinatubo.

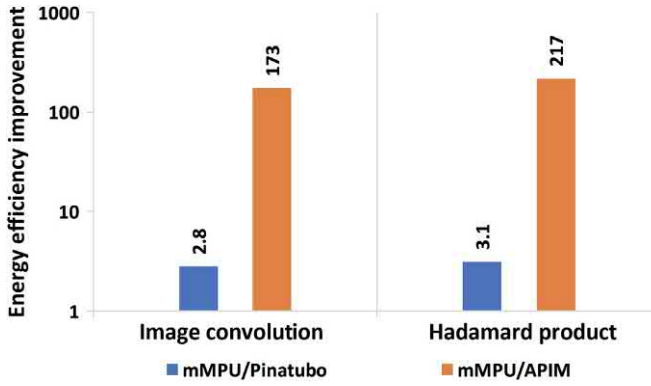


FIGURE 7.7 Energy efficiency improvement of the mMPU when compared with APIM and Pinatubo.

the energy efficiency by 2.95× and 195× over Pinatubo and APIM, on average, respectively. The improvement in energy efficiency over APIM shows the same trend as the improvement in speedup: the energy dissipated in the periphery when reading the data is dwarfed by the massive computation energy of performing MAGIC. Note that while the higher parallelism enabled in the mMPU reduces both the energy and latency, the total power may be a limitation.

7.6 Conclusions

In this chapter, we overviewed the different memristor-based in-memory logic techniques and showed their benefits in enabling efficient IMC. This was demonstrated in a case study conducted on MAGIC deployed in a mMPU. The

mMPU supports ample parallelism that was leveraged to perform image processing, consequently providing excellent performance and energy efficiency. Image processing is one application that could benefit from memristor-based IMC. We believe that many more data intensive and highly parallel applications could benefit from logic in memory with memristors.

References

[1] A. Pedram, S. Richardson, M. Horowitz, S. Galal, S. Kvatinsky, Dark memory and accelerator-rich system optimization in the dark silicon era, IEEE Des. Test. 34 (2) (2017) 39–50.

[2] P. Dlugosch, D. Brown, P. Glendenning, M. Leventhal, H. Noyes, An efficient and scalable semiconductor architecture for parallel automata processing, IEEE Trans. Parallel Distrib. Syst. 25 (12) (2014) 3088–3098.

[3] M. Oskin, F.T. Chong, T. Sherwood, Active pages: a computation model for intelligent memory, in: Proceedings. 25th Annual International Symposium on Computer Architecture, June 1998.

[4] D. Patterson, T. Anderson, N. Cardwell, R. Fromm, K. Keeton, C. Kozyrakis, et al., A case for intelligent RAM, IEEE Micro 17 (2) (1997) 34–44.

[5] V. Seshadri, D. Lee, T. Mullins, H. Hassan, A. Boroumand, J. Kim, et al., Ambit: in-memory accelerator for bulk bitwise operations using commodity DRAM technology, in: Proceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture, October 2017.

[6] C. Xu, D. Niu, N. Muralimanohar, R. Balasubramonian, T. Zhang, S. Yu, et al., Overcoming the challenges of crossbar resistive memory architectures, in: 2015 IEEE 21st International Symposium on High Performance Computer Architecture (HPCA), February 2015, pp. 476–488.

[7] S.P. Adhikari, C. Yang, H. Kim, L.O. Chua, Memristor bridge synapse-based neural network and its learning, IEEE Trans. Neural Netw. Learn. Syst. 23 (9) (2012) 1426–1435.

[8] A. Afifi, A. Ayatollahi, F. Raissi, Implementation of biologically plausible spiking neural network models on the memristor crossbar-based CMOS/nano circuits, Circuit Theory and Design, 2009. (ECCTD 2009). European Conference on Circuit Theory and Design, IEEE, 2009, pp. 563–566.

[9] K. Eshraghian, K.R. Cho, O. Kavehei, S.K. Kang, D. Abbott, S.M.S. Kang, Memristor MOS content addressable memory (MCAM): hybrid architecture for future high performance search engines, IEEE Trans. Very Large Scale Integr. (VLSI) Syst. 19 (8) (2011) 1407–1417.

[10] P.E. Gaillardon, L. Amaru, A. Siemon, E. Linn, R. Waser, A. Chattopadhyay, et al., The programmable logic-in-memory (PLiM) computer, in: 2016 Design, Automation Test in Europe Conference Exhibition (DATE), March 2016, pp. 427–432.

[11] Q. Guo, X. Guo, Y. Bai, E. Ipek, A resistive TCAM accelerator for data-intensive computing, in: Proceedings of the 44th Annual IEEE/ACM International Symposium on Microarchitecture, December 2011, pp. 339–350.

[12] M. Itoh, L.O. Chua, Memristor oscillators, Int. J. Bifurc. Chaos 18 (11) (2008) 3183–3206.

[13] S. Kvatinsky, E.G. Friedman, A. Kolodny, U.C. Weiser, The desired memristor for circuit designers, IEEE Circuits Syst. Mag. 13 (2) (May 2013) 17–22.

[14] S. Kvatinsky, N. Wald, G. Satat, A. Kolodny, U.C. Weiser, E.G. Friedman, MRL - memristor ratioed logic, in: 2012 13th International Workshop on Cellular Nanoscale Networks and Their Applications, August 2012, pp. 1–6.

[15] Y. Levy, J. Bruck, Y. Cassuto, E.G. Friedman, A. Kolodny, E. Yaakobi, et al., Logic operations in memory using a memristive akers array, Microelectron. J. 45 (11) (2014) 1429–1437.
[16] E. Linn, R. Rosezin, S. Tappertzhofen, U. Bottger, R. Waser, Beyond von Neumann-logic operations in passive crossbar arrays alongside memory operations, Nanotechnology 23 (30) (2012).
[17] X. Liu, Z. Zeng, S. Wen, Implementation of memristive neural network with full-function Pavlov associative memory, IEEE Trans. Circuits Syst. I: Regul. Pap. 63 (9) (2016) 1454–1463.
[18] A.K. Maan, D.S. Kumar, S. Sugathan, A.P. James, Memristive threshold logic circuit design of fast moving object detection, IEEE Trans. Very Large Scale Integr. (VLSI) Syst. 23 (10) (2015) 2337–2341.
[19] D. Mahalanabis, V. Bharadwaj, H.J. Barnaby, S. Vrudhula, M.N. Kozicki, A Nonvolatile sense amplifier flip-flop using programmable metallization cells, IEEE J. Emerg. Sel. Top. Circuits Syst. 5 (2) (2015) 205–213.
[20] A. Morad, L. Yavits, S. Kvatinsky, R. Ginosar, Resistive GP-SIMD processing-in-memory, CM. Trans. Architecture Code Optim. (TACO) 12 (4) (2016) 57:1–57:22.
[21] Y.V. Pershin, M. Di Ventra, Practical approach to programmable analog circuits with memristors, IEEE Trans. Circuits Syst. I: Regul. Pap. 57 (8) (2010) 1857–1864.
[22] A. Siemon, S. Menzel, R. Waser, E. Linn, A complementary resistive switch-based crossbar array adder, IEEE J. Emerg. Sel. Top. Circuits Syst. 5 (1) (2015) 64–74.
[23] D. Soudry, D. Di Castro, A. Gal, A. Kolodny, S. Kvatinsky, Memristor-based multilayer neural networks with online gradient descent training, IEEE Trans. Neural Netw. Learn. Syst. 26 (10) (2015) 2408–2421.
[24] D.B. Strukov, K.K. Likharev, CMOL FPGA: a reconfigurable architecture for hybrid digital circuits with two-terminal nanodevices, Nanotechnology 16 (6) (2005) 888.
[25] W. Wang, T.T. Jing, B. Butcher, FPGA based on integration of memristors and CMOS devices, in: Proceedings of the 2010 IEEE International Symposium on Circuits and Systems, May 2010, pp. 1963–1966.
[26] L. Xie, H.A.D. Nguyen, M. Taouil, S. Hamdioui, K. Bertels, Fast Boolean logic mapped on memristor crossbar, in: 2015 33rd IEEE International Conference on Computer Design (ICCD), October 2015, pp. 335–342.
[27] L. Yavits, S. Kvatinsky, A. Morad, R. Ginosar, Resistive associative processor, IEEE Comput. Architect. Lett. 14 (2) (2015) 148–151.
[28] J. Reuben, R. Ben-Hur, N. Wald, N. Talati, A. Haj-Ali, P.-E. Gaillardon, et al., Memristive logic: a framework for evaluation and comparison, in: 27th International Symposium on Power and Timing Modeling, Optimization and Simulation (PATMOS), September 2017.
[29] S. Li, C. Xu, Q. Zou, J. Zhao, Y. Lu, Y. Xie, Pinatubo: a processing-in-memory architecture for bulk bitwise operations in emerging non-volatile memories, in: 2016 53nd ACM/EDAC/IEEE Design Automation Conference (DAC), June 2016, pp. 1–6.
[30] P. Chi, S. Li, C. Xu, T. Zhang, J. Zhao, Y. Liu, et al., PRIME: a novel processing-in-memory architecture for neural network computation in ReRAM-based main memory, in: Proceedings of the 43rd International Symposium on Computer Architecture, 2016, pp. 27–39.
[31] A. Shafiee, A. Nag, N. Muralimanohar, R. Balasubramonian, J.P. Strachan, M. Hu, et al., ISAAC: a convolutional neural network accelerator with in-situ analog arithmetic in crossbars, in: 2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA), June 2016, pp. 14–26.

[32] L. Song, X. Qian, H. Li, Y. Chen, PipeLayer: a pipelined ReRAM-based accelerator for deep learning, in: 2017 IEEE International Symposium on High Performance Computer Architecture (HPCA), February 2017, pp. 541–552.
[33] S. Aga, S. Jeloka, A. Subramaniyan, S. Narayanasamy, D. Blaauw, R. Das, Compute caches, in: 2017 IEEE International Symposium on High Performance Computer Architecture (HPCA), February 2017, pp. 481–492.
[34] S. Kvatinsky, D. Belousov, S. Liman, G. Satat, N. Wald, E.G. Friedman, et al., MAGIC - Memristor-Aided Logic, IEEE Trans. Circuits Syst. II: Express Briefs 61 (11) (2014) 895–899.
[35] S. Kvatinsky, G. Satat, N. Wald, E.G. Friedman, A. Kolodny, U.C. Weiser, Memristor-based material implication (IMPLY) logic: design principles and methodologies, IEEE Trans. Very Large Scale Integr. (VLSI) Syst. 22 (10) (2014) 2054–2066.
[36] R. Ben-Hur, S. Kvatinsky, Memory processing unit for in-memory processing, in: 2016 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH), July 2016, pp. 171–172.
[37] R. Ben-Hur, N. Wald, N. Talati, S. Kvatinsky, SIMPLE MAGIC: Synthesis and in-memory MaPping of Logic Execution for Memristor-Aided loGIC, in: International Conference on Computer-Aided Design (ICCAD), November 2017.
[38] A. Haj-Ali, R. Ben-Hur, N. Wald, R. Ronen, S. Kvatinsky, IMAGING: In-Memory AlGorithms for Image processiNG, IEEE Trans. Circuits Syst. I: Regul. Pap. 65 (12) (2018) 4258–4271.
[39] A. Haj-Ali, R. Ben-Hur, N. Wald, S. Kvatinsky, Efficient algorithms for in-memory fixed point multiplication using MAGIC, in: 2018 IEEE International Symposium on Circuits and Systems (ISCAS), May 2018.
[40] N. Talati, S. Gupta, P. Mane, S. Kvatinsky, Logic design within memristive Memories Using Memristor-Aided loGIC (MAGIC), IEEE Trans. Nanotechnol. 15 (4) (2016) 635–650.
[41] N. Talati, A. Haj-Ali, R. Ben-Hur, N. Wald, R. Ronen, P.-E. Gaillardon, et al., Practical challenges in delivering the promises of real processing-in-memory machines, in: 2018 Design, Automation Test in Europe Conference Exhibition (DATE), March 2018.
[42] D.S. Bolme, J.R. Beveridge, B.A. Draper, Y.M. Lui, Visual object tracking using adaptive correlation filters, in: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, June 2010, pp. 2544–2550.
[43] R.A. Horn, The Hadamard product, Matrix Theory Appl. 40 (1990) 87–169.
[44] R. Jain, R. Kasturi, B.G. Schunck, Machine Vision, vol. 5, McGraw-Hill, New York, 1995.
[45] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems 25, December 2012, pp. 1097–1105.
[46] B. Shen, I.K. Sethi, V. Bhaskaran, DCT convolution and its application in compressed domain, IEEE Trans. Circuits Syst. Video Technol. 8 (8) (1998).
[47] A. Skodras, C. Christopoulos, T. Ebrahimi, The JPEG 2000 still image compression standard, IEEE Signal. Process. Mag. 18 (5) (2001) 36–58.
[48] R. Zhen, R.L. Stevenson, Image demosaicing, Color Image Video Enhancement (2015) 13–54.
[49] M. Imani, S. Gupta, T. Rosing, Ultra-efficient processing in-memory for data intensive applications, in: Proceedings of the 54th Annual Design Automation Conference 2017, June 2017.

[50] N. Wainstein, S. Kvatinsky, A lumped rf model for nanoscale memristive devices and nonvolatile single-pole double-throw switches, IEEE Trans. Nanotechnol. 17 (5) (2018) 873–883.

[51] S. Shirinzadeh, M. Soeken, P.-E. Gaillardon, R. Drechsler, Fast logic synthesis for RRAM-based in-memory computing using majority-inverter graphs, in: Proceedings of the 2016 Conference on Design, Automation & Test in Europe, March 2016, pp. 948–953.

[52] E. Linn, R. Rosezin, C. kugeler, R. Waser, Complementary resistive switches for passive nanocrossbar memories, Nat. Mater. 9 (5) (2010) 403.

[53] J. Borghetti, G.S. Snider, P.J. Kuekes, J.J. Yang, D.R. Stewart, R.S. Williams, 'Memristive' switches enable 'stateful' logic operations via material implication, Nature 464 (7290) (2010) 873–876.

[54] S. Kvatinsky, A. Kolodny, U.C. Weiser, E.G. Friedman, Memristor-based IMPLY logic design procedure, in: 2011 IEEE 29th International Conference on Computer Design (ICCD), October 2011, pp. 142–147.

[55] E. Lehtonen, J.H. Poikonen, M. Laiho, Two memristors suffice to compute all boolean functions, Electron. Lett. 46 (3) (2010) 239–240.

[56] E. Lehtonen, J.H. Poikonen, J. Tissari, M. Laiho, L. Koskinen, Recursive algorithms in memristive logic arrays, IEEE J. Emerg. Sel. Top. Circuits Syst. 5 (2) (2015) 279–292.

[57] D.B. Strukov, G.S. Snider, D.R. Stewart, R.S. Williams, The missing memristor found, Nature 453 (7191) (2008) 80.

[58] O. Catrina, A. Saxena, Secure computation with fixed-point numbers, Financ. Cryptogr. 6052 (2010) 35–50.

[59] H.J. Nussbaumer, Fast Fourier Transform and Convolution Algorithms, vol. 2, Springer Science & Business Media, 2012.

[60] R.C. Gonzalez, R.E. Woods, in: M. McDonald (Ed.), Digital Image Processing, Pearson Education, Inc, 2008.

[61] F. Alibart, L. Gao, B.D. Hoskins, D.B. Strukov, High precision tuning of state for memristive devices by adaptable variation-tolerant algorithm, Nanotechnology 23 (7) (2012).

[62] S.S. Sheu, P.C. Chiang, W.P. Lin, H.Y. Lee, P.S. Chen, Y.S. Chen, et al., A 5ns fast write multi-level non-volatile 1 K bits RRAM memory with advance write scheme, in: 2009 Symposium on VLSI Circuits, June 2009, pp. 82–83.

[63] M.-C. Wu, W.-Y. Jang, C.-H. Lin, T.-Y. Tseng, A study on low-power, nanosecond operation and multilevel bipolar resistance switching in Ti/ZrO 2 /Pt nonvolatile memory with 1T1R architecture, Semiconductor Sci. Technol. 27 (6) (May 2012).

[64] L. Zhang, D. Strukov, H. Saadeldeen, D. Fan, M. Zhang, D. Franklin, SpongeDirectory: flexible sparse directories utilizing multi-level memristors, in: Proceedings of the 23rd International Conference on Parallel Architectures and Compilation, August 2014, pp. 61–74.

[65] H.S. Malvar, L-w. He, R. Cutler, High-quality linear interpolation for demosaicing of bayer-patterned color images, in: 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 3, May 2004.

[66] A. Haj-Ali, R. Ben-Hur, N. Wald, R. Ronen, S. Kvatinsky, Not in name alone: a memristive memory processing unit for real in-memory processing, IEEE Micro 38 (5) (Sep 2018) 13–21.

[67] R. Ben-Hur, S. Kvatinsky, Memristive Memory Processing Unit (MPU) controller for in-memory processing, in: 2016 IEEE International Conference on the Science of Electrical Engineering (ICSEE), November 2016, pp. 1–5.

[68] C.R. Baugh, B.A. Wooley, A two's complement parallel array multiplication algorithm, IEEE Trans. Comput. C-22 (12) (1973) 1045–1047.

[69] S. Kvatinsky, M. Ramadan, E.G. Friedman, A. Kolodny, VTEAM: a general model for voltage-controlled memristors, IEEE Trans. Circuits Syst. II: Express Briefs 62 (8) (2015) 786–790.

[70] H.Y. Lee, Y.S. Chen, P.S. Chen, P.Y. Gu, Y.Y. Hsu, S.M. Wang, et al., Evidence and solution of Over-RESET problem for HfOX based resistive memory with sub-ns switching speed and high endurance, in: 2010 International Electron Devices Meeting, December 2010.

[71] A. Krizhevsky, G. Hinton, Learning Multiple Layers of Features from Tiny Images, Technical Report, University of Toronto, April 2009.

[72] Y. Cassuto, S. Kvatinsky, E. Yaakobi, Sneak-path constraints in memristor crossbar arrays, in: 2013 IEEE International Symposium on Information Theory, July 2013.

[73] M.A. Zidan, H.A.H. Fahmy, M.M. Hussain, K.N. Salama, Memristor-based memory: the sneak paths problem and solutions, Microelectron. J. 44 (2) (2013) 176–183.

Chapter 8

Hyperdimensional computing nanosystem: in-memory computing using monolithic 3D integration of RRAM and CNFET

Abbas Rahimi[1,2], Tony F. Wu[3], Haitong Li[3], Jan M. Rabaey[2], H.-S. Philip Wong[3], Max M. Shulaker[4] and Subhasish Mitra[3]

[1]Department of Information Technology and Electrical Engineering, ETH Zurich, Zurich, Switzerland, [2]Department of Electrical Engineering and Computer Sciences, UC Berkeley, Berkeley, CA, United States, [3]Department of Electrical Engineering, Stanford University, Stanford, CA, United States, [4]Electrical Engineering and Computer Science Department, MIT, Cambridge, MA, United States

8.1 Introduction

Over the past six decades, the semiconductor industry has been immensely successful in providing exponentially increasing computational power at an ever-reducing cost and energy footprint. Underlying this staggering evolution is a set of well-defined abstraction layers: starting from robust switching devices that support a deterministic Boolean algebra to a scalable and stored program architecture that is Turing complete and hence capable of tackling (almost) any computational challenge. Unfortunately, this abstraction chain is being challenged as scaling continues to nanometer dimensions, as well as by exciting new applications that must support a myriad of new data types. Maintaining the current deterministic computational model ultimately puts a lower bound on the energy scaling that can be obtained, set in place by fundamental physics that governs the operation, variability, and reliability of the underlying nanoscale devices [1–3].

At the same time, the nature of computing itself is evolving rapidly: for a vast number of emerging applications, cognitive functions such as classification, recognition, and learning are rapidly gaining importance. For efficient information-extraction, these applications require a fundamental departure from the traditional von Neumann architecture, where data have to be

Memristive Devices for Brain-Inspired Computing. DOI: https://doi.org/10.1016/B978-0-08-102782-0.00008-3

transported to the processing unit and back, creating the infamous memory wall. One of the most promising options for realizing such non-von Neumann architectures is to exploit beyond silicon materials and substrates that allow dense and 3D integration of memory and logic. However, such a dense and layered 3D system increases the risk of failures within the chip, and the system must be fault-tolerant. As has been realized for a long time, this resembles the way a brain computes. Hence, 3D integrated nanotechnologies combined with brain-inspired computational paradigms that support fast learning and fault tolerance could lead the way [4].

Emerging hyperdimensional (HD) computing [5] is based on the understanding that brains compute with *patterns of neural activity* that are not readily associated with scalar numbers. In fact, the brain's ability to calculate with numbers is feeble. However, due to the very size of the brain's circuits, we can model neural activity patterns with points of an HD space, that is, with hypervectors. In this formalism, information is represented in hypervectors as ultrawide words, for example, 10,000-bit. Such hypervectors can then be mathematically manipulated not only to classify but also to bind, associate, and perform other types of cognitive operations in a straightforward manner. In addition, these mathematical operations also ensure that the resulting hypervector is unique and thus the learning is one-shot or few-shot, meaning that object categories are learned from few examples in a *single pass* over the training data [6–10]. Thus, HD computing can substantially reduce the number of operations needed by conventional learning algorithms, thereby providing tremendous energy savings. Implementation of the HD computing in practical hardware needs large arrays of nonvolatile memory so that the learning is not "forgotten." A large-scale demonstration of HD computing includes 49 10,000-bit hypervectors that are encoded in 760,000 phase-change memory (PCM) devices performing analog in-memory computing [11]. Our approach is focused on potential low-voltage, nonvolatile Resistive Random Access Memory (RRAM) that can be integrated at high density with logic switches [12]. We further explore potential low-voltage approaches to logic transistors such as the carbon nanotube field-effect transistors (CNFETs) so that the overall supply voltage requirement and hence the energy dissipation can be lowered [13,14].

In this chapter, we present an HD computing nanosystem by efficient implementation of HD operations using emerging nanoscalable CNFETs and RRAM, and their monolithic 3D integration. The rest of this chapter is organized as follows. In Section 8.2, we briefly introduce HD computing and discuss some of its key properties including a well-defined set of arithmetic operations (Section 8.2.1), generality and scalability (Section 8.2.2), robustness (Section 8.2.3), and embarrassingly parallel operations (Section 8.2.4). In Section 8.3, we describe an application of HD computing in language recognition, and show how its operations can be used to solve various learning and classification tasks. In Section 8.4, we present the emerging technology for HD

computing and describe how the principal operations can be efficiently implemented in a 3D integrated architecture. Our experimental results for 3D architecture regarding robustness and energy efficiency are described in Section 8.5.

8.2 Background on HD computing

The difference between traditional computing and HD computing is apparent in the elements that the computer computes with. In traditional computing, the elements are Booleans, numbers, and memory pointers. In HD computing, they are multicomponent hypervectors, or tuples, where neither individual component nor a subset thereof has a specific meaning: a component of a hypervector and the entire hypervector represent the same. Furthermore, the hypervectors are ultrawide: the number of components is in the thousands and they are independent and identically distributed (IID).

We demonstrate the idea with a simple example from language [15,16]. The task is to identify the language of a sentence from its three-letter sequences called *trigrams*. We compare the trigram profile of a test sentence to the trigram profiles of 21 languages and choose the language with the most similar profile. A profile is essentially a histogram of trigram frequencies in the text in question.

The standard algorithm for computing the profile—the *baseline*—scans through the text and counts the trigrams. The Latin alphabet of 26 letters and the space give rise to $27^3 = 19{,}683$ possible trigrams, and so we can accumulate the trigram counts into a 19,683-dimensional vector and compare such vectors to find the language with the most similar profile. This is straightforward and simple with trigrams but it gets complicated with higher-order n-grams when the number of possible n-grams grows into the millions (the number of possible pentagrams is $27^5 = 14{,}348{,}907$). The standard algorithm generalizes poorly.

The HD algorithm starts by choosing a set of 27 letter hypervectors at random. They serve as *seed hypervectors*, and the same seeds are used with all training and test data. We have used 10,000-dimensional hypervectors of equally probable 0 s and 1 s as seeds (aka binary spatter coding [17]). From these we make trigram hypervectors by *rotating* the first letter hypervector twice, the second letter hypervector once, and use the third letter hypervector as is, and then by *multiplying* the three hypervectors component-by-component. Such trigram hypervectors resemble the seed hypervectors in that they are 10,000-D with equally probable 1 s and 0 s, and they are random relative to each other. A text's profile is then the sum of all the trigrams in the text: for each occurrence of a trigram in the text, we *add* its hypervector into the profile hypervector. The profile of a test sentence is then compared with the language profiles and the most similar one is returned as the system's answer, as above. In contrast to the standard algorithm, the HD algorithm generalizes readily to any n-gram size: the hypervectors remain 10,000-D.

8.2.1 Arithmetic operations on hypervectors

HD computing is based on the properties of hypervectors and operations on them. We will review them with reference to D-bit hypervectors, where $D = 10{,}000$ for example [18]. There are 2^D such hypervectors, also called *points*, and they correspond to the corners of a D-dimensional unit cube. The number of places at which two binary hypervectors differ is called the *Hamming distance* and it provides a measure of *similarity* between hypervectors. A peculiar property of HD spaces is that most points are relatively far from any given point. Hence, two D-bit hypervectors chosen at random are dissimilar with near certainty: when referenced from the center of the cube, they are nearly *orthogonal* to each other.

To combine hypervectors, HD computing uses three operations [5]: *addition* (which can be weighted), *multiplication*, and *permutation* (more generally, multiplication by a matrix). "Addition" and "multiplication" are meant in the abstract algebra sense where the sum of binary vectors $[A + B + \ldots]$ is defined as the componentwise majority function with ties broken at random, the product is defined as the componentwise XOR (addition modulo 2 denoted by $\oplus$), and permutation (ρ) shuffles the components. All these operations produce a D-bit hypervector, and we collectively call them Multiply-Add-Permute (MAP) operations [19].

The usefulness of HD computing comes from the nature of the MAP operations. Specifically, addition produces a hypervector that is *similar* to the argument hypervectors—the inputs—whereas multiplication and random permutation produce a *dissimilar* hypervector; multiplication and permutation are *invertible*, addition is approximately invertible; multiplication *distributes* over addition; permutation distributes over both multiplication and addition; multiplication and permutation *preserve similarity*, meaning that two similar hypervectors are mapped to equally similar hypervectors elsewhere in the space.

Operations on hypervectors can produce results that are approximate or "noisy" and need to be identified with the exact hypervectors. For that, we maintain a list of known (noise-free) seed hypervectors in a so-called *item memory* or *clean-up memory*. When presented with a noisy hypervector, the item memory outputs the most similar stored hypervector. High dimensionality is crucial to make that work reliably [18]. With 10,000-bit hypervectors, 1⁄3 of the bits can be flipped at random and the resulting hypervector can still be identified with the original stored one.

These operations make it possible to encode/decode and manipulate sets, sequences, and lists—in essence, any data structure. Such packing and unpacking operations are then viewed as mappings between points of the space suggesting a mechanism for analogy, with the analogy mapping being computed from examples. This enables implementing analogical reasoning to answer nontrivial queries, for example, "What's the Dollar of Mexico?" [20].

Fig. 8.1 shows how a data record consisting of variables x, y, z with values a, b, c can be encoded into a hypervector H and the value of x can be extracted from it. We start with randomly chosen seed hypervectors *X, Y, Z, A, B, C* for the variable and the values and store them in the item memory. We then encode the record by *binding* the variables to their values with multiplication and by adding the bound pairs:

$$H = [(X \oplus A) + (Y \oplus B) + (Z \oplus C)]$$

To find the value of x in H, we multiply it with the inverse of X, for which XOR is X itself: $A' = X \oplus H$. The resulting hypervector A' is given to the item memory which returns A as the most similar stored hypervector. An analysis of this example would show how the properties of the operations, as listed above, come in to play. One thing to note about the operations is that addition and multiplication approximate an algebraic structure called a field, to which permutation gives further expressive power.

HD computing has been described above in terms of binary hypervectors. However, the key properties are shared by hypervectors of many kinds, all of which can serve as the computational infrastructure. They include

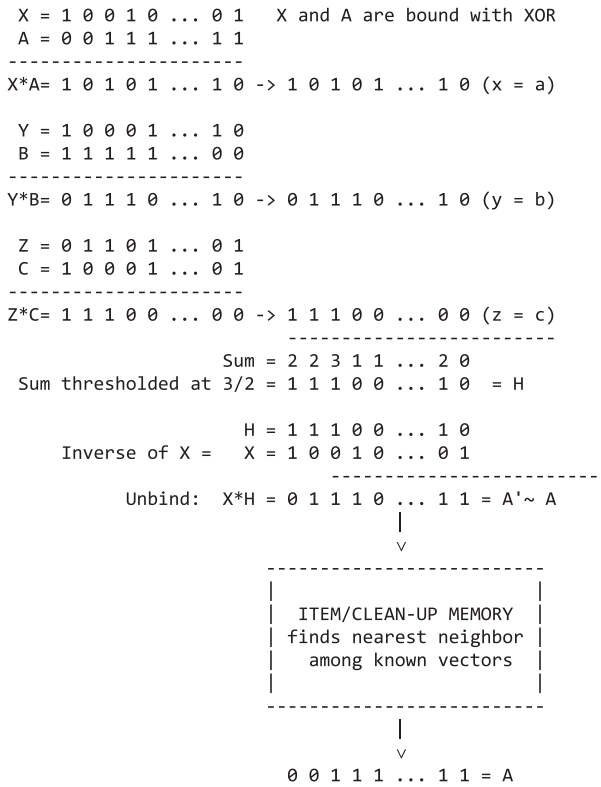


FIGURE 8.1 An example of encoding and decoding of a data structure using HD computing.

Holographic Reduced Representations (HRR) [21], Frequency-domain Holographic Reduced Representations (FHRR) [21], Binary Spatter Codes (BSC) [17], Multiply-Add-Permute (MAP) coding [19], Binary Sparse Distributed Codes (BSDC) [22], Matrix Binding of Additive Terms (MBAT) [23], and Geometric Analogue of Holographic Reduced Representations (GAHRR) [24]. Different representational schemes using high-dimensional vectors and operations on them are generally referred to as Vector Symbolic Architectures (VSA) [25] and the ultrahigh dimensionality is referred to as *hyperdimensional* [5].

8.2.2 General and scalable model of computing

HD computing is a complete computational paradigm that is easily applied to learning problems. Its main difference from other paradigms is that it can operate with data represented as approximate patterns, allowing it to scale to large learning applications. HD computing has been used commercially since 2008 for making semantic vectors for words—semantic vectors have the property that words with similar meanings are represented by similar vectors. The Random Indexing (RI) [28] algorithm for making semantic hypervectors was developed as an alternative to Latent Semantic Analysis (LSA) [29], which relies on compute-heavy Singular Value Decomposition (SVD). The original experiment used 37,000 "documents" on seven topics to compute 8000-dimensional semantic hypervectors of equal quality for 54,000 words. SDV-based LSA requires memory in proportion to the product: size of vocabulary $\times$ number of documents. By contrast RI requires memory in proportion to the size of the vocabulary, and the statistics of documents/contexts is learned through simple vector addition [28]. Thus, the complexity of the method grows linearly with the size of the training corpus and scales easily to millions of documents.

Multiplication and permutation make it possible to encode causal relations and grammar into these hypervectors, thereby capturing more and more of the meaning in language [15,30]. We have used HD computing successfully to identify the language of test sentences, as described at the beginning of this section [15,16] (also with sparse hypervectors [31]), to categorize News articles [26], and to classify DNA [32]; other applications to text include common substrings search [33] and recognition of permuted words [34]. HD computing has also been used in speech recognition [27,35]. All these applications have a single input stream (Fig. 8.2A), while HD computing provides a natural fit for applications with multiple sensory inputs, for example, biosignal processing (Fig. 8.2B). For instance, we have adapted the architecture for text analytics to the classification of hand gestures, when analog electromyography (EMG) signals are recorded simultaneously by four sensors [6,36,37] or even a larger flexible electrode array [7]. The template architecture is shown in Fig. 8.2. The architecture was further extended to

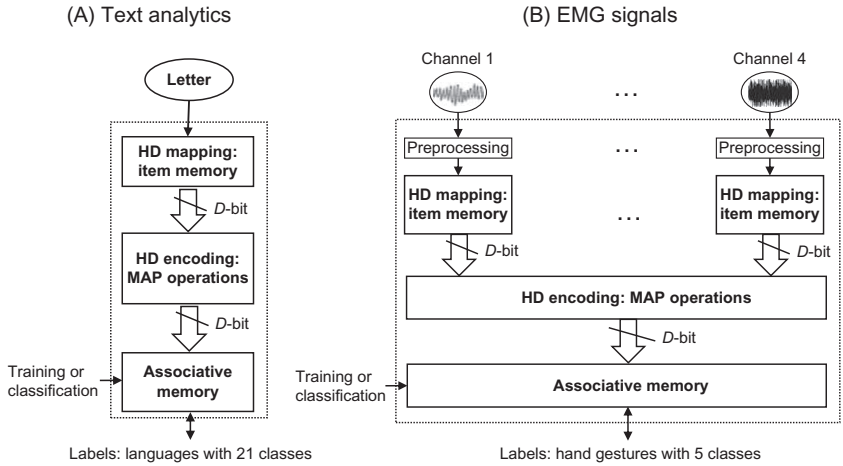


FIGURE 8.2 General and scalable HD computing for various cognitive tasks: (A) European languages recognition; (B) electromyography (EMG)-based hand gesture recognition.

TABLE 8.1 List of applications based on HD computing with different numbers of inputs and classes. The last two columns compare the classification accuracy of HD versus the baseline in that application domain.

Applications	Inputs (#)	Classes (#)	HD (%)	Baseline (%)
Language recognition [15,16]	1	21	96.7	97.9
Text categorization [26]	1	8	94.2	86.4
Speech recognition [27]	1	26	95.3	93.6
EMG gesture recognition [6]	4	5	97.8	89.7
Flexible EMG gesture recognition [7]	64	5	96.6	88.9
EEG brain–machine interface [9,10]	64	2	74.5	69.5
ECoG seizure detection [8]	100	2	95.4	94.3

ECoG, Electrocorticography; *EEG*, electroencephalography; *EMG*, electromyography.

operate on electroencephalography (EEG) [9,10] and electrocorticography (ECoG) [8] with up to 100 electrodes (See Table 8.1) .

Notably, the learning and classification tasks are performed on the same hardware construct by integrating the following three main steps: (1)

mapping to the HD space, (2) encoding with the MAP operations, and (3) associative memory (Fig. 8.2). The only difference is that during training, the associative memory updates the learned patterns with new hypervectors, while during classification it computes distances between a query hypervector and learned patterns. Hence, it is possible to build a general-purpose computational engine based on these operations to cover a variety of tasks with similar success rates. In Section 8.5, we show how such a computational engine can be efficiently realized by using emerging nanotechnologies. In addition, since the same hardware is used for learning and classification, the architecture is ideal for incremental or online learning.

8.2.3 Robustness of computations

HD computing is extremely robust. Its tolerance for low-precision and faulty components is achieved by brain-inspired properties of hypervectors: (pseudo) randomness, high-dimensionality, and fully distributed holographic representations. Symbols represented with hypervectors begin with IID components and when combined with the MAP operations, the resulting hypervectors also appear as identically distributed random hypervectors, and the independence of the individual components is mostly preserved. This means that a failure in a component of a hypervector is not "contagious." At the same time, failures in a subset of components are compensated for by the holographic nature of the data representation, that is, the error-free components can still provide a useful representation that is similar enough to the original hypervector. This inherent robustness eliminates the need for asymmetric error protection in memory units, making HD data representation suitable for operation at low signal-to-noise ratios (SNR).

8.2.4 Memory-centric with parallel operations

At its very core, HD computing is about manipulating and comparing large patterns within the memory itself. The MAP operations allow a high degree of parallelism by needing to communicate with only a local component or its immediate neighbors. Other operations such as the distance computation can be performed in a distributed fashion [38]. This is a fundamental difference from traditional computational architectures, where data have to be transported to the processing unit and back, creating the infamous memory wall. In HD processing, logic is tightly integrated with the memory and all computations are fully distributed. This translates into substantial energy savings, as global interconnects are accessed at a relatively low frequency.

8.3 Case study: language recognition

As a concrete application of HD computing, let us look at an implementation of the language recognition algorithm discussed in Section 8.2. The HD

algorithm generates trigram profiles as hypervectors and compares them for similarity. As shown in Fig. 8.3, the design is based on a memory-centric architecture where logic is tightly integrated with the memory and all computations are fully distributed. The HD architecture has two main parts: mapping and encoding modules, and a similarity search module (associative memory). The mapping and encoding module embeds an input text, composed of a stream of letters, to a hypervector in the HD space. Then, this hypervector is broadcast to the similarity search module for comparison with a set of *learned* language hypervectors. Finally, the search module returns the language that has the closest match based on Hamming distance similarity.

8.3.1 Mapping and encoding module

This module accepts the text as a stream of letters and computes its representation as a hypervector. The module has an item memory that holds a random hypervector (the "letter" hypervector) for each of the 26 letters and the space. The item memory is implemented as a lookup table that remains constant. In the dense binary coding [17], a letter hypervector has an approximately equal number of randomly placed 1 s and 0 s, hence the 27 hypervectors are approximately orthogonal to each other. As another alternative, mapping to binary hypervectors can be realized by *rematerialization* [39], for example, by using a cellular automaton exhibiting chaotic behavior [40].

The module computes a hypervector for each block of three consecutive letters as the text streams in. It consists of three stages in first in, first out style, each of which stores a letter hypervector. A trigram hypervector is created by successively permuting the letter vectors based on their order and binding them together, which creates a unique representation for each unique sequence of three letters. For example, the trigram "abc" is represented by the hypervector $\rho(\rho(A) \oplus B) \oplus C = \rho(\rho(A)) \oplus \rho(B) \oplus C$. Use of permutation and binding distinguishes the sequence "abc" from "acb", since a permuted hypervector is uncorrelated with all the other hypervectors.

The random permutation operation ρ is fixed and is implemented as a rotation to the right by 1 position as shown in Fig. 8.3. For instance, given the trigram "abc", the A hypervector is rotated twice ($\rho(\rho(A))$), the B hypervector is rotated once ($\rho(B)$), and there is no rotation for the C hypervector. Once "c" is reached, its corresponding C hypervector is fetched from the item memory and is written directly to the first stage of the encoder (i.e., Letter3 hypervector in Fig. 8.3). The two previous letters are rotated as they pass through the encoder and turn into $\rho\rho(A)$ and $\rho(B)$. Componentwise bindings (i.e., multiplication) are then applied between these three hypervectors to compute the trigram hypervector, that is, $\rho\rho(A) \oplus \rho(B) \oplus C$. Since the

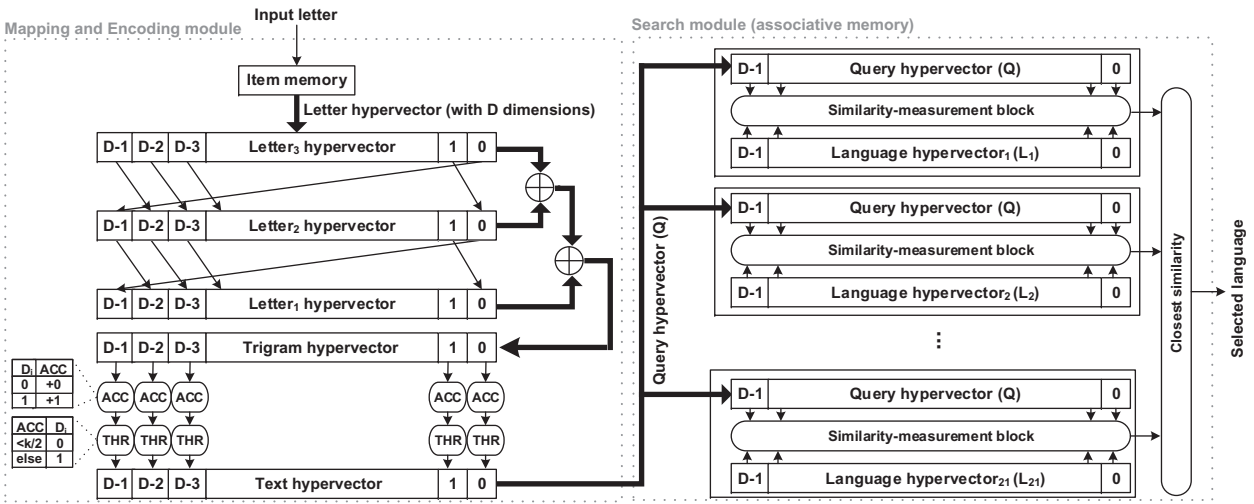


FIGURE 8.3 A 2D architecture of HD computing for language recognition task (Fig. 8.2A): mapping and encoding module, and search module.

trigram hypervector is binary, the binding between two hypervectors is implemented with D XOR gates.

The hypervector for the input text is computed by adding the hypervectors for all the trigrams in the text and by applying a threshold to retain them in the binary space. An input text of length $k + 2$ generates k trigram vectors. We implement the componentwise addition with a set of D accumulators (ACC in Fig. 8.3), one for each dimension of the hypervector, and count the number of 1 s in that component location. This componentwise accumulation produces a D-dimensional hypervector of integers. To compute the corresponding binary hypervector, the encoding module applies a threshold of $k/2$ (implementing the majority function $(k, k/2)$) to every accumulator value, where k is the number of trigrams accumulated from the input. The left side of Fig. 8.3 shows such a dedicated accumulation and thresholding for every hypervector component. The output of the module is the binary text hypervector.

The mapping and encoding module is used for both training and testing. During training when the language of the input text is known, we refer to the text hypervector as a *language* hypervector. Such language hypervectors are stored in the search module as learned patterns. When the language of a text is unknown, as it is during testing, we call the text hypervector a *query* hypervector. The query hypervector is sent to the similarity search module to identify its source language.

8.3.2 Similarity search module

The search module stores a set of language hypervectors that are precomputed by the mapping and encoding module. These language hypervectors are formed in exactly the same way as described above, by making the text hypervectors from samples of a known language. Therefore during the training phase, we feed texts of a known language to the mapping and encoding module and save the resulting text hypervector as a language hypervector in the search module. We consider 21 European languages and at the end of training have 21 language hypervectors, each stored in its own row of the search module.

The language of an unknown text is determined by comparing its query hypervector to all the language hypervectors. This comparison is done in a distributed fashion using an associative memory, and with the Hamming distance as the similarity function. Hamming distance counts the number of components at which two binary hypervectors disagree. The module uses a set of D XOR gates to identify mismatches between the two hypervectors. In this digital implementation, the similarity-measurement block compares only one component each clock cycle. Hence it takes $O(D)$ cycles to compute the Hamming distance between the two hypervectors [16]. This block is replicated 21 times (the number of languages in our application) within the search

module as shown in Fig. 8.3. The query hypervector is broadcast across the search module, hence all the similarity-measurement blocks compute their distance concurrently. Finally, a combinational comparison block selects the minimum Hamming distance and returns its associated language as the language that the unknown text has been written in.

8.4 Emerging technologies for HD computing

Several emerging nanotechnologies such as carbon nanotube field-effect transistors, resistive RAM, and monolithic 3D integration have been demonstrated to be particularly effective for implementation of HD computing as well as other computing paradigms. One or more of these technologies has been used in demonstrating HD computing operations [12] as well as in full system demonstrations [13,14]. These technologies are detailed in the following sections.

8.4.1 Carbon nanotube field-effect transistors

Carbon nanotube field-effect transistors (CNFETs) are an emerging transistor technology which promise an order of magnitude improvement in energy-delay-product (a metric of energy efficiency) for digital circuits [41]. CNFETs use multiple carbon nanotubes (CNTs), which are cylindrical structures of carbon atoms 1−2 nm in diameter, that act as channels. CNTs enable highly energy-efficient digital logic circuits due to their high carrier mobility and excellent electrostatic control in CNFETs [42]. High-performance complementary logic has been demonstrated using CNFETs with an I_{on}/I_{off} ratio (i.e., the ratio of drive current to the off-state leakage current) of about 10^6 [43,44]. They can be built at scaled gate lengths (5 nm) [45] and without hysteresis [46]. CNFETs have even been fabricated as negative capacitance FETs with sub-55 mV/decade subthreshold swing at room temperature [47]. CNFETs can be fabricated at low temperature ($\leq$250°C), which is key to enabling monolithic 3D integration (discussed later in this chapter).

When designing circuits with CNFETs, imperfections inherent in CNTs, such as mis-positioned CNTs (that can lead to stray conducting paths resulting in incorrect functionality) and metallic CNTs (i.e., CNTs with little or no bandgap), can be overcome using the imperfection-immune paradigm. The imperfection-immune paradigm uses a combination of fabrication and design techniques [48−50] to enable wafer-scale fabrication and VLSI-compatible design of CNFET circuits. This paradigm has enabled experimental demonstrations such as the first CNT computer, the first 3D nanosystem consisting of over 2 million CNFETs on a single die, and the first full-system demonstration of an HD computing nanosystem [13,51,52].

In addition to process variations that exist in silicon transistors (e.g., variations in threshold voltage, channel length, and oxide thickness), CNFETs

are subject to CNT-specific variations such as CNT count variations (i.e., variations in the number of CNTs in a CNFET). These variations cause drive current variations, which can manifest as delay variations in digital circuits. These variations can be suppressed using optimized process and circuit design. However, these inherent variations can be utilized in HD computing to generate the seed hypervectors, as demonstrated in Ref. [13] by capitalizing on the variations in CNT count and threshold voltage. This means that variability and randomness essentially become the sources for computation.

8.4.2 Resistive RAM

Resistive RAM (RRAM) is an emerging memory technology that promises high capacity, nonvolatile data storage (10-year retention), and can be fabricated at low temperature ($\leq$250°C) [53,54]. RRAM is fabricated as a metal oxide switching layer (insulator) sandwiched between two metallic electrodes and can be realized using various metal−insulator−metal material combinations.

Three main operations are typically performed on an RRAM cell: set, reset, and read. The set operation transforms the cell from high-resistance state (HRS) to low-resistance state (LRS) by applying a positive voltage (i.e., set voltage) across the top and bottom electrodes [53]. A transistor is typically used to limit the current for the set operation (called the compliance current). This creates or lengthens a filament of oxygen vacancies from the bottom electrode to the top electrode. As the length of the conductive filament increases, the resistance of the RRAM decreases [53]. In most cases a higher set voltage (called forming voltage) is applied to form the filament (once) after fabrication. However, forming-less RRAM cells have also been demonstrated [53,55]. A reset operation transforms the cell from LRS to HRS by applying a negative voltage (i.e., reset voltage) across the top and bottom electrodes, rupturing the filaments between the electrodes. RRAM cells with ≤ 2 V set/reset voltage (≈ 10 ns pulse duration) and 10−100 HRS/LRS resistance ratio have been demonstrated [53,56]. RRAM is also subject to variations in its resistance, stemming from the stochastic size and shape of the conductive filament after a set or reset operation. These variations can also be exploited to generate seed hypervectors as discussed in Section 5.1. A read operation detects the state of the cell (e.g., HRS or LRS) by sensing the current after applying a small voltage across the two electrodes. This voltage is small enough to not change the resistance of the cell. Although RRAM has limited write (i.e., set/reset) endurance (10^{12} cycles at the cell level [56] and 10^5−10^7 cycles at the array level [57,58]), HD computing is shown to be robust against such endurance-related errors (Section 5.1).

Many cell structures (e.g., 1 transistor-1 RRAM cell, 1 transistor-n RRAM cell, 1 selector-1 RRAM cell [53]) may be used for RRAM, with

each structure exhibiting a trade-off between cell density (i.e., the number of cells that can be placed in a given area) and the controllability (of the resistance during set or reset operations) or the ability to detect the state of the cell reliably. For example, the one transistor-1 RRAM (1T-1R) cell configuration can be effectively used to prevent current overshoot during the set operation and provide exceptional selectivity between cells during the read operation but has limited cell density due to each cell using a transistor (typically larger than the RRAM cell itself) [53]. Array-level implementations using the 1T-1R RRAM structure have been demonstrated up to 16 Gbits of capacity [59]. Moreover, RRAM can be vertically built (3D VRRAM) in a bit-cost scalable manner to improve the cell density [60]. HD computing demonstrations have used both the 1T-1R [13] configuration and 3D VRRAM [12].

A single RRAM cell can store a single bit or multiple bits [61]. To demonstrate multibit storage in RRAM cells, one or a combination of set or reset parameters are adjusted to change the resistance of the cell to an intermediate value (between LRS and HRS): compliance current in the set operation, reset voltage, and set or reset pulse duration. These parameters can also be adjusted to gradually increase the RRAM cell resistance (i.e., increasing the resistance incrementally). This gradual increase in RRAM cell resistance has been demonstrated for a variety of switching layers (i.e., the material in which the filament forms, such as HfO_X [62]) by using short pulses during the reset operation. This behavior, called *gradual reset*, can be employed to realize addition operations in hardware [13].

The RRAM has been demonstrated as digital storage and as incrementers using gradual reset (i.e., the ability to increment the RRAM resistance in a fine-grained manner), and for performing the bitwise operations necessary for HD computing [12,13].

8.4.3 Monolithic 3D integration

Monolithic 3D integration is a process whereby tiers of circuits (i.e., a layer of logic, memory, or sensors) are fabricated on top of each other on the same substrate. Monolithic 3D integration uses interlayer vias (ILVs), standard vias used to connect adjacent metal layers in the interconnect stack of today's silicon CMOS technologies, to connect between tiers of circuits. This is in contrast to chip stacking using through-silicon vias (TSVs) with typical pitches of around 10 μm [63]. The ILVs can have the same pitch as metal interconnects (100 nm at the 28 nm technology node [64]), enabling significantly denser vertical connectivity compared to TSVs [65]—a key to tight integration between logic and memory.

Monolithic 3D integration requires low temperature fabrication for upper tiers of circuits ($\leq$400°C) as higher temperatures can damage existing circuits (transistors and interconnects) on the bottom tiers. While this is difficult

for traditional silicon CMOS technologies (e.g., high temperature requirements for dopant activation $\geq 1000^{\circ}C$), it is naturally enabled by CNFETs and RRAM due to their low temperature fabrication [53,66]. In recent demonstrations of HD computing, all CNFETs and RRAM were fabricated with a maximum temperature of 200°C. Monolithic 3D integration of CNFETs, RRAM, and silicon transistors has been shown [43], demonstrating compatibility with silicon CMOS. HD computing has been shown to provide up to 35× energy-execution time product benefits when implemented using such monolithically integrated CNFETs and RRAM compared with the standard silicon CMOS approach [14].

8.5 Experimental demonstrations for HD computing

In this section we describe several experimental hardware demonstrations for HD computing, including in-memory MAP kernels using 3D VRRAM [12] and an end-to-end HD system with CNFETs and RRAM using monolithic 3D integration [13,14]. Electrical characterization results and system simulations are discussed to provide insights into how emerging device technologies can be utilized toward natural and efficient implementations of HD computing systems.

8.5.1 3D VRRAM demonstration: in-memory MAP kernels

One novel approach to serve memory-intensive MAP operations for HD computing is to directly construct a set of native MAP kernels within dense memories without moving data around. 3D vertical RRAM (VRRAM), built in a multilayer vertical architecture, has been demonstrated naturally suitable for native MAP kernel implementations toward an in-memory, nonvolatile HD system (Fig. 8.4) [12]. In this section we discuss in-memory computing opportunities associated with 3D VRRAM for HD computing, from data representation to manipulation.

First, HD data representation requires randomness in seed hypervectors while maintaining long-term stability. Targeting RRAM-centric HD systems, binary vectors should be initialized and stored in RRAM cells using resistance-based representation in a nonvolatile fashion. To meet this requirement inherent stochasticity of RRAM is employed. The fundamental switching mechanisms of resistive memories lead to statistical behaviors during programming. This phenomenon is typically characterized by physical resistances and voltage distributions under memory or storage use cases. When RRAM is operated at voltages below nominal SET/RESET voltages, stochastic switching in binary states emerges and can be modulated by pulse programming conditions (pulse width and pulse amplitude). Specifically, a higher probability of successful SET operation can be attained from increasing pulse amplitude or using longer pulses. This corresponds to a nonlinear

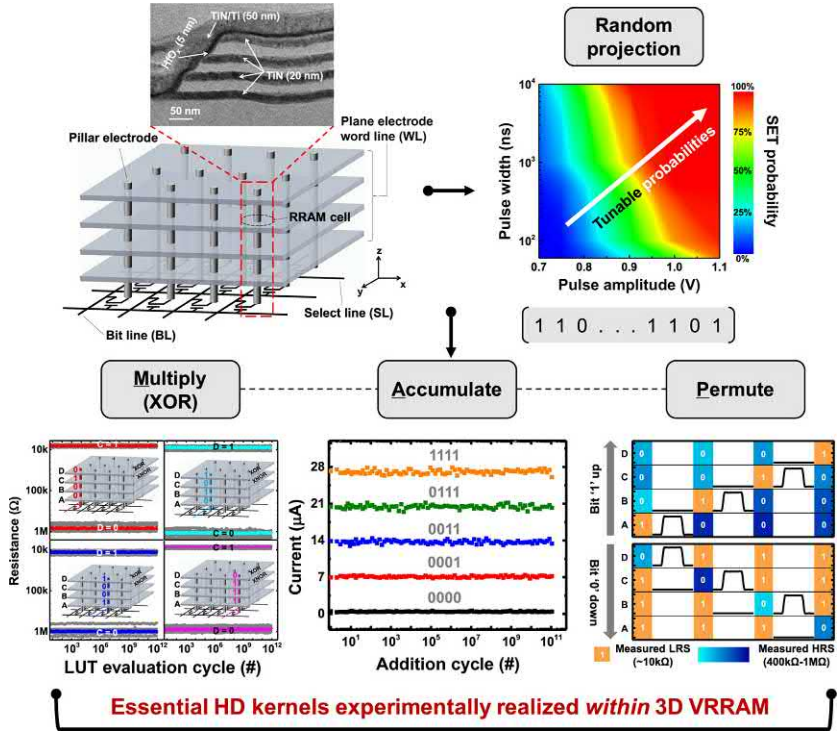


FIGURE 8.4 In-memory MAP kernels experimentally realized with 3D VRRAM, utilizing unique device-level and circuit-level characteristics including nonvolatility and 3D connectivity.

voltage−time relationship of RRAM stochastic switching in the below-threshold regime. Harnessing this stochastic behavior leads to an observation that seed hypervectors can be produced in situ with a 50% probability of switching from HRS (representing "0") to LRS (representing '1'). Standby power or refresh operations are not required due to 10-year retention of RRAM for the following HD-related data manipulation in memories.

After the initial mapping and random projection are realized directly in memories, the MAP kernels can be implemented by exploiting the circuit-level properties of the vertical 3D architecture, where multiple layers of RRAM cells share common vertical pillars connected to select transistors underneath [67]. Fig. 8.4 shows key experimental results for the essential operations using four-layer 3D VRRAM.

For multiply or equivalently bitwise XOR operations, we leverage the voltage dividers formed by the RRAM cells and select transistors underneath to construct nonvolatile XOR/XNOR look-up tables within 3D VRRAM. This architecture design allows us to perform only a few initial write operations for creating the look-up tables, while subsequent operations are read-

only, without the need of reprogramming RRAM. Specifically, the voltage divider structure in 3D leverages the interaction between RRAM layers and the select transistor (which can be operated in linear region or turned off). This interplay enables the resistance state of RRAM in one layer to impact the pulse programming results of RRAM cells in other layers, since HRS or LRS will create different voltage dividing scenarios along the common vertical pillar shared by different layers of RRAM and their select transistor underneath. The pulse programming scheme uses three voltages: full VDD, half VDD, and GND, which allows arbitrary Boolean functions to be programmed. For example, in the case of XOR in the four-layer 3D VRRAM structure, the lower two layers of RRAM function as input storage whereas the upper two layers serve as output storage. Following an XOR-specific pulse train programming, the upper two layers of RRAM store complete XOR/XNOR outputs, given different input combinations in the lower two layers. In this look-up table configuration, evaluation requires a simple decode-and-read operation to fetch the XOR results. In our electrical measurements up to 10^{12} XOR evaluation cycles are performed by pulse operations on four-layer 3D VRRAM, showing no disturb error in readout results (Fig. 8.4).

The 3D vertical connectivity also enables analog current summing among RRAM cells that share a vertical pillar. Thus, an accumulate operation can be performed by parallel read operations across layers. Total current along a vertical pillar comes from RRAM cells around the common pillar structure, modulated by specific HRS or LRS of each bit. We measure various 4-bit vectors that are initialized in four-layer 3D VRRAM, and obtain distinct and correct accumulated results with up to 10^{11} pulse operations showing no disturb error. Permute operations simply shift bits within a hypervector, which are realized through bit copy operations within 3D VRRAM. Building upon the methodology that the vertical 3D connectivity enables tight correlation among layers, when multiple RRAM cells share a common pillar electrode, programming of one cell can be intentionally modulated by resistance states of another cell. Here, similarly, the voltage divider scenario exhibits among RRAM cells. Hence, shifting "1" or "0" up and down in arbitrary orders can be realized through pairs of VDD/GND pulses that involve SET or RESET operations. However, this in-memory permute operation does not involve extra or separate read-out and write-back procedure.

Since using nonvolatile memory cells for both HD data representation and MAP operations involves write operations, endurance constraint (total number of write cycles before a hard error is produced) is also evaluated by conducting simulations on the language recognition task. During the training and inference on the task dataset, endurance failures (stuck at "1" or "0") may occur on RRAM cells. Under different levels of endurance characteristics at the device level, simulations show that a certain degree of robustness can be retained (Fig. 8.5), owing to the robust HD representation. In

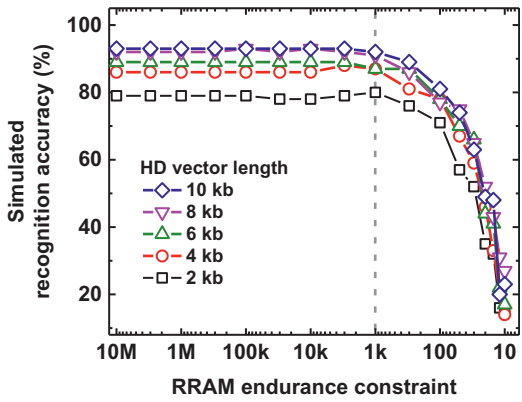


FIGURE 8.5 Evaluation of the interplay between HD representations and RRAM endurance constraints (stuck-at errors in memories).

summary, algorithm-level (e.g., error resilience in HD representations) and technology-level characteristics (stochasticity and 3D connectivity) are exploited for an in-memory, nonvolatile HD hardware demonstration.

8.5.2 System demonstration using monolithic 3D integrated CNFETs and RRAM

HD computing can be realized in hardware using monolithic 3D integration of CNFETs and RRAM [13,14] and has demonstrated pairwise classification of 21 languages with measured mean accuracy of up to 98% on $>20{,}000$ sentences. Unique properties of RRAM and CNFETs can be exploited to create area- and energy-efficient monolithic 3D circuit blocks that combine CNFETs with fine-grained access to RRAM memories (Fig. 8.6):

1. Circuits that embrace inherent variations in RRAM and CNFETs to generate hypervectors, with estimated $3\times$ lower dynamic energy (vs silicon CMOS implementations at the same technology node) stemming from both the circuit topology and the use of energy-efficient CNFETs.
2. Approximate incrementer circuits using gradual RRAM reset operations for HD accumulation use $30\times$ fewer transistors versus full-digital incrementer implementations.
3. Ternary content-addressable memory (TCAM) cells built using pairs of CNFETs and RRAM for computing Hamming distance use $19\times$ lower energy (simulated vs SRAM-based TCAM cells) due to reduced leakage of nonvolatile RRAM.

When such a system is implemented at smaller technology nodes (e.g., 28 nm node), it can simultaneously achieve lower energy and faster execution time compared to conventional silicon CMOS approaches (e.g., $7.6\times$ lower energy and $4.6\times$ faster execution time).

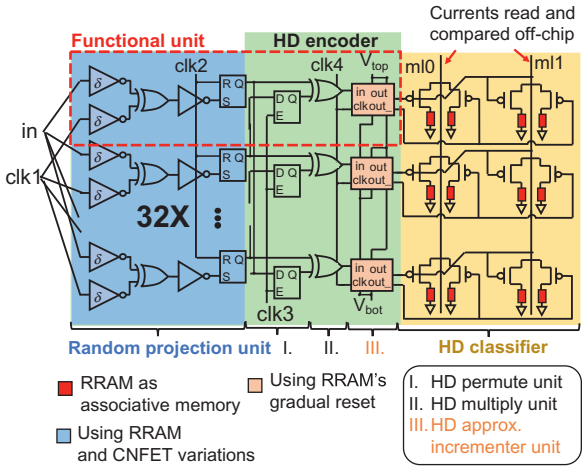


FIGURE 8.6 Schematic of Monolithic 3D HD system using CNFETs and RRAM.

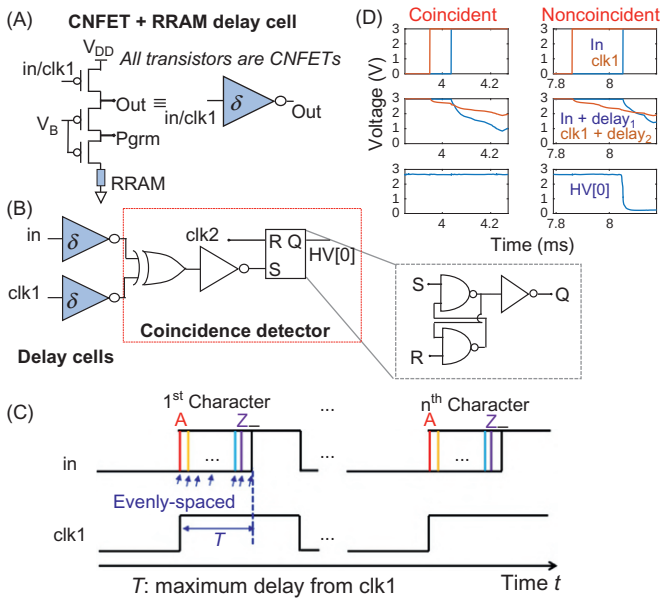


FIGURE 8.7 Delay cells exploiting inherent variations of CNFETs and RRAM.

To realize an item memory, with randomly generated seeds, to map input letters to hypervectors, inherent variations in RRAM and CNFETs can be exploited (Fig. 8.7). Delay cells can be used to translate device-level variations such as drive current variations resulting from variations in carbon nanotube (CNT) count (i.e., the number of CNTs in a CNFET) or threshold

voltage of CNFETs, and resistance variations of RRAM to delay variations. To generate hypervectors, each possible input (26 letters of the alphabet and the space character) is mapped to a delay from a reference clock edge (time-encoded). To calculate each bit of the hypervector, random delays are added to both the input signal and the reference clock using delay cells. If the resulting signals are coincident (the falling edges are close enough to set an SR latch), the output is '1'. Before training the system, to initialize delay cells, the RRAM resistance is first reset to HRS and then set to LRS.

To realize circuits to perform addition in hardware, an approximate incrementer with thresholding which leverages the multiple values of RRAM resistance that can be programmed by performing a gradual reset is used (Fig. 8.8). A digital buffer is used to transform (threshold) the sum to a binary hypervector. Each such approximate incrementer uses eight transistors and a single RRAM cell. In contrast, a digital 7-bit incrementer may use 240 transistors. Thus, when D (e.g., 10,000) accumulators are needed, the savings can be significant.

The search module (i.e., Hamming distance calculation) is implemented using 2T2R (2-CNFET transistor, 2-RRAM) ternary content-addressable memory (TCAM) cells to form an associative memory (Fig. 8.6) with each cell performing an XOR inside the memory. The TCAM cells perform an HD multiplication operation on the stored data and a query vector with the output on the match line (i.e., ml0 or ml1). During training, the match line corresponding to the language (e.g., ml0 for English and ml1 for Spanish) is set to a high voltage to write to the RRAM cells (e.g., 3 V), writing the query hypervector into the RRAM cells connected to the match line. During

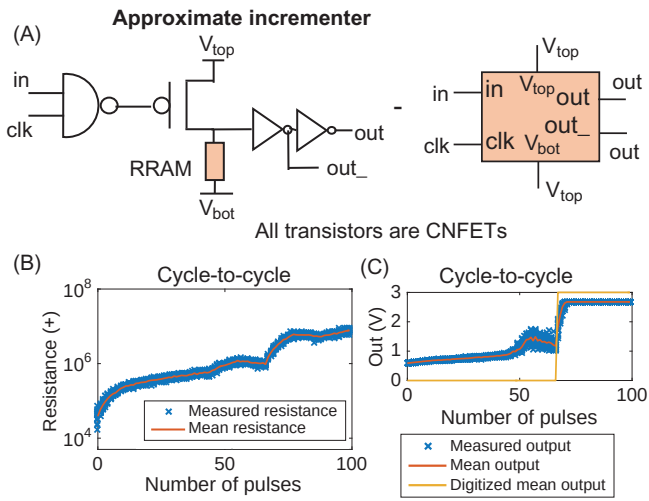


FIGURE 8.8 Approximate incrementer using the gradual reset property of RRAM. All transistors are CNFETs.

inference the match lines (i.e., ml0 and ml1) are set to a low voltage (e.g. 0.5 V), and the current on each match line is read as an output. When the query hypervector bit is equal to the value stored in a TCAM cell (match), the current is high. Otherwise (mismatch) the current is low. Cell currents are summed on each match line (i.e., the number of mismatches is counted to produce the Hamming distance). The line with the most current (i.e., smallest Hamming distance) corresponds to the output class.

8.6 Conclusion

The conventional von Neumann model of computing is deterministic, and the engineering and manufacturing effort to make computer circuits reliable is considerable. It is also costly in material and energy. By contrast HD computing uses randomness constructively and tolerates variation and errors in many of the components. Several experiments in this chapter illustrate how various properties of heterogeneous nanotechnologies can be effectively exploited and combined to realize brain-inspired HD computing architectures by tightly integrating computation and storage, and by embracing randomness.

References

[1] W. Dehaene, Sc1: Circuit design in advanced CMOS technologies: how to design with lower supply voltages, in: 2015 IEEE International Solid-State Circuits Conference – (ISSCC) Digest of Technical Papers, 2015, pp. 1–2. <https://doi.org/10.1109/ISSCC.2015.7063154>.

[2] S. Williams, E.P. DeBenedictis, OSTP Nanotechnology-Inspired Grand Challenge: Sensible Machines (Extended Version 2.5) (October 20, 2015).

[3] A. Rahimi, L. Benini, R.K. Gupta, Variability mitigation in nanometer CMOS integrated systems: a survey of techniques from circuits to software, Proc IEEE. 104 (7) (2016) 1410–1448. Available from: https://doi.org/10.1109/JPROC.2016.2518864.

[4] A. Rahimi, S. Datta, D. Kleyko, E.P. Frady, B. Olshausen, P. Kanerva, et al., High-dimensional computing as a nanoscalable paradigm, IEEE Transact. Circuits Systems I Regular Papers. 64 (9) (2017) 2508–2521. Available from: https://doi.org/10.1109/TCSI.2017.2705051.

[5] P. Kanerva, Hyperdimensional computing: an introduction to computing in distributed representation with high-dimensional random vectors, Cognitive Computation 1 (2) (2009) 139–159. Available from: https://doi.org/10.1007/s12559-009-9009-8. Available from: http://dx.doi.org/10.1007/s12559-009-9009-8.

[6] A. Rahimi, S. Benatti, P. Kanerva, L. Benini, J.M. Rabaey, Hyperdimensional biosignal processing: a case study for EMG-based hand gesture recognition, in: IEEE International Conference on Rebooting Computing, IEEE, San Diego, CA, 2016.

[7] A. Moin, A. Zhou, A. Rahimi, S. Benatti, A. Menon, S. Tamakloe, et al., An EMG gesture recognition system with flexible high-density sensors and brain-inspired high-dimensional classifier, in: IEEE International Symposium on Circuits and Systems, ISCAS, IEEE, Florence, Italy, 2018.

[8] L.B.A. Burrello, K. Schindler, A. Rahimi, One-shot learning for iEEG seizure detection using end-to-end binary operations: local binary patterns with hyperdimensional computing, in: IEEE Biomedical Circuits and Systems Conference (BioCAS), IEEE, Cleveland, OH, 2018.

[9] A. Rahimi, P. Kanerva, J.d.R. Millán, J.M. Rabaey, Hyperdimensional computing for non-invasive brain−computer interfaces: blind and one-shot classification of EEG error-related potentials, in: 10th ACM/EAI International Conference on Bio-Inspired Information and Communications Technologies (BICT), 2017.

[10] A. Rahimi, A. Tchouprina, P. Kanerva, J.d.R. Millán, J.M. Rabaey, Hyperdimensional computing for blind and one-shot classification of EEG error-related potentials, Mobile Networks Appl. (Oct 2017) <https://doi.org/10.1007/s11036-017-0942-6> and <https://doi.org/10.1007/s11036-017-0942-6>.

[11] G. Karunaratne, M.L. Gallo, G. Cherubini, L. Benini, A. Rahimi, A. Sebastian, In-memory hyperdimensional computing, Nature Electronics (2020). Available from: https://doi.org/10.1038/s41928-020-0410-3.

[12] H. Li, T.F. Wu, A. Rahimi, K.S. Li, M. Rusch, C.H. Lin, et al., Hyperdimensional computing with 3d vrram in-memory kernels: device-architecture co-design for energy-efficient, error-resilient language recognition, in: 2016 IEEE International Electron Devices Meeting (IEDM), 2016, pp. 16.1.1−16.1.4. <https://doi.org/10.1109/IEDM.2016.7838428>.

[13] T.F. Wu, H. Li, P. Huang, A. Rahimi, J.M. Rabaey, H.P. Wong, et al., Brain-inspired computing exploiting carbon nanotube fets and resistive ram: Hyperdimensional computing case study, in: 2018 IEEE International Solid State Circuits Conference (ISSCC), 2018, pp. 492−494. <https://doi.org/10.1109/ISSCC.2018.8310399>.

[14] T.F. Wu, H. Li, P. Huang, A. Rahimi, G. Hills, B. Hodson, et al., Hyperdimensional computing exploiting carbon nanotube fets, resistive ram, and their monolithic 3d integration, IEEE J Solid-State Circuits (2018) 1−14. Available from: https://doi.org/10.1109/JSSC.2018.2870560.

[15] A. Joshi, J.T. Halseth, P. Kanerva, Language geometry using random indexing, in: J.A. de Barros, B. Coecke, E. Pothos (Eds.), Quantum Interaction: 10th International Conference, QI 2016, San Francisco, CA, July 20−22, 2016, Revised Selected Papers, Springer International Publishing, Cham, 2017, pp. 265−274. <https://doi.org/10.1007/978-3-319-52289-0_21> and <https://doi.org/10.1007/978-3-319-52289-0_21>.

[16] A. Rahimi, P. Kanerva, J.M. Rabaey, A robust and energy efficient classifier using brain-inspired hyperdimensional computing, in: Low Power Electronics and Design (ISLPED), 2016 IEEE/ACM International Symposium on, 2016.

[17] P. Kanerva, Binary spatter-coding of ordered k-tuples, in ICANN'96, Proceedings of the International Conference on Artificial Neural Networks, Vol. 1112 of Lecture Notes in Computer Science, Springer, 1996, pp. 869−873.

[18] P. Kanerva, Sparse Distributed Memory, The MIT Press, Cambridge, MA, 1988.

[19] R.W. Gayler, Multiplicative binding, representation operators and analogy, in: D. Gentner, K.J. Holyoak, B.N. Kokinov (Eds.), Advances in Analogy Research: Integration of Theory and Data From the Cognitive, Computational, and Neural Sciences, New Bulgarian University, Sofia, Bulgaria, 1998, pp. 1−4. Available from: http://cogprints.org/502/.

[20] P. Kanerva, What we mean when we say "what's the dollar of mexico?": Prototypes and mapping in concept space, in: AAAI Fall Symposium: Quantum Informatics for Cognitive, Social, and Semantic Processes, 2010, pp. 2−6.

[21] T. Plate, Holographic Reduced Representations, CLSI Publications, 2003.

[22] D.A. Rachkovskij, Representation and processing of structures with binary sparse distributed codes, IEEE Trans. Knowledge Data Eng. 3 (2) (2001) 261–276.
[23] S.I. Gallant, T.W. Okaywe, Representing objects, relations, and sequences, Neural Comput. 25 (8) (2013) 2038–2078.
[24] D. Aerts, M. Czachor, B. De Moor, Geometric analogue of holographic reduced representation, J. Math. Psychol. 53 (2009) 389–398.
[25] R.W. Gayler, Vector symbolic architectures answer jackendoff's challenges for cognitive neuroscience, in: Proceedings of the Joint International Conference on Cognitive Science. ICCS/ASCS, 2003, pp. 133–138.
[26] F.R. Najafabadi, A. Rahimi, P. Kanerva, J.M. Rabaey, Hyperdimensional computing for text classification, in: Design, Automation Test in Europe Conference Exhibition (DATE), University Booth, 2016. <https://www.date-conference.com/system/files/file/date16/ubooth/37923.pdf>.
[27] M. Imani, D. Kong, A. Rahimi, T. Rosing, Voicehd: hyperdimensional computing for efficient speech recognition, in 2017 IEEE International Conference on Rebooting Computing (ICRC), 2017, pp. 1–8 <https://doi.org/10.1109/ICRC.2017.8123650>.
[28] P. Kanerva, J. Kristoferson, A. Holst, Random indexing of text samples for latent semantic analysis, in Proceedings of the 22nd Annual Conference of the Cognitive Science Society, Erlbaum, 2000, p. 1036. <http://www.rni.org/kanerva/cogsci2k-poster.txt>.
[29] T.K. Landauer, S.T. Dutnais, A solution to plato's problem: the latent semantic analysis theory of acquisition, induction, and representation of knowledge, Psychol. Rev. 104 (2) (1997) 211–240.
[30] G. Recchia, M. Sahlgren, P.K.M. Jones, Encoding sequential information in semantic space models. Comparing holographic reduced representation and random permutation, Comput. Intelligence Neurosci. (2015) 1–18.
[31] M. Imani, J. Hwang, T. Rosing, A. Rahimi, J.M. Rabaey, Low-power sparse hyperdimensional encoder for language recognition, IEEE Design Test 34 (6) (2017) 94–101. Available from: https://doi.org/10.1109/MDAT.2017.2740839.
[32] M. Imani, T. Nassar, A. Rahimi, T. Rosing, Hdna: energy-efficient DNA sequencing using hyperdimensional computing, in: 2018 IEEE EMBS International Conference on Biomedical Health Informatics (BHI), 2018, pp. 271–274 <https://doi.org/10.1109/BHI.2018.8333421>.
[33] D. Kleyko, E. Osipov, On bidirectional transitions between localist and distributed representations: the case of common substrings search using vector symbolic architecture, Procedia Comput. Sci. 41 (2014) 104–113.
[34] D. Kleyko, E. Osipov, R.W. Gayler, Recognizing permuted words with vector symbolic architectures: a Cambridge test for machines, Procedia Comput. Sci. 88 (2016) 169–175.
[35] O. Rasanen, Generating hyperdimensional distributed representations from continuous valued multivariate sensory input, in: Proceedings of the 37th Annual Meeting of the Cognitive Science Society, 2015, pp. 1943–1948.
[36] F. Montagna, A. Rahimi, S. Benatti, D. Rossi, L. Benini, Pulp-hd: accelerating brain-inspired high-dimensional computing on a parallel ultra-low power platform, in: Proceedings of the 55th Annual Design Automation Conference, DAC '18, ACM, New York, NY, 2018, pp. 111:1–111:6. <https://doi.org/10.1145/3195970.3196096>.
[37] D. Kleyko, A. Rahimi, D.A. Rachkovskij, E. Osipov, J.M. Rabaey, Classification and recall with binary hyperdimensional computing: tradeoffs in choice of density and mapping characteristics, IEEE Trans. Neural Networks Learning Systems (2018) 1–19. Available from: https://doi.org/10.1109/TNNLS.2018.2814400.

[38] M. Imani, A. Rahimi, D. Kong, T. Rosing, J.M. Rabaey, Exploring hyperdimensional associative memory, in: 2017 IEEE International Symposium on High Performance Computer Architecture (HPCA), 2017, pp. 445–456. <https://doi.org/10.1109/HPCA.2017.28>.

[39] M. Schmuck, L. Benini, A. Rahimi, Hardware Optimizations of Dense Binary Hyperdimensional Computing: Rematerialization of Hypervectors, Binarized Bundling, and Combinational Associative Memory, ACM J. Emerg. Technol. Comput. Syst. 15 (4) (2019) 25. Available from: https://doi.org/10.1145/3314326.

[40] O. Yilmaz, Symbolic computation using cellular automata-based hyperdimensional computing, Neural Comput. 27 (12) (2015) 2661–2692. pMID: 26496041. arXiv: https://doi.org/10.1162/NECO_a_00787.

[41] L. Chang, Iedm short course, 2012.

[42] J. Appenzeller, Carbon nanotubes for high-performance electronics—progress and prospect, Proc. IEEE 96 (2) (2008) 201–211. Available from: https://doi.org/10.1109/JPROC.2007.911051.

[43] M.M. Shulaker, G. Pitner, G. Hills, M. Giachino, H.P. Wong, S. Mitra, High-performance carbon nanotube field-effect transistors, in: 2014 IEEE International Electron Devices Meeting, 2014, pp. 33.6.1–33.6.4. <https://doi.org/10.1109/IEDM.2014.7047164>.

[44] Y. Yang, L. Ding, J. Han, Z. Zhang, L.-M. Peng, High-performance complementary transistors and medium-scale integrated circuits based on carbon nanotube thin films, ACS Nano 11 (4) (2017) 4124–4132. pMID: 28333433. arXiv:https://doi.org/10.1021/acsnano.7b00861.

[45] C. Qiu, Z. Zhang, M. Xiao, Y. Yang, D. Zhong, L.-M. Peng, Scaling carbon nanotube complementary transistors to 5-nm gate lengths, Science 355 (6322) (2017) 271–276. Available from: https://doi.org/10.1126/science.aaj1628. arXiv: http://science.sciencemag.org/content/355/6322/271.full.pdf..

[46] R.S. Park, G. Hills, J. Sohn, S. Mitra, M.M. Shulaker, H.-S.P. Wong, Hysteresis-free carbon nanotube field-effect transistors, ACS Nano 11 (5) (2017) 4785–4791. pMID: 28463503. arXiv:https://doi.org/10.1021/acsnano.7b01164.

[47] T. Srimani, G. Hills, M.D. Bishop, U. Radhakrishna, A. Zubair, R.S. Park, et al., Negative capacitance carbon nanotube fets, IEEE Electron Device Lett. 39 (2) (2018) 304–307. Available from: https://doi.org/10.1109/LED.2017.2781901.

[48] J. Zhang, A. Lin, N. Patil, H. Wei, L. Wei, H.P. Wong, et al., Carbon nanotube robust digital vlsi, IEEE Trans. Computer-Aided Design Integrated Circ Syst. 31 (4) (2012) 453–471. Available from: https://doi.org/10.1109/TCAD.2012.2187527.

[49] M.M. Shulaker, G. Hills, T.F. Wu, Z. Bao, H.P. Wong, S. Mitra, Efficient metallic carbon nanotube removal for highly-scaled technologies, in: 2015 IEEE International Electron Devices Meeting (IEDM), 2015, pp. 32.4.1–32.4.4. <https://doi.org/10.1109/IEDM.2015.7409815>.

[50] G. Hills, J. Zhang, M.M. Shulaker, H. Wei, C. Lee, A. Balasingam, et al., Rapid co-optimization of processing and circuit design to overcome carbon nanotube variations, IEEE Trans. Computer-Aided Design Integ. Circ. Syst. 34 (7) (2015) 1082–1095. Available from: https://doi.org/10.1109/TCAD.2015.2415492.

[51] M.M. Shulaker, G. Hills, N. Patil, H. Wei, H.-Y. Chen, H.S.P. Wong, et al., Carbon nanotube computer, Nature 501 (2013). 526 EP.

[52] M.M. Shulaker, G. Hills, R.S. Park, R.T. Howe, K. Saraswat, H.S.P. Wong, et al., Three-dimensional integration of nanotechnologies for computing and data storage on a single chip, Nature 547 (2017). 74 EP.

[53] H.S.P. Wong, H.Y. Lee, S. Yu, Y.S. Chen, Y. Wu, P.S. Chen, et al., Metal oxide RRAM, Proc. IEEE 100 (6) (2012) 1951–1970. Available from: https://doi.org/10.1109/JPROC.2012.2190369.

[54] S. Lee, J. Sohn, H.-Y. Chen, H.-S.P. Wong, Metal oxide resistive memory using graphene edge electrode, Nature Communications (September 25, 2015).

[55] Z. Fang, H.Y. Yu, X. Li, N. Singh, G.Q. Lo, D.L. Kwong, $HfO_x/TiO_x/HfO_x/TiO_x$ multilayer-based forming-free rram devices with excellent uniformity, IEEE Electron Dev. Lett. 32 (4) (2011) 566–568. Available from: https://doi.org/10.1109/LED.2011.2109033.

[56] Y. Kim, S.R. Lee, D. Lee, C.B. Lee, M. Chang, J.H. Hur, et al., Bi-layered rram with unlimited endurance and extremely uniform switching, in: 2011 Symposium on VLSI Technology – Digest of Technical Papers, 2011, pp. 52–53.

[57] A. Grossi, E. Nowak, C. Zambelli, C. Pellissier, S. Bernasconi, G. Cibrario, et al., Fundamental variability limits of filament-based rram, in: 2016 IEEE International Electron Devices Meeting (IEDM), 2016, pp. 4.7.1–4.7.4. <https://doi.org/10.1109/IEDM.2016.7838348>.

[58] Z. Chen, H. Wu, B. Gao, D. Wu, N. Deng, H. Qian, et al., Performance improvements by sl-current limiter and novel programming methods on 16mb rram chip, in: 2017 IEEE International Memory Workshop (IMW), 2017, pp. 1–4. <https://doi.org/10.1109/IMW.2017.7939097>.

[59] R. Fackenthal, M. Kitagawa, W. Otsuka, K. Prall, D. Mills, K. Tsutsui, et al., 19.7 a 16gb reram with 200mb/s write and 1gb/s read in 27nm technology, in: 2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2014, pp. 338–339. <https://doi.org/10.1109/ISSCC.2014.6757460>.

[60] H. Li, et al., Four-layer 3D vertical RRAM integrated with FinFET as a versatile computing unit for brain-inspired cognitive information processing, in: IEEE Symp. VLSI Technology, 2016.

[61] S. Sheu, M. Chang, K. Lin, C. Wu, Y. Chen, P. Chiu, et al., A 4mb embedded slc resistive-ram macro with 7.2ns read-write random-access time and 160ns mlc-access capability, in: 2011 IEEE International Solid-State Circuits Conference, 2011, pp. 200–202. <https://doi.org/10.1109/ISSCC.2011.5746281>.

[62] T. Cabout, J. Buckley, C. Cagli, V. Jousseaume, J.-F. Nodin, B. de Salvo, et al., Role of ti and pt electrodes on resistance switching variability of hfo2-based resistive random access memory, Thin Solid Films 533 (2013) 19–23. eMRS 2012 Symposium L. https://doi.org/10.1016/j.tsf.2012.11.050. http://www.sciencedirect.com/science/article/pii/S0040609012015477.

[63] P. Leduc, L.D. Cioccio, B. Charlet, M. Rousseau, M. Assous, D. Bouchu, et al., Enabling technologies for 3d chip stacking, in: 2008 International Symposium on VLSI Technology, Systems and Applications (VLSI-TSA), 2008, pp. 76–78. <https://doi.org/10.1109/VTSA.2008.4530806>.

[64] C.W. Liang, M.T. Chen, J.S. Jenq, W.Y. Lien, C.C. Huang, Y.S. Lin, et al., A 28nm poly/sion CMOS technology for low-power soc applications, in: 2011 Symposium on VLSI Technology - Digest of Technical Papers, 2011, pp. 38–39.

[65] P. Batude, M. Vinet, B. Previtali, C. Tabone, C. Xu, J. Mazurier, et al., Advances, challenges and opportunities in 3D CMOS sequential integration, in: 2011 International Electron Devices Meeting, 2011, pp. 7.3.1–7.3.4. <https://doi.org/10.1109/IEDM.2011.6131506>.

[66] N. Patil, A. Lin, E.R. Myers, K. Ryu, A. Badmaev, C. Zhou, et al., Wafer-scale growth and transfer of aligned single-walled carbon nanotubes, IEEE Trans., Nanotechnol. 8 (4) (2009) 498–504. Available from: https://doi.org/10.1109/TNANO.2009.2016562.

[67] H. Li, T.F. Wu, S. Mitra, H.-S.P. Wong, Resistive ram-centric computing: design and modeling methodology, IEEE Trans. Circ. Syst. I Regular Papers 64 (9) (2017) 2263–2273.

Chapter 9

Vector multiplications using memristive devices and applications thereof

Mohammed A. Zidan and Wei D. Lu
Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI, United States

9.1 Introduction

As we are approaching a new era that relies heavily on intelligent computing and data analysis, it becomes ever more challenging to fulfill these computing needs using classical computing systems. Current computing architectures, based on the concept of separated processing and memory modules, have now become a fundamental performance-limiting factor, leading to the so-called von Neumann bottleneck [1,2]. The problem gets even worse as data-centric applications are becoming the new norm these days. Historically, the performance improvement in computing systems can always be counted on by simply scaling down complementary metal−oxide−semiconductor (CMOS) transistors according to Moore's law, where the reduction in the device size leads to improved cost, speed, and power consumption. However, with the increased fabrication cost and impending fundamental physical limits, device scaling alone can no longer provide the desired performance gains and Moore's law may finally reach its limit [3,4].

The recent developments in emerging memory devices and new computing principles may provide promising, alternative solutions to modern computing challenges [5]. Memristors are one such technology [6,7] that have gained substantial interest as a candidate for future data storage and efficient in-memory computing paradigms [5,8−11]. The memristor device typically has a simple two-terminal structure form, where only two electrodes and a switching layer are needed. The compact two-terminal structure offers the potential for very-high-density integration and low-cost fabrication. By properly choosing the switching layer and the electrode materials, reliable resistive switching behaviors with rich internal dynamics can be obtained, where

Memristive Devices for Brain-Inspired Computing. DOI: https://doi.org/10.1016/B978-0-08-102782-0.00009-5

the switching material can be dynamically reconfigured when stimulated by electrical inputs [11,12]. In basic memory operations, the resistance of each device is used to represent the stored data [11]. For example, devices in high-resistance state are used to represent binary ZERO, while other devices in low-resistance state are used to represent binary ONE. The high integration density and low fabrication cost, combined with other device metrics in terms of speed, power, retention, and endurance, make memory/storage a natural target market for memristive devices.

However, the real impact of memristive devices may reach far beyond memory applications. Specifically, memristive crossbar arrays have shown extraordinary potential to perform in-memory vector−matrix multiplication (VMM) operations in a massively parallel and power efficient manner. Such operations are the basic building block for a vast number of modern, data-centric computing applications. For instance, state-of-the-art artificial neural networks (ANNs) can be readily mapped to memristive crossbar fabrics that perform weight storage and convolutions locally and in parallel. Tremendous advances have been made recently that demonstrated data classification [13−15], feature extraction [16,17], data clustering [18,19], signal processing [20,21], in-memory data processing [22−24], and security applications [25] by utilizing the in-memory computing capabilities offered by memristive crossbar arrays. It is worthwhile noting that interesting computing applications that are not based on VMM operations have also been developed using memristive devices, such as reservoir computing [26], stochastic computing [27], digital logic [28], field-programmable gate array [29], coupled oscillators [30], shortest path finding [31], and programmable circuitry [32]. However, these implementations are beyond the scope of this chapter.

This chapter is organized as follows. We will first discuss the mechanism that leads to natural implementation of vector−matrix operations in memristive crossbars. Then, we will survey notable examples from the literature that apply memristor-based in-memory computing systems to real-world applications, with an emphasis on hardware and experimental demonstrations. Our exploration of the memristive computing application will be divided into three categories. The first one is termed soft computing application, where qualitative answers are sufficient and expected from the computing system. ANNs are typical examples of soft computing tasks and can potentially be performed in an entirely analog domain, where low computing precision and device nonidealities can be tolerated without suffering from significant performance degradation. The second type of application is hard or precise computing applications. Here, accurate, high-precision numerical values are required and expected as the system output, with much less tolerance to device and circuit level nonidealities. The last category is general memristive computing architectures, which can perform different tasks, soft or hard, through simple software changes. Attempts to build such general computing platforms will be presented and analyzed.

9.2 Computing via physical laws

Resistive memories are typically fabricated in crossbar structures to allow higher density and easier accessibility, as shown in Fig. 9.1A. At the intersection of each two electrode lines lies a memristor device. As a memory block, each device is used to store a binary (or multilevel) value in the form of resistance, where the stored data can be retrieved in a random fashion. In the meantime, through simple changes of the periphery circuitry, the (programmable) resistive elements also allows very efficient in-memory computing implementations.

Specifically, according to Ohm's law, the current that passes through the memristor device is equivalent to the multiplication between the applied voltage and the device conductance, which is shown as follows:

$$i_m = \frac{v_m}{r_m} = v_m \times g_m \tag{9.1}$$

where i_m is the current that passes through the device when a voltage v_m is applied across its terminals, r_m is the device resistance, and g_m is the device conductance. To implement a VMM operation, input voltages are applied to the rows of the memristor crossbar in parallel (as shown in Fig. 9.1B), and the current measured at the columns is obtained following Kirchhoff's current law, such that

$$i_j = \sum_{k=0}^{n-1} g_{k,j} \times v_k \tag{9.2}$$

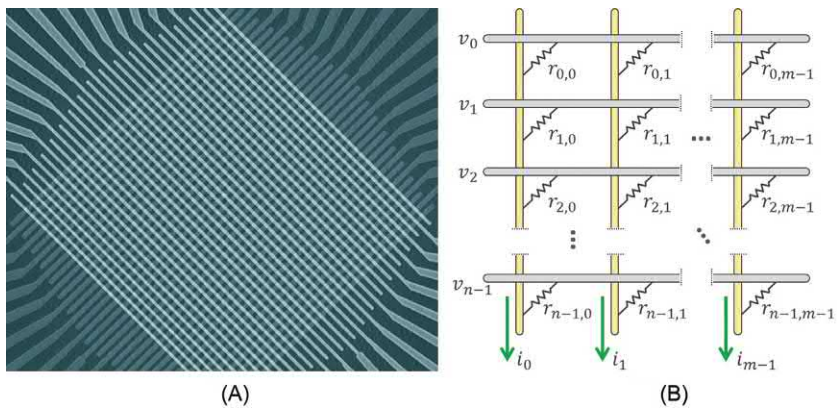


FIGURE 9.1 (A) A scanning electron microscopic (SEM) image of a 1k crossbar fabricated using an Ag/a-Si/Ni stack. (B) Schematic of in-memory computing in a memristor crossbar array, with all the rows activated with the input voltages and the output currents collected from the columns simultaneously. Each memristive device acts as a resistor whose resistance value represents the stored data. *Reproduced with permission from (A) S.H. Jo, K.H. Kim, W.D. Lu, High-density crossbar arrays based on a Si memristive system, Nano Lett. 9 (2) (2009), 870–874 [33]. Copyright 2009 American Chemical Society.*

where k is the row number, j is the column number, and n is the total number of rows. The output current at column j thus represents the dot product operation between the input vector and a stored feature vector:

$$i_j = G_j \cdot V \tag{9.3}$$

where V is the input voltage vector and G_j is a feature vector represented by the device conductances along column j. Note that by measuring output currents from all columns in the crossbar, which can be achieved through a parallel read process, the VMM output can be obtained in a single step as follows:

$$\begin{bmatrix} i_0 \\ i_1 \\ i_2 \\ \vdots \\ i_{m-1} \end{bmatrix} = \begin{pmatrix} g_{0,0} & g_{1,0} & \cdots & g_{n-1,0} \\ g_{0,1} & g_{1,1} & \cdots & g_{n-1,1} \\ g_{0,2} & g_{1,2} & \cdots & g_{n-1,2} \\ \vdots & \vdots & \ddots & \vdots \\ g_{0,m-1} & g_{1,m-1} & \cdots & g_{n-1,m-1} \end{pmatrix} \cdot \begin{bmatrix} v_0 \\ v_1 \\ v_2 \\ \vdots \\ v_{n-1} \end{bmatrix} \tag{9.4}$$

$$I = G \cdot V \tag{9.5}$$

where m is the total number of columns, G is the crossbar conductance matrix, and I is the output current vector.

As seen in the aforementioned example, a VMM operation between an input vector of N elements and a matrix of M elements would involve $N \times M$ multiply-and-accumulate (MAC) processes and is a computationally expensive operation. However, it can be implemented naturally in the memristor crossbar in a single step, in memory and in parallel. The ability of memristor crossbars to efficiently perform VMM operations in memory is a key enabling factor for the data-intensive computing tasks that will be discussed in this chapter.

9.2.1 Data mapping to the crossbar

Mapping of the matrix data to the crossbar can be done in several different ways based on the system requirements and the device and circuit properties. The most area-efficient approach, that is, the ideal case, is to store each parameter value in a single memristive device, whose conductance can be controlled to the desired precision, as shown in Fig. 9.2A. In this approach, the precision of the stored data would be limited by the device properties, where practical devices can offer only tens of different analog levels, equivalent to 4–6 bits. The second approach is to use multiple devices to represent one parameter, such that the effective precision that can be represented is expanded, as shown in Fig. 9.2B and C. This approach is analogous to the "bit-parallel" architecture in digital circuits. Here, the dot-product operation with the input vector will involve several columns in the crossbar, where the columns involved are termed a "slice" in our notation. In this case, each slice

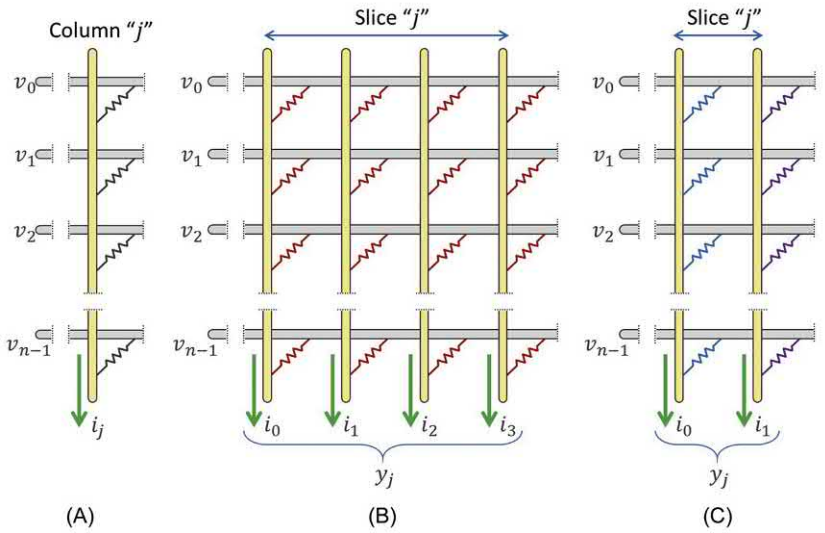


FIGURE 9.2 Three different techniques of mapping data to the memristor crossbar. Each parameter value is represented by the multilevel conductance value of (A) a single device, (B) a group of equally weighted devices, or (C) a group of weighted devices. The group of columns represents a 1D vector of the stored parameters, and is termed a slice.

is computationally equivalent to a single column in the ideal case. The different columns in a slice can either be treated identically (i.e., nonweighted) or in a weighted fashion, as shown in Fig. 9.2A and B, respectively. In the case of nonweighted mapping, all the columns are of the same weight. The current from each column represents a partial product, and the dot-product operation is obtained by the sum of the partial products from different columns in the slice as follows:

$$y_j = \sum_{l=0}^{s-1} i_l = \sum_{l=0}^{s-1} \sum_{k=0}^{n-1} g_{k,l} \times v_k \tag{9.6}$$

where y_j is the equivalent output from the slice, l is the column number within the slice, and s is the number of columns per slice. This approach enables more precise data representation beyond the precision limit of the physical devices, at the expense of increased area of the system.

A more area-efficient approach is to treat the columns of the slice in a weighted manner, following approaches already widely used in digital computing systems. For instance, the digit 8 in the tens' location is ten times larger than the digit 8 in the ones' location. The same principle applies to the binary number system used in digital computing, where each bit location is two times higher in weight compared with the one to its right. In general, in a base-α number system, each digit location is weighted by the powers of

α. So, if our memristor device can represent α number of levels, the output of each column of the slice would be weighted by the powers of α based on its location. The dot-product operation can be defined as follows:

$$y_j = \sum_{l=0}^{s-1} \alpha^l \times i_l = \sum_{l=0}^{s-1} \alpha^l \sum_{k=0}^{n-1} g_{k,l} \times v_k \tag{9.7}$$

The weighted approach is much more area and power efficient compared with the nonweighted ones. However, an extra processing step is required at the output side to set the proper weight factors. It should be noted that multiplying by the powers of α is a simple scaling process that can be relatively easily realized in either analog or digital preprocessing circuitry. In addition, it is worth mentioning that the columns forming a slice can be from different crossbars, rather than within the same crossbar. Organizing the slice columns over multiple crossbars may offer more power management freedom for a system that requires variable computing precisions. In this scenario, when low-precision computing is sufficient, some crossbars can be turned off to save a system's energy consumption.

9.2.2 Input data encoding

In the ideal case, the input data are represented by either the (analog) pulse width or the (analog) pulse amplitude of the voltage pulses applied to the rows of the crossbar. In the case of pulse width representation, as shown in Fig. 9.3A, the integrated current, that is, accumulated charge would represent the computation output. However, a full-analog approach may not always be favorable. First, a computing system would typically deal with the digital or quantized data. Second, analog signals are typically more difficult to handle, store, or move around the system. In these cases, quantized multilevel input signals would be used. A digital-to-analog (DAC) module is typically used to supply the input voltages to the crossbar. Each row DAC circuitry will transform an input digital data to a multilevel input signal to drive the crossbar. Typically, DACs can provide a number of voltage levels equivalent to a few binary bits. For applications requiring higher precision, a stream of discrete pulses can be used in a so-called bit-serial fashion. Collectively, each group of pulses would represent an input. Again, input pulses could be represented in a nonweighted or weighted format. In the case of nonweighted pulses, the output of the dot-product is obtained as follows:

$$y_j = \sum_{q=0}^{p-1} \sum_{k=0}^{n-1} g_{k,j} \times v_{q,k} \tag{9.8}$$

where y_j is the desired dot-product output, q is the input pulse number, p is the total number of pulses per input, and $v_{q,k}$ is the input voltage applied to

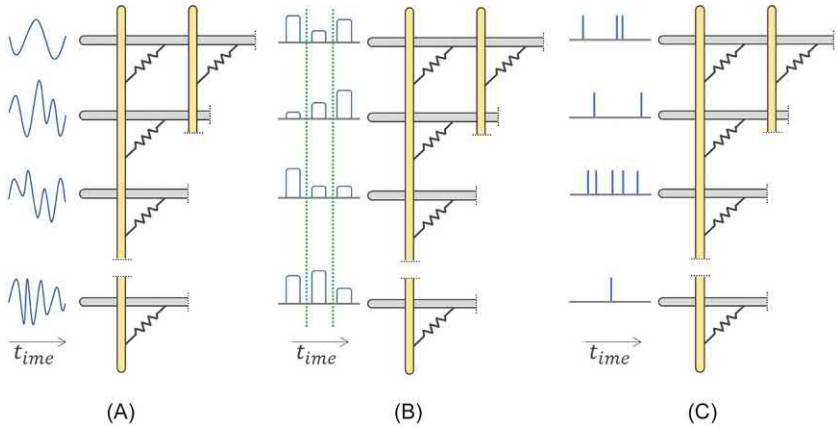


FIGURE 9.3 Three different techniques of encoding the crossbar input data, where data are represented using (A) an analog signal, (B) a stream of synchronized pulses in a bit-parallel or bit-serial fashion, or (C) a train of asynchronous spikes.

row k at pulse number q. Here, the output from each input pulse could be sampled independently and then summed together later. Alternatively, the current of the different pulses could be integrated internally and the accumulated charge, representing the final output, is sampled at the end of the input stream:

$$y_j = \frac{1}{p\tau}\int_0^{p\tau} i_j dt = \frac{1}{p\tau}\int_0^{p\tau}\left(\sum_{k=0}^{n-1} g_{k,j} \times v_k\right) dt \tag{9.9}$$

where τ is the pulse width of each input pulse. The integration could be simply implemented over a capacitor at the column output, where the capacitor voltage represents the integrated charge value. The system in this case would have an analog input/output interface.

Similar to the bit-parallel case, to avoid using too many pulses to represent an input, the pulses could be weighted, that is, in a bit-serial approach. Here, each pulse would have a weight in the power of β, where β is the voltage levels per pulse. In such a case, the output of the dot-product would be:

$$y_j = \sum_{q=0}^{p-1} \beta^q \sum_{k=0}^{n-1} g_{k,j} \times v_{q,k} \tag{9.10}$$

The weighted bit-serial scheme can save significant time and energy consumption, at the expense of extra processing steps. Like the bit-parallel case, it should be noted that multiplying by the powers of β is a scaling process, which can be readily realized. However, the output needs to be sampled for each pulse, as shown in Fig. 9.3B, since it is not trivial to perform the weighted integration in analog circuits, contrary to the case for nonweighted inputs as described in Eq. (9.7).

A third data encoding approach is a stream of (asynchronous) input spikes, as shown in Fig. 9.3C. Here, the input spikes are identical in voltage and duration, while their timing or rate represents the value of the input signals. Such spike-based approaches have been adopted in several applications, especially in bio-inspired neural networks and can lead to very low-power consumption systems [34].

While each of the encoding approaches offers its own advantages and disadvantages, we believe the discrete pulse approach to be more convenient in general. In this approach, the input data are represented digitally (by the pulse stream) allowing for more convenient interfacing with the rest of the system in digital domain. The crossbar would be treated as a digital entity by the system, while the actual computing performed inside the crossbar is achieved in analog domain. This approach allows modular integration to form a larger system, without having to modify the underlying analog computing processes within each crossbar.

9.2.3 Output data sampling

As discussed earlier, the crossbar output current that represents the computation result can be collected in an analog or digital manner. In the analog case, the current is collected and processed (and possibly integrated) in its native form, as shown in Fig. 9.4A. Such an approach may be suitable for low-precision applications. A notable example here is the integrate-and-fire neuron circuitry that can be integrated with the crossbar. In such a case, the neuron integrates the analog current output over time and fires an output spike when a given threshold is passed. However, for many other cases, it is more convenient to digitize the output current before further processing and data handling. Digitizing the output current is typically achieved using analog-to-digital converters (ADCs). During the conversion, all analog values within the quantization thresholds are represented by the same quantized value, thus eliminating noise in the signal so that the digitized data can be amplified (e.g., shifted) and added to improve the effective precision of the system.

The ADC circuitry could pose a significant design challenge to the system. This is because the area and the energy consumption of an ADC grows exponentially with the number of output bits. Theoretically, to perform precise dot-product computation in the crossbar, the output bits required for a column sampling ADC would be:

$$B_{ADC} = B_{in} + B_{device} + \log_2 \omega \tag{9.11}$$

where B_{in} is the number of bits of the input, B_{device} is the equivalent number of bits per memristor device, and ω is the number of activated crossbar rows. Eq. (9.11) suggests that the required ADC bit length could grow beyond practical limits if not handled carefully. For example, 8-bit inputs, 8-bit

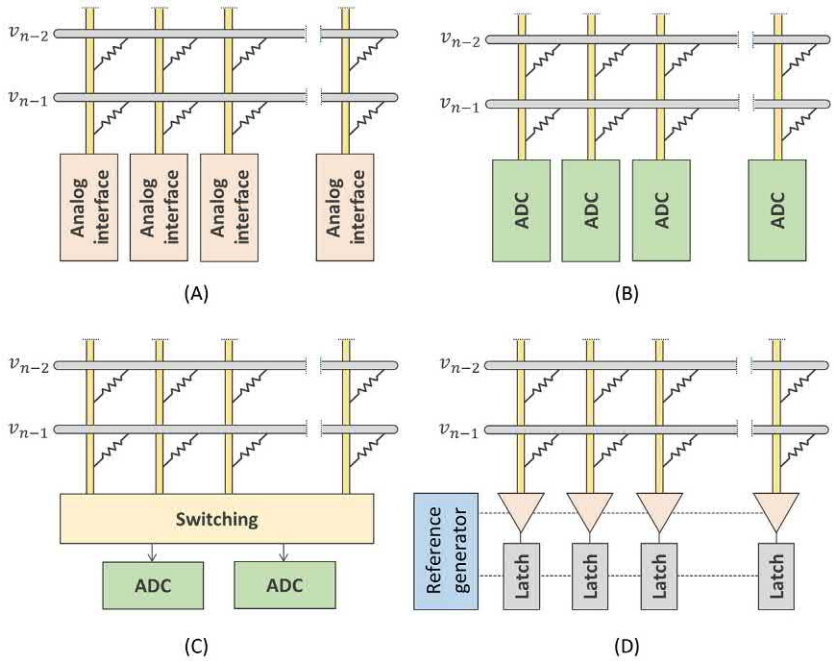


FIGURE 9.4 Four different approaches to sample output currents from a crossbar based on (A) analog circuit interface, (B) an ADC for each column, (C) a group of shared ADCs, or (D) array ADCs. A basic ramp ADC is illustrated in (D), where most of the circuit core is shared by all the columns.

weights, and 128 rows (7 bits) would theoretically require the ADC at each column to have 23 bits, which is impractical to build. Hence, one may need to limit the number of levels per device or input, and use bit-serial and/or bit-parallel techniques discussed earlier, to comply with ADC requirements. It is worth mentioning here that some applications may allow the dot-product results to be truncated after the computation is complete. In this case, truncation can be done at the sampling stage by using a lower precision ADC, where the actual dot product operation has already been performed at a higher accuracy (e.g., analog domain).

In general, ADC area and power optimization will be critical to the system performance. The straightforward approach is to use a stand-alone ADC at each crossbar column (termed parallel ADC approach), as shown in Fig. 9.4B. However, parallel ADCs may not be practical if the ADC size is large. A second option is to use shared ADCs, where multiple columns (or the whole array) share a single ADC, as shown in Fig. 9.4C. Here, the ADC sequentially samples one column at a time, for example, through time-multiplexing, where the system speed is traded for the ADC peripheral area. A potentially promising approach is to use array ADCs, where the common

circuit core is shared among different ADCs, thus reducing the overall area significantly, as shown in Fig. 9.4D. This approach has been historically used in column-parallel CMOS sensor arrays or multichannel biomedical and physical detectors. These ADCs are designed for parallel multinode sampling applications, which is a good match for sampling outputs from the crossbar.

9.2.4 Additional design considerations

It should be noted that different applications may require different sets of properties from the devices and circuit designs. The first step in the design process would be to decide on the memory device. Afterward, circuit or system level optimizations can be applied to mask some of device challenges while fully utilizing the more desirable device properties.

In general, in-memory computing designs will differ from memory-only designs. It may have been conventional wisdom in the memory domain to aim for larger crossbar sizes and high device nonlinearity. These metrics, however, do not necessarily align with in-memory computing system requirements. For instance, a larger array size will lead to increasing parasitic effects such as series line resistance, sneak currents, and imperfect virtual grounds, which could be very undesirable for in-memory applications [35,36]. Moreover, the larger crossbars generally lead to larger interface and sampling circuitry. For instance, as shown in Eq. (9.11), the ADC area is directly proportional to the number of the crossbar size (number of rows). Hence, using larger crossbars does not necessarily improve the crossbar/periphery ratio for in-memory computing applications, while employing smaller crossbar sizes (e.g., 64×64) may allow one to better deal with device and circuit nonidealities and improve the flexibility of the system design.

9.3 Soft computing applications

Arguably one of the most attractive properties of memristor crossbars is their potential to efficiently realize VMM operations. In particular, soft computing applications, which typically aim to obtain an approximate or qualitative solution, can effectively tolerate device and circuit nonidealities such as limited precision and device variabilities often encountered in memristor crossbars, and are natural fits to crossbar-based in-memory computing hardware systems [11,37,38]. For instance, a classification task would be to find out to which category the input belongs, rather than to perform quantitative operations on the input. The former can tolerate significant uncertainty and noise in the system, whereas the latter requires accurate numerical operations since even a single-bit flip in a typical multiplication operation can cause a significant error at the output. Furthermore, most of the soft applications involve an evolutionary training phase, which is typically flexible enough to

accommodate even sizable hardware nonidealities. It has been shown that soft computing tasks can survive with much lower precision, with some applications requiring only six bits or less [15].

An excellent example for soft computing applications is ANN systems. Recently, neuromorphic [39] and machine learning algorithms [40] have shown convincing results in processing cognitive and data-intensive tasks, with some deep neural networks yielding performance approaching or even surpassing those of humans in specific categories [40,41]. Generally, ANNs follow either a bio-inspired approach or an algorithmic machine learning approach, both of which can be traced to our desire to improve computing efficiency by emulating the structure and/or dynamics of the brain. While neuromorphic computing tries to mimic the biological system in device and in circuit domains faithfully, machine learning approaches mainly attempt to efficiently solve a practical problem using a given algorithm without worrying about the biological details. Both approaches have been shown to be able to solve complex problems with much better speed and power efficiency compared with classical techniques. However, mapping any of these techniques to classical computing hardware will unavoidably run into the same von Neumann bottleneck due to large amounts of data that need to be processed by the networks.

A neural network in its purest form is a set of neurons connected by weighted synaptic connections, as shown in Fig. 9.5A. However, modern networks can grow very deep with over 100 layers [42]. This results in an enormous number of synaptic connections, whose numerical representations,

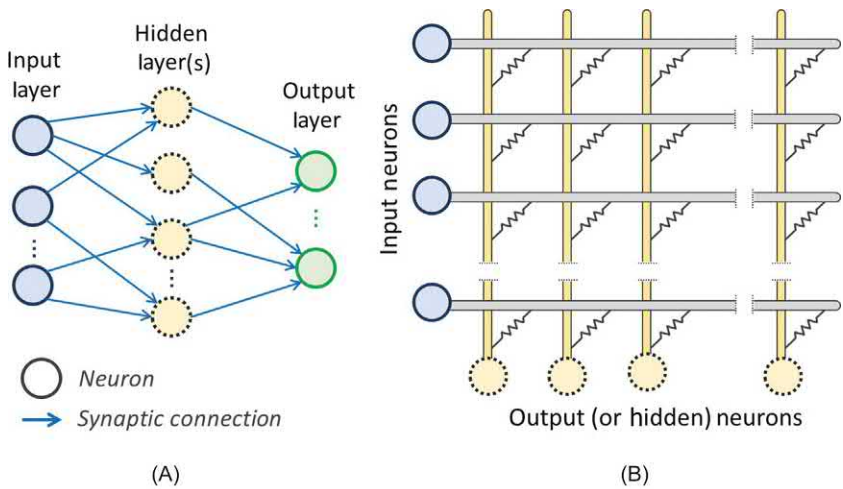


FIGURE 9.5 (A) A feed-forward neural network with an input layer, an output layer, and one (or multiple) hidden layers. (B) A single layer of the neural network is mapped to the crossbar architecture, where each memristive device represents a synaptic connection.

synaptic weights, are stored in off-chip memory and need to be continuously loaded into the processing unit to compute the desired output to the next neuron. As a result, the performance of neural networks on classical systems is still fundamentally limited by the von Neumann bottleneck [34] and requires enormous computing hardware resources and high-power consumption during operation. Conversely, memristive crossbars allow straightforward mapping of neural network layers, with each memristor device acting as a synapse connecting a pair of neurons, as shown in Fig. 9.5B. This approach represents a natural hardware implementation of the neural network at the fine grain. From a mathematical perspective, the output neuron's activation potential is typically determined by the inputs and the synaptic weights as follows:

$$y_j = \sum_{i=0}^{n-1} x_i \times w_{i,j} = X \cdot W_j \tag{9.12}$$

where $w_{i,j}$ is the synaptic weight strength between input neuron i and output neuron j, x_i is input from neuron i, y_j is the membrane potential of output neuron j, and n is the number of inputs. The expression can be written more precisely in the vector form, with X the input vector and W_j the neuron j's feature vector corresponding to the synaptic weights connected to neuron j. As discussed earlier, the input vector−feature vector dot-product operations shown in Eq. (9.12) can be natively mapped to the crossbar system, with the input voltage pulses representing X, memristor device conductance representing $W_{i,j}$, and the output current representing y_j. In addition, as shown in Fig. 9.5B, multiple dot-product operations can be performed in parallel by simply measuring the output currents at multiple columns, essentially performing the VMM operation through the parallel structure of the crossbar. As will be discussed in many examples in this chapter, the co-location of memory and logic and the device-level parallelism offered by the crossbar fabric allow the system to efficiently perform ANN and other data-intensive computing applications.

9.3.1 Data classification

Classification is one of the most successful applications of neural network algorithms. With the ability to identify small differences in subcategories, modern convolutional neural networks (CNNs) were able to approach the classification capabilities of humans [40]. Classification is typically done over two phases: the training stage and the inference stage. In the training stage, the neural network's weights evolve iteratively to reduce the error (defined by a cost function) during the classification of a training set (typically labeled). Afterward, the network is used to classify new inputs to the system in the inference stage using the already trained weights. Recent studies show that both training and inference phases can be efficiently mapped to

memristive crossbar systems, although to perform online training would typically require more demanding device properties, such as higher native precision and higher endurance compared with inference only systems [11]. As a result, a more practical approach would rely on offline training (via software) and map the trained weights to the memristor network afterward to perform inference only. However, the error rate during weight storage needs to be tightly controlled since the system in this approach loses the self-repairing properties offered by the training process. Ultimately, continued device and circuit optimizations may enable the online training to be more easily mapped to the crossbar devices. It should be noted here that the crossbar-based dot-product operation is the backbone for both the training and inference stages.

9.3.1.1 Bio-faithful networks

Generally, there are two different approaches to train the neural network. The first one is following a bio-faithful path that aims to implement training rules that are as close as possible to the observed behaviors in the biology. One example is the spike-timing-dependent plasticity (STDP) [43] behavior, where the weight of the synaptic connection is updated based on the timing difference between the presynaptic and the postsynaptic spikes as follows:

$$\Delta w_{i,j} \propto \text{sign}\left(\Delta t_{i,j}\right) e^{\frac{\Delta t_{i,j}}{\tau}} \tag{9.13}$$

where $w_{i,j}$ is the synaptic weight between the presynaptic neuron i and the postsynaptic neuron j, $\Delta t_{i,j}$ is the timing difference between the presynaptic and postsynaptic neuron spikes, and τ is a decay time constant. The goal is to update memristor conductance, which represents the synaptic weight, following Eq. (9.13), ideally through internal device dynamics that mimic the biological synapse's response to spikes (Fig. 9.6).

Ideally, the STDP system uses a spike-based date encoding (see Fig. 9.3C). The spikes from the presynaptic neurons pass through synaptic connections (memristor devices in the crossbar) and are used to excite postsynaptic neurons. The postsynaptic neuron fires (generates a spike) when its membrane potential crosses a threshold. The postsynaptic spike can back-propagate in the crossbar toward synaptic connections. As a result, the combined effects from the presynaptic and the postsynaptic spiking events may cause the synaptic weight to change, depending on the difference between the spike timing. In practice, the timing difference can be converted to a parameter that modulates the memristor conductance, for example, through carefully designed spike shapes [44,45]. However, this approach requires careful engineering of the pulse shapes and relies on the pulses to overlap with each other. These constraints make the implementation both expensive and less flexible.

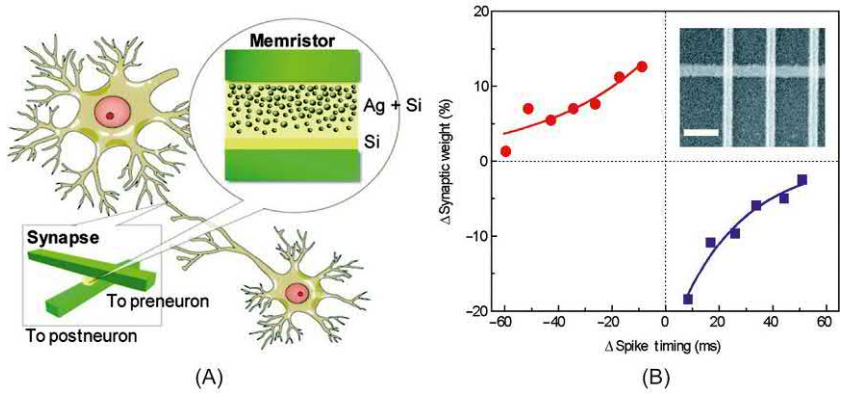


FIGURE 9.6 (A) Using a memristive device to emulate a biological synapse. (B) Measured changes of the memristor conductance versus the relative timing Δt of the neuron spikes, emulating the STDP behavior. Inset: SEM image of the memristor crossbar array. Scale bar: 300 nm. *Reproduced with permission from S.H. Jo, T. Chang, I. Ebong, B.B. Bhadviya, P. Mazumder, W. Lu, Nanoscale memristor device as synapse in neuromorphic systems, Nano Lett. 10 (4) (2010) 1297–1301. Copyright 2010 American Chemical Society.*

Recent developments of second-order memristor devices allow more biorealistic implementations of timing-based learning rules with simple, non-overlapping pulses, by utilizing internal short-term dynamics of the device as a local timing mechanism [46]. Other biologically observed behaviors such as rate-based plasticity [47,48] and neuron dynamics [49,50] have also been demonstrated by taking advantage of internal dynamic processes of memristors.

Network operations based on the STDP learning rule have been demonstrated experimentally [39,51–54] and through simulations based on the observed device characteristics [55–57]. Recent studies have also focused on utilizing the biorealistically implemented learning rules to perform functions such as classification [49,58]. For example, in Ref. [49], the authors used a 8×8 Hafnium oxide-based memristor network to implement a classifier, where diffusive silver oxide devices with short-term memory properties were used to mimic neuron functions.

9.3.1.2 *Machine learning model implementations—classification*

Although biorealistic implementations of neural networks will likely result in further improvements in compute efficiency, mapping algorithm-based machine learning models may be a more practical approach in the near term. In this approach, the memristor network is simply designed to perform VMM operations that are needed in the machine learning models, without worrying about any other biological details.

Several systems have been demonstrated recently. For example, in Ref. [59], a perceptron implemented using a 10×2 titanium oxide crossbar. The network was able to learn (either online or offline) and classify two different symbols; each is made of 3×3 pixels with various pixel flips. Later, the same authors expanded the network to 10×6 crossbars [13] and used the network to perform online learning of three different symbols with the aid of the Manhattan update rule, as shown in Fig. 9.7A. Here, six columns are used to realize three output neurons, where each synaptic weight is represented using two memristive devices to accommodate for the positive and negative weight values. Another example for an online perceptron training was presented in Ref. [14], where the authors used a 128×8 (1T1R) array to perform a face classification task. The network was trained using the Yale Face Database [60] and was able to successively classify three different face classes, as shown in Fig. 9.7B. A two-layer network that can perform MNIST (Modified National Institute of Standards and Technology database) classification was also recently demonstrated [61], achieving 91.71% classification accuracy based on a 128×64 Hafnium oxide memristor array.

In general, these examples utilize analog memristor devices to implement machine learning models. In such a case, each device is used to store a multi-bit value to represent the synaptic weight. However, practical, large-scale implementation of the analog memristor devices would still require extensive materials and device optimization. Using more mature devices such as binary resistive random-access memory (RRAM) devices that are already on the cusp of large-scale commercialization appears to be an attractive option instead, but at the expense of compute density, where a single

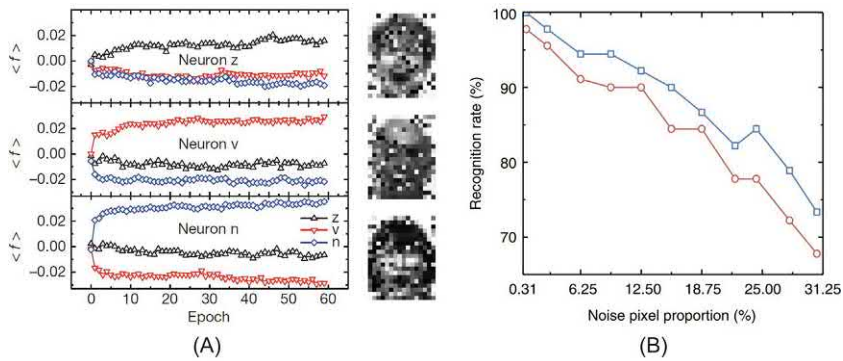


FIGURE 9.7 (A) Postsynaptic neuron output over the training epochs, for three different classes [13]. The training is based on the Manhattan supervised rule. (B) An example of the test pattern used in the face classification system with 100 noise pixels per image (left), and the recognition rate versus the noise percentage (right). The *blue curve* is obtained using a write-verify programming scheme during weight update, while the red curve is obtained without the write-verify stage. *Reproduced with permission from P. Yao, H. Wu, B. Gao, S.B. Eryilmaz, X. Huang, W. Zhang, et al., Face classification using electronic synapses, Nat. Commun. 8 (2017) 15199.*

synaptic weight may require multiple RRAM devices to represent, as discussed in Section 9.2.1.

In Ref. [15], the authors present a realization for a three-layer CNN that performs handwritten digit classification using the MNIST database [62]. The network was realized on a 16-Mb prototype with the ability of online backpropagation training. Another notable example is the work presented in Ref. [63], where the authors show a multilayer network as a MNIST classifier using mature PCM technology. Here, the network was tested for inference, but with a path to online training presented. Beyond experimental demonstrations, more complex and larger scale neural network systems have been extensively analyzed in simulation studies [24,64–67]. These studies verify the feasibility to implement different ANN models using memristor hardware with high energy efficiency once the device technologies are mature enough to support it.

9.3.2 Feature extraction

Another promising application for the memristor crossbar-based hardware is the feature extraction. A feature extraction algorithm would try to extract the common and uncorrelated features from an input data set in an unsupervised manner. Many such algorithms are inspired by the visual cortex, where the network would extract features from natural images that are visually similar to what we observe in the biological systems [68–70].

In a typical implementation, the network will be trained with patches of natural images using update rules such as Oja's rule [24], defined as follows:

$$\Delta w_{i,j} = \eta y_j \left(x_i - y_i \times w_{i,j}\right) \tag{9.14}$$

where $w_{i,j}$ is the synaptic weight strength between the input neuron i and the output neuron j, x_i is the input to the presynaptic neuron i, y_j is the output of the postsynaptic neuron j, and η is the learning rate. Here, the value of the output of the postsynaptic neuron y_j is the dot-product between the input data and the synaptic weight, which again can be readily mapped to memristive crossbars as discussed earlier (see Section 9.2). The Oja's rule is considered a stable version from the Hebbian learning rule. Typically, a Winner-Take-All (WTA) approach is used so that only the winning neuron's weights are updated following Eq. (9.14), thus significantly reducing the training cost.

Fig. 9.8A shows the conductance map measured from different neurons after online training using natural images following Oja's rule and WTA [16]. Other more complex learning rules, such as gradient descent can also be adopted [72]. It is worth mentioning that the term $(y_i \times w_{i,j})$ of Eq. (9.14) can also be performed over the crossbar by supplying the value of y_i as an input from the postsynaptic neuron side and collect the result from the presynaptic side (see Ref. [17] for more details). This approach allows the same

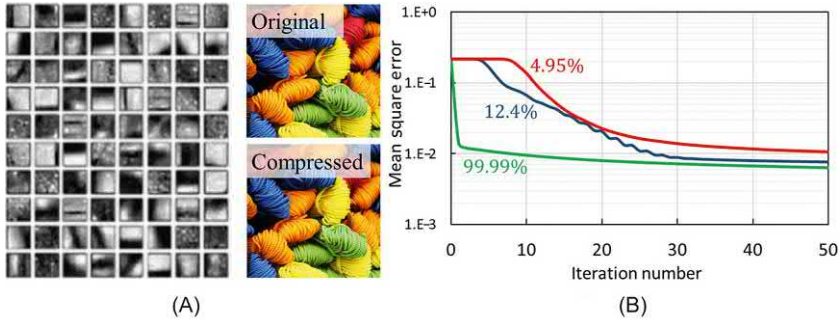


FIGURE 9.8 (A) Features obtained in a memristor crossbar following WTA and Oja's rule. (B) LCA implemented in a binary RRAM crossbar array (left). Right panel plots the mean square error between the original and the reconstructed image at different compression factors (right). *Reprinted with permission from (A) W. Ma, F. Caí, C. Du, Y. Jeong, M. Zidan, W.D. Lu, Device nonideality effects on image reconstruction using memristor arrays, in: IEEE International Electron Devices Meeting (IEDM), 2016, pp. 16.7.1–16.7.4. Copyright 2016 IEEE; (B) M.A. Zidan, Y. Jeong, W.D. Lu, Hybrid neural network using binary RRAM devices, in: IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH), 2017, pp. 81–82. Copyright 2017 IEEE.*

crossbar to also perform the output activation—transpose of weight matrix multiplication, which is frequently employed in many algorithms along with the forward input activation—weight matrix multiplication operations.

The extracted features during the training phase can be used as the biases for the feature extraction analysis (e.g., inference) applications. One notable example here is the sparse coding based approach, implemented in algorithms such as the locally competitive algorithm (LCA) [69]. Sparse coding aims to reconstruct the image using the dictionary set, with a goal to not only minimize the reconstruction error but also minimize the number of features used. The algorithm results in a compressed version of the original image while balancing sparsity and accuracy constraints. A notable work, presented in Ref. [17], shows successful experimental implementation of the LCA algorithm using memristor crossbars. Here, a 32×32 Tungsten oxide-based crossbar was used to map the weights and implement the LCA algorithm, assisted by periphery circuitry on a test board. By iteratively applying the forward input activation—weight matrix multiplication operations to obtain output neuron activities, and the output activation—transpose of weight matrix multiplications to obtain the reconstruction, the authors were able to implement lateral neuron competition, a key operation in LCA, using the same memristor and avoiding expensive all-to-all connections among the output neurons [17]. Later simulation studies using binary RRAM devices further verified the experimental results [24,71] (see Fig. 9.8B).

A second example of feature extraction is principal component analysis (PCA). PCA aims to transform the original data, represented in a large

number of possibly correlated variables (in a large dimension), into data that can be represented by a much smaller number of linearly uncorrelated variables [so-called principal components (PCs)]. PCA has been widely used in practice for feature extraction and data clustering. In general, PCA is a computationally expensive algorithm and in its native form cannot be directly mapped to a crossbar-based computing system. However, it has been shown that instead of directly solving the PCs, they can be learned through the generalized Hebbian rule (GHR) in an unsupervised fashion [73]. The GHR, which is also known as Sanger's rule, is defined as follows [73]:

$$\Delta w_{i,j} = \eta y_j \left(x_i - \sum_{k=0}^{j} y_k \cdot w_{i,k} \right) \tag{9.15}$$

where $w_{i,j}$ is the synaptic weight between the input neuron i and the output neuron j, x_i is the input from neuron i, y_j is the output from neuron j, and η is the learning rate. Here, the number of output neurons would represent the number of desired PCs to learn [18,74]. Similar to Oja's rule discussed earlier, the GHR rule can be readily mapped onto memristor crossbars. For instance, the values of y_j can be computed using a forward dot-product operation, while the $y_k \cdot w_{i,k}$ terms can be obtained from the backward dot-product operations by feeding the output neuron activation as inputs.

After the learning phase, each column of the crossbar would represent a PC vector [18,74]. The network can then be used to perform PCA on test dataset, where the original data can be mapped to the space represented by PCs through VMM operations between the input data and the PCs matrix, an operation that can be readily mapped on memristor crossbars.

Results from one such experimental demonstration [74] are shown in Fig. 9.9. A 9×2 array based on tantalum oxide memristors is used to perform the online learning and PCA analysis for a standard breast cancer screening data set. After projecting the original (nine-dimensional) data on the two-dimensional space represented by the first PCs, the benign and malignant cells are grouped into different clusters and can be reliably separated through a decision boundary drawn using supervised learning.

9.3.3 Data clustering

Data clustering is another example of data-intensive computing applications that can largely benefit from the matrix operation capabilities of the memristor crossbars. Here, a clustering algorithm divides the data into different groups in an unsupervised manner. One of the standard data clustering techniques is the K-means algorithm, which has been experimentally realized by memristor crossbars in Ref. [19]. The K-means algorithm groups unlabeled data into "K" different clusters based on minimizing the Euclidian distance between the data and the cluster centroids. The algorithm typically starts with a set of unlabeled points and allows new points to be added to the

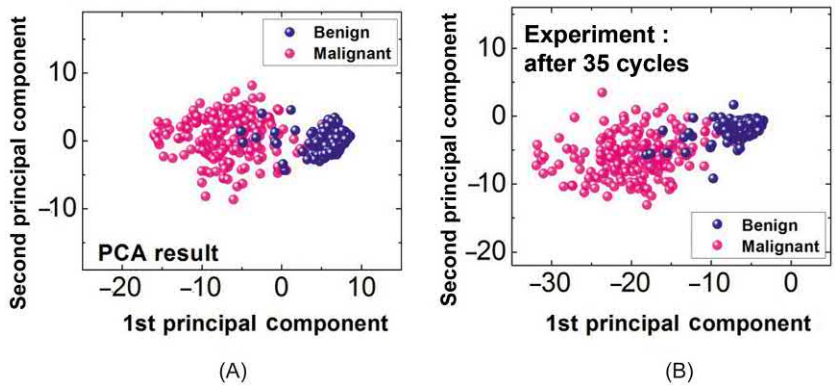


FIGURE 9.9 PCA analysis for malignant and benign cells. (A) Results obtained by directly solving the first two PCs through software and (B) results obtained experimentally from the memristor crossbar by learning the first two PCs using the GHR. *Reproduced with permission from S. Choi, J.H. Shin, J. Lee, P. Sheridan, W.D. Lu, Experimental demonstration of feature extraction and dimensionality reduction using memristor networks, Nano Lett. 17 (5) (2017), 3113–3118. Copyright 2017 American Chemical Society.*

cluster with the shortest Euclidian distance. The K-means algorithm has many similarities with other algorithms discussed earlier. On the other hand, instead of comparing the similarity of the input vector and a feature vector that can be represented by a dot-product, the algorithm relies on calculating the Euclidian distances, which offers additional challenges.

Specifically, the square of the Euclidian distance is defined as follows:

$$\|X_i - C_j\|^2 = \|X_i\|^2 + \|C_j\|^2 - 2X_i \cdot C_j \tag{9.16}$$

where X_i represents the data point i and C_j is the center of centroid j. While the $X_i \cdot C_j$ operation is a standard dot-product that can be directly mapped to the memristor crossbar, the $\|C_j\|^2$ operation requires further attention, since for features whose amplitudes are not normalized, the addition of the $\|C_j\|^2$ term suggests that the smallest dot-product output may not correspond to the shortest Euclidian distance. Since the $\|X_i\|^2$ is the same for all the clusters, it does not affect comparison of the Euclidian distances.

This challenge was addressed in Ref. [19], where the authors introduced an extra row that holds the value $\|C_j\|^2$ of different centroids to the crossbar, as shown in Fig. 9.10A. Hence, the expanded crossbar can now perform both the $X_i \cdot C_j$ term and the $\|C_j\|^2$ term and obtain the desired results shown in Eq. (9.16) directly in the crossbar, differing only by the constant $\|X_i\|^2$ term. After identifying the cluster whose centroid is closest to the data point, the second step is to update the centroid location accordingly with the addition of the new point [19]. This process is then repeated until all points have been assigned to a cluster.

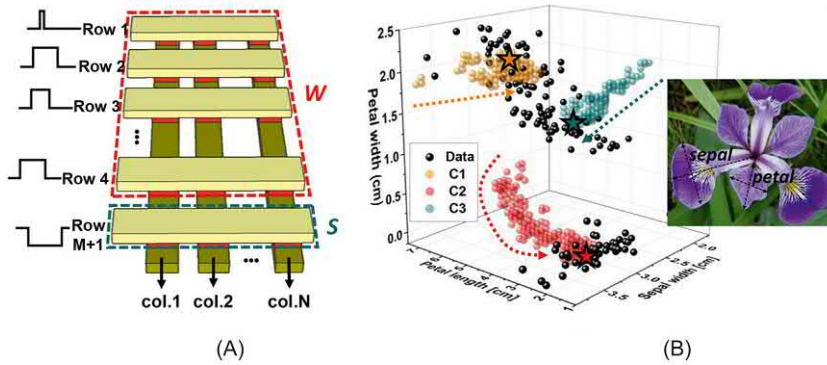


FIGURE 9.10 (A) Mapping of the K-means algorithm to a memristor crossbar, where the coordinates of each centroid are stored as the memristor conductance values [19]. The last row represents the square of the norm of the centroid vector. (B) Evolution of the centroid locations during online training with the IRIS dataset. Inset: Parameters of the IRIS dataset, including the length and width of the sepal and the petal. *Reproduced with permission from Y. Jeong, J. Lee, J. Moon, J.H. Shin, W.D. Lu, K-means data clustering with memristor networks, Nano Lett. 18 (7) (2018), 4447−4453. Copyright 2018 American Chemical Society.*

Fig. 9.10 shows the excremental results obtained using the Tantalum oxide crossbar for the K-means algorithm. Here, the authors used a standard IRIS dataset to test the algorithm operation with classification accuracy comparable to software implementations.

9.3.4 Signal processing

Signal processing is one of the fields that would largely benefit from in-memory and parallel computing capabilities. Various signal-processing techniques rely on vector−matrix operations, such that significant speed-up and energy savings can be obtained in memristor-based in-memory computing systems. Moreover, many signal-processing techniques are soft such that slight shifts in the computed values have little impact on the system performance. A good example here is the convolution filters, where an input signal is convolved with a filter or a mask to create the output. For instance, in case of 2D image processing, a convolution matrix is used to scan over the whole image, and at each step, the dot-product between the covered input image area and the matrix generates one pixel at the output image, as shown in Fig. 9.11A. A hardware implementation for image filtering was demonstrated in Ref. [20], using a Hafnium oxide-based memristor array. A 5×5 convolution matrix was used to process input images. The 25-element filter was realized using 50 memristor devices, with 2 devices representing each matrix element to accommodate negative values.

Another interesting signal processing application is lossy image compression. In lossy image compression, the size of the data is significantly reduced

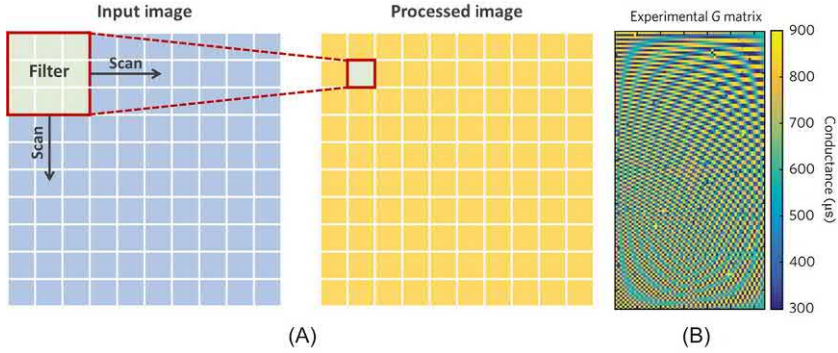


FIGURE 9.11 (A) Processing a 2D image using convolutional filters. The output pixel is the result of the dot-product between the filter and the covered input image pixels. (B) A discrete cosine transform basis matrix mapped to a memristor array and used to perform lossy image compression. *Reproduced with permission from C. Li, M. Hu, M. Hu, Y. Li, H. Jiang, N. Ge, et al., Analogue signal and image processing with large memristor crossbars, Nat. Electron. 1 (1) (2018), 52–59. Copyright 2018 Springer Nature.*

while minimizing the effect on its visual properties. The sparse coding image compression techniques presented in Ref. [17] also falls into this category. In the sparse coding case, common features of natural images are used to represent the original image in a sparse manner. Another approach of lossy image compression is the discrete cosine transform (DCT) technique. The DCT can be considered as a special form of discrete Fourier transform, where only real numbers are used. In this case, the input signal (image) can be represented using its frequency components, where higher frequencies can be eliminated without significant degradation to the image quality. This technique was implemented using memristor arrays in Ref. [20], where the authors used a 128×64 (1T1R) memristor array to store the DCT basis matrix (as shown in Fig. 9.11B). Afterward, the image compression can be achieved using VMM as follows:

$$Y = X \cdot M_{DCT} \tag{9.17}$$

where X represents the input signal vector, Y represent the output (i.e., compressed signal) vector, and M_{DCT} is the DCT basis matrix. Using memristor arrays to perform the VMM operation can provide a very low energy implementation that is highly desirable for IoT and edge computing systems. Relevant work, presented in Ref. [21], demonstrated the other face of the coin, where compressed signals are recovered. In Ref. [21], the authors employed a PCM array to perform the VMM required for compressed image recovery used for reconstruction. Finally, it is worth mentioning that the internal device dynamics can also be used for signal-processing techniques, particularly for temporal inputs, where streaming input pulses can excite the devices into different states depending on the temporal pattern of the inputs [75]. The excited

memristor devices can be treated as a reservoir, where handwritten digit recognition and nonlinear signal modeling have been successfully implemented using such memristor-based reservoir computing systems [26].

9.3.5 Security applications

Memristor dynamics have been shown to be very useful for security applications as well, in cases such as physical unclonable functions (PUFs) [76–81], true random number generators [82–85], and others [86]. In these applications, the intrinsic device variability is treated as a key property that the designers aim to exploit. For instance, PUFs can be built based on device-to-device variability [87], while random number generation can be based on cycle-to-cycle variability [27]. In this sense, one can categorize these applications to the soft computing category. It is also worthwhile to note that typically these applications may only depend on single-device properties and do not utilize the vector multiplication computing abilities offered by the memristor arrays. On the other hand, dot product operations can still be beneficial, as demonstrated in Ref. [25]. In this work, the authors experimentally demonstrated a PUF using two stacks of 10×10 memristor arrays based on aluminum and titanium oxides, where the arrays are filled with random data patterns. Typically, PUFs depend on a random physical factor introduced during the fabrication/manufacturing process of the system. The presented work in Ref. [25] utilizes the variations in the nonlinear $I-V$ characteristics of memristors as the source of the physical randomness. These intrinsic and unique variability patterns lead to similarly unique current patterns with a given input voltage that helps protect the circuit design.

9.4 Precise computing applications

So far, the focus of memristor-based hardware has been on "soft" computing tasks such as ANNs, where only qualitative solutions are required. The other class can be categorized as "hard" computational tasks, and would require accurate solutions and has been generally considered challenging to implement in memristor-based systems. For example, a well-designed memristor device may provide around 64 different resistance levels [88,89], which is equivalent to 6 bits. On the other hand, practical numerical tasks may require up to 64 bits (2^{64} levels) of precision. An in-memory computing platform that can process both soft and hard, high-precision computing tasks would thus greatly expand the potential of memristor hardware systems and allow the system to appeal to a broad range of use cases that deal with large amounts of data at high speed and with high energy efficiency.

In this section, we discuss hardware implementations of memristor-based computing systems that can effectively operate at high precision, making them suitable for such hard computing tasks.

9.4.1 In-memory arithmetic accelerators

Similar to soft tasks, the most promising approach for memristor hardware in numerical computations is as in-memory accelerators. A representative application here is the memristor-based in-memory partial differential equation (PDE) solver [23]. Solving PDE is a core operation for many engineering, economic, and scientific systems. In general, the majority of PDEs are solved using numerical methods that are computationally expensive, involving iterative VMMs with massive amounts of data. In fact, the most powerful supercomputers are generally built to carry out such large-scale numerical computation tasks. A functional, efficient, and parallel memristor-based high-precision computing system would be invaluable to such type of applications.

Without losing generality, many PDE systems can be formulated as solving a system of equations:

$$A \cdot X = B \tag{9.18}$$

where X is the unknown vector to be solved, A is the coefficient matrix, and B is a constant vector containing the boundary conditions. Such problems can be solved using several iterative numerical techniques such as the Jacobi method, the Gauss-Seidel method, successive overrelaxation, or the conjugate gradient method. Naturally, the core computing operation in these methods, VMM, can be implemented in memristor crossbars as in the soft computing cases (e.g., ANNs). However, solving PDEs would require very high precision since small errors will be accumulated and multiplied due to the iterative nature of the PDE solvers, while practical memristive devices cannot provide the number of bits or the accuracy required due to nonidealities in controlling the device resistance (e.g., cycle-to-cycle variations and switching stochasticity) and controlling the device uniformity (e.g., device-to-device variations) that lead to errors that are too high for PDE solvers.

One approach is to use a mixed-precision setup, as discussed in [22]. In such a case, the actual high-precision computation is still performed on a digital system. However, the memristive system leads to a significant speed-up of the system by providing a lower precision initial solution. The solutions are obtained in an iterative manner using the low-precision memristive system and the high-precision digital system using the Krylov-subspace numerical technique. The concept was demonstrated experimentally using a prototype PCM chip with 3 million integrated devices.

A different approach that can provide high-precision computing using memristive hardware alone was experimentally demonstrated in Ref. [23], using tantalum oxide RRAM devices. Several architectural innovations were employed in Ref. [23] to allow the mapping of the high-precision PDE solver on the physical RRAM-based in-memory computing system. The first problem solved was how to map the typically very large-sized PDE coefficient matrices to the physical crossbars. For example, a PDE problem in a small

2D grid of 100×100 would require a $10{,}000 \times 10{,}000$ coefficient matrix with 10^8 elements. Luckily, such matrices are typically very sparse in nature and can be divided into equally sized slices such that only the active slices (the ones containing non-zero elements) are needed to be mapped onto memristor crossbars. The second and more challenging problem is mapping the high-precision values of the coefficient to the memristive devices that can support only a limited number of levels. This was achieved with the aid of a precision extension technique. The effective precision of the system can be extended through the use of multiple crossbars, where each crossbar represents a given number of bits determined by the native device precision. This precision expansion approach is similar to the techniques used in digital circuits, where binary (two-level) physical values, such as capacitor voltages in dynamic random-access memory, are used as the basis of high-precision computing systems. Finally, the authors addressed the device variability problem at two different levels. At the device level, a write−verify technique was used to reduce the device variability to the minimum achievable level. At the circuit level, quantization techniques were employed to reduce the analog error before the digitized values are shifted and added. This approach prevented the analog errors from being amplified during the precision extension process and enabled the system to achieve an effective higher precision than that of individual devices.

The Jacobi method was used to iteratively solve the PDEs through the tantalum oxide memristor crossbars. An example of the experimental results is shown in Fig. 9.12, where up to 64 bit precision has been experimentally

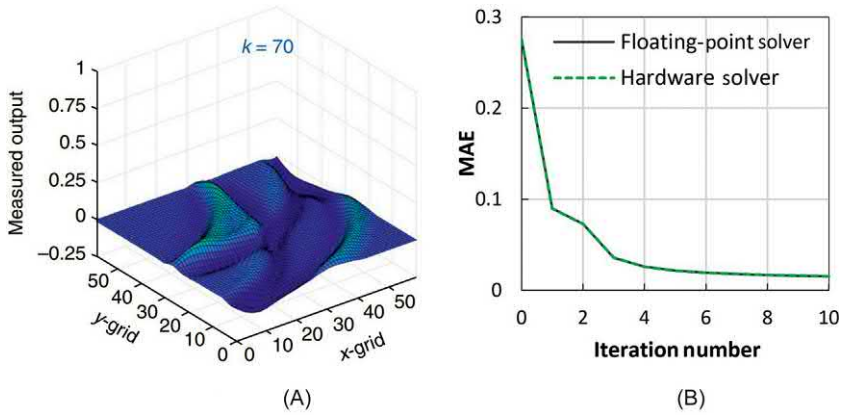


FIGURE 9.12 (A) Solutions of a time-evolving wave equation obtained from a memristor solver, simulating wave propagation in a shallow water system. (B) Evolution of the mean absolute error for the memristor-based hardware solver and a floating-point solver, measured against the exact numerical solution. *Reproduced with permission from M.A. Zidan, Y. Jeong, J. Lee, B. Chen, S. Huang, M.J. Kushner, et al., A general memristor-based partial differential equation solver, Nat. Electron. 1 (7) (2018) 411−420.*

demonstrated. By performing computing in-memory that minimizes the data movement and maximizes parallelism, memristor-based systems would provide significant performance improvement over their digital counterparts even for hard computing tasks [22,23].

Beyond the experimentally demonstrated memristor-based PDE solver, simulation studies have also suggested general arithmetic operations based on memristor hardware [24], using similar precision-extension techniques for the arguments involved and using an in-line data migration technique to store the intermediate results for the next operations.

9.4.2 Logic circuitry

Memristive devices have also been widely investigated as MOS transistor replacements in the digital logic circuitry due to their compact size and non-volatility [28,90–93]. In many cases, the device is used only as a switch, rather than a memory element. As a result, these approaches do not offer the full benefits of in-memory computing systems. However, dot-product or VMM operations can still be beneficial. One such example is lookup table (LUT) circuitry. In the classic version, LUTs are used to store the truth table of a Boolean function output, and by reading the entry equivalent to the input code, one gets the expected output. Typically, the LUT's size grows exponentially with the number of the inputs, so the high-density offered by RRAM crossbar becomes a very attractive feature. An example of RRAM-based LUT for logic operations is shown in Ref. [92], where polynomial expression results can be dynamically computed during runtime as needed. After programming, the crossbar acts like a typical LUT where an output can be found from the memory for every input for a given function. Here, different input combinations and the associated output values for a given function are stored in the form of device conductances in the crossbar, along the same row, as shown Fig. 9.13A. A Boolean expression is evaluated by applying voltage pulses representing the arguments to the crossbar columns, equivalent to a binary-input/binary-matrix dot-product. The row that stores the input combinations matching the actual input produces a low current and can be identified by comparing the current outputs with a predefined threshold. After identifying the correct row in the LUT, the target output can then be read out. This system was experimentally demonstrated using an aluminum oxide crossbar and used to implement essential logic functions such as full-adders.

Similar LUT functionality can be achieved by using a small number of crossbar rows by programming each new input to the devices' conductance, rather than keeping an entry for each input combination, as presented in Ref. [93]. This approach can be viewed as a tradeoff between the system speed and the device endurance from one side and the compute density from the other side. In addition, the LUT concept can be expanded to 3D, where

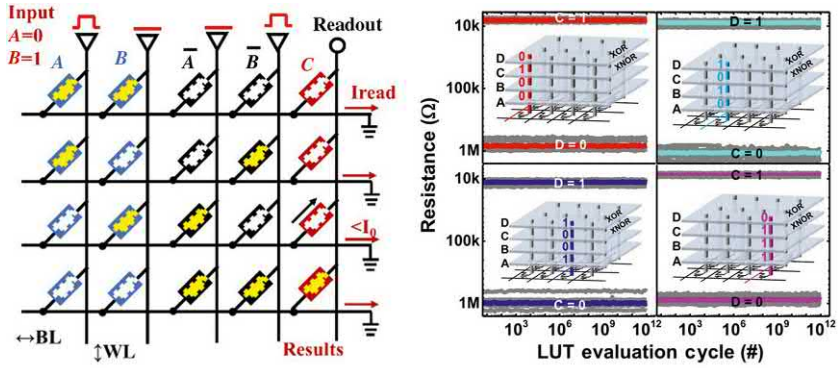


FIGURE 9.13 (A) Dynamically programmable logic LUT. (B) 3D-LUT demonstrations. Showing measured data of up to 10^{12} cycles of correct and robust XOR and XNOR functions for different input/output combinations. The logic input/output values are stored by RRAM cells in vertical pillars. *Reprinted with permission from (A) B. Chen, F. Cai, J. Zhou, W. Ma, P. Sheridan, W.D. Lu, Efficient in-memory computing architecture based on crossbar arrays, in: IEEE International Electron Devices Meeting (IEDM), 2015, pp. 17.5.1–17.5.4, Copyright 2015 IEEE; and (B) H. Li, T.F. Wu, S. Mitra, H.-P. Wong, Resistive RAM-centric computing: design and modeling methodology, IEEE Trans. Circuits Syst. I: Regul. Pap. 64 (9) (2017), 2263–2273. Copyright 2017 IEEE.*

RRAM cells along the 3D vertical pillars are programmed to represent various logic input/output data [94]. This 3D-LUT approach further improves the compute density and also effectively addresses the limited write endurance issue facing RRAM devices.

9.5 General memristor-based multiply-and-accumulate accelerators

By utilizing memristor crossbars to perform VMM (MAC) operations for either soft or hard computing tasks, general MAC or dot-product hardware accelerators have been proposed [95,96]. One of the first experimental works toward a memristor-based computing unit was the demonstration of hybrid memristor/CMOS integration presented in Ref. [95]. In this work, a 40×40 crossbar array was successfully integrated with CMOS peripheral circuit using local, high-density CMOS vias. Excellent memory performance was obtained from the integrated memristor/CMOS system, as shown in Fig. 9.14A. Another notable example is the dot-product engine presented in Ref. [35]. The authors demonstrated in-memory MAC operations using 256×256 1T1R arrays, with potential applications in signal-processing and classification domains. Fig. 9.14B shows the implemented chip with different array sizes. The work also highlighted the effect of device and circuit

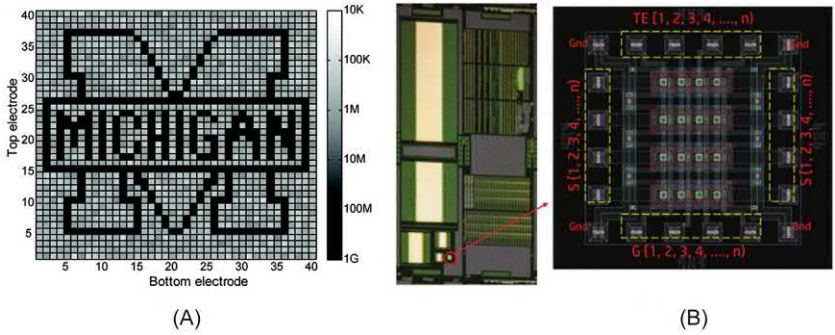


FIGURE 9.14 (A) A reconstructed bitmap image obtained by storing and retrieving data in the 40×40 crossbar array presented in Ref. [95]. (B) An optical image for the dot product engine presented in Ref. [35] and the layout of one of the 4×4 1T1R test crossbars. *Reproduced with permission from (A) K.-H. Kim, S. Gaba, D.C. Wheeler, J. Cruz-Albrecht, T. Hussain, N. Srinivasa, et al., A functional hybrid memristor crossbar-array/CMOS system for data storage and neuromorphic applications, Nano Lett. 12 (1) (2012) 389–395. Copyright 2012 American Chemical Society; (B) Reprinted with permission from M. Hu, J.P. Strachan, Z. Li, E.M. Grafals, N. Davila, C.E. Graves, et al., Dot-product engine for neuromorphic computing: programming 1T1M crossbar to accelerate matrix-vector multiplication, Design Automation Conference (DAC) (2016), 1–6. Copyright 2016 IEEE.*

nonidealities on the computing accuracy, where a system level algorithm was proposed to compensate for effects such as line resistance during the programming phase. A similar MAC engine was presented in Ref. [96], where layers of memristor crossbars were monolithically integrated over the CMOS base circuitry. A two-layer 24×36 system was demonstrated, where each device in the arrays is capable of storing multilevel data for targeted computing applications.

From an architecture perspective, an MAC accelerator engine could be used as a reconfigurable processor unit as discussed in Ref. [24], and shown in Fig. 9.15. A system can then be formed by identical memristive cores, where each core (or a fraction of it) can be configured for storage, analog computing, or digital computing tasks. The system would be fabricated using the same physical fabric, where different functions can be achieved in its optimal computing domain (soft or precise) through software-based reconfiguration. The natively modular design of the system in turn allows for a high degree of scalability and reconfigurability to tailor fit different workloads. For each task, the merge of compute and memory at the lowest physical level helps to achieve maximal efficiency and minimal data migration. Such types of in-memory processors could theoretically outperform their digital counterparts for a broad range of tasks [24]; however, detailed architecture design/analysis and device optimizations are still required for the realization of these types of systems.

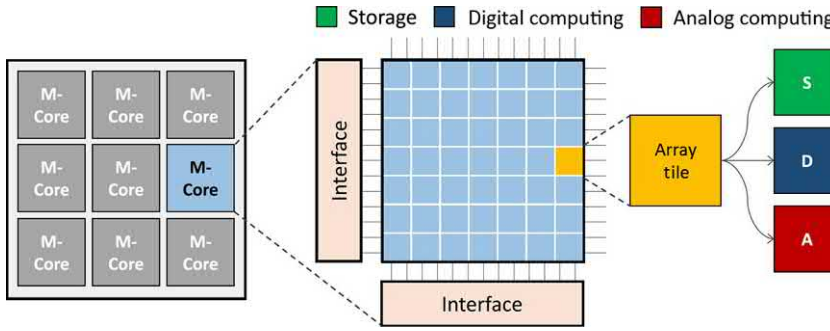


FIGURE 9.15 Schematic diagrams of a reconfigurable system, termed field-programmable crossbar array presented in Ref. [24], showing the crossbar-based blocks that can be reconfigured for different functions. *Reprinted with permission from M.A. Zidan, Y. Jeong, J.H. Shin, C. Du, Z. Zhang, W.D. Lu, Field-programmable crossbar array (FPCA) for reconfigurable computing, IEEE Trans. Multi-Scale Comput. Syst. 4 (2018), 698–710. Copyright 2018 IEEE.*

9.6 Conclusion

In this chapter, we discussed the ability of memristive crossbars to efficiently perform in-memory vector-matrix operations. We introduced the underlying theory and different approaches to map the VMM operations to the memristor crossbar fabric. Several interesting applications based on this approach have been analyzed. They include soft computing applications that can tolerate errors and hardware limitations and precise computing applications that require accurate numerical processing of the data. Different types of applications all benefit tremendously from the native and parallel MAC processing capabilities of the memristor crossbar.

In general, we can view an in-memory processor or accelerator as the natural evolution of the computing paradigm, following the trend of the shift from central processing units to graphics processing units for data-intensive tasks, moving towards finer-grained and highly parallel structures. As discussed in this chapter, such an in-memory processing system can find broad applications ranging from high-performance machine-learning systems to low-power embedded chips for edge computing. We would expect that initial implementations of such systems will be at the edge side, where energy efficiency will be critical. However, with continued advances in device technology larger scale implementations will become possible, which may find their place in servers and cloud systems.

Acknowledgments

The authors are indebted to helpful and stimulating discussions with their colleagues, Drs. Patrick Sheridan, Shinhyun Choi, Chao Du, Wen Ma, Yeongjoo Jeong, Fuxi Cai, Zhengya Zhang, and Michael Flynn. The results presented in this chapter have been supported

through numerous grants, including the SyNAPSE, UPSIDE, and ACCESS programs from the Defense Advanced Research Projects Agency (DARPA), the Air Force Office of Scientific Research (AFOSR) through MURI grant FA9550-12-1-0038 and the National Science Foundation (NSF) through grant CCF-1617315.

References

[1] P. Kogge, K. Bergman, S. Borkar, D. Campbell, W. Carlson, W. Dally, et al., Exascale computing study: technology challenges in achieving exascale systems, Def. Adv. Res. Proj. Agency Inf. (2008).

[2] S. Borkar, A.A. Chien, The future of microprocessors, Commun. ACM 54 (5) (2011) 67–77.

[3] M.M. Waldrop, The chips are down for Moore's law, Nature 530 (7589) (2016) 144.

[4] J. Shalf, R.W. Leland, Computing beyond Moore's Law, IEEE Computer 48 (12) (2015) 14–23.

[5] M.A. Zidan, J.P. Strachan, W.D. Lu, The future of electronics based on memristive systems, Nat. Electron. 1 (1) (2018) 22–29.

[6] L.O. Chua, S.M. Kang, Memristive devices and systems, Proc. IEEE 64 (2) (1976) 209–223.

[7] D.B. Strukov, G.S. Snider, D.R. Stewart, R.S. Williams, The missing memristor found, Nature 453 (7191) (2008) 80–83.

[8] J.J. Yang, D.B. Strukov, D. Stewart, Memristive devices for computing, Nat. Nanotechnol. 8 (1) (2013) 13–24.

[9] Y.V. Pershin, M.D. Ventra, Neuromorphic, digital, and quantum computation with memory circuit elements, Proc. IEEE 100 (6) (2012) 2071–2080.

[10] H.-S. Wong, H.-Y. Lee, S. Yu, Y.-S. Chen, Y. Wu, P.-S. Chen, et al., Metal–oxide RRAM, Proc. IEEE 100 (6) (2012) 1951–1970.

[11] M.A. Zidan, A. Chen, G. Indiveri, W. Lu, Memristive computing devices and applications, J. Electroceram. 39 (1–4) (2017) 4–20.

[12] J. Lee, W.D. Lu, On-demand reconfiguration of nanomaterials: when electronics meets ionics, Adv. Mater. 30 (1) (2018) 1702770.

[13] M. Prezioso, F. Merrikh-Bayat, B. Hoskins, G.C. Adam, K.K. Likharev, D.B. Strukov, Training and operation of an integrated neuromorphic network based on metal-oxide memristors, Nature 521 (7550) (2015) 61–64.

[14] P. Yao, H. Wu, B. Gao, S.B. Eryilmaz, X. Huang, W. Zhang, et al., Face classification using electronic synapses, Nat. Commun. 8 (2017) 15199.

[15] S. Yu, Z. Li, P.-Y. Chen, H. Wu, B. Gao, D. Wang, et al., Binary neural network with 16 Mb RRAM macro chip for classification and online training, in: IEEE Electron Devices Meeting (IEDM), 2016, pp. 16.2.

[16] W. Ma, F. Caí, C. Du, Y. Jeong, M. Zidan, W.D. Lu, Device nonideality effects on image reconstruction using memristor arrays, in: IEEE International Electron Devices Meeting (IEDM), 2016, pp. 16.7.1–16.7.4.

[17] P. Sheridan, F. Cai, C. Du, W. Ma, Z. Zhang, W.D. Lu, Sparse coding with memristor networks, Nat. Nanotechnol. 12 (8) (2017) 784–789.

[18] S. Choi, P. Sheridan, W.D. Lu, Data clustering using memristor networks, Sci. Rep. 5 (2015) 10492.

[19] Y. Jeong, J. Lee, J. Moon, J.H. Shin, W.D. Lu, K-means data clustering with memristor networks, Nano Lett. 18 (7) (2018) 4447–4453.

[20] C. Li, M. Hu, M. Hu, Y. Li, H. Jiang, N. Ge, et al., Analogue signal and image processing with large memristor crossbars, Nat. Electron. 1 (1) (2018) 52–59.
[21] M.L. Gallo, A. Sebastian, G. Cherubini, H. Giefers, E. Eleftheriou, Compressed sensing recovery using computational memory, in: IEEE International Electron Devices Meeting (IEDM), 2017, pp. 28.3.1–28.3.4.
[22] M.L. Gallo, A. Sebastian, R. Mathis, M. Manica, H. Giefers, T. Tuma, et al., Mixed-precision in-memory computing, Nat. Electron. 1 (4) (2018) 246–253.
[23] M.A. Zidan, Y. Jeong, J. Lee, B. Chen, S. Huang, M.J. Kushner, et al., A general memristor-based partial differential equation solver, Nat. Electron. 1 (7) (2018) 411–420.
[24] M.A. Zidan, Y. Jeong, J.H. Shin, C. Du, Z. Zhang, W.D. Lu, Field-programmable crossbar array (FPCA) for reconfigurable computing, IEEE Trans. Multi-Scale Comput. Syst. 4 (2018) 698–710.
[25] H. Nili, G.C. Adam, B. Hoskins, M. Prezioso, J. Kim, M.R. Mahmoodi, et al., Hardware-intrinsic security primitives enabled by analogue state and nonlinear conductance variations in integrated memristors, Nat. Electron. 1 (3) (2018) 197–202.
[26] C. Du, F. Cai, M.A. Zidan, W. Ma, S.H. Lee, W.D. Lu, Reservoir computing using dynamic memristors for temporal information processing, Nat. Commun. 8 (1) (2017) 2204.
[27] S. Gaba, P. Sheridan, J. Zhou, S. Choi, W. Lu, Stochastic memristive devices for computing and neuromorphic applications, Nanoscale 5 (13) (2013) 5872–5878.
[28] J. Borghetti, G.S. Snider, P.J. Kuekes, J.J. Yang, D.R. Stewart, R.S. Williams, 'Memristive' switches enable 'stateful' logic operations via material implication, Nature 464 (7290) (2010) 873–876.
[29] Y.Y. Liauw, Z. Zhang, W. Kim, A.E. Gamal, S.S. Wong, Nonvolatile 3D-FPGA with monolithically stacked RRAM-based configuration memory, in: IEEE International Solid-State Circuits Conference (ISSCC), 2012, pp. 406–408.
[30] S. Datta, N. Shukla, M. Cotter, A. Parihar, A. Raychowdhury, Neuro inspired computing with coupled relaxation oscillators, in: Design Automation Conference (DAC), 2014, pp. 1–6.
[31] Y.V. Pershin, M.D. Ventra, Solving mazes with memristors: a massively-parallel approach, Phys. Rev. E 84 (4) (2011) 046703.
[32] P.-E. Gaillardon, L. Amarú, A. Siemon, E. Linn, R. Waser, A. Chattopadhyay, et al., The programmable logic-in-memory (PLiM) computer, in: Design, Automation & Test in Europe Conference & Exhibition (DATE), 2016, pp. 427–432.
[33] S.H. Jo, K.H. Kim, W.D. Lu, High-density crossbar arrays based on a Si memristive system, Nano Lett. 9 (2) (2009) 870–874.
[34] P.A. Merolla, J.V. Arthur, R. Alvarez-Icaza, A.S. Cassidy, J. Sawada, F. Akopyan, et al., A million spiking-neuron integrated circuit with a, Science 345 (6197) (2014) 668–673.
[35] M. Hu, J.P. Strachan, Z. Li, E.M. Grafals, N. Davila, C.E. Graves, et al., Dot-product engine for neuromorphic computing: programming 1T1M crossbar to accelerate matrix-vector multiplication, in: Design Automation Conference (DAC), 2016, pp. 1–6.
[36] Y. Jeong, M.A. Zidan, W.D. Lu, Parasitic effects analysis in memristor array-based neuromorphic systems, IEEE Trans. Nanotechnol. 17 (1) (2018) 184–193.
[37] E. Neftci, B.U. Pedroni, S. Joshi, M. Al-Shedivat, G. Cauwenberghs, Stochastic synapses enable efficient brain-inspired learning machines, Front. Neurosci. 10 (2016) 241.
[38] S. Yu, P.-Y. Chen, Y. Cao, L. Xia, Y. Wang, H. Wu, Scaling-up resistive synaptic arrays for neuro-inspired architecture: challenges and prospect, in: IEEE International Electron Devices Meeting (IEDM), 2015, pp. 17.3.1–17.3.4.

[39] S.H. Jo, T. Chang, I. Ebong, B.B. Bhadviya, P. Mazumder, W. Lu, Nanoscale memristor device as synapse in neuromorphic systems, Nano Lett. 10 (4) (2010) 1297–1301.
[40] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.
[41] D. Silver, A. Huang, C.J. Maddison, A. Guez, L. Sifre, G.V.D. Driessche, et al., Mastering the game of Go with deep neural networks and tree search, Nature 529 (7587) (2016) 484–489.
[42] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[43] S. Song, K.D. Miller, L.F. Abbott, Competitive Hebbian learning through spike-timing-dependent synaptic plasticity, Nat. Neurosci. 3 (9) (2000) 919–926.
[44] M. Prezioso, F.M. Bayat, B.D. Hoskins, K.K. Likharev, D.B. Strukov, Self-adaptive spike-time-dependent plasticity of metal-oxide memristors, Sci. Rep. 6 (1) (2016). 21331–21331.
[45] B. Linares-Barranco, T. Serrano-Gotarredona, Exploiting memristance in adaptive asynchronous spiking neuromorphic nanotechnology systems, in: IEEE Conference on Nanotechnology (IEEE-NANO), 2009, pp. 601–604.
[46] S. Kim, C. Du, P. Sheridan, W. Ma, S. Choi, W. Lu, Experimental demonstration of a second-order memristor and its ability to biorealistically implement synaptic plasticity, Nano Lett. 15 (3) (2015) 2203–2211.
[47] T. Chang, S. Jo, W. Lu, Short-term memory to long-term memory transition in a nanoscale memristor, ACS Nano 5 (9) (2011) 7669–7676.
[48] Z. Wang, S. Joshi, S. Savel'ev, H. Jiang, R. Midya, P. Lin, et al., Memristors with diffusive dynamics as synaptic emulators for neuromorphic computing, Nat. Mater. 16 (1) (2017) 101–108.
[49] Z. Wang, S. Joshi, S. Savel'ev, W. Song, R. Midya, Y. Li, et al., Fully memristive neural networks for pattern classification with unsupervised learning, Nat. Electron. 1 (2) (2018) 137–145.
[50] S. Kumar, J.P. Strachan, R.S. Williams, Chaotic dynamics in nanoscale NbO2 Mott memristors for analogue computing, Nature 548 (7667) (2017) 318–321.
[51] T. Ohno, T. Hasegawa, T. Tsuruoka, K. Terabe, J.K. Gimzewski, M. Aono, Short-term plasticity and long-term potentiation mimicked in single inorganic synapses, Nat. Mater. 10 (8) (2011) 591–595.
[52] K. Seo, I. Kim, S. Jung, M. Jo, S. Park, J. Park, et al., Analog memory and spike-timing-dependent plasticity characteristics of a nanoscale titanium oxide bilayer resistive switching device, Nanotechnology 22 (25) (2011) 254023.
[53] P. Krzysteczko, J. Münchenberger, M. Schäfers, G. Reiss, A. Thomas, The memristive magnetic tunnel junction as a nanoscopic synapse-neuron system, Adv. Mater. 24 (6) (2012) 762–766.
[54] Z.Q. Wang, H. Xu, X.H. Li, H. Yu, Y. Liu, X.J. Zhu, Synaptic learning and memory functions achieved using oxygen ion migration/diffusion in an amorphous InGaZnO memristor, Adv. Funct. Mater. 22 (13) (2012) 2759–2765.
[55] M.A. Zidan, Y. Jeong, W.D. Lu, Temporal learning using second-order memristors, IEEE Trans. Nanotechnol. 16 (4) (2017) 721–723.
[56] N. Qiao, H. Mostafa, F. Corradi, M. Osswald, F. Stefanini, D. Sumislawska, et al., A reconfigurable on-line learning spiking neuromorphic processor comprising 256 neurons and 128K synapses, Front. Neurosci. 9 (141) (2015). 141.
[57] M. Suri, D. Querlioz, O. Bichler, G. Palma, E. Vianello, D. Vuillaume, et al., Bio-inspired stochastic computing using binary CBRAM synapses, IEEE Trans. Electron. Devices 60 (7) (2013) 2402–2409.

[58] A. Serb, J. Bill, A. Khiat, R. Berdan, R. Legenstein, T. Prodromakis, Unsupervised learning in probabilistic neural networks with multi-state metal-oxide memristive synapses, Nat. Commun. 7 (2016) 12611.
[59] F. Alibart, E. Zamanidoost, D.B. Strukov, Pattern classification by memristive crossbar circuits using ex situ and in situ training, Nat. Commun. 4 (2013) 2072.
[60] A. Georghiades, P. Belhumeur, D. Kriegman, Yale face database, Center for Computational Vision and Control at Yale University, 1997.
[61] C. Li, D. Belkin, D. Belkin, Y. Li, P. Yan, P. Yan, et al., Efficient and self-adaptive in-situ learning in multilayer memristor neural networks, Nat. Commun. 9 (1) (2018) 2385.
[62] Y. LeCun, C. Cortes, C.J. Burges, The MNIST database of handwritten digits. Available at: <http://yann.lecun.com/exdb/mnist/> (accessed 29.01.19).
[63] G. Burr, R. Shelby, C.D. Nolfo, J. Jang, R. Shenoy, P. Narayanan, et al., Experimental demonstration and tolerancing of a large-scale neural network (165 000 synapses) using phase-change memory as the synaptic weight element, in: IEEE International Electron Devices Meeting, 2015, pp. 29.5.1–29.5.4.
[64] A. Shafiee, A. Nag, N. Muralimanohar, R. Balasubramonian, J.P. Strachan, M. Hu, et al., ISAAC: a convolutional neural network accelerator with in-situ analog arithmetic in crossbars, in: ACM/IEEE Annual International Symposium on Computer Architecture (ISCA), 2016, pp. 14–26.
[65] P. Chi, S. Li, C. Xu, T. Zhang, J. Zhao, Y. Liu, et al., PRIME: a novel processing-in-memory architecture for neural network computation in ReRAM-based main memory, in: ACM/IEEE Annual International Symposium on Computer Architecture (ISCA), 2016, pp. 27–39.
[66] L. Song, X. Qian, H. Li, Y. Chen, PipeLayer: a pipelined ReRAM-based accelerator for deep learning, in: IEEE International Symposium on High Performance Computer Architecture (HPCA), 2017, pp. 541–552.
[67] Y. Jiang, J. Kang, X. Wang, RRAM-based parallel computing architecture using k-nearest neighbor classification for pattern recognition, Sci. Rep. 7 (2017) 45233.
[68] S. Yu, B. Gao, Z. Fan, H. Yu, J. Kang, H.-S.P. Wong, A low energy oxide-based electronic synaptic device for neuromorphic visual systems with tolerance to device variation, Adv. Mater. 25 (12) (2013) 1774–1779.
[69] C.J. Rozell, D.H. Johnson, R.G. Baraniuk, B.A. Olshausen, Locally competitive algorithms for sparse approximation, in: IEEE International Conference on Image Processing, 2007, pp. 169–172.
[70] B.A. Olshausen, D.J. Field, Emergence of simple-cell receptive field properties by learning a sparse code for natural images, Naturevolume 381 (6583) (1996) 607–609.
[71] M.A. Zidan, Y. Jeong, W.D. Lu, Hybrid neural network using binary RRAM devices, in: IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH), 2017, pp. 81–82.
[72] P. Sheridan, C. Du, W.D. Lu, Feature extraction using memristor networks, IEEE Trans. Neural Netw. 27 (11) (2016) 2327–2336.
[73] T.D. Sanger, Optimal unsupervised learning in a single-layer linear feedforward neural network, Neural Netw. 2 (6) (1989).
[74] S. Choi, J.H. Shin, J. Lee, P. Sheridan, W.D. Lu, Experimental demonstration of feature extraction and dimensionality reduction using memristor networks, Nano Lett. 17 (5) (2017) 3113–3118.
[75] I. Gupta, A. Serb, A. Khiat, R. Zeitler, S. Vassanelli, T. Prodromakis, Real-time encoding and compression of neuronal spikes by metal-oxide memristors, Nat. Commun. 7 (2016) 12805.

[76] R. Liu, H. Wu, Y. Pang, H. Qian, S. Yu, A highly reliable and tamper-resistant RRAM PUF: design and experimental validation, in: IEEE International Symposium on Hardware Oriented Security and Trust (HOST), 2016, pp. 13–18.

[77] Y. Gao, D.C. Ranasinghe, S.F. Al-Sarawi, O. Kavehei, D. Abbott, Memristive crypto primitive for building highly secure physical unclonable functions, Sci. Rep. 5 (10) (2015). 12785–12785.

[78] J. Kim, T. Ahmed, H. Nili, J. Yang, D.S. Jeong, P. Beckett, et al., A physical unclonable function with redox-based nanoionic resistive memory, IEEE Trans. Inf. Forensics Security 13 (2) (2018) 437–448.

[79] G.S. Rose, N.R. McDonald, L.-K. Yan, B.T. Wysocki, A write-time based memristive PUF for hardware security applications, in: IEEE/ACM International Conference on Computer-Aided Design (ICCAD), 2013, pp. 830–833.

[80] L. Gao, P.Y. Chen, R. Liu, S. Yu, Physical unclonable function exploiting sneak paths in resistive cross-point array, IEEE Trans. Electron. Devices 63 (8) (2016) 3109–3115.

[81] A. Chen, Utilizing the variability of resistive random access memory to implement reconfigurable physical unclonable functions, IEEE Electron. Device Lett. 36 (2) (2015) 138–140.

[82] S. Balatti, S. Ambrogio, Z. Wang, D. Ielmini, True random number generation by variability of resistive switching in oxide-based devices, IEEE J. Emerg. Sel. Top. Circuits Syst. 5 (2) (2015) 214–221.

[83] C.-Y. Huang, W. Shen, Y.-H. Tseng, Y.-C. King, C.-J. Lin, A contact-resistive random-access-memory-based true random number generator, IEEE Electron. Device Lett. 33 (8) (2012) 1108.

[84] H. Jiang, D. Belkin, S.E. Savel'ev, S. Lin, Z. Wang, Y. Li, et al., A novel true random number generator based on a stochastic diffusive memristor, Nat. Commun. 8 (2017) 882.

[85] T. Zhang, M. Yin, C. Xu, X. Lu, X. Sun, Y. Yang, et al., High-speed true random number generation based on paired memristors for security electronics, Nanotechnology 28 (45) (2017) 455202.

[86] H. Jiang, C. Li, R. Zhang, P. Yan, P. Lin, Y. Li, et al., A provable key destruction scheme based on memristive crossbar arrays, Nat. Electron. 1 (10) (2018) 548–554.

[87] A. Chen, Comprehensive assessment of RRAM-based PUF for hardware security applications, in: IEEE International Electron Devices Meeting (IEDM), 2015, pp. 10.7.1–10.7.4.

[88] F. Alibart, L. Gao, B.D. Hoskins, D.B. Strukov, High precision tuning of state for memristive devices by adaptable variation-tolerant algorithm, Nanotechnology 23 (7) (2012) 075201.

[89] E.J. Merced-Grafals, N. Dávila, N. Ge, R.S. Williams, J.P. Strachan, Repeatable, accurate, and high speed multi-level programming of memristor 1T1R arrays for power efficient analog computing applications, Nanotechnology 27 (36) (2016) 365202.

[90] S. Kvatinsky, N. Wald, G. Satat, A. Kolodny, U.C. Weiser, E.G. Friedman, MRL–memristor ratioed logic, in: International Workshop on Cellular Nanoscale Networks and Their Applications (CNNA), 2012, pp. 1–6.

[91] S. Kvatinsky, G. Satat, N. Wald, E.G. Friedman, A. Kolodny, U.C. Weiser, Memristor-based material implication (IMPLY) logic: design principles and methodologies, IEEE Trans. Very Large Scale Integr. (VLSI) Syst. 22 (10) (2014) 2054–2066.

[92] B. Chen, F. Cai, J. Zhou, W. Ma, P. Sheridan, W.D. Lu, Efficient in-memory computing architecture based on crossbar arrays, in: IEEE International Electron Devices Meeting (IEDM), 2015, pp. 17.5.1–17.5.4.

[93] P. Huang, J. Kang, Y. Zhao, S. Chen, R. Han, Z. Zhou, et al., Reconfigurable nonvolatile logic operations in resistance switching crossbar array for large-scale circuits, Adv. Mater. 28 (44) (2016) 9758–9764.

[94] H. Li, T.F. Wu, S. Mitra, H.-P. Wong, Resistive RAM-centric computing: design and modeling methodology, IEEE Trans. Circuits Syst. I: Regul. Pap. 64 (9) (2017) 2263–2273.

[95] K.-H. Kim, S. Gaba, D.C. Wheeler, J. Cruz-Albrecht, T. Hussain, N. Srinivasa, et al., A functional hybrid memristor crossbar-array/CMOS system for data storage and neuromorphic applications, Nano Lett. 12 (1) (2012) 389–395.

[96] B. Chakrabarti, M.A. Lastras-Montaño, G.C. Adam, M. Prezioso, B.J. Hoskins, M. Payvand, et al., A multiply-add engine with monolithically integrated 3D memristor crossbar/CMOS hybrid circuit, Sci. Rep. 7 (1) (2017) 42429.

Chapter 10

Computing with device dynamics

Stephanie Bohaichuk[1] **and Suhas Kumar**[2]
[1]*Stanford University, Stanford, CA, United States,* [2]*Hewlett Packard Labs, Palo Alto, CA, United States*

Although there are many systems and technologies that can produce a wide range of dynamical behavior, memristors are well suited for commercial electronics due to their scalability into a compact size down to a few nanometers [1,2]. Along with the reduced area footprint comes the advantages of reduced latency and lower energy expenditure. To understand how seeking the aid of device dynamics to perform computations could affect the fundamental architecture of a computing system, consider an extremely simplified sketch of the typical hierarchy of a computing system based on the von Neumann architecture (Fig. 10.1A). This consists of levels within the hierarchy going from materials to software, with each level being sufficiently functionally distinct from the others that it can be represented and handled with intermediate abstractions. This enables design of the individual levels

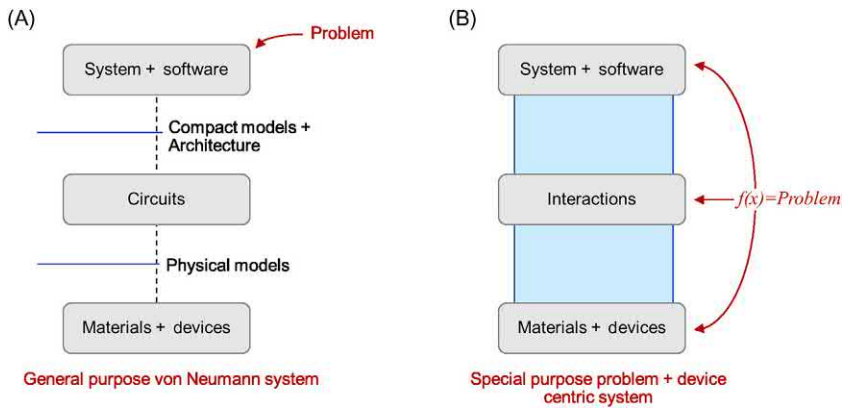


FIGURE 10.1 (A) Schematic of a von Neumann architecture-based general-purpose computing system. (B) Schematic of a non-von Neumann architecture special-purpose computing system.

Memristive Devices for Brain-Inspired Computing. DOI: https://doi.org/10.1016/B978-0-08-102782-0.00010-1

without being affected by most nonfunctional changes in the other levels. For example, the system architect does not (and should not) have to consider the behavior of the particular materials chosen for construction of the devices (e.g., transistors), as long as the devices faithfully reproduce the assigned steady-state function (e.g., encoding of 0s and 1s). Therefore only the highest levels are exposed to the problem that is being solved and computation can be extremely general. On the other hand when the computing function is directly aided by the dynamics of the devices, the architecture becomes less hierarchical (Fig. 10.1B) and more connected. Since the dynamical behavior of the materials within the devices are solving (a part of) the problem of interest, the construction of the other levels within the hierarchy are directly affected by the dynamical behavior of the materials/devices. Although this makes them more challenging to design, these interconnected systems offer the potential to solve certain complex problems efficiently, such as pattern recognition, image or speech processing, correlation detection, or combinatorial optimization.

Nearly all mathematical models of natural computing systems have discovered that any form of complex and emergent properties require highly nonlinear dynamics, such as chaotic behavior [3–5]. This ranges from microscopic edge-of-chaos behavior in individual neurons [6] to macroscopic chaotic dynamics in groups of individuals that are favored during evolutionary natural selection [5]. How dynamical behaviors of devices can be used in a computing system, similar to examples of computation in nature, depends on how intimately the dynamical process is connected to the problem of interest. There are a variety of different materials, device structures, and computational techniques by which device dynamics can be utilized within the hierarchy of a computing system.

10.1 Computation using oscillatory dynamics

One technique that has been gaining interest is to use oscillatory behavior in devices to perform computation, sometimes referred to as an oscillatory neural network. Materials that exhibit a volatile change in resistance can often be configured in such a way as to oscillate between a high-resistance "OFF" state (HRS) and a low-resistance "ON" state (LRS). An example of such a memristive material is a transition metal oxide used in RRAM (e.g., TaO_x, TiO_x) or CBRAM, which can exhibit volatile filament formation due to oxygen vacancy or metal–ion migration when the initial forming step is not done to completion [7]. Another prominent example is a Mott insulator–metal transition material (e.g., VO_2, NbO_2), with an abrupt volatile change in resistance spanning several orders of magnitude that is triggered by a critical temperature or voltage [8]. Any other physical process that can lead to an instability can also be utilized to construct a relaxation oscillator, including superlinear thermal feedback, tunneling, and so on [9,10].

A key indicator that the material has an inherent instability and is useful as an oscillator is the presence of a negative differential resistance (NDR) region in a static I−V curve (Fig. 10.2A). This instability can be accessed by placing the device in series with a resistor or transistor (i.e., a tunable resistor) to reach an appropriate load line that intersects the NDR region (Fig. 10.2A,B). A capacitor is always present in the circuit, intrinsic to the device itself or external (whether intentionally or parasitic). Typical oscillatory dynamics of such a circuit are illustrated in Fig. 10.2C−E. As the

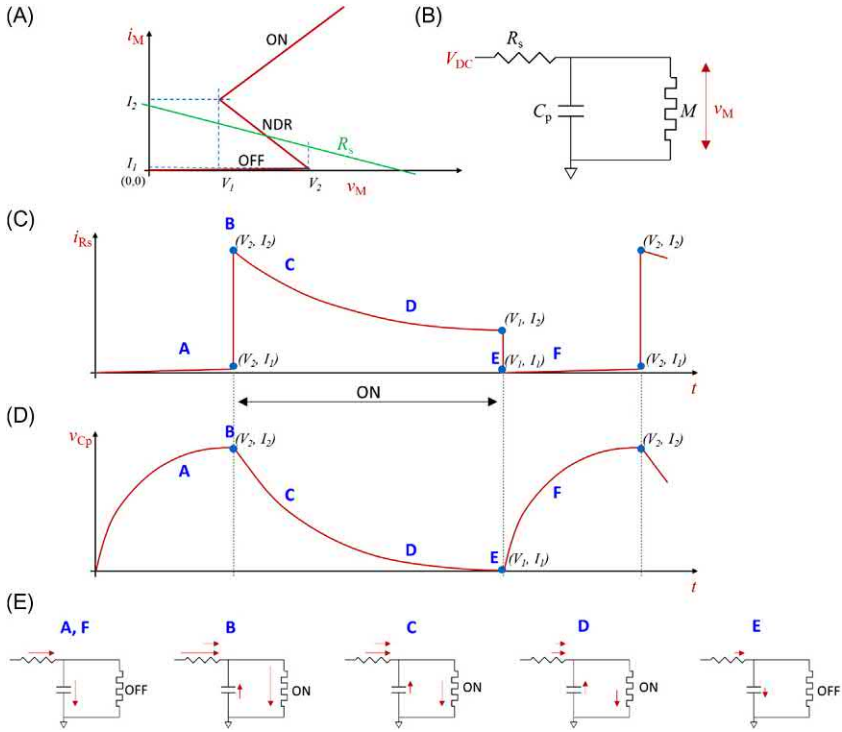


FIGURE 10.2 (A) Quasi-static current−voltage behavior of a memristive device exhibiting NDR. Green line illustrates a load line representing the series resistor R_s. (B) Schematic of a relaxation oscillator with the NDR device marked M, biased with V_{DC} to be consistent with the load line in (A) (i.e., to fix the operating point within the NDR region). (C)−(D) Illustration of the working of a relaxation oscillator by displaying the current through R_s and the voltage across the parallel capacitor C_p. (E) The waveform in (C) is illustrated with typical current flow in the circuit in (B) at different time steps (A−F). The labels from (B) are dropped for simplicity of presentation. The arrows in dark red indicate direction of current flow. Within the ON duration, the current through R_s consists of a sum of two currents (one through C_p and another through M), as indicated. Outside the ON duration, the current through R_s consists of only one current (through C_p), as indicated (provided the LRS state is well insulating). The length of the arrows also indicate the relative magnitudes of the currents. *Partly reproduced from Bohaichuk, S. M., et al., Fast Spiking of a Mott VO_2−Carbon Nanotube Composite Device, Nano Lett. 19, 6751−6755, doi: 10.1021/acs.nanolett.9b01554 (2019).*

capacitor gradually charges, the voltage across the NDR element also increases along with a gradual increase in the current through the series resistor. When this voltage reaches a critical voltage (V_2), the transition to LRS is abruptly triggered and a large current suddenly flows through both the memristor and the series resistor. The creation of a highly conductive path within the NDR element causes the voltage across it to fall and the capacitor to discharge, abruptly raising the current, with the voltage now mainly dropping across the series resistor instead (bearing some similarity to current overshoot in RRAM [11]). As the capacitor discharge slows and the memristor sees a lower field and power (especially with larger R_s), it gradually becomes less conductive. Once a threshold (V_1) is reached it abruptly returns to HRS and restricts the current flow. The cycle now repeats again, with the voltage rising across the now resistive device until the switch to LRS is triggered. This process repeats indefinitely creating oscillating signals. The electrical RC time constant determines the frequency of oscillations, which typically ranges from a few kilohertz to a few hundred megahertz and can be tuned via the series resistance or applied voltage [7,12].

Such schemes are not limited to electrically memristive devices, but also extend to any device with an instability. This includes mechanical memristive systems such as cantilever-based capacitive systems [13,14]. Oscillatory computations have even been proposed in complementary metal–oxide–semiconductor (CMOS) but require considerable area and resources. Furthermore, coupling of oscillators into networks need not be capacitive, but can be done optically (e.g., a thermo-optical effect via laser heating) [15] or magnetically as in spin torque oscillators (STOs) [16,17]. In an STO, spin polarized current is injected into a ferromagnetic layer that results in spin-transfer torque (the magnetic field of the layer tends to align with the injected spin). This can cause destabilization of the layer's magnetization under certain conditions, generating magnetic precession and spin waves. The oscillations in magnetization can be electrically detected using giant magnetoresistance (if two layers are magnetized parallel then a larger current will flow than if they are antiparallel).

A single oscillating device could be used as a component in a neural network: a neuristor (artificial neuron) that generates fast spikes (action potentials) which could be used as input events or used to adjust synaptic weights with high precision [18]. However, computation can also be done using a network of coupled neuron-like oscillators (Fig. 10.3A). If two oscillators are capacitively coupled together (at the node between R_s and M in Fig. 10.2B), then they will synchronize or desynchronize over time depending on the conditions, similar to a spring–mass system. If their individual frequencies are similar or the coupling is strong then the oscillations will synchronize to the same frequency, though they are generally out of phase (one memristor is in HRS while the other is in LRS) as shown in Fig. 10.3B. If very dissimilar in frequency, then the oscillators will drift relative to one another and they will

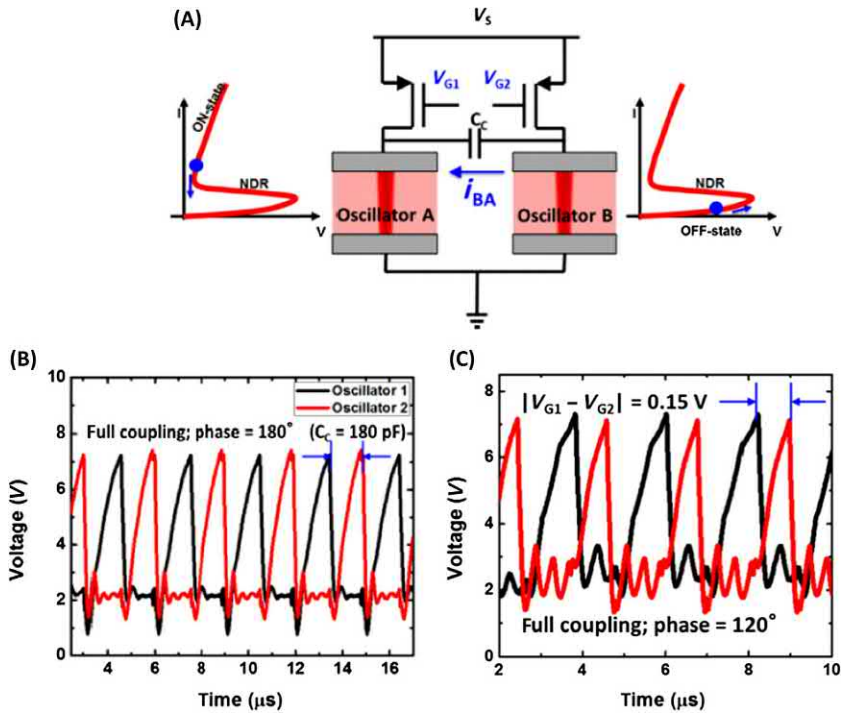


FIGURE 10.3 Dynamics of coupled oscillators. (A) Schematic of two relaxation oscillators capacitively coupled via C_c, with transistors used to tune each oscillator's frequency. (B) A scenario where the oscillators are well coupled (synchronized to the same frequency) but are 180° out of phase. This corresponds to one oscillator being ON and the other being OFF (A). (C) A scenario where the oscillators are well coupled, but a 120° phase shift is created by a difference in individual oscillator frequencies via the series resistances (via V_{G1} and V_{G2}). *Reproduced from Sharma, A.A., Bain, J.A. & Weldon, J.A. Phase Coupling and Control of Oxide-Based Oscillators for Neuromorphic Computing. IEEE J. Exploratory Solid-State Computational Devices Circuits 1, 58-66, doi:10.1109/jxcdc.2015.2448417 (2015).*

not be phase locked. The degree of synchronization and relative phases of the oscillators can be manipulated to do computation via R_s (changing the individual resonant frequencies) or C_c (the degree of coupling) [7,8,19,20]. An example is shown in Fig. 10.3C.

This property can be used to create an analog comparator that takes the difference norm between two inputs [21]. A transistor is used in series with each memristor, and each input is sent to a transistor gate. If the input values are similar, then the series resistance and therefore the resonant frequencies of the oscillators are well matched, leading to synchronization. Synchronization is measured by taking a time-averaged exclusive "or" operation (XOR) of the analog outputs on each side of the coupling capacitor after thresholding to digital values. The higher the XOR value the more dissimilar the inputs are, as

the two oscillator nodes spend more time out of sync. This can be used for image/pattern recognition by comparing each pixel to a stored image. Using a winner-take-all method, if a threshold number of pixels match then the images are said to match. With this method and all the required peripheral circuitry (e.g., the XOR in CMOS), a 20× reduction in power over an equivalent CMOS circuit was obtained [21]. This could potentially be extended to other image or data manipulation tasks such as edge detection.

Pattern recognition or correlation detection can also be performed using multiple relaxation oscillators coupled together. Computation can be done in a frequency-shift keying (FSK) regime, where patterns are encoded as oscillator frequencies, or in a phase-shift keying (PSK) regime, where patterns are identified via the relative phase between the oscillators [8]. In FSK, the frequency of each oscillator is intentionally shifted, for example, in proportion to the difference between each component/pixel of an input vector/image and a stored one. Averaging over the oscillators and summing over time will yield a large value if the oscillators have similar frequencies and thus synchronize in phase, corresponding to a matching pattern. In PSK, all oscillators are set with the same frequency but the coupling capacitances are varied. For example, these could be set according to a Hebbian learning rulewhich multiplies stored and input components resulting in a large value if they are correlated. If the phase difference is small (the coupling capacitance was large), then the pattern matches.

As an example implementation of an oscillatory network, consider the solution to the vertex coloring problem demonstrated using coupled relaxation oscillators [22–24]. Graph coloring has a similar form to many other problems of wide interest, including Sudoku, node classification, scheduling, register allocation, bandwidth allocation, and others. The goal of the problem is to find the minimum number of colors required to have each vertex colored, with adjacent vertices (directly connected by an edge) being a different color. Mathematically this problem can be solved using spectral algorithms that compute the eigenvector solutions of the adjacency matrix, in which each element of the matrix describes whether or not a pair of nodes is adjacent. Physically a network of relaxation oscillators can emulate this type of algorithm through the system's time evolution, with the oscillator connections representing the adjacency matrix and the dynamics solving for the eigenvectors. The relaxation oscillators used to solve this problem exploited the NDR instabilities within VO_2-based memristive devices along with an energy storing capacitor (Fig. 10.4A) [10]. An oscillator network can be constructed such that the connections among the oscillators represent edges of a graph, and the oscillators themselves represent the vertices. The coupled oscillators end up synchronizing to the same frequency but are out of phase with one another. The phases cluster in groups that can each be assigned a color, and this can be used to produce an optimal solution to the vertex coloring or equivalent color sorting problem (Fig. 10.4).

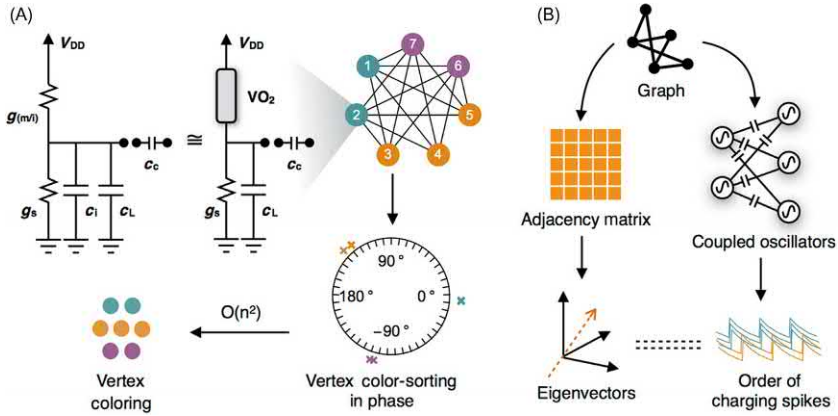


FIGURE 10.4 Schematic and operation of the coupled oscillator network to solve the vertex coloring problem. (A) Relaxation oscillators are constructed using the insulator−metal transition material VO_2, and capacitively coupled (C_c). Each oscillator represents a graph node, and each capacitor an edge. The oscillators will synchronize but be shifted in phase from one another, clustered into groups each of which can be assigned a color. (B) Mathematically, the connections between the coupled oscillators correspond to a physical implementation of the adjacency matrix of the graph, and the resulting phase ordering of spiking corresponds to the eigenvector solutions. *Reproduced from Parihar, A., Shukla, N., Jerry, M., Datta, S. & Raychowdhury, A. Vertex coloring of graphs via phase dynamics of coupled oscillatory networks. Sci. Rep. 7, 911, doi:10.1038/s41598-017-00825-1 (2017).*

Vertex coloring is thought to be an NP-hard (non-deterministic polynomial-resource) problem, so there is expected to be no polynomial-resource solution to the problem. In other words the resources required to solve such problems increase exponentially as the problem size increases linearly (typically at least 2^n operations). However, it was shown that the total number of operations (or corresponding time) required to arrive at this solution using an oscillator network was a polynomial function of the size of the graph (n^2 for a problem size of n). The fact that this solution is obtained in polynomial time may suggest that there could be an exponential expenditure of energy with a linear increase in graph size. It is likely that due to the complex nature of a large system's dynamics (with each oscillator being represented by at least two state equations, and each oscillator coupled to many others), the ensuing group dynamics will be chaotic in nature, with exponential excursions in currents and/or voltages leading to exponential energy expenditure. Although much recent progress has been made, the theoretical and experimental dynamics of such a network remain to be fully understood, especially at larger scales.

The problem here was encoded onto the hardware at the level of the devices and circuit. Hence a given hardware can only solve one specific instance of the graph coloring problem. Thus this system is not reprogrammable with the same ease at which most von Neumann systems are

programmable. Programming such a system would require rewiring of the entire network in accordance with a new problem. This provision limits the system to solving only different instances of the vertex coloring problem. Alongside this issue, actual implementation in hardware is a challenge, wherein inferring a solution requires measurement of many oscillating signals and calculating their phases and phase grouping. These observations fit into the general rule of thumb that programming closer to the hardware might provide highly attractive performance enhancements (such as polynomial-time solutions to NP-hard problems), but reduces programmability (or hardware feasibility) of the system. The case of vertex coloring with oscillator networks is an extreme example, which leads to dramatic speedups of solutions, but renders the system non-reprogrammable.

10.2 Control of memristor resistance

Another use of memristive devices in computation is by control of their crystallization or filament formation, enabling tunability of either their nonvolatile resistance levels or volatile dynamics, which can used to detect correlations in input signals or solve equations [25–28]. Most physical processes that enable memristive binary nonvolatile storage (two resistance states) also enable multilevel storage. For example in phase-change memory the resistance level depends on the volume of the crystalline region embedded within an amorphous region (or vice versa). If we start with a finite volume of an amorphous region, the conductance can be incrementally increased in small steps by injecting small amounts of energy sufficient to progressively crystallize the region. This can be represented within self-consistent solutions to the dynamical equations in Fig. 10.5, which rely on material-dependent thermodynamics. Similarly many oxides used in RRAM also show multilevel storage capability, with energy injection strengthening the formation of a filament (i.e., changing the vacancy/ion concentration, filament volume, and degree of connectivity across the electrodes). Depending on the material and operating conditions, input energy can produce a reduction in resistance that is either nonvolatile or decays with a characteristic time constant. Although detrimental to classic von Neumann architectures, well-characterized resistance drift could be useful in systems for processing dynamic or temporal information, where the outputs need to depend on the rate and relative timing of the inputs [29]. This physically implements a functionality similar to long-term potentiation or depression in neurons, where neurons display changes in activity based on whether or not spikes have recently been received by it.

Injection of limited energy can be achieved by using voltage pulses of controllable amplitude and duration. In principle such pulses could even be produced by using the rapid spikes of an oscillating memristor. In the case of correlation detection described below it is beneficial to have the conductance change linearly with pulse number to give a clear and even distinction

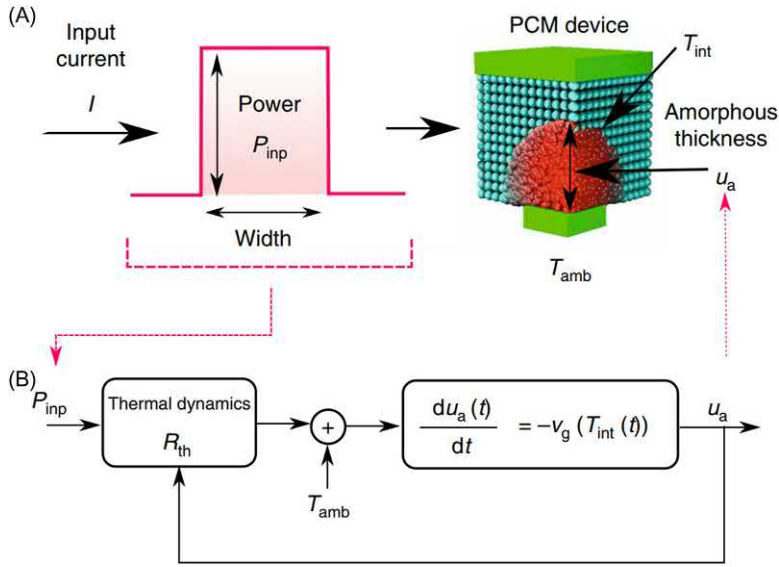


FIGURE 10.5 Schematic and dynamics of a crystallization-based memristor. (A) A low-energy pulse can be used to progressively crystallize part of a phase-change memory device. Similar dynamics (i.e., a gradual change in resistance) can be reproduced using other physical phenomena. (B) The thermal environment and input pulse characteristics determine the final output conductance via crystallization dynamics. *Reproduced from Sebastian, A. et al. Temporal correlation detection using computational phase-change memory. Nat. Commun. 8, 1115, doi:10.1038/s41467-017-0148-9 (2017).*

between levels. However in some systems it is better to have the conductance be a highly nonlinear function of pulse number [30]. For example this could allow the memristor to behave similarly to an integrate-and-fire neuron, which only switches to low resistance after enough energy has been accumulated over several pulses.

10.3 Correlation detection and nonlinear solvers

Now suppose that we have different incoming temporally varying signals (consider discrete-time binary signals for simplicity). If we seek to determine the correlation among any of the signals, one way to do it is by counting the number of times the signals under investigation together produce a high value. The idea of gradually changing the conductance of memristors can be used to implement this (Fig. 10.6). Distinct memristors are assigned to every signal, and at every time step there is a determination of the number of signals that contain a high value, the sum of which sets the amplitude or duration of a pulse applied to memristors that had a high signal. Therefore those memristors that correspond to signals that are highly correlated will end up

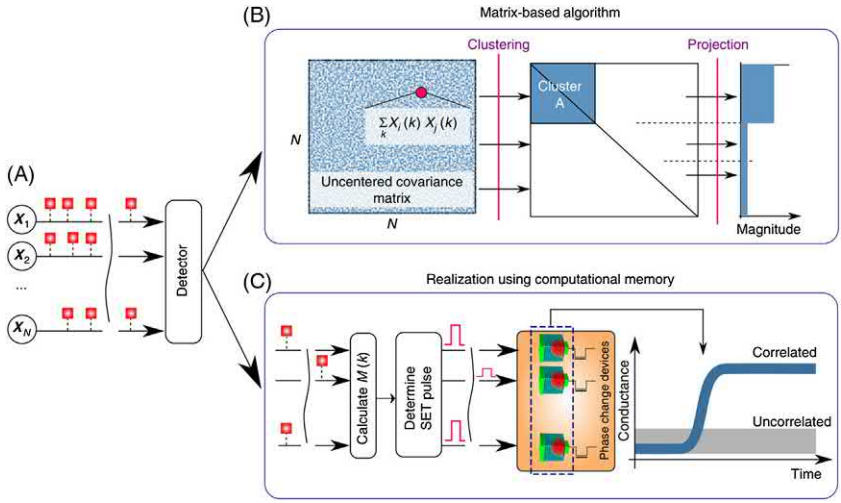


FIGURE 10.6 Comparison of a crystallization dynamics-based correlation detection system (lower panel) with a covariance calculation technique (upper panel). The dynamics-based system functions by changing the input pulse width to phase-change devices according to each input and accumulating correlation over time. *Reproduced from Sebastian, A. et al. Temporal correlation detection using computational phase-change memory. Nat. Commun. 8, 1115, doi:10.1038/s41467-017-0148-9 (2017).*

being exposed to higher energies (and hence have higher conductance) than those that correspond to uncorrelated signals. A mere measurement of the conductance of the memristors and a plot of the conductance distribution will reveal a set of correlated signals. This technique can be extended to detect multiple correlated sets of signals and continuous-time analog signals as well. Moreover apart from the example of using crystallization dynamics, identical results can be obtained using most other physical memristive processes [13,14].

Correlation detection has significant applications in many areas including classification, pattern recognition, weather prediction, etc. The system discussed in the preceding paragraph exploits the dynamics of memristive devices embedded within the network, and therefore the network's description is intimately tied to the dynamics of the particular device used. However, unlike the case of solving the graph coloring problem using couple oscillators, this network's construction does not represent any specific instance of the problem. Therefore the system can solve any arbitrary instance of a correlation problem. Thus this is a more generic solver compared with the case of coupled oscillators, but is still not a general purpose computing system. The correlation detection system based on crystallization dynamics provided a speed-up of about 200 times over the best GPU-based processors for detecting correlations in more than a million datasets. Recall

that the problem and instance-specific network of coupled oscillators used for vertex coloring provided a speed-up of $2^n/n^2$ times, which is several orders of magnitude for large problem sizes (but which may be throttled or even limited by circuit overheads of scaling).

Reservoir computing systems can also be used to detect correlations, especially temporal, and can require less training than other recurrent neural network approaches. Generally, weights in neural networks are used to describe the response strength or the connections between each neuron, and must be trained with an input data set by updating the weights to minimize the error between the actual and expected output. This training can be a computationally and time intensive task. However by first sending the input through a fixed reservoir that performs a nonlinear transformation on the data, dictated directly by the hardware (e.g., an array of connected nonlinear memristors), features in some nonlinear problems can be more easily extracted (Fig. 10.7A–C). Rather than training weights within the entire network, only weights associated with reading out the reservoir state need to be trained (not within the reservoir itself), which can use a simple algorithm such as linear regression. This can make reservoir computing systems faster and more efficient, especially to train [29,31].

When time-dependent behavior of devices in the reservoir is utilized, reservoir computing is suited for detecting patterns in time-dependent data. If devices have short-term memory, with a steady decay in conductance between input pulses, then the final conductance state of the reservoir will depend not only on the number of input pulses but also on their relative order and timing, allowing for an analysis of temporal data. Alternatively devices with short-term memory could be utilized to detect correlations in a nontemporal data set more efficiently. Normally each element in the data set needs a device to process it, where each device adds complexity to the reservoir and additional weights to be trained in the readout. Instead, by converting sections of data to pulse trains, the data can be processed by the devices sequentially as if it were temporal, while still detecting patterns. This conversion from spatial to temporal processing can reduce the memristor array size and complexity, with fewer weights to train. However this can come at the cost of increased processing time and a dependence on the choice of input pulse rate, which is strongly tied to the device dynamics. The time constants of short-term memory devices must be carefully characterized and designed relative to the input rate and readout times (if the time constant is too slow then it loses relative timing information and only counts pulses, but if too fast then the conductance will rapidly decay to the ground state).

While a classification system may be able to tolerate some initial variability (which could be dealt with during training) or slight drift in the time constants or conductance of all devices over time, significant cycle-to-cycle variation or drift in devices relative to one another may hinder accuracy. Large systems and datasets may also pose challenges, where it can be

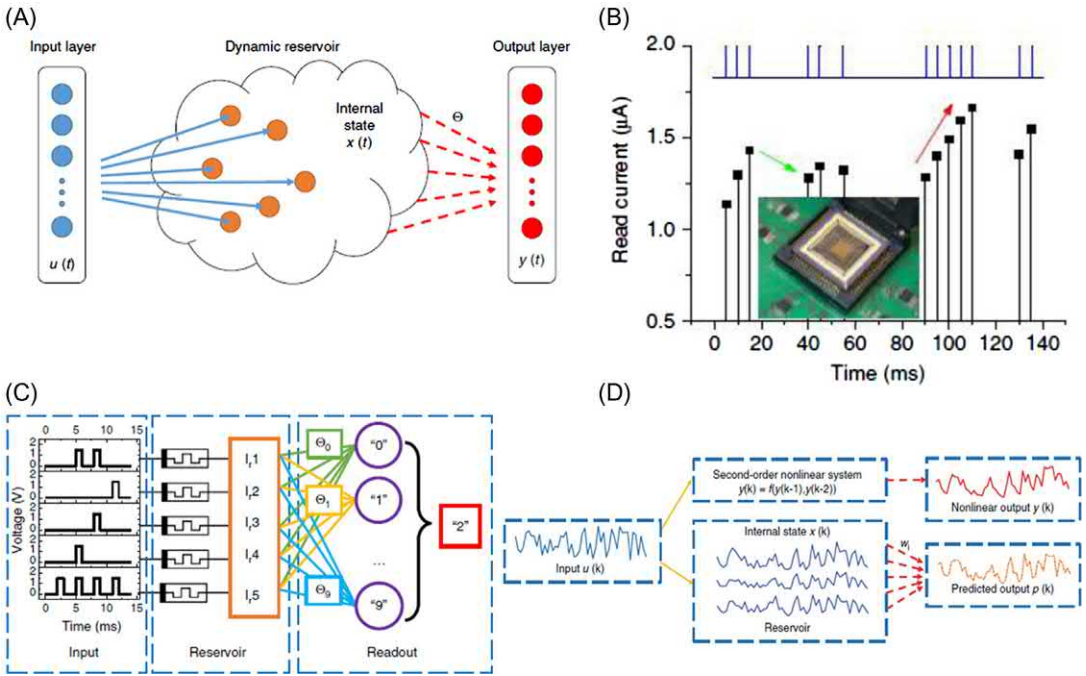


FIGURE 10.7 (A) A general reservoir computing system performs a fixed nonlinear transformation on input data, and a readout layer Θ is trained to convert the internal reservoir state $x(t)$ into an output $y(t)$. (B) Memristor devices with short-term memory can be used in the reservoir to transform input pulses (blue) into temporally correlated resistance changes (black). (C) A specific implementation of reservoir computation, with inputs as voltage pulses, a memristor cross-bar array as the reservoir, and a small neural network as the readout function. (D) The reservoir can also be used to compute nonlinear dynamics when an exact equation for the system is unknown. *Reproduced from Du, C. et al. Reservoir computing using dynamic memristors for temporal information processing. Nat. Commun. 8, 2204, doi:10.1038/s41467-017-02337-y (2017).*

increasingly difficult to distinguish two similar states. For example if two large datasets differ only slightly (e.g. two nodes are connected differently in a large graph), then the difference in memristor resistance or oscillator phase between the two states may not be accurately detected. This might limit data processing to small subsections of the data set at a time, presenting challenges to correlation detection on a large scale or in distant regions of an image. Further study is still needed to understand the scaling up of these systems to large networks, including the effects of variation and drift.

Because of the reservoir devices' nonlinearity, such a system can also be used to solve nonlinear equations (Fig. 10.7D) [29]. Often problems in mechanics, electromagnetics, circuits, and many other disciplines require predictions of a nonlinear system's behavior when in practice the exact equations describing the system are unknown. Using a sample set of data from the system being modeled, the readout weights can be trained such that the reservoir computing system can map time evolving input data into a predicted output without knowing the analytical expression of the nonlinear transfer function.

10.4 Optimization using Hopfield networks and chaotic devices

Finally volatile and nonvolatile memristors can be used in Hopfield networks, which make use of chaotic device dynamics originating from volatile memristors to accelerate solutions to many types of optimization problems [2]. The idea was based on the premise of being able to generate controlled chaos from a single electronic device, which had not been demonstrated before. It was recently shown that a relaxation oscillator constructed with nanoscale memristors can exhibit chaos if the memristors are made sufficiently small in size and contain sufficient nonlinearity (Fig. 10.8) [2]. As the memristors were made smaller, they became more susceptible to thermal fluctuations owing to the decreased thermal mass. The thermal fluctuations could couple with the temperature state equation and get amplified by the nonlinear transport when operated in a region of instability (namely, NDR). This drove the system into chaos, which could be used in system-level applications as described below.

A Hopfield network (Fig. 10.9A) stores a matrix representation of any problem that can be represented as a Hamiltonian matrix, which includes most optimization problems, making it extremely general purpose. The main algorithm of a Hopfield network is based on minimization of the energy E calculated using the Hamiltonian (feedback weights) matrix w and the solutions matrix s, the global minimum of which should correspond to the most optimal solution to the problem (Fig. 10.9B):

$$E = -\frac{1}{2}\sum_i \sum_{j \neq i} s_{i,j} \sum_k \sum_l s_{k,l} w_{(i,j),(k,l)} + \sum_i \sum_j s_{i,j}\theta \tag{10.1}$$

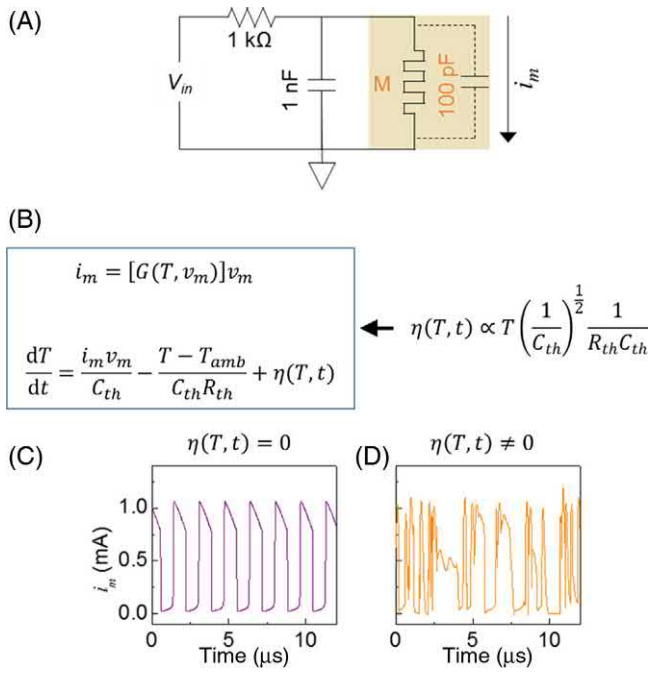


FIGURE 10.8 Schematic and origin of chaotic dynamics in a memristor-based relaxation oscillator circuit. *Partly reproduced from Kumar, S., Strachan, J.P., Williams, R.S. Chaotic dynamics in nanoscale NbO_2 Mott memristors for analogue computing. Nature 548, 318–321, doi:10.1038/nature23307 (2017).*

The solution matrix is typically a set of binary states (usually +1 and 0 or −1) describing the answer to the problem; for example, this can describe whether or not a node has been visited by a traveling salesman, whether a pixel is black or white, whether a magnet spin is up or down, etc. The weight matrix describes the strength of the connections between each state, and the sign can dictate whether a state is attracted or repelled toward a state or condition. The expression to calculate weights is dictated by the specifics of the problem being solved, which can include objectives and constraints. For example in the traveling salesmen problem, this could include that cities are only visited once, that all cities must be visited, and that the traveled distance should be minimized. θ can be included to represent an external input or to implement the commonly used weighted feedback in Eq. (10.2). The network iterates to minimize the energy and determines the final solution, using feedback based on the weights and solution at the current iteration. The states s are usually described using a threshold or sigmoidal function, such as the following equation with

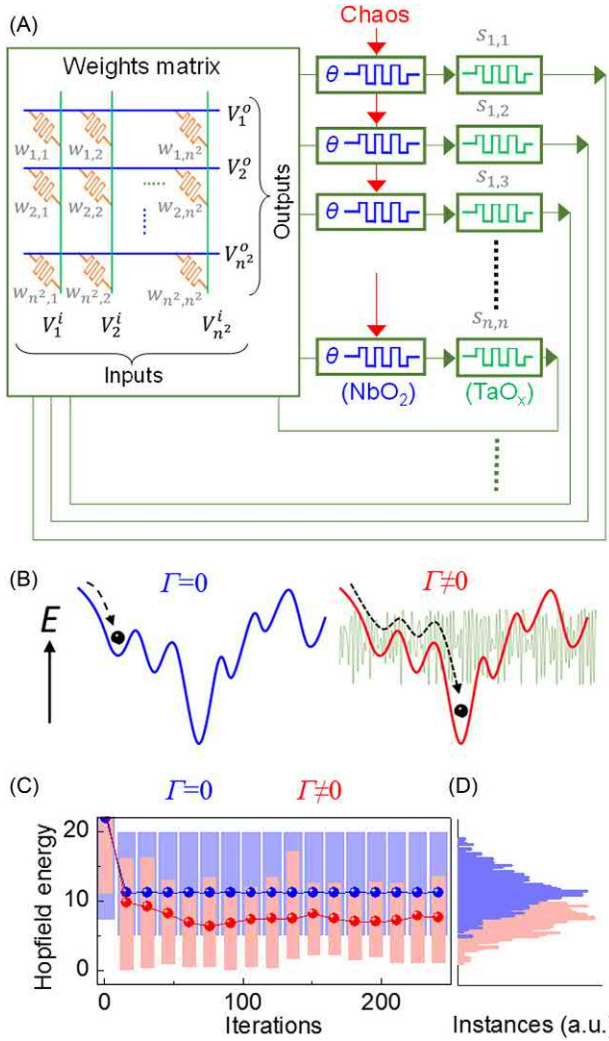


FIGURE 10.9 (A) Layout and operation of a chaos-driven Hopfield network. Weighted feedback is used to minimize the energy of the system and find a solution. (B) Chaotic behavior in memristors (gray) can be used to escape a local minimum that Hopfield networks often get stuck in. (C) Using chaos results in a lower energy and more optimal solution. *Reproduced from Kumar, S., Strachan, J.P., Williams, R.S. Chaotic dynamics in nanoscale NbO_2 Mott memristors for analogue computing. Nature 548, 318–321, doi:10.1038/nature23307 (2017).*

threshold value θ:

$$s_{i,j} = \begin{cases} 1 & if\ \theta < \sum_k \sum_l w_{(i,j),(k,l)} s_{k,l} \\ -1 & if\ \theta \geq \sum_k \sum_l w_{(i,j),(k,l)} s_{k,l} \end{cases} \tag{10.2}$$

Some easy problems have only one energy minimum, and hence it is easy for such a network to solve such a problem. But most real-world optimizations problems are sufficiently complex that they contain several energy

minima, while only the global minimum corresponds to a meaningful and optimal solution. The exact working of the Hopfield network and similar neural networks (such as the Boltzmann machine) is not crucial to the narrative of this chapter, but the interested reader can obtain more details elsewhere [2,32]. It suffices to state that one of the biggest problems with the Hopfield network that prevents it from being widely adopted is that it can only perform energy minimization (known as gradient descent) that often gets stuck in the local minima (wrong and/or invalid solutions) of complicated problems. A trick to get around this problem is to introduce some disturbance, typically in the form of uncorrelated functions such as stochasticity, chaos, pseudorandom sequences, etc., which aid the system in jumping out of the local minima when they are stuck (Fig. 10.9B). Quantum annealing schemes have made use of quantum fluctuations to achieve this. It was recently discovered that a single nanoscale volatile memristor can produce controlled chaos due to interactions with ambient thermal fluctuations at room temperature [2]. These thermal fluctuations produce a chaotic shift in the threshold switching voltage of the memristors, which can be used to slightly disturb the network and aid in energy minimization.

The various stages of such a Hopfield network can all be physically implemented using memristors. The binary nonvolatile storage of highly nonlinear memristors can be used to store the solution matrix s, while nonvolatile memristors with tunable resistance levels (less nonlinear) can be used to store the weight matrix w. Finally volatile memristors can be used to implement the weighted feedback of Eq. (10.2), with the threshold θ implemented as the threshold switching voltage. The small fluctuations in the threshold switching voltage shift θ chaotically at each iteration, enough to help prevent being stuck in a local minimum but not so much as to prevent convergence. This accelerates the solutions by at least a factor of 10 compared with the best CMOS-based systems [33,34]. The Hopfield network is very general and can solve almost any problem that can be represented by any constraints or optimizations, including all NP classes of problems.

10.5 Conclusions

It is notable here that the examples chosen so far have gotten progressively more general purpose (the coupled oscillator system being the least programmable and the Hopfield network being the most programmable), while the speed-ups offered by these systems over competing CMOS-based approaches have become progressively lower (though a speed-up of >10 is still impressive for a viable technology).

By now we have established that device dynamics can be used to solve problems in completely new ways or to accelerate solvers in known computing systems. Theoretical approaches have predicted chaos-driven polynomial-time solutions to all NP-hard problems [35]. There are even

some experimental claims on being able to solve any NP-hard problem using only polynomial resources using memristor dynamics, which remain controversial [36]. Encoding any part of the problem into the dynamics of the devices used for constructing the system invariably affects the architectural aspects of the system. The research community is presently exploring the use of new architectures such as analog neural networks and non-von Neumann systems to perform linear algebra better than digital CMOS-based systems. Dealing with extreme degrees of nonlinearity and using device dynamics to directly solve problems is only at the horizon of computing research. There have been several other approaches in this domain that are not covered in this chapter, which is meant to be an introduction to the concept rather than an exhaustive literature survey. The surge in interest in this futuristic concept, backed by solid system-level demonstrations of accelerating solutions to several problems, makes computing with memristor dynamics a promising and a nearly imperative future computing primitive.

References

[1] D.B. Strukov, G.S. Snider, D.R. Stewart, R.S. Williams, The missing memristor found, Nature 453 (2008) 80–83. Available from: https://doi.org/10.1038/nature06932.

[2] S. Kumar, J.P. Strachan, R.S. Williams, Chaotic dynamics in nanoscale NbO_2 Mott memristors for analogue computing, Nature 548 (2017) 318–321. Available from: https://doi.org/10.1038/nature23307.

[3] L.O. Chua, Local activity is the origin of complexity, Int. J. Bifurc. Chaos 15 (2005) 3435–3456. Available from: https://doi.org/10.1142/S0218127405014337.

[4] J. Whitfield, Complex systems: order out of chaos, Nature 436 (2005) 905–907. Available from: https://doi.org/10.1038/436905a.

[5] R. Lewin, Complexity: Life at the Edge of Chaos, University of Chicago Press, 1999.

[6] L. Chua, V. Sbitnev, H. Kim, Neurons are poised near the edge of chaos, Int. J. Bifurc. Chaos 22 (2012) 1250098. Available from: https://doi.org/10.1142/S0218127412500988.

[7] A.A. Sharma, J.A. Bain, J.A. Weldon, Phase coupling and control of oxide-based oscillators for neuromorphic computing, IEEE J. Exploratory Solid-State Computational Devices Circuits 1 (2015) 58–66. Available from: https://doi.org/10.1109/jxcdc.2015.2448417.

[8] D.E. Nikonov, et al., Coupled-oscillator associative memory array operation for pattern recognition, IEEE J. Exploratory Solid-State Computational Devices Circuits 1 (2015) 85–93. Available from: https://doi.org/10.1109/jxcdc.2015.2504049.

[9] G.A. Gibson, et al., An accurate locally active memristor model for S-type negative differential resistance in NbO_x, Appl. Phys. Lett. 108 (2016) 023505. Available from: https://doi.org/10.1063/1.4939913.

[10] M.D. Pickett, R.S. Williams, Sub-100 fJ and sub-nanosecond thermally driven threshold switching in niobium oxide crosspoint nanodevices, Nanotechnology 23 (2012) 215202. Available from: https://doi.org/10.1088/0957-4484/23/21/215202.

[11] H.J. Wan, et al., In situ observation of compliance-current overshoot and its effect on resistive switching, IEEE Electron. Device Lett. 31 (2010) 246–248. Available from: https://doi.org/10.1109/led.2009.2039694.

[12] N. Shukla, et al., Synchronized charge oscillations in correlated electron systems, Sci. Rep. 4 (2014) 4964. Available from: https://doi.org/10.1038/srep04964.
[13] S. Kumar, et al., Direct observation of localized radial oxygen migration in functioning tantalum oxide memristors, Adv. Mater. 28 (2016) 2772–2776. Available from: https://doi.org/10.1002/adma.201505435.
[14] J. Zhang, et al., Thermally induced crystallization in NbO_2 thin films, Sci. Rep. 6 (2016) 34294. Available from: https://doi.org/10.1038/srep3429.
[15] M. Zhang, et al., Synchronization of micromechanical oscillators using light, Phys. Rev. Lett. 109 (2012) 233906. Available from: https://doi.org/10.1103/PhysRevLett.109.233906.
[16] K. Yogendra, D. Fan, K. Roy, Coupled spin torque nano oscillators for low power neural computation, IEEE Trans. Magnetics 51 (2015) 1–9. Available from: https://doi.org/10.1109/tmag.2015.2443042.
[17] S. Kaka, et al., Mutual phase-locking of microwave spin torque nano-oscillators, Nature 437 (2005) 389–392. Available from: https://doi.org/10.1038/nature04035.
[18] M.D. Pickett, G. Medeiros-Ribeiro, R.S. Williams, A scalable neuristor built with Mott memristors, Nat. Mater. 12 (2013) 114–117. Available from: https://doi.org/10.1038/nmat3510.
[19] A. Parihar, N. Shukla, M. Jerry, S. Datta, A. Raychowdhury, Computing with dynamical systems based on insulator-metal-transition oscillators, Nanophotonics 6 (2017) 601–611. Available from: https://doi.org/10.1515/nanoph-2016-0144.
[20] T.C. Jackson, A.A. Sharma, J.A. Bain, J.A. Weldon, L. Pileggi, Oscillatory neural networks based on TMO nano-oscillators and multi-level RRAM cells, IEEE J. Emerg. Sel. Top. Circuits Syst. 5 (2015) 230–241. Available from: https://doi.org/10.1109/jetcas.2015.2433551.
[21] Shukla, N. *et al*. *IEEE International Electron Devices Meeting* (San Francisco, CA, 2014).
[22] J. Wu, L. Jiao, R. Li, W. Chen, Clustering dynamics of nonlinear oscillator network: Application to graph coloring problem, Phys. D: Nonlinear Phenom. 240 (2011) 1972–1978. Available from: https://doi.org/10.1016/j.physd.2011.09.010.
[23] C.W. Wu, Graph coloring via synchronization of coupled oscillators, IEEE Trans. Circuits Syst. I: Fundamental Theory Appl. 45 (1998) 974–978. Available from: https://doi.org/10.1109/81.721263.
[24] A. Parihar, N. Shukla, M. Jerry, S. Datta, A. Raychowdhury, Vertex coloring of graphs via phase dynamics of coupled oscillatory networks, Sci. Rep. 7 (2017) 911. Available from: https://doi.org/10.1038/s41598-017-00825-1.
[25] A. Sebastian, et al., Temporal correlation detection using computational phase-change memory, Nat. Commun. 8 (2017) 1115. Available from: https://doi.org/10.1038/s41467-017-0148-9.
[26] M.A. Zidan, et al., A general memristor-based partial differential equation solver, Nat. Electron. 1 (2018) 411–420. Available from: https://doi.org/10.1038/s41928-018-0100-6.
[27] T. Tuma, M. Le Gallo, A. Sebastian, E. Eleftheriou, Detecting correlations using phase-change neurons and synapses, IEEE Electron. Device Lett. 37 (2016) 1238–1241. Available from: https://doi.org/10.1109/led.2016.2591181.
[28] M. Hu, et al., Memristor-based analog computation and neural network classification with a dot product engine, Adv. Mater. 30 (2018). Available from: https://doi.org/10.1002/adma.201705914.
[29] C. Du, et al., Reservoir computing using dynamic memristors for temporal information processing, Nat. Commun. 8 (2017) 2204. Available from: https://doi.org/10.1038/s41467-017-02337-y.

[30] C.D. Wright, P. Hosseini, J.A.V. Diosdado, Beyond von-Neumann computing with nanoscale phase-change memory devices, Adv. Funct. Mater. 23 (2013) 2248−2254. Available from: https://doi.org/10.1002/adfm.201202383.

[31] G. Tanaka, et al., Recent advances in physical reservoir computing: a review, Neural Netw. 115 (2019) 100−123. Available from: https://doi.org/10.1016/j.neunet.2019.03.005.

[32] J.J. Hopfield, D.W. Tank, "Neural" computation of decisions in optimization problems, Biol. Cybern. 52 (1985) 141−152. Available from: https://doi.org/10.1007/BF00339943.

[33] Hu, M. *et al*. *Design Automation Conference* 1−6 (IEEE, 2016).

[34] C. Li, et al., Analogue signal and image processing with large memristor crossbars, Nat. Electron. 1 (2018) 52. Available from: https://doi.org/10.1038/s41928-017-0002-z.

[35] M. Ercsey-Ravasz, Z. Toroczkai, Optimization hardness as transient chaos in an analog approach to constraint satisfaction, Nat. Phys. 7 (2011) 966−970. Available from: https://doi.org/10.1038/nphys2105.

[36] F.L. Traversa, C. Ramella, F. Bonani, M. Di Ventra, Memcomputing NP-complete problems in polynomial time using polynomial resources and collective states, Sci. Adv. 1 (2015) e1500031. Available from: https://doi.org/10.1126/sciadv.1500031.

Chapter 11

Exploiting the stochasticity of memristive devices for computing

Alice Mizrahi[1], Raphaël Laurent[2], Julie Grollier[1] and Damien Querlioz[3]

[1]*Unité Mixte de Physique, CNRS, Thales, Université Paris-Saclay, 91767 Palaiseau, France,* [2]*HawAI.tech S.A.S., Grenoble, France,* [3]*Centre for Nanoscience and Nanotechnology, Universite Paris-Saclay Palaiseau, France*

Memristive devices are usually noisy devices. If we program a device several times using exactly the same procedure, we typically get significantly different results. This stochasticity is very hard to fight at the material level: it is often due to the nature of resistive switching, which involves phenomena at the atomic level. The unpredictability of memristive devices is a concern for applications. To fight it system designers have two options: (1) they can program memristive devices using strong currents or voltages, where devices typically behave more reliability or (2) they can rely on system-level solutions, like error-correcting codes, which use supplementary devices to detect and correct errors occurring due to memristive unreliability. Both solutions are expensive in terms of energy consumption. However, there are ideas for computing approaches where elementary device unpredictability is not a problem and is even needed for the quality of the computation.

This chapter provides an introduction to such concepts, and discusses to what extent they can be applied to memristive-based computing. First we explain why harnessing randomness instead of fighting against it is an attractive strategy for low-power applications. We present various ways in which randomness can be used and how they are relevant to novel computing. Then we introduce different memristive devices in which randomness can be exploited. Finally we show some realizations of systems relying on the ideas previously introduced in the chapter. Additional description of recent works on the topic can be found in the review by Carboni and Ielmini [1].

Memristive Devices for Brain-Inspired Computing. DOI: https://doi.org/10.1016/B978-0-08-102782-0.00011-3

11.1 Harnessing randomness

The reliability of electronic devices has been an important issue since the early days of computing machines. In his 1956 lecture, von Neumann expressed concern about computation errors due to the intrinsic physics of components. As vacuum tubes were replaced by transistors, components became highly reliable. Transistors could be shrunk down and assembled in always larger circuits at a steady pace, following the prediction of Moore. But since the late 1990s, scientists have feared the end of Moore's law. Not only transistors are increasingly difficult and costly to manufacture as they reach nanometric dimensions but their reliability also decreases [2]: some circuits become significantly sensitive to noise and device variations and lose their perfectly deterministic behavior. Facing these issues, several approaches are possible. The most obvious approach is to fight unreliability. This means programming voltages cannot be scaled down anymore and error correction schemes need to be used sometimes [3,4]. However, these solutions are energetically costly. In consequence two more radical paths have been proposed: trading-off reliability for low-power consumption and embracing unreliability.

11.1.1 Trading-off reliability for low-power consumption

As early as 2001 the International Technology Roadmap for Semiconductors (ITRS) stated that "relaxing the requirement of 100% correctness for devices and interconnects" would allow drastic cost reductions [5]. A lot of effort has been put into designing computing architectures that produce reliable results with unreliable components [6]. For example the group of Krishna Palem at Rice University showed that a small loss in reliability can lead to important energy savings: allowing for a few percentage of error on an inverter circuit allows a three-fold energy reduction.

It should be noted that not all computing applications are suited for such "approximate computing." However, many important computing applications nowadays do not require maximal precision to be useful. Image, audio, and video processing can allow various levels of imprecision, depending on the expected quality. Tasks such as recognition of patterns and classification often only require the most important features of the input.

This approach is not limited to logic and naturally generalizes to memory, memristive, or not. It has been studied how approximate storage allows energy cuts as well. Sampson et al. showed that reducing the precision of less significant memory bits stored with phase-change and Flash technologies enables to save energy and increases the lifetime of the memory cells [7]. Locatelli et al. studied approximate storage in magnetic random access memories (MRAM) [8].

Fig. 11.1 represents the energy reduction as compared to the 10^{-10} error rate case versus the error rate, for the programming of a magnetic memory cell. A 10^{-3} error probability can still be tackled for some applications and cuts the energy consumption down by a factor two ("approximate memory" regime). However, to drastically reduce energy costs, one must reach high error rates. For instance reducing the energy by 90% (i.e., a 10-fold energy gain) would require an error rate close to 100%. In this regime (indicated as "stochastic device" in Fig. 11.1), the programming is fully random. This calls for nonconventional architectures and computing schemes where noise is exacerbated and entirely embraced.

11.1.2 Embracing unreliability by using noise

This path is an even further break from the traditional approach. It consists in fully embracing noise and stochasticity and using schemes where they can have a positive impact. That noise can be useful counter-intuitive, but has been observed in different fields of science. The benefits of noise in physical systems were first made famous in the early 1980s with the development of a theory showing that an optimal amount of noise can amplify the response of a bistable system to a weak signal [9]. This phenomenon was called *stochastic resonance*. One difficulty in approaching the abundant literature concerning stochastic resonance is that a wide range of phenomena have been labeled this way.

11.1.2.1 Canonical model of stochastic resonance

The first and most popular model of stochastic resonance corresponds to a bistable system submitted to a periodic drive. A comprehensive and detailed mathematical derivation of the phenomenon can be found in Ref. [10]. Here we aim at presenting the phenomenon and providing to the reader an understanding of why it has generated so much interest.

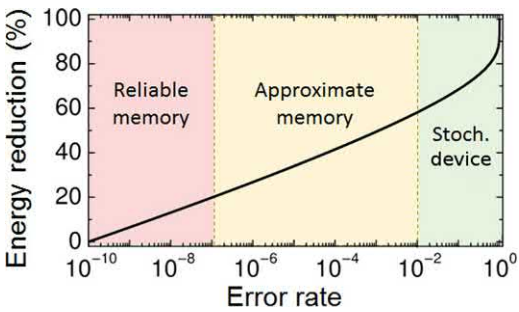


FIGURE 11.1 Numerical simulations of the energy reduction as compared to the 10^{-10} error rate case for the programming of a magnetic memory cell versus the error rate. Three regimes can be distinguished: reliable memory (red), approximate memory (yellow), and stochastic device (green).

We consider a double-well potential landscape as depicted in Fig. 11.2A:

$$V(x) = \frac{1}{4}x^4 - \frac{1}{2}x^2. \tag{11.1}$$

The two stable states of the system correspond to the local minima, which are located at ± 1 and are separated by a potential barrier $\Delta V = \frac{1}{4}$. A particle in this potential is submitted to noise so that its equation of motion is:

$$\frac{dx}{dt} = -\frac{dV}{dx} + \chi(t) \tag{11.2}$$

where $\chi(t)$ is a white Gaussian random function with mean zero, autocorrelation function $\langle \chi(t)\chi(0)\rangle = 2D\delta(t)$ and intensity D. In case of thermal noise $D = k_B T$ with k_B the Boltzmann constant and T the temperature. The particle motion is driven by noise and it can be shown that it randomly

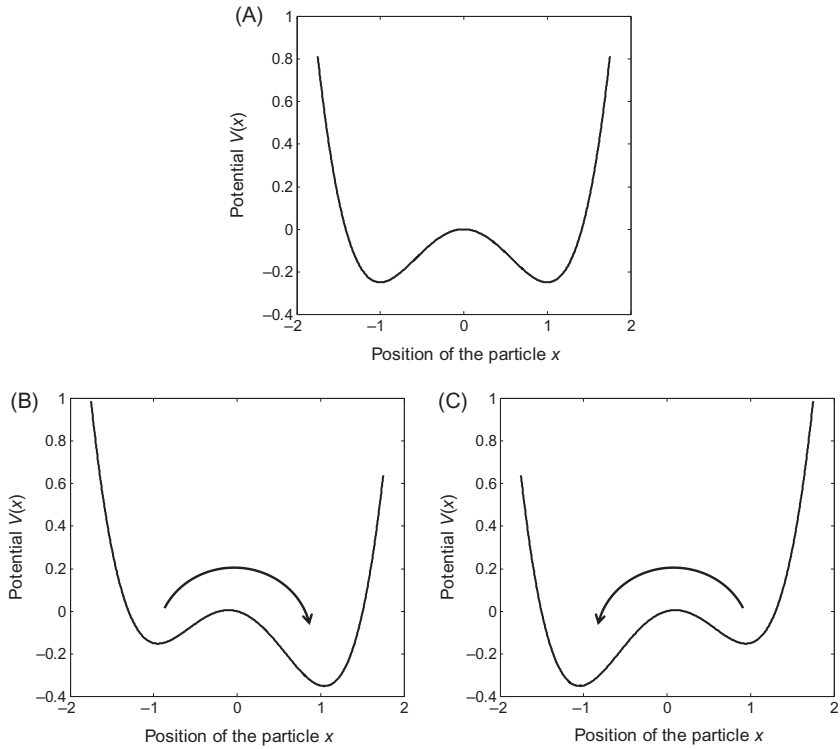


FIGURE 11.2 Energy potential $V(x)$ in function of the position of the particle x. (A) No drive is applied. (B) The value of the drive is $-A_0 = -0.1$. (C) The value of the drive is $+A_0 = +0.1$. In panels (B) and (C), the arrows represent the fact that the particle has a strong probability to jump from the high-potential well to the low-potential well.

switches symmetrically from well to well (i.e., between positions around 1 and -1) with the escape rate [11]:

$$r_K = \frac{1}{\sqrt{2\pi}} exp\left(-\frac{\Delta V}{D}\right). \tag{11.3}$$

The particle is now submitted to a periodic drive so that its equation becomes:

$$\frac{dx}{dt} = -\frac{dV}{dx} + A_0 cos(2\pi Ft) + \chi(t) \tag{11.4}$$

where A_0 is the amplitude of the drive and F is its frequency. The effect of the drive on the potential is depicted in Fig. 11.2B and C. As a consequence, the drive influences the motion of the particle. When it raises the potential of one well, it increases the probability for the particle to leave this well and jump to the lower potential well (as depicted by the arrows in Fig. 11.2B and C). Here we suppose that the amplitude of the drive is too weak to trigger forced oscillations of the particle between the wells. The intuitive reasoning would be to expect that the noise is detrimental to the motion of the particle following the periodic drive. However, it is observed that there is an optimal value of noise for which the response of the particle to the drive is maximal! For small drive amplitudes, the average position of the particle over many trials is indeed [10]:

$$\langle x(t) \rangle = X(D) cos(2\pi Ft + \Phi(D)), \tag{11.5}$$

where X is the amplitude of the periodic response of the system and is plotted in function of noise in Fig. 11.3:

$$X(D) = \frac{A_0}{D} \frac{2r_K(D)}{\sqrt{4r_K(D)^2 + 4\pi^2 F^2}}, \tag{11.6}$$

and Φ is a phase-lag equal to arc tan $(2\pi F/2r_K)$.

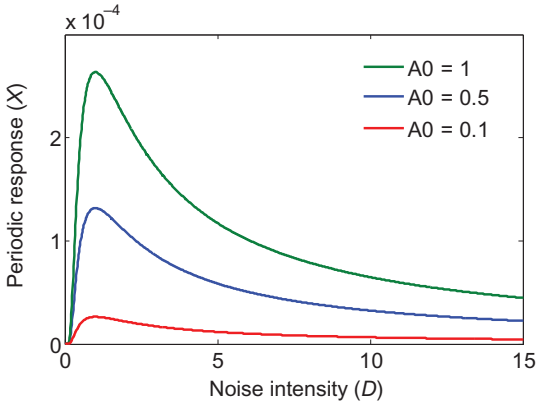


FIGURE 11.3 Periodic response X versus noise intensity D. Eq. 11.6 is plotted for a fixed drive frequency $F = 100$ and varying drive amplitude $A_0 = 1$ (green), $A_0 = 0.5$ (blue) and $A_0 = 0.1$ (red).

The periodic response X depends on the noise intensity D, and surprisingly, its dependency is nonmonotonous and exhibits a maximum for a positive value, as shown in Fig. 11.3. The higher the drive amplitude, the stronger the response of the system. This phenomenon has been called *stochastic resonance* [10]. Indeed the bell shape of the X versus D curve reminds of the amplitude versus frequency curve of traditional resonance.

A useful metric to quantify the response of the system to the periodic drive is the spectral density of the motion of the particle. It exhibits several peaks at odd-number multiples of the drive frequency F. Jung and Hanggi showed that the integrated power associated with the peak at F is $p = \pi X(D)^2$ [12]. In consequence the spectral power of the peak at the drive frequency takes a maximal value for an optimal level of noise as well.

The qualitative interpretation of the phenomenon is the following. For low noise intensities, the switches from well to well are rare (most of the motion is intra-well). There is little or no response to the periodic drive. As the noise intensity is increased, the switches are more frequent. They are facilitated by the periodic drive, which sees its action effectively amplified. When the noise becomes too intense, random switches dominate the motion of the particle and the influence of the drive loses its relevance. There is thus an optimal level of noise for which the response of the system to the drive is maximal. This optimal noise level depends on the amplitude of the drive A_0 as well as its frequency F. The response is maximal when the escape rate induced by the noise are close to twice the frequency of the drive: the time scale of the system (its "natural frequency") and the time scale of the drive match (as two switches are needed to complete one full oscillation).

These are two important features of stochastic resonance. First stochastic resonance is a resonance in noise intensity and not in frequency. The periodic response decreases monotonously as the frequency of the drive increases. Second stochastic resonance amplifies the response of the bistable system to a periodic drive but it does not correspond to synchronization. Noise can induce synchronization in bistable systems but under stricter conditions [13].

On top of its nonintuitive character that makes it scientifically fascinating, stochastic resonance is promising for applications. Usually noise is detrimental to the detection of weak signals and forces the observer to use stronger signals. But here noise which is often free (whether it is thermal noise from the room temperature or various fluctuations due to a real world environment) enables the use of weaker signals and thus allows a lower energy consumption.

11.1.2.2 Various types of stochastic resonance

As noise is ubiquitous in physical systems, the idea that it can be useful has sparked a lot of interest. Stochastic resonance has indeed been quickly extended from the canonical model to a more general framework [14].

Aperiodic stochastic resonance and nonlinear systems

Collins et al. showed that stochastic resonance can occur not only for a periodic drive but for any signal. They labeled this as *aperiodic stochastic resonance* [15]. Furthermore they showed that stochastic resonance (periodic or aperiodic) can occur not only in bistable systems but also in excitable systems (such as neurons that fire when their input crosses a threshold for instance) [15]. They extended their study to show that stochastic resonance only needs a nonlinear system and a weak input signal to occur [15].

Fig. 11.4 illustrates stochastic resonance in the case of a single threshold system. Here the system detects any signal whose amplitude is above a given threshold. (A) A sinusoidal signal (solid line) has to be detected but its amplitude is below the detection threshold (dashed line). (B) Noise is added to the signal and enables it to pass the threshold, triggering detection. Here detection is only possible because of the presence of noise. Having too much noise into the system would lead to false positive detection events. The response of the system to the signal (e.g., its detection) is therefore maximal for an optimal amount of noise. The shape of the signal (whether it is periodic or not, etc.) is not determinant for the occurrence of the phenomenon.

Suprathreshold stochastic resonance

From what has been described previously it seems that stochastic resonance can only occur for subthreshold inputs. And this is indeed the case for single element systems. Let us look again at the example depicted in Fig. 11.4: if the signal amplitude is high enough to cross the threshold and trigger detection, adding any level of noise will only produce false positive events and decrease the performance. However, stocks showed that this is not true in systems composed of many elements [16]. All elements have the same threshold, and receive individual inputs and independent noise. The output of the system is the sum of the individual outputs. In this case noise can improve the final output even for suprathreshold inputs [14].

Fig. 11.5, inspired by the results in Ref. [17], provides an illustration of suprathreshold stochastic resonance. A gray-scale picture is submitted to a threshold

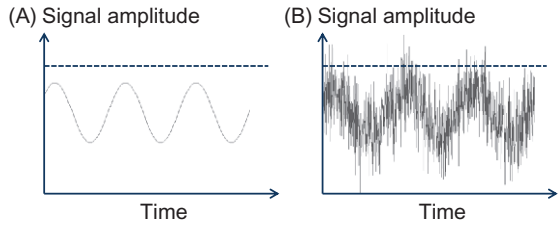


FIGURE 11.4 (A) Example of signal detection through stochastic resonance. A signal is plotted versus time. The dashed line is the detection threshold. (B) The same signal with additional white Gaussian noise is plotted versus time.

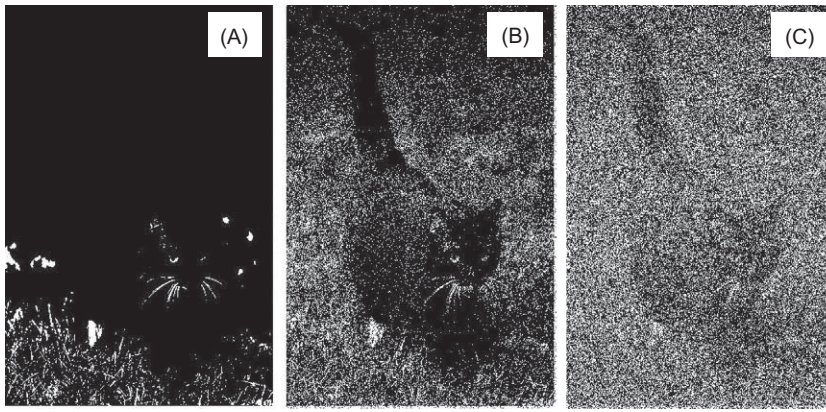


FIGURE 11.5 (A) Binarized image of a cat noise. (B) Picture with an optimal level of noise. (C) Picture with high noise.

operation so that each pixel becomes black (value below the threshold) or white (value above the threshold), giving the black and white picture of Fig. 11.5A. White Gaussian noise is then added to the input: a positive or negative random number is added to the value of each pixel. Some black pixels become white and some white pixels become black. Our eye performs a local average operation of neighboring pixels, thus leading to an impression of gray scale (even though the picture is only composed of black and white pixels). For an optimal level of noise (Fig. 11.5B), this improves our visual perception of the picture. Too much noise (Fig. 11.5C) makes the picture blurry and damages perception. The response of the system to the signal (perception of the picture—and recognition of the person) is maximal for a nonzero level of noise, which is exactly stochastic resonance.

In consequence stochastic resonance can occur in nonlinear systems submitted to a signal, provided that the global response of the system to the signal is suboptimal in the absence of noise. This condition is extremely broad and as a consequence stochastic resonance has been studied both theoretically and experimentally in different systems and fields. In climatology the canonical model of stochastic resonance explains how small variations of the eccentricity of the earth's orbit around the sun induced glacial ages every 10^5 years [9]. This specific case launched the field of stochastic resonance. In neuroscience a complete review has been written by Moss et al. [18]. Sensory receptors often exhibit threshold effects, which allow observing periodic and aperiodic resonance. Systems involving ensemble of receptors undergo suprathreshold resonance. Stochastic resonance has also been studied extensively in artificial systems. Examples of stochastic resonance in electrical circuits as well as corresponding mathematical models can be found in the review by Harmer et al. [19]. Badzey and Mohanty provided the

first experimental demonstration of stochastic resonance in a nanoscale system [20]. They reported the noise-induced amplification of the response of a nanomechanical silicon bistable oscillator to a periodic signal. Venstra et al. showed how stochastic resonance could allow the detection of a weak signal by a bistable cantilever [21]. In magnetic systems, with the emergence of spintronics, stochastic resonance was studied in several bistable nanomagnetic systems: domain wall motion [22], spin valves [23], and recently superparamagnetic tunnel junctions [24,25].

11.1.2.3 Relevance of stochastic resonance for computing

Among the systems in which stochastic resonance has been observed, neurons have received a specific focus. Models of various types of neural systems have been studied, including spiking neurons [26,27]. Stochastic resonance has been observed in in-vitro experiments on rats neurons, suggesting the positive role of noise in sensory tasks [28]. Hidaka et al. showed in-vivo that submitting a venous blood pressure receptor to a weak periodic drive and an arterial blood pressure receptor to noise can improve the human blood pressure regulatory system [29]. This suggests that stochastic resonance occurs as the signals from both receptors are combined in the brain. Nevertheless, it has not been demonstrated that the nervous system actually uses stochastic resonance. Whether our brain uses noise to perform computations is a fascinating and still open question. However, we know that both neural models and real neurons are able to exhibit stochastic resonance, and that the brain operates in an environment where noise can be found at high levels and under various forms [30]. And we know that the brain is able to perform complex computations, while only consuming 20 W.

Regardless of whether noise is really used by the brain, these facts suggest that stochastic resonance is an interesting path for the design of computing systems that are inspired from biology. A straightforward and attractive idea is to use stochastic resonance for weak signal detection by artificial neurons (in smart sensors for instance). However, using stochastic resonance for detection has not been translated into real applications, despite many attempts. Some possible reasons for this fact are that stochastic resonance is limited to weak signal detection, which is not the most useful effect for computing. Additionally an appropriate system for noise to be used has not been proposed yet. As a consequence in Section 11.1.2.4, we review other noise-induced phenomena and in Section 11.1.2.5, we explain why noise-induced synchronization is promising.

11.1.2.4 Broader paradigm of stochastic facilitation

Interesting noise-induced phenomena are numerous and various. They include stochastic resonance, noise-induced synchronization [31,32], coherence resonance [33], noise-induced phase transitions [34], noise-induced chaos [35], noise-induced pattern formation [36], diversity resonance [37], and gain enhancement [38].

It should be noted that the phrase *stochastic resonance* is often used in the literature to refer to any phenomenon where noise has a positive impact, which leads to confusion. This is why it has been proposed to designate these phenomena by the expression *stochastic facilitation* [39]. Stochastic facilitation is thus a generalization of stochastic resonance. It describes the fact that the performance of a system is maximal for an optimal level of noise which is nonzero, as depicted in Fig. 11.6A. Two important conditions should be reunited for stochastic facilitation to occur: the system should be nonlinear, and the performance in the absence of noise should be suboptimal.

11.1.2.5 Noise-induced synchronization for low-power computing?

Noise can induce the synchronization of a system with an external signal drive. As for noise-induced detection in the case of stochastic resonance, this phenomenon occurs for subthreshold signals whose amplitudes are too low to trigger synchronization in the absence of noise. Noise-induced synchronization has been studied extensively theoretically [32]. The most striking signature of noise-induced synchronization is the evolution of the frequency of the system versus the level of noise. As depicted in Fig. 11.6B, the frequency of the system plateaus at the frequency of the drive within an optimal range of noise for which synchronization and phase-locking are achieved. This plateau has been experimentally observed in various systems ranging from Schmidt triggers [40] to laser rings [41] and biological sensors in fish [42], as well as stochastic memristive devices, as shown later [24,25].

There has also been a specific interest for noise-induced synchronization in neural networks. It has been demonstrated theoretically [43] and observed

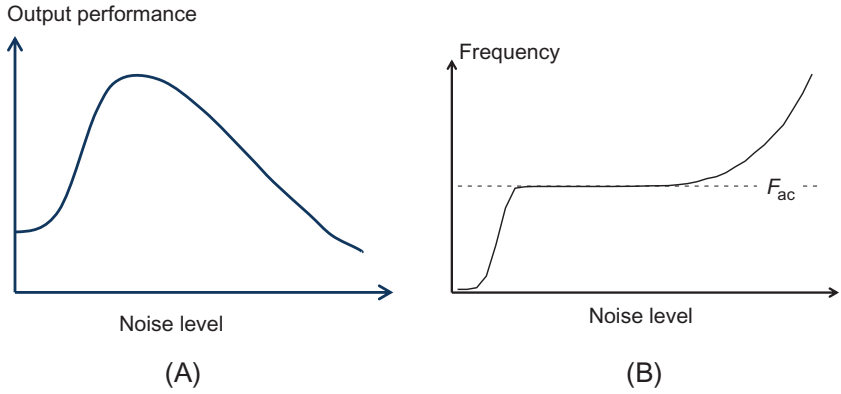


FIGURE 11.6 Noise-induced synchronization: (A) Sketch of the typical output performance versus noise level curve in a system which exhibits stochastic facilitation. (B) Sketch of the typical evolution of the frequency of a system submitted to a subthreshold signal versus the level of noise. The horizontal dashed line represents the frequency of the drive F_{ac}.

experimentally in rats [44] and paddle-fish [13]. Synchronization plays an important role in the brain: phase-locking of neuronal oscillations might be a key element in the processes of learning and memorization. Inspired by these observations many theoretical schemes using synchronization of oscillators to perform tasks such as pattern recognition and classification have been proposed [45,46]. Implementing these schemes in hardware holds the promise of fast and low-power cognitive computing. Specifically several groups have worked on how to use networks of coupled spin torque nano-oscillators [47,48]. In nano-scale devices, the fluctuations of thermal noise have an important impact. In particular noise is detrimental to traditional synchronization. Being able to use noise-induced synchronization would not only settle the issue of noise but also allow further power consumption gains (as lower drive amplitudes would be required).

After describing how noise in physical systems can have a positive impact, we now turn to the direct benefits of randomness for computing. The following section explains how random numbers encoding probabilities can allow performance of computations at low energy cost.

11.1.3 Computing with probabilities: stochastic computing

Since the beginnings of computing, it has been proposed to encode information by probabilities and not numbers [49–51]. The series of 0 and 1 bits no longer use the traditional binary representation but form a random bitstream whose average value (i.e., the probability to have a 1 bit) is the encoded number. For example the number 1010010111 should be read as the probability $6/10 = 0.6$ (instead of in binary representation). Larger numbers of bits allow encoding numbers with a higher precision. A small bitstream will give a rough approximation of the result, thus allowing a progressive precision gain. Using random bitstreams to encode numbers and the ensemble of schemes to perform operations on them is called *stochastic computing*. The main asset of this probabilistic representation is its tolerance to errors. With traditional binary representation, errors can be fatal if they concern the most significant bits. On the contrary, the bits composing random bitstreams do not have a hierarchy of significance: each bit contributes equally to the average. For example, coding 0010010111 instead of 1010010111 (i.e., an error on one bit) leads to representing 0.5 instead of 0.6. A much smaller error than representing 151 instead of 663!

Many schemes allowing to perform computations on random bitstreams have been developed [52]. An important asset of stochastic computing has emerged: many operations that are complex and thus costly with the traditional binary representation are simple when dealing with probabilities. For example, implementing the multiplication of two binary numbers requires counters and full adders. On the other hand, a single AND gate is sufficient to multiply two probabilities, as shown in Fig. 11.7. The random bitstream

FIGURE 11.7 Implementation of a multiplication with a single AND gate. The probabilities P_1 and P_2 associated with the inputs as well as the probability P associated with the output are indicated.

resulting from the AND operation of two random bitstreams has an average value equal to the product of the average values of the two inputs. Many studies have shown that using stochastic computing enables to considerably reduce the chip area used and the power cost [52]. In the early days of stochastic computing, studies have focused on implementing the building blocks of general purpose computing (matrix operations [53], division and square rooting [54], etc.). Neural networks have been explored [55], as well as hybrid analog-digital computing [56].

The strengths of stochastic computing are the small area used and the potential for low energy cost, the high tolerance to errors, and its character of progressive precision. However, these strengths have not been sufficient to alter the domination of conventional deterministic computing. Complementary metal–oxide–semiconductor (CMOS) transistors have quickly become highly reliable and the advances of Moore's law have allowed for lower and lower energy costs, making stochastic computing promises unnecessary. Moreover stochastic computing has critical issues. The generation of random bitstreams with CMOS technology is costly in terms of energy, and the same is true about analog-random bitstream conversions. Furthermore the random bitstreams in a stochastic logic unit (e.g., an AND gate for multiplication) should be uncorrelated for the operation to be accurate. This makes cascading logic gates uneasy and requires to regenerate random bitstreams frequently, which consumes even more energy. The energy gain due to the logic simplification of stochastic computing is annihilated by the cost of generating the stochasticity. For these reasons stochastic computing has been unable to compete with conventional computing yet.

As Moore's law is coming to an end and as energy consumption is becoming a burning issue, stochastic computing appears increasingly appealing. Research in this field has focused on more appropriate applications: image processing [57], pattern recognition [58], error-correcting codes [59], signal processing [60], computation in harsh radiation environments [61], synaptic sampling machines [62], Bayesian inference [63,64], Markov-chain Monte Carlo sampling [65] and more modern neural networks [66]. These applications are costly to implement by conventional computing. By allowing easy implementation of complex mathematical functions and parallel computing, stochastic computing brings drastic energy gains. In the light of the decreasing reliability of electronic components when the nanoscale is reached, a broader paradigm—"stochastic electronics"—has been called for

[67]. It proposes to put randomness at the core of the hardware and to use noise through stochastic facilitation (see Section 11.1.2). Although promising this new view of stochastic computing does not address the issue of random bitstream generation. There is the need for a device which is able to generate random numbers at low energy cost. Several different media have been proposed and will be detailed in the following section of this chapter.

Overall several ways to harness randomness for computing exist and have been studied thoroughly. However, they have not yet led to actual computing applications. Indeed it is difficult to identify a perfect stochastic building block. This device should feature an intrinsic stochastic behavior which is well understood and controlled. Ideally it should have a nonlinear dynamics in order to allow stochastic facilitation phenomena. Of course its state should be easily readable, and the device needs to be compatible with CMOS technology.

In the following section, we review various proposed candidates. We investigate how memristive and spintronic technology can answer this need.

11.2 Proposals of stochastic building blocks

Several systems, which differ fundamentally, have been proposed to implement novel forms of computing that harness randomness. However, computing with these stochastic devices is still at a very early stage and not much has been demonstrated yet. We present the most significant of these devices and explain why they can be promising.

Sections 11.2.1 and 11.2.2 present two historical ideas of computing systems subject to randomness: quantum dots and molecular approaches. Sections 11.2.3 and 11.2.4 describe approaches using memristive nanotechnology. Memristive devices are particularly interesting for stochastic computing applications because they exhibit complex dynamics and their states can be read and controlled more easily than the ones of molecular devices. Furthermore it has been shown that nanodevices can emulate some functionalities of the components of the brain: synapses and neurons. This makes them promising candidates for the hardware implementation of bioinspired computing schemes.

11.2.1 Quantum dots cellular automata

A cellular automaton is a grid of identical unit cells. Each cell has a finite number of states, and the change of state of each cell depends on a set of rules conditioned by the state of its neighbor cells. It has been demonstrated that both deterministic [68] and probabilistic [69] cellular automata can perform computing tasks. While most cellular automata are implemented in software, there have been several hardware realizations, the most studied being quantum dot cellular automata and magnetic cellular automata. In the quantum dot cellular automata, demonstrated experimentally as early as 1997 [70], if two cells are placed next to each other, they will take the same polarization because of Coulomb repulsion.

This allows creating wires of quantum dots cellular automatathatpropagate binary states through space. Because of their quantum nature, these devices exhibit intrinsically random behaviors. As a consequence a recent study by Purkayastha et al. proposed to use quantum dot cellular automata as true random number generator [71]. A true random number generator uses an intrinsically stochastic phenomenon to generate random numbers, on the contrary of a pseudo random number generator which uses an algorithm to generate, from a seed value, a deterministic sequence of numbers close to random statistics. Purkayastha et al. proposed an architecture and an algorithm to generate random ciphers. These results could be useful for cryptography applications. However, despite the facts that this stochasticity has been widely studied and that computing schemes with probabilistic cellular automata have been proposed, there has been no use of quantum dots for stochastic computing. The works on computing with quantum dots rather focus on how to fight their randomness, mostly with redundancy [72].

11.2.2 Molecular approaches

11.2.2.1 Biomolecular automata

Biomolecular automata are molecular systems that go through a predetermined sequence of state-to-state transitions. These transitions can be deterministic or stochastic. Biomolecular automata can perform computing [73]. The input, output, software and hardware of such a *biomolecular computer* are biomolecules. The input molecule undergoes a sequence of operations, which are determined by another molecule—the software molecule. These operations change the input molecule into the output molecule which is the result of the computation. In this biomolecular automata the input is a DNA molecule. The transitions that the input will undergo are encoded by another set of DNA molecules (software). The hardware is composed of DNA-manipulating enzymes. The digestion of the input DNA by the enzymes depends on the software DNA. At the end of the reaction the input is transformed into the output DNA molecules, which encodes the result of the computation.

Biomolecular computers are particularly relevant for the analysis of biomedical information, as they can provide a direct analysis of biological signals without having to convert information into electronic form. For instance a biomolecular computer could perform direct recognition of molecular disease indicators and automatically release a biologically active molecule as a drug. Because biological processes are highly stochastic, it is a good strategy to use a computing method which takes advantage of this randomness. Therefore, stochastic computing is appropriate for biomolecular computing, and processing biological information in general. Adar et al. experimentally demonstrated the implementation of such a system [74]. The probabilities of the transitions outcomes are controlled by choosing the concentrations of the various software molecules. This idea is exciting because molecules can be synthesized in huge numbers with atomic precision.

11.2.2.2 Resonant energy transfer between chromophores

The *chromophore* is the part of a fluorescent molecule which is responsible for its color. A chromophore has two molecular orbitals whose energy difference correspond to visible light. As a consequence it can absorb a photon of a given wavelength, exciting an electron from its ground state to a higher energy orbital. The excited chromophore can then re-emit a photon of the same wavelength. When two chromophores are close to each other, *resonant energy transfer* can occur: the donor absorbs a photon, then transfers energy to the acceptor which in turn can emit a photon. As only a fraction of the excitation energy is transferred, the light emitted by the acceptor has a larger wavelength than the light absorbed by the donor [75]. The time between the absorption by the donor and the emission by the acceptor is random and has an inverse exponential probability distribution. It constitutes a continuous time random Markov chain. Wang et al. showed that observing the transitions means sampling from a probability distribution [76]. It allows sampling both continuous and discrete variables (by discretizing the time in time steps). By modifying the types of chromophores and the topology of the network, one can implement many Markov chains and thus many probability distributions to sample from. This is particularly useful to implement computing schemes that require many random numbers (discrete or continuous) from specific probability distributions. Wang et al. specifically proposed to use resonant energy transfer between chromophores for Bayesian computing tasks. Building a chromophores network matching the considered Bayesian network enables to directly sample from the wanted probability distribution. This allows to perform Bayesian tasks without computing complex probability distributions and artificially generated random numbers, which is costlier in terms of time and energy.

Several approaches to use molecules as building blocks for stochastic computing have therefore been proposed. These approaches are exciting, nevertheless, molecular computers still have a long way to go before becoming a mature technology. In particular, it is difficult to read and control the state of molecules.

11.2.3 Charge-based memristive devices

Several groups have proposed to embrace the stochasticity of memristors and use it for computing applications. Here we present the most significant contributions.

11.2.3.1 Memristors as random bitstream generators

Gaba et al. have investigated the use of memristors to generate random bitstreams for stochastic computing [77]. The considered device is composed of amorphous silicon sandwiched between a silver and a poly-silicon electrode, and switches by the formation and dissolution of silver filaments through the amorphous silicon.

In this device the process of filament formation is intrinsically stochastic, as it involves the motion of atoms. When a large enough voltage is applied,

the waiting time before the formation of the filament is random and follows a Poisson process [77], as depicted in Fig. 11.8A. The probability for the filament to form in a Δt interval within a time t is:

$$P(t) = \frac{\Delta t}{\tau} exp\left(-\frac{t}{\tau}\right), \tag{11.7}$$

where τ is the characteristic time, which decreases when the applied voltage increases. Integrating Eq. (11.7) leads to the cumulative switching probability:

$$C(t) = 1 - exp\left(-\frac{t}{\tau}\right). \tag{11.8}$$

If a voltage pulse of given amplitude and width is applied to the device, the probability to switch the resistance is given by Eq. 11.8. Applying a succession of identical pulses and observing the state of the memristor generates a random bitstream, as depicted in Fig. 11.8B. Each attempt to induce switching represents a bit: a successful switch corresponds to 1 while no switch corresponds to 0. The probability encoded by the bitstream can be raised (reduced) by increasing (decreasing) the amplitude and the width of the voltage pulses. This memristor therefore implements a true random bitstream generator, and the same principle has been explored by other groups with other memristive technologies [78]. It could be used for stochastic computing applications. A critical issue of such devices is their relatively high power consumption. For instance Fig. 11.8 presents results obtained for an applied voltage of 2.5 V, which is higher than the supply voltage of traditional computing circuits.

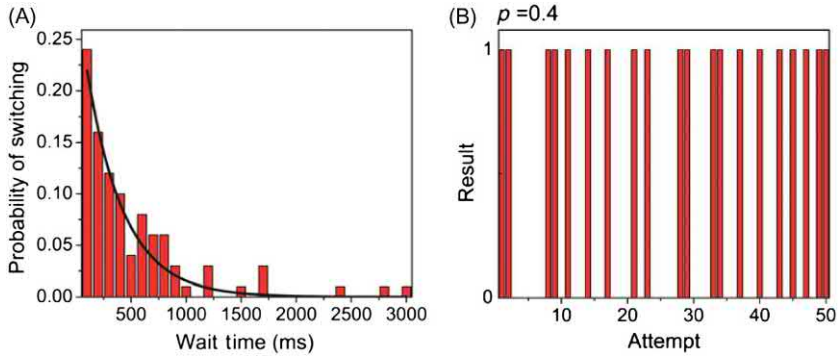


FIGURE 11.8 (A) Random wait time distribution for an applied voltage of 2.5 V. Red bars: experimental data. Solid black line: fitting of Eq. 11.7 using parameter $\tau = 340$ ms. (B) Result of the switching attempt ("0" means failure while "1" means success) versus the number of the attempt. Each attempt corresponds to a voltage pulse of amplitude 2.5 V and width 300 ms. This bitstream has a probability $p = 0.4$. *Reproduced from S. Gaba, P. Sheridan, J. Zhou, S. Choi, W. Lu, Stochastic memristive devices for computing and neuromorphic applications, Nanoscale 5 (2013) 5872.*

11.2.3.2 Memristors as stochastic integrate and fire neurons

We now present an implementation of stochastic integrate and fire neurons with phase-change memristors, proposed by Tuma et al. [79]. Various types of neurons have been observed and many theoretical models have been proposed. One of the most popular models is the *integrate and fire neuron*. This type of neuron receives voltage inputs and integrates them. When the cumulative value of the received inputs exceeds a threshold, the neuron fires a voltage pulse (also called *spike*). The neuron is reset and integration can start from zero. In biological neurons, inputs are stored by the potential of the membrane. Integrating inputs and firing outputs are partly stochastic processes. This is due to the noise of ionic conductance, the effect of thermal noise on the motion of charge carriers, interneuron morphological variability and background noise. Stochasticity is an important component of neural behavior. Some studies have even suggested that it is used for information encoding [80]. Specifically, many schemes using assemblies—called populations—of stochastic neurons have been proposed. More details are given at the end of this chapter.

The memristor of Tuma et al. is composed of a chalcogenide-based phase-changing material, with two possible states: low-resistivity crystalline or high-sensitivity amorphous. This device is called mushroom-type because only a dome-shaped region of the material actually undergoes phase transitions while the rest remains crystalline, and the conductance of the device is directly linked to the thickness of the amorphous region. Applying voltage pulses lets current flow through the device, which generates a Joule heating. For the right pulses amplitude, the temperature is above the glass transition but below the melting point of the material. The heating therefore allows crystal growth and the thickness of the amorphous region decreases. The conductance of the device thus increases. When it reaches a threshold, the current flowing through the device is large enough for the temperature to reach the melting point. As the pulse is cut-off abruptly, the material within the region quenches into the amorphous phase. This corresponds to the firing of the neuron and resets the device to a low conductance.

The thickness of the amorphous region and its evolution are ruled by processes that are intrinsically stochastic, for several reasons. First the thickness of the amorphous region and its internal atomic configuration created by melting and quenching when the neuron fires are never the same after reset. Second during melting, the atomic mobility is high. Small variations in the initial conditions or pulse characteristics lead to different states and thus different growth velocity. Finally at each reset the amorphous region has different crystalline nuclei. This leads to different thickness evolution when pulses are applied. Additional crystalline nuclei can appear and alter the thickness evolution. This stochasticity is translated into the distribution of the interspike intervals. The mean interval decreases exponentially when the pulse width increases, and the distributions of interspike intervals are Gaussian. The larger the pulse width, the lower the mean interval and the more narrow the distribution.

A population of such stochastic neurons, composed of an assembly of memristors, can encode information in a way that is not possible with a single neuron or a population of deterministic neurons. Because of device-to-device variability, the frequencies of the artificial neurons follow a roughly Gaussian distribution. All individual frequencies are below 20 kHz. Each memristor of the population is submitted to a triangular input signal of frequency $F = 10$ kHz. This means that the signal is broadband as all its components have a frequency above 10 kHz. It can be observed that the number of firing devices versus time follows the same shape as the input signal. The population is therefore able to encode a broadband signal even though the individual frequencies of the devices were all under $2F$. This would have not been possible with a single device, due to the Nyquist−Shannon theorem. With a population of deterministic spiking neurons, which fire at a periodic pace, this would have been challenging. Using stochastic rather than deterministic neurons allows in this specific case to encode a signal with a better time resolution. The representation error decreases with the number of devices, which outlines the strength of the population coding.

These results on true random bit generation and stochastic neurons are promising for the implementation of computing systems using randomness. As mentioned previously, memristors have many attractive qualities to be computational building blocks. However, charge-based memristors have some limitations in this context. The stochastic behavior of memristors come from physical phenomena that are multiple and not well controlled. The physics of filament formation and destruction in memristors are yet not fully understood. Although the resistance switching mechanism can be described as a Poisson process, its characteristic time and the way it varies from device to device are not well modeled. Similarly the randomness in the interspike intervals of phase-change memristors comes from many noncontrollable physical sources. Stochastic behaviors in memristors originate from defects and are hard to predict parameters. A challenge for the future will be to manufacture large numbers of memristors which random behaviors follow probability distributions with random parameters, although this would be a requirement to build computing systems. Finally the described memristors exhibit resistance changes due to structural variations or atomic motion. Performing these changes degrades the device. As a consequence these memristors have a limited endurance.

11.2.4 Spintronics

We now turn to the specific field of spintronics, which can provide spin-based memristive devices. Stochastic spintronic devices indeed feature high reliability and endurance, together with well understood and controlled stochasticity. Spintronics is a contraction of spin electronics. It concerns the study of the physics of the spin of the electrons and of devices aiming at coding and processing information not only with the charge of the electrons (which corresponds to conventional electronics) but also with their spin.

The flagship device of modern spintronics is the magnetic tunnel junction. Its primary use is to be a nonvolatile memory as the unit cell of MRAMs. A junction in the parallel state codes for 0, whereas a junction in the antiparallel state codes for 1. The read operation is done through a measurement of the electrical resistance. The most recent generations of MRAMs use the phenomenon of spin transfer torque to perform the write operation through current injection (these are called STT-MRAMs).

The most important feature of magnetic tunnel junctions is their tunnel magnetoresistance (TMR), which characterizes the difference between the electrical resistance of a magnetic tunnel junction in the parallel (R_P) and antiparallel (R_{AP}) states. It can be defined as:

$$TMR = \frac{R_{AP} - R_P}{R_P} = \frac{2P_1P_2}{1 - P_1P_2}, \tag{11.9}$$

where P_i is the spin polarization of the magnetic layer i. The value of the polarization depends on the materials of the magnetic layers as well as the tunnel barrier. The materials used for the magnetic layers of magnetic tunnel junctions are typically metallic alloys such as cobalt−iron−boron (CoFeB) or nickel−iron (NiFe). Currently the most common material for the tunnel barrier is manganese oxide (MgO). To reduce its stray field the reference layer is usually made not with a single magnetic layer but with a synthetic antiferromagnet, composed of two magnetic layers separated by a nonmagnetic one.

The other two important features of a magnetic tunnel junction for memory applications are its energy barrier (i.e., the height of the energy well) and its critical current. The energy barrier should be large to guarantee the stability of the stored state. The critical current should be small to limit the energetic cost of write operations. Lowering the critical current while maintaining the energy barrier is a crucial challenge of MRAM design and fabrication [81].

The notion of whether a magnetic tunnel junction is stable or unstable is entirely relative to the relevant time scale. When destined to memory applications, magnetic tunnel junctions are designed so that no junction on the memory chip will switch over for 10 years. This leads to energy barriers so that $\Delta E/k_BT > 60$ [82]. The energy barrier is proportional to the surface of the base of the junction [83]. Thus smaller devices are less stable.

Magnetic tunnel junctions come in two types: in plane and out of plane. In-plane junctions have magnetizations that are parallel to the plane of the thin films while out-of-plane junctions, also called perpendicular junctions, have magnetizations that are perpendicular to the plane of the thin films (Fig. 11.9). The advantage of perpendicular magnetic tunnel junction is that, at equal stability, they exhibit lower critical currents [82], and the latest MRAM generations are made with out-of-plane magnetic tunnel junctions.

Magnetic tunnel junctions are attractive for the following reasons. Their physics is well understood. They are reliable. The switching process does not

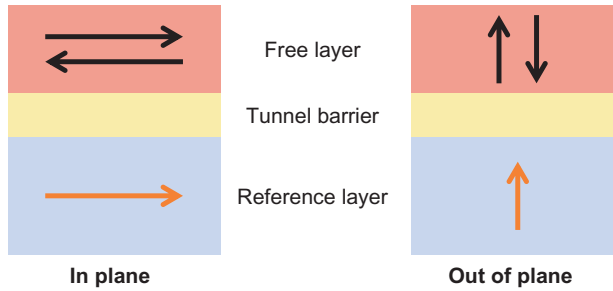


FIGURE 11.9 (Left) In-plane magnetic tunnel junction. (Right) Out-of-plane magnetic tunnel junction. Each stack is composed of a reference layer (blue), a tunnel barrier (yellow), and a free layer (red). The orange arrow is the reference magnetization and the black arrow is the free magnetization.

damage them (contrary to filamentary memristors for instance); therefore they exhibit outstanding endurance [81]. They are compatible with CMOS technology. Their read-write operations are fast (from 2 to 30 ns) [81]. Several companies including big players such as Samsung, Globalfoundries or TSMC, now manufacture MRAMs. Despite these significant advances, MRAMs face a critical challenge: magnetic tunnel junctions lose their stability when their size is decreased. While these unstable devices (superparamagnetic tunnel junctions) are useless for memory applications, they are promising for unconventional computing, as shown in Section 11.3.3.

11.2.4.1 Modifying the magnetic state—spin torques

Electrical currents can not only enable to read a magnetic state but also to modify it. The phenomena where a charge current influences the magnetization of a material are called spin torques, and currently, the most important spin torque is the spin transfer torque. Here, we describe how it affects the dynamics of the magnetization of a nanomagnet. More details are found in Ref. [84].

Dynamics of the magnetization of a nanomagnet

Let us consider a monodomain nanomagnet made of a thin ferromagnetic ellipse. Its magnetization $m^{\rightarrow}$ can be described by a single macro-spin rather than many individual electronic spins. At zero temperature, the dynamics of its magnetization follows the Landau−Lifshitz−Gilbert equation:

$$\frac{1}{\gamma}\frac{d\,m^{\rightarrow}}{dt} = \vec{m} \times \left(H_{eff}^{\rightarrow} - \frac{\alpha}{m}\vec{m} \times H_{eff}^{\rightarrow}\right), \tag{11.10}$$

where $\gamma = \frac{2\mu_B}{\hbar}$ is the gyro-magnetic ratio and α is the damping coefficient, which depends on material parameters. $H_{eff}^{\rightarrow}$ is the effective magnetic field, and includes the external magnetic field, as well as sources of magnetic

anisotropy that define the easy axis of the junction. In the case of an out-of-plane magnetization, $\vec{H_{eff}}$ is perpendicular to the plane of the magnet. In the case of an in-plane magnetization, $\vec{H_{eff}}$ keeps the magnetization in the plane. Following Eq. 11.10, the magnetization processes around the easy axis. In this process, the magnetization is subject to two torques. The field torque drives the precession while the damping brings the magnetization back to the easy axis. This corresponds to oscillations in one of the potential wells.

The disorder induced by thermal noise is small compared to the exchange energy, which tends to align the spins. In consequence thermal noise does not reduce the magnetization but rather kicks it around randomly. Finite temperature can thus be modeled by adding a Langevin random field $\vec{H_L}$ in Eq. 11.10 [85]:

$$\frac{1}{\gamma}\frac{d\vec{m}}{dt} = \vec{m} \times \left(\vec{H_{eff}} + \vec{H_L} - \frac{\alpha}{m}\vec{m} \times \vec{H_{eff}}\right). \tag{11.11}$$

$H_{L,i} = \sqrt{\frac{2\alpha k_B T}{\gamma m}} I_{rand,i}(t)$ for $i = x, y, z$. $I_{rand}(t)$ is a random number with Gaussian distribution of mean zero and standard deviation one. The three x, y, and z random components are uncorrelated. The amplitude of the field is deduced from the fluctuation−dissipation theorem. The dynamics of the magnetization is therefore thermally activated. The random fluctuations of thermal noise can cause the magnetization to leave its potential well, following a Poisson process. The average time spent by the magnetization in the well is the characteristic time of the Poisson law. This life time inside the well or dwell time is [85]:

$$\tau = \tau_0 exp\left(\frac{\Delta E}{k_B T}\right), \tag{11.12}$$

where k_B is the Boltzmann constant, T is the temperature, $\tau_0 \simeq \frac{1}{\gamma H_k}$ is the attempt time and ΔE is the potential barrier height of the well.

Spin transfer torque

A spin-polarized charge current is now also injected in the thin film, perpendicularly to its plane. The spin-polarization has a value:

$$\eta = \frac{I_\uparrow - I_\downarrow}{I_\uparrow + I_\downarrow} \tag{11.13}$$

where $I_\uparrow$ and $I_\downarrow$ are the currents in the two spin channels (parallel and anti-parallel to the spin-polarization). The unit vector of the polarization is $\vec{n_s}$.

The spin-polarization of the current is modified by the magnetization. Reciprocally, some spin angular momentum of the current is absorbed by the magnetization. As a consequence the magnetization precession is affected. The resulting torque has two components: the spin transfer torque which is colinear to the damping and the field-like torque which is colinear to the

field torque. The field-like torque is weaker than the spin transfer torque but is still significant in magnetic tunnel junctions, where it is typically around $30\% - 40\%$ of the spin transfer torque [86]. Here we focus on the dominant term, the spin transfer torque. It can be shown that in the macrospin case, the transverse component of the spin−torque is [87]:

$$\vec{\Gamma} = I\eta\frac{\hbar}{2e}\frac{1}{m^2}\left(\vec{n_s} \times \vec{m}\right) \times \vec{m}, \tag{11.14}$$

where e is the charge of the electron and $\hbar$ is the reduced Planck constant. This component is proportional to the number of electrons flowing through the layer and their spin-polarization. The transverse component of the spin−torque has to be added to the right member of Eq. 11.11, which then becomes:

$$\frac{1}{\gamma}\frac{d\vec{m}}{dt} = \vec{m} \times \left(\vec{H_{eff}} + \vec{H_L} - \frac{\alpha}{m}\vec{m} \times \left(\vec{H_{eff}} + \vec{H_s}\right)\right), \tag{11.15}$$

where

$$\vec{H_s} = I\eta\frac{\hbar}{2e}\frac{1}{m\alpha}\vec{n_s}. \tag{11.16}$$

The spin torque does not contain a fluctuating field because it is already a dissipating force itself. In the small cone limit of an out of plane magnetization (e.g., the magnetization does not go too far away from its easy axis), $\vec{H_{eff}}$ and $\vec{H_s}$ are approximately parallel. In consequence, Eq. 11.15 is equivalent to Eq. 11.11 where the damping α is replaced by $\tilde{\alpha}$:

$$\frac{1}{\gamma}\frac{d\vec{m}}{dt} = \vec{m} \times \left(\vec{H_{eff}} + \vec{H_L} - \frac{\tilde{\alpha}}{m}\vec{m} \times \vec{H_{eff}}\right). \tag{11.17}$$

The effective damping is expressed as follows:

$$\tilde{\alpha} = \alpha\left(1 + \frac{H_s}{H_{eff}}\right) = \alpha\left(1 + \frac{I}{I_c}\right). \tag{11.18}$$

The critical current I_c corresponds to the current required to switch the magnetization at zero temperature:

$$I_c = \frac{1}{\eta}\frac{2e}{\hbar}m\alpha H_{eff}. \tag{11.19}$$

When the effective damping becomes negative, the disturbances to the magnetization equilibrium are amplified rather than damped out. This instability leads to magnetization reversal. The spin transfer torque is often referred to as an *anti-damping*. The sign of the critical current depends on the sign of m and thus on the orientation of the magnetization. In consequence each polarity of

the current stabilizes one orientation of the magnetization and destabilizes the other. A negative current switches the magnetization of the free layer in the AP state while a positive current switches it in the P state. Because the critical currents for P and AP states are different, the curve exhibits a hysteresis loop. Critical current densities typically range from 10^6 to 10^7 A/cm^2.

The thermal term $H_L^{\rightarrow}$ is not modified by the current. In consequence it corresponds to an effective temperature $\tilde{T}$ such that $\tilde{\alpha}\tilde{T} = \alpha T$. Therefore, the dwell-time of the magnetization in the potential well is modified as follows by the spin transfer torque [88]:

$$\tau = \tau_0 exp\left(\frac{\Delta E}{k_B \tilde{T}}\right) = \tau_0 exp\left(\frac{\Delta E}{k_B T}\frac{\tilde{\alpha}}{\alpha}\right) \tag{11.20}$$

$$\tau = \tau_0 exp\left(\frac{\Delta E}{k_B T}\left(1 - \frac{I}{I_c}\right)\right). \tag{11.21}$$

This expression is valid for a current of amplitude inferior to I_c. The value and sign of I_c depend on the orientation of the magnetization. Thus a given current polarity stabilizes one orientation and destabilizes the other. A current with an opposite polarity will have the inverse effect.

Stochastic switching of magnetic tunnel junctions

The state of magnetic tunnel junctions can be read by an electrical resistance measurement. Similarly, the state of the free layer can be written by the injection of an electrical current through the device as shown in Fig. 11.10. The reference layer acts as a spin polarizer. The newly spin-polarized current can then modify the magnetization of the free layer through spin transfer torque.

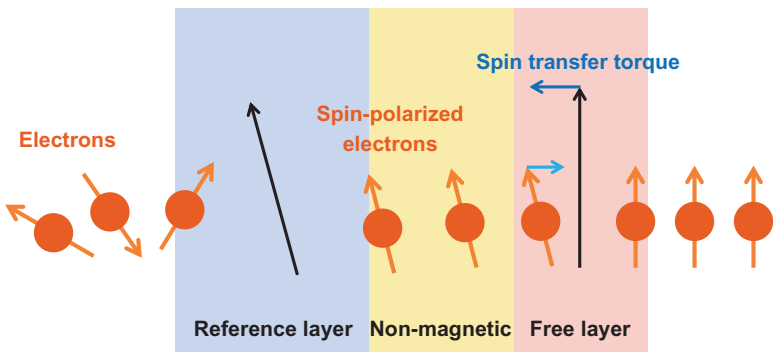


FIGURE 11.10 Schematic of the mechanism of spin transfer torque in a multilayer device. The electrons (orange) flow through the reference layer (blue) which spin-polarizes them. The electrons then flow through the free layer (red) which modifies their spin-polarization (light blue arrow). Reciprocally, the magnetization absorbs some spin-angular momentum (dark blue arrow).

It is important to remark that any nanomagnetic device has an intrinsically stochastic behavior. This randomness is directly due to the fluctuations of thermal noise and arises for any finite temperature. Contrary to the devices studied in Section 11.2.3, stochasticity does not come from defects. This allows us to understand better the stochastic behavior and, more importantly, to control it. Nanomagnets naturally amplify the thermal noise into full amplitude magnetization reversals. In the case of magnetoresistive devices, this translates into full amplitude oscillations of the resistance.

11.3 Test cases of stochastic computation: case of magnetic tunnel junction

To highlight the variety of computation schemes exploiting stochastic properties of memristive devices, we now focus on magnetic tunnel junctions, as we have seen that they feature highly controllable and well-understood stochasticity. We present several uses of such devices, going from a relatively conventional application of random bit generation, to more radical ideas.

11.3.1 Spin dice: a true random number generator

Fukushima et al. proposed to use magnetic tunnel junctions as true random number generators[1] [89]. The magnetic tunnel junction starts in the parallel state. A current pulse of amplitude I and width Δt is applied to the junction. The probability for the junction to switch to the antiparallel state is [90]:

$$P(I) = 1 - exp\left(-\frac{\Delta t}{\tau_0} exp\left(-\frac{\Delta E}{k_B T}\left(1 - \frac{I}{I_c}\right)\right)\right). \tag{11.22}$$

The current amplitude and width are chosen so that the probability is 0.5. If the junction switches, the value encoded is 1. If it does not switch, it is 0. The junction is then reset in the parallel state by a negative current pulse and can be excited again as shown in Fig. 11.11. The system proposed by Fukushima et al.—called Spin Dice—is composed of eight out-of-plane magnetic tunnel junctions. This allows to represent a 8-bit random number.

Spin Dice is a true random number generator, which can be useful for applications such as cryptography or stochastic computing. It has passed usual random numbers tests from the National Institute of Standard and Technology [89] and the Special Publication 800 Statistical Tests Suite[2]. It can be scaled to larger systems composed of many magnetic tunnel junctions.

1. A random number generator is the randomness which comes from a physical process and not a deterministic algorithm.
2. See http://csrc.nist.gov/ for more details.

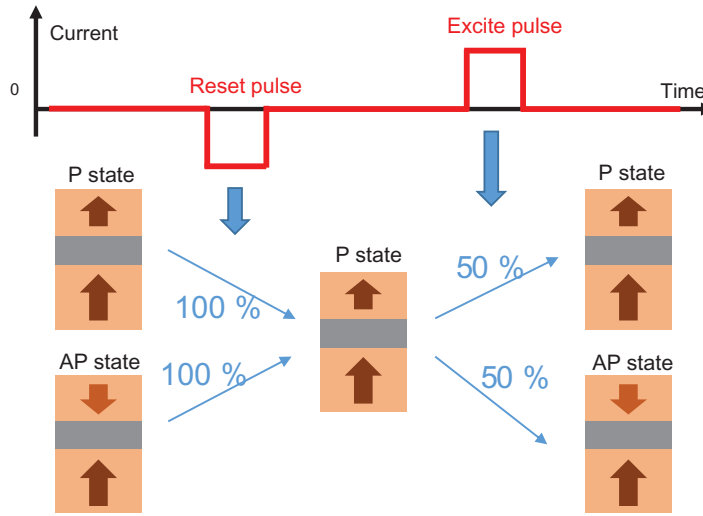


FIGURE 11.11 (Top) current injected through the magnetic tunnel junction versus time. (Bottom) corresponding evolution of the junction's state. The junction can be P or AP, a negative pulse resets it into the P state with 100% probability. Then a positive pulse excites it in the P or AP state with a 50% probability for each. *Adapted from A. Fukushima, T. Seki, K. Yakushiji, H. Kubota, H. Imamura, S. Yuasa, et al., Spin dice: A scalable truly random number generator based on spintronics, Appl. Phys. Express 7 (2014) 083001.*

11.3.2 Stochastic synapses

Vincent et al. proposed to leverage the stochastic programming of magnetic tunnel junctions for bioinspired computing in a more unconventional way [91]. The proposed system is a neural network. The integrate and fire spiking neurons are implemented in CMOS and the synapses are magnetic tunnel junctions. Each input neuron is connected to each output neuron through a crossbar of synapses—as depicted in Fig. 11.12A. Vincent et al. demonstrated that this system can perform a classification task. The input is data from a bioinspired artificial retina and is a video of a highway with several lanes, where cars go by (some data are shown in Fig. 11.12B). The task is to delimit the lanes and count the cars passing in each of them.

Each input neuron corresponds to a pixel of the input data. When there is a high value in the corresponding pixel, an input neuron fires, emitting a voltage pulse to the crossbar, through the magnetic tunnel junctions. This voltage leads to currents received by the output neurons. The output neurons integrate these currents and each fires when it has received a certain amount of current. If the system is operating successfully, each output neuron specializes in a specific lane and fires if and only if a car passes through it. The success of this classification task depends on the states of the magnetic tunnel junctions (i.e., synaptic weights) connecting the input and output neurons.

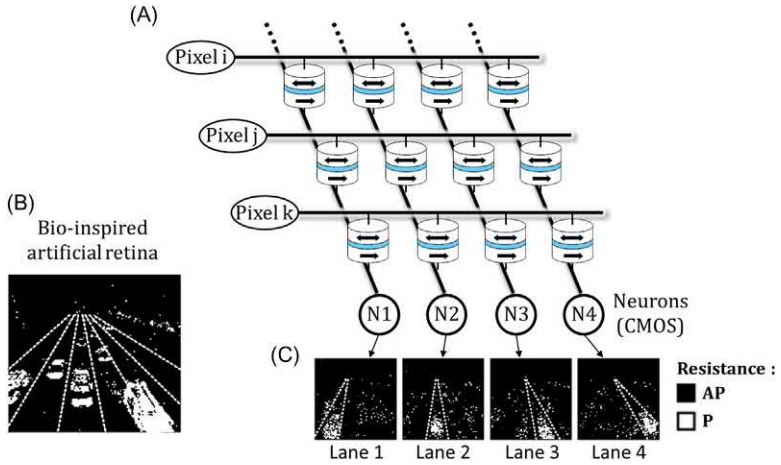


FIGURE 11.12 (A) Input from a bioinspired artificial retina. The dashed lines provide a guide to the eye to delimit the highway lanes. (B) Proposed system: a crossbar of magnetic tunnel junctions. Each line has the value of one pixel of the camera as input. Each column is linked to an output neuron. (C) Resistance map for each column. The black (white) pixels correspond to junctions in the AP (P) state. *Adapted from A.F. Vincent, J. Larroque, N. Locatelli, N.B. Romdhane, O. Bichler, C. Gamrat, et al., Spin-transfer torque magnetic memory as a stochastic memristive synapse for neuromorphic systems, IEEE Trans. Biomed. Circuits Syst. 9 (2015) 166–174, and J. Grollier, D. Querlioz, M.D. Stiles, spintronic nanodevices for bioinspired computing, Proc. IEEE 104 (2016) 2024–2039.*

The synaptic weights are initially random. The input neurons fire pulses accordingly to the data from the artificial retina. First the output neurons fire randomly as the voltage they receive depends on the random weights. Progressively the weights are adjusted by a simple learning rule. The rule is implemented only when an output neuron fires and goes as follows: if the output spike is shortly after an input spike, the weight is incremented. Otherwise the weight is decreased.

After the learning, each output neuron specializes in one specific lane, as can be observed in Fig. 11.12C. Practically, this means that the synapses connecting the considered output neuron and the input neurons corresponding to the pixels of the associated lane have a high conductance (low resistance). The great advantage of this learning method is that it is unsupervised. There is no need to tell the system what are the correct answers (the different lanes in this case). This rule is a simplified version of the spike timing dependent plasticity (STDP), a bioinspired learning rule.

The ability of memristors to exhibit STDP and use it for classification tasks has been shown by several groups [93,94]. Traditionally it requires analog or at least many-level synapses. However, it has been demonstrated that the same results can be achieved with binary synapses which are programmed in a stochastic way [95]. Indeed the requirement for analog weights is due to the fact that the weights need to be only slightly modified by each

spiking event. Otherwise the system is highly sensitive to noise and isolated events (for instance a single event could saturate some weights). If the increase/decrease of the binary weights is implemented with a given probability, it is equivalent at the system level to an increase/decrease by small steps. Using binary stochastic synapses instead of deterministic multilevel ones enables a much simpler implementation as fabricating and controlling multilevel stable devices in a reliable way is challenging. Vincent et al. showed that the magnetic tunnel junction is a particularly appropriate device to emulate a binary stochastic synapse. They also demonstrated that the system exhibits a strong robustness to device variability.

11.3.3 Stochastic computation with superparamagnetic tunnel junctions

We have seen that magnetic tunnel junctions can be used to generate random bits with the spin dice concept. The associated energy consumption is the energy to program a device, and is therefore in the order of picoJoules per bit. Such an energy consumption is attractive for cryptographic applications of random bit generation. However, for computing schemes that rely on stochasticity, it might be too high to allow ultralow-power computation. It is in fact possible to generate random bits with magnetic tunnel junctions at much lower energies. We have indeed already mentioned that these devices become unstable when their area is decreased: they switch between their two states, due to thermal noise only, a property known as superparamagnetism (Fig. 11.13). For memory applications, this behavior is of course unacceptable, but we can by contrast fabricate intentionally superparamagnetic tunnel junctions for stochastic applications [24,25]. In Ref. [96], it is shown that if we measure the states of such junctions at regular frequency, we can get random bits, with extremely reduced energy consumption. No programming operation is indeed needed, and reading the state of a memristive device is much less costly than programming it. The authors estimate, including the different CMOS overhead,

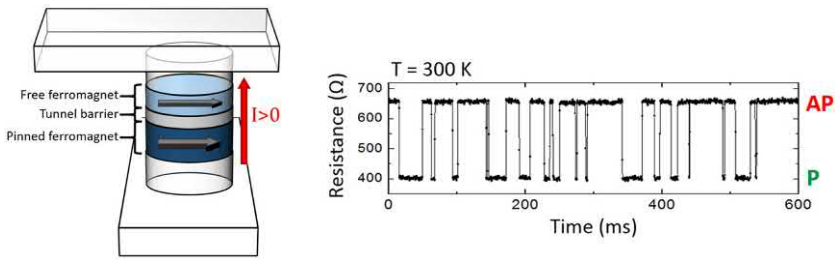


FIGURE 11.13 (A) Simplified illustration of a typical superparamagnetic tunnel junction. (B) Example measurement of the electrical resistance of a superparamagnetic tunnel junction as a function of time.

the required energy at 20 femtojoules per bit, which is orders of magnitude below the CMOS state of the art.

By contrast, these random bits may not be ideal for cryptographic applications: the superparamagnetic tunnel junctions are sensitive and can be influenced by external conditions (magnetic field, temperature). Also the limit for superparamagnetic switching events to remain uncorrelated is the Gigahertz. However, superparamagnetic devices are perfect for all kinds of stochastic computing applications. Ref. [25] shows that superparamagnetic tunnel junctions can exhibit noise-induced synchronization and stochastic resonance. Ref. [96] argued for their use for probabilistic computation, following approaches similar to Section 11.1.3. In particular algorithms of Bayesian reasoning can map very well to implementations with stochastic computing [63–65,97]. Bayesian algorithms manipulate probabilities. Therefore, it is quite natural to compute them with stochastic computing, literally using probabilities to represent probabilities.

11.3.4 Population coding-based stochastic computation

We now present an alternative form of stochastic computation, inspired by the brain, to which memristive and spintronic devices could be especially adapted. To some extent, information coding in traditional stochastic computing of Section 11.1.3 resembles neurons: real numbers are coded by temporal bitstreams on a single wire, a little like neurons that can code some information by spike trains on axons. It is therefore natural to push the analogy further, and develop more brain-inspired stochastic computation. This idea takes special meaning when looking at some neuroscience works on sensory neurons. In several parts of the brain, such as the visual cortex, information appears to be coded by populations of neurons. For example for coding an orientation, the cortex uses a population of neurons, each responding to a special range of angles. The ensemble can code an orientation in a very robust form, which has been theorized extensively in computational neuroscience [80]. In most theoretical models, the answer of the neurons is assumed to be stochastic (following a Poisson process), which allows neurons to compute using models that resemble stochastic computing. Very interestingly this fits well with the answer of some stochastic devices such as the ones described in Section 11.2.3.2, and even better with the behavior of superparamagnetic tunnel junctions [98].

In the case of superparamagnetic tunnel junctions, this is easy to see from Eq. 11.21. If we apply a voltage to a superparagnetic tunnel junction, the mean frequency follows

$$F = \frac{F_0}{cosh\left(\frac{\Delta E}{k_B T}\frac{V}{V_c}\right)}, \tag{11.23}$$

where $F_0 = \frac{1}{2\tau_0 exp\left(\frac{\Delta E}{k_B T}\right)}$. The peak of the tuning curve F_0 is determined by the value of the attempt time $\tau_0 = 1ns$, and the energy over temperature ratio

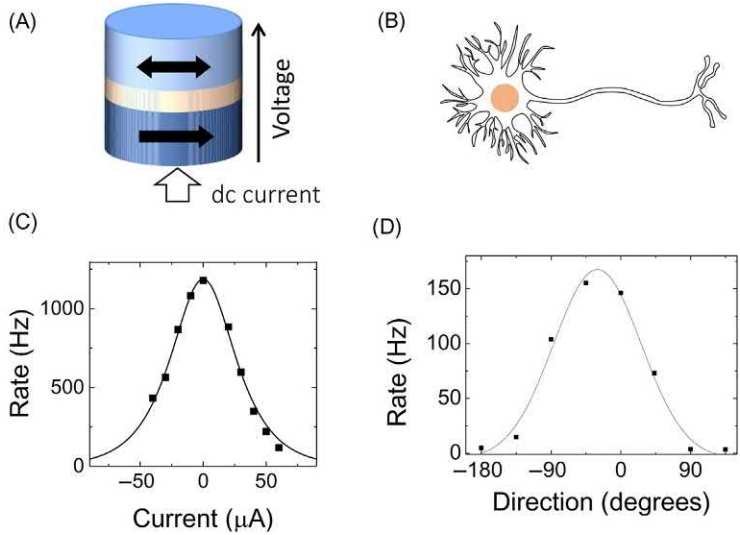


FIGURE 11.14 Comparison between the frequency versus current curve (C) of a superparamagnetic tunnel junction (A) with the tuning curve (D) of a biological sensory neuron (B). *Data from H. Kumano, T. Uka, The spatial profile of macaque MT neurons is consistent with Gaussian sampling of logarithmically coordinated visual representation, J. Neurophysiol. 104 (2010) 61–75.*

$\Delta E/k_BT$. The width of the tuning curve is determined by the value of $\frac{\Delta E}{k_BT}\frac{1}{V_c}$. Just as a neuron with a bell-shaped tuning curve, as shown in Fig. 11.14, the superparamagnetic tunnel junction has a preferred stimulus for which its frequency is maximal ($F = F_0$). When the junction is unbiased, the preferred stimulus is $V_{dc} = 0$ V. This curve is very similar to the tuning curve of sensory neurons. The major difference is that the response of superparamagnetic tunnel junction is a telegraph signal, and not a spiking response. If the junction is biased, the tuning curve can be shifted, allowing the creations of populations, similarly to biology.

In Ref. [98], the authors push this idea to propose a full system based on superparamagnetic tunnel junctions and stochastic computing. The system appears to be very interesting for low energy approximate computation. In particular it naturally includes an analog-to-digital feature: like sensory neurons, it takes an analog signal, codes it in a stochastic population form, and is able to perform computation in these schemes.

11.4 Conclusion

Randomness is a crucial issue faced by all devices used for computing when scaled down to the nanoscale, whether they are CMOS transistors or emerging technologies such as memristive and spintronic devices. Nevertheless

randomness can be provided at a very small energy cost by many physical phenomena which can be exploited for computing applications. Specifically we have targeted and presented noise-induced synchronization and stochastic computing. We showed that these schemes need a stochastic building block to fulfill their potential. We have described several memristive stochastic behaviors and the strategies employed to harness this randomness. Spintronics is an especially interesting case in this regard. Even though spintronics suffers from some limitations (low read-out signals and high programming currents), its flagship device—the magnetic tunnel junction—is an appropriate stochastic building block.

These results once again highlight that the potential of memristive technology can go beyond memory. Their physics can be the basis for alternative stochastic computation schemes.

References

[1] R. Carboni, D. Ielmini, Stochastic memory devices for security and computing, Adv. Electron. Mater. (2019) 1900198.

[2] S. Borkar, Designing reliable systems from unreliable components: the challenges of transistor variability and degradation, IEEE Micro 25 (2005) 10–16.

[3] K. Nikolic, A. Sadek, M. Forshaw, Architectures for reliable computing with unreliable nanodevices, in: Proceedings of the 2001 1st IEEE Conference on Nanotechnology, 2001. IEEE-NANO 2001, pp. 254–259.

[4] D. Ernst, N.S. Kim, S. Das, S. Pant, R. Rao, T. Pham, et al., Razor: a low-power pipeline based on circuit-level timing speculation, in: 36th Annual IEEE/ACM International Symposium on Microarchitecture, 2003. MICRO-36. Proceedings, pp. 7–18.

[5] International Technology Roadmap for Semiconductors, 2001. http://www.itrs2.net/itrs-reports.html.

[6] K.V. Palem, Computational proof as experiment: probabilistic algorithms from a thermodynamic perspective, in: N. Dershowitz (Ed.), Verification: Theory and Practice, Number 2772 in Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2003, pp. 524–547. Available from: https://doi.org/10.1007/978-3-540-39910-0_23.

[7] A. Sampson, J. Nelson, K. Strauss, L. Ceze, Approximate storage in solid-state memories, ACM Trans. Computer Syst. 32 (2014) 1–23.

[8] N. Locatelli, A. Vincent, S. Galdin-Retailleau, J.-O. Klein, D. Querlioz, Approximate programming of magnetic memory elements for energy saving, IEEE (2015) 1–2.

[9] R. Benzi, G. Parisi, A. Sutera, A. Vulpiani, Stochastic resonance in climatic change, Tellus (1982).

[10] L. Gammaitoni, P. HÃnggi, P. Jung, F. Marchesoni, Stochastic resonance, Rev. Mod. Phys. 70 (1998) 223–287.

[11] H.A. Kramers, Brownian motion in a field of force and the diffusion model of chemical reactions, Physica 7 (1940) 284–304.

[12] P. Jung, P. HÃnggi, Stochastic nonlinear dynamics modulated by external periodic forces, Europhys. Lett. (EPL) 8 (1989) 505–510.

[13] A.B. Neiman, D.F. Russell, Synchronization of noise-induced bursts in noncoupled sensory neurons, Phys. Rev. Lett. 88 (2002) 138103.

[14] M.D. McDonnell, Theoretical aspects of stochastic signal quantisation and suprathreshold stochastic resonance., Thesis, 2006.
[15] J.J. Collins, C.C. Chow, A.C. Capela, T.T. Imhoff, Aperiodic stochastic resonance, Phys. Rev. E 54 (1996) 5575–5584.
[16] N.G. Stocks, Suprathreshold stochastic resonance in multilevel threshold systems, Phys. Rev. Lett. 84 (2000) 2310–2313.
[17] E. Simonotto, M. Riani, C. Seife, M. Roberts, J. Twitty, F. Moss, Visual perception of stochastic resonance, Phys. Rev. Lett. 78 (1997) 1186–1189.
[18] F. Moss, L.M. Ward, W.G. Sannita, Stochastic resonance and sensory information processing: a tutorial and review of application, Clin. Neurophysiol. 115 (2004) 267–281.
[19] G. Harmer, B. Davis, D. Abbott, A review of stochastic resonance: circuits and measurement, IEEE Trans. Instrum. Meas. 51 (2002) 299–309.
[20] R.L. Badzey, P. Mohanty, Coherent signal amplification in bistable nanomechanical oscillators by stochastic resonance, Nature 437 (2005) 995–998.
[21] W.J. Venstra, H.J. Westra, H.S. van der Zant, Stochastic switching of cantilever motion, Nat. Commun. 4 (2013).
[22] E. Martinez, G. Finocchio, M. Carpentieri, Stochastic resonance of a domain wall in a stripe with two pinning sites, Appl. Phys. Lett. 98 (2011) 072507.
[23] X. Cheng, C.T. Boone, J. Zhu, I.N. Krivorotov, Nonadiabatic stochastic resonance of a nanomagnet excited by spin torque, Phys. Rev. Lett. 105 (2010) 047202.
[24] N. Locatelli, A. Mizrahi, A. Accioly, R. Matsumoto, A. Fukushima, H. Kubota, et al., Noise-enhanced synchronization of stochastic magnetic oscillators, Phys. Rev. Appl. 2 (2014) 034009.
[25] A. Mizrahi, N. Locatelli, R. Lebrun, V. Cros, A. Fukushima, H. Kubota, et al., Controlling the phase locking of stochastic magnetic bits for ultra-low power computation, Sci. Rep. 6 (2016) 30535.
[26] H.E. Plesser, W. Gerstner, Noise in integrate-and-fire neurons: from stochastic input to escape rates, Neural Computation 12 (2000) 367–384.
[27] A. Patel, B. Kosko, Stochastic resonance in continuous and spiking neuron models with levy noise, IEEE Trans. Neural Netw. 19 (2008) 1993–2008.
[28] J.J. Collins, T.T. Imhoff, P. Grigg, Noise-enhanced information transmission in rat SA1 cutaneous mechanoreceptors via aperiodic stochastic resonance, J. Neurophysiol. 76 (1996) 642–645.
[29] I. Hidaka, D. Nozaki, Y. Yamamoto, Functional stochastic resonance in the human brain: noise induced sensitization of baroreflex system, Phys. Rev. Lett. 85 (2000) 3740–3743.
[30] M.D. McDonnell, D. Abbott, What is stochastic resonance? definitions, misconceptions, debates, and its relevance to biology, PLoS Comput. Biol. 5 (2009) e1000348.
[31] A. Neiman, A. Silchenko, V. Anishchenko, L. Schimansky-Geier, Stochastic resonance: Noise-enhanced phase coherence, Phys. Rev. E 58 (1998) 7118–7125.
[32] A. Neiman, L. Schimansky-Geier, F. Moss, B. Shulgin, J.J. Collins, Synchronization of noisy systems by stochastic signals, Phys. Rev. E 60 (1999) 284–292.
[33] A.S. Pikovsky, J. Kurths, Coherence resonance in a noise-driven excitable system, Phys. Rev. Lett. 78 (1997) 775–778.
[34] W. Horsthemke, Noise induced transitions, in: P.D.C. Vidal, P.D.A. Pacault (Eds.), Non-Equilibrium Dynamics in Chemical Systems, number 27 in Springer Series in Synergetics, Springer Berlin Heidelberg, 1984, pp. 150–160. Available from: https://doi.org/10.1007/978-3-642-70196-2_23.
[35] F. Monifi, J. Zhang, A.K. Zdemir, B. Peng, Y.-X. Liu, F. Bo, et al., Optomechanically induced stochastic resonance and chaos transfer between optical fields, Nat. Photonics 10 (2016) 399–405.

[36] G. Stegemann, A.G. Balanov, E. Schll, Noise-induced pattern formation in a semiconductor nanostructure, Phys. Rev. E 71 (2005) 016221.
[37] C.J. Tessone, C.R. Mirasso, R. Toral, J.D. Gunton, Diversity-induced resonance, Phys. Rev. Lett. 97 (2006) 194101.
[38] J. Lengler, F. Jug, A. Steger, Reliable neuronal systems: the importance of heterogeneity, PLoS ONE 8 (2013) e80694.
[39] M.D. McDonnell, L.M. Ward, The benefits of noise in neural systems: bridging theory and experiment, Nat. Rev. Neurosci. 12 (2011) 415−426.
[40] B. Shulgin, A. Neiman, V. Anishchenko, Mean switching frequency locking in stochastic bistable systems driven by a periodic force, Phys. Rev. Lett. 75 (1995) 4157−4160.
[41] S. Barbay, G. Giacomelli, F. Marin, Stochastic resonance in vertical cavity surface emitting lasers, Phys. Rev. E 61 (2000) 157.
[42] S. Bahar, A. Neiman, L.A. Wilkens, F. Moss, Phase synchronization and stochastic resonance effects in the crayfish caudal photoreceptor, Phys. Rev. E 65 (2002) 050901.
[43] C. Kurrer, K. Schulten, Noise-induced synchronous neuronal oscillations, Phys. Rev. E 51 (1995) 6213−6218.
[44] B.J. Gluckman, T.I. Netoff, E.J. Neel, W.L. Ditto, M.L. Spano, S.J. Schiff, Stochastic resonance in a neuronal network from mammalian brain, Phys. Rev. Lett. 77 (1996) 4098−4101.
[45] T. Aonishi, Phase transitions of an oscillator neural network with a standard Hebb learning rule, Phys. Rev. E 58 (1998) 4865−4871.
[46] F.C. Hoppensteadt, E.M. Izhikevich, Oscillatory neurocomputers with dynamic connectivity, Phys. Rev. Lett. 82 (1999) 2983−2986.
[47] M.R. Pufall, W.H. Rippard, G. Csaba, D.E. Nikonov, G.I. Bourianoff, W. Porod, Physical implementation of coherently coupled oscillator networks, IEEE J. Exploratory Solid-State Computational Devices Circuits 1 (2015) 76−84.
[48] K. Yogendra, D. Fan, K. Roy, Coupled spin torque nano oscillators for low power neural computation, IEEE Trans. Magnetics 51 (2015) 1−9.
[49] C.E. Shannon, J. McCarthy, Automata studies. (AM-34), Princeton University Press, 2016. Google-Books-ID: adLfCwAAQBAJ.
[50] W.J. Poppelbaum, C. Afuso, J.W. Esch, Stochastic computing elements and systems, ACM Press, 1967, p. 635.
[51] B.R. Gaines, Stochastic computing systems, in: J.T. Tou (Ed.), Advances in Information Systems Science, Advances in Information Systems Science, Springer, US, 1969, pp. 37−172.
[52] A. Alaghi, J.P. Hayes, Survey of stochastic computing, ACM Trans. Embedded Comput. Syst. 12 (2013) 1−19.
[53] P. Mars, H. Mclean, High-speed matrix inversion by stochastic computer, Electron. Lett. 12 (1976) 457.
[54] S. Toral, J. Quero, L. Franquelo, Stochastic pulse coded arithmetic, Presses Polytech. Univ. Romandes (2000) 599−602.
[55] B. Brown, H. Card, Stochastic neural computation. I. Computational elements, IEEE Trans. Computers 50 (2001) 891−905.
[56] V. Vujicic, S. Milovancev, M. Pesaljevic, D. Pejic, I. Zupunski, Low-frequency stochastic true RMS instrument, IEEE Trans. Instrum. Meas. 48 (1999) 467−470.
[57] T. Hammadou, M. Nilson, A. Bermak, P. Ogunbona, A 96 Ã— 64 intelligent digital pixel array with extended binary stochastic arithmetic, volume 4, IEEE, 2003, pp. IV−772−IV−775.

[58] A. Morro, V. Canals, A. Oliver, M.L. Alomar, J.L. Rossello, Ultra-fast data-mining hardware architecture based on stochastic computing, PLoS ONE 10 (2015) e0124176.
[59] X.-R. Lee, C.-L. Chen, H.-C. Chang, C.-Y. Lee, A 7.92 Gb/s 437.2 mW stochastic LDPC decoder chip for IEEE 802.15. 3c applications, IEEE Trans. Circuits Syst. I: Regul. Pap. 62 (2015) 507–516.
[60] R. Frisch, R. Laurent, M. Faix, L. Girin, L. Fesquet, A. Lux, et al., A bayesian stochastic machine for sound source localization, in: Rebooting Computing (ICRC), 2017 IEEE International Conference on, IEEE, pp. 1–8.
[61] A. Coelho, R. Laurent, M. Solinas, J. Fraire, E. Mazer, N.-E. Zergainoh, et al., On the robustness of stochastic Bayesian machines, IEEE Trans. Nucl. Sci. 64 (2017) 2276–2283.
[62] E.O. Neftci, B.U. Pedroni, S. Joshi, M. Al-Shedivat, G. Cauwenberghs, Stochastic synapses enable efficient brain-inspired learning machines, Front. Neurosci. 10 (2016).
[63] J.S. Friedman, L.E. Calvet, P. Bessiere, J. Droulez, D. Querlioz, Bayesian inference with Muller C-elements, IEEE Trans. Circuits Syst. I: Regul. Pap. 63 (2016) 895–904.
[64] A. Coninx, R. Laurent, M.A. Aslam, J. Lobo, P. Bessière, E. Mazer, et al., Bayesian sensor fusion with fast and low power stochastic circuits, in: Rebooting Computing (ICRC), IEEE International Conference on, IEEE, pp. 1–8.
[65] M. Faix, R. Laurent, P. Bessière, E. Mazer, J. Droulez, Design of stochastic machines dedicated to approximate Bayesian inferences, IEEE Transactions on Emerging Topics in Computing (2016).
[66] V. Canals, A. Morro, A. Oliver, M.L. Alomar, J.L. RossellÃ, A new stochastic computing methodology for efficient neural network implementation, IEEE Trans. Neural Netw. Learn. Syst. 27 (2016) 551–564.
[67] T.J. Hamilton, S. Afshar, A. van Schaik, J. Tapson, Stochastic electronics: a neuro-inspired design paradigm for integrated circuits, Proc. IEEE 102 (2014) 843–859.
[68] S. Wolfram, A New Kind of Science (2002).
[69] M. Schle, T. Ott, R. Stoop, Computing with probabilistic cellular automata, in: D. Hutchison, T. Kanade, J. Kittler, J.M. Kleinberg, F. Mattern, J.C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M.Y. Vardi, G. Weikum, C. Alippi, M. Polycarpou, C. Panayiotou, G. Ellinas (Eds.), Artificial Neural Networks ICANN 2009, volume 5769, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 525–533.
[70] A.O. Orlov, I. Amlani, G.H. Bernstein, C.S. Lent, G.L. Snider, Realization of a functional cell for quantum-dot cellular automata, Science 277 (1997) 928–930.
[71] T. Purkayastha, D. De, K. Das, A novel pseudo random number generator based cryptographic architecture using quantum-dot cellular automata, Microprocessors Microsyst. 45 (2016) 32–44.
[72] T.J. Dysart, P.M. Kogge, Probabilistic Analysis of a Molecular Quantum-Dot Cellular Automata Adder, in: 22nd IEEE International Symposium on Defect and Fault-Tolerance in VLSI Systems (DFT'07), 2007, pp. 478–486.
[73] Y. Benenson, T. Paz-Elizur, R. Adar, E. Keinan, Z. Livneh, E. Shapiro, Programmable and autonomous computing machine made of biomolecules, Nature 414 (2001) 430–434.
[74] R. Adar, Y. Benenson, G. Linshiz, A. Rosner, N. Tishby, E. Shapiro, Stochastic computing with biomolecular automata, Proc. Natl Acad. Sci. 101 (2004) 9960–9965.
[75] V. Helms, Principles of Computational Cell Biology, John Wiley & Sons, 2008.
[76] S. Wang, A.R. Lebeck, C. Dwyer, Nanoscale resonance energy transfer-based devices for probabilistic computing, IEEE Micro 35 (2015) 72–84.

[77] S. Gaba, P. Sheridan, J. Zhou, S. Choi, W. Lu, Stochastic memristive devices for computing and neuromorphic applications, Nanoscale 5 (2013) 5872.
[78] S. Balatti, S. Ambrogio, Z. Wang, D. Ielmini, True random number generation by variability of resistive switching in oxide-based devices, IEEE J. Emerg. Sel. Top. Circuits Syst. 5 (2015) 214–221.
[79] T. Tuma, A. Pantazi, M. Le Gallo, A. Sebastian, E. Eleftheriou, Stochastic phase-change neurons, Nat. Nanotechnol. 11 (2016) 693–699.
[80] B.B. Averbeck, P.E. Latham, A. Pouget, Neural correlations, population coding and computation, Nat. Rev. Neurosci. 7 (2006) 358–366.
[81] A.D. Kent, D.C. Worledge, A new spin on magnetic memories, Nat. Nanotechnol. 10 (2015) 187–191.
[82] E. Chen, D. Apalkov, Z. Diao, A. Driskill-Smith, D. Druist, D. Lottis, et al., Advances and future prospects of spin-transfer torque random access memory, IEEE Trans. Magnetics 46 (2010) 1873–1878.
[83] H. Sato, E.C.I. Enobio, M. Yamanouchi, S. Ikeda, S. Fukami, S. Kanai, et al., Properties of magnetic tunnel junctions with a MgO/CoFeB/Ta/CoFeB/MgO recording structure down to junction diameter of 11 nm, Appl. Phys. Lett. 105 (2014) 062403.
[84] D. Ralph, M. Stiles, Spin transfer torques, J. Magnetism Magnetic Mater. 320 (2008) 1190–1216.
[85] W.F. Brown, Thermal fluctuations of a single-domain particle, Phys. Rev. 130 (1963) 1677–1686.
[86] H. Kubota, A. Fukushima, K. Yakushiji, T. Nagahama, S. Yuasa, K. Ando, et al., Quantitative measurement of voltage dependence of spin-transfer torque in MgO-based magnetic tunnel junctions, Nat. Phys. 4 (2008) 37–41.
[87] J. Slonczewski, Current-driven excitation of magnetic multilayers, J. Magnetism Magnetic Mater. 159 (1996) L1–L7.
[88] Z. Li, S. Zhang, Thermally assisted magnetization reversal in the presence of a spin-transfer torque, Phys. Rev. B 69 (2004) 134416.
[89] A. Fukushima, T. Seki, K. Yakushiji, H. Kubota, H. Imamura, S. Yuasa, et al., Spin dice: A scalable truly random number generator based on spintronics, Appl. Phys. Express 7 (2014) 083001.
[90] J. Sun, Spin angular momentum transfer in current-perpendicular nanomagnetic junctions, IBM J. Res. Dev. 50 (2006) 81–100.
[91] A.F. Vincent, J. Larroque, N. Locatelli, N.B. Romdhane, O. Bichler, C. Gamrat, et al., Spin-transfer torque magnetic memory as a stochastic memristive synapse for neuromorphic systems, IEEE Trans. Biomed. Circuits Syst. 9 (2015) 166–174.
[92] J. Grollier, D. Querlioz, M.D. Stiles, Spintronic nanodevices for bioinspired computing, Proc. IEEE 104 (2016) 2024–2039.
[93] T. Serrano-Gotarredona, T. Masquelier, T. Prodromakis, G. Indiveri, B. Linares-Barranco, Stdp and stdp variations with memristors for spiking neuromorphic learning systems, Front. Neurosci. 7 (2013) 2.
[94] O. Bichler, D. Querlioz, S.J. Thorpe, J.-P. Bourgoin, C. Gamrat, Extraction of temporally correlated features from dynamic vision sensors with spike-timing-dependent plasticity, Neural Netw. 32 (2012) 339–348.
[95] W. Senn, S. Fusi, Convergence of stochastic learning in perceptrons with binary synapses, Phys. Rev. E 71 (2005).

[96] D. Vodenicarevic, N. Locatelli, A. Mizrahi, J.S. Friedman, A.F. Vincent, M. Romera, et al., Low-energy truly random number generation with superparamagnetic tunnel junctions for unconventional computing, Phys. Rev. Appl. 8 (2017) 054045.

[97] M. Faix, E. Mazer, R. Laurent, M.O. Abdallah, R. Le Hy, J. Lobo, Cognitive computation: a bayesian machine case study, in: 2015 IEEE 14th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC), IEEE, pp. 67–75.

[98] A. Mizrahi, T. Hirtzlin, A. Fukushima, H. Kubota, S. Yuasa, J. Grollier, et al., Neural-like computing with populations of superparamagnetic basis functions, Nat. Commun. 9 (2018) 1533.

[99] H. Kumano, T. Uka, The spatial profile of macaque mt neurons is consistent with gaussian sampling of logarithmically coordinated visual representation, J. Neurophysiol. 104 (2010) 61–75.

Part III

Deep learning

Chapter 12

Memristive devices for deep learning applications

Damien Querlioz[1], **Sabina Spiga**[2], **Abu Sebastian**[3] and **Bipin Rajendran**[4]

[1]*Centre for Nanoscience and Nanotechnology, Universite Paris-Saclay, Palaiseau, France,* [2]*CNR−IMM, Agrate Brianza (MB), Italy,* [3]*IBM Research − Zurich, Rüschlikon, Switzerland,* [4]*Department of Engineering, Kings College London, London, United Kingdom*

The progress of deep learning in recent years—relying on deep neural networks—has driven an incredible progress in artificial intelligence [1]. Computers can now outperform professional go and poker players, recognize images better than humans in many situations, and convincingly tackle complex tasks such as text translations. These advances are changing many professional fields and might even transform our societies.

Nevertheless deep learning comes with challenges. One of them is its important energy consumption [2]. In the widely publicized first victory of a computer against a professional Go player, the computer cluster consumed hundreds and thousands of watts [3], whereas a human player played only using his brain, which consumed only 20 W. Despite considerable progress since this game [4], the energy cost of operating deep neural networks means that they are typically run in data centers and not on consumer-embedded devices. This nonlocality adds to their power consumption the cost of transferring data between data center and users and raises major privacy concerns. The energy consumption of data centers is also starting to become a serious environmental concern.

These considerations drive a considerable effort to develop specialized hardware for implementing deep learning. This chapter serves as an introduction to the next two chapters, which show how memristive devices can be a key asset in this quest. In the first section we introduce the general concepts of deep neural networks and their modern evolution. This section does not intend to be a comprehensive textbook about this topic, as can be found elsewhere [1]. It means to give an intuitive feel of the general ideas necessary to approach the literature. In the second section we answer the question of why

Memristive Devices for Brain-Inspired Computing. DOI: https://doi.org/10.1016/B978-0-08-102782-0.00012-5

deep neural networks operated on graphics cards or computers consume so much energy than the brain. This allows us to understand why memristive devices have so much potential for improving the energy efficiency of deep learning, as well as the associated challenges.

12.1 Quick introduction to deep learning

12.1.1 Simple neural network

Fig. 12.1 presents a very simple neural network. It is composed of N input neurons, one output neurons, and N synapses that connect the input neurons to the output neuron. The input neurons represent an image: each of them is connected to a pixel of the image, and the value of the input neuron (between 0 and 1) is the grayness level of the pixel. Each synapse is associated with a real-valued "synaptic weight," noted as w_i if the synapse is connected to the i-th input neuron. These values constitute the parameters of the neural network: a set of w_i values turn the neural network into a cat detector, whereas a different set of values could make it a dog detector.

The value of the output neuron is computed using simple equations. We first compute the sum of the inputs weighted by the synaptic weights:

$$z = \sum_{i=1}^{N} w_i x_i. \tag{12.1}$$

A nonlinear "activation function" is then applied to this sum. Different choices are possible for the activation function, which is discussed later. Here we take a sigmoid function:

$$a = \sigma(z) = \frac{1}{1 + e^{-z}}, \tag{12.2}$$

whose values can be between 0 and 1.

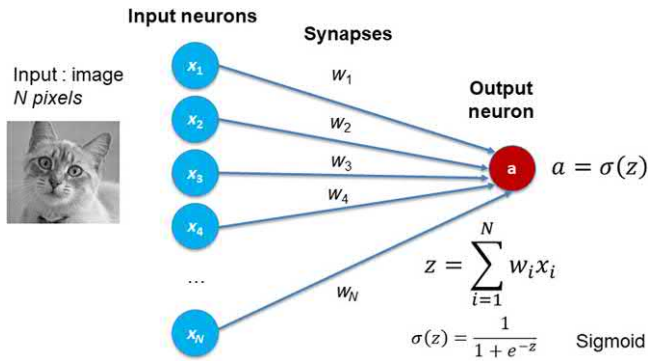


FIGURE 12.1 The simple one-layer cat detector network.

We now would want our neural network to become a cat detector. This means that whenever an image of a cat is presented to the network, the output neuron value a approaches 1, and whenever this image is not a cat, a approaches 0. The first step is to achieve, that is to be able to measure how well the network is doing. For this we need a metric. Here we present a metric that has been successful. If we present a cat to the network, we introduce the "cost" value:

$$L = -\ln a. \tag{12.3}$$

If a approaches 1, the neural network is giving a correct answer, then L approaches 0. If a is not 1, then L is a positive value, and if a is from 1, then L is larger. On the other hand if the presented image is not a cat, then a is as close as possible to 0. An appropriate cost value is:

$$L = -\ln (1 - a). \tag{12.4}$$

If a is close to 0, L is close to 0. If a is not 0, L takes a positive value.

It is possible to combine Eqs. (12.3) and (12.4) into a single equation, which is more convenient for mathematical derivations. We introduce $y = 1$ if the presented image is a cat, and $y = 0$ if it is not a cat. Then a general equation for L is:

$$L = -y \ln a - (1 - y)\ln(1 - a). \tag{12.5}$$

Now let us imagine that we have a collection of images, some of them are cats, some of are not, which are used to train our network to recognize cats (this is called a "training set"). We can calculate a total cost function as

$$J = \sum_{\text{all images}} L. \tag{12.6}$$

The J value indicates how good the neural network is working on the whole dataset. When we train the neural network, our goal is to find the w_i values that make the neural network do as well as possible on the whole dataset, and therefore minimize J. For this purpose we use a famous optimization algorithm known as gradient descent. This algorithm consists in computing, for all synapses, $\partial J/\partial w_i$. Then we update the weights as

$$w_i \leftarrow w_i - \alpha \frac{\partial J}{\partial w_i}, \tag{12.7}$$

where α is a parameter known as "learning rate," which should be chosen carefully. We then repeat this procedure until the algorithm converges and the cost value J reaches a stable value.

Computing $\partial J/\partial w_i$ is surprisingly easy. As

$$\frac{\partial J}{\partial w_i} = \sum_{\text{all images}} \frac{\partial L}{\partial w_i}, \tag{12.8}$$

We only need to compute the $\partial L/\partial w_i$. Based on Eq. (12.5) using the chain rule, we can drive them as:

$$\frac{\partial L}{\partial w_i} = \frac{dL}{da}\frac{\partial a}{\partial w_i} \tag{12.9}$$

$$= \frac{dL}{da}\frac{da}{dz}\frac{\partial z}{\partial w_i} \tag{12.10}$$

Now we can use the property of the sigmoid function $da/dz = a(1-a)$, and we find,

$$\frac{\partial L}{\partial w_i} = \left(-\frac{y}{a} + \frac{1-y}{1-a}\right) a(1-a)x_i \tag{12.11}$$

$$\frac{\partial L}{\partial w_i} = (a-y)x_i, \tag{12.12}$$

which is a very simple equation and shows that training a one-layer neural network is actually a simple task.

12.1.2 Backpropagation

Unfortunately, one-layer neural networks are very limited. A cat detector trained this way will make mistakes, considerably more than humans. A solution to improve the performance of such networks, known since the 1980s [5] is to add more layers to the neural network. Let us look at a simple case, where we have added a single layer of neurons between the input neurons and the output neurons (Fig. 12.2). Such neurons are called "hidden neurons" as their value is used only internally by the neural network.

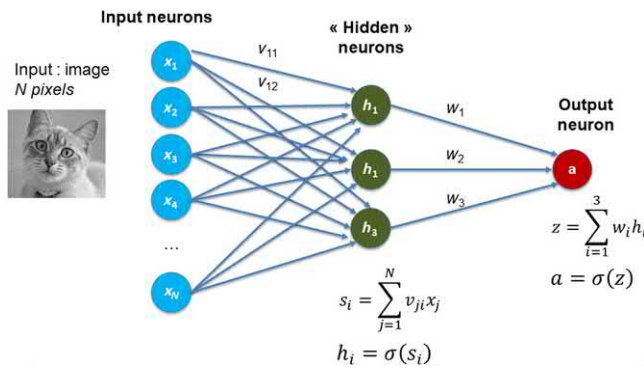


FIGURE 12.2 The two-layer cat detector network.

v_{ij} is the weight connecting input neuron j and hidden neurons i, and w_i is the weight connecting hidden neuron i to the output neuron. To compute the hidden neurons values, we first calculate the weighted sums:

$$s_i = \sum_{j=1}^{N} v_{ij} x_j, \tag{12.13}$$

and then apply a sigmoid activation function.

$$h_i = \sigma(s_i) \tag{12.14}$$

The value of the output neuron is computed as in the one-layer case with:

$$z = \sum_{i=1}^{3} w_i h_i, \tag{12.15}$$

and

$$a = \sigma(z). \tag{12.16}$$

To apply gradient descent, we need the values of $\partial L/\partial w_i$ and $\partial L/\partial v_{ij}$. For $\partial L/\partial w_i$, the derivation in the one-layer case still holds, and we find:

$$\frac{\partial L}{\partial w_i} = (a - y) h_i \tag{12.17}$$

For $\partial L/\partial v_{ij}$, we use a powerful approach known as "error backpropagation," which consists in applying the chain rule repeatedly:

$$\frac{\partial L}{\partial v_{ij}} = \frac{dL}{da}\frac{da}{dz}\frac{\partial z}{\partial v_{ij}} \tag{12.18}$$

$$= \frac{dL}{da}\frac{da}{dz}\sum_{k=1}^{3}\frac{\partial z}{\partial h_k}\frac{\partial h_k}{\partial v_{ij}} \tag{12.19}$$

In this simple case, only one term in the sum is non-zero:

$$\frac{\partial L}{\partial v_{ij}} = \frac{dL}{da}\frac{da}{dz}\frac{\partial z}{\partial h_i}\frac{\partial h_i}{\partial v_{ij}} \tag{12.20}$$

$$= \frac{dL}{da}\frac{da}{dz}\frac{\partial z}{\partial h_i}\frac{dh_i}{ds_i}\frac{\partial s_i}{\partial v_{ij}} \tag{12.21}$$

It is then easy to obtain:

$$\frac{\partial L}{\partial v_{ij}} = (a - y) w_i h_i (1 - h_i) x_j \tag{12.22}$$

We can generalize the backpropagation approach to neural networks with more layers. The derivation is very straightforward, but the equations

become more complicated. For example if we add another layer of hidden neurons g_i between the input layer and the hidden neurons h_i, with weight u_{kl} (Fig. 12.3), we find:

$$\frac{\partial L}{\partial u_{kl}} = (a - y) \sum_{i=1}^{3} w_i h_i (1 - h_i) v_{ki} g_k (1 - g_k) x_l. \tag{12.23}$$

The derivation in the general case (arbitrary number of layers) can be achieved using simple linear algebra and can be found in most neural network textbooks [1].

12.1.3 Why going deep helps?

In recent developments of artificial intelligence we use neural networks with many layers—sometimes hundreds—therefore called "deep" neural networks. However it has not always been obvious that having many layers would be a beneficial thing. In fact, it has been shown mathematically that all functions that can be learned by a neural network can be learned by a neural network with only one layer of hidden neurons (universal approximation theorem) [6].

However this mathematical result is misleading: solving difficult tasks with a single hidden layers can require unrealistically large numbers of parameters, while such tasks may be tractable with deep neural networks. For a long time, this fact was misunderstood, as training deep neural networks is actually a difficult task. When we apply the chain rule repeatedly as in Eq. (12.23), the results can becomes very small values, a phenomenon known as "vanishing gradient" effect. Due to this effect, it can be very hard to learn the weights of the first layers, as they get barely changed every time Eq. (12.7) is applied.

However in recent years, the point of view on this question has changed radically. In 2012 Geoffrey Hinton and his coworkers showed that a deep

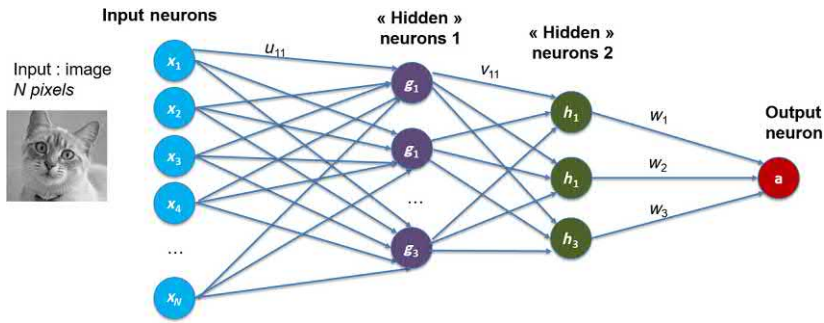


FIGURE 12.3 The three-layer cat detector network.

learning neural network could perform image recognition on a difficult dataset (ImageNet) with a performance that has never been achieved before by an artificial neural network [7]. This has led to explosion of works in the field of deep learning [8], and numerous achievements in artificial intelligence, such as victory in professional go board game players [3].

Why are deep neural network now so successful, after having been overlooked for a long time? First, the neuron activation function is not necessarily a sigmoid anymore. It can be a tanh function, or most often a rectifying linear unit (ReLU) function: $ReLU\ (z) = \max(0,\ z)$, as such activation function reduces the vanishing gradient effect and lead to neural networks that are easier to train.

But most importantly, we have now access to gigantic amounts of data and to very big computers. This allows training deep neural networks on billions of examples and to train all layers of weights, despite the vanishing gradient issue. Actually big neural networks are typically trained on graphics processing units (GPUs) and not on central processing units (CPUs). The parallelism of GPUs allows accelerating deep learning considerably.

12.1.4 Modern deep neural networks

Modern deep neural networks differ significantly from the simple networks that we have presented.

12.1.4.1 Multiple output neural networks

First neural networks do not usually have a single output. A neural network that is solely a cat detector is not that useful, usually, one wants to recognize cats, dogs, humans, cars, etc. In a neural network that has several output neurons a_i with associated weighted sum z_i, the "softmax" activation function usually gives the best result:

$$a_i = \frac{e^{z_i}}{\sum_j e^{z_j}}, \tag{12.24}$$

where the denominator is a sum of all output neurons. This activation function naturally leads to situations where a single output neuron has a value close to one and usually leads to the best result for classification tasks. In the case of a single output neuron, it reduces to a sigmoid activation.

12.1.4.2 Convolutional and recurrent neural networks

The architecture of neural networks that we have presented until now is called "fully connected": each neuron in a layer is connected to each neuron in the next layer. This kind of networks is very powerful, but the number of synapses can become extremely high. For example let us imagine a colored 1000×1000 pixels image, which is relatively low resolution. Presenting the

input requires $3 \times 1000 \times 1000 = 3{,}000{,}000$ pixels (factor 3 corresponding to the three Red/Green/Blue colors). If these neurons are fully connected to 10,000 hidden neurons, thirty billion synapses will be required! This is a huge number, which means high memory usage, high computational cost, and above all a high difficulty for training a model with so many parameters. The solution in wide use today is to rely on "convolutional neural networks," as illustrated in Fig. 12.4A [7]. Such networks have three distinct types of layers: convolutional, pooling, and fully connected.

The convolutional layers constituted weights kernels, which can be applied on small pieces of the input. These kernels are applied to the inputs as convolutional kernels, that is, the input is divided into subimages, and the same kernels are applied to all subimages. Typically, these kernels identify patterns, such as angles in the subimages.

After one or several convolutional layers, the neural network usually features a pooling layer (also called subsampling layers). This type of layer reduces the number of neurons, by averaging several neurons or retaining only the highest valued neurons. This allows compressing the information and going toward more abstracted representations.

After an ensemble of convolutional and pooling layers, the neural network is finished by an ensemble of fully connected layers, similar to the ones given in the previous section.

Convolutional neural networks are trained by backpropagation through the entire network, just like the fully connected network presented

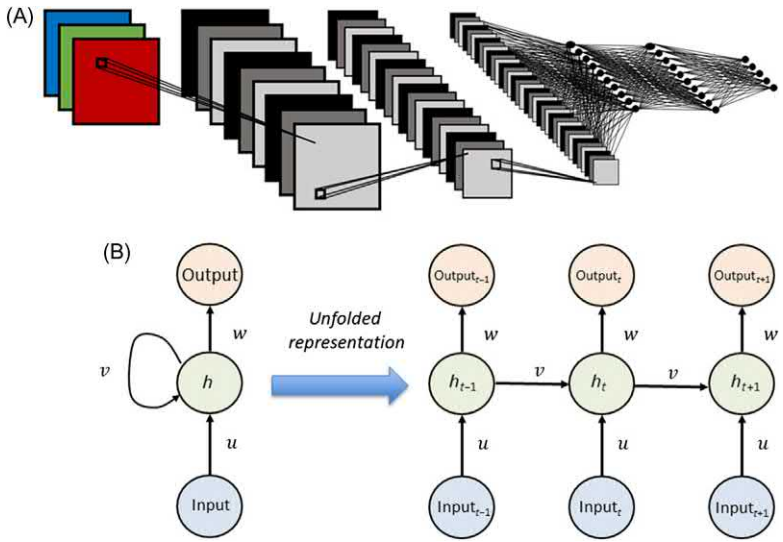


FIGURE 12.4 Schematization of (A) a convolutional neural network and (B) a recurrent neural network. *u*, *v*, and *w* are weight values.

previously. Convolutional neural networks have reached astonishing performance, often outperforming humans on image recognition tasks. Nevertheless it is not always clear to understand how they actually work. What seems clear is that the layer start by identifying relatively low level features in the images, such as edges, and the more we progress in the network the more abstract the features become. This is relatively similar to the working of human visual cortex.

Another challenge of the neural networks that we have presented is that they feature a fixed number of inputs. This is fine for image processing applications, but it is a challenge for others. For example in natural language processing tasks such as text translation, sentences have varying number of words. An adaptation of neural networks can deal with inputs such as recurrent neural networks (Fig. 12.4B).

In a recurrent neural network, the state of a neuron does not depend only on the state of the neurons in the previous layer, but also on the state of the neurons of the current layer at the previous time step. This way, the neural network features memory: applying the same input to the neural network (e.g., a word) will have a different effect depending on the inputs that came before. This also makes these networks particularly useful on language processing tasks, where the meaning of words is extremely context dependent.

Recurrent neural networks are also trained by error backpropagation, which might sound difficult, but is well illustrated in Fig. 12.4B: backpropagation can be applied "through time," as if the previous states of the layer in time were previous layers in a deep neural network.

Today the most widely used recurrent neural network is the Long Short-Term Memory (LSTM) [9]. The LSTM has a relatively complicated architecture, which allows it to remember information over varying time scales. This is invaluable in practice, as in a sequence, the timescale along which the sequence should be interpreted vary greatly (e.g., in language), sometimes long or short sequences of words "go together." LSTMs are therefore very flexible and have been used in very different kinds of contexts.

12.1.4.3 Techniques for implementing learning

The last evolution between the neural networks that we have described and modern deep neural networks is that they typically use learning rule that are a little more complex than Eq. (12.7). A challenge of Eq. (12.7), for example, is that it can lead easily to an oscillation of the weight values around their optimal values, avoiding the convergence. A commonly used technique, among others, to improve this concern is the Adaptive Moment Estimation (Adam) algorithm [10], which requires several additional parameters and variables.

Additionally it is often not optimal to apply Eq. (12.7) with $\partial J/\partial w_i$ calculated hundreds of examples, known as mini-batches, allows updating the weights more frequently and leads to faster learning.

Overall getting a neural network to learn efficiently requires significant tuning, and a trial and error process. This process is very empirical and relies on the experience of the practitioner. In practice this work is usually done using deep learning frameworks, such as Tensorflow [11], PyTorch [12], or Caffe [13]. Such libraries are convenient to use, as they calculate to perform backpropagation algorithm automatically, even in complicated neural networks. And they are extremely optimized for high performances on CPUs or GPUs.

12.2 Why do deep neural networks consume more energy than the brain, and how memristive devices can help

12.2.1 Separation of logic and memory

A fundamental difference between the operation of deep neural networks on CPUs or GPUs and the brain is how they deal with memory [14–16]. Processors, even in their most modern implementations, are tied to the von Neumann architecture of computers. One of the most fundamental principle of this architecture is the separation between processing elements and memory. Let us illustrate this principle for example with the computation of the neural network equation: $z = \sum_{i=1}^{N} w_i x_i$ (Fig. 12.5). Processors feature floating point units that can do additions and multiplications very efficiently. Values of w_i and x_i are stored in memory. To compute the equation

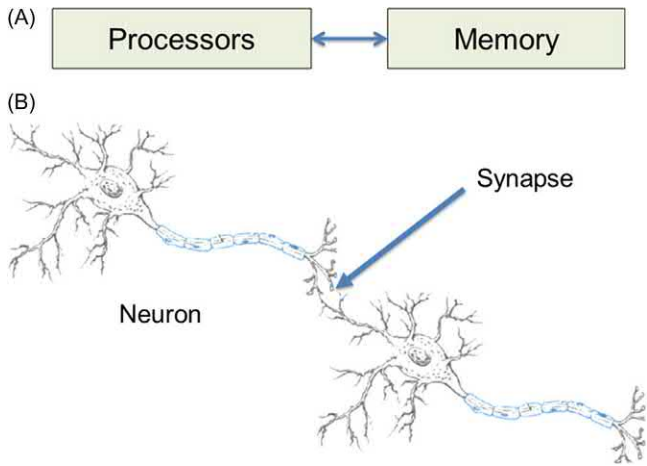


FIGURE 12.5 (A) Computers separate logic and memory physically, whereas (B) the brain colocates neurons and synapses.

(N additions and N multiplications), the processor needs to load $2N$ values in memory. However the cost of reading information from memory, in modern computers, is considerably higher than the cost of arithmetic operations [17]: this means that the ultra-dominant energy cost of operating neural networks on processors is actually accessing memory! This importance of memory access on energy consumption is more pronounced for neural networks than most algorithms, as the ratio of memory access to actual computation is particularly high: this is due to the high number of adjustable parameters in a neural network and the relatively simple equations that they are based on. The brain has an entirely different strategy: it does not separate arithmetic operation and memory. The synapses do an analogue for "multiplication" locally, while neurons do an analogue for "addition" locally (see Part IV for precise operations done by biological neurons and synapses). The brain does not need to access a remote memory. Therefore the ultra-dominant source of energy consumption when implementing neural networks on computers just does not exist in the brain.

In this discussion we have focused on energy consumption. In the past to compute $z = \sum_{i=1}^{N} w_i x_i$ would have needed to do all additions and multiplications sequentially (one after one). Today with multicore CPUs and GPUs it is no longer the case: efficient implementations of neural networks are massively parallel. Therefore the largest benefit of the brain's strategy of associating computing and memory is more about energy than speed.

These insights suggest that memristive devices can be a major lead to reduce the energy cost of neural network inference. Inference in deep neural networks can indeed be done within memristive devices. This is due to the nature of inference equations of neural networks such as Eq. (12.1) or (12.13). These equations are local topologically. This means that we can design a system with a topology imitating the topology of the neural network and perform the arithmetic operations very close to the memory. In the next two chapters we show that neural network inference is an excellent application for this idea, which allows operating neural network in a way that is closer to how the brain neural networks and can save considerable energy.

Implementing neural network learning within memory is a more difficult problem. Backpropagation equations, Eq. (12.22) or (12.23), are indeed not local topologically. The next two chapters discuss ideas for implementing them in memory.

12.2.2 Reliance on approximate computing

A second reason for which the brain is more energy efficient than deep neural networks operated on CPUs or GPUs comes from the nature of computation performed by these systems. CPUs and GPUs compute neural networks with floating point operations, which are extremely precise. This precision comes with an important energy cost. The brain, by contrast, computes

mostly using an analog computing approach, which is believed to be highly imprecise [18,19]. This kind of "approximate computation," where the basic operations are imprecise, but still manage a reliable result at the system level, has the potential of being much more energy efficient.

The precision of CPUs and GPUs is essential for many applications, but it has been recently discovered that in the case of deep neural network, we can actually reproduce the approximate computing strategy of the brain. Indeed, less precise and more energy efficient fixed point operation can be used instead of floating point. Precision as low as eight bits has no impact on neural network inference [20]. Google has already designed a (purely CMOS) system that exploits this property of deep neural networks [21]. It is possible to go even further in the direction of low precision. If a neural network has been *specifically trained* with the idea that it is going to be operated with low-precision arithmetics, we can use synapses that have lesser than eight bits: this approach corresponds the concept of quantized neural network introduced in Ref. [22]. Finally if we are willing to increase the number of neurons, we can even use binary synapses and still manage state-of-the-art performance on difficult image classification tasks [23–25].

These results are exciting for memristive implementations of deep neural networks: they mean that digital memristive neural networks can be implemented with few bits, such as reduced number of memristive devices. They would also use simple fixed point arithmetic, and therefore save a lot of energy. They also suggest that we may actually reproduce the strategy of the brain and rely on analog computation. As many memristive devices can be used as analog memory, this means that we might get away with using one memristive device to implement a synaptic weigh.! In the next two chapters, we will see extensive developments of this idea, and particularly efficient techniques to implement the equations of neural networks using analog memristive devices. We will also see how to take into account the reality of device behavior—and of their imperfections—in such approaches, and how different types of memristive devices may bring different types of opportunities and challenges.

12.2.3 Cost of clock

Finally a last source of energy consumption in CPUs and GPUs is absent in brains: clocking. In current large microelectronics systems, distributing the clock has become a major source of energy consumption, while brains operate in an entirely asynchronous fashion and therefore avoid this energy cost entirely. This strategy is however extremely difficult to imitate in the memristive implementation of a deep neural network, as the equations of a neural network need to be computed in order. On the other hand, the final part of this book details that research aiming at implementing more bioinspired

neural networks may use asynchronous electronics and also avoid the energy cost of clocking.

12.2.4 Is backpropagation hardware compatible?

We have seen that implementing inference on deep neural networks is a highly promising lead, as inference equations are topologically local, and inference in neural networks is compatible with low-precision computing. Learning is a more difficult problem, as the backpropagation equations are not topologically local, and not tolerant to low-precision computing. These issues become particularly problematic when training a system with many layers.

Realizing a memristive system capable of inference only would in fact already have tremendous applications. For many applications in situ learning is not necessary. Learning can be done on big GPUs by the service provider and the resulting weights are transferred on the low-energy memristive systems. This being said, learning on chip with memristive devices is not an impossible task, as the brains are able to learn probably using topologically local "equations" and low-precision arithmetics. Therefore currently considerable research is trying to adapt backpropagation to these challenges. Some of this research is described in the next two chapters and in part IV of the book.

We would also like to finish this discussion by a more long-term thought: is deep learning similar to brain learning? In fact there are considerable differences. Deep neural networks learn based on massive dataset, which can necessitate billions of examples. If a neural network is trained to recognize giraffes, it should be trained with images representing every way a giraffe can look (Fig. 12.6). By contrast a child can learn the concept of giraffe

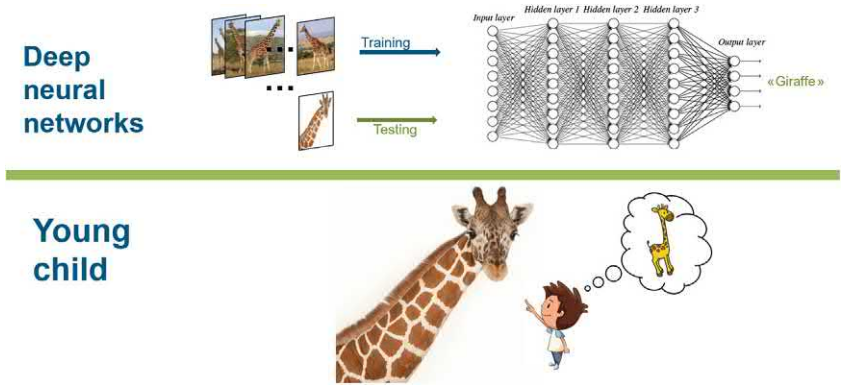


FIGURE 12.6 A deep neural networks need billions of examples to learn a concept, while a child can learn from a single example.

from a single cartoon and then recognize a real giraffe even if he or she has never seen one before. Human learning is therefore a lot more "intelligent" than deep learning, mainly due to our capability to generalize and to transfer concepts from old knowledge to new knowledge. For many researchers this suggests that the brain might be using learning principles that are quite different than standard backpropagation. This is why, for implementing learning with memristive devices, they take more inspiration in neuroscience than in deep learning. This approach is covered in part IV.

12.3 Conclusion

In this chapter we have summarized the basic principles of modern deep neural networks, and the sources of their important energy consumption when operated on CPUs and GPUs. We have seen that memristive devices offer a clear potential to implement lower energy deep neural networks, by providing a possibility to merge computation and memory, and to function in approximate computing regime. Memristive devices might also allow energy efficient learning, but this raises more challenges, as vanilla backpropagation does not allow the merging of computation and memory easily and is not as compatible with approximate computing. These considerations open up extremely interesting research, in terms of system design, application, and optimization of memristive devices for such applications. The next two chapters describe some of these exciting works with more technical details.

References

[1] I. Goodfellow, Y. Bengio, A. Courville, Y. Bengio, Deep Learning, vol. 1, MIT Press, Cambridge, 2016.
[2] Editorial, Big data needs a hardware revolution, Nature 554 (2018) 145.
[3] D. Silver, A. Huang, C.J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, et al., Mastering the game of go with deep neural networks and tree search, Nature 529 (2016) 484.
[4] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, et al., Mastering the game of go without human knowledge, Nature 550 (2017) 354.
[5] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by back-propagating errors, Nature 323 (1986) 533.
[6] G. Cybenko, Approximation by superpositions of a sigmoidal function, Math. Control Signals Syst. 2 (1989) 303–314.
[7] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, Neural Information Processing Systems Foundation, pp. 1097–1105.
[8] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (2015) 436–444.
[9] F.A. Gers, J. Schmidhuber, F. Cummins, Learning to forget: continual prediction with LSTM, Neural Comput. 12 (10) (2000) 2451–2471.
[10] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, arXiv preprint arXiv:1412.6980, 2014.

[11] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, et al., Tensorflow: a system for large-scale machine learning, in: OSDI, *16*, 2016, 265–283.

[12] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, et al., Automatic differentiation in PyTorch, In NIPS 2017 Autodiff Workshop: The Future of Gradient-based Machine Learning Software and Techniques, Long Beach, CA, 2017.

[13] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, et al., Caffe: Convolutional architecture for fast feature embedding, in: Proceedings of the 22nd ACM International Conference on Multimedia, ACM, 2014, pp. 675–678.

[14] G. Indiveri, S.-C. Liu, Memory and information processing in neuromorphic systems, Proc. IEEE 103 (2015) 1379–1397.

[15] D. Querlioz, O. Bichler, A.F. Vincent, C. Gamrat, Bioinspired programming of memory devices for implementing an inference engine, Proc. IEEE 103 (2015) 1398–1416.

[16] A. Sebastian, M. Le Gallo, G.W. Burr, S. Kim, M. BrightSky, E. Eleftheriou, Tutorial: Brain-inspired computing using phase-change memory devices, J. App. Phys. 124 (2018) 111101.

[17] A. Pedram, S. Richardson, M. Horowitz, S. Galal, S. Kvatinsky, Dark memory and accelerator-rich system optimization in the dark silicon era, IEEE Design Test 34 (2017) 39–50.

[18] A.A. Faisal, L.P. Selen, D.M. Wolpert, Noise in the nervous system, Nat. Rev. Neurosci. 9 (2008) 292.

[19] E. Marder, J.-M. Goaillard, Variability, compensation and homeostasis in neuron and network function, Nat. Rev. Neurosci. 7 (2006) 563.

[20] D. Lin, S. Talathi, S. Annapureddy, Fixed point quantization of deep convolutional networks, in: International Conference on Machine Learning, 2016, pp. 2849–2858.

[21] N.P. Jouppi, C. Young, N. Patil, D. Patterson, G. Agrawal, R. Bajwa, et al., In-datacenter performance analysis of a tensor processing unit, in: Computer Architecture (ISCA), 2017 ACM/IEEE 44th Annual International Symposium on, IEEE, pp. 1–12.

[22] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, Y. Bengio, Quantized neural networks: Training neural networks with low precision weights and activations, J. Machine Learn. Res. 18 (2017) 6869–6898.

[23] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, Y. Bengio, Binarized neural networks: training deep neural networks with weights and activations constrained to + 1 or-1, arXiv preprint arXiv:1602.02830, 2016.

[24] M. Rastegari, V. Ordonez, J. Redmon, A. Farhadi, Xnor-net: Imagenet classification using binary convolutional neural networks, in: European Conference on Computer Vision, Springer, 2016, pp. 525–542.

[25] M. Bocquet, T. Hirztlin, J.-O. Klein, E. Nowak, E. Vianello, J.-M. Portal, et al., In-memory and error-immune differential rram implementation of binarized deep neural networks, in: 2018 IEEE International Electron Devices Meeting (IEDM), IEEE, pp. 20–26.

Chapter 13

Analog acceleration of deep learning using phase-change memory

Pritish Narayanan, Stefano Ambrogio, Hsinyu Tsai, Charles Mackin, Robert M. Shelby and Geoffrey W. Burr
IBM Research– Almaden, 650 Harry Road, San Jose, CA, United States

13.1 Introduction

Deep learning networks and algorithms [1–4], trained using backpropagation [5], have revolutionized machine learning in recent years, matching or improving upon human-level performance in diverse fields such as image recognition [6], speech recognition [7], and language translation [8–10]. This revolution has been achieved by the confluence of two important factors—(1) the availability of copious amounts of data and (2) the use of high performance graphical processing units (GPUs) to train large networks with several millions of adjustable parameters (referred to henceforth as "weights," "synapses," or "synaptic weights").

However, there are some key challenges when it comes to building hardware for deep learning. First, even with cutting edge GPUs, training deep networks can often take days to weeks. Second, large pretrained networks need to be often deployed in either highly power- and energy-constrained settings ("edge" devices) or in enterprise/cloud settings, where high throughput and/or low latency may be all important. This has led to a significant interest, in both academia and industry, on hardware acceleration of deep learning.

Digital accelerators, such as those from Google tensor processing unit (TPU) [11], IBM [12], or NVIDIA deep learning GPUs [13], seek to make computations more efficient by applying a series of deep learning-focused optimizations. At the software level, this implies frameworks and libraries to make the best use of the underlying hardware. At the hardware level, optimizations target one or more facets of the well-established von Neumann architecture—in which a processing unit and a memory unit are distinct entities

Memristive Devices for Brain-Inspired Computing. DOI: https://doi.org/10.1016/B978-0-08-102782-0.00013-7

connected by a bus. For instance, one could improve memory capacity and bandwidth by using high-bandwidth memory (HBM) [14] and design systolic arrays [15] to efficiently implement multiply−accumulate operations within the processing unit, by using deep learning-aware compression schemes [16] to reduce the amount of data transferred through the bus (the von Neumann bottleneck), or by using reduced precision arithmetic to address all three facets at once [17,18]. In these accelerators, the underlying technological framework (Silicon MOSFETs, CMOS logic, off-chip or embedded DRAMs) is assumed to be unchanged.

On the other hand, more radical/exploratory analog approaches seek to achieve far greater performance and energy benefits by rethinking the entire design stack across devices, circuits, and architecture. Since the conductance of some emerging memory technologies or embedded flash [19] transistors can be tuned over a continuous range, these can be used to implement the weights of the neural network [20]. Furthermore, the multiply−accumulate operation, which accounts for a substantial fraction of deep learning workloads, can be implemented directly on crossbar arrays of such analog devices using the inherent physics of these structures, enabling a path toward acceleration. This is an example of a "non Von Neumann" architecture, where computation is performed at the location of the data.[1]

Among analog device candidates for deep learning, phase-change memory (PCM) [22,23] is particularly promising. As described earlier in this book, it is a relatively mature technology with good endurance, and a wide resistance contrast attributed to its tunability from fully amorphous to fully crystalline states. Conversely, phenomena such as resistance drift due to amorphous relaxation, nonlinear conductance response versus programming pulses, and sharp asymmetry between SET and RESET operations can be detrimental to deep learning [24,25].

A primary concern for research in this field is as follows: Can a deep neural network implemented with large arrays of PCM (or other NVM) devices achieve the same accuracy on a training or inference task as a GPU or a digital accelerator? On the one hand, the ready availability of several deep learning benchmarks gives us the foundations to start tackling this question. On the other hand, at the time of writing, end-to-end analog hardware systems with several millions of synapses are not yet available. Furthermore, this seems to be a classic case of chicken-and-egg, where the considerable monetary investment to build such a system can be justified only if such a radical approach can be shown to work in the first place.

While several research groups have focused on simulation studies of PCM and other NVM devices (for a review, please see Ref. [26]), the nonergodic and time-dependent change in device characteristics of real devices is

1. One exception to this analog/digital categorization is the IBM TrueNorth chip [21], which is a digital non−von Neumann system.

exceedingly difficult to model through simulation alone. To this end, we have proposed a mixed hardware–software experimental framework, wherein we use physical PCM devices for every single conductance used in the network but combine this with software simulations of neuron functions such as squashing functions and derivatives.[2]

The rest of the chapter focuses on the use of PCM technology for accelerating deep learning. We will start with a brief overview of deep learning in Section 13.2, along with an introduction to the major types of neural networks in use today including perceptrons [27], convolution neural network [1,6], and LSTMs [2,3]. We then present an overview of recent research in the field using PCM for deep learning and the associated challenges (Section 13.3) followed by our recent results on achieving software-equivalent neural network accuracy in training with PCM devices (Section 13.4) using our mixed hardware–software experiments. Building from the techniques introduced in Section 13.4, we further discuss NVM device requirements for deep learning through a series of simulation studies (Section 13.5) and conclude in Section 13.6.

13.2 Deep learning with nonvolatile memory—an overview

We briefly discuss the basic computational needs of deep learning. More information can be found from several online resources including the following tutorials, courses, and published works [28–34].

From a hardware designer's perspective, deep learning is attractive because (1) the number of different computational primitives needed tends not to be very large and (2)) a large fraction of the computational effort is concentrated on a small subset of these primitives. Among these, the multiply–accumulate (MAC) operation is one of the most important. In this operation, a vector of neuron activations x_i is multiplied by a matrix of weights w_{ij}. An "activation function" f is then applied to the accumulated sum, generating a new vector of neuron excitations for the next layer y_j. These nonlinear activation functions include tanh, sigmoid, or rectified linear units (ReLUs) and are needed to ensure that the neural network does not reduce to a simple linear transformation of the inputs.

MAC operations are the workhorse of both fully connected and convolution layers. These layers differ from one another in the total number of neurons and synapses involved and how they are organized, as shown in Fig. 13.1. Nevertheless, both types of layers feature prominently in widely used neural networks. Multilayer perceptrons (MLPs) [27] are a series of

2. Although we have used PCM extensively in our research, the mixed hardware–software experimental framework and other concepts that we describe in this chapter such as jump tables and multiple conductances of varying significance are applicable to other materials/devices too.

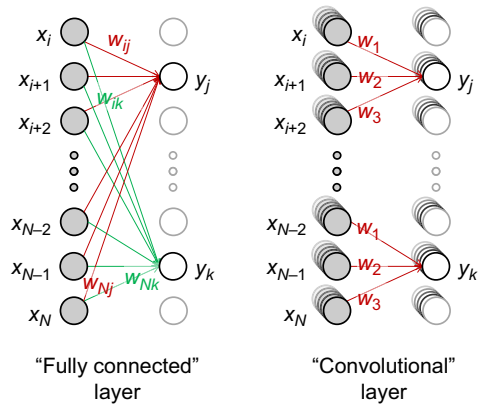


FIGURE 13.1 (A) In a fully connected (FC) deep neural network layer, each pair of neurons across the two neighboring neuron layers shares a unique weight (but no connections within layers). (B) In contrast, a convolutional (CONV) layer contains many neurons, often organized into planes. *Adapted with permission from Tsai H.Y., Ambrogio S., Narayanan P., Shelby R. and Burr G.W., J. Phys. D Appl. Phys. 51 (2018) [35].*

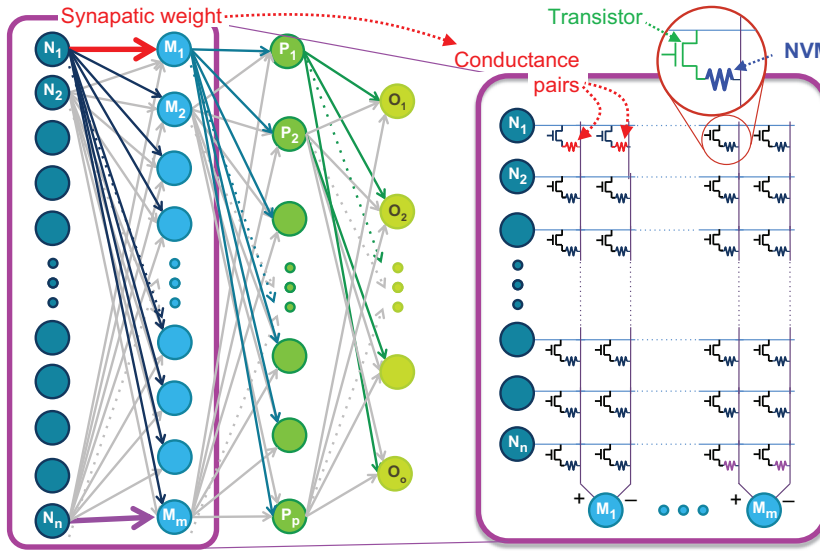


FIGURE 13.2 (Left) A multilayer perceptron (MLP) with four neuron layers and three fully connected weight layers. (Right) Mapping of the first fully connected layer to a NVM crossbar array. Every weight is encoded into a pair of conductances G^+ and G^-. Access transistors (or back-end-of-the-line selectors) may be needed for selective programming operations, just as in a conventional memory array. *Adapted with permission from Burr G.W., Shelby R.M., Sidler S., Di Nolfo C., Jang J., Boybat I., et al. IEEE Trans. Electron Dev. 62 (2015), 3498–3507 [36].*

fully connected layers, as shown in Fig. 13.2 (left). Convolution neural networks [1,6] contain several convolution layers with zero or more fully connected layers at the end. Long short-term memories (LSTMs) [2,3] combine fully connected layers with element-wise multiplication and time-evolving state variables for sequence prediction.

The training of a neural network is typically done in a supervised fashion. During "forward propagation," input training data presented to the network are propagated through multiple layers, generating a final set of outputs. For instance, in a typical image processing application, the input is an image and the output is a classification label (e.g., this is a cat). Errors are calculated by comparing the classification labels generated by the network against known correct answers for the training data. These errors are backpropagated through the network from the output to the input, adjusting the weights of the network along the way, often using a "gradient-descent" rule [4,5] that induces a change in the weight along the direction of steepest descent, as described in the previous chapter of this book. By presenting copious amounts of training data and applying corrections, the network improves at the particular task it is being trained on.

It must be noted that this simplified description hides several important subtleties. The chief among these is that the operations as described here will eventually (given sufficient data, time, and weight parameters) lead to the network actually "memorizing" the entire input dataset, also known as "overfitting." However, the practical use of deep neural networks (DNNs) is in being able to make decisions on data not seen before, but nevertheless similar enough to the training data. This is known as generalization, and achieving good generalization performance is one of the key challenges in the DNN design. Another important factor is the use of "mini-batches," that is, combining multiple input vectors into an input matrix and converting vector−matrix operations into matrix−matrix manipulations that can better utilize digital compute resources. This works particularly well in training—in most cases, the entire training set is available a priori, and achieving high throughput in terms of the effective number of examples processed per second is far more important than the latency of any single example.

Once the neural network is trained, it can be deployed to do "inference" (or forward inference) on new unseen data. The computations are identical to the forward propagate phase during training. Inference does not need the computations involved in back propagation and weight update. Furthermore, it typically requires less numeric precision provided the forward propagate step during training is also done at similarly reduced precision [37].

Another important distinction is that the hardware opportunity for inference strongly depends on the use case being considered. In the embedded domain, one might target low-power or low-energy implementation, whereas in the enterprise domain, this could mean throughput. In many cases, latency could be important too, for example, when populating, an entire large mini-batch cannot be taken for granted. Therefore it is conceivable that the hardware design approaches for accelerating inference would be considerably

different than for training, and indeed from one another, despite the fact that a significant fraction of the computations are identical.

Nevertheless, a key consideration for digital accelerator design is minimizing the amount and the distance of data to be moved. Convolution layers [1,6], which feature significant reuse of weight data (as the same sets of weights need to be rastered across all the pixels of an image) are less likely to be memory bottlenecked compared with fully connected layers, where each input/output neuron pair has a unique weight associated with it, and reuse is entirely dependent on minibatching. Consequently, digital hardware designs focusing on optimizing convolution layers could work well for applications such as image processing, which rely heavily [38] if not almost entirely on these. However, they are more likely to be bottlenecked when considering recurrent neural networks such as long short-term memory (LSTM) [2] or gated recurrent units (GRUs) [9], which are used in tasks such as machine translation and captioning, which need several fully connected layers with limited data reuse.

On the other hand, the analog memory-based accelerators are uniquely well suited for fully connected layers. The basis of any analog-based acceleration approach is a memory array that can store weights in an analog fashion (Fig. 13.2, right). Typically, this is done by encoding the weight into the conductance(s) of a resistive element (which is often an NVM element, but could also be a flash transistor [19]). Positive or negative weights can be encoded by either using a pair of conductances (where the weight is represented as $w_{ij} = G_{ij}^{+} - G_{ij}^{-}$) [20] or using a single conductance per weight alongside a common reference conductance that is shared across all the weights. More advanced schemes use more than one conductance pair, increasing the dynamic range of the weight being encoded and/or providing fault resilience. The conductance pairs could be of varying [39] or equal significance [40].

By encoding the neuron activation x_i as either a voltage or a pulse duration, a multiplication operation can be implemented at the site of each conductance using Ohm's law ($I = GV$). Furthermore, by applying the right neuron activations to all rows at once, the current contributions of all individual conductances along a column can be added up, achieving the MAC operation at the location of the data, using the inherent physics of the devices and the crossbar array. Successive layers of a neural network can be implemented on multiple crossbars, with mechanisms for converting the accumulated charge (typically stored on a "south-side" capacitor in one array), into a digital or analog signal that is passed through an activation function and delivered to the "west side"' inputs of the next array (Fig. 13.3A). Reverse propagation for training tasks can be implemented in a similar fashion, with inputs (specifically the corrections δ_j) arriving on the south-side columns and accumulating along the rows, producing outputs on the west side (Fig. 13.3B). Weight update can be done either using a crossbar-compatible weight update rule applying a series of pulses on the rows and columns

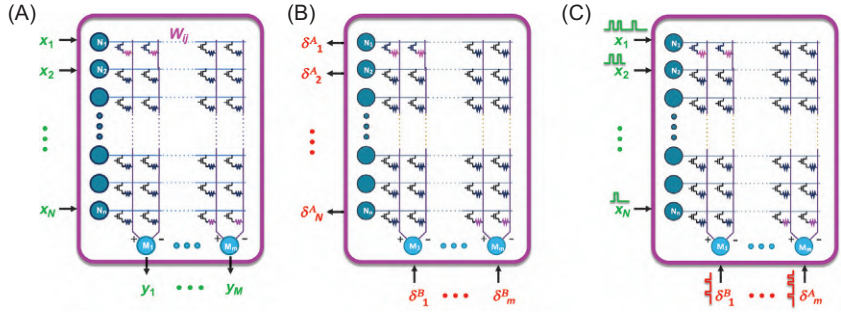


FIGURE 13.3 Signal flow for (A) forward inference: "read" $\mathbf{W}$ $y_j = f(\sum w_{ij} x_i)$, (B) reverse—backpropagate errors: "read" $\mathbf{W}^{\mathbf{T}}$ $\delta^A{}_i = f'(x_i^A) \bullet (\sum w_{ij}\, \delta^B{}_j)$, and (C) weight update operations on a crossbar array—"crossbar-compatible" weight update: "write" $\Delta\mathbf{W}$.

based on the x_i and δ_j values (Fig. 13.3C), with an overlap of pulses programming the crosspoints [24], a stochastic variant of this rule [25], or offline methods [41] more conducive to minibatching.

The above discussion outlines the basic needs for MLPs. LSTMs and GRUs would additionally need extra flexible signal routing capability as well as modules for vector−vector multiplication. Training hardware for these recurrent networks would also require some storage resources to hold on to values of state variables in previous time steps to accomplish backpropagation through time.

Convolution layers involve significant reuse of both the neuron activations and the weight kernels. For forward inference, this could be potentially implemented by making multiple copies of the weights on different crossbar arrays and additional orchestration circuitry to deliver the right neuron activations to all of these arrays. Circuitry for max pooling must also be included. However, it will be necessary to study the tradeoffs end to end to make careful estimations of whether forward inference convolution networks with analog memory are feasible and provide enough upside over digital approaches with all of these caveats included. For training, this is even more complicated—using a single copy of the weights and arranging the inputs to pass through sequentially [42] incurs high latency, especially for earlier layers where the input spatial dimensions are large. On the other hand, using multiple copies of the weights makes both the orchestration of the right x_i and δ_j values to the right spot at the right time quite complicated. Furthermore, reconciling weight updates across copies that ought to be identical is also decidedly nontrivial. Again, the time and energy tradeoffs of such an implementation must be carefully considered. Given these reasons, most of the rest of this chapter will not focus on convolution. A notable exception is given in Section 13.4, where we will discuss using a transfer learning approach to train on some image processing benchmarks.

13.3 Recent progress on phase-change memory for deep learning

In recent years, several non-von Neumann approaches involving resistive memories and, among them, phase-change devices, have been proposed for performing logic or computation at the location of data. In particular, PCM has been a commonly used technology due to its relative maturity and the availability of large arrays. Among these demonstrations, the first results in the field demonstrated the application of PCM for the spike-timing-dependent plasticity (STDP) algorithm [20,40,43–46]. In this implementation, the conductance of the PCM encodes the strength of the synaptic connection between two neurons called the preneuron and the postneuron. In case the preneuron fires a pulse before the postneuron, the synapse undergoes potentiation, namely the device conductance is increased. Conversely, if the postneuron fires before the preneuron, conductance is decreased. While hardware for STDP has been a popular field of research due to its presumed neuromorphicity (i.e., there is some evidence that this is one mechanism by which synaptic strength between neurons in the brain is modulated), it is still unclear how (or indeed if) this local learning rule can be adapted/scaled up to meaningful or commercially interesting learning tasks. Other in-memory computing applications, as the name suggests, consider performing arithmetic computation directly at the location of data, such as mentioned in Ref. [47]. Here, systems of linear equations of the type $Ax = b$ are iteratively solved using crossbar arrays of PCM devices [47]. The solution is achieved by guessing an initial approximate solution z and calculating the error $r = Az$. These operations, which are computationally intensive but at low precision, are performed in the PCM crossbar array. Then, high-precision calculations of the solution $x = x + z$ and of the error $r = b - Ax$ are performed, updating the result. Since these are not computationally intensive, they can be performed in a digital unit, speeding up the overall computation time [47].

In this section, we overview non-von Neumann coprocessors based on phase-change memory devices serving as synaptic weights. As mentioned earlier, PCM devices are typically organized in crossbar arrays to implement dense matrices of weights. An early demonstration using large arrays of real PCM devices has been reported in Ref. [24], where an MLP with four neuron layers has been trained on a subset of 5000 images of the MNIST dataset. Experimental training results are reported in Fig. 13.4, where a test accuracy of around 82% was achieved after two epochs of training. Results are consistent with a physics-matched simulator, using which it is possible to extrapolate the impact of training up to 20 epochs. In addition, the color map shows the conductance values of the PCM devices after training. In this demonstration, the weight is encoded as $W = G^+ - G^-$, where both G^+ and G^- are implemented using PCM

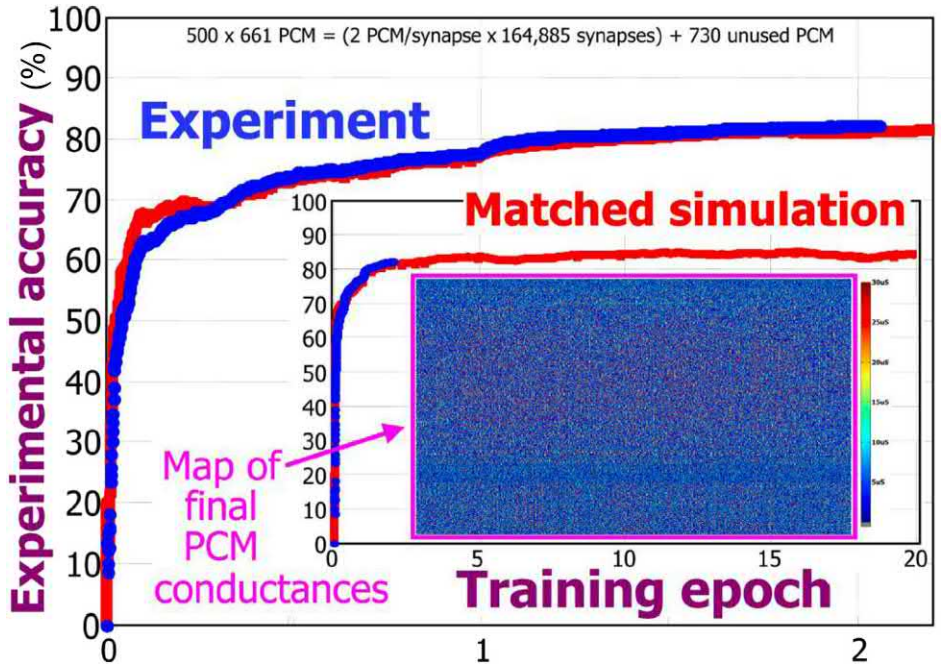


FIGURE 13.4 Experimental and matched simulated training results on a subset of the MNIST dataset, with weights encoded with two PCMs as $W = G^{+} - G^{-}$. The inset shows an extension of the simulation to 20 epochs, plus the conductance map of the trained weights after two epochs. *Adapted with permission from Burr G., Shelby R.M., di Nolfo C., Jang J., Shenoy R., Narayanan P., et al., IEDM Technical Digest T29.5 (2014) [24].*

devices. Training the same network with the same 5000 examples in software would produce an accuracy of around 94%, which is considerably larger than what was obtained in this mixed hardware–software experiment. This accuracy drop is caused by several issues due to PCM nonidealities such as limited conductance range, nonlinear and asymmetric update, and so on as described in detail in Refs. [24,36].

Some approaches have been developed to overcome these issues. For example, Boybat et al. [40] introduce the concept of a single weight composed of multiple conductances of equal significance. In this scheme, as shown in Fig. 13.5, a single weight is composed of N different PCM devices, Fig. 13.5A. The communication between the presynaptic and postsynaptic neurons happens through the application of the same voltage over all N devices, each of them contributing with an individual current. The sum of all currents, which is proportional to the sum of all single conductances, is then read by the postsynaptic neuron. This scheme enables an increased conductance range since now the maximum achievable conductance is roughly equal to N times the conductance of a single PCM device.

During weight update, one single PCM device is selected and programmed, as shown in Fig. 13.5B. Since weight update is performed only on one single device, the smallest conductance step is still equal to the change in conductance obtained with weights encoded with one single PCM. However, to avoid always programming the same device over the N available, a counter-based arbitration scheme is adopted, which is shown in Fig. 13.5C. The selection counter (which is shown in figure for a case with

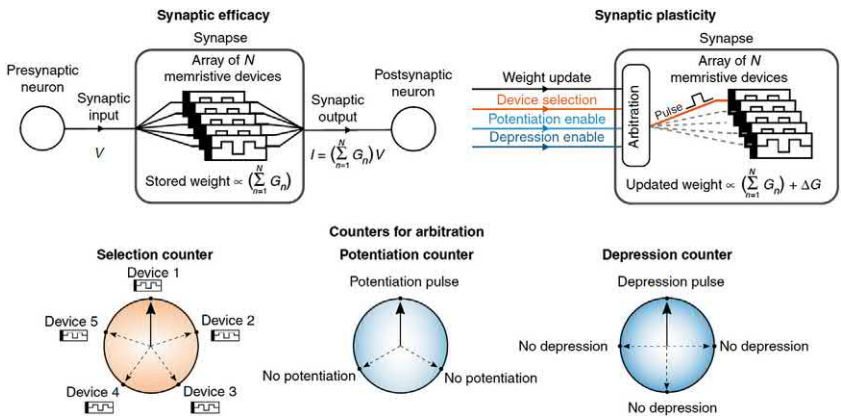


FIGURE 13.5 (A) Weight implementation with N conductances of equal significance. (B) The selection of the device to be updated and the operation to be performed (potentiation or depression) is obtained using (C) arbitration counters that selectively choose the device and the rates of potentiation and depression. *Adapted with permission from Boybat I., Gallo M.L., R.N.S., Moraitis T., Parnell T., Tuma T., Rajendran et al., Nature Communications 9 2514 (2018) [40].*

$N = 5$) is updated with a fixed increment rate, which is chosen to be coprime with the number N of devices in the weight. This enables proper distribution of weight updates over all the N available devices. To control the rate of potentiation or depression, two other counters are used, where only one position corresponds to the actual weight update. In this way, programming rates can be lowered based on a reduced probability of hitting the programming pulse. In addition, since devices often show different rates for potentiation or depression, two independent counters can be used, which decouple the two operations and compensate for eventual device asymmetries. This weight scheme can be implemented in a nondifferential approach, which is the one just described, or in a differential approach, where the $W = G^+ - G^-$ is obtained by dividing the N devices in two $N/2$ groups, one representing G^+ and the other G^-. In this scheme, weight potentiation is obtained increasing the conductance of the $N/2$ devices describing G^+, while depression is achieved by increasing the conductance of the $N/2$ devices for G^-.

The most interesting feature of this approach is the crossbar compatibility, since the N devices can be organized along a bit line, with the corresponding current contributions naturally summing in the bit line. The arbitration schemes are then implemented by turning on selected word lines.

Fig. 13.6 shows the simulated results of an MLP training on the MNIST dataset as a function of the number N of PCM devices in a single weight. Results show that increasing N leads to higher results, around 90%, for both differential and nondifferential schemes. However, accuracy is still below the full floating-point implementation (97.8% accuracy with 60,000 training examples), which is represented by the dashed line.

Another approach to deep learning is represented by the work of Nandakumar et al. [41]. The idea is schematically shown in Fig. 13.7. The premise of this approach is that during training of a neural network, both

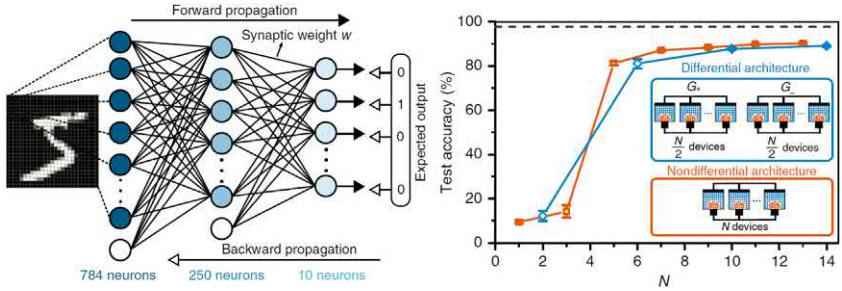


FIGURE 13.6 MLP trained on the MNIST dataset and corresponding simulated accuracy results for increasing number N of devices inside one single weight. Accuracy increases for increasing N, reaching values slightly lower than the full floating-point implementation, represented with a dashed line. Both results for differential and nondifferential architectures are reported. *Adapted with permission from Boybat I., Gallo M.L., R.N.S., Moraitis T., Parnell T., Tuma T., Rajendran et al., Nature Communications 9 2514 (2018) [40].*

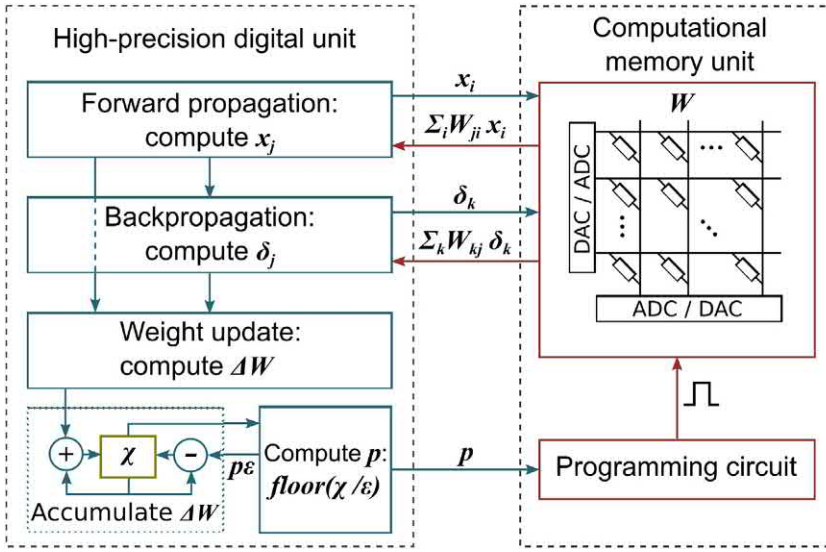


FIGURE 13.7 Hybrid implementation of the training of MLP using high-precision digital units to calculate x and δ and computational memory unit for calculating ΣW_x and ΣW_δ. Weight update is calculated with high precision on the χ variable. Only when χ is multiple times ϵ, representing the smallest achievable conductance step in a real PCM, a real programming operation is performed in the array. *Adapted with permission from Moraitis, R. N. S., Gallo M.L., Boybat I., Rajendran B., Sebastian A. and Eleftheriou E., IEEE ISCAS Proc. (2018), 1–5 [41].*

forward and backpropagation steps can be performed at reduced precision, provided that the weight update is computed at high precision. Based on this idea, the training of an MLP is performed using a hybrid approach. In this case, the multiply–accumulate operations corresponding to forward and reverse directions are implemented on the crossbar. The calculation for the subsequent x and δ values is done in high-precision digital circuitry, with ADCs/DACs interfacing to the crossbar arrays.

The weight update is also computed and accumulated in the digital domain using a floating-point variable χ. Only when χ is at least p times ϵ, where ϵ corresponds to the smallest conductance step implementable in the physical PCM cells, an integer number p of pulses is sent to the corresponding PCM device, and χ is updated in the digital domain subtracting $p\epsilon$. In case p is positive, the weight is potentiated, otherwise it is depressed. No check is performed on the accuracy of the programmed conductance in the PCM. This method avoids performing weight updates, which are too small to be implemented in a PCM. While the approach was shown to work for MNIST in simulation, it has not been tried in hardware. It will also need separate hardware resources for calculating the weight updates at every example, storing all the weight updates χ, and determining when to execute write operations to the array.

13.4 Achieving software-equivalent accuracy in DNN training

One approach to quantify the impact of NVM nonideality on neural network behavior is to use a simulation framework, where the NVM is modeled using a closed form analytical expression of the conductance change over a number of applied pulses. However, NVM device research is constantly evolving, and there are considerable differences in the NVM properties based on the materials, processes, and interfaces used or even in the electrical parameters such as pulse width, voltage, number of pulses, etc. Another even more important concern is that often times, analytical models may not be able to capture the full extent of variability across hundreds of thousands to millions of devices or even cycle to cycle variation within a single device.

In earlier work, we have proposed the idea of a "jump table" [24,36] as a generic statistical construct for modeling NVM devices (e.g., a PCM jump table is shown in Fig. 13.8A). In a jump table, the *X*-axis represents the instantaneous conductance expressed either in absolute or relative (percentage of maximum conductance) terms, and the *Y*-axis represents the conductance change (or "jump") at a particular conductance value. Instead of a single point on the *XY* plain, we use a cumulative distribution function that captures the full extent of the statistical distribution of this conductance jump for every interval in the *X*-axis. In simulation studies of large neural

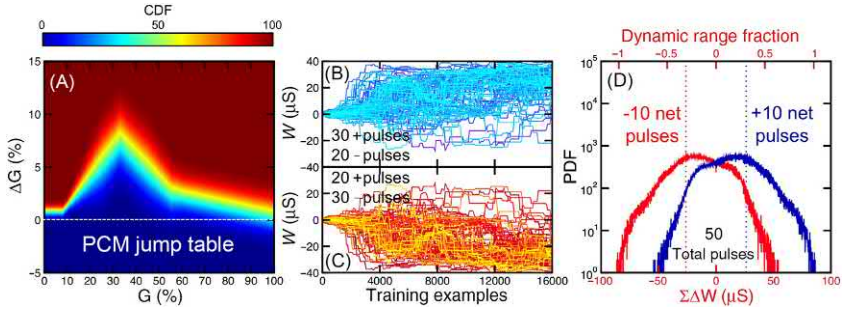


FIGURE 13.8 (A) A "jump table" with cumulative density function (CDFs) of conductance change versus absolute conductance, based on the experimental characterization of ~330,000 devices from Ref. [24]. (B) Possible trajectories of weight evolution, when starting from an initial net weight of zero, and applying either 30 up/20 down (top) or 20 up/30 down update pulses, simulated using the jump table data and randomizing the ordering of the up and down pulses. (C) Probability density function (PDF) of final weights obtained from 3×10^5 random trajectories of net +10 or net −10 weight changes. Ideal behavior (linear and symmetric conductance change) would result in delta functions at the dotted vertical lines. Nonideality in device characteristics leads to the considerable spread in final conductance values. *Adapted with permission from Ambrogio S., Narayanan P., Tsai H., Shelby R.M., Boybat I., di Nolfo C., et al., Nature 558 (2018) 60 and Cristiano G., Giordano M., Ambrogio S., Romero L., Cheng C., Narayanan P., et al., J. Appl. Phys. 124 (2018) 151901 [39,48].*

networks, every NVM participating in a weight update step undergoes a weight change that is independently sampled from one of these distributions depending on its instantaneous conductance.

While the jump table can be a more comprehensive approach to capture variability with respect to closed form analytical expressions, there are certain limitations: (1) simulations tend to be inherently slower, as every conductance change needs to be sampled, (2i) large device arrays may be needed to ensure that one has enough characterization data to build a meaningful jump table, and (3) there is an implicit assumption of ergodicity (i.e., the weight update pulses applied across many different devices are equally representative of all the devices in the array). In reality, different devices could have different tendencies based on local variations.

Nevertheless, the jump table construct can be tremendously useful in understanding the impact of NVM nonideality on neural networks. For instance, during training, every weight in the neural network might undergo a large number of weight increases and an almost equal number of weight decreases. The expectation in training is that weight updates are symmetric, that is, one positive weight update exactly cancels a negative weight update, and the effective weight changes after any number of examples is proportional to the difference. For instance, one might consider a net change of $+10$ (-10) after 30 positive (negative) and 20 negative (positive) pulses. However, we know that this might not really work in analog hardware given the nonlinearity, asymmetry, and variability of the PCM devices. We can study this issue by setting up a random order of positive and negative weight updates (total 50, with a net difference of 10) and tracking the evolution of the weights by sampling conductance changes from the jump table distribution. Typical trajectories of this behavior are shown in Fig. 13.8B.

At the end of 50 total pulses, we can plot distributions of the effective change in weights starting from different initial conditions, as shown in Fig. 13.8C. In an ideal case, this distribution would be a delta function at $+10$ and -10. However, due to nonidealities discussed earlier, the actual distributions are quite broad with considerable overlap, which ultimately leads to low accuracy as discussed in Ref. [39].

The ideal NVM for training is one which is perfectly linear in conductance change for both increasing and decreasing pulses [25]. While several research efforts are being undertaken to build such an ideal device [49], the rest of this section focuses on how we can improve accuracy with existing technology using device and circuit techniques.

13.4.1 PCM + 3T1C

We discuss the concept of "multiple conductances of varying significance," wherein the net synaptic weight is distributed across not one, but

two (or more) conductance pairs ([39] and also from Ref. [50]). This can be mathematically expressed as $W = F\ (G^+ - G^-) + g^+ - g^-$, where G^+ and G^- are the higher/most significance conductance pair (MSP) whose contribution to the weight is scaled by a factor $F > 1$, and g^+ and g^- are the lower significance conductance pair (LSP). In this cell, all parallel weight updates during a training operation are applied to the LSP. Given the need to process training examples quickly, this is an "open-loop" programming, meaning one does not have the time to verify that the right change in weight was achieved at each cell. Periodically, one may execute an occasional transfer step, transferring the weight information from the LSP to the MSP. Since this is done infrequently and at a time when another array in the network is implementing neural network operations, it can be closed loop with multiple successive write/read operations as necessary to verify that the correct conductance change is applied to the MSP.

This is diagrammatically shown in Fig. 13.9. An initial net weight W, represented by the height above or below a centerline, initially has contributions from all four conductances. In these "G-diamonds," projections to the tilted axes represent the respective components. A weight transfer is visualized as zeroing out the lower significances and tuning the higher significances such that the net weight remains the same. However, it is still possible that weight was not successfully transferred in its entirety to the higher conductance pair. In this case, we have the option of sacrificing some of the dynamic range of the lower significance pair to compensate for

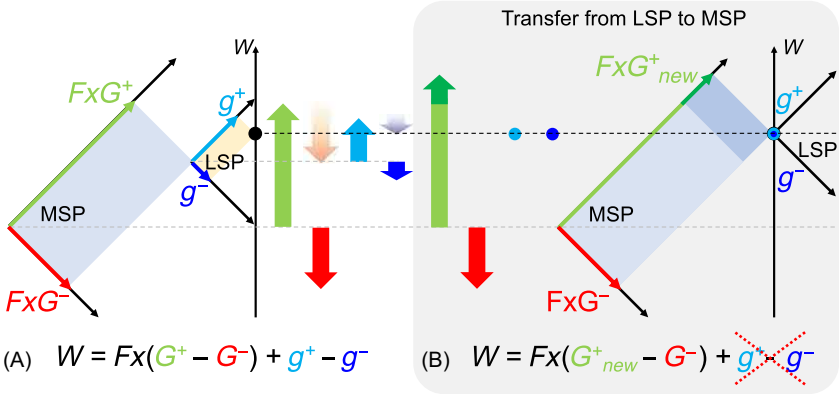


FIGURE 13.9 (A) A G-diamond, showing the net weight as a height above or below a horizontal axis, with higher significance (G + and G- arrows) and lower significance (g + and g- arrows) representing the relative contributions of the individual conductances. (B) Transfer operation, where weight information accumulated during training on the least significant pair is transferred over to the most significant pair. *Adapted with permission from Cristiano G., Giordano M., Ambrogio S., Romero L., Cheng C., Narayanan P., et al., J. Appl. Phys. 124 (2018) 151901 [48].*

the residual error. This approach is called posttransfer tuning (not shown in Fig. 13.9).

The multiple conductances idea immediately improves the total dynamic range of the synaptic weights. However, if both pairs are implemented with PCM, we would still have the issue of nonlinear and asymmetric open-loop conductance change during training. To address this question, we proposed using a capacitive unit cell to implement the LSP. In such a 3T1C cell (Fig. 13.10A [39]), weight information is stored on a capacitive element. The current through a "read" transistor (Tr) is proportional to the voltage on the capacitor (Fig. 13.10D). This voltage can be changed in an analog fashion by applying pulses to the pFET (Tp) or nFET (Tn) devices during open-loop training, sourcing, or sinking small amounts of current from the capacitor node. By operating these transistors in saturation, the programming current can be kept nearly independent of the capacitor (drain) voltage, implying linearity, that is, the conductance/voltage change can be almost constant over a wide range of instantaneous conductance/voltage values (Fig. 13.10B and C). The programming pulse amplitude applied to Tp and Tn can be adjusted to keep increases and decreases balanced, implying symmetric weight updates, at least in a nominal scenario.

However, given that the capacitor undergoes decay over time, the occasional transfer step needs to happen often enough to transfer training information to a nonvolatile higher significance conductance pair, which in this case can be PCM. Since this occasional transfer step is closed loop, one is not as sensitive to nonlinearity and asymmetry in PCM programming, as opposed to the example-by-example weight updates that have to be done in an open-loop fashion for performance reasons.

Finally, since the 3T1C cell conductance can be either increased or decreased linearly, we can save area by using a shared g cell as a common reference and adjusting the conductance of the individual g cells up or down in relation to this shared g (Fig. 13.10A). In our experiments, we assume a sharing of 3 g^{shared} cells for every 128 g cells.

13.4.2 Polarity inversion

While in principle the 3T1C cell can be made linear and symmetric, in practice, CMOS parameter variability implies that not every pFET and nFET used for increasing or decreasing the conductance is perfectly balanced. In fact, since the transistors are operated in near-threshold to ensure small programming currents through Tp and Tn and consequently small conductance changes on the capacitors, random variations in the FETs induced by effects such as dopant fluctuation can have a big impact.

By carrying out 1000 Monte Carlo simulations of the 3T1C cells using publicly available SPICE models [51,52] of a 90 nm CMOS technology, we obtained families of curves of current versus conductance for the three FETs

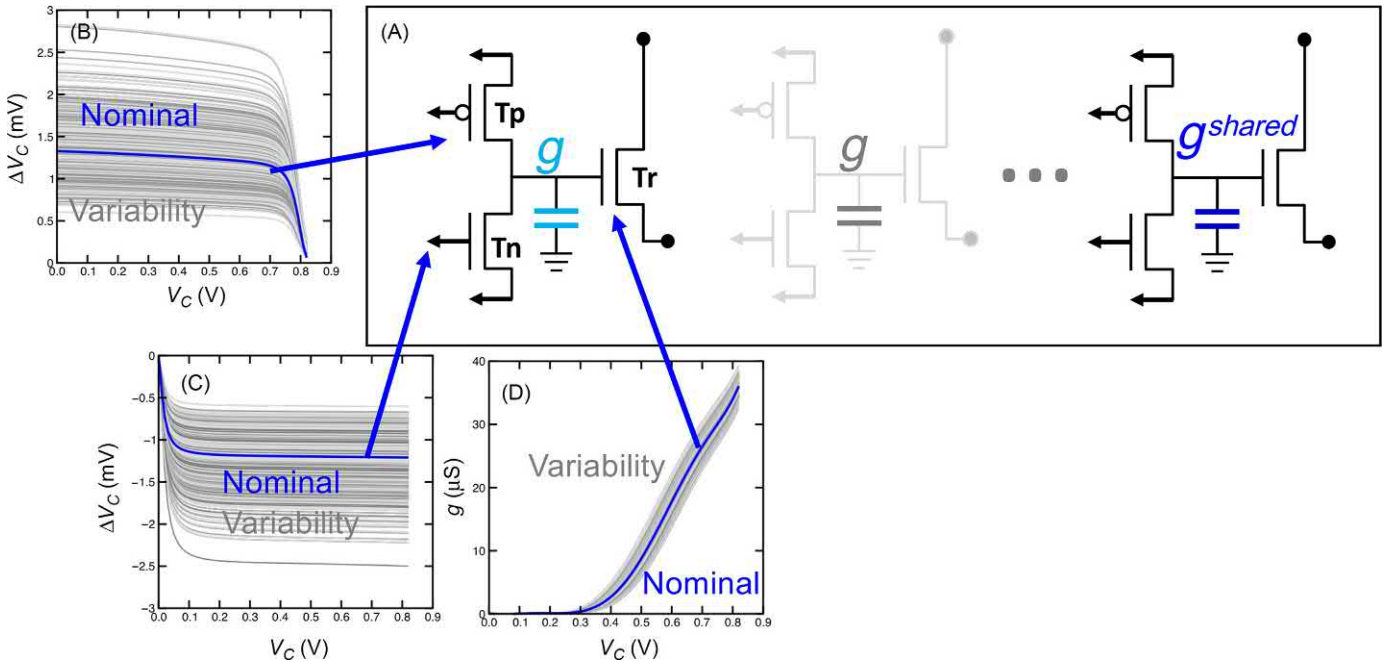


FIGURE 13.10 (A) A 3T1C unit cell: voltage on the capacitor represents a synaptic weight, which can be sensed as a current through a read transistor (Tr). P-type MOSFET (metal oxide semiconductor field-effect transistor) (PMOS) and N-type MOSFET (NMOS) field-effect transistor (FETs) (Tp and Tn) are used to increase or decrease the voltage on the capacitor. (B) Change in voltage as a function of the instantaneous capacitor voltage, when charge is added using the PMOS FET Tp. (C) Change in voltage as a function of the instantaneous capacitor voltage, when charge is removed using the NMOS FET Tn. (D) Effective conductance as a function of capacitor voltage, as sensed through Tr. In all cases, the nominal circuit simulation characteristic, and variability spread from Monte Carlo simulations are also shown. *Adapted with permission from Ambrogio S., Narayanan P., Tsai H., Shelby R.M., Boybat I., di Nolfo C., et al., Nature 558 (2018) 60[39].*

in the 3T1C cell (*gray* curves in Fig. 13.10B–D). To study the impact of this variability, we evaluate the trajectories of weight evolution. In this case, 1000 update pulses are applied, requesting a net change of +100 or −100, that is, 550 positive, 450 negative, or 450 positive, 550 negative pulses permuted in a random order. Once again, at the end of this weight evolution, we investigate the distributions of the final weights, and notice that there is a considerable spread.

The issue of asymmetry stems from having imbalance between the p- and n-FET devices. Having a too strong p-FET can lead to large weight increases, but small weight decreases, deviating considerably from the idealized weight cancellation concept, as shown in Fig. 13.11A and B. To fix this, a polarity inversion technique (Fig. 13.12) was introduced. In polarity inversion, the sign of the LSP is toggled periodically, that is, a weight initially represented by $W = F\ (G^+ - G^-) + (g^+ - g^-)$ is changed to $W = F\ (G^+ - G^-) - (g^+ - g^-)$. When this happens, the roles of the p- and n-programming FETs are reversed. Now the pFET, while still increasing the conductance of g^+ is actually decreasing the overall weight, and the nFET, while still decreasing the conductance of g^-, are actually increasing the overall weight. A strong pFET implies that before polarity inversion, the weight tended to drift toward higher values, but after the inversion, the same physical asymmetry (pFET stronger than nFET) causes the *weight* to drift toward lower values. This technique requires a re-initialization of the lower significance to allow the inversion and can very naturally be implemented at each transfer operation.

Fig. 13.11C shows the weight evolution trajectories with polarity inversion. Now, one can see that the expected weight change after two full transfer cycles is distributed quite tightly around the ideal values. With this final piece in place, we will show that we are able to achieve the goal of software-equivalent training accuracy with physical PCM devices.

13.4.3 Mixed hardware–Software experiment

We incorporated the above techniques into our mixed hardware–software experimental methodology (Fig. 13.13). In this methodology, the NVM cells are implemented using memory arrays of PCM. More specifically, every weight in the network is assigned to two physical PCM devices in the memory array. We perform initialization, read, and program operations on these devices allowing for the full extent of variability, defects, nonergodicity of conductance changes and other nonidealities of nonvolatile memory. The CMOS components, namely, the neuron circuitry and the 3T1C cells, are implemented in the software framework. To account for the variability in the 3T1C cells, we independently assign to each cell one weight increase, one weight decrease, and one read current characteristic

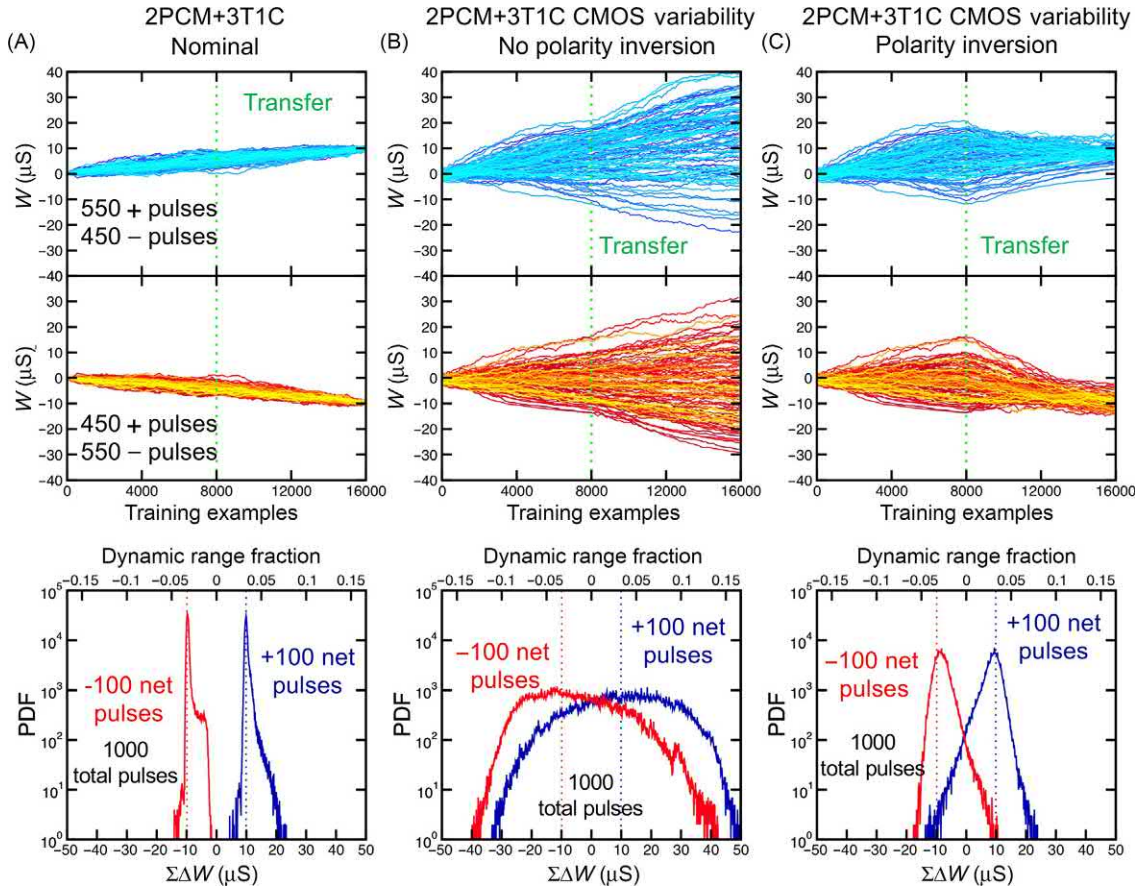


FIGURE 13.11 Trajectories of weight evolution and associated probability density functions of overall weight change for open-loop programming in 3T1C cells. (A) Nominal case, no variability, (B) fixed asymmetry between Tp and Tr weight changes due to variability, and (C) including variability and polarity inversion. *Adapted with permission from Ambrogio S., Narayanan P., Tsai H., Shelby R.M., Boybat I., di Nolfo C., et al., Nature 558 (2018) 60 [39].*

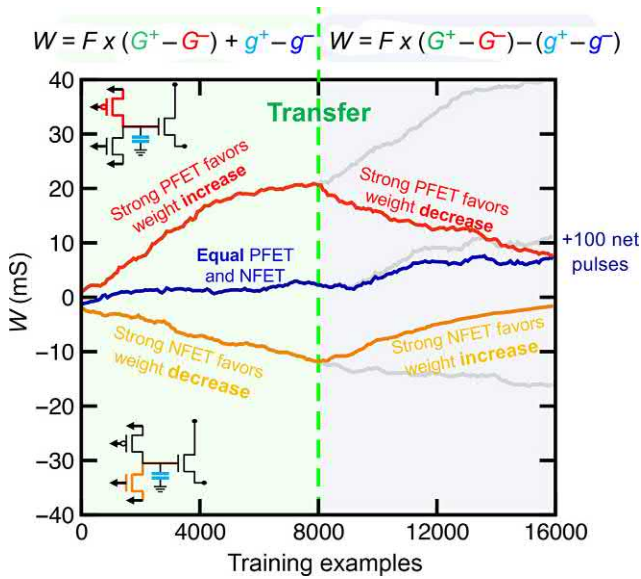


FIGURE 13.12 After every transfer operation, the polarity of the least significance pair is inverted. Fixed asymmetries that tended to cause the net weight to increase or decrease in the previous interval will now have the opposite effect. For instance, strong PFETs (N-type field-effect transistors, NFETs) that pushed weights to higher (lower) values in the earlier interval will now cause them to decrease (increase).

from the 1000 Monte Carlo SPICE simulations shown in Fig. 13.10B–D (10^9 combinations of characteristics possible).

With the weights of the network defined, for every training image, we calculate a set of neuron activations for each layer in software executing the forward pass and a set of errors executing the backward pass. We then apply weight updates to the 3T1C cells, using the families of SPICE simulations as described earlier. We also account for the decay of the capacitance charge, based on an RC time constant extracted from SPICE simulations.

We carry out the transfer operation once every 8000 examples to change PCM conductances, along with the needed polarity inversion and posttransfer tuning. In our experiments, the transfer step in this framework is done for all the devices at once, with the training stopped. In a complete implementation, the latency of the transfer step could be hidden, that is, arrays not involved in any deep learning operations at any given time might be executing transfer operations. Furthermore, the transfer step could be executed one row/column at a time due to circuit considerations, as opposed to the entire array at once.

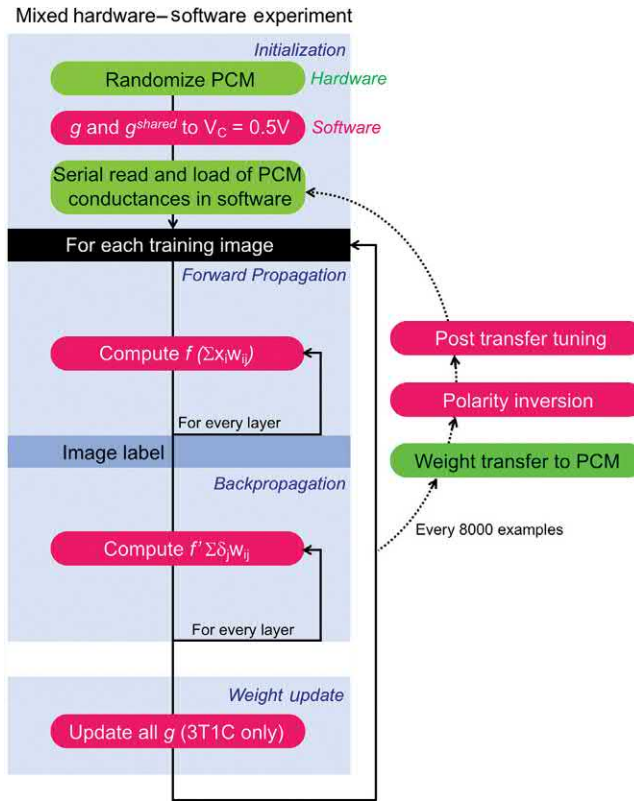


FIGURE 13.13 Mixed hardware–software experiment: PCM read and programming operations were carried out on real memory arrays, the 3T1C cell behavior was sampled from Monte Carlo SPICE simulations incorporating variability, and neuron functionality was emulated with software. *Adapted with permission from Ambrogio S., Narayanan P., Tsai H., Shelby R.M., Boybat I., di Nolfo C., et al., Nature 558 (2018) 60 [39].*

13.4.4 Results

We studied four different neural network benchmarks—MNIST [1], MNIST with background noise [53] CIFAR-10 and CIFAR-100 [54]. In each case, we established software baselines using TensorFlow. Our goal with these baselines was not to reach the best possible accuracy on a particular task, but to evaluate what an "ideal" software-implemented network of the same size and topology would achieve on the given benchmark, if it were allowed to take advantage of software techniques such as unbound rectified linear units (ReLUs), cross-entropy training, optimizers such as AdaGrad or ADAM, etc [34].

MNIST is the "hello world" of image classification tasks, where the goal is to recognize handwritten digits. The training data are composed of 60,000 28×28 handwritten images, and the trained model is tested on 10,000 images. We cropped MNIST to 22×24 since the outer pixels are largely empty. We then trained on a network with one input neuron layer (528 neurons), two hidden neuron layers (250, 125 neurons each), and an output neuron layer (10 neurons), with three fully connected weight layers in between (shorthand 528-250-125-10). The total number of synapses after accounting for one bias neuron per layer is 164,885, corresponding to 329,770 PCM devices. In Fig. 13.14A, we show training and test accuracies obtained from our mixed hardware–software experiment over 20 epochs and compare these against TensorFlow experiments with different random seeds. Experimental accuracy closely matches the average TensorFlow accuracy of 97.94%.

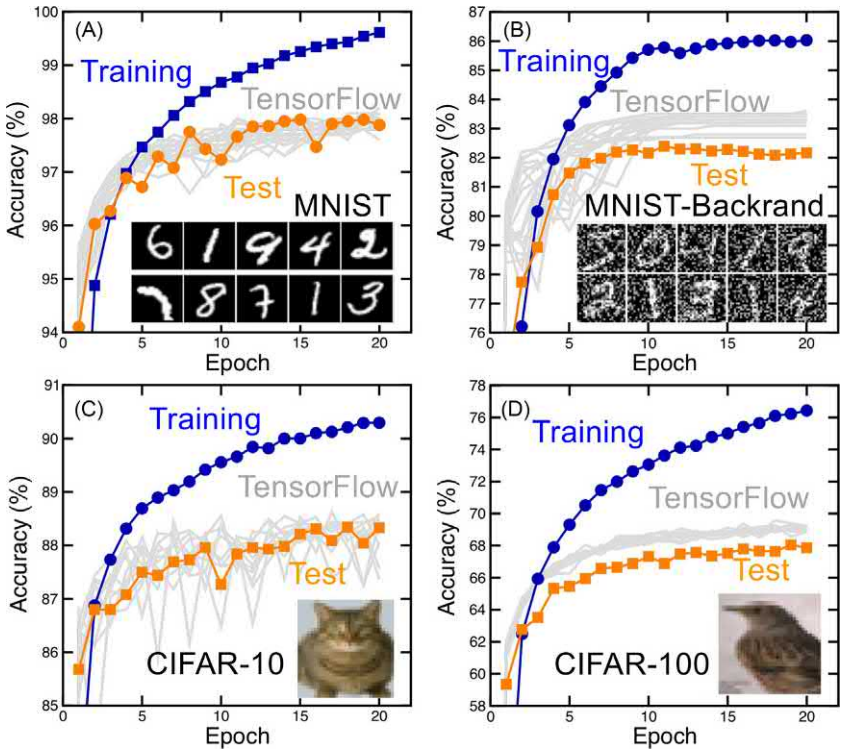


FIGURE 13.14 Results from mixed hardware–software experiments comparing test accuracy versus a TensorFlow baseline for (A) MNIST, (B) MNIST with background noise, (C) transfer learning with CIFAR-10, and (D) transfer learning with CIFAR-100. *Adapted with permission from Ambrogio S., Narayanan P., Tsai H., Shelby R.M., Boybat I., di Nolfo C., et al., Nature 558 (2018) 60 [39].*

A variant of MNIST, MNIST with background noise is a much more challenging dataset. Not only is there uniform background noise inserted into each image, the number of training examples is reduced to 12,000 and the number of test examples is increased to 50,000. We trained on a 784-180-100-10 network. We did not crop the images as the outer pixels contain noise that is relevant to the training task. The experimental test accuracy of 82.13% was only slightly below the TensorFlow accuracy of 83.3% (Fig. 13.14B).

The previous experiments focused on training networks composed entirely of fully connected layers. Today's convolution neural networks, considered the state-of-the-art for image classification, typically use convolution layers that act as feature extractors, alongside fully connected classification layers that combine the extracted features [1,6,55]. There are several challenges to implementing convolution layers with crossbar arrays including the large neuron to weight ratio, the need for complicated data flow schemes to allow neuron activation reuse, and the fact that convolution layers may not be memory bound in the first place (and hence well suited to digital accelerators such as GPUs). Therefore, training on convolution networks while maintaining the performance benefits of analog crossbar arrays is still a field of open research.

However, a transfer learning approach [56] can be used to reuse pretrained convolution layer weights for a new dataset, so long as the relevant features needing extraction in the new dataset are similar enough to the features learned from the original training dataset. In this case, it is sufficient to retrain just the fully connected classification layers at the end of the network since now these extracted features need to be combined in new ways to reflect the categories present in the new dataset.

We explored the option of using transfer learning for CIFAR-10 and CIFAR-100 datasets, Fig. 13.14C and D. We obtained original convolution weights from a 70-layer network (Google Inception-v3), which had been trained on ImageNET. To ensure that the 32×32 CIFAR images could be used with this network, which was originally trained on 299×299 ImageNET images, we used an image rescaling script (https://www.tensorflow.org/tutorials/imageretraining) provided by Google. We trained the last fully connected layer in each of these networks (2048-10 for CIFAR-10 or 2048-100 for CIFAR-100) using our analog memory approach and compared against TensorFlow-based training. While on CIFAR-10, test accuracy is equivalent and there is a small dropoff for CIFAR-100 (slightly more than 1%).

To quantify the relative impacts of the various techniques introduced in this section, we carried out simulation studies where we matched our neural network training simulation to the real experiment and then removed one technique at a time. Variability and mismatch in the FET characteristics have the most significant impact, and the polarity inversion technique is

critical to counteract them. Other important factors are the posttransfer tuning of the lower significance conductance to correct for residual errors in transfer and modulation of the neuron activation (x) and error (δ) pulses during weight update. Please see Ref. [39] for a more detailed analysis of the relative impacts.

13.5 Nonvolatile memory device requirements for deep learning revisited

The drawback of the 3T1C structure includes large area and power/energy consumption. One way to solve these issues would be to use two resistive devices instead of the 3T1C, as depicted in Fig. 13.15. Interestingly, the MSP and LSP devices will have different requirements, since their roles are different. In this way, constraints are more relaxed and diversified, as opposed to achieving all the necessary criteria from a single device to perform highly accurate training. In this section, a list of specifications is reported, based on jump table simulations [48].

To model different device behaviors, two similar jump tables have been used, as shown in Fig. 13.16A and B, where axes show percentage values. The large initial step (LIS) model shows large initial conductance step ΔG_0, as in RRAMs, while the S-shape model reveals a more PCM-like behavior. For the sake of simplicity, the following simulations will use the LIS model. However, results do not change markedly between LIS and S-shape models. Both show conductance saturation at large G, as demonstrated by ΔG going to be zero. The single device step shows an intrinsic noise σ_{intra}, which models the intradevice (or cycle-to-cycle) variability. To model the interdevice

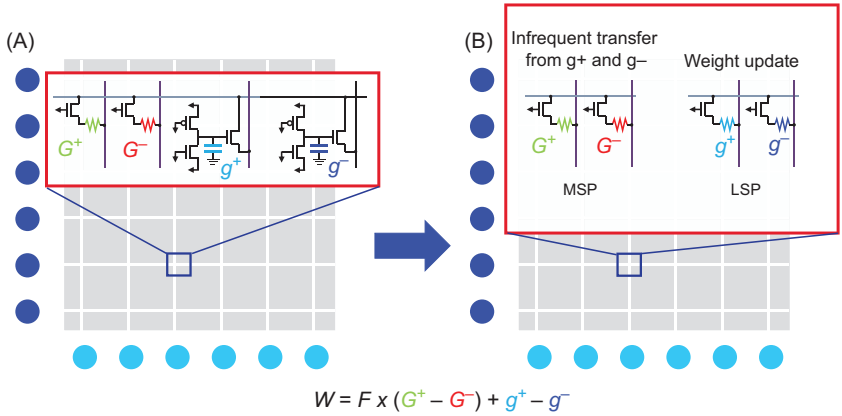


FIGURE 13.15 (A) Weight architecture proposed in Ref. [39] and (B) proposed updated version with four resistive devices [48]. *Adapted with permission from Cristiano G., Giordano M., Ambrogio S., Romero L., Cheng C., Narayanan P., et al., J. Appl. Phys. 124 (2018) 151901 [48].*

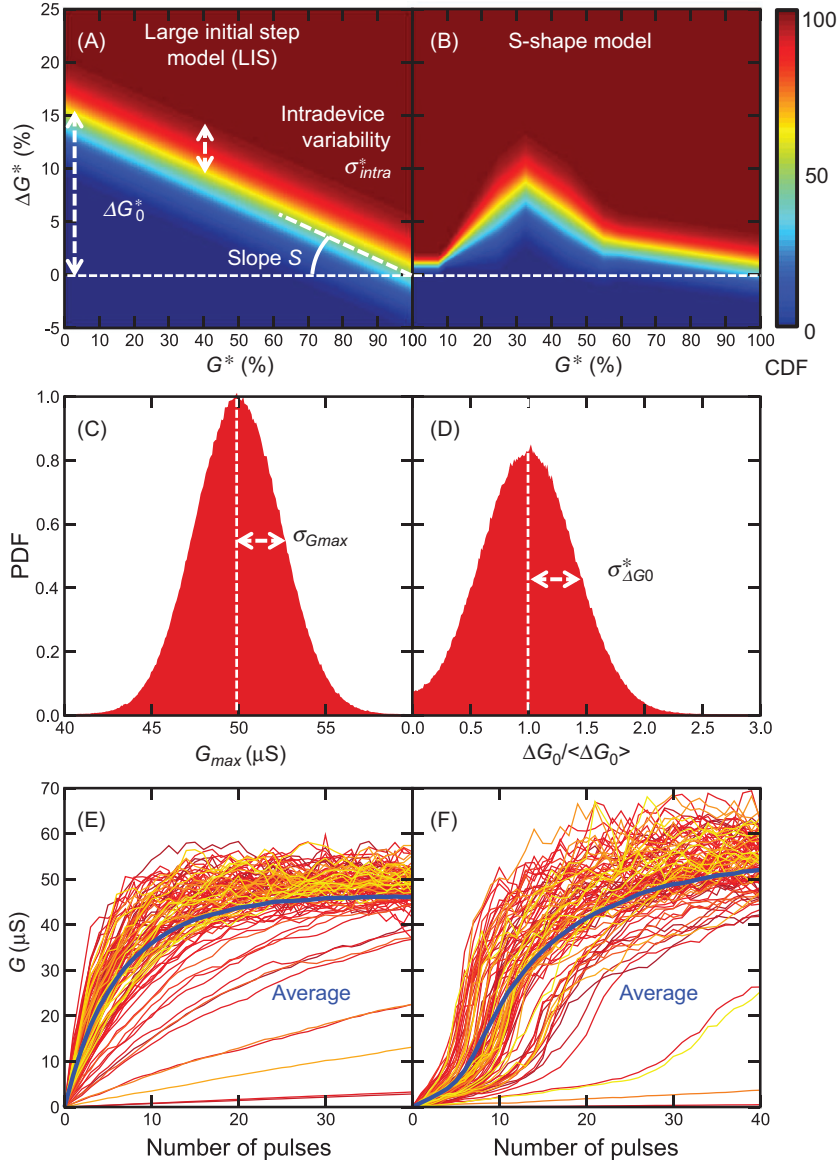


FIGURE 13.16 Simulated device model of a resistive device. (A) Two jump tables for large initial step (LIS) or (B) S-shaped models are studied, providing intradevice variability. The inter-device variability is obtained with variation of the maximum device conductance from (C) and initial ΔG_0 step from (D). Fifty simulated curves are shown (E) for LIS and (F) for S-shape models. *Adapted with permission from Cristiano G., Giordano M., Ambrogio S., Romero L., Cheng C., Narayanan P., et al., J. Appl. Phys. 124 (2018) 151901 [48].*

(or device-to-device) variability, the maximum achievable conductance G_{max} is extracted from the Gaussian distribution in Fig. 13.16C, while the initial step ΔG_0 shows a broadening described in Fig. 13.16D. Sampling from the described jump table models leads to the conductance traces reported in Fig. 13.16E for the LIS model and Fig. 13.16F for the PCM-like model, revealing a similar behavior with real data [48].

13.5.1 Most significant pair programming

The devices that represent the MSP are only programmed during transfers; therefore such devices are only touched every thousands of training images. For this reason, more time can be spent in the programming process, since such programming can be hidden into the neural network training, while a core is idling. Therefore, we can use a closed-loop tuning procedure (CLT), which is described in Fig. 13.17 [48].

The goal of CLT is to program a particular target weight into the MSP pair, for simplicity just represented by $W = G^+ - G^-$. Based on the actual error, defined as $|Target - (G^+ - G^-)|$, a variable number of programming pulses, from a minimum of zero (in case the error is small enough) to a maximum of 3 (for large errors) is fired on the devices, as shown in Fig. 13.17A. The single weight is read and reprogrammed up to 20 times (Fig. 13.17B), and three different targets of 5 μS, −5 μS and 45 μS are simulated in the G-diamond plots in Fig. 13.17C. In each case, we plot 50 different simulated runs. The simulations show an increased precision with a higher number of retries. Since MSP devices are programmed with CLT, there is no need for linear conductance change [48].

13.5.2 Dependence of accuracy on device nonidealities

To evaluate the impact of device nonidealities on neural network training, we start by defining a baseline. By using the jump tables shown in Fig. 13.18A and B for, respectively, MSP and LSP pairs, we achieve software-equivalent training accuracy on a four-layer MLP for MNIST, as shown in Fig. 13.18C. While the average accuracy is around 97.95%, we define as an acceptable tolerance in accuracy, a lower limit of 97.5% accuracy, which is based on the spread that twenty TensorFlow runs show [48].

In our experiments, we perform 20-epochs training simulations on the MNIST dataset, gradually increasing one nonideality of the MSP or the LSP and recording the corresponding test accuracy. Fig. 13.19A shows the sweep of the initial ΔG_0 step, which affects the number of available steps (or, equivalently, dynamic range) that the device shows, since a larger ΔG_0 covers a higher portion of the available conductance range.

Test accuracy results for MSP (Fig. 13.19B) and LSP (Fig. 13.19C) reveal that the MSP is highly resilient to ΔG_0 since the CLT procedure will

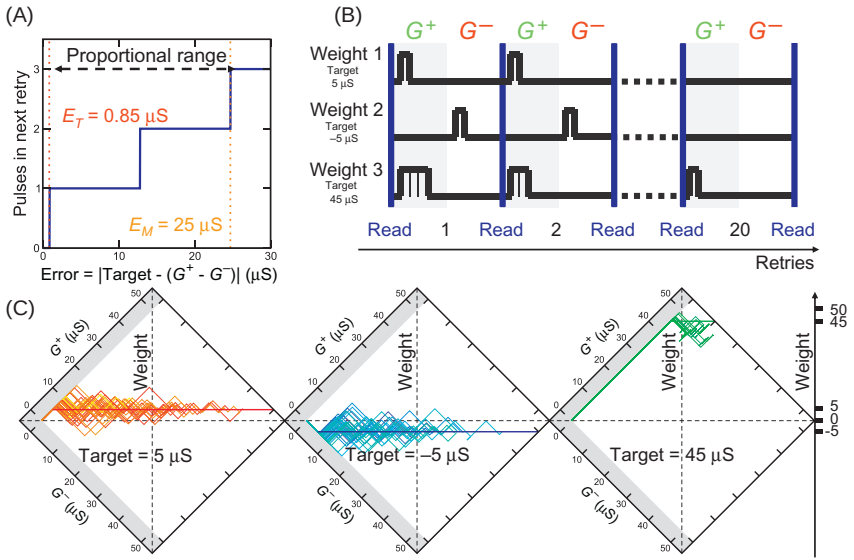


FIGURE 13.17 Closed-loop tuning procedure. Based on the error between the target and the actual weight, (A) a variable number of programming pulses is sent over G^+ or G^-, (B) with a maximum of 20 read and program operations. Three different targets are shown, 5 µS, −5 µS, and 45 µS in (C), with 50 different simulation runs for every target. *Adapted with permission from Cristiano G., Giordano M., Ambrogio S., Romero L., Cheng C., Narayanan P., et al., J. Appl. Phys. 124 (2018) 151901 [48].*

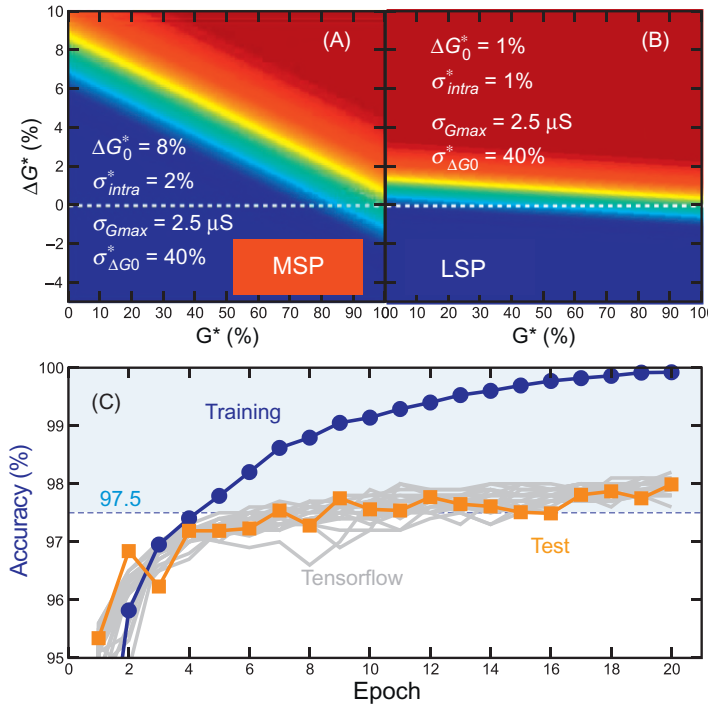


FIGURE 13.18 Ideal device model parameters defined for (A) MSP and (B) LSP. Using such devices, the accuracy obtained after training a MLP on the MNIST dataset is equivalent to that achieved with (C) TensorFlow. *Adapted with permission from Cristiano G., Giordano M., Ambrogio S., Romero L., Cheng C., Narayanan P., et al., J. Appl. Phys. 124 (2018) 151901 [48].*

exploit the smaller steps appearing at higher device conductance, even with a large initial conductance jump. On the other hand, a too small ΔG_0 leads to larger errors since, after 20 retries, the devices may still be near the initial value. The LSP requires a small enough ΔG_0 for frequent weight update, which corresponds to at least 110 device steps. This requirement is much more relaxed than the required 1000 steps for tuning a network with just one type of device [25].

Fig. 13.20 shows a summary of all results obtained varying many different device nonidealities. What is interesting is that different properties require different specifications on MSP and LSP, thus relaxing the overall constraints. In particular, while intradevice variability represents a concern for both MSP and LSP, interdevice variability shows little impact due to the adoption of CLT for MSP and Polarity Inversion for LSP. The dependence on faulty devices, namely, devices that are dead (conductance always zero) or stuck-on (conductance always high), reveal a larger tolerance for stuck-on in MSP thanks to CLT (even if one device shows high conductance, by

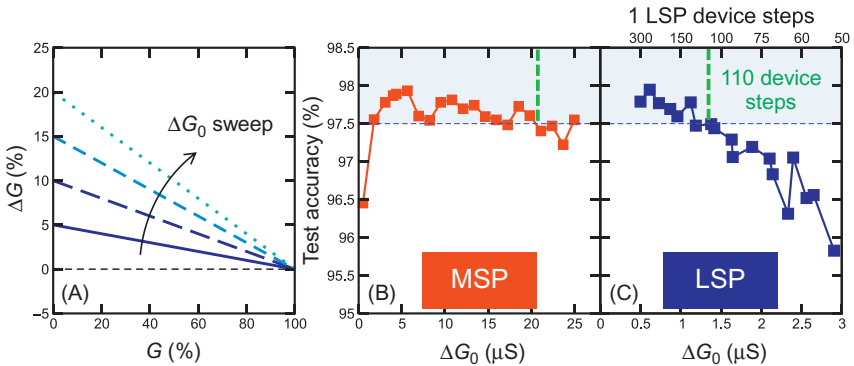


FIGURE 13.19 (A) Test accuracy results by gradually increasing the initial conductance step ΔG_0. (B) MSP shows a relatively high resiliency due to CLT, while LSP shows a degradation for larger ΔG_0. A minimum of 110 device steps is required to achieve acceptable accuracy. *Adapted with permission from Cristiano G., Giordano M., Ambrogio S., Romero L., Cheng C., Narayanan P., et al., J. Appl. Phys. 124 (2018) 151901 [48].*

Specifications	Parameter	MSP	LSP
Initial step-size	ΔG_0 (ΔG_0^*)	< 21 μS (42%)	< 1.4 μS (2.8%)
Intradevice variability	σ_{intra}	< 1.5 μS	< 0.8 μS
Interdevice variability	σ_{Gmax}	< 10 μS	< 12 μS
	$\sigma^*_{\Delta G0}$	< 200%	< 95%
Faulty devices	Dead C.R.	< 7%	< 7%
	Stuck on C.R.	< 35%	< 10%
Dynamic range	Number of levels	> 13	> 110
Retention	Time before data loss	Higher	Lower
Endurance	Number of set/reset	Lower	Higher

FIGURE 13.20 Summary of the tolerance results with respect to device nonidealities. *Adapted with permission from Cristiano G., Giordano M., Ambrogio S., Romero L., Cheng C., Narayanan P., et al., J. Appl. Phys. 124 (2018) 151901 [48].*

tuning the other device accordingly the error can be reduced). In addition, long-term retention is only needed for MSP devices since LSP are updated very frequently, with multiple updates in less than hundreds of $\mu - sec$. Finally, LSP endurance would need to be higher since LSP devices are frequently programmed, while MSP are only programmed during transfers [48].

13.6 Conclusions

We reviewed prior work on using PCMs for accelerating the training of deep neural networks, highlighting the challenge of achieving software-equivalent accuracy in the presence of nonlinear, asymmetric, and variable conductance change. While continued improvements in device engineering would help bridge the gap, in this chapter, we outlined a series of complementary techniques that can simplify device requirements. The multiple conductances of varying significance idea improve dynamic range and calls for a diversification between MSP and LSP devices. Specifically LSP devices would need linear and need high endurance, but can have significant device-to-device and cycle to cycle variability and asymmetry, whereas MSP devices have relaxed linearity and endurance targets, but would need sufficient dynamic range and long-term retention. The polarity inversion technique can provide resilience against fixed device asymmetries, averaging them out over longer time scales. By using these techniques, along with the 2T2R + 3T1C cell design, we demonstrated software-equivalent accuracy on MNIST and CIFAR benchmarks and quantified the requirements for an eventual MSP/LSP pair suitable for training.

References

[1] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Proceedings of the IEEE 86 (1998) 2278–2324.

[2] S. Hochreiter, J. Schmidhuber, Neural Computation 9 (1997) 1735–1780.

[3] J. Schmidhuber, Neural Networks 61 (2015) 85–117.

[4] Y. LeCun, Y. Bengio, G. Hinton, Nature 521 (2015) 436–444.

[5] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Nature 323 (1986) 533–536.

[6] Krizhevsky A., Sutskever I. and Hinton G.E. 2012 Imagenet classification with deep convolutional neural networks, In: P. Bartlett (Ed.), Advances in Neural Information Processing Systems 25: Proceedings of the 26th Annual Conference on Neural Information Processing Systems 2012.

[7] G. Hinton, L. Deng, D. Yu, G.E. Dahl, A.R. Mohamed, N. Jaitly, et al., IEEE Signal Processing Magazine 29 (2012) 82–97.

[8] Sutskever I., Vinyals O. and Le Q.V. 2014 Sequence to sequence learning with neural networks, In *Advances in Neural Information Processing Systems*, pp 3104–3112.

[9] Cho K., Van Merriënboer B., Bahdanau D. and Bengio Y. 2014 *arXiv preprint arXiv:1409.1259*.

[10] Bahdanau D., Cho K. and Bengio Y. 2014 *arXiv preprint arXiv:1409.0473*.

[11] Jouppi N.P., Young C., Patil N., Patterson D., Agrawal G., Bajwa R., et al., 2017 In-datacenter performance analysis of a tensor processing unit, In *Proceedings of the 44th Annual International Symposium on Computer Architecture* (ACM), pp 1–12.

[12] Fleischer B et al., 2018 A scalable multiteraops deep learning processor core for ai training and inference, In *2018 Symposium on VLSI Circuits Digest of Technical Papers* (Honolulu, HI).

[13] Sijstermans F. 2018 The NVIDIA Deep Learning Accelerator *Proceedings of Hot Chips 30* (Cupertino, CA).

[14] Lee D.U., Kim K.W., Kim K.W., Kim H., Kim J.Y., Park Y.J., et al., 2014 25.2 a 1.2v 8gb 8-channel 128gb/s high-bandwidth memory (hbm) stacked dram with effective microbump i/o test methods using 29nm process and tsv, *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, pp 432–433, ISSN 0193-6530.

[15] Chen C., Choi J., Gopalakrishnan K., Han V., Srinivasan V. and Zhang J. Matrix multiplication on a systolic array, US patent 20,180,267,938.

[16] Chen C., Choi J., Gopalakrishnan K., Srinivasan V. and Venkataramani S. 2018 Exploiting approximate computing for deep learning acceleration, *2018 Design, Automation Test in Europe Conference Exhibition (DATE)*, pp 821–826, ISSN 1558-1101.

[17] S. Gupta, A. Agrawal, K. Gopalakrishnan, P. Narayanan, Deep learning with limited numerical precision, In, in: F. Bach, D. Blei (Eds.), Proceedings of the 32nd International Conference on Machine Learning (Proceedings of Machine Learning Research, vol 37, PMLR, Lille, France, 2015, pp. 1737–1746. URL http://proceedings.mlr.press/v37/gupta15.html.

[18] Courbariaux M., Bengio Y. and David J.P. 2015 Binaryconnect: Training deep neural networks with binary weights during propagations, *Advances in Neural Information Processing Systems*, pp 3123–3131.

[19] Fick D. and Henry M. 2018 Analog computation in flash memory for datacenter-scale ai inference in a small chip, *Proceedings of Hot Chips 30* (Cupertino, CA).

[20] Suri M., Bichler O., Querlioz D., Cueto O., Perniola L., Sousa V., et al., 2011, Phase change memory as synapse for ultra-dense neuromorphic systems: Application to complex visual pattern extraction, *2011 International Electron Devices Meeting*, pp. 79–82.

[21] P.A. Merolla, J.V. Arthur, R. Alvarez-Icaza, A.S. Cassidy, J. Sawada, F. Akopyan, et al., Science 345 (2014) 668–673. ISSN 0036- 8075 (*Preprint* http://science.sciencemag.org/content/345/6197/668.full.pdf) URL http://science.sciencemag.org/content/345/6197/668.

[22] G.W. Burr, B.N. Kurdi, J.C. Scott, C.H. Lam, K. Gopalakrishnan, R.S. Shenoy, IBM Journal of Research and Development 52 (2008) 449–464. ISSN 0018-8646.

[23] H.P. Wong, S. Raoux, S. Kim, J. Liang, J.P. Reifenberg, B. Rajendran, et al., Proceedings of the IEEE 98 (2010) 2201–2227. ISSN 0018-9219.

[24] Burr G., Shelby R.M., di Nolfo C., Jang J., Shenoy R., Narayanan P., et al., 2014 *IEDM Technical Digest* T29.5.

[25] T. Gokmen, Y. Vlasov, Frontiers in Neuroscience 10 (2016).

[26] G.W. Burr, R.M. Shelby, A. Sebastian, S. Kim, S. Kim, S. Sidler, et al., Advances in Physics: X 2 (2017) 89–124.

[27] Rosenblatt F. 1957 The perceptron–a perceiving and recognizing automaton, Technical Report 85-460-1, Cornell Aeronautical Laboratory.

[28] V. Sze, IEEE Solid-State Circuits Magazine 9 (2017) 46–54. ISSN 1943-0582 sze:2017.

[29] V. Sze, Y.H. Chen, T.J. Yang, J.S. Emer, Proceedings of the IEEE 105 (2017) 2295–2329.

[30] P. Narayanan, A. Fumarola, L. Sanches, S. Lewis, K. Hosokawa, R.M. Shelby, et al., IBM Journal of Research and Development 61 (11) (2017) 1–11.
[31] Ng A. Machine learning cs229.stanford.edu.
[32] Li F.F. and Karpathy A. Convolutional neural networks for visual recognition cs231n.stanford.edu.
[33] Socher R. Deep learning for natural language processing cs224d.stanford.edu.
[34] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, Cambridge, MA, 2016. Available from: http://www.deeplearningbook.org.
[35] H.Y. Tsai, S. Ambrogio, P. Narayanan, R. Shelby, G.W. Burr, Journal of Physics D: Applied Physics 51 (2018).
[36] G.W. Burr, R.M. Shelby, S. Sidler, C. Di Nolfo, J. Jang, I. Boybat, et al., IEEE Transactions on Electron Devices 62 (2015) 3498–3507.
[37] Choi J., Wang Z., Venkataramani S., Chuang P.I., Srinivasan V. and Gopalakrishnan K. 2018 *CoRR* abs/1805.06085 (*Preprint* 1805.06085), URL http://arxiv.org/abs/1805.06085
[38] He K., Zhang X., Ren S. and Sun J. 2015 *CoRR* abs/1512.03385 (*Preprint* 1512.03385), URL http://arxiv.org/abs/1512.03385
[39] S. Ambrogio, P. Narayanan, H. Tsai, R.M. Shelby, I. Boybat, C. di Nolfo, et al., Nature 558 (2018) 60.
[40] I. Boybat, M.L. Gallo, R.N.S. Moraitis, T. Parnell, T. Tuma, T. Rajendran, et al., Nature Communications 9 (2018) 2514.
[41] S. R. Nandakumar, M. Le Gallo, I. Boybat, B. Rajendran, A. Sebastian and E. Eleftheriou, "Mixed-precision architecture based on computational memory for training deep neural networks," *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, Florence, 2018, pp. 1–5.
[42] Gokmen T., Onen O.M. and Haensch W. 2017, Training deep convolutional neural networks with resistive cross-point devices, *Front. Neurosci.* **11**, 2017, 538. http://frontiersin.org/article/10.3389/fnins.2017.00538
[43] D. Kuzum, R. Jeyasingh, B. Lee, H.S.P. Wong, Nano Letters 12 (2011) 2179–2186.
[44] O. Bichler, M. Suri, D. Querlioz, D. Vuillaume, B. DeSalvo, C. Gamrat, IEEE Transactions on Electron Devices 59 (2012) 2206–2214.
[45] B.L. Jackson, B. Rajendran, G.S. Corrado, M. Breitwisch, G.W. Burr, R. Cheek, et al., ACM Journal on Emerging Technologies in Computing Systems (JETC) 9 (2013) 12.
[46] S. Ambrogio, N. Ciocchini, M. Laudato, V. Milo, A. Pirovano, P. Fantini, et al., Frontiers in Neuroscience 10 (2016) 56.
[47] M.L. Gallo, A. Sebastian, R. Mathis, M. Manica, H. Giefers, T. Tuma, et al., Nature Electronics 1 (2018) 246–253.
[48] G. Cristiano, M. Giordano, S. Ambrogio, L. Romero, C. Cheng, P. Narayanan, et al., Journal of Applied Physics 124 (2018) 151901.
[49] J. Woo, S. Yu, IEEE Nanotechnology Magazine 12 (2018) 36–44.
[50] Agarwal S., Gedrim R.B.J., Hsia A.H., Hughart D.R., Fuller E.J., Talin A.A., et al., 2017 Achieving ideal accuracies in analog neuromorphic computing using periodic carry, In *VLSI Technology, 2017 Symposium on* (IEEE), pp T174–T175.
[51] Cao Y. 2009 *SIGDA Newsletter* 39 ISSN 0163-5743, URL http://doi.acm.org/10.1145/1862891.1862892
[52] Cao Y. Predictive Technology Model (PTM) [Online], ptm.asu.edu
[53] Variations on the mnist digits [Online], www.iro.umontreal.ca/lisa/twiki/bin/view.cgi/Public/MnistVariations

[54] Krizhevsky A. 2009 Learning multiple layers of features from tiny images [Online], https://www.cs.toronto.edu/kriz/cifar.html
[55] Inception in TensorFlow [Online], github.com/tensorflow/models/tree/master/inception
[56] How to retrain inception's final layer for new categories [Online], www.tensorflow.org/tutorials/imageretraining

Chapter 14

RRAM-based coprocessors for deep learning

Ying Zhou[1], Bin Gao[1], Chunmeng Dou[2], Meng-Fan Chang[2] and Huaqiang Wu[1]

[1]Institute of Microelectronics, Tsinghua University, Beijing, P.R. China, [2]Electrical Engineering Department, National Tsing Hua University, Hsinchu, Taiwan, ROC

14.1 Introduction

The most crucial elements of brains are neurons and synapses. Synapses transfer signals and serve as the connection of two neurons and include excitatory and inhibitory synapses. The connection strength between two neurons is characterized by the synaptic weight. A significant feature of synapses is their plasticity, which allows learning. In emerging neuromorphic computing, various devices and structures have been proposed to implement the function of synapses, such as phase change memory (PCM) [1], complementary metal oxide semiconductor (CMOS) [2], spin transfer torque magnetic random access memory [3], conductive bridge random access memory (CBRAM) [4], and ferroelectric random access memory [1], and are introduced throughout this book. However, their high area and energy cost can make it difficult to develop a deep learning system based on those technologies. In recent years, resistive random access memory (RRAM) has drawn a lot of attention from researchers for its similarity with plastic synapses of the brain, giving it the potential to provide learning ability. RRAM is a promising candidate in neuromorphic computing due to its simple device structure, low energy consumption, fast operation speed, high scalability, and easiness to integrate in a 3D manner [5–7]. Furthermore, the fabrication process of RRAM is compatible with existing CMOS technology [5]. RRAM has adjustable resistance switching characteristic between a high-resistance state (HRS) and a low-resistance state (LRS). The device conductance can be modulated by biasing voltage on the electrodes of the device. Owing to functional similarity with synapse, RRAM plays a role of synapse in neuromorphic computing. Unfortunately, due to dynamic stochastic generation recombination and drift diffusion of multiple oxygen ions/vacancies, the SET

Memristive Devices for Brain-Inspired Computing. DOI: https://doi.org/10.1016/B978-0-08-102782-0.00014-9

(transition from low conductance state to high conductance state) and RESET (transition from high conductance state to low conductance state) processes of RRAM exhibit various nonideal characteristics, which are the inevitable challenges for RRAM-based neural network (NN).

To implement NNs with good performance, many requirements of device characteristic have been suggested. First, RRAM's stochastic variation is a key issue in array integration. RRAM suffers from two types of variation: device-to-device (also called nonuniformity) and cycle-to-cycle. Bad uniformity of device is an obstacle toward practical application [7]. Retention characteristic is another evaluation aspect for device variation. It is defined as the ability of devices of maintaining conductance state and reflects a variation in the time dimension. A long retention/relaxation time is required for a stable and high accuracy in some NN application. Besides, endurance is a significant factor that affects the reliability of devices and influences the robustness of NN. Due to frequent programming operation, good endurance characteristic is also required for online training [8].

In addition to factors mentioned above, RRAM can be divided into three categories according to conductance state: binary RRAM, multilevel RRAM, and analog RRAM. For binary RRAM, two conductance states are existing: HRS and LRS. In the $I-V$ curve of binary RRAM, SET process is abrupt and RESET process exhibits gradual transition [9]. RRAM devices also feature the potential to obtain multiple conductance levels and even continuous analog conductance [10], as shown in Fig. 14.1A. The LRS indeed exhibits high dependence on the current compliance in the SET operation, I_{SET}. The stop voltage in RESET operation, V_{stop}, accounts for HRS tuning. In multilevel conductance state adjustment process, program-and-verify method is usually utilized [13,14]. To obtain RRAM's multilevel conductance state, two programming methods have been introduced. First, LRS could be tuned by carefully controlling SET compliance current [10]. The modulation result is illustrated in Fig. 14.1B. The second method consists in adjusting the HRS by controlling RESET stop voltage, as shown in Fig. 14.1C. As for tuning different HRS, larger V_{stop} results in larger HRS. However, obtained HRS levels can easily be overlapping between each other in this operation method. This method has been investigated worldwide to practically obtain separated conductance level. Another type of methods is currently emerging: devices exhibit various conductance states under varying linearly RESET pulse amplitudes. In addition to adjusting programming amplitude, longer or shorter applied pulse width can change the state of devices more or less rapidly [15]. Therefore modulating the width of the voltage pulse is also a practical method to get multilevel conductance state [16,17]. The width of the voltage pulse can alternatively be implemented by applying a number of fixed pulse width voltage pulses.

Finally, by carefully controlling SET current compliance and RESET voltage during the programming process, analog characteristic can

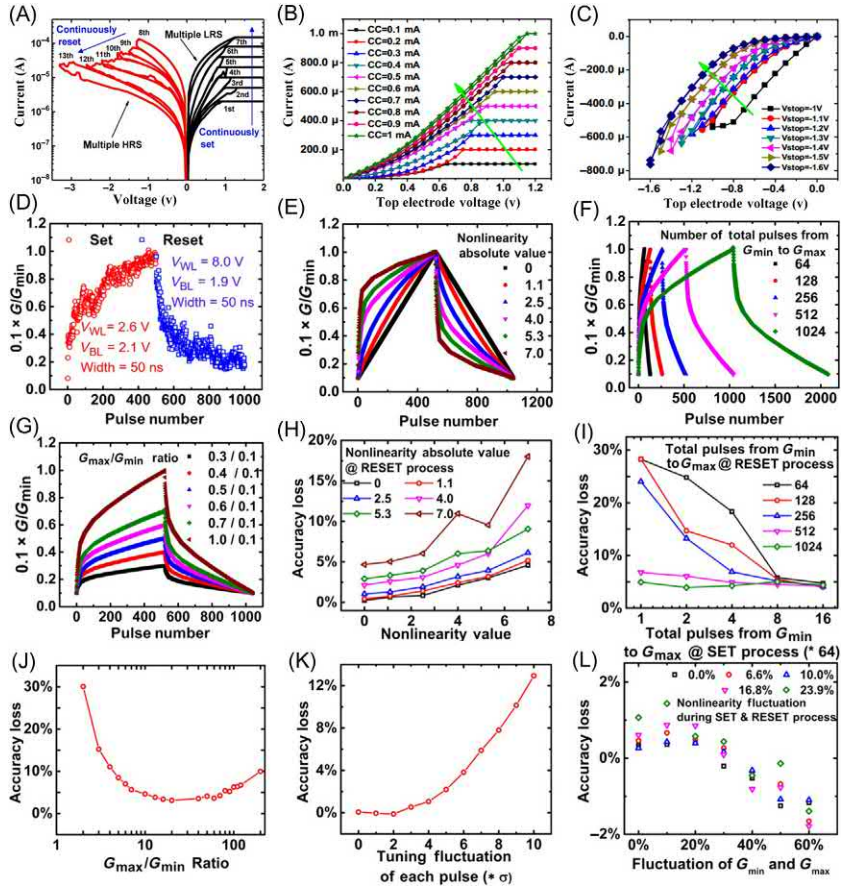


FIGURE 14.1 Multilevel operation results (A). *I*−*V* curves of RRAM device. Multiple LRS states are obtained by setting varying SET current compliances. Different RESET stop voltages account for multiple HRS states (B). In SET process, device's conductance state is increasing with the increase in SET current compliance (C). In RESET process, device's conductance state is decreasing with the increase in RESET stop voltage amplitude. (D) Conductance change curve under fixed pulses. Device shows bidirectional and analog characteristics. (E−G) Conductance change curve under different nonlinearity values, various total pulses, different G_{max}/G_{min} ratios. (H−K) Accuracy losses of recognition network under different nonlinearity values, total pulses from G_{min} to G_{max}, G_{max}/G_{min} ratios, tuning fluctuation of each pulse, nonlinearity oscillation, and G_{min} and G_{max} fluctuation, respectively. *Adapted from H. Wu, et al., 2017. IEEE International Electron Devices Meeting, pp. 11.5.1−11.5.4 [6]; S. Yu, Y. Wu, R. Jeyasingh, D. Kuzum, H.S.P. Wong, An electronic synapse device based on metal oxide resistive switching memory for neuromorphic computation. IEEE Trans. Electron. Devices 58 (2011) 2729−2737 [11]; Y. Wu, S. Yu, H.S.P. Wong, Y.S. Chen, 2015 IEEE International Memory Workshop, pp. 1−4 [12].*

sometimes be realized [10,11]. As shown in Fig. 14.1D, SET and RESET transition curves in analog RRAM are continuously gradual, which is a highly desirable characteristic for an electronic synapse capable of online training [6,18]. Bidirectional and analog switching behaviors are the most preferable characteristics for NN application. Unlike abrupt conductance change in other resistive devices, conductance of bidirectional analog RRAM could be increased/decreased gradually in SET and RESET process. Thus this type of RRAM device is a promising candidate for realizing the long-term potentiation or long-term depression associated with the online learning process [6]. Linearity characteristics, G_{max}/G_{min} ratio, fluctuation, and total number of pulses from G_{min} to G_{max} are of great importance factors for online training. Linearity means the linear conductance change under successive programming pulses. Fig. 14.1E shows conductance change under different nonlinearity values. Each synaptic weight can be potentiated or depressed by pulses, unconcerned of the initial state, which characterizes the high linearity of a device [19]. Nonlinearity results in the degradation of network performance. For example, in a face classification experiment, the inference accuracy is higher when the nonlinearity value is close to zero [6]. Fig. 14.1F and G, respectively, shows conductance change under different total pulses and G_{max}/G_{min} ratios. Under various nonideal factors, accuracy losses of image recognition network are respectively illustrated in Fig. 14.1H–K. In particular, accuracy is not higher under higher G_{max}/G_{min} ratio according to results. Besides, effect of fluctuation is illustrated in Fig. 14.1L, including conductance tuning fluctuation at each pulse, nonlinearity oscillation, and G_{min} and G_{max} fluctuation [6].

There are approximately 10^{15} synapses and 10^{12} neurons in human brain, at a cost of about 20 W energy in processing a task while multicore supercomputing requires 1000 times more energy [20]. To approach human brain synapse density, NN implementation really is a challenge. As in NNs, the number of synapses is much larger than neurons, synapse implementation is more crucial. Unfortunately, CMOS synapses occupy large area, which severely limits the scale of NN implementation [1]. Instead, many emerging synaptic elements have drawn attention, such as PCM [21], CBRAM [4], and RRAM [22]. The simple structure of RRAM makes it possible to integrate synaptic devices in a chip to obtain high device density. Furthermore, RRAM's conductance bidirectional adjustable ability makes it promising synaptic device in NN implementation [6]. In this chapter, we would introduce NN implementation work based on RRAM in detail, including significant researches in demonstration and circuit realization.

As seen in other chapters of the book, 1T1R [5], 1S1R [23], and 0T1R [24] are existing RRAM structures. RRAM's nonvolatile characteristic makes it capable of both computation and storage. Selector device plays a role of current limiter, which is capable of avoiding current overshooting phenomenon. Selection wires are regular grid distribution with two terminal

RRAM cells located in each cross point. This architecture is widely called crossbar. Wires play a role of axons and dendrites. RRAM device corresponds to biological synapses [5]. RRAM crossbar architecture is suitable for vector-matrix multiplication (VMM). In this process, dense voltage vector is applied in the input wires of array. According to Ohm's law and Kirchhoff's Current Law, current flow each output wire corresponds to the product result of vector and matrix. Thus this basic computation operation in NN could be easily realized. Peripheral CMOS circuit plays selection wire part to address each RRAM cell. This computation method mimics massive parallel computing in biological brain. VMM method saves lots of energy consumption and processing time. Brain-inspired computing, which is also called neuromorphic computing, is a computing revolution for accelerating NN. In [25], two discrete RRAM devices connect with amplifier to realize simple accelerator. Furthermore, more complicated RRAM-based accelerators have been proposed [26–28]. They are effectively applied in NN demonstration. In learning process of NN, RRAM cells are modulated to target weight matrix value by online or offline training approach, using various training algorithms. Several RRAM-based demonstrations of NN are introduced in the following parts.

14.2 NN applications based on RRAM

14.2.1 Related simulation work

NN implementation has successfully realized many intelligent tasks, for example, pattern classification, position detection, data clustering, and information's storage and recovery. In this part, we will introduce previous NN simulation work based on RRAM.

To be adapted to binary RRAM devices, a modified K-nearest neighbor algorithm's implement approach is proposed in Ref. [9]. The RRAM crossbar is employed to realize pattern classification. A simulated architecture is validated with high recognition accuracy using handwritten digits on Mixed National Institute of Standards and Technology database (MNIST) database. The simulation result exhibits high tolerance to input noise. The deviation of weight values also has little impact on robustness of network and final recognition effect [29].

Besides, Ref. [30] implements the simulation of analog neuromorphic network to achieve the recognition of gray scale images. Hybrid CMOS and RRAM integration architecture is exploited in this research. The whole demonstration structure is illustrated in Fig. 14.2A. Synaptic RRAM device characteristic has multiple piecewise transitions. Various training stages are utilized and correspond to different gray scale bars, as shown in Fig. 14.2B. Simulation results are obtained and include various recognition accuracy, training speed, and energy consumption.

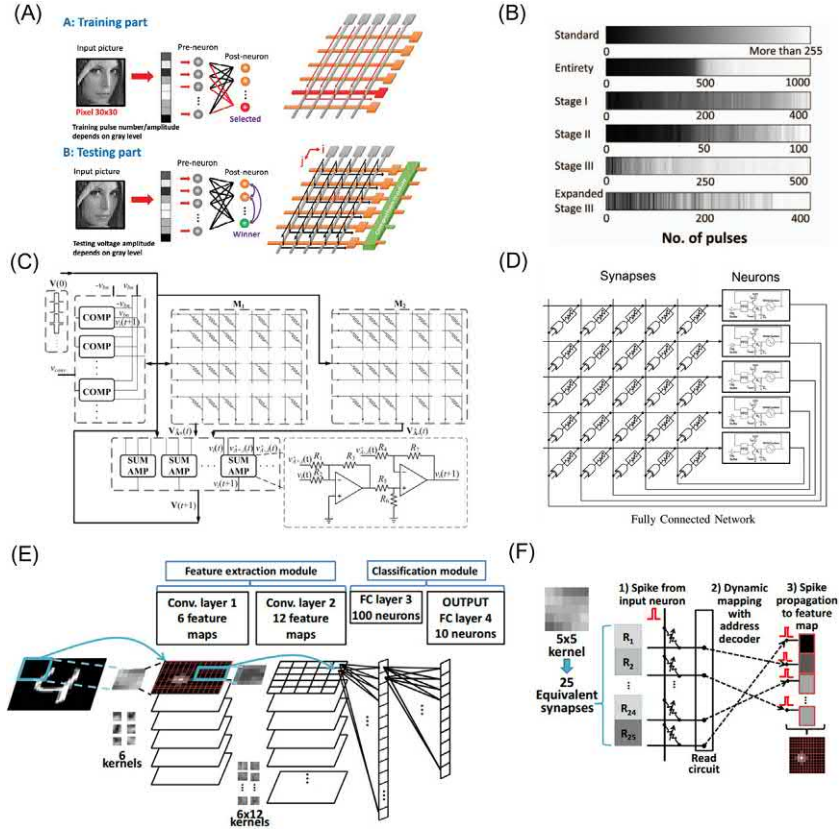


FIGURE 14.2 (A) Gray scale images' recognition implementation architecture. Training and test processes are also included. Two-layer perceptron network is utilized with integrated hybrid CMOS and RRAM arrays. Postneurons receive signals from preneurons with synaptic weights. (b) Various gray scale bars correspond to different training stages. Some are smoothly, and some are unevenly. Among them, stage I has the smoothest transition of conductance. (C) Implementation circuit design of BSB recall network. It contains two RRAM array, summing amplifiers and comparators and so on. (D) ONN implementation architecture including neurons and synapses. Synapse consists of XOR gate and RRAM, and XOR ensure that the RRAM's voltage does not exceed the upper limit of state change. Neurons compose of phase-locked loop- and RRAM-based oscillator. (E) CNN architecture for MNIST recognition task. (F) Process of convolutional kernel mapping to network; each synapse is composed of 20 RRAM devices. *Adapted from Z. Chen, B. Gao, Z. Zhou, P. Huang, 2015 IEEE International Electron Devices Meeting, pp. 17.7.1–17.7.4 [30]; M. Hu, et al., Memristor crossbar-based neuromorphic computing system: a case study. IEEE Trans. Neural Netw. Learn. Syst. 25 (2014) 1864–1878 [31]; T.C. Jackson, A.A. Sharma, J.A. Bain, J.A. Weldon, L. Pileggi, Oscillatory neural networks based on TMO nano-oscillators and multi-level RRAM cells. IEEE J. Emerg. Sel. Top. Circ. Syst. 5 (2015) 230–241 [32]; D. Garbin, O. Bichler, E. Vianello, Q. Rafhay, 2014 International Electron Devices Meeting, pp. 28.4.1–28.4.4 [33].*

In addition to pattern recognition, information's recovery is an attractive research topic. Autoassociative NN is a kind of frequently utilized NN. It is capable of recalling prestored information. In this network, the desired patterns are stored as stable state. Among its description theories and models, the brain-state-in-a-box (BSB) model is a notable model. A realization framework on hardware of this algorithm is introduced in Ref. [31]. Two training methods of implementation of BSB NN have been proposed, including mapping and online training, with Delta rule [34]. In Monte Carlo simulation process, two RRAM crossbar arrays are employed to implement both positive and negative weight values. A complete synapse corresponds to two RRAM cells, as illustrated in Fig. 14.2C. Other processing circuit's design is also considered, such as summing amplifier, comparator. At the end of the training process, this architecture achieves success in English character retrieval task. It shows a tolerance to image's random defects, including line defects and point defects. The summing amplifier's resolution and correlation of the two crossbar arrays have most impact on overall performance.

Another significant model of associative memory, Hopfield NN (HNN) is previously demonstrated in [35,36]. HNNs containing 4 neurons or 64 neurons are simulated while excitatory and inhibitory synapses are realized by 1T1R configuration of RRAM. Above investigations pave toward hardware implementation of associative memory.

Oscillatory NN (ONN) is a fully connected neuromorphic network, which is capable of summing signal and storing and recovering patterns. It can be seen as a kind of associative memory. However, enormous digital-to-analog converters are utilized to realize analog synapse in CMOS technology. It makes implementation circuit more complicated. RRAM's intrinsic analog programming characteristic is ideally suitable for ONN implementation. Ref. [32] proposes RRAM-based ONN demonstration architecture, as shown in Fig. 14.2D. In this NN, weight values are learned by a Hebbian rule. Each RRAM device connects with additional XOR gates to compose a complete synapse. Compared with CMOS-based ONN, these extra gates slightly increase area and power consumption. In addition, using simple logic gate, each neuron completes function of phase initialization and locking, namely phase-locked loops. Neurons receive signals from other connected neurons and sum it. Their output signal phase is jointly determined by connected synapse value. Eight neurons and 20 neurons have been simulated using offline training. Some simple black and white patterns are utilized in this simulation work. White and black squares in stored patterns represent 180° and 0°, respectively. In the form of output neuron's signal phase, desired data are memorized in network. The ONN has been validated to have the ability to store and recall stored patterns. When fed into noise pattern, network would settle down to the nearest desired pattern, which is memorized as energy minima of whole network.

Convolutional NN (CNN) implementation is also a significant research direction. With graphics processing units (GPU) and CPU, implementation of CNN on software has shown an outstanding ability for pattern classification, which has been widely used in the commercial field. However, the multiplication in convolution computation costs lots of area and energy and is an obstacle to integration into further portable application. Hardware implementation therefore becomes an urgent problem to be solved, and the intrinsic characteristic of RRAM provides a new way toward CNN's hardware implementation.

A spiking architecture of CNN has been first realized, while RRAM accomplishes convolutional kernel function in Ref. [33]. CNN implementation process is shown in Fig. 14.2E. The realization of system simulation of CNN has been implemented with HfO_2-based RRAM. The whole architecture is composed of two cascaded convolutional layers to extract image feature and two fully connected layer to complete recognition ability. In convolutional layer, each kernel synapse is composed of 20 RRAM device in array's same row, as illustrated in Fig. 14.2F. They are dynamically mapped to output neurons due to addressing decoder. Input neurons are applied with spike pulse. Signals are spread to neurons in output layer after multiplication and sum operation. Final recognition class corresponds to the output Integrate and Fire neuron with the highest spike signal. Kernel synapses are programmed by two unsupervised learning methods, offline and online. Because of synaptic sharing in the inference process, it is possible to reduce array size at a cost of more programming operation. Switching event of fully connected NN is increased, which requires high endurance and low energy consumption. In simulation, the visual classification of handwritten digit is successfully realized in this work and high recognition accuracy is obtained. This work proposed spike-based CNN with RRAM and validates architecture's feasibility for CNN implementation.

Besides, Ref. [37] mentions that resistive processing units (RPUs) could successfully perform CNN training besides fully connected network. The mapping method of convolutional computation in a single layer is illustrated in Fig. 14.3A. All parameters of one convolutional layer for training are stored in matrix K. Different convolutional kernels are separately stored as column vector. In forward computation, the most significant operation is the convolution, which is exactly matrix−matrix multiplication. In each convolutional layer, multiplication of kernel and input data matrix is mapped to RPU array by following rearrangement operation. Original convolutional operation is separated to the following equation: $y = Kx$. Input vector x is a part of whole input vector. By repeatedly moving computation location of x, complete output results are obtained. By above reconfiguration operation, matrix−matrix multiplication $Y = KX$ is achieved, with matrix Y corresponding to the output result. In backward convolutional computation, backward cycle of convolutional layer can be calculated by equation $Z = K^T D$. D is

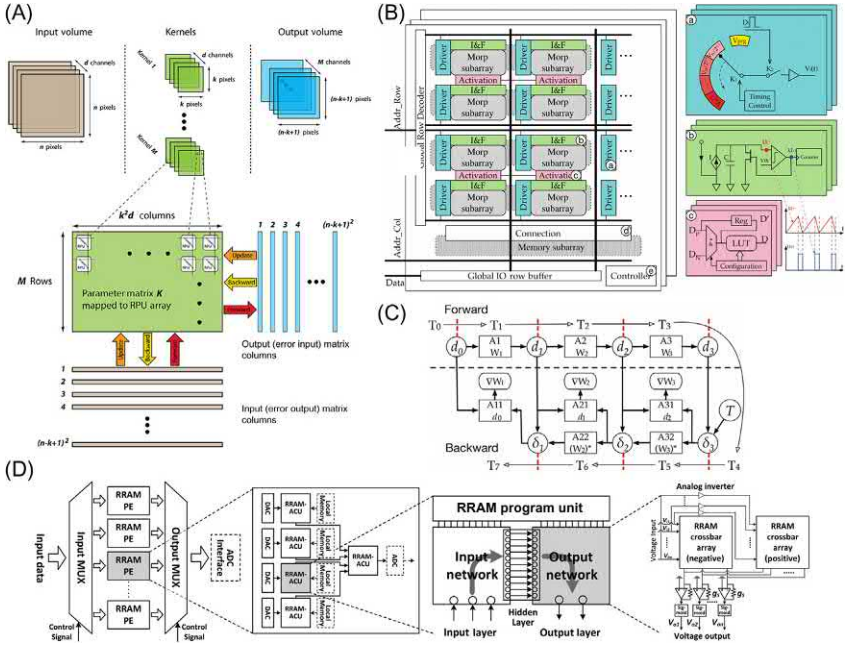


FIGURE 14.3 (A) Convolutional mapping method in single layer. Dimension of input matrix X is $k^2d \times (n-k+1)^2$ while k is $M \times k^2d$ dimension. As a result, output matrix Y is obtained with $M \times (n-k+1)^2$. (B) PipeLayer architecture. Detailed design parts contain spike driver, activation function part and integration and fire circuit. (C) CNN training process with PipeLayer. (D) Analog RRAM-based ACU framework. (B) Three-layer network architecture, including input, output, and hidden layers. *Adapted from B. Li, et al., RRAM-based analog approximate computing. IEEE Trans. Comput. Des. Integr. Circ. Syst. 34 (2015) 1905–1917 [14]; T. Gokmen, M. Onen, W. Haensch, Training deep convolutional neural networks with resistive cross-point devices. Front. Neurosci. 11 (2017) [37]; L. Song, X. Qian, H. Li, Y. Chen, 2017 IEEE International Symposium on High PERFORMANCE Computer Architecture, pp. 541–552 [38].*

error matrix while K^T is the transpose of parameter matrix. Updated difference value of matrix K is calculated by D and X. As accelerator, RPU could naturally execute parallel computing, saving many resources and running time. Each convolutional layer is followed by a pooling layer. Secondary pooling layer's output connects with fully connected layer. The entire network architecture exhibits good recognition accuracy in simulation result. Without any extra analog circuit, digital programming circuit could address the abovementioned analog computation problems, such as increased sensitivity to noise. Improved update rules and device optimization could further ensure the implementation of CNN training. All results exhibit practicability of RPU-based accelerator for CNN training.

A great amount of computation tasks are sequentially executed in CNN implementation: inference and update data transformation in multiple layer, which result in potential timing problems. In Ref. [38], a special layer is proposed to accelerate both CNN's training and test process, the PipeLayer. Computation is executed without redundant processing unit with PipeLayer architecture. By contrast, another proposed famous architecture, PRIME, has no detailed data transfer and convolutional kernel mapping operation. Moreover, in ISAAC [39], which features pipeline architecture, at the end of each cycle, results are calculated and transferred to next the cycle after massive new data are entered in network. The pipeline organization increases the throughput of the NN. However, when limited amount of data is utilized in the training process, ISAAC's pipeline is not realistic. By contrast, the PipeLayer architecture is exploited to support training and test operation, as shown in Fig. 14.3B. RRAM arrays are divided into two types, morphable subarrays and memory subarrays. Morphable subarrays play the role of both memory and computation, while memory subarrays are only used as storage buffer. Because of calculation dependence to results before the layer, the training process is more complicated than test process. The entire pipeline organization is divided into several stages, including forward and backward computation. Data transfer into network and execute computation as consecutive pipeline, as illustrated in Fig. 14.3C. In each computation stage, calculation result is fed into next stage and new image data are fed into first stage. In backward computation stage, error results of each layer and partial derivatives of weight value are temporarily stored in memory subarrays. The whole weight matrix keeps unchanged. After a batch of experiment images is fed into network, weight values are updated according to previous all intermediate results stored in memory subarrays. In PipeLayer, VMM operation could be performed without participation of additional processing units. With parallelism of intra- and interlayer, simulation results prove the efficiency of PipeLayer architecture. Compared with implementation in GPU platform, it exhibits a high superiority in energy-saving and speedup.

Approximate computing is also an NN research field. It consists in a tradeoff between efficiency and power consumption [40]. Various common functions could be performed in this way such as dot product, log function, and sinusoidal function. An analog RRAM-based approximate computing unit (ACU) framework is proposed in Ref. [14] to accelerate approximate computing operation, as shown in Fig. 14.3D. Approximate computation consists of two types of operation, VMM and sigmoid activation function. With efficient proposed scheme, data to be calculated are programmed to RRAM array. The feasibility of RRAM-based ACU is also simulated in three-layer perceptron network. Simulation results exhibit that analog RRAM-based approximate computing is more efficient in power consumption compared with digital implementation in CPU, GPU, and field-programmable gate array.

Speech recognition is also an attractive research field. With RRAM-based analog neuromorphic system, Ref. [41] presents first auditory recall demonstration. Leaky Integrate and Fire neuron model is utilized in this work. Synaptic weights are exploited fully unsupervised learning rule. Using electroencephalography (EEG) experiment, speech system is successfully demonstrated. In this work, three vowels are utilized with large difference in frequency analysis result. In EEG experiment, these vowel signals are extracted from eight electrodes in the right and left temporal areas. The human brain exhibits different activation situations to various vowel speeches, including activation area and response timing. The whole speech experiment flow consists of signal preprocessing, speech learning, and test process. A total of 1000 extracted spike signals in input layer are randomly connected with 400 neurons in the first layer of feed-forward spiking NN. This network use 400 synaptic connected RRAM in connection layer and a single output neuron. Winner-take all rule is exploited to obtain output result. Demonstration result exhibits a good speech performance. Prediction success rate could achieve more than 90%. This work paves the way toward biological interfaces and attractive speech application with RRAM device.

14.2.2 Experimental implementation

14.2.2.1 Associative memory

In biological level, one of the most famous examples of associative memory is the Pavlov's dog experiment [42]. This experiment exhibits that dog obtains the association of sight of food and special sound with salivation phenomenon after several training periods. It is a simplified realization of famous Hebbian rule. In Ref. [43], experimental demonstration of Pavlov's dog has been implemented. Two discrete connection RRAM devices and three electronic neurons (two input neurons and one output neuron) are applied in this experiment.

Many theories and models have been proposed to describe associative memory. Among them, HNN, a kind of recurrent NN (RNN), is one of the most notable theories, as shown in Fig. 14.4A. HNN's implementation in hardware has been successfully realized using RRAM in Ref. [44]. Some desired patterns can be memorized in network in the form of local energy minima. By modulating synaptic weight matrix, the network can store different information. In the experimental process, weight values are learned using Hebbian rules in software. Target weight values are mapped to RRAM devices through repeated programming operation. What is more, owing to intrinsic symmetric characteristic of HNN, six discrete RRAM devices in total are employed in the experimental demonstration process, as shown in Fig. 14.4B. The entire network is realized using RRAM and other electronic elements. Other processing parts are integrated on a printed circuit board

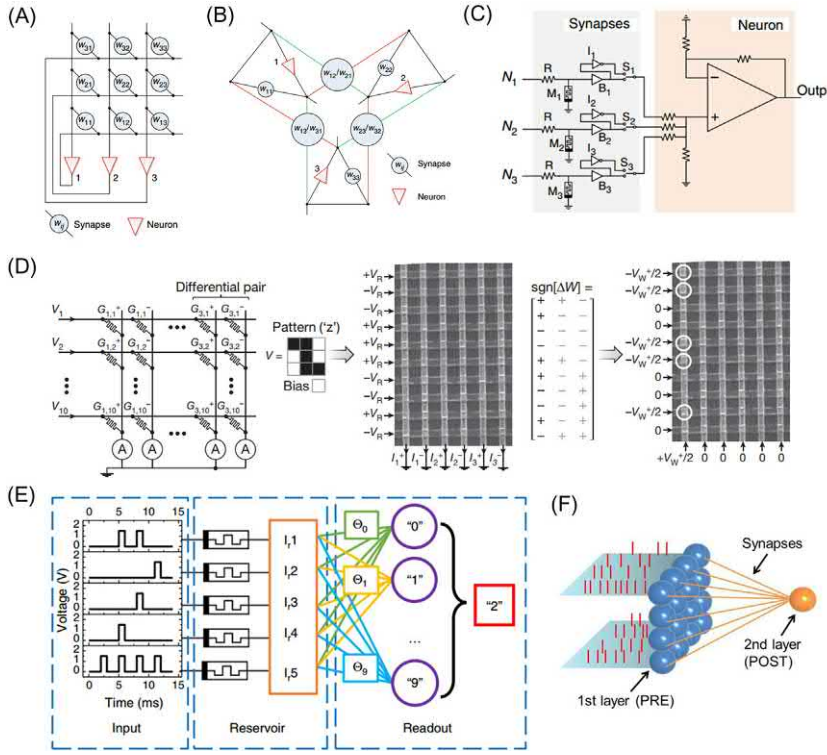


FIGURE 14.4 (A) HNN structure consisting of three neurons. (B) HNN simplified structure due to its symmetric characteristic. (C) Circuit schematic representation of HNN. (D) Pattern classification implementation experiment. Binary image's 9 pixels and bias correspond to 10 inputs. Outputs of network correspond to three types of recognition result. (E) Reservoir example. Binary image is transferred to a temporal series of pulse. (F) NN structure consists of 16 presynaptic neurons and one postsynaptic neuron. Synapses connect two neurons in two layers. *Adapted from M. Prezioso, et al., Training and operation of an integrated neuromorphic network based on metal-oxide memristors. Nature 521 (2015) 61–64 [24]; S.G. Hu, et al., Associative memory realized by a reconfigurable memristive Hopfield neural network. Nat. Commun. 6 (2015) 7522 [44]; C. Du, et al., Reservoir computing using dynamic memristors for temporal information processing. Nat. Commun. 8 (2017) 2204 [45]; G. Pedretti, et al., Memristive neural network for on-line learning and tracking with brain-inspired spike timing dependent plasticity. Sci. Rep. 7 (2017) [46].*

(PCB), including transmission gate chips, operational amplifiers, and comparator chips. The whole circuit is connected with RRAM using external wire. Fig. 14.4C shows a 3-bit neuron circuit schematic representation. In the programming phase, RRAM devices are adjusted to positive or negative values. In the recall phase, three output neurons separately receive signals with synaptic weight and update their states asynchronously. Neuron's output result is fed into input terminal again. Whole process would not stop until

the outputs converge to stable state. This work exhibits that HNN's capable of storage and retrieval of 3-bit data, including single-associative memory and multiassociative memory, and is a successful step toward hardware implementation of associative memory.

14.2.2.2 Pattern recognition

The feasibility of neuromorphic systems for pattern recognition has been discussed in Ref. [47]. CMOS neurons circuit is employed in this work, including pulse generation part, connection, and NN, while RRAM plays the role of a synapse. In this way, neurons and two RRAM devices have been experimentally confirmed. Using the STDP learning rule, two distinct modes including training and test are designed. Further circuit and system-level simulation are also executed. This architecture is capable of realizing digit recognition.

In Ref. [24], integrated RRAM crossbar is used to implement a single-layer perceptron. Transistor-free device architecture makes high density of circuit possible. Recognition of three types of 3×3 pixels binary images has been successfully realized. Each synapse corresponds to two RRAM devices, and weight value is described as following equation: $W_{ij} = G_{ij}^{+} - G_{ij}^{-}$. The entire network consists of 30 synapses, including 10 inputs and 3 outputs and the crossbar performs analog multiplication of matrix and vector. Fig. 14.4D depicts an example of classification operation and weight adjustment. An online trained network is capable of recognize noise patterns.

A larger 128×64 RRAM array is utilized in Ref. [48] to implement NN application. Each device could achieve 6-bit precision. Dot Product Engine (DPE) presented in this work is a platform for reprogrammable analog computing. This array is utilized to implement large single layer of NN. Forward inference on RRAM array successfully realizes MNIST image recognition task, achieving 89.9% recognition accuracy. Because of array architecture, massive parallel VMM is naturally executed in array. It reduces many NN procedures on platform based on traditional von Neumann architecture. Whole NN's speed could be increased, which significantly affect the feasibility of large matrix application.

Most of these works mainly processes task with time-independent input and good retention characteristic of devices is required to achieve high performance. By contrast, in Ref. [45], short-term RRAM array is proposed to implement a dynamic reservoir. WO_x material is used as switching layer in this research. Different from RRAM devices mentioned in previous works, short-term RRAM device's state is dependent on not only programming pulse but also pulse interval. With even small-scale RRAM array, it is capable of solving complex temporal data problem effectively. With only 88 dynamic reservoir and readout function (which is composed of 176×10

RRAM), complex MNIST classification can also be realized. Each row of binary images forms a specific input stream. These are fed into reservoir part. State of every RRAM device is relative to temporal input. Consecutive pixels' information corresponds to unique different conductance states. Conductance of all RRAM in reservoir needs to be collected and sent to readout network. Then, the classification result is a label corresponding to the maximum value of VMM. The whole process is depicted in Fig. 14.4E. This system can also successfully address second-order nonlinearity task, that is, prediction of output without knowing initial time-dependent transfer function. The reservoir is composed of 90 RRAM devices, and the readout network architecture is similar to readout function described above.

Compared with neuromorphic recognition with common supervised learning algorithm, unsupervised learning method is more attractive for some recognition tasks, as it requires no identified label. Unsupervised learning has been successfully demonstrated with STDP and SRDP learning rule. Refs. [49,50] demonstrate STDP characteristic in HfO_2 RRAM device. These fully connected networks consist of 64 presynaptic neurons and one postsynaptic neuron. Unsupervised online learning rule is utilized. Moreover, pattern recognition is executed with a two-layer perceptron network. The whole network is composed of 3×3 RRAM (1T1R)-based synapses. Weight value is trained by STDP rule. Furthermore, a 4-transistor-1-resistor (4T1R) synapse architecture is proposed to implement SRDP characteristic, which can also realize pattern learning. These training and recognition works are a step toward brain-inspired network using unsupervised learning scheme. In addition to works mentioned above, in Ref. [46], static pattern's unsupervised learning and dynamic pattern's adaptation are demonstrated. Two-layer perceptron network is composed of 16 presynaptic neurons and one postsynaptic neuron, as shown in Fig. 14.4F. Connected synapses are trained utilizing STDP learning rule. 1T1R cell with two conductance states plays a role of synapse. Demonstration results illustrates that binary states of synaptic RRAM can sometimes be sufficient for neuromorphic recognition.

In future complicated hardware implementation of NN, analog-to-digital (A/D) and digital-to-analog (D/A) conversions are an inevitable problem. They cost increased energy consumption and chip area. In Ref. [6], a binarized-hidden-layer (BHL) is proposed to address this issue in handwritten digit classification task. This introduced network consists of three perceptron layers, as shown in Fig. 14.5A. BHL architecture means that binarized hidden neuron layer is either 0 or 1, while input and output are analog formats. BHL shows its satisfied accuracy and its potential in future development of NN. With this modified MLP algorithm, an RRAM-based taped-out chip is also mentioned, which is capable of MNIST recognition function. Whole network consists of three perceptron layers, 784, 100, and 10 neurons in each layer. Each synapse is realized by two differential RRAM devices. Weight values can be trained by online learning scheme. Fig. 14.5B depicts

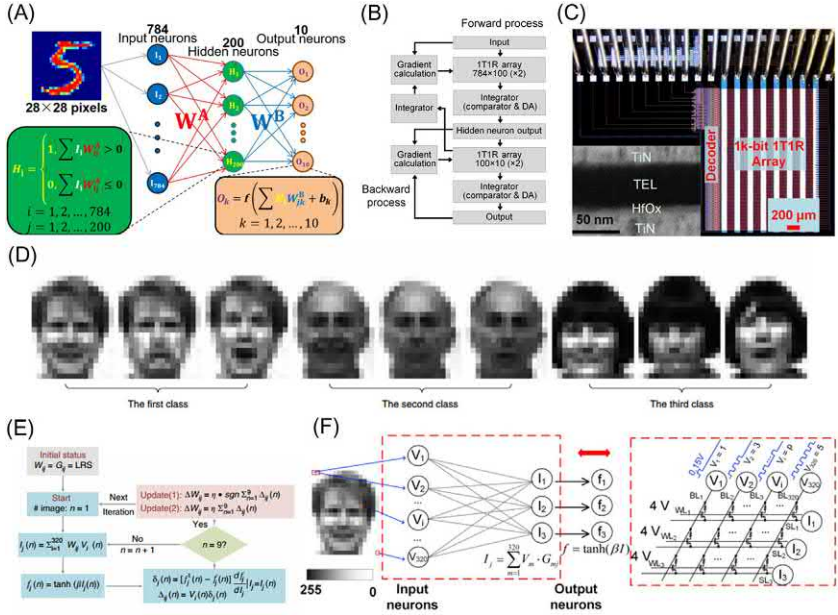


FIGURE 14.5 Pattern and face classification implementation. (A) Modified MLP algorithm. In three-layer perceptron network, the output of hidden neurons is binarized. (B) Architecture of three-layer perceptron chip. (C) Picture of 1k-RRAM array and TEM image of RRAM device. (D) Utilized gray scale face images from Yale Face Database, and they could be divided into three classes. (E) Online training algorithm in face classification experiment. (F) Each gray scale pixel's value in face images ranges from 0 to 255 corresponds to number of voltage pulse. *Adapted from P. Yao, et al. Face classification using electronic synapses. Nat. Commun. 8 (2017) 15199 [5]; H. Wu, et al., 2017 IEEE International Electron Devices Meeting, pp. 11.5.1–11.5.4 [6].*

architecture of this chip. Comparator plays a role of tunable ADC and 1-bit sense amplifier (SA). It could convert current into voltage pulse, which would be transferred to input of next layer.

Among pattern classification application, face classification is also a significant research branch. In Refs. [5,6], face classification is successfully implemented in integrated 1k-cell 1T1R analog RRAM array, as illustrated in Fig. 14.5C. Transistor plays a role of current limiter and operation switch in this architecture. With these 1k arrays, three types of gray scale face images from the Yale Face Database can be recognized, as seen in Fig. 14.5D. The network training algorithm is shown in Fig. 14.5E. Each pixel's value in face images ranges from 0 to 255. Different from previous binary input value, multi-value input corresponds to matching input pulse number during 255 time slices in inference process, as illustrated in Fig. 14.5F. The current flow through each SL is collected. This operation process completes the multi-valued VMM. Single layer perceptron's weight

is trained online in parallel. This training method is effective to address RRAM cells' imperfect and achieve high recognition accuracy. Two types of weight programming scheme are used in online training process, verify (delta rule) and without verify (Manhattan rule). In experiment, network could converge with both these two schemes. The system has high noise tolerance and compared with the same network implemented by Intel Xeon Phi processor with off-chip weight storage costs, 1000 times lower energy consumption. This research further validates the feasibility of energy-efficient neuromorphic computing based on analog RRAM.

14.2.2.3 information processing

Data signal and image process are also an attractive application of NN: spectrum analysis, image edge extraction, image process, and so on. In many real-life situations, many machine learning tasks are processing data and information with no adequate label. PCA is a crucial algorithm in machine learning, which performs functions of feature extraction and dimensionality reduction, and is widely utilized in many applications, such as image analysis, medical disease diagnosis, and disease treatment. PCA hardware implementation has been demonstrated in Ref. [51]. In experiment, 9×2 RRAM array and other peripheral processing circuit are integrated in test board. The RRAM crossbar implements VMM process. The input vectors have nine components; each one corresponds to a measurement standard. Initially, conductance matrix is randomly distributed. After training using online and unsupervised learning method, trained matrix's first and secondary columns, respectively, determine data's primary and secondary principle components. In this way, high-dimensional data can be transferred into low-dimensional data. With linear classification, clustering data could further complete tasks, such as classification and prediction. After clustering, medical data are separated into two types: benign and malignant. Result validates that medical data could be clustered and prediction function could be executed successfully with small nonideal RRAM array.

Besides, sparse coding is also common way to realize data dimensionality reduction in machine learning. In [52], signal and image process are successfully implemented with sparse coding. Previous work's analog vector is realized by transferring amplitude to pulse width [5,51,53]. This method needs more read and operation time. In circuit design, there are additional points to be considered. With 128×64 large-scale RRAM crossbar, analog VMM first executes complete multiplication operation of analog amplitude vector and analog weight matrix. $I-V$ characteristic results for all cells in different conductance states are very linear. This characteristic is very suitable for analog computing. Different amplitude vector corresponds to various amplitudes of operation voltage. Using this scheme, discrete cosine transformation (DCT), which is useful in digital signal and image/video processing, has been

implemented in array. First, signal spectrum analysis result executed by crossbar exhibits high agreement with MATLAB output result. Besides, image information is transformed in signal data. Using same method, two-dimensional (2D) DCT is successfully utilized in image processing. In two compression rates 20:1 and 2:1, image information can both be compressed and reconstructed. Furthermore, convolutional image filtering is realized in crossbar array using 2D DCT. Filtering image is obtained by image vector and convolutional matrix multiplication. Compared with VMM using digital ASIC, since no need to ADCs requirement, crossbar array's energy efficiency is 17 higher.

14.2.2.4 Scaling the demonstrations

To process highly complicated tasks, hardware implementation of very deep NNs is an urgent problem to be addressed. Compared with analog RRAM, binary RRAM in NN could avoid nonlinearity. A complete binary NN (BNN) function is demonstrated in Ref. [8]. In this multilayer perceptron network, neurons and synapses are binarized. Input layer contains 400 neurons and 200 neurons composed of hidden layer. Output layer consists of 10 neurons respectively corresponding to 10 classification categories. The 16-Mb RRAM chip is exploited to demonstrate BNN. This chip is one of the most scaled RRAM arrays, containing totally 16 blocks. Each block is composed of two 512k RRAM array (1T1R). Through experiment, the whole network can realize MNIST classification with back propagation algorithm used offline. Offline training result is successfully programmed into the RRAM array and the recognition accuracy is close to software executed results, even with imperfect yield of RRAM array. Furthermore, online training process is executed using simulation, but results show requirement of a high precision of synapse (multiple bits). Unfortunately, high precision could also decrease energy efficiency. These two factors need to tradeoff in implementation.

Ref. [54] experimentally demonstrates MLP network in two 20×20 RRAM crossbar arrays. These arrays are integrated with CMOS components in two PCB boards. As shown in Fig. 14.6A, three perceptron layers are composed of 16 input neurons, 10 hidden neurons, and 4 output neurons. With additional bias input, 170 and 44 synapses in two connection layers are implemented. Each synaptic weight is comprised of two RRAM devices. The whole network is capable of recognizing 4×4 pixels binary image with both ex situ and in situ training methods. Satisfied classification result is obtained with ex situ method.

Ref. [55] also exhibits successful demonstration of multilayer NN in integrated 128×64 analog RRAM array. High recognition accuracy could be obtained in this network with a certain array defect. Compared with offline training, online training method could successfully compensate for inevitable imperfection of RRAM layer. Fig. 14.6B shows the implementation

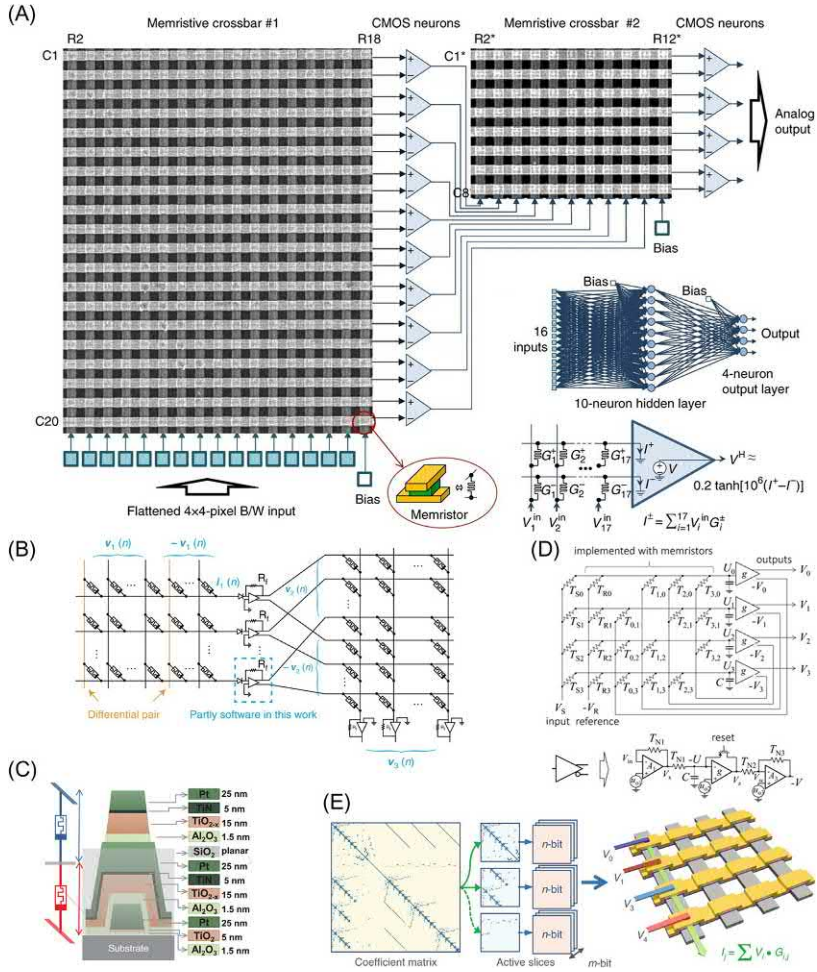


FIGURE 14.6 (A) MLP demonstration architecture in RRAM array; graphical representation of the MLP and perceptron circuit. (B) Implementation architecture based on RRAM array. Difference of two cells on two marking columns represents a complete synapse. (C) Two-layer vertical stacked 3D RRAM crossbar's architecture. RRAM devices in two layers share intermediate layer as electrode. (D) HNN ADC implementation architecture with 4-bit precision. (E) Partial differential equation solving process. Sparse coefficient matrix is divided into small equal slices, which would be mapped into equally sized crossbar. Each bit is represented by multiple RRAM arrays to enhance computing precision. Crossbar executes VMM function. PDE's final solution is calculated by iterative VMM operation. *Adapted from F.M. Bayat, et al., Implementation of multilayer perceptron network with highly uniform passive memristive crossbar circuits. Nat. Commun. 9 (2017) 2331 [54]; C. Li, et al., Efficient and self-adaptive in-situ learning in multilayer memristor neural networks. Nat. Commun. 9 (2018) 2385 [55]; G.C. Adam, et al., 3-D memristor crossbars for analog and neuromorphic computing applications. IEEE Trans. Electron. Devices 64 (2016) 312–318 [56]; X. Guo, et al., Modeling and experimental demonstration of a Hopfield network analog-to-digital converter with hybrid CMOS/memristor circuits. Front. Neurosci. 9 (2015) [57]; D. Tank, J.J. Hopfield, Simple 'neural' optimization networks: an A/D converter, signal decision circuit, and a linear programming circuit. IEEE Trans. Circ. Syst. 33 (1986) 533–541 [58]; M.A. Zidan, et al., A general memristor-based partial differential equation solver. Nat. Electron. 1 (2018) 411–420 [59].*

architecture of multilayer NN. Synaptic strengths in network are trained by stochastic gradient descent learning scheme, including 64×54 synapses in the first layer and 54×10 synapses in the second connection layer. Difference of two RRAM devices represents a weight value. RRAM array completes VMM operation. This experimental result exhibits good recognition accuracy in MNIST database. Furthermore, extended simulation work exhibits the feasibility and higher recognition capability of larger-scale multilayer neuron network.

In addition to existing 2D architecture, RRAM device can be stacked vertically in 3D RRAM integrated with FinFET. This approach increases array density significantly and makes it possible to form dense NN connection. In Ref. [56], two-layer vertically stacked 10×10 RRAM crossbar array is manufactured, which share an intermediate electrode. The entire structure of 3D RRAM crossbar array is illustrated in Fig. 14.6C. RRAM devices in both bottom and top layers show good reliable analog property. Multiple stable and independent conductance states could be programmed. This work paves the way toward brain-inspired computation with analog-integrated 3D RRAM array. Besides, a four-layer 3D vertical RRAM array is fabricated in Ref. [60]. Device model is established with experimental data and utilized in system-level simulation. Result shows feasibility and reliability of this structure for neuromorphic computing. Furthermore, Ref. [61] proposes experimental demonstration of neuromorphic computing accelerator for 3D RRAM application. A monolithically integrated two-layer 3D RRAM crossbar array architecture is exploited to realize DPE function. This structure significantly relieves previous accelerator's limitation of memory bandwidth. Two experiment methods are utilized in this work, computation in the same layer or in the different layers. High computation density's potential is experimental demonstrated. In addition, as for training for vertical stacked RRAM device, variation of device is an inevitable factor. To address this problem, Ref. [62] mentions a new training scheme: to restrain inherent characteristic of RRAM, several RRAM devices compose of a synapse. Final simulation results validate this novel scheme's feasibility in 3D neuromorphic computing.

In addition to above introduced demonstration, HNN is a kind of RNN, which can solve many optimization problems owing to its energy equation. HNN can be applied in many fields, such as image storage and retrieval, controlling, and signal processing. Among them, HNN realizes that ADC function is a rich investigation direction. However, it is a real challenge to complete the function of HNN ADC with the combination of CMOS components and RRAM devices. In Ref. [57], HNN's implementation with RRAM device has validated the feasibility to realize ADC up to 8-bit precision in simulation work. Furthermore, an experiment is demonstrated to implement ADC with 4-bit precision with discrete synaptic RRAM devices and some peripheral inverting amplifiers compose of neurons, as shown in Fig. 14.6D.

In the experiment process, all devices are connected in breadboard with external wires. The relationship between number of weight cells and number of neurons is square. Before connection operation, RRAM device is programmed using ex situ learning method. Afterward, reference weight would be adjusted with in situ to get better experiment result.

Most of the works described earlier is not highly vulnerable to RRAM device imperfection due to their inherently tolerant algorithms. However, high computation precision and accurate result are required in many special applications. In Ref. [59], RRAM-based demonstration with high precision is made. Partial differential equation (PDE) is solved with limited precision and size of RRAM crossbar. PDE is essential in various research works and practical tasks involving enormous operation of vector and matrix. By discretizing variable, partial derivatives in each point are relative to values in neighbor point. Equation solution is obtained by iterative multiplication of input vector and coefficient matrix. Sparse coefficient matrix (many values is zero in matrix) is separated into small sized matrixes. Only active slices correspond to scale-limited crossbar and zero slice matrixes are no need to map in RRAM array. This method would significantly reduce computation resource and power consumption. Besides, to expand computing precision and obtain accurate result, multiple RRAM arrays represent 1 bit together. Fig. 14.6E shows PDE solving process in RRAM crossbar. Examples in this work exhibit high efficiency of this accurate computing system.

14.3 Circuit and system-level implementation

14.3.1 Latest progress on circuit and system based on RRAM for NN processing

RRAM-based nonvolatile computing-in-memory (nvCIM) macros for NN processing can actually be developed based on normal memory macro, as shown in Fig. 14.7A. A CIM macro should be able to support both memory and CIM mode, so it can store and update the weight data in the memory mode and perform NN processing in CIM model. Its block diagram mainly differs from the memory macro by its dual mode WL drivers and the sensing circuits with CIM functions. While the WL driver activates a single WL in the memory mode, it can activate multiple WLs at the same time in NN processing mode. Similarly, the sensing circuits can satisfy not only the requirement of the memory mode to read logic “0” and “1” but also the need for sensing a number of MAC values to output the multi-bits results [63,65]. Fig. 14.7B further shows the detailed information of the MAC operation in the NN mode, where the weight “0” and “1” can be represented by the HRS and LRS states, respectively. Once the input signals “1” (VDD) or “0” (0 V) are applied to the WLs, the multiply results of the inputs and the weights can be given by the cell current and the accumulated multiply results can be read

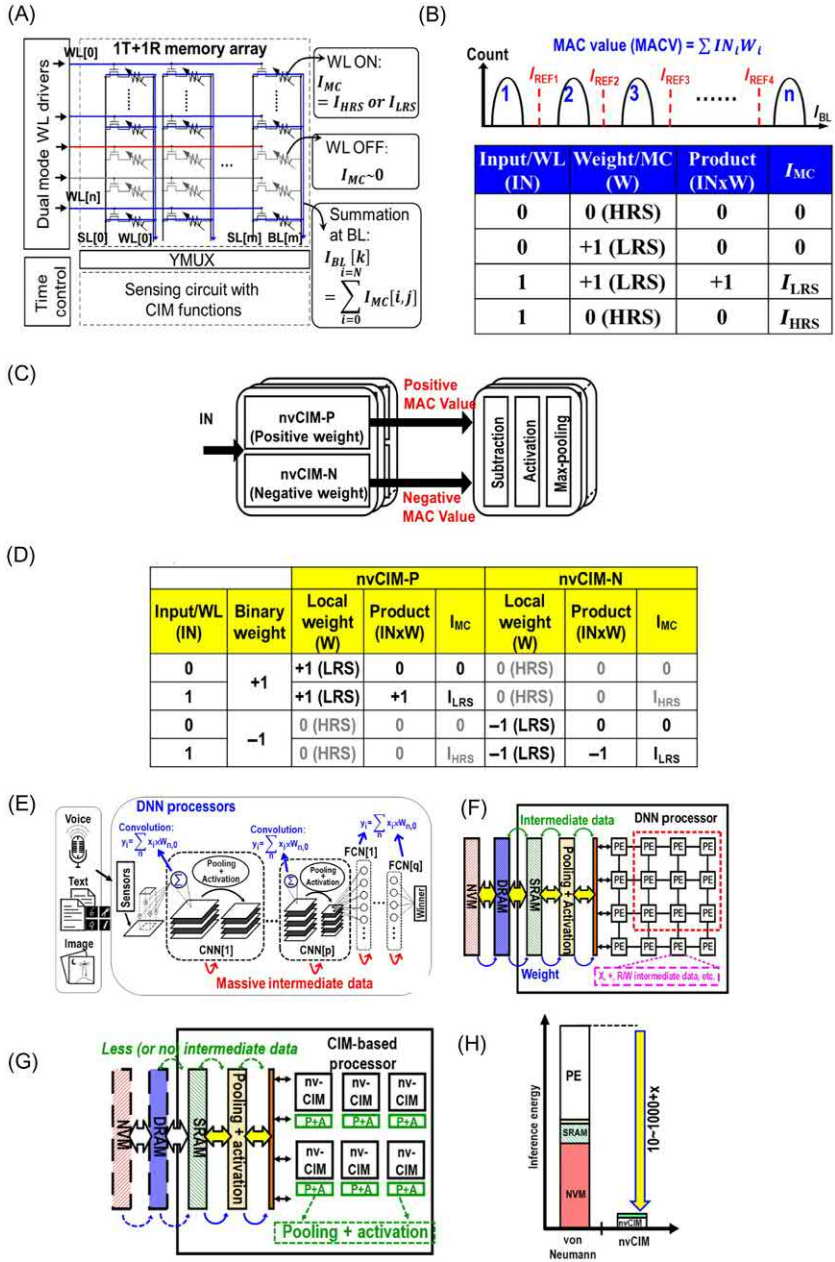


FIGURE 14.7 (A) Block diagram of typical memory macro with computing-in-memory functions and (B) input, weight, and product result in CIM macro. (C) Representation of positive and negative MAC values in different macro and (D) input, weight, and product result in CIM macros with positive/negative weights. (E) DNN processing for a wide range of recognition and classification tasks. DNN processing systems based on digital NN accelerator (F) and nvCIM-based accelerator (G), and (H) comparisons on energy consumption of different types of DNN processors. *Adapted from M.-F. Chang, 2018 IEEE International Solid-State Circuits Conference (ISSCC) Tutorial, 2018 [63]; W.H. Chen, et al., 2018 Solid-State Circuits Conference, pp. 494−496 [64].*

from BL. By comparing the BL signals (voltage or current) with several references, the SA with CIM functions can directly output the MAC values without moving the stored weight information to digital circuits.

Another issue raised from NN processing is that both of positive and negative weights are required in most of algorithms. Because of the mechanism for MAC operations in RRAM-based CIM macro, memory cells in 1T1R configuration cannot contribute negative current to the BL and thus negative weights cannot be represented. This issue can be solved by using different macros to represent positive and negative weights. Fig. 14.7C shows the computation flow, and Fig. 14.7D shows the weight representation in different macros. While the MAC operations can be done in macros stored positive (nvCIM-P) and negative (nvCIM-N) weights, the results can be obtained by performing subtraction, activation, and max-pooling in sequence by additional digital circuits placed near the memory macro. It is also noteworthy that both of algorithms based on binary weights and ternary weights have been successfully demonstrated by this method recently [63].

DNN is considered as one of the indispensable elements that forms the foundation of deep learning. Recent progress has demonstrated its outstanding performance in a broad range of recognition and classification tasks, such as pattern, text, and voice [63]. Fig. 14.7E shows a typical DNN that consists of multiple layers of convolution (CNN) and fully connect NNs (FCNN). Combined with several layers of CNN and FCNN, DNN can extract the features of the inputs with a high-level abstraction and thereby classify the inputs into different catalogs. In a CNN layer, the convolution operations between the inputs and the kernel (or filter) weights are first carried out. The output can be given after pooling and activation. On the other hand, in FCNN layer, the output is given by the weighted sum of the inputs after activation. Although CNN differs from FCNN in many ways, both need massive amounts of MAC operations for convolution and weighted sum.

To satisfy the requirements of high parallelism required to process NN, DNN processors based on the von Neumann architectures usually contain a large array of processing elements (PEs), as shown in Fig. 14.7F. PEs are capable to execute simple operations (e.g., multiplication, summation) and work in parallel. Although this approach eases parallel computation, it is still bottlenecked by the memory accessing because (1) weight data need to be transferred from off-chip NVMs and (2) large amounts of intermediate data generated in DNN processing may need to be repeatedly transferred between the PE arrays and the memory. As a result, the energy efficiency and the performance of DNN processors tend to be limited by the need of frequent fetching data through the memory hierarchy and the heavy cross-hierarchy data transferring.

On the other hand, Fig. 14.7G conceptually shows the structure of DNN processors based on CIM architecture. CIM can enhance DNN processing performance in several aspects as follows. First, all of weight data can be stored in the nonvolatile CIM macro. Hence, no cross-memory-hierarchy

access is required to fetch the weight data. Besides, off-chip NVM is not necessary, thanks to the on-chip NVM. Second, intermediate data that are produced during DNN processing can be effectively reduced. Because different layers of NN can be processed by different CIM macros, the outputs of a given CIM macro can be directly input into its neighboring macros. As a result, the intermediate data do not need to be frequently transferred between logic and memory circuits as that required in the previous architecture, which can result in a 10−1000× energy reduction (Fig. 14.7H). Lastly, CIM macros can execute multiple MAC operations in a signal cycle and therefore fully support the computation with high parallelism.

Despite the great potential of CIM architecture, practical implementation of CIM array based on emerging NVM is still facing several challenges due to the device characteristics of the memory devices. Several issues associated with device-circuit interactions that must be taken into considerations to design CIM macro are discussed as below.

14.3.2 Practical challenges of implementing RRAM macros for DNN processing

14.3.2.1 Sneak current and array architecture

Because multiple WLs are needed to be activated for CIM operations, more leakage paths can exist in CIM mode compared with those in memory mode. Therefore the memory array architectures need to be optimized according to the CIM functions. Fig. 14.8A−D comparatively shows four types of array structure, including crossbar array and 1T1R arrays with different input schemes and different direction of SL. Here, we also assume a case with two inputs those are equal to 1 and 0, respectively. While the supply voltage (VDD) is applied when the input is 1, the input node is grounded (0 V) when input is 0. It can be observed that sneaky current paths exist in the crossbar array and the 1T1R array based on the SL input scheme. Although the sneak current in the crossbar structure can be suppressed by adopting additional diode or selector devices [66], it may require special process. As for 1T1R array, the sneak current path can be effectively eliminated by using the WL input schemes as shown in Fig. 14.8C and D. By applying VDD to SL and controlling the access transistor with the input voltages, MAC operation can be achieved by reading the BL current. It should also note that because the foundry solutions for 1T1R ReRAM array usually have parallel aligned SL and BL, the array structure shown in Fig. 14.8D with WL input scheme is a more practical solution for CIM arrays.

14.3.2.2 The influence of resistances of access device and memory cell

Because a series of data patterns need to be sensed for NN processing, CIM macro has posed new challenges on read circuit designs, which requires high

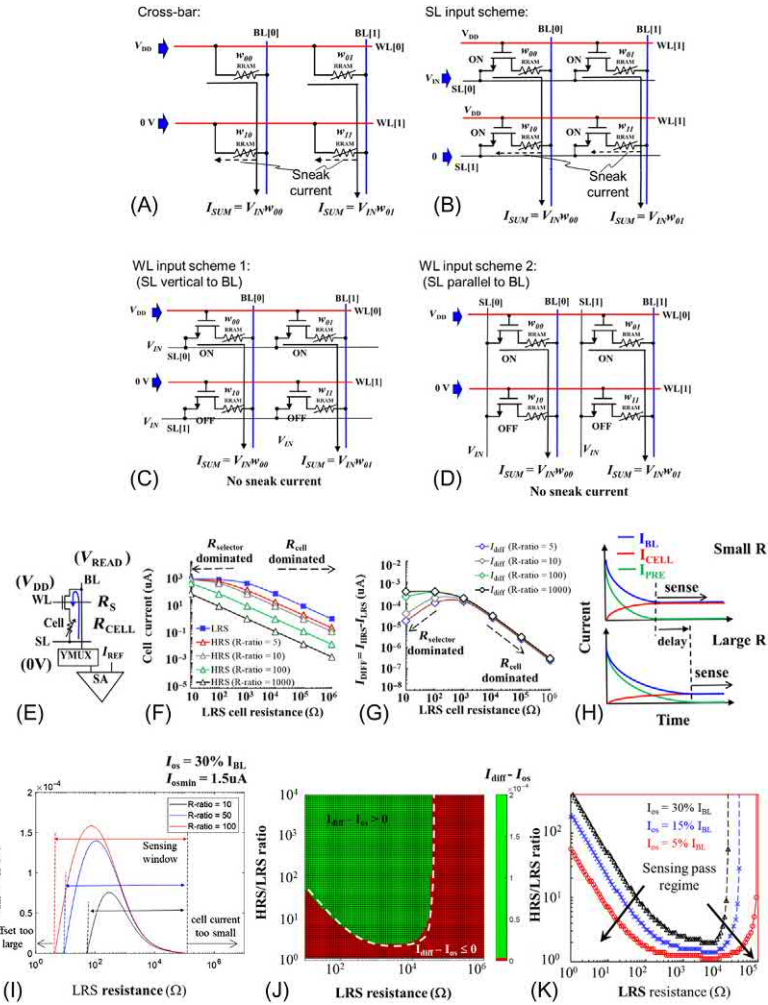


FIGURE 14.8 (A–D) Comparison on different array architectures considering the case with two different inputs. The influence of cell resistance on the sensing circuits: (E) simplified current sensing schemes, (F) cell currents at HRS and LRS of RRAM cells with different resistances and R-ratios, (G) differential mode inputs given by HRS and LRS states, and (H) the influence of cell resistance on read access time. The influence of SA offset on the sensing regime: (I) the dependence of differential mode input current (I_{DIFF}) subtracted by SA offset current (I_{OS}) as functions of LRS resistance and on/off ratio, (J) sensing pass regime of RRAM cells with a typical SA, and (K) sensing pass regime considerably increases with decreasing SA offset current. *Adapted from C.-M. Dou, et al., 2018 Symposium on VLSI Circuits Dig. Tech. Papers, 2018, pp. 171–172 [65].*

sensing yield with small read margins. As a result, it becomes more important to increase the differential mode inputs to SA caused by different data patterns to improve sensing yield. Fig. 14.8E shows simplified current sensing schemes in memory macros. The cell current of 1T1R cell and the ratio between the common-mode (I_{COM}) and differential-mode currents (I_{DIFF}) as a function of LRS resistance (R_{LRS}) in the typical current-mode sensing scheme are shown in Fig. 14.8F and G, respectively. An excessively small R_{LRS} can reduce the ratio I_{DIFF}/I_{COM} required for a good sensing yield, whereas an excessively large R_{LRS} tends to increase the read access time because of reduced cell current (Fig. 14.8H). This means that cell resistance must be optimized taken the trade-off between the read margin and speed into considerations.

14.3.2.3 Influence of SA offset

Except from the magnitude of differential input signal, the sensing yield is also determined by SA offset. In the current-type sensing scheme, the sensing yield can be evaluated by the value of differential mode input current (I_{DIFF}) subtracted by SA offset current (I_{OS}). The correct results can be only readout when $I_{DIFF} - I_{OS} > 0$. In typical SAs, I_{OS} is proportional to I_{DIFF} with a minimum offset current I_{OSMIN}. Because the existence of I_{OS}, there is a maximum sensing regime given by a given SA. Fig. 14.8I shows the $I_{DIFF} - I_{OS}$ as functions of LRS resistance and R-ratio. The requirements on cell resistance and R-ratio given by I_{OS} can be clearly observed. When LRS resistance is too small, I_{OS} is relatively large because of the increased cell currents. Therefore a relatively large R-ratio is also required to increases I_{DIFF}. On the other hand, LRS resistance can also not be too large. If the cell current is smaller than I_{OSMIN}, no reliable readout process can be done. Here, we assume that I_{OS} is about 30% of I_{DIFF} with an $I_{OSMIN} = 1.5$ μA, which is a typical value of a conventional latch-type current SA [64]. Fig. 14.8J further shows the evaluated maximum sensing regime of a given SA. The green region outlines the range of cell resistance and R-ratio that can be sensed. Fig. 14.8K reveals the maximum sensing regime as a function of SA offset. It can be clearly showing that maximum sensing regime increases with decreased SA offset. With decreased SA offset, it is possible to detect smaller change of the differential mode inputs.

14.3.2.4 Read margin degradation with increasing number of activated WLs

Fig. 14.9A shows the voltage-model sensing scheme in a 32×32 CIM macro, which use a binary local weight. The array can execute MAC operations with 32 inputs and give a 3-bit output at each BL. The BL voltage of a given MAC value (MACV) is determined by the voltage division between the parallel-connected RRAM cells with activated WLs and that of the

clamping transistor. Eight MAC values range from 4 to 32 with a step of 4 are divided. The simulated BL voltage distributions of different MACVs as a function of activated WLs (N_{WL}) are shown in Fig. 14.9B. The read margin of a given MAC value (V_{RM}) can be defined as its voltage different to the minimum BL voltage of its neighboring MACV. Fig. 14.9C further shows the minimum V_{RM} ($V_{RM,MIN}$) of different MACVs as a function of N_{WL}. It clearly reveals that the read marge of a given MAC value degrades as the number of activated WL increases. This originates from the leakage current from the HRS cells with activated WLs, as illustrated by Fig. 14.9D. Although the multiply results of these HRS cells represent 0, they still contribute leakage currents to BL and thus results in the variations of BL voltage of a given MACV.

14.3.3 Advanced design techniques for performance and reliability enhancement

In this section, by reviewing state-of-the-art RRAM CIM macros, we will introduce advanced design techniques for implementing RRAM CIM macros with high performance and reliability for NN processing. In the rest of this section, RRAM macros with highly energy-efficient architecture for low power applications, self-write termination (SWT) schemes to deal with cell-to-cell variations, and input-aware generation scheme to solve the small read margins challenges will be introduced in sequence.

Fig. 14.9E and F show the die-photo and key chip information of the fabricated 150 nm nonvolatile intelligent processor (nv-Processor) having superior energy efficiency [66]. It consists of CPU, NN processing units based on HfOx RRAM CIM macros, and related interface and controlling modules. Nonvolatile flip-flops are adopted to backup and restore the key data the states in the processors, so it can exhibit high energy-efficiency during frequent power-interrupted scenarios for the internet-of-thing (IoT) edge applications. The structure of NN processing unit is shown in Fig. 14.9G, which uses separated macros to store positive and negative weights. The high energy efficiency of the NN processing roots from several key features discussed as below. First, 1T1R RRAM array with WL input schemes can effectively reduce the sneak current paths as discussed in the previous section. Second, the NN processing unit is used to perform binary weight NN. It not only improves the reliability considering relatively small on/off ratio of RRAM cells but also removes the energy/area overhead required for multi-level cell (MLC) operations. Lastly, thanks to the engineered sensing amplifiers with CIM functions, the energy efficiency and area efficiency are significantly improved because no A/D and D/A converters, as shown in Fig. 14.9H. Because of these features discussed above, this architecture can deliver high energy efficiency and thus offer a competent solution of IoT edge inference.

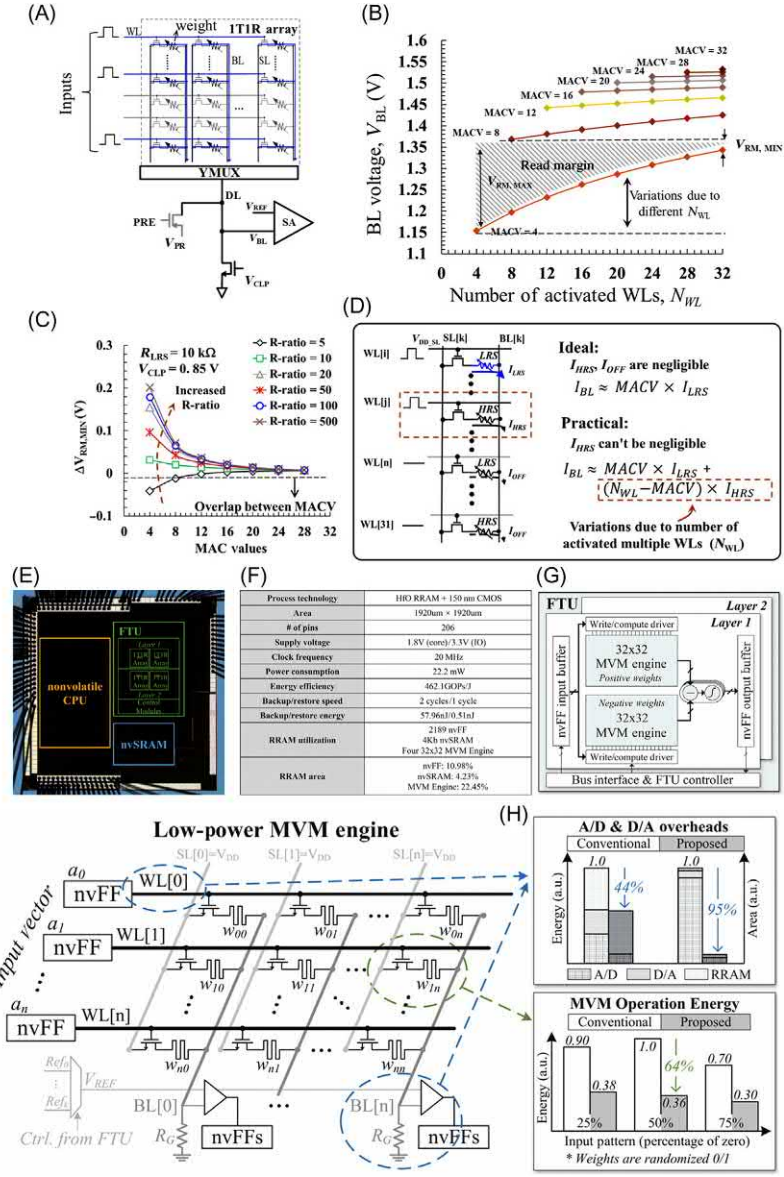


FIGURE 14.9 (A−D) Sensing margin versus number of activated WLs. (A) Voltage-mode sensing scheme. (B) Simulated BL voltage distribution of different multiply and accumulation values as a function of activated WLs (NWLs). (C) The minimum read margin ($V_{RM,MIN}$) of different MACVs as a function of R-ratio. (D) The mechanism of read margin degradation due to HRS leakage. nonvolatile intelligent processor (nv-Processor) with RRAM-based NN processing unit: (E) die-photo and (F) key chip information of fabricated, (G) structure of the NN processing units based on RRAM CIM macros, and (H) key features and advantages of the fabricated RRAM CIM macros. *Adapted from C.-M. Dou, et al., 2018 Symposium on VLSI Circuits Dig. Tech. Papers, 2018, pp. 171−172 [65]; F. Su, et al., 2017 Symposium on VLSI Circuits Dig. Tech. Papers, pp. T260−T261 [66].*

As previously discussed, one of the critical challenges for implementing RRAM CIM macro comes from the large cell-to-cell characteristics variations of RRAM devices, which leads to significant degradation on CIM functions yields. It has been demonstrated that this problem can be effectively mitigated by using SWT scheme [67]. Fig. 14.10A shows the die-photo and key chip information of fabrication 150 nm 16 M RRAM CIM macro for logic operation with write termination schemes. Fig. 14.10B shows the logic values representations in the CIM mode and the positions of reference levels for different logic operations. It has similar operational method compared with that used for NN processing. By activating two WLs and selecting corresponding reference levels, the computing results of the logic operations of stored data can be obtained by comparing the BL and reference signals. Fig. 14.10D and E show circuit schematics and operational waveforms for set and reset termination process in cases of slow and fast bits. In set process, when the resistance of RRAM cells becomes smaller the target LRS value, the SETEND switches to high and turns off P2 to disconnect DL from VSET, resulting in a drop in DL voltage and termination of the SET operation. In the reset process, when the cell resistance exceeds the target HRS value, the RSTEND switches to low and turns off N1 to disconnect DL from VRESET (controlled by N2), resulting in a rise in DL voltage and termination of the RESET operation. Fig. 14.10C describes the BL current distributions of different logic states with and without SWT schemes. It can be clearly observed that without the SWT scheme, considerable overlaps between different logic states exist and directly results in read errors. On the other hand, the SWT scheme can effectively remove these overlaps and thus enable high reliable computations.

To tackle with the small read margin due to the limited R-ratio and the HRS cell leakages when multiple WLs are activated as discussed in Section 3.2, the input-aware referent generation (IA-REF) scheme has been proposed particular to solve this issue [64]. Fig. 14.10F shows the die-photo and key chip information of the fabricated 65 nm 1 M RRAM CIM macro. It supports dual mode (memory and NN modes) operations with the abilities to carry out both of fully connected (FCN) and CNN. Besides, this macro has highlights on its large capacity and fast access time (Fig. 14.10G), which is of importance to accommodate and process modern neural network with ever-increasing model size. Fig. 14.10H further reveals the challenges caused by the HRS cell leakage. Because BL current distribution of different MACVs depends on the numbers of activated WLs, it is difficult to use fixed reference levels to distinguish difference MACVs with varying number of activated WLs (NWLs). For instance, if we used fixed reference levels as shown by the dash line, the readout results at NWL = 7 will be wrong, although those at NWL = 2 are right. Fig. 14.10I shows the schematics of the IA-REF scheme, which comprising of a reference-WL controller (RWLC), input-counter (ICN), and input-aware replica rows (IA-RR).

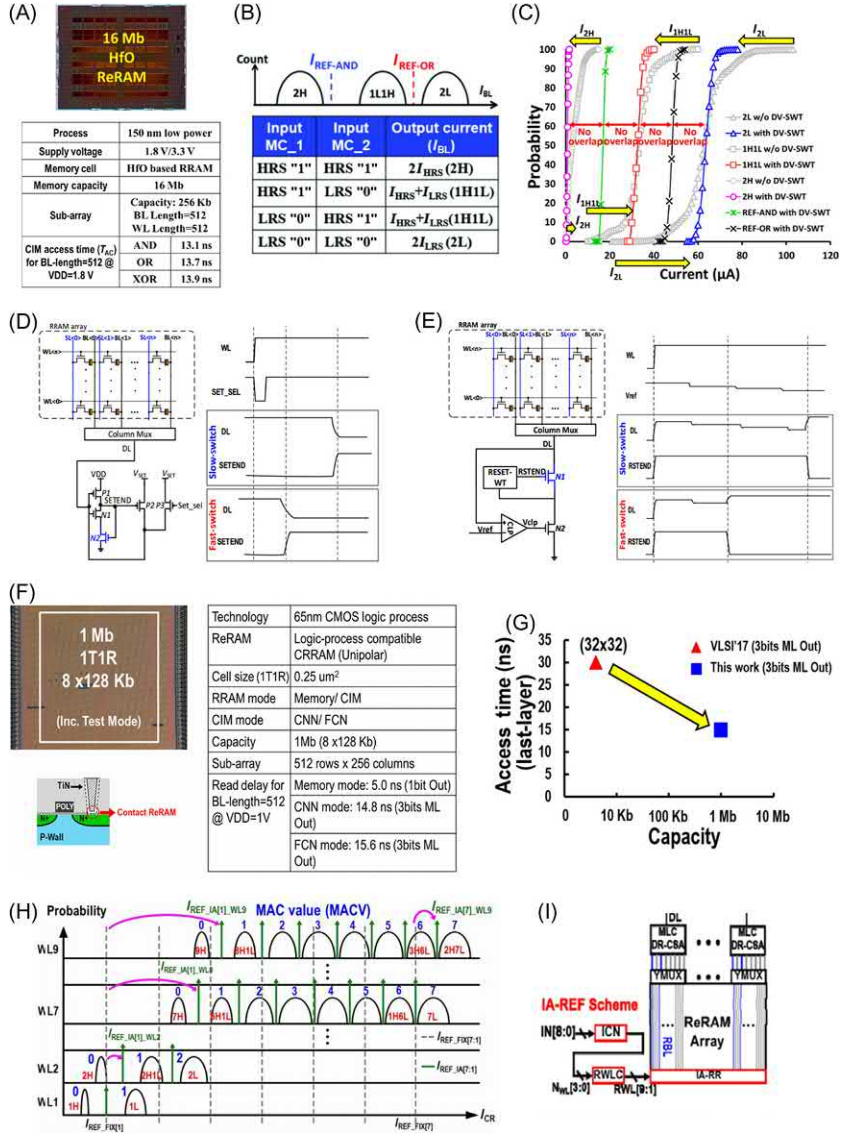


FIGURE 14.10 CIM macro with SWT schemes: (A) die-photo and key chip information of fabrication 150 nm 16 M RRAM CIM macro for logic operation with write termination schemes, (B) logic values representations in the CIM mode and the positions of reference levels for different logic operations, (C) BL current distributions of different logic states with and without SWT schemes, (D and E) circuit schematics and operational waveforms for set and reset termination process. CIM macro with input-aware reference generation (IA-REF) scheme: (F) die-photo and key chip information of the fabricated 65 nm 1 M RRAM CIM macro with IA-REF scheme. The inset conceptually shows the structure of contact RRAM cell used in this work, (G) benchmark on access time and capacity of this work, (H) BL current distribution of different MACVs as a function of different numbers of activated WLs, and (I) the schematics of the IA-REF scheme. *Adapted from W.H. Chen, et al., 2018 Solid-State Circuits Conference, pp. 494–496 [64]; W.H. Chen, et al., 2017 IEEE International Electron Devices Meeting, 2017, pp. 28.2.1–28.2.4 [67].*

In a typical inference process, the ICN counts the number inputs equaling 1, which is NWL, in the input pattern, and outputs NWL to RWLC. Then, RWLC turns on NWL replica WLs (RWLs), from RWL[1] to RWL[NWL], which can generate corresponding reference levels by using IA-RR. As a result, the reference level can be adaptively adjusted based on the inputs. Considering the limited R-ratio and nonignorable HRS leakage currents, this technique can play as a key enabler to increase the number of maximum inputs of CIM macro in a single clock cycle and thus further accelerate the DNN processing.

14.4 Summary

In the coming information explosion era, traditional computing method cannot meet the requirement of Big Data. Brain-like computing methods are potential candidates for providing a solution and an emerging research highlight. In recent years, owing to the fine plasticity of RRAM devices, they have emerged as promising electronic synapses for the realization of hardware NN. This chapter briefly introduced the required characteristic of RRAM device in NN hardware implementation. In addition, we have introduced previous demonstrations of neuromorphic computing, including simulation works and experimental implementations. The demonstrated functions range from classification to information processing. This chapter also mentioned the significant challenges in circuit implementation of NN processing. To address these issues, the chapter introduced several advances in circuits. These works pave the way further towards portable NN applications, which could take place in every aspect of our future lives.

References

[1] S.B. Eryilmaz, et al., Brain-like associative learning using a nanoscale non-volatile phase change synaptic device array, Front. Neurosci. 8 (2014). 205–205.
[2] M. Verleysen, B. Sirletti, A. Vandemeulebroecke, P.G.A. Jespers, A high-storage capacity content-addressable memory and its learning algorithm, IEEE Trans. Circ. Syst. 36 (1989) 762–766.
[3] Y. Zhang, et al., 2014 IEEE International Conference on Nanotechnology, pp. 233–236.
[4] M. Suri, O. Bichler, D. Querlioz, G. Palma, 2012 IEEE international Electron Devices Meeting, pp. 10.3.1–10.3.4.
[5] P. Yao, et al., Face classification using electronic synapses, Nat. Commun. 8 (2017) 15199.
[6] H. Wu, et al., 2017 IEEE International Electron Devices Meeting, pp. 11.5.1–11.5.4.
[7] H. Liu, et al., Uniformity improvement in 1T1R RRAM with gate voltage ramp programming, IEEE Electron. Device Lett. 35 (2014) 1224–1226.
[8] S. Yu, et al., 2017 IEEE international Electron Devices Meeting, pp. 16.2.1–16.2.4.
[9] C. Liu, et al., 2017 International Symposium on VlSI Technology, Systems and Application, pp. 1–2.

[10] B. Long, Y. Li, R. Jha, Switching characteristics of {Ru/HfO}_{2}{/TiO}_{2-x}{/Ru} RRAM devices for digital and analog nonvolatile memory applications, Electron. Device Lett. IEEE 33 (2012) 706–708.
[11] S. Yu, Y. Wu, R. Jeyasingh, D. Kuzum, H.S.P. Wong, An electronic synapse device based on metal oxide resistive switching memory for neuromorphic computation, IEEE Trans. Electron. Devices 58 (2011) 2729–2737.
[12] Y. Wu, S. Yu, H.S.P. Wong, Y.S. Chen, 2015 IEEE International Memory Workshop, pp. 1–4.
[13] F. Bedeschi, R. Fackenthal, C. Resta, E.M. Donze, A bipolar-selected phase change memory featuring multi-level cell storage, IEEE J. Solid State Circ. 44 (2009) 217–227.
[14] B. Li, et al., RRAM-based analog approximate computing, IEEE Trans. Comput. Des. Integr. Circ. Syst. 34 (2015) 1905–1917.
[15] S.H. Jo, et al., Nanoscale memristor device as synapse in neuromorphic systems, Nano Lett. 10 (2010) 1297–1301.
[16] S. Yu, Y. Wu, Y. Chai, J. Provine, 2011 International Symposium on Vlsi Technology, Systems and Applications, pp. 1–2.
[17] H.Y. Lee, et al., 2008 IEEE International Electron Devices Meeting, pp. 1–4.
[18] C.C. Chang, et al., 2017 IEEE International Electron Devices Meeting, pp. 11.6.1–11.6.4.
[19] D. Ielmini, Brain-inspired computing with resistive switching memory (RRAM): devices, synapses and neural networks, Microelectronic Eng. (2018).
[20] S. Mandal, A. Elamin, K. Alexander, B. Rajendran, R. Jha, Novel synaptic memory device for neuromorphic computing, Sci. Rep. 4 (2014) 5333.
[21] D. Kuzum, R.G.D. Jeyasingh, B. Lee, H.S.P. Wong, Nanoelectronic programmable synapses based on phase change materials for brain-inspired computing, Nano Lett. 12 (2011) 2179–2186.
[22] S. Yu, et al., A low energy oxide-based electronic synaptic device for neuromorphic visual systems with tolerance to device variation, Adv. Mater. 25 (2013) 1774–1779.
[23] P.Y. Chen, S. Yu, Compact modeling of RRAM devices and its applications in 1T1R and 1S1R array design, IEEE Trans. Electron. Devices 62 (2015) 4022–4028.
[24] M. Prezioso, et al., Training and operation of an integrated neuromorphic network based on metal-oxide memristors, Nature 521 (2015) 61–64.
[25] L. Gao, F. Alibart, D.B. Strukov, 2015 Ieee/ifip International Conference on Vlsi and System-On-Chip, pp. 88–93.
[26] Q. Xia, et al., Memristor-CMOS hybrid integrated circuits for reconfigurable logic, Nano Lett. 9 (2009) 3640.
[27] X. Liu, et al., 2015 Design Automation Conference, pp. 1–6.
[28] J. Sandrini, et al., Co-design of ReRAM passive crossbar arrays integrated in 180 nm CMOS technology, IEEE J. Emerg. Sel. Top. Circ. Syst. 6 (2016) 339–351.
[29] Y. Jiang, J. Kang, X. Wang, RRAM-based parallel computing architecture using k-nearest neighbor classification for pattern recognition, Sci. Rep. 7 (2017) 45233.
[30] Z. Chen, B. Gao, Z. Zhou, P. Huang, 2015 IEEE International Electron Devices Meeting, pp. 17.7.1–17.7.4.
[31] M. Hu, et al., Memristor crossbar-based neuromorphic computing system: a case study, IEEE Trans. Neural Netw. Learn. Syst. 25 (2014) 1864–1878.
[32] T.C. Jackson, A.A. Sharma, J.A. Bain, J.A. Weldon, L. Pileggi, Oscillatory neural networks based on TMO nano-oscillators and multi-level RRAM cells, IEEE J. Emerg. Sel. Top. Circ. Syst. 5 (2015) 230–241.

[33] D. Garbin, O. Bichler, E. Vianello, Q. Rafhay, 2014 International Electron Devices Meeting, pp. 28.4.1–28.4.4.
[34] J.A. Anderson, J.W. Silverstein, S.A. Ritz, R.S. Jones, Distinctive features, categorical perception, and probability learning: some applications of a neural model, Psychol. Rev. 84 (1977) 413–451.
[35] V. Milo, E. Chicca, D. Ielmini, 2018 IEEE International Symposium on Circuits and Systems, pp. 1–5.
[36] V. Milo, D. Ielmini, E. Chicca, 2017 IEEE International Electron Devices Meeting, pp. 11.2.1–11.2.4.
[37] T. Gokmen, M. Onen, W. Haensch, Training deep convolutional neural networks with resistive cross-point devices, Front. Neurosci. 11 (2017).
[38] L. Song, X. Qian, H. Li, Y. Chen, 2017 IEEE International Symposium on High PERFORMANCE Computer Architecture, pp, 541–552.
[39] A. Shafiee, et al., ISAAC: a convolutional neural network accelerator with in-situ analog arithmetic in crossbars, ACM Sigarch Computer Architecture N. 44 (2016) 14–26.
[40] B. Li, P. Gu, Y. Wang, H. Yang, Exploring the precision limitation for RRAM-based analog approximate computing, IEEE Des. Test. 33 (2016) 51–58.
[41] S. Park, A. Sheri, J. Kim, J. Noh, 2013 IEEE International Electron Devices Meeting, pp. 25.6.1–25.6.4.
[42] I.P. Pavlov, Conditioned reflexes: an investigation of the physiological activity of the cerebral cortex, Ann. Neurosci. 8 (2010) 136–141.
[43] Y. Pershin, M.D. Ventra, Experimental demonstration of associative memory with memristive neural networks, Neural Netw. Off. J. Int. Neural Netw. Soc. 23 (2010) 881.
[44] S.G. Hu, et al., Associative memory realized by a reconfigurable memristive Hopfield neural network, Nat. Commun. 6 (2015) 7522.
[45] C. Du, et al., Reservoir computing using dynamic memristors for temporal information processing, Nat. Commun. 8 (2017) 2204.
[46] G. Pedretti, et al., Memristive neural network for on-line learning and tracking with brain-inspired spike timing dependent plasticity, Sci. Rep. 7 (2017).
[47] S. Park, H. Kim, M. Choo, J. Noh, 2012 IEEE International Electron Devices Meeting, pp, 10.2.1–10.2.4.
[48] M. Hu, et al., Memristor-based analog computation and neural network classification with a dot product engine, Adv. Mater. 30 (2018) 1705914.
[49] V. Milo, et al., 2017 IEEE International Electron Devices Meeting, pp. 16.8.1–16.8.4.
[50] S. Ambrogio, et al., Neuromorphic learning and recognition with one-transistor-one-resistor synapses and bistable metal oxide RRAM, IEEE Trans. Electron. Devices 63 (2016) 1508–1515.
[51] S. Choi, J.H. Shin, J. Lee, P. Sheridan, W.D. Lu, Experimental demonstration of feature extraction and dimensionality reduction using memristor networks, Nano Lett. 17 (2017) 3113–3118.
[52] C. Li, et al., Analogue signal and image processing with large memristor crossbars, Nat. Electron. 1 (2018) 52–59.
[53] P.M. Sheridan, et al., Sparse coding with memristor networks, Nat. Nanotechnol. 12 (2017) 784–789.
[54] F.M. Bayat, et al., Implementation of multilayer perceptron network with highly uniform passive memristive crossbar circuits, Nat. Commun. 9 (2017) 2331.
[55] C. Li, et al., Efficient and self-adaptive in-situ learning in multilayer memristor neural networks, Nat. Commun. 9 (2018) 2385.

[56] G.C. Adam, et al., 3-D memristor crossbars for analog and neuromorphic computing applications, IEEE Trans. Electron. Devices 64 (2016) 312–318.
[57] X. Guo, et al., Modeling and experimental demonstration of a Hopfield network analog-to-digital converter with hybrid CMOS/memristor circuits, Front. Neurosci. 9 (2015).
[58] D. Tank, J.J. Hopfield, Simple 'neural' optimization networks: an A/D converter, signal decision circuit, and a linear programming circuit, IEEE Trans. Circ. Syst. 33 (1986) 533–541.
[59] M.A. Zidan, et al., A general memristor-based partial differential equation solver, Nat. Electron. 1 (2018) 411–420.
[60] H. Li, et al., 2016 IEEE Symposium on VLSI Technology, pp. 1–2.
[61] M.A. Lastras-Montaño, B. Chakrabarti, D.B. Strukov, K.T. Cheng, 2017 Design, Automation & Test in Europe Conference & Exhibition, pp. 1257–1260.
[62] B. Gao, et al., Ultra-low-energy three-dimensional oxide-based electronic synapses for implementation of robust high-accuracy neuromorphic computation systems, ACS Nano 8 (2014) 6998.
[63] M.-F. Chang, 2018 IEEE International Solid-State Circuits Conference (ISSCC) Tutorial, 2018.
[64] W.H. Chen, et al., 2018 Solid-State Circuits Conference, pp. 494–496.
[65] C.-M. Dou, et al., 2018 Symposium on VLSI Circuits Dig. Tech. Papers, 2018, pp. 171–172.
[66] F. Su, et al., 2017 Symposium on VLSI Circuits Dig. Tech. Papers, pp. T260–T261.
[67] W.H. Chen, et al., 2017 IEEE International Electron Devices Meeting, 2017, pp. 28.2.1–28.2.4.

Part IV

Spiking neural networks

Chapter 15

Memristive devices for spiking neural networks

Bipin Rajendran[1], Damien Querlioz[2], Sabina Spiga[3] and Abu Sebastian[4]

[1]Department of Engineering, King's College London, London, United Kingdom, [2]Centre for Nanoscience and Nanotechnology, Universite Paris-Saclay, Palaiseau, France, [3]CNR-IMM, Agrate Brianza, Italy, [4]IBM Research – Zurich, Rüschlikon, Switzerland

15.1 Introduction

Deep learning algorithms have demonstrated unprecedented successes in a wide variety of cognitive tasks such as computer vision, natural language processing, and control tasks based on the same underlying network structure [1]. These networks are loosely inspired by the fundamental architecture of the brain, which comprises a large interconnected mesh of neurons that communicate with each other through synapses. It is believed that learning and memory in the brain are made possible by the neuronal activity-dependent modifications in the effective conductivity of synaptic junctions [2]. On the other hand artificial deep learning networks rely on gradient descent-based methods to adjust synaptic weight parameters to optimize objective functions for learning. Implementing these learning algorithms in today's von Neumann computing platforms is very costly (in terms of both energy and time) due to two fundamental reasons.

- Information in these networks is encoded and transmitted using real numbers in floating-point (or other reduced precision) formats and the operations for parameter optimization involve large numbers of matrix–vector multiplications involving such numbers.
- Owing to the large number of neuronal and synaptic parameters that are involved in state-of-the-art networks, the computational performance is also limited by the constant shuffling of date between the physically separated processor and data storage units.

These factors have prompted research efforts to develop brain-inspired learning algorithms that encode and process information using the time of

Memristive Devices for Brain-Inspired Computing. DOI: https://doi.org/10.1016/B978-0-08-102782-0.00015-0

arrival of binary events (aka "spikes") [3] and their implementation and acceleration on dedicated hardware platforms [4], based on in-memory computing architectures [5].

15.2 Signal encoding and processing with spikes

Second-generation artificial neural networks (ANNs) use real numbers to encode information (Fig. 15.1). Information from neurons in a layer propagate to the next layer through synaptic connections—the output of a neuron, y_j in a layer is determined based on the weighted sum of the inputs x_i of all the neurons in the previous layer through their respective synaptic weights w_{ij}, as

$$I_j = \left(\sum_i w_{ij} x_i\right) \quad y_j = f(I_j) \tag{15.1}$$

Here, f is a non-linear transformation function, such as sigmoid, ReLU (rectified linear unit), etc.

On the other hand sensory organs as well as neurons in the brain encode information in an event-driven manner using electrical signals called action potentials or spikes. Spikes can be thought of as binary signals, whose time or rate of arrival is used to encode information; their amplitude or shape on the other hand does not convey any information. The dynamics of a simple leaky integrate-and-fire (LIF) spiking neuron can be expressed as follows.

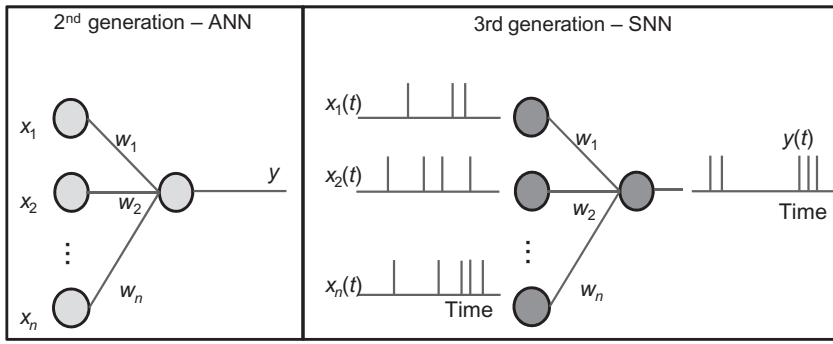


FIGURE 15.1 Second-generation artificial neural networks (ANNs) use real numbers to encode information, while third-generation spiking neural networks (SNNs) use time of arrival of binary signals (spikes) to encode information. Note that neurons and synapses in SNNs have time-dependent transfer functions.

$$I_j(t) = \left(\sum_i w_{ij} x_i(t) \right)^* \alpha(t)$$
$$v_j(t) = I_j(t)^* \exp(-t/\tau) \quad y_j(t) = 1 \quad \text{if } v_j(t) > \theta, 0 \text{ otherwise.} \tag{15.2}$$

Here * is the convolution operator, $\alpha(t)$ is a synaptic current kernel, τ is the neuronal integration time constant, and θ is a threshold for spike detection. The input $x_k(t)$ is a sum of delta functions; if the i^{th} spike arrive at time t_k^i in the k^{th} synapse, $x_k(t) = \sum_i \delta(t - t_k^i)$. Thus SNNs naturally incorporate the time-based information encoding and processing aspects of the human brain. While the LIF model is often used for computing purposes, neurons in the brain exhibit many complex spiking behaviors [6].

A wide variety of signal encoding schemes have been observed in nature [7], some of which are described below. In the rate based coding scheme the number of spikes arriving or issued in a specific time interval encodes the information. For instance the arrival rate of spikes may be proportional to the intensity of a real-world signal. In a latency based code the precise time of arrival may be used to encode information, for instance, the latency may be proportional to the signal intensity. Rhythmic oscillatory patterns of spiking activity are observed in the brain and phase codes use the relative phase of a neuronal spike with respect to these background oscillations to encode information.

Irrespective of the coding mechanism employed, since information is conveyed by the time of arrival and not the amplitude or shape, the encoding mechanism is essentially binary and information transmission through synapses is triggered by the arrival of spikes. If this principle is employed in designing artificial networks, then the standard synaptic communication operation, which involves determining the weighted sum of the neuronal inputs, can be computed without multiplication operations as $x_i(t)$ is 1 only if a spike arrived at time t, and is 0 otherwise. This is markedly different from the standard second-generation networks which necessarily require the use of real numbers in floating-point or reduced precision formats to achieve high accuracy in learning tasks.

In addition to the above computational advantage, there is also a potential to improve the overall efficiency further if real-world signals are encoded using spikes in a *sparse* manner. Consider two adjacent layers of a fully connected network with N neurons. In a second-generation implementation of this network, forward transmission requires N^2 multiplication operations per input. In an equivalent spiking network implementation, if the same input is represented using k samples in the time domain, and if the average probability of an input spike at any sample point in time is p, the forward communication will require pkN^2 addition operations. If the computational cost for multiplication and addition are represented as C_m and C_a, then as long as

$$pkC_a < C_m, \tag{15.3}$$

there will be an overall computational efficiency improvement due to the spike-based signal encoding and transmission strategy, provided all the neuronal and synaptic data are available at the processor when required. Hence, p and k should be chosen as small as possible ("sparse encoding") to obtain the maximum benefit from spike-based networks. Note that in most modern processors, $C_m/C_a > 3$ [8].

15.3 System architecture

Another inherent advantage of spiking networks comes from the fact that their hardware implementations lend naturally to non-von Neumann architectures. Most modern processors have physically separated memory and processor units, and overall performance is limited by the energy and latency costs associated with shuttling data back and forth through a serial data transfer bus, the von Neumann bottleneck. This is especially true for the implementation of second-generation neural networks whose operations involve large matrix multiplications.

To overcome this fundamental limitation for implementing artificial neural networks in dedicated hardware, tiled arrays of cross-bars have been proposed, with each cross-bar array closely integrating the logic and memory components of the networks to be implemented. The neuronal logic, as well as learning circuitry, is implemented in the periphery, while the synaptic weights are stored in the conductance of the memory devices in the cross-bar. Depending on the size of the memory cell, IR drop on the wires, and size of peripheral circuits, most cross-bars are configured to have less than 2048×2048 memory devices.

In a cross-bar architecture, the computations within the core can be conducted in analog or digital mode, based on the nature of the memory devices, and the configuration of the peripheral circuits. However the communication between the tiles has to be digital, as signals have to be transported over long distances between the tiles when realizing large networks. In the case of conventional neural networks, this means that real-valued multi-bit neuronal variables have to be transported between cores from each neuron to its target for every input, which can be a costly operation. On the other hand for spiking networks, only address packets corresponding to sparse binary spikes need to be transported [9], potentially enabling significant performance benefits.

15.4 Memristive devices for Spiking neural networks

Several CMOS-based designs have been demonstrated that leverages these aspects to implement spiking networks in an efficient manner [10]. Some

notable examples include the TrueNorth chip from IBM [11], Loihi from Intel [12], NeuroGrid from Stanford [13], DYNAP from INI Zurich [14], and ODIN from Catholic University Louvain [15]. While these implementations have demonstrated the potential for significant energy and performance benefits, they still lag significantly behind the performance of the human brain.

Nanoscale memristive devices can play a key role in addressing this challenge [16]. Lever-aging the inherent properties of engineered materials at the nanoscale, several novel device structures have been proposed to natively implement the computations necessary for neuronal integration, synaptic communication, and plasticity, employing voltages and currents much lower than that used in conventional CMOS devices. However several challenges have to be overcome to enable the utilization of these devices to realize large, reliable, and efficient neuromorphic learning systems.

The chapters in this section provide an in-depth description of these aspects. Chapter 16 by Zhongrui Wang, Rivu Midya and J. Joshua Yang at the University of Massachusetts, Amherst discusses various approaches to realize neuronal devices based on memristive materials. V. Milo, T. Dalgaty, Daniele Ielmini, and Elisa Vianello discuss synaptic realizations based on memristive devices in Chapter 17. Chapter 18 by Giacomo Indiveri, Bernabe Linares Barranco, and Melika Payvand describes various aspects of system-level integration in neuromorphic co-processors. Finally, M. E. Fouda, F. Kurdahi, A. Eltawil, and E. Neftci in Chapter 19 describe a memristor-based design perspective for inference and learning with Spiking Neural Networks.

15.5 Future outlook

In spite of the potential advantages of spiking networks described in this chapter, they have not found wide successful adoption by the industry and academia so far. One of the fundamental reasons for this is the lack of efficient learning algorithms for SNNs, similar to the backpropagation algorithm used for second-generation deep learning networks. Considerable progress has been made in this domain over the past decade, especially in deriving the basic algorithmic approaches to implement supervised and semi-supervised learning with spikes for small networks [17]. However these methods have not yet been applied successfully to demonstrate the applicability of multi-layered SNNs to solve large benchmark problems in machine learning.

Hence the ultimate success of spiking networks for realizing efficient learning systems will depend on the development of learning algorithms that allow network optimization in an event-driven manner, as well as hardware platforms based on nanoscale devices that mitigate the limitations of conventional von Neumann systems.

References

[1] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (2015) 436–444.

[2] E.R. Kandel, Nobel Lecture, Physiology or Medicine, 2000. https://www.nobelprize.org/uploads/2018/06/kandel-lecture.pdf

[3] W. Maas, Networks of spiking neurons: the third generation of neural network models, Trans. Soc. Comput. Simul. Int. 14 (4) (1997) 1659–1671. URL http://dl.acm.org/citation.cfm?id = 281543.281637.

[4] C. Mead, Neuromorphic electronic systems, Proc. IEEE 78 (10) (1990) 1629–1636. Available from: https://doi.org/10.1109/5.58356.

[5] D. Ielmini, H.S.P. Wong, In-memory computing with resistive switching devices, Nat. Electron. 1 (6) (2018) 333–343. Available from: https://doi.org/10.1038/s41928-018-0092-2. URL https://doi.org/10.1038/s41928-018-0092-2.

[6] E.M. Izhikevich, Which model to use for cortical spiking neurons? IEEE Trans. Neural Netw. 15 (5) (2004) 1063–1070. Available from: https://doi.org/10.1109/TNN.2004.832719.

[7] E.T. Rolls, A. Treves, The neuronal encoding of information in the brain, Prog. Neurobiol. 95 (3) (2011) 448–490.

[8] A. Pedram, S. Richardson, M. Horowitz, S. Galal, S. Kvatinsky, Dark memory and accelerator-rich system optimization in the dark silicon era, IEEE Des. Test. 34 (2) (2017) 39–50. Available from: https://doi.org/10.1109/MDAT.2016.2573586.

[9] A. Mortara, E. Vittoz, P. Venier, A communication scheme for analog VLSI perceptive systems, IEEE J. Solid-State Circuits 30 (6) (1995) 660–669. Available from: https://doi.org/10.1109/4.387069.

[10] B. Rajendran, A. Sebastian, M. Schmuker, N. Srinivasa, E. Eleftheriou, Low-power neuromorphic hardware for signal processing applications: a review of architectural and system-level design approaches, IEEE Signal. Process. Mag. 36 (6) (2019) 97–110. Available from: https://doi.org/10.1109/MSP.2019.2933719.

[11] P.A. Merolla, J.V. Arthur, R. Alvarez-Icaza, A.S. Cassidy, J. Sawada, F. Akopyan, et al., A million spiking-neuron integrated circuit with a scalable communication network and interface, Science 345 (6197) (2014) 668–673. Available from: https://doi.org/10.1126/science.1254642.

[12] M. Davies, N. Srinivasa, T. Lin, G. Chinya, Y. Cao, S.H. Choday, et al., Loihi: A neuromorphic manycore processor with on-chip learning, IEEE Micro 38 (1) (2018) 82–99. Available from: https://doi.org/10.1109/MM.2018.112130359.

[13] B.V. Benjamin, P. Gao, E. McQuinn, S. Choudhary, A.R. Chandrasekaran, J.-M. Bussat, et al., Neurogrid: a mixed-analog-digital multichip system for large-scale neural simulations, Proc. IEEE 102 (5) (2014) 699–716.

[14] S. Moradi, N. Qiao, F. Stefanini, G. Indiveri, A scalable multicore architecture with heterogeneous memory structures for dynamic neuromorphic asynchronous processors (DYNAPs), IEEE Trans. Biomed. Circuits Syst. 12 (1) (2018) 106–122.

[15] C. Frenkel, M. Lefebvre, J. Legat, D. Bol, A 0.086-mm^2 12.7-pJ/SOP 64k-synapse 256-neuron online-learning digital spiking neuromorphic processor in 28-nm CMOS, IEEE Trans. Biomed. Circuits Syst. 13 (1) (2019) 145–158. Available from: https://doi.org/10.1109/TBCAS.2018.2880425.

[16] S.R. Nandakumar, S.R. Kulkarni, A.V. Babu, B. Rajendran, Building brain-inspired computing systems: examining the role of nanoscale devices, IEEE Nanotechnol. Mag. 12 (3)

(2018) 19–35. Available from: https://doi.org/10.1109/MNANO.2018.2845078. URL https://ieeexplore.ieee.org/document/8438410/.

[17] O. Simeone, B. Rajendran, A. Gruning, E.S. Eleftheriou, M. Davies, S. Deneve, et al., Learning algorithms and signal processing for brain-inspired computing [from the guest editors], IEEE Signal. Process. Mag. 36 (6) (2019) 12–15. Available from: https://doi.org/10.1109/MSP.2019.2935557.

Chapter 16

Neuronal realizations based on memristive devices

Zhongrui Wang, Rivu Midya and J. Joshua Yang
Department of Electrical and Computer Engineering, University of Massachusetts, Amherst, MA, United States

16.1 Introduction

16.1.1 Spiking neuron network

AlphaGo and its variants have demonstrated the potential of deep neural networks in making strategic decisions [1,2]. To make neuromorphic computing more bio-realistic, investigations have been made on various representations of neuronal signals, which is usually classified into three generations [3].

- Generation 1: McCulloch−Pitts neurons with discrete outputs
- Generation 2: Neurons with analog outputs (or continuous activation function)
- Generation 3: Spiking neurons with time-domain signal outputs

The third generation, or the spiking neural network (SNN), is the most bio-realistic of the three, where information is encoded into the timing and frequency of spikes like those of biological neural circuits [4,5]. SNN-based computation operates on the time-dependent state evolution of a dynamical system. This property equips SNNs with physiological spatiotemporal learning rules [e.g., Hebbian rules like spike-timing-dependent plasticity (STDP)], without the need of gradient-based optimization techniques that are popular with nonspiking artificial neural networks. In addition, SNN is more noise-proof [5], suitable for time-dependent problems [6], and energy efficient for a variety of tasks [7−9].

16.1.2 Conventional transistor-based spiking neurons

Transistors have long been used to build large-scale SNNs, such as TrueNorth [9], Loihi [10], NeuroGrid [11], HICANN [12] (or BrainScaleS

Memristive Devices for Brain-Inspired Computing. DOI: https://doi.org/10.1016/B978-0-08-102782-0.00016-2

[13]), and SpinNaker [14]. These SNNs could execute advanced algorithms more efficiently when compared to CPU or GPU based implementations. However, the complexity of such hardware SNNs is limited by the fundamental building block of these systems, the complementary metal–oxide–semiconductor (CMOS) transistor. As Moore's law is becoming increasingly obsolete, significant improvements of synapse and neuron density via scaling or 3D stacking while maintaining comparable power efficiency may be more feasible with compact analog state devices. The decision-making capability of a neural network is generally determined by its size (i.e., how deep the network is and how many hidden neurons there are per layer). A powerful and yet energy-efficient system could be built with memristors, which are novel two terminal analog devices that are compact, fast, and less power hungry. Such devices have been used to build artificial synapses by mapping their conductance to synaptic weights in both artificial networks and SNNs [15–22]. More recently, memristive devices have been used to bio-realistically implement the temporal plasticity of chemical synapses [18,23–25]. Apart from synapses, temporal synaptic integration has recently been reported with artificial neurons based on phase-change memristors [26], redox memristors [27–29], Ovonic switches [4], Mott memristors [30–32], and magnetic tunneling junctions [33,34]. Such neurons show stochastic integrate-and-fire properties, resembling their biological counterparts closely. This has enabled the building of hardware spiking neuron networks.

In this chapter, different approaches to build neurons with memristive devices are surveyed keeping in mind the underlying physical mechanisms. The unsupervised learning protocols enabled by such neurons are discussed when the neurons are paired with emerging memristive synapses.

16.2 Novel memristor-based neurons

The memristor is a mathematical concept conceived by Chua linking the differential change of electrical charge and that of magnetic flux [35]. A generalized memristor has been realized by HP labs in 2008 with a TiO_x resistive switching element [36]. Memristive devices exhibit voltage history dependent electrical resistance, which could be realized with various atomic and electronic effects, including but not limited to phase change, redox reactions, Ovonic switching, Mott insulator-to-metal transition, and magnetoresistance.

16.2.1 Phase-change memristor

Tuma et al. pioneered in exploiting the accumulative switching properties of chalcogenide-based phase-change materials to implement the neural integrate-and-fire dynamics [26]. Phase-change memristors operate on amorphous-to-crystal phase transitions of a nanometric volume of $Ge_2Sb_2Te_5$

sandwiched between two electrodes. The phase configuration of the material, quantified in this case in the form of memristor conductance, is mapped onto the electrochemical membrane potential of biological neurons, as schematically illustrated in Fig. 16.1A.

A subthreshold voltage spike train, with an amplitude low enough to avoid global melting but still high enough to induce substantial crystal growth, serves as the presynaptic signal. As shown in Fig. 16.1B, the conductance of the memristor rises slowly with the number of pulses before an abrupt increment of conductance. The rapid onset of the conductance change, or equivalently the firing event, is associated with a positive conductance and joule heating feedback mechanism (i.e., the larger the conductance, the larger the heating power, and the faster the crystallization), which bears a

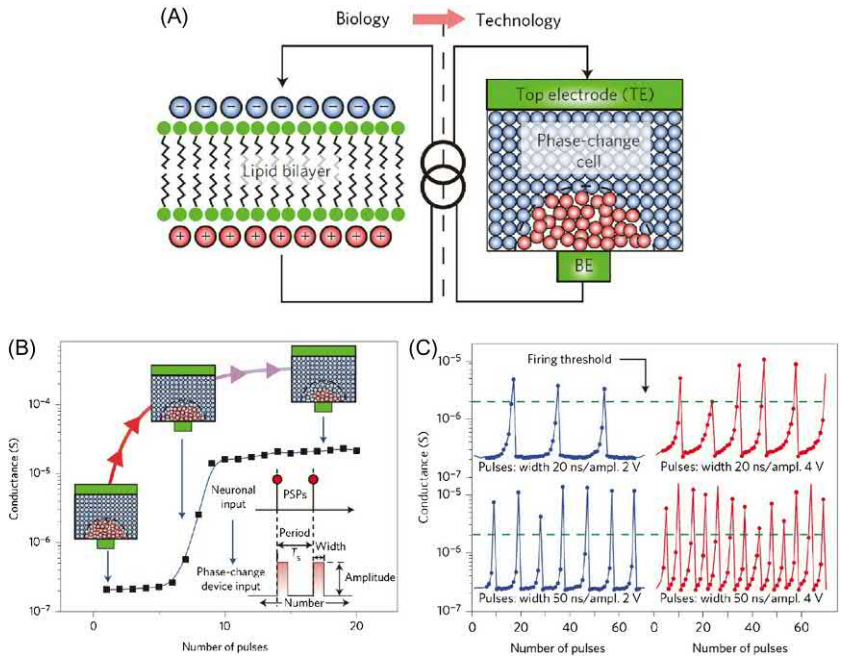


FIGURE 16.1 The stochastic phase-change integrate-and-fire neuron. (A) The key computational element is the neuronal membrane, which stores the membrane potential in the phase configuration of a nanoscale phase-change device. (B) The evolution of the phase-change device conductance as a function of the number of crystallizing pulses. After approximately six pulses, the measured conductance rises sharply, causing the neuron to fire. (C) The integrate-and-fire dynamics in a phase-change neuron. After reaching a conductance threshold of 2 μS, the phase-change device is automatically reset to the initial state, which results in a sequence of firing events. The neuron fires with a frequency that is determined by the amplitude (power) and the duration of the crystallizing pulses. *Reprinted from T. Tuma, A. Pantazi, M. Le Gallo, A. Sebastian, E. Eleftheriou, Stochastic phase-change neurons, Nat. Nanotechnol. 11 (2016) 693–699. Copyright 2016, Springer Nature Limited.*

similarity to the dynamics of voltage-gated ion channels of biological neurons. The rate of neuron firing (with an automatic RESET) depends on both the electrical power of the presynaptic pulse as well as its time duration (see Fig. 16.1C).

Since phase-change memristors are nonvolatile, a RESET mechanism is needed to bring the excited membrane potential back to the resting value after each firing event. One of the solutions is to deploy an analog comparator circuit as investigated by Cobley et al. and Tuma et al. [26,37]. In the proposed method, a phase-change memristor is in series with a resistor connected to the ground terminal. The switching of the phase-change device shifts the voltage divisions between the phase-change memristor and the resistor thus triggering the response of an analog comparator which has a fixed reference voltage. The output of analog comparator is fed back to the phase-change memristor via a delay block to RESET the phase-change cell to its low conductance state, which would subsequently disable the comparator output. This proposed technique is easy to implement and robust, at the expense of an increased number of transistors per neuron.

16.2.2 Redox and electronic memristor

Redox memristors, particularly those based on Ag diffusion and electrochemical reactions (e.g., diffusive memristors), could show threshold resistance switching due to minimization of interfacial energy between Ag and host dielectrics [25,38,39]. Wang et al. [40] and Zhang et al. [41] have both reported similar realization of stochastic integrate-and-fire neurons by combining such Ag-based diffusive memristors with external capacitance, bearing resemblance with the neuristor concept to be discussed in later sections. In such a configuration, the threshold behavior of the diffusive memristor [42] can mimic the switching of an ion channel located near the soma of a neuron, whereas the membrane capacitance and axial resistance are equivalent, respectively, to the capacitor parallel to the memristor and a resistor in series with this combination, as schematically shown in Fig. 16.2A. During integration, the spatially summed temporal presynaptic signal is applied to the capacitor which charges up the equivalent membrane capacitance, activating the diffusive memristor "ion channels" if the accumulated charge reaches the threshold to fire the neuron, which shares a strong resemblance with the temporal summation process of a biological neuron near the soma. Fig. 16.2B illustrates the accumulation of membrane potential due to capacitor charging and the consecutive firing of the diffusive memristor neuron. Larger capacitance yields slower integration process due to an increased amount of charge needed to reach the threshold potential. Larger series resistance with the diffusive memristor reduces the equivalent conductance of the memristor, resulting in smaller charging current and slow charge buildup, as summarized in Fig. 16.2B and C.

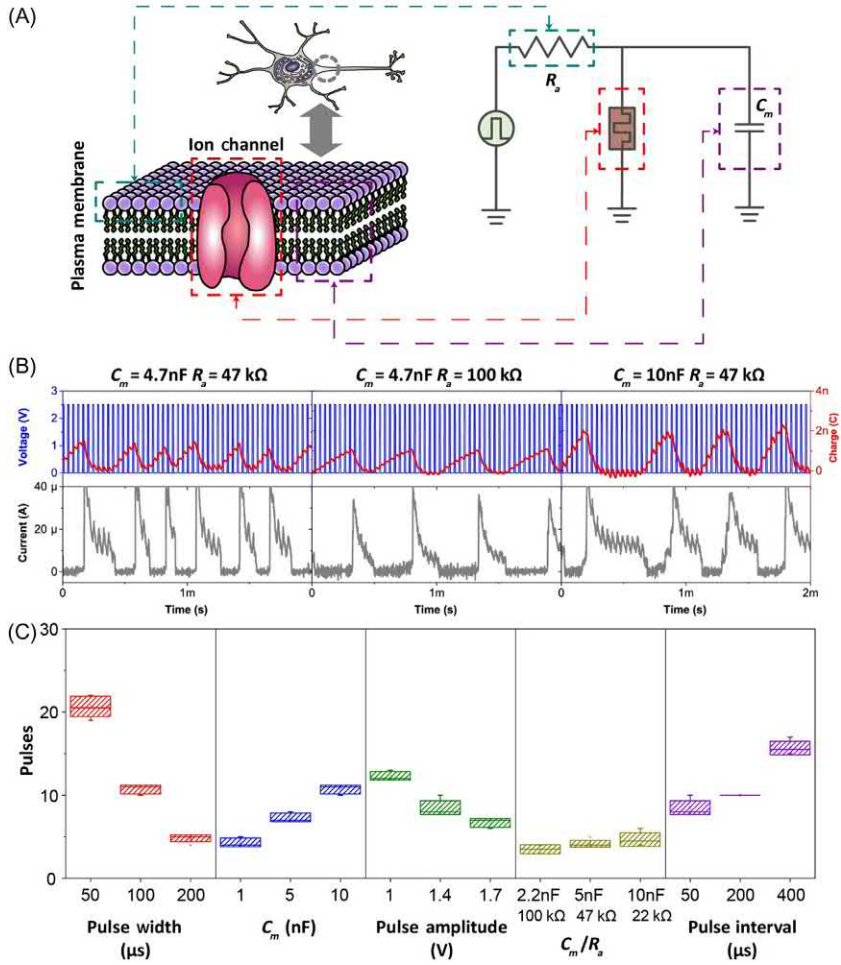


FIGURE 16.2 A diffusive memristor neuron with a parallel capacitance. (A) Illustration of an ion channel embedded in the cell membrane near the soma of a biological neuron. The inputs from the dendrites are integrated on the capacitance of the membrane and the ion channel opens if the threshold condition is reached. Also shown is the analogous electrical integrate-and-fire circuit of the artificial neuron in which the diffusive memristor functions as the ion channel and the capacitor acts as the membrane. (B) Response of the integrate-and-fire circuit to multiple consecutive pulses and the influence of varying membrane capacitance (C_m) and axial resistance (R_a) shows how the number of pulses required to charge the capacitor up to the memristor threshold increases with increasing C_m or R_a. The current pulse across the diffusive memristor coincides with the discharge of the capacitor, clearly demonstrating that the device is actively firing a pulse of stored charge. (C) Controlled firing response of the integrate-and-fire circuit under different input and circuit conditions. A similar effect as in (B) can be observed by changing the input parameters such as the pulse width (shorter pulses result in larger number of pulses before firing), pulse interval (shorter intervals result in smaller pulse number), and circuit parameters such as capacitance (larger capacitance delays the firing). Changing the input resistance while keeping the *RC* constant results in a small or no change in the firing. *Reprinted from Z. Wang, et al., Fully memristive neural networks for pattern classification with unsupervised learning, Nat. Electron. 1 (2018) 137−145. Copyright 2018, Springer Nature Limited.*

In addition, akin to a biological neuron, the input stimuli affect the integration process. For instance, shorter spikes increase the number of required pulses per firing. Notice that the finite resistance of the diffusive memristor naturally replicates the leaky effect of biological neurons, which makes the integration pulse interval dependent.

The SET transition of Ag-based threshold switches and nonvolatile conducting-bridge cells have shown accumulated switching similar to that of phase-change memristors. This has been exploited to build a single memristor neuron with self-relaxation. Fig. 16.3A depicts an example of such an integrate-and-fire neuron, fabricated by sandwiching an Ag-doped dielectric material between two metal plate electrodes. With voltage pulse trains applied to this artificial neuron (with a series resistor to limit the current, the parasitic capacitance of the diffusive memristor corresponds to the parallel capacitor symbol in Fig. 16.3A), the neural current or equivalently the memristor conductance is negligible in the first few spikes of each pulse train before a sharp rise, which is attributed to the gradual Ag filament formation modulated by the internal Ag dynamics, as shown in Fig. 16.3B. The internal Ag dynamics of diffusive memristors originates from complicated multiphysics effects including field-induced Ag mass transport from the electrodes, such as Ag diffusion and redox reaction [43–47], which is also stochastic in nature (see Fig. 16.3C).

The major functional difference between the neuron implementation with external capacitance and that without it is the activation dependence on the amplitude of the presynaptic signals. For the neuron with an external capacitance, during charging up, doubling the amplitude of input pulse may reduce the integration time approximately by half. However, doubling the amplitude may reduce the integration time to less than 1% of the original one with the single diffusive memristor neuron. The difference in the activation process makes the two neuron architectures suit different tasks.

Furthermore, Mehonic and Kenyon have shown that the instability of SiO_2 switching could be exploited to integrate input current pulses and produce output voltage transients [28]. Such voltage transients, reflected by the voltage drop across the memristor, naturally mimics the action potential spiking activity observed in a biological neuron.

16.2.3 Ovonic chalcogenide glass

Ovonic switching was first reported in the 1960s by Ovshinsky [48]. It is widely believed that the coupled electrothermal effects on carrier population are responsible for the observed resistive threshold switching in chalcogenide materials [49–53].

Lim et al. devised a hybrid structure consisting of a nonideal operational amplifier in series with a Pearson–Anson oscillator [4]. The latter comprises of an Ovonic chalcogenide threshold switch, a capacitor, and two resistors,

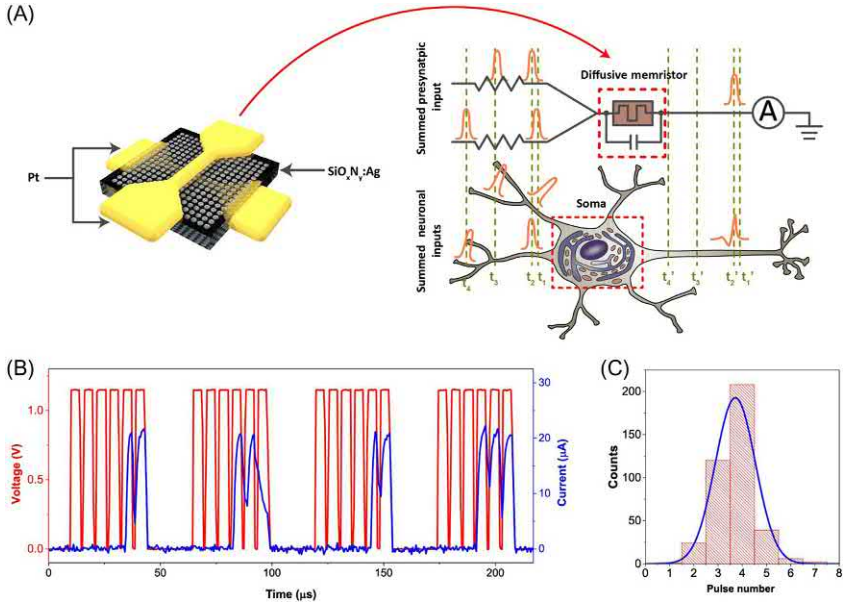


FIGURE 16.3 Intrinsic integrate and fire of a single diffusive memristor. (A) Schematic illustration of a cross-point diffusive memristor, which consists of a SiO_xN_y:Ag layer between two Pt electrodes. The artificial neuron receives software summed weighted presynaptic inputs via a pulsed voltage source and an equivalent synaptic resistor (e.g., 20 μs in this case). Both the artificial and biological neurons integrate input stimuli (*orange*) beginning at t_1 and fire when the threshold condition is reached (i.e., at t_2'). The integrated signal decays over time such that input stimuli spaced too far apart will fail to reach threshold (i.e., the delay between t_3 and t_4). (B) Experimental response of the device to multiple subthreshold voltage pulses followed by a rest period of 200 μs (only 20 μs is shown for convenience). The device required multiple pulses to reach the threshold and "fire." (C) Histogram of the number of subthreshold voltage pulses required to successfully fire the artificial neuron (*red*) compared to a Gaussian distribution (*blue*). *Reprinted from Z. Wang, et al., Fully memristive neural networks for pattern classification with unsupervised learning, Nat. Electron. 1 (2018) 137–145. Copyright 2018, Springer Nature Limited.*

as schematically shown in Fig. 16.4A. A sufficiently large input voltage could turn ON the Ovonic switch and pull up the output voltage. Meanwhile, the ON switching of the Ovonic switch makes the voltage drop across it decrease, which subsequently RESETs the switch back to its OFF state and zeroes the output voltage. These two opposite changes in output voltage take place in close succession, producing an output spike with the temporal width modulated by the capacitor and input signal (see Fig. 16.4B). Lim et al. have also assembled a pair of neurons linked via a synaptic resistor as shown in Fig. 16.4C. The spiking of the presynaptic neuron is modulated by the synaptic weight before being fed to the postsynaptic neuron and triggers delayed firing events (see Fig. 16.4D and E).

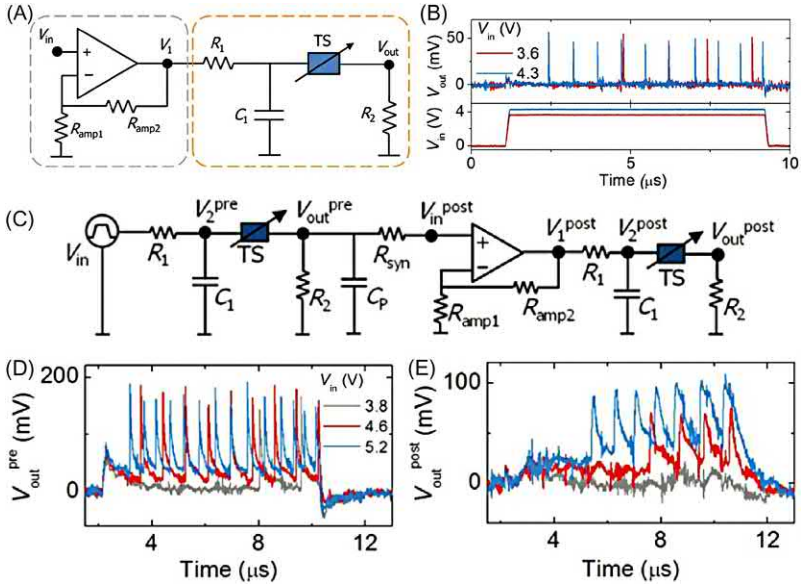


FIGURE 16.4 (A) Equivalent circuit of the relaxation oscillator leaky integrate-and-fire neuron. The gray and orange boxes indicate a nonideal op-amp and Pearson−Anson oscillator encompassing the Ovonic threshold switch, respectively. (B) Oscillating outputs of the Pearson−Anson oscillator under the corresponding square voltage pulses. (C) Equivalent circuit of a pair of neurons that are connected through a synaptic resistor ($R_{syn} = 1\,k\Omega$ and $R_{amp2} = 100\,k\Omega$). C_P indicates the parasitic capacitance originating from the wiring. (D) Spike bursts in the presynaptic neurons at different constant voltages (3.8, 4.6, and 5.2 V). (E) The consequent postsynaptic spike bursts. *Reprinted from H. Lim, et al., Relaxation oscillator-realized artificial electronic neurons, their responses, and noise, Nanoscale 8 (2016) 9629−9640. Copyright 2016, Royal Society of Chemistry.*

16.2.4 Mott insulators

Strongly correlated Mott insulators have long been researched for their reversible insulator-to-metal transition upon the perturbation of either temperature or electric field [54,55]. Pickett et al. have built neuristors, short for neuron-like resistors, based on dynamical resistance behaviors possessed by Mott insulator memristors similar to Hodgkin−Huxley ion channels [30], which has equipped neuristors with important features such as threshold-driven spiking, lossless spike propagation at a constant velocity with uniform spike shape, and a refractory period . A neuristor consists of two identical Mott memristors (see Fig. 16.5A) with parallel capacitors. The two memristors are biased with opposite polarities to mimic the sodium and potassium channels of the Hodgkin−Huxley model, with a linked load resistor.

Fig. 16.5B shows the simulated all-or-nothing action potential response of the neuristor. With a superthreshold input pulse, the neuristor could be

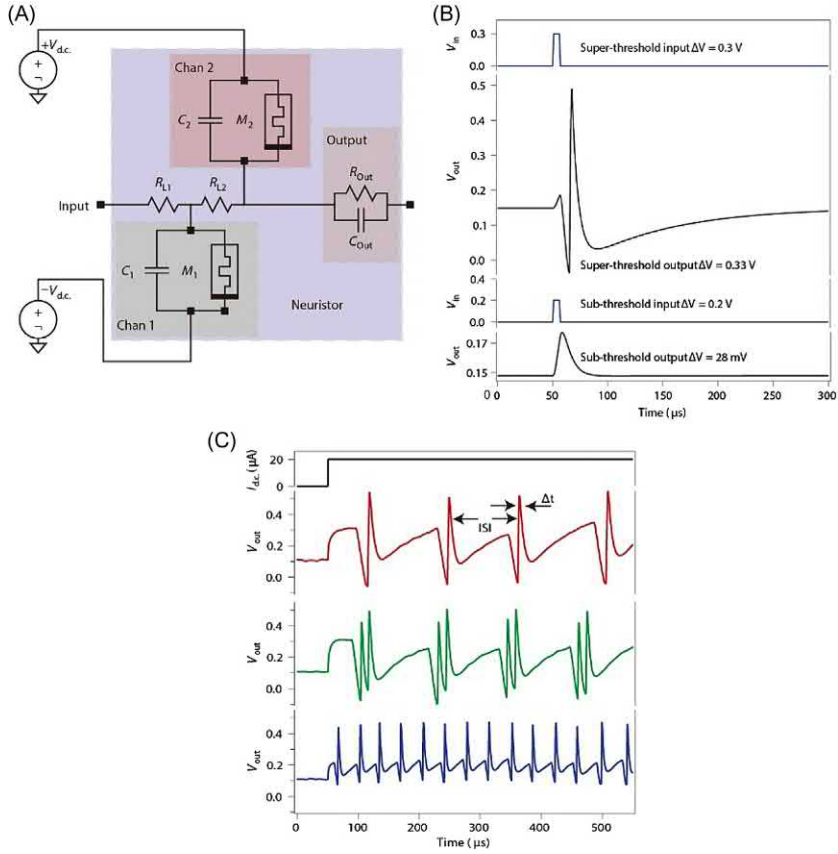


FIGURE 16.5 (A) Circuit diagram of the lumped neuristor. The channels consist of Mott memristors (M1 and M2), each with a characteristic parallel capacitance (C_1 and C_2, respectively) and are biased with opposite polarity DC voltage sources. (B) All-or-nothing response of the neuristor. The two upper panels show the input of the superthreshold 0.3 V input pulse and its corresponding spike output. The two lower panels show the input of the subthreshold 0.2 V input and its attenuated output. (C) Various neuristor voltage spiking output patterns with a current-sourced input. As the channel capacitances C_1 and C_2 are adjusted, the interspike interval (ISI) and spike width (Δt) are modified such that the neuristor exhibits: regular spiking ($C_1 = 5.1$ nF, $C_2 = 0.75$ nF), chattering ($C_1 = 5.1$ nF, $C_2 = 0.5$ nF), and fast spiking ($C_1 = 1.6$ nF, $C_2 = 0.5$ nF). C_1 controls the ISIs and C_2 controls the spike width (Δt). *Reprinted from M.D. Pickett, G. Medeiros-Ribeiro, R.S. Williams, A scalable neuristor built with Mott memristors, Nat. Mater. 12 (2013) 114–117. Copyright 2013, Springer Nature Limited.*

excited to produce an action potential with an amplitude of 0.33 V, larger than that of the input spike. On the contrary, a subthreshold 0.2 V voltage pulse will be attenuated to 0.028 V, indicating the thresholding capability of the neuristor. Another merit of the neuristor is the diversity of spiking behaviors when using a current source. By pairing different capacitances, the

neuristor could output various biomimetic spiking patterns such as regular spiking, chattering, and fast spiking, as shown in Fig. 16.5C.

Lim et al. [31] and Parihar et al. [56] have studied the noise due to the stochasticity of the threshold switching in the neuristor-based leaky integrate-and-fire neuron. This simplified neuristor-based leaky integrate-and-fire neuron, consisting of one memristor and one parallel capacitor, has been reported as a compact implementation of oscillatory neurons by Chen et al. [57] and Lee et al. [58] for memristive neural networks.

Mott memristors may also show accumulative switching without the assistance of external capacitance. Stoliar et al. have investigated the threshold switching of lacunar spinel compounds containing transition-metal tetrahedral clusters with narrow Mott−Hubbard gap of 0.1−0.3 eV [32]. Fig. 16.6A shows the measurement of a single $GaTa_4Se_8$ Mott insulator memristor neuron in series with a load resistor at cryogenic temperature. With a superthreshold input voltage spike, the neuron current rises slowly in the first-half of the pulse before an abrupt increment after 89 μs, illustrating the accumulative switching capability of the neuron (Fig. 16.6B). By replacing the single superthreshold pulse with multiple subthreshold ones, the leaky integrate-and-fire behavior could be implemented. Fig. 16.6C illustrates such an integration process where a train of short pulses with 20 μs duration

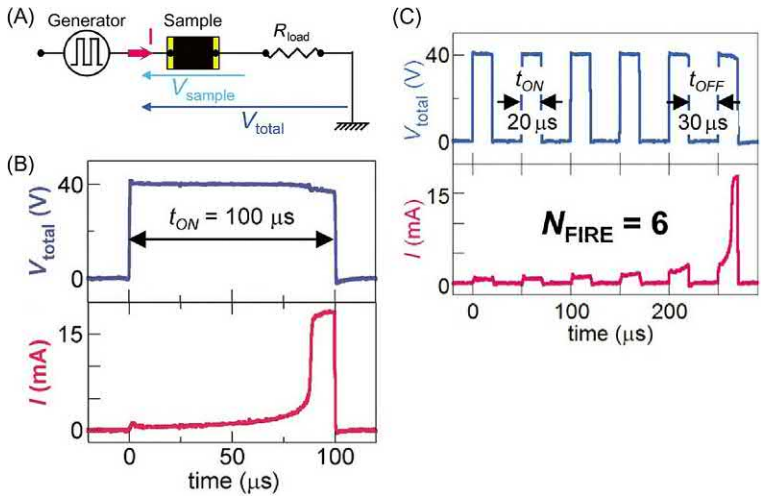


FIGURE 16.6 (A) Measurement setup of the Mott insulator neuron. (B) Accumulative threshold switching of the neuron by applying a long single voltage pulse at 74K. The memristor is a slice of $GaTa_4Se_8$ crystal with interelectrode distance of about 40 μm. (C) Integrate and fire characteristics of the neuron by applying a train of short pulses of 20 μs width and 30 μs interval. The neuron fires after six spikes. *Reprinted from P. Stoliar, et al. A leaky-integrate-and-fire neuron analog realized with a Mott insulator, Adv. Funct. Mater. 27 (2017) 1604740. Copyright 2017, WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim.*

and 30 μs separation is applied. The Mott insulator memristor switches after six pulses.

16.2.5 Magnetic tunneling junction

Unlike the aforementioned memristors which physically form and rupture conducting paths, magnetic tunneling junctions rely on magnetization modulated tunneling resistance. Such memristors usually have two ferromagnetic layers separated by a tunneling oxide barrier, as shown in Fig. 16.7A[33]. The magnetization direction of one of the two ferromagnetic layers could be switched by a spin polarized current, which results in two stable resistance states, the parallel spin state with larger conductance and antiparallel spin state with lower conductance. An energy barrier exists along the transition path from one state to the other (see Fig. 16.7B).

Sengupta et al. have simulated the accumulative switching or integrate-and-fire capability of the magnetic tunneling junction neuron. The membrane potential of the biological neuron could be mapped to the in-plane

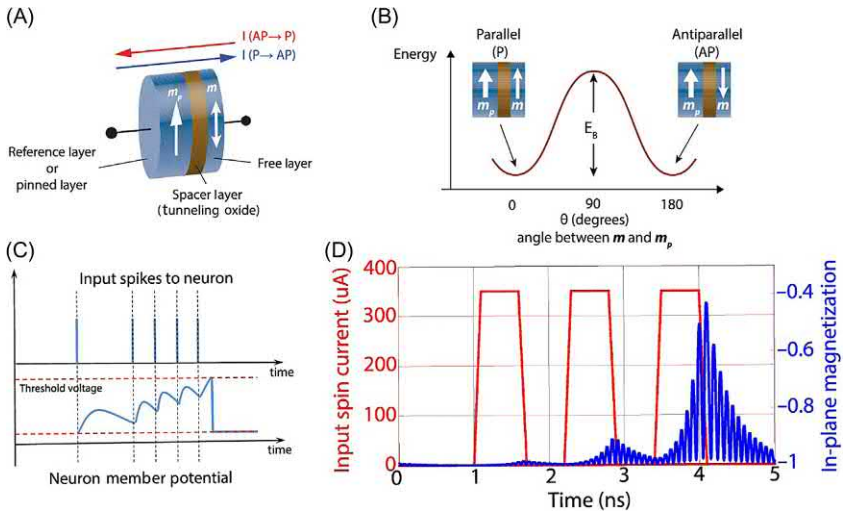


FIGURE 16.7 (A) A magnetic tunnel junction consists of two magnetic layers sandwiching a spacer layer. While the magnetization direction of the reference layer is pinned, the magnetization of the free layer can be manipulated by an input charge current. The magnetic tunneling junction has two stable resistance states, namely, the parallel *(P)* and antiparallel *(AP)* configuration. (B) The barrier height (E_B) makes the P and AP states thermally stable. (C) The membrane potential of a biological neuron integrates input spikes and leaks when there is no input. It spikes when the membrane potential crosses the threshold. (D) Magnetic tunneling junction neuron due to the application of three input pulses. The in-plane magnetization starts integrating due to the pulses and then starts leaking once the pulse is removed. *Reprinted from A. Sengupta, P. Panda, P. Wijesinghe, Y. Kim, K. Roy, Magnetic tunnel junction mimics stochastic cortical spiking neurons. Sci. Rep. 6 (2016) 30039.*

magnetic anisotropy of the magnetic tunneling junction. Fig. 16.7C shows the analogy between the leaky integration of a biological neuron and that of the simulated MTJ neuron. The neuron receives three successive voltage pulses which increases the magnetization, as shown in Fig. 16.7D. However, the extent of stimulation is insufficient, and the magnetization starts to leak once bias is removed. When the stimulation is large enough, the neuron will fire in the form of magnetization reversal of the in-plane component. Such a neuron works in two phases, one to integrate-and-fire while the other to read the resultant state and reset the MTJ back to its initial state in case the neuron fires.

16.3 Unsupervised programming of the synapses

In biological neural systems, neurons not only integrate the dendritic stimulus modulated by the synaptic junctions, but also gradually change the weights of those synapses via various synaptic plasticity mechanisms which depend on the spiking history of the pre- and postsynaptic neurons. Such synaptic weight updates are performed without supervision, which forms the basis of learning for all biological neural systems.

Although artificial memristor neurons have been shown to work with emerging artificial memristor synapses [4,33,34,40,41,57–59], a natural question is whether they could work like their biological counterparts to program the affiliated synapses without a supervisor signal. Here, we discuss the investigation of such neuron–synapse interaction with both phase-change memristor neurons and Ag-based threshold switching neurons.

16.3.1 Phase-change memristor neuron and synapse interaction

Pantazi et al. used nanoscale phase-change memristors to emulate both neuronal dynamics as well as synaptic plasticity [58]. A chip consisting of 2×2 phase-change memristor arrays (1-million devices per array) is used. Each array is equipped with dedicated on-chip addressing circuit, an analog-to-digit converter for reading device conductance, and an associated write circuitry for single memristor programming. The synapses, whose weights are represented by the conductance of single phase-change memristors, form a 400×3 fully connected neural network (see Fig. 16.8A). These phase-change synapses could be updated with a simplified asymmetric STDP learning rule such that a single SET or RESET pulse is used to update their weights during learning.

Three phase-change neurons interface with the 400 presynaptic neurons via phase-change memristor synapses. The phase-change memristor neurons are based on that shown in Fig. 16.1 in which the neural membrane potential is mapped to the atomic configuration of the phase-change memristor.

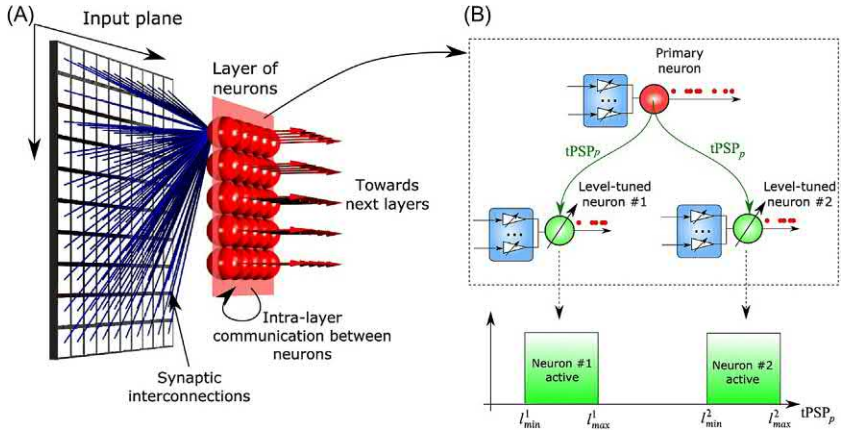


FIGURE 16.8 Schematic of the 400×3 phase-change memristor neural network with level-tuned neurons. (A) SNN overview: a one-layer feedforward network in which postsynaptic neurons are interconnected based on their internal states. (B) Concept of level-tuned neurons. The internal state of a primary neuron provides information on the cumulative characteristics of the input and is used to enable a set of level-tuned neurons. *Reprinted from A. Pantazi, S. Wozniak, T. Tuma, E. Eleftheriou, All-memristive neuromorphic computing with level-tuned neurons, Nanotechnology 27 (2016) 355205. Copyright 2016, IOP Publishing Ltd.*

The three neurons are mutually connected based on a bioinspired level-tuning concept, as shown in Fig. 16.8B. One of the three neurons serve as the primary neuron, or a trigger to selectively enable integrate-and-fire of the other neurons. This is achieved by sending the total postsynaptic potential (tPSP), or the weighted sum of the inputs, of the primary neuron to the rest neurons. In case the tPSP falls into a predefined range of a level-tuned neuron, the latter will integrate and fire. Such an approach provides a convenient implementation of lateral inhibition as explained below.

Fig. 16.9 illustrates the detection of spatial−temporal patterns. There are two subsets of presynaptic neurons (e.g., neuron 1 to 50, neuron 51 to 80 in Fig. 16.9A) such that all neurons of the same subset always fire together. The rest 320 presynaptic neurons fire randomly without any correlation.

Fig. 16.9B reveals the time evolution of the tPSP of the primary neuron. The amplitude of tPSP, once stabilized, is likely to be proportional to the number of concurrently firing neurons, so that its peaks could serve as the trigger signal for level-tuned neurons. The weights of phase-change memristor synapses affiliated to the primary neuron are updated by the synergy of both the pre- and postsynaptic neuron spikes (see Fig. 16.9C) via the simplified STDP rule which potentiates synapses interfacing the two presynaptic neuron subsets while depresses the other synapses of the primary postsynaptic neuron. Due to the threshold-enabling conditions, the firing of presynaptic neuron subsets always activates the corresponding level-tuned postsynaptic

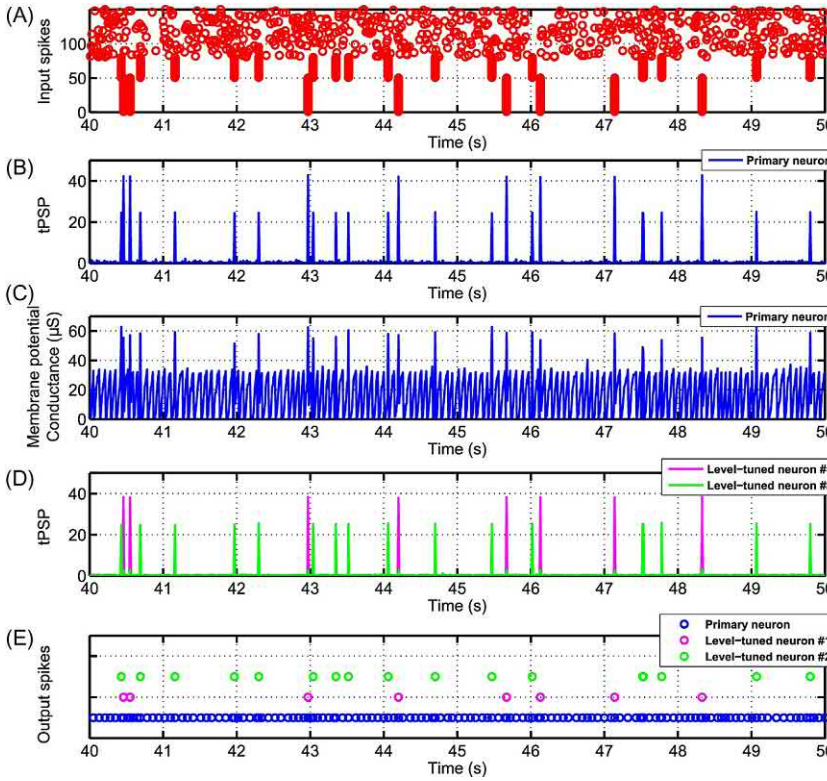


FIGURE 16.9 Unsupervised learning to detect spatial−temporal patterns with both phase-change memristor neurons and synapses. (A) Presynaptic neuron spiking patterns. Presynaptic neuron subsets (e.g., neuron 1 to 50, neuron 51 to 80) where neurons of the same subset always fire together. The rest 320 presynaptic neurons fires randomly. (B) The tPSP of the primary neuron. The amplitude of tPSP, once stabilized, is likely to be proportional to the number of concurrently firing presynaptic neurons. (C) The conductance of the primary neuron which is mapped to the membrane potential. (D) The tPSP of the level-tuned neurons. (E) The neural outputs of both primary and level-tuned postsynaptic neurons. *Reprinted from A. Pantazi, S. Wozniak, T. Tuma, E. Eleftheriou, All-memristive neuromorphic computing with level-tuned neurons, Nanotechnology 27 (2016) 355205. Copyright 2016, IOP Publishing Ltd.*

neuron, as revealed by Fig. 16.9D and the time-resolved spiking activities of postsynaptic neurons in Fig. 16.9E.

16.3.2 Redox memristor neuron

Redox memristors have been used to demonstrate the direct physical interaction between the neurons and synapses, the first of its kind, relying on simple voltage divisions between the memristors to perform unsupervised synaptic updates.

Diffusive memristor neurons based on either external capacitance (see Fig. 16.2) or intrinsic Ag accumulative switching (see Fig. 16.3) have been used to update Ta/HfO_x/Pt memristive synapses in an unsupervised fashion [40]. Fig. 16.10 illustrates this idea with a mini network which consists of 2×2 synapses connected to two diffusive memristor neurons at each output. Synapses are RESET to small conductance states before weight updates (see Fig. 16.10A). A triangular voltage pulse (for neurons with parallel capacitance, see the first column in Fig. 16.10) or a train of rectangular spikes (for neurons with intrinsic accumulative switching, see the third column in Fig. 16.10) are applied to the first row of synapses, representing an input signal "1." The presynaptic neuron to the second row is silent, or an input signal "0." As shown in the first and third columns of Fig. 16.10A and B, the

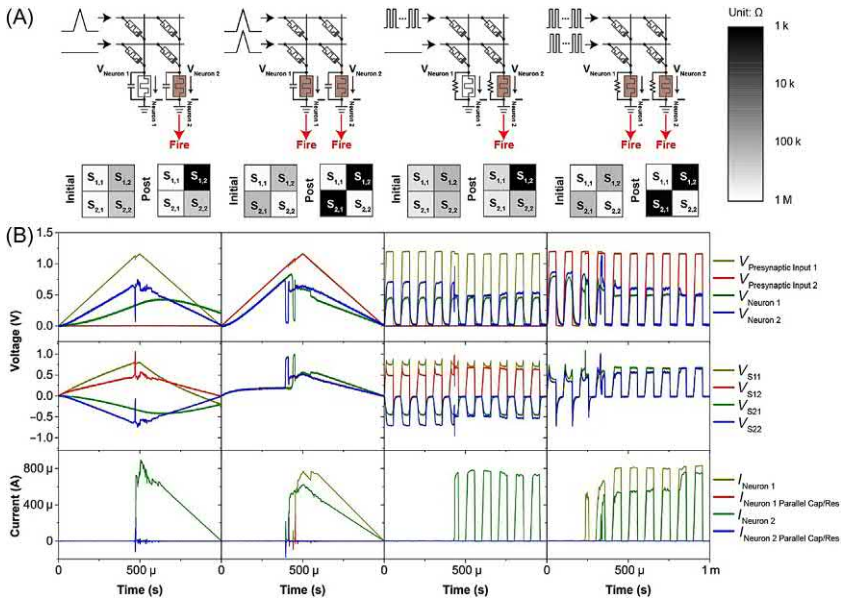


FIGURE 16.10 Experimental demonstration of unsupervised synaptic weight update using a 2×2 drift memristor array interfaced with two diffusive memristor artificial neurons. (A) Schematics of the circuits, the presynaptic inputs, the postsynaptic neuron outputs, and conductance map of the synapse array before and after update, respectively. All synapses were initialized to the high resistance state with some stochastic variation before training. (B) The measured presynaptic signals, the potentials across neurons and synapses, and the neural currents. Upon receiving a "10" input vector, the right neuron fires with both external capacitance (first column) and internal accumulative switching (third column) mechanisms, which programs the synapse S_{12}. The input vector "11" results in the firing of both neurons and programs both S_{12} and S_{21} at the same time, with both RC (second column) and internal Ag dynamics (fourth column) mechanisms. *Reprinted from Z. Wang, et al., Fully memristive neural networks for pattern classification with unsupervised learning. Nat. Electron. 1 (2018) 137–145. Copyright 2018, Springer Nature Limited.*

right postsynaptic neuron N_2 fires, which is likely due to the relatively larger initial weights of affiliated synapses. The firing of the neuron pulls down the voltage of the bottom electrodes of S_{12} and S_{22}, producing a large voltage spike (red lines of the middle panels in Fig. 16.10B) which is capable to SET S_{12} to an even larger conductance. Similarly, with an input pattern "11," both memristor postsynaptic neurons fire and potentiate the synapses S_{12} and S_{21}. (see the second and fourth columns in Fig. 16.10B). This simple hardware neuron−synapse interaction provides a novel method to implement synaptic plasticity on electronic devices.

This voltage division-based unsupervised synapse programming could be applied to a larger fully connected network. Fig. 16.11 shows such an example where an 8×3 memristive synaptic array with three diffusive memristor neurons receive inputs from a software pooling layer. Lateral inhibition

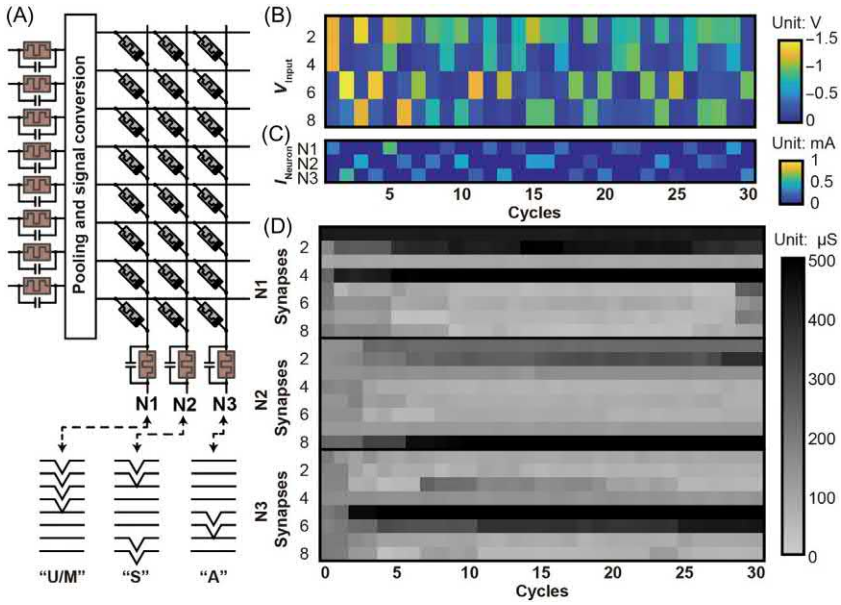


FIGURE 16.11 Unsupervised training of a fully connected network based on the integrated all-memristive neural network. (A) The schematic diagram of the 8×3 network with inputs based on the outputs of the neurons in Fig. 16.4. The prototypical patterns of neurons after training correspond to the input letters "U/M," "S," and "A" in Fig. 16.4, respectively. (B−D) The input patterns (peak voltages of triangular waveforms), peak neuronal currents, and synaptic weights at each training cycle. The synapses of the N1, N2, and N3 neurons quickly diverge from the initial 100 μs and evolve by self-organizing processes to patterns with increasing similarities to one of the prototypical patterns in (A). The magnitude of input patterns in (B) reduces in the first few cycles and becomes stable due to conductance saturation of the diverged drift memristor synapses. *Reprinted from Z. Wang, et al., Fully memristive neural networks for pattern classification with unsupervised learning. Nat. Electron. 1 (2018) 137−145. Copyright 2018, Springer Nature Limited.*

between diffusive memristor neurons are implemented with a dynamic voltage scaling scheme and associated feedback circuitry. The synapses are initialized to states with moderate conductance ($\sim 100\,\mu s$). Once diffusive memristor neuron fires, the weights of the synapses start to diverge via the simplified STDP rule to gain similarity to one of the prototypical input patterns (see Fig. 16.11A). The learning rates between synapses are not identical due to the stochastic switching nature (e.g., device-to-device variation of the threshold conditions) of drift memristors. For instance, the third synapse of N1 and the seventh synapse of N2 are much less potentiated. The learning quickly converges as reflected by the weights evolution in Fig. 16.11D and the reduction of magnitude of input patterns in Fig. 16.11B, illustrating the clustering capability of the network with the hardware encoded unsupervised learning.

16.4 Conclusion

In summary, the signal history dependent conductance of memristors provides a compact and efficient way to realize biological neural dynamics, using various atomic and electronic effects such as phase-change phenomena, redox reactions, Ovonic switching, Mott insulator-to-metal transition, and tunneling magnetoresistance. Such memristor neurons have been demonstrated with integrate-and-fire function with either external capacitance or intrinsic accumulative switching. The projected small fingerprint, good 3D integration capability, low power operation, and small fabrication cost are attractive compared to transistor-based artificial neurons.

Moreover, memristor spiking neurons could work with emerging artificial synapses, particularly the prevalent memristive synapses, leading to hardware neuromorphic computing systems with simple and robust resistive coupling between elements. Such integrated neuron−synapse circuits can implement unsupervised learning protocols, which has been demonstrated on both phase-change and redox memristors-based networks, suggesting a bio-realistic approach for developing future computing platforms.

References

[1] D. Silver, et al., Mastering the game of Go with deep neural networks and tree search, Nature 529 (2016) 484−489.

[2] D. Silver, et al., Mastering the game of Go without human knowledge, Nature 550 (2017) 354.

[3] W. Maass, Networks of spiking neurons: the third generation of neural network models, Neural. Netw. 10 (1997) 1659−1671.

[4] H. Lim, et al., Relaxation oscillator-realized artificial electronic neurons, their responses, and noise, Nanoscale 8 (2016) 9629−9640.

[5] I. Sourikopoulos, et al., A 4-fJ/spike artificial neuron in 65 nm CMOS technology, Front. Neurosci. 11 (2017) 123.

[6] S. Ghosh-dastidar, H. Adeli, Spiking neural networks, Int. J. Neural Syst. 19 (2009) 295–308.
[7] Y. Cao, Y. Chen, D. Khosla, Spiking deep convolutional neural networks for energy-efficient object recognition, Int. J. Computer Vis. 113 (2014) 54–66.
[8] B. Han, A. Sengupta, K. Roy, On the energy benefits of spiking deep neural networks: a case study, Neural Networks (IJCNN), 2016 International Joint Conference on, IEEE, 2016, pp. 971–976.
[9] P.A. Merolla, et al., A million spiking-neuron integrated circuit with a scalable communication network and interface, Science 345 (2014) 668–673.
[10] C.-K. Lin, et al., Programming spiking neural networks on Intel's Loihi, Computer 51 (2018) 52–61.
[11] B.V. Benjamin, et al., Neurogrid: a mixed-analog-digital multichip system for large-scale neural simulations, Proc. IEEE 102 (2014) 699–716.
[12] J. Schemmel, et al., A wafer-scale neuromorphic hardware system for large-scale neural modeling, Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on, IEEE, 2010, pp. 1947–1950.
[13] S. Friedmann, et al., Demonstrating hybrid learning in a flexible neuromorphic hardware system, IEEE Trans. Biomed. Circuits Syst. 11 (2017) 128–142.
[14] S.B. Furber, F. Galluppi, S. Temple, L.A. Plana, The SpiNNaker Project, Proc. IEEE 102 (2014) 652–665.
[15] S.H. Jo, et al., Nanoscale memristor device as synapse in neuromorphic systems, Nano Lett. 10 (2010) 1297–1301.
[16] S. Yu, Y. Wu, R. Jeyasingh, D. Kuzum, H.S.P. Wong, An electronic synapse device based on metal oxide resistive switching memory for neuromorphic computation, IEEE Trans. Elect. Dev. 58 (2011) 2729–2737.
[17] T. Ohno, et al., Short-term plasticity and long-term potentiation mimicked in single inorganic synapses, Nat. Mater. 10 (2011) 591–595.
[18] Z.Q. Wang, et al., Synaptic learning and memory functions achieved using oxygen ion migration/diffusion in an amorphous InGaZnO memristor, Adv. Funct. Mater. 22 (2012) 2759–2765.
[19] J.J. Yang, D.B. Strukov, D.R. Stewart, Memristive devices for computing, Nat. Nanotechnol. 8 (2013) 13–24.
[20] H. Lim, I. Kim, J.S. Kim, C.S. Hwang, D.S. Jeong, Short-term memory of TiO_2-based electrochemical capacitors: empirical analysis with adoption of a sliding threshold, Nanotechnology 24 (2013) 384005.
[21] S. La Barbera, D. Vuillaume, F. Alibart, Filamentary switching: synaptic plasticity through device volatility, ACS Nano 9 (2015) 941–949.
[22] M. Prezioso, et al., Training and operation of an integrated neuromorphic network based on metal-oxide memristors, Nature 521 (2015) 61–64.
[23] C. Du, W. Ma, T. Chang, P. Sheridan, W.D. Lu, Biorealistic implementation of synaptic functions with oxide memristors through internal ionic dynamics, Adv. Funct. Mater. 25 (2015) 4290–4299.
[24] S. Kim, et al., Experimental demonstration of a second-order memristor and its ability to biorealistically implement synaptic plasticity, Nano Lett. 15 (2015) 2203–2211.
[25] Z. Wang, et al., Memristors with diffusive dynamics as synaptic emulators for neuromorphic computing, Nat. Mater. 16 (2016) 101–108.
[26] T. Tuma, A. Pantazi, M. Le Gallo, A. Sebastian, E. Eleftheriou, Stochastic phase-change neurons, Nat. Nanotechnol. 11 (2016) 693–699.

[27] M. Al-Shedivat, R. Naous, G. Cauwenberghs, K.N. Salama, Memristors empower spiking neurons with stochasticity, IEEE Trans. Emerg. Sel. Top. Circuits Syst. 5 (2015) 242–253.
[28] A. Mehonic, A.J. Kenyon, Emulating the electrical activity of the neuron using a silicon oxide RRAM cell, Front. Neurosci. 10 (2016) 57.
[29] I. Gupta, et al., Real-time encoding and compression of neuronal spikes by metal-oxide memristors, Nat. Commun. 7 (2016) 12805.
[30] M.D. Pickett, G. Medeiros-Ribeiro, R.S. Williams, A scalable neuristor built with Mott memristors, Nat. Mater. 12 (2013) 114–117.
[31] H. Lim, et al., Reliability of neuronal information conveyed by unreliable neuristor-based leaky integrate-and-fire neurons: a model study, Sci. Rep. 5 (2015) 9776.
[32] P. Stoliar, et al., A leaky-integrate-and-fire neuron analog realized with a Mott insulator, Adv. Funct. Mater. 27 (2017) 1604740.
[33] A. Sengupta, P. Panda, P. Wijesinghe, Y. Kim, K. Roy, Magnetic tunnel junction mimics stochastic cortical spiking neurons, Sci. Rep. 6 (2016) 30039.
[34] A. Jaiswal, S. Roy, G. Srinivasan, K. Roy, Proposal for a leaky-integrate-fire spiking neuron based on magnetoelectric switching of ferromagnets, IEEE Trans. Elect. Dev. 64 (2017) 1818–1824.
[35] L. Chua, Memristor-The missing circuit element, IEEE Trans. Circuit Theory 18 (1971) 507–519.
[36] D.B. Strukov, G.S. Snider, D.R. Stewart, R.S. Williams, The missing memristor found, Nature 453 (2008) 80–83.
[37] R.A. Cobley, H. Hayat, C.D. Wright, A self-resetting spiking phase-change neuron, Nanotechnology 29 (2018) 195202.
[38] Z. Wang, et al., Threshold switching of Ag or Cu in dielectrics: materials, mechanism, and applications, Adv. Funct. Mater. 28 (2018) 1704862.
[39] W. Wang, et al., Surface diffusion-limited lifetime of silver and copper nanofilaments in resistive switching devices, Nat. Commun. 10 (2019) 81.
[40] Z. Wang, et al., Fully memristive neural networks for pattern classification with unsupervised learning, Nat. Electron. 1 (2018) 137–145.
[41] X. Zhang, et al., An artificial neuron based on a threshold switching memristor, IEEE Elect. Dev. Lett. 39 (2018) 308–311.
[42] Z. Wang, et al., Memristors with diffusive dynamics as synaptic emulators for neuromorphic computing, Nat. Mater. 16 (2017) 101–108.
[43] T. Tsuruoka, et al., Effects of moisture on the switching characteristics of oxide-based, gapless-type atomic switches, Adv. Funct. Mater. 22 (2012) 70–77.
[44] I. Valov, et al., Atomically controlled electrochemical nucleation at superionic solid electrolyte surfaces, Nat. Mater. 11 (2012) 530–535.
[45] I. Valov, et al., Nanobatteries in redox-based resistive switches require extension of memristor theory, Nat. Commun. 4 (2013) 1771.
[46] F. Messerschmitt, M. Kubicek, J.L.M. Rupp, How does moisture affect the physical property of memristance for anionic-electronic resistive switching memories? Adv. Funct. Mater. 25 (2015) 5117–5125.
[47] I. Valov, W.D. Lu, Nanoscale electrochemistry using dielectric thin films as solid electrolytes, Nanoscale 8 (2016) 13828–13837.
[48] S.R. Ovshinsky, Reversible electrical switching phenomena in disordered structures, Phys. Rev. Lett. 21 (1968) 1450–1453.
[49] A.C. Warren, Reversible thermal breakdown as a switching mechanism in chalcogenide glasses, IEEE Trans. Elect. Dev. 20 (1973) 123–131.

[50] A. Redaelli, A. Pirovano, A. Benvenuti, A.L. Lacaita, Threshold switching and phase transition numerical models for phase change memory simulations, J. Appl. Phys. 103 (2008) 111101.

[51] D. Adler, H.K. Henisch, S.N. Mott, The mechanism of threshold switching in amorphous alloys, Rev. Mod. Phys. 50 (1978) 209–220.

[52] D. Ielmini, Y. Zhang, Analytical model for subthreshold conduction and threshold switching in chalcogenide-based memory devices, J. Appl. Phys. 102 (2007) 054517.

[53] D. Ielmini, Threshold switching mechanism by high-field energy gain in the hopping transport of chalcogenide glasses, Phys. Rev. B 78 (2008) 035308.

[54] L. Cario, C. Vaju, B. Corraze, V. Guiot, E. Janod, Electric-field-induced resistive switching in a family of Mott insulators: towards a new class of RRAM memories, Adv. Mater. 22 (2010) 5193–5197.

[55] N.F. Mott, L. Friedman, Metal-insulator transitions in VO_2, Ti_2O_3 and $Ti_{2-x}V_xO_3$, Philos. Mag. 30 (1974) 389–402.

[56] A. Parihar, M. Jerry, S. Datta and A. Raychowdhury, Stochastic IMT (insulator-metal-transition) neurons: an interplay of thermal and threshold noise at bifurcation, Front. Neurosci. 12 (2018). 210

[57] P.-Y. Chen, J.-s. Seo, Y. Cao, S. Yu, Compact oscillation neuron exploiting metal-insulator-transition for neuromorphic computing, Proceedings of the 35th International Conference on Computer-Aided Design, ACM, 2016, pp. 1–6.

[58] D. Lee, et al., NbO_2-based frequency storable coupled oscillators for associative memory application, IEEE J. Electron. Devices Soc. 6 (2018) 250–253.

[59] A. Pantazi, S. Wozniak, T. Tuma, E. Eleftheriou, All-memristive neuromorphic computing with level-tuned neurons, Nanotechnology 27 (2016) 355205.

Chapter 17

Synaptic realizations based on memristive devices

Valerio Milo[1], Thomas Dalgaty[2], Daniele Ielmini[1] and Elisa Vianello[2]
[1]*Department of Electronics, Information and Bioengineering, Polytechnic University of Milan and IU.NET, Milan, Italy,* [2]*University of Grenoble Alpes, CEA, LETI, Grenoble, France*

17.1 Introduction

Since the seminal works of Rosenblatt [1] and Minsky and Papert [2], the neural network has been recognized as a powerful machine learning model which, similar to a biological nervous system, demonstrates a certain level of *intelligence* on a technological substrate. Among many topologies, the deep neural network (DNN) has demonstrated the state-of-the-art performance in recognizing objects, images, and speech [3,4]. However, DNNs require extensive datasets and large amounts of energy during the training phase to parameterize the network model—often using backpropagation with gradient descent algorithm. Such a learning scheme can be seen simply as a mathematical method to improve the fitting of existing data by iteratively updating the synaptic weight between pairs of neurons, which lacks a biological analogy with real nervous systems. In contrast, spiking neural networks (SNNs) aim to reproduce some of these biological mechanisms with the aim of arriving at a learning machine that operates in a manner closer to biology and therefore with less data and using minimal power and largely relies on the exchange of spikes among neurons to process information [5]. This is the so-called *neuromorphic approach*, in which systems aim to incorporate the biologically relevant architecture, information coding, and the learning mechanisms of the human brain. In neuromorphic SNNs, spikes coordinate learning via Hebbian rules such as the spike-timing-dependent plasticity (STDP) and the spike-rate-dependent plasticity (SRDP).

To implement DNNs and SNNs in hardware circuits and systems, complementary metal-oxide semiconductor (CMOS) technology has been traditionally adopted in both digital and analog (or mixed) circuits [6,7]. CMOS circuits combine design flexibility, scalability, and the possibility to operate transistors in the power-efficient subthreshold regime. However, CMOS

Memristive Devices for Brain-Inspired Computing. DOI: https://doi.org/10.1016/B978-0-08-102782-0.00017-4

circuits lack a nonvolatile memory element capable of storing a synaptic weight in the desired long-term and multistate manner. Emerging nonvolatile memories, such as resistive switching random access memory (RRAM) [8], phase-change memory (PCM) [9], and spin-transfer torque magnetic random access memory (STT-MRAM) [10], instead naturally provide a more appropriate synaptic element required for hardware DNNs and SNNs. These types of memories are small, scalable and have a two-terminal resistive structure where the resistance can be modulated by the application of electrical pulses. These memory devices have been miniaturized to dimensions of approximately 10 nm in height and hundreds of nanometers in width [11]. In particular, RRAM and PCM also display analog switching behavior, where the resistance can be increased or decreased gradually by the application of suitable pulses [12]. Emerging memories can be easily implemented in CMOS circuits, with the help of the back-end-of-line (BEOL) integration techniques [13]. Furthermore, RRAM and PCM crossbars and matrices enable fast and energy-efficient in-memory computing [14] for free through the Ohm's law and Kirchhoff's laws [15–17] and the noniterative solution of linear algebra problems [18]. Given these multiple advantages from the physical, architectural, and scaling perspectives, nonvolatile resistive memories have been recognized as a promising technology to implement synaptic elements within high-density neuromorphic systems [19].

This chapter presents the hardware implementation of memristive synapses with biorealistic plasticity. First, biological plasticity is reviewed with reference to in vivo and in vitro experiments. Implementing such bioinspired plasticity rules in hardware is essential for realizing SNNs that emulate some of the cognitive functions of nervous systems such as pattern recognition, association, attention, and planning. Synaptic implementations are then discussed by describing RRAM synapses, PCM synapses, STT-MRAM synapses, and various hybrid structures combining one or more transistors with resistive devices to enable higher functionality and flexibility of the synaptic circuit. Nonoverlap, differential, 3D, and three-terminal synaptic transistor concepts are also presented to provide a comprehensive overview of various architectural approaches to STDP synapses. Triplet and SRDP learning synapses are also introduced with their applications in learning and filtering of spiking information. Finally, this chapter provides an overview of full-hardware implementations of SNNs for learning of patterns, thus further supporting the relevance of biological learning rules for enabling brain-inspired functions in silico.

17.2 Biological synaptic plasticity rules

The computational elements of nervous systems the neurons and synapses continuously adapt their properties for the purposes of homeostasis, short-term adaptation, and long-term changes for learning and memory formation.

This adaptation takes place by modifying the properties and number of ion channels on their cell membrane. These modifications result in changes of ion-channel efficacy and temporal dynamics of ion exchange. In the case of synapses, these modifications are usually abstracted to the idea of a change in a synaptic weight, which can be expressed as a function of the spike timing or spiking rate of the pre- and postsynaptic neurons. A body of literature uses this organizational perspective to derive *learning rules* that govern synaptic weight modification on the basis of data derived from biological experiments. Well-known learning algorithms are the STDP rule, which induces changes triggered by pairs of pre- and postsynaptic spikes, and the SRDP, where synaptic potentiation and depression are controlled by high- and low-presynaptic spike rates, respectively. The changes can be persistent for long-term plasticity or nonpersistent for short-term plasticity (STP). The following sections summarize the most common models of plasticity.

17.2.1 Long-term spike-timing-dependent plasticity and spike-rate-dependent plasticity

Changes in the synaptic weight are believed to encode the memory behavior and serve as the principal mechanism for learning in nervous systems. The most known STDP rule is the long-term plasticity induced by pairs of presynaptic and postsynaptic spikes, which was first experimentally observed in 1998 [20]. The changes in synaptic weight depend on the difference in spike timing between a pre- and a postsynaptic neuron and are persistent. The direction of the weight change depends on the polarity of this timing difference. The synaptic weight between two neurons increases for the case of the presynaptic neuron firing before the postsynaptic neuron, leading to the so-called long-term potentiation (LTP). On the other hand, the synaptic weight between two neurons decreases for the case of the presynaptic neuron firing after the postsynaptic neuron, leading to the so-called long-term depression (LTD) (Fig. 17.1). The weight change is higher when the spike time interval is short and it tends to zero for increasing spiking interval, which is consistent with the Hebb's postulate [21]. The dependence between the spike time interval and the weight change can be modeled as a piecewise function of two exponentials. Other shapes for the STDP characteristic have been observed, such as a symmetric dependence or anti-Hebbian plasticity where the time dependence is reversed compared to the classical time dependence [22–25]. In all these cases, the change in the synaptic weight depends on the relative timing of the pre- and postsynaptic spikes, which is the core principle of the pair-based STDP rule. However, pair-based STDP fails to replicate the results of richer experimentally observed biological features. In particular, it has been demonstrated that a triplet rule (i.e., a rule that considers sets of three spikes, two pre and one post or two post and one pre) is more biologically realistic [26].

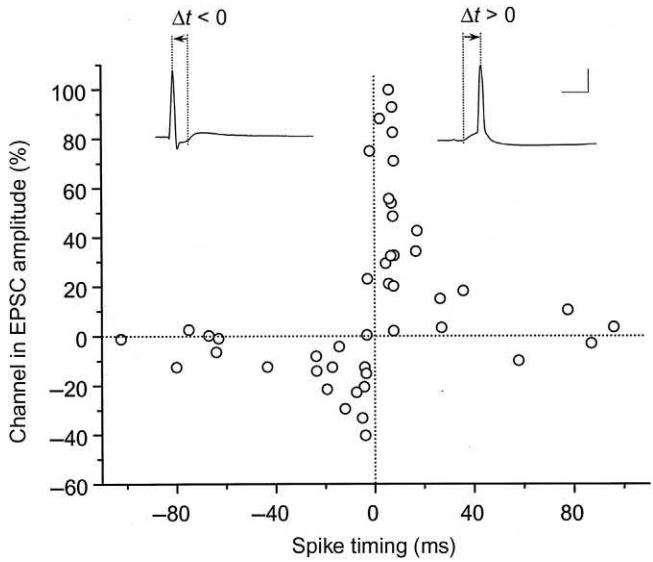


FIGURE 17.1 Experimentally observed pair-based STDP characteristics. *Reprinted from G.-Q. Bi, M.-M. Poo, Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and post synaptic cell type, J. Neurosci. 18 (24) (1998) 10464–10472, doi: 10.1523/JNEUROSCI.18-24-10464.1998. Copyright [1998] Society for Neuroscience.*

SRDP is another paradigm for implementing the Hebbian synaptic plasticity. The SRDP induction protocol is predominantly based on the neuronal firing rate to change the sign and magnitude of synaptic plasticity [27–29]. As observed in the hippocampus/neocortex, the postsynaptic terminations underwent LTP when the presynaptic neuron fired with a high frequency (20–100 Hz), while LTD was observed instead for low-frequency spiking (1–5 Hz). A simple and effective learning rule to implement SRDP, often called the Fusi rule, relies on the postsynaptic firing rate instead [27]. After a presynaptic pulse, the synapse can be depressed or potentiated depending on whether the postsynaptic membrane potential is low or high gated by an additional calcium variable, which is determined by the neurons firing rate. Synapse potentiation is inhibited when the calcium variable is above a certain threshold, while synapse depression is inhibited when the calcium variable is below another threshold.

17.2.2 Short-term plasticity

Long-term STDP and SRDP induce persistent synaptic weight changes. On the other hand, short-term nonpersistent synaptic weight changes can also take place after the synapse has propagated a spike. Following a presynaptic spike, the weight of the synapse can either transiently decrease (depression)

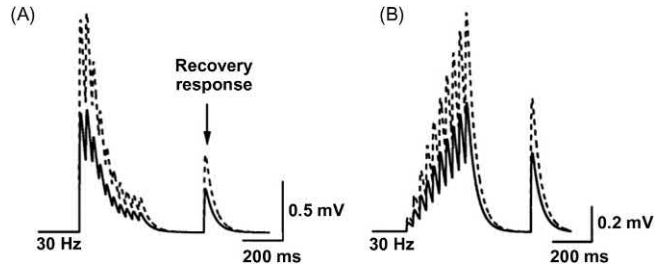


FIGURE 17.2 Experimentally observed short-term plasticity. (A) Example of short-term depression. (B) Example of short-term facilitation. *Adapted from H. Markram, D. Pikus, A. Gupta, M. Tsodyks, Potential for multiple mechanisms, phenomena and algorithms for synaptic plasticity at single synapses, Neuropharmacology 37(4–5) (1998) 489–500. Copyright 1998 Elsevier Science Ltd.*

or increase (facilitation), followed by a decayof the synaptic weight toward its baseline with time. As is the case for long-term plasticity, STP has been observed in biological experiments [30–32]. STP may result in either a depression when each presynaptic spike induces a decrease of the synaptic weight (Fig. 17.2A) or a facilitation when each presynaptic spike induces an increase of the synaptic weight (Fig. 17.2B). As the changes induced by STP only take effect during a short period and rapidly fade with time, it is not sufficient to cause stable learning. However, it has interesting properties that contribute to the efficiency of the neural network. One notion is that the short-term change behaves as a temporal filter of spiking trains. For instance, a synapse exhibiting short-term depression acts as a low-pass filter since the high-frequency presynaptic activity is attenuated in the synapse before it can excite the postsynaptic neuron. The contrary is true for short-term facilitation whereby only a high rate of presynaptic activity is sufficient to achieve a synapse strong enough to significantly excite a postsynaptic cell.

17.2.3 State-dependent synaptic modulation

Further synaptic temporary modulation can be induced by signals from neuromodulatory neurons dependent on the state of the animal [33]. An example can be found in the elementary motion detection system of drosophila where the neuromodulator octopamine tunes neuronal properties in the visual system as a function of whether the insect is resting or flying. This allows the insect to adapt its sensitivity to different velocities of stimulus as well as reduce power consumption while it is in a resting state. In Fig. 17.3 the response of Lobula Plate tangential cells, well-characterized neurons dedicated to the processing of optical flow, is reported for Drosophila stimulated with a moving grating when it is in resting and flying states. The area under the curve for the insect in its resting state is greatly reduced compared to

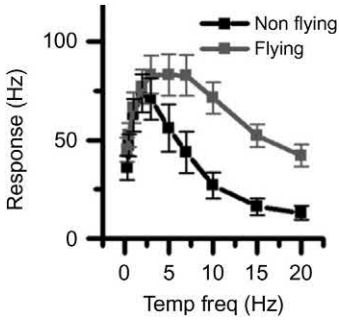


FIGURE 17.3 Temporal frequency sensitivity tuning curve of the mean response of Lobula Plate, a neuron dedicated to the processing of optic flow, in drosophila resting and flying states. *Adapted from S.N. Jung, A. Borst, J. Haag, Flight activity alters velocity tuning of fly motion-sensitive neurons. J. Neurosci. 31 (25) (2011) 9231–9237. Copyright [2011] Society for Neuroscience.*

that of its flying state which is thought to be an evolutionary adaptation to optimize the energy consumption.

17.3 Memristive implementations

To develop bioinspired neuromorphic hardware, the implementation of biological synaptic plasticity rules, such as STDP and SRDP, is essential. Indeed, a key enabling feature of neuromorphic circuits is their ability for learning and adaptation, which requires synaptic plasticity as in the human brain. This has thus led to a significant effort in the exploration of novel devices that could replicate bioinspired learning rules with simple algorithms, low-energy consumption, and high density of synaptic connections. To this purpose, memristive devices appear as a promising technology to emulate the synaptic behavior in artificial neural networks. In particular, strong interest was gained by a class of memristors including RRAM and PCM, also called first-order memristors [34], depicted in Fig. 17.4A. In this type of devices, STDP can be achieved solely by the application of overlapping spikes at device terminals as schematically depicted in Fig. 17.4B [35]. In addition to first-order memristors, another class of memristors, called second-order memristors (Fig. 17.4C), has been recently proposed [34], evidencing that resistive switching phenomena can be induced by nonoverlapping spikes applied across memristor device with variable positive/negative relative delay Δt (Fig. 17.4D). The nonoverlap resistance switching can be explained by the occurrence of short-term conductance changes controlled by second-order internal variables such as the internal temperature [35]. This feature is extremely important to implement at device-level significant processes such as the Ca^{2+} short-term dynamics [32], thus enabling the gradual weight update shown by biological STDP [20] and SRDP [27–29] with higher detail than the synaptic implementations with first-order memristors. Taking inspiration from these schemes, several hardware implementations of nanoscale synapses based on memristive materials capable of replicating

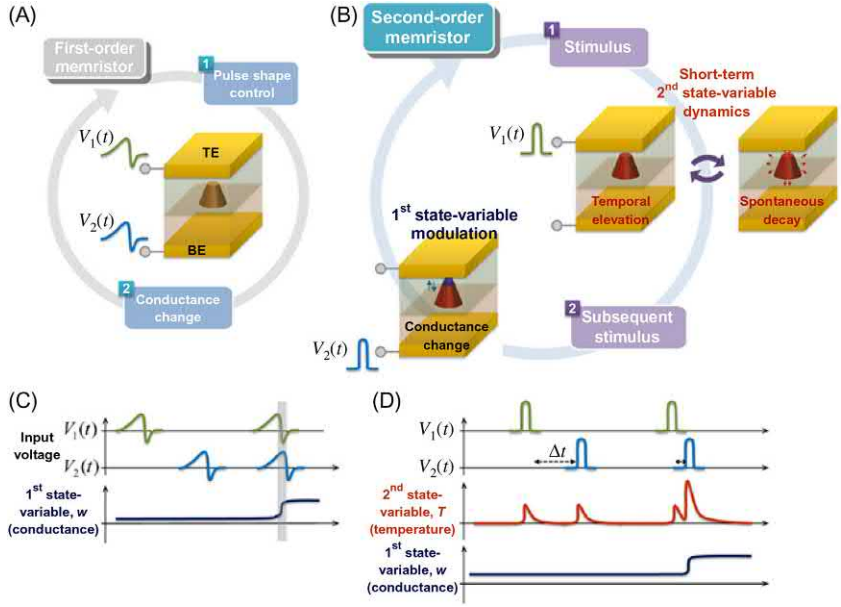


FIGURE 17.4 Comparison between (A) a first-order memristor where (B) only overlapping spikes applied at terminals can induce a conductance modification and (C) a second-order memristor where (D) the conductance can be changed depending on the sign and magnitude of relative timing of applied spikes thanks to second-order variables (e.g., temperature) displaying a short-term dynamics. *Reprinted with permission from S. Kim, C. Du, P. Sheridan, W. Ma, S.H. Choi, W.D. Lu, Experimental demonstration of a second-order memristor and its ability to biorealistically implement synaptic plasticity, Nano Lett. 15 (3) (2015) 2203–2211. Copyright (2015) American Chemical Society.*

synaptic plasticity have been developed and the most significant prototypes are discussed in the following section.

17.3.1 Resistive switching random access memory synapses

In the past decade, RRAM technology has been intensively investigated to design memristive synapses capable of STDP for biorealistic neuromorphic systems [35–43]. Indeed, RRAM combines low-voltage operation, large window, analog-type multilevel operation, good cycling endurance, and strong reliability [8].

Fig. 17.5A illustrates the ideal concept of the RRAM-based synapse, where a memory element within a high-density cross-point array can serve as the synaptic connection between artificial neurons, similar to the biological synapse in the brain [37]. Interestingly, both the biological synapse and the memristive RRAM rely on the ionic diffusion for the plasticity mechanism [43]. One of the earliest implementations of RRAM-based synapses

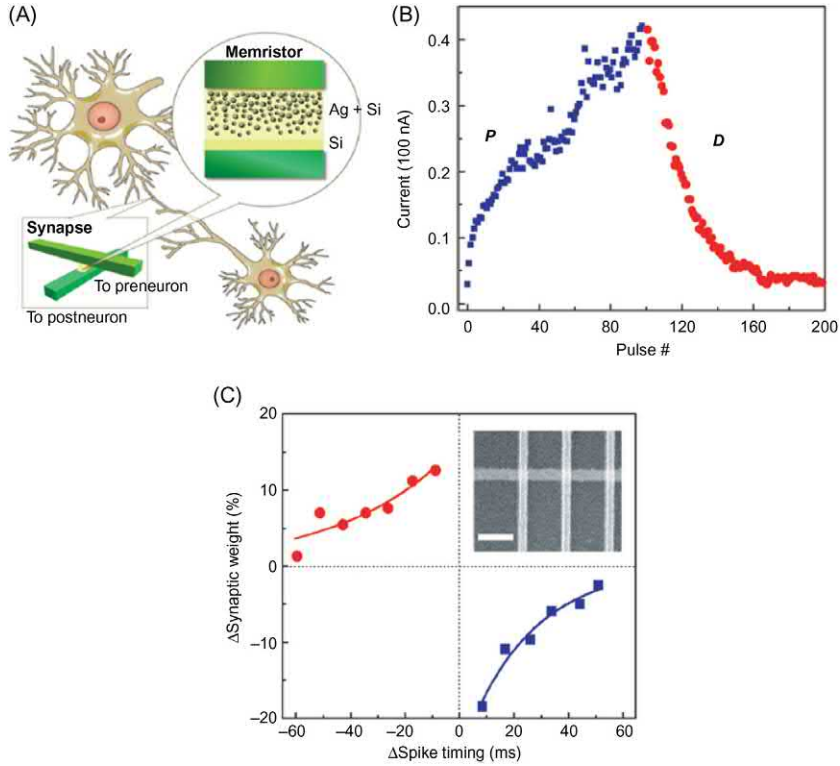


FIGURE 17.5 (A) Sketch of a synapse connection between a PRE and a POST neuron implemented by a memristor element. (B) Current response of Ag−Si RRAM device as a function of the number of applied pulses for both potentiation (current increase) and depression (current decrease). (C) STDP implementation for Ag−Si memristor at experimental and simulation level by application of PRE/POST spikes with variable time delay. *Adapted with permission from S.H. Jo, T. Chang, I. Ebong, B.B. Bhadviya, P. Mazumder, W. Lu, Nanoscale memristor device as synapse in neuromorphic systems, Nano Lett. 10 (4) (2010) 1297−1301. Copyright (2010) American Chemical Society.*

used a programmable metallization cell with an Ag-a/Si active layer where two regions with high- and low-Ag ion concentration, respectively, are formed by suitably setting the gradient of the Ag/Si mixture ratio [37]. Unlike memristors such as HfO_x or TiO_x-based RRAM, which sometimes exhibits abrupt resistive transitions due to the formation and rupture of a conductive filament, the resistance of this device can be tuned with analog precision by controlling the motion of Ag ions between Ag-rich and Ag-poor regions by application of an external voltage. To test the synaptic behavior of this device, a DC characterization study consisting of the application of two consecutive series of 100- and 300-μs-long pulses of amplitude 3.2 and −2.8 V, respectively, was performed. As a result, Fig. 17.5B shows the

incremental increase of the current during a first series of positive voltage pulses and the incremental decrease of current during the following series of negative voltage pulses, thus supporting the memristor capability of analog potentiation/depression at positive/negative bias. Based on the characterization study at device level, STDP measurements were carried out. To capture STDP characteristics by the Ag−Si RRAM device, a CMOS circuit was realized with two integrate-and-fire neurons connected through a RRAM memristor capable of mapping the relative time delay between occurrence times of PRE and POST spikes ($\Delta t = t_{PRE} - t_{POST}$) into the width of a pulse to be applied to synaptic device via a time-division multiplexing (TDM) scheme with globally synchronized time frames. According to this scheme, if the PRE spike precedes the POST spike, a potentiation pulse, with pulse width being an exponentially decaying function of Δt, is applied to the synapse. Otherwise, if the PRE spike follows the POST spike, a depression, pulse, with the pulse width being an exponentially decaying function of $|\Delta t|$, is applied to the device. Fig. 17.5C shows the resulting STDP characteristics obtained by measuring the percentage of synaptic weight update as a function of Δt which display an exponential decay of potentiation and depression in agreement with in vivo experimental data.

Although the results in Fig. 17.5 demonstrated the possibility to achieve STDP in silico for the first time, the TDM approach requires significant circuit complexity. To reduce the complexity of the STDP scheme, a direct overlap scheme was implemented using a bipolar RRAM device with one-resistor (1R) structure based on a TiN/HfO_x/AlO_x/Pt stack [38]. Fig. 17.6A shows the $I-V$ characteristics of the RRAM device with a relatively abrupt set transition and a more gradual reset transition whereas Fig. 17.6B shows the $I-V$ curves obtained by a continuous increase in the compliance current I_C from 1 μA to 200 μA, which allows to set the device at increasingly high conductance. Also, the application of a reset sweep with incremental maximum voltage $|V_{stop}|$ from −1.3 V to −3.3 V allows to reset the device at higher resistance. Therefore, the controllable set/reset operations support the multiple resistance states of the RRAM [44−47], enabling analog synaptic potentiation/depression via continuous set/reset processes. Fig. 17.6C further highlights the multilevel operation capability of the RRAM, showing that the measured resistance of synaptic RRAM device in response to the application of individual 50-ns-long positive/negative pulses with increasing amplitudes. Starting from an intermediate initial state between 200 and 300 kΩ, the device resistance can be gradually increased for pulse amplitudes varying from −2.4 V to −2.8 V, or the resistance can be gradually decreased for pulse amplitudes varying from 1.6 V to 2 V. The figure thus supports the ability to modulate the synaptic weight by applying short pulses of variable amplitude. The multilevel operation controlled by pulse amplitude was thus used as a basis to demonstrate the STDP learning rule at device level. To achieve this objective, PRE and POST spikes were properly designed via a

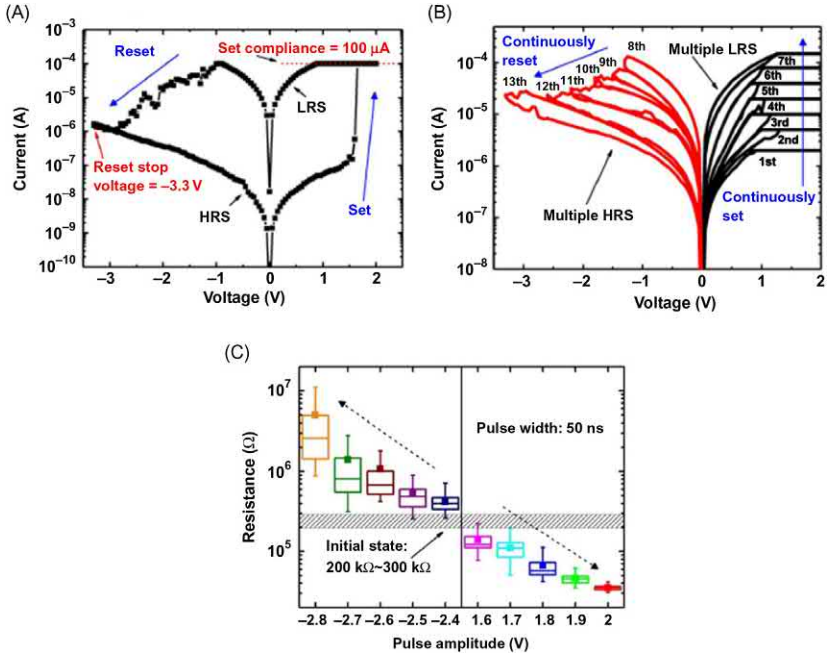


FIGURE 17.6 (A) Current−Voltage ($I-V$) characteristics of the HfO_x/AlO_x RRAM device with compliance current $I_C = 100\ \mu A$ and $V_{stop} = -3.3$ V. (B) $I-V$ characteristics for increasing I_C, which results in multiple LRS, and increasing V_{stop}, which leads to multiple HRS. (C) Resistance response for HfO_x/AlO_x RRAM device evidencing a gradual resistance decrease/increase for positive/negative pulses of increasing amplitude and fixed 50 ns duration. *Adapted with permission from S. Yu, Y. Wu, R. Jeyasingh, D. Kuzum, H.-S.P. Wong, An electronic synapse device based on metal oxide resistive switching memory for neuromorphic computation, IEEE Trans. Electron. Devices 58 (8) (2011) 2729−2737. Copyright 2011 IEEE.*

sequence of single pulses in consecutive timeslots, namely a negative pulse of period 1 μs followed by five positive pulses with identical period and decreasing amplitudes, such that only their overlap can effectively induce a synaptic weight modulation.

Fig. 17.7A shows the waveforms of two spikes that were devised such that if the relative timing between PRE and POST spikes, which is defined as $\Delta t = t_{post} - t_{pre}$ in this report, is positive, a single positive voltage pulse capable of triggering the set process is applied across the device causing potentiation, whereas if Δt is negative, a single negative voltage pulse capable of triggering the reset process is applied across the device causing depression. As a result of the application of this overlap approach, an analog STDP behavior more similar to the biological one was captured in simulation. The resulting STDP characteristic is shown in Fig. 17.7B, which supports the HfO_x/AlO_x RRAM and the overlap scheme as a promising approach for hardware neuromorphic systems capable of learning function.

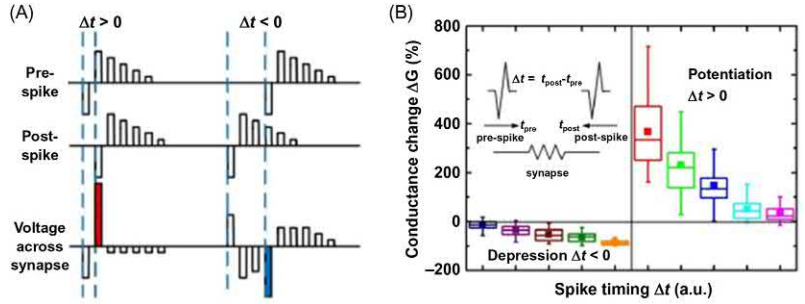


FIGURE 17.7 (A) Programming scheme based on the overlap of PRE and POST spikes to capture synaptic potentiation and depression according to STDP rule. (B) Calculated relative change in conductance as a function of the relative time delay between PRE and POST spikes suggesting the capability of HfO_x/AlO_x RRAM device of mimicking biological STDP rule. *Adapted with permission from S. Yu, Y. Wu, R. Jeyasingh, D. Kuzum, H.-S.P. Wong, An electronic synapse device based on metal oxide resistive switching memory for neuromorphic computation, IEEE Trans. Electron. Devices 58 (8) (2011) 2729–2737. Copyright 2011 IEEE.*

The engineering of pulse shape/width of PRE and POST spikes applied to memristor terminals plays a crucial role to achieve the memristor conductance modulation, hence the synaptic weight update. This is because the conductance changes at a given time in first-order memristors used in such synaptic structures [34] are solely governed by the voltage/current input applied to the device and the conductance state at that time. On the other hand, there is another class of memristors, referred to as second-order memristors [34], where the conductance is also controlled by one or more second-order state variables, which provide an additional degree of freedom to achieve the implementation of synaptic mechanisms increasingly similar to biorealistic processes.

In this regard, Kim et al. presented in [35] a second-order Ta_2O_{5-x}-based memristor device capable of replicating STDP rule with nonoverlapping spikes exploiting the short-term dynamics of internal temperature, which thus serves as second-order state variable making weight modulation timing dependent. To capture STDP, memristor device was presented with nonoverlapping PRE and POST spikes at two terminals (Fig. 17.8A) which, as evidenced in Fig. 17.8B, consist of two consecutive pulses with different amplitude and duration. In detail, PRE spike includes the sequence of a 20-ns-long programming pulse of amplitude 1.6 V followed, after a time interval of 1 μs, by a longer pulse of amplitude 0.7 V and width 1 μs for heat generation, whereas the POST spike coincides with PRE spike except for the amplitude of first pulse which is 1.1 V. The application of the PRE and POST spikes at top electrode (TE) and bottom electrode (BE), respectively, causes an overall voltage across device given by $V_{pre} - V_{post}$ which changes as shown in Fig. 17.8C depending on whether the PRE spike precedes the

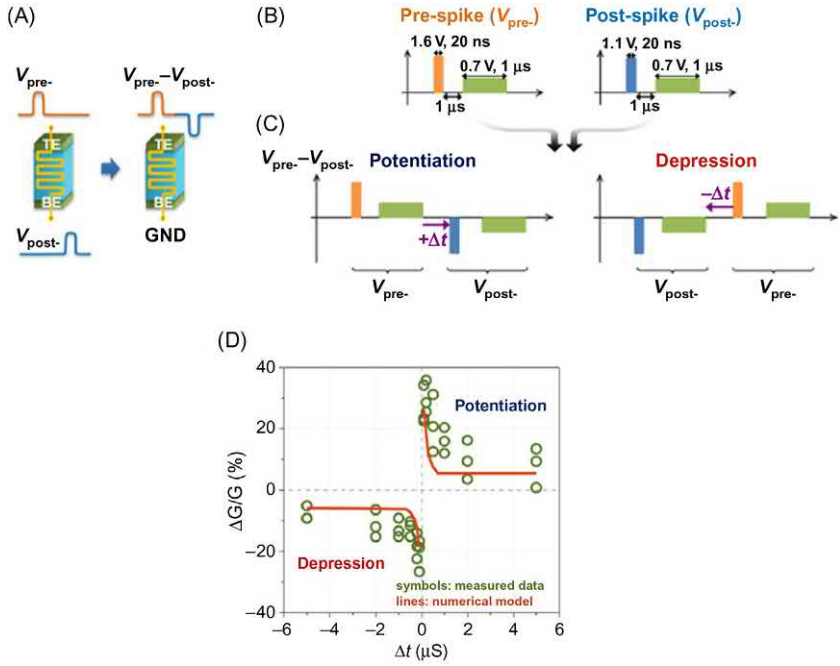


FIGURE 17.8 (A) Sketch of a memristive device where nonoverlapping voltage pulses are applied to the terminals. (B) The application of PRE and POST spikes, which consist of sequences of two positive pulses with different amplitude and width, at TE and BE, respectively, results in (C) a voltage $V_{PRE} - V_{POST}$ across the memristive element exhibiting two consecutive spikes with no overlaps capable of inducing a conductance change depending on the order of presentation (sign of Δt) and the short-term dynamics of internal temperature after pulse application (magnitude of Δt). (D) Experimental STDP characteristics achieved in a second-order Ta_2O_{5-x}-based memristor compared with a characteristic calculated by a numerical model. *Reprinted with permission from S. Kim, C. Du, P. Sheridan, W. Ma, S.H. Choi, W.D. Lu, Experimental demonstration of a second-order memristor and its ability to biorealistically implement synaptic plasticity, Nano Lett. 15 (3) (2015) 2203−2211. Copyright (2015) American Chemical Society.*

POST spike (left) or the PRE spike follows the POST spike (right). In the first case, which is featured by a positive time delay Δt between two spikes, the application of first spike (PRE spike) induces a temperature increase that affects the following spike (POST spike). The heat generated by the second spike is added to the temporal heating effect from the first spike causing a memristor conductance increase. Due to the elevated temperature, the second spike dominates the overall effect causing potentiation. On the contrary, for negative Δt, an identical mechanism based on short-term dynamics of internal temperature leads the memristor device to undergo a conductance decrease activated by second spike (PRE spike) which is higher than the conductance increase due to the first spike (POST spike), resulting in an overall

conductance decrease within memristor or synaptic depression. Importantly, note that in both cases the shorter/longer is Δt, the more/less pronounced the impact of Joule heating summation effect on memristor conductance upon the occurrence of the second spike, which thus results in an increasing/decreasing update of synaptic weight. As shown in Fig. 17.8D, this internal mechanism based on heat summation enables a second-order memristor to achieve at device level a very faithful replication of STDP characteristics observed in biological experiments where the relative change in conductance is a function of both the sign and the magnitude of Δt [20].

17.3.2 Phase-change memory synapses

In addition to RRAM technology, other novel nonvolatile memory devices have been investigated as potential candidates to build electronic synapses. Among various types of memristors, PCM has received a strong interest mainly for its high-resistance controllability via the gradual crystallization dynamics of chalcogenide-based active layer and the large resistance window ($\sim 10^3$) which is very suitable for efficient multilevel operation [48].

Similar to the approach described in [38] for RRAM synapses, a scheme based on the overlap between PRE and POST pulsed voltage signals at device terminals was designed in [49] to demonstrate STDP in single-element PCM-based synapses. As shown in Fig. 17.9A, POST signal consists of 8-ms-long negative pulse whereas PRE signal includes two sequences of 6 consecutive pulses of high- and low-positive voltages, respectively, separated by a zero period of 8 ms. In the first series, the pulses were designed with width of 50 ns, period of 10 ms, and linearly increasing amplitudes to achieve synaptic depression. On the other hand, the following series includes pulses that were designed with width of 1 μs, period of 10 ms, and linearly decreasing amplitudes to achieve synaptic potentiation. To validate this overlap scheme, relative time delays Δt of opposite signs between PRE and POST signals were applied by shifting the POST spike relative to the PRE spike. While Fig. 17.9A depicts the case for $\Delta t = 0$, Fig. 17.9B shows the overlapping spikes for a positive delay ($\Delta t = 20$ ms) evidencing that the net voltage across the synaptic device given by $V_{pre} - V_{post}$ crosses the minimum voltage threshold, thus leading to the increase of synaptic weight. Otherwise, if the relative delay is negative ($\Delta t = -40$ ms), the voltage subtraction across PCM results in a single pulse of amplitude higher than the minimum voltage threshold, thus activating depression process (Fig. 17.9C). Based on these particular cases, the application of variable delay values ranging from -40 ms to 40 ms allowed Kuzum et al. to achieve STDP capability at device level. This is confirmed by STDP measurements shown in Fig. 17.9D where the resulting STDP curve exhibits a nice agreement with biological data presented in [20]. This approach also offers great flexibility enabling to tune the time constant of measured STDP characteristics by changing the

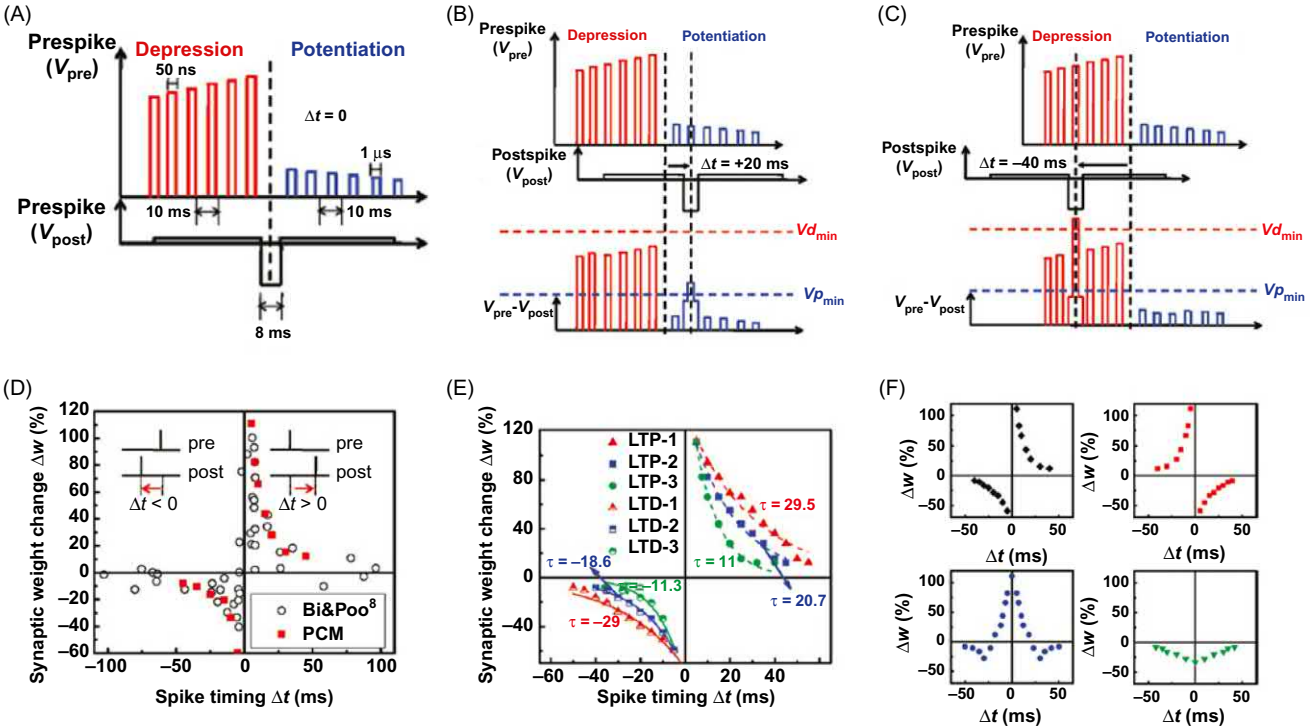


FIGURE 17.9 (A) Programming scheme based on the overlap between pulses within PRE and POST spikes that is adopted to implement STDP in a PCM synaptic device. (B) If relative delay is positive ($\Delta t = 20$ ms), spike overlap results in a voltage drop $V_{pre} - V_{post}$ across the PCM cell where a single 1-μs-long pulse can cross the set threshold, thus inducing synaptic potentiation. (C) On the contrary, if relative delay is negative ($\Delta t = -40$ ms), spike overlap results in a voltage drop $V_{pre} - V_{post}$ across PCM cell where a single 50-ns-long pulse can overcome the reset threshold, thus leading to synaptic depression. (D) STDP characteristics achieved by application of the programming scheme on PCM cell against experimental data collected in [20]. (E) Measured STDP curves for variable time constants τ obtained tuning the pulse amplitude/width within the programming scheme. (F) Various asymmetric and symmetric STDP characteristics that can be implemented at device level changing the order of pulse sequences. *Adapted with permission from D. Kuzum, R.G. D. Jeyasingh, B. Lee, H.-S.P. Wong, Nanoelectronic programmable synapses based on phase change materials for brain-inspired computing, Nano Lett. 12 (5) (2012) 2179−2186. Copyright (2012) American Chemical Society.*

amplitude and separation of pulses within PRE spike. Specifically, gradually decreasing the spacing between consecutive pulses such that the highest pulses within each PRE sequence are the closest ones allows to reduce the time constants of STDP exponential curves, which are significant biological parameters marking synapses in the brain. The application of this scheme thus leads to the implementation of measured STDP characteristics for variable time constants $|\tau|$ in the range 10 ms−30 ms shown in Fig. 17.9E, which supports the capability of PCM synapses of emulating various types of synapses with different biological functions. Finally, as shown in Fig. 17.9F, the modulation of the order of pulses within the PRE spike for synaptic potentiation and depression was also tested allowing to demonstrate two asymmetric STDP kernels and two symmetric STDP kernels, thus paving the way to the opportunity to build neuromorphic systems based on nanoscale memristive synapses increasingly approaching the complex operation of human brain.

17.3.3 Spin-transfer torque magnetic random access memory synapses

While RRAM and PCM have received more investigation, STT-MRAM has also been explored as potential synaptic technology in recent years [50−52]. Fig. 17.10A shows the structure of a perpendicular magnetic tunnel junction (pMTJ) based on dual-MgO/CoFeB interfaces and a Co/Pt multilayer synthetic antiferromagnetic (SAF) pinned layer that was experimentally investigated in [52] to demonstrate neuron and synaptic functionality at the level of individual memristive device. In this work, it has been demonstrated that the

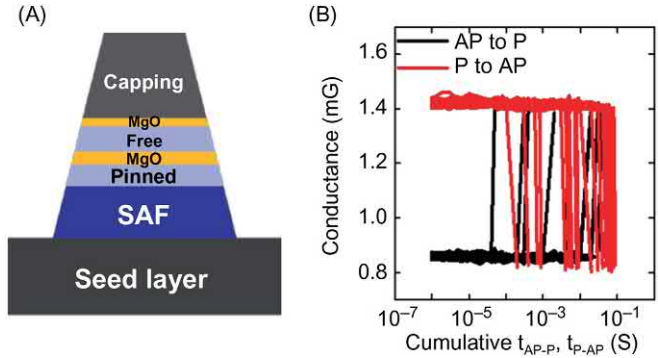


FIGURE 17.10 (A) Sketch of an STT-MRAM device structure including dual-MgO/CoFeB interfaces and a Co/Pt multilayer synthetic antiferromagnetic (SAF) pinned layer. (B) Conductance response of STT-MRAM device as operated at currents below a certain threshold supporting its use as stochastic binary synapse. *Adapted with permission from M.-H. Wu, M.-C. Hong, C.-C. Chang, P. Sahu, J.-H. Wei, H.-Y. Lee, et al., Extremely compact integrate-and-fire STT-MRAM neuron: a pathway toward all-spin artificial deep neural network. IEEE Symposium VLSI Technol. (VLSI Technol.). (2019) T34−T35. Copyright 2019 IEEE.*

adoption of a tailored pinned layer into an STT-MRAM device structure leads it to exhibit a highly asymmetric characteristic and unexpected spikes into its conductance response as a constant bias above a certain current threshold is applied. This behavior has been explained in [52] as a result of the stochastic oscillation between the parallel (P) and the anti-parallel (AP) state of magnetic polarization within the STT-MRAM free-layer induced by a back-hopping mechanism assisted by a field-like torque, and was exploited to implement a novel current-driven integrate-and-fire spiking neuron at single device level. Moreover as shown in Fig. 17.10B, this memristive device was demonstrated to display a stochastic but bistable switching behavior if operated at currents lower than the threshold, which supports its application not only as neuron but also as stochastic binary synapse, thus opening the way for the building of full all-spin-based neuromorphic networks in hardware.

17.4 Hybrid complementary metal-oxide semiconductor/memristive synapses

17.4.1 One-transistor/one-resistor synapses

Although single-element memristive synapses offer the opportunity to build extremely dense neuromorphic circuits, their use in crossbar arrays however can lead to significant concerns such as leakage currents due to sneak paths and high power consumption caused by the lack of current limiters. To bypass these issues while keeping relatively high integration density, a technological solution extensively adopted in recent years has been the use of a field-effect transistor (FET) in series to the memristor device, which led to the development of hybrid CMOS/memristive synaptic structures such as the one-transistor/one-resistor (1T1R) structure [53–60].

Fig. 17.11 shows (A) a 1T1R structure based on a serial connection of a FET to a Ti/HfO_x/TiN RRAM cell and (B) its $I-V$ characteristic, which clearly displays the current limitation to $I_C = 50\,\mu A$ during set transition achieved by FET. To operate this structure as an electronic synapse, the circuit scheme of Fig. 17.11C can be adopted [59]. According to this implementation, the PRE drives the gate terminal of FET, thus enabling synapse activation only as PRE spike occurs, whereas the POST controls the TE voltage V_{TE} which is generally set at low constant voltage to allow for communication between PRE and POST via the synapse. In this phase, the application of a PRE spike at the FET gate when the TE is biased at communication voltage induces a current proportional to the synaptic conductance across the device which is collected along with all the currents triggered by other activated PREs at the input of the POST. Then, the sum of these currents is integrated by the POST causing an increase in its internal potential until it exceeds a threshold, eventually leading to the emission of a fire spike by

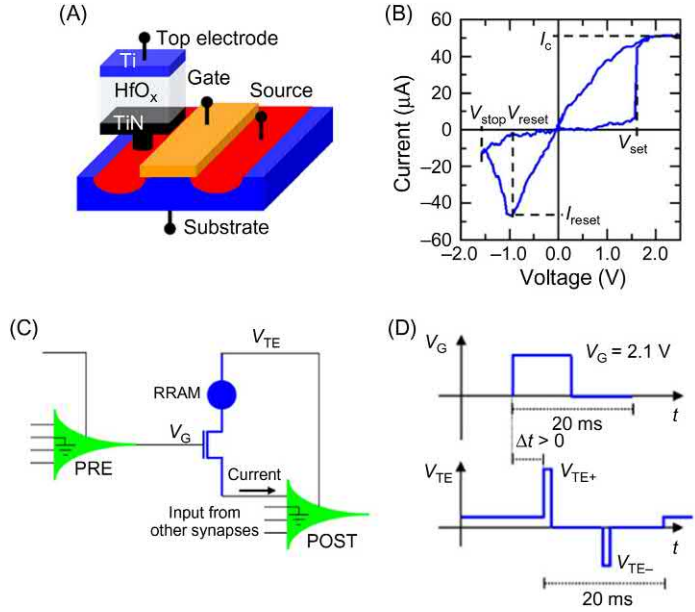


FIGURE 17.11 (A) Sketch of a 1T1R cell based on a Ti/HfO$_x$/TiN RRAM device. (B) $I-V$ characteristic of 1T1R RRAM structure. (C) Fundamental block using 1T1R cell as synaptic element connecting PRE neuron with POST neuron. (D) Programming strategy used to capture potentiation in 1T1R synapse: as Δt is positive, only the positive pulse within POST spike applied to the TE can overlap with the PRE spike applied to the gate terminal, thus activating a set transition, hence a weight change from HRS to LRS. *Adapted with permission from S. Ambrogio, S. Balatti, V. Milo, R. Carboni, Z. Wang, A. Calderoni, et al. Neuromorphic learning and recognition with one-transistor-one-resistor synapses and bistable metal oxide RRAM. IEEE Trans. Electron. Devices 63 (4) (2016) 1508–1515. Copyright 2016 IEEE.*

POST which is delivered at TE to update the synaptic weight according to the STDP rule. If the relative delay Δt between the PRE spike, which was designed as a 10-ms-long pulse of amplitude 2.1 V followed by a zero period of 10 ms, and the POST spike, which was designed as a 1-ms-long positive pulse followed by 1-ms-long negative pulse after a zero period of 10 ms, is positive, only the short positive pulse of amplitude $V_{TE+} > V_{set}$ within POST spike overlaps with PRE spike, thus inducing a set transition in the RRAM cell resulting in the potentiation of synaptic weight (Fig. 17.11D). Otherwise, if Δt is negative, only the short negative pulse of amplitude $V_{TE-} < V_{reset}$ in the POST spike takes place at TE during PRE spike, thus causing a reset transition in the RRAM cell leading to depression of synaptic weight.

This synaptic operation scheme was validated by the experimental measurements shown in Fig. 17.12A demonstrating the relative change of conductance in a single 1T1R synapse as a function of Δt for variable initial state from the full LRS ($R_0 = 25\ \text{k}\Omega$) and full HRS ($R_0 = 500\ \text{k}\Omega$) [59].

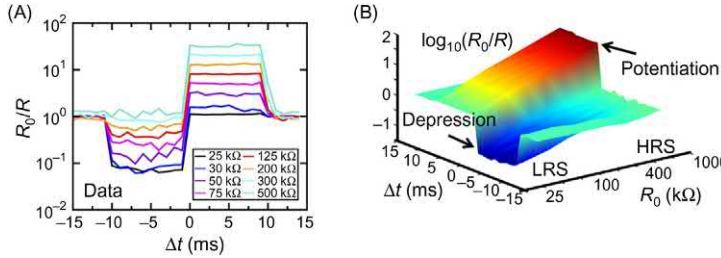


FIGURE 17.12 (A) Measured STDP characteristics achieved in the 1T1R HfO_x RRAM device for variable initial state from HRS to LRS. (B) Color plot of experimental STDP implemented in the 1T1R RRAM cell. *Adapted with permission from (A) S. Ambrogio, S. Balatti, V. Milo, R. Carboni, Z. Wang, A. Calderoni, et al., Neuromorphic learning and recognition with one-transistor-one-resistor synapses and bistable metal oxide RRAM, IEEE Trans. Electron. Devices 63(4) (2016) 1508–1515. Copyright 2016 IEEE; (B) V. Milo, G. Pedretti, R. Carboni, A. Calderoni, N. Ramaswamy, S. Ambrogio, et al., Demonstration of hybrid CMOS/RRAM neural networks with spike time/rate-dependent plasticity, in: IEEE Int. Electron. Devices Meet. (IEDM), 2016, pp. 440–443, doi: 10.1109/IEDM.2016.7838435 Copyright 2016 IEEE.*

These characteristics first show that the more resistive the initial state, the higher the weight change via potentiation event, and the less resistive the initial state, the higher the weight change via depression event. Also, note that although the measured STDP characteristics show the synaptic potentiation/depression for positive/negative delays as expected by STDP biological protocol, their behavior is however uniform within the overlap window of $|\Delta t| < 10$ ms for any initialization because of binary operation of RRAM device. The positive pulse at TE always leads the device to the full LRS set by I_C (controlled via V_G) whereas the negative pulse always leads the device to the full HRS, irrespective of Δt. This is also confirmed by the color plot of measured STDP characteristics shown in Fig. 17.12B where the maximum potentiation for positive Δt is achieved starting from HRS whereas the maximum depression for negative Δt is obtained as the initial state is programmed in LRS [60].

In addition to the 1T1R RRAM synapses, 1T1R synaptic structures including a PCM cell as memristive element have also been investigated [53,55,58]. In this frame, Bichler et al. devised the so-called 2-PCM synapse shown in Fig. 17.13A, which is capable of implementing potentiation and depression by two 1T1R PCM structures referred to as LTP cell and LTD cell, respectively, using chalcogenide crystallization process in both cases [55]. In this way, a significant power saving due to the non-use of reset pulses at high current (hundreds of microampere) for depression phase can be achieved. Also, since the progressive crystallization of chalcogenide active layer is carried out by application of sequences of voltage pulses with the same amplitude, pulse generation is easier than the scheme adopted in [49]. In terms of functionality, this synaptic structure was used to capture the

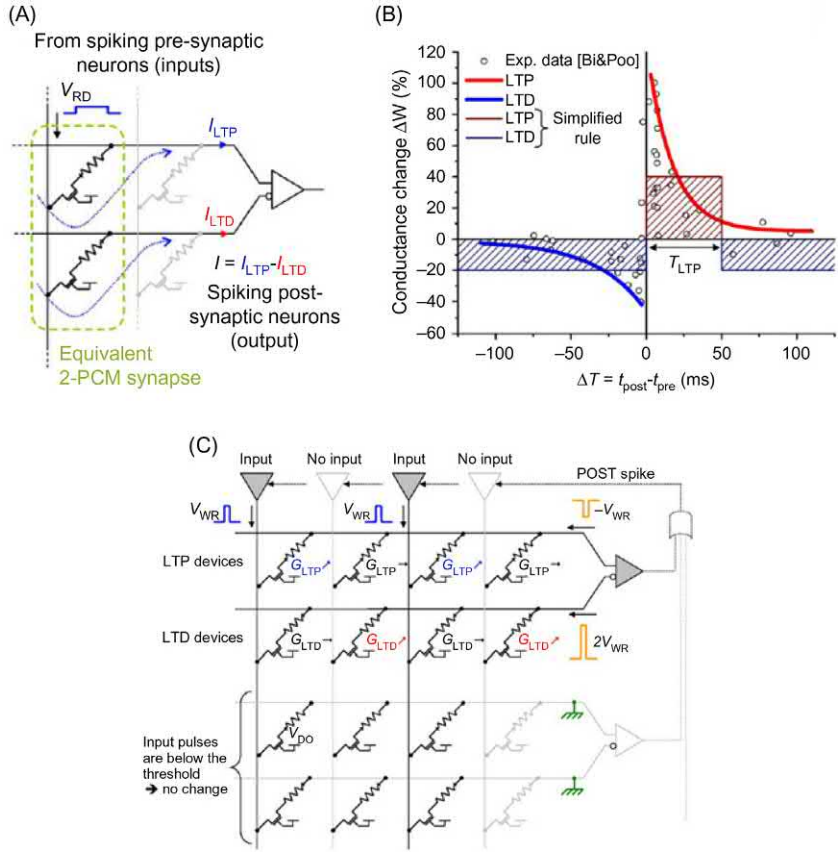


FIGURE 17.13 (A) Schematic representation of 2-PCM synapse whose weight is given by the conductance difference between LTP device and LTD device. (B) STDP learning rule implemented by 2-PCM synapse against biological STDP. (C) Programming algorithm used to implement potentiation and depression in 2-PCM synapses according to simplified STDP rule shown in (B). *Reprinted with permission from O. Bichler, M. Suri, D. Querlioz, D. Vuillaume, B. De Salvo, C. Gamrat, Visual pattern extraction using energy-efficient '2-PCM synapse' neuromorphic architecture, IEEE Trans. Electron. Devices 59 (8) (2012) 2206–2214. Copyright 2012 IEEE.*

simplified STDP characteristics shown in Fig. 17.13B, according to which synaptic potentiation can occur only for a specific range of positive time delays between PRE and POST spikes of length T_{LTP}. In particular, to demonstrate this weight update rule, the pulse scheme for write operations schematically described in Fig. 17.13C was designed. Each time an input neuron is activated, it enters or re-enters an LTP internal state for the duration of the LTP window (T_{LTP}). When an output neuron fires, it transmits a POST spike signal to every input neuron, signaling write operation. In write operation, input neurons generate a positive pulse of amplitude V_{WR} such that

$V_{WR} < V_{set} < 2V_{WR}$, only if they are in the LTP window. The output neuron delivers at the same time voltage pulses of amplitude $-V_{WR}$ and $2V_{WR}$ at BEs of LTP PCM cells and LTD PCM cells, respectively. When an input neuron pulse interacts with the output neuron pulses, the effective voltage across the LTP device is $2V_{WR} > V_{set}$, and the voltage across the LTD device is $V_{WR} < V_{set}$. As a result, the conductance of LTP cell is increased, while the corresponding LTD cell remains unchanged (synaptic potentiation). If input neurons do not generate pulses, namely in all the cases with time delays outside LTP window, $-V_{WR}$ drops on the LTP cell and $2V_{WR}$ on the LTD cell, which results in a conductance increase for LTD cell with unchanged conductance of LTP cell (synaptic depression). The application of this plasticity scheme requires the execution of an additional refresh operation whenever the conductance of one of the two cells within 2-PCM synapses saturates to the full LRS, which consists of a reinitialization in HRS of both devices followed by the application of a series of set pulses to the LTP cell to restore the effective synaptic weight. To overcome this limitation, it has been experimentally demonstrated in [61] that an adaptation of the PCM structure—narrow heater bottom electrode-based phase-change memory—associated with short programming pulses (<50 ns) can naturally implement gradual LTP and depression. This solution allows for replacing the 2-PCM synapse with a 1T1R structure. Since the number of conductance levels that can be achieved during LTD (gradual amorphization) is lower than the number of conductance levels that can be achieved during LTP (gradual crystallization), the 1T1R PCM structure suffers from a small reduction of the recognition rate with respect to the 2-PCM (76% vs 80% for classification on the MNIST database).

17.4.2 Two-transistor/one-resistor synapses

Although very compact 1T1R synapses have been demonstrated to be capable of achieving neuromorphic applications such as visual pattern recognition via simplified STDP learning rules [53–60], more complex architectures have been developed to gain higher flexibility and more detail in the emulation of biological processes. To this end, a hybrid CMOS/memristive synaptic structure called two-transistor/one-resistor (2T1R) structure has been recently proposed using both RRAM device [62] and PCM device [63].

Fig. 17.14A shows a 2T1R synaptic structure based on the serial connection of a TiN/HfO_x/TiN RRAM device and two transistors arranged into a parallel configuration where the gate terminal of the left transistor, called communication gate (CG), and the TE of RRAM device are controlled by the PRE circuit, while the gate terminal of the right transistor, called fire gate (FG), and the BE of RRAM device are driven by POST circuit [62]. In this architecture, the communication phase between PRE and POST is enabled through the application by PRE of a short-voltage pulse V_{CG} at CG

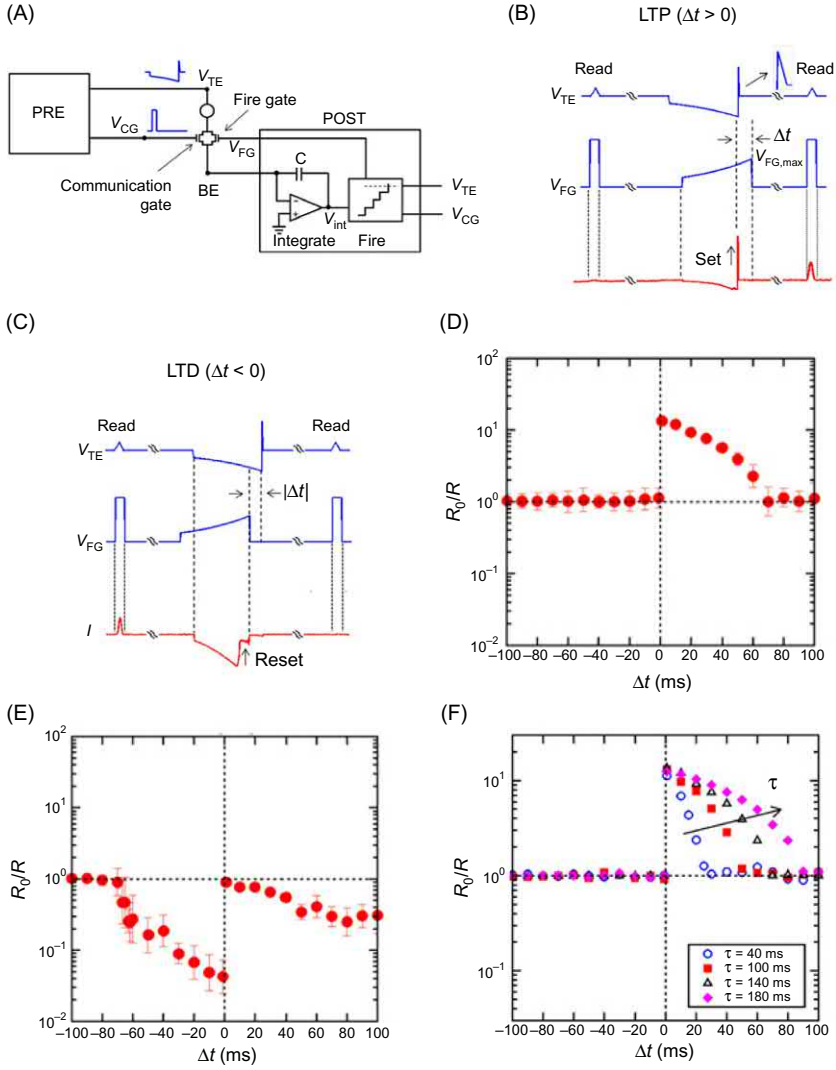


FIGURE 17.14 (A) Schematic representation of 2T1R RRAM synapse in PRE-synapse-POST circuit including voltage signals applied to synaptic terminals during the communication phase. Overlap between TE voltage and FG voltage triggering (B) set transition for RRAM device, hence potentiation for 2T1R synapse in case of positive Δt, and (C) reset transition for RRAM device, hence depression for 2T1R synapse, in case of negative Δt. STDP characteristics achieved by 2T1R RRAM structure for (D) potentiation and (E) depression, which can also occur for high positive Δt. (F) STDP characteristics under potentiation mode for variable time constant τ of FG pulse. *Adapted from Z.-Q. Wang, S. Ambrogio, S. Balatti, D. Ielmini, A 2-transistor/1-resistor artificial synapse capable of communication and stochastic learning for neuromorphic systems, Front. Neurosci. 8 (2015) 438.*

and a voltage signal consisting of an exponentially increasing negative pulse followed by a short positive pulse at TE, whose overlap results in the activation of a current proportional to the device conductance at that time. This current spike is transmitted to POST via BE and then integrated by the integrate-and-fire stage by causing an increase in the POST internal potential V_{int} which, after a certain number of PRE events, reaches a threshold leading to the generation of a fire spike.

As the fire spike sent by POST at FG overlaps with the PRE spike at TE, the following synaptic operation phase called plasticity phase takes place, enabling to modulate the weight of 2T1R synapse via potentiation and depression processes based on the STDP rule. Specifically, synaptic potentiation is achieved if the PRE voltage spike applied to the TE, which consists of the sequence of a negative 150-ms-long exponential pulse and a very short (1 ms) positive pulse (top), precedes ($\Delta t > 0$) the truncated positive exponential POST pulse applied to FG (center), in that their superposition results in a very sharp current increase (bottom) inducing set transition of RRAM device (Fig. 17.14B). On the contrary, as described in Fig. 17.14C, if the POST spike precedes the PRE spike ($\Delta t < 0$), their overlap causes a reset transition within RRAM device leading to the depression of 2T1R synapse. Applying the PRE and POST spikes at the 2T1R synapse with continuous change of Δt from -100 to 100 ms, its ability to capture biorealistic analog behavior of potentiation and depression according to STDP was experimentally validated as evidenced by measured characteristics shown in Fig. 17.14D and E, respectively. In particular, note that a weak synaptic depression can also be obtained for very large positive Δt as a result of competition between the two synaptic processes. Importantly this structure also offers an additional degree of freedom compared to 1T1R configuration namely the opportunity to change both potentiation characteristics (Fig. 17.14F) and depression characteristics (data not shown) by proper tuning of time constant τ of FG voltage spike, which can serve as useful tool to replicate other biological phenomena.

Fig. 17.15A shows an alternative 2T1R synapse using a PCM cell as memristive element [63]. Here PCM cell is connected to the intermediate node between two transistors, called LIF transistor (top) and STDP transistor (bottom), respectively. This structure is connected to the PRE by the gate terminals of the LIF and STDP transistors, and to the POST by the drain terminal of LIF and the BE of the PCM device. Similar to the 2T1R RRAM synapse [62], two distinct paths were designed to achieve communication (LIF) and plasticity (STDP) operation modes, respectively. During LIF phase, which is explained in Fig. 17.15B, upon PRE spike, the LIF WL pulse generator included in the PRE circuit enables LIF transistor with STDP transistor turned off, leading to the discharge of the capacitor of leaky-integrate-and-fire POST circuit as long as the voltage across the capacitor V_{cap} decreases below V_{th}. At that point POST fires, by activating after a time

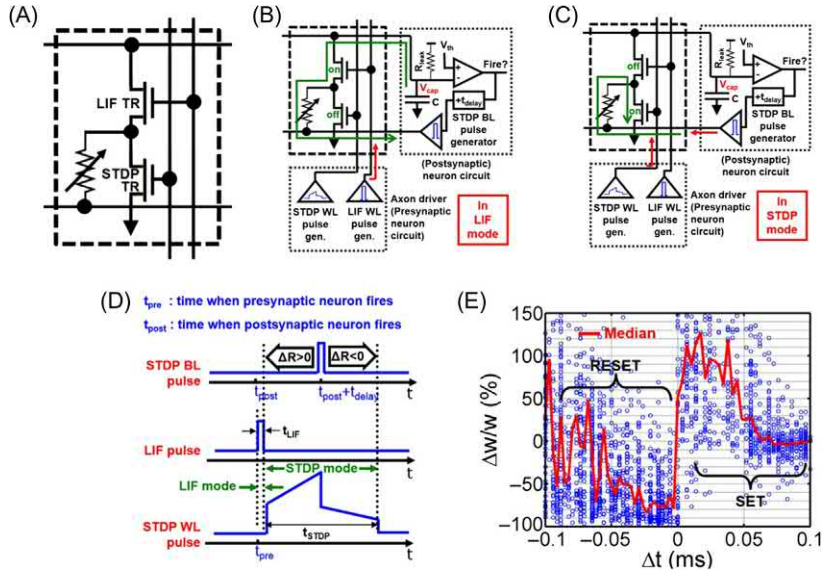


FIGURE 17.15 (A) Scheme of the 2T1R PCM synapse where a transistor is used for leaky-integrate-and-fire phase (LIF TR) whereas the other one for weight update phase (STDP TR). Schematic representation of 2T1R synapse operation during (B) LIF mode and (C) STDP mode. (D) Programming strategy used in 2T1R PCM synapse circuit to achieve potentiation and depression depending on timing of overlapping STDP BL pulse and STDP WL pulse. (E) Measured STDP characteristics demonstrated via 2T1R PCM synapse. *Adapted with permission from S. Kim, M. Ishii, S. Lewis, T. Perri, M. BrightSky, W. Kim, et al., NVM neuromorphic core with 64k-cell (256-by-256) phase change memory synaptic array with on-chip neuron circuits for continuous in-situ learning, IEEE Int. Electron. Devices Meet. (IEDM) (2015) 443–446. Copyright 2015 IEEE.*

delay t_{delay} the STDP BL pulse generator that delivers a short positive pulse at the BE of PCM cell. After LIF mode, the PRE circuit disables LIF transistor and activates the STDP transistor via a slowly varying voltage signal emitted by STDP WL pulse generator, thus leading 2T1R synapse in STDP mode (Fig. 17.15C). In STDP mode 2T1R synapse can update its weight according to the STDP rule through the overlap of STDP BL pulse and STDP WL pulse. As shown in Fig. 17.15D, depression ($\Delta R > 0$) can be achieved for $t_{\text{PRE}} > t_{\text{POST}}$, namely as STDP BL pulse overlaps with increasing part of STDP WL signal since it induces high current programming the PCM cell into HRS. Otherwise, potentiation ($\Delta R < 0$) can be achieved for $t_{\text{PRE}} < t_{\text{POST}}$, since in this case the overlap of STDP BL pulse and decreasing part of STDP WL signal results in a lower current, leading the PCM device to be programmed to the LRS. Most importantly, this 2T1R synaptic implementation allows to capture the gradual nature of potentiation and depression dynamics via the properly designed STDP WL signal. This is confirmed by measured relative weight change as a function of Δt shown in Fig. 17.15E,

which supports 2T1R PCM synapse as a valuable electronic synapse for neuromorphic applications.

17.4.3 Differential synapses

As already discussed in Sections 17.4.1 and 17.4.2, the use of memristive devices such as RRAM and PCM in hybrid synaptic architectures involves a certain overhead in terms of complexity of structure and algorithm to capture biological behavior. First, these circuits require long overlapping spikes at PRE and POST terminals to trigger weight updates via atom configuration modifications, which results in a significant reduction of data throughput in large-scale neuromorphic networks. In addition to this, write operation of memristive devices governed by spike-based algorithms can require high-programming currents, which has a detrimental impact on power consumption and circuit size [64]. To tackle these issues featuring the majority of recently developed hybrid CMOS/memristive synapses, a novel memristive-based synaptic circuit, referred to as differential memristive synapse, was proposed in [64]. This hybrid synapse architecture, which is based on two memristors and 20 transistors, stores the synaptic weight within the difference of the conductances of two memristive devices where one of them provides a positive term whereas the other one a negative term. Specifically, increasing positive conductance with simultaneous reduction of negative conductance leads the synapse to undergo potentiation whereas the reduction of positive conductance with simultaneous increase in the negative conductance results in the synaptic depression. Nair et al. first demonstrated that their differential synapse operated in read mode allows to downscale the currents flowing in the memristors into currents of the range of pA, thus enabling the POST circuit to integrate very small currents. This results in the opportunity to build more compact and energy-efficient POST circuits. Also it was validated that the combination of differential operation mode and normalizing capability in this synaptic architecture allows to be more insensitive to memristor variability and to increase the dynamic range of weights by implementation of inhibitory and excitatory weights. Finally, most importantly, it seems very suitable to faithfully implement biorealistic weight update schemes thanks to the ability to operate by using nonoverlapping spikes at synaptic terminals, which was tested via learning simulations at network level by achieving significant performance in single pattern binary classification and multipattern classification [64].

17.4.4 Multimemristive synapses

As already discussed, the ideal RRAM synapse presents an analog conductance modulation under identical pulses in both programming directions: the conductance gradually increases (decreases) when a train of set (reset)

pulses is applied. However, in most RRAM and PCM devices developed for standard memory applications it is difficult to achieve both potentiation and depression in a gradual manner. In RRAM devices the set operation is usually noncumulative: the switch from the high to the low resistance state is abrupt [38]. In PCM memories, the amorphization process tends to be abrupt unlike synaptic depression [55]. Clearly, advances in materials science and device technology could play a key role in addressing some of these challenges [61], but equally important are innovations in the synaptic architectures. One example is the 2-PCM synapse presented in [55] (Section 17.4.1). However, the need to refresh the device conductance frequently to avoid conductance saturation could potentially limit the applicability of this approach. In another approach proposed, several binary memristive devices are programmed and read in parallel to implement a synaptic element, exploiting the probabilistic switching exhibited by certain types of memristive devices [57]. Since it may be challenging to achieve fine-tuned probabilistic switching reliably across a large number of devices, pseudo-random number generators could be used to implement this probabilistic update scheme with deterministic memristive devices. Moreover, a multimemristive synaptic architecture with an efficient global counter-based arbitration scheme has been proposed in [65]. A tradeoff associated with the proposed hybrid approach is the silicon area consumption for each synapse proportional to the number of devices needed to obtain a multilevel behavior. A possible solution to overcome this problem is the adoption of the Vertical RRAM (VRRAM) technology, which consists of RRAM cells integrated in multilayered VNAND−like structure and uses simple and cost-effective 3D processes to achieve high memory density [41,66,67]. Fig. 17.16A shows the vertical RRAM (VRRAM) structure presented in [66]. It consists of a stacked VRRAM, which includes a TiN/SiO_2 double layer with a TiN liner operating as BE surrounded by cylindrical-shaped HfO_2 switching layer and Ti-based TE, serially connected to a FET serving as selector and current limiter during set operation. This architecture allows to build a 1T-nR structure which, thanks to the multiple binary RRAM devices connected in parallel configuration, exhibits conductance change with gradual dynamics [57]. In particular, it exhibited a strong potential as electronic synapse in auditory pattern extraction applications enabling the implementation of a simplified stochastic STDP-based learning rule similar to that proposed in [54], which is shown in Fig. 17.16B, via intrinsic variability of set and reset processes in RRAM elements. Another hardware implementation of 3D hybrid CMOS/memristive synapse was proposed in [67]. It is a four-layer 3D VRRAM, consisting of a TiN/Ti layer as common TE, a HfO_x film as switching layer and 4 TiN layers as BEs, integrated with a p-channel FinFET operating as 3D selector (Fig. 17.16C). To implement synapses capable of stochastic learning, the intrinsic switching variability within

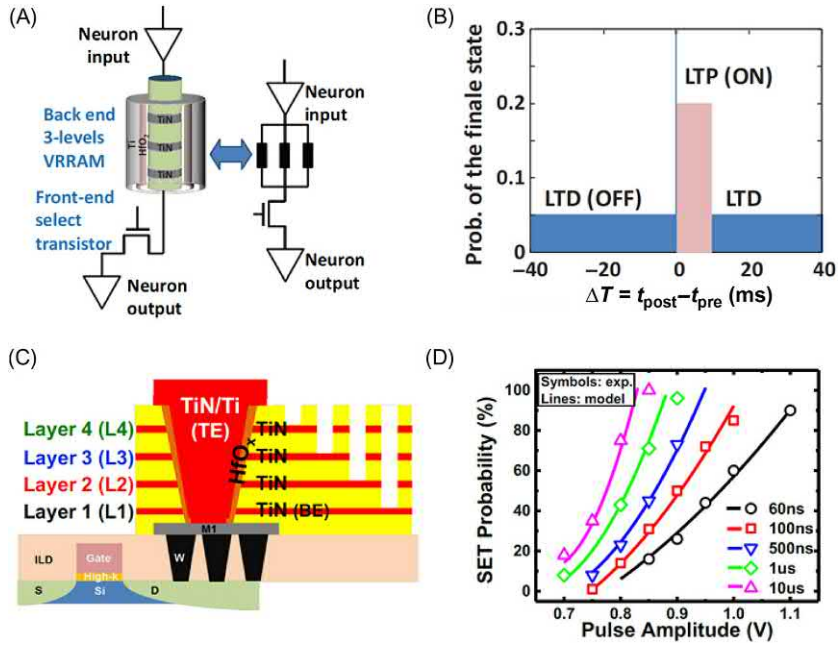


FIGURE 17.16 (A) Schematic representation of 3D 1T-nR synapse and (B) probabilistic STDP learning rule implemented at synaptic level. (C) Sketch of four-layered 3D TiN/Ti/HfO$_x$/TiN VRRAM synapse. (D) Experimental and calculated behavior of set probability as a function of amplitude of applied pulse for increasing pulse width evidencing that the longer the pulse, the lower the pulse amplitude needed to achieve the set transition with high probability. *Adapted with permission from (A and B) G. Piccolboni, G. Molas, J.M. Portal, R. Coquand, M. Bocquet, D. Garbin, et al., Investigation of the potentialities of Vertical Resistive RAM (VRRAM) for neuromorphic applications, IEEE Int. Electron. Devices Meet. (IEDM) (2015) 447–450. Copyright 2015 IEEE; (C and D) H. Li, K.-S. Li, C.-H. Lin, J.-L. Hsu, W.-C. Chiu, M.-C. Chen, et al., Four-layer 3D vertical RRAM integrated with FinFET as a versatile computing unit for brain-inspired cognitive information processing, IEEE Symposium VLSI Technol. (VLSI Technol.) (2016) 1–2. Copyright 2016 IEEE.*

RRAM was exploited. Specifically, the proper design of applied pulses in terms of duration and amplitude can be exploited to define the switching probability (Fig. 17.16D).

17.5 Synaptic transistors (3-terminal synapses)

The adoption of two-terminal memristive devices as electronic synapses offers the great advantage to realize very compact neuromorphic networks thanks to their very small size. However, although the introduction of a third terminal via 1T1R structure increases the synaptic footprint, it enables a higher conductance/weight tunability and the removal of sneak paths into the

arrays. For this reason, various three-terminal synaptic transistor concepts, such as solid electrolyte transistor also called electrochemical random access memory (ECRAM) [68–70], ionic floating-gate memory [71], ferroelectric field-effect transistor (FeFET) [72], spin–orbit torque magnetic random access memory (SOT-MRAM) [73], and memtransistor [74], have recently attracted strong interest.

Fig. 17.17A schematically shows a FeFET device with a gate stack including a Si:HfO_2 ferroelectric layer (FE) sandwiched between a TiN layer and a SiON layer [72] where the application of a series of positive/negative voltage pulses with increasing amplitude or width at gate terminal allows to gradually increase/decrease its channel conductivity via lowering/increase of its threshold voltage. As reported in [72], this three-terminal device was used in the circuit shown in Fig. 17.17B to implement an electronic synapse interconnecting PRE and POST neurons. According to this scheme, the PRE sends spikes at the FeFET gate and to the resistor connecting the gate and drain terminals, whereas the POST sends spikes at bulk terminal, which is

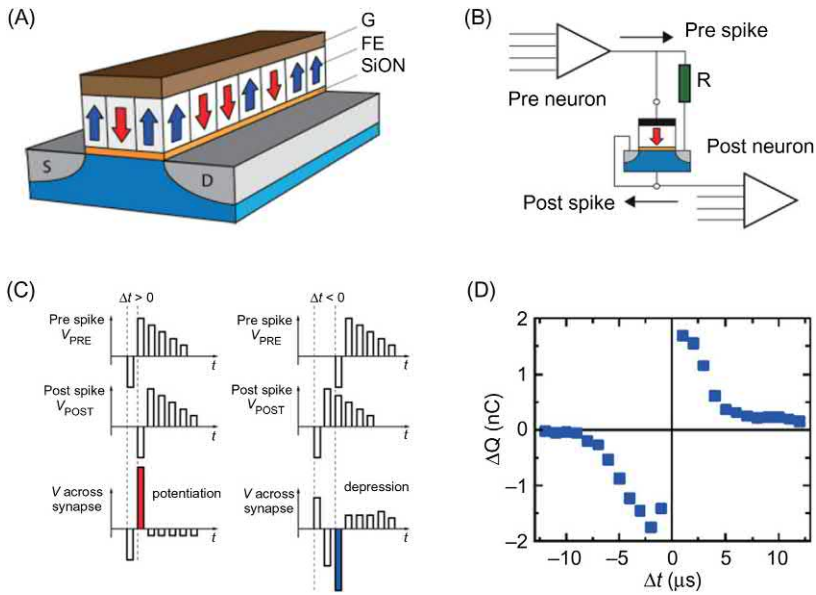


FIGURE 17.17 (A) Illustrative scheme of a FeFET device where a HfO_2 ferroelectric layer serves as dielectric layer into the gate stack. (B) Neuromorphic building block exhibiting the FeFET device as synaptic connection between the PRE neuron and POST neuron. (C) Programming scheme used to implement synaptic potentiation and depression into FeFET synapse. (D) Measured STDP characteristics achieved in a FeFET synapse by using a time delay–voltage amplitude conversion approach. *Adapted with permission from H. Mulaosmanovic, J. Ocker, S. Müller, M. Noack, J. Müller, P. Polakowski, et al., Novel ferroelectric FET based synapse for neuromorphic systems, IEEE Symposium VLSI Technol. (VLSI Technol.) (2017) T176–T177. Copyright 2017 IEEE.*

connected with the source. As external voltage spikes are emitted by PRE, signal transmission in the synaptic device is enabled by the flow of a drain current, which also induces a voltage across the resistor between gate and drain terminals and therefore an electric field able to activate the polarization switching process into the FE layer, hence a conductance change within the synaptic device. Thanks to the application of pre- and postsynaptic spikes shown in Fig. 17.17C, the overlap of spike voltage waveforms across FeFET device led it to undergo synaptic potentiation (PRE−POST sequence or $\Delta t > 0$) and depression (POST−PRE sequence or $\Delta t < 0$), thus allowing to demonstrate the ability of this synaptic transistor to achieve STDP functionality in experiments as shown in Fig. 17.17D.

17.6 Triplet-based synapses

Pair-based synaptic modulation has been a staple in the implementation of neuromorphic computing systems capable of learning, thanks to the algorithm's simplicity. However, beyond experiments where synaptic efficacy is measured after pairs of pre- and postsynaptic spikes as a function of their relative timing, the plasticity rule fails to replicate the results of more complicated experiments [26]. This is believed to result from an asymmetry in the impacts of the spike timings of the pre- and postsynaptic cells in favor of the postsynaptic one in biology. In order to break the symmetry of pair-based STDP in artificial systems, extensions of the pair-based algorithms have been proposed and are often termed triplet (or quadruplet) rules harking back to the experiments which motivated their development [26]. Typical pair-based STDP rules make use of one local variable each at the pre- and postsynapse which exponentially decay in time with the weight change being a function of the two states: 'o' represents the exponentially decaying postsynaptic variable, while 'r' denotes the presynaptic variable in Fig. 17.18. The values of these local variables can be thought of as being "stamped" in time giving the famous form of the synaptic weight change expression (Δw), as a function of the spike times:

$$\Delta w\left(t_{\text{pre}}, t_{\text{post}}\right) = \begin{cases} Ae^{\left(\frac{t_{\text{pre}} - t_{\text{post}}}{\tau}\right)}, t_{\text{post}} > t_{\text{pre}} \\ -Ae^{\left(\frac{t_{\text{post}} - t_{\text{pre}}}{\tau}\right)}, t_{\text{pre}} > t_{\text{post}} \end{cases} \tag{17.1}$$

where, A is the amplitude of the maximum synaptic efficacy change, t_{pre} is the timestamp of the last presynaptic spike, t_{post} is the timestamp of the last postsynaptic spike, τ is time constant of the decay from maximum synaptic change to zero change, and w is the synaptic efficacy. As an extension, triplet-based rules make use of an extra exponentially decaying variable per pre- and postsynapse and explicitly use their value in time to update the

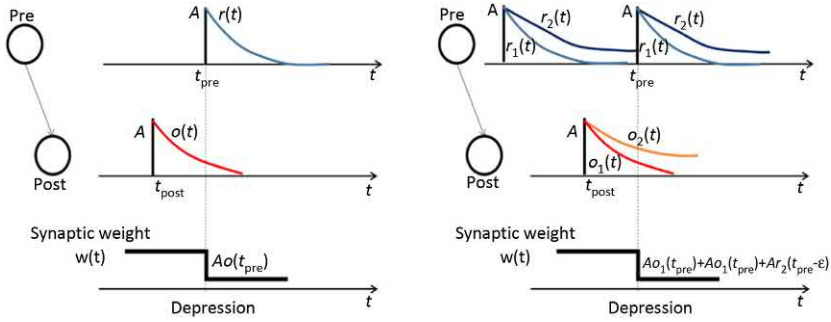


FIGURE 17.18 Schematic representation of the pair-based (left) and of the triplet-based STDP rules (right). Synaptic weight change (depression event) is shown. *Adapted with permission from J. Pfister, W. Gerstner. Triplets of spikes in a model of spike timing-dependent plasticity, J. Neurosci. 26 (38) (2006) 9673–9682. Copyright 2006 Society for Neuroscience.*

synaptic weight. These synaptic variables are stepped by a constant value when their respective neuron fires. This can be written as follows:

$$\frac{dx(t)}{dt} = \frac{-x(t)}{\tau}, \text{at spike arrival}\left(t = t_{\text{pre}} \vee t = t_{\text{post}}\right), x \rightarrow x + 1 \tag{17.2}$$

In the formulation of the triplet rule, each presynaptic spike t_{pre} induces an increase of two presynaptic variables, r_1 and r_2, and each postsynaptic spike t_{post} induces an increase of two postsynaptic variables, o_1 and o_2. All these variables, o_1, o_2, r_1, and r_2 follow Eq. (17.2) where the time constant for each variable is independent as in Fig. 17.18. Using these four synaptic time-dependent variables, Eq. (17.3) describes the triplet rule synaptic updates where the ratios between the time constants of $o_{2/1}$ and $r_{1/2}$ introduce the asymmetry in favor of the postsynapse.

$$\Delta\omega(t) = \begin{cases} -o_1(t)\left(A_2^- + A_3^- r_2(t-\varepsilon)\right), \text{if } t = t_{\text{pre}} \\ r_1(t)\left(A_2^+ + A_3^+ o_2(t-\varepsilon)\right), \text{if } t = t_{\text{post}} \end{cases} \tag{17.3}$$

where o_1 and o_2 are the postsynaptic variables that vary in time as described in Eq. (17.2), r_1 and r_2 are the presynaptic variables that vary in time as described in Eq. (17.2), A_2 is the maximum amplitude of change resulting from pairing of two spikes as in standard STDP, A_3 is the maximum amplitude of change resulting from pairing of three spikes extending the original STDP update to triplet STDP. Note that for the case of setting the constants A_3 to zero, Eq. (17.3) assumes an alternate form of Eq. (17.2) where the local variables are explicitly written instead of the spike time. It is therefore important to realize that triplet STDP is not a novel rule but a higher order extension of pair-based STDP—analogous to using a higher order function to better fit the data. This work has motivated the development of synapses capable of implementing triplet learning algorithms for

neuromorphic computing systems [75]. The work is based on the assumption that a resistive memory follows a behavioral model: the resistance of the device decreases exponentially if the applied voltage to the two terminals of the device (Δv) is higher than a given threshold (v_{th}), while it increases exponentially if the applied voltage is lower than $-v_{th}$, as described in Eq. (17.4).

$$f(\Delta v) = \begin{cases} I_0 \times \Delta v \left(e^{\frac{\Delta v - v_{th}}{v_0}} \right), |\Delta v| > |v_{th}| \\ 0, |\Delta v| < |v_{th}| \end{cases} \tag{17.4}$$

where $f(\Delta v)$ is a function which returns a change in the current passing through the resistive memory given an applied voltage Δv, I_0 and v_0 are two fitting parameters. Since the synaptic variables are exponential functions of time, a parallel exists with the exponential dependence on applied voltage of the resistance. It is then possible to use two resistive memories per triplet rule synapse whose superposition encodes the total synaptic weight (Fig. 17.19). One memory code for the base-pair change as in standard STDP and the other for the extra change that results from the triplet rule. It is possible to simplify the triplet algorithm by removing the higher order change during presynaptic events, at the expense of slightly less biological correspondence, as in the spike-time dependent form written in Eq. (17.5).

$$\Delta w(t_{pre}, t_{post}) = \begin{cases} -Ae^{\left(\frac{t_{post} - t_{pre}}{\tau_1}\right)}, if\ t = t_{pre} \\ Ae^{\left(\frac{t_{pre} - t_{post}}{\tau_2}\right)} + Ae^{\left(\frac{t_{pre} - t_{post}}{\tau_2}\right)} \times e^{\left(\frac{t_{post(n)} - t_{post(n-1)}}{\tau_3}\right)}, if\ t = t_{post} \end{cases} \tag{17.5}$$

where A is the amplitude of maximum synaptic efficacy change, t_{pre} is the timestamp of the last presynaptic spike, t_{post} is the timestamp of the last postsynaptic spike, τ is the time constant of the decay from maximum synaptic change to zero change, w is the synaptic efficacy, $t_{post(n)}$ is the most recent postsynaptic spike time, $t_{post(n-1)}$ is the second most recent postsynaptic

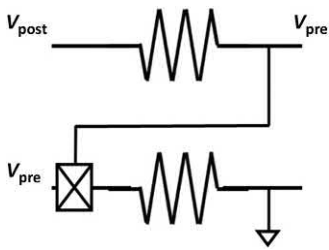


FIGURE 17.19 Two resistive memories synapse proposed in [75] to implement the triplet rule. The circuit comprises of two resistive memories and a multiplier/rectifier circuit shown as a crossed square. *Adapted with permission from M.R. Azghadi, B. Linares-Barranco, D. Abbott, P.H.W. Leong, A hybrid CMOS-memristor neuromorphic synapse, IEEE Trans. Biomed. Circuits Syst. 11(2) (2017) 434–445. Copyright 2017 IEEE.*

spike time. With suitably generated voltages, which are a function of spike events, their combination over the terminals of the simple circuit of Fig. 17.19 can result in changes to the two devices such that their superimposed weight changes in the manner of a triplet rule.

17.7 Spike-rate-dependent plasticity synapses

17.7.1 One-resistor synapses

In the human brain, crucial cognitive functionalities such as memory and learning are governed by complex synaptic mechanisms that are not yet fully understood. Some experimental studies such as the ones reported in [28,29] have revealed that, in addition to the timing of spikes underlying the well-known STDP learning rule, the repetition rate of spikes also plays a key role in such processes. For this reason, the biorealistic SRDP phenomenon taking into account the effect of spike rate on synaptic plasticity has attracted much attention to achieve a more faithful reproduction of synaptic behavior in hardware. Because of the limitations due to the abrupt nature of resistive switching process in RRAM materials highlighted in [76], the implementation of SRDP rule at device level has required the exploration of alternative devices/structures such as single-element Ag_2S inorganic synapses [77], one-selector/one-resistor (1S1R) structures equipped with SiO_xN_y:Ag diffusive memristors [43] and second-order memristors [35].

In [35], rate-based potentiation process was experimentally studied by applying a sequence of PRE spikes to the TE of Ta_2O_{5-x} RRAM device with grounded BE, which consist of a negative 20-ns-long set pulse of amplitude -1.1 V followed by a 1-μs-long pulse of amplitude -0.7 V for heat generation, separated by a time interval Δt as shown in Fig. 17.20A. In this manner, the shorter/longer Δt, the stronger/weaker the temporal heat accumulation effect on memristor conductance change already discussed in Section 17.3.1, which results in an increasing/decreasing synaptic potentiation. This is supported by SRDP characteristics for synaptic potentiation shown in Fig. 17.20B which show both an increase in conductance change for increasing number of applied spikes and a higher final weight for increasing stimulation frequency. Similar results were also obtained in rate-based synaptic depression experiments evidencing a stronger/weaker conductance decrease for high/low frequency stimulation of second-order memristor by programming pulses within PRE spikes with positive voltage polarity to reach reset transition (positive or negative polarity of heating pulses is unimportant). Therefore, these experimental results corroborate the ability of second-order memristors to implement another biorealistic long-term plasticity rule.

Although the key role played by long-term plasticity in fundamental brain functionalities such as memory and learning has been supported by several

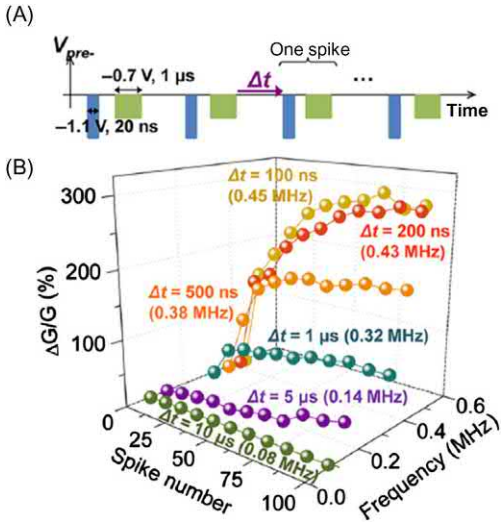


FIGURE 17.20 (A) SRDP implementation in a second-order Ta_2O_{5-x}-based memristor by application of a series of set/heating pulses for variable time interval Δt. (B) Measured SRDP characteristics as a function of number of the applied spikes with decreasing Δt from 10 μs to 100 ns. *Adapted with permission from S. Kim, C. Du, P. Sheridan, W. Ma, S.H. Choi, W.D. Lu, Experimental demonstration of a second-order memristor and its ability to biorealistically implement synaptic plasticity, Nano Lett. 15 (3) (2015) 2203−2211. Copyright (2015) American Chemical Society.*

biological experiments, the number of processes controlling real synaptic behavior is much wider and not yet totally understood. Among these additional effects, one of the most important ones is Short Term Plasticity [32]. Motivated by experimental observations [30−32], solutions aiming at capturing STP by various memristive devices have been proposed in recent years [43,77−79]. An interesting approach is the one presented by Werner et al. in [78] where STP was implemented using nonvolatile RRAM devices. To achieve STP, 10 Ti/HfO_2 RRAM cells were arranged in parallel configuration to realize a single synapse and the programming scheme described in Fig. 17.21A was implemented. According to this scheme, every PRE spike applied to all RRAM TEs causes abrupt reset transitions within resistive synapse (weight decrease) which are followed by weak set transitions at each period ΔT with no input, thus gradually restoring the initial synaptic state.

Based on this strategy, Fig. 17.21B shows the experimental and calculated evolution of synaptic weight $y(t)$ as a function of time evidencing short-term changes that can be tuned to control set/reset probabilities ($p_{set} = 0.05$ and $p_{reset} = 0.5$ in this case). The advantage of this strategy is that it does not require a new technology to implement STP in that the same nonvolatile RRAM technology is adopted for both STP and long-term plasticity, thus simplifying the fabrication process. Thanks to long-term plasticity, the system is capable of learning using an unsupervised paradigm. STP allows improving accuracy in the presence of significant background noise in the input data. Volatile resistive switching memories have also been proposed to implement short-term synapses. In this context, particular focus should be attributed to Ag_2S-based inorganic synapses presented in [77] where STP is captured by spontaneous rupture of the metallic filament

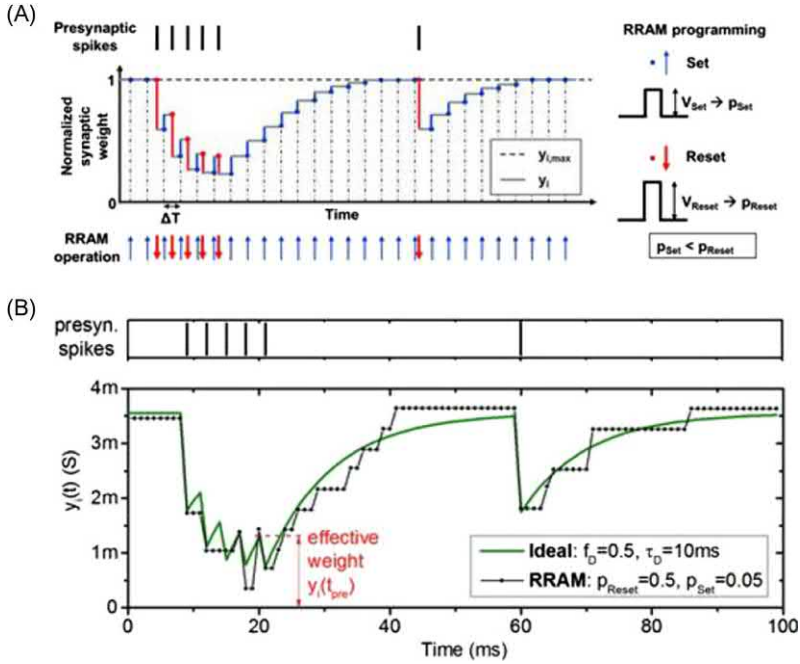


FIGURE 17.21 (A) Programming strategy used to achieve short-term plasticity (STP) in stochastic synapse based on 10 RRAM devices in parallel according to which each incoming PRE spike leads to abrupt depression. Probabilistic set events can occur at each time slot ΔT with no external input, thus enabling to recover initial high conductance state. (B) STP implementation at experimental and simulation level based on the pulse scheme shown in (A). *Adapted with permission from T. Werner, E. Vianello, O. Bichler, A. Grossi, E. Nowak, J.-F. Nodin, et al., Experimental demonstration of short and long-term synaptic plasticity using OxRAM multi k-bit arrays for reliable detection in highly noisy input data, IEEE Int. Electron. Devices Meet. (IEDM) (2016) 432–435. Copyright 2016 IEEE.*

induced by low-frequency spiking stimulation, and to diffusive SiO_xN_y:Ag memristor [43] which is capable of implementing short-term depression and facilitation, similar to [79], thanks to the diffusive dynamics of Ag ions in response to low-frequency spike trains.

17.7.2 Four-transistors/one-resistor synapses

Because of the abrupt nature of resistive switching mechanism in many RRAM materials, most RRAM devices do not reproduce SRDP protocol without using complex synaptic structures and programming schemes. In this regard, a synapse circuit based on a hybrid CMOS/RRAM structure capable of SRDP functionality was presented in [60,80].

As shown in Fig. 17.22A, PRE and POST blocks are connected by a hybrid synaptic structure called four-transistors/one-resistor (4T1R) synapse

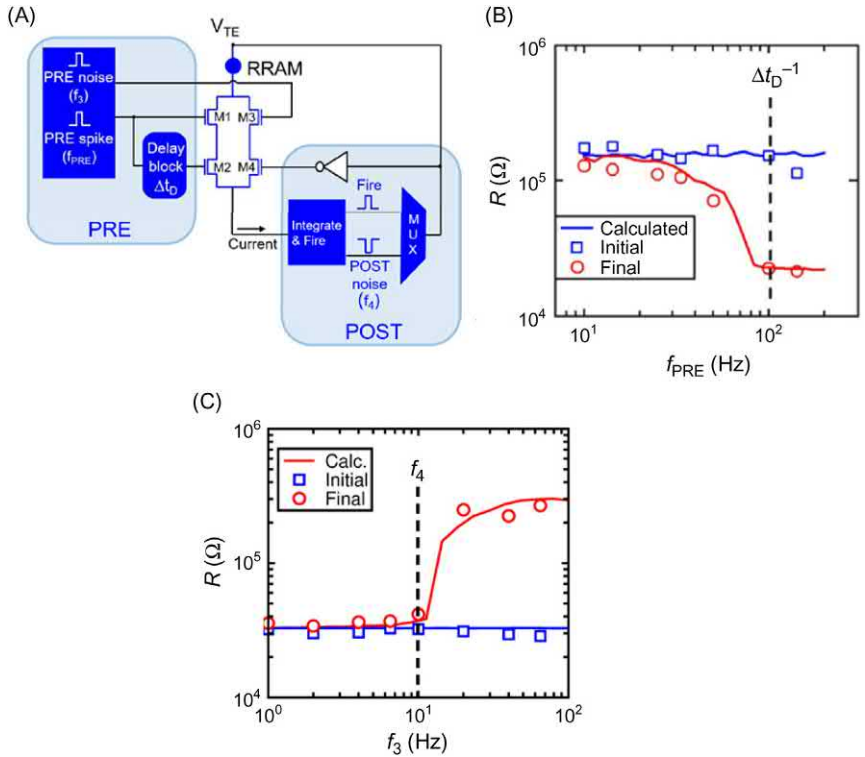


FIGURE 17.22 (A) Schematic representation of hybrid 4T1R RRAM synapse capable of replicating SRDP biorealistic rule. Experimental demonstration of (B) synaptic potentiation for $f_{PRE} > \Delta t^{-1}$ and (C) synaptic depression for $f_3 > f_4$ when $f_{PRE} \ll \Delta t^{-1}$. *Adapted with permission from V. Milo, G. Pedretti, R. Carboni, A. Calderoni, N. Ramaswamy, S. Ambrogio, et al., Demonstration of hybrid CMOS/RRAM neural networks with spike time/rate-dependent plasticity, IEEE Int. Electron. Devices Meet. (IEDM) (2016) 440–443. Copyright 2016 IEEE.*

which comprises of a HfO_2 RRAM device and two parallel branches each of which includes a pair of FETs, M_1/M_2 for left branch and M_3/M_4 for right branch, in serial configuration. In this circuit scheme, PRE block includes a spike generator emitting Poisson distributed asynchronous PRE spikes, which are applied to the gate of M_1 and, after being shifted by a delay Δt_D, to the gate of M_2, and a generator of PRE noise spikes driving the gate of M_3, whereas the POST block consists of an integrate-and-fire circuit followed by a multiplexer (MUX) and an inverter. As the external stimulation rate (f_{PRE}) is higher than ${\Delta t_D}^{-1}$, the probability that M_1 and M_2 are simultaneously enabled by PRE spikes and their delayed copies, respectively, is high, thus enabling a current to flow across the M_1/M_2 branch. This current is integrated by POST causing an increase in the POST internal potential approaching it to a threshold. As the threshold is hit, POST internal potential is reset, and a fire pulse is emitted by the POST fire channel and then delivered backward to the TE leading to a set transition of RRAM device, hence the synapse potentiation, thanks to overlapping voltages driving M_1, M_2, and TE, which implement a PRE-PRE-POST modified triplet-based weight scheme similar to ones described in [26,81]. Note that the fire pulse, after being inverted by the inverter gate within the POST circuit block, is applied to the gate of M_4, thus disabling the M_3/M_4 branch during the potentiation mode. Hence, M_1/M_2 is the only branch designed to achieve synaptic potentiation in the 4T1R RRAM synapse. On the other hand, as f_{PRE} is much lower than ${\Delta t_D}^{-1}$, spike coincidences at the inputs of potentiation branch occur. Therefore, a second branch based on M_3/M_4 pair was added in parallel to capture weight decrease at low f_{PRE}. Its operation relies on the stochastic activation of M_3 via noise spikes emitted by PRE at frequency f_3, and of M_4 and TE by POST noise spikes, which are randomly emitted by POST at frequency f_4 and transmitted via the MUX as the POST fire channel is inactive. Note that both f_3 and f_4 are set lower than f_{PRE} to prevent any interference between potentiation branch and depression branch during synaptic operation. As these three random noise voltage pulses overlap, the M_3/M_4 branch is enabled and a stochastic reset transition is triggered in the RRAM device leading to a weight decrease, given the negative polarity of voltage pulse at TE. As a result, 4T1R RRAM synapse operation allows for SRDP implementation by selective synaptic potentiation for high-frequency spiking stimulation and a stochastic synaptic depression for low-frequency spiking stimulation by using biologically inspired stochastic noise spikes emitted by PRE and POST [82].

The ability of the 4T1R RRAM synapse circuit to implement high-frequency potentiation and low-frequency depression was also validated in experiments studying potentiation and depression operation modes separately in 2T1R integrated structures. As shown in Fig. 17.22B, given a delay $\Delta t_D = 10$ ms, a resistance change from HRS to LRS in RRAM device, hence synaptic potentiation, can be achieved only for $f_{PRE} >= 100$ Hz, that is

$\Delta t_D{}^{-1}$, thus supporting high-frequency potentiation. On the other hand, a resistance transition from LRS to HRS in the RRAM device can be triggered by PRE and POST noise spikes provided $f_3 > f_4$, as shown in Fig. 17.22C where f_4 was set to 10 Hz. This result also confirms the feasibility of stochastic synaptic depression and, consequently, of SRDP operation in the 4T1R RRAM synapse.

17.7.3 One-selector/one-resistor synapses

In parallel to hybrid CMOS/RRAM structures capable of mimicking synaptic behavior using nonvolatile resistive switching phenomenon in various RRAM devices such as 1T1R synapse (Section 17.4.1) and 2T1R synapse (Section 17.4.2), other attractive hybrid structures based on memristor devices are being intensively explored to achieve a more detailed replication of biological dynamics. Among them, strong interest was gained by 1S1R structure with RRAM devices based on material stacks such as SiO_xN_y:Ag [43], Cu/SiO_x [83], and Ag/SiO_x [83] exhibiting volatile resistive switching as a result of spontaneous retraction of metallic filaments within a short retention time in the range from few microseconds to few milliseconds.

Fig. 17.23 shows (A) the stack of volatile switching Ag/SiO_x/C RRAM device and (B) its measured $I-V$ characteristics collected after three cycles , which display a bidirectional switching behavior where the application of both a positive threshold voltage V_{T+} and a negative threshold voltage V_{T-} leads to the on-state via an abrupt increase in current limited to $I_C = 35\ \mu A$ via introduction of a select transistor in series. Also, one can note that as the positive/negative voltage applied across RRAM becomes lower/higher than characteristic holding voltage V_{H+}/V_{H-}, an abrupt current drop due to a spontaneous filament disconnection takes place leading the RRAM device from the on-state to the off-state [79,83]. In addition to this, Fig. 17.23B also displays a very large resistance window between on-state and off-state ($> 10^7$) and very steep resistance transitions, which are fundamental device features that are captured with nice agreement by a compact model described in [79]. Thanks to its ability to predict the Ag/SiO_x RRAM experimental characteristics with good detail, this physics-based analytical model was used to investigate SRDP in a 1S1R structure where the volatile switching Ag/SiO_x RRAM cell serves as select device. Fig. 17.23C and D show the scheme of a 1S1R cell based on a nonvolatile RRAM device serially connected with a volatile RRAM selector and its corresponding $I-V$ characteristics calculated by combined use of the physics-based analytical model for nonvolatile RRAM presented in [84] and the physics-based model for the volatile RRAM presented in [79], respectively. Fig. 17.23E also shows the calculated current response of the 1S1R device for variable spiking stimulation regimes. In particular, it should be noted that the application of a spike train at high frequency ($f_{spike} = 2$ kHz) leads to a gradual current

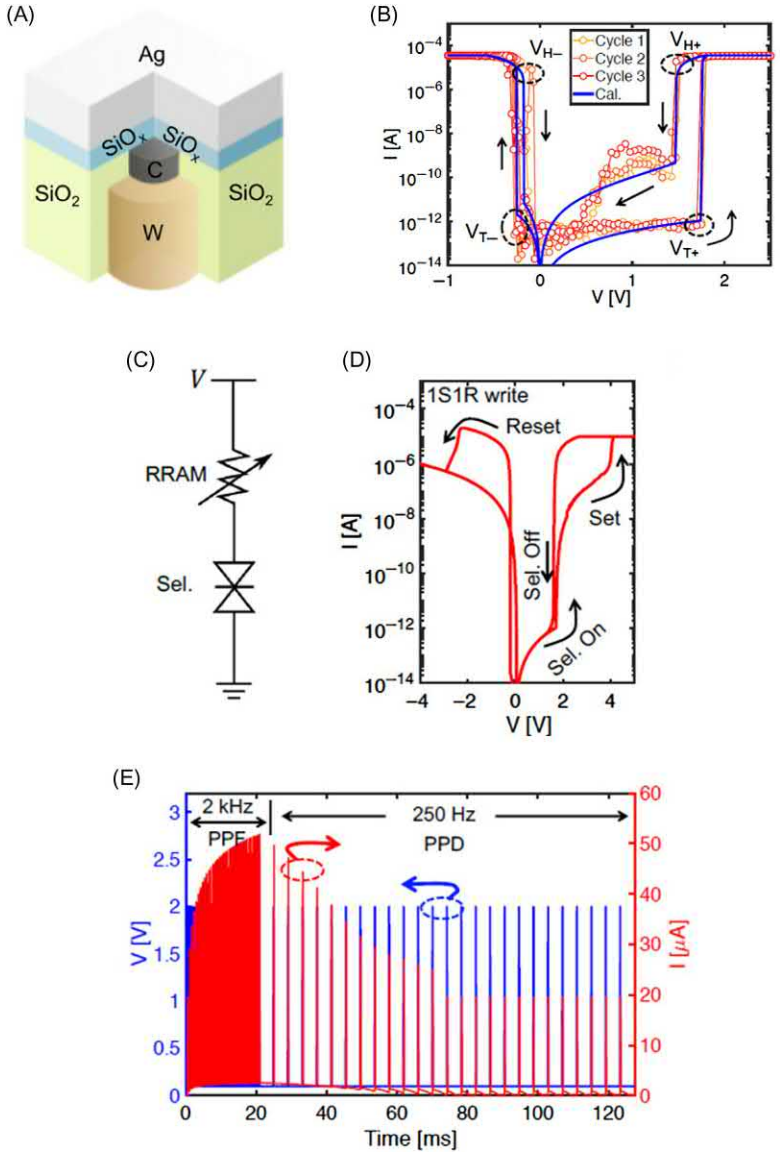


FIGURE 17.23 (A) Sketch of an Ag/SiO_x RRAM device based on a stack with an Ag top electrode, a SiO_x switching layer and a graphitic carbon bottom electrode. (B) Measured $I-V$ characteristics of an Ag/SiO_x RRAM device exhibiting volatile resistive switching against a calculated curve obtained by the compact model presented in [79]. (C) Scheme a 1S1R structure obtained combining a nonvolatile RRAM device with a volatile RRAM select device and (D) its $I-V$ characteristics. (E) Calculated current response of 1S1R device evidencing SRDP capability via paired-pulse facilitation (PPF) for high-frequency spiking stimulation and paired-pulse

(*Continued*)

(conductance) increase thanks to the gradual growth of filament induced by spikes, which results in the so-called paired-pulse facilitation (PPF). On the contrary, under a low-frequency spiking stimulation ($f_{spike} = 250$ Hz), conductance gradually decreases because the filament dissolution dominates over its growth, leading to another regime known as paired-pulse depression (PPD). The implementation of these two processes thus suggests the ability of volatile RRAM devices in 1S1R cell to capture biologically inspired SRDP algorithm with the added value, compared to the 4T1R RRAM synapse described in Section 17.7.2, to achieve a significant area saving, which makes it very promising for building of dense crosspoint synaptic networks capable of brain-inspired cognitive functionalities.

17.8 Self-learning networks with memristive synapses

In recent years, we have seen a boost in the performance and applications of machine learning (ML), driven by several factors: (1) the availability of large datasets for training and models and (2) the increased computational power of modern computers (GPUs are an excellent match for ML thanks to the high degree of parallelization). Among many fields of ML, deep learning (DL) is the most popular. DNNs fall into three classes of architectures: fully connected neural network (FCNN), convolutional neural network (CNN), and recurrent neural network (RNN).

As shown in Fig. 17.24, an FCNN is composed of fully-connected layers, each of which contains a collection of processing units (neurons) and weights (synapses). The neurons of a given layer are connected to every neuron of the previous layer by synapses. Raw data (e.g., video, audio, biological data, etc.) is fed as the values of the first layer (the input layer). The output layer corresponds to the inference classes (each output neuron is associated to a class of objects, e.g., dog, cat, car, etc.). The number of weights and operations is directly proportional to the dimensions of the layers. On the other hand, CNN is composed of one or more convolutional layers, pooling or subsampling layers, and fully connected output layers (Fig. 17.25). In a convolutional layer, a small set of synapses constituting a kernel allows subsequent network layers to extract spatially localized features before the information is subsampled and pooled and often used to drive further convolutional

◀ depression (PPD) for low-frequency spiking stimulation. *(A) Adapted from A. Bricalli, E. Ambrosi, M. Laudato, M. Maestro, R. Rodriguez, D. Ielmini, Resistive switching device technology based on silicon oxide for improved on-off ratio—Part II: Select devices, IEEE Trans. Electron. Devices 65 (1) (2018) 122–128; (B–E) Adapted with permission from W. Wang, A. Bricalli, M. Laudato, E. Ambrosi, E. Covi, D. Ielmini, Physics-based modeling of volatile resistive switching memory (RRAM) for crosspoint selector and neuromorphic computing, IEEE Int. Electron. Devices Meet. (IEDM) (2018) 932–935. Copyright IEEE 2018.*

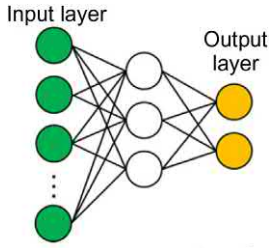


FIGURE 17.24 Example of two-layer Fully Connected Neural Network (FCNN).

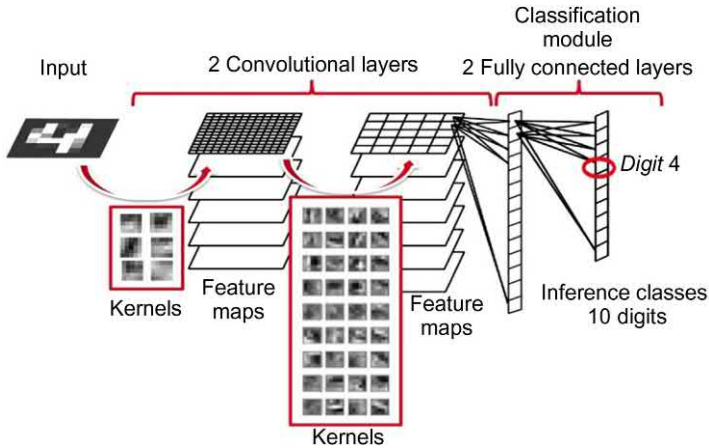


FIGURE 17.25 Schematic of Convolutional Neural Network (CNN) used for handwritten digits recognition (MNIST database). *Adapted with permission from D. Garbin, O. Bichler, E. Vianello, Q. Rafhay, C. Gamrat, L. Perniola, et al., Variability-tolerant Convolutional Neural Network for pattern recognition applications based on OxRAM synapses, IEEE Int. Electron. Devices Meet. (IEDM) (2014) 661−664. Copyright IEEE 2014.*

layers. The outputs of the convolutional layers, also called feature maps, contain information about the locations where the features extracted by learned kernels are present in the input. The fully connected layer (classification module) is applied to complete the classification. Inference in CNN is identical to that of FCNN. The input data initialize the processing units of the first layer and the algorithm moves forward layer by layer. The activity of the processing units in the output layer corresponds to the inferred classes as for the FCNN. CNN can achieve superb classification accuracy for image processing at much lower weight count than FCNN. Unlike FCNNs and CNNs, RNNs have loops enabling information to persist since the input at each step is composed of the data at that step in conjunction with the network output obtained at the previous step (Fig. 17.26). They are the natural architecture to use for sequential or temporal data. In the last few years, there has been incredible success applying RNN to a variety of problems such as speech

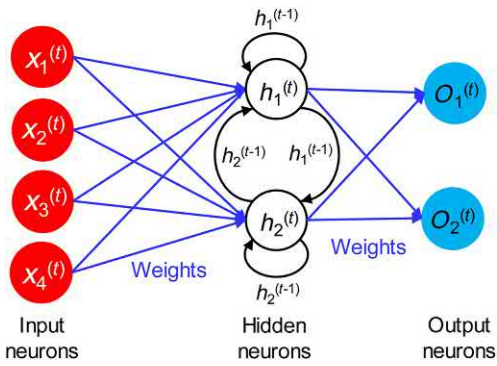


FIGURE 17.26 Sketch of a multilayer Recurrent Neural Network (RNN).

recognition, language modeling, and language translation. In particular, the well-known long-short-term memory (LSTM) RNN has recently found extensive application in text and speech recognition tasks.

There are two modes of operation for neural network models. Before a network can be applied to an application, it must be trained whereby the parameters of the network model are determined, normally, through a mathematical optimization that minimizes some metric of error given input training data. Through repeated presentations of the training data, and application of a learning algorithm or rule, networks update their parameters such that their responses become closer to the desired output. Three broad classes of learning approaches exist—supervised, unsupervised, and reinforcement. For supervised learning, the training data is labeled (each example is a pair consisting of input data and a desired output). In contrast, the training phase in unsupervised learning receives only the input data without a desired output. After a network has been trained, inference can take place. Inference takes new un-seen data and uses the trained model to make output predictions (recognize images, spoken words, a blood disease, etc.). During both training and inference steps, the neural network performs very large multiply accumulate (MAC) operations between the weights of the model and the new input data. It therefore often requires hardware optimized for matrix multiplications such as GPUs. RRAM arrays are promising candidates to implement the MAC operation, where the multiply operation is performed at every crosspoint by Ohm's law and the resulting currents are summed along rows or columns. Moreover, since they are fabricated in the BEOL, they are increasingly attractive for their high density. In addition, there is interest to use RRAM in more biologically inspired architectures and learning rules as presented in the previous sections. Bioinspired algorithms such as STDP are also thought to enable learning.

Hardware implementations of the inference operation in neuromorphic hardware have been presented in the literature [86–93]. An RRAM

perceptron classifier implemented entirely in integrated hardware is presented in [90]. Multivalued resistance levels are stored in the RRAM cells. The test chip, with 2 M synapses integrated into 130 nm CMOS, results in 90.8% MNIST recognition rate (ex situ training). A small-scale perceptron classifier based on RRAM crossbar array board integrated with discrete CMOS components is presented in [91]. The network was trained both in situ and ex situ to perform classification of 4×4 pixel images.

Brain-inspired learning in SNNs with RRAM synapses has been widely explored in recent years [85,94−104]. A perceptron-like neuromorphic hardware capable of STDP-based unsupervised learning was presented in [98]. This hardware network consists of a fully connected perceptron neural network (16 PREs and 2 POSTs) where all the PREs were connected with each POST by individual 1T1R RRAM synapses identical to the ones described in Section 17.4.1. Inhibitory synapses between the two POST neurons enable the implementation of the well-known winner-take-all scheme [105] according to which the POSTs are not allowed to fire together in order to maximize the storage capability of multiple visual patterns. The system was implemented on a PCB connecting an Arduino Due microcontroller and synaptic elements with 1T1R integrated structure. The learning of two patterns has been experimentally demonstrated: two patterns and random noise were stochastically submitted to the 1st layer of the network. Noise submission induces depression within background synapses, thus allowing to "forget" the previously learnt pattern when a new one is submitted. Noise is shown to decrease learning time and reduce the probability of "false firing". However excessive noise results in unstable learning by increasing the probability of false firing [99,100].

In addition to hardware demonstration of ability to learn static visual patterns via STDP, the 1T1R RRAM synapses adopted in [98,99] were also used to connect 16 PREs with a single POST in a perceptron network in order to implement learning of spatiotemporal sequences [102]. To this end, PREs were subjected to the presentation of spatiotemporal patterns consisting of sequences of four spikes that were labeled as true/false patterns according to a teacher signal. Fig. 17.27A shows the experimental demonstration of learning of spatiotemporal patterns in the same perceptron network with 1T1R RRAM synapses showing (top) the supervision signal and V_{int} measured in response to the sequence submission during training, (center) true fire, false fire and false silence spikes generated during the experiment, and (bottom) the color plot of potentiation/depression behavior of all the synaptic weights at increasing training cycle which suggests that 1-4-9-16 sequence was chosen as the true spatiotemporal pattern. Fig. 17.27B shows some experimental results for recognition phase following training phase. Fig. 17.27B (top) shows that the submission of true pattern allows V_{int} to cross voltage threshold, thus supporting the network ability to capture the true sequence learnt during training. In addition to this, as shown in

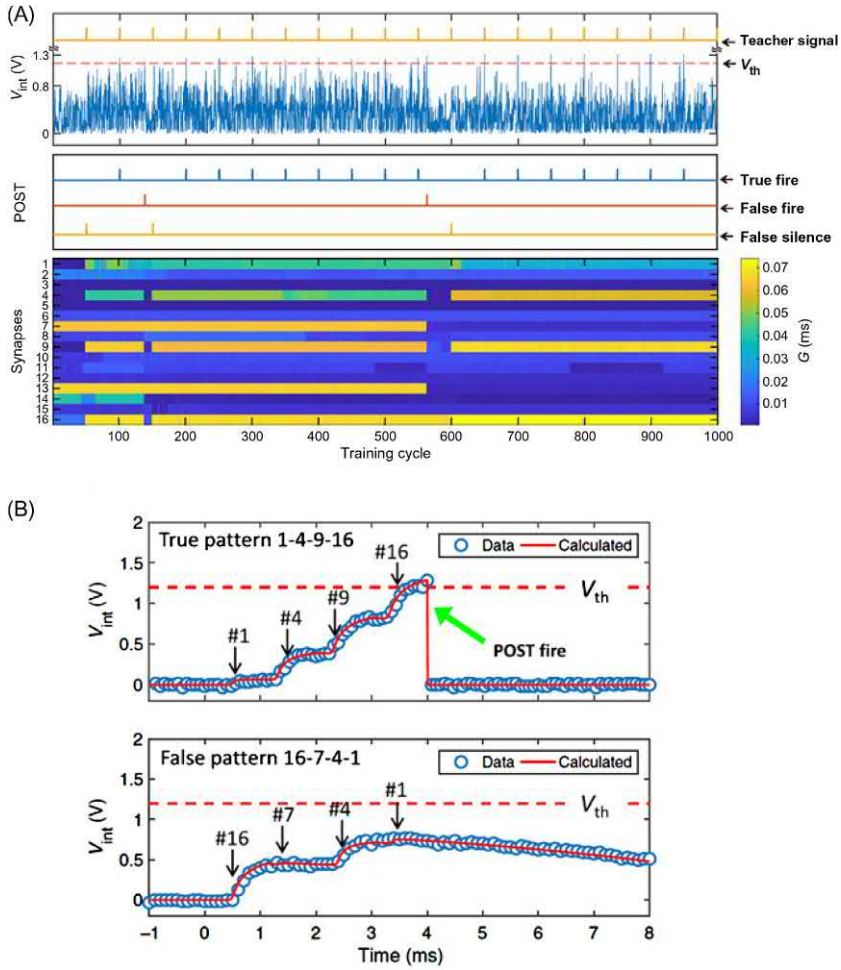


FIGURE 17.27 (A) Experimental training of spatiotemporal patterns captured by a perceptron network with 16 RRAM synapses. (Top) Sequence of teaching spikes used as labels for true pattern and measured evolution of V_{int} during training phase. (Center) True fire, false fire, and false silence spikes occurring during training and (Bottom) evolution of measured conductance for each synapse during training which evidences potentiation of weights associated to the true pattern 1-4-9-16. (B) Experimental results for recognition phase evidencing that crossing of voltage threshold by V_{int} marks if (Top) the true sequence is effectively recognized or (Bottom) if submitted sequence is a false pattern. *Adapted from W. Wang, G. Pedretti, V. Milo, R. Carboni, A. Calderoni, N. Ramaswamy, et al., Learning of spatiotemporal patterns in a spiking neural network with resistive switching synapses, Sci. Adv. 4 (9) (2018) eaat4752.*

Fig. 17.27B (bottom), the network is able to recognize false patterns submitted at input layer, as for instance 16-7-4-1 sequence, since V_{int} cannot hit the voltage threshold in these cases.

The role of synaptic variability (due to the intrinsic cell-to-cell and cycle-to-cycle variability) during unsupervised learning by STDP is investigated in [104] by means of system-level simulations calibrated on the characterization of a 4-kbit RRAM array. A fully connected feed-forward neural network topology with leaky integrate-and-fire neurons and RRAM-based synapses is adopted. A detection task in dynamic input data is investigated. The network is composed of one-layer fully connected network topology. The input layer is an image sensor composed of 128×128 spiking pixels, fully connected to an input layer of 60 neurons. The results are based on system-level simulations, calibrated on the experimental data (measurements have been performed on a 4-kbit 1T1R array). The results demonstrate that, similarly to biology, SNNs are not only robust to variability but a certain amount of it can improve the network performance. More precisely, the performance of the proposed application for measured RRAM conductance distributions and an artificial device with zero variability are studied. The RRAM Memory Window is defined as the ratio between the high conductance value at minus three standard deviation (-3σ) and the low conductance value at 3σ of the cumulative conductance distribution. For a memory window of 2.25 the detection score is 0.63 for the artificial synapse with no variability and 0.952 for the real RRAM. The increase of both conductance variability and memory window allows for an increase of the ratio between the conductance values of potentiated and depressed synapses, thus improving the learning accuracy.

17.9 Conclusion

This chapter reviews the realization of synaptic elements within neuromorphic hardware by using memory and memristive devices. RRAM and PCM synapses show analog switching, scalable size, low voltage/power, thus offering a promising technology for achieving brain-inspired cognitive computing in both SNN and DNN architectures. To emulate the learning processes of the human brain, bioinspired STDP and SRDP rules can be realized by using either overlap or nonoverlap algorithms. The physics of RRAM devices can be used to naturally implement STDP and SRDP, for example, by thermal effects or ionic diffusion at the nanoscale. By combining neuron and synapse elements within a neuromorphic circuit, learning and recognition functions can be achieved, thus allowing to benchmark CMOS and memristive technologies for cognitive computing.

Acknowledgments

This work has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement no. 648635).

References

[1] F. Rosenblatt, The Perceptron: A Perceiving and Recognizing Automaton, Report 85-460-1, Cornell Aeronautical Laboratory, Buffalo, New York, 1957.

[2] M.L. Minsky, S.A. Papert, Perceptrons: An Introduction to Computational Geometry, The MIT Press, Cambridge MA, 1972.

[3] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (2015) 436–444. Available from: https://doi.org/10.1038/nature14539.

[4] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324. Available from: https://doi.org/10.1109/5.726791.

[5] W. Maass, Networks of spiking neurons: the third generation of neural network models, Neural Netw. 10 (9) (1997) 1659–1671. Available from: https://doi.org/10.1016/S0893-6080(97)00011-7.

[6] P.A. Merolla, J.V. Arthur, R. Alvarez-Icaza, A.S. Cassidy, J. Sawada, F. Akopyan, et al., A million spiking-neuron integrated circuit with a scalable communication network and interface, Science 345 (6197) (2014) 668–673. Available from: https://doi.org/10.1126/science.1254642.

[7] E. Chicca, F. Stefanini, C. Bartolozzi, G. Indiveri, Neuromorphic electronic circuits for building autonomous cognitive systems, Proc. IEEE 102 (9) (2014) 1367–1388. Available from: https://doi.org/10.1109/JPROC.2014.2313954.

[8] D. Ielmini, Resistive switching memories based on metal oxides: mechanisms, reliability and scaling, Semicond. Sci. Technol. 31 (6) (2016) 063002. Available from: https://doi.org/10.1088/0268-1242/31/6/063002.

[9] S. Raoux, W. Welnic, D. Ielmini, Phase change materials and their application to non-volatile memories, Chem. Rev. 110 (1) (2010) 240–267. Available from: https://doi.org/10.1021/cr900040x.

[10] C. Chappert, A. Fert, F.N. Van Dau, The emergence of spin electronics in data storage, Nat. Mater. 6 (2007) 813–823. Available from: https://doi.org/10.1038/nmat2024.

[11] B. Govoreanu, G.S. Kar, Y.-Y. Chen, V. Paraschiv, S. Kubicek, A. Fantini, et al., 10×10 nm^2 Hf/HfO_x crossbar resistive RAM with excellent performance, reliability and low-energy operation, IEEE Int. Electron. Devices Meet. (IEDM) (2011) 729–732. Available from: https://doi.org/10.1109/IEDM.2011.6131652.

[12] F. Nardi, S. Larentis, S. Balatti, D.C. Gilmer, D. Ielmini, Resistive switching by voltage-driven ion migration in bipolar RRAM—Part I: Experimental study, IEEE Trans. Electron. Devices 59 (9) (2012) 2461–2467. Available from: https://doi.org/10.1109/TED.2012.2202319.

[13] F. Arnaud, et al., Truly innovative 28 nm FDSOI technology for automotive micro-controller applications embedding 16MB phase change memory, IEEE Int. Electron. Devices Meet. (IEDM) (2018) 424–427. Available from: https://doi.org/10.1109/IEDM.2018.8614595.

[14] D. Ielmini, H.-S.P. Wong, In-memory computing with resistive switching devices, Nat. Electron. 1 (2018) 333–343. Available from: https://doi.org/10.1038/s41928-018-0092-2.

[15] C. Li, M. Hu, Y. Li, H. Jiang, N. Ge, E. Montgomery, et al., Analogue signal and image processing with large memristor crossbars, Nat. Electron. 1 (2018) 52–59. Available from: https://doi.org/10.1038/s41928-017-0002-z.

[16] P.M. Sheridan, F. Cai, C. Du, W. Ma, Z. Zhang, W.D. Lu, Sparse coding with memristor networks, Nat. Nanotechnol. 12 (2017) 784–789. Available from: https://doi.org/10.1038/nnano.2017.83.

[17] M. Le Gallo, A. Sebastian, R. Mathis, M. Manica, H. Giefers, T. Tuma, et al., Mixed-precision in-memory computing, Nat. Electron. 1 (2018) 246–253. Available from: https://doi.org/10.1038/s41928-018-0054-8.

[18] Z. Sun, G. Pedretti, E. Ambrosi, A. Bricalli, W. Wang, D. Ielmini, Solving matrix equations in one step with crosspoint resistive arrays, Proc. Natl. Acad. Sci. (PNAS) 116 (10) (2019) 4123–4128. Available from: https://doi.org/10.1073/pnas.1815682116.

[19] D. Ielmini, Brain-inspired computing with resistive switching memory (RRAM): devices, synapses and neural networks, Microelectron. Eng. 190 (2018) 44–53. Available from: https://doi.org/10.1016/j.mee.2018.01.009.

[20] G.-Q. Bi, M.-M. Poo, Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and post synaptic cell type, J. Neurosci. 18 (24) (1998) 10464–10472. Available from: https://doi.org/10.1523/JNEUROSCI.18-24-10464.1998.

[21] D.O. Hebb, The Organization of Behavior, JohnWiley & Sons, New York, 1949.

[22] M.A. Woodin, K. Ganguly, M.M. Poo, Coincident pre- and postsynaptic activity modifies GABAergic synapses by postsynaptic changes in Cl- transporter activity, Neuron 39 (5) (2003) 807–820. Available from: https://doi.org/10.1016/S0896-6273(03)00507-5.

[23] Y. Luz, M. Shamir, Balancing feed-forward excitation and inhibition via hebbian inhibitory synaptic plasticity, PLoS Computational Biol. 8 (1) (2012) 1–12. Available from: https://doi.org/10.1371/journal.pcbi.1002334.

[24] L.F. Abbott, S.B. Nelson, Synaptic plasticity: taming the beast, Nat. Neurosci. 3 (2000) 1178–1183. Available from: https://doi.org/10.1038/81453.

[25] T.P. Vogels, R.C. Froemke, N. Doyon, M. Gilson, J.S. Haas, R. Liu, et al., Inhibitory synaptic plasticity: spike timing-dependence and putative network function, Front. Neural Circuits 7 (2013) 119. Available from: https://doi.org/10.3389/fncir.2013.00119.

[26] J. Pfister, W. Gerstner, Triplets of spikes in a model of spike timing-dependent plasticity, J. Neurosci. 26 (38) (2006) 9673–9682. Available from: https://doi.org/10.1523/JNEUROSCI.1425-06.2006.

[27] J.M. Brader, W. Senn, S. Fusi, Learning real-world stimuli in a neural network with spike-driven synaptic dynamics, Neural Computation 19 (11) (2007) 2881–2912. Available from: https://doi.org/10.1162/neco.2007.19.11.2881.

[28] M.F. Bear, A synaptic basis for memory storage in the cerebral cortex, Proc. Natl. Acad. Sci. USA 93 (1996) 13453–13459. Available from: https://doi.org/10.1073/pnas.93.24.13453.

[29] P.J. Sjöström, G.G. Turrigiano, S.B. Nelson, Rate, timing, and cooperativity jointly determine cortical synaptic plasticity, Neuron 32 (2001) 1149–1164. Available from: https://doi.org/10.1016/S0896-6273(01)00542-6.

[30] M.F. Bear, R.C. Malenka, Synaptic plasticity: LTP and LTD, Curr. Opin. Neurobiol. 4 (3) (1994) 389–399. Available from: https://doi.org/10.1016/0959-4388(94)90101-5.

[31] H. Markram, D. Pikus, A. Gupta, M. Tsodyks, Potential for multiple mechanisms, phenomena and algorithms for synaptic plasticity at single synapses, Neuropharmacology 37 (4-5) (1998) 489–500. Available from: https://doi.org/10.1016/S0028-3908(98)00049-5.

[32] R.S. Zucker, W.G. Regehr, Short-term synaptic plasticity, Annu. Rev. Physiol. 64 (1) (2002) 355–405. Available from: https://doi.org/10.1146/annurev.physiol.64.092501.114547.

[33] S.N. Jung, A. Borst, J. Haag, Flight activity alters velocity tuning of fly motion-sensitive neurons, J. Neurosci. 31 (25) (2011) 9231–9237. Available from: https://doi.org/10.1523/JNEUROSCI.1138-11.2011.

[34] Y.V. Pershin, M. di Ventra, Neuromorphic, digital, and quantum computation with memory circuit elements, Proc. IEEE 100 (6) (2012) 2071–2080. Available from: https://doi.org/10.1109/JPROC.2011.2166369.

[35] S. Kim, C. Du, P. Sheridan, W. Ma, S.H. Choi, W.D. Lu, Experimental demonstration of a second-order memristor and its ability to biorealistically implement synaptic plasticity, Nano Lett. 15 (3) (2015) 2203–2211. Available from: https://doi.org/10.1021/acs.nanolett.5b00697.

[36] G.S. Snider, Spike-timing-dependent learning in memristive devices, IEEE/ACM Int. Symposium Nanoscale Architectures (NANOARCH 2008) (2008) 85–92. Available from: https://doi.org/10.1109/NANOARCH.2008.4585796.

[37] S.H. Jo, T. Chang, I. Ebong, B.B. Bhadviya, P. Mazumder, W. Lu, Nanoscale memristor device as synapse in neuromorphic systems, Nano Lett. 10 (4) (2010) 1297–1301. Available from: https://doi.org/10.1021/nl904092h.

[38] S. Yu, Y. Wu, R. Jeyasingh, D. Kuzum, H.-S.P. Wong, An electronic synapse device based on metal oxide resistive switching memory for neuromorphic computation, IEEE Trans. Electron. Devices 58 (8) (2011) 2729–2737. Available from: https://doi.org/10.1109/TED.2011.2147791.

[39] K. Seo, I. Kim, S. Jung, M. Jo, S. Park, J. Park, et al., Analog memory and spike-timing-dependent plasticity characteristics of a nanoscale titanium oxide bilayer resistive switching device, Nanotechnology 22 (25) (2011) 254023. Available from: https://doi.org/10.1088/0957-4484/22/25/254023.

[40] T. Serrano-Gotarredona, T. Masquelier, T. Prodromakis, G. Indiveri, B. Linares-Barranco, STDP and STDP variations with memristors for spiking neuromorphic learning systems, Front. Neurosci. 7 (2013) 2. Available from: https://doi.org/10.3389/fnins.2013.00002.

[41] I.-T. Wang, Y.-C. Lin, Y.-F. Wang, C.-W. Hsu, T.-H. Hou, 3D synaptic architecture with ultralow sub-10 fJ energy per spike for neuromorphic computation, IEEE Int. Electron. Devices Meet. (IEDM) (2014) 665–668. Available from: https://doi.org/10.1109/IEDM.2014.7047127.

[42] M. Prezioso, F. Merrikh Bayat, B. Hoskins, K. Likharev, D. Strukov, Self-adaptive spike-time-dependent plasticity of metal-oxide memristors, Sci. Rep. 6 (2016) 21331. Available from: https://doi.org/10.1038/srep21331.

[43] Z. Wang, S. Joshi, S.E. Savel'ev, H. Jiang, R. Midya, P. Lin, et al., Memristors with diffusive dynamics as synaptic emulators for neuromorphic computing, Nat. Mater. 16 (2017) 101–108. Available from: https://doi.org/10.1038/nmat4756.

[44] S. Yu, Y. Wu, H.-S.P. Wong, Investigating the switching dynamics and multilevel capability of bipolar metal oxide resistive switching memory, Appl. Phys. Lett. 98 (10) (2011) 103514. Available from: https://doi.org/10.1063/1.3564883.

[45] S. Balatti, S. Larentis, D.C. Gilmer, D. Ielmini, Multiple memory states in resistive switching devices through controlled size and orientation of the conductive filament, Adv. Mater. 25 (10) (2013) 1474–1478. Available from: https://doi.org/10.1002/adma.201204097.

[46] L. Zhao, H.-Y. Chen, S.-C. Wu, Z. Jiang, S. Yu, T.-H. Hou, et al., Multi-level control of conductive nano-filament evolution in HfO_2 ReRAM by pulse-train operations, Nanoscale 6 (11) (2014) 5698–5702. Available from: https://doi.org/10.1039/C4NR00500G.

[47] A. Prakash, J. Park, J. Song, J. Woo, E.-J. Cha, H. Hwang, Demonstration of low power 3-bit multilevel cell characteristics in a TaO_x-based RRAM by stack engineering, IEEE Electron. Device Lett. 36 (1) (2015) 32–34. Available from: https://doi.org/10.1109/LED.2014.2375200.

[48] A. Athmanathan, M. Stanisavljevic, N. Papandreou, H. Pozidis, E. Eleftheriou, Multilevel-cell phase-change memory: a viable technology, IEEE J. Emerg. Sel. Top. Circuits Syst. (JETCAS) 6 (1) (2016) 87–100. Available from: https://doi.org/10.1109/JETCAS.2016.2528598.

[49] D. Kuzum, R.G.D. Jeyasingh, B. Lee, H.-S.P. Wong, Nanoelectronic programmable synapses based on phase change materials for brain-inspired computing, Nano Lett. 12 (5) (2012) 2179–2186. Available from: https://doi.org/10.1021/nl201040y.

[50] A.F. Vincent, J. Larroque, N. Locatelli, N. Ben Romdhane, O. Bichler, C. Gamrat, et al., Spin-transfer torque magnetic random access memory as a stochastic memristive synapse for neuromorphic systems, IEEE Trans. Biomed. Circ. Syst. 9 (2) (2015) 166–174. Available from: https://doi.org/10.1109/TBCAS.2015.2414423.

[51] G. Srinivasan, A. Sengupta, K. Roy, Magnetic tunnel junction based long-term short-term stochastic synapse for a spiking neural network with on-chip STDP learning, Sci. Rep. 6 (2016) 29545. Available from: https://doi.org/10.1038/srep29545.

[52] M.-H. Wu, M.-C. Hong, C.-C. Chang, P. Sahu, J.-H. Wei, H.-Y. Lee, et al., Extremely compact integrate-and-fire STT-MRAM neuron: a pathway toward all-spin artificial deep neural network, IEEE Symposium VLSI Technol. (VLSI Technol.) (2019) T34–T35. Available from: https://doi.org/10.23919/VLSIT.2019.8776569.

[53] M. Suri, O. Bichler, D. Querlioz, O. Cueto, L. Perniola, V. Sousa, et al., Phase change memory as synapse for ultra-dense neuromorphic systems: application to complex visual pattern extraction, IEEE Int. Electron. Devices Meet. (IEDM) (2011) 79–82. Available from: https://doi.org/10.1109/IEDM.2011.6131488.

[54] M. Suri, O. Bichler, D. Querlioz, G. Palma, E. Vianello, D. Vuillaume, et al., CBRAM devices as binary synapses for low-power stochastic neuromorphic systems: auditory (cochlea) and visual (retina) cognitive processing applications, IEEE Int. Electron. Devices Meet. (IEDM) (2012) 235–238. Available from: https://doi.org/10.1109/IEDM.2012.6479017.

[55] O. Bichler, M. Suri, D. Querlioz, D. Vuillaume, B. De Salvo, C. Gamrat, Visual pattern extraction using energy-efficient '2-PCM synapse' neuromorphic architecture, IEEE Trans. Electron. Devices 59 (8) (2012) 2206–2214. Available from: https://doi.org/10.1109/TED.2012.2197951.

[56] S. Ambrogio, S. Balatti, F. Nardi, S. Facchinetti, D. Ielmini, Spike-timing dependent plasticity in a transistor-selected resistive switching memory, Nanotechnology 24 (2013) 384012. Available from: https://doi.org/10.1088/0957-4484/24/38/384012.

[57] D. Garbin, E. Vianello, O. Bichler, Q. Rafhay, C. Gamrat, G. Ghibaudo, et al., HfO_2-based OxRAM devices as synapses for convolutional neural networks, IEEE Trans. Electron. Devices 62 (8) (2015) 2494–2501. Available from: https://doi.org/10.1109/TED.2015.2440102.

[58] S. Ambrogio, N. Ciocchini, M. Laudato, V. Milo, A. Pirovano, P. Fantini, et al., Unsupervised learning by spike timing dependent plasticity in phase change memory (PCM) synapses, Front. Neurosci. 10 (2016) 56. Available from: https://doi.org/10.3389/fnins.2016.00056.

[59] S. Ambrogio, S. Balatti, V. Milo, R. Carboni, Z. Wang, A. Calderoni, et al., Neuromorphic learning and recognition with one-transistor-one-resistor synapses and bistable metal oxide RRAM, IEEE Trans. Electron. Devices 63 (4) (2016) 1508–1515. Available from: https://doi.org/10.1109/TED.2016.2526647.

[60] V. Milo, G. Pedretti, R. Carboni, A. Calderoni, N. Ramaswamy, S. Ambrogio, et al., Demonstration of hybrid CMOS/RRAM neural networks with spike time/rate-dependent

plasticity, IEEE Int. Electron. Devices Meet. (IEDM) (2016) 440–443. Available from: https://doi.org/10.1109/IEDM.2016.7838435.

[61] S. La Barbera, D.R.B. Ly, G. Navarro, N. Castellani, O. Cueto, G. Bourgeois, et al., Narrow heater bottom electrode-based phase change memory as a bidirectional artificial synapse, Adv. Electron. Mater. 4 (2018) 1800223. Available from: https://doi.org/10.1002/aelm.201800223.

[62] Z.-Q. Wang, S. Ambrogio, S. Balatti, D. Ielmini, A 2-transistor/1-resistor artificial synapse capable of communication and stochastic learning for neuromorphic systems, Front. Neurosci. 8 (2015) 438. Available from: https://doi.org/10.3389/fnins.2014.00438.

[63] S. Kim, M. Ishii, S. Lewis, T. Perri, M. BrightSky, W. Kim, et al., NVM neuromorphic core with 64k-cell (256-by-256) phase change memory synaptic array with on-chip neuron circuits for continuous in-situ learning, IEEE Int. Electron. Devices Meet. (IEDM) (2015) 443–446. Available from: https://doi.org/10.1109/IEDM.2015.7409716.

[64] M.V. Nair, L.K. Muller, G. Indiveri, A differential memristive synapse circuit for on-line learning in neuromorphic computing systems, Nano Futures 1 (2017) 035003. Available from: https://doi.org/10.1088/2399-1984/aa954a.

[65] I. Boybat, M. Le Gallo, S.R. Nandakumar, T. Moraitis, T. Parnell, T. Tuma, et al., Neuromorphic computing with multi-memristive synapses, Nat. Commun. 9 (2018) 2514. Available from: https://doi.org/10.1038/s41467-018-04933-y.

[66] G. Piccolboni, G. Molas, J.M. Portal, R. Coquand, M. Bocquet, D. Garbin, et al., Investigation of the potentialities of vertical resistive RAM (VRRAM) for neuromorphic applications, IEEE Int. Electron. Devices Meet. (IEDM) (2015) 447–450. Available from: https://doi.org/10.1109/IEDM.2015.7409717.

[67] H. Li, K.-S. Li, C.-H. Lin, J.-L. Hsu, W.-C. Chiu, M.-C. Chen, et al., Four-layer 3D vertical RRAM integrated with FinFET as a versatile computing unit for brain-inspired cognitive information processing, IEEE Symposium VLSI Technol. (VLSI Technol.) (2016) 1–2. Available from: https://doi.org/10.1109/VLSIT.2016.7573431.

[68] J. Shi, S.D. Ha, Y. Zhou, F. Schoofs, S. Ramanathan, A correlated nickelate synaptic transistor, Nat. Commun. 4 (2013) 2676. Available from: https://doi.org/10.1038/ncomms3676.

[69] E.J. Fuller, F. El Gabaly, F. Léonard, S. Agarwal, S.J. Plimpton, R.B. Jacobs-Gedrim, et al., Li-ion synaptic transistor for low power analog computing, Adv. Mater. 29 (2017) 1604310. Available from: https://doi.org/10.1002/adma.201604310.

[70] J. Tang, D. Bishop, S. Kim, M. Copel, T. Gokmen, T. Todorov, et al., ECRAM as scalable synaptic cell for high-speed, low-power neuromorphic computing, IEEE Int. Electron. Devices Meet. (IEDM) (2018) 292–295. Available from: https://doi.org/10.1109/IEDM.2018.8614551.

[71] E.J. Fuller, et al., Parallel programming of an ionic floating-gate memory array for scalable neuromorphic computing, Science 364 (6440) (2019) 570–574. Available from: https://doi.org/10.1126/science.aaw5581.

[72] H. Mulaosmanovic, J. Ocker, S. Müller, M. Noack, J. Müller, P. Polakowski, et al., Novel ferroelectric FET based synapse for neuromorphic systems, IEEE Symposium VLSI Technol. (VLSI Technol.) (2017) T176–T177. Available from: https://doi.org/10.23919/VLSIT.2017.7998165.

[73] M. Cubukcu, O. Boulle, M. Drouard, K. Garello, C.O. Avci, I.M. Miron, et al., Spin-orbit torque magnetization switching of a three-terminal perpendicular magnetic tunnel junction, Appl. Phys. Lett. 104 (2014) 042406. Available from: https://doi.org/10.1063/1.4863407.

[74] V.K. Sangwan, H.S. Lee, H. Bergeron, I. Balla, M.E. Beck, K.S. Chen, et al., Multi-terminal memtransistors from polycrystalline monolayer molybdenum disulfide, Nature 554 (7693) (2018) 500–504. Available from: https://doi.org/10.1038/nature25747.

[75] M.R. Azghadi, B. Linares-Barranco, D. Abbott, P.H.W. Leong, A hybrid CMOS-memristor neuromorphic synapse, IEEE Trans. Biomed. Circuits Syst. 11 (2) (2017) 434–445. Available from: https://doi.org/10.1109/TBCAS.2016.2618351.

[76] W. He, K. Huang, N. Ning, K. Ramanathan, G. Li, Y. Jiang, et al., Enabling an integrated rate-temporal learning scheme on memristor, Sci. Rep. 4 (2014) 4755. Available from: https://doi.org/10.1038/srep04755.

[77] T. Ohno, T. Hasegawa, T. Tsuruoka, K. Terabe, J.K. Gimzewski, M. Aono, Short-term plasticity and long-term potentiation mimicked in single inorganic synapses, Nat. Mater. 10 (8) (2011) 591–595. Available from: https://doi.org/10.1038/nmat3054.

[78] T. Werner, E. Vianello, O. Bichler, A. Grossi, E. Nowak, J.-F. Nodin, et al., Experimental demonstration of short and long-term synaptic plasticity using OxRAM multi k-bit arrays for reliable detection in highly noisy input data, IEEE Int. Electron. Devices Meet. (IEDM) (2016) 432–435. Available from: https://doi.org/10.1109/IEDM.2016.7838433.

[79] W. Wang, A. Bricalli, M. Laudato, E. Ambrosi, E. Covi, D. Ielmini, Physics-based modeling of volatile resistive switching memory (RRAM) for crosspoint selector and neuromorphic computing, IEEE Int. Electron. Devices Meet. (IEDM) (2018) 932–935. Available from: https://doi.org/10.1109/IEDM.2018.8614556.

[80] V. Milo, G. Pedretti, R. Carboni, A. Calderoni, N. Ramaswamy, S. Ambrogio, et al., A 4-transistors/1-resistor hybrid synapse based on resistive switching memory (RRAM) capable of Spike-Rate-Dependent Plasticity (SRDP), Trans. Very Large Integr. (VLSI) Syst. 26 (12) (2018) 2806–2815. Available from: https://doi.org/10.1109/TVLSI.2018.2818978.

[81] J. Gjorgjieva, C. Clopath, J. Audet, J.-P. Pfister, A triplet spike-timing-dependent plasticity model generalizes the Bienenstock-Cooper-Munro rule to higher-order spatiotemporal correlations, Proc. Natl. Acad. Sci. USA 108 (48) (2011) 19383–19388. Available from: https://doi.org/10.1073/pnas.1105933108.

[82] A.A. Faisal, L.P.J. Selen, D.M. Wolpert, Noise in the nervous system, Nat. Rev. Neurosci. 9 (4) (2008) 292–303. Available from: https://doi.org/10.1038/nrn2258.

[83] A. Bricalli, E. Ambrosi, M. Laudato, M. Maestro, R. Rodriguez, D. Ielmini, Resistive switching device technology based on silicon oxide for improved on-off ratio—Part II: Select devices, IEEE Trans. Electron. Devices 65 (1) (2018) 122–128. Available from: https://doi.org/10.1109/TED.2017.2776085.

[84] S. Ambrogio, S. Balatti, D.C. Gilmer, D. Ielmini, Analytical modeling of oxide-based bipolar resistive memories and complementary resistive switches, IEEE Trans. Electron. Devices 61 (7) (2014) 2378–2386. Available from: https://doi.org/10.1109/TED.2014.2325531.

[85] D. Garbin, O. Bichler, E. Vianello, Q. Rafhay, C. Gamrat, L. Perniola, et al., Variability-tolerant convolutional neural network for pattern recognition applications based on OxRAM synapses, IEEE Int. Electron. Devices Meet. (IEDM) (2014) 661–664. Available from: https://doi.org/10.1109/IEDM.2014.7047126.

[86] G.W. Burr, R.M. Shelby, C. di Nolfo, J.W. Jang, R.S. Shenoy, P. Narayanan, et al., Experimental demonstration and tolerancing of a large-scale neural network (165,000 synapses) using phase-change memory as the synaptic weight element, IEEE Int. Electron. Devices Meet. (IEDM) (2014) 697–700. Available from: https://doi.org/10.1109/IEDM.2014.7047135.

[87] M. Prezioso, F. Merrikh-Bayat, B.D. Hoskins, G.C. Adam, K.K. Likharev, D.B. Strukov, Training and operation of an integrated neuromorphic network based on metal-oxide

memristors, Nature 521 (7550) (2015) 61–64. Available from: https://doi.org/10.1038/nature14441.

[88] S. Yu, Z. Li, P.-Y. Chen, H. Wu, B. Gao, D. Wang, et al., Binary neural network with 16 Mb RRAM macro chip for classification and online training, IEEE Int. Electron. Devices Meet. (IEDM) (2016) 416–419. Available from: https://doi.org/10.1109/IEDM.2016.7838429.

[89] P. Yao, H. Wu, B. Gao, S.B. Eryilmaz, X. Huang, W. Zhang, et al., Face classification using electronic synapses, Nat. Commun. 8 (2017) 15199. Available from: https://doi.org/10.1038/ncomms15199.

[90] R. Mochida, K. Kouno, Y. Hayata, M. Nakayama, T. Ono, H. Suwa, et al., A 4M synapses integrated analog ReRAM based 66.5 TOPS/W neural-network processor with cell current controlled writing and flexible network architecture, IEEE Symposium VLSI Technol. (VLSI Technol.) (2018) 175–176. Available from: https://doi.org/10.1109/VLSIT.2018.8510676.

[91] F. Merrikh-Bayat, M. Prezioso, B. Chakrabarti, H. Nili, I. Kataeva, D. Strukov, Implementation of multilayer perceptron network with highly uniform passive memristive crossbar circuits, Nat. Commun. 9 (2018) 2331. Available from: https://doi.org/10.1038/s41467-018-04482-4.

[92] C. Li, D. Belkin, Y. Li, P. Yan, M. Hu, N. Ge, et al., Efficient and self-adaptive in-situ learning in multilayer memristor neural networks, Nat. Commun. 9 (2018) 2385. Available from: https://doi.org/10.1038/s41467-018-04484-2.

[93] V. Milo, C. Zambelli, P. Olivo, E. Pérez, M.K. Mahadevaiah, O.G. Ossorio, et al., Multilevel HfO_2-based RRAM devices for low-power neuromorphic networks, APL. Mater. 7 (2019) 081120. Available from: https://doi.org/10.1063/1.5108650.

[94] S. Yu, B. Gao, Z. Fang, H. Yu, J. Kang, H.-S.P. Wong, Stochastic learning in oxide binary synaptic device for neuromorphic computing, Front. Neurosci. 7 (2013) 186. Available from: https://doi.org/10.3389/fnins.2013.00186.

[95] E. Covi, S. Brivio, A. Serb, T. Prodromakis, M. Fanciulli, S. Spiga, Analog memristive synapse in spiking networks implementing unsupervised learning, Front. Neurosci. 10 (2016) 482. Available from: https://doi.org/10.3389/fnins.2016.00482.

[96] A. Serb, J. Bill, A. Khiat, R. Berdan, R. Legenstein, T. Prodromakis, Unsupervised learning in probabilistic neural networks with multi-state metal-oxide memristive synapses, Nat. Commun. 7 (2016) 12611. Available from: https://doi.org/10.1038/ncomms12611.

[97] T. Werner, E. Vianello, O. Bichler, D. Garbin, D. Cattaert, B. Yvert, et al., Spiking neural networks based on OxRAM synapses for real-time unsupervised spike sorting, Front. Neurosci. 10 (2016) 474. Available from: https://doi.org/10.3389/fnins.2016.00474.

[98] G. Pedretti, V. Milo, S. Ambrogio, R. Carboni, S. Bianchi, A. Calderoni, et al., Memristive neural network for on-line learning and tracking with brain-inspired spike timing dependent plasticity, Sci. Rep. 7 (2017) 5288. Available from: https://doi.org/10.1038/s41598-017-05480-0.

[99] G. Pedretti, V. Milo, S. Ambrogio, R. Carboni, S. Bianchi, A. Calderoni, et al., Stochastic learning in neuromorphic hardware via spike timing dependent plasticity with RRAM synapses, IEEE J. Emerg. Top. Circuits Syst. (JETCAS) 8 (1) (2018) 77–85. Available from: https://doi.org/10.1109/JETCAS.2017.2773124.

[100] G. Pedretti, S. Bianchi, V. Milo, A. Calderoni, N. Ramaswamy, D. Ielmini, Modeling-based design of brain-inspired spiking neural networks with RRAM learning synapses, IEEE Int. Electron. Devices Meet. (IEDM) (2017) 653–656. Available from: https://doi.org/10.1109/IEDM.2017.8268467.

[101] Z. Wang, S. Joshi, S. Savel'ev, W. Song, R. Midya, Y. Li, et al., Fully memristive neural networks for pattern classification with unsupervised learning, Nat. Electron. 1 (2018) 137−145. Available from: https://doi.org/10.1038/s41928-018-0023-2.

[102] W. Wang, G. Pedretti, V. Milo, R. Carboni, A. Calderoni, N. Ramaswamy, et al., Learning of spatiotemporal patterns in a spiking neural network with resistive switching synapses, Sci. Adv. 4 (9) (2018) eaat4752. Available from: https://doi.org/10.1126/sciadv.aat4752.

[103] T. Dalgaty, E. Vianello, D. Ly, G. Indiveri, B. De Salvo, E. Nowak, et al., Insect-inspired elementary motion detection embracing resistive memory and spiking neural networks, in: V. Vouloutsi, et al. (Eds.), Biomimetic and Biohybrid Systems. Living Machines 2018. Lecture Notes in Computer Science, vol. 10928, Springer, Cham, 2018. Available from: https://doi.org/10.1007/978-3-319-95972-6_13.

[104] D. Ly, A. Grossi, C. Fenouillet-Beranger, E. Nowak, D. Querlioz, E. Vianello, Role of synaptic variability in resistive memory-based spiking neural networks with unsupervised learning, J. Phys. D: Appl. Phys. 51 (2018) 44. Available from: https://doi.org/10.1088/1361-6463/aad954.

[105] T. Masquelier, R. Guyonneau, S.J. Thorpe, Competitive STDP-based spike pattern learning, Neural Computation 21 (5) (2009) 1259−1276. Available from: https://doi.org/10.1162/neco.2008.06-08-804.

Chapter 18

System-level integration in neuromorphic co-processors

Giacomo Indiveri[1], Bernabé Linares-Barranco[2] and Melika Payvand[1]
[1]*Institute of Neuroinformatics, University of Zurich and ETH Zurich, Zürich, Switzerland,* [2]*Instituto de Microelectrónica de Sevilla IMSE-CNM, CSIC and Universidad de Sevilla, Sevilla, Spain*

18.1 Neuromorphic computing

Neuromorphic computing systems typically comprise neuron and synapse circuits arranged in a massively parallel manner to support the emulation of large-scale spiking neural networks. Different approaches have been proposed for hardware implementations of neuromorphic computing systems, ranging from digital CMOS ones based on synchronous [1] and asynchronous logic [2,3], to analog and mixed-signal ones based on standard strong inversion circuits [4] and weak-inversion circuits [5–7]. Although implemented in pure conventional CMOS technology, most of these neuromorphic architectures are optimally suited for cointegration with memristive devices, which can be used to both emulate synaptic function and to support nonvolatile local storage of network parameters [8,9]. In such architectures, memory elements (e.g., that store the synaptic state) are used also as computing elements (i.e., that convert input pulses into weighted synaptic currents) and are placed just next to the main processing units (i.e., the neurons that integrate all synaptic inputs and produce output spikes). These architectures are radically different from the ones based on the classical von Neumann computer one, in which memory and compute elements are implemented in separate and distinct blocks that exchange data across a common shared bus, as quickly as possible. The spiking neural networks implemented by the neuromorphic architectures can be configured to carry out multiple types of signal processing tasks, ranging from sensory signal processing [10] to pattern recognition [6], to finite-state-machine like computation [11]. These spiking neural networks represent new brain-inspired computing paradigms and the hardware architectures that implement them have the potential of solving the von Neumann memory bottleneck problem [12] efficiently [13]: given their colocalization of memory and computation, no fast exchange of data across

Memristive Devices for Brain-Inspired Computing. DOI: https://doi.org/10.1016/B978-0-08-102782-0.00018-6

different memory and compute blocks takes place. In addition, these architectures can minimize power consumption by performing data-driven computation (i.e., carrying out computations only when there are data to drive the circuits), and processing the data at a rate that matches the time constants of the input signals and the real-time requirements of the task at hand. Setting the time constants of the processing elements to match those of the signals that need to be processed can reduce the power consumption and data bandwidth requirements by orders of magnitude, compared to synchronous clock-driven conventional computer approaches. Despite the use of slow, reduced precision, and variable computational elements, these architectures can achieve fast, robust, and reliable computation by virtue of their massively parallel mode of operation. Given this design strategy, these architectures can also exhibit remarkable fault-tolerance features, by taking advantage of the inherent redundancy in the use of their components. Indeed, while the memory-related constraints of conventional computers require high-speed data transfers using reliable bit-precise devices, these brain-inspired computing systems, as well as the biological nervous systems they emulate, are able to perform fast and robust computation, using memory and computing elements that are slow, in-homogeneous, stochastic, and faulty [8,14,15].

18.2 Integrating memristive devices as synapses in neuromorphic computing architectures

Typically in neuromorphic computing architectures, very large arrays of synaptic elements are connected to a smaller number of neuron circuits. In addition to being compatible with designs that faithfully model biological neural networks, architectures comprising neurons connected to a large number of many parallel synapses support the implementation of a wide range of spiking neural network topologies, including multilayer or deep neural networks and recurrent neural networks. In order to configure the desired network topology and program the connectivity among neurons, neuromorphic systems typically assign an address to each neuron and encode the spikes produced by such neurons as "address events." In these systems, information is typically encoded in the timing of the address events. Specifically, the interval between successive address events produced by the same neuron (i.e., the inter-spike interval) can be used to represent analog values. Neural processing of analog variables can be achieved by using the digital address event representation (AER) [16–19] and connecting multiple neurons together with different types of AER connectivity schemes. Spikes produced by source neurons are transmitted to one or more destination synapse circuits that integrate them with different gain factors and convey them to the post-synaptic neuron. Unlike classical digital logic circuits, these networks are typically characterized by very large fan-in and fan-out numbers. For example, in cortical networks, neurons project on average to about 10,000

destinations. The type of processing and functionality of these spiking neural networks is determined by the their specific structure and parameters, such as the properties of the neurons or the weights of the synapses [20]. It is therefore important to design neuromorphic computing platforms that can be configured to support the construction of different network topologies, with different neuron and synapse properties. This requires the development of both configurable neuron/synapse circuits, and of programmable event-based routing and communication schemes. The latter elements are particularly important, because the scalability of neuromorphic systems is mainly restricted by communication requirements. Figs. 18.1 and 18.2 show examples of architectures that follow two complementary approaches for achieving the implementation of large-scale neural networks. Each of the populations of neurons shown in these figures could be implemented in a single "core" and multipopulation (e.g., multilayer) networks can be implemented by designing multicore neuromorphic processors [1–3,7].

The high-density multicore approach of Fig. 18.1 aims to capitalize on the nanoscale size of memristive devices and uses them as simple synaptic elements in dense cross-bar arrays [21–23]. In this approach the single synapse element is kept as simple and compact as possible using either single passive memristive device element per synapse arranged in "1R" cross-bar arrays, or single memristive devices connected to a "select" transistor or "selector" device in "1T-1R" cross-bar arrays (see also Chapter 5, Selector Devices for Emerging Memories). The state of a specific memristive synapse, corresponding to its synaptic weight, can be changed by appropriately setting the voltage values of the row and column lines connected to the top and bottom electrodes of the target device (e.g., with a large voltage difference ΔV), while setting the ΔV of all other devices to a low value (or ideally zero). Programming circuits that implement these operations can be designed and placed at the periphery of the cross-bar arrays. Similarly these

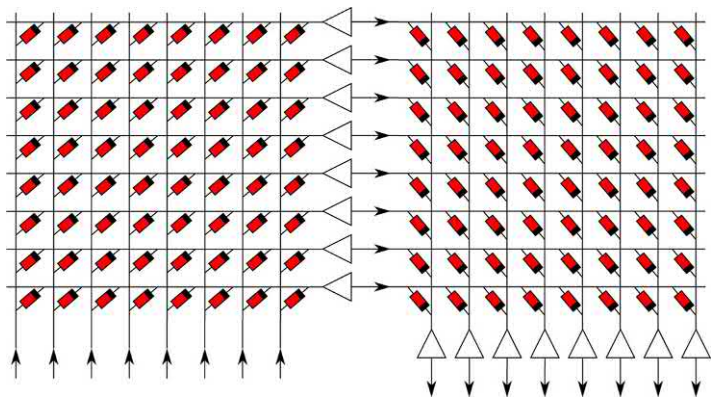


FIGURE 18.1 Example of crossbar multineuron architecture.

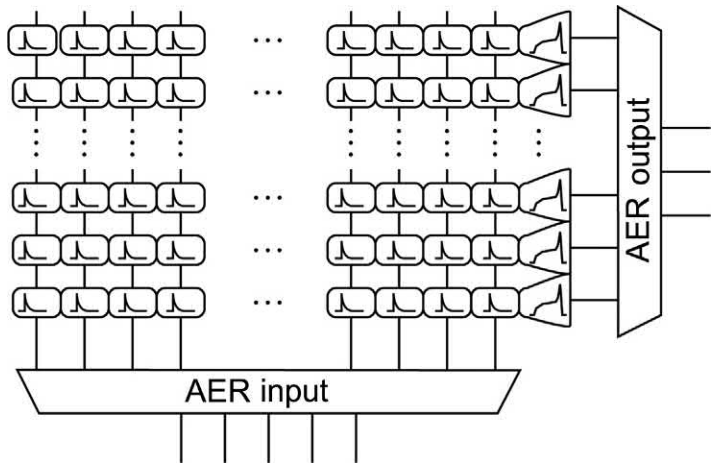


FIGURE 18.2 Example of address event representation (AER) multineuron architecture.

programming blocks can be driven by spike-based learning circuits that evaluate the state of the input and output pulses and produce an appropriate voltage waveform to be applied between the top and bottom electrodes of the addressed synapse. While this approach has the advantage of being able to support very dense cross-bar array fabrications, it has the disadvantage that it requires large overhead circuitry at the periphery for the address encoders, decoders, programming logic, and voltage drivers. Depending on the technology node used, these peripheral circuits can be made quite small. However, since the technology has to support sufficiently large voltage values for forming the memristive devices and updating their state, the corresponding minimum feature size is typically not very small and the overall area used by the overhead circuitry can become prohibitive for typical neuromorphic computing designs. Another disadvantage of this approach, in case on-chip learning features are desired, is given by the fact that learning and weight updates require the cross-bar row and column select lines to be actively driven for the full duration of the memristive device "set" operation, which is determined by the length of the ΔV pulse used to change the memristor state. Therefore the bandwidth of the data flow in these architectures is limited by the minimum duration of the ΔV pulse. Since typical spike timing-dependent plasticity (STDP) learning protocols require presynaptic input pulses to overlap with postsynaptic output pulses in order to produce the right ΔV pulse, these operations can keep the row and column select lines busy for very long periods, reaching even milliseconds (e.g., for learning protocols that model real synapses and/or that are used for processing sensory signals), and severely limit the overall system bandwidth, as well as power consumption.

Conversely the multicore approach of Fig. 18.2 forgoes the attempt to maximize density at the cross-bar level to allow the use of larger synapse blocks at the benefit of adding additional complexity within each synapse and enabling more sophisticated and massively parallel computations. Examples of more complex "compound" synapses that comprise multiple memristive devices per synapse have been shown to enable more precise modulation of the synaptic weight over a wide dynamic range [24] and to reduce the effect of device variability [9] (Fig. 18.3). Following this approach, both input spikes into the individual synapses and output spikes generated by the postsynaptic neurons are short asynchronous digital pulses encoded using the AER protocol. If the activity of the neuron circuits is sparse and their firing rates are biologically plausible (e.g., ranging from a few spikes per second to a few hundred spikes per second), then it is possible to trade-off space with speed very effectively, by time-multiplexing a single (very fast) digital bus to represent many (very slow) neuron axons. The AER input circuits in Fig. 18.2 receive input address events and decode them to stimulate corresponding columns of synaptic cells. These circuits transmit

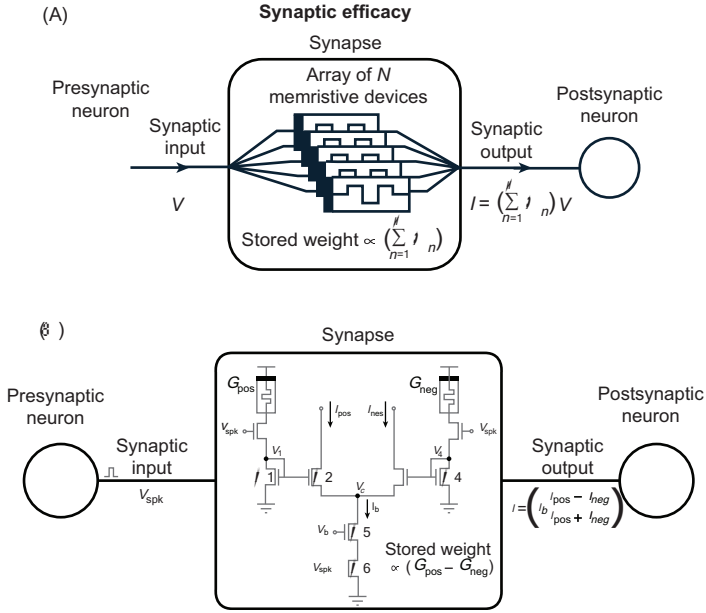


FIGURE 18.3 Example of compound synapse blocks: (A) multimemristive synapse concept, in which the synaptic weight is represented by the combined conductance of multiple devices; (B) differential memristive synapse circuit, in which voltage pulses are directly converted to currents via current-mode normalizer circuits. *Adapted from (A) I. Boybat, M.L. Gallo, T. Moraitis, et al., "Neuromorphic computing with multi-memristive synapses," Nat. Commun., 9, 2018, p. 2514, and (B) M.V. Nair and G. Indiveri, A differential memristive current-mode circuit, European Patent Application EP 17183461.7, Filed 27.07.2017, 2017.*

the events as soon as they arrive and as quickly as possible (e.g., within a few nano-seconds). This frees the shared communication bus to transmit spikes to the cross-bar array, increasing the throughput of the network by means of time-multiplexed communication resources. Integrators and pulse-extenders can be used inside each synapse circuit in the array to convert the fast AER pulses into slower dynamic signals to emulate synaptic and neural dynamics. On the output side, an AER output circuit arbitrates the spikes produced by the neurons and queues output events in case of collisions. The circuit converts the asynchronous spikes produced by the neurons into fast address events and transmits them on the shared output bus [17].

In both high-density and high-complexity memristive synapse approaches, the neurons are typically implemented using Integrate and Fire (I&F) models: they receive input currents that represent the weighted sum of all synaptic contributions, integrate them over time, and produce an output spike when their integrated value exceeds the spiking threshold. As a consequence, the timing of the output spike is directly related to the amplitude of the total input current, which in turn depends on the relationship between the timing of the input spike on each synapse and its synaptic weight. In addition to the precise timing of spikes, the neurons can also encode signals with their average firing rates (the average number of spikes produced per second), which are also proportional to their input currents. Therefore the same neuron circuit and architecture can be used to carry out computation on the precise timing of the input/output network signals, or on analog variables encoded in the average firing rates of inputs and outputs. Indeed, it is even possible to use both signal representations (single spike timing and average firing rate) together to carry out complex computations.

18.3 Spike-based learning mechanisms for hybrid memristive-CMOS neuromorphic synapses

Synaptic plasticity plays a crucial role in allowing neural networks to learn and adapt to various input environments. Neuromorphic electronic systems that seek to emulate these learning abilities struggle to meet the seemingly incompatible requirements of reproducing complex plasticity mechanisms in each neuromorphic synapse circuit, while keeping the size of the plastic synapse circuit small, to integrate large numbers of synapses per neuron. While many attempts have been made in this respect with pure CMOS mixed-signal analog/digital circuits [2,6,26–28], several challenges related to the circuit size and the volatility of the learned synaptic weights are still open. In this respect, memristive devices can play a key role in the design of mixed memristive-CMOS learning mechanisms.

Here we review different spike-based plasticity models that lend themselves well to hardware implementation using mixed CMOS–memristive circuits. The learning mechanisms and their corresponding circuit design

solutions have different properties that trade-off the complexity of the plasticity model emulated with the size and complexity of the circuit proposed.

18.3.1 STDP mechanism

STDP is the ability of natural or artificial synapses to change their strength according to the precise timing of individual pre and/or postsynaptic spikes [29–32]. A comprehensive overview of STDP and its history can be found in Ref. [33]. STDP learning in biology is inherently asynchronous and *on line*, meaning that synaptic incremental update occurs while neurons and synapses transmit spikes and perform computations. This contrasts to more traditional learning rules, like back propagation [34,35], where first neurons and synapses perform signal aggregation and neural state update (we call this here "inference phase") and then error signals are computed and corresponding weight updates are applied (we call this here "weight update phase") and alternate these two phases during training.

Even early proposals for memristor-based STDP learning implementations used artificial time-multiplexing to alternate continuously and synchronously between "inference" and "weight update" phases, thus requiring global system-wide synchronization. This can become a severe handicap when scaling up systems to arbitrary size. However it is possible to do a fully asynchronous implementation of memristor-based STDP where "inference" and "weight update" phases happen simultaneously in a natural manner, as in biology [36], where there is no need for any global synchronization.

Fig. 18.4A shows the change of synaptic strength (in percent) measured experimentally from biological synapses as a function of relative timing $\Delta T = t_{pos} - t_{pre}$ between the arrival time t_{pre} of a presynaptic spike and the time t_{pos} of generation of a postsynaptic spike. Although the data show stochasticity, we can infer an underlying interpolated function $\xi(\Delta T)$ as shown in Fig. 18.4B

$$\xi(\Delta T) = \begin{cases} a^{+} e^{-\Delta T/\tau^{+}} & if \quad \Delta T > 0 \\ -a^{-} e^{\Delta T/\tau^{-}} & if \quad \Delta T < 0 \end{cases} \tag{18.1}$$

For a causal pre to postspike timing relation ($\Delta T > 0$) the strength of the synapse is increased, while for an anti-causal relation ($\Delta T < 0$) it is decreased. In the case of synapses with negative synaptic strength (as in some artificial realizations), the reversed version shown in Fig. 18.4C can be used. Pure CMOS-based very large scale integration (VLSI) circuit implementations of STDP rules that follow the description of Eq. (18.1) have been reported [37,38], which result in more than about 30 transistors per plastic synapse, thus demonstrating the very high cost of their hardware realization. However combining memristive crossbars (as shown in Fig. 18.1) with neurons that comprise learning circuits which send spikes both forward and backward, it is possible to realize the STDP learning rule in a fully

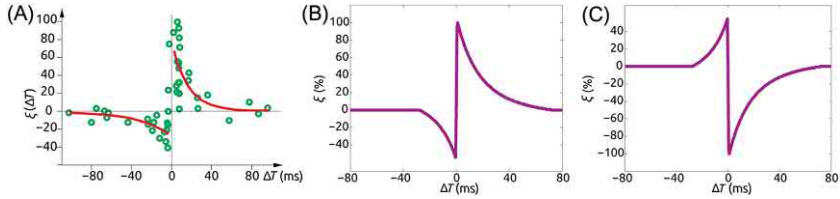


FIGURE 18.4 Spike timing-dependent plasticity (STDP) learning rule. (A) Experimentally measured STDP function $\xi(\Delta T)$ on biological synapses, (B) Ideal STDP update function used in computational models of STDP synaptic learning. (C) Anti-STDP learning function for inhibitory STDP synapses. *(A) Data adapted from G.-Q. Bi and M.-M. Poo, "Synaptic modifications in cultured hippocampal neurons: Dependence on spike timing, synaptic strength, and postsynaptic cell type," J. Neurosci., 18, 24, 1998, pp. 10464–10472.*

asynchronous manner [39]. Furthermore, by changing the analog shape of the forward and backward voltage pulses produced by the neuron learning circuits, it is possible to massage the STDP learning function. The main drawback of this approach is that the crossbar lines need to be kept "active" for sufficiently long times to guarantee proper overlap between pre and postsynaptic spikes (sometimes even for up to many milliseconds). This implies that crossbar lines are kept busy for long times, and also that important power is dissipated by the resistive memristors while the spikes are applied. Moreover, such design requires the design of amplifiers which can drive resistive loads. Hence, at least a two-stage amplifier is needed with a high current drive to support such load. However, a problem arises since the resistive load is changing in the process of learning: the compensation required to stabilize the amplifier should be designed to adapt to the changing load. Such design is not only complicated but also is power hungry and consumes a large amount of area on silicon. These issues however can be solved by engineering second-order memristors having internal dynamics, on which STDP can be induced without forcing pre and postsynaptic spikes to overlap in time [40]. Under these circumstances, spikes can be made fast in time and resistive power dissipation can be reduced to a minimum.

18.3.2 Spike timing- and rate-dependent plasticity mechanism

The spike timing- and rate-dependent plasticity (STRDP) model was proposed in Ref. [41], and is based on work originally presented in Ref. [42]. In this model synapses have two stable states on long-time scales (the potentiated and depressed states), but multiple transient states, on short timescales, that enable a gradual transition between the two stable states. The synaptic weight $w(t)$ is expressed as a function of the synapse internal state variable $X(t)$. Examples of such function can be equivalence $w(t) = X(t)$) or binary threshold: $w(t) = 1$ *if* $X(t) > w_{th}$ and 0 otherwise, where w_{th} is an arbitrary

threshold value. The internal state variable $X(t)$ is updated upon the arrival of a presynaptic spike, at time t_{pre}. The direction of the weight update (increase or decrease) depends on the value of the postsynaptic neuron membrane voltage $V_{mem}(t_{pre})$ (whether it is above or below a set threshold θ_V). An additional third factor, the variable $C(t)$, is used to model the intracellular calcium concentration and to determine whether to actually do the weight update or not. Specifically, upon the arrival of the presynaptic spike at time t_{pre}, the synaptic weight is updated according to the following equations:

$$\mathrm{X} \rightarrow \mathrm{X} + a \quad if \quad V_{\mathrm{mem}}(t_{\mathrm{pre}}) > \theta_V \quad and \quad \theta^l_{\mathrm{up}} < C(t_{\mathrm{pre}}) < \theta^h_{\mathrm{up}} \tag{18.2}$$

$$\mathrm{X} \rightarrow \mathrm{X} - b \quad if \quad V_{\mathrm{mem}}(t_{\mathrm{pre}}) \leq \theta_V \quad and \quad \theta^l_{\mathrm{down}} < C(t_{\mathrm{pre}}) < \theta^h_{\mathrm{down}} \tag{18.3}$$

where a and b are jump sizes, θ_V is a voltage threshold, and θ^l_{up}, θ^h_{up}, θ^l_{down}, and θ^h_{down} are thresholds on the calcium variable. In other words, $X(t)$ is increased if $V_{\mathrm{mem}}(t)$ is elevated (above θ_V) when the presynaptic spike arrives and decreased if $V_{\mathrm{mem}}(t)$ is lower than θ_V at time t_{pre}, provided that the calcium variable $C(t)$ is in the correct range. $C(t)$ is an auxiliary variable that corresponds to a low-pass filtered version of the postsynaptic spikes: $C(t)$ is incremented by J_{C} (which corresponds to magnitude of spike-triggered calcium influx into the cell) at each postsynaptic spike time t_i, and decays with a time constant τ_{C}:

$$\frac{\mathrm{dC}(t)}{\mathrm{d}t} = -\frac{1}{\tau_{\mathrm{C}}} C(t) + J_{\mathrm{C}} \sum_i \delta(t - t_i) \tag{18.4}$$

The dependence of the weight updates on $C(t)$ allows the learning rule to enable/disable the weight updates based on the long-term average of postsynaptic activity. This implements a "stop-learning" condition that allows the network to stop changing weights when it is performing correctly, and allows the system to keep the learning process always enabled, without having to separate a "training phase" from a "test phase" as is typically done in conventional artificial neural network applications.

In parallel to the spike-driven weight updates described above, $X(t)$ is continuously and slowly driven toward one of two stable values, depending on whether it is above or below an additional threshold parameter θ_X:

$$\frac{\mathrm{dX}}{\mathrm{d}t} = \alpha \quad \text{if} \quad X > \theta_X \tag{18.5}$$

$$\frac{dX}{dt} = -\beta \quad \text{if} \quad X \leq \theta_X \tag{18.6}$$

The state variable $X(t)$ is bounded above and below by the two stable states X_{high} and X_{low} that are not shown in the equations to simplify the notation. Fig. 18.5 illustrates the relevant waveforms and parameters of the spike-based voltage-dependent learning rule.

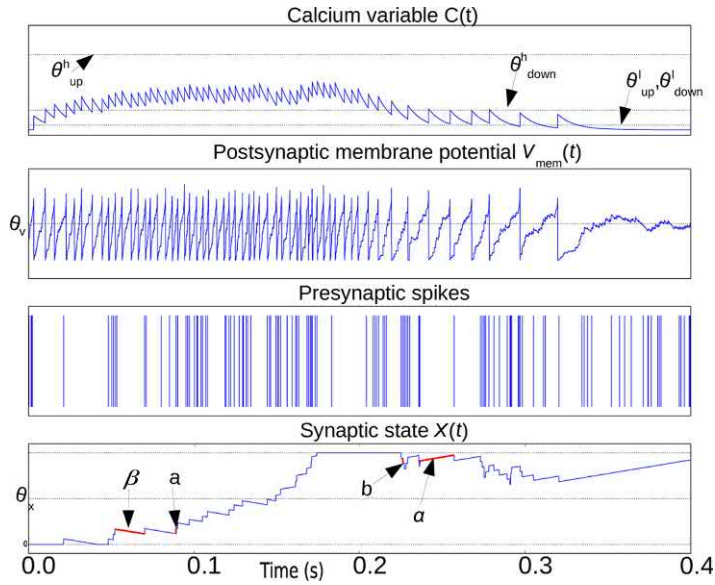


FIGURE 18.5 Illustration waveforms of the STDRP learning rule showing key parameters from Eqs. 18.2 to 18.6. *Adapted from H. Mostafa, A. Khiat, A. Serb, et al., "Implementation of a spike-based perceptron learning rule using TiO2-x memristors," Front. Neurosci., 9, 2015, 357. doi: 10.3389/fnins.2015.00357.*

Although this learning rule has been shown to reproduce, on average, the classical STDP phenomenology [41], it differs from the vast majority of STDP rules in that it does not explicitly depend on the precise timing of both pre and postsynaptic neuron spikes. The compatibility with the classical STDP learning rule comes about through the rule's dependence on the postsynaptic neuron's membrane potential: a presynaptic spike that occurs when the postsynaptic membrane potential is high will potentiate the synapse and will likely produce a postsynaptic spike shortly after. Thus the synapse tends to get potentiated in pre before post scenarios. The synapse also tends to get depressed in post before pre scenarios because the membrane potential is usually low for some time after a postsynaptic spike is emitted, and a presynaptic spike arriving in this interval will depress the synapse. However, as this rule also has access to postsynaptic neuron's rate information through the $C(t)$ signal, it can reproduce effects beyond classical pair-wise STDP, such as increased potentiation at high postsynaptic firing rates and increased depression at low postsynaptic firing rates [44].

18.3.3 Spike-based stochastic weight update rules

Filamentary memristive devices have large variations in their operational parameters that stem from the underlying switching mechanisms

that are based on filament formation. This switching mechanism exhibits stochastic behavior due to the thermally activated filament formation process [22,45–47]. Filament formation in memristive devices is typically bias-dependent and can be explained by the hopping of positively charged particles in a thermally activated process [22]. The hopping rate is exponentially related to the activation energy and linearly dependent in time:

$$\Gamma = 1/\tau = \upsilon e^{-E_a(V)/k_B T} \tag{18.7}$$

where υ is the attempt frequency for particle hopping, k_B is the Boltzmann constant, and T is the absolute temperature. The bias-dependent nature of the switching characteristics results in a stochastic process that follows a Poisson distribution. The Poisson distribution implies that the switching events are independent from one another. Therefore the probability of a switching event occurring within Δt at time t is:

$$P(t) = \frac{\Delta t}{\tau} e^{-t/\tau} \tag{18.8}$$

where τ is the characteristic wait time and is an exponential function of the voltage applied across the device:

$$\tau(V) = \tau_0 e^{-V/V_0} \tag{18.9}$$

The parameters τ_0 and V_0 are fitting parameters that depend on the physical characteristics of the memristive device and can be found by experimental measurements [22]. This intrinsic probabilistic property of memristive devices can be exploited for implementing stochastic learning in neuromorphic architectures to overcome the limitations and problems typically found with nonlinear conductance changes in deterministic learning approaches [48] and to reduce the network sensitivity to their variability [49].

The learning algorithm that we use for achieving such features is based on the *delta rule* [50]. This rule minimizes the cost function of a single-layer neural network defined as the error between a desired target value T and the variable y calculated as weighted sum of the network input:

$$y = \sum_i (w_i x_i) \tag{18.10}$$

where w_i and x_i are the i th synaptic weight and input, respectively. In the original formulation [50], the network inputs x_i were binary signals and the output was a thresholded and binarized function of the weighted sum variable. In neuromorphic architectures, the x_i inputs are spiking events and the network output corresponds to the activity of the spiking neurons. If one considers the average firing rate of the neurons, then the output is a sigmoidal

function of y. According to this rule, the weight change of the *ith* input synapse should be proportional to: $\Delta W_{ji} \propto (T - y)x_i$ [51]. If we consider binary synapses with synaptic weights represented by set or reset memristive devices, the analog nature of the weight change can be interpreted as the probability of switching rather than a gradual change in the device conductance. Taking into account the stochastic filament formation behavior described above, the synapse probability of switching for $t << \tau$ can be written as

$$P(t) = \frac{\Delta t}{\tau} = \frac{\Delta t}{\tau_0 e^{V/V_0}} \tag{18.11}$$

To map this stochastic property with the delta rule, the voltage across the device should be set to:

$$V/V_0 = \ln(T - y) \tag{18.12}$$

Therefore with this setting, upon the arrival of an input spike event x_i, the probability of switching becomes:

$$P(t) = \Delta t e^{\ln(T_j - y_j)} x_i = \Delta t(T - y)x_i \tag{18.13}$$

The requirement to linearly change synaptic weights with gradual changes in memristive device conductances has been converted to switching binary devices with a probability that is linearly proportional to the error.

The circuits required to implement this stochastic rule on a neuromorphic chip have been recently proposed in Ref. [49]. They comprise the following:

- a block that compares the sum of the synaptic currents (obtained by simply using Kirchhoff's current law on the neuron's input node) with a target current provided by the system.
- a circuit that produces a ramping voltage whose slope is proportional to the logarithmic value of the error, as specified in Eq. (18.2). Such ramp voltage has been shown to be able to modulate the probabilty of the switching of the device depending on the final value it reaches by the end of a programming time [52].

Behavioral simulations of such probabilistic mechanism [49] have shown that this stochastic learning rule provides promising results, for example in classical benchmark tasks, such as the classification of handwritten MNIST characters [53], and that performance improves when combining memristive single devices into compound multimemristive devices, as explained in Section 18.2.

Recent results on STDP learning with binary weights following stochastic updates with additional regularizations, such as homeostasis and moving and double thresholds [54], present an excellent potential to be exploited on memristors restricted to binary values.

18.3.4 Comparison between the spike-based learning architectures

STDP and STRDP are spike-based Hebbian rules which at a network level could be utilized, for example, in Hopfield networks, Restricted Boltzmann Machines and Deep-Belief networks where they minimize the energy function of such networks. Moreover, they are used as unsupervised learning paradigm to cluster the data in competitive learning structures in which the distance between the moving average of the cluster and each data point is minimized [55]. However, to learn a specific target function in the hardware, it is much more desirable to define a more specific cost function and employ gradient descent for its optimization. Deriving an update rule based on this optimization algorithm results in Delta rule for a one-layer network and extends to "back propagation" using chain rule for deeper networks. Back propagation update rule however does not pass the locality criteria required for hardware implementation. However, recently it has been shown that such update rule can be implemented in a local fashion to approximate back propagation which makes such an algorithm even more powerful [56,57].

Given the 7 bits of precision reported in Ref. [58], this learning rule can take advantage of the "almost" analog nature of the memory by translating the error of the network to the number of pulses applied to the device. In Ref. [59] a circuit is proposed in which the network error modulates the frequency of a ring oscillator, thus allowing the device to be tuned more precisely to a desire value.

18.4 Spike-based implementation of the neuronal intrinsic plasticity

Neurons have multiple parameters that elicits certain computational features. Such properties include, but are not limited to its time constant, refractory period, and adaptation time constants. These parameters are determined by the conductance of the nerurons' membrane proteins. Biological neurons continuously adapt these conductances to maximize information transfer and minimize power consumption [60]. This local adaptive mechanism is described as neuronal intrinsic plasticity, which is shown to act as a neuronal homeostatic plasticity regulating the network's activity [61]. These neuron parameters are in the range of biological time constants in the order of milliseconds which are emulated in hardware using subthreshold circuits [5] biased using a central bias generator [62].

Currently the state-of-the-art neural processors compromise variety and share these parameters for all the neurons and synapses on a single core (thousands of neurons and synapses). If parameters were not shared, the static power consumption and the area consumed by wires (metal lines) running across the chip for connecting the biases grows linearly with the number of parameters.

As memristive devices are adaptable conductances, it is a natural implementation to use them as the replacement for the biases and use local plasticity rules to adapt them based on the neuron's activity. However given the millisecond range of time constant required, the memristive devices should always be in their high resistive state (HRS). Measurements in Ref. [52] show that the mean HRS value of memristive devices in a 4 kb array is an exponential function of the reset voltage applied across the device that follows a lognormal distribution. The circuit introduced in Section 18.3 could be used to apply the appropriate reset voltage acorss the device until the neuron's firing rate falls into the target range of activity. In such hybrid systems, each circuit model has its own parameters set by the incorporated RRAM without area or static power overhead which also enables the self-organiszation of individual parameters locally.

18.5 Scalable mixed memristive–CMOS multicore neuromorphic computing systems

To create complete neuromorphic computing systems that can support large-scale networks, including deep neural network and convolutional neural network architectures, it is necessary to develop an infrastructure to integrate in a single VLSI die, multiple spiking neural network cores and to connect the neurons and synapses in each core to all other neurons and synapses in all other cores. Examples of pure CMOS neuromorphic computing systems that comprise auxiliary circuits for interconnecting multiple cores and creating large-scale neural processing systems have already been developed in the past [1–4,7]. These systems make use of the AER communication protocol to route spikes from source neurons to destination synapses, and implement different types of routing schemes. One of the most critical factors for building routing schemes that can support very large-scale networks is the memory required to store the network connectivity parameters (routing tables and weights). While some solutions have resorted to using external DRAM memory chips [1], others chose to use on-chip SRAM circuits [2,3,7]. Both solutions have advantages and disadvantages: external DRAM chips can store very large amounts of memory, but the energy required to transfer the data from the memory chip to the neuromorphic processor is substantially larger than that used by on-chip SRAM circuits; on the other hand SRAM cells require at least 6 transistors per bit, and therefore occupy a substantial amount of area on the silicon die. The use of memristive devices can have a significant impact on the design of future neuromorphic processor chips also if used as classical digital memory circuits to store the routing data [63]. They promise to significantly increase the density and at the same time reduce the power consumption, as the writing of the data would be performed only at the onset of the experiment, during the definition of the network connectivity tables. Furthermore, when stored in DRAM or SRAM

cells, the network parameters would be deleted when the chips are reset and when power is removed. This would require users to upload all the parameters again when the system is powered up. This can be particularly problematic in large networks, where the storing and initialization of all the system parameters can take a significant amount of time. By virtue of their nonvolatility [64], these memristive devices will also save the configuration/start-up time, when booting up the system.

18.6 Conclusions and discussion

In this chapter, we have given an overview of some techniques to exploit memristive devices for implementing neuromorphic computing and learning systems. In such systems, most of chip area is devoted to the implementation of synapses. Therefore, it is in these components where the use of memristors can provide a significant boost for system density. We have discussed two main approaches: one, in which a synapse is built using one single memristor, giving the highest possible density, but at the cost of introducing significant overheads for the peripheral circuits and introducing timing and power constraints (unless second-order memristors are used, which is not discussed in this chapter); and a second one, in which synapses are made more complex by combining memristors and CMOS transistors, resulting in less-complex and more efficient peripheral circuits. Several STDP learning mechanisms exploiting memristors are discussed, ranging from plain STDP, STRDP, to stochastic STDP. Finally some quick scalability considerations are discussed.

Overall it is clear that memristive devices have a great potential for providing new ways of performing neuromorphic computing, including learning, with highly compact as well as low-power physical realizations. On the other hand, it is also true that today such realizations remain at the laboratory level, at small scale, and practical issues for mass production remain to be solved. For example, at the time of this writing, we are only aware of one hybrid CMOS-memristor technology being available for circuit design researchers as multiproject wafers through Europractice (http://www.europractice-ic.com/) or CMP (http://mycmp.fr/) that offers monolithic 1T-1R (one memristor with one NMOS selector transistor) devices that can store 1-bit. These memristors require one selecting transistor in series, thus increasing significantly its size, while occupying Si area, as opposed to pure memristors that could be added on top of the CMOS transistor layer without consuming transistor space. These selectors are necessary to address the problem of sneak-paths when selecting single devices in crossbars and can be used to limit the current during forming and writing operations. However, researchers are investigating non-transistor-based selectors that could share the same footprint as the memristors [65]. This together with the successful use of second-order memristors for learning could potentially yield to

compact and low-power self-learning neuromorphic systems ideally suited for "edge computing" and Internet-of-Things (IoT) Artificial Intelligence (AI) applications that require ultra low-power operation and do not need to be necessarily connected to the internet for "cloud computing" via the use of power-hungry server farms.

References

[1] S. Furber, F. Galluppi, S. Temple, L. Plana, The SpiNNaker project, Proc. IEEE 102 (5) (2014) 652–665.

[2] M. Davies, N. Srinivasa, T.H. Lin, et al., Loihi: aneuromorphic manycore processor with on-chip learning, IEEE Micro. 38 (1) (2018) 82–99.

[3] P.A. Merolla, J.V. Arthur, R. Alvarez-Icaza, et al., A million spikingneuron integrated circuit with a scalable communication network and interface, Science 345 (6197) (2014) 668–673. issn: 0036-8075, 1095-9203.

[4] J. Schemmel, D. Bruderle, A. Grubl, et al., A wafer-scale neuromorphic hardware system for large-scale neural modeling, Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on, IEEE, 2010, pp. 1947–1950.

[5] E. Chicca, F. Stefanini, C. Bartolozzi, G. Indiveri, Neuromorphic electronic circuits for building autonomous cognitive systems, Proc. IEEE 102 (9) (2014) 1367–1388. issn: 0018-9219.

[6] N. Qiao, H. Mostafa, F. Corradi, et al., A reconfigurable on-line learning spiking neuromorphic processor comprising 256 neurons and 128k synapses, Front. Neurosci. 9 (141) (2015) 1–17.

[7] S. Moradi, N. Qiao, F. Stefanini, G. Indiveri, A scalable multicore architecture with heterogeneous memory structures for dynamic neuromorphic asynchronous processors (DYNAPs), IEEE Trans. Biomed. Circuits Systems 12 (1) (2018) 106–122.

[8] G. Indiveri, B. Linares-Barranco, R. Legenstein, G. Deligeorgis, T. Prodromakis, Integration of nanoscale memristor synapses in neuromorphic computing architectures, Nanotechnology 24 (38) (2013). Available from: https://doi.org/10.1088/0957-4484/24/38/384010. p. 384 010, [Online]. Available: http://stacks.iop.org/0957-4484/24/i = 38/a = 384010.

[9] M. Payvand, M. Nair, L. Müller, G. Indiveri, A neuromorphic systems approach to in-memory computing with non-ideal memristive devices: from mitigation to exploitation, Faraday Discuss. (2018) 1–13. Available from: https://doi.org/10.1039/C8FD00114F [Online]. Available: https://doi.org/10.1039/C8FD00114F.

[10] S.-C. Liu, T. Delbruck, Neuromorphic sensory systems, Curr. Opin. Neurobiol. 20 (3) (2010) 288–295. Available from: https://doi.org/10.1016/j.conb.2010.03.007.

[11] E. Neftci, J. Binas, U. Rutishauser, et al., Synthesizing cognition in neuromorphic electronic systems, Proc. Natl Acad. Sci. U S A 110 (37) (2013) E3468–E3476.

[12] J. Backus, Can programming be liberated from the von neumann style?: a functional style and its algebra of programs, Commun. ACM 21 (8) (1978) 613–641. Available from: https://doi.org/10.1145/359576.359579 [Online]. Available. Available from: http://doi.acm.org/10.1145/359576.359579.

[13] G. Indiveri, S.-C. Liu, Memory and information processing in neuromorphic systems, Proc. IEEE 103 (8) (2015) 1379–1397. Available from: https://doi.org/10.1109/JPROC.2015.2444094.

[14] S. Habenschuss, Z. Jonke, W. Maass, Stochastic computations in cortical microcircuit models, PLoS Comput. Biol. 9 (11) (2013) e1003311.

[15] W. Maass, Noise as a resource for computation and learning in networks of spiking neurons, Proc. IEEE 102 (5) (2014) 860–880. Available from: https://doi.org/10.1109/JPROC.2014.2310593. issn: 0018-9219.

[16] S. Deiss, R. Douglas, A. Whatley, A pulse-coded communications infrastructure for neuromorphic systems, in: W. Maass, C. Bishop (Eds.), Pulsed Neural Networks, MIT Press, 1998, pp. 157–178. ch. 6.

[17] K. Boahen, Point-to-point connectivity between neuromorphic chips using address-events, IEEE Trans. Circuits Syst. II 47 (5) (2000) 416–434.

[18] P. Merolla, J. Arthur, B. Shi, K. Boahen, Expandable networks for neuromorphic chips, IEEE Trans. Circuits Syst. I 54 (2) (2007) 301–311.

[19] E. Chicca, A. Whatley, P. Lichtsteiner, et al., A multi-chip pulse-based neuromorphic infrastructure and its application to a model of orientation selectivity, IEEE Trans. Circuits Syst. I 5 (54) (2007) 981–993. Available from: https://doi.org/10.1109/TCSI.2007.893509.

[20] P. Dayan, L. Abbott, Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems, MIT Press, Cambridge, MA, USA, 2001, ISBN: 9780262541855, .

[21] M. Prezioso, F. Merrikh-Bayat, B. Hoskins, et al., Training and operation of an integrated neuromorphic network based on metal-oxide memristors, Nature 521 (7550) (2015) 61–64. Available from: https://doi.org/10.1038/nature14441.

[22] S.H. Jo, T. Chang, I. Ebong, et al., Nanoscale memristor device as synapse in neuromorphic systems, Nano Lett. 10 (4) (2010) 1297–1301.

[23] M. Zidan, H. Omran, R. Naous, et al., Single-readout high-density memristor crossbar, Sci. Rep. 6 (2016) 18 863.

[24] I. Boybat, M.L. Gallo, T. Moraitis, et al., Neuromorphic computing with multi-memristive synapses, Nat. Commun. 9 (2018) 2514.

[25] M.V. Nair and G. Indiveri, *A differential memristive current-mode circuit*, European patent application EP 17183461.7, Filed 27.07.2017, 2017.

[26] S. Nease, E. Chicca, Floating-gate-based intrinsic plasticity with lowvoltage rate control, International Symposium on Circuits and Systems ISCAS 2016, IEEE, 2016, pp. 2507–2510.

[27] F.L.M. Huayaney, S. Nease, E. Chicca, Learning in silicon beyond STDP: Aneuromorphic implementation of multi-factor synaptic plasticity with calcium-based dynamics, IEEE Trans. Circuits Syst. I: Regul. Pap. 63 (12) (2016) 2189–2199.

[28] S. Schmitt, J. Klähn, G. Bellec, et al., "Neuromorphic hardware in the loop: training a deep spiking network on the BrainScaleS wafer-scale system," in *2017 International Joint Conference on Neural Networks (.CNN)*, 2017, pp. 2227–2234.

[29] G.-Q. Bi, M.-M. Poo, Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type, J. Neurosci. 18 (24) (1998) 10 464–10 472.

[30] W. Gerstner, R. Kempter, J.L. van Hemmen, H. Wagner, A neuronal learning rule for sub-millisecond temporal coding, Nature 383 (6595) (1996) 76.

[31] T. Masquelier, R. Guyonneau, S. Thorpe, Spike timing dependent plasticity finds the start of repeating patterns in continuous spike trains, PLoS ONE 3 (1) (2008) e1377. Available from: https://doi.org/10.1371/journal.pone.0001377.

[32] H. Markram, J. Lübke, M. Frotscher, B. Sakmann, Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs, Science 275 (1997) 213–215.

[33] H. Markram, W. Gerstner, P. Sjöström, Spike-timing-dependent plasticity: a comprehensive overview, Front. Synaptic Neurosci. 4 (2) (2012) 1−3. Available from: https://doi.org/10.3389/fnsyn.2012.00002.

[34] P. Werbos, The Roots of Backpropagation: From Ordered Derivatives to Neural Networks and Political Forecasting, vol. 1, Wiley, 1994.

[35] D.E. Rumelhart, G.E. Hintont, R.J. Williams, Learning representations by back-propagating errors, Nature 323 (6088) (1986) 533−536.

[36] S. Saighi, C. Mayr, B. Linares-Barranco, et al., Plasticity in memristive devices, Front. Neurosci. 9 (51) (2015).

[37] G. Indiveri, E. Chicca, R. Douglas, A VLSI array of low-power spiking neurons and bistable synapses with spike−timing dependent plasticity, IEEE Trans. Neural Networks 17 (1) (2006) 211−221.

[38] M.R. Azghadi, N. Iannella, S. Al-Sarawi, G. Indiveri, D. Abbott, Spike-based synaptic plasticity in silicon: design, implementation, application, and challenges, Proc. IEEE 102 (5) (2014) 717−737. Available from: https://doi.org/10.1109/JPROC.2014.2314454. issn: 0018-9219.

[39] T. Serrano-Gotarredona, T. Masquelier, T. Prodromakis, G. Indiveri, B. Linares-Barranco, STDP and STDP variations with memristors for spiking neuromorphic learning systems, Front. Neurosci. 7 (2) (2013). Available from: https://doi.org/10.3389/fnins.2013.00002. issn: 1662-453X, [Online]. Available: http://www.frontiersin.org/neuroscience/10.3389/fnins.2013.00002/full.

[40] S. Kim, C. Du, P. Sheridan, et al., Experimental demonstration of a second-order memristor and its ability to biorealistically implement synaptic plasticity, Nano Lett. 15 (3) (2015) 2203−2211.

[41] J.M. Brader, W. Senn, S. Fusi, Learning real-world stimuli in a neural network with spike-driven synaptic dynamics, Neural Computation 19 (11) (2007) 2881−2912.

[42] S. Fusi, M. Annunziato, D. Badoni, A. Salamon, D. Amit, Spike−driven synaptic plasticity: theory, simulation, VLSI implementation, Neural Computation 12 (2000) 2227−2258.

[43] H. Mostafa, A. Khiat, A. Serb, et al., Implementation of a spike-based perceptron learning rule using TiO2-x memristors, Front. Neurosci. 9 (357) (2015). Available from: https://doi.org/10.3389/fnins.2015.00357.

[44] P. Sjöström, G. Turrigiano, S. Nelson, Rate, timing, and cooperativity jointly determine cortical synaptic plasticity, Neuron 32 (6) (2001) 1149−1164 [Online]. Available: https://doi.org/10.1016/S0896-6273(01)00542-6.

[45] S. Gaba, P. Sheridan, J. Zhou, S. Choi, W. Lu, Stochastic memristive devices for computing and neuromorphic applications, Nanoscale 5 (13) (2013) 5872−5878.

[46] S. Ambrogio, S. Balatti, V. Milo, et al., Neuromorphic learning and recognition with one-transistor-one-resistor synapses and bistable metal oxide RRAM, IEEE Trans. Electron. Devices 63 (4) (2016) 1508−1515.

[47] J.J. Yang, D.B. Strukov, D.R. Stewart, Memristive devices for computing, Nat. Nanotechnol. 8 (1) (2013) 13−24.

[48] J. Woo, K. Moon, J. Song, et al., Improved synaptic behavior under identical pulses using AlOx/HfO2 bilayer RRAM array for neuromorphic systems, IEEE Electron. Device Lett. 37 (8) (2016) 994−997.

[49] M. Payvand, L.K. Muller, G. Indiveri, Event-based circuits for controlling stochastic learning with memristive devices in neuromorphic architectures, Circuits and Systems (ISCAS), 2018 IEEE International Symposium on, IEEE, 2018, pp. 1−5.

[50] B. Widrow, M. Hoff, Adaptive switching circuits, 1960 IRE WESCON Convention Record, Part 4, IRE, New York, 1960, pp. 96–104 [Online]. Available: http://isl-www.stanford.edu ~ widrow/papers/c1960adaptiveswitching.pdf.

[51] J. Hertz, A. Krogh, R. Palmer, Introduction to the Theory of Neural Computation, Addison-Wesley, Reading, MA, 1991.

[52] T. Dalgaty, M. Payvand, B. De Salvo, et al., Hybrid cmos-rram neurons with intrinsic plasticit, International Symposium on Circuits and Systems (ISCAS), 2019, IEEE, 2019.

[53] The MNIST database of handwritten digits, Yann LeCun's web-site, 2012. [Online]. Availablefrom http://yann.lecun.com/exdb/mnist/.

[54] A. Yousefzadeh, E. Stromatias, M. Soto, T. Serrano-Gotarredona, B. Linares-Barranco, On practical issues for stochastic stdp hardware with 1-bit synaptic weights, Front. Neurosci. 12 (2018).

[55] R. Kreiser, T. Moraitis, Y. Sandamirskaya, G. Indiveri, On-chip unsupervised learning in winner-take-all networks of spiking neurons, Biomedical Circuits and Systems Conference, (BioCAS), 2017, IEEE, 2017, pp. 424–427.

[56] J. Sacramento, R.P. Costa, Y. Bengio, and W. Senn, "Dendritic error backpropagation in deep cortical microcircuits," arXiv preprint arXiv:1801.*00062*, 2017.

[57] E.O. Neftci, C. Augustine, S. Paul, G. Detorakis, Event-driven random back-propagation: enabling neuromorphic deep learning machines, Front. Neurosci. 11 (2017) 324. Available from: https://doi.org/10.3389/fnins.2017.00324. issn: 1662-453X, [Online]. Available: https://www.frontiersin.org/article/10.3389/fnins.2017.00324.

[58] S. Stathopoulos, A. Khiat, M. Trapatseli, et al., Multibit memory operation of metal-oxide bi-layer memristors, Sci. Rep. 7 (1) (2017) 17 532.

[59] M. Payvand, G. Indiveri, Spike-based plasticity circuits for always-on on-line learning in neuromorphic systems, International Symposium on Circuits and Systems (ISCAS), 2019, IEEE, 2019.

[60] M. Stemmler, C. Koch, How voltage-dependent conductances can adapt to maximize the information encoded by neuronal firing rate, Nat. Neurosci. 2 (1999) 521–527.

[61] J. Triesch, Synergies between intrinsic and synaptic plasticity mechanisms, Neural Computation 19 (2007) 885–909.

[62] T. Delbruck, R. Berner, P. Lichtsteiner, C. Dualibe, 32-bit configurable bias current generator with sub-off-current capability, *International Symposium on Circuits and Systems, (ISCAS), 2010*, IEEE, IEEE, Paris, France, 2010, pp. 1647–1650. Available from: http://doi.org/10.1109/ISCAS.2010.5537475.

[63] S. Moradi, R. Manohar, The impact of on-chip communication on memory technologies for neuromorphic systems, J. Phy. D: Appli. Phy. 52 (1) (2018) 014003.

[64] D. Ielmini, R. Waser, Resistive Switching: From Fundamentals of Nanoionic Redox Processes to Memristive Device Applications, John Wiley & Sons, 2015.

[65] Z. Wang, S. Joshi, S.E. Savel'ev, et al., Memristors with diffusive dynamics as synaptic emulators for neuromorphic computing, Nat. Mater. 16 (1) (2017) 101.

Chapter 19

Spiking neural networks for inference and learning: a memristor-based design perspective

Mohammed E. Fouda[1], Fadi Kurdahi[1], Ahmed Eltawil[1] and Emre Neftci[2]

[1]*Department of Electrical Engineering and Computer Science, University of California—Irvine, Irvine, CA, United States,* [2]*Department of Cognitive Sciences and Department of Computer Science, University of California—Irvine, Irvine, CA, United States*

19.1 Introduction

Machine learning and particularly deep learning have become the de facto choice in solving a wide range of problems when adequate data are available. So far machine learning has been mainly concerned more by the "learning" rather than the "machine." This focus is natural given that von Neumann computers and graphics processing units capable of general purpose processing offer excellent performance per unit of monetary cost. As the scalability of such processors hits difficult scalability and energy efficiency challenges, interest in dedicated, multicore, and multiprocessor systems is increasing. This calls for increased efforts on improving the physical instantiations of "machines" for machine learning. Physical instantiation of computations is challenging because the structure and nature of the physical substrate severely restrict the basic computational operations it can carry out. However, if the computations can be formulated in a way that they exploit the physics of the devices, then the efficiency and scalability can be drastically improved. In this line of thought, the field of neuromorphic engineering is arguably the one that has attracted the most attention and effort. The field's core ideas communicated by R. Feynmann, C. Mead, and other researchers in a series of lectures called physics of computation elaborate on the analogies between the physics of ionic channels in the brain and those of complementary metal−oxide−semiconductor (CMOS) transistors [1]. By building synapses,

Memristive Devices for Brain-Inspired Computing. DOI: https://doi.org/10.1016/B978-0-08-102782-0.00019-8

neurons, and circuits modeled after the brain and driven by similar physical laws, neuromorphic engineers would "understand by building" and help engineer novel computing technologies equipped with the robustness and efficiency of the brain. In the last decade there have been enormous advances in building and scaling neuromorphic hardware using mixed-signal [2–5] and digital [6–8] technologies, including embedded learning capabilities and scales achieving 1 M neurons per chip. A major limitation in these technologies is the memory required to store the state and parameters of the system. For example in both mixed-signal and digital technologies, synaptic weights are typically stored in static random-access memory (SRAM), the densest, fastest, and most energy-efficient memory we can currently locate next to the computing substrate [6,8,9]. Unfortunately SRAMs are expensive in terms of area and energy, making on-chip memory small given the computational requirements. In fact learning often requires higher precision parameters to average out noise and ambiguities in real world data, especially in the case of gradient-based learning in neural networks [10].

In this chapter, we explore promising learning and inference algorithms compatible with neuromorphic hardware, with a special focus on spiking neural networks (SNNs). We highlight their hardware requirements, discuss their possible implementations with memristors, and identify a class of computations they are particularly suitable for. In doing so, we will view SNNs as types of recurrent neural networks (RNN) and explain which hardware nonidealities are detrimental, and which can be exploited for improving computations.

19.2 Spiking neural networks and synaptic plasticity

We start with a description of neuron models commonly used in neuromorphic hardware and describe the tools used to analyze and develop algorithms on them. The most common model is the Leaky Integrate and Fire (LI&F) [11]. While several variations of LI&F neuron models of it exist, including mixed-signal current mode and digital implementations, the base model can be formally described in differential form as:

$$\tau_{\text{mem}} \frac{\mathrm{d}U_i}{\mathrm{d}t} = -(U_i - U_{\text{rest}}) + RI_i - S_i(t)(U_i - U_{\text{rest}}), \tag{19.1}$$

$$\tau_{\text{syn}} \frac{\mathrm{d}I_i}{\mathrm{d}t} = -I_i(t) + \tau_{\text{syn}} \sum_j W_{ij} S_j(t), \tag{19.2}$$

where $U_i(t)$ is the membrane potential of neuron i, U_{rest} is the resting potential, τ_{mem} and τ_{syn} are the membrane time constants, R is the input resistance, and $I_i(t)$ is the synaptic current [12]. Eq. (19.1) shows that U_i acts as a leaky integrator of the input current I_i, which itself is a leaky, weighted integration of the spike train S_j. Spike trains are described as a sum of delta Dirac

functions $S_j(t) = \sum_k \delta(t - t_j^k)\ t_j^k$ where t_j^k are spike times of neuron j. In other words, for each incoming spike, the synaptic current undergoes a jump of height W_{ij} and otherwise decays exponentially with a time constant τ_{syn}. Note that the expression $\sum_j W_{ij} S_j(t)$ has the structure of a vector−matrix multiplication and forms the basis of RRAM implementations of SNN discussed later in this chapter. Because S_j represents spikes ($S_j = 0$ or 1), these operations consist of additions and scaling of the result by τ_{syn}.

Neurons emit spikes when the membrane voltage reaches the firing threshold, V_{th}. After each spike, the membrane voltage U_i is reset to the resting potential U_{rest} (Fig. 19.1A). In Eq. (19.1), the reset is carried out by the term $S_i(t)(U_i - U_{\text{rest}})$ when the membrane potential reaches V_{th}. This reset acts as a refractory mechanism since the neuron must reach the firing threshold from U_{rest} to fire again.

A simple circuit implementation of the LI&F is based on resistor−capacitor circuits with a switch to reset the voltage across the capacitor as shown in Fig. 19.1B. The mathematical model of this circuit is equivalent to Eq. (19.1) where $\tau_{\text{mem}} = RC$. Several variations of this circuit have been demonstrated [11]. Synaptic circuits that reproduce the dynamics of I can be implemented in analog VLSI using a Differential-Pair Integrator (DPI) or other similar circuits [13].

To make the mathematical analysis and digital implementations more intuitive, it is useful to write LI&F in the discrete form:

$$U_i[n+1] = \beta U_i[n] + U_{rest} + I_i[n] - S_i[n](U_i[n] - U_{rest})$$

$$I_i[n+1] = \alpha I_i[n] + \sum_j W_{ij} S_j[n]$$

$$S_i[n] = \Theta(U_i[n]) = \begin{cases} 1 & U_i[n] \geq V_{th} \\ 0 & \text{Otherwise} \end{cases}$$

where $\alpha = \exp\left(-\frac{\Delta t}{\tau_{\text{mem}}}\right)$ and $\beta = \exp\left(-\frac{\Delta t}{\tau_{\text{syn}}}\right)$ are constants that capture the decay dynamics of states U and I, respectively, during a Δt timestep, Θ is the step function, and V_{th} is the firing threshold.

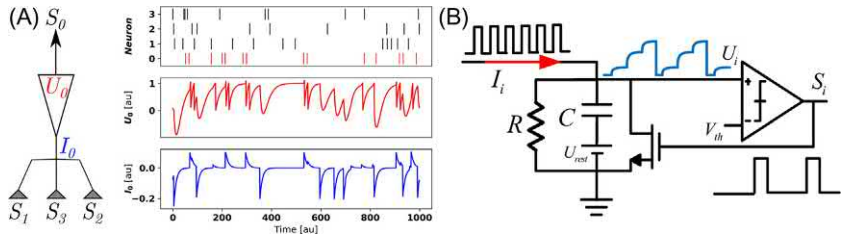


FIGURE 19.1 (A) Example Leaky Integrate and Fire (LI&F) Neuron dynamics for $U_{rest} = 0$ and $V_{th} = 1$ and (B) basic neuron circuit implementing LI&F dynamics.

Several digital implementations of SNNs use these update equations [6,14]. These equations can be described using an artificial RNN [15] formalism. The recurrent aspect arises from the statefulness of the neurons (when $\alpha > 0$ and $\beta > 0$). In addition, if there exist connections from the neuron to itself, then these connections can be viewed on the same footing as recurrent connections in artificial RNN. As we shall later see, the artificial RNN view of SNNs enables the derivation of biologically credible learning rules with the capabilities of machine learning algorithms.

This description of the SNN generalizes to a non-recurrent artificial neural network where activations are binary. In fact, replacing α and β with 0 and ignoring the reset, the equations above become:

$$U_i[n+1] \quad = \sum_j W_{ij} S_j[n], S_i[n] = \Theta(U_i[n]). \tag{19.3}$$

The dynamics above are those of the standard artificial neural network (without any multiplications, as described above) followed by a spiking nonlinearity, that is, they are Perceptrons. Neural networks with binary neural activations and/or weights were proposed as efficient implementations of conventional neural networks [16,17]. Such devices are promising for energy-efficient implementations of deep learning processors in full-digital technologies [18,19] as well as with in-memory processing with emerging memory devices [20].

19.3 Memristive realization and nonidealities

Neuromorphic hardware implementations require a device or a circuit that mimics the synapse behavior. A minimum requirement for neural network applications is a memory to store the synaptic weight. Memristive devices can be used to realize such synapses. A device is referred as a memristive device if it exhibits pinched hysteresis behavior in the current−voltage plane that indicates a memory behavior in its resistance. Many physical devices exhibit memristive behaviors such as phase change memory (PCM), ferroelectric RAM (FeRAM), spin-transfer torque magnetic RAM (STT-MRAM), and resistive RAM (RRAM or ReRAM). RRAMs have a promising potential for neuromorphic applications due to high area density, stackability, and low write energy compared with the other emerging devices [21]. In this section, we focus on RRAM, without loss of generality, to discuss the physical limitations and problems facing deploying such technology for neuromorphic hardware.

A common building block in both spiking and non-spiking NN is the weighted summation of inputs (see Eq. (19.3)). This can be performed in a single step using a crossbar structure, unlike the conventional computing methods that typically require $N \times M$ steps or clock cycles. Fig. 19.2 shows a single-layer crossbar-based resistive neural network with M inputs and N

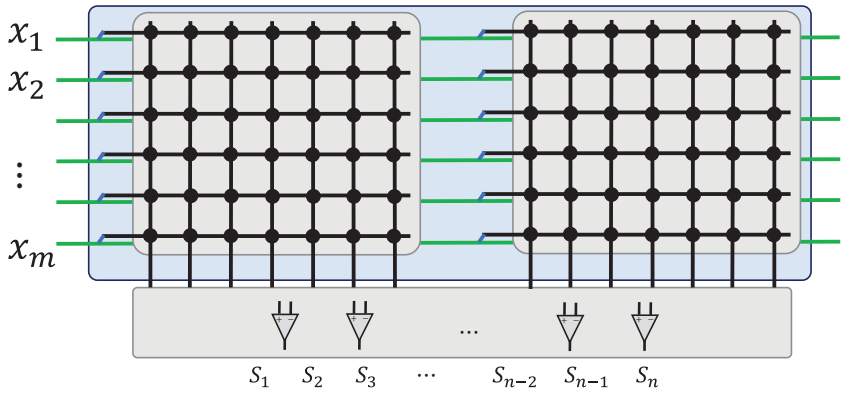


FIGURE 19.2 Crossbar array realization of one layer neural network.

outputs representing N perceptrons with M inputs each and the weights are stored in the memristors. The inputs to the perceptrons (presynaptic signals) are encoded in the input voltages and the output of each perceptron is the sum of the currents passing through each memristor. This way all currents within the same column can be linearly summed to obtain the postsynaptic currents, that is equivalent of Eq. (19.2). The total postsynaptic currents need sensing and shaping circuits to convert them into voltages and to be passed to subsequent neurons. In the non-spiking NN, the postsynaptic current is summed through the sensing circuit and passed through another shaping circuit to create the required neural activity such as sigmoid, tanh, or rectified linear functions. With spiking neurons, the output of the current sensing circuit is instead passed through a LI&F circuit.

In neural networks, both positive (excitatory) and negative (inhibitory) connections are required. However, the RRAM conductance is positive by definition, which only supports excitatory or inhibitory connections. Two weight realization techniques are possible to create both excitatory and inhibitory connections;

1. using two RRAMs per weight [22,23] which is referred to as balanced realization or
2. using one RRAM as weight in addition to one reference RRAM having a conductance set to $G_r = (G_{max} + G_{min})/2$ which is referred to as unbalanced realization [24,25].

The first realization has double the dynamic range $w \in [-\Delta G, \Delta G]$, where $\Delta G = G_{max} - G_{min}$, making it more immune to variability at a cost of double the area, double power consumption during inference and requires additional programming operations. The second technique has one RRAM

device, meaning that $w \in [-\Delta G/2, \Delta G/2]$ making it more prone to variability but the overall area is smaller, requires less power, and is easier to program (programming only one RRAM per weight). Due to high variations in the existing devices, the first approach is commonly used with one big crossbar or two crossbars (one for positive weights and the other one for negative weights as shown in 2). The output of the memristive neural network can be written as

$$S_i = \sum_{j=1}^{m} G_{ij} V_j \tag{19.4}$$

where S_i is the output of the i^{th} neuron and $G_{ij} = G_{ij}^{+} - G_{ij}^{-}$ is the synaptic weight, and V_j is the j^{th} input. The crossbar array forms the majority of the research in using RRAMs for neuromorphic computation [26]. In practice, RRAMs and crossbar structures suffer from many problems and do not behave ideally for computational purposes. These nonidealities can severely undermine the overall performance of applications unless they are taken into consideration during the training operation. After defining the mapping of synaptic weights to RRAM conductances, the following sections will overview these nonidealities in light of neuromorphic computation and learning.

19.3.1 Weight Mapping

As discussed above, each weight is translated into two conductances, that is one-to-two mapping and can be mathematically formulated as

$$G = G^{+} - G^{-} = \frac{W}{W_{max}} \Delta G, \tag{19.5}$$

where W_{max} is the maximum value of the weight. If it is required to realize W_{max}, G^{+} and G^{-} are set to G_{max} and G_{min}, respectively. The difference between the two conductances is constant and proportional to the required weight value and each conductance is constrained to be between G_{min} and G_{max} as shown in Fig. 19.3. Thus, there are many possible realizations for

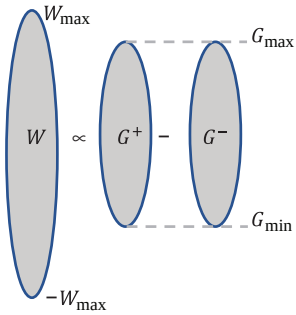


FIGURE 19.3 Mapping synaptic weight into conductances.

each weight, for example the zero weight can be realized with any equal values of G^+ and G^-. Therefore a criterion for selecting the weight mapping should be defined. This can be formulated as an optimization problem as follows:

$$\begin{aligned} &\text{minimize} \quad \mathcal{L}(G^+, G^-) \\ &\text{subject to} \\ &\quad G_{min} \leq G^+, \quad G^- \leq G_{max}, \text{and} \\ &\quad G^+ - G^- = \frac{W}{W_{max}} \Delta G \end{aligned} \tag{19.6}$$

where $\mathcal{L}$ is the objective function, G^+ and G^- are the positive and negative conductance matrices, respectively. Since many conductance configurations are possible to obtain the same effective weight, additional criteria such as power consumption while reading (important for inference) or writing (important in online training) can be introduced. These constraints can be taken into consideration while training the network with a regularization term in the loss function (see Section 4). We note that the mapping is more important for offline training where the optimization is completed on software and the final weight values are transferred to the hardware.

19.3.2 RRAM endurance and retention

An attractive purpose of RRAMs is to accelerate inference and training. However, endurance is a critical obstacle to RRAM deployment in neuromorphic hardware. In online learning, the devices are frequently updated, and especially so during gradient-based learning as in artificial neural networks. However, each device has a limited number of potentiation and depression cycles [27,28]. Endurance depends on the device's switching and electrode materials. For example HfO_x devices can achieve endurance up to 10^{10} cycles, but Al_2O_3 devices achieve endurance only up to 10^4 [29]. With limited endurance, it is necessary to complete the training before the devices degrade. The endurance requirement for learning is application dependent. In standard deep learning, weight updates are usually performed every batch. Classification benchmarks such as MNIST handwritten digit recognition require writing around 10^4 cycles. However, even gradient-based training of neural networks can easily scale to 10^8 cycles.

Neuromorphic hardware can be multipurpose (i.e., the same device can be used to perform many different tasks), where a complete training of the network is performed for every task. Consequently the device endurance should be high enough to cover its lifetime use. There are some solutions to mitigate the endurance problem in machine learning scenarios:

- Full-offline training: Training is completed on software and the final weights are transferred to the RRAM-based hardware. This requires an

accurate model of the devices, the crossbar array, and the sensing circuitry in the training procedure, and to verify the response of each part of the network to make sure that the response matches the simulated one [30].

- Semi-online training: A complete training cycle is performed offline, then the new weights are transferred to the devices. Then online retraining is carried out to reduce performance loss due to the existing impairments. Due to the smaller number of writing cycles, this solution would relax the endurance requirements. In Ref. [25], it was noticed that the network was able to recover the original accuracy after $10\% - 20\%$ of the training epochs.

Once the online or the offline training is performed, the network can operate in the inference mode where only reading cycles are performed. In this case, the retention of the stored values becomes an important issue. As with endurance, RRAM retention is also dependent on the device materials and temperature. For example, the HfO_x devices have around 10^4 seconds (2.78 hours) retention [31]. Although this might be sufficient for certain single-use scenarios, such as biomedical applications, it is inadequate for IoT and autonomous control applications. In this case, the retention values need to be more than 10^6 seconds across different temperature values (since retention degrades with increasing the temperature).

Full online learning requires high endurance and moderate retention, but semi-online requires moderate endurance and retention. Thus, while both endurance and retention are important for machine inference and learning tasks, the learning approach may require one of them to be far superior than the other.

19.3.3 Sneak Path Effect

The sneak path problem, also is referred to as IR drop, arises from the existence of the wire resistance that is inevitable in nanostructure crossbar arrays. The wire resistance creates many paths to the signal from each input port to the output port. These multiple paths create undesired currents that perturb the reading of the weight. It is expected that the wire residence would reach around 92 Ω for 5 nm feature size [32], which is the expected feature size for crossbar technology according to International Technology Roadmap of Semiconductors (ITRS) [21]. Fig. 19.4A and B show an example of the sneak path problem in 512×512 with random weights. A linear switching device having a $10^6\Omega$ high-resistance state and $10^3\Omega$ low-resistance state is used while the wire resistance is 0.1 Ω. Ideally, the measured weights should be similar to the programmed weights, as shown in Fig. 19.4C. Despite the small value of the wire resistance, it has a significant effect on the weights stored in the crossbar arrays (Fig. 19.4A). The weights are exponentially decaying across the diagonal of the array where the cell (1,1) has the least sneak path effect and the cell (N,M) has the worst sneak path effect.

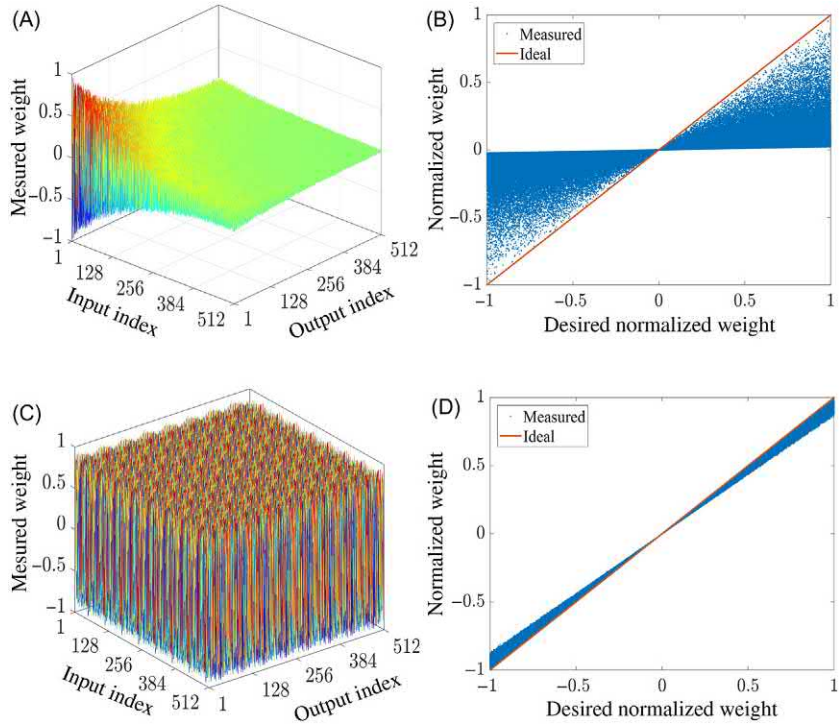


FIGURE 19.4 Effect of the wire resistance on the measured weights for 512×512 crossbar array with 0.1 Ω wire resistance. 3D plots of random weights distributed across the array (A) without partitioning and (C) with partitioning into 64×64 crossbar arrays; and the measured weights with the sneak path problem versus the desired values for (B) the entire array without partitioning and (D) with partitioning.

Some devices have a voltage-dependent conductance where the conductance is an exponential or quadratic function of the applied voltage [32]. This conductance nonlinearity can help reduce the sneak path problem in resistive memories on crossbar or crosspoint structures [33,34] due to single cell reading. But, in neuromorphic applications, this adds an exponential behavior to vector−matrix multiplication (VMM) which becomes

$$S_i = \sum_{j=1}^{m} G_{ij} \sinh(aV_j). \tag{19.7}$$

This exponential nonlinearity perturbs the VMM operation, which deteriorates the training performance [35]. Some algorithms were developed to take the effect of the device's voltage-dependency into consideration while training non-SNNs [35]. The same algorithm idea can be extended to SNNs.

Partitioning of Large Layer Matrices The sneak path problem prohibits the implementation of large matrices using a single large crossbar array. One possible solution is to partition the large matrices into small matrices that can be implemented using realizable crossbar arrays. Fig. 19.5 shows the partitioned crossbar arrays and the interconnect fabric between them to realize the complete VMMs where the large crossbar array, having $N \times M$ RRAMs, is partitioned into $n \times m$ crossbar arrays. In order to have the same structure of a large crossbar array, vertical and horizontal interconnects are placed under the crossbar arrays. This horizontal interconnect is used to connect the inputs between the crossbar arrays within the same array rows. The vertical interconnect is used to connect the outputs of the vertical crossbar arrays. The vertical interconnects are grounded through the sensing circuit to absorb the currents within the same vertical wire. The sensed currents are then connected to the neuronal activity. It is worth highlighting that each crossbar array may require input drivers (buffers) to reduce the loading effect of the vertical interconnect and crossbar arrays. These buffers are not shown in Fig. 19.5 for clarity. Moreover, they can be placed under the crossbar arrays to save the wafer area where the crossbar arrays are usually fabricated between higher metal layers. Fig. 19.4C shows the measured random synaptic weights with the same aforementioned parameters after partitioning the 512×512 crossbar arrays into 64 of 64×64 crossbar arrays. The weight variations due to their locations in the crossbar array became much smaller as shown in Fig. 19.4D and can be considered with the device variation.

Although partitioning the array mitigates the sneak path problem, it might cause routing problems where the nonidealities (e.g., parasitics) of the routing fabric will affect the performance. Thus, routing's nonidealities must be simulated in the case of full-offline learning. Also, additional algorithmic work is needed to overcome the residual sneak path problem after partitioning (especially with the aforementioned high-wire resistance expected to be

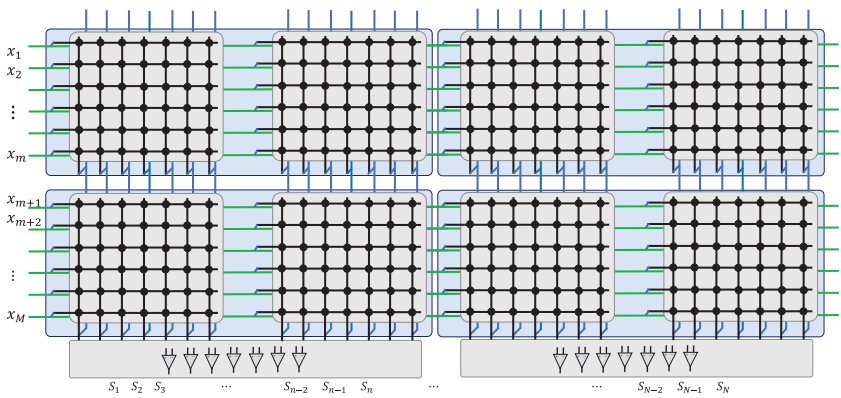


FIGURE 19.5 Realization of the partitioned matrices.

10 Ω per cell), such as the masking technique proposed in binary neural networks [25]. In the masking technique, the exponentially decaying profile is used to capture the effect of the sneak path problem during learning by multiplying the trained weights element wise.

19.3.4 Delay

Signal delay determines the speed at which computations can be run on hardware. While delays are not an issue for neuromorphic hardware designed to run with real-work time constants [2,9], other models are accelerated [36]. Due to the parallel VMM operation, the memristive hardware would be dedicated to an accelerated regime. For this, it is necessary to reduce delay caused by the device and structure parasitics and circuits.

In Ref. [32], a complete mathematical model for the crossbar delay is discussed. The model showed that the delay is a function of the weights stored in the crossbar arrays. The higher the device resistance, the more delay to the signal. For 1 MΩ switching resistance, the maximum delay of the crossbar arrays is expected to be in the range of nanoseconds. Also, there is another delay resulting from the sensing circuit that is expected to be around 10 ns.

The partitioning and drivers add extra delay factors caused by the wire resistance of the interconnect fabric and the input capacitance of the drivers. Delay can be calculated using the Elmore delay model [32]. The wire resistance of the interconnect per array is nR_w where n is the number of columns per array and R_w is the wire resistance per cell. The Elmore delay of such an interconnect wire is $0.67nR_wC_d$, where C_d is the input capacitance of the buffer. Thus, the total input delay is $0.67(N-n)R_wC_d + (N/n)\tau_d$, where N/n is the number of horizontal crossbar arrays and τ_d is the driver delay. The delay resulting from the partitioning and drivers is expected to be in the range of nanoseconds. Thus, the total delay of the entire layer would be in the range of 20−100 ns. It is worth mentioning that the effect of the capacitive parasitics of the crossbar array is often ignored because the feature size of the fabricated devices is in the range of sub micrometers, that is $F = 200$ nm [23]. However, for nano-scale structures, $F = 10$ nm, the capacitive parasitics may cause leakage paths at high frequency, where the impedance between the interconnects would be comparable to or less than the switching device's impedance, which would affect the performance. Thus, a more detailed analysis of the capacitive parasitics of the crossbar array must be considered on a case-to-case basis.

19.3.5 Asymmetric nonlinearity conductance update model

Several RRAM devices demonstrating promising synaptic behaviors are characterized by nonlinear and asymmetric update dynamics, which is a

major obstacle for large-scale deployment in neural networks [26], especially for learning tasks. Applying the vanilla backpropagation algorithms without taking into consideration the device nonidealities does not guarantee the convergence of the network. Thus, a closed-form model for the device nonlinearity must be derived based on the device dynamics and added to the neural network training algorithm to guarantee the convergence to the optimal point (minimal error).

Most of the potentiation and depression behaviors have exponential dynamics versus the programming time or the number of pulses. In practice, the depression curve has a higher slope compared with the potentiation curve, which causes asymmetric programming. The asymmetric nonlinearity of the RRAM's conductance update can be fitted to the following model

$$G(t) = \begin{cases} G_{max} - \beta_P e^{-\alpha_1 \phi(t)} & v(t) > 0 \\ G_{min} + \beta_D e^{-\alpha_1 \phi(t)} & v(t) < 0 \end{cases} \tag{19.8}$$

where G_{max} and G_{min} are the maximum and minimum conductances, respectively, $\alpha_1, \alpha_2, \beta_P$ and β_D are fitting coefficients. and β_D are related to the difference between G_{max} and G_{min} and $\phi(t)$ is the time integral of the applied voltage.

Updating the RRAM conductance is commonly performed through positive/negative programming pulses for potentiation/depression with pulse width T and constant programming voltage V_p. As a result, the discrete values of the flux are $\phi(t = nT) = V_p nT$ where n is the number of applied pulses. This technique provides precise and accurate weight updates. For $t = n\Delta T$ and substituting back into Eq. (19.8), the potentiation and depression conductances become:

$$G_{LTP} = G_{max} - \beta_P e^{-\alpha_P n}, \text{ and} \tag{19.9}$$

$$G_{LTD} = G_{min} + \beta_D e^{-\alpha_D n}, \tag{19.10}$$

respectively, where n is the pulse number, $\alpha_P = |V_p|\alpha_1 T$, and $\alpha_D = |V_D|\alpha_2 T$. The rate of change in conductance with respect to n becomes

$$\frac{dG}{dn} = \begin{cases} \beta_P \alpha_P e^{\alpha_P n}, & \text{for LTP} \\ -\beta_D \alpha_D e^{\alpha_D n}, & \text{for LTD} \end{cases}. \tag{19.11}$$

One way to quantify the device potentiation and depression asymmetry and linearity is the asymmetric nonlinearity factors [37]. The effect of these factors is reflected in the coefficients $\alpha_P, \alpha_D, \beta_P$ and β_D, which are used for the training. The potentiation asymmetric nonlinearity (PANL) factor and depression asymmetric nonlinearity (DANL) are defined as $PANL = G_{LTP}(N/2)/\Delta G - 0.5$ and $DANL = 0.5 - G_{LTD}(N/2)/\Delta G$, respectively, where N is the total number of pulses to fully potentiate the device. $PANL$ and $DANL$ are between $[0, 0.5]$. The sum of both potentiation

and depression asymmetric nonlinearities represents the total asymmetric nonlinearity (ANL), which can be written as follows for the proposed RRAM model:

$$ANL = 1 - \frac{\beta_P e^{-0.5\alpha_P N} + \beta_D e^{-0.5\alpha_D N}}{\Delta G}. \tag{19.12}$$

19.3.5.1 Asymmetric nonlinearity behavior example

An example of a synaptic device is a nonfilamentary (oxide switching) TiO_x based RRAM with a precision measured to 6 bits [38]. The $Mo/TiO_x/TiN$ device was fabricated based on a redox reaction at Mo/Tio_x interface that forms conducting MoO_x. This type of interface-based switching devices exhibit good switching variability across the entire wafer and guarantees reproducibility [38]. The asymmetric nonlinear behavior of this device is shown in Fig. 19.7A.

The proposed model was fitted and parameters were extracted for the three programming cases $\{\pm 2V,\ \pm 2.5V,$ and $\pm 3V\}$. Tables 19.1 and 19.2 show the extracted model identification parameters of the device for the three reported voltages with negligible root mean square errors (RMSE). According to the results, the higher the applied voltage, the higher the switching range. Clearly, the model parameters are the function of the applied voltage. Thus, each parameter can be modeled as a function of the applied voltage that would help to interpolate potentiation and depression curves if nonreported responses are required to be tested. The interpolated models are reported in the tables as functions of the applied voltage.

Practically, $V_p = \pm 3V$ cases would be considered since it has the widest switching range. Fig. 19.6 shows the curve fitted model on top of the reported conductance for both potentiation and depression scenarios. This device has $PANL = 0.32$ and $DANL = 0.45$ with $ANL = 0.77$.

TABLE 19.1 Extracted potentiation parameters of the $Mo/TiO_x/TiN$ device.

$V_p(V)$	G_{max} (nS)	$\alpha_P \times 10^{-3}$	$\beta_P \times 10^{-9}$	RMSE
3	674	30.58	626.8	9.07
2.5	252.7	18.23	220.22	0.6416
2	83.38	19.19	71.7	0.2276
V_p	$2.968e^{1.823V_p} - 30.4$	$2.019 \times 10^{-9} e^{7.51V_p} + 18.28$	$1.522e^{2.014V_p} - 13.78$	–

Adapted from Jaesung Park et al. Tio x-based rram synapse with 64-levels of conductance and symmetric conductance change by adopting a hybrid pulse scheme for neuromorphic computing. IEEE Electron. Device Lett., 37(12):1559–1562, 2016.

TABLE 19.2 Extracted depression parameters of the $Mo/TiO_x/TiN$ device.

$V_p(V)$	G_{min} (nS)	$\alpha_D \times 10^{-3}$	$\beta_D \times 10^{-9}$	RMSE
−3	32.95	353.4	921.9	23.696
−2.5	186.3	35.29	410.9	10.3215
−2	340.5	20.55	330.8	6.12
V_p	$307.6V_p + 955.5$	$8.14 \times 10^{-6} e^{-5.48V_p} + 20.5$	$0.009e^{-3.706V_p} + 315.9$	–

Adapted from Jaesung Park and et al. Tio x-based rram synapse with 64-levels of conductance and symmetric conductance change by adopting a hybrid pulse scheme for neuromorphic computing. IEEE Electron. Device Lett., 37(12):1559–1562, 2016.

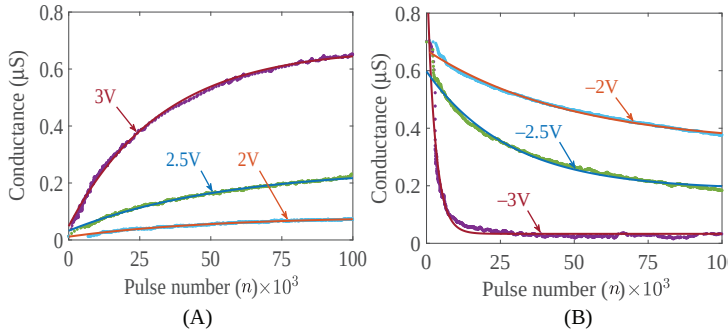


FIGURE 19.6 RRAM's conductance update (A) long-term potentiation (B) long-term depression. *Reproduced from M.E. Fouda, E. Neftci, A. Eltawil, and F. Kurdahi. Independent component analysis using rrams. IEEE Trans. Nanotechnol., 18:611–615, 2019.*

Device variations are important issues to be taken into consideration during training. There are two types of variations in RRAMs: (1) the variation during the write operation where a slightly different value is written in the device because of the randomness in the voltage variation and switching materials. This randomness can be mitigated with write–verify techniques where the written value is read to verify the value and corrected until the desired value is obtained [39]. (2) Independent device-to-device variation due to fabrication mismatch and material inhomogeneity. These variations can be included in the model by treating each parameter in the model as an independent random variable. Fig. 19.7B shows the conductance variations of multiple devices during the potentiation and depression cycles with $\pm 3V$ programming pulses. The model parameters are sampled from Gaussian sources with 25% tolerance (Variance/mean) for α, and 1% and 5% tolerances for the maximum and minimum conductances, respectively.

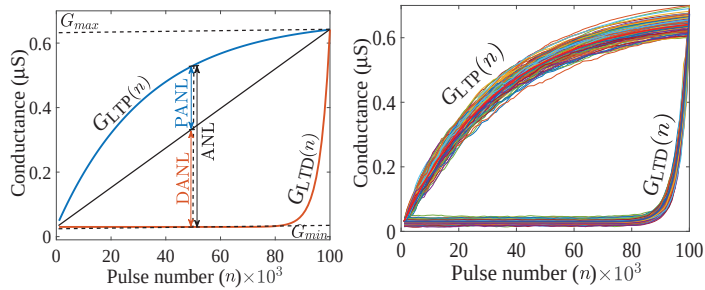


FIGURE 19.7 Nonidealities of the RRAM: (A) asymmetric nonlinear weight update and (B) device variations. *Reproduced from M.E. Fouda, E. Neftci, A. Eltawil, and F. Kurdahi. Independent component analysis using rrams. IEEE Trans. Nanotechnol., 18:611–615, 2019.*

The effect of the variation in the parameter β is considered inside the variations of α. β is modeled as a lognormal variable to have a monotonic increasing or decreasing conductance update. Thus, the second term of the conductance update has log Gaussian variable, which is e^z, multiplied by $e^{\alpha n}$ where z and α are Gaussian variables. Since the sum of two Gaussian random variables is a Gaussian random variable, the variation of β and α can be included in either of them.

19.3.5.2 RRAM updates for training

In Refs. [33,34], we proposed a method to have the resistive devices behave exactly like the learning rule where the change in each weight must be proportional to the change in the RRAM's conductance, $\Delta\mathbf{G} \propto \Delta\mathbf{w}$. To achieve this, the asymmetric nonlinear behavior of potentiation and depression is included in the learning rule. We first calculate the change in the weights for both potentiation and depression cases taking into effect the asymmetric nonlinearity of the RRAM model. In general the change in the LTP's and LTD's conductance due to applying Δn is

$$\begin{aligned}\Delta G_{LTP} &= (G_{max} - G)\left(1 - e^{-\alpha_P \Delta n}\right), \text{ and}\\ \Delta G_{LTD} &= (G - G_{min})(e^{-\alpha_D \Delta n} - 1),\end{aligned} \tag{19.13}$$

respectively, where G is the previous conductance. Clearly the relation between the rate of change in conductance and Δn is an injective function. Thus, the number of pulses to cause ΔG_{LTP} and ΔG_{LTD} are

$$\Delta n_{LTP} = -\frac{1}{\alpha_P}\ln\left(1 - \frac{\Delta G_{LTP}}{G_{max} - G(n)}\right), \text{ and} \tag{19.14}$$

$$\Delta n_{LTD} = -\frac{1}{\alpha_D}\ln\left(\frac{\Delta G_{LTD}}{G(n) - G_{min}} + 1\right), \tag{19.15}$$

respectively. After learning, $\Delta\mathbf{G}$ becomes $\mathbf{0}$. As a result, $\Delta\mathbf{n}$ also becomes zero. The update Eqs (19.15) and (19.16) require the knowledge of the weight value, meaning a read operation is needed to calculate the required number of pulses to update. In addition, they are nonlinear functions that are expensive to implement in hardware (e.g., they require log amplifiers). Thus, both can be linearized as $\ln(1-x) \approx -x(1+0.5x) \approx -x$ and $\ln(1+x) \approx x(1-0.5x) \approx x$ for $x \ll 1$. This leads to the potentiation and depression updates as given by

$$\Delta n_{LTP} = \frac{1}{\alpha_P} \frac{\Delta G_{LTP}}{G_{max} - G(n)}, \quad \Delta G_{LTP} \ll G_{max} - G(n), \text{ and} \tag{19.16}$$

$$\Delta n_{LTD} = \frac{1}{\alpha_D} \frac{|\Delta G_{LTD}|}{G(n) - G_{min}}, \quad |\Delta G_{LTD}| \ll G(n) - G_{min}, \tag{19.17}$$

It is worth mentioning that deploying the linearized update equation might result in increased training time. Thus, there is a trade-off between the training time and the complexity of the weight update hardware.

19.4 Synaptic plasticity and learning in SNN

As RRAM arrays provide a scalable physical substrate for implementing neural computations and plasticity, we now turn to the modeling of synaptic plasticity. Synaptic plasticity in the brain is realized using some constraints as in RRAMs. One of these constraints is that information necessary for performing efficient weight updates must be available at the physical location where the weights updates are computed and stored.

The brain is capable of learning and adapting at different time scales. Generally, learning processes operate in the hours to years range, which is thought to be implemented by synaptic plasticity and structural plasticity. Focusing on the former, a common synaptic plasticity process dictates that synaptic weights changes according to a modulated Hebbian-like process [40], which can be written in a functional form as:

$$\Delta W_{ij} = f(W_{ij}, S_i, S_j, M_i)$$

where M_i is some modulating function that is not yet specified. A common biologically inspired model is STDP. The classical STDP rule modifies the synaptic strengths of connected pre- and postsynaptic neurons based on the spike history in the following way: if a postsynaptic neuron generates action potential within a time interval after the presynaptic neuron has fired multiple spikes, then the synaptic strength between these two neurons becomes stronger (causal updates, long-term potentiation (LTP)). Note that STDP does not use the modulation term. Formally

$$\Delta W_{ij}^{STDP} \propto S_i(t)(\epsilon_{pre} * S_j(t)) - S_j(t)(\epsilon_{post} * S_i(t))$$

where ϵ_{post} and ϵ_{pre} are two kernels, generally of first- or second-order (exponential or double exponential) filters as they relate to the neuron dynamics, Eqs. (19.1) and (19.2). The convolution terms $\epsilon^* S(t) = \int \mathrm{d}s\epsilon(s)S(t-s)$ capture the trace of the spiking activity and serve as key building blocks for synaptic plasticity. These terms are key for learning in SNN as they provide eligibility traces or memory of the neural activity history. These traces emerge from the gradients on the neural membrane potential dynamics [41].

STDP captures the change in the postsynaptic potential amplitude in an experimental setting [42] where the pair of neurons is elicited to fire at precise times. As such, it only captures a particular temporal aspect of the synaptic plasticity dynamics. Experimental work argues that STDP alone does not account for several observations in synaptic plasticity [43]. Theoretical work suggested that synapses require complex internal dynamics on different time scales to achieve extensive memory capacity [44]. Furthermore, error-driven learning rules derived from first principles are not directly compatible with pair-wise STDP [45]. These observations are not in contradiction with seminal work of [42] as considerable variation in LTP and LTD is indeed observed. However, Ref. [45] suggests that STDP is an incomplete description of synaptic plasticity.

On the flip-side, normative approaches derive synaptic plasticity dynamics from mathematical principles. While several normative approaches exist, in the following section, we focus on three-factor rules that are particularly well suited for neuromorphic applications.

19.4.1 Gradient-based learning in SNN and three-factor rules

Three-factor rules can be viewed as extensions of Hebbian learning and STDP and are derived from a normative approach [46]. The first two factors are functions of pre- and postsynaptic activity, and the third factor is a modulating function that is relevant to the learning task. Such rules have been shown to be compatible with a wide number of unsupervised, supervised, and reinforcement learning paradigms [46], and implementations can have scaling properties comparable to that of STDP [14].

Three-factor rules can be derived from gradient descent on the SNN [15,45]. Such rules are often "local" in the sense that all the information necessary for computing the gradient is available at the postsynaptic neuron [47]. Recent digital implementations of learning use three-factor rules, where the third factor is a modulation term that depends on internal synaptic [6] or postsynaptic neuron states [14].

Three-factor rules are motivated by biology, where there are additional extrinsic factors that modulate learning, for example, dopamine, acetylcholine, or noradrenaline in reward-based learning [48], or GABA neuromodulator controlling spike time-dependent plasticity [49]. The three-factor learning rule can be written as follows:

$$\Delta W_{ij}^{3F} \propto f_{pre}(S_j(t)) f_{post}(S_i(t)) M_i \tag{19.18}$$

where f_{pre} and f_{post} correspond to functions over pre- and postsynaptic variables, respectively, and M_i is the modulating term of postsynaptic neuron i. The modulating term is a task-dependent function, which can represent error, surprise, or reward.

The equivalence of SNN with artificial neural networks discussed in Section 2, paired with synaptic plasticity derived from gradient descent suggests that same approaches used for training artificial networks can be applied to SNN. In other words, the synaptic plasticity rule can be formulated in a way that it optimizes a task-relevant objective function [47]. A machine learning description of SNN training consists of three parts: the objective function, the (neural network) model, and the optimization strategy. The objective, written as $\mathcal{L}(\mathbf{S}(\Omega), \mathbf{S_{data}})$, is a scalar function describing the performance of the task at hand (e.g., classification error, reconstruction error, free energy, etc.), where Ω are trainable parameters and $\mathbf{S}$ and $\mathbf{S_{data}}$ are neural states (spikes, membrane potentials, etc.) and input spikes, respectively, dictated by the SNN dynamics. If operating in a firing rate mode (where spike counts or mean firing rates are carriers of task-level information), $\mathbf{S}$ and $\mathbf{S_{data}}$ can be interpreted as firing rates instead. The optimization strategy consists of a parameter update derived from gradient descent on $\mathcal{L}$. If this update rule can be expressed in terms of variables that are local to the connection, then the learning rule will be called a synaptic plasticity rule.

Gradient-based approaches have been used in a wide range of work. For examples, the Tempotron is a learning rule using a membrane potential-based objective function to learn to discriminate between inputs on the basis of the spike train statistics [50]; [45] expresses the neuron model as a stochastic generative model and derive learning rules by maximizing the likelihood of a target spike train; and SpikeProp [51] is a spike-based gradient backpropagation algorithm. Several other approaches that can collectively be described as surrogate gradient descent [15] rely on approximations of the gradient to perform SNN parameter updates [41,52–54].

While the above models are computationally promising, there are important challenges in learning multilayer models on a physical substrate such as the brain. The physical substrate defines what variables are available to which processes and when. This is in stark contrast to von Neumann computers where learning processes have access to shared memory. One consequence of this limitation is the weight transport problem of gradient backpropagation, where the symmetric transpose of the weights is necessary to train deep networks. In many cases, however, the neurons and weight tables cannot be "reversed" in this fashion. Another less-studied consequence is the temporal locality due to the continual dynamics of SNN: solving the credit assignment problem in RNNs requires some memory of the previous states and inputs, either in the form of buffers or eligibility traces [55]. Both

these problems, namely the weight transport problem and temporal credit assignment, must be solved in order to successfully implement memristor-based machine inference and learning. Below we describe some promising approaches that overcome these two problems.

Feedback Alignment and Event-Driven RBP One way to alleviate the weight transport problem is to replace the weights used in backpropagation with fixed, random ones [56]. Theoretical work suggests that the network adjusts its feed-forward weights such that they align with the (random) feedback weights, which is arguably equally good in communicating gradients. This approach is naturally extended to SNN to overcome the weight transport problem. Event-driven random back propagation (eRBP) is one such rule that extends feedback alignment to meet the other constraints of learning in SNN, namely that weight updates are event-based (no separate forward and backward phases) and errors are maintained on a dedicated compartment of the neuron, rather than in a globally accessible buffer. Because it is local, it can be implemented as a presynaptic spike-driven plasticity rule modulated by top−down errors and gated by the state of the post-synaptic neuron and is simple to implement. The learning rule can be written as:

$$M_i \quad = \sum_k g_{ik} Error_k \Delta W_{ij}^{eRBP} \propto M_i Boxcar(U_i) S_j(t) \tag{19.19}$$

where g_{ik} are fixed, random weights and *Boxcar* is a symmetric function that is equal to 1 in the vicinity of $U_i = 0$, and zero otherwise. Here, M_i represents the state of the neural compartment that modulates the plasticity

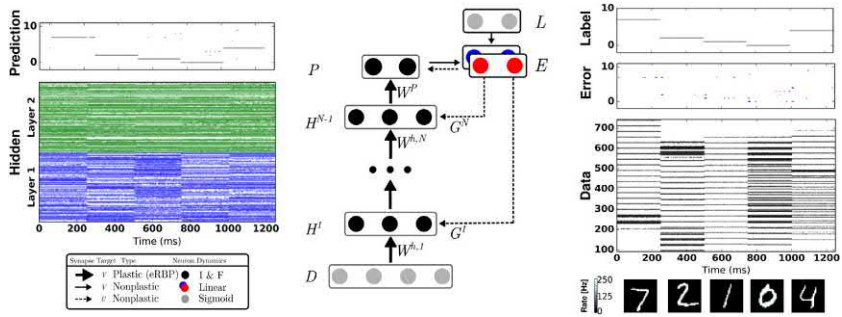


FIGURE 19.8 Network Architecture for Event-driven Random Backpropagation (eRBP). The network consists of feed-forward layers (784-200-200-10) for prediction and feedback layers for supervised training with labels (targets) $\mathcal{L}$. Full arrows indicate synaptic connections, thick full arrows indicate plastic synapses, and dashed arrows indicate synaptic plasticity modulation. Neurons in the network indicated by black circles were implemented as two-compartment LI&F neurons. The top-layer errors are proportional to the difference between labels (L) and predictions (P) and is implemented using a pair of neurons coding for positive error (blue) and negative error (red). Each hidden neuron receives inputs from a random combination of the pair of error neurons. Output neurons receive inputs from the pair of error neurons in a one-to-one fashion. *Reproduced from E. Neftci, C. Augustine, S. Paul, and G. Detorakis. Event-driven random back- propagation: Enabling neuromorphic deep learning machines. In* 2017 IEEE International Symposium on Circuits and Systems, *May 2017.*

rule according to top−down errors. Its functionality is to maintain a trace of $Error_k = target_k - S_k$ when an input spike occurs. ERBP solves the non-locality problems, leading to remarkably good performance on MNIST handwritten digit recognition tasks (Fig. 19.8), achieving close to 2% error compared to 1.9% error using gradient backpropagation on the same network architecture.

One limitation of eRBP is related to the "loop duration", that is the duration necessary from the input onset to a stable response in the error neurons. A related problem is layerwise locking in deep neural networks: because errors are backpropagated from the top layers, hidden layers must wait until the prediction is made available [58]. This duration scales with the number of layers, limiting eRBP scalability for very deep networks. The loop duration problem is caused by the temporal dynamics of the spiking neuron, which are not taken into account in Eq. (19.19).

This problem can be partly overcome by maintaining traces of the input spiking activity and a solution was reported in Ref. [41]. Their rule called Superspike is derived from gradient descent over the LI&F neuron dynamics, resulting in the following three-factor rule:

$$\Delta W_{ij}^{SS} \propto \alpha^*(Error_i \rho'(U_i)(\epsilon_{pre}{}^* S_j))) \tag{19.20}$$

where ρ describes the slope of the activation function ρ at the membrane potential U_i, and ϵ_{pre} here is the response of the postsynaptic neuron to a presynaptic spike (the impulse response function at U).

Interestingly, both Eqs. (19.20) and (19.19) are reminiscent of STDP but include further terms that vary according to some external modulation, itself related to some task. Unsurprisingly, the three terms in Eq. (19.18) can be related to the classical Widrow-Hoff (Delta) rule, where the first term is the error, the second is the derivative of the output activation function, and the third term is the input.

The loop duration problem is only partly solved with Eq. (19.20), as α and ϵ introduce memory of the previous activity into the synapses. However, this is only an approximation, as the dynamics of every layer leading to the top layer must be taken into account with one additional temporal convolution per layer. As a result, Eqs. (19.20) and (19.19) do not scale well with multiple layers.

Local Errors A more effective method to overcome the loop duration and the layerwise locking problem is to use synthetic gradients: gradients that can be computed locally, at every layer. Synthetic gradients were initially proposed to decouple one or more layers from the rest of the network to prevent layerwise locking [58]. Synthetic gradients usually involve an outer loop consisting of a full backpropagation through the network. While this provides convergence guarantees, a full backpropagation step cannot be done locally in SNNs. Instead, relying on initialization of the local random

classifier weights and forgoing the outer loop training yields good empirical results [59].

Using layerwise local classifiers [59], the gradients are computed locally using pseudotargets (for classification, the labels themselves). To take the temporal dynamics of the neurons into account, the learning rule is similar to SuperSpike [41]. However, the gradients are computed locally through a fixed random projection of the network activities into a local classifier. Learning is achieved using a local rate-based cost function reminiscent of readout neurons in liquid state machines [60], but where the readout is performed over a fixed random combination of the neuron outputs. The readout does not have a temporal convolution term in the cost function, the absence of which enables linear scaling, and does not prevent learning precise temporal spike trains (Fig. 19.9). The resulting learning dynamics are called DEep COntinuous Local LEarning (DECOLLE), and written as:

$$\Delta W_{ij}^{DECOLLE} \propto \left(\sum_k g_{ik} Error_k \right) \rho'(U_i)(\epsilon_{pre} * S_j). \tag{19.21}$$

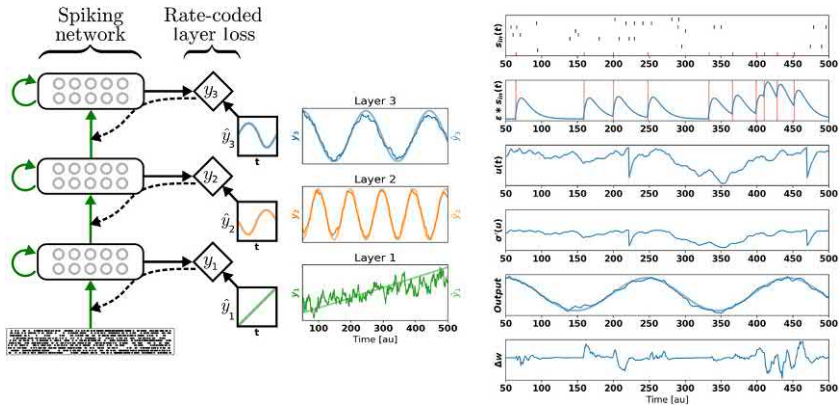


FIGURE 19.9 Deep continuous local learning example. (Left) Each layer consists of spiking neurons with continuous dynamics. Each layer feeds into a local classifier through fixed, random connections (diamond-shaped, y). The classifier is trained to produce auxiliary targets $\hat{y}$. Errors in the local classifiers are propagated through the random connections to train the input weights, but no further (curvy, dashed line). To simplify the learning rule and enable linear scaling of the computations, the cost function is formulated using a rate code. The states of the spiking neurons (membrane potential, synaptic states, refractory state) are carried forward in time. Consequently, even in the absence of recurrent connections, the neurons are stateful in the sense of RNNs. (Right) Snapshot of the neural states illustrating the DECOLLE learning rule in the top layer. In this example, the network is trained to produce three time-varying pseudotargets $\hat{y}_1$, $\hat{y}_2$, and $\hat{y}_3$. *Reproduced from E. Neftci, C. Augustine, S. Paul, and G. Detorakis. Event-driven random back-propagation: Enabling neuromorphic deep learning machines. In* 2017 IEEE International Symposium on Circuits and Systems*, May 2017.*

Here the error is computed with respect to a random linear combination of the neuron outputs: $Error_k = target_k - \sum_i g_{ik} S_i$. While SuperSpike scales at least quadratically with the number of neurons, learning with local errors scales linearly [61]. Linearity greatly improves the memory and computational cost of computing the weight updates and simplifies potential RRAM implementations (see Section 5.2).

Independent Component Analysis with Three-Factor Rule Independent component analysis (ICA) is a very powerful tool to solve the cocktail party problem (blind source separation), feature extraction (sparse coding) and can be utilized in many applications such as denoising images, electroencephalograms (EEG) signals, and telecommunications [62]. ICA consists of finding mutually independent and non-Gaussian hidden factors (components), $\mathbf{s}$, that form a set of signals or measurements, $\mathbf{x}$. This problem can be mathematically described for linearly mixed components as $\mathbf{x} = \mathbf{A}\mathbf{s}$ where A is the mixing matrix. Both $\mathbf{A}$ and $\mathbf{s}$ are unknowns. In order to find the independent components (sources), the problem can be formulated as $\mathbf{u} = \mathbf{W}\mathbf{x}$ where $\mathbf{x}$ is the mixed signals (inputs of ICA algorithm), $\mathbf{W}$ is the weight matrix (demixing matrix), $\mathbf{u}$ is the outputs of the ICA algorithm (independent components). ICA's strength lies in utilizing the mutual statistical independence of components to separate the sources.

Recently Isomura and Toyoizumi proposed a biological plausible learning rule called *Error-Gated Hebbian Rule (EGHR)* inspired from the standard Hebbian rule [63] that enables local and efficient learning to find the independent components. The EGHR learning rule can be written as

$$\Delta \mathbf{W}^{EHGR} = \eta \langle (E_o - E(\mathbf{u})) g(\mathbf{u}) \mathbf{x}^T \rangle \tag{19.22}$$

where η is the learning rate, $\langle \cdot \rangle$ is the expectation over the ensemble (training samples), $g(u_i)x_j$ is the Hebbian learning rule, $g(u_i)$ and x_j are the postsynaptic and presynaptic terms of the neuron, respectively, $(E_o - E(\mathbf{u}))$ is the global error signal which consists of E_o which is a constant, and $E(\mathbf{u})$ which is the surprise or reward that guides the learning. The cost function of EGHR is defined as $\mathcal{L} = c\frac{1}{2}\langle (E_0 - E(\mathbf{u}))^2 \rangle$. It was proven mathematically and numerically that this learning rule is robust, stable and its equilibrium point is proportional to the inverse of the mixture matrix, that is the solution of ICA. This learning rule is a clear example of three-factor learning where the modulating is represented in the surprise, $(E_o - E(\mathbf{u}))$. This learning rule is a three-factor rule and can be performed using spiking neuron by following an implementation similar to [64].

ICA assumes that the sources are linearly mixed using a mixture matrix $\mathbf{A}$. The final weight matrix, $\mathbf{w}$, should converge to $c\mathbf{A}^{-1}$, which is still a valid solution since c is a scaling factor. As previously discussed, ΔG_{ij} can be replaced by $\eta' \Delta w_{ij}$ to match the synaptic dynamics [33,34], where η' is the scaled learning rate, and Δw_{ij} is given by EGHR. Thus the final equations for potentiation and depression pulses can be written as follows:

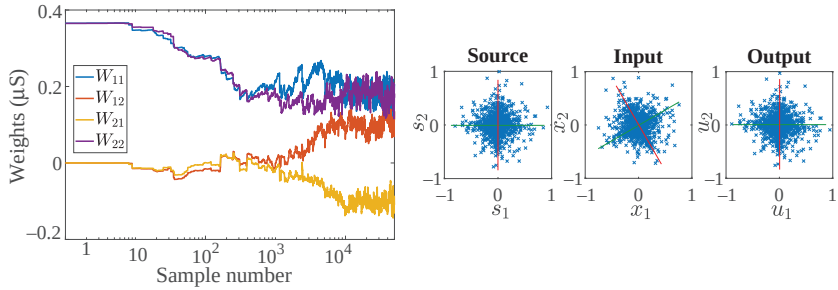


FIGURE 19.10 The online training results versus training time. (A) Evolution of the weights and (B) visual results of the input and the output. *Reproduced from J. Kaiser, H. Mostafa, and E. Neftci. Synaptic plasticity for deep continuous local learning.* arXiv preprint arXiv:1811.10766*, 2018.*

$$\Delta n_{ij}|_{LTP} \approx \frac{1}{\alpha_P}\left(\frac{\eta'(E_o - E(\mathbf{u}))g(u_j)x_i}{G_{max} - G_{ij}(n)}\right), \text{ and} \tag{19.23}$$

$$\Delta n_{ij}|_{LTD} = -\frac{1}{\alpha_D}\left(\frac{\eta'(E_o - E(\mathbf{u}))g(u_j)x_i}{G_{ij}(n) - G_{min}}\right), \tag{19.24}$$

respectively. By programming the RRAMs using the previous equations, the circuit behaves as required and compensates for the asymmetric nonlinearity of the devices.

As a test bench for the proposed technique, we considered two Laplacian random variables that are generated and mixed using a mixture matrix that is set to a rotation matrix $\mathbf{A} = (\cos\theta, -\sin\theta; \sin\theta, \cos\theta)$ with $\theta = \pi/6$. Fig. 19.10 shows the results of the online learning of independent components of the mixed signals. Fig. 19.10A shows the weight evolution during the training. Clearly there are some oscillations in the weights around the final solution after 10^4 samples because of the continuous online learning and the device variations. This can be avoided by using dynamic learning rate such as Adaptive Moment Estimation (Adam). A visual representation of the signals before mixing, after mixing, and after training is shown in Fig. 19.10B, which depicts the similarity between the source and output signals.

19.5 Stochastic SNNs

Till now, we have considered fully deterministic neuron and synapse models. However, as discussed in Section 3.5.2, the writing (and in some cases, reading) of RRAMs values are stochastic. Additionally analog VLSI neuron circuits have significant variability across neurons due to fabrication mismatch

(fixed pattern noise) and behave stochastically due to noise intrinsic to the device operation. The variability at the device level can be taken into account in SNN models and sometimes be exploited for improving learning performance and implementing probabilistic inference [65,66]. Here we list avenues for implementing online learning in-memory devices that exploit the stochasticity in the neurons and synapses.

A stochastic model of the neurons can be expressed as:

$$P(S_i|\mathbf{s}) = \rho(U_i(t)) \tag{19.25}$$

where ρ_i is the stochastic intensity (the equivalent of the activation function in artificial neurons) and η and ϵ are kernels that reflect neural and synaptic dynamics, for example, refractoriness, reset, and postsynaptic potentials [40]. The stochastic intensity can be derived or estimated experimentally if the noiseless membrane potential ($U_i(t)$) can be measured at the times of the spike [67]. This type of stochastic neuron model drives numerous investigations in theoretical neuroscience and forms the starting point for other types of adapting SNNs capable of efficient communication [68].

19.5.1 Learning in stochastic SNNs

Neural and synaptic unreliability can induce the necessary stochasticity without requiring a dedicated source of stochastic inputs, for example, the unreliable transmission of synaptic vesicles in biological neurons. This is a well-studied phenomenon [69,70], and many studies suggested it as a major source of stochasticity in the brain [71–74]. In the cortex, synaptic failures were argued to reduce energy consumption while maintaining the computational information transmitted by the postsynaptic neuron [75]. More recent work suggested synaptic sampling as a mechanism for representing uncertainty in the brain, and its role in synaptic plasticity and rewiring [76].

Strikingly, the simplest model of synaptic unreliability, a "*blank-out*" synapse, can improve the performance of SNNs in practical machine learning tasks over existing solutions, while being extremely easy to implement in hardware [77] and often naturally occurring in emerging memory technologies [78–80].

One approach to learning with such neurons and synapses is Event-Driven Contrastive Divergence (ECD), using ideas borrowed from Contrastive Divergence in restricted Boltzmann machines [81]. The stochastic neural network produces samples from a probability distribution, and STDP carries out the weight updates according to the Contrastive Divergence rule in an online, asynchronous fashion. In terms of the three-factor rule above, ECD can be written as:

$$\Delta W_{ij}^{ECD} = M_i(t)\Delta W_{ij}^{STDP} \tag{19.26}$$

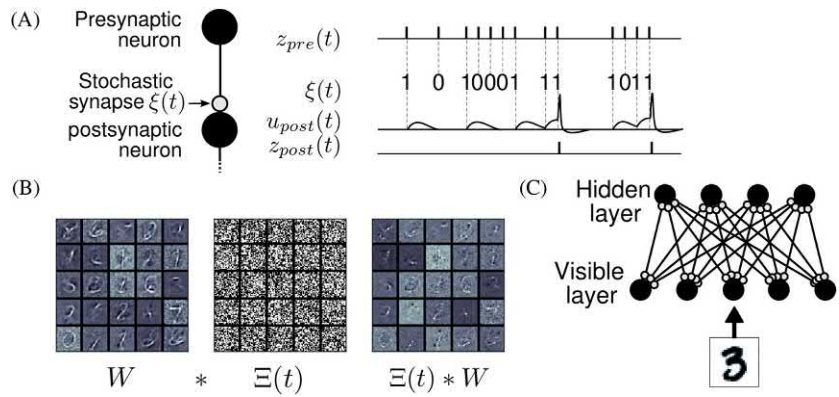


FIGURE 19.11 The synaptic sampling machines (SSM). (A) At every occurrence of a presynaptic event, a pre-synaptic event is propagated to the postsynaptic neuron with probability p. (B) Synaptic stochasticity can be viewed as a continuous DropConnect method [83] where weights are masked by a binary matrix $\Theta(t)$, where * denotes element-wise multiplication. (C) SSM Network architecture, consisting of a visible and a hidden layer. *Reproduced from E.O. Neftci, B.U. Pedroni, S. Joshi, M. Al-Shedivat, and G. Cauwenberghs. Stochastic synapses enable efficient brain-inspired learning machines. Front. Neurosci., 10(241), 2016.[84]*

where $M_i(t) = 1$ during the "data" phase and $M_i(t) = -1$ during the "reconstruction" phase. These neural networks can be viewed as a stochastic counterpart of Hopfield networks [82], but where stochasticity is caused by multiplicative noise at the connections (synapses) or at the nodes (neurons) (Fig. 19.11).

ECD requires symmetric weights, which is difficult to achieve due to the weight transport problem discussed above. A variation of ECD called random Contrastive Hebbian learning (rCHL) [85], replaces the transpose of the synaptic weights with fixed random matrices. This was performed similar to Feedback Alignment (FDA) [56]. Contrastive Hebbian Learning (CHL) is similar to Contrastive Divergence, but it employs continuous nonlinear dynamics at the neuronal level. Like Contrastive Divergence, it does not rely on a special circuitry to compute gradients (but can be interpreted as the gradient of an objective function), allows information to flow in a coupled, synchronous way, and is grounded upon Hebb's learning rule. CHL uses feedback to transmit information from the output layer to hidden(s) layer(s), and in instances when the feedback gain is small (such as in the clamped phase), has been demonstrated by Xie and Seung to be equivalent to Backpropagation [86]. Using this approach, the information necessary for learning propagates backward, although it is not transmitted through the same axons (as required in the symmetric case), but instead via separate pathways or neural populations.

Equilibrium propagation (EP) describes another modification of CHL that generalizes the objective functions it can solve and improve on its theoretical groundings. In EP, the neuron dynamics are derived from the energy

function, although it requires symmetric weights. The energy function used in EP includes a mean-squared error term on the output units, allowing the output to be weakly clamped to the true outputs (e.g., labels). The neuron model takes a form which is reminiscent of the LI&F neuron. The recurrent dynamics in the network affect the other layers in the network, similar to CHL. Both rCHL and EP were formulated for continuous (rate-based) neurons, although their implementation with spikes is straightforward following the same approach as SSMs.

Learning in stochastic neuron networks can also be performed using the surrogate gradient approach and three-factor rules. In this case, a simple approach is to use the stochastic intensity ρ as a drop-in replacement of the neural activation function for purposes of computing the weight updates [15]. Stochasticity plays a regularization role similar to dropout in deep learning [87] and weight normalization [88], an online modification of batch normalization.

19.5.2 Three-factor learning in memristor arrays

So far, we have discussed how to implement gradient-based learning as local synaptic plasticity rules in SNN. In many cases, gradient-based learning provides superior results compared to STDP and takes the form of three-factor rules. These rules are biologically credible since pre- and postsynaptic activities are available at the level of the neuron and neurotransmitters in the brain can carry the extrinsic factor. However, besides the LTP and LTD asymmetry problems already discussed, the implementation of the three factors in memristor arrays come with certain challenges. The first challenge concerns the implementation of the synaptic traces. In certain simple cases such as when the subthreshold neural dynamics are linear, only one trace for each neuron involved in the learning connections is required for learning [61], similar to the STDP case. Previous work has demonstrated STDP learning in RRAM and hence capture some form of a neural trace. The majority of these include additional CMOS circuitry in a "hybrid configuration" [89] to enable STDP. The simplest of these implementations consists of a 2T1R configuration that enables an update when both terminals are high (both spike). While this is sufficient for the case where $\alpha = \beta = 0$, an additional mechanism that filters the spike is necessary to recover STDP like curves when $\alpha > 0$ or $\beta > 0$. This can be achieved with a circuit that is similar to that of the synapse [90] or calcium variables [91]. For more complex neuron dynamics (such as nonlinear neuron dynamics), at least one trace per synapse is required [92], and ideally, one trace per connection and per neuron [55].

Since the synaptic trace dynamics follow similar first-order (RC) dynamics, the same circuits used to implement first-order synaptic dynamics can be used to implement synaptic traces [93] or dedicated calcium dynamics circuits [94]. However, scalability can become an issue: the same amount of memory

for storing the weights is necessary for computing the traces, but the latter must be carried out continuously. One potential solution comes from recent work in using diffusive memristors to implement the leaky dynamics of integrate and fire neurons [95], which can be used for computing neural traces.

The second issue concerns the modulation. In many gradient-based three-factor rules, the modulation of the learning rule is specific to each neuron, not each synapse. This means that a similar approach to eRBP Eq. (19.19), where a separate neuron compartment is used for maintaining the modulation factor can in principle be used in memristor arrays, that is the weight update can consist in the two factors $(\epsilon_{pre} * S_j))$, where $M_i \rho'(U_i)$.

Additionally, the variability in the conductance reading and writing can cause the learning to fail or slow down. Independent noise in the read or write is not a problem and can even help learning, as discussed in the stochastic SNN section. Fixed pattern noise, however, can be problematic as it translates into variable learning rates per weight and can impair learning.

19.6 Concluding remarks

In this chapter, we presented neural and synaptic models for learning in SNNs. In particular we focused on synaptic plasticity models that use approximate gradient-based learning that is potentially compatible with a neuromorphic implementation and memristor arrays. Gradient-based learning in SNNs generally provides the best performances on many applications, such as image recognition, probabilistic inference, and ICA. The mathematical modeling of the synaptic plasticity revealed that these dynamics take the form of three-factor rules, which can be viewed as a type of modulated Hebbian learning. While Hebbian learning or its spiking counterpart, STDP have been previously demonstrated in memristors, three-factor rules also require modulation of the learning at the postsynaptic neurons. Furthermore, if the neuron and synapse model are equipped with temporal dynamics, then it may become necessary to maintain pre- and postsynaptic activity traces in the crossbar to address the temporal credit assignment problem. Through the mathematical models, we identified the approaches viable for implementing the modulation and the neural traces with memristors.

References

[1] C. Mead, Neuromorphic electronic systems, Proc. IEEE 78 (10) (1990) 1629–1636.

[2] B.V. Benjamin, P. Gao, E. McQuinn, S. Choudhary, A.R. Chandrasekaran, J. Bussat, et al., Neurogrid: a mixed-analog-digital multichip system for large-scale neural simulations, Proc. IEEE 102 (5) (2014) 699–716.

[3] E. Chicca, F. Stefanini, and G. Indiveri. Neuromorphic electronic circuits for building autonomous cognitive systems. *Proc IEEE*, 2013.

[4] J. Park, S. Ha, T. Yu, E. Neftci, and G. Cauwenberghs. A 65k-neuron 73-mevents/s 22-pj/event asynchronous micro-pipelined integrate-and-fire array transceiver. In *Biomedical Circuits and Systems Conference (Bio-CAS)*. IEEE, Oct. 2014.

[5] J. Schemmel, D. Brüderle, A. Grübl, M. Hock, K. Meier, and S. Millner. A wafer-scale neuromorphic hardware system for large-scale neural modeling. In *International Symposium on Circuits and Systems, ISCAS 2010*, pages 1947−1950. IEEE, 2010.

[6] M. Davies, N. Srinivasa, T.H. Lin, G. Chinya, P. Joshi, A. Lines, et al. Loihi: a neuromorphic manycore processor with on-chip learning. *IEEE Micro*, PP(99):1−1, 2018. ISSN 0272-1732. https://doi.org/10.1109/MM.2018.112130359.

[7] S.B. Furber, F. Galluppi, S. Temple, L. Plana, et al., The spinnaker project, Proc. IEEE 102 (5) (2014) 652−665.

[8] P.A. Merolla, J.V. Arthur, R. Alvarez-Icaza, A.S. Cassidy, J. Sawada, F. Akopyan, et al., A million spiking-neuron integrated circuit with a scalable communication network and interface, Science 345 (6197) (2014) 668−673.

[9] N. Qiao, H. Mostafa, F. Corradi, M. Osswald, F. Stefanini, D. Sumislawska, et al., A reconfigurable on-line learning spiking neuromorphic processor comprising 256 neurons and 128k synapses, Front. Neurosci. 9 (2015).

[10] M. Courbariaux, Y. Bengio, and J.-P. David. Low precision arithmetic for deep learning. *arXiv preprint arXiv:1412.7024*, 2014.

[11] G. Indiveri, B. Linares-Barranco, T.J. Hamilton, A. van Schaik, R. Etienne-Cummings, T. Delbruck, et al., Neuromorphic silicon neuron circuits, Front. Neurosci. 5 (2011) 1−23. ISSN 1662-453X. https://doi.org/10.3389/fnins.2011.00073.

[12] W. Gerstner, W.M. Kistler, R. Naud, L. Paninski, Neuronal Dynamics: From Single Neurons to Networks and Models of Cognition, Cambridge University Press, 2014.

[13] C. Bartolozzi, G. Indiveri, Synaptic dynamics in analog VLSI, Neural Computation 19 (10) (Oct 2007) 2581−2603. Available from: https://doi.org/10.1162/neco.2007.19.10.2581.

[14] G. Detorakis, S. Sheik, C. Augustine, S. Paul, B.U. Pedroni, N. Dutt, et al., Neural and synaptic array transceiver: a brain-inspired computing framework for embedded learning, Front. Neurosci. 12 (2018) 583. ISSN 1662-453X. https://doi.org/10.3389/fnins.2018.00583. URL https://www.frontiersin.org/article/10.3389/fnins. 2018.00583.

[15] E.O. Neftci, H. Mostafa, and F. Zenke. Surrogate gradient learning in spiking neural networks. *arXiv preprint arXiv:1901.09948*, 2019.

[16] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, and Y. Bengio. Binarized neural networks: training deep neural networks with weights and activations constrained to + 1 or-1. *arXiv preprint arXiv:1602.02830*, 2016.

[17] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi. Xnor-net: imagenet classification using binary convolutional neural networks. In *European Conference on Computer Vision*, pages 525−542. Springer, 2016.

[18] R. Andri, L. Cavigelli, D. Rossi, and L. Benini. Yodann: a ultra-low power convolutional neural network accelerator based on binary weights. In *2016 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pages 236−241. IEEE, 2016.

[19] Y. Umuroglu, N.J. Fraser, G. Gambardella, M. Blott, P. Leong, M. Jahre, and et al. Finn: a framework for fast, scalable binarized neural network inference. In *Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, pages 65−74. ACM, 2017.

[20] X. Sun, X. Peng, P.-Y. Chen, R. Liu, J-sun Seo, and S. Yu. Fully parallel rram synaptic array for implementing binary neural network with (+ 1,- 1) weights and (+ 1, 0) neurons. In *Design Automation Conference (ASP-DAC), 2018 23rd Asia and South Pacific*, pages 574−579. IEEE, 2018.

[21] L. Wilson. *International technology roadmap for semiconductors. Semiconductor Industry Association*, 2013.

[22] Can Li, et al., Efficient and self-adaptive in-situ learning in multilayer memristor neural networks, Nat. Commun. (2018) 2385.

[23] M. Prezioso, F. Merrikh-Bayat, B.D. Hoskins, G.C. Adam, K.K. Likharev, D.B. Strukov, Training and operation of an integrated neuromorphic network based on metal-oxide memristors, Nature 521 (7550) (2015) 61−64.

[24] Chih-Cheng Chang and et al. Mitigating asymmetric nonlinear weight update effects in hardware neural network based on analog resistive synapse. *IEEE J Emerg Selected Topics Circuits Systems*, 2017.

[25] M.E. Fouda, J. Lee, A.M. Eltawil, and F. Kurdahi. Overcoming crossbar nonidealities in binary neural networks through learning. In *2018 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH)*, pages 1−3. IEEE, 2018b.

[26] S. Yu, Neuro-inspired computing with emerging nonvolatile memorys, Proc. IEEE 106 (2) (2018) 260−285.

[27] B. Chen, Y. Lu, B. Gao, Y.H. Fu, F.F. Zhang, P. Huang, et al. Physical mechanisms of endurance degradation in tmo-rram. In *2011 International Electron Devices Meeting*, pages 12−3. IEEE, 2011.

[28] M. Zhao, H. Wu, B. Gao, X. Sun, Y. Liu, P. Yao, et al. Characterizing endurance degradation of incremental switching in analog rram for neuromorphic systems. In *2018 IEEE International Electron Devices Meeting (IEDM)*, pages 20−2. IEEE, 2018.

[29] C. Nail, G. Molas, P. Blaise, G. Piccolboni, B. Sklenard, C. Cagli, et al. Understanding rram endurance, retention and window margin trade-off using experimental results and simulations. In *2016 IEEE International Electron Devices Meeting (IEDM)*, pages 4−5. IEEE, 2016.

[30] S. Jain, A. Sengupta, K. Roy, and A. Raghunathan. Rx-caffe: framework for eval- uating and training deep neural networks on resistive crossbars. *arXiv preprint arXiv:1809.00072*, 2018.

[31] M. Azzaz, E. Vianello, B. Sklenard, P. Blaise, A. Roule, C. Sabbione, et al. Endurance/retention trade off in hfox and taox based rram. In *2016 IEEE 8th International Memory Workshop (IMW)*, pages 1−4. IEEE, 2016.

[32] M.E. Fouda, A.M. Eltawil, F. Kurdahi, Modeling and analysis of passive switching crossbar arrays, IEEE Trans. Circuits Syst. I: Regul. Pap. 65 (1) (2018) 270−282.

[33] M.E. Fouda, E. Neftci, A. Eltawil, F. Kurdahi, Independent component analysis using rrams, IEEE Trans. Nanotechnol. 18 (2019) 611−615. ISSN 1536-125X. https://doi.org/10.1109/TNANO.2018.2880734.

[34] M.E. Fouda, A.M. Eltawil, and F. Kurdahi. On resistive memories: one step row readout technique and sensing circuitry. *arXiv preprint arXiv:1903.01512*, 2019b.

[35] H. Kim, T. Kim, J. Kim, J.-J. Kim, Deep neural network optimized to resis- tive memory with nonlinear current-voltage characteristics, ACM J. Emerg. Technol. Comput. Syst. (JETC) 14 (2) (2018) 15.

[36] J. Schemmel, J. Fieres, and K. Meier. Wafer-scale integration of analog neural networks. In *Proceedings of the IEEE International Joint Conference on Neural Networks*, 2008.

[37] J. Woo, S. Yu, Resistive memory-based analog synapse: the pursuit for linear and symmetric weight update, IEEE Nanotechnol. Mag. 12 (3) (2018) 36−44.

[38] Jaesung Park, et al., Tio x-based rram synapse with 64-levels of conductance and symmetric conductance change by adopting a hybrid pulse scheme for neuromorphic computing, IEEE Electron. Device Lett. 37 (12) (2016) 1559−1562.

[39] F.M. Puglisi, C. Wenger, P. Pavan, A novel program-verify algorithm for multi-bit operation in hfo 2 rram, IEEE Electron. Device Lett. 36 (10) (2015) 1030−1032.

[40] W. Gerstner, W. Kistler, Spiking Neuron Models. Single Neurons, Populations, Plasticity, Cambridge University Press, 2002.
[41] F. Zenke and S. Ganguli. Superspike: supervised learning in multi-layer spiking neural net- works. *arXiv preprint arXiv:1705.11146*, 2017.
[42] G.-Q. Bi, M.-M. Poo, Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type, J. Neurosci. 18 (24) (1998) 10464−10472.
[43] H.Z. Shouval, S.S.-H. Wang, G.M. Wittenberg, Spike timing dependent plasticity: a consequence of more fundamental learning rules, Front. Computational Neurosci. 4 (2010) 19.
[44] S. Lahiri and S. Ganguli. A memory frontier for complex synapses. In C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 1034−1042. 2013.
[45] J.-P. Pfister, T. Toyoizumi, D. Barber, W. Gerstner, Optimal spike-timing-dependent plasticity for precise action potential firing in supervised learning, Neural Computation 18 (6) (2006) 1318−1348.
[46] R. Urbanczik, W. Senn, Learning by the dendritic prediction of somatic spiking, Neuron 81 (3) (2014) 521−528.
[47] E.O. Neftci, Data and power efficient intelligence with neuromorphic learning machines, iScience 5 (2018) 52−68. ISSN 2589-0042. https://doi.org/10.1016/j.isci.2018.06.010. URL http://www.sciencedirect. com/science/article/pii/S2589004218300865.
[48] W. Schultz, Getting formal with dopamine and reward, Neuron 36 (2) (2002) 241−263.
[49] V. Paille, E. Fino, K. Du, T. Morera-Herreras, S. Perez, J.H. Kotaleski, et al., Gabaergic circuits control spike-timing-dependent plasticity, J. Neurosci. 33 (22) (2013) 9353−9363. ISSN 0270-6474. https://doi.org/10.1523/JNEUROSCI.5796-12.2013. URL http://www.jneurosci.org/content/33/22/9353.
[50] R. Gütig, H. Sompolinsky, The tempotron: a neuron that learns spike timing−based decisions, Nat. Neurosci. 9 (2006) 420−428. Available from: https://doi.org/10.1038/nn1643.
[51] S.M. Bohte, J.N. Kok, J. A La Poutré, Spikeprop: backpropagation for networks of spiking neurons, ESANN (2000) 419−424.
[52] D. Huh and T.J. Sejnowski. Gradient descent for spiking neural networks. *arXiv preprint arXiv:1706.04698*, 2017.
[53] N. Anwani and B. Rajendran. Normad-normalized approximate descent based supervised learning rule for spiking neurons. In *Neural Networks (IJCNN), 2015 International Joint Conference on*, pages 1−8. IEEE, 2015.
[54] S.B. Shrestha and G. Orchard. Slayer: spike layer error reassignment in time. *arXiv preprint arXiv:1810.08646*, 2018.
[55] R.J. Williams, D. Zipser, A learning algorithm for continually running fully recurrent neural networks, Neural Computation 1 (2) (1989) 270−280.
[56] T.P. Lillicrap, D. Cownden, D.B. Tweed, C.J. Akerman, Random synaptic feedback weights support error backpropagation for deep learning, Nat. Commun. 7 (2016).
[57] E. Neftci, C. Augustine, S. Paul, and G. Detorakis. Event-driven random back-propagation: enabling neuromorphic deep learning machines. In *2017 IEEE International Symposium on Circuits and Systems*, May 2017a.
[58] M. Jaderberg, W.M. Czarnecki, S. Osindero, O. Vinyals, A. Graves, and K. Kavukcuoglu. Decoupled neural interfaces using synthetic gradients. *arXiv preprint arXiv:1608.05343*, 2016.
[59] H. Mostafa, V. Ramesh, and G. Cauwenberghs. Deep supervised learning using local errors. *arXiv preprint arXiv:1711.06756*, 2017.

[60] W. Maass, T. Natschläger, H. Markram, Real-time computing without stable states: a new framework for neural computation based on perturbations, Neural Computation 14 (11) (2002) 2531–2560.
[61] J. Kaiser, H. Mostafa, and E. Neftci. Synaptic plasticity for deep continuous local learning. *arXiv preprint arXiv:1812.10766*, 2018.
[62] Aapo Hyvärinen, et al., Independent Component Analysis, volume 46, John Wiley & Sons, 2004.
[63] T. Isomura, T. Toyoizumi, A local learning rule for independent component analysis, Sci. Rep. 6 (2016) 28073.
[64] C. Savin, P. Joshi, J. Triesch, Independent component analysis in spiking neurons, PLoS Computational Biol. 6 (4) (2010) e1000757.
[65] R. Naous, M. AlShedivat, E. Neftci, G. Cauwenberghs, K.N. Salama, Memristor- based neural networks: synaptic versus neuronal stochasticity, Aip Adv. 6 (11) (2016) 111304.
[66] D. Querlioz, O. Bichler, A.F. Vincent, C. Gamrat, Bioinspired programming of memory devices for implementing an inference engine, Proc. IEEE 103 (8) (2015) 1398–1416.
[67] R. Jolivet, A. Rauch, H.-R. Lüscher, W. Gerstner, Predicting spike timing of neocortical pyramidal neurons by simple threshold models, J. Computational Neurosci. 21 (1) (2006) 35–49.
[68] D. Zambrano and S.M. Bohte. Fast and efficient asynchronous neural computation with adapting spiking neural networks. *arXiv preprint arXiv:1609.02053*, 2016.
[69] B. Tiago, K. Staras, The probability of neurotransmitter release: variability and feedback control at single synapses, Nat. Rev. Neurosci. 10 (5) (2009) 373–383.
[70] B. Katz, Nerve, Muscle, and Synapse, McGraw-Hill, New York, 1966.
[71] L.F. Abbott, W.G. Regehr, Synaptic computation, Nature 431 (2004) 796–803.
[72] A.A. Faisal, L.P.J. Selen, D.M. Wolpert, Noise in the nervous system, Nat. Rev. Neurosci. 9 (4) (2008) 292–303.
[73] R. Moreno-Bote, Poisson-like spiking in circuits with probabilistic synapses, PLoS Computational Biol. 10 (7) (2014) e1003522.
[74] Y. Yarom, J. Hounsgaard, Voltage fluctuations in neurons: signal or noise? Physiological Rev. 91 (3) (2011) 917–929.
[75] W.B. Levy, R.A. Baxter, Energy-efficient neuronal computation via quantal synaptic failures, J. Neurosci. 22 (11) (2002) 4746–4755.
[76] D. Kappel, S. Habenschuss, R. Legenstein, and W. Maass. Network plasticity as bayesian inference. *arXiv preprint arXiv:1504.05143*, 2015.
[77] D.H. Goldberg, G. Cauwenberghs, A.G. Andreou, Probabilistic synaptic weighting in a reconfigurable network of VLSI integrate-and-fire neurons, Neural Netw. 14 (6–7) (2001) 781–793.
[78] M. Al-Shedivat, R. Naous, E. Neftci, G. Cauwenberghs, and K.N. Salama. Inherently stochastic spiking neurons for probabilistic neural computation. In *IEEE EMBS Conference on Neural Engineering*, Apr 2015.
[79] S. Saïghi, C.G. Mayr, T. Serrano-Gotarredona, H. Schmidt, G. Lecerf, J. Tomas, et al., Plasticity in memristive devices for spiking neural networks, Front. Neurosci. 9 (2015) 51.
[80] S. Yu, B. Gao, Z. Fang, H. Yu, J. Kang, H.-S.P. Wong, Stochastic learning in oxide binary synaptic device for neuromorphic computing, Front. Neurosci. 7 (2013).
[81] G.E. Hinton, Training products of experts by minimizing contrastive divergence, Neural Computation 14 (8) (2002) 1771–1800.
[82] J.J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, Proc Natl Acad. Sci. U S A 79 (8) (1982) 2554–2558.

[83] L. Wan, M. Zeiler, S. Zhang, Y.L. Cun, R. Fergus, Regularization of neural networks using dropconnect, Proc. 30th Int. Conf. Mach. Learn. (ICML-13) (2013) 1058–1066.

[84] E.O. Neftci, B.U. Pedroni, S. Joshi, M. Al-Shedivat, G. Cauwenberghs, Stochastic synapses enable efficient brain-inspired learning machines, Front. Neurosci. 10 (241) (2016). ISSN 1662-453X. https://doi.org/10.3389/fnins.2016.00241.

[85] G. Detorakis, T. Bartley, and E. Neftci. Contrastive hebbian learning with random feedback weights. *Neural Networks*, 2018a. URL https://arxiv.org/abs/1806.07406. (accepted).

[86] X. Xie, H.S. Seung, Equivalence of backpropagation and contrastive hebbian learning in a layered network, Neural Computation 15 (2) (2003) 441–454.

[87] E.O. Neftci, C. Augustine, S. Paul, G. Detorakis, Event-driven random back- propagation: enabling neuromorphic deep learning machines, Front. Neurosci. 11 (2017) 324. ISSN 1662-453X. https://doi.org/10.3389/fnins.2017.00324.

[88] E. Neftci. Stochastic synapses as resource for efficient deep learning machines. In *Electron Devices Meeting (IEDM), 2017 IEEE International*, pages 11–1. IEEE, 2017.

[89] D. Ielmini, Brain-inspired computing with resistive switching memory (rram): devices, synapses and neural networks, Microelectronic Eng. 190 (2018) 44–53.

[90] C. Bartolozzi and G. Indiveri. A silicon synapse implements multiple neural computational primitives. The Neuromorphic Engineer, 2008.

[91] S. Mitra, S. Fusi, and G. Indiveri. A VLSI spike-driven dynamic synapse which learns only when necessary. In *International Symposium on Circuits and Systems (ISCAS), 2006*, pages 2777–2780. IEEE, May 2006. https://doi.org/10.1109/ISCAS.2006.1693200.

[92] G. Bellec, F. Scherr, E. Hajek, D. Salaj, R. Legenstein, and W. Maass. Biologically inspired alternatives to backpropagation through time for learning in recurrent neural nets. *arXiv preprint arXiv:1901.09049*, 2019.

[93] C. Bartolozzi, G. Indiveri, Silicon synaptic homeostasis, Brain Inspired Cognit. Systems, BICS 2006 (2006) 1–4.

[94] F.L.M. Huayaney, S. Nease, E. Chicca, Learning in silicon beyond STDP: a neuromorphic implementation of multi-factor synaptic plasticity with calcium-based dynamics, IEEE Trans. Circuits Syst. I: Regul. Pap. 63 (12) (2016) 2189–2199. Available from: https://doi.org/10.1109/tcsi.2016.2616169. dec.

[95] Z. Wang, S. Joshi, S. Savelev, W. Song, R. Midya, Y. Li, et al., Fully memristive neural networks for pattern classification with unsupervised learning, Nat. Electron. 1 (2) (2018) 137.

Index

Note: Page numbers followed by "*f*" and "*t*" refer to figures and tables, respectively.

C

D

E

F

G

H

I

J

K

L

M

O

P

Q

R

S

T

U

V

W

Y

Z

WOODHEAD PUBLISHING SERIES IN ELECTRONIC AND OPTICAL MATERIALS

Memristive Devices in Brain-Inspired Computing: From Materials, Devices and Circuits to Applications - Computational Memory, Deep Learning, and Spiking Neural Networks reviews the latest developments in materials and devices engineering for optimizing memristive devices beyond storage applications and towards brain-inspired computing. The book provides readers with an understanding of four key areas: material and device aspects; algorithmic aspects comprising basic concepts of neuroscience as well as various computing concepts such as in-memory computing; the circuits and architectures implementing those algorithms; and target applications such as deep learning and spiking neural networks.

This comprehensive book could serve as a valuable resource for an interdisciplinary audience, including materials scientists, physicists, electrical engineers and computer scientists.

Key features

- Provides readers with an overview of four key concepts in this emerging research topic, namely, material and device aspects, algorithmic aspects, circuits and architectures, and target application
- Covers a broad range of applications, including computational memory, deep learning and spiking neural networks
- Provides a unique interdisciplinary look at the field by including perspectives from a wide range of disciplines, including materials science, electrical engineering and computing

About the Editors

Sabina Spiga is a Senior Researcher at the Institute for Microelectronics and Microsystems (IMM), National Research Council (CNR), Agrate Brianza, Italy. She received her degree in Physics in 1995 from Università di Bologna and her Ph.D. in Material Science from the Università di Milano in 2002.

Abu Sebastian is a Principal Research Staff Member and technical manager at IBM Research – Zurich. He received a B. E. (Hons.) degree in Electrical and Electronics Engineering from BITS Pilani in 1998, and his M.S. and Ph.D. degrees in Electrical Engineering from Iowa State University in 1999 and 2004, respectively.

Damien Querlioz is a CNRS Researcher at the Centre for Nanoscience and Nanotechnology of Université Paris-Saclay. He completed his predoctoral education from Ecole Normale Supérieure, Paris, and received his Ph.D. from Université Paris-Sud in 2009.

Bipin Rajendran is a Reader in Engineering at King's College London, UK. He received a B.Tech degree from I.I.T. Kharagpur in 2000, and his M.S. and Ph.D. degrees in Electrical Engineering from Stanford University in 2003 and 2006, respectively.

TECHNOLOGY & ENGINEERING / Electronics / General / Material Science

ISBN 978-0-08-102782-0

9 780081 027820

ELSEVIER

WP WOODHEAD PUBLISHING

An imprint of Elsevier
elsevier.com/books-and-journals