Hao Yu · Yuhao Wang

Design Exploration of Emerging Nano-scale Non-volatile Memory

Springer

Design Exploration of Emerging Nano-scale Non-volatile Memory

Hao Yu • Yuhao Wang

Design Exploration of Emerging Nano-scale Non-volatile Memory

Springer

Hao Yu
School of EEE
Nanyang Technological University
Singapore, Singapore

Yuhao Wang
School of EEE
Nanyang Technological University
Singapore, Singapore

ISBN 978-1-4939-0550-8 ISBN 978-1-4939-0551-5 (eBook)
DOI 10.1007/978-1-4939-0551-5
Springer New York Heidelberg Dordrecht London

Library of Congress Control Number: 2014935240

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

The analysis of big data at exascale (10^{18} bytes or flops) has introduced the emerging need to reexamine the existing hardware platform that can support intensive data-oriented computing. A big-data-driven application requires huge bandwidth with maintained low-power density. For example, web-searching application involves crawling, comparing, ranking, and paging of billions of web pages with extensive memory access. At the same time, the analysis of such a huge data at exascale is a national interest due to cybersecurity need. One needs to provide scalable big-data storage and processing solution that can detect malicious attack from the sea of data, which is beyond the capability of a pure software-based data analytic solution. The key bottleneck is from the current data storage and processing hardware, which has not only the well-known memory wall and power wall with limited accessing bandwidth but also large leakage power at advanced CMOS technology nodes. One needs to design an energy-efficient hardware platform for future big-data storage that can also support data-intensive processing for data recognition in both image and security applications.

Memory is any physical device that is able to temporarily or permanently hold the state of information. Memories can be generally classified into two categories: volatile and nonvolatile. Static and dynamic random-access memories (SRAM and DRAM) are examples of volatile memories that can be accessed in nanosecond of speed, but the stored data will be lost when powered off. The flash and hard disk drive (HDD) are examples of nonvolatile memories. Imagine the life where one can start a computer in the blink of an eye, without having to wait for the operation system to load, or transfer full-length high-definition movie by memory stick in seconds rather than hours. Such a life would happen if a universal nonvolatile memory could be developed that not only can retain information without external power but also can be accessed in high speed. In general, the following criteria examine the new memory technologies: (1) Scalability for high-density integration (2) Low energy consumption for mobile access (3) High endurance capable of 10^{12} writing/erasing cycles

The existing memory technologies have critical challenges of scaling at nanoscale due to process variation, leakage current, and I/O access limitations.

Recently, there are two research trends that attempted to alleviate the memory-wall and power-wall issues for future big-data storage and processing system. Firstly, the emerging nonvolatile memory technologies such as the resistive RAM (ReRAM), spin-transfer torque RAM (STT-RAM), and domain-wall nanowire racetrack memory have shown significantly reduced standby power and increased integration density, as well as close to DRAM/SRAM access speed. Therefore, they are considered as promising candidates of universal memory for big-data applications. Secondly, due to high data-level parallelism in big-data applications, a large number of application-specific accelerators can be deployed for data processing. However, such a memory–logic integration approach will still incur I/O overhead. Instead, an in-memory-based domain-specific computation will be highly desired with less dependence on I/Os.

In order to achieve low power and high throughput (or energy efficiency) in big-data computing, one can build an in-memory nonvolatile memory (NVM) hardware platform, where both the memory and computing resources are based on NVM devices with instant switch-on as well as ultralow leakage current. This can result in significant power reduction due to the nonvolatility. Moreover, one can develop NVM logic accelerator that can perform domain-specific computation such as machine learning in a logic-in-memory fashion. In contrast, for the conventional memory–logic integration architecture, the storage data must be loaded into the volatile main memory, processed by logic, and written back afterwards with significant I/O communication overhead.

In this book, we plan the following research studies in this regard. Firstly, we introduce a nonvolatile big-data storage design platform that can evaluate both the current NVM technology and future NVM technology. We develop a SPICE-like simulator NVM SPICE, which implements physical models for nonvolatile devices in a similar way as the BSIM model for MOSFET. We further develop an advanced NVM design platform that can provide evaluation of various memory cell structures as well as corresponding readout circuits. As such, one can perform an accurate and efficient estimation of memory performance at microarchitecture level. Secondly, we study in-memory NVM computing architecture for domain-specific big-data storage and processing. We illustrate the NVM-based basic memory and logic components and find significant power reduction. We further illustrate in-memory machine learning such as extreme learning machine (ELM) for big-data image recognition as well as security classification, which can be evaluated based on both developed NVM design platforms.

This book provides a state-of-the-art summary for the latest literature on emerging nonvolatile memory technologies and covers the entire design flow from device, circuit, to system perspectives, which is organized into five chapters. Chapter 1 covers the basics of memory and review of existing memory technologies and emerging nonvolatile memory technologies. Chapter 2 introduces the physics of the emerging nonvolatile memory as well as the agreeing computing architecture. Chapter 2 details the device characterization for the emerging nonvolatile memory by nonelectrical states. Chapter 4 explores the circuit level design techniques for the emerging nonvolatile memory. Chapter 5 presents the system-level architectures

with applications for the emerging nonvolatile memory. This book assumes that readers have basic knowledge of semiconductor device physics. This book will be a good reference for senior undergraduate and graduate students who are performing researches on nonvolatile memory technologies.

Finally, the authors would like to thank their colleagues at CMOS Emerging Technology Group at Nanyang Technological University: Wei Fei, Yang Shang, Xiwei Huang, Chun Zhang, Sai Manoj Pudukotai Dinakarrao, and Shuai Chen. The authors also owe their grateful discussion to Prof. Roy Kaushik, Prof. Dennis Sylvester, Prof. Kevin Cao, Prof. Weisheng Zhao, Prof. Yuan Xie, Prof. Yiran Chen, Prof. Hai Li, Dr. Tanay Karnik, Dr. Jing Li, Prof. Wei Zhang, Prof. Tony Kim, Prof. Wen-Siang Lew, Prof. Chip-hong Chang, and Prof. Kiat-Seng Yeo. Their support is invaluable to us during the writing of this book. The relevant research is funded by MOE Tier-2 (MOE2010-T2-2-037), A*STAR PSF (1120120 2015), and NRF CRP (NRF-CRP9-2011-01) from Singapore as well as industry research collaboration fund from Huawei Shannon Lab.

Singapore — Hao Yu
Singapore — Yuhao Wang
January 1, 2014

Contents

Chapter 1
Introduction

Abstract Computer memory is any physical device capable of storing data temporarily or permanently. It covers from the fastest yet most expensive static random-access memory to the cheapest but slowest hard drive disk, while in between there are many other memory technologies that make trade-offs among cost, speed, and power consumption. The variety of memory technologies introduces the most important concept as memory hierarchy, which exploits the strength and avoids the weakness of different memory technologies. However, the memory hierarchy is only the temporary solution to alleviate the memory-wall issue, and the ultimate solution requires a breakthrough on memory technology. Fortunately, many newly introduced emerging nonvolatile memory technologies have exhibited great potential for the future universal memory. This chapter reviews the existing semiconductor memory technologies as well as the emerging nonvolatile memory technologies.

Keywords Memory technology • Static random-access memory • Dynamic random-access memory • Non-Volatile memory

1.1 Memory Design

Before introducing specific memory technologies, it is important to understand the basic electronic components of which the memory is made up of. A memory chip consists of millions to billions of memory cells and takes binary address as input and finds target cells correspondingly, so that read and write operations can be performed. In order to efficiently perform this, memory cells are organized in certain fashion, as illustrated in Fig. 1.1.

Enormous data cells are divided into multiple data arrays, which are connected by the H-tree network. The input address is logically divided into two parts and will be interpreted respectively. The first part of the address indicates the position of the data array, in which the target cells are kept. The second part of the address reveals

H. Yu and Y. Wang, *Design Exploration of Emerging Nano-scale Non-volatile Memory*,
DOI 10.1007/978-1-4939-0551-5_1,

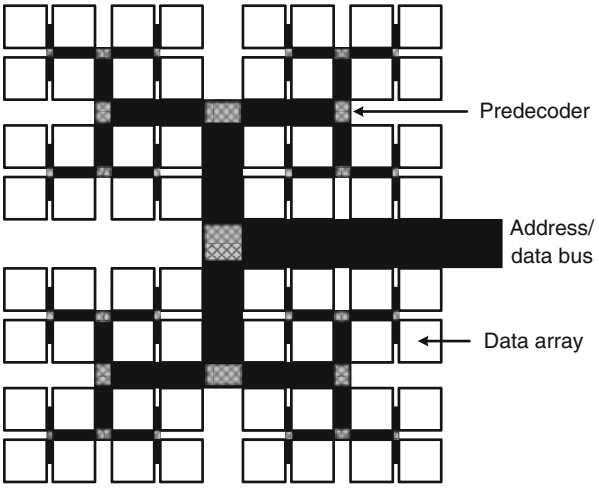


Fig. 1.1 Memory organization in H-tree network

the position of the target cells inside the data array. The data array identifier will be used by the predecoders along the H-tree network paths, to route electrical signals to target the data array. The most noticeable advantage of H-tree network is that it can ensure an identical signal propagating distance to every data array. This is important to ensure the system is deterministic and the access latency is fixed.

The storage unit in the memory is the data array, whose structure is shown in Fig. 1.2. All memory cells lie at the cross-points of the word-lines and bit-lines. Word-lines and bit-lines have metal wires that propagate signals and will incur certain wire delay due to its parasitic wire resistance and capacitance; therefore, the larger the data array is, the longer access latency can be expected. Each cell stores one bit of information, represented by high or low logic, and its value can be read out by sense amplifiers. If every single cell is directly connected with outside I/O, billions of I/Os will be required which is practically impossible to achieve; therefore the decoders are used to take binary address to operate on designated cells.

A decoder converts binary address from n input lines to a maximum of 2^n unique output lines. Figure 1.3 shows the circuit of a 2–4 decoder with its truth table. In the memory array, the output lines of word-line decoders are connected to the word-lines, which enable an entire row of data array specified by the address. Because the electrical signals are bidirectional on the bit-line, that is, the bit-line can drive the cell in the write operation and the cell can drive the bit-line in the read operation, the bit-line decoder output lines are connected to the multiplexer, which selectively allows electrical signal of a specific column to pass.

The objective of the readout circuit is to distinguish the bistable states of the memory cells. For conventional memory technologies like static random-access memory (SRAM) and dynamic random-access memory (DRAM), the memory state is represented by electrical voltage: V_{DD} for logic 1 and GND for logic 0. Therefore, a readout circuit can be designed as a voltage comparator that compares

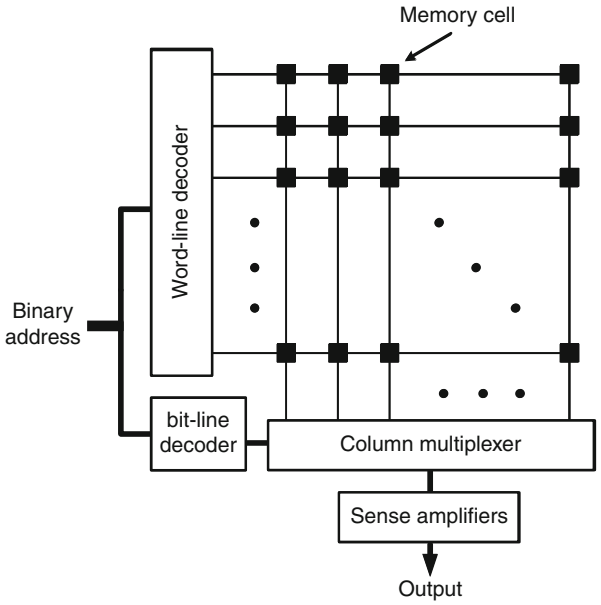
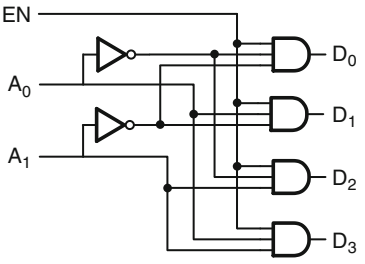


Fig. 1.2 The structure of memory array



EN	A_0	A_1	D_0	D_1	D_2	D_3
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1
0	X	X	0	0	0	0

Fig. 1.3 Two–four decoder logic and truth table

state voltage with intermediate voltage, namely $V_{DD}/2$. However in practice, due to the charge sharing between the memory cell and bit-line parasitic capacitor, the detectable voltage margin is reduced to less than one-tenth of previous value, and more sensitive readout circuit is required. Figure 1.4 shows the circuit of a voltage mode typical latch-type sense amplifier, which is able to detect voltage difference smaller than 100 mV at nanosecond scale. Its working mechanism can be described as follows. At first, the BLP and BLN are precharged at $V_{DD}/2$ and EN is kept low. When the readout is performed, the BLP voltage will increase or decrease slightly (ΔV) due to charge share. If differential structure is used, the BLN will change $-\Delta V$. Note that the ΔV is determined by the ratio of memory cell capacitance and bit-line capacitance, and when the array is too large, ΔV will

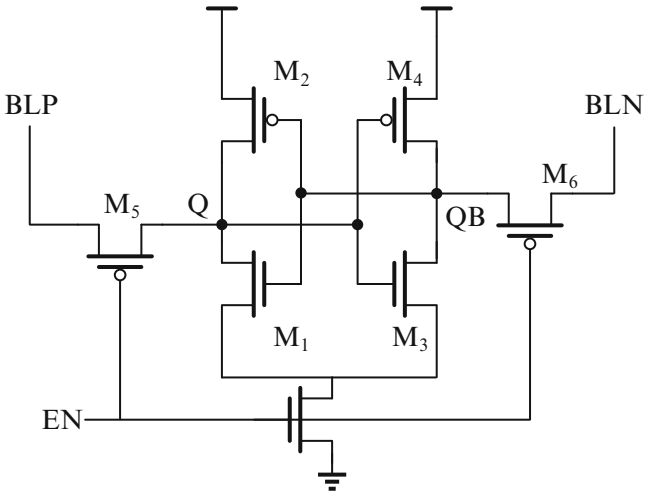


Fig. 1.4 Latch-type sense amplifier

become too small to detect. This is the major reason why the memory cannot be made into one single array and the division is required. After that, the sense amplifier is enabled by set EN signal high. This isolates the BLP and BLN with Q and QB, and Q and QB will be compared by the latch. The latch compares voltages in a positive feedback fashion. Assume there is a ΔV_Q, and as the input of the inverter at the right-hand side ($M3$ and $M4$) it will lower the voltage of V_{QB}, as shown in the transfer curve of inverter. And as the input of the inverter at the left-hand side ($M1$ and $M2$), the decrease in V_{QB} will in turn increase the value of ΔV_Q. As such, the two cross-coupled inverters reinforce each other and enter a positive feedback loop until they reach the final stable state in which the ΔV_Q is V_{DD} and V_{QB} is 0. Without the pass transistor $M5$ and $M6$, the latch will have to drive the entire bit-line, which greatly affects the convergence speed and incurs more energy consumption.

1.2 Traditional Semiconductor Memories

1.2.1 Overview

Semiconductor memories refer to the silicon-transistor-based physical devices in computer systems that are used to temporarily or permanently store programs or data. In contrast to storage media like hard disks and optical disc, of which the accesses have to follow a predetermined order due to the mechanical drive limitations, semiconductor memories possess the property of random access, which means that the time it takes to access any data is identical regardless of the data location.

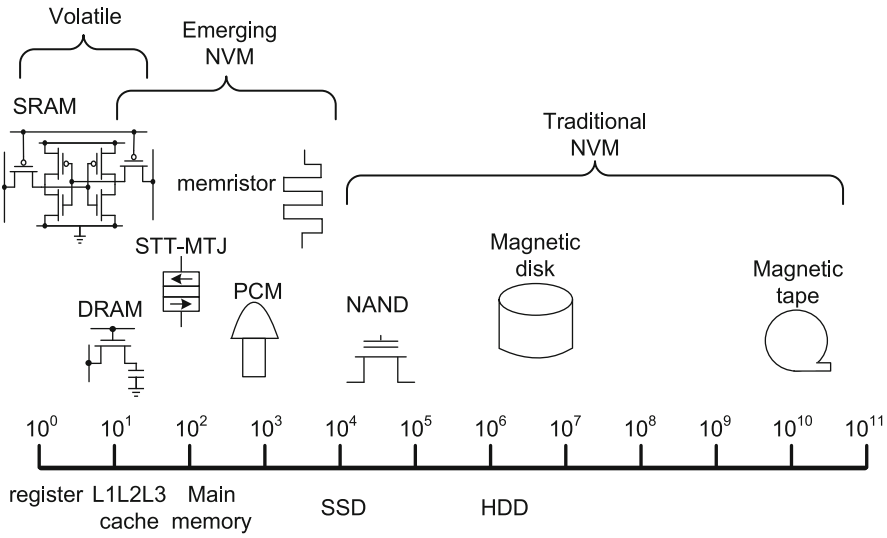


Fig. 1.5 Typical memory hierarchy of a computer system (latency measured in processor cycles)

According to volatility, the semiconductor memories can be further classified into volatile memory and nonvolatile memory. Volatile memory stores data as electrical signals and loses its data when the chip is powered off. Currently, the most commonly used volatile memories are SRAM and DRAM, whose data is indicated by the electrical voltage levels. On the contrary, nonvolatile memory is able to retain its data even when the chip is turned off, as its data is mostly preserved by nonelectrical states. For instance, bits in programmable read-only memory (PROM) are denoted by whether the fuses of individual memory cells are burned. In this part, the principles and operations of volatile SRAM/DRAM and nonvolatile flash memory will be briefly introduced (Fig. 1.5).

1.2.1.1 Static Random-Access Memory

A typical CMOS SRAM cell consists of six transistors, as shown in Fig. 1.6. The flip-flop formed by $M1$–$M4$ holds the stored bit. The term *static* is derived from the fact that the cell does not need to be refreshed like dynamic RAM and the data can be retained as long as the power is supplied. $M5$ and $M6$, connected with the word-line and two bit-lines, are used as access transistors to select target cells.

There are three operation states for a SRAM cell: write, read, and standby. To a write operation, the value to be written needs to be applied on both bit-lines, namely, BL and $\overline{\text{BL}}$, in a complementary manner. Assume we wish to write "0" to the cell, i.e., Q to be "0" and $\overline{Q}$ to be "1"; the BL is driven low and $\overline{\text{BL}}$ high. Once $M5$ and $M6$ are turned on by setting WL "1," the bit-line drivers will override the previous

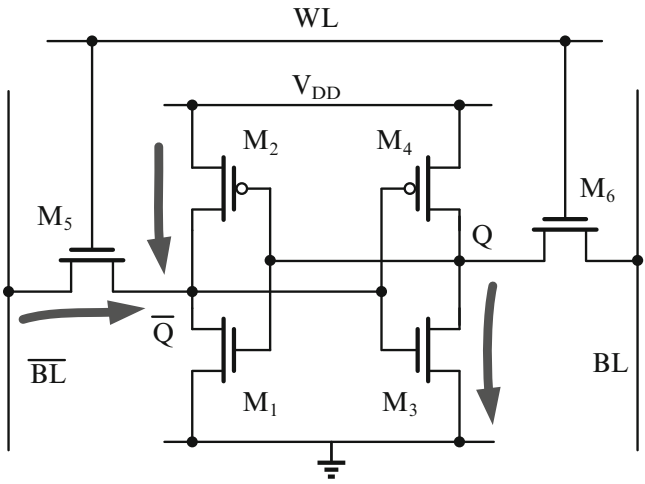


Fig. 1.6 A 6T SRAM cell structure with leakage paths in standby state. The bit-lines are precharged high and assume the stored data is "1" at Q

stored value. In order to easily override the previous state in the self-reinforced flip-flop, the bit-line drivers are required to be designed stronger than the transistors in flip-flop.

For a read operation, both bit-lines are precharged high before the start of the read cycle, and the turning on of the word-line signifies the start of the read operation. Because of the opposite voltages at Q and $\overline{Q}$, one of the bit-lines will be pulled down by the cell, and the discharging of one of the bit-lines is then detected by the bit-line sense amplifier. In voltage mode sensing scheme, the sign of bit-line voltage difference ΔV (V_{BL} minus $V_{\overline{\mathrm{BL}}}$) determines the value of stored bit. A ΔV in tens of millivolts is significant enough to efficiently distinguish which bit-line is being discharged. For example, assume the stored bit is "1" at Q and "0" at $\overline{Q}$, and once the WL is asserted, the $\overline{\mathrm{BL}}$ will discharge towards "0"; when a positive ΔV of tens of millivolts is gained, the latch-based sense amplifier will amplify the small voltage difference with positive feedback and finally output logic "1" as result.

When the word-line (WL) is connected to ground, turning off the two access transistors, the cell is in the standby state. During the standby state, the two cross-coupled inverters will reinforce each other due to the positive feedback, the value is preserved as long as the power is supplied. One prominent problem regarding SRAM in standby state is severe subthreshold leakage. Subthreshold leakage is the drain–source current of a transistor when the gate–source voltage is less than the threshold voltage. The subthreshold current depends exponentially on threshold voltage, which results in large subthreshold current in deep-submicron regime. Figure 1.6 shows three leakage paths in one SRAM cell, assuming the stored bit is "1" at Q. Note that the BL and $\overline{\mathrm{BL}}$ are always precharged to V_{DD} to facilitate future read operation. Regardless of the stored value, there always will be three transistors consuming leakage power.

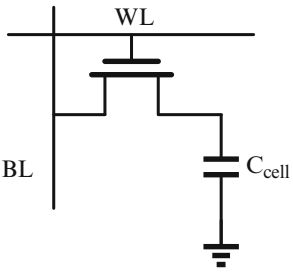


Fig. 1.7 The circuit diagram of 1T1C DRAM cell structure

Compared to other memory technologies, SRAM is able to provide fastest access speed, but the advantage comes as a tradeoff against density and power. As one SRAM cell requires silicon area for six transistors, SRAM has very limited density and hence is more expensive than other memory technologies. In addition, it is very power consuming due to the leakage problem at standby state. Therefore, SRAM serves best in applications where high performance is the main concern and the capacity is not significant, namely the caches for processors.

1.2.1.2 Dynamic Random-Access Memory

The philosophy behind DRAM is simplicity. Unlike SRAM where one cell is composed of six transistors, each individual DRAM cell consists of only one capacitor and one access transistor. The data "0" or "1' is represented by whether the capacitor is fully charged or discharged. However, the electrical charge on the capacitor will gradually leak away, and after a period of time, the voltage on the capacitor is so low for the sense amplifier to differentiate between "1" and "0." Therefore, unlike SRAM in which the data can be retained as long as the power is supplied, the retention time for DRAM is finite and all DRAM cells need to be read out and written back periodically to ensure data integrity. Typically, the cells are refreshed once every 32 ms or 64 ms. This process is known as *refresh*, and this is how the name of dynamic RAM is derived. Figure 1.7 shows the circuit diagram of such 1T1C structure DRAM cell.

To write a DRAM cell, the bit-line is first set high or low based on the value to write. After the access transistor is turned on by the asserting word-line, the capacitor in the selected cell is charged to "1" or discharged to "0." Because the access takes place through an NMOS transistor, there exists a V_{th} drop during the write "1." In order to prevent this V_{th} drop and maintain a long refresh period, the word-line driving voltage is usually boosted to $V_{PP} = V_{DD} + V_{\text{th}}$.

To read a DRAM cell, the bit-line is precharged to $V_{DD}/2$ and then the word-line is enabled. Due to the charge sharing, the bit-line voltage will slightly decrease or increase depending on the voltage that the capacitor was previously charged to, i.e., V_{DD} or 0. If it is previously charged, the charge sharing will slightly boost the bit-line voltage; otherwise, some charge will be distributed from bit-line to cell

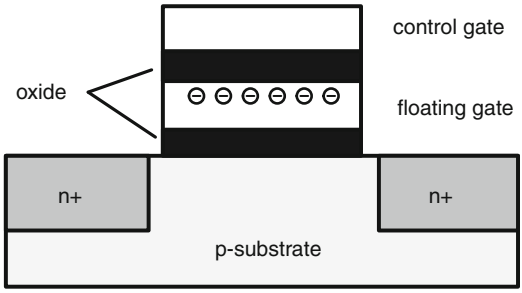


Fig. 1.8 The cross section of a floating gate transistor

capacitor. In both cases, the voltage of the storage capacitor will be changed after the read operation; thus the read operation is called destructive, and an instant write back is required. A slight voltage change on the bit-line can be calculated by

$$\Delta V = \pm \frac{V_{DD}}{2} \frac{C_{\text{cell}}}{C_{\text{cell}} + C_{\text{bitline}}}. \tag{1.1}$$

The sign of ΔV depends on the state of the storage capacitor. In modern DRAM devices, the capacitance of a storage capacitor is far smaller than the capacitance of the bit-line. Typically, the capacitance of a storage capacitor is one-tenth of the capacitance of the long bit-line that is connected to hundreds or thousands of other cells. The relative capacitance values create the scenario that when the small charge contained in a storage capacitor is placed on the bit-line, the resulting voltage on the bit-line is small and difficult to measure in an absolute sense. In DRAM devices, the voltage sensing problem is resolved through the use of a differential sense amplifier that compares the voltage of the bit-line to a reference voltage.

The use of differential sense amplifier, in turn, introduces some requirements on the DRAM array structure. Particularly, instead of a single bit-line, a pair of bit-lines needs to be used to sense the voltage value contained in any DRAM cell. In addition, in order to ensure that the voltage and capacitance values on the pair of bit-lines are closely matched, the bit-lines must be closely matched in terms of path lengths and the number of cells attached. The above requirements lead to two distinctly different array structures: open bit-line structures and folded bit-line structures.

1.2.1.3 Flash Nonvolatile Memory

Flash memory is the most widely used nonvolatile memory technology today. The key device in this prevailing memory is floating gate transistors. A figure of cross section of a floating gate transistor is shown in Fig. 1.8. Unlike a MOSFET transistor, an additional floating gate is added between the control gate and channel. Isolated by oxide layers, floating gate is able to trap charges and keep them for years. Therefore, the FG-transistor encodes data based on whether electrons are trapped and is able to retain data without power. That is where "nonvolatile" is derived from.

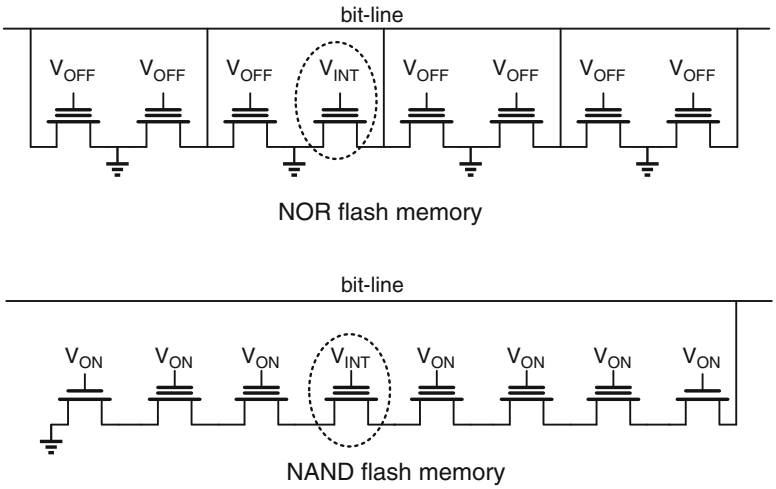


Fig. 1.9 Two common layouts for flash memory: NOR flash memory and NAND flash memory

The principle of read operation can be described as follows. When no charges are trapped in floating gate, the FG-transistor has a threshold voltage of V_{th0}; when negative charges are trapped, they attract positive charges of control gate, thus higher control gate voltage is required to turn on the channel,which produces a higher threshold voltage V_{th1}. By applying intermediate control gate voltage V_{th0} and V_{th1} and measuring current, the state of device can be known.

The write operation of FG-transistor involves injecting or pulling electrons across the oxide barrier. There are two ways to achieve this, quantum tunneling and hot electron injection. In quantum tunneling scenario, high voltage is applied on control gate, quantum tunneling will take place between the floating gate and channel, and electrons can travel across the oxide barrier. For hot electron injection scenario, electrons are accelerated under high electrical field in the channel till its energy is high enough to penetrate the oxide layer. Note that electrons with high energy will damage the oxide lattice, and such damage will accumulate and lead to a limited write cycles, which is typically around 10^5 cycles.

There are two common layouts for flash memory, as shown in Fig. 1.9, NAND flash memory with FG-transistors in series and NOR flash memory with FG-transistors in parallel. The names NAND and NOR are derived from the fact that their connection fashion in series or parallel resembles a NAND gate or NOR gate. NAND layout has the density advantage over NOR layout because each row only has one ground connection and thus is widely used for external storage. NOR layout has lower latency and thus is widely used in embedded systems, where high performance is required. Figure 1.9 also shows how the read of NAND and NOR flash memory can be achieved. The relationship between the different applied voltage magnitudes is shown as follows:

$$V_{\text{OFF}} < V_{th0} < V_{\text{INT}} < V_{th1} < V_{\text{ON}} \ll |V_{\text{HIGH}}|. \tag{1.2}$$

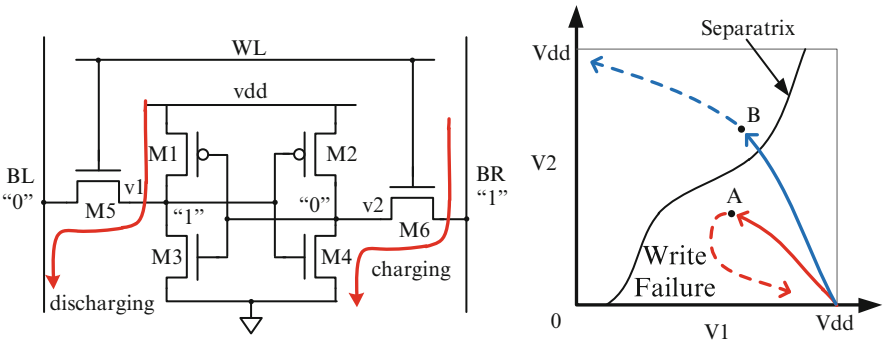


Fig. 1.10 Illustration of SRAM write failure

1.2.2 Nanoscale Limitations

As the technology scaling advances into nanoscale, the conventional memory design faces certain limitations. In the regime where the classic physics still rules, such limitations mainly come from two major aspects. Firstly, due to process variation, the mismatch among transistors may lead to functional failures. Secondly, positive feedback loop between leakage power and heat may result in thermal runaway failure. In this part, we review the physical mechanisms of such failures induced in nanoscale designs, including write, read, and hold failures, as well as the thermal runaway failure. For the simplification of illustration, only variation of threshold voltage is considered for the functional failures.

1.2.2.1 Functional Failures by Variation

Write Failure

Write failure is defined as the inability to write data properly into the SRAM cell. During the write operation, both access transistors should be strong enough to pull down or pull up the voltage level at internal nodes. As shown in Fig. 1.10, the write operation can be described on the variable plane as the process of pulling the operating point from initial state (bottom-right corner) to the target state (top-left corner). The crossing line named *separatrix* divides the variable plane into two convergent regions. Given enough time, the operating point in any region will converge to the nearest stable equilibrium state at either the top-left or bottom-right corner. The write operation is aimed at pulling the operating point into the targeted convergent region such that the operating point can converge to the closest equilibrium state after the operation finishes, which is shown by point B in Fig. 1.10.

However, an increase in threshold voltage due to variation can reduce the transistor driving strength and vice versa for a decrease in threshold. The increase

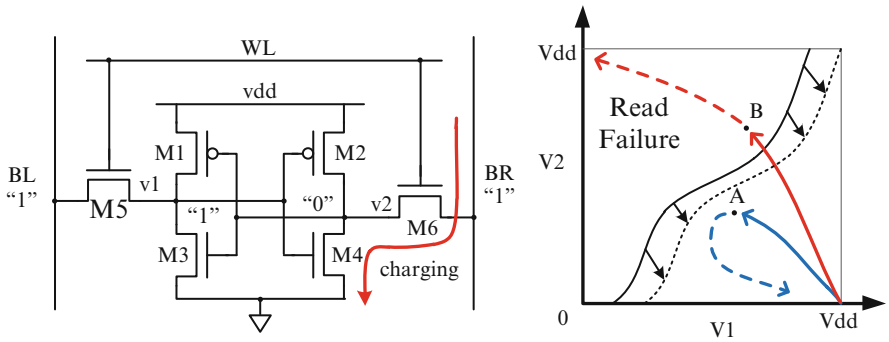


Fig. 1.11 Illustration of SRAM read failure

of V_{th} in $M6$ along with the decrease of V_{th} in $M4$ can result in difficulty in pulling down v_2. On the variable plane, it becomes more difficult for the operating point to move towards the target state. If the operating point cannot cross the *separatrix* before access transistors are turned off, it goes back to the initial state, which means a write failure.

Read Failure

Read failure refers to the loss of the previously stored data. Before the read operation is performed, both BR and BL are precharged to v_{dd}. Suppose the previous internal states in SRAM are $v_1 = v_{dd}$ and $v_2 = 0$; electric charge on BR is discharged through $M6$ and $M4$ while that on BL remains the same. As such, a small voltage difference between BR and BL is generated which will be detected and amplified. In this way, data stored in the SRAM can be read. Note that access transistors need careful sizing such that their pull-up strength is not strong enough to pull the stored "0" to "1" during the read operation.

On the variable plane, the operating point is inevitably perturbed and pulled towards the *separatrix*. If the read operation does not last too long, access transistors can be shut down before the operating point crosses the *separatrix*. As such, the operating point returns to the initial state in the end, as point A in Fig. 1.11, which means a read success.

Even though all the sizing is carefully taken, threshold variations may still result in read failure. For example, variation caused by mismatch between $M4$ and $M6$ may result in unbalanced pulling strength, and v_2 can be pulled up more quickly. As a result, the operating point crosses the *separatrix* before the read operation ends, as point B in Fig. 1.11.

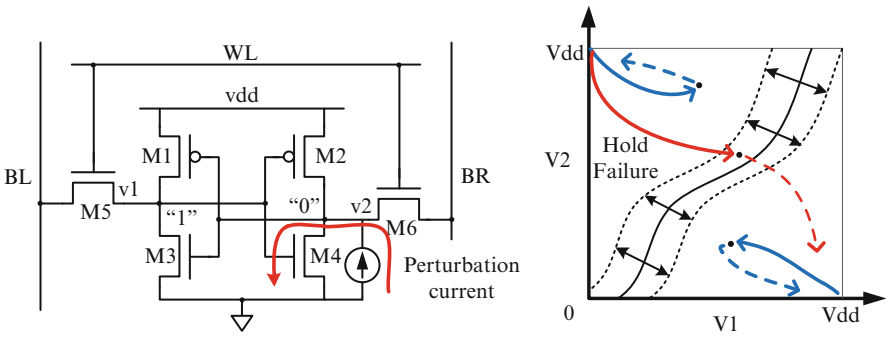


Fig. 1.12 Illustration of SRAM hold failure

Hold Failure

Hold failure happens when the SRAM fails to retain the stored data. It can be caused by external noise or single event upset (SEU). The external perturbation can be modeled as noise current injected into SRAM. Similar to the read operation, the operating point is expected to converge back to the initial state after settling down from disturbance. Otherwise, it will cross to the other convergent region.

While access transistors have no impact on the retention of SRAM data, $M1$–$M4$ together can determine the likelihood of hold failure by finding the position of the *separatrix* and thus threshold variation may cause failure by perturbing the *separatrix* as shown in Fig. 1.12. A such, one needs to verify if the SRAM is still tolerable to the injected noise in the presence of threshold voltage variation.

1.2.2.2 Functional Failure by Thermal Runaway

Thermal runaway [16, 22] refers to a situation where an increase in temperature changes the conditions in a way that causes a further increase in temperature in positive feedback. In memory system, it is associated with the electrical–thermal coupling between leakage power and temperature. Note that leakage current in memory can be modeled by

$$I_{\text{leakage}} = \underbrace{A_s \cdot \frac{W_d}{L_d} \cdot v_T^2 \left(1 - e^{\frac{-V_{DS}}{v_T}}\right) \cdot e^{\frac{V_{GS}-V_{\text{th}}}{ns \cdot v_T}}}_{I_{\text{subthreshold}}} + \underbrace{W_d \cdot L_d \cdot A_J \left(\frac{T_{\text{oxr}}}{T_{\text{ox}}}\right)^{nt} \frac{V_g \cdot V_{\text{aux}}}{T_{\text{ox}}^2} e^{-B_J T_{\text{ox}}(a-b|V_{\text{ox}}|)(1+c|V_{\text{ox}}|)}}_{I_{\text{gate\_leakage}}}, \quad (1.3)$$

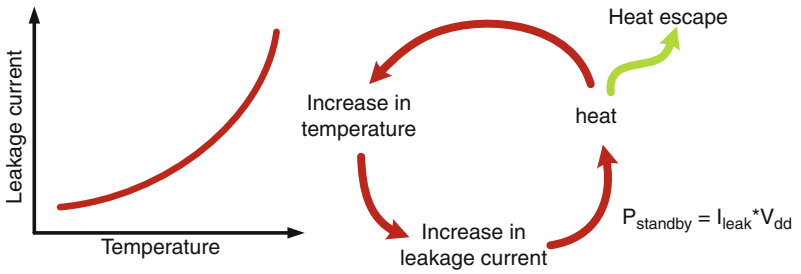


Fig. 1.13 Illustration of SRAM thermal runaway failure by positive feedback between temperature and leakage power

where

- $v_T = \frac{k \cdot T}{q}$ is the thermal voltage,
- V_{DD} is the supply voltage with ΔV swing,
- L_d and W_d are the effective device channel length and width,
- ns is the subthreshold swing coefficient,
- A_s, A_J, B_J, a, b, and c are technology-dependent constants,
- nt is a fitting parameter,
- T_{ox} and T_{oxr} are gate dielectric and reference oxide thickness, respectively, and
- V_{aux} is auxiliary temperature-dependent function for density of tunneling carriers.

As the technology node scales down, the controlling ability of transistors becomes weaker, and hence larger leakage current will be experienced. As such, thermal runaway becomes a prominent limitation for large-scale big-data memory integration in advanced technology nodes. The course of potential thermal runaway is illustrated in Fig. 1.13. At the very beginning, memory works at room temperature with moderate leakage current, which will consistently produce heat. If the thermal source grows much faster than the heat removal ability of heat sink, there will be thermal accumulation that leads to temperature increase of the memory chip. Due to the exponential relationship between temperature and leakage current, the increase of memory temperature will in turn provoke larger leakage current, which in turn increases the leakage current. Such uncontrolled positive feedback will continue and finally lead to destructive high temperature, melting silicon and permanently damaging the memory cells.

Thermal runaway temperature $T_{\text{threshold}}$ is temperature at which thermal runaway failure happens. When the temperature goes beyond $T_{\text{threshold}}$, the system temperature will rise rapidly with resulting thermal runaway. As shown in Fig. 1.14b, the $T_{\text{threshold}}$ represents the maximum heat removal ability of the heat sink beyond which thermal runaway failure happens. To avoid thermal runaway, we can either maintain a low thermal source or improve the thermal removal ability. To reduce the thermal source, low-power techniques such as power gating or more sophisticated memory cell structures can be applied. For thermal removal, close heat removal path to heat sink is favored.

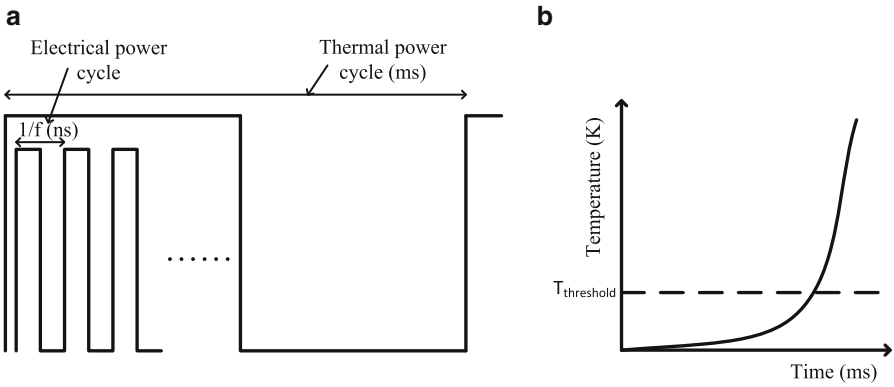


Fig. 1.14 (**a**) Thermal power and electrical power cycles. (**b**) Thermal runaway and threshold temperature

1.3 Recent Nanoscale Nonvolatile Memories

1.3.1 Overview

All well-established memory technologies introduced above have certain limitations. SRAM is the fastest memory technology currently available, but significant leakage power is experienced to retain the stored data, which gets worse when scaled down to deep-submicron regime. In addition, its capacity is limited to a few megabytes due to its high cost led by an area-consuming 6-transistor structure. DRAM is second to SRAM in terms of speed, but this capacitor-based memory needs to be refreshed periodically, which produces large power consumption for DRAM in large capacity. Flash memory overcomes the power issue by its nonvolatility, but has slower speed and very limited endurance; thus it mainly serves as storage for data that does not need to be frequently accessed.

In order to exploit the strengths and avoid the drawbacks of different memories, the memory hierarchy has been formed based on the roles the memories play (Fig. 1.15). Generally, memories at the bottom of the pyramid (Fig. 1.16) tend to be slow, less costly, and nonvolatile; and memories at the top of the pyramid tend to be fast, expensive, and power hungry. Such memory hierarchy has been proven very efficient for decades, but by no means the final solution for memory system design. The challenge that the current memory system faces is the well-known memory-wall issue, meaning that the memory performance accelerates at a much lower rate than that of processors. Therefore, memory usually becomes the bottleneck of the overall computing system. The inefficiency and complexity of data transmitting vertically among different levels of memory hierarchy are the de facto hindrance to bridge the performance gap between the memory and processor. For example, the retrieve latency of HDD is several orders longer than that of main memory, and in cases where the data to be executed has not yet been loaded into the main memory, there will be a big performance loss incurred for the processor waiting.

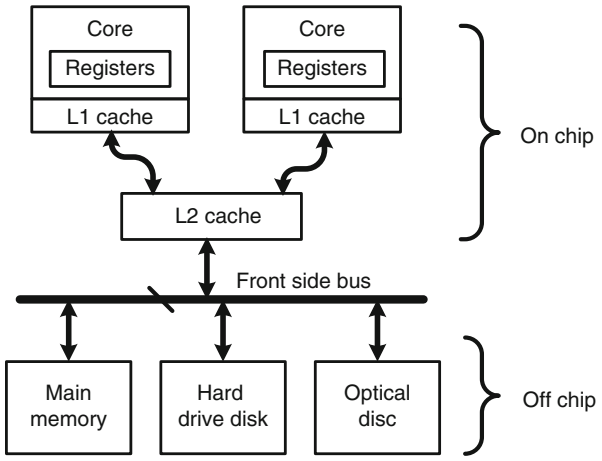


Fig. 1.15 Memory architecture in a typical system

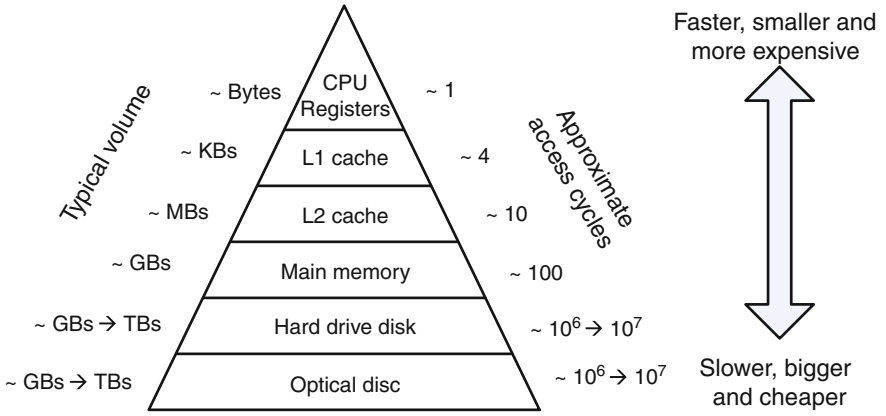


Fig. 1.16 Memory hierarchy with typical volume and access cycles

Ideal memory needs to possess the properties of small cell area and great scalability so that low cost can be achieved, nonvolatility so that low power will be consumed, and low access latency so that performance can be guaranteed. If such candidate exists, the memory hierarchy can be flattened or even eliminated. All the data can be retained without powering, energy will be consumed only when data is accessed, and required data can be directly fetched for computation almost instantly without processor pausing. Towards this end, great research effort has been done to find the potential universal memory candidate. Fortunately, featured with fast access speed, high density, and zero standby power, the emerging nonvolatile memories at nanoscale such as spin-transfer torque memory [14], phase-change memory [38], conductive-bridge random-access memory (CBRAM) [15], racetrack memory [20],

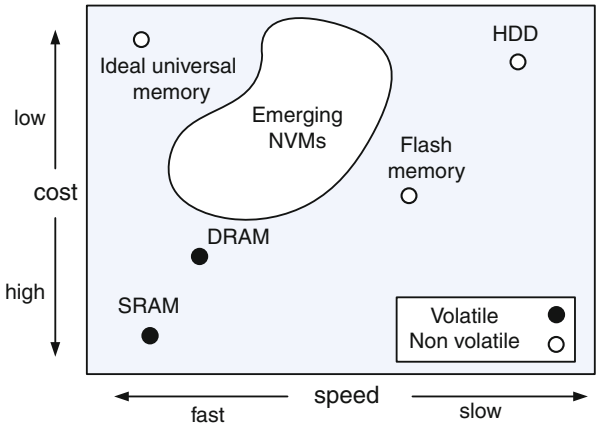


Fig. 1.17 Current status of researches on emerging nonvolatile memories towards ideal universal memory

and memristor [26] have introduced a promising future for the universal memory. Some of them, like racetrack memory and CBRAM, may still be in their infancy and only have small-scale demonstration [5, 20], while spin-transfer torque RAM (STT-RAM) and phase-change memory are already commercialized and available in the market (Fig. 1.17).

1.3.1.1 Resistive Random-Access Memory

Memristor

The existence of memristor was first predicted by Leon Chua in 1976 [4], but not until nearly 30 years later was it first discovered in nanoscale devices at HP Labs [26]. The name memristor is derived from the fact that its resistance is determined by the current passed through in the past as if it can remember its history. The memory effect specifically is the time integral of the current flowing through the device: when current flows in one direction through a memristor, its electrical resistance increases; when current flows in the opposite direction, resistance decreases; and when the current is stopped, the memristor retains its previous resistance. The bistable states of the device are defined as the high-resistance state and low-resistance state. The write operation is achieved by applying large current, which changes its resistance rapidly. To read the cell, small current is applied to detect its resistance without significantly changing its resistance. In 2008, the first physical realization of a memristor was demonstrated by HP Labs: the memristive effect was achieved by moving the doping front along a TiO_2 thin-film device [26]. The materials at the different sides of the doping front have different resistivities, and the overall resistance is calculated as the two resistors in series (Figs. 1.18–1.20).

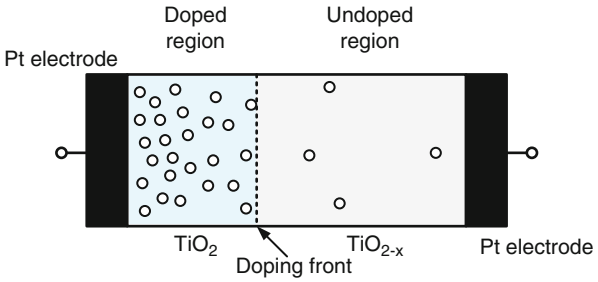


Fig. 1.18 The diagram of TiO_2/TiO_{2-x}-based memristor cell structure

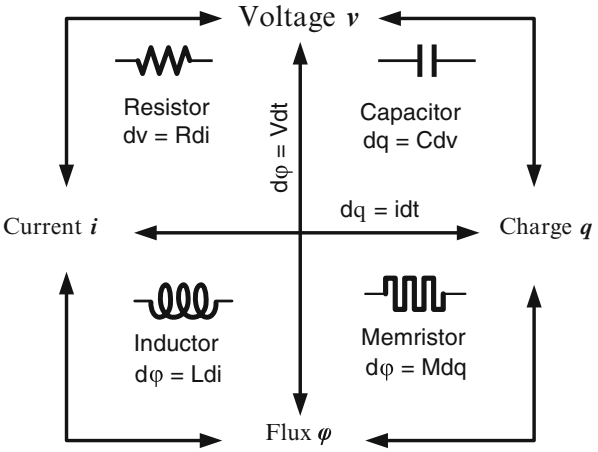


Fig. 1.19 The relationship between the fundamental elements and the prediction of the fourth element: memristor

Conductive Bridging Memory

CBRAM, also known as programmable metalization cell (PMC), is an emerging two-terminal cylinder-shaped nanoscale device. Each CBRAM memory cell is made up of two metal electrodes, one relatively inert (e.g., tungsten) and the other electrochemically active (e.g., silver or copper), with a solid electrolyte between them. Within the solid electrolyte, there exists a vertical conductive filament that grows from the inert electrode. The dynamics of CBRAM can be summarized as the physical relocation of ions between the active electrode and conductive filament. Specifically, when a positive bias voltage is applied, the active electrode will dissolve and the metal ions will accumulate on the filament, resulting in the vertical growth of filament until the two electrodes are bridged together. This sets the CBRAM into a low-resistance or ON-state. Similarly, when a negative bias voltage is applied, the ions of the conductive filament travel back the active electrode, resulting in the vertical shrink of the filament, which disconnects the two electrodes and sets the CBRAM in high-resistance or OFF-state (Fig. 1.21).

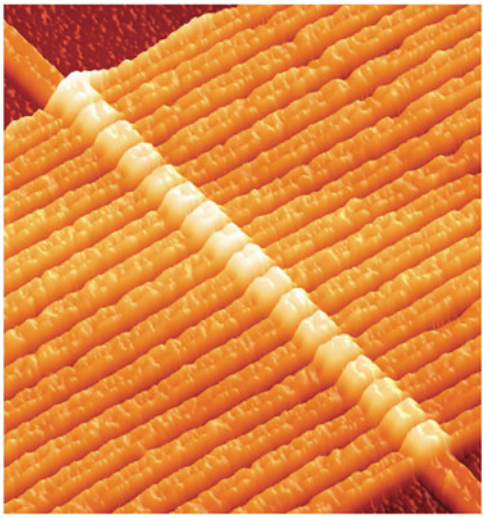


Fig. 1.20 STM photo of memristor array from HP Labs [37]

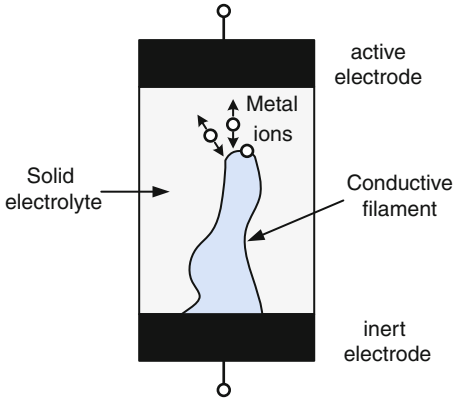


Fig. 1.21 The diagram of conductive bridging memory cell structure

1.3.1.2 Magnetoresistive Random-Access Memory

Giant magnetoresistance (GMR) was discovered in 1988 independently by the groups of Albert Fert of the University of Paris-Sud, France, and Peter Grünberg of Forschungszentrum Jülich, Germany [2]. The 2007 Nobel Prize in Physics was awarded to Albert Fert and Peter Grünberg for the discovery of GMR.

GMR is observed as a significant change in the electrical resistance depending on whether the magnetization of adjacent ferromagnetic layers is in a parallel or an antiparallel alignment. The overall resistance is relatively low for parallel

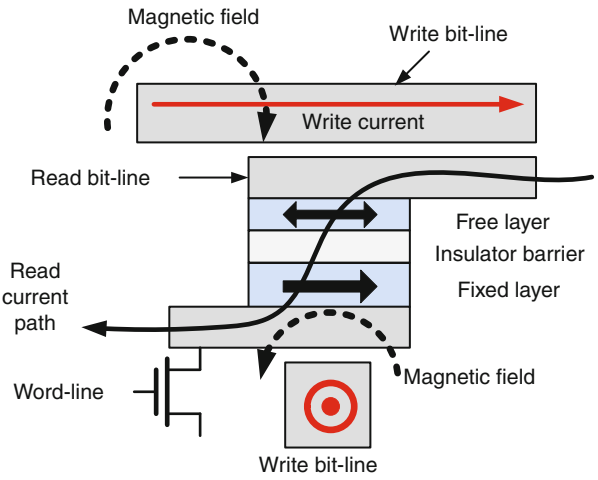


Fig. 1.22 The diagram of toggle MRAM cell structure

alignment and relatively high for antiparallel alignment.Motivated by the discovery of GMR, magnetoresistive random-access memory (MRAM) was under great interest since then.

Toggle Mode MRAM

Toggle mode MRAM [1,6,9] was the first generation of the MRAM, where external magnetic field needs to be introduced to operate the device. The diagram of a toggle mode MRAM cell is shown in Fig. 1.22. Each MRAM cell is composed of three layers: free magnetic plate on the top, fixed magnetic plate at the bottom, and insulator plate sandwiched in between. The fixed layer is strongly magnetized; thus its magnetization cannot be altered. The stored bit is denoted by the magnetization orientation in free layer, which is made of thin-film ferromagnetic material and its magnetization can be changed by applying external magnetic field, whose strength exceeds original field. To be compatible with the electronic system, current-induced magnetic field is used. Due to the magnetic tunneling effect, the electrical resistance of the cell changes due to the orientation of the fields in the two plates. By measuring the resulting current, the resistance inside any particular cell can be determined and from this the polarity of the writable plate.

MRAM has similar performance to SRAM, similar density to DRAM but much lower power consumption. For instance, IBM researchers have demonstrated MRAM devices with access times on the order of 2 ns, somewhat better than even the most advanced DRAMs built on much newer processes [6, 8, 9]. However, as mentioned above, the most basic MRAM cell suffers from the half-select problem, which limits cell sizes to around 180 nm or more. Thus avoid the interference of current-generated magnetic field among adjacent cells.

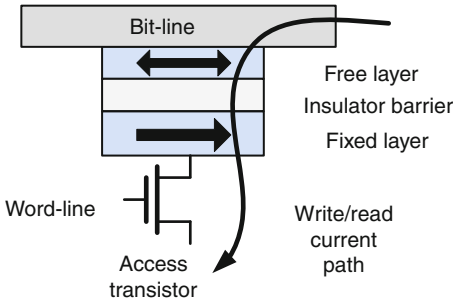


Fig. 1.23 The diagram of spin-transfer torque-based MRAM cell structure

Spin-Transfer Torque MTJ

Toggle mode MRAM faces severe scaling issues because the current required to generate the magnetic field increases as the magnetic pillar becomes smaller. The high write power consumption severely limits scaling of conventional MRAM. STT-RAM [11, 14], which is based on spin-polarized current-induced magnetic tunneling junction (MTJ) switching, has attracted a lot of attention due to its fast speed, low power consumption, and better scalability.

STT-RAM is based on the spin-transfer torque effect, in which magnetization orientations (corresponding to low-high resistance states) in magnetic multi-layer nano-structures can be manipulated via spin polarized current. The spin-transfer phenomena occur for the electric current flowing through two ferromagnetic layers separated by a thin nonmagnetic spacer layer. The current becomes spin polarized by transmission through or reflection from the first ferromagnetic layer (the pinned reference layer) and mostly maintains this polarization as it passes through the nonmagnetic spacer and enters and interacts with the second ferromagnetic layer (the free layer). This interaction exerts a spin torque on the magnetic moment of the free layer (FL) through a transfer of angular momentum from the polarized current to the FL magnetization. This spin torque can oppose the intrinsic damping of the FL causing the magnetization precession (exiting spin waves) and/ or reverse the direction of the magnetization with sufficient current strengths. Spin transfer can have important implications on electronic device applications since it provides a local means of magnetization manipulation rather than using the long-range Oersted field generated by a remote current.

This new MRAM design, called STT-RAM, is believed to have better scalability than conventional MRAM because its switching current is proportional to the MTJ size (Fig. 1.23).

Racetrack Memory

Racetrack memory [20, 27, 29], or domain-wall memory, is considered as the next generation of MRAM after STT-MRAM and also the potential solution as a future universal memory due to extremely high density and high performance. Racetrack

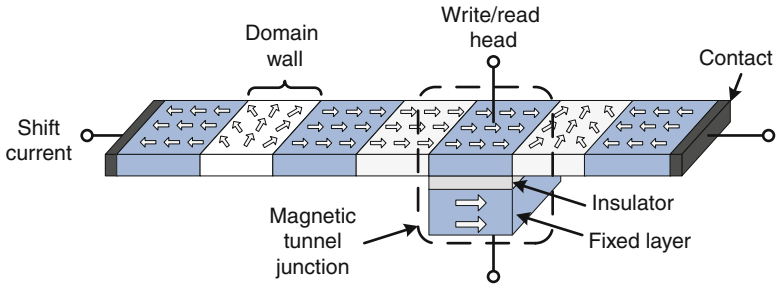


Fig. 1.24 The diagram of racetrack memory nanowire structure

memory is a thin ferromagnetic film strip that consists of many magnetic domains, where in each domain the stored information is denoted as the magnetization orientation (Fig. 1.24). The train-like domains can be pushed to move bidirectionally synchronously by a steady direct current inside and along the strip via spin-transfer torque effect. An MTJ-like junction is constructed at certain position of the ferromagnetic strip, and the domain that aligns with the fixed layer serves as the free layer in MTJ. As such, the read and write operations on the racetrack can be achieved through this constructed head, and together with the shift operation the three basic operations of racetrack memory are formed. Racetrack memory is not random-access memory as the target cell to be operated needs to be shifted to the read and write head first; instead, it works similarly as shift register.

Based on spin-transfer torque effect, racetrack memory is expected to have similar performance as its precursor STT-MRAM, but provide much higher density due to its highly packed fashion.

1.3.1.3 Phase-Change Random-Access Memory

Phase-change random-access memory (PC-RAM) device is based on the phase-changing technology of chalcogenide materials such as Ge_2SbTe_5 (GST) [10]. The chalcogenide materials have two different stable states: crystalline and amorphous, as shown in Fig. 1.25c. The resistance of chalcogenide in amorphous state is larger, in several orders of magnitude, than that in crystalline state. In addition, the chalcogenide crystal structure can be thermally switched between the two states. Exploiting such unique behaviors of chalcogenide, the phase-change memory has been made possible in recent years [3, 13, 19].

A typical mushroom-shaped phase-change memory cell is illustrated in Fig. 1.25a. A stick of heater is placed right beneath the GST layer with small contact area and is surrounded by thermally insulating materials. This structure under current will produce heat at a higher rate than it can be dissipated, which leads to the temperature surge of bottom GST; and by applying current pulse with different shapes to the heater, the phase-changing process can be activated

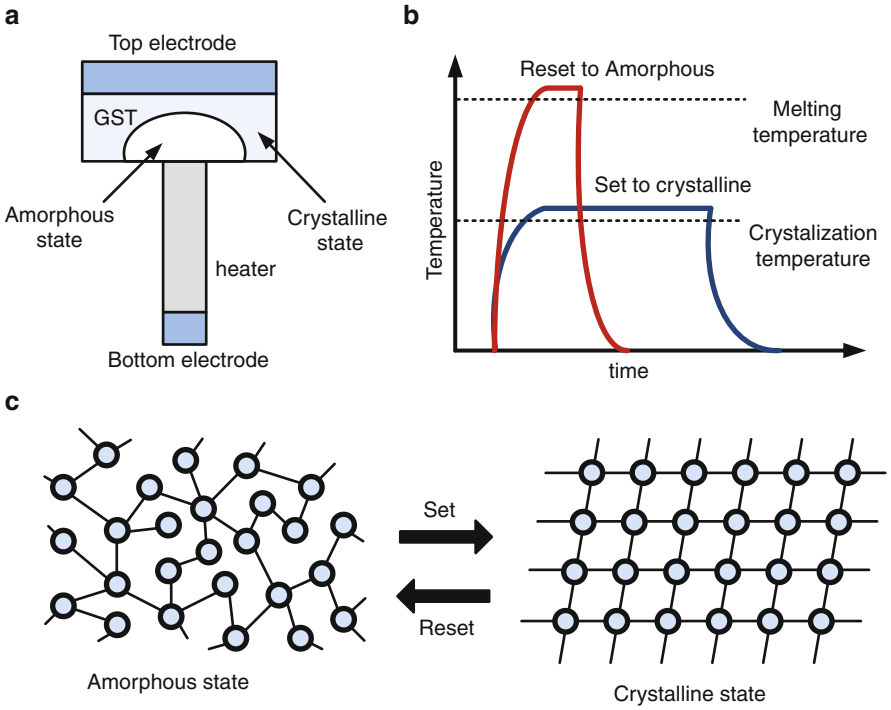


Fig. 1.25 (**a**) The cross section of phase-change memory cell. (**b**) Temperature profile of chalcogenide material in SET and REST operations. (**c**) Phase changes between the amorphous phase and polycrystalline phase

bidirectionally. As illustrated in Fig. 1.25b, when a narrow pulse of large current is applied, the temperature of GST will surge above its melting point, leading to the activation of amorphousizing process. On the other hand, when a long pulse of small current is applied, the temperature of GST will increase but stay below its melting point, resulting in the activation of crystallizing process. Similar with the previously introduced nonvolatile memories, the state readout of phase-change memory also involves the detection of resistance.

1.3.1.4 Ferroelectric Random-Access Memory

Ferroelectric random-access memory, FeRAM in short, has a cell structure that resembles that of DRAM (Fig. 1.26). However, instead of using traditional capacitor, ferroelectric capacitor is used for each cell, where the dielectric material is replaced by ferroelectric material. Ferroelectric capacitors possess two characteristics required for a nonvolatile memory cell, that is, they have two stable states corresponding to the two binary levels in a digital memory, and they retain their

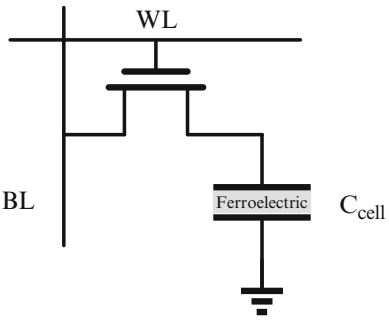


Fig. 1.26 The circuit diagram of 1T1C FeRAM cell structure

states without electrical power. The bistable states are denoted by the polarization states of the ferroelectric material, either positive or negative. In order to write a binary digit to a cell, either a positive voltage or negative voltage (normally with the amplitude of the power supply voltage, V_{DD}) is applied across the ferroelectric capacitor. A cell can be read by floating the bit-line and applying a positive voltage (V_{DD}) to the drive line while asserting the word-line. If the initial polarization state of the capacitor is negative (positive), reading the cell develops a relatively large (small) signal on the bit-line. Since this operation is destructive, the data must be written back into the cell after each read (data restore). The amount of charge needed to flip the memory cell to the opposite state is measured and the previous state of the cell is revealed.

As a nonvolatile memory, low power consumption can be expected for FeRAM when compared with SRAM and DRAM, for which leakage power and refresh power will be dissipated. Compared to the prevailing nonvolatile flash memory, FeRAM also has several advantages. Firstly, compared to the limited write cycles of flash memory ($\sim 10^5$), FeRAM has practically unlimited write–erase cycles ($\sim 10^{15}$) [12]. Secondly, FeRAM is able to provide better performance than flash memory. Unlike the write operation of flash memory, where a considerable time is introduced for charge pump to build high voltage, the FeRAM write time is almost symmetric as the read operation, both typically within 100 ns [12].

The main disadvantages of FeRAM are that the storage density for data is considerably less than flash memory and it faces scaling difficulties. One reason is that when scaled down, the memory tends to stop being ferroelectric when they are too small. Besides, FeRAM is more difficult to produce than other memory technologies, as the ferroelectric layer can be easily degraded during silicon chip manufacturing. Due to above reasons, FeRAM tends to cost more than flash memory, and therefore is often used in portable computer-based devices like smart cards tied to security systems to enter buildings and radio frequency identifier (RFID) tags used on consumer products to track inventory, where low capacity and high performance are needed.

1.3.2 NVM Design Challenges

Although the emerging NVM technologies possess features like faster speed, lower power, and higher density, they are still in their infancy and facing some design challenges. The challenge from device level is the lack of design verification platform. For CMOS technology, all devices have well-established device models that are used in SPICE simulator for design verification. However, different from conventional devices, the emerging NVM devices possess unique physics and their states are represented by a nonelectrical variable. Therefore, emerging NVM devices are not compatible with current SPICE simulator if no modification is provided. The work in [18] proposed a NVM simulation theory which is however limited for spintronic NVM technologies. In [7, 23, 24, 30, 40], one universal simulation theory that considers the nonvolatile state variables of all NVMs is developed. In Chaps. 2 and 3, we will discuss how to effectively describe the nonvolatile states of all NVM devices.

From circuit level point of view, the current cell structures and operating circuits for the existing memory technologies cannot be adopted for the emerging NVM technologies. The fundamental reason is that the working mechanism is no longer electron based. Even for different emerging NVM technologies, their working mechanisms are different from each other. This requires deliberately designed cell structures as well as agreeing peripheral circuits for each emerging NVM technology. In addition, due to immature fabrication process and the high-dimensional sensitivity of NVM devices, large device variations will be incurred, which poses even higher requirement for peripheral circuit for robust and reliable operations. In [28, 31–33, 35], different memory cell structures as well as peripheral circuits are explored for different NVMs. We will discuss the circuit level design challenges in Chap. 4 in details.

Lastly from system level, as the strengths of emerging NVM are different from conventional ones, it brings many possibilities to come up with emerging NVM-specific memory hierarchy and architectures to take advantage of performance improvement. For example, various works on replacing cache and main memory with emerging NVM technologies are attempted in [17, 21, 25, 39]. Besides simply substituting existing memory with emerging NVM, novel architectures that can exploit the performance potentials of emerging NVM are also studied, such as in-memory architecture [29, 34, 36, 41], where computation can be partially done inside memory instead of fully executed in logic units. We will discuss the system level design exploration in Chap. 5 accordingly.

References

1. Andre TW, Nahas JJ, Subramanian CK, Garni BJ, Lin HS, Omair A, Martino WL Jr (2005) A 4-mb 0.18-μm 1t1mtj toggle mram with balanced three input sensing scheme and locally mirrored unidirectional write drivers. IEEE J Solid-State Circuits 40(1):301–309

2. Baibich MN, Broto J, Fert A, Van Dau FN, Petroff F, Etienne P, Creuzet G, Friederich A, Chazelas J (1988) Giant magnetoresistance of (001) fe/(001) cr magnetic superlattices. Phys Rev Lett 61(21):2472
3. Bedeschi F, Bez R, Boffino C, Bonizzoni E, Buda E, Casagrande G, Costa L, Ferraro M, Gastaldi R, Khouri O et al (2004) 4-mb mosfet-selected phase-change memory experimental chip. In: IEEE proceeding of the 30th European solid-state circuits conference ESSCIRC 2004, pp 207–210
4. Chua L (1971) Memristor-the missing circuit element. IEEE Trans Circuit Theor 18(5):507–519
5. Dietrich S, Angerbauer M, Ivanov M, Gogl D, Hoenigschmid H, Kund M, Liaw C, Markert M, Symanczyk R, Altimime L et al (2007) A nonvolatile 2-mbit cbram memory core featuring advanced read and program control. IEEE J Solid-State Circuits 42(4):839–845
6. Engel B, Akerman J, Butcher B, Dave R, DeHerrera M, Durlam M, Grynkewich G, Janesky J, Pietambaram S, Rizzo N et al (2005) A 4-mb toggle mram based on a novel bit and switching method. IEEE Trans Magn 41(1):132–136
7. Fei W, Yu H, Zhang W, Yeo KS (2012) Design exploration of hybrid cmos and memristor circuit by new modified nodal analysis. IEEE Trans Very Large Scale Integr (VLSI) Syst 20(6):1012–1025
8. Gallagher WJ, Parkin SS (2006) Development of the magnetic tunnel junction mram at ibm: from first junctions to a 16-mb mram demonstrator chip. IBM J Res Dev 50(1):5–23
9. Gogl D, Arndt C, Barwin JC, Bette A, DeBrosse J, Gow E, Hoenigschmid H, Lammers S, Lamorey M, Lu Y et al (2005) A 16-mb mram featuring bootstrapped write drivers. IEEE J Solid-State Circ 40(4):902–908
10. Horii H, Yi J, Park J, Ha Y, Baek I, Park S, Hwang Y, Lee S, Kim Y, Lee K et al (2003) A novel cell technology using n-doped gesbte films for phase change ram. In: IEEE 2003 symposium on VLSI technology. Digest of technical papers, pp 177–178
11. Hosomi M, Yamagishi H, Yamamoto T, Bessho K, Higo Y, Yamane K, Yamada H, Shoji M, Hachino H, Fukumoto C et al (2005) A novel nonvolatile memory with spin torque transfer magnetization switching: Spin-ram. In: IEEE international electron devices meeting, 2005. IEDM technical digest, pp 459–462
12. ITRS (2010) International technology roadmap of semiconductor. http://www.itrs.net
13. Kang S, Cho WY, Cho BH, Lee KJ, Lee CS, Oh HR, Choi BG, Wang Q, Kim HJ, Park MH et al (2007) A 0.1-μm 1.8-v 256-mb phase-change random access memory (pram) with 66-mhz synchronous burst-read operation. IEEE J Solid-State Circ 42(1):210–218
14. Kawahara T, Takemura R, Miura K, Hayakawa J, Ikeda S, Lee Y, Sasaki R, Goto Y, Ito K, Meguro I et al (2007) 2mb spin-transfer torque ram (spram) with bit-by-bit bidirectional current write and parallelizing-direction current read. In: IEEE international solid-state circuits conference, 2007. ISSCC 2007. Digest of technical papers, pp 480–617
15. Kund M, Beitel G, Pinnow CU, Rohr T, Schumann J, Symanczyk R, Ufert KD, Muller G (2005) Conductive bridging ram (cbram): an emerging non-volatile memory technology scalable to sub 20nm. In: IEEE international electron devices meeting, 2005. IEDM Technical Digest, pp 754–757
16. Lacey A (1995) Thermal runaway in a non-local problem modelling ohmic beating: part 1: model derivation and some special cases. Eur J Appl Math 6(2):127–144
17. Lee BC, Ipek E, Mutlu O, Burger D (2009) Architecting phase change memory as a scalable dram alternative. ACM SIGARCH Comput Architect News 37(3):2–13
18. Manipatruni S, Nikonov DE, Young IA (2012) Modeling and design of spintronic integrated circuits. Circuits and Systems I: Regular Papers, IEEE Trans on 59(12):2801–2814
19. Oh HR, Cho Bh, Cho WY, Kang S, Choi Bg, Kim Hj, Kim Ks, Kim De, Kwak Ck, Byun HG et al (2006) Enhanced write performance of a 64-mb phase-change random access memory. IEEE J Solid-State Circ 41(1):122–126
20. Parkin SS, Hayashi M, Thomas L (2008) Magnetic domain-wall racetrack memory. Science 320(5873):190–194

21. Qureshi MK, Srinivasan V, Rivers JA (2009) Scalable high performance main memory system using phase-change memory technology. ACM SIGARCH Comput Architect News 37(3):24–33
22. Schatz R, Bethea C (1994) Steady state model for facet heating leading to thermal runaway in semiconductor lasers. J Appl Phys 76(4):2509–2521
23. Shang Y, Fei W, Yu H (2012) Analysis and modeling of internal state variables for dynamic effects of nonvolatile memory devices. IEEE Trans Circ Syst I Regular Pap 59(9):1906–1918
24. Shang Y, Fei W, Yu H (2012) Fast simulation of hybrid cmos and stt-mtj circuits with identified internal state variables. In: IEEE 2012 17th Asia and South Pacific design automation conference (ASP-DAC), pp 529–534
25. Smullen CW, Mohan V, Nigam A, Gurumurthi S, Stan MR (2011) Relaxing non-volatility for fast and energy-efficient stt-ram caches. In: IEEE 2011 17th international symposium on high performance computer architecture (HPCA), pp 50–61
26. Strukov DB, Snider GS, Stewart DR, Williams RS (2008) The missing memristor found. Nature 453(7191):80–83
27. Venkatesan R, Kozhikkottu V, Augustine C, Raychowdhury A, Roy K, Raghunathan A (2012) Tapecache: a high density, energy efficient cache based on domain wall memory. In: Proceedings of the 2012 ACM/IEEE international symposium on low power electronics and design, ACM, pp 185–190
28. Wang Y, Yu H (2012) Design exploration of ultra-low power non-volatile memory based on topological insulator. In: IEEE 2012 IEEE/ACM international symposium on nanoscale architectures (NANOARCH), pp 30–35
29. Wang Y, Yu H (2013) An ultralow-power memory-based big-data computing platform by nonvolatile domain-wall nanowire devices. In: IEEE 2013 international symposium on low power electronics and design (ISLPED), pp 329–334
30. Wang Y, Fei W, Yu H (2012) Spice simulator for hybrid cmos memristor circuit and system. In: IEEE 2012 13th international workshop on cellular nanoscale networks and their applications (CNNA), pp 1–6
31. Wang Y, Shang Y, Yu H (2012) Design of non-destructive single-sawtooth pulse based readout for stt-ram by nvm-spice. In: IEEE 2012 12th annual non-volatile memory technology symposium (NVMTS), pp 68–72
32. Wang Y, Zhang C, Nadipalli R, Yu H, Weerasekera R (2012) Design exploration of 3d stacked non-volatile memory by conductive bridge based crossbar. In: 2011 IEEE International 3D systems integration conference (3DIC), pp 1–6
33. Wang Y, Zhang C, Yu H, Zhang W (2012) Design of low power 3d hybrid memory by non-volatile cbram-crossbar with block-level data-retention. In: Proceedings of the 2012 ACM/IEEE international symposium on low power electronics and design, ACM, pp 197–202
34. Wang Y, Kong P, Yu H (2013) Logic-in-memory based mapreduce computing by nonvolatile domain-wall nanowire devices. In: IEEE 2013 13th Annual Non-Volatile Memory Technology Symposium (NVMTS)
35. Wang Y, Yu H, Zhang W (2013) Nonvolatile cbram-crossbar-based 3-d-integrated hybrid memory for data retention. IEEE Trans Very Large Scale Integr (VLSI) Syst PP(99):1–1. doi:10.1109/TVLSI.2013.2265754
36. Wang Y, Kong P, Yu H, Sylvester D (2014) Energy efficient in-memory aes encryption based on nonvolatile domain-wall nanowire. In: IEEE design automation and test conference in Europe (DATE)
37. Williams R (2008) How we found the missing memristor. IEEE Spectr 45(12):28–35
38. Wong HP, Raoux S, Kim S, Liang J, Reifenberg JP, Rajendran B, Asheghi M, Goodson KE (2010) Phase change memory. Proc IEEE 98(12):2201–2227

39. Xu W, Sun H, Wang X, Chen Y, Zhang T (2011) Design of last-level on-chip cache using spin-torque transfer ram (stt ram). IEEE Trans Very Large Scale Integr (VLSI) Syst 19(3):483–493
40. Yu H, Fei W (2010) A new modified nodal analysis for nano-scale memristor circuit simulation. In: IEEE proceedings of 2010 IEEE international symposium on circuits and systems (ISCAS), pp 3148–3151
41. Yu H, Wang Y, Chen S, Fei W, Weng C, Zhao J, Wei Z (2014) Energy efficient in-memory machine learning for data intensive image-processing by non-volatile domain-wall memory. In: IEEE 2014 19th Asia and South Pacific design automation conference (ASP-DAC)

Chapter 2
Fundamentals of NVM Physics and Computing

Abstract The bistable states are the foundation of all memory devices to store data. For conventional memory devices, the bistable states are represented by voltage levels and the transition is described by the charging and discharging of the capacitors. The transition dynamics is critical in order to obtain important figures of merit such as device operation speed and energy. Therefore, it is of great importance to quantitatively understand the physical mechanism and transition dynamics of the emerging nonvolatile devices, whose states are represented by nonelectrical variables. For the magnetoresistive random-access memory family, including toggled MRAM, STT-MRAM, and racetrack memory, the magnetization dynamics is the fundamental physics behind, while for the resistive random-access memory category, including memristor and CBRAM, the ion migration effect is the shared physics. In this chapter, both the magnetization dynamics and ion migration dynamics are introduced.

Keywords Magnetization • Ion migration • Logic-in-memory architecture • In-memory computing

2.1 Nonvolatile Memory Physics

2.1.1 Magnetization

A large portion of the emerging nonvolatile memory devices are magnetization based. The MRAM usually is formed by one insulator in the middle, sandwiched by two ferromagnetic layers, namely fixed layer that is strongly magnetized and free layer that can be easily changed. Differed by the approaches for writing, there are several phases of MRAM technology. The first-generation MRAM needs external magnetic field to switch the free layer magnetization. The second-generation STT-RAM is introduced and the free layer magnetization can be altered by polarized current, which brought significant advantages such as easy

H. Yu and Y. Wang, *Design Exploration of Emerging Nano-scale Non-volatile Memory*,
DOI 10.1007/978-1-4939-0551-5_2,

integration with current CMOS technology and high density, high reliability, etc. Recently, the third-generation domain-wall racetrack is introduced, with a series of magnetization domains in one ferromagnetic thin-film nanowire and additional shift ability. The shift is also current-induced operation. In this section, the magnetization dynamics under external field and spin current in nanosecond regime will be introduced.

2.1.1.1 Basic Magnetization Process

As an intrinsic property, electrons spin about its axis and produce magnetic field like current-carrying wire loop. From macrospin point of view, the relation between magnetization M and angular momentum associated with electron spin S can be expressed as

$$M = -\gamma S, \tag{2.1}$$

where $\gamma = 2.21 \times 10^5\ \mathrm{mA^{-1}\ s^{-1}}$ is the gyromagnetic ratio. A uniform magnetic field exerts no net force on a current loop, but it does exert a net torque, and the torque T, on the current-carrying loop under applied magnetic field H, can be expressed as

$$T = M \times H. \tag{2.2}$$

By definition, the time derivative of angular momentum is called torque. The relation between the angular momentum L and torque T reads

$$\frac{dL}{dt} = T. \tag{2.3}$$

The quantum form of Eq. (2.3) still remains valid, and then we have

$$\frac{dS}{dt} = T. \tag{2.4}$$

By combining Eqs. (2.1), (2.2), and (2.4), we can obtain the motion equation of magnetization under applied magnetic field:

$$\frac{dM}{dt} = -\gamma M \times H. \tag{2.5}$$

The processional motion described by Eq. (2.5) indicates that the magnitude of magnetization will not change and also the angle between H and M will not change, which is depicted in Fig. 2.1. This is based on that no energy loss is assumed during this process.

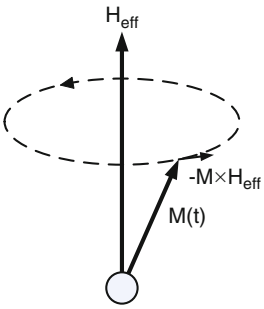


Fig. 2.1 The magnetization precession

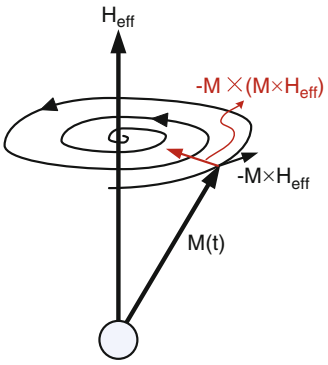


Fig. 2.2 The magnetization precession with damping

2.1.1.2 Magnetization Damping

In real systems, however, energy is dissipated through various means and the magnetization motion is damped until an equilibrium is reached. Energy dissipation can occur through the spin–spin, spin–photon, and spin–electron interactions through which the spin energy is transferred. The approach followed by Landau and Lifshitz is to introduce dissipation in a phenomenological way. In fact, they introduce an additional torque term that pushes magnetization in the direction of the effective field. Landau–Lifshitz equation in the Gilbert form, or LLG equation, then reads

$$\frac{dM}{dt} = -\gamma M \times H + \frac{\alpha}{M_s} M \times \frac{dM}{dt}. \tag{2.6}$$

The magnetization dynamics described by Eq. (2.6) is sketched in Fig. 2.2.

2.1.1.3 Spin-Transfer Torque

In 1996, Berger [3] and Slonczewski [14] predicted, which later has been confirmed experimentally [7, 16, 20], that electrons that carry enough angular momentum are able to cause magnetization precession and switching by spin-transfer torque effect. When a current passes through a ferromagnetic layer, its electron spins are polarized

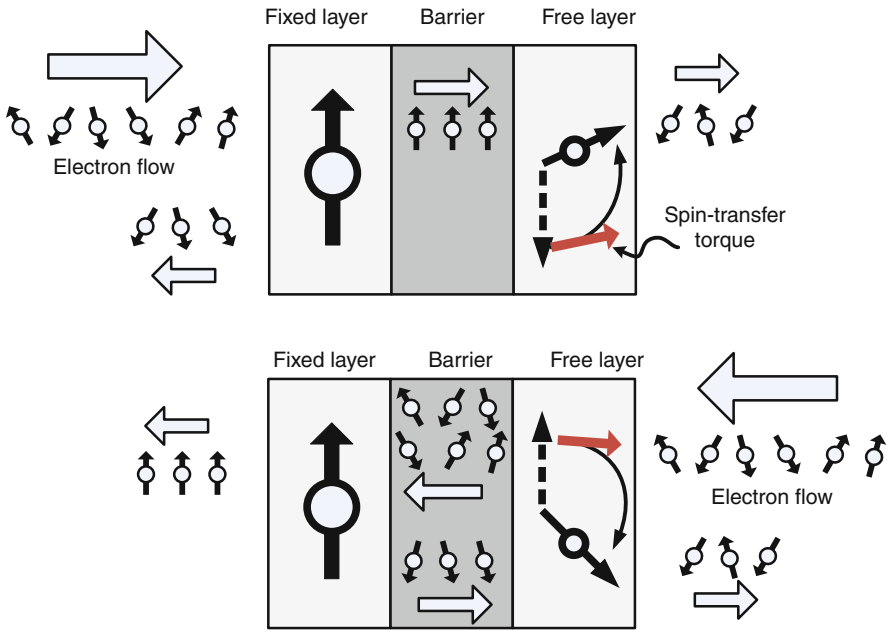


Fig. 2.3 The spin-transfer torque effect

along the magnetization direction and the current becomes spin polarized and hence carries angular momentum. And the spin-polarized current, when flows through a second ferromagnetic layer, exerts a spin torque on the local magnetic moment of the magnetic layer and causes the magnetization precession and switching when the current is large enough (Fig. 2.3).

Therefore, the dynamics of the free layer magnetization can be determined by the LLG equation in conjunction with an additional term for spin-transfer torque:

$$\frac{dM}{dt} = -\gamma M \times H + \frac{\alpha}{M_s} M \times \frac{dM}{dt} - \frac{a_J}{M_s}[M \times (M \times P)], \tag{2.7}$$

where P is the magnetization of fixed layer, M_s is the saturation magnetization, a_J is a factor related to the interfacial interaction between magnetic moment and spin-polarized electrons, which is proportional to the current density, and the sign of a_J depends on the direction of current. When applied properly, the current is able to cancel the damping and switch the magnetization of free layer by spin-transfer torque.

For the current-induced magnetization precession, there exists a threshold current density J_{c0}, and by applying current larger than J_{c0} the magnetization can be switched back and forth:

$$J_{c0} = \frac{2e\alpha M_s t_F (H_K + H_{\text{ext}} + 2\pi M_s)}{\hbar \eta}, \tag{2.8}$$

where e is the electron charge, α the damping constant, M_s the saturation magnetization, t_F the thickness of free layer, $\hbar$ the Planck constant, η the spin-transfer efficiency, H_K the anisotropy field, and H_{ext} the external applied magnetic field.

There are three modes for the current-driven magnetization switching: thermal activation associated with switching time longer than 10 ns, processional switching associated with switching time less than a few nanoseconds, and dynamic reversal as a compound process of both. The above three modes reveal the switching time and current density relationship.

For the fast processional switching, the switching time is reversely proportional to the applied current:

$$\tau_p \propto \frac{1}{(J - J_{c0})} \ln\left(\frac{\pi}{2\theta_0}\right), \tag{2.9}$$

where θ_0 is the initial magnetization angle deviated from the easy axis. At finite temperature, θ_0 is determined by thermal distribution. For the fast processional switching in the regime of nanosecond, it usually takes a current density that is several times greater than J_{c0}.

In the slow thermal-activated switch regime, the switching current is dependent on the current pulse width and thermal stability factor $\Delta = K_u V / k_B T$ of the free layer. Interestingly, the current density can be smaller than the critical density and therefore is useful for current reduction. In this case, the standard thermal agitation will be assisted by spin current, which introduces extra energy to reach enough energy for the magnetization switching. The relation reads

$$J(\tau) = J_{c0}\left[1 - \frac{K_u V}{k_B T} \ln\left(\frac{\tau}{\tau_0}\right)\right], \tag{2.10}$$

where $\tau_0 \propto 1$ ns is the inverse of the attempt frequency and $K_u V$ the anisotropy energy.

As the fast processional switching requires large current density which reduces the robustness and causes undesired switching and the slow thermal activation process takes too long time, the most interesting switching mode is the dynamic reversal at intermediate current pulses. Although the dynamic switching mode corresponds to the operating speed of practical STT-RAM, the explicit formula is hard to be derived due to its complicated process. Therefore, the dynamic reversal is usually studied as a combination of processional and thermally activated switching.

2.1.1.4 Magnetization Dynamics

Magnetic domains are formed by the competition between the various energy terms involved in a magnetic object. The energy of a magnetic structure is the sum of the exchange energy, the anisotropy energy, the Zeeman energy, and the demagnetization energy. The magnetic system seeks to minimize its overall

free energy. Since the magnitude of the magnetization cannot change the way to minimize the energy is to vary the direction of the magnetization. The exchange energy seeks to align the spins with each other, the anisotropy energy seeks to align the spins with an axis determined by the crystal structure, and the Zeeman energy aligns the spins with an external field. When the magnetostatic dipole–dipole interaction is also taken into account, known as the demagnetization energy, a nonuniform magnetization will generally be found as the lowest compromise of overall energy. Short-range exchange energy will prevail a configuration with the spins aligned, and long-range dipole–dipole interaction will however prevail a magnetic state with minimal net magnetization. In the macrospin model of magnetization dynamics study, the short-range exchange energy can be ignored. The energy associated with anisotropy field can be written as

$$\epsilon = K(1 - m_x^2), \tag{2.11}$$

where K is the anisotropy constant and m_x the normalized magnetization in x-direction, defined as the in-plane easy axis.

For the Zeeman energy by external applied field, we have

$$\epsilon = -\mu_0 M H_{\text{ext}} \tag{2.12}$$

in which μ_0 is called the vacuum permeability.

Demagnetization field represents the work necessary to assemble magnetic poles in a given geometric configuration. Since the thickness is so small compared to the in-plane dimensions, the dominant term is approximately the demagnetizing field of a uniformly magnetized thin film with infinite lateral dimensions, namely $H_D = [0, 0, -M_S m_z]$. The associated energy density is

$$\epsilon = -\frac{1}{2}\mu_0 M H_D = -\frac{1}{2}\mu_0 M_s^2 H_D^2. \tag{2.13}$$

Combining all three terms together, we have the overall energy:

$$\epsilon = K(1 - m_x^2) + \frac{1}{2}\mu_0 M_s^2 m_z^2 - \mu_0 M \cdot H_{\text{ext}}. \tag{2.14}$$

When pulled out of equilibrium, the magnetization is subject to an effective field. Therefore, we are able to calculate the effective field H in the LLG Eq. (2.7):

$$H_{\text{eff}} = -\frac{1}{\mu_0 M_s}\frac{\delta \epsilon}{\delta m} = [H_x^{\text{ext}} + H_K m_x, 0, -M_s m_z], \tag{2.15}$$

where $H_K = 2K/(\mu_0 M_s)$ is the anisotropy field.

In dimensionless form, we have

$$\omega = \frac{1}{2}Q(1 - m_x^2) + \frac{1}{2}m_z^2 - m \cdot h^{\text{ext}} \tag{2.16}$$

and

$$h_{\text{eff}} = -\frac{\delta\omega}{\delta m} = [h_x^{\text{ext}} + Qm_x, 0, -m_z] \tag{2.17}$$

with $Q = 2K/(mu_0M_s^2) = H_K/M_s$. Since the magnitude of m does not change, which suggests that it can be rewritten in spherical coordinates,

$$\frac{d\theta}{d\tau} = h_\phi - \alpha \sin\theta \frac{d\phi}{d\tau} \tag{2.18}$$

$$\sin\theta \frac{d\phi}{d\tau} = -h_\theta + \alpha \frac{d\theta}{d\tau} \tag{2.19}$$

by multiplying α to Eq. (2.18), adding Eq. (2.19), multiplying α to Eq. (2.19), and subtracting Eq. (2.19), we could obtain a set of first-order differential equations:

$$(1+\alpha^2)\frac{d\theta}{d\tau} = h_\phi - \alpha h_\theta \tag{2.20}$$

$$(1+\alpha^2)\sin\theta \frac{d\phi}{d\tau} = -h_\theta + \alpha h_\phi. \tag{2.21}$$

The stable processional states can be obtained by numerical integration of Eqs. (2.20) and (2.21), as first demonstrated by Sun [17]. With approximation, the current threshold for the establishment of a stable magnetization trajectory may be simply derived from standard perturbation theory. Clearly, in the absence of current and under the action of any applied field $h_x^{\text{ext}} > 0$, the stable magnetization direction satisfies $m_x = 1$, or, $\theta = \pi/2$, $\phi = 0$. As the damping constant α is generally small, so that in the investigated trajectory zone, the spin-transfer torque roughly balances the damping. Therefore χ is of the same order of magnitude as α, and both can be treated as small parameters. To investigate the stable processional states, we focus on trajectories in which the magnetization is close to its equilibrium states. This suggests the replacement $\theta = \pi/2 + \xi$, so that ξ and ϕ can be treated as small. Taking $1 + \alpha^2 \cong 1$ leads to the following linearized equations of magnetization motion:

$$\frac{d\xi}{d\tau} = h_\phi - \alpha h_\theta \tag{2.22}$$

$$\frac{d\phi}{d\tau} = -h_\theta + \alpha h_\phi \tag{2.23}$$

with

$$h_\theta = -(1 + Q + h_x^{\text{ext}})\xi - \chi\theta \tag{2.24}$$

$$h_\phi = +\chi\xi - (Q + h_x^{\text{ext}})\phi. \tag{2.25}$$

Let $u = Q + h_x^{\text{ext}}$ and λ be the first derivative operator $d/d\tau$; the first-order differential equation then reads

$$\lambda^2 + [\alpha + 2(\alpha u - \chi)]\lambda + u(1+u) = 0 \tag{2.26}$$

solution precession

$$\theta = e^{-\frac{t}{t_0}} \cos(\omega t + \Phi_0), \tag{2.27}$$

where

$$t_0 = \frac{1}{\gamma_0 M_s(\chi_{\text{crit}} - \chi)} \tag{2.28}$$

$$\omega = \gamma_0 M_s \sqrt{u(1+u) - (\chi_{\text{crit}} - \chi)^2} \tag{2.29}$$

with

$$\chi_{\text{crit}} = \alpha(\frac{1}{2} + Q + h_x^{\text{ext}}) \cong \frac{\alpha}{2} \tag{2.30}$$

if $Q, h_x^{\text{ext}} \ll 1$.

2.1.1.5 Domain-Wall Propagation

Recalling spin-transfer torque effect that when a current is passed through a ferromagnetic material, electrons will polarize, that is, the spin of the conduction electron will align with the spin of the local electrons carrying the magnetic moment of the material. When the conduction electrons subsequently enter a region of opposite magnetization they will eventually become polarized again, thereby transferring their spin momentum to the local magnetic moment, as required by the law of conservation of momentum. Therefore, when many electrons are traversing a domain wall (DW), magnetization from one side of the DW will be transferred to the other side. Effectively the electrons are able to push the DW in the direction of the electron flow i.

The influence of current on DW dynamics is often treated by including two spin torque terms in the LLG equation, Eq. (2.6). When the current, with current density J, is flowing in one direction, the x-direction, the LLG equation including the spin torque terms can be written as

$$\frac{\partial M}{\partial t} = -\gamma M \times H + \frac{\alpha}{M_s} M \times \frac{\partial M}{\partial t} - \eta J \frac{\partial M}{\partial x} + \beta \eta J M \times \frac{\partial M}{\partial x}, \tag{2.31}$$

where last two terms are added to the regular LLG equation to describe the effect of current on the magnetization dynamics. The first of these terms expresses the

adiabatic spin-transfer torque as exerted by a current on magnetic DWs with η the strength of the effect. The second STT term in the equation describes the nonadiabatic current-induced effect which relative strength is parameterized by β. The strength of the adiabatic spin torque, η, is widely agreed on [2, 10, 18, 19] and given by

$$\eta = \frac{g\mu_B P}{2eM_s}, \tag{2.32}$$

where g is the Land factor, μ_B the Bohr magneton, e the electron charge, M_s the saturation magnetization, and P the electron polarization, all of which values are very well known except for the electron polarization. Estimates for P range from $P = 0.4$ to $P = 0.7$ [1].

2.1.2 Ion Migration Dynamics

Unlike the nonvolatile memories from MRAM family that they share same physical dynamics as magnetization precession, the physics behind other emerging memories are more or less different from each other. From a mechanistic point of view, therefore, it is hard to conclude one universal physical model that is suitable for all. Nevertheless, they still share many similarities that make it possible for all to have similar phenomenological model, especially for various resistive random-access memory (ReRAM) technologies. For example, the ion migration model proposed by Mott–Gurney [13] is used to describe the kinetics of both CBRAM and memristor devices according to [15, 21], though their types of ion and reaction equations are different from each other. In this section, we will review the ion migration physics that is able to describe dynamics of thin-film ReRAM devices.

The ion migration is involved in many ReRAM switching dynamics. For example, for CBRAM device, its metal electrode will dissolve and the metal ions will transport under applied electric field, and for memristor, oxygen ion and oxygen vacancy migrate according to applied electric field. For thick films, the ion transport follows linear velocity and electric field relationship due to its low electric field [5]; however, for very thin film, which is the case for most nanoscale ReRAM devices, electric field may be high and linear drift law will be invalidate. For example, when a thin-film ReRAM device with $\sim$10 nm thickness is operated at a few volts, its electric field may exceed 1 MV/cm. Such a high electric field can significantly reduce the activation barrier for migration inside the solid, and hence it is natural to suspect there might be strongly nonlinear ionic transport. In fact, such nonlinear ionic transport under strong electric field has been experimentally confirmed [4]. Figure 2.4 shows the linear ionic transport for small electric field and strongly nonlinear transport while electric field is large.

The hyperbolic sine function in the nonlinear drift model predicts the existence of threshold voltage for thin-film ReRAM devices. As the drift velocity has positive

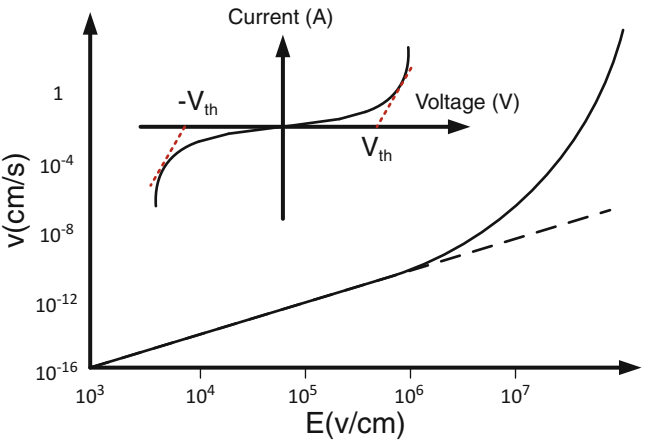


Fig. 2.4 The nonlinear ion drift velocity under strong electric field with introduced bidirectional thresholds for thin-film ReRAM devices

correlation with current density, the nonlinear drift therefore brings nonlinear I-V curve, as shown in Fig. 2.4. Similar to diode, the nonlinear I-V curve also brings a threshold voltage, below which the device can be considered turned off. Different from diode, the hyperbolic sine function indicates that the thin-film ReRAM devices will behave like a bidirectional diode with one positive voltage threshold as well as a mirrored negative voltage threshold.

According to [5], such relationship can be described by hyperbolic sine function,

$$J = 4avn_0 e^{-\frac{U_a}{kT}} \sinh\left(\frac{qaE}{kT}\right), \tag{2.33}$$

where N_i is the density of the metal ions in the solid electrolytes, f is the attempt-to-escape frequency, E_a is the activation energy, E is the electric field, kT is the thermal energy, and a is the effective hopping distance. The illustration of energy profile with definitions of variables is shown in Fig. 2.5.

A simplified derivation of above continuum transport equation that takes account of both concentration gradients and high electrostatic potential gradients will be reviewed in the following. When dealing with equilibrium phenomena of ion migration, one must take consideration of electrochemical potential as they carry a net charge, instead of only normal chemical potential. The electrochemical potential of a species can be defined by

$$\overline{\mu} = \mu_0 + kT \ln(n) + q\phi, \tag{2.34}$$

where μ_0 is the standard chemical potential of the species, q is the charge, ϕ is the local electrostatic potential, and n is the species concentration which can be approximated by

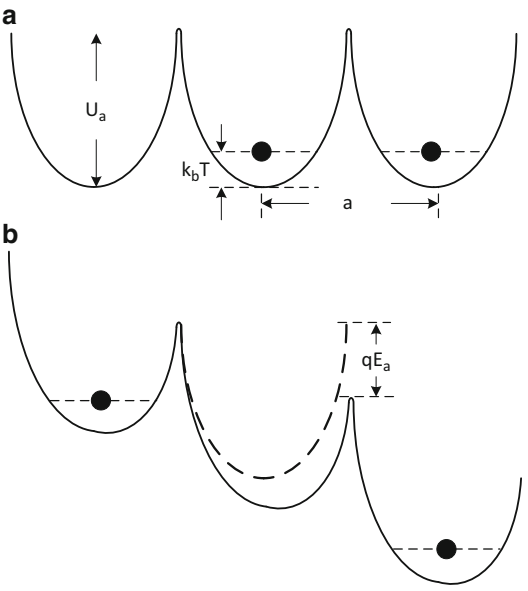


Fig. 2.5 Illustration of vacancy diffusion (**a**) without and (**b**) with an applied electric field with definitions of physical parameters

$$n \propto \exp\left(\frac{-q\phi}{kT}\right). \tag{2.35}$$

It is clear that any valid continuum equation must satisfy the condition that for zero electrochemical potential gradient, the flux density of the corresponding species must be zero. Thus

$$J = 0 \text{ for } \frac{\partial \overline{\mu}}{\partial x} = 0, \tag{2.36}$$

where x is a position variable and J the flux density of the species in the x-direction.

At low field strengths, any general continuum transport equation must reduce to the normal linear transport phenomenological equations. Thus, the following equation can be written as

$$J = -D\frac{\partial n}{\partial x} + nvE\frac{q}{|\,q\,|} \text{ for small } E, \tag{2.37}$$

where D is the diffusion coefficient, v the mobility of the migrating species, and E the electric field strength in the x-direction. By substituting the Einstein relation $v/D = |q|/kT$, Eq. (2.37) can be rewritten in the alternative form:

$$J = \frac{-Dn}{kT}\frac{\partial \overline{\mu}}{\partial x} \text{ for small } E. \tag{2.38}$$

Apparently, Eq. (2.38) satisfies the boundary condition (2.36). And in the absence of a concentration gradient, and for high field strengths, we shall for present purposes regard the flux density as given by equation

$$J \propto \sinh\left(\frac{\mu^* E}{kT}\right) \text{ for } \frac{\partial n}{\partial x} = 0, \tag{2.39}$$

where μ^* is a phenomenological coefficient with the dimensions of an electric dipole. In fact, Eq. (2.39) is constructed more from an experimental phenomenological approach than it is from theoretical mechanistic approach. Experimental justification for the hyperbolic sine function dependence on the field, and for its limiting form an exponential dependence on field, can be found in [13]. Then we have

$$J = \frac{-Dn}{kT} \cdot \frac{\partial \overline{\mu}}{\partial x} \cdot \frac{\sinh(\mu^* E/kT)}{\mu^* E/kT}. \tag{2.40}$$

It is then obvious that Eq. (2.40) satisfies boundary condition (2.36) over the entire range of all the variables. In addition, when field strength is sufficiently low (i.e., $\mu^* E/kT \ll 1$), the sinh function could be replaced by its argument, which reduces conditions Eq. (2.40) to Eq. (2.38). Again for zero concentration gradient, in the case of $\partial\overline{\mu}/\partial x = -qE$, Eq. (2.40) reduces to Eq. (2.39). Equation (2.40) is therefore consistent with the asymptotic equations (2.38) and (2.39) and furthermore satisfies condition (2.36). It is clear that Eq. (2.40) cannot be regarded as unique in satisfying these conditions.

By substituting Eq. (2.34), $D = 4a^2 v \exp(-U_a/kT)$ and $\mu^* = qa$, Eq. (2.40) can be rewritten as

$$J = 4avn_0 e^{-\frac{U_a}{kT}}\left[\sinh\left(\frac{qaE}{kT}\right) - \frac{\partial \ln(n)}{\partial x} a \frac{\sinh(\mu^* E/kT)}{\mu^* E/kT}\right]. \tag{2.41}$$

Practically for thin films it can be approximated that the concentration of mobile ions at the interface towards which the ions are moving can have no direct influence on the transport rate; the latter terms can be omitted, and thus Eq. (2.41) becomes Eq. (2.33).

2.2 Nonvolatile In-Memory Computing

The memory technology can affect the computing performance from two aspects. The first aspect is from memory technology itself. For example, memory access latency, access energy as well as memory density are important figures of merit of memory that tell how fast and how efficiently data can be stored and retrieved. Besides the memory technology itself, the second aspect is the way how memory is integrated with logic. This will greatly affect how fast and how efficiently the retrieved data can be processed by logic units. In this part, the memory and logic integration issues will be discussed.

2.2.1 Memory-Logic-Integration Architecture

Current memory and logic integration has hit the memory wall. That is to say, the memory is the bottleneck of the whole system which is not able to provide data at the rate that processor requires. In this case, the processor is not fully operating. Such hardware resource waste is especially severe for data-intensive applications. This is because current memory and logic components are separated. The data required by logic components will be read out from memory and write back the results to memory through I/O after execution is done. In other words, the linking bridge between logic and memory is the limited I/Os.

The memory-logic throughput can be determined by two factors: I/O pin numbers and how fast the I/O can be operated. Speed-wise, the current I/O can be operated ranging from 100 MB per second per pin for flash memory to 10,000 MB per second per pin for GDDR memory. Although higher I/O operating frequency is desirable, it has fundamental limits such as signal propagation physics (Maxwell's equation) and signal integrity issues (cross talk, loss, and reflection). On the other hand, the I/O number depends on the packaging technology, number of ball bumps, for example. Wider I/O requires higher power and cost budget, and current maximum is about 512 bits. Higher number of pins requires packaging and interconnect breakthroughs. As a rule of thumb, a new generation of packaging technology comes at every six years: 1994 lead technology TSOP, 2000 FBGA (0.8 mm), 2006 PoP/MCM (0.4 mm) and 2012 die stack (40–50 μm). As the pitch of interconnects is getting smaller, it is promising to have more I/O for memory chip.

2.2.2 Logic-in-Memory Architecture

Instead of improving memory bandwidth, it is also possible to reduce the required data communication traffic between memory and processor. The basic idea behind this is that, instead of feeding processor large volume of raw data, it is beneficial to preprocess the data and provide processor only intermediate result. The key to lower communication traffic is the operand reduction. For example, to perform a sum of ten number, instead of transmitting ten number to processor, in-memory architecture is able to obtain the sum by in-memory logic and transmit only one result, thus reducing traffic by 90%. To perform in-memory logic, it is necessary to implement logic inside memory so that preprocessing logic can be done. Such architecture is called logic-in-memory architecture [6,8,9,11,12]. Figure 2.6 shows logic-in-memory architecture at memory cell level. The example illustrated here is an in-memory full adder with both *sum* logic and *carry* logic.

The basic circuitry, including the access transistor, word-line, and bit-lines, is to ensure memory access. The data is stored in nonvolatile memory devices which have either low or high resistance. Redundant data is required for each bit of data for logic purpose. Combinational logic circuit is added inside which the nonvolatile devices are equivalent to transistors: considered turned on if at low-resistance state

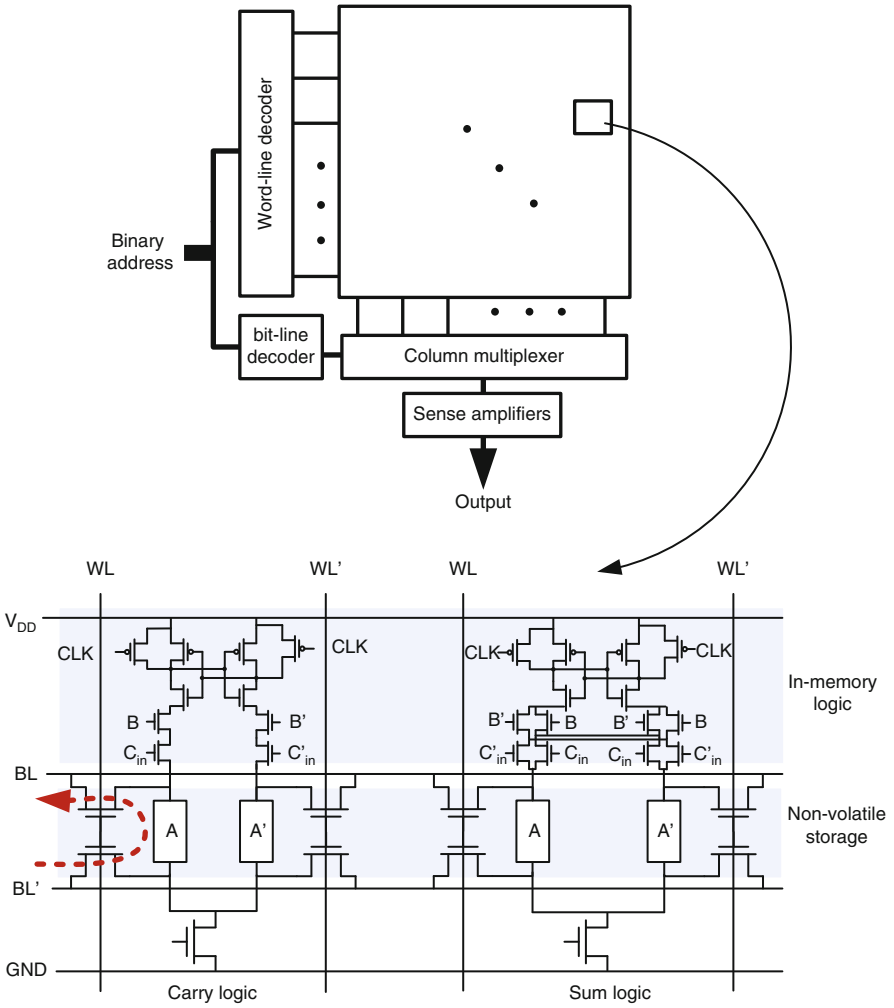


Fig. 2.6 In-memory computing architecture at memory cell level

or turned off if at high-resistance state. In such architecture, the desired result can be obtained immediately without reading operands as if the results are already stored in data array and it can just be "read out." This is very useful for some specific applications as this architecture is able to preprocess data without loading data to processor with extremely short latency.

As the logic is inserted to one cell or a few cells, it is limited to small size; thus, it cannot be made complex. Usually only simple logic is suitable for such architecture; otherwise, the overhead would be overwhelming. However, though simple logic in such architecture is able to share the workload of processor, its effect to reduce communication traffic is not obvious due to limited operand reduction. In addition,

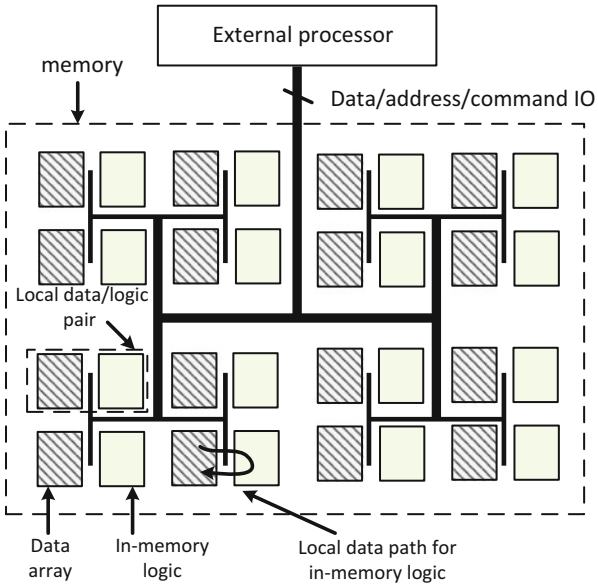


Fig. 2.7 In-memory computing architecture at memory block level

similar to the operation of memory, for the whole data array, only a few logic can be active concurrently at one time. This leads most logic circuit to be idle at most of the time, which not only is a waste of computational resources but also incurs leakage power for CMOS logic. Another disadvantage is that the data needs to be stored in a very strict manner, determined by how the in-memory is organized.

An alternative in-memory architecture at block level which is more suitable for traffic reduction is illustrated in Fig. 2.7. A memory data is usually organized in H-tree fashion, and the data block can be the data array or a number of data arrays that belong to same "H-tree" branch. Instead of inserting in-memory logic at memory cell level inside the data array, the architecture in Fig. 2.7 pairs each block of data with in-memory logic (accelerators). Different from the cell-level in-memory architecture, the accelerators can be made with higher complexity, and the number of accelerators for each data block can also be customized. The data flow of the block-level in-memory architecture is to read out data from data block to in-memory logic, which performs particular functionality and then writes back the result. The data also needs to be stored in assigned blocks, but it is much more flexible than that of cell-level in-memory architecture. The block-level in-memory architecture is very effective to reduce communication traffic between memory and processor. This is because high operand reduction can be achieved due to higher accelerator complexity. For example, for face recognition in image processing application, instead of transmitting a whole image to obtain a Bool result, the result can be directly gained through in-memory logic. In other words, the block-level in-memory architecture is suitable for big data-driven applications where traffic reduction is more important than latency reduction.

References

1. Beach G, Tsoi M, Erskine J (2008) Current-induced domain wall motion. J Magn Magn Mater 320(7):1272–1281
2. Berger L (1978) Low-field magnetoresistance and domain drag in ferromagnets. J Appl Phys 49(3):2156–2161
3. Berger L (1996) Emission of spin waves by a magnetic multilayer traversed by a current. Phys Rev B 54(13):9353
4. Cabrera N, Mott N (1949) Theory of the oxidation of metals. Rep Progress Phys 12(1):163
5. Dignam M (1968) Ion transport in solids under conditions which include large electric fields. J Phys Chem Solids 29(2):249–260
6. Hanyu T, Teranishi K, Kameyama M (1998) Multiple-valued logic-in-memory vlsi based on a floating-gate-mos pass-transistor network. In: Solid-state circuits conference, 1998. Digest of technical papers, 1998 IEEE International. IEEE, New York, pp 194–195
7. Katine J, Albert F, Buhrman R, Myers E, Ralph D (2000) Current-driven magnetization reversal and spin-wave excitations in co/cu/co pillars. Phys Rev Lett 84(14):3149
8. Kautz WH (1969) Cellular logic-in-memory arrays. IEEE Trans Comp 100(8):719–727
9. Kimura H, Hanyu T, Kameyama M, Fujimori Y, Nakamura T, Takasu H (2004) Complementary ferroelectric-capacitor logic for low-power logic-in-memory vlsi. Solid State Circ IEEE J 39(6):919–926
10. Li Z, Zhang S (2004) Domain-wall dynamics and spin-wave excitations with spin-transfer torques. Phys Rev Lett 92(20):207–203
11. Matsunaga S, Hayakawa J, Ikeda S, Miura K, Hasegawa H, Endoh T, Ohno H, Hanyu T (2008) Fabrication of a nonvolatile full adder based on logic-in-memory architecture using magnetic tunnel junctions. Appl Phys Expr 1(9):1301
12. Matsunaga S, Hayakawa J, Ikeda S, Miura K, Endoh T, Ohno H, Hanyu T (2009) Mtj-based nonvolatile logic-in-memory circuit, future prospects and issues. In: Proceedings of the conference on design, automation and test in Europe. European Design and Automation Association, Leuven, pp 433–435
13. Mott NF, Gurney RW (1964) Electronic processes in ionic crystals. Dover, New York
14. Slonczewski JC (1996) Current-driven excitation of magnetic multilayers. J Magn Magn Mater 159(1):L1–L7
15. Strukov DB, Williams RS (2009) Exponential ionic drift: fast switching and low volatility ofáthin-film memristors. Appl Phys A 94(3):515–519
16. Sun J (1999) Current-driven magnetic switching in manganite trilayer junctions. J Magn Magn Mater 202(1):157–162
17. Sun J (2000) Spin-current interaction with a monodomain magnetic body: a model study. Phys Rev B 62(1):570
18. Tatara G, Kohno H (2004) Theory of current-driven domain wall motion: spin transfer versus momentum transfer. Phys Rev Lett 92(8):086–601
19. Thiaville A, Nakatani Y, Miltat J, Suzuki Y (2005) Micromagnetic understanding of current-driven domain wall motion in patterned nanowires. EPL (Europhys Lett) 69(6):990
20. Tsoi M, Jansen A, Bass J, Chiang WC, Seck M, Tsoi V, Wyder P (1998) Excitation of a magnetic multilayer by an electric current. Phys Rev Lett 80(19):4281
21. Yu S, Wong HS (2011) Compact modeling of conducting-bridge random-access memory (cbram). Electron Dev IEEE Trans 58(5):1352–1360

Chapter 3
Nonvolatile State Identification and NVM SPICE

Abstract Hybrid integration of CMOS and nonvolatile memory (NVM) devices has become the technology foundation for emerging nonvolatile memory based computing. Therefore, it is under great interest inincluding the emerging new NVM devices in the standard CMOS design flow. The primary challenge to validate a hybrid design with both CMOS and nonvolatile devices is to develop a SPICE-like simulator that can simulate the dynamic behavior accurately and efficiently. The previous approaches either ignore dynamic effect without considering nonvolatile states for dynamic behavior or need complex equivalent circuits to represent those devices. This chapter details a new modified nodal analysis for nonvolatile memory devices with identified nonelectrical state variables for dynamic behavior. As such, compact SPICE-like implementation can be derived for the new nonvolatile memory devices in the hybrid NVM/CMOS designs. As demonstrated by a number of examples, the developed NVM-SPICE simulator can not only capture dynamic behaviors of emerging NVM devices but also improve simulation efficiency by around 100× compared to the previous equivalent circuit based approaches.

Keywords SPICE • Modified nodal analysis • CMOS/NVM co-simulation • Non-volatile device modeling

3.1 SPICE Formulation with New Nanoscale NVM Devices

In order to deal with a design composed of large number of nonvolatile memory (NVM) devices but also the other traditional devices such as CMOS transistors, the new NVM element needs to be included into a circuit simulator like SPICE [34] with state explicitly described. Traditional nodal analysis (NA) only contains nodal voltages at terminals of devices. Since an inductor is short at *dc* and its two terminal voltages become dependent, the state matrix is indefinite at *dc*. This problem is resolved by a modified nodal analysis (MNA) [19], which modifies the NA by adding branch currents as state variables. However, many nontraditional devices

H. Yu and Y. Wang, *Design Exploration of Emerging Nano-scale Non-volatile Memory*,
DOI 10.1007/978-1-4939-0551-5_3,

(memristor, spin-transfer torque device, etc.) introduced at the nanoscale have to be described by state variables different from the traditional nodal voltages and branch currents, i.e., electrical states. As such, the conventional circuit formulation in MNA may not be able to include these new nanodevices, which contain the nonelectrical or nonvolatile states.

Due to the lack of development of related circuit simulator, all circuits with aforementioned NVM devices are currently designed in very limited scalability. The challenges faced when integrating with CMOS devices remain unsolved. With the aid of one SPICE-like simulator for NVM devices developed in this chapter to describe both electrical and nonelectrical states, the hybrid CMOS/NVM co-simulation can be efficiently conducted with high accuracy.

3.1.1 Traditional Modified Nodal Analysis

Kirchhoff's current law (KCL) and Kirchhoff's voltage law (KVL) are two fundamental equations governing the electric property of a circuit. These two laws can be compactly formulated by an incidence matrix determined by the topology of circuits. Assuming n nodes and b branches, the incident matrix $E \in (R^{n\times b})$ is defined by

$$e_{i,j} = \begin{cases} 1, & \text{if branch } j \text{ flows into node } i \\ -1, & \text{if branch } j \text{ flows out of node } i \\ 0, & \text{if branch } j \text{ is not included at node } i\,. \end{cases} \tag{3.1}$$

By further denoting branch current as j_b, branch voltages as v_b, and nodal voltages as v_n, KCL and KVL can be described by (Table 3.1)

$$\begin{cases} E\,j_b = 0, & \text{KCL} \\ E^T v_n = v_b, & \text{KVL.} \end{cases} \tag{3.2}$$

Ideally, the branch current vector is a function purely dependent on the nodal voltages under the device branch equation:

$$j_b = \frac{d}{dt} q\left(E^T v_n, t\right) + j(E^T v_n, t). \tag{3.3}$$

However, as inductor and voltage source become indefinite at *dc* when using the nodal voltages only (NA), the MNA breaks the branch current vector into four pieces with four corresponding incident matrices and deploys branch-inductive current j_l and branch source current j_i as new state variables. As such, the KCL and KVL in Eq. (3.2) become

Table 3.1 Definitions of variables used for SPICE-like simulator

Variables	Definitions
s_m	Nonvolatile state variable for new MNA
v_n, j_b	Traditional state variables for MNA: nodal voltage and branch current
v_b, j_i, j_l	Branch voltage, source current, and inductor current
G, C, L_l	Traditional conductance, capacitance, and inductance
S	New Jacobian as memductance
$K_v^F, K_s^F, K_v^G, K_s^G$	New Jacobians introduced by s_m
E_c, E_g, E_l, E_i	Incident matrix for capacitor, resistor, inductor, and current source
E_m	New incident matrix introduced for NVM devices
f, g	Functions introduced by s_m for new state equations

$$\begin{cases} \dfrac{d}{dt} E_c q\left(E_c^T v_n, t\right) + E_g j\left(E_g^T v_n, t\right) + E_l j_l + E_i j_i = 0, \\ L_l \dfrac{d}{dt} j_l - E_l v_n = 0, \\ E_i^T v_n = 0. \end{cases} \tag{3.4}$$

Here the four incident matrices $[E_c, E_g, E_l, E_i]$ describe the topological connections of capacitive, conductive, inductive, and voltage-source elements. Introducing the state variable $x = [v_n, j_l, j_i]^T$, the above MNA formulation can be denoted shortly by a differential algebra equation (DAE) below:

$$F\left(x, \dot{x}, t\right) = \frac{d}{dt} q\left(x, t\right) + j\left(x, t\right) = 0. \tag{3.5}$$

3.1.2 New MNA with Nonvolatile State Variables

In order to handle the dynamic models of NVM device with nonvolatile state variables, one needs to develop a new MNA by adding nonvolatile state variables into the traditional MNA (for CMOS devices). As shown in Fig. 3.1b, these nonvolatile state variables, termed as s_m, determine the conductance of all NVM devices, termed as memductance (memory conductance) here, and therefore can be categorized as one new device branch. Note that the simulation time is directly related to the total number of state variables generated for the circuit. Compared to the traditional approaches with complex equivalent circuits, the introduction of memductance adds much fewer state variables by including nonvolatile state variables to represent dynamic effect of NVM device. Take memristor, for example; its equivalent circuit requires dozens of additional nodal voltages to characterize the circuit behavior, while only two nodal voltages and one nonelectrical state

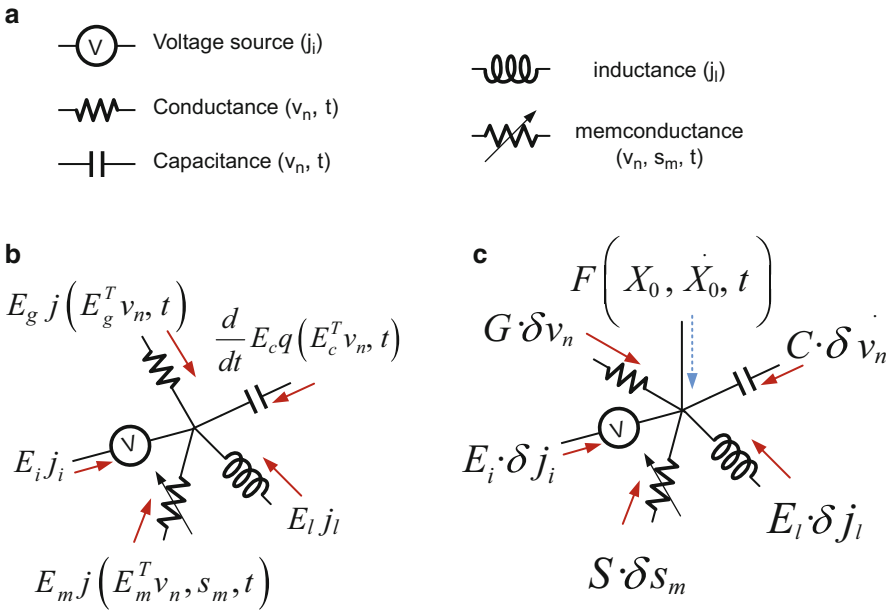


Fig. 3.1 New MNA with (**a**) components and state variables; (**b**) large-signal KCL; and (**c**) small-signal KCL

variable are required in our memductance approach. Therefore, the introduction of memductance greatly simplifies the model complexity and in turn largely reduces the verification and design cost.

One NVM device may require one or multiple nonvolatile state variables to accurately describe its dynamic behaviors. The corresponding incident matrix is termed as E_m, in which the nonvolatile device branch current is obtained as $E_m j(E_m^T v_n, s_m, t)$. As Fig. 3.1b shows, combined with branch currents from the traditional CMOS devices, a new KCL equation is formed by

$$\begin{cases} \dfrac{d}{dt} E_c q\left(E_c^T v_n, t\right) + E_m j\left(E_m^T v_n, s_m, t\right) + E_g j\left(E_g^T v_n, t\right) + E_l j_l + E_i j_i = 0, \\ L_l \dfrac{d}{dt} j_l - E_l v_n = 0, \\ E_i^T v_n = 0, \\ f(E_m^T v_n, s_m, t) + \dfrac{d}{dt} g(E_m^T v_n, s_m, t) = 0. \end{cases} \tag{3.6}$$

Here functions f and g are the additional state equations for nonvolatile devices with s_m. Moreover, with the new state variable vector $X = [v_n, j_i, j_l, s_m]^T$, the above new MNA can still be described by the DAE as Eq. (3.5).

We can further derive the Jacobian as generalized conductance G, capacitance C, and memductance S. The additional term of memductance S is introduced to describe the conductance of nonvolatile devices for the induced current change under the change of nonvolatile state s_m. At one biasing point X_0, the first-order derivative or Jacobian of the new DAE with respect to X is

$$\begin{cases} G = \left(E_g \dfrac{d}{d v_b^g} j\left(v_b^g, t\right) E_g^T + E_m \dfrac{d}{d v_b^m} j\left(v_b^m, s_m, t\right) E_m^T \right) ||_{X=X_0}, \\ C = \left(E_c \dfrac{d}{d v_b^c} q\left(v_b^c, t\right) E_c^T \right) ||_{X=X_0}, \\ S = \left(E_m \dfrac{d}{d s_m} j\left(v_n^m, s_m, t\right) E_m^T \right) ||_{X=X_0}, \end{cases} \tag{3.7}$$

where $v_b^g = E_g^T v_n, v_b^m = E_m^T v_n, v_b^c = E_c^T v_n$.

In addition, there are four additional Jacobian terms introduced from functions f and g due to the new state variable s_m for NVM device:

$$\begin{cases} G = \left(E_g \dfrac{d}{d v_b^g} j\left(v_b^g, t\right) E_g^T \right) ||_{X=X_0}, v_b^g = E_g^T v_n, \\ C = \left(E_c \dfrac{d}{d v_b^c} q\left(v_b^c, t\right) E_c^T \right) ||_{X=X_0}, v_b^c = E_c^T v_n, \\ S = \left(E_l \dfrac{d}{d \Phi_b^l} j\left(\Phi_b^l, t\right) E_l^T \right) ||_{X=X_0}, \Phi_b^l = E_l^T \Phi_n, \\ W = \left(E_m \dfrac{d}{d \Phi_b^m} q\left(\Phi_b^m, t\right) E_m^T \right) ||_{X=X_0}, \Phi_b^m = E_m^T \Phi_n. \end{cases} \tag{3.8}$$

The subscripts v and s denote the derivatives to nonvolatile device branch voltage v_b^m and its nonvolatile state variable s_m, respectively. The superscripts F and G correspond to functions f and g, which are current and charge terms for the KCL equation, respectively.

In addition, with the new small-signal current introduced by s_m as shown in Fig. 3.1c, the new linearized small-signal DAE becomes

$$\begin{cases} G \cdot \delta v_n + C \cdot \delta \dot{v_n} + E_i \cdot \delta j_i + E_l \cdot \delta j_l + S \cdot \delta s_m = -F\left(X_0, \dot{X}_0, t\right), \\ E_i^T \cdot \delta v_n = 0, \\ E_l^t \cdot \delta v_n = L_l \cdot \delta \dot{j}_l, \\ K_v^F \cdot \delta v_n + K_s^F \cdot \delta s_m + K_v^G \cdot \delta \dot{v_n} + K_s^G \cdot \delta \dot{s_m} = 0. \end{cases} \tag{3.9}$$

Therefore, we have the following matrix formed for linearized system equation with all Jacobian matrices $(G, C, L_l, S, K_v^F, K_s^F, K_v^G, K_s^G)$ and incident matrices (E_c, E_g, E_l, E_i):

$$\begin{bmatrix} G & E_i & E_l & S \\ -E_i & 0 & 0 & 0 \\ -E_l & 0 & 0 & 0 \\ K_v^F & 0 & 0 & K_s^F \end{bmatrix} \begin{bmatrix} \delta v_n \\ \delta j_i \\ \delta j_l \\ \delta s_m \end{bmatrix} + \begin{bmatrix} C & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & L_i & 0 \\ K_v^G & 0 & 0 & K_s^G \end{bmatrix} \begin{bmatrix} \delta \dot{v_n} \\ \delta \dot{j_i} \\ \delta \dot{j_l} \\ \delta \dot{s_m} \end{bmatrix} = -F(X_0, \dot{X}_0, t). \tag{3.10}$$

Note that both large-signal and small-signal system equations can be obtained when introducing nonvolatile state variables. Its implementation for NVM devices is shown in the following section.

3.2 ReRAM Device Model

3.2.1 Memristor

The memristor was theoretically predicted by Chua in 1976 [9], but not until nearly 30 years later was it first discovered in nanoscale devices at HP Labs [43]. The memristor is a two-terminal nonlinear passive electrical device which maintains a relationship between the time integrals of current and voltage and is regarded as the fourth passive element besides resistor, capacitor, and inductor. It has attracted numerous studies recently for a variety of purposes. For example, the crossbar-structured memristors are very promising for high-density nanoscale NVM [24, 50]. Also, hybrid CMOS-memristor-based reconfigurable computing has been demonstrated in [52]. In addition, memristors can be utilized to mimic the behavior of synapses in neuromorphic systems [25].

3.2.1.1 Nonvolatile State Identification

A comprehensive dynamic memristor model which takes charge-induced-drifting effect, slowdown effect, and strong-electric-field effect into consideration is illustrated in Fig. 3.2a with doping and undoping regions. The term doping ratio (w) is introduced to numerically represent the ratio of doping region (W) over the whole thickness of memristor film (D). As there is a large difference of resistivity in doping and undoping regions, the total resistance is changing with doping ratio. For instance, when $w = 1$, the memristor is at ON-state, where the resistance is a thousand times smaller than the resistance at OFF-state with $w = 0$.

The operating principle of memristor device can be summarized as charge-induced state (doping ratio w) shifting. As illustrated in Fig. 3.2a, the memristor can be seen as two regions in series, doping region and undoping region. The domain

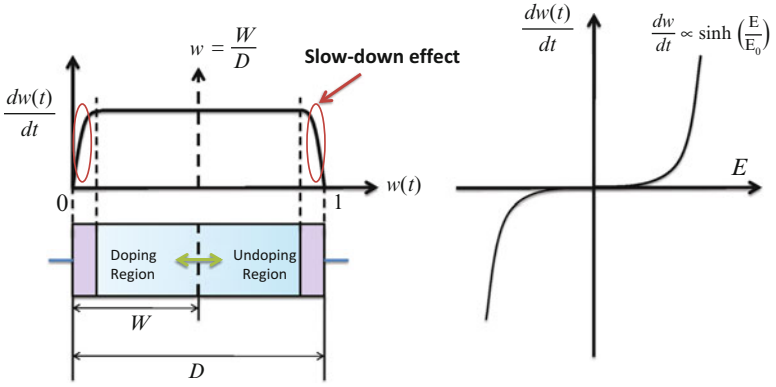


Fig. 3.2 Structure of memristor and nonlinear effects for dynamic model: (**a**) slowdown effect at boundary, (**b**) exponential relation between drift velocity and electric field

wall that separates two regions can be shifted by electric charge that flows through. When the positive charge flows towards the right-hand side of memristor junction, the domain wall shifts rightwards; thus, the doping ratio (w) increases and the resistance reduces accordingly and vice versa for the charge flowing in the reverse direction. Traditionally, this mechanism can be described as generalized voltage-controlled memristive system [9] with the following equations:

$$\begin{aligned} v(t) &= R\,(w, v)\, i\,(t) \\ \frac{dw(t)}{dt} &= f\,(w, v)\,, \end{aligned} \tag{3.11}$$

where R is the resistance of memristor and f is the explicit function of w and applied voltage v; dw/dt is the normalized drift velocity determined by w and external voltage v. Note that the complementary current-controlled memristive system can be defined similarly. The drifting velocity of memristor is a function of applied current that can be initially modeled in the following piecewise linear form [43]:

$$\frac{dw(t)}{dt} = \begin{cases} \mu \frac{R_{\text{on}}}{D^2} i\,(t) & \text{mode1 : normal operation} \\ 0 & \text{mode2 : } w = 0 \text{ or } 1 \\ 0 & \text{mode3 : } |v| < V_{\text{th}}. \end{cases} \tag{3.12}$$

In addition, strong-electric-field effect stands for the exponential increase in the drift velocity under a strong-electric-field condition (>1 MV/cm) [42]. It usually happens in the nanoscale devices such as the newly fabricated memristor at a scale of 20 nm. As illustrated in Fig. 3.2b, the phenomenon can be approximated by a sin$h$ function [42]:

Table 3.2 Definitions of variables used for memristor device

Variables	Definitions
M	Memristance, i.e., resistance of memristor
G	Memconductance, i.e., conductance of memristor
Φ	Magnetic flux
q	Charge
W	Length of the doped region
D	Thickness of memristor film
$w(t)$	Doping ratio (W/D), $w(t) \in [0, 1)$ (nonvolatile state variable)
$dw(t)/dt$	Normalized drift velocity
R_{on}	ON-state resistance
R_{off}	OFF-state resistance
$F(x)$	Window function defining the boundary condition of $w(t)$
μ	Mobility at small electric field
E_0	Characteristic field for a particular mobile atom in the crystal

$$\frac{dw(t)}{dt} = \frac{\mu E_0}{D} \sinh\left(\frac{E}{E_0}\right), \tag{3.13}$$

where $E = v(t)/D$ is the applied field strength and definitions of remaining variables are shown in Table 3.2.

Moreover, the slowdown effect of memristor is illustrated in Fig. 3.2a, when w is approaching the boundaries where $w = 0$ or 1 and the drift velocity drops quickly and is no longer linear with the current applied. This nonlinear dynamic effect can be characterized by either of the following window functions [3, 26]:

$$F(w) = 1 - (2w - 1)^{2p} \tag{3.14}$$

$$F(w) = 1 - [(w - u(-i(t)))^{2p}]. \tag{3.15}$$

Therefore, the complete nonlinear dynamic memristor model relevant to doping factor w considering the charge-induced-drifting effect, slowdown effect, and strong-electric-field effect can be obtained from Eqs. (3.11)–(3.15):

$$\frac{v(t)}{i(t)} = R_{\text{off}} - (R_{\text{off}} - R_{\text{on}})\, w(t),$$

$$\frac{dw(t)}{dt} = \frac{\mu E_0}{D} \sinh\left(\frac{v(t)}{D E_0}\right) F(w(t)). \tag{3.16}$$

Applying Eqs. (3.14)–(3.16) derived in the last section to the memristor dynamic function described in Eq. (3.10), we can obtain the required Jacobian terms as follows:

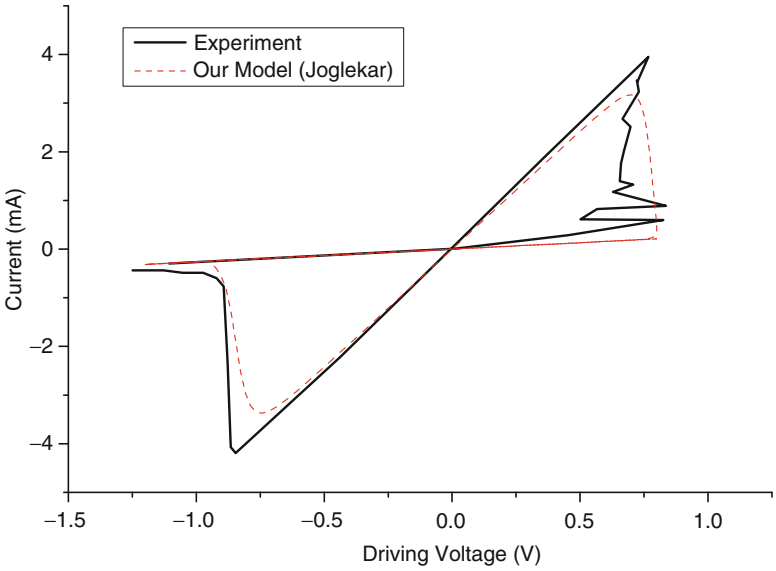


Fig. 3.3 Memristor device simulation with the Joglekar window function: I–V curve, which is consistent to the measured results reported in [43]

$$G = \begin{bmatrix} G & -G \\ -G & G \end{bmatrix}; S = \begin{bmatrix} v_n \times \frac{dG}{dw_m} \\ -v_n \times \frac{dG}{dw_m} \end{bmatrix}$$

$$K_v^F = \left[-c_1 c_2 F(w_m) \cosh(c_2 v_n) \; c_1 c_2 F(w_m) \cosh(c_2 v_n) \right];$$

$$K_s^F = -c_1 \sinh(c_2 v_n) \times \frac{dF(w_m)}{dw_m}; K_v^G = 0; K_s^G = 1,$$

where G is the conductance and $\frac{dG}{dw_m}$ is the nonvolatile state variable derivative of conductance; $c_1 = \frac{\mu E_0}{D}$, $c_2 = \frac{1}{DE_0}$ are constants; and $F(w)$ is the window function. Recall that the other parameters are defined in Table 3.2. Therefore, all Jacobian terms required for the linearized system equation (3.10) can be established, with which the memristor device simulation considering the nonvolatile state variables can be implemented in the SPICE-like simulator accordingly.

In order to verify the dynamic memristor model and our simulator, simulation to generate memristor hysteresis loop has been conducted to validate against experimental measurement in [43]. The device parameters are set as $d = 10\,\text{nm}$, $R_{\text{on}} = 200\,\Omega$, and $R_{\text{off}} = 3{,}800\,\Omega$; the initial resistance is set as $R_{\text{init}} = 3{,}790\,\Omega$, $\mu = 1 \times 10^{-17}\text{m}^2/(\text{V}\cdot\text{s})$, and $E_0 = 1\,\text{MV/cm}$; and the Joglekar window function is applied with $p = 2$. The dynamic behavior is investigated under the driving voltage of $v(t) = -0.2 + \sin(200\pi t)$. As shown in Fig. 3.3, simulation result can well capture the experiment results reported in [43].

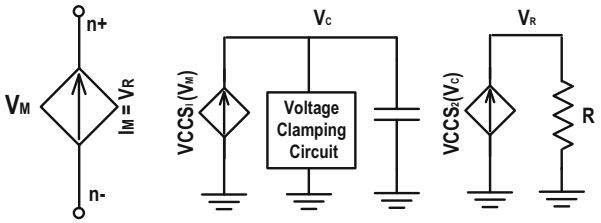


Fig. 3.4 Equivalent circuit of memristor model

3.2.1.2 Simulation Results

Current approach to describe a memristor element in circuit simulation is to utilize its equivalent circuit [20, 41]. However, very often they require the use of many additional circuit components to capture this nonlinear dynamics, hence leading to large complexity for large-scale circuit simulation. For instance, as depicted in Fig. 3.4 to implement an equivalent circuit of one memristor shown in [41], many circuit components have to be added, including a resistor, capacitor, diode, voltage-controlled current source, etc. In contrast, our approach only requires one nonvolatile state variable for a memristor.

The CPU runtime comparison between our nonvolatile state variable based simulation and equivalent circuit based approach is shown in Fig. 3.5. The equivalent circuit is based on [41]. The circuit test bench is the memristor crossbar [43] with varying number of memristor devices. For a 4-memristor-device simulation, the CPU time of our nonvolatile state variable based simulation is only half of the equivalent circuit based approach. Note that the time reduction ratio obtained from the experiment largely depends on the complexity of equivalent circuit and the number of memristor devices under study. With the increasing number of devices, the overall netlist complexity of equivalent circuit approach grows much faster than our approach. As such, this time reduction keeps increasing with the enlarged number of memristor devices. For a circuit with 1,024 memristor devices, the CPU runtime of our nonvolatile state variable based simulation is 40 times less compared to the equivalent circuit based simulation.

3.2.2 Conductive Bridge

In this section, the CBRAM device physics is illustrated with discussion of its nonvolatile states, and the hybrid CMOS-CBRAM circuit simulation is discussed as well.

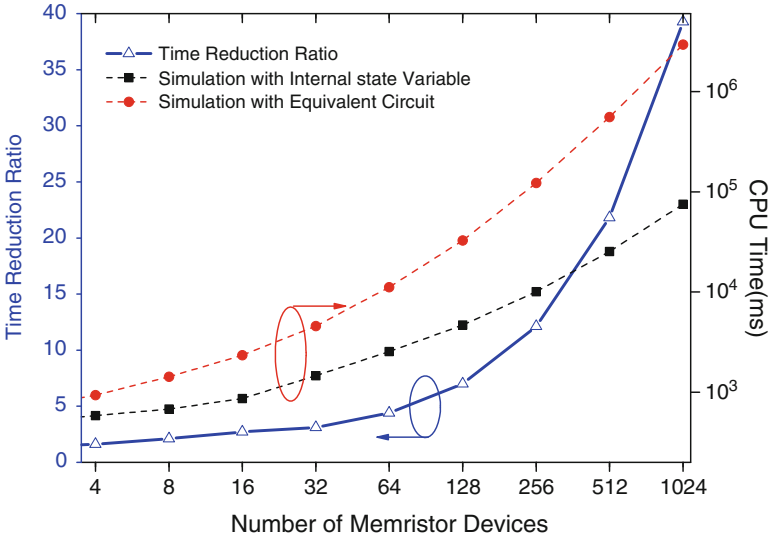


Fig. 3.5 CPU runtime comparison between our method and equivalent circuit based model for the memristor

3.2.2.1 Nonvolatile State Identification

CBRAM, also known as programmable metalization cell (PMC) [29, 38] or NanoBridge [27, 44], is an emerging two-terminal NVM device that can be fabricated by depositing an active anode layer, a solid-electrolyte layer, and an inert cathode layer, where the solid-electrolyte layer is sandwiched by the other two electrode layers. A variety of electrode and solid-electrolyte material combinations have been reported in the literature [16,17,39,40]. CBRAM-based memory has also been successfully fabricated in the chip level [10].

The physical working mechanism of the CBRAM device is illustrated in Fig. 3.6a, which can be summarized as the shape morphing of the conductive metal filament within the ambient-resistive solid electrolyte. When a positive bias voltage is applied, the conductive filament, modeled as a cone with fixed base radius, first grows vertically by accumulating metal ions until the two electrodes are bridged together. Then the cone begins to morph into a cylinder, which will eventually set the CBRAM into low-resistance or ON-state. Similarly, when a negative bias voltage is applied, the cylinder-shaped conductive filament dissolves into a cone with same base area, disconnects the two electrodes, shrinks vertically, and then turns the CBRAM in high-resistance or OFF-state. From the defined geometric variables shown in Fig. 3.6b, it can be observed that the variables h and r are capable of modeling the shape-morphing process, where detailed equations are given in [54].

We find that the angle θ from the side of the cone to the bottom can determine the shape of filament, when assuming that a cylinder is simply a cone whose apex

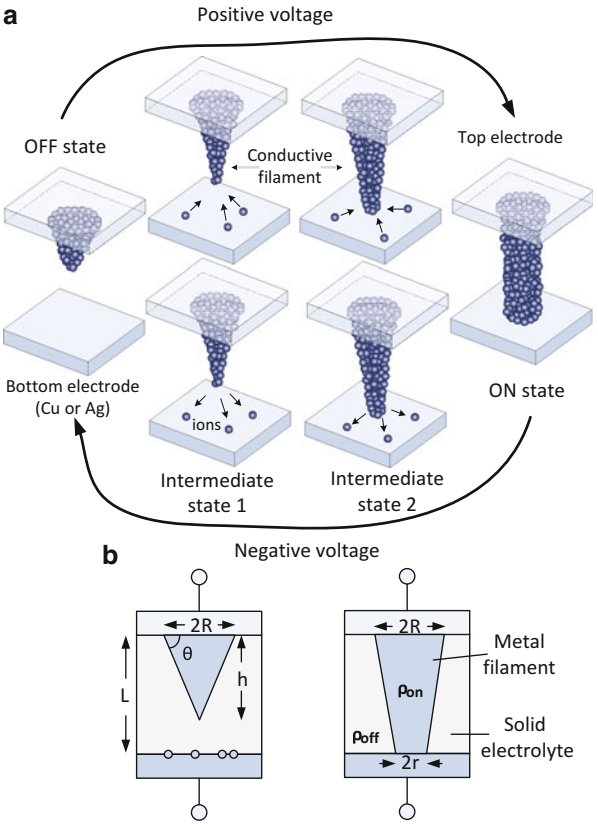


Fig. 3.6 (**a**) Working mechanism of CBRAM with the shape morphing of conductive filaments in several phases illustrated between ON-state and OFF-state; (**b**) cross section of CBRAM device with defined geometric variables

is at infinity. Thus the nonvolatile state variable s or $\tan\theta$, i.e., the height of cone in a projective perspective to base radius, is selected to determine the resistance of one CBRAM device. Hence s becomes a shape-determining state variable in which a large value of s indicates the filament more cylinder shaped while a small value for more cone shaped. As such, the number of state variable for each device can be reduced from two to one, which will greatly improve the simulation speed.

As such, by modifying the equations from [54], the dynamic behavior of CBRAM device based on variable s can be modeled as

$$\frac{ds}{dt} = \begin{cases} \frac{k_1}{R} \cdot \sinh\left(\frac{k_2 \cdot \mathbf{V}}{L + k_3 \cdot R \cdot s}\right) & s \le \frac{L}{R} \\ \frac{s^2 \cdot k_1}{L} \cdot \sinh(k_4 \cdot \mathbf{V}) & s > \frac{L}{R}, \end{cases} \tag{3.17}$$

where k_1, k_2, k_3, and k_4 are constants, and their detailed definitions can be found in [54]. Additionally, $\mathbf{V}$ is the applied voltage on the CBRAM device, L is the thickness of CBRAM, and R is the constant base radius.

The equivalent resistance of CBRAM with respect to variable s can be calculated by

$$\begin{cases} R_{\text{off}} = \dfrac{\rho_{\text{on}} \cdot s \cdot R + \rho_{\text{off}}(L - s \cdot R)}{\pi \cdot R^2} & s \leq \dfrac{L}{R} \\ R_{\text{on}} = \dfrac{\rho_{\text{on}} \cdot L}{\pi \cdot R \cdot (R - L/s)} & s > \dfrac{L}{R}, \end{cases} \tag{3.18}$$

where ρ_{on} and ρ_{off} are the conductive filament resistivity and nonconductive solid-electrolyte resistivity, respectively.

Two observations from Eqs. (3.17) and (3.18) can be drawn for the fast write speed of CBRAM. Firstly, there exists an exponential relation between the state changing speed ds/dt and the applied voltage $\mathbf{V}$. In the literature, CBRAMs with less than 50 ns write latency have been demonstrated under 1.5 V bias voltage [10, 14]. Secondly, the morphing steps between OFF-state and intermediate state 1 as well as that between ON-state and intermediate state 2 are much more time-consuming than that between intermediate states 1 and 2. Practically, by properly defining the intermediate states 1 and 2 and operating the devices within this range, the unnecessary time for morphing at two ends can be eliminated, thus significantly improving the write speed while still achieving acceptable *off/on* resistance ratio.

In addition, it can be observed that the switching speed indicated by Eq. (3.17) and also R_{on} and R_{off} indicated by Eq. (3.18) are all dependent only on the shape of internal conductive filament, rather than the external sizes of the upper and bottom contacts. As the filament size is significantly smaller than that of contacts, the CBRAM device exhibits a great potential of scalability. The scalability of the CBRAM device has been confirmed by [30], within which it has been demonstrated that the threshold voltage, R_{on}, and R_{off} are feature-size independent from μm to nm regions.

Combining Eqs. (3.17) and (3.18) together, we can reach the conclusion that the branch current of the CBRAM device can be decided by the conductive filament shape-determining variable s and also the applied voltage V. As such, the MNA that takes s as the additional new nonvolatile variable can be built into NVM SPICE and then solved by it.

As discussed previously, to perform transient analysis together with NVM devices, the linearized circuit Eq. (3.10) has to be established.

The new state variable vector $X = [v_n, j_i, j_l, s_m]^T$ contains nodal voltage, source current, inductor current, and nonvolatile state variable of NVM devices, respectively. $K_v^g, K_s^g, K_v^c, and K_s^c$ are the additional Jacobian terms introduced from NVM device equations. As such, for CBRAM memory circuit designs, our approach avoids the use of equivalent circuit components, which is not scaled

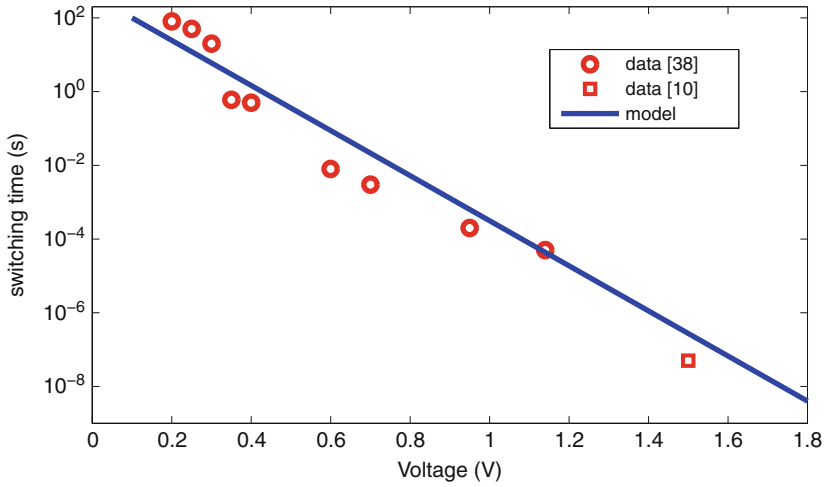


Fig. 3.7 CBRAM model validation against the published measurement data [10, 38]

with dependence on geometries and not scalable for large-scale memory design. Moreover, the reduction of the number of CBRAM state variables from two to only one will also greatly reduce the size of state matrix in Eq. (3.10).

Applying Eqs. (3.17) and (3.18) to (3.10), we can obtain the required Jacobian terms as follows:

$$G = \begin{bmatrix} g(s) & -g(s) \\ -g(s) & g(s) \end{bmatrix}, S = \begin{bmatrix} V \cdot \frac{dg(s)}{ds} \\ -V \cdot \frac{dg(s)}{ds} \end{bmatrix},$$

$$K_v^g = \begin{bmatrix} -\frac{df(V,s)}{dv} & \frac{df(V,s)}{dv} \end{bmatrix}, K_s^g = -\frac{df(V,s)}{ds},$$

$$K_v^c = 0, K_s^c = 1, \tag{3.19}$$

where $g(s) = \frac{1}{R}$ is the CBRAM conductance and $f(V,s) = \frac{ds}{dt}$ as indicated in Eq. (3.17). As such, the new MNA for CBRAM is formulated and can be directly applied in transient analysis in a SPICE-like simulator.

3.2.2.2 Simulation Results

The CBRAM device model parameters are based on [10]. Besides, 100 nm L_{pitch}, 1.8 V v_w, 0.9 V v_r, and 6.4 fF distributed C are utilized as settings. Figure 3.7 shows the validation of CBRAM device model against the published measurement data [10, 38]. It can be seen that the exponential relation between applied voltage and switching time is valid.

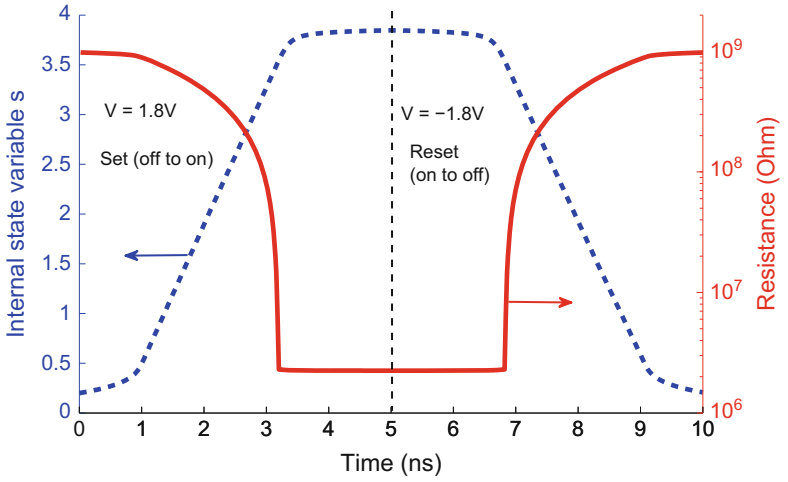


Fig. 3.8 Transient response of CBRAM device set and reset when v_w amplitude of 1.8 V is applied

The transient analysis for CBRAM switching is conducted in the NVM-SPICE simulator with results shown in Fig. 3.8. The model parameters k_1, k_2, k_3, and k_4 are based on [10] with 1.8 V voltage supply. 2 MΩ and 1 GΩ are assumed as thresholds for on and off states of CBRAM, respectively. A pitch size of 100 nm is assumed for the crossbar structure, with 50 nm × 50 nm cross-sectional area of nanowire made of copper. Multiple supply voltages are used, where 65 nm technology node with 1.2 V is assumed for CMOS logic and 1.8 V for CBRAM-crossbar operations. The CBRAM is initialized at OFF-state, and 1.8 V voltage is applied to set the device (off to on). The nonvolatile state variable s grows from 0.2 to around 4, indicating that the conductive filament first grows its height as a cone, reaches the top electrode, and then transforms towards the cylinder. This is also confirmed by the change of resistance from GΩ scale to MΩ scale. The CBRAM has been successfully set after around 5 ns, and then a −1.8 V voltage is applied to reset the device, where a reverse process can be observed.

3.3 Spintronics Device Model

3.3.1 Spin-Transfer Torque Magnetic Tunneling Junction

Based on spin-transfer torque effect, STT-RAM is seen as the second phase of magnetization-based NVM technology after the toggle MRAM. Since it has great scalability thanks to current-induced magnetization switch instead of external magnetic field, it has attracted a lot of attention. In the following, the nonvolatile state variables of spin-transfer torque magnetic tunneling junction (STT-MTJ) are identified based on the magnetization physics.

Table 3.3 Definitions of variables used for STT-MTJ device

Variables	Definitions
θ, ϕ	Shown in Fig. 3.9, azimuthal angles of magnetization orientation in x-z and x-y plane
R_H, R_L	Resistance values of antiparallel state and parallel state
ΔR_{GMR}	Difference between R_H and R_L
α	Damping constant
e	Electron charge
A	Area of STT-MTJ cross section
l_m	Thickness of oxide barrier
H^{ext}, H_k	External applied field and shape anisotropy field
M	Magnetization
M_s	Saturation magnetization of material
m, h	Normalized magnetization and effective magnetic field
η	Spin-transfer efficiency
$\hbar$	Reduced Planck constant
τ	Normalized time,$\tau = \gamma_0 M_s t$
τ_0	Inverse of attempt frequency (1 ns)
Δ	Thermal stability factor
I_c	Critical current for magnetization switching

3.3.1.1 Nonvolatile State Identification

In this section, the operation of STT-MTJ device is discussed. Then the nonvolatile state variables of STT-MTJ devices are identified for magnetization angles with the consideration of arbitrary driving condition. All variables used in this section are summarized in Table 3.3.

A typical STT-MTJ device structure appears as a sandwich with two ferromagnetic layers and one oxide barrier in between [4, 48]. One STT-MTJ device has two stable states: parallel state (P) or antiparallel state (AP), where the free layer magnetization is in the same or the opposite direction with hard axis (magnetization direction of fixed layer), respectively. One giant magnetoresistance (GMR) at AP state (R_H) is higher than the GMR at P state (R_L)[15]. The angle θ is one magnetization angle between the free layer and hard axis, and the GMR can be expressed as [1]

$$\begin{cases} R(\theta) = R_L + \dfrac{R_H - R_L}{2}(1 - \cos(\theta)) \\ = R_L + \dfrac{\Delta R_{\text{GMR}}}{2}(1 - \cos(\theta)). \end{cases} \tag{3.20}$$

It can be easily derived from Eq. (3.20) that $R(\theta = 0) = R_L$ at P state and $R(\theta = \pi) = R_H$ at AP state.

The operating principle of STT-MTJ device can be summarized as: the external current induces the state (P or AP) change. STT-MTJ is switched from P to AP

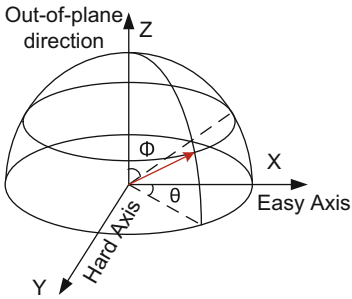


Fig. 3.9 Spherical coordinates with two magnetization angles: θ and ϕ

with sufficient forward-biased current and from AP to P with sufficient reverse-biased current. In order to model this mechanism accurately, we need to look into two dominant effects in the device physics of STT-MTJ: tunneling effect [4] and spin-transfer torque effect.

Tunneling effect of STT-MTJ can be understood as a parabolic relationship between the junction conductance and the applied voltage [4]. Tunneling effect normally becomes dominant when the applied voltage is relatively small such that it will not trigger the STT-MTJ to change state. This relation can be approximated by

$$\begin{cases} R_l(V) = \dfrac{R_{l0}}{1 + c_l V^2} \\ R_h(V) = \dfrac{R_{h0}}{1 + c_h V^2}, \end{cases} \tag{3.21}$$

where c_l and c_h are voltage-dependent coefficients for parallel state and antiparallel states, respectively.

As introduced in Chap. 2, spin-transfer torque effect is able to cause magnetization reversal in the free layer of STT-MTJ when the spin-polarized current is larger than the critical value I_c. Magnetization reversal is not an instantaneous process, and the switching time required (T_s) for magnetization reversal decreases exponentially with the current applied (I_0):

$$T_s = \tau_0 \exp\left(\Delta\left(1 - \frac{I_0}{I_c}\right)\right), \tag{3.22}$$

where $I_c = (2eA\alpha l_m M_s(H + H_k + 2\pi M_s))/\eta$, τ_0 is the switching time at $I_0 = I_c$, and the definitions of remaining variables are in Table 3.3. Note that T_s from Eq. (3.22) is the minimum pulse width requirement for certain I_0. For the write operation in STT-MTJ device-based memory, one must make sure that the constant current larger than I_c is applied within a period of T_s.

To better understand T_s is important for modeling STT-MTJ devices. This requires the analysis for the dynamic behavior of STT-MTJ device under arbitrary

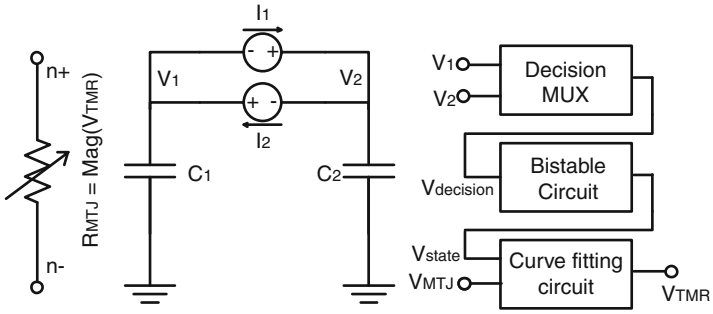


Fig. 3.10 Equivalent circuit of STT-MTJ model

driving condition. As described in Chap. 2, the Landau–Lifshitz–Gilbert equation (LLG) is deployed for this purpose, and the dynamics can be expressed as

$$\theta = \theta_0 \mathrm{Exp}\left(-\frac{t}{t_0}\right) \cdot \cos(\phi) \tag{3.23}$$

$$\omega = \frac{d\phi}{dt} = k_1\sqrt{k_2 - (k_3 - k_4 I)^2}, \tag{3.24}$$

where θ_0 is the initial value of θ, slightly tilted from the stable x or $-x$ directions; t_0 is procession time constant; ω is the angular speed of ϕ; k_1 to k_4 are the magnetic parameters with detailed explanation in Chap. 2; and I is the spin current that causes the magnetization precession.

3.3.1.2 Simulation Results

The 1T-1MTJ structure STT-RAM array with transient analysis for a write operation is set as the test bench circuit. The circuit netlist is described in two versions which only differ in terms of STT-MTJ model, one using the equivalent circuit (Fig. 3.10) based SPICE macromodel in [18] and one using the intrinsic physics-based model in NVM SPICE. Necessary modifications are made to port the HSPICE subcircuit netlist provided by [18] compatible with Berkeley SPICE-styled NVM SPICE. The simulation duration is set to 20 ns with time step of 0.1 ns.

Table 3.4 shows the runtime comparison of the two different simulation approaches for different array sizes. It can be observed that the simulation using NVM-SPICE intrinsic physics-based model is 10–117× faster than the equivalent circuit approach. The advantage will be even larger when the array size increases. For typical memory array size of hundreds by hundreds, the speedup can be expected more than 100×.

Table 3.4 Simulation time comparison for STT-RAM array using different simulation approaches (unit in second)

Array size	Macromodel in [18]	NVM-SPICE	Speedup
8×8	2.522	0.257	10×
16×16	98.131	1.87	52×
32×32	1119.99	11.533	97×
64×64	22188.8	189	117×

3.3.2 Topological Insulator

Topological insulator (TI) is a recently discovered nanodevice whose bulk acts as an insulator but surface behaves as metal. As state information in a TI device is conducted by ordered spins, it draws tremendous interest for ultra-low-power computing.

The scattering, in which electrons deviate from their trajectory resulting in dissipation, is the fundamental reason of power consumption. As such, instead of manipulating electrons, it is envisioned that spintronics [47, 51] can be developed by controlling pure spin current and spin accumulation for the ultra-low-power information storage, transmission, and processing. Topological insulator (TI) [2, 12, 21, 33] is recently found as a means for realizing spintronics not only for the underpinning of computer logic in microprocessors but also for hard disk, written, read, or rewritten with significantly reduced power.

Materials like Bi_2Se_3, Bi_2Te_3, and Sb_2Te_3 are experimentally observed as three-dimensional (3D) TI devices [6, 22, 53]. TI has insulating bandgap state in the bulk and gap-less metallic state at surface. The gap-less metallic state at surface in a 3D TI is very robust under perturbations. Due to the strong spin–orbit coupling, electrons in TI move along their surface into two distinguished directions without scattering according to their spins. This works similarly to that vehicles can move in two opposite directions at two sides of a highway without disturbing each other. Without scattering, power and further thermal dissipation can be significantly reduced in this type of devices.

The quantum spin Hall effect (QSHE) behavior has been observed for TI devices [5, 7, 55], where a quantized topological surface states form the Landau levels in the presence of external magnetic field. This becomes the foundation for its application in magnetic NVM design [13]. However, no design exploration is performed before on the potential use of TI in NVM. From a modeling perspective, due to its unique device properties, there is no physical model to fully capture the states in TI devices. From design perspective, there is no NVM design based on TI on how to apply the external field for read and write operation. There is a need to develop a design platform for TI-based NVM design, including SPICE-like simulation of TI device and its agreeing memory design.

Table 3.5 Definitions of variables used for TI device

Variables	Definitions
θ, ϕ	Shown in Fig. 3.9, azimuthal angles of magnetization orientation in x-z and x-y plane
α	Damping constant
H_{eff}	Effective field
H^e, H_k	External applied field and shape anisotropy field
M	Magnetization
M_s	Saturation magnetization of material
m, h	Normalized magnetization and effective magnetic field
coeff	Coefficient between read current and produced external magnetic field
σ_{H}, V_H	Quantum Hall conductance and quantum Hall voltage

3.3.2.1 Nonvolatile State Identification

An STT-MTJ device is found with state depending on the magnetization angle. Similarly, a topological insulator (TI) device also has nonconventional electrical states to describe. In this section, the working mechanism of TI is discussed with additional state variable identified as well. Then, similar to the BSIM model for MOSFET, the agreeing device model to stamp a TI device in SPICE is described. The deployed variables and terms are summarized in Table 3.5.

A typical TI-based memory device is built by a two-layer structure with ferromagnetic layer on the top and topological insulator layer on the bottom as shown in Fig. 3.11a. The TI device has four terminals, with two controlling terminals along x-axis and two Hall terminals attached to the lateral sides. As discussed in [13], one bit can be then stored by the perpendicular magnetization of the ferromagnetic layer. Programming a bit requires an external magnetic field whose field strength is exceeding the coercivity of ferromagnetic layer. Under the magnetic field of the ferromagnetic layer, the topological insulator exhibits a quantum Hall conductance as shown in Fig. 3.11b. It can be observed that the sign of the quantum Hall conductance is determined by the magnetic field orientation; thus, the stored bit can be read out by detecting the sign of the quantum Hall voltage:

$$V_H = \frac{I_R}{\sigma_{\text{H}}}, \tag{3.25}$$

where I_R is the applied read-current pulse along the x-axis.

The Hall conductance σ_{H} can be calculated by [37]

$$\sigma_{\text{H}} = \frac{e^2}{\hbar} \int \frac{d^2\mathbf{k}}{(2\pi)^2} (f_c - f_v)(\mathbf{k}) \Omega_z(\mathbf{k}), \tag{3.26}$$

where the $\mathbf{k}$ is the wave vector, Ω_z the Berry curvature, and f_c and f_v the Fermi–Dirac distributions of conduction band and valence band, respectively.

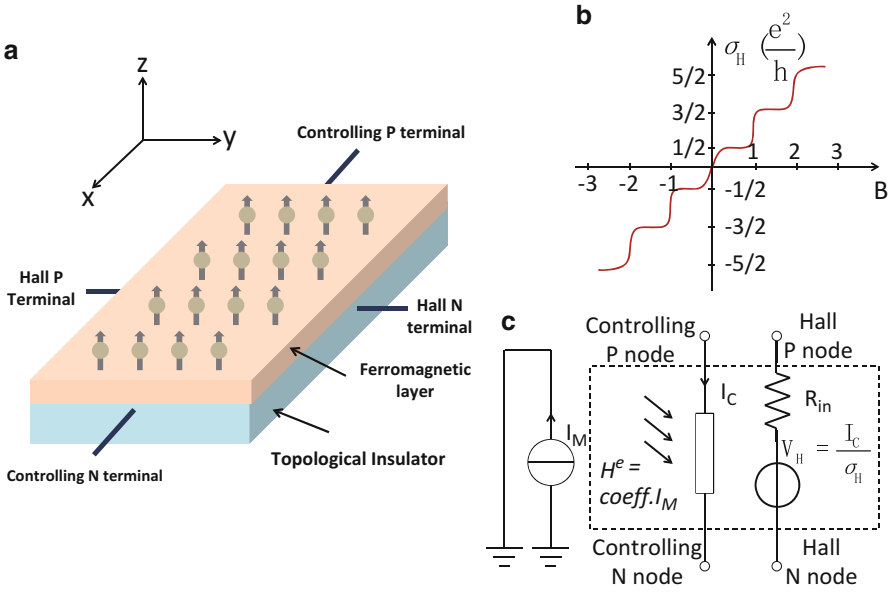


Fig. 3.11 (**a**) Device structure; (**b**) schematic diagram of quantum Hall conductance; and (**c**) abstracted equivalent device circuit model

It can be seen that the Hall conductance σ_{H} is a function of bandgap, Fermi level, and temperature. When the bandgap $\Delta \gg k_B T$, Eq. (3.26) becomes

$$\sigma_{\mathrm{H}} \approx \frac{e^2}{2h} sgn(M). \tag{3.27}$$

The h here is the Planck constant, e is the charge of electron, and $sgn(M)$ is the orientation of magnetization. As such, the quantum Hall conductance is a constant approximately equals to 19.4 μS. Note that the TI device is insensitive to disorder, imperfection, and cell geometry, which ensures a constant readout voltage even in the presence of perturbations.

From Eq. (3.27), the quantum Hall voltage can be regarded as a current-controlled voltage source with the coefficient of σ_{H}, as shown in Fig. 3.11c. Note that this equivalent quantum Hall voltage source has very limited driving ability, and a large internal resistance R_{in} is introduced for an accurate modeling.

More importantly, in order to model the dynamic behavior of the TI device during programming procedure, the magnetization trajectory needs to be studied, which is described by the normalized LLG at macroscale:

$$\frac{dm}{d\tau} = -m \times h + \alpha \left(m \times \frac{dm}{d\tau} \right), \tag{3.28}$$

where the normalized effective field h equals to $-\frac{\delta \epsilon}{\delta m}$.

The ϵ is the normalized energy density:

$$\epsilon = \frac{1}{2}h_k(1 - m_z^2) - m \cdot h^e, \tag{3.29}$$

where the two energy density contributions are associated with anisotropy field and the external field, respectively.

Note that the required external magnetic field for device programming is generated by a current, as shown in Fig. 3.11c. There exists a coefficient between the read current and the produced field:

$$H^e = \text{coeff} \cdot I_M. \tag{3.30}$$

We assume the external magnetic field only have a perpendicular component along the easy axis z. Thus, normalized effective field $h = \text{coeff} \cdot I_M / M_s + h_k \cdot m_z$ can be obtained.

The solution of Eq. (3.28) can be interpreted as the change of the normalized magnetization (m) over time. The normalized magnetization m can be expressed in spherical coordinates with variables θ and ϕ as shown in Fig. 3.9. The dynamic behavior described by θ and ϕ can be finally calculated by LLG:

$$\theta = \theta_0 \text{Exp}\left(-\frac{t}{t_0}\right) \cdot \cos(\phi) \tag{3.31}$$

$$\omega = \frac{d\phi}{dt} = k_c \sqrt{k_d - \left(\alpha h_k - \alpha h_x^e\right)^2}, \tag{3.32}$$

where θ_0 is the initial value of θ, slightly tilted from the stable z or $-z$ directions; t_0 is the procession time constant; ω is the angular speed of θ; $k_c = \gamma_0 \cdot M_s$ is the product of gyromagnetic ratio and saturation magnetization; and $k_d \approx \frac{H_k}{M_s}$. Definitions of the remaining variables are also shown in Table 3.5.

The new state variable vector $X = [v_n, j_i, j_l, s_m]^T$ contains nodal voltage, source current, inductor current, and the new state variable s_m (magnetization angles θ and ϕ) for a TI device, respectively. We assume that (i) the conductance along the x-axis shows only weak dependency on the new state variables θ and ϕ and (ii) the magnetization is only subject to the external field, and hence $S \approx 0$, $K_v^f \approx 0$, and $K_v^g \approx 0$.

Applying Eqs. (3.17) and (3.18) to (3.10), we can obtain the required Jacobian terms as follows:

$$G = \begin{bmatrix} \sigma_x & -\sigma_x \\ -\sigma_x & \sigma_x \end{bmatrix};$$

$$K_s^f = \begin{bmatrix} 1 & -\frac{df(\phi_m,t)}{d\phi_m} \\ 0 & 0 \end{bmatrix}; K_s^g = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix},$$

where $f(\phi_m, t)$ is the right-hand side of Eq. (3.34) and σ_x is the conductance along the x-axis provided as a model parameter. Due to the non-scattering property of TI surface, an extremely high σ_x can be expected, which will contribute to an ultra-low-power consumption.

Besides the magnetization dynamics, the derived MNA also has to model the quantum Hall voltage readout behavior. As discussed in the last subsection, the quantum Hall voltage can be modeled as a current-controlled voltage source, with coefficient as $\frac{1}{\sigma_{\mathrm{H}}}$. As such, applying Eq. (3.25), the incident matrix for the quantum Hall voltage source can be obtained as

$$E_i = \begin{bmatrix} 1 \\ -1 \end{bmatrix}.$$

Meanwhile, I_c/σ_{H} has to be added to the corresponding right-hand side of Eq. (3.10), where I_c is the read current flowing through x-axis, and σ_{H} can be calculated by Eq. (3.26). Therefore, all Jacobian terms required for the linearized system Eq. (3.10) can be established, with which the TI device simulation considering the new state variables can be implemented in the SPICE-like simulator accordingly.

3.3.2.2 Simulation Results

In this section, the design of TI device-based NVM is discussed. Firstly, a memory cell circuit is proposed with addressability achieved. Then, a memory array design is further illustrated with word read and write operations.

3.3.2.3 Memory Cell Circuit Design

Inspired by the toggle MRAM design [11], the cell structure of a TI device for NVM application is proposed in Fig. 3.12. To program a selected TI cell, both word-write-line (WWL) and bit-write-line (BWL) produce half of the required external magnetic field H_e.

In order to achieve addressability, the amplitude of H_e is subject to

$$H_e/2 \leq H_c \leq H_e, \tag{3.33}$$

where the H_c is the coercivity of the magnetic surface. Here the currents along WWL and BWL to produce $H_e/2$ are defined as programming current I_{PW} and I_{PB}, respectively. The upward I_{PB} and leftward I_{PW} are defined as positive. If both I_{PW} and I_{PB} are applied, there exists an external magnetic field of H_e that exceeds the coercive field of ferromagnetic layer to program a cell. If I_{PW} is applied only or I_{PB} is applied only, the cell is exposed to a magnetic field of $H_e/2$, which is insufficient to switch the magnetization. A zero magnetic field, when no current is

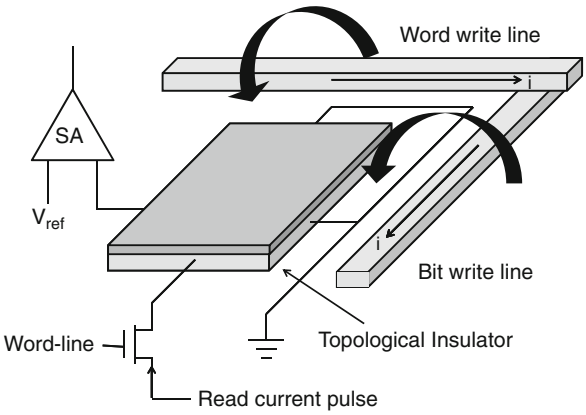


Fig. 3.12 Cell circuit of topological insulator based memory

applied for both WWL and BWL, is also not able to switch the magnetization. Thus, the cell addressability can be achieved under programming operation. Moreover, to read a cell, the cell is first selected by the signal through word-line, and then a read-current pulse I_R is applied through bit-line. The corresponding quantum Hall voltage V_H will be either positive or negative depending on the status of magnetization orientation. Therefore, by comparing V_H with reference voltage, the bit stored by magnetization orientation can be known.

3.3.2.4 Memory Array Circuit Design

Figure 3.13 shows a 4×4 memory array design based on the TI memory cell. It can be observed that it is impossible to write 1 bit and 0 bit for a word at one time since they require opposite I_{PW} and I_{PB} directions. Thus, to write a word, writing 1 bit or 0 bit is conducted separately. To write 1001 for $word0$ for instance, WWL0, BWL0, and BWL3 are applied with positive I_{PW} and I_{PB}, while other WWLs and BWLs are applied with no current. As discussed in last subsection, the addressability allows only $bit0$ and $bit3$ to be programmed into 1, while other bits remain their status. Then WWL0, BWL1, and BWL2 are applied with negative I_{PW} and I_{PB} so that $bit1$ and $bit2$ are programmed into 0. As such, 1001 is successfully written to $word0$ in two separate steps. Because there exists an inversely proportional relationship between current-induced magnetic field and distance in space and also due to the programming threshold for the magnetic field, the operations on target cells will not interfere their neighboring cells. To read out a word, its corresponding word-line is selected, and a current pulse I_R is applied for all bit-lines. As a result, the V_H for each bit can be interpreted by sense amplifiers for each bit-line.

A TI-based NVM design platform is developed. As CMOS circuits are still required as interfacing part, hybrid CMOS-TI simulation is required. Similar to a BSIM model for MOSFET, the physical model of TI device is implemented into a SPICE-like simulator NGspice [35]. Based on the developed SPICE-like simulator,

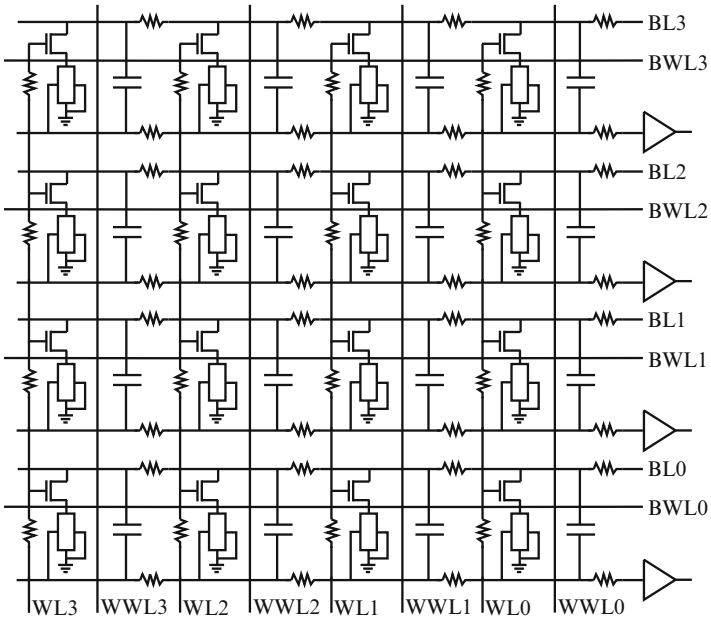


Fig. 3.13 One 4 × 4 topological insulator based NVM array

a number of experiments have been conducted for TI-based NVM designs. In the following numerical experiments, we first validate our physical model against the reported measurement data [28]. Then, we design the cell and array memory circuits of TI and verify them by the SPICE-like simulator under both read and write operations. Finally, we compare the performance of read and write power with the other emerging NVM devices. All numerical experiments are implemented in C and are conducted on the same workstation with Intel Core i5 CPU and 8G RAM.

3.3.2.5 Validation of Dynamic Effect with New State Variable of TI Device

In order to validate the dynamic model of magnetization for TI device, we show the device-level simulation results based on soft ferromagnet material $Ni_{60}Fe_{40}$ with parameters extracted from the measurement in [28]. The lateral dimension of TI device is 0.8 by 1.6 μm^2, with ferromagnet layer thickness of 5 nm. The saturation magnetization is set to 740 kA/M, the shape anisotropy field is 1.72 kA/M, the damping constant is 0.01, and the current-to-magnet coefficient is 10^6. Same parameters are assumed in the following experiments for consistency.

Figure 3.14 shows the magnetization switching time versus the applied external magnetic field amplitude. We can see the results produced by our simulator fit well with the measured data reported in [28]. A nearly inversely proportional relationship can be observed between switching time and magnetic field. For example, in order to

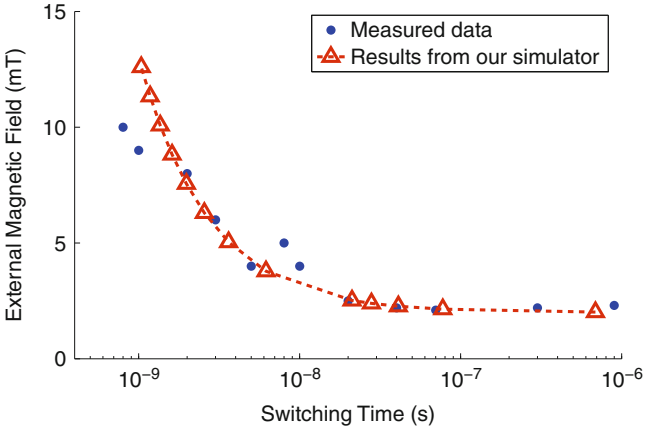


Fig. 3.14 Validation of the switching time and external magnetic field relationship for magnetization dynamics

achieve a faster memory programming speed, a stronger magnetic field is desirable. Moreover, a switching threshold can be observed from Fig. 3.14. So only when the external magnetic field exceeds a certain strength, approximately 2 mT observed from figure, can the magnetization be switched. This is also consistent with [28], where the magnetic coercivity is reported as 2 mT.

As such, the magnetization dynamics has been verified to validate for our developed TI device simulator. Note that for TI with other materials as ferromagnetic layer, parameters of saturation magnetization, shape anisotropy field, and damping constant need to be specified for the simulator in a model file.

3.3.2.6 Hybrid Simulation with CMOS for TI-Based Memory Cell

In order to investigate the performance of TI device-based memory, transient analysis is conducted for one TI memory cell circuit as illustrated in Sect. 3.3.2.3. The technology node is 65 nm for CMOS part, and VDD is 1.2 V. The aspect ratios of transistors are set to 3. The read-current pulse is set to be 1μA with a pulse width of 5 ns. The programming current of both WWL I_{PW} and BWL I_{PB} is generated by their respective current sources. As discussed above, strong external magnetic field is desirable for a fast programming speed. Moreover, in order to achieve memory cell addressability, however, the field strength should be subject to Eq. (3.33). So in this work, the magnetic coercivity H_c is 2 mT, and H_e is designed at 3 mT. Parameters for ferromagnetic layer are the same as last section.

Figure 3.15 shows the dynamic response when using the new state variables. Both programming currents I_{PW} and I_{PB} are applied at zero second. It can be observed that θ starts to deviate from the original angle once H_e is applied. Its maximum deviation increases exponentially with time before the reversal happens.

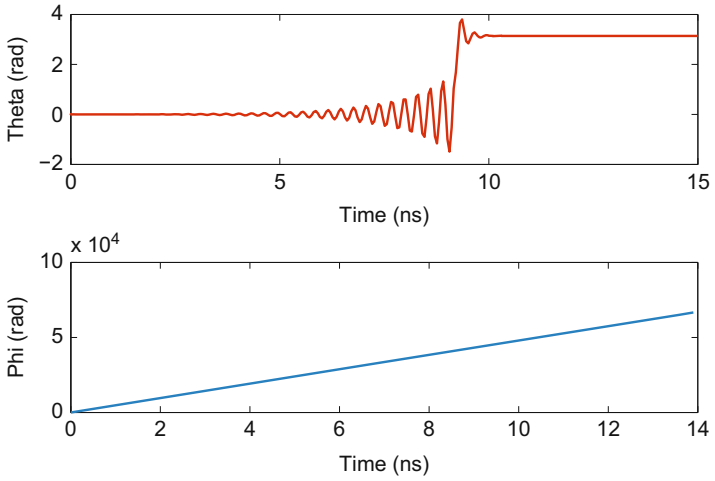


Fig. 3.15 Dynamic response of topological insulator new state variables

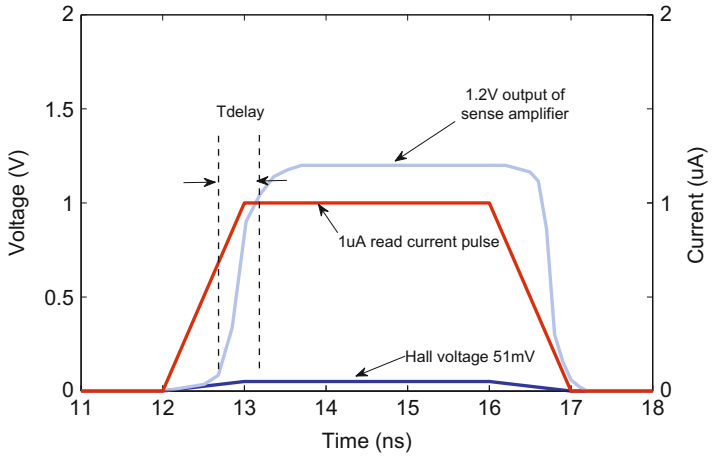


Fig. 3.16 Read operation for topological insulator based memory cell

The fluctuation decays very fast after reversal since the shape anisotropy field alters its direction to strengthen H_e, and then the device enters into the other stable state. The whole process indicates that a write latency, i.e., switching time, of about 10 ns is achieved under 3 mT magnetic field. The continuously increasing ϕ shows a very fast circulating frequency in the x-y plane, which causes θ to fluctuate in the same frequency.

The simulation result for the cell circuit readout operation is shown in Fig. 3.16. Firstly, the word-line signal is set to logic-1 before the readout operation. Then at the time of 12 ns, a read-current pulse with amplitude of 1μA is applied through

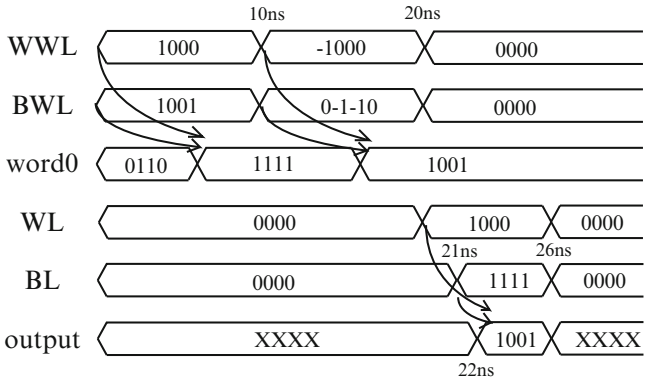


Fig. 3.17 Timing diagram for word write and read operations in a 4 × 4 TI array

bit-line. The quantum Hall voltage responds simultaneously and is fed into the sense amplifier (SA) later. After a time of T_{delay}, which is the combination of bit-line delay and SA sensing delay, the SA outputs the stored logic. Note that the T_{delay} depends on the array size and quantum Hall voltage amplitude in practice, and the read-current pulse width is set to 5 ns to secure a successful readout operation.

3.3.2.7 Performance Comparison of TI-Based Memory Array

Here we further investigate the TI-based memory array design as shown in Fig. 3.13. The circuit settings are same with Sect. 3.3.2.6. The transient analysis conducted is to write 1001 to $word0$ as discussed. The $word0$ is initialized 0110 for better illustration, the other bits are initialized all zeros.

Figure 3.17 shows the timing diagram of write and read operations. Note that the positive and negative currents are indicated by 1 and −1 in this figure. It can be observed that the write operation is executed in two phases. First, the first bit and fourth bit of $word0$ are written to 1 in the first 10 ns. From 10 ns to 20 ns, the second and third bits are written to 0. To read out $word0$, $word0$ is selected through word-line, and bit-lines of all bits are applied with read-current pulse. A correct readout can be observed. It has also been verified that all the other bits remain their initial values.

Table 3.6 shows the comparison of performance for different memory technologies. Compared with the other emerging NVMs, such as phase-change memory (PCM), STT-MTJ, and ReRAM, the TI device-based memory shows both faster read and write latencies. It is also noticed that the TI-based memory exhibits in several orders of magnitude lower write energy. The write energy of TI is calculated by Eq. (3.29) with device dimension. Actually, the read energy of TI is also extremely low due to the non-scattering property. Simulation shows a read energy of 1.2e-17J/bit. The data for other memory technologies is extracted from ITRS 2011 [23].

Table 3.6 Performance comparison for different memory technologies

Memory technology	Write latency (ns)	Read latency (ns)	Write energy (J/bit)
SRAM	0.2	0.2	5e-16
DRAM	2–10	2–10	4e-15
PCM	100	12	6e-12
STT-MTJ	35	35	2.5e-12
FeRAM	40	60	3e-14
TI	20	5	1e-17

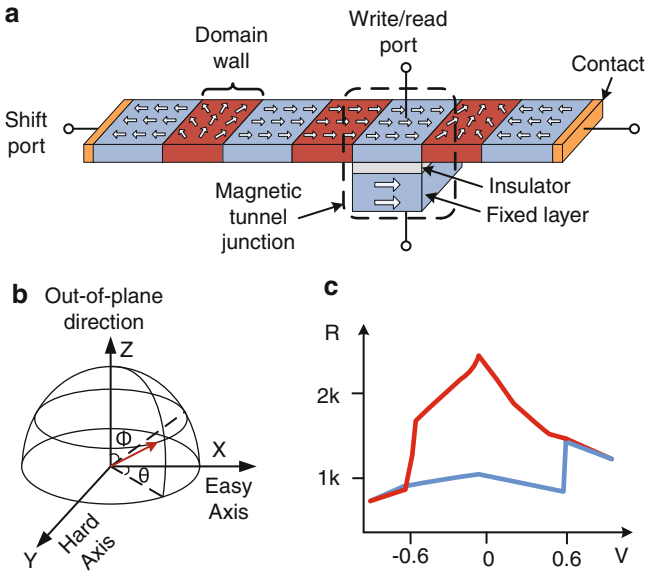


Fig. 3.18 (**a**) Schematic of domain-wall nanowire structure with access port and shift port; (**b**) magnetization of the free layer in spherical coordinates with defined magnetization angles; and (**c**) typical R–V curve for MTJ

3.3.3 Racetrack and Domain Wall

Domain-wall nanowire, also known as racetrack memory [36, 45, 46], is a newly introduced NVM device in which multiple bits of information are stored in single ferromagnetic nanowire. As shown in Fig. 3.18a, each bit is denoted by the leftward or rightward magnetization direction, and adjacent bits are separated by domain walls. By applying a current through the shift port at the two ends of the nanowire, all the domain walls will move left or right at the same velocity, while the domain width of each bit remains unchanged; thus, the stored information is preserved. Such a tape-like operation will shift all the bits similarly like a shift register.

In order to access the information stored in the domains, a strongly magnetized ferromagnetic layer is placed at desired position of the ferromagnetic nanowire and

is separated by an insulator layer. Such a sandwich-like structure forms a magnetic tunnel junction (MTJ), through which the stored information can be accessed. In the following, the write, read, and shift operations are modeled, respectively.

3.3.3.1 Magnetization Reversal

The write access can be modeled as the magnetization reversal of MTJ free layer, i.e., the target domain of the nanowire. Note that the dynamics of magnetization reversal can be described by the precession of normalized magnetization m or state variables θ and ϕ in spherical coordinates as shown in Fig. 3.18b. The spin-current-induced magnetization dynamics described by θ and ϕ is given by

$$\theta = \theta_0 \mathrm{Exp}\left(-\frac{t}{t_0}\right) \cdot \cos(\phi) \tag{3.34}$$

$$\omega = \frac{d\phi}{dt} = k_1\sqrt{k_2 - (k_3 - k_4 I)^2}, \tag{3.35}$$

where θ_0 is the initial value of θ, slightly tilted from the stable x or $-x$ directions; t_0 is the procession time constant; ω is the angular speed of ϕ; k_1 to k_4 are the magnetic parameters with detailed explanation in Chap. 2; and I is the spin current that causes the magnetization precession.

3.3.3.2 MTJ Resistance

A typical R–V curve for MTJ is shown in Fig. 3.18c with two regions: GMR region and tunneling region. Depending on the alignment of magnetization directions of the fixed layer and free layer, parallel or antiparallel, the MTJ exhibits two resistance values R_l and R_h. As such, the general MTJ resistance can be calculated by the GMR effect:

$$R(\theta_u, \theta_b) = R_{l0} + \frac{R_{h0} - R_{l0}}{2}(1 - \cos(\theta_u - \theta_b)), \tag{3.36}$$

where θ_u and θ_b are the magnetization angles of upper free layer and bottom fixed layer and R_{l0} and R_{h0} are the MTJ resistances when the applied voltage is subtle. When the applied voltage increases, there exists tunneling effect caused by voltage-dependent resistance roll-off:

$$\begin{cases} R_l(V) = \dfrac{R_{l0}}{1 + c_l V^2} \\ R_h(V) = \dfrac{R_{h0}}{1 + c_h V^2}, \end{cases} \tag{3.37}$$

where c_l and c_h are the voltage-dependent coefficients for parallel state and antiparallel state, respectively.

3.3.3.3 Domain-Wall Propagation

Like a shift register, the domain-wall nanowire shifts in a digital manner; thus, it could be digitalized and modeled in the unit of domains, in which a bit is stored. Note that except the bit in the MTJ, the other bits denoted by the magnetization directions are only affected by their adjacent bits. In other words, the magnetization of each bit is controlled by the magnetization in adjacent domains. Inspired by this, we present a magnetization-controlled magnetization (MCM) device-based behavioral model for domain-wall nanowires. Unlike the current-controlled and voltage-controlled devices, the control in MCM device needs to be triggered by the rising edge of one SHF signal, which can be formulated as

$$\begin{aligned}\theta &= f(T_{\mathrm{sl}}, \theta_r, T_{\mathrm{sr}}, \theta_l, \theta_c) \\ &= T_{\mathrm{sl}}\theta_r + T_{\mathrm{sr}}\theta_l + \overline{T}_{\mathrm{sl}}\overline{T}_{\mathrm{sr}}\theta_0\end{aligned} \tag{3.38}$$

in which T_{sl} and T_{sr} are the shift-left and shift-right commands; θ_r and θ_l are the magnetization angles in the right adjacent cell and left adjacent cell, respectively; and θ_c is the current state before the trigger signal. This describes that the θ-state will change when triggered and will remain state if no shift signal is issued.

For the bit in MTJ, the applied voltage for spin-based read and write will also determine the θ-state as discussed previously. Therefore, we have

$$\theta = f(T_{\mathrm{sl}}, \theta_r, T_{\mathrm{sr}}, \theta_l, \theta_0) + g(Vp, Vn, \theta_c), \tag{3.39}$$

where Vp and Vn are the MTJ positive and negative nodal voltages and $g(Vp, Vn, \theta_0)$ is the additional term that combines Eqs. (3.34)–(3.37).

In addition, the domain-wall propagation velocity can be mimicked by the SHF-request frequency. The link between the SHF-request frequency and the propagation velocity is experimentally observed by current–velocity relation [8]:

$$v = k(J - J_0), \tag{3.40}$$

where J is the injected current density and J_0 is the critical current density.

By combining Eqs. (3.34)–(3.39) together, with the magnetization angles θ and ϕ as nonvolatile state variables other than electrical voltages and currents, one can fully describe the behaviors of the domain-wall nanowire device, where each domain is modeled as the proposed MCM device. As such, the MNA can be built in NVM SPICE to verify circuit designs by domain-wall nanowire devices.

3.4 Phase-Change Device Model

PCM device is an emerging NVM technology based on the phase-changing technology of chalcogenide materials such as Ge_2 $SbTe_5$ [31], of which the crystal structure can be thermally switched between two different stable states: crystalline

and amorphous. PCM exploits the unique behavior of chalcogenide material as the resistance in amorphous state is significantly larger than that in crystalline state,; therefore, the resistance of two states is utilized to denote stored data and can be switched bidirectionally by thermal approach.

3.4.1 Nonvolatile State Identification

Typically, PCM device has a mushroom structure as demonstrated in Fig. 1.25. This structure makes the heat dissipation rate at top surface much higher than that of bottom surface. As such, the thermal activation of phase changing always begins from the bottom side. Note that the resistance of PCM device is mainly determined by the active region shown in Fig. 1.25.

The operation of PCM device can be summarized as thermally activated state (crystallizing and amorphousizing) change. As illustrated in Fig. 1.25, a narrow pulse of temperature surge above the melting temperature (T_m) will RESET the active region to the amorphous state, and a long pulse of temperature surge between melting temperature (T_m) and crystallization temperature (T_X) will SET the active region to the crystalline state. The dynamic effects required to describe the behavior of PCM device is discussed in Appendix C with details. As discussed, we choose crystallization ratio C_X and active region temperature T_M as nonvolatile state variables to describe the dynamic behavior of PCM device.

Two dominant effects have to be considered to describe the dynamic behavior of PCM effects: thermal effect and crystallization kinetics [32]. Thermal effect can be subdivided into two different processes: heat generation and heat dissipation. The heat generation power is mainly contributed by the Joule heat, which can be represented as

$$W_j = I_M \times V_M, \tag{3.41}$$

where I_M and V_M are current and voltage of one PCM device. The Joule heat generated will be dissipated through the surrounding media of active region. The heat dissipation power can be simplified as

$$W_d = k_e \cdot \bigtriangledown T, \tag{3.42}$$

where k_e is the effective thermal conductivity of surrounding material and $\bigtriangledown T$ is the temperature gradient between active region and ambient environment. As such, the temperature of active region (T_M) will increase when $W_j > W_d$ and will decrease when $W_j < W_d$. This state of thermal effect can be represented by an integration function of the generated net heat:

$$T_M = \int \frac{W_j - W_d}{C \cdot V} dt, \tag{3.43}$$

where C is the specific thermal capacity and V is the volume of active region.

Table 3.7 Definitions of variables used for phase-change device

Variables	Definitions
T_M	Active region temperature of chalcogenide material
K_0	Frequency factor
K_B	Boltzmann's constant
E_a	Activation energy
T_m	Melt temperature
T_X	Crystallization temperature
n	Avrami exponent

Moreover, crystallization kinetics is actuated when the temperature of active region is above crystallization temperature (T_X). Such a state change can be described by the Johnson–Mehl–Avrami–Kolmogorov (JMAK) equations [32]:

$$\begin{cases} C_X = 1 - \exp(-Kt^n) \\ K = K_0 \exp\left(\dfrac{-E_a}{K_B T_M}\right), \end{cases} \tag{3.44}$$

where C_X is the crystallization ratio; $n \in [1, 4]$ is the Avrami exponent; and K is the effective crystallization rate determined by the temperature of active region (T_M). The definitions of remaining parameters are given in Table 3.7.

When the temperature of active region is higher than T_m, the chalcogenide material melts in the active region, and C_X is reset to zero. As such, the total active region resistance of PCM device (R_M) can be approximated as a function of C_X:

$$\begin{cases} \dfrac{1}{R_M} = \dfrac{C_X}{R_\mathrm{SET}} + \dfrac{1 - C_X}{R_\mathrm{RESET}} & \text{when } V_M < V_\mathrm{th}, \\ R_M = R_B & \text{when } V_M \geq V_\mathrm{th}, \end{cases} \tag{3.45}$$

where R_SET and R_RESET are the PCM device resistance at crystalline and amorphous states, respectively. V_th is the breakdown threshold voltage and R_B is the breakdown-state resistance.

Similar to other NVM devices, the new state variable vector is introduced as $X = [v_n, j_i, j_l, C_X, T_X]^T$, and the Jacobian terms G, S, K_v^F, K_s^F, K_v^G, and K_s^G in Eq. (3.5) for PCM device are then obtained by

$$G = \begin{bmatrix} g(C_X, T_M) & -g(C_X, T_M) \\ -g(C_X, T_M) & g(C_X, T_M) \end{bmatrix}, S = \begin{bmatrix} V \cdot \frac{dg(C_X, T_M)}{dT_M} & V \cdot \frac{dg(C_X, T_M)}{dC_X} \\ -V \cdot \frac{dg(C_X, T_M)}{dT_M} & -V \cdot \frac{dg(C_X, T_M)}{dC_X} \end{bmatrix},$$

$$K_v^F = \begin{bmatrix} -\frac{dh(T_M, v_n, t)}{dv_b} & \frac{dh(T_M, v_n, t)}{dv_b} \\ 0 & 0 \end{bmatrix}, K_s^F = \begin{bmatrix} 1 & 0 \\ -\frac{df(T_M, v_n, t)}{dT_M m} & 1 \end{bmatrix},$$

$$K_v^G = 0, K_s^G = 0, \tag{3.46}$$

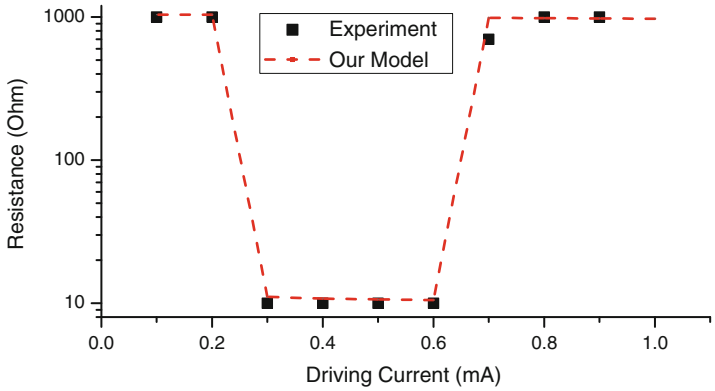


Fig. 3.19 PCM device simulation: R–I curve, which is consistent to the measured results reported in [49]

where g is the conductance and $\frac{dg(C_X,T_M)}{dT_M}$, $\frac{dg(T_M,v_n,t)}{dT_M m}$ are the temperature and crystalline fraction derivatives; $\frac{dh(T_M,v_n,t)}{dv_b}$ and $\frac{df(T_M,v_n,t)}{dT_M m}$ are derived from Eqs. (3.43) and (3.44), respectively. Note that $h(T_M, v_n, t)$ is the function at the right-hand side of Eq. (3.43), and $f(T_M, v_n, t)$ is the function at the right-hand side of Eq. (3.44). Recall that the other parameters are defined in Table 3.7.

3.4.2 Simulation Results

To further validate the PCM model, the device-level simulation results are compared to the experiment results shown in [49] under the same device parameters. The crystallization and melt temperature (T_X and T_m) are 155 °C and 620 °C, respectively. The crystalline state resistance (R_{SET}) is 10 KΩ and the amorphous state resistance (R_{RESET}) is 1 MΩ . We select the activation energy (E_a) to be 2.24 eV for a typical chalcogenide material $Ge_2Sb_2Te_5$ [31], the Avrami exponent (n) is 1.5, and the frequency factor is set to 1.2e7.

Figure 3.19 shows the resulting resistance of PCM device after a 300 ns current pulse, of which the amplitude is changed from 0.1 to 1 mA. We can see that our model fits well with the experiment results reported in [49]. Moreover, we can see the allowable current range for a 300 ns current pulse to change the PCM into crystalline state is between 0.3 and 0.6 mA. Out of this range the PCM remains in the amorphous state.

The dynamic responses of nonvolatile state variables, temperature and crystalline fraction of the active region under SET and RESET operations, are illustrated in Fig. 3.20. We can see that a 0.3 mA current pulse of 400 ns raises the temperature of the active region well above the crystallization temperature and the active region gradually changes from amorphous to crystalline state. Then a 0.65 mA current

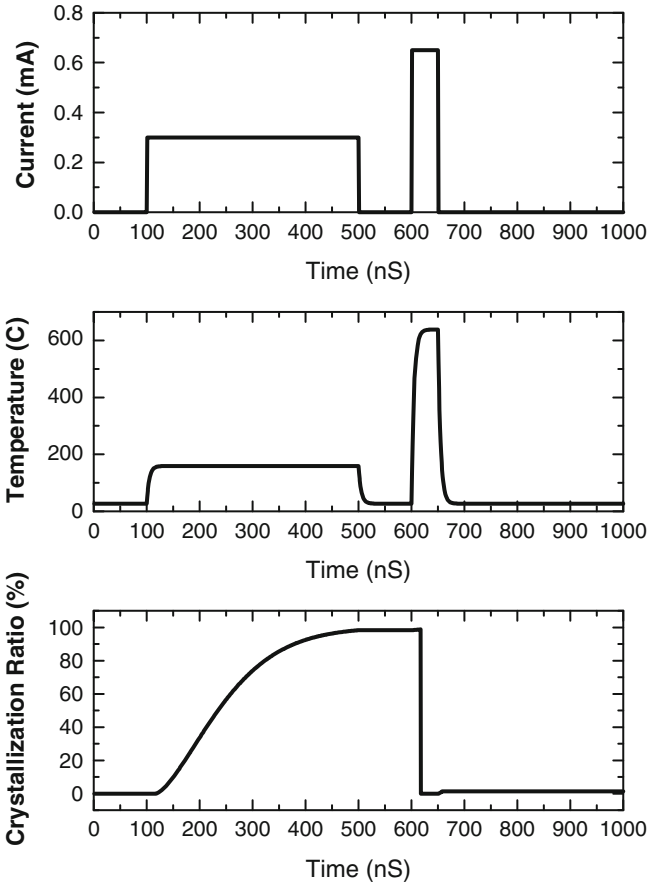


Fig. 3.20 Dynamic responses of nonvolatile state variables of PCM device under current driving SET and RESET operations

pulse of 50 ns increases the temperature of the active region higher than the melt temperature of 620 °C, and then the active region changes back to amorphous state. Actually, the storage temperature of PCM device is limited by the crystallization temperature. If the PCM memory is placed in the environment higher than the crystallization temperature, all the data stored will be corrupted.

An equivalent circuit of PCM device [49] is shown in Fig. 3.21 with dozens of circuit elements required to realize the same dynamics mentioned above. This introduces large overhead for large-scale circuit simulation. With the increasing size of memory, the overall netlist complexity of equivalent circuit approach grows much faster than our approach. As such, the simulation time reduction keeps increasing with the enlarged size of memory, as shown in Fig. 3.22. For a 512-bit memory, compared to the equivalent circuit based simulation, the CPU time of our nonvolatile state variable based simulation is 69 times smaller.

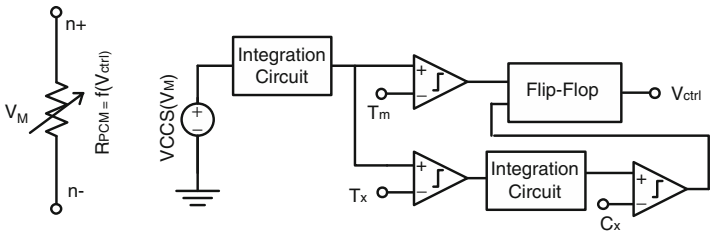


Fig. 3.21 Equivalent circuit of phase-change device model

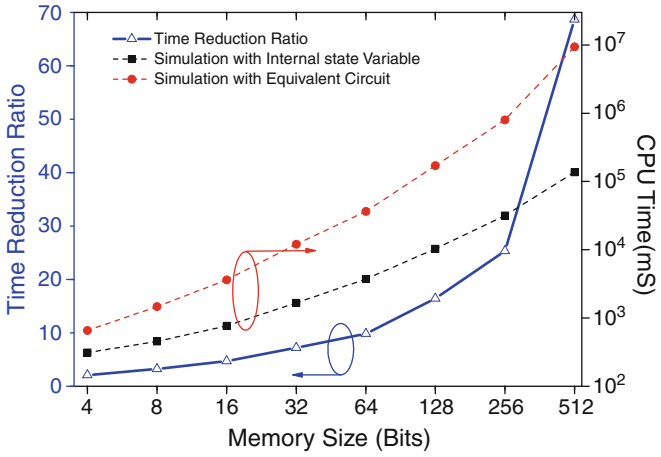


Fig. 3.22 CPU runtime comparison between our method and equivalent circuit based model for phase-change device

References

1. Baibich MN, Broto J, Fert A, Van Dau FN, Petroff F, Etienne P, Creuzet G, Friederich A, Chazelas J (1988) Giant magnetoresistance of (001) fe/(001) cr magnetic superlattices. Phys Rev Lett 61(21):2472
2. Bernevig BA, Hughes TL, Zhang SC (2006) Quantum spin hall effect and topological phase transition in hgte quantum wells. Science 314(5806):1757–1761
3. Biolek Z, Biolek D, Biolkova V (2009) Spice model of memristor with nonlinear dopant drift. Radioengineering 18(2):210–214
4. Brinkman W, Dynes R, Rowell J (1970) Tunneling conductance of asymmetrical barriers. J Appl Phys 41(5):1915–1921
5. Brüne C, Liu C, Novik E, Hankiewicz E, Buhmann H, Chen Y, Qi X, Shen Z, Zhang S, Molenkamp L (2011) Quantum hall effect from the topological surface states of strained bulk hgte. Phys Rev Lett 106(12):126,803
6. Chen Y, Analytis J, Chu JH, Liu Z, Mo SK, Qi XL, Zhang H, Lu D, Dai X, Fang Z et al (2009) Experimental realization of a three-dimensional topological insulator, bi2te3. Science 325(5937):178–181
7. Cheng P, Song C, Zhang T, Zhang Y, Wang Y, Jia JF, Wang J, Wang Y, Zhu BF, Chen X et al (2010) Landau quantization of topological surface states in bi_ {2} se_ {3}. Phys Rev Lett 105(7):076,801

8. Chiba D, Yamada G, Koyama T, Ueda K, Tanigawa H, Fukami S, Suzuki T, Ohshima N, Ishiwata N, Nakatani Y et al (2010) Control of multiple magnetic domain walls by current in a co/ni nano-wire. Appl Phys Expr 3(7):3004
9. Chua L (1971) Memristor-the missing circuit element. IEEE Trans Circuit Theor 18(5):507–519
10. Dietrich S, Angerbauer M, Ivanov M, Gogl D, Hoenigschmid H, Kund M, Liaw C, Markert M, Symanczyk R, Altimime L et al (2007) A nonvolatile 2-mbit cbram memory core featuring advanced read and program control. IEEE J Solid-State Circuits 42(4):839–845
11. Engel B, Akerman J, Butcher B, Dave R, DeHerrera M, Durlam M, Grynkewich G, Janesky J, Pietambaram S, Rizzo N et al (2005) A 4-mb toggle mram based on a novel bit and switching method. IEEE Trans Magn 41(1):132–136
12. Fu L, Kane CL, Mele EJ (2007) Topological insulators in three dimensions. Phys Rev Lett 98(10):106,803
13. Fujita T, Jalil MBA, Tan SG (2011) Topological insulator cell for memory and magnetic sensor applications. Appl Phys Expr 4(9):094,201. doi:10.7567/APEX.4.094201. http://apex.jsap.jp/link?APEX/4/094201/
14. Gopalan C, Ma Y, Gallo T, Wang J, Runnion E, Saenz J, Koushan F, Blanchard P, Hollmer S (2011) Demonstration of conductive bridging random access memory (cbram) in logic cmos process. Solid-State Electron 58(1):54–61
15. Grünberg P, Schreiber R, Pang Y, Brodsky M, Sowers H (1986) Layered magnetic structures: Evidence for antiferromagnetic coupling of fe layers across cr interlayers. Phys Rev Lett 57(19):2442
16. Guan W, Long S, Liu Q, Liu M, Wang W (2008) Nonpolar nonvolatile resistive switching in cu doped ZrO2. IEEE Electron Device Lett 29(5):434–437
17. Haemori M, Nagata T, Chikyow T (2009) Impact of cu electrode on switching behavior in a cu/hfo2/pt structure and resultant cu ion diffusion. Appl Phys Expr 2(6):1401
18. Harms JD, Ebrahimi F, Yao X, Wang JP (2010) Spice macromodel of spin-transfer torque-operated magnetic tunnel junctions. IEEE Trans Electron Dev 57(6):1425–1430
19. Ho CW, Ruehli A, Brennan P (1975) The modified nodal approach to network analysis. IEEE Trans Circ Syst 22(6):504–509
20. Ho Y, Huang GM, Li P (2011) Dynamical properties and design analysis for nonvolatile memristor memories. IEEE Trans Circ Syst I Regular Pap 58(4):724–736
21. Hsieh D, Qian D, Wray L, Xia Y, Hor YS, Cava R, Hasan MZ (2008) A topological dirac insulator in a quantum spin hall phase. Nature 452(7190):970–974
22. Hsieh D, Xia Y, Qian D, Wray L, Dil J, Meier F, Osterwalder J, Patthey L, Checkelsky J, Ong N et al (2009) A tunable topological insulator in the spin helical dirac transport regime. Nature 460(7259):1101–1105
23. ITRS (2010) International technology roadmap of semiconductor. http://www.itrs.net
24. Jo SH, Kim KH, Lu W (2009) High-density crossbar arrays based on a si memristive system. Nano Lett 9(2):870–874
25. Jo SH, Chang T, Ebong I, Bhadviya BB, Mazumder P, Lu W (2010) Nanoscale memristor device as synapse in neuromorphic systems. Nano Lett 10(4):1297–1301
26. Joglekar YN, Wolf SJ (2009) The elusive memristor: properties of basic electrical circuits. Eur J Phys 30(4):661
27. Kaeriyama S, Sakamoto T, Sunamura H, Mizuno M, Kawaura H, Hasegawa T, Terabe K, Nakayama T, Aono M (2005) A nonvolatile programmable solid-electrolyte nanometer switch. IEEE J Solid-State Circ 40(1):168–176
28. Koch R, Deak J, Abraham D, Trouilloud P, Altman R, Lu Y, Gallagher W, Scheuerlein R, Roche K, Parkin S (1998) Magnetization reversal in micron-sized magnetic thin films. Phys Rev Lett 81(20):4512
29. Kozicki MN, Balakrishnan M, Gopalan C, Ratnakumar C, Mitkova M (2005) Programmable metallization cell memory based on ag-ge-s and cu-ge-s solid electrolytes. In: IEEE non-volatile memory technology symposium, 2005, p 7

30. Kund M, Beitel G, Pinnow CU, Rohr T, Schumann J, Symanczyk R, Ufert KD, Muller G (2005) Conductive bridging ram (cbram): an emerging non-volatile memory technology scalable to sub 20nm. In: IEEE international electron devices meeting, 2005. IEDM technical digest, pp 754–757
31. Lee BS, Abelson JR, Bishop SG, Kang DH, Cheong Bk, Kim KB (2005) Investigation of the optical and electronic properties of ge2sb2te5 phase change material in its amorphous, cubic, and hexagonal phases. J Appl Phys 97(9):093,509–093,509
32. McHenry M, Johnson F, Okumura H, Ohkubo T, Ramanan V, Laughlin D (2003) The kinetics of nanocrystallization and microstructural observations in finemet, nanoperm and hitperm nanocomposite magnetic materials. Scripta Mater 48(7):881–887
33. Moore JE (2010) The birth of topological insulators. Nature 464(7286):194–198
34. Nagel LW, Pederson DO (1973) SPICE: simulation program with integrated circuit emphasis. Electronics Research Laboratory, College of Engineering, University of California
35. Nenzi P, Holger V (2010) Ngspice users manual. http://www.itrs.net
36. Parkin SS, Hayashi M, Thomas L (2008) Magnetic domain-wall racetrack memory. Science 320(5873):190–194
37. Qi XL, Wu YS, Zhang SC (2006) Topological quantization of the spin hall effect in two-dimensional paramagnetic semiconductors. Phys Rev B 74(8):085,308
38. Russo U, Kamalanathan D, Ielmini D, Lacaita AL, Kozicki MN (2009a) Study of multilevel programming in programmable metallization cell (pmc) memory. IEEE Trans Electron Dev 56(5):1040–1047
39. Sakamoto T, Lister K, Banno N, Hasegawa T, Terabe K, Aono M (2007) Electronic transport in ta2o5 resistive switch. Appl Phys Lett 91(9):092,110–092,110
40. Schindler C, Thermadam SP, Waser R, Kozicki MN (2007) Bipolar and unipolar resistive switching in cu-doped sio2. IEEE Trans Electron Dev 54(10):2762–2768
41. Shin S, Kim K, Kang SM (2010) Compact models for memristors based on charge-flux constitutive relationships. IEEE Trans Comput Aid Des Integr Circ Syst 29(4):590–598
42. Strukov DB, Williams RS (2009) Exponential ionic drift: fast switching and low volatility ofáthin-film memristors. Appl Phys A 94(3):515–519
43. Strukov DB, Snider GS, Stewart DR, Williams RS (2008) The missing memristor found. Nature 453(7191):80–83
44. Tada M, Sakamoto T, Banno N, Aono M, Hada H, Kasai N (2010) Nonvolatile crossbar switch using tiox/tasioy solid electrolyte. IEEE Trans Electron Dev 57(8):1987–1995
45. Thomas L, Yang SH, Ryu KS, Hughes B, Rettner C, Wang DS, Tsai CH, Shen KH, Parkin SS (2011) Racetrack memory: a high-performance, low-cost, non-volatile memory based on magnetic domain walls. In: 2011 IEEE international electron devices meeting (IEDM), pp 24–2
46. Venkatesan R, Kozhikkottu V, Augustine C, Raychowdhury A, Roy K, Raghunathan A (2012) Tapecache: a high density, energy efficient cache based on domain wall memory. In: Proceedings of the 2012 ACM/IEEE international symposium on Low power electronics and design, ACM, pp 185–190
47. Wang KL, Zhao Z, Khitun A (2008) Spintronics for nanoelectronics and nanosystems. Thin Solid Films 517(1):184–190
48. Wang X, Zhu W, Siegert M, Dimitrov D (2009) Spin torque induced magnetization switching variations. IEEE Trans Magn 45(4):2038–2041
49. Wei X, Shi L, Walia R, Chong T, Zhao R, Miao X, Quek B (2006) Hspice macromodel of pcram for binary and multilevel storage. IEEE Trans Electron Dev 53(1):56–62
50. Williams R (2008) How we found the missing memristor. IEEE Spectr 45(12):28–35
51. Wolf S, Awschalom D, Buhrman R, Daughton J, Von Molnar S, Roukes M, Chtchelkanova AY, Treger D (2001) Spintronics: a spin-based electronics vision for the future. Science 294(5546):1488–1495
52. Xia Q, Robinett W, Cumbie MW, Banerjee N, Cardinali TJ, Yang JJ, Wu W, Li X, Tong WM, Strukov DB et al (2009) Memristor-cmos hybrid integrated circuits for reconfigurable logic. Nano Lett 9(10):3640–3645

53. Xia Y, Qian D, Hsieh D, Wray L, Pal A, Lin H, Bansil A, Grauer D, Hor Y, Cava R, et al (2009) Observation of a large-gap topological-insulator class with a single dirac cone on the surface. Nat Phys 5(6):398–402
54. Yu S, Wong HS (2011) Compact modeling of conducting-bridge random-access memory (cbram). IEEE Trans Electron Dev 58(5):1352–1360
55. Zyuzin A, Burkov A (2011) Thin topological insulator film in a perpendicular magnetic field. Phys Rev B 83(19):195,413

Chapter 4
Nonvolatile Circuit Design

Abstract Memory design is commonly composed of two parts: the data arrays and the peripheral circuits. The data array is essentially a two-dimensional expansion of memory cells repetitively, which determines the way to retrieve data from particular cells in the array with limited I/O interface. The peripheral circuits mainly include many levels of decoders as well as readout sense amplifiers. Due to the use of nonelectrical states of emerging nonvolatile memory devices, new cells structures as well as agreeing readout circuits are needed for their unique read and write operations with performance evaluation. In this chapter, three different memory cell designs, crossbar structure for ReRAM, 1T1R structure for STT-RAM, and tape-like structure for domain-wall nanowire, are discussed with the agreeing readout circuits illustrated. Their performance models are presented as well if they are different from traditional designs.

Keywords Nonvolatile memory design • Nonvolatile logic • Analog learning circuit design

4.1 Memory and Readout Circuit

4.1.1 Crossbar Resistive Memory

Crossbar memory, also known as cross-point memory, structure has been popular ever since the advent of memristor array in crossbar structure fabricated by HP Labs [32]. Crossbar structure is mostly associated with resistive-RAM (RRAM) devices with nonlinear I–V characterization, memristor, and CBRAM, for instance, so that the half selection scheme can be applied. This leads to the most noticeable feature of crossbar, that is, unlike the conventional 1T1C/1T1R structure, access transistors are not required for every cell. In this section, we will introduce the crossbar-based memory design.

H. Yu and Y. Wang, *Design Exploration of Emerging Nano-scale Non-volatile Memory*,
DOI 10.1007/978-1-4939-0551-5_4,

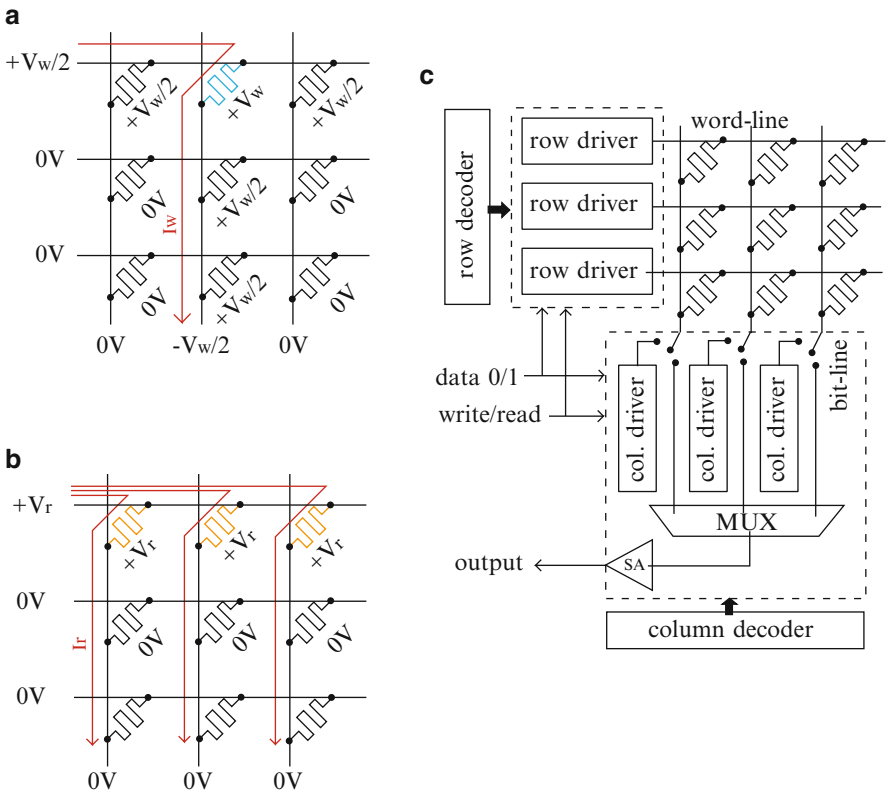


Fig. 4.1 Crossbar operations and peripheral circuits: (**a**) write operation, (**b**) read operation, and (**c**) operation of peripheral circuit

4.1.1.1 Crossbar Memory Array

Figure 4.1 illustrates the design for CBRAM-crossbar structure. Compared to the 1T1R structure, the crossbar structure has two major advantages. Firstly, an extremely high integration density can be realized due to the small pitch size of nano-bars [12, 32]. Secondly, the crossbar structure can be stacked on top of the active transistor layer in a 3D fashion [2, 11, 27], which further reduces the area overhead and improves communication bandwidth. Figure 4.2a shows the approach to fabricate the CBRAM-crossbar structure within the interconnect layers, where the CBRAM devices are deposited at the bottom of the copper vias [11]. Another approach of CBRAM-crossbar fabrication, shown in Fig. 4.2b, is to stack the crossbar structure on top of the interconnect layer, which incurs the least modifications to the conventional CMOS process [27].

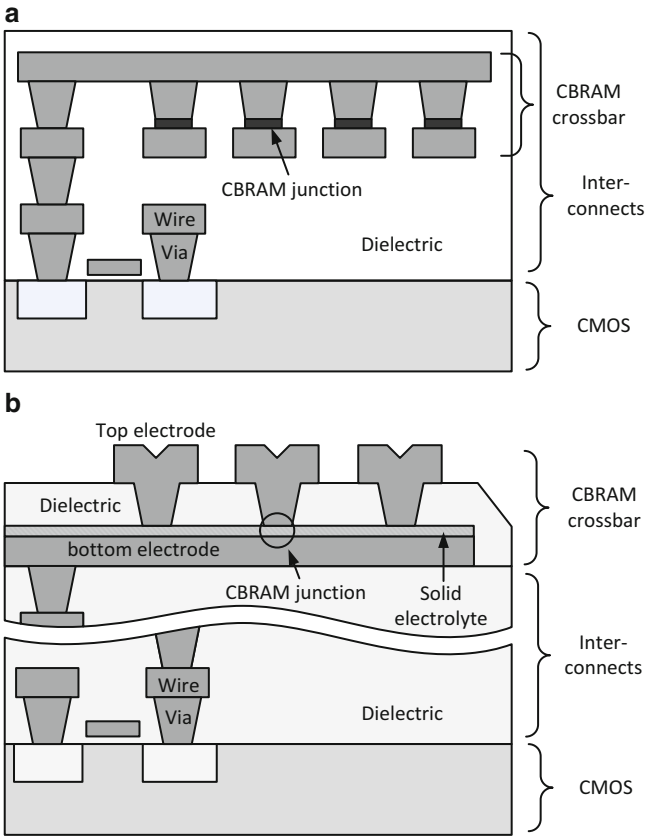


Fig. 4.2 Two approaches for 3D stacked CBRAM-crossbar fabrication: (**a**) crossbar structure integrated within interconnect layers and CBRAM devices fabricated at the *bottom* of vias and (**b**) crossbar structure fabricated on the *top* of interconnect layers

The conventional voltage-divider-based readout design is shown in Fig. 4.3, where R_l follows $\frac{R_{off}}{R_l} = \frac{R_l}{R_{on}}$. As such, V_o will be logic-1 when the target cell is in low-resistance state (LRS) and be logic-0 when in high-resistance state (HRS). Note that all the unselected word-lines and bit-lines need to be floated to avoid sensing current branches before reaching R_l. This will incur sneak-path issue. When the target cell is in HRS and is surrounded by other cells in LRS, there will be a significant leaked current flowing through the neighboring cells, which may lead to misinterpretation of the stored bit. The sneak-path issue can be addressed by adding a selection device for each CBRAM device [6, 17]. The selection device works like a bidirectional diode to ensure no current flowing through paths with more than one CBRAM device.

Instead of applying selection devices, alternative operations are proposed here for the CBRAM-crossbar readout circuit to avoid the sneak-path issue. The *write* and

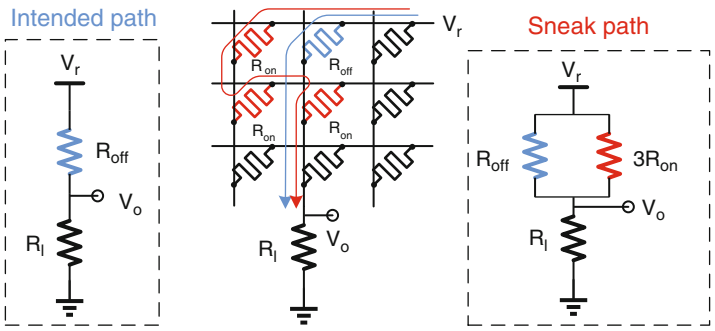


Fig. 4.3 Conventional crossbar read operation with incurred sneak-path issue

read operations for CBRAM-crossbar are shown in Fig. 4.1a and b, respectively. To write the CBRAM-crossbar, the bias-voltage v_w needs to be applied on the designated cell through multiplexed voltage selector. As such, the corresponding word-line and bit-line are applied with $v_w/2$ and $-v_w/2$ voltages, respectively. Such method is called half selection. By controlling the voltage level and polarity, one can determine the write-speed and change the on/off states of the cell. The cells that are half-selected will not change their states. To read the CBRAM-crossbar, the bias-voltage v_r (normally $v_r < v_w$) is applied on the corresponding word-line. By measuring the current of the designated column, the on/off state of the target cell can be detected. Note that because all unselected word-lines and bit-lines are grounded rather than floated, the voltage-divider-based readout scheme is incapable to function. As such, more sophisticated readout circuit design is required and is discussed later in this section.

Figure 4.1c further shows the peripheral circuit design for one CBRAM-crossbar. During write operation, the corresponding word-line and bit-line are applied with writing voltage through address decoding. The voltage selection is done by the row drivers and column drivers, which switch among different voltage supplies according to the command issued. During read operation, word-lines still need voltage selection while bit-lines are switched to current sensing circuit.

4.1.1.2 Crossbar Readout Circuit

To sense the current of the designated column in the crossbar structure, readout circuit illustrated in Fig. 4.21 is proposed. The sensing is done in two steps here. Firstly, a current mirror is deployed to amplify the current determined by the state of the target cell. The bias current I_{bias} is applied to ensure that transistor M1 works in saturation region. The bias-voltage V_{bias} has to be deliberately chosen according to I_{bias} to achieve a virtually grounded node A, i.e., a 0 V column voltage required in Fig. 4.1b. Otherwise, the current to detect will have to branch at node A to other cells at the same column, which will weaken the current signal and thus lead to a

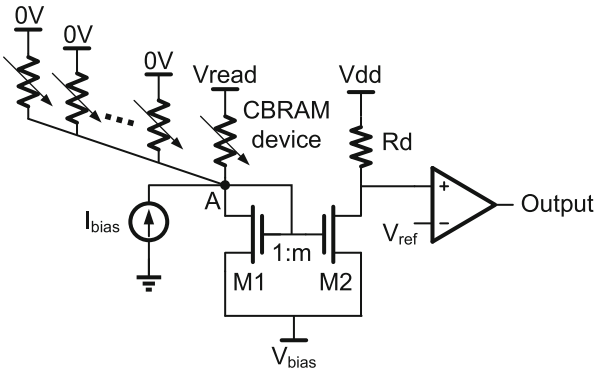


Fig. 4.4 Readout circuit design for CBRAM-crossbar structured memory

degradation of sensing accuracy. Secondly, the amplified current signal is converted into voltage signal and is further compared with the reference voltage to decide and output on/off state denoted by different logic levels (Fig. 4.4).

Another issue in readout operation is the severe device resistance variations. It is reported that 5×∼100× R_{off} and 2x∼10× R_{on} variations can be expected based on a variety of measurement data [4]. For one crossbar array, there exists a valid reference voltage in the readout circuit only when the minimum R_{off} of all CBRAM devices is greater than the maximum R_{on} of all devices, i.e.,

$$min(R_{off}) > max(R_{on}) \tag{4.1}$$

which, however, may not hold true if variation is significant. A crossbar with device resistance variations that cannot meet above condition is called a failure. As such, the sense amplifier is not able to guarantee successful readout for every operation. In the following, we will show the failure rate for different crossbar-array sizes and possible solutions to address such issue when failure rate is high.

Monte Carlo simulation with 5,000 times is conducted in order to investigate the device resistance variations impact on readout operation, which is shown in Fig. 4.5. In each iteration, a 100 × 100 crossbar array is generated, where R_{off} and R_{on} of all CBRAM devices have a normal distribution with μ of 1 GΩ and 2 MΩ and deviation δ that follows the variation in [4], i.e., 10× for R_{on} and 100× for R_{off}. The $max(R_{on})$ and $min(R_{off})$ are calculated and then used as the x-axis and y-axis values for each point. The whole domain is divided into two regions, pass region and fail region, separated by the dashed line $max(R_{on}) = min(R_{off})$. It can be observed that there are 9 failed cases, which lead to a 0.18 % failure rate for the 100 × 100 crossbar size. In other words, 9 out of 5,000 crossbars in this size do not have the ideal reference voltage value.

The Monte Carlo simulations for crossbar sizes starting at 50 × 50 to 600 × 600 with a step of 50 are conducted with results shown in Table 4.1. It can be observed that the failure rate increases as the crossbar size enlarges. For large crossbar arrays

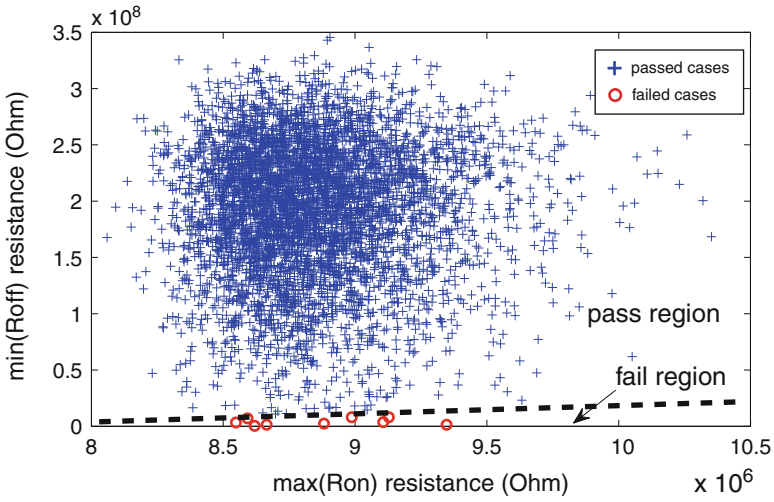


Fig. 4.5 Monte Carlo simulation of 100 × 100 crossbar array with device resistance variations

Table 4.1 Device resistance variation caused sense amplifier failure rate for different crossbar-array sizes

Array size	50^2	100^2	150^2	200^2	250^2	300^2
Failure rate	0.05 %	0.18 %	0.85 %	0.95 %	2.65 %	3.20 %
Array size	350^2	400^2	450^2	500^2	550^2	600^2
Failure rate	4.30 %	5.50 %	5.55 %	7.65 %	10.85 %	13.40 %

where the failure rate is not negligible, robust readout schemes, namely, the self-reference [8] and runtime ECC/ECP [21] can be applied to increase the readout reliability. Thus it is favorable to limit the crossbar size within a few hundreds by a few hundreds.

4.1.1.3 Crossbar Circuit Performance Evaluation

Based on the circuit design for CBRAM-crossbar, here we propose its system-level delay, power, and area models like the CMOS memory design tool CACTI [28]. We further verify such models by the transistor-level SPICE-level simulator developed in the previous section.

Delay Model

Generally, the time needed for write operation in CBRAM-crossbar is the signal propagation delay on wires (i.e., word-line and bit-line), denoted by D^w_{wire}, plus the

device switching time $D_{switching}$; for read operation, it is composed of wire delay D^r_{wire} and sensing delay $D_{sensing}$. Therefore we have

$$D_{write} = D^w_{wire} + D_{switching} \tag{4.2}$$

$$D_{read} = D^r_{wire} + D_{sensing}. \tag{4.3}$$

Different from DRAM/SRAM cell, the CBRAM device has asymmetrical write-/read-delays. Since the write operation requires physical change of CBRAM cell, i.e., the shape morphing of the conductive filament as illustrated in Fig. 3.6, it usually takes much longer time than simply detecting the resistance of CBRAM cell in the read operation. The CBRAM switching time $D_{switching}$ can be obtained from the proposed CMOS-CBRAM simulator, where high accuracy can be achieved thanks to the use of a physical model developed in the last chapter.

Due to the existence of leakage current in crossbar structure, CBRAM cells at different positions suffer from different amounts of reduced applied voltages introduced by IR-drop along the word-line and bit-line. Since the switching time is very sensitive to even the slightest of voltage deviation from the expected value, CBRAM cells at different positions of the crossbar array exhibit different values of switching time. Therefore, the exponential relation between $D_{switching}$ and the applied voltage v_w, represented as a lookup table, is built into CACTI.

Different from SRAM/DRAM sensing schemes where the subtle voltage/current swing or capacitor driving signal needs to be detected, the current signal in Fig. 4.21 is amplified and driven by power source; thus the sensing can be performed really fast. Note that the sensing delay can also be obtained by performing SPICE-level simulation. In the following, we focus on the modeling and calculation of wire delay D_{wire} for read/write operations.

For conventional 1T-1R structure, word-line delay can be calculated by distributed RC-line delay and the bit-line delay can be estimated by Seevinck model [22]. However, although the same approach has been applied to estimate the wire delay of crossbar structure in [33], it lacks accuracy due to the following reasons: Firstly, since there is no transistor in a crossbar, the word-line and bit-line delays are symmetric. In other words, the word-line and bit-line delays have to be modeled in the same manner. Moreover, the leakage current of cells along word-line and bit-line is a phenomenon specific to crossbar structure, which is not considered by conventional RC-line delay model. The leakage current will weaken the driving ability of row and column drivers, and hence a longer delay can be predicted compared to the conventional RC-line delay.

Figure 4.6a and b illustrates the crossbar delay model for read/write operation, with leakage path of cell i along the word-line and bit-line modeled as parallel R_{li}, whose value is its corresponding CBRAM on/off resistance. For the read operation, the bit-line is virtually grounded as illustrated in Fig. 4.1; thus only the word-line delay for the propagation of v_r contributes to the wire delay. For the write operation,

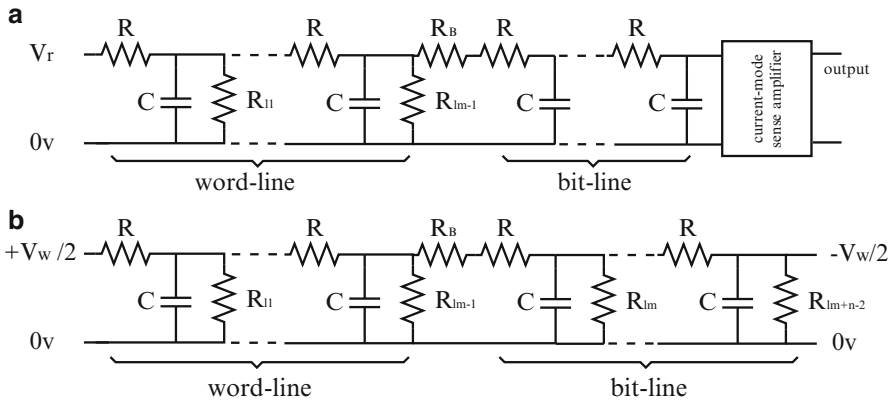


Fig. 4.6 RC-delay model of one CBRAM-crossbar for: (**a**) read operation; (**b**) write operation

both word-line driving voltage $v_w/2$ and bit-line voltage $-v_w/2$ will propagate to the target cell, and the total wire delay is determined by the slower one. Therefore, for a CBRAM-crossbar with m rows and n columns, the worst-case wire delay for both read operation and write operation can be calculated respectively by

$$D^r_{wire} = \alpha RCn^2 \tag{4.4}$$

$$D^w_{wire} = max(\alpha RCn^2, \alpha RCm^2) \tag{4.5}$$

where R and C are parasitic resistor and capacitor in unit length similar as the distributed RC-line model. Note that when the CBRAM device is scaled down, the R and C will be reduced accordingly, which produces a smaller wire delay. In addition, the location-dependent switching speed issue incurred by the IR-drop along the word-line and bit-line will be relieved as well.

Compared to the conventional RC-line delay expression, a fitting parameter α has been added to approximate the expected longer delay due to the CBRAM-crossbar structure, such as the effect introduced by R_{li} in Fig. 4.6. Practically, α can be obtained by fitting with a few samples obtained by simulating an entire CBRAM-crossbar in different sizes using the developed SPICE-like simulator.

Figure 4.7 shows the verification of the proposed crossbar specific delay model against accurate simulation results obtained through the developed SPICE-like simulator. It can be observed that the proposed model with fitting parameter $\alpha = 1.2$ is able to predict the crossbar delay well. As expected in Sect. 4.1.1.3, the crossbar wire delay, calculated by $1.2\,\mathrm{RCn}^2$, is thereby more than twice longer compared to the conventional distributed RC-line delay calculated by $0.5\,\mathrm{RCn}^2$. In other words, an error of more than 50 % will be incurred if the conventional distributed RC-line delay model is used instead.

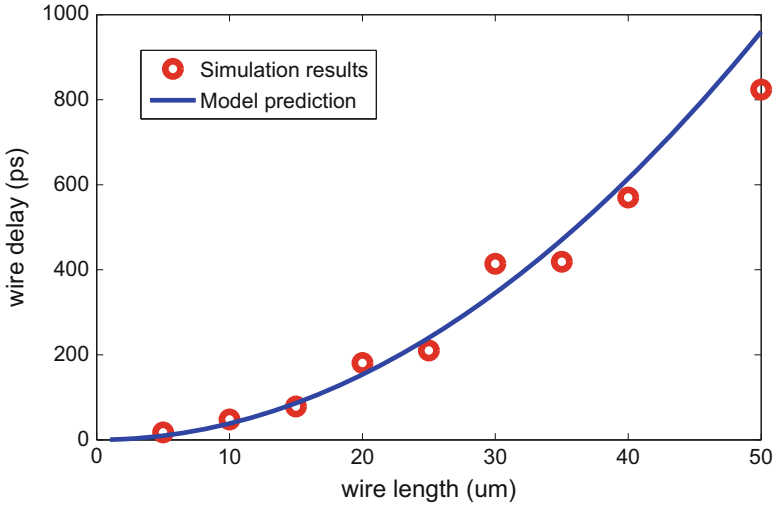


Fig. 4.7 Verification of proposed CBRAM-crossbar specific wire delay model against simulation results, with fitting parameter $\alpha = 1.2$

Power Model

The energy per write-access for CBRAM-crossbar is composed of two parts: the energy consumed to switch the target cell state and the energy dissipated along the word-line and bit-line. Consider a CBRAM-crossbar with m rows and n columns; the write-access energy E_{write} can be calculated as

$$E_{write} = E_{switching} + E^{w}_{static} + E^{w}_{dynamic} \tag{4.6}$$

where $E_{switching}$ is the energy for changing the target CBRAM cell state and needs to be obtained through the developed SPICE-like simulator since its dynamics is hard to be approximated using a resistor; $E^{w}_{dynamic}$ is the energy for charging parasitic capacitors along word-line and bit-line; and E^{w}_{static} is Joule heat dissipated on the half-selected CBRAM cells along word-line and bit-line and can be modeled as resistors since their resistance values remain constant during write operation.

The E^{w}_{static} and $E^{w}_{dynamic}$ can be calculated by

$$E^{w}_{static} = \left(k \cdot \frac{{v_w}^2}{4 \cdot R_{on}} + l \cdot \frac{{v_w}^2}{4 \cdot R_{off}} \right) \cdot D_{write} \tag{4.7}$$

$$E^{w}_{dynamic} = \frac{1}{8} C \cdot {v_w}^2 \cdot (m + n - 2) \tag{4.8}$$

where v_w is the write voltage, R_{on} and R_{off} are the *on-*/*off*-state resistance of CBRAM, D_{write} is the crossbar write-delay, C is the distributed unit capacitance

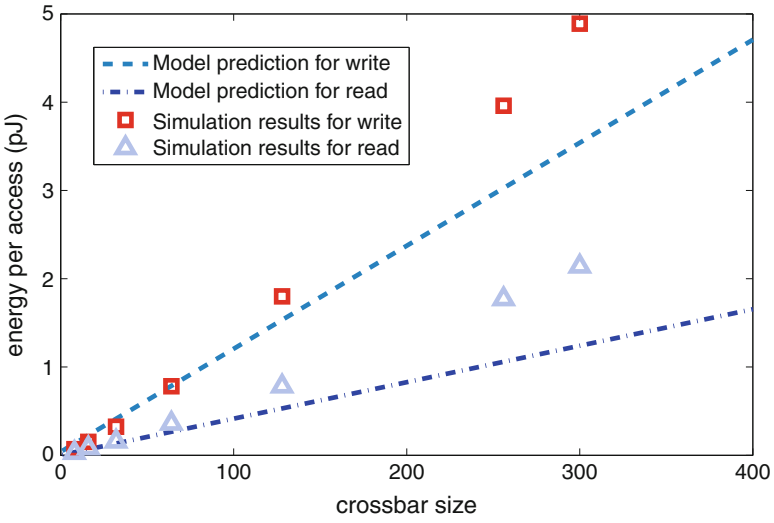


Fig. 4.8 Verification of proposed CBRAM-crossbar specific power model for read/write operations against simulation results

of crossbar wire, and k and l are the numbers of CBRAM cells in ON-state and OFF-state, respectively, along the path, following $k + l = m + n - 2$.

Similarly, the read-access energy E^r_{read} for crossbar can be calculated by

$$E_{read} = E^r_{static} + E^r_{dynamic} \tag{4.9}$$

Note that for read operation, all the bit-lines are virtually grounded and only cells in the target row consume power as shown in Fig. 4.1; thus we have

$$E^r_{static} = \left(k \cdot \frac{{v_r}^2}{R_{on}} + l \cdot \frac{{v_r}^2}{R_{off}} \right) \cdot D_{read} \tag{4.10}$$

$$E^r_{dynamic} = \frac{1}{2} C \cdot {v_r}^2 \cdot n \tag{4.11}$$

where v_r is the read voltage and D_{read} is the crossbar read-delay. Moreover, k and l are the numbers of CBRAM cells in ON-state and OFF-state in the target row, respectively, and they satisfy $k + l = n$. Note that the scalability of the CBRAM device is beneficial to dynamic power reduction. When scaled down, the smaller pitch size will reduce the wire capacitance with the power reduction at the word-line and bit-line.

The verification of the power model against simulation results is shown in Fig. 4.8. The simulation results are obtained by simulating $n \times n$ square CBRAM-crossbars with different n values, where n is used to denote crossbar size. It can be

observed that the proposed model for read/write operations is able to capture the trend with minor error when crossbar size increases.

Area Model

The area model consists of two parts: A_c for the pure area crossbar structure and A_p for corresponding CMOS peripheral circuits. Utilizing the 3D integration technique shown in Fig. 4.2, the crossbar is stacked over the active layer where its peripheral circuits are located. As such, the total area becomes

$$A = max(A_p, A_c). \tag{4.12}$$

For a CBRAM-crossbar with M rows and N columns, its area can be calculated by

$$A_c = M \cdot N \cdot L_{pitch}^2 \tag{4.13}$$

where L_{pitch} is the nano-bar pitch size, determined by the technology node of the CBRAM device. Therefore, at advanced technology node, extremely small CBRAM-crossbar area can be achieved due to the scalability of the CBRAM device. Besides the reduced wire delay inside the CBRAM-crossbar, the addressing delay outside the CBRAM-crossbar can be greatly reduced as well, which together will lead to significant memory access latency reduction. Note that a similar area model can be developed for the peripheral circuits based on [28].

4.1.2 3D Crossbar Resistive Memory

4.1.2.1 Diode-Added Memristor Memory

Recent device research has made it possible to fabricate each cross-point with a memristor and a pn-junction connected in series [16]. Though the junction could be modeled as a memristor in series with a diode, the pn-junction does not prevent setting the memristor value in the reverse direction [16]. In this way, the sneak path can be extensively reduced and a large portion of power consumption is saved.

Figure 4.9 shows the structure of a 4×4 memristor crossbar, whose read-access is controlled by two 4-to-1 switch MUXes connected with the voltage source. Cross-points of the memory crossbar (red circle) can be implemented using either one pure resistive memristor or one diode-added memristor.

How our design prevents the sneak paths is illustrated in Fig. 4.10. Here, only cross-points with an ON-state memristor are shown for visual clarity. The solid blue line indicates the current path to read the cell in the 1st row and 1st column. Two possible sneak paths are shown with red dotted lines when pure resistive

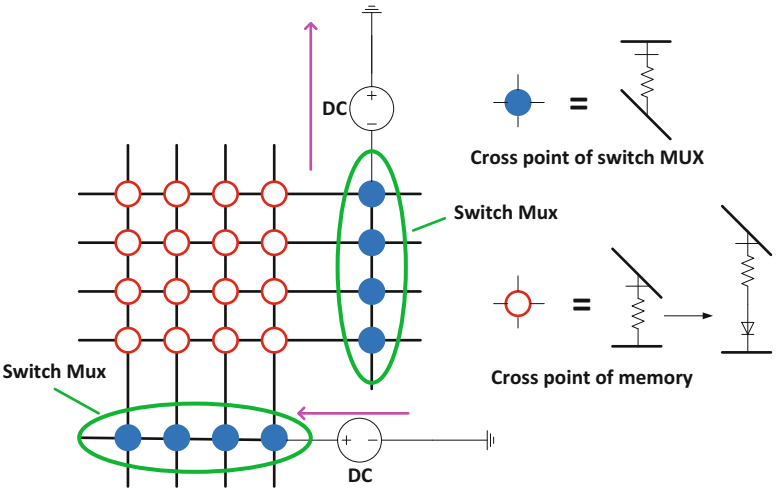


Fig. 4.9 A 4×4 crossbar memory for read operation

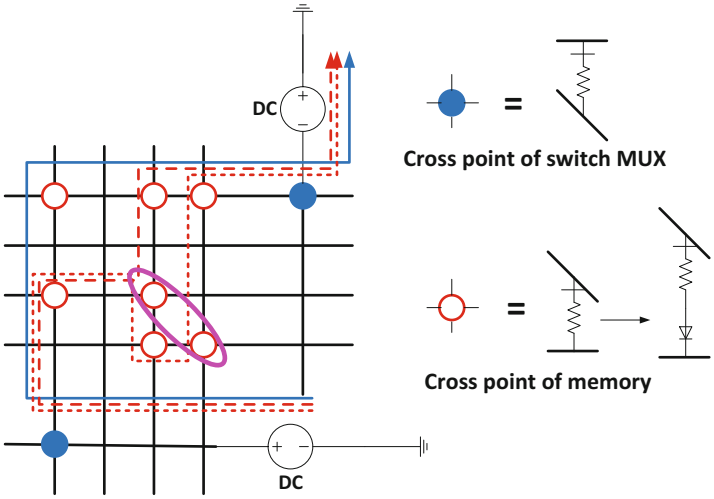


Fig. 4.10 Sneak-path prevention

memristors are used in the memory cells. Each sneak path may be composed of 3, 5, or more (odd number) ON-state cells. Their resistances are connected in parallel with the reading path, resulting in not only larger power consumption but also large performance degradation. When diode-added memristors are used for memory cells instead, current can only flow in one direction for a read operation. As shown in the figure, current can only flow from vertical bars to horizontal bars. Therefore, there is no way for a sneak path to go through other paths without getting blocked by one diode. The two cells marked by one pink circle can block the previous sneak paths.

Table 4.2 Read performance for crossbar memories

Crossbar structure	2D 4X4 with diode	2D 4X4 without diode	3D 4X8 with diode
I_{on} | all other cells off (nA)	38.349	53.587	38.47
I_{on} | all cells on (nA)	38.478	65.893	38.688
I_{off} | all other cells on (nA)	0.81361	57.872	1.275
I_{off} | all cells off (nA)	0.46612	0.63801	0.69832
I_{on} range (nA)	38.349 to 38.478	53.587 to 65.893	38.47 to 38.688
I_{off} range (nA)	0.46612 to 0.81361	0.63801 to 57.872	0.69832 to 1.275
Wost case I_{on}/I_{off}	47.13	0.93 (fail)	30.17
Power range (nW)	0.746 to 61.6	1.02 to 105	1.12 to 61.9

Adding the diode into the memory cells does not change the memory writing scheme described in [31]. Instead, it only introduces different energy requirements for setting and resetting of memristor states [16]. The comparisons between the two designs based on diode-added memristor and resistive memristor are analyzed in terms of functionality and power consumption by our circuit simulator.

However, the new crossbar is not strictly resistive any more due to the embedded diode. This leads to some drawbacks. For example, depletion capacitor in reverse-biased diode increases capacitive load and may cause some delay. Moreover, threshold voltage of the diode can reduce output voltage swing as well. Therefore, appropriate sizing and doping are required to minimize these drawbacks. However, the superior performance and tremendous reduction in power consumption brought by the diode-based design motivate us to explore the potential of diode-added memristor for memory.

To analyze the effect of the new diode-added memristor, each cross-point is modeled as a memristor connected in series with a diode to form a new 4×4 crossbar. A read-function is then operated in comparison with the crossbar by pure resistive memristors. A switch-MUX is implemented similarly to [31]. For simplicity, memristors for memory and switch-MUX are set with the same parameters except the threshold voltage, which is set larger for switch-MUX to prevent unwanted value-changing during the write-function. The parameter settings are the same as in our designed decoder. With ±0.8 (V) as reading voltages, the output current is used to determine ON/OFF state stored in memory cells.

Simulation results are shown in Table 4.2 where performance and power consumptions are compared. In Table 4.2, I_{on} and I_{off} indicate the resulted output currents when reading an ON-state or OFF-state, respectively. The worst case is to read an ON-state while all other cells are in OFF-states and to read an OFF-state while all other cells are in ON-state. These two operations generate the minimum I_{on} and maximum I_{off}, whose ratio (worst-case I_{on}/I_{off}) is viewed as a measure for memory performance. As Table 4.2 indicates, the I_{on}/I_{off} ratio for the read-function improves tremendously when diode-added memristor is used, while the power consumption is also decreased greatly. Note that although the minimum power consumption for the memory without diode is only 1.02 (0.638A×1.6V) nW,

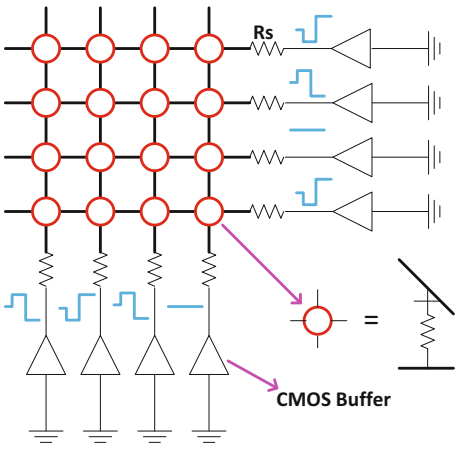


Fig. 4.11 A 4×4 crossbar with various inputs

due to the existence of a sneak path, the power consumption can rise to 92.6 (57.87A × 1.6 V) nW when reading an OFF-state (I_{off} | all other cells on). When diode-added memristor is used, on the other hand, high power consumption only appears when reading an ON-state. Also, the maximum power consumption is almost halved. Therefore, the total power consumption can be improved around four times. When the memory size increases, this improvement is expected to further increase.

As mentioned earlier, the existence of sneak path limits the maximum memory size for a proper operation. As shown in Table 4.2, the 4×4 crossbar memory built with pure resistive memory already fails because it cannot distinguish an ON-state and OFF-state (worst I_{on}/I_{off} ratio <1). Therefore, the maximum memory size achievable with the given device parameters is less than 4×4. Since parts of the peripheral components would not shrink the size along with memory [14], this limitation in size can result in limitation on device density, which is resolved when diode-added memristors are deployed instead.

We can also efficiently evaluate the process variation of the memristive circuits by applying Monte Carlo simulations within the new simulator. A 4×4 crossbar memory is implemented with memristors used for variation analysis of the write operation. As Fig. 4.11 shows, three different input patterns (step functions switching between ±4 V) are fed to 8 bars through buffers to write the memory cells at the junction. A ±30 % variation is assumed for memristor-device length (D), resulting in a distinct I–V hysteresis path for each memristor. For simplicity, R_{on}, R_{off}, and D are assumed to be not correlated. Parameters are set: R_{on} = 3.33e7 Ω, R_{off} = 3.33e10 Ω, μ_v = 2.5e-6 ($m^2s^{-1}V^{-1}$), D = 1e-8±30 % m, R_s = 1e7 Ω. All memristances are set to Roff at the beginning.

Diverse input voltages and variation in parameter D can lead to complicated transient paths for memristor values in the crossbar. Figure 4.12 shows the transient change of memristance for one of the memristors (W1-1). As the figure indicates, the memristance is successfully written despite the variations in D. In our experiment, all 16 memristors are written to the expected values. On the other hand,

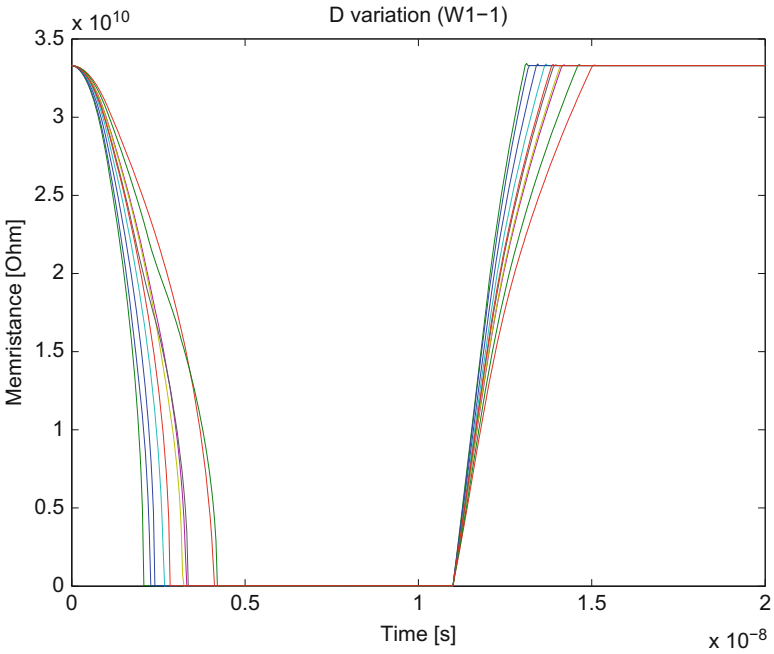


Fig. 4.12 Transient path of value for one memristor (W1-1) with ±30 % variation of device lengths (D) for all 16 memristors. Only parts of the results are shown in the plot for visual clarity

the transient path for the memristor value is very sensitive to D. A Monte Carlo analysis (Fig. 4.13) shows that a ±30 % variation in D leads to more than ±50 % variation in time delay of the write operation.

4.1.2.2 3D Crossbar Memory

Based on the previously discussed building blocks, we further discuss memory architecture by introducing a 3D crossbar-based design with the use of diode-added memristors. The pioneering idea to explore the nano-electronic at the architecture level is from the work of CMOS and molecular logic circuit (CMOL) [7]. CMOL adds the nanowire crossbar on top of CMOS stack, so as to further increase the device density. This hybrid architecture can be used to implement memory, reconfigurable logic, and neuromorphic networks [7].

Figure 4.14a indicates that the traditional CMOL uses a special pin to reach the top layer of the crossbar. However, fabrication variation may cause this pin to entangle with the bottom layer of crossbar and hence may result in missing contacts and defective circuits. To solve this problem, a modified CMOL, called field programmable nanowire interconnect (FPNI), is developed in [26]. As shown in Fig. 4.14b, FPNI uses large-size nano-pads to contact with CMOS stack, leading

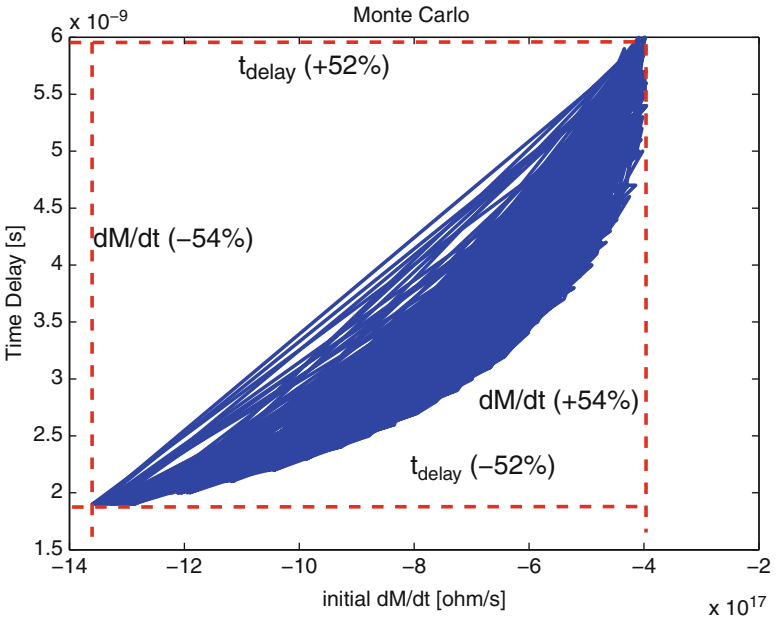


Fig. 4.13 Monte Carlo analysis for parameter Ds impact on the transient changing path of memristance in the crossbar. For a ±30 % variation of D, the initial changing speed of memristance has a mean value of 8.75 (Ωs^{-1}) and a variation of ±54 %, and the time delay before a successful write has a mean value of 3.95 (ns) and a variation of ±52 %

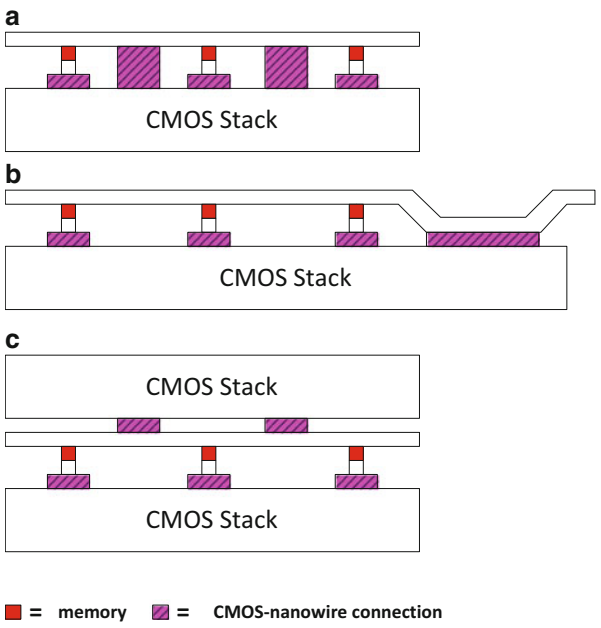


Fig. 4.14 Different architectures for CMOL: (**a**) traditional CMOL, (**b**) FPNI, and (**c**) 3D CMOL

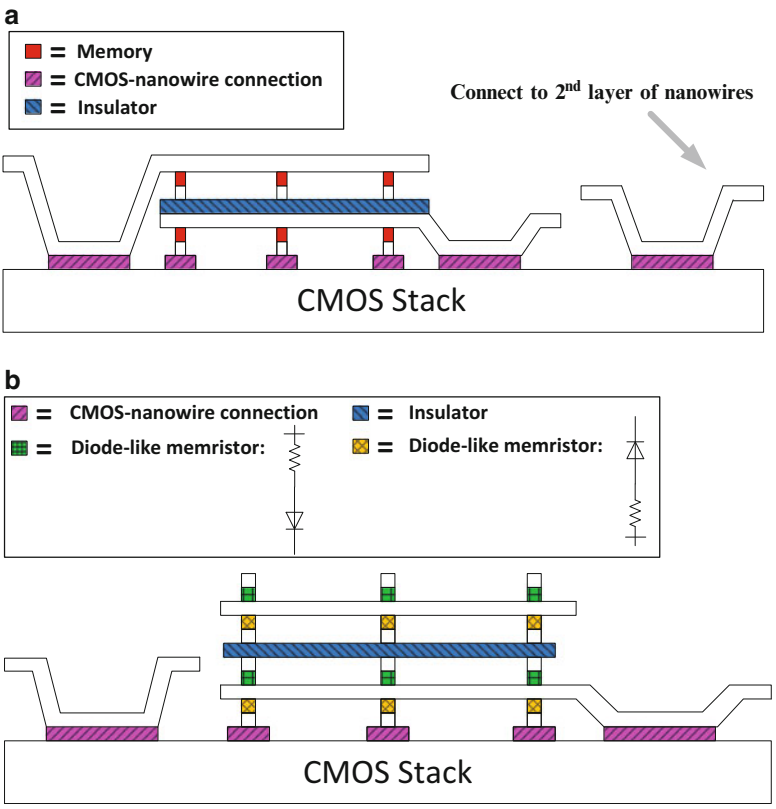


Fig. 4.15 Architecture for a 3D crossbar memory: (**a**) 3D RRAM and (**b**) our design

to a fabrication with high defect tolerance. However, due to the large size of pads, a low device density is resulted. Another solution is to introduce a 3D CMOL with 2 CMOS stacks and 1 crossbar layer in between [30]. As shown in Fig. 4.14c, each CMOS stack only needs to contact with the nearer nanowire layer of the crossbar. However, since in memory design, the CMOS peripheral area is relatively small, only one CMOS stack is needed below the nanowire crossbars. In addition, 3D memory design is also discussed in [13, 14], where multiple layers of nanowire crossbars are fabricated above one CMOS stack to form the 3D resistive RAM (RRAM). The crossbars are separated with each other by insulator layers (Fig. 4.15a). Nanowires are then contacted with CMOS stack in a similar way to FPNI, leading to large peripheral area.

Apart from the limitations for each of the architectures discussed above, pure resistive crossbar-based memory also has a common limitation on its maximum size achievable for implementing one function. The number of sneak paths increases as

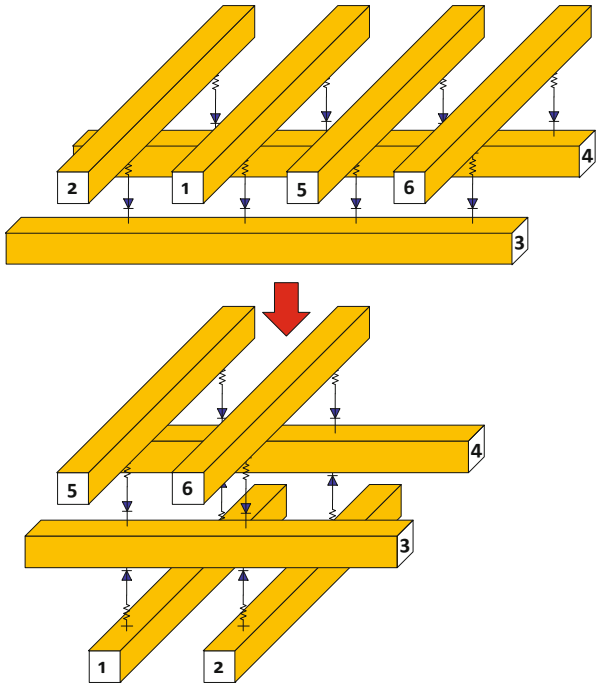


Fig. 4.16 Folding of the nanowire crossbar

the memory size rises, causing large size crossbar memory to fail in one operation. In this book, we propose a different 3D crossbar architecture with the use of diode-added memristors (Fig. 4.15b). Obviously, the sneak path can be prevented in this design. The limitation on crossbar size for proper operation is therefore released. Larger-sized crossbar can be built with a much smaller peripheral area overhead.

Moreover, we can further reduce the memory area and increase the device density by folding a two-layer nanowire crossbar into a three-layer crossbar. As shown in Fig. 4.15b, two nanowire layers now share one perpendicular nanowire layer. The memory folding detail is shown in 4.16. Because the diode directions for two adjacent crossbars are opposite to each other, the folded crossbar memory can function correctly. Due to the folding of the longer dimension of the memory, there is an estimated 33 % increase in the memory density that can be built for the same technology.

Using the proposed architecture in Fig. 4.16, two 4×4 crossbars are merged together on top of CMOS stack to form a folded 3D 4×8 memory. With the same memristor, switch-MUX, and reading voltages implemented in the 2D 4×4 crossbar memory, the resulted output current and power consumption are shown in Table 4.2. As Table 4.2 indicates, the Ion/Ioff ratio for the read operation degrades a bit when compared to 2D memory, which could be justified by the increase in memory size. As the memory size doubles compared to 2D memory, Ioff is expected to rise

due to increase in leakage current paths, while Ion should not be affected much. This is proved by the measured data in Table 4.2. More importantly, the 3D power consumption remains the same level as the 2D crossbar memory although memory size is doubled. This benefit comes from prevention of sneak path, which highly decreases the power consumption.

4.1.3 1T-1R Spintronic Memory

Thanks to the strongly nonlinear I–V curve of ion migration kinetics, ReRAM devices have threshold effect so that half selection technique can be applied. This enables crossbar memory structure and transistor free feature. This is why crossbar memory architecture is often associated with ReRAM technologies. Spintronic memory, on the other hand, is similar to conventional SRAM/DRAM technologies which require transistors to control. Thus spintronic memory is often associated with 1T-1R structure, where "T" stands for transistor and "R" denotes one nonvolatile device whose state is represented by resistance.

4.1.3.1 STT-RAM Memory

A typical hybrid STT-RAM cell is shown in Fig. 4.17a with one transistor and one STT-MTJ in series connection. The structure is identical to that of DRAM cell except that the capacitor is replaced by STT-MTJ device. The gate of transistor is connected to word-line, which serves to select target cells in the same word-line. When enabled, two bit-lines (namely, bit-line and select line) can be driven to have V_w or $-V_w$ depending on the desired data to write, 1 or 0. In the "write 1" operation, WL is connected to VDD, and BL and SL are connected to V_{DD} and ground, respectively. In the "write 0" operation, the polarities of SL and BL line are interchanged. Readout operation can be performed in a similar way by using V_r, and the current loop will eventually be measured by readout circuit, which determines device state by current amplitude.

4.1.3.2 STT-RAM Readout Circuit

Existing STT-RAM readout schemes to avoid disturbance of large STT-MTJ resistance variation usually require several steps, which slows down the read latency. We show that by applying a single-sawtooth pulse and exploiting the resistance roll-off of STT-MTJ, the robust readout can be achieved within one cycle.

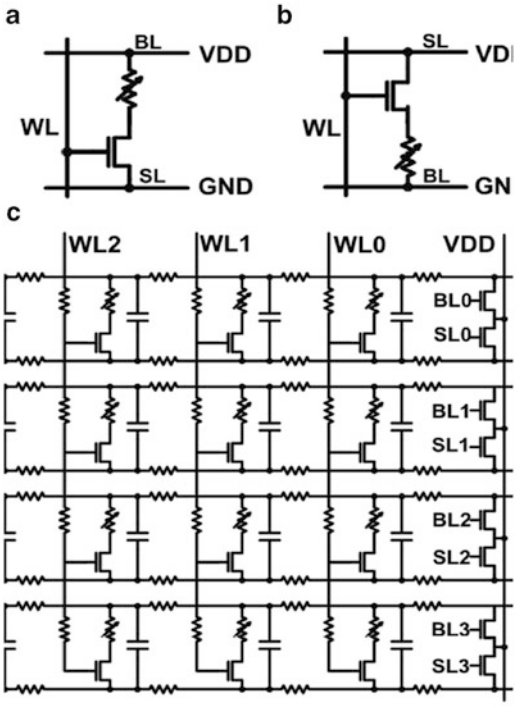


Fig. 4.17 Circuit diagrams of 1T-1R STT-RAM memory cell: (**a**) simplified memory cell for write 1; (**b**) simplified memory cell for write 0; and (**c**) 16-bit STT-RAM with 4 bit-lines and 4 word-lines

Basic STT-RAM Readout Circuit

The basic voltage sensing scheme for the popular 1T-1MTJ structure STT-RAM is shown in Fig. 4.18a. The reference voltage is set to satisfy

$$I_r \cdot (R_{AP} + R_t) > V_{ref} > I_r \cdot (R_P + R_t)$$

where I_r is the applied read current and R_{AP}, R_P, and R_t are the MTJ antiparallel state resistance, parallel state resistance, and cell transistor ON-state r_{ds}, respectively. However, in the presence of bit-to-bit MTJ resistance variation, the reference voltage has to fulfill

$$Min(V_{BL,AP}) > V_{ref} > Max(V_{BL,P})$$

where a satisfying V_{ref} may not exist when variation is large.

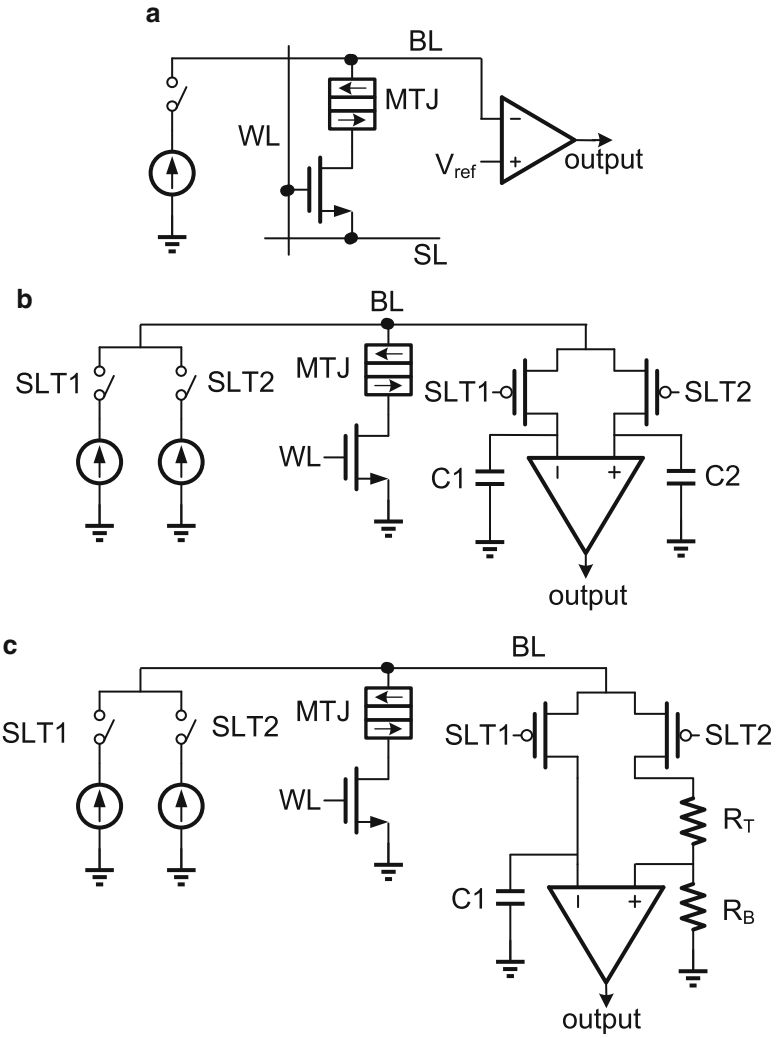


Fig. 4.18 The existing schemes for STT-RAM readout: (**a**) basic STT-RAM readout, (**b**) destructive self-reference readout in [9], and (**c**) nondestructive self-reference readout in [5]

Destructive Self-Reference Readout Circuit

In order to achieve reliable readout in the presence of large MTJ resistance variation, a self-reference readout is presented in [9], whose diagram is shown in Fig. 4.18b. The read operation is done in five phases:

- The read-current I_r is applied and its bit-line voltage is stored in C1.
- The "0" (parallel state) value is written to the target cell.

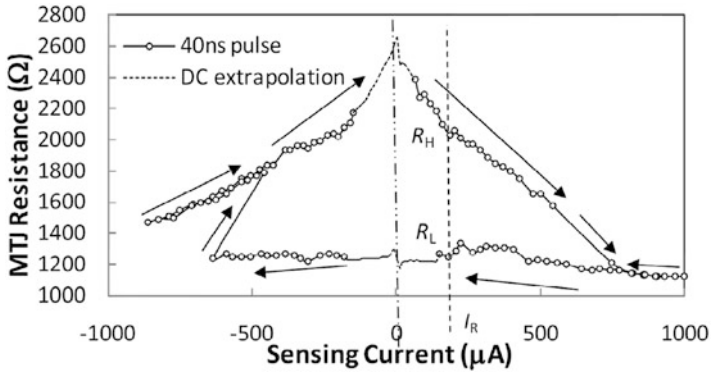


Fig. 4.19 The measured R–I sweep curve of a typical MgO-based MTJ in [9]

- The read-current I_r is applied again and its bit-line voltage is stored in C2.
- The sense amplifier is enabled and voltages of C1 and C2 are compared and the output is "1" (antiparallel state) if V_{C1} is greater than V_{C2} and "0" otherwise.
- The output value has to be written back to the destructed cell.

Therefore, in terms of both speed and power, the overhead brought by write-back may be large for this scheme.

Nondestructive Self-Reference Readout Circuit

The current-dependent resistance roll-off can be observed for STT-MTJ as shown in Fig. 4.19. By exploiting the fact that the roll-off slope of the antiparallel state is much greater than that of parallel state, a nondestructive self-reference readout is proposed in [5] with a diagram shown in Fig. 4.18c.

The read operation is done in three phases:

- The read-current I_{r1} is applied to achieve its corresponding resistance R_1.
- The read-current I_{r2} is applied to achieve its corresponding resistance R_2.
- The sense amplifier is enabled and R_1 and R_2 are compared. The output is "1" if two values are significantly different and "0" otherwise.

As such, the nondestructive self-reference readout can improve the read latency by eliminating the two time-consuming write phases. However, this scheme still has limited performance in read latency and sensing margin as discussed below.

4.1.3.3 Single-Sawtooth-Pulse-Based Readout Circuit

Although the variation can be overcome by the two self-reference schemes above, they still involve several phases which slow down the read latency. A single-sawtooth-pulse-based readout is proposed in Fig. 4.20 to reduce the read latency

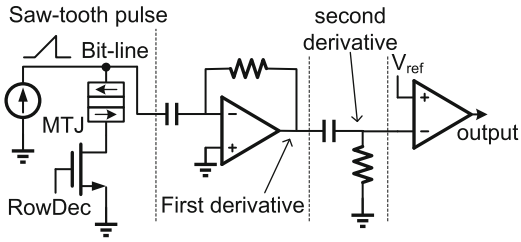


Fig. 4.20 Diagram of proposed single-sawtooth-pulse-based readout

into one cycle. In the following, we will show that within one cycle, by applying a single-sawtooth pulse to bit-line and obtaining the second derivative of corresponding bit-line voltage, the variation disturbance during readout can be totally avoided.

Assume the applied sawtooth pulse to bit-line can be expressed as

$$i(t) = k_s \cdot t \tag{4.14}$$

where the k_s denotes the current rising rate. Also, the R–I curve slope in Fig. 4.19 is assumed linear for simplification; thus the current-dependent resistance can be expressed as

$$\begin{aligned} R_{AP}(i) &= R_H - k_{AP} \cdot i \\ R_P(i) &= R_L - k_P \cdot i \end{aligned} \tag{4.15}$$

where R_H and R_L are the resistances of R_{AP} (antiparallel) and R_P (parallel) when $i = 0$. Therefore, the bit-line voltage produced by the applied sawtooth pulse can be expressed as

$$\begin{aligned} V_{BL,AP}(t) &= i(t) \cdot R(i) = R_H \cdot k_s \cdot t - k_{AP} \cdot k_s^2 \cdot t^2 \\ V_{BL,P}(t) &= i(t) \cdot R(i) = R_L \cdot k_s \cdot t - k_P \cdot k_s^2 \cdot t^2 \end{aligned}. \tag{4.16}$$

It can be observed that the bit-line voltage depends on the R_H and R_L, which will introduce readout errors in the presence of large variations. Nevertheless, the readout dependency on R_H and R_L can be eliminated if the second derivative of bit-line voltage can be obtained:

$$\begin{aligned} \frac{d^2 V_{BL,AP}}{dt^2} &= -2 \cdot k_{AP} \cdot k_s^2 \\ \frac{d^2 V_{BL,P}}{dt^2} &= -2 \cdot k_P \cdot k_s^2 \end{aligned} \tag{4.17}$$

Note that the current-dependent resistance roll-off slopes k_{AP} and k_P are not fabrication process sensitive, and k_P is a close-to-zero while k_{AP} is much larger as indicated in Fig. 4.19; the robust readout can be easily achieved under the proposed scheme right after the sawtooth pulse is applied. Thus compared with previous work where several steps are required, the proposed scheme can potentially reduce the read latency into one cycle time.

The circuit to implement second-derivative operation can be designed as two differentiators in series, and each differentiator can be implemented as either an OPAMP-based feedback circuit or an RC-based high-pass filter. An RC-based high-pass filter provides simple differentiation but has limited gain, while an OPAMP-based one can generate output with significant sensing margin. The three different circuits to implement the second-derivative operations with trade-off between circuit complexity and readout sensing margin are as follows:

- Pure OPAMP: two OPAMP-based in series
- Hybrid: first stage with OPAMP-based and second stage with RC-based high-pass filter
- Pure RC: two RC-based high-pass filters in series

In this experiment, the hybrid approach is deployed for the single-sawtooth-pulse-based readout as shown in Fig. 4.20.

The proposed single-sawtooth-pulse-based readout scheme in Fig. 4.20 and the STT-RAM array circuit are simulated together using NVM-SPICE intrinsic STT-MTJ model. The STT-MTJ model parameters are set with $cap = 0.9$, $cp = 0.1$, $rap = 2650$, and $rp = 1230$ with explanations for each parameter in Table A.2; the BSIM4.7 model is used for all the transistors with $L = 90\,\text{nm}$ and $W = 2\,\mu\text{m}$; the $I_w = \pm 900\,\mu A$ is used for write operation, and read-current I_r rises from 0 to $200\,\mu\text{A}$ within 25 ns, which produces a sawtooth pulse with $k_s = 8000\,\text{A/s}$.

The single-sawtooth-pulse-based readout with second-derivative circuit achieved in all three ways are simulated with results shown in Fig. 4.21. It can be observed that for the bit-line response to the applied sawtooth pulse current, the bit-line of AP state STT-MTJ exhibits some nonlinearity while that of P state is almost linear. The first derivative of bit-line signals shows larger difference between AP state and P state cases, where that of P state is almost constant while that of AP state has a considerable slope. The output difference in this stage still cannot avoid the readout fault caused by resistance variation according to Eq. (4.16). The second derivation of bit-line voltage, which is variation tolerant according to Eq. (4.17), shows separated voltage level for AP state and P state, which is also easy to distinguish by later sensing stage. The pure OPAMP-based one can produce voltage levels separated by around 200 mV between AP state and P state, and that of hybrid-based one is around 30 mV and less than 0.2 mV for pure RC-based one. Table 4.3 shows the readout performance comparison between the proposed scheme and previous work.

4.1.4 Domain-Wall Spintronic Memory

Compared with the conventional SRAM or DRAM by CMOS, the domain-wall nanowire-based memory (DWM) can demonstrate two major advantages. Firstly, extremely high integration density can be achieved since multiple bits can be packed in one macro-cell. Secondly, zero standby power can be expected as a nonvolatile device does not require to be powered to retain the stored data. In addition, the resistance-detection-based readout does not require bit-line precharging, which

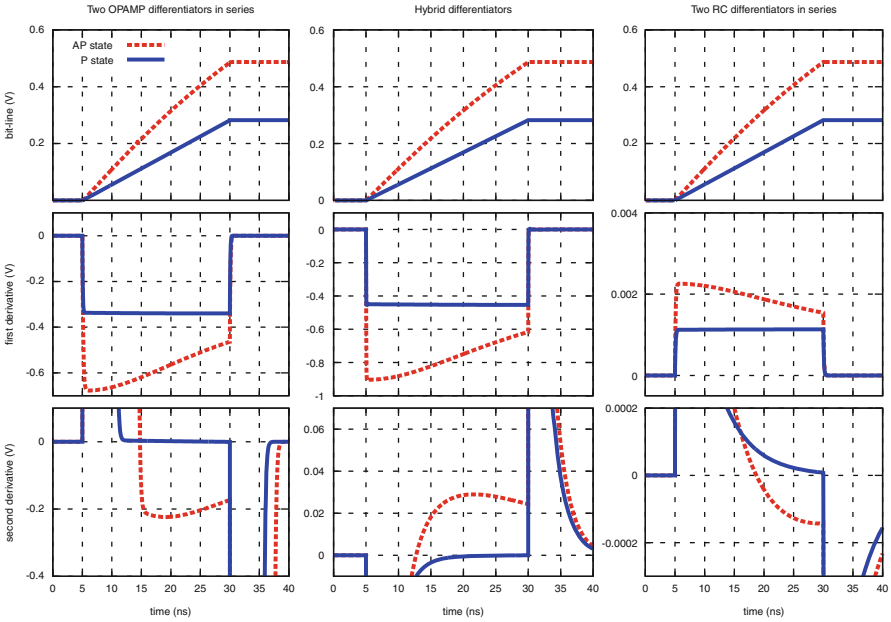


Fig. 4.21 Transient response of bit-line voltage, first derivative, and second derivative to the applied sawtooth pulse for pure OPAMP, hybrid, and pure RC-based circuit, respectively

Table 4.3 Comparison of different readout schemes for STT-RAMs

Scheme	Read latency	Sense margin (mV)
Destructive readout [9]	2 read cycles + 2 write cycles	76.6
Nondestructive readout [5]	2 read cycles	12.1
Proposed	1 read cycle	15 (hybrid) 100 (OPAMP)

avoids the subthreshold leakage of the access transistors. In this section, we will present DWM-based design with macro-cell memory: structure, modeling, and data organization.

4.1.4.1 DWM Memory

Figure 4.22a shows the design of domain-wall nanowire-based memory (DWM) macro-cell with access transistors. The access port lies in the middle of the nanowire, which divides the nanowire into two segments. The left-half segment of the nanowire is used for data storage while the right-half segment is reserved for shift operation in order to avoid information loss. In order to access the leftmost bit, the reserved segment has to be at least as long as the data segment. In such case, the data utilization rate is only 50 %. In order to improve the data utilization rate,

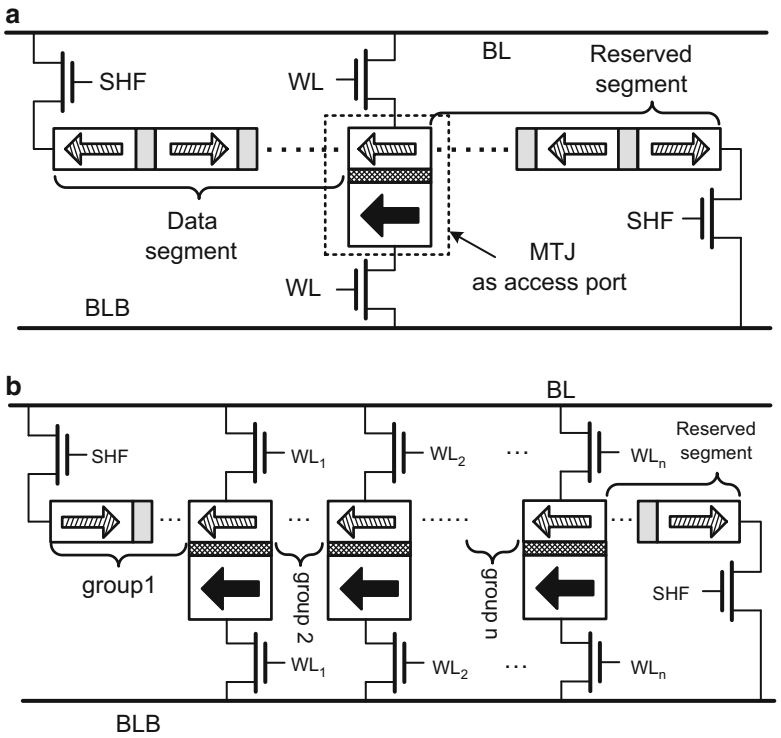


Fig. 4.22 A macro-cell of DWM with (**a**) a single access port and (**b**) multiple access ports

a multiple port macro-cell structure is presented in Fig. 4.22b. The access ports are equally distributed along the nanowire, which divides the nanowire into multiple segments. Except the rightmost segment, all other segments are data segments with the bits in one segment forming a *group*. In such case, to access an arbitrary bit in the nanowire, the shift-offset is always less than the length of one segment; thus the data utilization rate is greatly improved.

Thus, the number of bits in one macro-cell can be calculated by

$$N_{cell-bits} = (N_{rw-ports} + 1)N_{group-bits} \tag{4.18}$$

in which $N_{rw-ports}$ is the number of access ports. Then the macro-cell area can be calculated by

$$A_{nanowire} = N_{cell-bits} L_{bit} W_{nanowire} \tag{4.19}$$

$$\begin{aligned} A_{cell} =& A_{nanowire} + 2A_{shf-nmos} \\ &+ 2A_{rw-nmos} N_{rw-ports} \end{aligned} \tag{4.20}$$

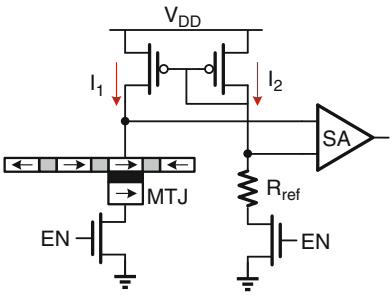


Fig. 4.23 Sensing circuit design for domain-wall nanowire

where L_{bit} is the pitch size between two consecutive bits, $W_{nanowire}$ is the width of domain-wall nanowire, and $A_{shf-nmos}$ and $A_{rw-nmos}$ are the transistor size at shift port and access port, respectively.

Moreover, the bit-line capacitance is crucial in the calculation of latency and dynamic power. The increased bit-line capacitance due to the multiple access ports can be obtained by

$$\begin{aligned} C_{bit-line} =& (N_{rw-ports} C_{drain-rw} \\ &+ C_{drain-shf} + C_{bl-metal}) \times N_{row} \end{aligned} \tag{4.21}$$

in which $C_{bl-metal}$ is the capacitance of bit-line metal wire per cell and the $C_{drain-rw}$ and $C_{drain-shf}$ are the access-port and shift-port transistor drain capacitances, respectively. Note that the undesired increase of per-cell capacitance will be suppressed by the reduced number of rows due to higher nanowire utilization rate.

Additionally, the domain-wall nanowire specific behaviors will incur in-cell delay and energy dissipation. The magnetization reversal energy 0.27 pJ and delay 600 ps can be obtained through the transient analysis by NVM SPICE as discussed in Chap. 3. The read energy is in fJ scale and thus can be omitted. Also, the read operation will not contribute to in-cell delay. The delay of shift operation can be calculated by

$$T_{shift} = L_{bit}/v_{prop} \tag{4.22}$$

in which v_{prop} is the domain-wall propagation velocity that can be calculated by Eq. (3.40). The Joule heat caused by the injected current is calculated as the shift-operation dynamic energy.

4.1.4.2 DWM Readout Circuit

The readout circuit for domain-wall memory is similar to that of STT-RAM, both based on the GMR effect. The basic readout circuit for DWM is shown in Fig. 4.23. To obtain the bit information from a specific domain, the domain needs to be

first shifted and aligned with the fixed layer so that the readout operation can be performed. Its resistance, determined by the GMR effect, is then compared to the reference resistance, and the result that indicates its state will then be output. Note that the readout circuit in Fig. 4.23 is the most basic readout circuit; more sophisticated readout techniques like self-reference readout and nondestructive readout circuit described in Sect. 4.1.3.2 can also be applied.

4.2 Nonvolatile Logic Circuit

4.2.1 ReRAM Crossbar Logic

4.2.1.1 Decoder

Decoders are the essential peripheral circuits to support memory access and are also important building blocks for memory-based logic. The diode-added memristor can be used to further improve the decoder as follows. The previous DEMUX-based decoders are usually implemented using the preprogrammed pure resistive memristors. However, due to the nature of the resistive crossbar, the DEMUX functions as a voltage divider with the large power consumption and the performance degradation from sneak paths. By adding the pn-junction, i.e., the diode into the memristor, the diode-added memristor crossbar can be developed.

Figure 4.24 shows one decoder implemented with a bistable diode-added crossbar. Here, ON-state (low-resistance) cross-points are marked with circles, while the other cross-points are in OFF-states (high resistance). Since the crossbar does not include the inversion function, the address-signal (A0-A2) and their complements are needed. The circuit is essentially a lookup table. The ON-state diode-added

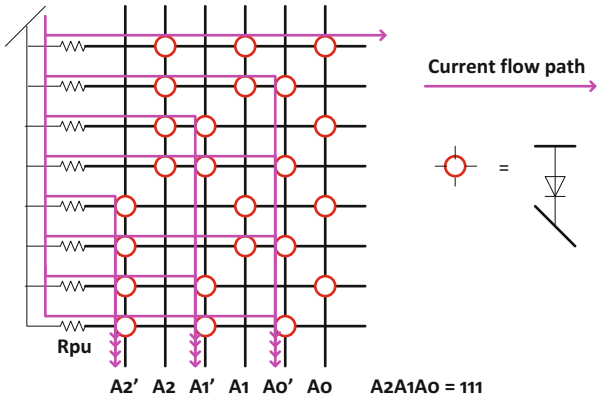


Fig. 4.24 One 3-8-decoder based on a bistable diode crossbar

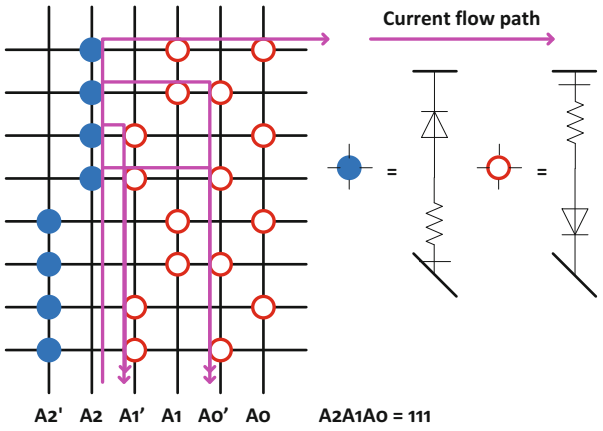


Fig. 4.25 One low-power 3-8-decoder based on a diode-added memristor crossbar

Table 4.4 DEMUX performance comparison

Crossbar Implementation	Memristor	Diode	Diode-added memristor
Output1 (V)	1.497	1.4776	1.0614
Output2 (V)	−0.499	0.5386	0.496
Output3 (V)	−0.499	0.5386	0.496
Output4 (V)	−0.499	0.4896	0.4302
Total power (nW)	1805.4	44.334	8.5953

memristors at the cross-points form an AND gate in each row. The line is selected and remains on high voltage when its cross-points are all connected to the logic 1. On the other hand, one or more ON-state cross-points in nonselected rows connect to the logic 0, which acts as one current sink to pull down the nonselected lines. The current flow paths are marked by pink lines. We can see four subcurrents flowing through A0-A2 in Fig. 4.24, and hence the power consumption is large.

In order to save power, a new decoder structure is proposed in Fig. 4.25. The current paths are now reduced to half of the last design. By using the pure resistive memristors in the columns for the first input signal, the current from A2 or A2′ can flow into all the rows through the resistive cross-points. In this way, the leaking subcurrents in nonselected lines can be reduced to two. Hence, the new decoder can be used for write operation together with the diode-added memristor memory to save the total power.

Two decoder designs and the DEMUX structure proposed by HP [31] are used to construct a 2-to-4 DEMUX for the decoder. Their performances are compared in Table 4.4. The parameters of memristors are set: Ron = 1e7 Ω, Roff = 1e10 Ω, μ_v = 2.5e-6 ($m^2 s^{-1} V^{-1}$), D = 1e-8 m, $V_{thd} = 2$ V, and $V_{thr} = 4$ V, except for memristors

in the first two columns in Fig. 4.25, whose R_{on} and R_{off} values are set ten times larger to assist voltage division. Here, V_{thd} and V_{thr} are the threshold voltage for programming diode-added memristor and pure resistive memristor, respectively. Similarly, the pull-up resistors (Fig. 4.24) are set ten times of R_{on} ($R_{pu} = 1e8\,\Omega$) for better performance. All outputs are loaded with $R_{load} = 10R_{off} = 1e11\,\Omega$. The threshold voltage for the diode is set as 0.43 V. Input voltage level of 0.5 V is used for HP's design, and 1.5 V for the other two designs. By adding state variable Φ, our simulator is able to handle historical information of memristor and therefore handle hybrid memristor–CMOS diode circuit easily. Simulation results are shown in Table 4.4.

As Table 4.4 indicates, the power consumption decreases tremendously when diode-added memristors are used. Distinct output voltage levels are important for operations in memory. The output voltage levels in the later two DEMUX structures are limited by the threshold voltage of the diode. Note that the diode threshold voltage is an unwanted feature in diode-added memristor and should be minimized. Therefore, the actual performance can be improved when diode threshold voltage can be lowered.

4.2.1.2 Adder

Decoders can be used to implement memory-based logic and CMOS buffers. In this book, the proposed decoder with diode-added memristors is used to implement a full adder (Fig. 4.26b). For comparison, another full adder (Fig. 4.26a) is implemented by pure resistive memristor-based crossbar method [3]. As Fig. 4.26a shows, the logic is again realized by the voltage dividing.

In Fig. 4.26a, each line of memristors with an inverter forms a NOR gate demonstrated by HP in [3]. The parameters for memristor are set the same as in the decoder design. To design the inverter, parameters for NMOS are set as W/L $= 100\,\mu m/0.24\,\mu m$, $\mu nCox = 117.7e\text{-}6$ (AV-2), $V_{th} = 0.43$ (V), $\lambda = 0.06$ (V^{-1}). A 33 kΩ resistor is connected in series with NMOS to form the inverter. The design in [36] is used in Fig. 4.26b with the decoder changed to the newly designed one as in Fig. 4.25 for the second full-adder design. Two pull-down resistors (Rpd) are set to be 100 times Ron of the memristors in the decoder. A small CMOS buffer is then used for obtaining the output. A 3V supply voltage is used for both adders. The simulator now handles the hybrid circuits with both memristors and various CMOS components. The inputs and outputs of two designs could be viewed in Fig. 4.27. The power consumptions of memristor-based logic are compared for the two full adders. The experiment results show that the power consumption improves from around 3.5 (μW) to around 0.18 (μW) when shifted to the diode-added memristor, saving 95 % of power while maintaining the same performance.

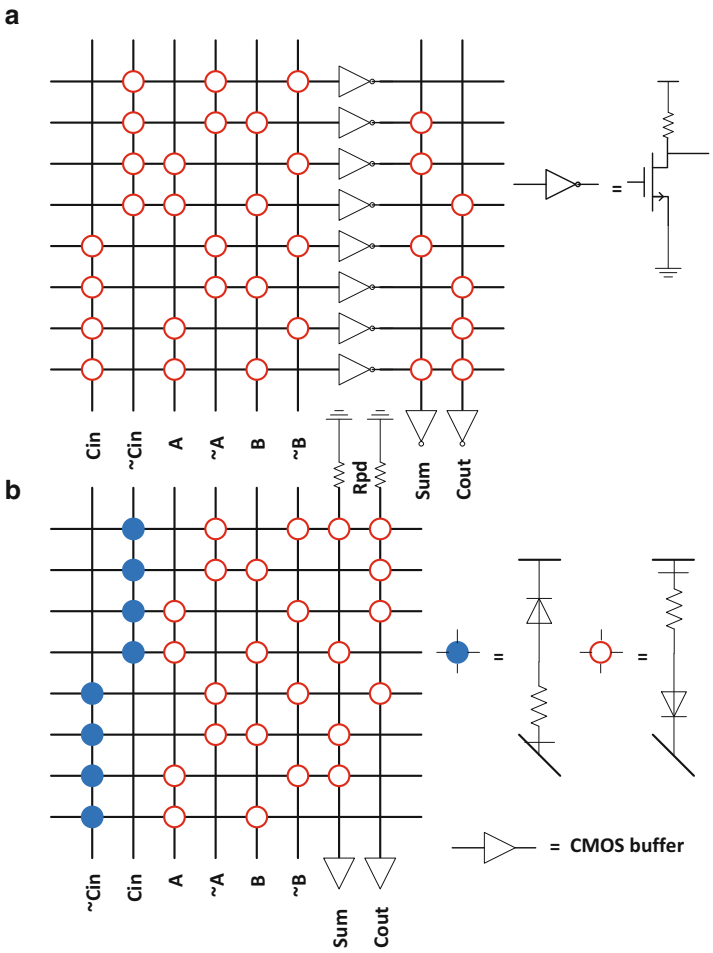


Fig. 4.26 Two full adders implemented by (**a**) pure resistive memristor crossbar and CMOS inverters and (**b**) proposed low-power decoder with CMOS buffers

4.2.2 Domain-Wall Logic

4.2.2.1 XOR

The magnetization switching with sub-nanosecond speed and sub-pJ energy have been experimentally demonstrated [20, 34, 35]. As such, the domain-wall nanowire-based logic can be further explored for logic-in-memory-based computing. In this section, we show how to further build DWL-based XOR logic and how it is applied for low-power ALU design for comparison and addition operations.

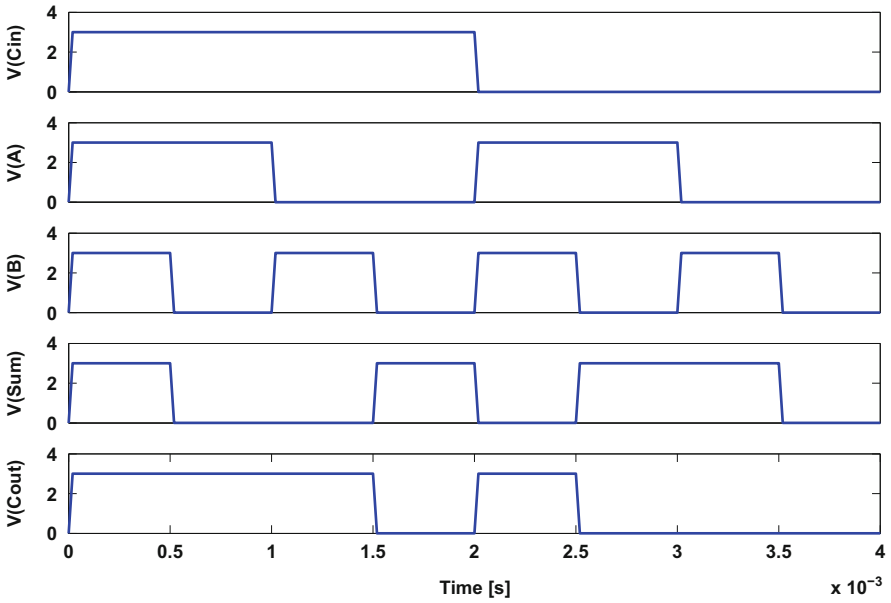


Fig. 4.27 Inputs and outputs of two full adders. V(Cin), V(A), and V(B) are the inputs to the adders, while V(Sum) and V(Cout) are the outputs

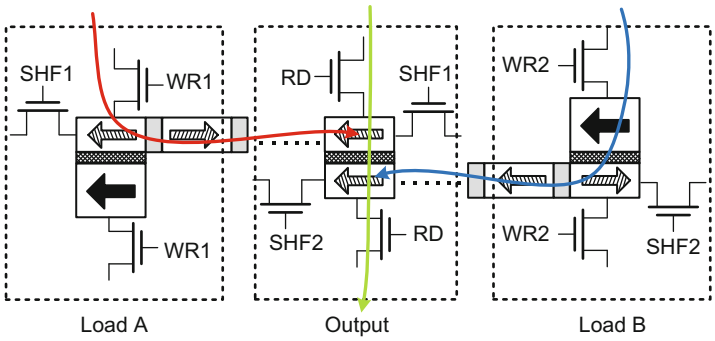


Fig. 4.28 Low-power XOR logic implemented by two domain-wall nanowires

The GMR effect can be interpreted as the bitwise-XOR operation of the magnetization directions of two thin magnetic layers, where the output is denoted by high or low resistance. In a GMR-based MTJ structure, however, the XOR logic will fail as there is only one operand as variable since the magnetization in fixed layer is constant. Nevertheless, this problem can be overcome by the unique domain-wall shift operation in the domain-wall nanowire device, which enables the possibility of DWL-based XOR logic for computing.

A bitwise XOR logic implemented by two domain-wall nanowires is shown in Fig. 4.28. The proposed bitwise-XOR logic is performed by constructing a new

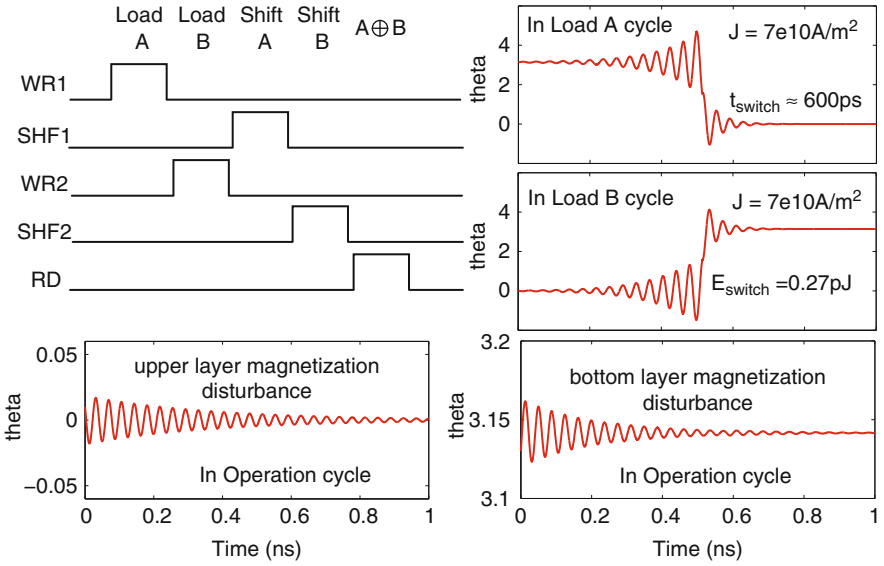


Fig. 4.29 The timing diagram of DWL-XOR with SPICE-level simulation for each operation

read-only port, where two free layers and one insulator layer are stacked. The two free layers are in the size of one magnetization domain and are from two respective nanowires. Thus, the two operands, denoted as the magnetization direction in free layer, can both be variables with values assigned through the MTJ of the agreeing nanowire. As such, it can be shifted to the operating port such that the XOR logic is performed.

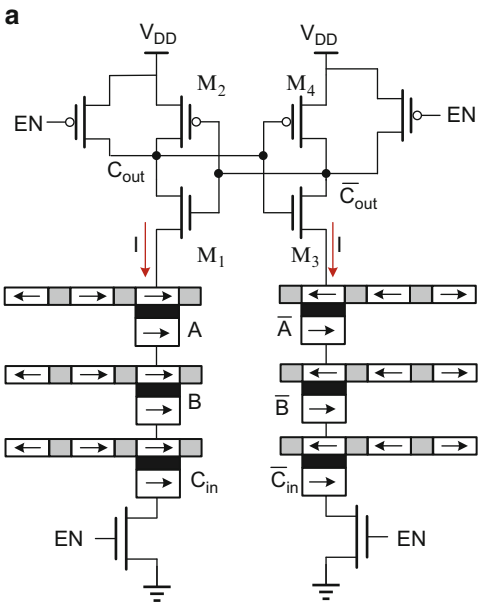
For example, the $A \oplus B$ can be executed in the following steps:

- The operands A and B are loaded into two nanowires by enabling WL_1 and WL_2, respectively.
- A and B are shifted from their access ports to the read-only ports by enabling SHF_1 and SHF_2, respectively.
- By enabling RD, the bitwise-XOR result can be obtained through the GMR effect.

Note that in the x86 architecture processors, most XOR instructions also need a few cycles to load its operands before the logic is performed, unless the two operands are both in registers. As such, the proposed DWL-based XOR logic can be a potential substitution of the CMOS-based XOR logic. Moreover, similar as the DWM macro-cell, zero leakage can be achieved for such XOR logic.

The transient analysis of the domain-wall nanowire XOR structure has been performed in the SPICE simulator, with both controlling timing diagram and operation details shown in Fig. 4.29.

The current density of $7e10\ \mathrm{A/m^2}$ is utilized for magnetization switching. The θ states of the nanowire that takes A are all initialized at 0 and the one that takes B



b

EN	A	B	C	R_L>R_R?	C_{out}
1	0	0	0	false	0
1	0	0	1	false	0
1	0	1	0	false	0
1	0	1	1	true	1
1	1	0	0	false	0
1	1	0	1	true	1
1	1	1	0	true	1
1	1	1	1	true	1
0	X	X	X	false	0

Fig. 4.30 The carry out logic achieved by domain-wall nanowires (**a**) Domain-wall memory cell structure (**b**) LUT by domain-wall nanowire array with parallel output

all at π. Only two bits per nanowire are assumed for both nanowires. The operating port is implemented as a developed magnetization controlled magnetization (MCM) device, with internal state variables θ and ϕ for both the upper layer and bottom layer. In the cycles of *loadA* and *loadB*, the precession switching can be observed for the MTJs of both nanowires. Also, the switching energy and time have been calculated as 0.27 pJ and 600 ps, which is consistent with the reported devices [20, 34, 35]. In the *shift* cycles, triggered by the SHF-control signal, the dynamics θ and ϕ of both the upper and bottom layers are updated immediately. In the *operation* cycle, a subtle sensing current is applied to provoke GMR effect. Subtle magnetization disturbance is also observed in both layers in the MCM device, which validates the read operation. The θ values that differ from initial values in the *operation* cycle also validate the successful domain-wall shift.

4.2.2.2 Adder

To realize a full adder, one needs both *sum* logic and *carry* logic. As the domain-wall nanowire-based XOR logic has been achieved, the sum logic can be readily realized by deploying two units: $Sum = (A \oplus B) \oplus C$. As for carry logic, spintronics-based carry operation is proposed in [29], where a precharged sensing amplifier (PCSA) is used for resistance comparison. The carry logic by PCSA and two branches

of domain-wall nanowires is shown in Fig. 4.30a. The three operands for carry operation are denoted by resistance of MTJ (low for 0 and high for 1) and belong to respective domain-wall nanowires in the left branch. The right branch is made complementary to the left one. Note that the C_{out} and $\overline{C_{out}}$ will be precharged high at first when PCSA EN signal is low. When the circuit is enabled, the branch with lower resistance will discharge its output to "0." For example, when left branch has no or only one MTJ in high resistance, i.e., no carry out, the right branch will have three or two MTJs in high resistance, such that the C_{out} will be 0. The complete truth table is shown in Fig. 4.30b, which is able to confirm carry logic by this circuit. The domain-wall nanowire works as the writing circuit for the operands by writing values at one end and shift it to PCSA. Note that with the full adder implemented by domain-wall nanowires and intrinsic shift ability of domain-wall nanowire, a shift/add multiplier can be further achieved purely by domain-wall nanowires.

4.2.2.3 LUT

Figure 4.31a shows the structure of the cell in the LUT array. The access port lies in the middle of the nanowire, which divides the nanowire into two segments. The left-half segment of the nanowire is used for data storage while the right-half segment is reserved for shift operation in order to avoid information loss.

Figure 4.31b shows the domain-wall nanowire-based LUT array. The input of the function implemented by LUT is represented as binary address. The address is fed into word-line decoder and bit-line MUX to find the target domain-wall nanowire cell, where the multiple-bit result is kept. The LUT array size depends on the domain, range, and precision of the function to perform.

Based on the way data is organized, the result can be output in serial manner or parallel manner. In serial output scenario, the binary result is stored in single domain-wall nanowire that is able to hold multiple bits of information. Assume each cell has only one access port and the first bit of result is initially aligned with access port; the way to output result is to iteratively readout and shift one bit until the last bit is output. In parallel output scenario, the multiple-bit result is distributed into different nanowires. Because each cell has their own access port, the multiple bits can be output concurrently. The design complexity of parallel output scheme is that, to find the relative position of the result within the nanowire, a variable access time will be introduced. For example, if luckily the result is stored at first bit of the nanowires, the result can be readout in one cycle; on the contrary if the result is kept at the very last bit of the nanowires, it will take tens of cycles to shift first before the result is output. Therefore, the choice between serial output and parallel output is the trade-off between access latency and design complexity.

Figure 4.32 shows the power characterization of DW-LUT in different array sizes. To obtain the area, power, and speed of DW-LUT, the memory modeling tool CACTI [28] has been extended with domain-wall nanowire model as discussed in Chap. 4. In terms of dynamic energy per lookup operation, the parallel output scenario is much more power efficient than serial output scenario, and the gap

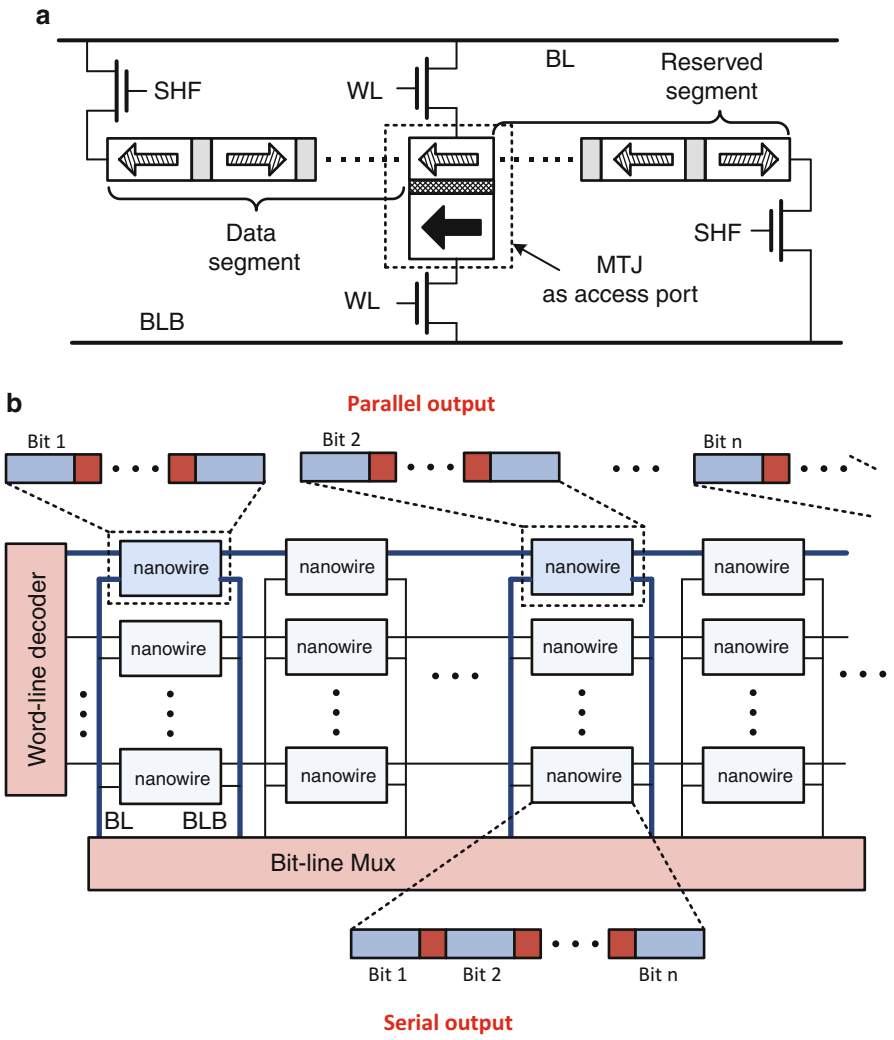


Fig. 4.31 LUT by domain-wall nanowire array (**a**) with parallel output and (**b**) with serial output

enlarges when array size increases. This is because more cycles are required to output results in serial than in parallel; therefore more access operations are involved. However, the serial scenario is able to avoid the variable access latency issue, which reduces the design complexity of the controller. For leakage power, the nonvolatility leads to extremely low leakage power in nW scale, which is negligible compared with its dynamic power. For volatile SRAM and DRAM, the leakage power may consume as large as half the total power especially in advanced technology node [28].

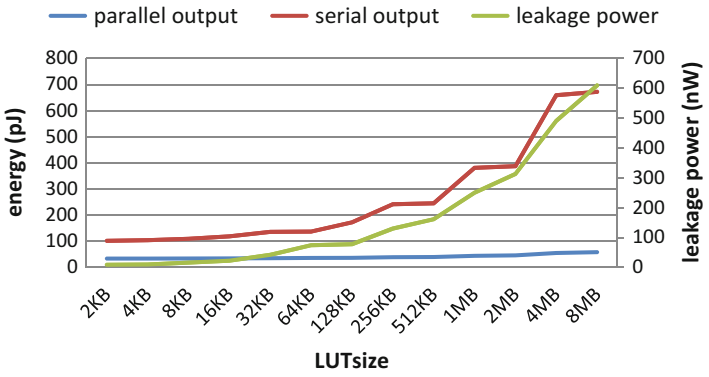


Fig. 4.32 Power characterization for DW-LUT in different sizes

Once the domain, range, and precision of function are decided, the DW-LUT size can be determined accordingly. Therefore, the power characterization can be used as a quick reference to estimate the power profile of specific function to perform system-level design exploration and performance evaluation.

4.3 Nonvolatile Analog Circuit

4.3.1 ReRAM Synapse for Analog Learning

The value-adaptive nature of memristor can lead to potential application in neuromorphic systems. There are many recent researches conducted on implementing memristors in neural network and other biological circuits [1, 10, 19, 25].

In [19], a memristive circuit (Fig. 4.33) is used to model amoebas' learning behavior. When exposed to the periodic environment change, amoeba is able to remember the change and adapts its behavior for the next stimuli. By using a simple RLC circuit together with a memristor, this learning process can be emulated. According to the author, this model may also be extended and applied in neural network.

To examine the full learning process, a CMOS spike generator is cascaded with the amoeba model to emulate the changing environment. Parameters for the memristor are Ron $= 3\,\Omega$, Roff $= 20\,\Omega$, $\mu_v = 2.5\text{e-}6$ ($\text{m}^2\text{s}^{-1}\text{V}^{-1}$), D $=$ 1e-8 m, $V_{thd} = 2.5$ (V). The rest of the model is set as R $= 0.195\,\Omega$, L $= 0.02$ (H), C $= 0.01$ (F). As shown in Figs. 4.34 and 4.35, the memristor adjusts its value to facilitate oscillation when facing periodic spikes. As memristance becomes larger, when a following spike is fed to the circuit again, the oscillation becomes less attenuated and stays longer. This can be viewed as the emulation for amoeba to remember the environment change and adapt its behavior to anticipate the next stimulus.

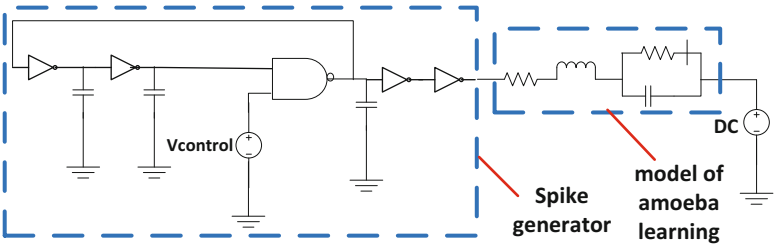


Fig. 4.33 Memristive model for the amoeba learning together with the spike generator

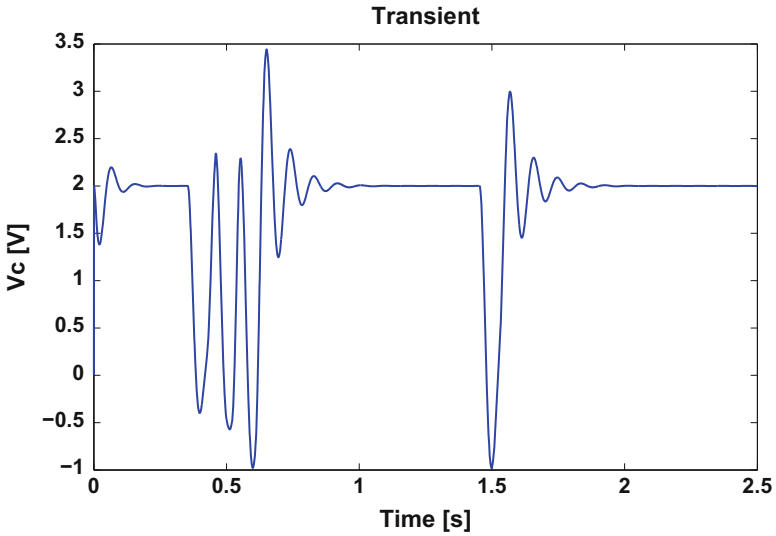


Fig. 4.34 Outputs of a memristive model for amoeba learning. Periodic spike input causes memristor to adjust its value, leading to longer oscillation when the spike is fed again

On the other hand, when non-periodic inputs are fed, the adjustment is much less obvious as the case under the periodic input. Here the added state variable Φ keeps information for both memristor and inductor. Simulation results in Fig. 4.21 show that the proposed simulator works well with analog simulation of hybrid memristor–CMOS circuits.

In the Pavlov's classic experiment, an unconditional food stimulus can always elicit unconditional salivation response of dogs, and a neutral stimulus of bell sound is not able to elicit salivation at first. However, after repeated sound-food pairing stimuli, the salivation can be elicited by only sound stimulus without food. The neutral bell stimulus then becomes a conditional stimulus which is able to elicit conditional response. This conditioning behavior can be strengthened or weakened by more sound-food pairing training or no more food association with sound.

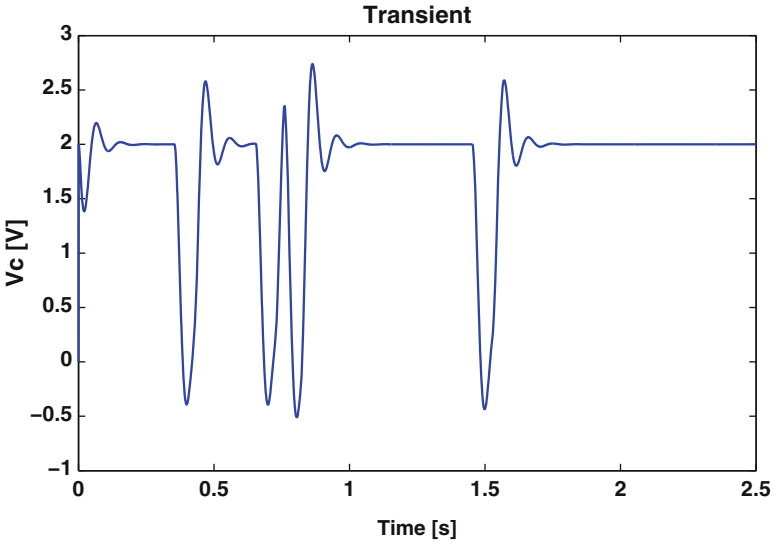


Fig. 4.35 Outputs of a memristive model for amoeba learning. Non-periodic spike input results in less obvious adjustment

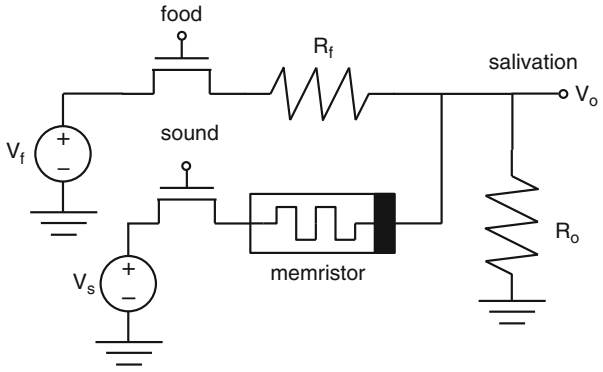


Fig. 4.36 A simple hybrid CMOS–memristor circuit to model the dynamic conditioning behavior

Such a training behavior has been modeled in [18]. Memristors have been utilized as synapses to denote the dynamically changing strong or weak bond between neural cells by its programmable ability. However their model of memristor ideally assumes a threshold for memristor programming and therefore cannot model the behavior that the conditioning can be dynamically weakened. Here we propose a simple hybrid CMOS–memristor circuit, as shown in Fig. 4.36, which can better model the conditioning behavior.

The interaction between unconditional food stimulus and salivation response can be modeled as a resistor R_f with very small resistance, therefore, to indicate that

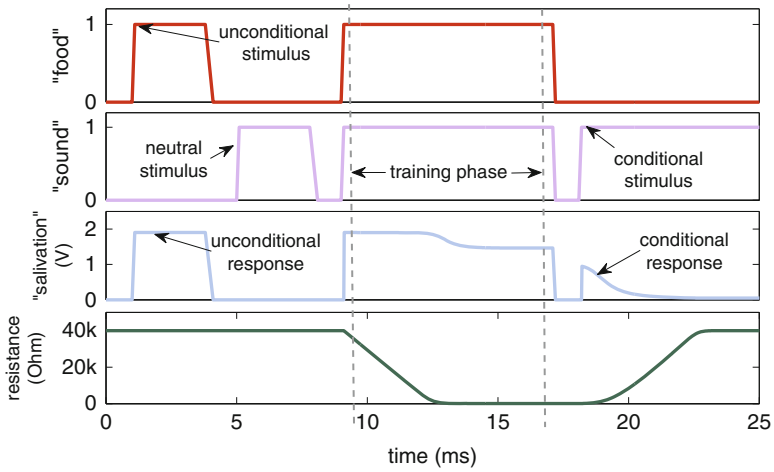


Fig. 4.37 The simulation results of the hybrid CMOS–memristor design for simple classic conditioning

the food stimulus signal can easily pass through to generate salivation output. Since the sound stimulus is not able to elicit salivation at first, the memristor is initially programmed at OFF-state to prevent the salivation response. R_o is designed with resistance somewhere between ON-state and OFF-state resistance. The sound and food stimuli can be controlled by CMOS transistors. The unconditional stimulus voltage V_f is set with higher voltage than neutral stimulus V_s. Therefore before training, when unconditional food stimulus is applied, the salivation output will always respond $V_o \approx V_f$. The sound stimulus alone will result in $V_o \approx 0$. The training can be achieved by applying both food and sound stimuli. The resistance relationship will lead to $V_o \approx V_f > V_s$. The negative voltage drop on memristor will slowly program it from OFF-state to ON-state. After training, the sound stimulus alone can also elicit salivation output $V_o \approx V_s$, and hence the conditioning is established. During the conditioning action, there exists a small positive voltage drop on memristor which will eventually program memristor from ON-state to OFF-state. This can be observed as the conditioning getting weakened.

The above hybrid design for classic conditioning behavior is simulated within NVM SPICE. R_o is set to $2\,\text{k}\Omega$, R_f $100\,\Omega$, V_f DC 2 V, and V_s DC 1 V. The memristor is set with $R_{on} = 100\,\Omega$, $R_{off} = 40\,\text{k}\Omega$, and other parameters with default values as described in Table A.1. Two ideal transistor switches are controlled by the "food" and "sound" signals. Transient analysis is conducted and the results are shown in Fig. 4.37.

The simulation results, shown in Fig. 4.37, are consistent with previous analysis. Before training, the unconditional food stimulus can elicit an unconditional salivation response, while the neutral sound stimulus is unable to do so due to the high OFF-state resistance of memristor which cuts down the signal transmission. During the training phase, the sound stimulus is associated with food stimulus. This is

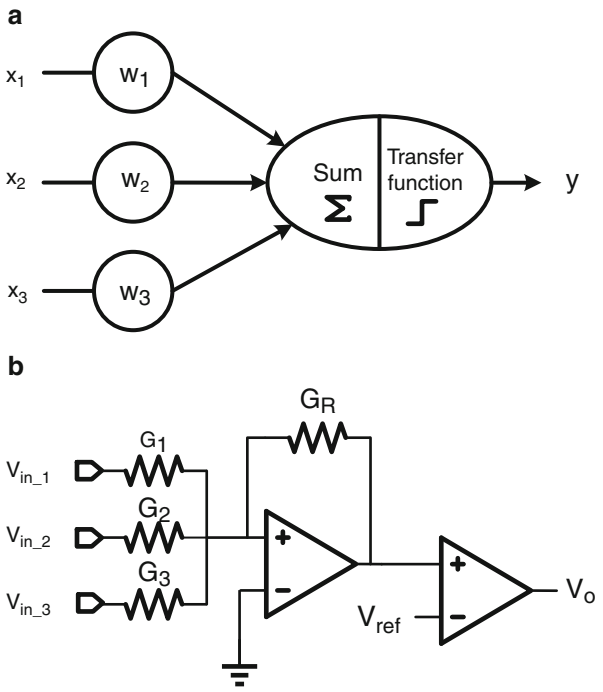


Fig. 4.38 (**a**) A neuron in feed forward neural network (**b**) CMOS implementation of neuron

physically achieved by a negative voltage drop on memristor which programs it from OFF-state to ON-state. This can be observed in the fourth subfigure. After training, the neutral sound stimulus then becomes conditional stimulus which can elicit conditional salivation response. The response is then weakened as there is no reward of food. The conditioning then disappears. This behavior is physically achieved because the memristor is programmed from ON-state back to OFF-state. Note that by scaling the settings the training speed and weakening speed can be controlled. For instance, a higher V_f will contribute to a faster training, and a lower V_s will slow down the conditioning weakening process.

4.3.2 Domain-Wall Neuron for Analog Learning

Neural network is one of the most commonly used learning paradigm, whose basic unit structure consisting of synapses and neuron is shown in Fig. 4.38a. As each synapse stands a weight, all the inputs will be weighted and their sum will be feeded to the neuron. The neuron has a transfer function, and the most typical one is a

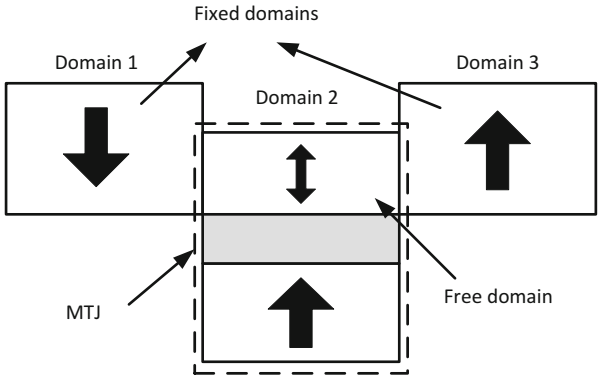


Fig. 4.39 A neuron implemented by a domain-wall nanowire device with modified structure

thresholding function: when the sum is above a threshold value then the output would be 1; otherwise -1 will be the output. The above function can be expressed as

$$V_o = f_t(\sum V_{in_i} G_i) \tag{4.23}$$

where V_{in_i} is the ith input voltage and G_i is the conductance of ith weight resistor. $f_t(x)$ is the transfer function:

$$f_t(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ -1 & \text{otherwise} \end{cases} \tag{4.24}$$

Its CMOS circuit hardware implementation is shown in Fig. 4.38b. The circuit has two stages. The first stage performs the weighted sum (dot product) operation by OPAMP, and the second stage achieves the transfer function by a comparator. Tt can be predicted that it would be extremely power consuming for a neural network with tens or hundreds of neurons.

While the ReRAM device has the ability to adapt its weight and work as synapse, the domain-wall nanowire-based neuron has been also investigated [23,24]. A nanowire that is used to implement neuron is different from the typical domain-wall nanowire structure in a few aspects. The domain-wall nanowire device as a neuron has a fixed number of three magnetic domains, as depicted in Fig. 4.39. Domain 1 at the left side and domain 3 at the right side are strongly magnetized as fixed domains, and domain 2 in the middle is free domain, whose spin polarity can be written parallel or antiparallel to the two fixed spin-domain 1 and 3, depending on the direction of current polarity. For example, magnetization of domain 2 will be polarized to upwards according to domain 1 when a current is flowing from left to right, and downwards when current flowing from right to left. Similar as typical domain-wall nanowire, an MTJ structure is also constructed as a read port to detect the state of domain 2.

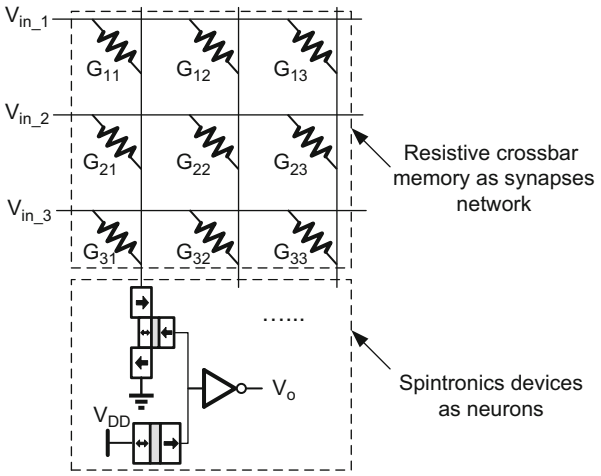


Fig. 4.40 A feed forward neural network with synapses implemented by resistive crossbar memory and neurons by domain-wall nanowire devices

Therefore, through the MTJ this device can detect the direction of current flow across its free domain, which is suitable to implement the transfer function of neuron. An ideal threshold in this operation to detect current direction is zero. However, due to the small hysteresis that exists in the magnetization switching characteristics, the domain-wall-based neuron will not behave exactly as the step transfer function of an ideal comparator. Practically by device scaling as well as the use of lower anisotropy barrier for the magnetic material, the small hysteresis will be weakened and the switching threshold can be effectively lowered to close-to-zero value [15].

By combining above domain-wall neurons with resistive crossbar synapses, a feed forward neural network can be efficiently implemented. A 3×3 neural network circuit using domain-wall neurons and resistive synapses is shown in Fig. 4.40. The resistance of each resistive memory device at cross-point of horizontal and vertical bars represents its synapse weight. One terminal of domain-wall neuron is connected with the end of each column and the other terminal is grounded. Due to the low-resistance value of domain-wall neuron compared to resistive nonvolatile devices, connecting point between crossbar and domain-wall neuron is virtually grounded as well. Therefore, by applying different voltage inputs at every row of crossbar, the inputs will be weighted by resistive synapse devices, and for each column the current will merge as input of its domain-wall neuron. The direction of merged current flow will be evaluated by domain-wall neuron and readout as output. The readout is achieved by a voltage divider as illustrated in Fig. 4.40, where a MTJ is used as resistance reference.

The energy dissipation for above neuron-synapse unit consists of Joule heat on resistive synapse devices and the read operation energy for MTJ. According to [23,

24], both energy terms are at sub-fJ scale. Compared to conventional analog circuits, the nonvolatile memory-based analog learning can achieve $\sim$100$\times$ lower power, and compared to 45 nm-CMOS digital ASIC, an improvement of $\sim$1,000$\times$ energy efficiency can be achieved [24].

References

1. Afifi A, Ayatollahi A, Raissi F (2009) Implementation of biologically plausible spiking neural network models on the memristor crossbar-based cmos/nano circuits. In: IEEE European Conference on Circuit theory and design, 2009. ECCTD 2009. pp 563–566
2. Ben-Jamaa MH, Gaillardon PE, Clermidy F, O'Connor I, Sacchetto D, De Micheli G, Leblebici Y (2011) Silicon nanowire arrays and crossbars: Top-down fabrication techniques and circuit applications. Sci Adv Mater 3(3):466–476
3. Borghetti J, Li Z, Straznicky J, Li X, Ohlberg DA, Wu W, Stewart DR, Williams RS (2009) A hybrid nanomemristor/transistor logic circuit capable of self-programming. Proc Natl Acad Sci 106(6):1699–1703
4. Chen A, Lin MR (2011) Variability of resistive switching memories and its impact on crossbar array performance. In: 2011 IEEE International Reliability physics symposium (IRPS), pp MY–7
5. Chen Y, Li H, Wang X, Zhu W, Xu W, Zhang T (2010) A nondestructive self-reference scheme for spin-transfer torque random access memory (stt-ram). In: Design, automation & test in europe conference & exhibition (DATE), 2010, IEEE, pp 148–153
6. Gopalakrishnan K, Shenoy R, Rettner C, Virwani K, Bethune D, Shelby R, Burr G, Kellock A, King R, Nguyen K, et al (2010) Highly-scalable novel access device based on mixed ionic electronic conduction (miec) materials for high density phase change memory (pcm) arrays. In: IEEE 2010 symposium on VLSI technology (VLSIT), pp 205–206
7. Hu XS, Khitun A, Likharev KK, Niemier MT, Bao M, Wang K (2008) Design and defect tolerance beyond cmos. In: Proceedings of the 6th IEEE/ACM/IFIP international conference on Hardware/Software codesign and system synthesis, ACM, Springer, New York, pp 223–230
8. Jeong G, Cho W, Ahn S, Jeong H, Koh G, Hwang Y, Kim K (2003a) A 0.24-μm 2.0-v 1t1mtj 16-kb nonvolatile magnetoresistance ram with self-reference sensing scheme. IEEE J Solid-State Circ 38(11):1906–1910
9. Jeong G, Cho W, Ahn S, Jeong H, Koh G, Hwang Y, Kim K (2003b) A 0.24-μm 2.0-v 1t1mtj 16-kb nonvolatile magnetoresistance ram with self-reference sensing scheme. IEEE J Solid-State Circ 38(11):1906–1910
10. Jo SH, Chang T, Ebong I, Bhadviya BB, Mazumder P, Lu W (2010) Nanoscale memristor device as synapse in neuromorphic systems. Nano lett 10(4):1297–1301
11. Kaeriyama S, Sakamoto T, Sunamura H, Mizuno M, Kawaura H, Hasegawa T, Terabe K, Nakayama T, Aono M (2005) A nonvolatile programmable solid-electrolyte nanometer switch. IEEE J Solid-State Circ 40(1):168–176
12. Kim KH, Gaba S, Wheeler D, Cruz-Albrecht JM, Hussain T, Srinivasa N, Lu W (2011) A functional hybrid memristor crossbar-array/cmos system for data storage and neuromorphic applications. Nano lett 12(1):389–395
13. Kügeler C, Meier M, Rosezin R, Gilles S, Waser R (2009) High density 3d memory architecture based on the resistive switching effect. Solid-State Electron 53(12):1287–1292
14. Lewis DL, Lee HH (2009) Architectural evaluation of 3d stacked rram caches. In: IEEE International Conference on 3D system integration, 2009, 3DIC 2009, pp 1–4
15. Morris D, Bromberg D, Zhu JGJ, Pileggi L (2012) mlogic: Ultra-low voltage non-volatile logic circuits using stt-mtj devices. In: Proceedings of the 49th Annual Design Automation Conference, ACM, pp 486–491

16. Mouttet BL (2007) Programmable crossbar signal processor. US Patent 7,302,513
17. Park WY, Kim GH, Seok JY, Kim KM, Song SJ, Lee MH, Hwang CS (2010) A pt/tio2/ti schottky-type selection diode for alleviating the sneak current in resistance switching memory arrays. Nanotechnology 21(19):195,201
18. Pershin YV, Di Ventra M (2010) Experimental demonstration of associative memory with memristive neural networks. Neur Netw 23(7):881–886
19. Pershin YV, La Fontaine S, Di Ventra M (2009) Memristive model of amoeba learning. Phys Rev E 80(2):021,926
20. Rowlands G, Rahman T, Katine J, Langer J, Lyle A, Zhao H, Alzate J, Kovalev A, Tserkovnyak Y, Zeng Z, et al (2011) Deep subnanosecond spin torque switching in magnetic tunnel junctions with combined in-plane and perpendicular polarizers. Appl Phys Lett 98(10):102,509–102,509
21. Schechter S, Loh GH, Straus K, Burger D (2010) Use ecp, not ecc, for hard failures in resistive memories. In: ACM SIGARCH computer rachitecture news, ACM, vol 38, Springer, Newyork, pp 141–152
22. Seevinck E, van Beers PJ, Ontrop H (1991) Current-mode techniques for high-speed vlsi circuits with application to current sense amplifier for cmos sram's. IEEE J Solid-State Circ 26(4):525–536
23. Sharad M, Fan D, Roy K (2013a) Spin-neurons: A possible path to energy-efficient neuromorphic computers. J Appl Phys 114(23):234,906
24. Sharad M, Fan D, Roy K (2013b) Ultra low power associative computing with spin neurons and resistive crossbar memory. In: Proceedings of the 50th Annual Design Automation Conference, ACM, p 107
25. Snider G (2007) Self-organized computation with unreliable, memristive nanodevices. Nanotechnology 18(36):365,202
26. Snider GS, Williams RS (2007) Nano/cmos architectures using a field-programmable nanowire interconnect. Nanotechnology 18(3):035,204
27. Tada M, Sakamoto T, Banno N, Aono M, Hada H, Kasai N (2010) Nonvolatile crossbar switch using tiox/tasioy solid electrolyte. IEEE Trans electron Dev 57(8):1987–1995
28. Thoziyoor S, Muralimanohar N, Ahn JH, Jouppi NP (2008) Cacti 5.1. HP Laboratories, April 2
29. Trinh HP, Zhao W, Klein JO, Zhang Y, Ravelsona D, Chappert C (2012) Domain wall motion based magnetic adder. Electron lett 48(17):1049–1051
30. Tu D, Liu M, Wang W, Haruehanroengra S (2007) Three-dimensional cmol: Three-dimensional integration of cmos/nanomaterial hybrid digital circuits. Micro Nano Lett, IET 2(2):40–45
31. Vontobel PO, Robinett W, Kuekes PJ, Stewart DR, Straznicky J, Williams RS (2009) Writing to and reading from a nano-scale crossbar memory based on memristors. Nanotechnology 20(42):425,204
32. Williams R (2008) How we found the missing memristor. IEEE Spectrum 45(12):28–35
33. Xu C, Dong X, Jouppi NP, Xie Y (2011) Design implications of memristor-based rram cross-point structures. In: Design, Automation and Test in Europe Conference and Exhibition (DATE), 2011, IEEE, pp 1–6
34. Zhao H, Lyle A, Zhang Y, Amiri P, Rowlands G, Zeng Z, Katine J, Jiang H, Galatsis K, Wang K, et al (2011) Low writing energy and sub nanosecond spin torque transfer switching of in-plane magnetic tunnel junction for spin torque transfer random access memory. J Appl Phys 109(7):07C720–07C720
35. Zhao H, Glass B, Amiri PK, Lyle A, Zhang Y, Chen YJ, Rowlands G, Upadhyaya P, Zeng Z, Katine J, et al (2012) Sub-200 ps spin transfer torque switching in in-plane magnetic tunnel junctions with interface perpendicular anisotropy. J Phys D: Appl Phys 45(2):025,001
36. Ziegler MM, Stan MR (2003) Cmos/nano co-design for crossbar-based molecular electronic systems. IEEE Trans Nanotechnol 2(4):217–230

Chapter 5
Nonvolatile Memory Computing System

Abstract The analysis of big-data at exascale (10^{18} bytes or flops) has introduced the emerging need to reexamine the existing hardware platform that can support memory-oriented computing. A big-data-driven application requires huge bandwidth with maintained low-power density. For example, web-searching application involves crawling, comparing, ranking, and paging of billions of web pages with extensive memory access. However, the current data-processing platform has well-known memory wall with limited accessing bandwidth but also large leakage power at advanced CMOS technology nodes. As such, a power-efficient memory-based design is highly desirable for future big-data processing. From memory design perspective, hybrid memory architecture can be built to exploit the strengths and avoid the weaknesses of different memory technologies. From logic computation perspective, nonvolatile memory based computing is favored to achieve power-efficient computing with high parallelism. In this chapter, the NVM system designs have been explored as potential solutions for future big-data computing platform.

Keywords Advanced encryption standard • Machine learning • Data retention • In-memory computing

5.1 Hybrid Memory System with NVM

3D die stacking [26] is promising to integrate hybrid memory components with high density and low latency. One can design a hybrid memory system with each tier by different memory technology stacked by through-silicon vias (TSVs) [26]. As such, advantages of different memory technologies can be leveraged with compact vertical integration. However, as leakage power is the primary concern of a memory system, one needs to have well-designed data-retention scheme such that power gating can be effectively deployed to reduce leakage power yet without degrading performance. Traditionally, the common approach for data retention of SRAM/DRAM [12] is to deploy a small retention voltage for all memory cells in sleep mode, which

H. Yu and Y. Wang, *Design Exploration of Emerging Nano-scale Non-volatile Memory*,
DOI 10.1007/978-1-4939-0551-5_5,

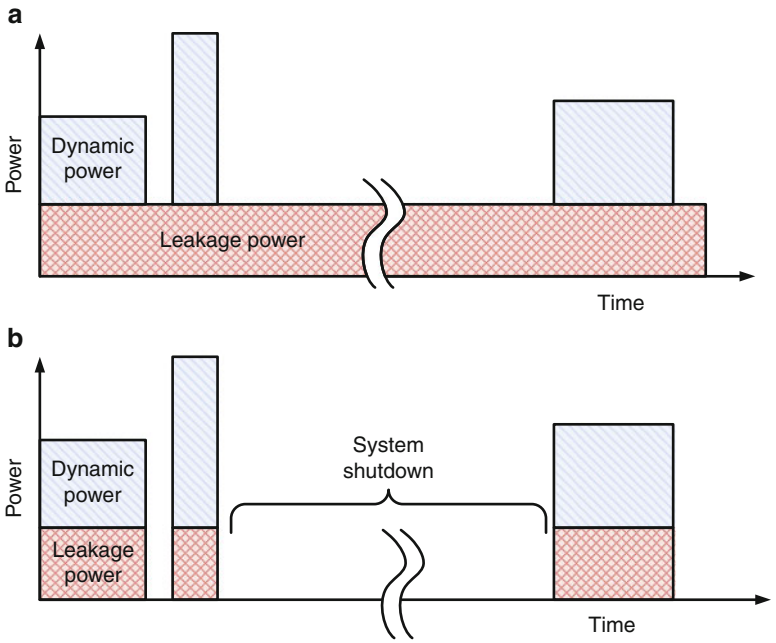


Fig. 5.1 (**a**) Conventional computing system power profile. (**b**) Ideal instant on/off computing system power profile

still has non-negligible leakage power. Recently, the work in [43] applies the nonvolatile PCRAM for *system-level* data retention of DRAM. DRAM layer and PCRAM layer are stacked in 3D fashion and connected by TSVs. Benefited from the increased number of vertical data paths, the bus bandwidth is significantly improved compared to 2D scenario. However, due to the asymmetric performance between DRAM and PCRAM, the data-transfer frequency and bandwidth are limited by the low write latency of PCRAM. Also, asynchronous hand-shaking protocol is required that incurs additional overhead. Another recent work in [21] has *bit-level* data retention by embedding one FeRAM device for each SRAM cell with bit-wise data-transfer controllers. Although concurrent bit-level data migration achieves fast speed, the overhead is overwhelmed with additional bit-wise data-transfer controllers in SRAM cells, which degrades the SRAM performance during the normal active mode. In this part, we will introduce a 3D hybrid memory architecture, in which CBRAM-crossbar is used to reduce system leakage power by block-level data retention.

The motivation of data retention by nonvolatile memory (NVM) is illustrated in Fig. 5.1. Without any power saving technique, the system will consume both dynamic power while it is executing and the leakage power when it is idle, and this results in a power profile as shown in Fig. 5.1a. This is because even when the system is idle, both memory and logic components of computing system will have leakage power. For memory, data needs to be retained in the memory for future

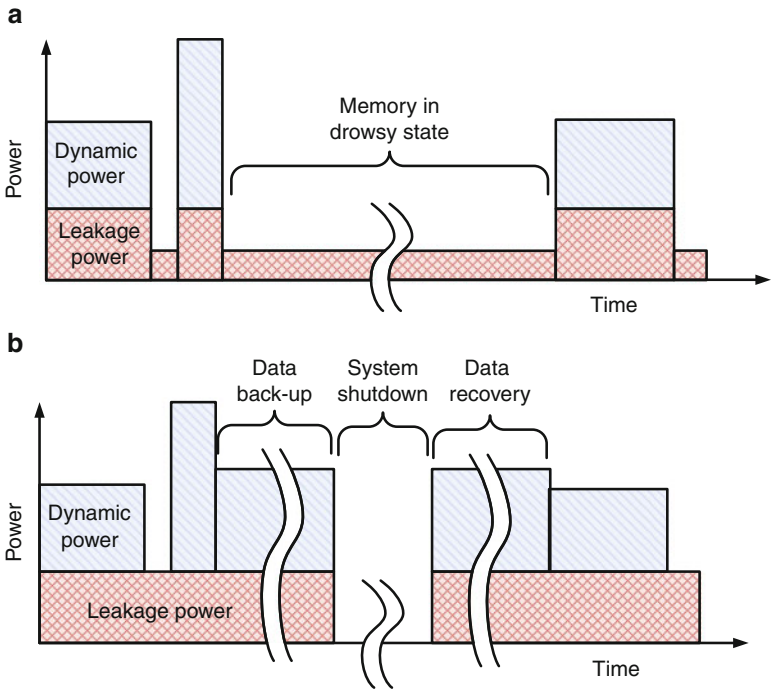


Fig. 5.2 Data retention for leakage reduction by (**a**) drowsy memory and (**b**) data backup and recovery

usage, and conventional volatile memory needs to be powered on always. Ideally, a system can be most power efficient by only turning it on while it is executing, and turning if off when idle, which leads to the power profile in Fig. 5.1b.

Practically, the logic components can be controlled to shut down when system is idle. However, the current volatile memory still has performance advantage and will still prevail; therefore the use of volatile memory will still bring inevitable leakage power. Nevertheless, the memory leakage power can be saved in two ways. The first way, shown in Fig. 5.2a, is to apply the drowsy memory technique and put memory into sleep mode when system is idle. In sleep mode, the memory supply voltage is reduced while it is still able to keep data, so that leakage power is, but not completely, reduced. The second way is to deploy data-retention scheme. In this scheme, the still can be totally shunt down, both for logic and memory, by migrating the memory content into hard-disk drive while it is about to be off and restoring when it is on. Its power profile is illustrated in Fig. 5.2b. Although a complete shut down can be achieved to avoid leakage power when it is idle, the data migration power overhead will be incurred. Therefore, which of the two ways is better depends on how frequently the system is in idle state and how long idle state will keep. In the

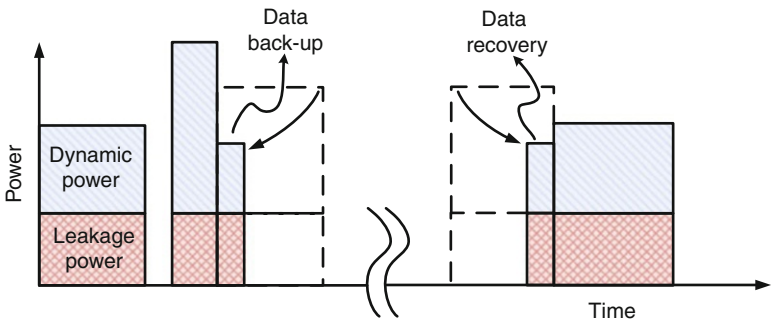


Fig. 5.3 More power-efficient instant on/off computing by incremental backup as well as 3D CBRAM-crossbar memory

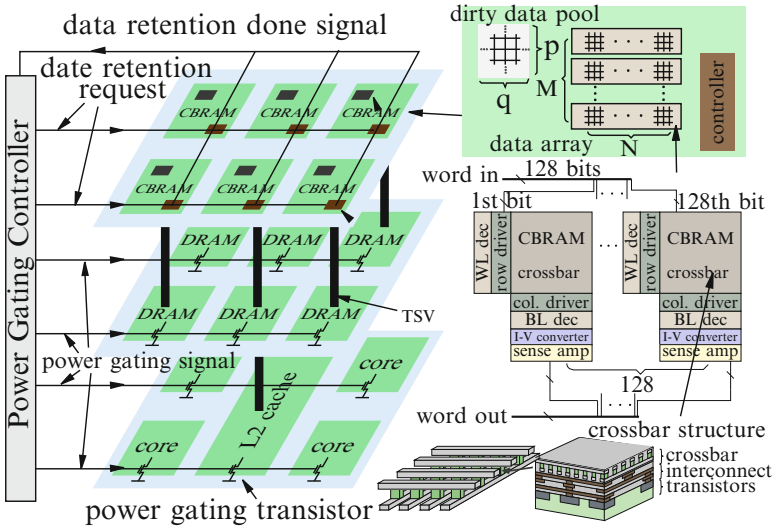


Fig. 5.4 3D hybrid memory system with CBRAM-crossbar-based data retention

following, we will show how the NVM can assist data-retention scheme to reduce data migration overhead, reaching a power profile of Fig. 5.3.

5.1.1 Overview of CBRAM-Based Hybrid Memory System

Figure 5.4 illustrates the overall system architecture of the proposed 3D hybrid memory, which is composed of embedded DRAM, SRAM, and CBRAM crossbar located at three layers (or tiers) connected by TSVs. Similar to the common memory organization [36], the entire CBRAM-crossbar memory is broken into *banks*, where

Table 5.1 Notations

Notation	Descriptions
T_{Hi}/P_{Hi}	Hibernating transition time/power for bank i
T_{Wi}/P_{Wi}	Wakeup transition time/power for bank i
P_b/P_m	Bank/mat power
E_b/E_m	Bank/mat access energy
C_b/C_m	Memory bank/mat capacity
N_b/N_m	Number of memory banks/mats
W_d/W_a	Data/address bus width of each bank
W_D/W_A	Total data/address bus width
M/N	Number of row/column mats of each bank
B_C/B_D	Cache line/DRAM page size

each bank can be accessed independently with dedicated data and address buses. Each bank is further broken into a $M \times N$ *mat* array, where each mat consists of one $m \times n$ CBRAM crossbar. In addition, the terms *sleep mode* and *active mode* are used to denote whether the system is power gated or not. The active to sleep mode transition is called *hibernating transition*, and term *wakeup transition* is used vice versa. For better presentation, Table 5.1 summarizes the notations used throughout this section.

To reduce the sneak-path power in crossbar structure [19], we propose to distribute the multi-bit data into different mats of the same row concurrently, where each mat (i.e., one CBRAM crossbar) accepts one bit of the written data at each cycle. To achieve this, the data address is decoded to find the exact same crossing point of each CBRAM-crossbar mat where one bit of the arriving data is to be kept. For read operation, the same address decoding is carried out and the bits from each mat are combined together as output. As such, during one hibernating transition, each SRAM/DRAM bank is associated with one dedicated CBRAM-crossbar bank of the same capacity for data retention. Therefore, power gating is employed at bank level (i.e., block level). Once the hibernating transition begins, all the data in the specified SRAM/DRAM bank must be copied or migrated to the corresponding CBRAM-crossbar memory bank through dedicated data and address buses implemented by TSVs in the vertical direction. On the other hand, all data must be migrated from CBRAM-crossbar back to the SRAM/DRAM during the wakeup transition.

However, the primary design challenges to develop the abovementioned 3D hybrid memory with CBRAM-crossbar are from twofold. Firstly, there is no design platform for CBRAM device and CBRAM-crossbar circuit such that the delay, area, and power can be estimated and optimized. Secondly, there is no memory controller developed and verified for the CBRAM-crossbar-based data retention, which can perform efficient data migration for SRAM/DRAM. In the following, we show the development of one design platform for CBRAM device and CBRAM-crossbar circuit. Moreover, we show one memory controller design for CBRAM-crossbar using an incremental block-level data retention.

5.1.2 Block-Level Incremental Data Retention

In this section, the proposed *block-level* data retention is discussed in details for the 3D hybrid memory architecture with CBRAM crossbar.

The data is migrated between memory blocks (i.e., banks) sequentially through dedicated 3D TSV buses. Compared to the *bit-level* data-retention scheme [21], where each SRAM memory cell is associated with one neighboring FeRAM cell with cell-wise controllers, our block-level approach can achieve much smaller area overhead since the data migration controller is shared by all memory cells of the same block. In addition, our block-level approach will not degrade the SRAM performance since no change is made inside the SRAM memory cell. Furthermore, [43] has a system-level data retention by updating checkpoints. Because the use of PCRAM limits the frequency and amount of data that are retained, the system in [43] can only keep system checkpoints between relatively longer time intervals, which is insufficient when more fine-grained data retention is required. Based on the CBRAM-crossbar structure, this section has introduced one block-level data retention, which includes the block-level memory controller with two operations: dirty-bit setup and incremental write-back.

5.1.2.1 Dirty Bit Setup

The target for data retention is to synchronize the data of CBRAM crossbar with corresponding SRAM/DRAM contents at block level. For any SRAM/DRAM bank i, the time needed to copy the data to CBRAM-crossbar is decided by

$$T_{Hi} = \frac{M_b}{f_b \cdot W_d}, \tag{5.1}$$

where f_b is the write-frequency of CBRAM-crossbar memory limited by its latency, W_d is the bank-level data bandwidth, and M_b is the amount of data to be migrated, which equals to the bank capacity C_b in the brute-force approach.

Clearly, reducing M_b directly reduces the T_{Hi}. Considering the common law of locality [11], the system tends to access relatively local-memory regions during a given period of time. As such, between two successive power gating stages, only part of the content in CBRAM-crossbar and SRAM/DRAM becomes unsynchronized, which is denoted as *dirty data* throughout this part. By only incrementally writing dirty data to CBRAM-crossbar, significant amount of migration data and power can be saved.

As shown in Fig. 5.4, to keep the dirty-data status information, an extra CBRAM-crossbar called *dirty-data pool* is embedded to each CBRAM-crossbar bank, where each bit in the pool, referred as *dirty bit*, indicates the dirty status of a few continuous bytes of data in SRAM/DRAM, referred as *dirty-data group*. Empirically, we design the group granularity G_d as the cache-line size B_C for SRAM or the page size B_D for DRAM.

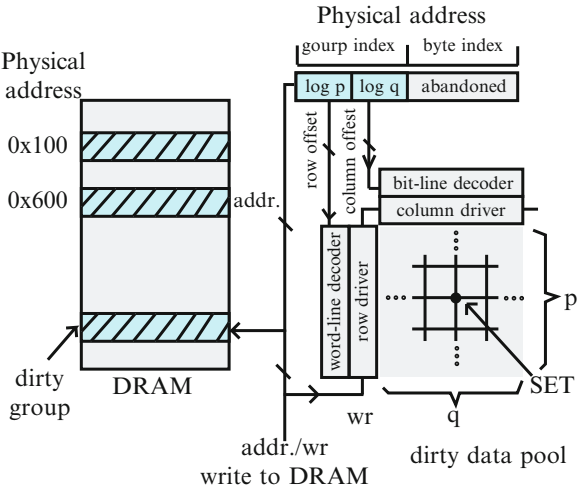


Fig. 5.5 Circuit diagram for dirty bit setup at active mode

The dirty-bit setup occurs simultaneously each time when the content of memory is changed during active mode. As shown in Fig. 5.5, each CBRAM-crossbar bank *listens* to all memory write operations issued to its corresponding SRAM/DRAM bank to update the dirty pool during active mode. Once the SRAM/DRAM write-action is detected, the corresponding data group becomes dirty.

As such, the corresponding bit in the dirty pool needs to be in SET state. The dirty pool size C_p is decided by

$$C_p = \frac{C_b}{G_d} = p \cdot q, \tag{5.2}$$

where p and q is the CBRAM-crossbar dimension of dirty pool. Therefore, the designated dirty bit position can be located by decoding the first $\log(p)$ and the following $\log(q)$ bits of the physical memory write-address, respectively.

5.1.2.2 Incremental Write-Back

The flow to write back dirty SRAM/DRAM data to CBRAM-crossbar during hibernating transition is illustrated in Fig. 5.6. Specifically, one address counter is used to check the status of all dirty bits in the dirty-data pool. Once the dirty bit in SET state is detected, the corresponding data group needs to be copied to the CBRAM-crossbar. Due to the limited data-bus bandwidth W_d, the group data of size G_d needs to be written back to CBRAM-crossbar in several cycles. As such, the address counter generates the memory address of the next piece of data to be copied from SRAM/DRAM by adding W_d-offset each cycle, and the read-signal to DRAM and write-signal to CBRAM-crossbar are issued for data migration.

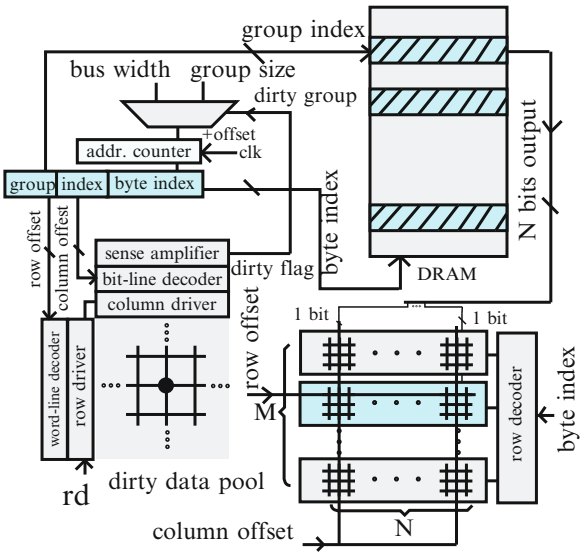


Fig. 5.6 Circuit diagram for dirty-data write-back at hibernating transition

Once finished, the corresponding dirty bit is RESET. During wakeup transition, all data will be copied from CBRAM-crossbar back to SRAM/DRAM with similar hardware supports.

Here we will discuss how the use of CBRAM-crossbar is able to instinctively facilitate the dirty-data write-back without additional design efforts. As discussed above, we can see that intensive bit operations are required for dirty flags update and read out. For the conventional 1T1R-structured memory, accesses are done in the unit of byte or word. Therefore, to read a bit, the byte or word which contains the target bit is first read out, then OR, AND, or SHIFT instruction is further applied to obtain one bit; and similarly to write a bit, the byte or word is first read out, then merged with the bit by OR or AND operation; and finally the bit-modified byte or word can be written back.

As such, the bit operations completed in multiple cycles may not be able to meet the real-time dirty flag update requirement. Also, significant power overhead may be incurred. Therefore, without additional design efforts, the byte-addressable or word-addressable conventional memory is not suitable for dirty-data write-back where intensive bit operations are required. On the other hand, as discussed in Sect. 4.1.1.1, bit operation is the instinctive way that CBRAM-crossbar is operated. Byte or word operations are achieved by multiple identical CBRAM-crossbar units to work in parallel. In other words, both bit operations required by dirty-data pool and word operations can coexist by using identical CBRAM-crossbar units naturally. Additionally from the physical design point of view, each CBRAM-crossbar block at top tier in Fig. 5.4 needs to have smaller size compared to their counterpart memory block in other tiers. This requirement is to ensure one vertical data path between the pairs and can be achieved since the CBRAM-crossbar has very high density.

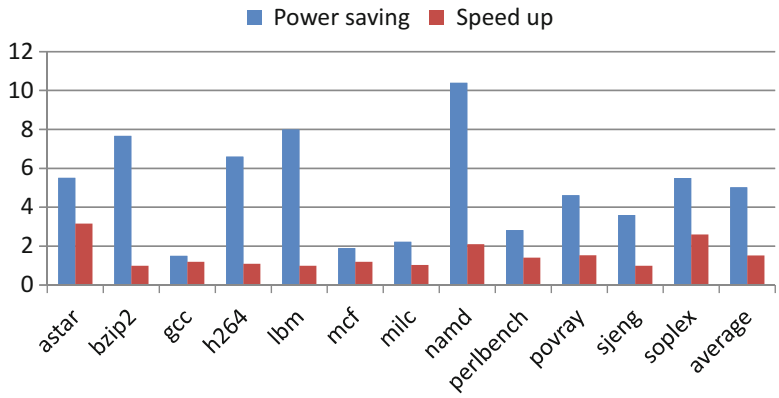


Fig. 5.7 Hibernating power and time reduction by incremental dirty-data write-back

Next, we evaluate the proposed block-level data retention with the use of incremental write-back. We also present the comparison with system-level and bit-level data retention, respectively.

In order to evaluate the dirty-data write-back strategy, a set of benchmark programs are selected from SPEC2006 suite and are run in gem5 simulator [4], where memory-access traces are generated. As the advantage of the dirty-data write-back strategy may depend on the memory-access patterns of executed programs, the benchmarks with different memory-access characteristics are picked in general. For example, mcf and lbm have high cache-miss rates while $h264$ and $namd$ have low cache-miss rates; $perlbench$ and gcc have intensive store instruction while $astar$ and $namd$ have low store instruction [5]. For each benchmark, its dirty-data flags are updated according to the memory-access trace. Then, the dirty-data write-back strategy is deployed to evaluate the power saving and speedup during data migration. Figure 5.7 compares the hibernating transition power and time with and without using dirty-data write-back strategy. Averagely, system with dirty-data write-back strategy achieves 5× reduction in power and 1.5× reduction in time during the hibernating transition.

5.1.3 Design Space Exploration and Optimization

5.1.3.1 Design Space Construction

In this part, based on the developed CBRAM-crossbar models, we further show how to optimize the 3D hybrid memory system by performing design space exploration under different optimization objectives.

Given design freedoms of parameters shown in Table 5.1, the total capacity of SRAM, DRAM, or CBRAM is calculated by

$$C_M = N_b \cdot C_b = N_b \cdot N_m \cdot C_m = N_b \cdot M \cdot N \cdot m \cdot n, \tag{5.3}$$

where $M \cdot N$ is the dimension of mat array for each bank and $m \cdot n$ is the dimension of crossbar array for each mat.

As indicated by Eqs. (5.1) and (5.3), different combinations of bank number, mat array dimension, and crossbar array dimension need to be explored for the optimal performance. For example, based on Eq. (5.1) it is desired to maximize the data-bus width W_d and CBRAM working frequency f_b. There are several design constraints to be considered as illustrated below one by one:

Data bandwidth constraint: as mentioned in Sect. 5.1.1, since each CBRAM-crossbar supports only one-bit-write for each cycle, we need to design the multi-bit data to be distributed into different crossbars at the same row. Consequently, the data-bus width W_d must equal to N, which is the number of crossbars (i.e., mats) that each row contains.

Transition power constraint: during the hibernating and wakeup transition, the concurrent data migration will induce high transition power:

$$\sum P_{bi} = \sum f_b W_{di} E_{mi} \leq P, \tag{5.4}$$

where W_{di} and E_{mi} is the bus-width and energy consumed of bank i. The transition power needs to be limited by reliability concerns for hot-spots of thermal and supply-current.

TSV density constraint: since the TSVs for data transmission occupy certain area, diameter in $5\mu m$, for example [15], they lead to larger footprint as well as difficulties in placement and routing. As a result, the average TSV density must be limited which leads to an upper bound on data bandwidth:

$$\frac{W_D + W_A}{A} \leq D, \tag{5.5}$$

where A is the chip area.

For our 3D hybrid memory system, one can perform design space exploration based on design parameters such as bank capacity C_b, data-migration bus bandwidth W_d, and CBRAM-crossbar write-frequency f_b. Figure 5.8 shows the performance objective with different trade-offs bounded by design constraints, which can be summarized as follows:

- For speed optimization, i.e., minimizing hibernating/wakeup transition time, large bandwidth is desirable. As such, small C_b, large W_d, and high f_b are preferred. However, it is limited by TSV density and power constraints.

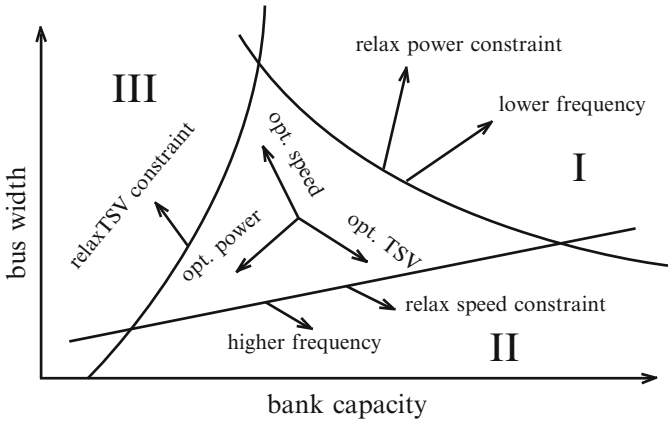


Fig. 5.8 Design space exploration of 3D hybrid memory with CBRAM-crossbar

- For power optimization, i.e., minimizing transition power, small C_b and W_d architecture working in a low f_b is favorable, which is mainly limited by TSV density and speed constraints.
- For memory performance optimization, i.e., minimizing memory SRAM/DRAM performance degradation due to high TSV density, small W_d and large C_b help reduce TSV density, which in turn alleviates memory performance degradation, mainly limited by speed constraint.

For example, regions I, II, and III in Fig. 5.8 are corresponding to the limiting design constraints of TSV density, power, and speed, respectively.

5.1.3.2 Optimization

This part shows the simulation experiment evaluation of the 3D hybrid memory from twofold. Firstly, the evaluation and optimization of the stacked CBRAM-crossbar memory is discussed. Based on the optimized CBRAM-crossbar memory design, the comparison between block-level data retention with the other schemes is then discussed.

TSVs length of 50 μm is assumed. The TSV power and delay models in [33] are integrated into the extended CACTI [36]. Here we adjust the design parameters C_b, W_d, and f_b under constraints of the maximal 10 W transition power, maximal 60 mm^2 TSV density, and maximal 3 ms transition speed.

Table 5.2 evaluates and compares the performance of the stacked CBRAM-crossbar memory with the other memory technologies for the same capacity of 16 MB. The data for SRAM and DRAM are generated by CACTI with default settings, and PCRAM data is extracted from PCRAMsim [7]. As shown in Table 5.2, mainly due to the fast device-level accessing speed by CBRAM device as well as the

Table 5.2 Performance comparison of 16 MB SRAM, DRAM, PCRAM, and CBRAM memories

Feature	SRAM	DRAM	PCRAM	CBRAM
Area (mm^2)	27.4	4.19	3.77	2.33
Read latency (ns)	3.43	2.25	2.54	3.90
Write latency (ns)	3.43	2.55	RESET 42.54 SET 102.54	8.01
Read energy (nJ)	0.83	0.61	0.73	1.8
Write energy (nJ)	0.75	0.61	RESET 6.66 SET 2.28	2.0

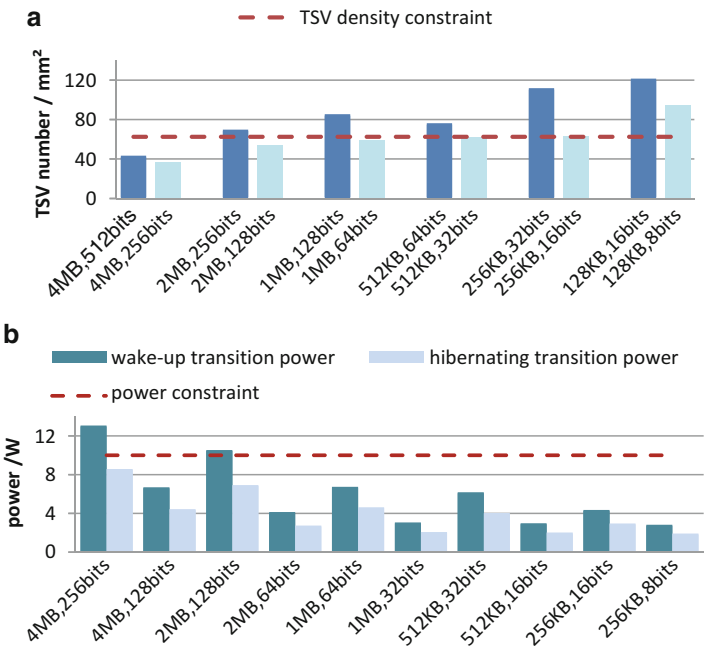


Fig. 5.9 (**a**) TSV density and (**b**) mode transition power under different architecture-level parameters

high density of crossbar structure, the CBRAM-crossbar performance, especially the accessing latency for read/write operations, is already close to DRAM, which shows its potential for the future application as the main memory. Moreover, compared to PCRAM, the CBRAM-crossbar shows 9× faster write latency, 1.6× smaller area, 4.5× less write energy per access, and 1.5× slower read latency on average. Only a slightly slower read latency is observed for the CBRAM-crossbar.

When employing CBRAM-crossbar for hybrid memory design, one needs to decide the optimal architecture-level parameters for performance objectives under certain constraints. As one example, we show the procedure of transition time (i.e., speed) optimization. Figure 5.9 shows the trend of TSV density and P_H/P_W with

Table 5.3 Optimized performance for different design objectives

Objective	Speed	Power	MP
C_b	256 KB	256 KB	4 MB
W_d (bits)	16	8	128
P_H/P_W (W)	2.3/9.0	***1.8/1.8***	4.3/7.1
T_H/T_W (ms)	***0.5/1.5***	3/3	1.9/3
TSV density (mm^2)	60	48	***26***

C_b/W_d. Among the available combinations, the one with C_b of 256 KB and W_d of 16 bits is chosen since it results in largest W_D and power margin when increasing frequency and when satisfying the defined constraints. As such, when working at its maximal allowed frequency, 0.5 ms T_W and 1.5 ms T_h can be achieved.

Table 5.3 demonstrates the optimal results for different performance objectives. An optimal 0.5 ms hibernating transition time and 1.5 ms wakeup transition time are achieved for the speed optimization. For transition power optimization, 5× less wakeup transition power and 1.6× less hibernating transition power are achieved with 6× wakeup transition time and 2× hibernating transition time penalties. For memory performance (MP in Table 5.3) optimization, the lowest TSV density is obtained as the minimized memory performance degradation in normal active mode. The design exploration leaves designer the freedom to choose different design parameters based on system requirements. The configuration with speed optimized is used in the following experiments.

5.1.4 Performance Evaluation and Comparison

The system under investigation is composed of one 2 MB level-2 SRAM cache, one 64 MB embedded DRAM, and one 66 MB CBRAM for data retention. We compare the performance of data retention and leakage power in sleep mode for the following data-retention schemes:

- *STD:* standard SRAM/DRAM without power gating
- *DPG:* data-retentive power gating (DPG) of both SRAM [32] and DRAM [30] with reduced supply voltage
- *PCRAM:* system-level data retention by PCRAM [43] which is used to keep the entire SRAM/DRAM contents
- *FeRAM:* bit-level data retention by FeRAM [21]
- *CBRAM:* our proposed block-level data retention with incremental dirty-data write-back by CBRAM

Table 5.4 shows the full comparison results in terms of sleep-mode leakage power for SRAM and DRAM (P_s^L/P_d^L), hibernating and wakeup transition time (T_H/T_W), and hibernating and wakeup transition power (P_H/P_W). The leakage power for SRAM/DRAM under standard scheme is generated by CACTI at 65 nm technology node. Leakage power under DPG scheme is calculated by the leakage reduction

Table 5.4 Data-retention performance comparison for different leakage reduction schemes

Scheme	P_s^L / P_d^L (mW)	T_H/T_W (ms)	P_H/P_W (W/MB)
STD	209/220	NA	NA
DPG	21/22	1e-4	0
PCRAM	0/0	3.7/1.3	0.072/0.1
FeRAM	0/22	1	0.75
CBRAM	0/0	0.33/1.5	0.007/0.14

Table 5.5 Cache performance comparison between block-level data retention (CBRAM) and bit-level data retention (FeRAM) in active mode

Cache capacity	Feature	Block-level CBRAM	Bit-level FeRAM
128 KB	Access time (ns)	1	1.8 −80%)
	Access energy (nJ)	0.13	0.21 (−62%)
	Area (mm^2)	0.82	7.6 (−827%)
512 KB	Access time (ns)	1.5	2.8 (−87%)
	Access energy (nJ)	0.33	0.6 (−82%)
	Area (mm^2)	3.8	28.6 (−642%)
2 MB	Access time (ns)	2.7	4.3 (−59%)
	Access energy (nJ)	0.6	1.3 (−117%)
	Area (mm^2)	14.2	113 (−696%)

factors reported in [30, 32] for both SRAM and DRAM. Block-level CBRAM-based data-retention performance is derived from our platform, combining the architecture optimization results and power and time reduction results. PCRAM-based scheme performance estimation is based on [43], with PCRAM memory performance obtained from PCRAMsim [7]. For FeRAM-based scheme, bit-to-bit data-retention performance is extracted from [21]. Because all data is migrated concurrently in this scheme, the data-retention power performance can be estimated by multiplication and speed performance is the same with bit-to-bit performance.

Due to the use of NVM for data retention, PCRAM-, FeRAM-, and CBRAM-based memory systems all outperform the STD and DPG schemes in terms of leakage power reduction. Moreover, our proposed 3D hybrid CBRAM-crossbar memory system allows a shutdown of both SRAM and DRAM during the power gating. As a result, compared to the block-level PCRAM-based data retention, we achieve 11× faster hibernating transition time and 10× smaller hibernating transition power with the same number of TSVs. As shown in Table 5.2, the hibernating transition performance improvement comes from the advantageous CBRAM-crossbar memory performance and also the block-level incremental dirty-data write-back strategy. However, the wakeup transition time and power are slightly inferior to PCRAM-based scheme. This is mainly because the incremental dirty-data write-back strategy does not apply to the wakeup transition. In addition, the proposed CBRAM system also outperforms the FeRAM by 107×/5.4× hibernating/wakeup transition power saving and around 3× faster hibernating transition time.

Further illustrated in Table 5.5, when compared with the block-level CBRAM-based data retention, the bit-level FeRAM-based data retention induces the

overwhelming cache performance degradation in normal active mode due to bit-wise controllers embedded in the SRAM. For bit-level FeRAM-based data-retention scheme, the area overhead for retaining one bit data is 6.1 um^2, which is only 0.36 um^2 in our CBRAM system, estimated by the developed CBRAM-crossbar-based CACTI. Due to such a large area overhead, the FeRAM-based system shows an 84% SRAM access time degradation, 74% SRAM access energy overhead on average. The performance of bit-level FeRAM-based cache is derived from CACTI by replacing SRAM cell with FeRAM-SRAM cell pair. Therefore, our CBRAM-crossbar-based data retention outperforms not only the system-level PCRAM-based data retention but also the bit-level FeRAM-based data retention.

5.2 In-Memory-Computing System with NVM

Domain-wall nanowire [31, 39, 40], or racetrack memory, is a newly introduced spintronic NVM device. It has not only potential for high-density and high-performance memory design but also interesting in-memory-computing capability. In the following, we will show how the domain-wall memory is exploited for in-memory computing.

5.2.1 Overview of Domain-Wall-Based In-Memory-Computing Platform

The analysis of big-data at exascale (10^{18} bytes or flops) has introduced the emerging need to reexamine the existing hardware platform that can support memory-oriented computing. A big-data-driven application requires huge bandwidth with maintained low-power density. For example, web-searching application involves crawling, comparing, ranking, and paging of billions of web pages with extensive memory access. However, the current data-processing platform has well-known memory-wall issue with limited accessing bandwidth but also large leakage power at advanced CMOS technology nodes. As such, an energy-efficient memory-based design is highly desirable for future big-data processing.

The current Von Neumann architecture has well-known memory-wall issue, which makes memory the bottleneck of whole computing system due to slow access latency of memory as well as the limited bandwidth for bus between computing resources and memory elements. The in-memory-computing architecture, as shown in Fig. 5.10, is promising as solution for the memory-wall, where lots of big-data-processing domain-specific accelerators are integrated into memory, so that the data will be preprocessed before they are read out to processor via I/O.

Furthermore, from memory technology perspective, emerging NVM technologies such as ReRAM, STT-RAM, and domain-wall nanowire racetrack memory

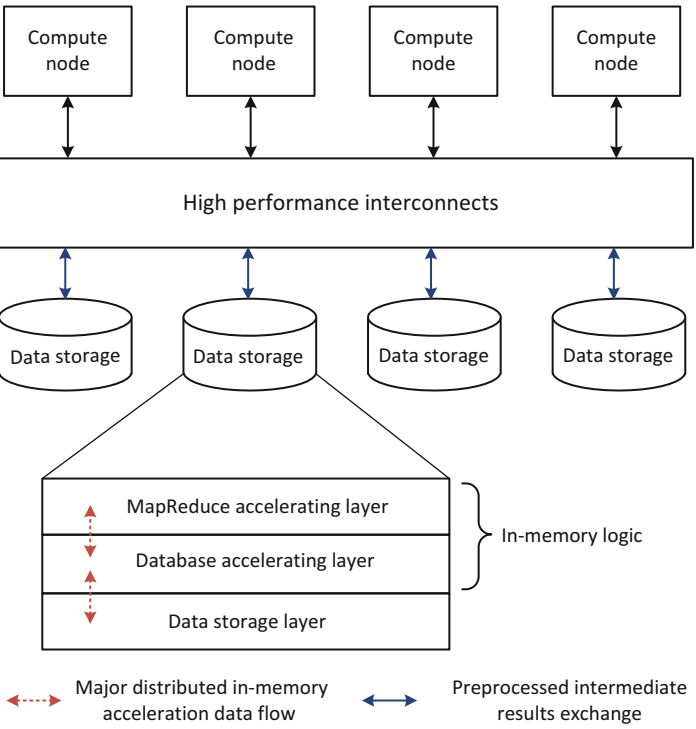


Fig. 5.10 The overview of in-memory architecture with distributed memory for data server

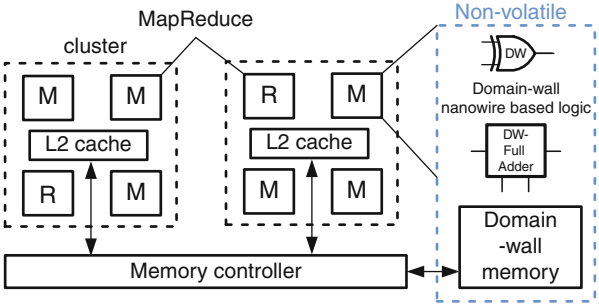


Fig. 5.11 The overview of the big-data computing platform by domain-wall nanowire devices

have been introduced recently, which show high power efficiency, high integration density, as well as close-to-DRAM/SRAM access speed. Among these emerging memory technologies, the domain-wall nanowire, or racetrack memory, not only shows potential as future universal memory, but it also can be exploited as logic units. One general purpose of nonvolatile in-memory-computing platform based on domain-wall nanowire is shown in Fig. 5.11. The big-data applications are

compiled by Map-Reduce-based parallel-computing model to generate scheduled tasks. A memory-based computing system is organized with integrated many-core microprocessor and main memory, which are mainly composed of the nonvolatile domain-wall nanowire devices. The many-core microprocessors are further classified into clusters. Each cluster shares an L2 cache and accesses the main memory by shared memory bus. Each core works highly independently for allocated tasks such as Map or Reduce functions.

In this platform, the domain-wall nanowire is intensively utilized towards the ultra-low-power big-data processing in both memory and logic, simultaneously. The domain-wall nanowire-based main memory can significantly reduce both the leakage and operating power of the main memory. What is more, large-volume of memory can be integrated with high density for data-driven applications. As such, one can build a hybrid memory system with CMOS-based cache as well as domain-wall nanowire-based main memory, whose compositions can be optimized by studying the accessing patterns under big-data applications. More importantly, the domain-wall nanowire is also explored for computing purpose. Specifically, the domain-wall nanowire-based XOR logic for comparison and addition is studied in details based on the following observations. Firstly, at instruction level, the web-searching-orientated big-data applications usually involve intensive string operations, namely, comparison, where the XOR and Adder logics will be visited more frequently than usual. Moreover, from logic level, the transistors to implement XOR gates in ALU account for more than half of the total number, due to its much higher complexity compared to the NAND, NOR, and NOT gates. As such, an optimized design of XOR logic by new technology such as domain-wall nanowire may provide the largest margin to optimize hardware for big-data processing.

5.2.1.1 DWM-Based Main Memory

There are two potential problems for such DWM macro-cell. Firstly, there exists variable access latencies for the bits that locate at different positions in the nanowire. Secondly, if the required bits are all stored in the same nanowire, very long access latency will be incurred due to the sequential access.

It is important to note that the data exchange between main memory and cache is always in the unit of a cache-line size of data, i.e., the main memory will be read-accessed when last-level cache miss occurs and will be write-accessed when a cache line needs to be evicted. Therefore, instead of the per access latency, the latency of the data block in the size of a cache line becomes the main concern. Based on such fact, we present a cluster-group-based data organization. The idea behind $cluster$ is to distribute data in different nanowires thus they can be accessed in parallel to avoid the sequential access; and the idea behind $group$ is to discard the within-group addressing, and transfer the $N_{\text{group}-\text{bits}}$ bits in $N_{\text{group}-\text{bits}}$ consecutive cycles, to avoid the variable latency. Specifically, a cluster is the bundle of domain-wall nanowires that can be selected together through bit-line multiplexers. The number of nanowires in one cluster equals the I/O bus bandwidth of the memory. Note that

the data in one cache-line have consecutive addresses. Thus, by distributing the bits of N consecutive bytes, where N is decided by the I/O bus bandwidth, into different nanowire within a cluster, the required N bytes can be accessed in parallel to avoid the sequential access. In addition, within each nanowire in the cluster, the data will be accessed in the unit of group, i.e. the bits in each group will be accessed in consecutive cycles with a similar fashion as DRAM.

The number of groups per nanowire is thus decided by

$$N_{\text{group-bits}} = N_{\text{line-bits}} / N_{\text{bus-bits}}. \tag{5.6}$$

For example, in system with cache-line size of 64 byte, and memory I/O bus bandwidth of 64 bit, the group size is 8 bit. As such, the DWM with cluster-group-based data organization can be operated in the following steps:

- Step1: The group head initially is aligned with the access port; thus the distributed first eight consecutive bytes can be first transferred between memory and cache.
- Step2: Shift the nanowire with 1-bit offset, and transfer the following eight consecutive bytes. Iterate this step 6 more times until the whole cache-line data is transferred.
- Step3: After the data transfer is completed, the group head is relocated to the initial position as required in step 1.

As mentioned in Eq. (3.40), the current-controlled domain-wall propagation velocity is proportional to the applied shift current. By applying a larger shift current, a fast one-cycle cluster head relocation can be achieved. In such a manner, the data transfer of cache block will be able to achieve a fixed and also lowest possible latency.

The domain-wall nanowire-based memory differs from the conventional CMOS-based memory in many aspects; thus CACTI has been extended with the domain-wall nanowire-based memory model for the DWM-based main memory, with accurate device operation energy and delay data obtained from NVM-SPICE. The memory configuration is shown in Table 5.6

Table 5.7 shows the 128 MB memory-bank comparison between CMOS-based memory (or DRAM) and domain-wall nanowire-based memory (or DWM). The number of access ports in main memory is varied for design exploration. The results of DRAM are generated by configuring the original CACTI with 32 nm technology node, 64 bit of I/O bus width with leakage optimized. The results of the DWM are obtained by the modified CACTI with the same configuration.

It can be observed that the memory area is greatly reduced in the DWM designs. Specifically, the DWMs with 1/2/4/8 access ports can achieve the area saving of 57%,70%,70% and 72%, respectively. The trend also indicates that the increase of number of access ports will lead to higher area saving. This is because of the higher nanowire utilization rate and is consistent with the analysis discussed in Sect. 4.1.4. Note that the area saving in turn results in a smaller access latency, and hence the DWM designs on average provide 1.9× improvement on the access latency.

Table 5.6 System configuration

Processor	
Technology node	65 nm
Number of cores	4
Frequency	1 GHz
Architecture	×86, O3, issue width - 4, 32 bits
Functional units	Integer ALU - 6
	Complex ALU - 1
	Floating point unit - 2
Cache	L1: 32 KB - 8 way/32 KB - 8 way
	L2: 1 MB - 8 way
	Line size - 64 bytes
Memory	
Technology node	32 nm
Memory size	2 GB - 128 MB per bank
IO bus width	64 bits

Table 5.7 Performance comparison of 128 MB memory bank implemented by different structures

Memory structure	Area (mm^2)	Access energy (nJ)	Access time (ns)	Leakage (mW)
DRAM	20.5	0.77	3.46	620.2
DWM/1 port	8.9	0.65	1.90	48.4
DWM/2 ports	6.2	0.72	1.71	30.1
DWM/4 ports	6.2	0.89	1.69	24.3
DWM/8 ports	5.7	1.31	1.88	19.0

However, the DWM needs one more cycle to perform shift operation, which will cancel out the latency advantage. Overall, the DWM and DRAM have similar speed performance. In terms of power, the DWM designs also exhibit benefit with significantly leakage power reduction. The designs with 1/2/4/8 access ports can achieve 92%, 95%, 96% and 97% leakage power reduction rates, respectively. The advantage mainly comes from the nonvolatility of domain-wall nanowire-based memory cells. The reduction in area and decoding peripheral circuits can further help leakage power reduction in DWM designs. In addition, the DWM designs have the following trend of access energy when increasing the number of access ports. The designs with 1/2 ports require 16% and 6% less energy, while designs with 4/8 ports incur 15% and 70% higher access energy cost. This is because when the number of ports increases, there are more transistors connected to the bit line which leads to increased bit-line capacitance.

In addition, as the data-orientated applications may have unique memory-access patterns, thus in order to study the memory dynamic power under different benchmarks, gem5 [4] simulator is employed to take both SPEC2000 and Phoenix benchmarks [35] and generate memory-accessing traces. The runtime dynamic power comparison under different benchmark programs are shown in Fig. 5.12a. It can be seen that the dynamic power is very sensitive to the input benchmark, and the results of the Phoenix benchmarks show no significant difference from

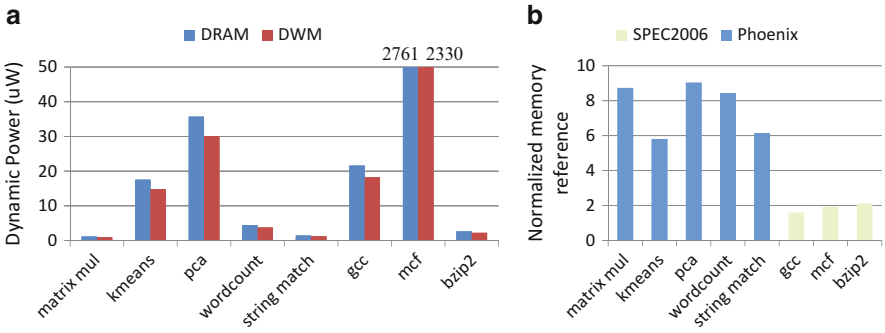


Fig. 5.12 (**a**) The runtime dynamic power of both DRAM and DWM under Phoenix and SPEC2006; (**b**) the normalized intended memory accesses

those in SPEC2006. This is because the dynamic power is effected by both the intended memory-access frequency and the cache-miss rate. Figure 5.12b shows the normalized intended memory reference rate, and as expected the data-driven Phoenix benchmarks have several times higher intended memory reference rate. However, both L1 and L2 cache-miss rates of Phoenix benchmarks are much lower than SPEC2006, which is due to the very predictable memory-access pattern when exhaustively handling data in Phoenix benchmarks. Overall, the low cache-miss rates of Phoenix benchmarks cancel out the higher memory reference demands, which leads to a modest dynamic power. Also, the runtime dynamic power contributes much less to the total power consumption compared to leakage power; thus the leakage reduction should be the main design objective when determining the number of access ports.

5.2.1.2 DWL-Based ALU

The DWL-XOR can be applied into the ALU design in two function units, the N-input XOR and N-input full adder for comparison and addition operations, respectively. Such two units account for more than half of the total transistors in ALU and also are the most frequently visited units, especially in big-data applications. The N-input XOR logic can be realized by employing N-bit-wise DWL-XOR, which is able to take the highly intensive $comparison$ instruction.

In the following, we present a full-adder design by DWL-XOR as shown in Fig. 5.13. As each CMOS-based XOR logic is built with four two-input NAND gates, the substitution by domain-wall nanowire logic (DWL)-based XOR logic thereby can reduce roughly three-quarters of the leakage power. Note that the addition of DWL-based full adder is also carried out in multiple cycles. Taking $A + B + C_{\text{in}}$ as an example, the full-adder executes in the following steps:

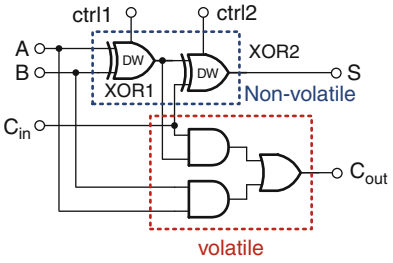


Fig. 5.13 Full-adder design with DWL-based XOR logic

- The operands A and B are loaded to the domain-wall nanowires in XOR1.
- The two operands are shifted to the read-only port and the internal $A \oplus B$ result is generated.
- The CMOS logic generates the C_{out} immediately.
- The internal $A \oplus B$ result and C_{in} are loaded into XOR2.
- The two operands are shifted to the read-only port and the sum $S = A \oplus B \oplus C_{\text{in}}$ can be obtained.

As such, by connecting N full adders together, an N-bit full adder can be achieved and integrated into the ALU. Note that there are two more control signals $crtl1$ and $crtl2$ used for nanowire operation control. A full adder is able to execute both ADD and SUB instructions; thus together with the N-input DWL-XOR, a very large portion of the instructions can be optimized in terms of power reduction. In addition, the stalls caused by the slightly longer cycles can be greatly suppressed in the out-of-order super-scalar processor.

In order to evaluate the DWL-XOR-based ALU design, gem5 simulator [4] is used to take both SPEC2000 and Phoenix benchmarks [35] and generate the instruction traces, which is then analyzed with the statistics of instructions that can be executed on the proposed XOR and adder for logic evaluation. McPAT [23] is then extended with additional power models of DWL-based XOR-logic. As such, by taking the instruction analysis from gem5, the extended McPAT is able to evaluate both accurate dynamic power and leakage power information of DWL-based ALU.

The system configuration is shown in Table 5.6. The 32-bit 65 nm processor is assumed with four cores integrated. In each core, there are six integer ALUs which are able to perform XOR, OR, AND, NOT, ADD, and SUB operations, and complex integer operations like MUL and DIV are executed in integer MUL. The instruction controlling decoder circuit is also considered during the power evaluation. The leakage power of both designs is calculated at gate level by the McPAT power model.

Figure 5.14 presents the per-core ALU power comparison between the conventional CMOS design and DWL-based design. Benefited from the use of DWL, both the dynamic power and leakage power can be greatly reduced. It can be observed that the set of Phoenix benchmarks consume higher dynamic power compared to those of SPEC2006, which is due to the high parallelism of Map-Reduce framework

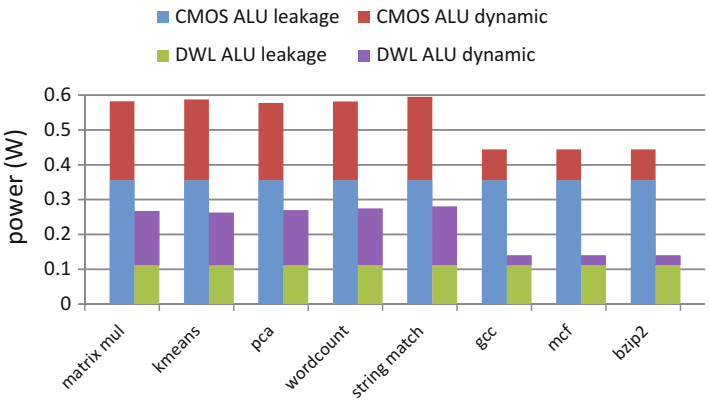


Fig. 5.14 The per-core ALU power comparison between the CMOS design and DWL-based design

with high utilization rate of the ALUs. Among each set, the power results exhibit a low sensitivity to the input, which indicates that percentages of instructions executed in the XOR and adder of ALU are relatively stable even for different benchmarks. The stable improvement ensures the extension of the proposed DWL to other applications. Averagely, a dynamic power reduction of 31% and leakage power reduction of 65% can be achieved for the ALU logic based on the eight benchmarks.

5.2.2 Matrix Multiplication

As shown in Fig. 5.15, the proposed NVM-based computing platform is composed of three parts. Firstly, nonvolatile domain-wall nanowire-based lookup table (DW-LUT) is utilized for configuring general logic. In this part, multiple LUTs are configured as different functions according to the program to be executed. Conventionally, program that intends to achieve complex functionality will be decomposed into basic instructions that the ALU can take by compiler. Therefore, it needs multiple clock cycles to be executed due to the decomposition. This compromises efficiency in order to gain better generality. However, in big-data applications, programs are usually intensive with a set of domain-specific functions without generality such that the coarse granularity can be introduced. With the basic functions implemented by the LUTs, programs can be executed with greatly augmented execution performance as well as power efficiency. Secondly, the physics of domain-wall nanowire device is exploited for special logic (DW-SL). Although most of the functions are covered by the LUTs, some commonly executed functions such as the XOR and shift can be economically implemented by domain-wall nanowire device directly. Thirdly, NVM is deployed for the data storage. Obviously, the three main parts of the proposed computing system are rich of nonvolatile

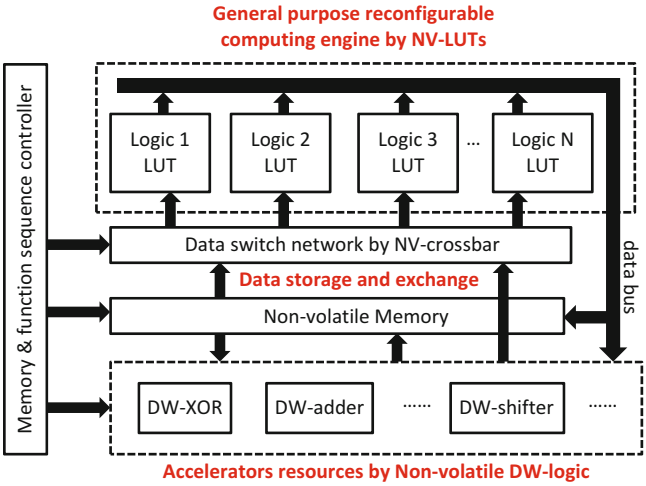


Fig. 5.15 System overview of the proposed NVM-based logic-in-memory-computing platform

devices, which can be significantly helpful to achieve high power efficiency and high bandwidth. In the following, we will explore detailed design by nonvolatile domain-wall nanowire device for each part.

5.2.2.1 Map-Reduce Programming Paradigm

Matrix multiplication is one of the essential functions in big-data applications like data mining and web searching. For instance, singular value decomposition (SVD), which can be used for deep learning in neural networks [22], involves iterations of matrix multiplication. Google PageRank, which intends to provide relative importance of billions of web pages according to searching query, also involves large amount of matrix multiplication operations [24]. In the following, we will show how matrix multiplication can be computed in parallel and how it can be mapped to the proposed platform.

Map-Reduce [6] is a parallel programming model to efficiently handle large volume of data. The idea behind Map-Reduce is to break down a large task into multiple sub-tasks, and each sub-task can be independently processed by different $Mapper$ computing units, where intermediate results are emitted. The intermediate results are then merged together to form the global results of the original task by the $Reducer$ computing units.

The problem to solve is $x = M \times v$. Suppose M is an $n \times n$ matrix, with element in row i and column j denoted by m_{ij}, and v is a vector with length of n. Hence, the product vector x also has the length of n and can be calculated by

$$x_i = \sum_{j=1}^{n} m_{ij} v_j .$$

Algorithm 1 Matrix multiplication in Map-Reduce form

```
function MAPPER(partitioned matrix p ∈ M, v)
    for all elements m_ij ∈ p do
        emit(i, m_ij v_j) to list l_i
    end for
end function

function REDUCER(key q, s ∈ l_q)
    sum ← 0
    for all values r_k ∈ s do
        sum ← sum + r_k
    end for
    emit(q, sum)
end function
```

The pseudo-code of matrix multiplication in Map-Reduce form is demonstrated in Algorithm 1. Matrix M is partitioned into many blocks, and each Mapper function will take the entire vector v and one block of matrix M. For each matrix element m_{ij} it emits the key-value pair $(i, m_{ij}v_j)$. The sum of all the values with same key will make up the matrix–vector product x. A reducer function simply has to sum all the values associated with a given key i. The summation process can also be executed concurrently by iteratively partial summing and emitting until one key-value pair left for each key, namely the (i, x_i).

5.2.2.2 Matrix Multiplication Task Mapping

Figure 5.16 shows how the Map-Reduce-based matrix multiplication is mapped into the proposed NVM-based computing platform. Before execution, the DW-LUTs are configured to execute integer multiplication. In addition, the matrix multiplication workload needs to be compiled into a task queue and stored into predefined region in the memory. The purpose of compilation is to break down the workload into smaller tasks that can be executed concurrently. In this example, the matrix M is partitioned in the unit of rows, so each task only requires dot product of two vectors. Each task instruction includes information like the entry address of specific row of matrix M and vector v, as well as length of row n.

When the computation of the matrix multiplication starts, the controllers, also achieved by the reconfigurable LUTs, will fetch the tasks from the task queue. For each task in the queue, the controller will fetch its required data according to the address, and then dispatch the multiplication tasks to mappers based on the availability. The controllers will keep fetching tasks, interpreting tasks, and dispatched tasks until the queue is empty. The DW-LUT in each mapper will compute the dispatched multiplication tasks and emit the <key, value> pairs as intermediate results. The <key, value> pairs are further combined by the reducers. Specifically, each reducer will take two pairs with same key from the intermediate results pool, and combine the value by addition, and emit a new pair to the

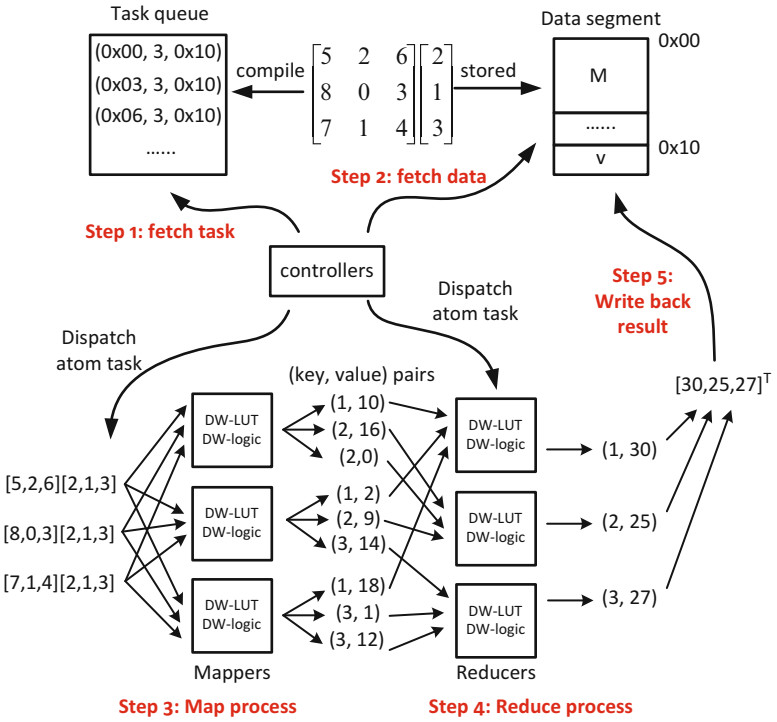


Fig. 5.16 Matrix multiplication mapping to proposed domain-wall nanowire-based computing platform

intermediate results pool. The core operation in reduce process is addition, which is mainly achieved by the DW-XOR-based adder. The reduce process works in an iterative manner, combining two pairs to one pair until the intermediate results cannot be further combined. As such, the final results are obtained, and the results write-back signify the end of the whole process.

5.2.2.3 Performance Evaluation and Comparison

Table 5.8 shows the power, area, and performance comparison between the proposed platform and conventional multi-core platform. The workload is the matrix multiplication in the scale of million by million, and all matrix elements are 8 bit integer numbers. In addition, serial output scenario is adopted, and each nanowire is composed of 32 bits, out of which 16 bits are used for storing results and the other 16 bits are reserved for shift operation, leading to an LUT array size of 64K for one multiplication operation. The comparison focuses on the computation resources and is exclusive in memory components. The 32 nm technology node is assumed for both scenarios.

Table 5.8 Platform comparison

General settings		
Platform	Multi-core	Proposed
Technology	32 nm	
Workload	1M×1M matrix multiplication	
Clock rate	3.4 GHz	500 MHz
Under 100W power budget		
Platform	Multi-core	Proposed
# of computing resources	8 cores	58716 LUTs+5963 adders
Performance	24.48 GOPS	917.44 GOPS
Area	145 mm^2	1292 mm^2
Under 145 mm^2 silicon area budget		
Platform	Multi-core	Proposed
# of computing resources	8 cores	6591 LUTs+669 adders
Performance	24.48 GOPS	102.98 GOPS
Power	100 W	11.23 W

The multi-core-based scenario consists of eight Xeon cores where the Map-Reduce-based matrix multiplication is executed. The evaluation flow is in two steps. Firstly, gem5 [4] simulator is employed to take Map-Reduce-based matrix multiplication from Phoenix benchmark suites [35] and to generate the runtime utilization rate of core components. Next, the generated statistics is taken by McPAT [23], which is able to provide core area, power, and performance results. For the domain-wall nanowire-based computing platform, the matrix multiplication is translated into the task list. The DW-LUT and DW-SL evaluation are gained from previous sections, and combined with the operation count, the performance of proposed platform can be estimated.

The results indicate that when power constraint is assumed for both systems, the proposed memory-based platform exhibits 37× higher throughput, but at the cost of 9× larger silicon area. This is because the NVM-based computing platform has very high power efficiency; thus more computation resources can be afforded to gain better performance. In the case if area constraint is adopted, proposed system shows 4.2× better performance and 88.77% less power consumption.

5.2.3 AES Encryption

Due to instant-on power-up and ultra-low leakage power, the newly introduced nanoscale NVM has shown great potential for future big-data storage. However, the sensitive data will not be lost during reboot or suspension and hence is susceptible to attack. Further, large volumes of data must be encrypted with high throughput and low power. Traditional memory–logic integration-based design incurs large

overhead when performing encryption by logic through I/Os. Therefore, in-memory encryption would be preferred to achieve high energy efficiency during data protection.

Advanced encryption standard (AES) is the most widely used encryption algorithm, and various CMOS-based hardware implementations for AES have been presented [16, 25]. In scenarios where energy efficiency is critical, CMOS-based ASIC implementations tend to incur significant leakage power in current deep submicron regime with limited throughput. In [1], a memristive CMOL implementation by hybrid CMOS and ReRAM design is introduced to facilitate AES application. However, while the ReRAM serves as reconfigurable interconnection, it is not used for in-memory computation-based encryption.

As spintronic devices have shown great scalability [42], it is promising to build big-data storage with in-memory logic-based computing such as encryption. In this work, we propose a full domain-wall nanowire-device-based in-memory AES computing, called DW-AES. The nonvolatile domain-wall nanowire devices are both used as storage element and deployed for logic computing in AES encryption. For example, *ShiftRow* transformation is facilitated by the unique shift operation of the domain-wall nanowire; *AddRoundKey* and *MixColumns* transformations benefit from the domain-wall nanowire-based XOR logics (DW-XOR); and *SubBytes* and *MixColumns* transformations are assisted by the domain-wall nanowire-based lookup table (DW-LUT). As such, all four fundamental AES transformation can be fully mapped to the nonvolatile domain-wall nanowire-based design.

5.2.3.1 Advanced Encryption Standard

In the AES, the standard input length is 16 bytes (128 bits), which are internally organized as a two-dimensional four rows by four columns array, called *state matrix*. During the AES algorithm, a sequence of transformations are applied to the state matrix, after which the input block is considered encrypted and then output. The flow chart of the AES algorithm is shown in Fig. 5.17. For simplicity, the initial round and final round are not shown, which slightly differ from the iterative rounds. In order to study the AES from hardware implementation point of view, the hardware complexity is analyzed for each transformation module as well as the dominant resources within each module. Gate utilization data is obtained by synthesizing public domain AES Verilog code from [38]. Each module is briefly described as follows:

- SubBytes: each byte in the state matrix will be updated by a nonlinear transformation independently. The nonlinear transformation is often implemented as a substitution table, called S-box. S-box takes one byte as input and outputs one byte to its original position. The SubBytes module accounts for half of the total gates in the AES. Within this block the dominant hardware resources are registers, used as lookup table (LUT) elements.

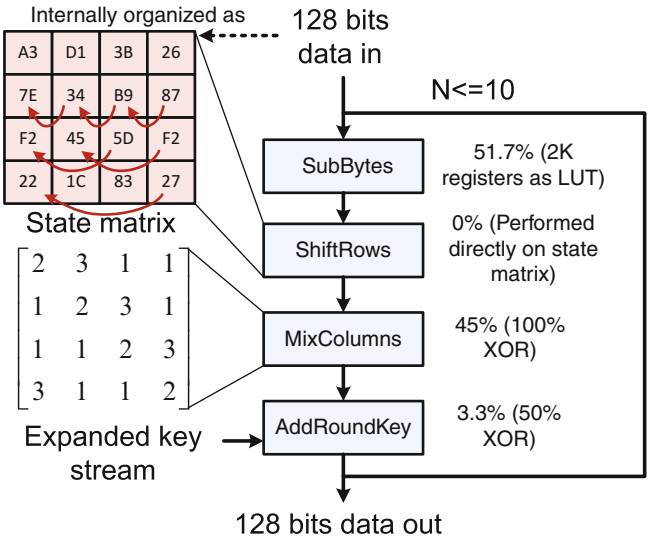


Fig. 5.17 The flow chart of the AES algorithm with gate utilization analysis

- ShiftRows: each row of the state matrix will be cyclically shifted left by different offset values. As shown in Fig. 5.17, the top row is not shifted, the second row is shifted by one byte position, the third by two, and the fourth row by three. In ASIC design, the ShiftRows transformation does not require additional logic gates; it can be performed on shift registers where the state matrix is held.
- MixColumns: each column of the state matrix is multiplied by the known matrix as shown in Fig. 5.17. The multiplication operation is defined as: multiplication by 1 means no change, multiplication by 2 means left shift, and multiplication by 3 means left shift and then XOR with the initial unshifted value. The step serves as an invertible linear transformation that takes four bytes in a column as input and outputs four bytes to their original position, where each input byte affects all four output bytes. This module accounts for nearly half of total gates, where all are XOR gates.
- AddRoundKey: state matrix is combined with the round keys. The round keys are also 16 bytes and organized in 4×4 array fashion as state matrix, and each byte of state matrix will be updated by bit-wise XOR with its counterpart byte in the round key matrix. Therefore, the AddRoundKey module is purely built by XOR gates, which accounts for 3.3% of the total gates.

Although in different designs the percentages may vary, the basic operations are without exception XOR, shift, and table lookup. This motivates the utilization of the domain-wall nanowire devices, which is ideally suited to the AES encryption application.

5.2.3.2 AES Task Mapping

In-memory encryption offers two major advantages over existing approaches. Firstly, all domain-wall-based AES ciphers (DW-AES) can be integrated inside the memory, and AES encryption is performed directly on target data stored in nonvolatile domain-wall memory. This is significantly different from the conventional memory–logic architecture in which the nonvolatile storage data to process must be loaded into volatile main memory, processed by logic, and written back afterwards. Secondly, the DW-AES cipher is implemented purely by domain-wall nanowire devices, which are identical to the storage elements. This provides good integration compatibility between the DW-AES ciphers and the memory elements, as well as the ability to reuse peripheral circuits like decoders and sense amplifiers. In this section, the detailed domain-wall nanowire-based in-memory encryption will be discussed.

Data Organization of State Matrix

Because in-memory encryption is performed directly on data cells, the data needs to be organized in certain fashion to facilitate the AES algorithm. As discussed in Sect. 3.3.3, domain-wall nanowires only support serial access, that is, one bit of information can be accessed from a domain-wall nanowire at one time. In order to access multiple bits within one cycle, the data needs to be distributed into separate nanowires so that they can be operated concurrently. In the AES algorithm, the basic processing unit is each byte in the state matrix. Therefore, the state matrix is split into eight 4×4 arrays, as illustrated in Fig. 5.18, where each entry of each array becomes one bit instead of one byte. By distributing the bytes and operating eight arrays together, the byte access requirement in the AES algorithm is satisfied. In addition, to facilitate the ShiftRows transformation by exploiting the shift property of domain-wall nanowire, each row of an array needs to be stored within one domain-wall nanowire. In this case, each array is composed of four nanowires, and within each nanowire, the four bits data are kept along with some redundant bits used for efficient circular shift. Details regarding redundant bits will be discussed later in ShiftRows transformation. By organizing each 16 bytes of data in the above manner, the AES algorithm can be applied efficiently.

SubBytes

In this step, each byte in the state matrix will undergo an invertible nonlinear transformation. This transformation is commonly implemented as an LUT, called substitution box (S-box). S-box LUT, essentially a pre-configured memory array, takes 8 bit input as a binary address, finds target cells that contain 8 bit result through decoders, and finally outputs correspondingly by sense amplifiers. With 2^8

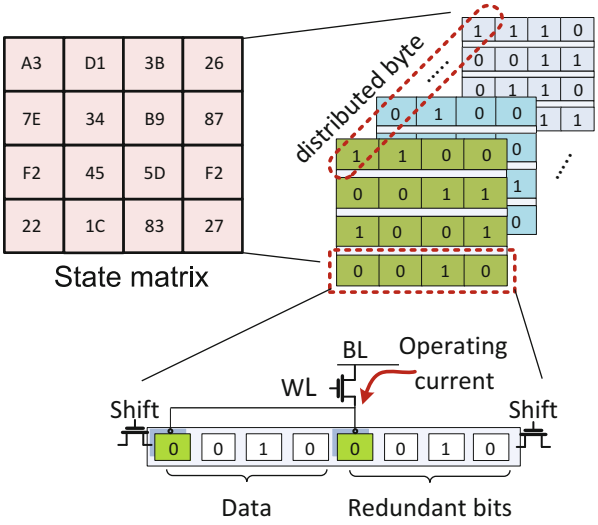


Fig. 5.18 Data organization of state matrix by domain-wall nanowire devices in distributed manner

possible input scenarios, and each scenario having 8 bit result, the LUT size can be determined as $2^8 \times 8 = 2,048$ bits. The LUT is conventionally implemented by SRAM cells, which in this size will incur significant leakage power.

In our proposed DW-AES design, the LUT is implemented by nonvolatile domain-wall nanowire devices, i.e., DW-LUT, as shown in Fig. 5.19. By distributing the 8-bit results in separated nanowires, the transformation can be done fast. In this parallel output scenario, eight sense amplifiers are required for each DW-LUT. As such, SubBytes transformation can be realized in a nonvolatile fashion, that is, both the in-memory state matrix and S-box are made up of nonvolatile domain-wall nanowires, which will enable significant leakage reduction. In addition, the memory and DW-LUT can share decoders and sense amplifiers, which leads to further power and area savings.

ShiftRows

The ShiftRows transformation can be efficiently achieved by exploiting the unique shift property of domain-wall nanowire. Due to the distributed data organization, in the ShiftRows transformation, the second row needs to be left shifted cyclically by one bit, the third row by two bits, and the fourth row by three bits, while the top row remains unshifted. In order to accomplish the circular shift in an elegant manner, i.e., without writing back the most significant bits to the least significant bits, redundant bits are used to form a virtual circle on the nanowire, as illustrated in Fig. 5.20.

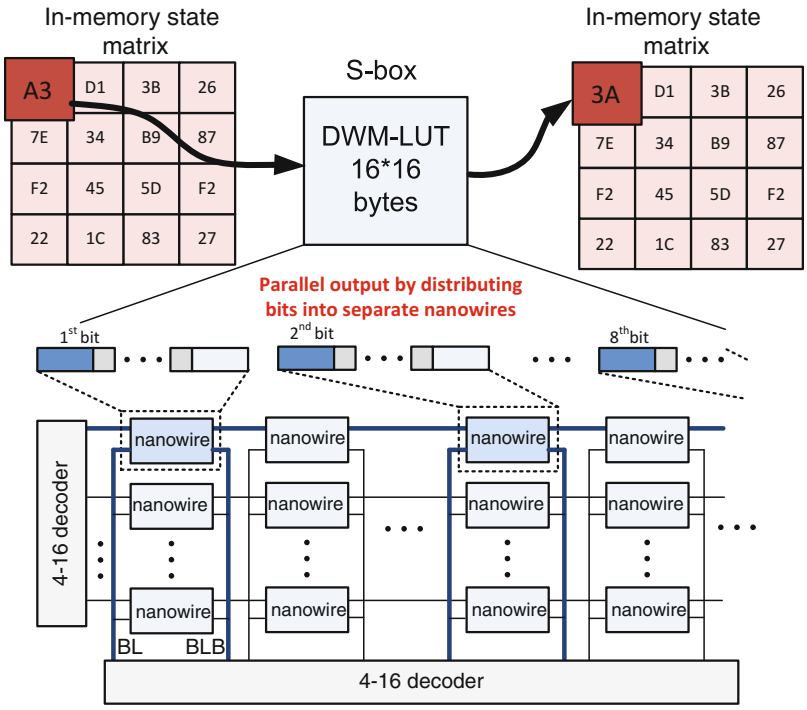


Fig. 5.19 SubBytes step with S-box function achieved by domain-wall memory-based lookup table

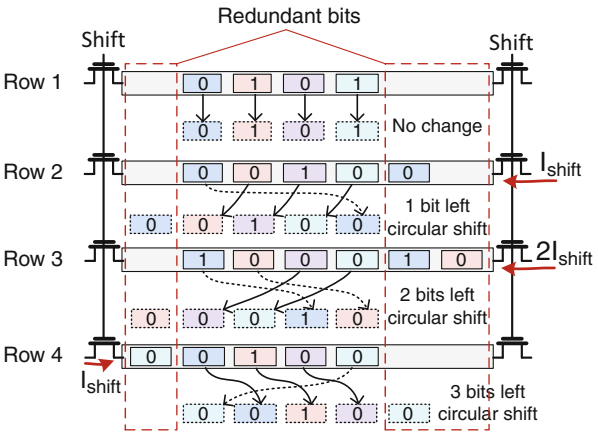


Fig. 5.20 ShiftRows transformation by domain-wall nanowire shift operations

Since each row has predetermined shift operation, the number of redundant bits of each row can be readily determined: one redundant bit is required for the second row, two bits for the third row, and three bits for the last row, all attached to the least significant bit from the right side. In order to achieve all shifts in one cycle, shift currents of different amplitude are applied to each row according to the linear current–velocity relationship of shift operation [3]. In other words, the third row and fourth row are the applied shift current that is twice and three times the amplitude applied to the second row. Consider the equivalent operations in the circular shift with four bits: $LS1 \stackrel{\text{def}}{=\!=} RS3$, $LS2 \stackrel{\text{def}}{=\!=} RS2$, $LS3 \stackrel{\text{def}}{=\!=} RS1$, where *LS* and *RS* indicate the left and right shift and the number denotes the length to shift. This means in the last row, instead of shifting 3 bits leftward, only right shift 1 bit needs to be performed. This helps to reduce the redundant data from 3 bits to 1 bit, as well as to reduce the applied shift current to one-third of previously required amplitude. The bits in the same color indicate that they are synchronized bits. To ensure correct circular shift, the redundant bits need to be synchronized with their counterparts. As a result, during changes in the matrix state, the redundant bits must also be updated.

In contrast with conventional computing flow, in which data needs to be moved to computing units for execution and written back to memory afterwards, the ShiftRows transformation is done directly on the stored data by in-memory-computing fashion.

AddRoundKey

In the AddRoundKey step, each byte in the state array will be updated by bit-wise XOR with corresponding key byte. As the dominant operation in this step is XOR, we propose a nanowire-based XOR logic (DW-XOR) for leakage free computing. As described in Sect. 3.3.3, the GMR-effect can be interpreted as the bit-wise-XOR operation of the magnetization directions of two thin magnetic layers, where the output is denoted by high or low resistance. In a GMR-based MTJ structure, however, the XOR logic will fail as there is only one operand as variable since magnetization in the fixed layer is constant. This problem is overcome by the unique domain-wall shift operation in the domain-wall nanowire device, which enables DW-XOR for computing.

The AddRoundKey with bit-wise-XOR logic implemented by two domain-wall nanowires is shown in Fig. 5.21. The proposed bit-wise DW-XOR logic is performed by constructing a new read-only port, where two free layers and one insulator layer are stacked. The two free layers each have the size of one magnetization domain and are from two respective nanowires. Thus, the two operands, representing the magnetization direction in each free layer, can both be variables with values assigned through the MTJs of their own nanowire. These assigned values are then shifted to the operating port such that the XOR can be performed.

Given A and B are 1 bit operands from state and key byte, respectively, and eight identical DW-XORs are used for bit-wise XOR between state byte and key byte, the $state \oplus key$ can be executed in the following steps:

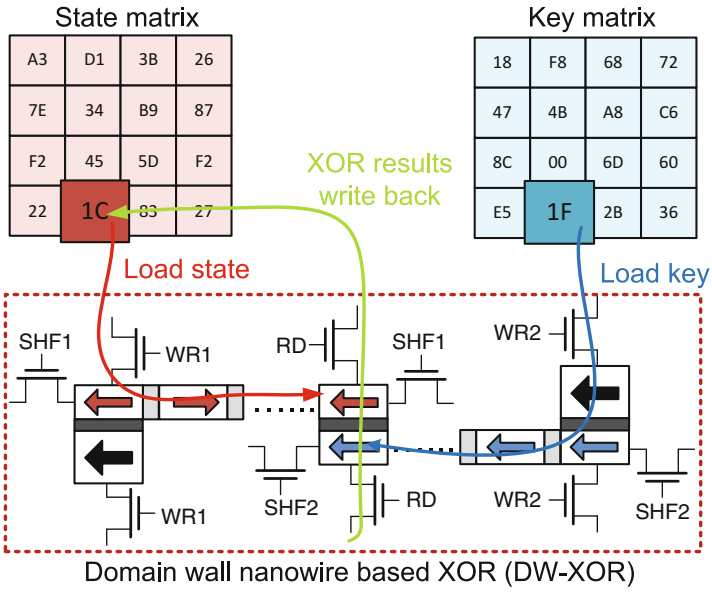


Fig. 5.21 AddRoundKey step with XOR logic achieved by domain-wall nanowire

- The A and B are loaded into two nanowires by enabling WL_1 and WL_2, respectively.
- A and B are shifted from their access ports to the read-only ports by enabling SHF_1 and SHF_2, respectively.
- By enabling RD, the bit-wise-XOR result can be obtained through the GMR effect.

MixColumns

The MixColumns transformation can be expressed as the state matrix multiplied by the known matrix shown in Fig. 5.17. Specifically for the ith column x_i $(0 \leq i \leq 3)$, the column after transformation becomes

$$
\begin{aligned}
y_{0,i} &= 2 * x_{0,i} \oplus 3 * x_{1,i} \oplus x_{2,i} \oplus x_{3,i} \\
y_{1,i} &= x_{0,i} \oplus 2 * x_{1,i} \oplus 3 * x_{2,i} \oplus x_{3,i} \\
y_{2,i} &= x_{0,i} \oplus x_{1,i} \oplus 2 * x_{2,i} \oplus 3 * x_{3,i} \\
y_{3,i} &= 3 * x_{0,i} \oplus x_{1,i} \oplus x_{2,i} \oplus 2 * x_{3,i}.
\end{aligned}
$$

The operations needed are multiplication by two (*xtime-2*), multiplication by three (*xtime-3*), and bit-wise XOR. The *xtime-2* is defined by left shift by 1 bit, and bit-wise XOR with *0x1B* if the most significant bit is 1. The *xtime-3* is defined

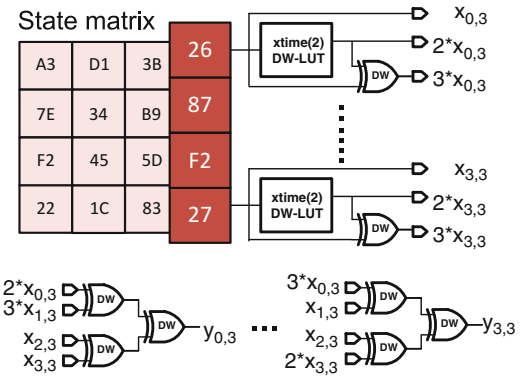


Fig. 5.22 MixColumns transformation by DW-LUT and DW-XOR

as *xtime-2* result XOR with its original value . Therefore, there are only two de facto atomic operations: (1) bit-wise XOR, executed by proposed DW-XOR, and (2) *xtime-2*. Although *xtime-2* can be implemented by in-memory shift together with additional DW-XOR, it is more efficient to use 8-bit input 8-bit output DW-LUT due to its branch operations depending on its most significant bit. As such, the MixColumns transformation can be purely performed by DW-LUT and DW-XOR, as shown in Fig. 5.22.

5.2.3.3 Performance Evaluation and Comparison

To evaluate DW-AES cipher, the following design platform has been set up. Firstly at device level, the transient simulation of MTJ read and write operations are performed within NVM-SPICE to obtain an accurate operation energy and timing for domain-wall nanowire. The shift-operation energy is modeled as the Joule heat dissipated on the nanowire when shift current is applied. The shift-current density and shift-velocity relationship are based on [3]. The area of one domain-wall nanowire is calculated by its dimension parameters. Specifically, the technology node of 32 nm is assumed with width of 32 nm, length of 64 nm per domain, and thickness of 2.2 nm for one domain-wall nanowire; the R_{off} is set at 2,600 Ω, the R_{on} at 1,000 Ω, the writing current at 100 μA, and the current density at 6×10^8 A/cm^2 for shift operation. Secondly at circuit level, the memory modeling tool CACTI [36] is modified with name as DW-CACTI. It can provide accurate power and area information for domain-wall nanowire memory peripheral circuits such as decoders and sense amplifiers (SAs). Together with the device level performance data, the DW-XOR as well as the DW-LUT can be evaluated at circuit level. The additional sequential controller of DW-AES is described by Verilog HDL, which is synthesized with area and power profiles. Finally at system level, an AES behavioral simulator is developed to emulate the AES cipher, as well as to explore the trade-offs among power, area, and speed.

Table 5.9 AES for 128 bits encryption performance comparisons

Implementation	Leakage (μW)	Total power power(μW)	Area (μm^2)	cycles
C code [27] on GPP	1.3e+6	4e+5	2.5e+6	2309
CMOS ASIC [16]	120.54	154.74	953.05	534
Memristive CMOL [1]	102.35	119.04	251.5	534
DW-AES	14.602	21.568	78.121	1022

The proposed DW-AES cipher is compared with both CMOS-based ASIC design [16, 25] and hybrid CMOS/ReRAM (CMOL) design [1]. For these implementations, performance data is extracted from the reported results in [1, 16, 25] with necessary technology scaling. C-code-based software implementation that runs on a general purpose processor (GPP) is also compared. Evaluation of the AES software implementation is done in two steps. Firstly, gem5 [4] simulator is employed to take AES binary, compiled from C-code obtained from [27], which generates the runtime utilization rate of core components. Next, the generated statistics are taken by McPAT [23], which provides core power and area model. All hardware implementations run at the clock rate of 3 MHz, while the processor is operated at 2 GHz for the software implementation. Table 5.9 compares the different implementations of AES cipher, and the results are discussed as follows.

- Power: as expected, the DW-AES cipher has the smallest leakage power due to the use of nonvolatile domain-wall nanowire devices. The remaining small leakage power is introduced by its CMOS peripheral circuits, i.e., decoders, sense amplifiers, as well as simple sequential controllers. Specifically, DW-AES cipher achieves a leakage power reduction of 88% and 86% compared to the CMOS ASIC and memristive CMOL designs, respectively. The leakage power can be further reduced if the decoders and SAs of the memory can be reused by the DW-AES ciphers.
- Area: the area breakdown with resource utilization for each module in DW-AES is illustrated in Fig. 5.23. Benefitting from the high density of domain-wall nanowire devices, the DW-AES cipher shows significant area reduction. In particular, highly area efficient DW-LUTs are deployed in the most resource consuming two modules, Namely, *SubBytes* and *MixColumns*, which contribute to the substantial area saving. Overall, the DW-AES cipher shows area reductions of 97% and 87% compared to the CMOS ASIC and memristive CMOL designs, respectively.
- Speed: the trade-off made in the DW-AES cipher is a larger number of cycles required compared to other hardware implementations. This is caused by the multiple-cycle operations of DW-XOR and its DW-LUT, where the shift operation needs to be performed first in order to align the target cell with MTJ to operate. Note that while small latency between the raw data in and the encrypted data out is critical in real-time systems, in big-data applications the most significant figures of merit are throughput and energy efficiency.

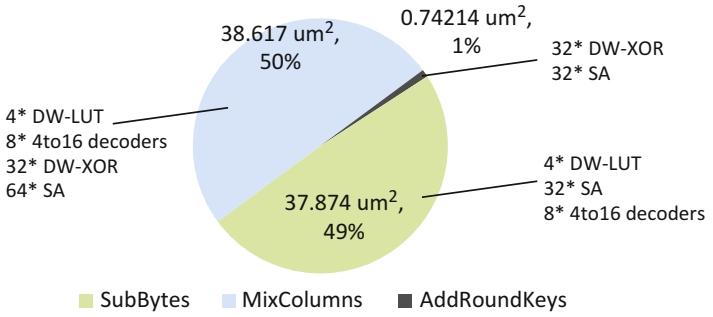


Fig. 5.23 The area breakdown of DW-AES with utilized resources in each module

Table 5.10 System configurations

AES computing platforms configurations under 10 mm^2 area design budget		
Platforms	# of AES ciphers	Clockrate
C code [27] on GPP	4 cores	2 GHz
CMOS ASIC [16]	10493	3 MHz
Pipelined ASIC [25]	499	
Memristive CMOL [1]	39761	
DW-AES	128010	
Memory configurations		
Memory capacity	1 GB	
Bus width	128 bits	
I/O energy overhead	3.7 nJ per access	

5.2.3.4 Throughput and Energy Efficiency Comparison

In the following, the proposed in-memory DW-AES is compared with other implementations at the system level. For each AES computing platform, the number of AES units is maximized subject to a fixed area constraint. All AES units are encrypting input data stream concurrently due to the high data parallelism. With the exception of the proposed in-memory DW-AES, all platforms will incur I/O energy overhead accessing data. Given a 10 mm^2 area design budget, the system configurations for different platforms are summarized in Table 5.10. The memory I/O energy overhead is obtained from CACTI.

Figure 5.24 compares throughput, power, and energy efficiency of the different AES computing platforms. All AES hardware implementations have several orders of magnitude throughput and energy efficiency improvement compared to the software implementation on GPP, as expected. Among all the hardware implementations, the proposed DW-AES computing platform provides the highest throughput of 5.6 GB/s. This throughput is 6.4× higher than that of the CMOS ASIC-based platform with a power saving of 29%; 2.5× higher than that of the pipelined CMOS ASIC platform with 30% power reduction; and 1.7× higher than that of

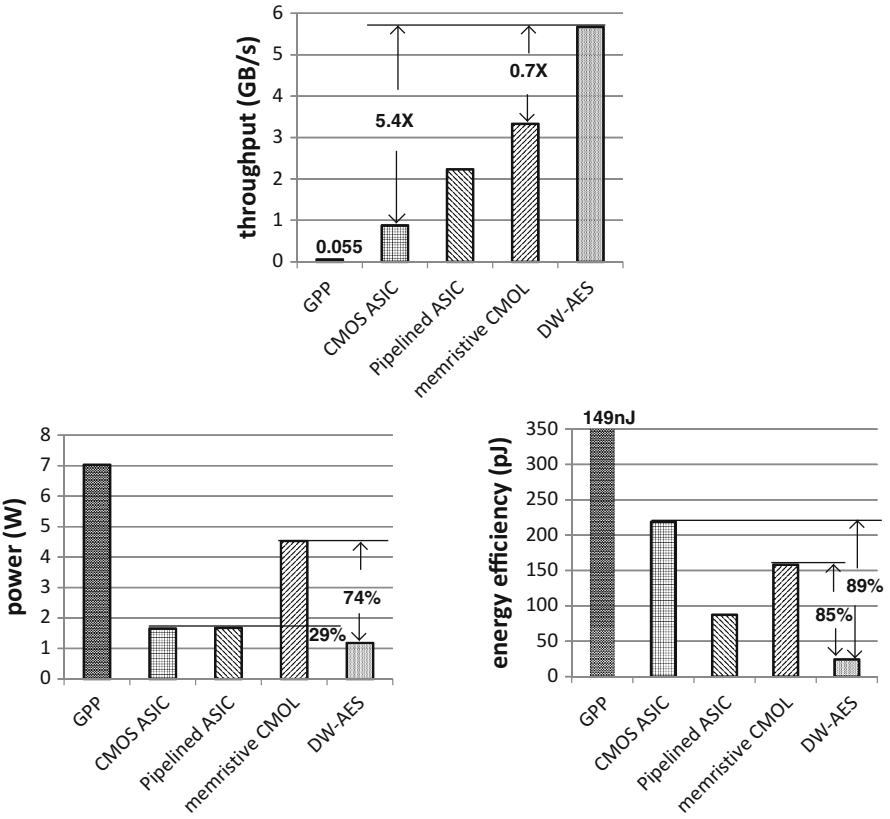


Fig. 5.24 In-memory encryption throughput, power, and energy efficiency comparisons between different AES platforms

memristive CMOL-based platform with 74% power saving. Due to the in-memory encryption computing and nonvolatility, the proposed DW-AES computing platform can achieve the best energy efficiency of 24pJ/bit, which is 9×, 3.6×, 6.5× higher than its counterpart: the CMOS ASIC, pipelined CMOS ASIC, and memristive CMOL-based platforms, respectively.

5.2.3.5 Pipelined DW-AES

Similar to the CMOS ASIC implementation, the DW-AES can also be implemented in pipelined fashion. As introduced above, AES has four stages: *SubBytes*, *AddRoundKeys*, *ShiftRows*, and *MixColumns*. In CMOS ASIC implementation the pipeline can be readily achieved as the combinational circuit for each stage takes single cycle to execute and four stages are separated by registers. In the DW-AES, however, each stage has different cycle numbers due to the multiple-cycle nature

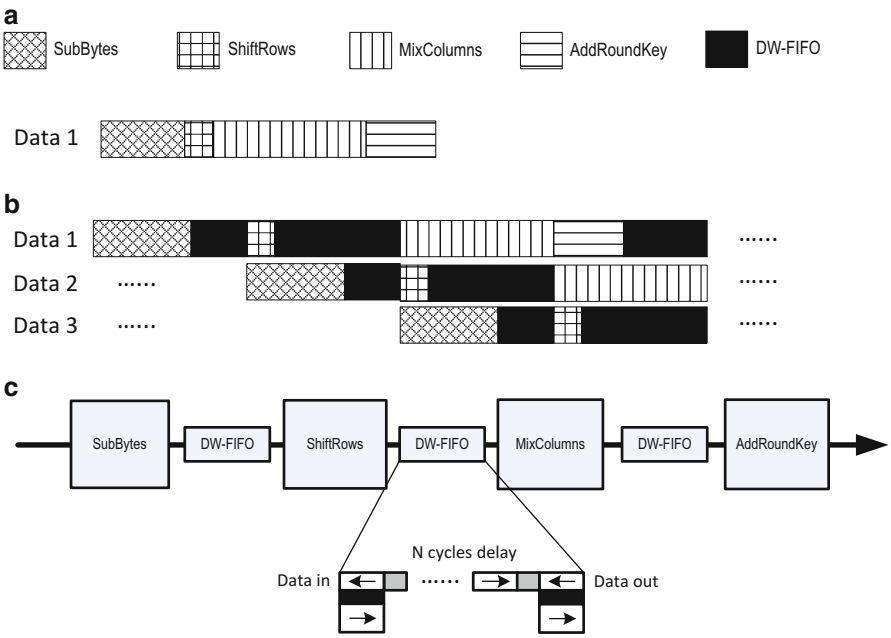


Fig. 5.25 (**a**) DW-AES without pipeline; (**b**) Pipelined DW-AES by inserting DW-FIFO; (**c**) Stages balancing by the cycles delay of DW-FIFO

of DW logic units. Figure 5.25a shows the DW-AES in which four stages have different cycle numbers. Assume the cycle numbers for the four stages are six, two, eleven, and five; the pipeline is obviously limited by the ShiftRows stage, i.e., the most time-consuming stage. It means the AddRoundKey result has to wait for nine cycles before it takes new input data from SubBytes stage and its result enters next ShiftRows stage and MixColumns stage also has to wait until ShiftRows feeds its result.

By exploiting the shift operation of domain-wall nanowire, such cycle delay can be easily accomplished instead of using timers. Figure 5.25b shows the timing diagram of pipelined DW-AES with DW-FIFO inserted. The DW-FIFO is illustrated in Fig. 5.25c which is essentially a nanowire with different lengths. By configuring the length of nanowire and shifting one domain per cycle, any number of cycle delay can be achieved.

Previously, the objective of DW-AES design optimization can be formulated as

$$\text{Minimize} \quad \text{Sum}(t_{\text{SubBytes}}, t_{\text{AddRoundKeys}}, t_{\text{ShiftRows}}, t_{\text{MixColumns}}). \tag{5.7}$$

As such the DW-AES can produce a throughput of

$$T(\text{Bytes/s}) = \frac{16(\text{Bytes})}{t_{\text{MinSum}}}. \tag{5.8}$$

However, for the pipelined scenario, the case is a little bit different. As the most time-consuming stage, defined as critical stage, is the limiting factor of throughput, the objective of pipelined DW-AES design optimization can be formulated as

$$\text{Minimize} \quad \text{Max}(t_{\text{SubBytes}}, t_{\text{AddRoundKeys}}, t_{\text{ShiftRows}}, t_{\text{MixColumns}}). \tag{5.9}$$

And in this case, the throughput can be calculated as

$$T_{\text{pipeline}}(\text{Bytes/s}) = \frac{16(\text{Bytes})}{t_{\text{MinMax}}}. \tag{5.10}$$

Ideally if the four stages have the same cycle delay, the pipelined DW-AES will produce four times higher throughput, and in the worst-case scenario where one stage is dominant the throughput for both cases will be almost the same. Therefore, the key to optimize pipelined DW-AES is to balance the cycle number of four stages. Considering that practically the resources, may it be power budget or area budget, is highly likely to be restricted, it is only possible to relax noncritical stages and improve critical stage by adjusting resources between different stages. Specifically, it needs to relocate computation resources from noncritical stages to critical stage. Given above objective and constraints, design exploration can be performed to find the pipelined DW-AES configuration that is able to produce highest throughput.

5.2.4 Machine Learning

One exciting feature of future big-data storage system is to find implicit pattern of data and excavate valued behavior behind by big-data analytics such as image feature extraction during image search. Instead of performing the image search by calculating pixel similarity, image search by machine learning is a similar process as human brains. For example, each image feature extraction is performed to obtain the characteristics first and then is matched by keywords. As such, the image search becomes a traditional string matching problem to solve.

However, to handle big image data at exascale, there is memory wall that has long memory-access latency as well as limited memory bandwidth. For the example of the image search in one big-data storage system, there may be billions of images. In order to perform feature extraction for one of the images, it will lead to significant congestion at I/Os when migrating data between memory and processor. Note that in-memory-computing system [10,17,20,28,29] is promising as one future big-data solution to relieve the memory-wall issue. For example, domain-specific accelerators can be developed within memory for big-data processing such that the data will be preprocessed before they are readout with the minimum number of data migrations.

In this book, the big image data-processing algorithm by machine learning is examined within the in-memory-computing system architecture. Among numerous

machine learning algorithms [8, 9, 34, 44], neural-network-based algorithm has shown low complexity with genetic adaptability. In particular, the extreme learning machine (ELM) [13, 14] has one input layer, one hidden layer, and one output layer, and hence it has tuning-free feature without expensive iterative training process, which makes it suitable for the low-cost hardware implementation. As such, the in-memory hardware accelerators of ELM are studied here for the big image data processing.

The proposed in-memory ELM computing system is examined by the nanoscale NVM devices. Domain-wall nanowire or racetrack memory is a newly introduced spintronic NVM device that has not only the potential for high density and high-performance memory storage, but also feasible in-memory-computing capability. In this book, we show the feasibility of mapping the ELM to a full domain-wall nanowire-based in-memory neural network computing system, called DW-NN. Compared to the scenario that ELM is executed in CMOS-based GPP, the proposed DW-NN improves the system throughput by 11.6× and energy efficiency by 92×.

5.2.4.1 Extreme Learning Machine

We first review the basic of the neural-network-based ELM algorithm. Among numerous machine learning algorithms [8, 9, 14, 34, 44], support vector machine (SVM) [8, 34] and neural network (NN) [9, 44] are widely discussed. However, both two algorithms have major challenging issues in terms of slow learning speed, trivial human intervene (parameter tuning), and poor computational scalability [14]. ELM was initially proposed [13, 14] for the single-hidden-layer feed-forward neural networks (SLFNs) (Fig. 5.26). Compared with traditional neural networks, ELM eliminates the need of parameter tuning in the training stage and hence reduces the training time significantly. The output function of ELM is formulated as (only one output node is considered)

$$f_L = \sum_{i=1}^{L} \beta_i h_i(X) = h(X)\beta, \tag{5.11}$$

where $\beta = [\beta_1, \beta_2, \cdots, \beta_L]^T$ is the output weight vector storing the output weights between the hidden layer and output node. $h(X) = [h_1(X), h_2(X), \cdots, h_L(X)]^T$ is the hidden layer output matrix given input vector X and performs the transformation of input vector into L-dimensional feature space. The training process of ELM aims to obtain output weight vector β and minimize the training error as well as the norm of output weight

$$\text{Minimize} : \|H\beta - T\| \text{ and } \|\beta\| \tag{5.12}$$

$$\beta = H^{\dagger}T, \tag{5.13}$$

where $H^{\dagger}$ is the Moore—Penrose generalized inverse of matrix H.

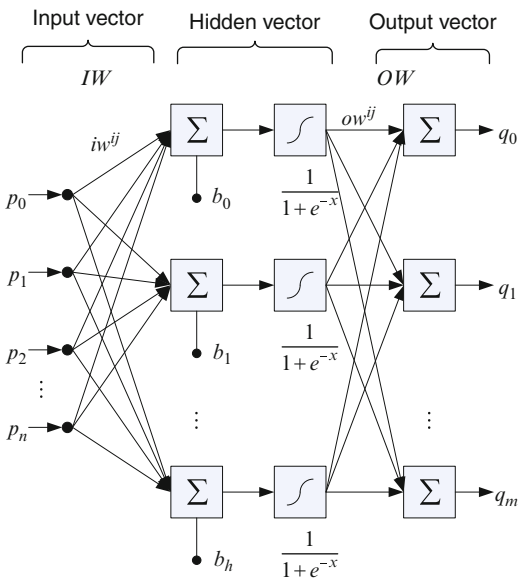


Fig. 5.26 The working flow of extreme learning machine

The application of ELM for image processing in this book is an ELM-based image super-resolution (SR) algorithm [2], which learns the image features of a specific category of images and improves low-resolution figures by applying learned knowledge. Note that ELM-SR is commonly used as preprocessing stage to improve image quality before applying other image algorithms. It involves intensive matrix operation, such as matrix addition, matrix multiplication as well as exponentiation on each element of a matrix. Figure 5.31 illustrates the computation flow for ELM-SR, where input vector obtained from input image is multiplied by input weight matrix. The result is then added with bias vector b to generate input of sigmoid function. Lastly sigmoid function outputs are multiplied with output weight matrix to produce final results. In the following, we will demonstrate how to map the fundamental addition, multiplication, and sigmoid function to domain-wall nanowires.

5.2.4.2 In-Memory MapReduce ELM Architecture

Conventionally, all the data is maintained within memory that is separated from the processor but connected with I/Os. Therefore, during the execution, all data needs to be migrated to processor and written back afterwards. In the data-oriented applications, however, this will incur significant I/O congestion and hence greatly degrade the overall performance. In addition, significant standby power will be consumed in order to hold the large volume of data.

To overcome the above two issues, the in-memory nonvolatile computing architecture is introduced. The overall architecture of domain-wall memory-based

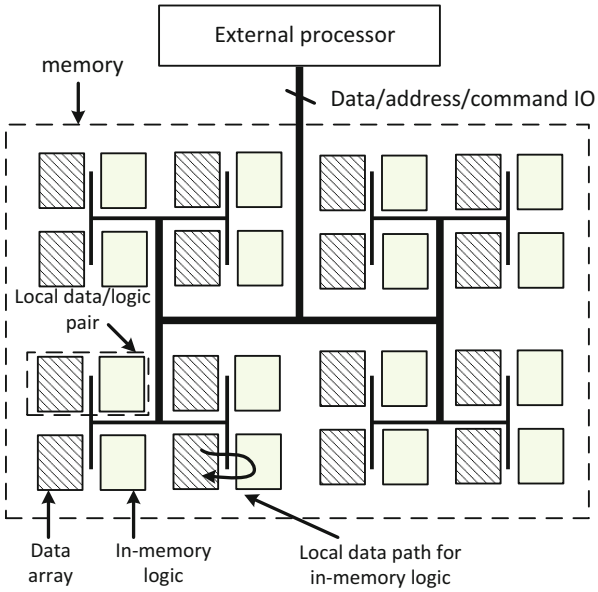


Fig. 5.27 The overview of the in-memory-computing architecture

in-memory-computing platform is illustrated in Fig. 5.27. In particular, domain-specific in-memory accelerators are integrated locally together with the stored data in distributed manner such that the frequently involved operations can be performed without much communication with external processor. In addition, the distributed local accelerators can also provide great thread-level parallelism and thus the throughput can be improved.

Figure 5.28 shows how the in-memory distributed Map-Reduce [6] data processing is performed locally between one data array and local logic pair. Firstly, the external processor will issue commands to specific pair to perform in-memory logic computing. The commands will be received and interpreted by a controller in accelerator. Secondly, the controller will request related data to the data array with a read operation. As a result, the neural-network-based processing in ELM, mainly including weighted sum and sigmoid function, can be performed in a Map-Reduce fashion. Lastly, the results are written back to the data array.

5.2.4.3 ELM Task Mapping

5.2.4.4 Vector Inner Product by Domain-Wall Adder and Multiplier

The GMR-effect can be interpreted as the bit-wise XOR operation of the magnetization direction of two thin magnetic layers, where the output is denoted by high or low resistance. In a GMR-based MTJ structure, however, the XOR logic will fail

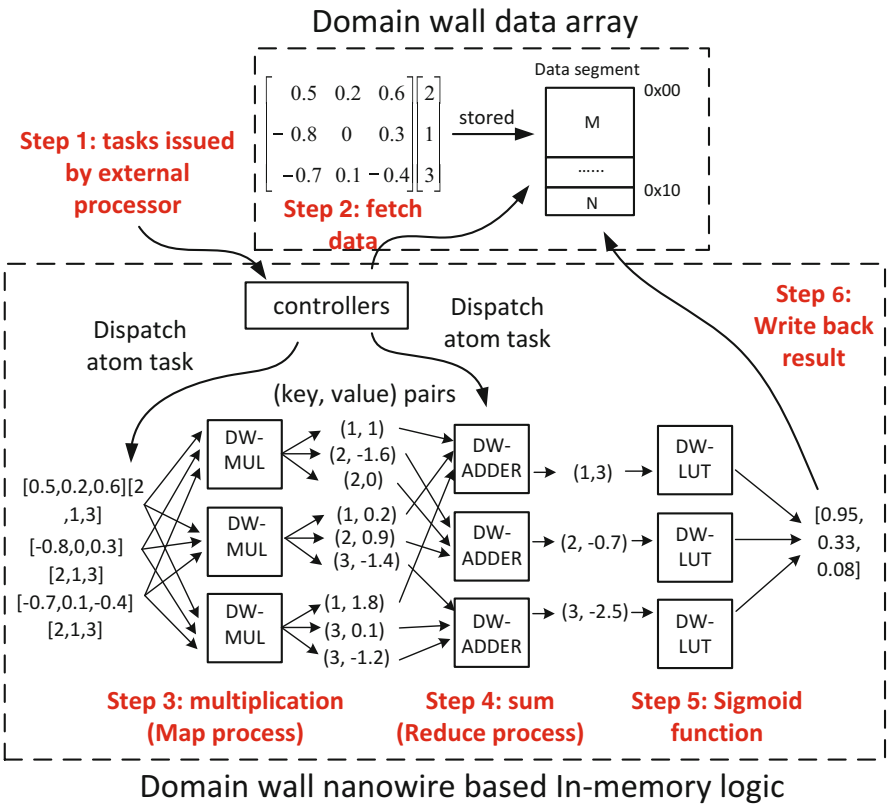


Fig. 5.28 Detailed in-memory domain-wall nanowire-based machine learning platform in Map-Reduce fashion

as there is only one operand as the variable since the magnetization in fixed layer is constant. Nevertheless, this problem can be overcome by the unique domain-wall shift operation in the domain-wall nanowire device, which enables the possibility of DWL-based XOR logic for computing.

A bit-wise-XOR logic implemented by two domain-wall nanowires is shown in Fig. 5.29. The bit-wise-XOR logic is performed by constructing a new read-only port, where two free layers and one insulator layer are stacked. The two free layers are in the size of one magnetization domain and are from two respective nanowires. Thus, the two operands, denoted as the magnetization direction in free layer, can both be variables with values assigned through the MTJ of the agreeing nanowire. As such, it can be shifted to the operating port such that the XOR logic is performed.

In addition, to realize a full adder, the carry operation is also needed. Spintronics-based carry operation is proposed in [37], where a precharge sensing amplifier (PCSA) is used for resistance comparison. The carry logic by PCSA and two branches of domain-wall nanowires is shown in Fig. 5.29. The three operands for

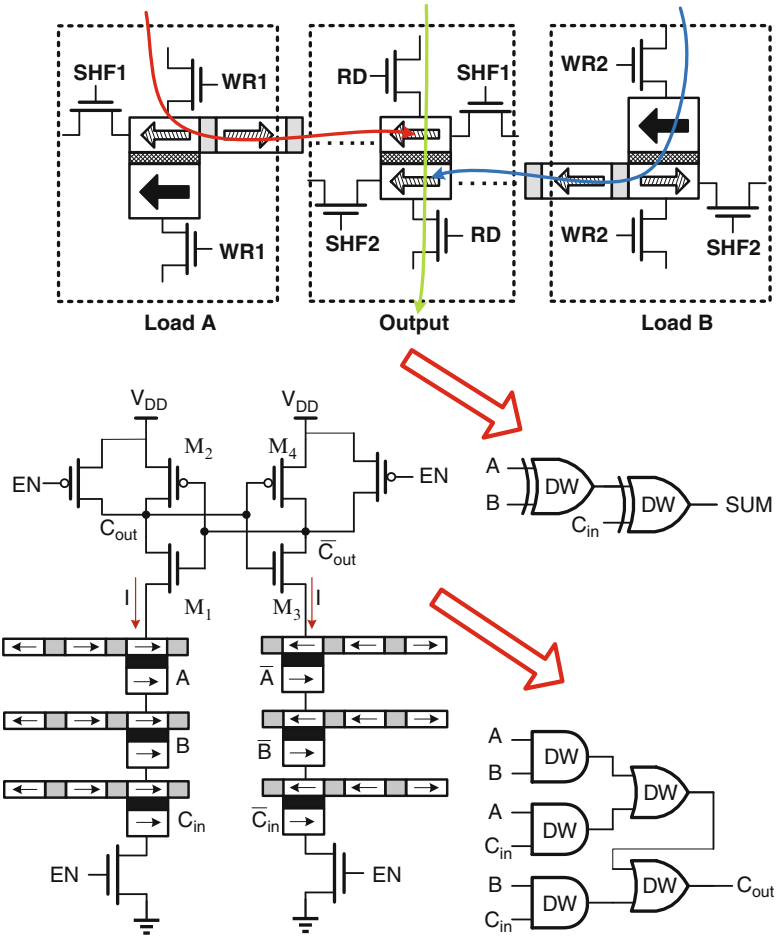


Fig. 5.29 Domain-wall nanowire-based full adder with SUM operation by DW-XOR logic and carry operation by resistor comparator

carry operation are denoted by resistance of MTJ (low for 0 and high for 1) and belong to respective domain-wall nanowires in the left branch. The right branch is made complementary to the left one. Note that the C_{out} and $\overline{C_{out}}$ will be precharged high at first when PCSA EN signal is low. When the circuit is enabled, the branch with lower resistance will discharge its output to "0". For example, when left branch has no or only one MTJ in high resistance, i.e., no carry out, the right branch will have three or two MTJs in high resistance, such that the C_{out} will be 0. The complete truth table will confirm carry logic by this circuit. The domain-wall nanowire works as the writing circuit for the operands by writing values at one end and shift it to PCSA.

Note that with the full adder implemented by domain-wall nanowires and intrinsic shift ability of domain-wall nanowire, a shift/add multiplier can be readily achieved purely by domain-wall nanowires.

5.2.4.5 Sigmoid Function by Domain-Wall Lookup Table

Sigmoid function includes exponentiation, division, and addition, which is a computing intensive operation in ELM application. In particular, the exponentiation will take many cycles to execute in the conventional processor due to the lack of corresponding accelerator. Therefore, it is extremely economic to perform exponentiation by the LUT. The LUT, essentially a pre-configured memory array, takes a binary address as input, finds target cells that contain result through decoders, and finally outputs correspondingly by sense amplifiers.

A domain-wall nanowire-based LUT (DW-LUT) is illustrated in Fig. 5.30. Compared with the conventional SRAM or DRAM by CMOS, the DW-LUT can demonstrate two major advantages. Firstly, extremely high integration density can be achieved since multiple bits can be packed in one nanowire. Secondly, zero standby power can be expected as a nonvolatile device does not require to be powered to retain the stored data. By distributing the multiple bits of results in separated nanowires, the serial operation of nanowire can be avoided and the function can be done fast.

Note that the LUT size is determined by the input domain, the output range, and the required precision for the floating point numbers. Figure 5.30 shows the ideal logistic curve and approximated curves by the LUTs. It can be observed that the output range is bounded between 0 and 1, and although the input domain is infinite, it is only informative in the center around 0. The LUT visually is the digitalized logistic curve, and the granularity, i.e., precision, depends on the LUT size. For machine learning application, the precision is not as sensitive as scientific computations. As a result, the LUT size for sigmoid function can be greatly optimized and leads to high energy efficiency for sigmoid function execution.

To compare proposed in-memory DW-NN platform and conventional GPP-based platform, ELM-based super-resolution (ELM-SR) application is executed as the workload. The evaluation of ELM-SR in GPP platform is based on gem5 [4] and McPAT [23] for core power and area model. DW-NN is evaluated in our developed self-consistent simulation platform based on NVMSPICE, DW-CACTI, and DW-NN behavioral simulator. The processor runs at 3 GHz while the accelerators run at 500 MHz. System memory capacity is set as 1 GB, and bus width is set as 128 bits. Based on [18], 3.7 and 6.3 nJ per access are used for on-chip and off-chip I/O overhead, respectively.

Table 5.11 compares ELM-SR in both DW-NN and GPP platforms. Due to the deployment of in-memory accelerators and high data parallelism, the throughput of DW-NN improves by 11.6× compared to GPP platform. In terms of area used by computational resources, DW-NN is 2.7% higher than that of GPP platform. Additional 0.5 mm^2 is used to deploy the domain-wall nanowire-based accelerators.

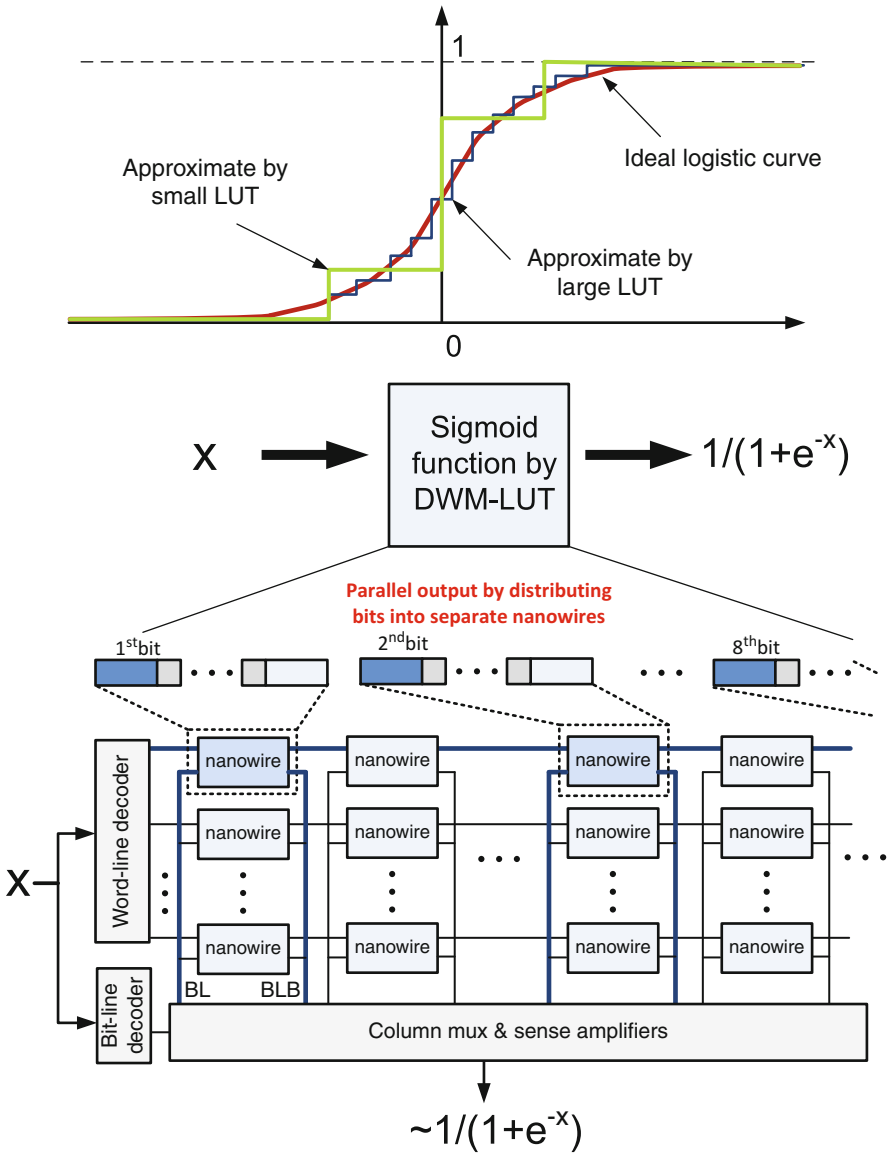


Fig. 5.30 (**a**) Sigmoid function implemented by domain-wall nanowire-based lookup table (DW-LUT); (**b**) DW-LUT size effect on the precision of the sigmoid function. The larger the LUT, the smoother and more precise the curve is

Thanks to the high integration density of domain-wall nanowires, the numerous accelerators are brought with only slight area overhead. In DW-NN, the additional power consumed by accelerators is compensated by the saved dynamic power of

Table 5.11 System area, power, throughput, and energy efficiency comparison between in-memory architecture and conventional architecture

Platform	DW-NN	GPP with on-chip memory	GPP with off-chip memory
# of computational units utilized	1×processor 7714×DW-ADDER 7714×DW-MUL 551×DW-LUT 1×controller	1×processor	1×processor
Area of computational units (mm^2)	18 (processor) + 0.5 (accelerators)	18	18
Power (Watt)	10.1	12.5	12.5
Throughput (MBytes/s)	108	9.3	9.3
Energy efficiency (nJ/bit)	7	389	642

Fig. 5.31 (**a**) Original image before ELM-SR algorithm (SSIM value is 0.91); (**b**) Image quality improved after ELM-SR algorithm by DW-NN hardware implementation (SSIM value is 0.94); (**c**) Image quality improved by GPP platform (SSIM value is 0.97)

processor, since the computation is mostly performed by the in-memory logic. Overall, DW-NN achieves a power reduction of 19%. The most noticeable advantage of DW-NN is its much higher energy efficiency compared to GPP. Specifically, it is 56× and 92× better than that of GPP with on-chip and off-chip memory respectively. The advantage comes from three aspects: (a) in-memory-computing architecture that saves I/O overhead; (b) nonvolatile domain-wall nanowire devices that are leakage free; and (c) application specific accelerators.

Figure 5.31 shows the image quality comparison between the proposed in-memory DW-NN hardware implementation and the conventional GPP software implementation. To measure the performance quantitatively, structural similarity (SSIM) [41] is used to measure image quality after ELM-SR algorithm. It can be observed that the images after ELM-SR algorithm in both platforms have higher image quality than the original low-resolution image. However, due to the use of

LUT, which trades off precision against the hardware complexity, the image quality in DW-NN is slightly lower than that in GPP. Specifically, the SSIM is 0.94 for DW-NN, 3% lower than 0.97 for GPP.

References

1. Abid Z, Alma'Aitah A, Barua M, Wang W (2009) Efficient cmol gate designs for cryptography applications. IEEE Trans Nanotechnol 8(3):315–321
2. An L, Bhanu B (2012) Image super-resolution by extreme learning machine. In: 19th IEEE international conference on Image processing (ICIP) 2012, IEEE, Washington, pp 2209–2212
3. Augustine C, Raychowdhury A, Behin-Aein B, Srinivasan S, Tschanz J, De VK, Roy K (2011) Numerical analysis of domain wall propagation for dense memory arrays. In: 2011 IEEE international Electron devices meeting (IEDM). IEEE, Washington, pp 17–6
4. Binkert N, Beckmann B, Black G, Reinhardt SK, Saidi A, Basu A, Hestness J, Hower DR, Krishna T, Sardashti S et al (2011) The gem5 simulator. ACM SIGARCH Comput Architect News 39(2):1–7
5. Bird S, Phansalkar A, John LK, Mericas A, Indukuru R (2007) Performance characterization of spec cpu benchmarks on intel's core microarchitecture based processor. In: SPEC Benchmark Workshop
6. Dean J, Ghemawat S (2008) Mapreduce: simplified data processing on large clusters. Commun ACM 51(1):107–113
7. Dong X, Jouppi NP, Xie Y (2009) Pcramsim: system-level performance, energy, and area modeling for phase-change ram. In: Proceedings of the 2009 international conference on computer-aided design. ACM, New York, pp 269–275
8. Furey TS, Cristianini N, Duffy N, Bednarski DW, Schummer M, Haussler D (2000) Support vector machine classification and validation of cancer tissue samples using microarray expression data. Bioinformatics 16(10):906–914
9. Hagan MT, Demuth HB, Beale MH et al (1996) Neural network design. Pws Pub., Boston
10. Hanyu T, Teranishi K, Kameyama M (1998) Multiple-valued logic-in-memory vlsi based on a floating-gate-mos pass-transistor network. In: 1998 IEEE International Solid-state circuits conference, 1998. Digest of Technical Papers. IEEE, NJ, pp 194–195
11. Hennessy JL, Patterson DA (2012) Computer architecture: a quantitative approach. Elsevier, Waltham
12. Hua CH, Cheng TS, Hwang W (2005) Distributed data-retention power gating techniques for column and row co-controlled embedded sram. In: 2005 IEEE international workshop on Memory technology, design, and testing, 2005 (MTDT 2005). IEEE, Washington, pp 129–134
13. Huang GB, Zhu QY, Siew CK (2004) Extreme learning machine: a new learning scheme of feedforward neural networks. In: 2004 IEEE international joint conference on Neural networks 2004 Proceedings, vol 2. IEEE, Washington, pp 985–990
14. Huang GB, Zhu QY, Siew CK (2006) Extreme learning machine: theory and applications. Neurocomputing 70(1):489–501
15. ITRS (2010) International technology roadmap of semiconductor. http://www.itrs.net
16. Kaps JP, Sunar B (2006) Energy comparison of aes and sha-1 for ubiquitous computing. In: Emerging directions in embedded and ubiquitous computing. Springer, New York, pp 372–381
17. Kautz WH (1969) Cellular logic-in-memory arrays. Comput IEEE Trans 100(8):719–727
18. Kim JK, Choi JH, Shin SW, Kim CK, Kim HY, Kim WS, Kim C, Cho SI (2004) A 3.6 gb/s/pin simultaneous bidirectional (sbd) i/o interface for high-speed dram. In: IEEE international Solid-state circuits conference, 2004. Digest of technical papers, ISSCC 2004. IEEE, Washington, pp 414–415

19. Kim KH, Gaba S, Wheeler D, Cruz-Albrecht JM, Hussain T, Srinivasa N, Lu W (2011) A functional hybrid memristor crossbar-array/cmos system for data storage and neuromorphic applications. Nano Lett 12(1):389–395
20. Kimura H, Hanyu T, Kameyama M, Fujimori Y, Nakamura T, Takasu H (2004) Complementary ferroelectric-capacitor logic for low-power logic-in-memory vlsi. Solid-State Circuits IEEE J 39(6):919–926
21. Koga M, Iida M, Amagasaki M, Ichida Y, Saji M, Iida J, Sueyoshi T (2010) First prototype of a genuine power-gatable reconfigurable logic chip with feram cells. In: 2010 International conference on field programmable logic and applications (FPL). IEEE, Washington, pp 298–303
22. Lee SJ, Ouyang CS (2003) A neuro-fuzzy system modeling with self-constructing rule generationand hybrid svd-based learning. Fuzzy Syst IEEE Trans 11(3):341–353
23. Li S, Ahn JH, Strong RD, Brockman JB, Tullsen DM, Jouppi NP (2009) Mcpat: an integrated power, area, and timing modeling framework for multicore and manycore architectures. In: 42nd annual IEEE/ACM international symposium on Microarchitecture, 2009 MICRO-42. IEEE, New York, pp 469–480
24. Lin J, Dyer C (2010) Data-intensive text processing with mapreduce. Synth Lect Hum Lang Tech 3(1):1–177
25. Lin SY, Huang CT (2007) A high-throughput low-power aes cipher for network applications. In: Proceedings of the 2007 Asia and South Pacific Design Automation Conference, IEEE Computer Society, pp 595–600
26. Loh GH (2008) 3d-stacked memory architectures for multi-core processors. In: ACM SIGARCH computer architecture news, vol 36. IEEE Computer Society, Los Alamitos, pp 453–464
27. Malbrain K (2009) Byte-oriented-aes: a public domain byte-oriented implementation of aes in c. https://code.google.com/p/byte-oriented-aes/
28. Matsunaga S, Hayakawa J, Ikeda S, Miura K, Hasegawa H, Endoh T, Ohno H, Hanyu T (2008) Fabrication of a nonvolatile full adder based on logic-in-memory architecture using magnetic tunnel junctions. Appl Phys Expr 1(9):1301
29. Matsunaga S, Hayakawa J, Ikeda S, Miura K, Endoh T, Ohno H, Hanyu T (2009) Mtj-based nonvolatile logic-in-memory circuit, future prospects and issues. In: Proceedings of the conference on design, automation and test in Europe. European Design and Automation Association, Leuven, pp 433–435
30. Nagai T, Wada M, Iwai H, Kaku M, Suzuki A, Takai T, Itoga N, Miyazaki T, Takenaka H, Hojo T et al (2006) A 65nm low-power embedded dram with extended data-retention sleep mode. In: IEEE international solid-state circuits conference 2006, ISSCC 2006. Digest of technical papers. IEEE, Washington, pp 567–576
31. Parkin SS, Hayashi M, Thomas L (2008) Magnetic domain-wall racetrack memory. Science 320(5873):190–194
32. Qin H, Cao Y, Markovic D, Vladimirescu A, Rabaey J (2004) Sram leakage suppression by minimizing standby supply voltage. In: Proceedings of 5th international symposium on Quality electronic design, 2004. IEEE, pp 55–60
33. Shang Y, Zhang C, Yu H, Tan CS, Zhao X, Lim SK (2013) Thermal-reliable 3d clock-tree synthesis considering nonlinear electrical-thermal-coupled tsv model. In: ASP-DAC, pp 693–698
34. Suykens JA, Vandewalle J (1999) Least squares support vector machine classifiers. Neural Process Lett 9(3):293–300
35. Talbot J, Yoo RM, Kozyrakis C (2011) Phoenix++: modular mapreduce for shared-memory systems. In: Proceedings of the 2ndnd international workshop on MapReduce and its applications. ACM, New York, pp 9–16
36. Thoziyoor S, Muralimanohar N, Ahn JH, Jouppi NP (2008) Cacti 5.1. HP Laboratories, 2 Apr 2008
37. Trinh HP, Zhao W, Klein JO, Zhang Y, Ravelsona D, Chappert C (2012) Domain wall motion based magnetic adder. Electron Lett 48(17):1049–1051

38. Usselmann R (2002) Advanced encryption standard / rijndael ip core. http://opencores.org/project,aes_core
39. Venkatesan R, Kozhikkottu V, Augustine C, Raychowdhury A, Roy K, Raghunathan A (2012) Tapecache: a high density, energy efficient cache based on domain wall memory. In: Proceedings of the 2012 ACM/IEEE international symposium on low power electronics and design. ACM, New York, pp 185–190
40. Wang Y, Yu H (2013) An ultralow-power memory-based big-data computing platform by nonvolatile domain-wall nanowire devices. In: 2013 IEEE international symposium on low power electronics and design (ISLPED). IEEE, New York, pp 329–334
41. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image quality assessment: From error visibility to structural similarity. Image Process IEEE Trans 13(4):600–612
42. Wang X, Chen Y, Li H, Dimitrov D, Liu H (2008) Spin torque random access memory down to 22 nm technology. Magnetics IEEE Trans 44(11):2479–2482
43. Xie J, Dong X, Xie Y (2010) 3d memory stacking for fast checkpointing/restore applications. In: 2010 IEEE international conference on 3D systems integration conference (3DIC). IEEE, Munich, pp 1–6
44. Yegnanarayana B (2004) Artificial neural networks. PHI Learning Pvt. Ltd., New Delhi

Appendix A
NVM-SPICE Design Examples

All's well that ends well

A.1 Memristor Model Card in NVM-SPICE

NVM-SPICE is developed with NVM nonlinear dynamic models added by extending NGspice.The syntax generally follows NGspice style. One slight difference is one more identifier for NVM device type is required. For example, the general form of model card of a memristor element is

$$n < \text{name} > \ \text{memristor} \ < + \text{node} > < - \text{node} > < \text{model} > < \text{params} > .$$

The first letter of the element name specifies the element type. Here the memristor has been assigned a letter starting with n. The <name> column is the name of device, which can be arbitrary alphanumeric strings. The following "memristor" is used to specify the type of NVM device to be the memristor. The following two columns are used to indicate positive and negative nodes for its topological connection in circuit. The <model> column is to apply a predefined model (a set of specified model parameters) to the device. The parameters for memristor device instances can be specified in the <params> column.

Table A.1 shows the full list of parameters for the implemented nonlinear dynamic memristor model. For not specified parameters, the default values will be assigned. Note that when all the four parameters *rhoon, rhooff, w*, and *l* are specified, the *ron* and *roff* will be calculated by

$$R_{\text{on}} = \frac{\rho_{\text{on}} \cdot d}{w \cdot l}, \quad R_{\text{off}} = \frac{\rho_{\text{off}} \cdot d}{w \cdot l}. \tag{A.1}$$

H. Yu and Y. Wang, *Design Exploration of Emerging Nano-scale Non-volatile Memory*,
DOI 10.1007/978-1-4939-0551-5, © Springer Science+Business Media New York 2014

Table A.1 A full list of parameters for nonlinear dynamic memristor model

Name	Model parameter	Units	Default	Example
ron	Resistance of memristor for conducting state	Ω	100	50
roff	Resistance of memristor for nonconducting state	Ω	16 k	100 k
height	Thickness of memristor film	m	50 n	50 n
mu	Mobility at small electric field	$m^2/(V{\cdot}s)$	0.01 f	0.01 f
e0	Characteristic field for a particular mobile atom in the crystal	V/m	100 meg	100 meg
wf	The type of window function	–	1	2
p	The slowdown effect parameter in window function	–	2	5
rhoon	The electrical resistivity of conducting part of memristor	Ω·m	Not specified	10 u
rhooff	The electrical resistivity of nonconducting part of memristor	Ω· m	Not specified	20 m
length	The length of cross-section area of memristor	m	Not specified	50 n
width	The width of cross-section area of memristor	m	Not specified	50 n
Name	Instance parameter	Units	Default	Example
rinit	Initial resistance of memristor	Ω	100	50 k

and the specifications of ron and $roff$ will be ignored. The Joglekar window function is applied by default with $wf = 1$. Alternatively, the Biolek window function can be applied by setting $wf = 2$ or no window function $wf = 0$.

Some examples for memristor element description are shown below:

```
.model nvm_mem_model1 memristor ron=0.1k roff=14k wf=1 p=5
n1 memristor 2 0
nref memristor 7 3 rinit=0.1k
nr5c5 memristor 16 2 nvm_mem_model1
```

A.2 Transient CMOS/Memristor Co-simulation Examples by NVM-SPICE

Here, we will illustrate how to use NVM-SPICE for hybrid CMOS/NVM design co-simulation with simple circuits for memristor, shown in Fig. A.1. This toy example circuit intends to study the SET operation of a memristor device.

The corresponding netlist to describe the circuit in Fig. A.1 can be written below:

```
* memristor SET operation study
.model nvmmod memristor ron=1k roff=16k
.model nmos nmos level=54 version=4.7.0
vdd nvdd 0 3.3v
n1 memristor nvdd d nvmmod rinit=15.9k
vcontrol g 0 pwl(0 0 10us 0 11us 3.3 90us 3.3 91us 0 100us 0)
m1 d g s 0 nmos l=90n w=2u
.tran 10n 100us
.end
```

After running transient analysis of above netlist, the following **PLOT** command can be used to investigate the change of internal state doping ratio for memristor:

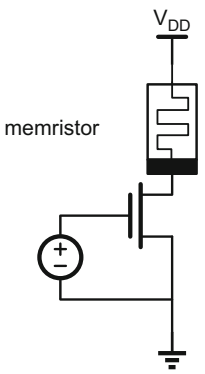


Fig. A.1 1T1R structure for memristor device based memory cell

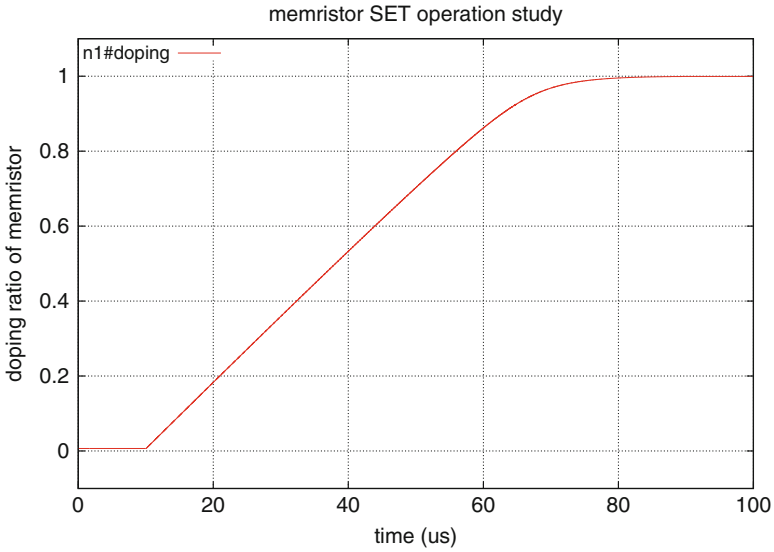


Fig. A.2 Dynamics of doping ratio in memristor set operation under transient analysis

plot n1#doping

Then we obtain the wave shown in Fig. A.2. It can be seen that the doping ratio changes from 0 (at 11 µs) to 1 (at around 82 µs) which indicates its resistance is switched from 15.9 to 1 kΩ within 71 µs under 3.3 V programming voltage. To further verify this, we can plot the actual resistance of memristor by

plot (v(nvdd)-v(d))/i(vdd)

We can get Fig. A.3, from which it is clear that the resistance does change from 15.9 to 1 kΩ. The internal state variables of NVM devices are usually associated with the external resistance; thus knowing the internal states, the way to obtain external resistance is to calculate it referring to its model equations.

A.3 STT-MTJ Model Card in NVM-SPICE

Similar to memristor, the general form of model card of a STT-MTJ element is

$$n<\text{name}> \ sttmtj \ <+\text{node}> <-\text{node}> <\text{model}> <params> .$$

Table A.2 shows the full list of parameters for the implemented STT-MTJ model. Some examples for STT-MTJ element description are shown below:

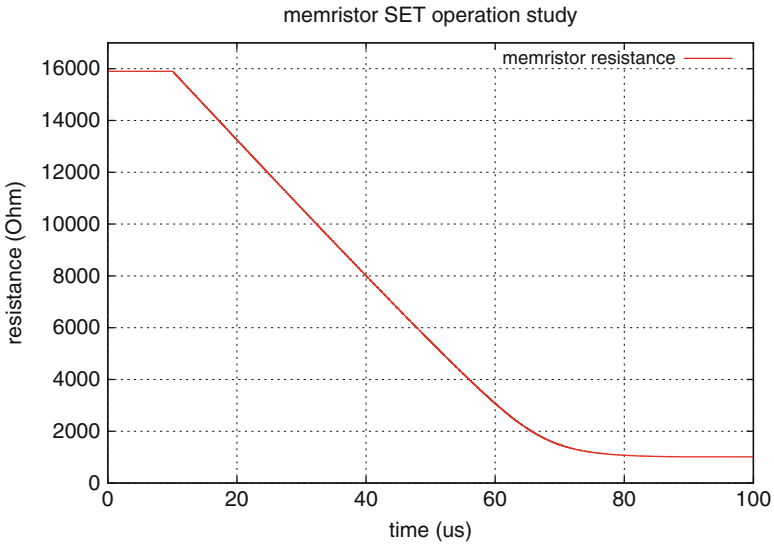


Fig. A.3 Plot of time-varying resistance of memristor for verification

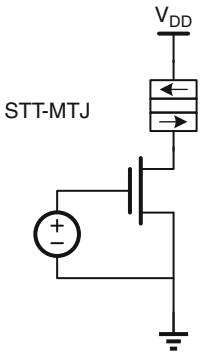


Fig. A.4 1T1R structure for STT-MTJ device based memory cell

.model nvm_sttmtj_model1 sttmtj rl=2k rh=4k ms=30k ms=700k ka=0.2 kb=0.5
n1 sttmtj 2 0
nref sttmtj 7 3 r0=550k
nr5c5 sttmtj 16 2 nvm_sttmtj_model1

A.4 Hybrid CMOS/STT-MTJ Co-simulation Example

We could write the netlist for STT-MTJ SET operation circuit depicted in Fig. A.4 as follows:

Table A.2 A full list of parameters for STT-MTJ model

Name	Model parameter	Units	Default	Example
vcp	Voltage-dependent coefficient for parallel state	–	0.01	0.1
vcap	Voltage-dependent coefficient for antiparallel state	–	0.9	0.65
p	Pre-factor of the spin transfer term and driving current ratio	–	6.37	6.37
gamma	Electron gyromagnetic ratio in Landau–Lifshitz–Gilbert equation	$(sA/m)^{-1}$	221 k	221 k
ms	Saturation magnetization of material	kA/m	800 k	800 k
hk	Effective anisotropy field	kA/m	29.05 k	29.05 k
rp	Resistance value of parallel state	Ω	1230	1 k
rap	Resistance value of antiparallel state	Ω	2650	5 k
damping	Damping constant in Landau–Lifshitz–Gilbert equation	–	0.01	0.005
Name	Instance parameter	Units	Default	Example
phi0	Initial radian for internal state variable ϕ	rad	1	1
theta0	Initial radian for internal state variable θ	rad	0.001	0.005

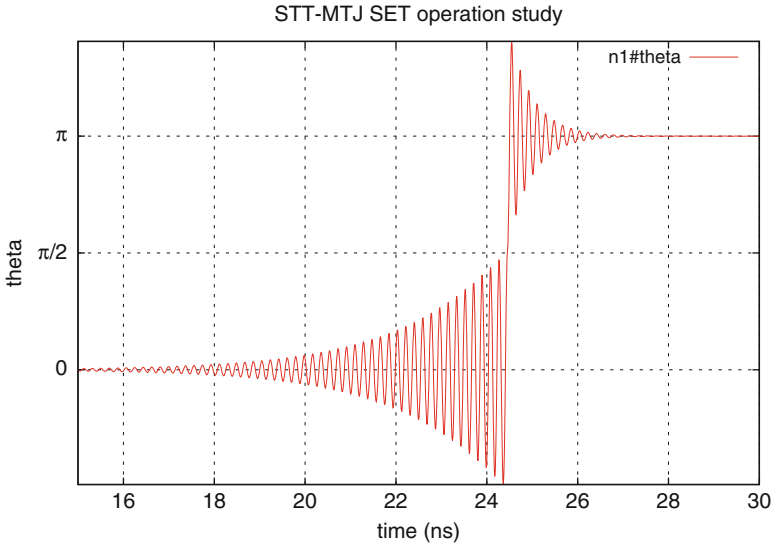


Fig. A.5 Plot of time-varying internal state theta of STT-MTJ

```
* STT-MTJ SET operation study
.model nvmmod2 sttmtj vcp=0 vcap=0 rap=1000 rp=500
.model nmos nmos level=54 version=4.7.0
v1 nvdd 0 pwl(0 0 5ns 0 6ns 1.2v)
vcontrol g 0 pwl(0 0 4ns 0 5ns 1.2v)
m1 d g 0 0 nmos l=90n w=2u
n1 sttmtj nvdd d nvmmod2 theta0=0.01
.tran 0.01n 30ns
.end
```

Similarly, we run the commands to plot internal state and external resistance, and results are shown in Figs. A.5 and A.6.

```
plot n1#theta
plot (v(nvdd)-v(d))/i(v1)
```

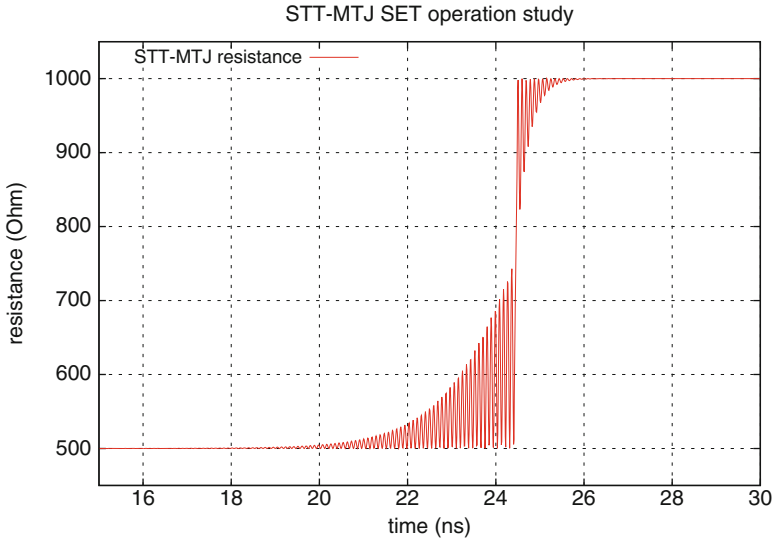


Fig. A.6 Plot of time-varying resistance of STT-MTJ

Index

H. Yu and Y. Wang, *Design Exploration of Emerging Nano-scale Non-volatile Memory*,
DOI 10.1007/978-1-4939-0551-5, © Springer Science+Business Media New York 2014