Title
Memristor-based ternary content addressable memory for data-intensive applications

Author
Zheng, Le

Publication Date
2015

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA
SANTA CRUZ

MEMRISTOR-BASED TERNARY CONTENT ADDRESSABLE MEMORY FOR DATA-INTENSIVE APPLICATIONS

A dissertation submitted in partial satisfaction
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

ELECTRICAL ENGINEERING

by

Le Zheng

March 2015

The Dissertation of Le Zheng is approved:

Professor Sung-Mo Steve Kang, Chair

Professor Kenneth Pedrotti

Maya Gokhale, Ph.D.

Tyrus Miller
Vice Provost and Dean of Graduate Studies

Table of Contents

List of Figures

List of Tables

Abstract

Memristor-based Ternary Content Addressable Memory for Data-Intensive Applications

by

Le Zheng

Data-intensive storage and computing systems call for continuing advancements both in data latency and energy efficiency. However, the major benefits from CMOS technologies, such as high packing density and processing speed, are getting desensitized primarily due to the prohibitively increasing power density. In order for the next-generation storage and computing systems to be capable of high performance data-intensive applications, it is necessary to continue innovations in creating new circuits and system architectures, together with searching for new materials and devices.

The memristor, short for memory resistor, is a two-terminal passive device whose resistance is controlled by external electrical signals and exhibits non-volatile memory function. Due to its capabilities of non-volatile resistive memories, nanoscale miniaturization in an ultra-high packing density, and intriguing nonlinear dynamics, the memristor is being widely investigated to create new advanced circuit functions and to complement CMOS systems. The memristor technologies will lead current CMOS-based storage and computing systems to the data-intensive electronic systems, with significantly reduced stand-by power, form factor and manufacturing cost.

Developing compact models for the memristor is essential to facilitate circuit analyses and designs with memristors. While the previously reported memristor models exhibit limitations in the model stability, versatility and adaptability, we propose a new module-based memristor model that covers a wide range of device behaviors. Coincident to the theoretic memristor behaviors, the proposed model uniquely reveals that an effective charge-flux constitutive relationship can always be obtained from various types of memristors. The stability of the proposed model is also significantly enhanced by adapting the new charge (or flux)-based window function.

Associative lookup functions with high throughputs are widely implemented in Ternary Content Addressable Memories (TCAMs). The TCAM holds the potential to curb the latency and power requirements of data-intensive systems. However existing TCAMs that commonly utilize Static Random Access Memories (SRAMs) as the storage units exhibit low storage capacity/density and high cost-per-bit due to bit cells with large areas. We propose a memristor-based ternary content addressable memory (mTCAM) for data-intensive applications. A novel bit cell structure is presented that not only minimizes the bit cell area but also is capable of performance optimizations on the latency and energy consumption. Detailed design issues such as voltage compliance to ensure correct write/search operations, parameter-dependent sensing margins and device variations are also discussed. Circuit level simulations have demonstrated functionalities of the mTCAM. Performance evaluation has shown that

mTCAM achieves impressive storage density, search latency and energy consumption. The proposed mTCAM is an attractive candidate in building future computing systems for data-intensive applications.

To my parents,

Weiying Yang and Guoxin Zheng

for their endless love and support.

Acknowledgements

This dissertation was impossible to accomplish without the help and support of many individuals.

I want to express my sincere gratitude to my advisor Prof. Sung-Mo Steve Kang for his insightful guidance and generous support. Prof. Kang provided me with the opportunity to continue my Ph. D. study and leaded me into this exciting field. His scientific enthusiasm as well as the depth and breadth of his knowledge have been an absolute inspiration.

I cannot thank enough Dr. Sangho Shin. He has been an incredible mentor for the past few years. He has a deep understanding of a wide range of technical topics and has been extremely insightful during discussions. Without his guidance and encouragement, this dissertation simply was not possible. He has always been an admirable model to me.

I thank my thesis committee members, Prof. Ken Pedrotti and Dr. Maya Gokhale for their support.

I thank Prof. Nobuhiko Kobayashi for his encouragement and support throughout the years.

I thank my coauthors and collaborators: Kuanfu Chen, Scott Llyod. Kuanfu's logical approach to problems has been inspirational. I will always remember insightful discussions with him. Scott has been very helpful in explaining basic concepts of content addressable memory. His collaboration is very much appreciated.

I thank Imran Hossain for his help in proofreading this dissertation.

I thank all the staff members of BSOE for their great efforts in building such a pleasant place to work. I want to particularly thank Carol Mullane and Emily Gregg for their genuine patience and support.

My friends: Hanyu Wang, Yang Song, Yu Han, Jian Lao, Andy Jeros, Emily Tucker, Jeremy Tucker, have been a huge treasure for me for the past five years. They are the source of my happiness. I will always remember them.

Finally I thank my parents Weiying Yang and Guoxin Zheng for their endless love and support. They have been extremely encouraging when I was facing obstacles. They will always be in my heart.

Chapter 1 Introduction

We are living in an era where two important trends are fundamentally changing the world around us. On one hand, we are entering a data-centric time where an enormous amount of data is generated every second via an ever-expanding collection of media: from countless mobile devices to billion-user social media, from fluctuating financial markets to various medical records, from computational biology to gigantic physical particle colliders. The immense amount of data affects our social behaviors, alters the landscape of the economy, and ultimately changes our perception of the rest of the world. On the other hand, the explosive increase of the data transforms the information technology industry where innovative software and hardware are demanded to digest the data and provide people with unprecedented experiences, services, and information.

1.1 Background

Over the past few decades, the fabrication process for integrated circuits has scaled well according to the Moore's law. The continuous scaling has benefited the industry with ever-increasing data processing capability on a chip at an ever-decreasing cost. For example, the processing power of an iPad is billions of times more than the first computer, ENIAC, and yet costs tens of thousands times less.

However the major benefits from CMOS technologies, such as high packing density, high processing speed, and low manufacturing cost, are getting desensitized primarily due to a number of factors. The most important one is the so-called 'power wall' [10]. As more transistors are integrated on a chip, the power-density of a chip does not stay constant but rather steadily increases over the years. It is predicted that future exascale systems using today's building blocks will consume power in the order of giga watts [11]. The main reason behind this power-limiting factor is the threshold voltage and in turn the supply voltage have not been scaling at the same pace of the feature size in part due to the leakage current concern [10, 12]. Another reason is that in pursuit of higher performance, the frequency has scaled up ahead of the technology until it tapered at around 3GHz [13].

The power issue seems more severe when the energy consumption from different hierarchies of memories is included. It is estimated that the leakage power from the

last-level cache could be larger than the power of a simple core running full out. For example, a breakdown of the power of a recent processor shows that nearly 50% of total power is consumed by caches and register files [10]. As far as DRAM (Dynamic Random Access Memory) is concerned, a significant energy overhead is added to the entire computing system due to the nature of energy-intensive DRAM accesses (on the order of few nJ compared to tens of pJ for internal cache accesses or logic operations).

Therefore, drastic innovations are needed in both hardware and software to design energy-efficient circuits and systems. On the hardware side, existing efforts include multi-core architectures, heterogeneous computing, power-aware dynamic control, and new memory technologies. While the first three focus on providing computation power with better processor energy efficiency, the last one aims at building memories with higher capacity, faster speed, lower cost, and less energy consumption. What is particularly exciting of new memory technologies is they have the potential of revolutionizing the existing computing architecture by collapsing memory hierarchies into fewer levels where both the access time/energy and the storage density will be dramatically improved, hence paving the road of building next-generation exascale computing systems [11].

1.2 Memristor

The memristor, short for memory resistor, was first postulated in 1971 by Chua as the fourth fundamental circuit element, with the rest being resistors, capacitors, and inductors [14]. Directly establishing the missing link between the charge and the magnetic flux, the memristors are essentially two-terminal devices whose resistances are dependent on the history of the supplied electrical signal. One of the most prominent signatures of the memristor is that under quasi-static external signals, a pinched hysteresis is observed on the input-output plot. Several years later in 1976, the concept of memristors was broadened to memristive systems/devices by Chua and Kang [15] where the dynamics of the systems can be described by a set of equations. The introduction of memristive systems successfully classified a large class of systems exhibiting pinched hysteresis, including thermistors, ionic systems, discharge tubes, and the Hodgkin-Huxley model for action potentials in neurons [16].

However, the significance of the two theoretical papers mentioned above was not recognized immediately. Although the pinched hysteresis has long been observed from various devices and systems, no connection was intentionally sought between the memristor theory and the observed experimental phenomenon. The breakthrough came in 2008, where researchers at Hewlett-Packard Laboratories announced the first nanoscale memristor was discovered [1]. A hysteresis was clearly observed, indicating the device underwent resistance change according to external stimulus. The device was able to retain its resistance when the power was shut down. A preliminary model was

also brilliantly proposed that not only described the device behavior but also fit well into the definition of memristive devices/systems.

Numerous memristors built from various material (e.g., TiOx [1, 17, 18], Ag/Si [19, 20], HfOx [21-24], TaOx [25, 26],) have been reported ever since. Depending on their applications, memristors are engineered to exhibit distinctive switching characteristics. For example, when used in resistive random access memories (RRAM), memristors are desired to have a large ratio of HRS (high resistance state) /LRS (low resistance state) and abrupt switching between the states; whereas analog or neuromorphic computing requires that intermediate resistance states of memristors can be reached reliably with good accuracies.

Despite the great variety of device material and switching mechanisms, existing memristors share properties that make them distinct from conventional CMOS devices. First, the miniature physical size (~10nm or even smaller) enables aggressive massive-scale integrations, which is especially beneficial in building high-density storage and computing systems. Second, memristors are promising candidates not only for future non-volatile memories but also other analog/digital computing systems with much reduced stand-by power consumption. Last but not least, the intriguing nonlinear dynamics of memristors resembling some of the basic elements in human brains could lead to ultra-high density neuromorphic computing systems that are much more energy-

efficient than their CMOS counterparts. Extensive research activities have been devoted to exploit many new possibilities offered by the new state variable 'resistance'. Memristors are widely investigated to build next-generation circuits/systems with much reduced form factor, stand-by power and manufacturing cost. Promising applications of memristors include: resistive random access memory (RRAM) [27-34]; reconfigurable nanoelectronic systems [35-37]; ultra-high density resistive Boolean logic and signal processing [38-42]; non-volatile VLSI computing [38, 43, 44]; nanoscale neuromorphic computing [20, 45-47].

A compact circuit model for memristors is essential in analyzing and designing circuits with memristors. While previously reported memristor models exhibit limitations in the model stability, versatility, and adaptability, we propose a new module-based memristors model that covers a wide range of diverse memristor behaviors. Coincident to the theoretic memristor behaviors, the proposed model uniquely reveals that an effective charge-flux constitutive relationship can always be obtained from various types of memristors. The stability of the proposed model is also significantly enhanced by adapting the new charge (or flux)-based window function. Simulation results demonstrate the proposed model is able to represent a wide variety of memristors.

1.3 CAM systems for data-intensive applications

Content addressable memory (CAM) takes a search content as the input, compares it with an array of stored data, and returns the matched address as the output. As the associative lookups are performed with high throughputs (theoretically single clock cycle), CAM provides superior search performance than other hardware- or software-based memory systems [48]. Conventional CMOS-based CAM systems use SRAMs as core storage elements which offer fast access at the cost of large cell area and significant dynamic power. As a result, CAMs are most commonly used in high-performance network routers/switches where fast lookups are carried out to determine the data flow within the network. But the size of a typical CAM is limited to Mb range and the cost-per-bit is much higher than traditional random access memories such as DRAM or SRAM [49, 50]. The large cell area of CMOS-based CAMs becomes a more severe issue when building ternary CAM (TCAM) systems where two bits are used in one storage unit. To improve the power and speed performance of CAM systems, innovations on both architectural and local levels are proposed including power-saving architectures and various matchline and searchline schemes [48].

In recent years, as conventional random access memories are continuously seen to be insufficient to keep up with the pace of data processing, the potential of using CAM systems in data-intensive applications are widely explored, where CAMs with high storage density, low energy consumption, and low search latency are desired. Particularly, emerging non-volatile memory (NVM) technologies, such as phase

change memory, magnetoresistive memory, and resistive memory, are actively investigated to take advantage of the unique properties of NVM devices, including: miniature physical dimension, fast access time, low energy read/write, and non-volatility. NVM-based CAMs offer exciting opportunities to dramatically increase storage capacity and achieve competitive energy and latency performances. Note that NVM devices are capable of storing information even when the power is shut off. This is especially attractive in building energy-efficient CAM systems where the system can be shut down instead of being kept running just to keep the stored information. We propose a memristor-based TCAM for data-intensive applications. A novel bit cell structure is presented that minimizes the cell area and achieves optimizations between latency and power consumption. Detailed design issues are discussed. The functions of the proposed mTCAM are demonstrated through circuit level simulations. Latency and power performance of the mTCAM are modeled, evaluated and projected to more advanced technology nodes. Among existing CMOS-based and NVM-based CAM systems, the proposed mTCAM achieves superior storage density, impressive energy efficiency and search speed.

Chapter 2 Review of memristor device and modeling

2.1 Theories of memristor

Traditionally three passive components have been studied in circuit theory: resistor, capacitor and inductor. Together they define three links between voltage, current, flux and charge, as shown in Figure 2.1. Resistor defines the relationship between voltage and current via Ohm's law. Capacitor defines the relationship between voltage and charge via $Q = C \cdot V$. Inductor defines the relationship between current and flux via $\Phi = L \cdot I$. The relationships between (Φ, V) and (Q, I) are governed by the

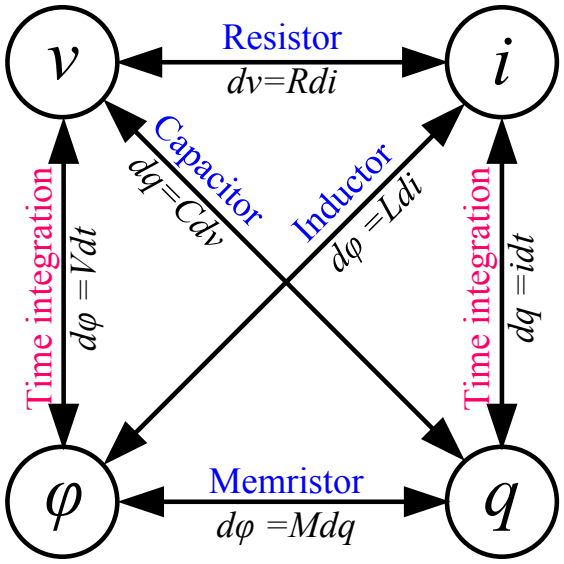


Figure 2.1 The relationships between voltage, current, flux, and charge.

definition of temporal integration. Therefore there is only one missing link between flux (Φ) and charge (Q). The concept of memristor was first discussed in 1971 by Chua who postulates memristor as the fourth fundamental circuit element to directly bridge a link between flux and charge [14]. As will be derived later in this chapter, memristor has the unit of ohm, but its resistance is not fixed but rather dependent on the history of the electrical signal. In other words, memristor is a resistor with memory effects (hence memory resistor, or memristors in short). As also proved by the original paper, memristor is a passive device where the instantaneous resistance is always non-negative.

In 1976, the concept of memristor was extended to 'memristive system/device' to accommodate a wide range of systems/devices that exhibit similar behaviors as memristors [15]. Chua and Kang stated that any system/device that can be described by a set of equations (Equations (2.1) and (2.2)) belongs to the category of memristive systems/devices.

$$\dot{x} = f(x, u, t) \tag{2.1}$$

$$y = g(x, u, t) \cdot u \tag{2.2}$$

The first equation reflects the dynamics of the system, stating the temporal change rate of the state variable x as a function of related parameters. The second equation defines the relationship between input signal u and output signal y. Note that both $f(\cdot)$ and $g(\cdot)$

can be functions of input u, state variable x and time t. For example, if output signal y represents voltage and input signal u represents current, the function g is defined as the memristance of the device. The concept of memristive system/device is rather broad that covers systems/devices such as: thermistor, Hodgkin-Huxley model of the action potential, discharge tube, among many others. Equations (2.1) and (2.2) establish the foundation of modeling any memristive devices. Almost all the modeling efforts of memristive devices have focused on how to describe the dynamics of the devices using this set of equations.

2.2 Evolution of memristor device

As it turns out, a great many devices have been reported to have similar characteristics as the originally memristor. In fact, the experiments history dates back two centuries; hysteresis loops are frequently observed in systems or devices such as biological ion channels, human blood, discharge lamps, electric arc, thermistors, among many others [51]. A general theory that explains all these interesting phenomena is that "hysteresis is typically noticed in systems and devices that possess certain inertia, causing the value of a physical property to lag behind changes in the mechanism causing it, manifesting memory" [51].

It is not until 2008 when the first nanoscale memristor was intentionally built by researchers from HP Labs [1]. The device is formed by inserting specially designed switching materials in between top and bottom platinum electrodes. The switching material contains two layers: one layer of titanium oxide (TiO_2) and another layer of oxygen-deficient titanium oxide (TiO_{2-x}). TiO_2 is an insulating material while the oxygen vacancies in TiO_{2-x} make it a highly conductive material. The resistance of the device is thus the combination of resistances from each individual layers. The applied voltage on the device modulates the boundary between the two layers, hence changing the device's resistance. For example, positive voltage will repel the positive-charged oxygen vacancies in the TiO_{2-x} layer into the TiO_2 layer, which moves the boundary downwards, causing the resistance to decrease; on the other hand, negative voltage will attract oxygen-vacancies towards the top electrode, which moves the boundary upwards, causing the resistance to increase. With the movement of the boundary dependent on both the amplitude and duration of the voltage, we now have a device whose resistance is controlled by the history of the externally applied electrical signals. Since the device will hold its resistance state when the power supply is shut down, the memristor is considered to be a non-volatile device.

Experiments presented in [1] demonstrate that under sinusoidal stimulus, the device exhibits hysteretic input-output relationships which match those illustrated in the original theoretical papers [14, 15]. To quantitatively explain the underlying

mechanism for switching, the paper propose a set of equations with the same format as Equations. (2.1) and (2.2) to describe the ion movement as a function of external signal and successfully correlate that with the resistance of the device. A good match is observed between simulations and measurements.

The discovery of nanoscale memristor inspires extensive and exciting research activities to explore different memristive material, investigate and understand switching mechanisms, and further engineer memristor devices for various applications. Memristors can be loosely classified as chemical devices and physical devices [44].

Chemical memristors rely on chemical reactions inside the device (e.g. redox) to switch between resistance states whereas physical memristors exhibit physical changes. Current research on chemical memristors have studied well beyond the original TiO_2 and expanded to material including oxide insulators (e.g., MgO, TiO_x, HfO_x, NbO_x, etc.) and non-oxide insulators (e.g., AlN, ZnTe, ZnSe, etc.). A widely accepted theory behind the switching mechanism is that a nanoscale conducting channel formed by electric field or joule heating alters the resistance between the two electrodes. By modulating the physical dimension of the channel, memristive effects can be observed. Note that the actual switching behavior is not only determined by the switching material itself, but also the interactions between the switching material and the electrodes.

Memristors based on physical changes include magnetic tunnel junctions, ferro-resistive switches, phase-change switches, etc. Among them, magnetic tunnel junctions and phase-change switches have been most intensively investigated. In magnetic tunnel junctions, two ferromagnets are separated by an insulator. Depending on the magnetic orientations, the tunneling current through the insulator can be different; hence distinct resistance states are realized. Phase-change switches alter their resistances by changing the morphology of the material (e.g., usually chalcogenide glass) between amorphous and crystalline. Although the concepts of these devices are attractive, they are not in the scope of this dissertation.

Over the years, the behaviors of memristors have evolved significantly to accommodate different applications. The evolution of memristors can be observed in the following aspects: ON/OFF ratio, endurance, retention, and *i*-*v* nonlinearity. The ON/OFF ratio is the ratio between the high resistance state (HRS) and the low resistance state (LRS) of the device. This ratio dictates the sensing margin when memristors are used in memory applications where binary information is manifested by HRS/LRS values. The larger the ratio, the easier it is to distinguish between '0' and '1'. In addition to large ON/OFF ratio, it is also desired that the absolute values of HRS and LRS are high to reduce the static power consumed during write/read operations. Endurance describes the number of switching cycles after which the device is considered non-functional. It is generally found that after certain number of switching

cycles, memristors freeze at a resistance state (HRS, LRS or intermediate state) and stop switching. Significant improvements have been made over the years to extend the endurance of memristors from 10^3 cycles to well beyond 10^{12} cycles [26]. The retention describes how well the device holds its resistance value. It is measured in number of years at a certain temperature. While different applications impose different temperature ratings, devices with retention of 10 years at 85°C are routinely reported and improvements on the retention time are expected [26, 52]. Current-Voltage (*i*-*v*) nonlinearity has been engineered into the device's switching to build energy-efficient crossbar RRAM array [25]. In a selection-device-free crossbar RRAM array, accessing a cell results sneak current paths appearing at half-selected cells, which are not on the desired signal path but experience finite voltage drops. To reduce the additional power consumption and sensing ambiguity from the sneak currents, nonlinear switching is desired where the current at low voltage amplitude is significantly smaller than that at high voltage amplitude [25, 44].

2.3 Existing memristor models

To use memristors to build next-generation electronic circuits/systems, it is necessary to have a circuit model for memristors to facilitate the circuit analysis and design. Ideally a memristor model would reflect the behavior of the device in both static

and transient conditions. The model is expected to be compatible with common circuit simulation environments such as SPICE, Verilog-A, etc. Moreover, the model is also desired to reflect important secondary characteristics of the device such as statistical behavior, retention time, endurance, etc., in both short-term and long-term time scales.

Memristor models with distinct structures and purposes have been proposed. While some models target superior model accuracies by adopting complex numerical equations, others focus on compact modeling with less computation complexity. The rest of this section reviews the existing memristor models and identifies their respective advantages and disadvantages.

2.3.1 HP's preliminary physical model

As the first attempt, a preliminary model was proposed to explain the behavior of the first nanoscale memristor [1]. The concept and the formulation of the model are illustrated in Figure 2.2. The model considers that the memristor has two layers of material: TiO_2 (undoped) and TiO_{2-x} (doped). The total resistance of the device is the sum of the resistance from each layer. When the device is supplied with external electrical signals, the boundary between the two layers moves, causing the resistance of both regions to change, hence the change of the total resistance. For example, a positive voltage will repel positively-charge oxygen vacancies into the insulating TiO_2

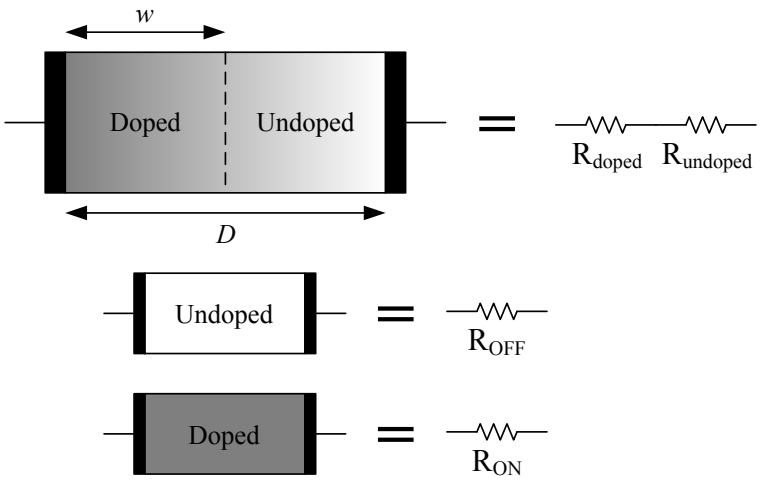


Figure 2.2 HP's preliminary memristor model.

layer, causing the total resistance to decrease, and vice versa. To model the device dynamics, a state variable w is chosen to represent the width of the TiO_{2-x} region. The relationship between the state variable and the electrical signal is described as,

$$\frac{dw(t)}{dt} = \mu_V \frac{R_{ON}}{D} i(t) \tag{2.3}$$

where μ_V is the dopant mobility, R_{ON} is LRS of the device, D is the semiconductor film thickness. Equation (2.3) states that the speed of the boundary's movement is directly dependent on technology-dependent constants (μ_V, R_{ON} and D) and the instantaneous current ($i(t)$). The relationship between voltage and current is described as,

$$v(t) = (R_{ON} \frac{w(t)}{D} + R_{OFF} \frac{D - w(t)}{D}) \cdot i(t) \tag{2.4}$$

where R_{OFF} is HRS of the device. Equation (2.4) states a modified Ohm's law where the memristance is dependent on a time-varying state variable w. The presentation of the above two equations is significant: for the first time, mathematic equations that have the same format as those in [15] are used to describe the behavior of a memristor, successfully linking a physical device to the concept of memristors.

The memristor model based on Equations (2.3) and (2.4) needs further modifications to reflect practical characteristics of the device. For example, the resistance of a memristor cannot change without boundaries, implying that the state variable w should be restricted between 0 and D. Furthermore, the movement of the boundary is not necessarily always linear to the applied current: nonlinear dopant drift phenomenon is observed and should be captured by the model as well. The concept of window functions is proposed to modify the equation that describes the dynamics of the state variable, Equation (2.3). For example, a window function $H(w)$ is proposed as follows,

$$H(w) = 4 \cdot \frac{w(D-w)}{D}, 0 \leq w \leq D \tag{2.5}$$

where $H(w)$ is zero when w is either zero or D and $H(w)$ reaches to its maximum value of 1 when $w = D / 2$. The modified dynamic equation for state variable is thus,

$$\frac{dw}{dt} = \mu_V \frac{R_{ON}}{D} i(t) \cdot H(w) \tag{2.6}$$

As can be seen, $H(w)$ has two functionalities. First, $H(w)$ is designed such that when the state variable w reaches its boundaries, it will be held at the boundary value. This

defines a valid range of w to change, which successfully limits the memristance between HRS and LRS values. Second, the shape of $H(w)$ helps to mimic the nonlinear dopant drift effect caused by the enormous electric field due to the miniature size of the device. Under the same current strength, the state variable w changes fastest when it is at the middle of the valid range whereas it takes a lot more charge (time-integrated current) to change w when w is closer to the two boundaries.

One of the primary issues with HP's preliminary physical model is the model stability. The window function described in Equation (2.6) is dependent on the state variable itself and nullifies when the state variable has a value of 0 or D. As a result, once the state variable reaches the boundary values, it can no longer change anymore. From the perspective of the memristance, this causes the device to freeze at either LRS or HRS forever. Obviously this violates the actual experimental results. This stability issue is termed 'backing problem' hereafter.

2.3.2 Modifications on HP's preliminary physical model

To improve the modeling accuracy and enhance the modeling stability, a group of models have been proposed that follow the same switching mechanism proposed by [1] but certain modifications are included.

Memristor models presented in [3] and [53] focus on creating more sophisticated window functions that are supposed to model the nonlinear dopant drift effect more accurately. Compared to the second-order polynomial in [1], the exponential function in [3] enhances the switching nonlinearity by forcing more abrupt attenuations when the state variable is approach boundary values. A versatile window function is proposed in [53] that not only models the switching nonlinearity but also is adjustable to capture a wide range of device dynamics. Nonetheless, the models proposed above still suffer from the backing problem which inspires the following models.

Biolek et al. proposed a window function to solve the backing problem, as shown below [4],

$$f(x) = 1 - \left(x - stp(-i)\right)^{2p}, (0 \le x \le 1) \tag{2.7}$$

where

$$x = \frac{w}{D} (0 \le w \le D) \tag{2.8}$$

p is a constant that adjusts the shape of the window function and $stp(\cdot)$ is the step function which is described as follows,

$$stp(i) = \begin{cases} 1 & i \ge 0 \\ 0 & i < 0 \end{cases} \tag{2.9}$$

When the current is positive, w or x will increase according to (2.7) but the rate of the increase dx/dt is continuously adjusted by $f(x) = 1 - x^{2p}$ so that it achieves its maximum value when $x = 0$ and until it is nullified when $x = 1$. As soon as the current negates its polarity, $f(x)$ experiences a sudden change from 0 to 1, which enables w or x to move away from the boundary values. As a result, window function described in (2.7) successfully solves the backing problem by setting different dx/dt values depending on the current polarity. The shape of the window function is illustrated in Figure 2.3 for $p = 5$.

Another attempt to solve the backing problem is proposed by [54] where the change rate of state variable is written as follows,

$$\frac{dx}{dt} = x(t) \cdot \big(1 - x(t)\big) \cdot \frac{1}{C_m} \cdot i(t) + \kappa(0.5 - x(t)) \tag{2.10}$$

where κ is a small constant. Note that while the first part of the right hand side has the same format as the window function proposed in (2.5), the backing problem is mitigated by the finite value of the second part when $x = 0$ or $x = 1$.

Modified window functions in [4] and [54] have their limitations as well. The asymmetric shape seen in Figure 2.3 indicates the speed at which the state variable enters a boundary is drastically different from that at which it leaves from there. This phenomenon is rarely seen from experimental results and is difficult to imagine. The

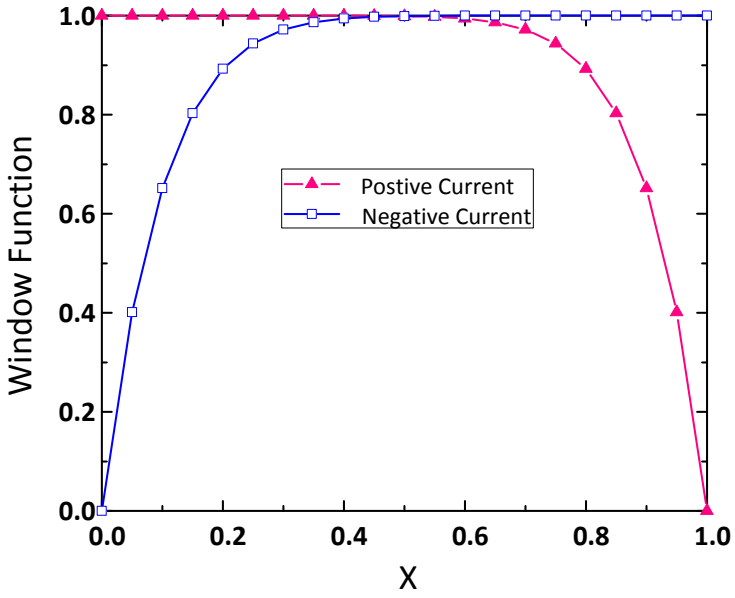


Figure 2.3 Window function proposed in [3] when p = 5. Note that the shape of the window function depends on the current polarity.

finite component in (2.10), however small, causes the resistance to change beyond LRS and HRS values, which alters the device properties and may violate the design rules especially in long-term simulations.

2.3.3 Advanced physical models

As the understanding of the switching mechanisms of memristors has been continuously improved, a group of more advanced physical models are proposed to achieve a better agreement between simulations and measurement results. These models rely on state-of-the-art knowledge on nanoscale material and are supported by physical evidences observed from various microscopy methods (TEM, SEM, AFM,

etc.) and chemical studying tools (XRD, etc.). Commonly focusing on a specific type of material and switching mechanism, these models commonly adopt more complex computation formula to achieve accurate results.

The switching dynamics of TiO_2 memristors are investigated in [18]. A circuit model for the device is proposed that consists of an Ohmic resistor in series with an electron tunnel barrier. Remarkably, a single state variable *w*, which is the width of the tunnel barrier, emerges from the analysis. Simmons *i*-*v* equations are used in the model that fits the data well with physically reasonable parameters. The analysis revealed a notable nonlinear property of memristors that the energy required to switch the device is exponentially dependent on the current. The SPICE version of the model is implemented in [55]

Memristors using tungsten oxide (WO_x) have been fabricated [56]. The memristive effect is attributed to the migration of oxygen vacancies, which modulates the interplay between Schottky barrier emission and tunneling at the WO_x/electrode interface. Instead of using length index as the state variable, this paper proposes using an area index as the new state variable. The growth of the state variable upon the external bias is in parallel with existing current path instead of in series as in [18]. The proposed model successfully reflects the nonlinear dynamics of the WO_x memristor.

A model is proposed to explain the dynamic resistive phenomena observed in a wide range of memristive devices using physical mechanisms [57]. The model is based on the general framework presented in (2.1) and (2.2). The state variable takes the form of filamentary dimension of the conducting region inside the device. The rate equation describes the nonlinear relationship between the growth rate of the filament and the applied voltage. A complex *i-v* equation was also established based on the assumption that the tunneling current dominates the total current. Although the entire model is constructed by continuous equations, it is able to reflect phenomena such as threshold effect, voltage-dependent switching time, and multi-level effect under complex circuit conditions. By using the proposed modeling framework, secondary effects such as lateral expansion, joule heating can also be represented.

2.3.4 Generic models

Memristor models discussed so far focus on devices with certain material and switching mechanisms. Although they provide superior accuracy (especially advanced

physical models), their adaptability is limited. Generic memristors models that can reflect common devices characteristics and represent a wide range of memristors are valuable to circuit designers especially when experimenting with different types of devices is necessary.

In [2], a general modeling framework is proposed where the state variable is the memristance M itself. The change rate of M is described as follows,

$$\frac{dM}{dt} = f(V)[\theta(V) \cdot \theta(M - M_1) + \theta(-V) \cdot \theta(M_2 - M)] \tag{2.11}$$

where $\theta(\cdot)$ is the step function, M_1 and M_2 are memristance boundary values and $f(\cdot)$ represents certain function between the effective driving force to change M and voltage input V. The boundary assurance is enabled by the second term on the right hand side of (2.11) where both the polarity of the input voltage and the memristance value are

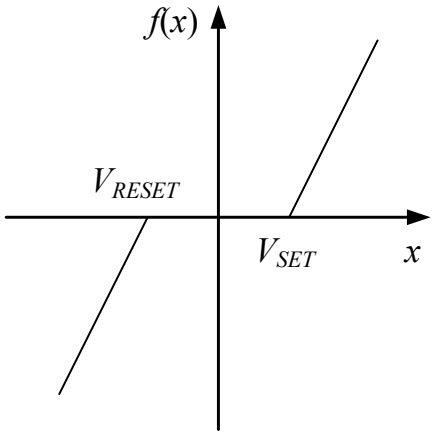


Figure 2.4 Piece-wise linear function used in [2] to model SET/RESET thresholds.

detected to generate 1 or 0. The function $f(\cdot)$ offers additional flexibility to shape the switching dynamics. For example, $f(\cdot)$ can adopt piece-wise linear function as shown in Figure 2.4 to reflect the threshold voltages for SET and RESET.

The model presented in [8] employs three boundary conditions to ensure the state variable stays within boundary values without experiencing backing problems. The threshold effect is naturally included in the three conditions to allow different SET/RESET threshold magnitudes. One of the limitations of the proposed model is that it only allows a linear relationship between the state variable change rate and the voltage input.

Lehtonen et al. propose a generic exponential model for thin-film memristive devices [58]. The model is developed based on two assumptions: 1) the *i-v* characteristic of the device is dominated by tunneling phenomenon; 2) the state variable is assumed to depend highly nonlinearly on the voltage across the device. Specifically the model is described as follows,

$$I = c_1 \cdot w \cdot \sinh(d_1 V(t)) \tag{2.12}$$

$$\frac{dw}{dt} = c_2 \cdot \sinh(d_2 V(t)) \tag{2.13}$$

where w is the state variable ($0 \leq w \leq 1$) and c_1, c_2, d_1, d_2 are constants for fitting purposes. Simulations on the model have demonstrated the threshold voltage depends

on the time scale of the operation, which was previously reported in experiments [19]. Yakopcic et al. add features to the above model including a voltage threshold for state variable motion and a nonlinear velocity function for oxygen vacancy or dopant drift [6, 59]. The model is claimed to be the first one to be quantitatively correlated to multiple devices for both sinusoidal and repetitive sweeping inputs.

Finally memristor models are developed to retain the theoretical foundation of memristors, i.e., constitutive charge-flux relationship (or CCR hereafter). Biolkova et al. propose a model that directly establishes a single-valued function between charge and flux based on which the voltage- (or current-) controlled memristance are derived [60]. However it might not be straightforward to use it to reflect actual device dynamics because instead of current-voltage relationships, CCRs are not readily available from device characterizing setups. Moreover, practical phenomena observed from actual devices (e.g., bounded memristance range, threshold voltages, etc.) cannot be directly reflected in the model. Shin et al. develop a compact model based on CCR [5]. Analysis shows that ideal CCR does not exist in actual memristors with bounded memristance range. Instead an equivalent CCR is obtained where excessive voltage and current are discarded when the memristance reaches one of the boundaries. A SPICE model is developed where the bounded memristance was implemented by adopting diodes and voltage sources. Shin's model reveals that while modern memristors differ greatly in

their materials and switching mechanisms, they could all exhibit equivalent constitutive charge-flux relationships as theoretical memristors do.

2.4 Applications of memristors

Memristors' unique properties such as miniature physical size, non-volatility, low switching energy, and high switching speed have inspired extensive efforts in exploring the potential from utilizing resistance instead of charge as the state variable.

One of the prominent and straightforward applications of memristor is resistive random access memory (RRAM) [27-34]. Memristors in RRAMs are used as non-volatile memory components where binary information is stored when the device is at either HRS or LRS state. The unique properties of the memristor enables building high-density non-volatile memories that possess fast access time and low write energy. Such memories have the potential to replace existing non-volatile memories such as flash memory and hard disk with much higher storage density and lower manufacturing cost. Current research activities on RRAMs focus on architecture, circuit, and device itself. Existing architectures include 0T1R and 1T1R crossbars where the former does not include any access device and the latter as one transistor in series with one memristor in each bit cell [23, 31, 32, 61-63]. The tradeoffs between storage density, energy consumption, and reliability have been widely explored in both architectures. Energy-

efficient peripheral circuits have been designed to support reliable read/write operations on memristors. Research on memristor devices has focused on engineering nonlinearities into the switching dynamics and improving variability, endurance of the devices [21, 25, 26].

Memristors can be used to implement Boolean logic, including threshold logic [35, 64], and stateful logic [38-40, 42, 43, 65-67]. Memristors are also used to build reconfigurable FPGA [36, 37]. Finally memristor-based neuromorphic computing systems are proposed where memristors are employed in building neural networks that help study and emulate cognitive systems [20, 45-47, 56, 68, 69].

Chapter 3 Modular compact memristor model

In this chapter, we present in details of the proposed module-based compact model for a wide range of memristors. After discussing the motivations, the mathematical formulation of the model is explained. The modular structure of the model is presented where each block is designed to reflect certain aspects of the device's behaviors. Computer simulations on three types of memristors demonstrate the stability, versatility, and adaptability of the modular compact memristor model. After comparing the proposed model with existing memristors models, a conclusion is drawn.

3.1 Motivation

Chapter 2 discusses that various memristor models are developed based on different mathematical formulations, different material, and switching mechanisms. While each model is able to represent certain aspects of the device behavior, the limitations of existing memristor models are summarized as follows. First, important device switching phenomena are missing from some models. For example, threshold voltages are not captured by [3, 4, 53, 54, 70, 71] and nonlinear state variable change rate is not reflected in [2, 4, 7, 8, 54, 72, 73]. Second, the 'backing problem' is commonly observed due to the fact that the window function is dependent on the state

variable itself [3, 4, 6, 53, 59, 70]. Third, while most of the modeling efforts have focused on current-voltage relationship, we argue that charge-flux relationship is an essential part of memristor dynamics and should be included in the model as well [5, 12]. Fourth, some physical models are highly restricted to certain device material with specific switching mechanisms and are less successful in simulating diverse device behaviors [6, 18, 55, 57, 59, 73-77]. Finally, secondary effects including process variation and voltage-dependent switching time are neglected by some models.

It is thus desired to have a memristor model that is compatible with common circuit simulation environments and has great versatility and adaptability to assist circuit designers to explore a large variety of memristors and optimize the system's performance accordingly. Moreover the model should mitigate problems from previously reported models. We propose a module-based modeling approach that can cover a wide range of memristive devices from theoretical memristors to practical RRAM devices. Constructed in a modular structure, the model is able to capture various device behaviors such as threshold voltages for SET/RESET, nonlinear memristance switching rate, finite memristance switching range, nonlinear *i*-*v* relationships, and device parameters with statistical variations. As the proposed model is derived from the original memristor definition, it offers a unique perspective: although memristive devices may exhibit drastically distinct behaviors, equivalent charge-flux constitutive relationships can always be obtained. The window function in the proposed model is

naturally defined in the memristive charge or flux domain (defined as the temporal integration of the current or voltage that causes the device to behave memristively), which solves the 'backing problem' yet still can model the nonlinear memristance switching rate. Simulations on three types of memristive devices demonstrate the proposed model can be used to describe a broad range of memristive devices.

3.2 Modeling for memristors

Memristors are classified into multiple categories depending on their voltage-current dynamics. Theoretical memristors are postulated to provide constitutive relationship between charge (q) and flux-linkage (φ). Practical memristors are vastly different from ideal memristors in terms of device dynamics, mainly due to limited memristance switching range, linear/nonlinear memristance switching rate, and threshold effects. In this section, we discuss the theoretical analyses and modeling approaches for different types of memristive devices ranging from theoretical memristors to practical RRAM devices. We show that such studies are necessary in developing a modular compact model for memristors.

3.2.1 Theoretical memristor

The theoretical memristor was proposed to directly establish a constitutive relationship between charge (q) and flux (φ), i.e., $q = f(\varphi)$, or $\varphi = g(q)$. For a voltage-actuated memristor, for example, the constitutive q-φ relationship can be expressed as:

$$q = f(\varphi) \rightarrow \frac{dq}{dt} = \frac{df(\varphi)}{d\varphi} \cdot \frac{d\varphi}{dt} \tag{3.1}$$

By defining the flux-controlled memductance G_m (short for memory conductance) as:

$$G_m(\varphi) = \frac{dq}{d\varphi} = \frac{df(\varphi)}{d\varphi} \tag{3.2}$$

(3.2) can be further reduced to:

$$i = G_m(\varphi) \cdot v \tag{3.3}$$

where by definition $i = \mathrm{d}q/\mathrm{d}t$ and $v = \mathrm{d}\varphi/\mathrm{d}t$. Similar to the approach presented in [5, 12], this work uses a measureable quantity, G_m, as the state variable in modeling the device's dynamics. In such a way, the model is no longer restricted to devices with specific physical mechanisms. The change rate of memductance, $\dot{G}_m \equiv \mathrm{d}G_m/\mathrm{d}t$, can be obtained by taking time derivative on both sides of (3.3), as:

$$\dot{G}_m(\varphi, v) = \frac{dG_m(\varphi)}{dt} = \frac{d^2 f(\varphi)}{d\varphi^2} \cdot v \tag{3.4}$$

Letting $h(\varphi) \equiv \mathrm{d}G_m/\mathrm{d}\varphi \equiv \mathrm{d}^2 f(\varphi) / \mathrm{d}\varphi^2$ will reduce (3.4) as:

$$\dot{G}_m(\varphi, v) = h(\varphi) \cdot v = \alpha \cdot H(\varphi) \cdot v \tag{3.5}$$

where $H(\varphi) = |h(\varphi)| / |h(\varphi)|_{MAX}$ and $\alpha = h(\varphi)/H(\varphi)$. Note that $H(\varphi)$ is a normalized function of $h(\varphi)$. Equation (3.5) reveals that:

a) the change rate of G_m is proportional to the instantaneous voltage, i.e., $\dot{G}_m \propto \alpha \cdot v$;

b) the change rate of G_m is also modulated by $H(\varphi)$, which is a function of flux.

In fact, if we consider an idealistic case where $\tilde{\dot{G}}_m = \alpha \cdot v$, $H(\varphi)$ becomes a masking function that further modulates $\tilde{\dot{G}}_m$ to yield $\dot{G}_m$, that is:

$$\dot{G}_m = \tilde{\dot{G}}_m \cdot H(\varphi) \tag{3.6}$$

Hence $H(\varphi)$ will be termed as the 'window function' hereafter. Note that $H(\varphi)$ reflects the memory effect of the device since its value is a function of the history of the actuated voltage. To correctly model the ideal memristors, $H(\varphi)$ cannot be any arbitrary function. For voltage-actuated memristive devices, physically it makes sense that as the flux increases, the memductance changes monotonically. Since ideal memristors do not have boundaries in their memductances, $h(\varphi)$ should be any function with a single polarity, hence the normalized $H(\varphi)$ can be any function with non-negative values and a maximum of unity across the entire φ domain.

Similar to voltage-actuated memristors, current-actuated memristors can be modeled with a charge-controlled memristance R_m whose change rate is proportional to the instantaneous current and a charge-dependent window function, as:

$$v = R_m(q) \cdot i \tag{3.7}$$

$$\dot{R}_m = \tilde{\dot{R}}_m \cdot H(q) \tag{3.8}$$

where $\tilde{\dot{R}}_m = \alpha \cdot i$ and $H(q)$ is the charge-dependent window function.

The conceptual formulations of the flux-controlled memductance change rate, $\dot{G}_m$ and the charge-controlled memristance change rate, $\dot{R}_m$ is depicted in Figure 3.1, where the rate functions are the products of idealistic rate functions and corresponding

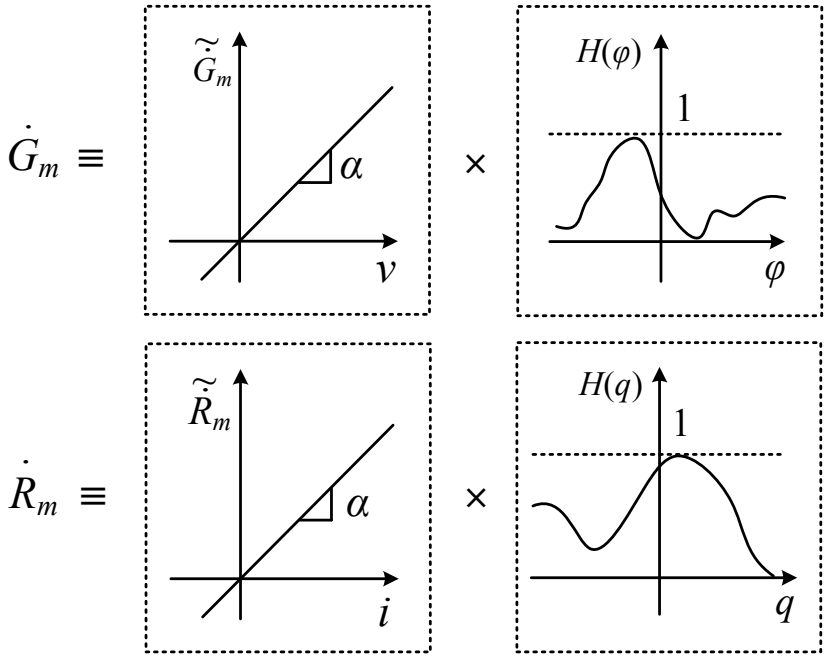


Figure 3.1 Conceptual formulation of $\dot{G}_m$ and $\dot{R}_m$ for theoretical memristors.

window functions, i.e., (3.6) and (3.8). Note that the idealistic rate function is a linear function in the domain of voltage or current, whereas the window function is defined in the entire flux or charge domain. Once the rate function is established, the device's memductance or memristance can be attained by simply integrating the rate function over time. As an example, Figure 3.2 illustrates the realization of a voltage-actuated theoretical memristor. The input to the model is the voltage v across the device. The flux φ is obtained by integrating v over time with the initial condition of φ_0. $\dot{G}_m$ is calculated by multiplying the window function $H(\varphi)$ with the scaled instantaneous voltage $\alpha \cdot v$. G_m is from integrating $\dot{G}_m$ over time with initial condition of G_{m0}. Finally the current passing through the device is attained by multiplying v with G_m, according to (3.3).

From Figure 3.2, the computed memductance value can be expressed as:

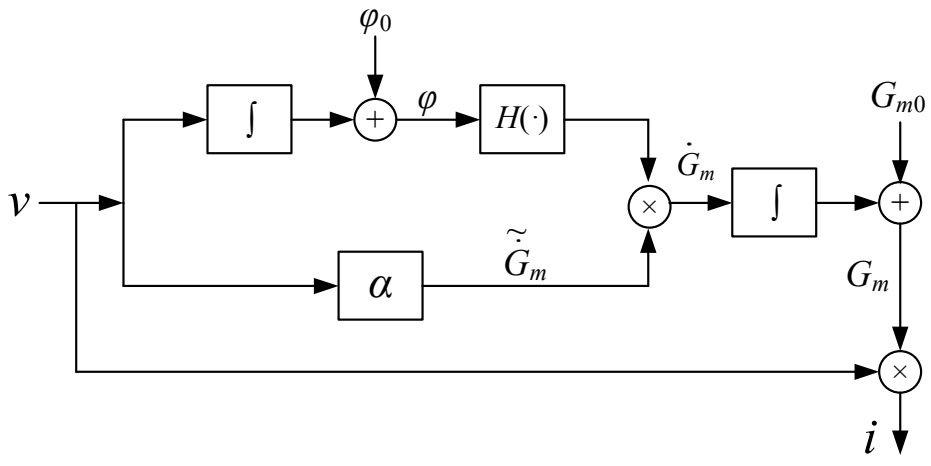


Figure 3.2 Modular model for theoretical memristors.

$$G_m = G_{m0} + \int_{-\infty}^{t} [\alpha \cdot v \cdot H(\varphi)] \cdot d\tau \tag{3.9}$$

With a special uniform window function, i.e., $H(\varphi) = 1$, the memductance G_m becomes linearly proportional to the flux φ, that is:

$$G_m = G_{m0} + \int_{-\infty}^{t} (\alpha \cdot v) \cdot d\tau = G_{m0} + \alpha \cdot \varphi \tag{3.10}$$

As can be imagined, a current-actuated ideal memristor can be modeled using the similar schematic in Figure 3.2 by replacing the quantities of voltage, flux, memductance with current, charge, and memristance. The computed memristance value can be expressed as:

$$R_m = R_{m0} + \int_{-\infty}^{t} [\alpha \cdot i \cdot H(q)] \cdot d\tau \tag{3.11}$$

Similarly a uniform window function, i.e., $H(q) = 1$, the memristance is linearly proportional to the charge q, that is:

$$R_m = R_{m0} + \int_{-\infty}^{t} (\alpha \cdot i) \cdot d\tau = G_{m0} + \alpha \cdot q \tag{3.12}$$

3.2.2 Memristor with bounded memristance

The memristance of a theoretical memristor can be freely changed without boundaries and the device is always operating memristively. Practical memristors however have limited memristance switching range, i.e., LRS $\leq R_m \leq$ HRS where LRS (low resistance state) and HRS (high resistance state) are the minimum and the maximum boundaries. As discussed in [5], practical memristors behave like linear resistors at memristance boundaries and the original charge-flux constitutive relationship no longer holds. Instead, an equivalent constitutive relationship between memristive-charge (q_m) and memristive-flux (φ_m) can be observed by siphoning the excessive input at memristance boundaries. In addition, the switching mechanisms of the practical devices require window functions with specific shapes. For example, the

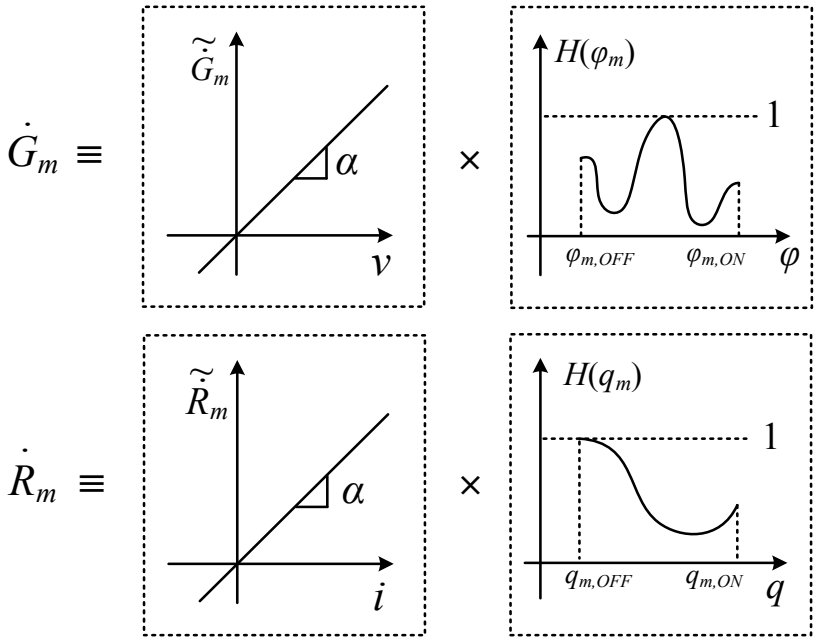


Figure 3.3 Conceptual formulation of $\dot{G}_m$ and $\dot{R}_m$ for memristors with bounded memristances.

memristor in [1] has the following properties: 1) bounded memristance switching range where the memristance stops changing once it reaches the boundary values; 2) the memristance switching rate could be linearly or nonlinearly dependent on the instantaneous input (widely referred as linear/nonlinear dopant drift effect).

To model memristors with boundary assurances, the following steps are implemented. First, boundaries are applied on q_m or φ_m within their respective limits ($q_{m,OFF} \leq q_m \leq q_{m,ON}$, $\varphi_{m,OFF} \leq \varphi_m \leq \varphi_{m,ON}$). This is realized by discarding the actuated current or voltage when q_m or φ_m reaches either of the boundaries until the current or the voltage changes its polarity. Second, R_m or G_m should stop changing as soon as q_m or φ_m reaches the boundaries. This is done by forcing the window function $H(q_m)$ or $H(\varphi_m)$ to zero at boundaries. The concepts of formulating $\dot{G}_m$ and $\dot{R}_m$ are drawn in Figure 3.3 where the window functions $H(\varphi_m)$ and $H(q_m)$ are valid only between two boundaries in φ_m and q_m domains respectively and are set to zero at the boundaries.

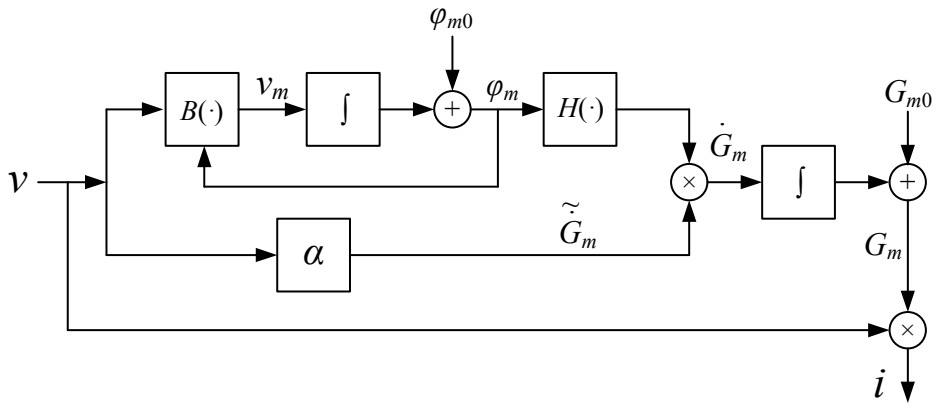


Figure 3.4 Modular model for memristors with bounded memristances.

As an example, a voltage-actuated memristors with bounded memristance can be modeled by a modified schematic shown in Figure 3.4, where a new function block $B(\cdot)$ is introduced and $H(\cdot)$ is modified. $B(\cdot)$ takes both v and φ_m as inputs, and computes the memristive voltage (v_m) to be used to calculate φ_m. The behavior of $B(\cdot)$ is summarized as:

$$B(v, \varphi_m) = \begin{cases} v, & (v > 0 \; and \; \varphi_m < \varphi_{m,ON}) | (v < 0 \; and \; \varphi_m > \varphi_{m,OFF}) \\ 0, & otherwise \end{cases} \quad (3.13)$$

$B(\cdot)$ is designed such that when φ_m is between boundary values, the input voltage v is allowed to be freely integrated over time; when φ_m reaches boundary values, $B(\cdot)$ prevents φ_m from further updating by forcing v_m to zero until v reverses its polarity. As a result, the memristive flux φ_m is effectively confined between boundary values. The window function $H(\cdot)$ is:

$$H(\varphi_m) = \begin{cases} g(\varphi_m), & \varphi_{m,OFF} < \varphi_m < \varphi_{m,ON} \\ 0, & otherwise \end{cases} \quad (3.14)$$

where $0 \leq g(\cdot) \leq 1$ and is determined by the detailed switching mechanism inside the device. For example,

$$g(\varphi_m) = 1 \quad (3.15)$$

models devices with the linear dopant drift effect; and

$$g(\varphi_m) = \left(\varphi_m - \frac{\varphi_{m,ON} + \varphi_{m,OFF}}{2}\right)^2 \Big/ \left(\frac{\varphi_{m,ON} - \varphi_{m,OFF}}{2}\right)^2 \tag{3.16}$$

is an example to represent the nonlinear dopant drift where $\dot{G}_m$ experiences fastest changes in the middle of the valid φ_m range and gradually slows down to zero as φ_m reaches to boundaries.

Similarly a current-actuated memristive device with bounded memristance can be modeled using the schematic shown in Figure 3.4 by replacing quantities and modifying $B(\cdot)$, $H(\cdot)$ as:

$$B(i, q_m) = \begin{cases} i, & (i > 0 \; and \; q_m < q_{m,ON}) | (i < 0 \; and \; q_m > q_{m,OFF}) \\ 0, & otherwise \end{cases} \tag{3.17}$$

$$H(q_m) = \begin{cases} g(q_m), & q_{m,OFF} < q_m < q_{m,ON} \\ 0, & otherwise \end{cases} \tag{3.18}$$

The backing problem has been observed from previously reported window functions [3, 4, 6, 53, 59, 70] where the device can be locked to a memristance state and fails to acknowledge external electrical signals. The primary reason is the window function is directly defined in the memristance (or memductance) domain. In the proposed model, the window function is established in q_m or φ_m domain to indirectly control R_m or G_m. This allows the modeling of the bounded memristance with the linear/nonlinear dopant drift effect and solving the backing problem at the same time.

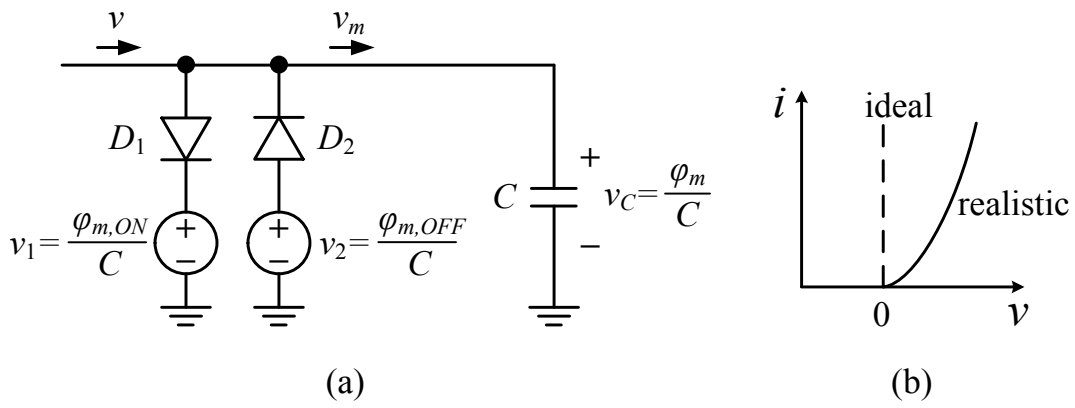


Figure 3.5 (a) Schematic of the clamping circuit, and (b) *i*-*v* characteristics of ideal and realistic diodes.

The bounded memristance is modeled by anti-parallel diodes and voltage sources (shown in Figure 3.5(a)) in previous SPICE models [5, 78]. The current flowing into this circuit represents the actuated v (or i) and the current flowing into the capacitor C represents v_m (or i_m), thus the voltage across the capacitor is a scaled version of φ_m (or q_m), i.e., $v_C = \varphi_m / C$ (or $v_C = q_m / C$). The current-voltage characteristics of an ideal diode and a realistic diode are illustrated in Figure 3.5(b). Since the ideal diode offers abrupt resistance change at threshold voltages, the clamping circuit will limit v_C exactly between two limits set by v_1 and v_2, i.e., $v_2 \leq v_C \leq v_1$. If v_1 and v_2 symbolize the scaled boundary values, φ_m (or q_m) is successfully bounded. On the other hand, realistic diodes do not exhibit clear threshold voltages due to their gradual resistance change, so careful simulations are required to tune v_1 and v_2 to achieve the desired boundary values.

3.2.3 Memristor with threshold voltages and nonlinear *i-v* relationship

Many fabricated memristors exhibit threshold voltages where the state of the device is not significantly changed unless the applied voltage exceeds certain values. Devices with such behaviors are especially valuable in applications such as RRAM where clear threshold voltages are preferred for read/write operations. The threshold voltages can be modeled by applying clipping functions on the actuated voltage v before it is used to calculate the state variable. The shape of the clipping function determines the amount of attenuation applied on v when it is below the threshold. Ideal and realistic clipping functions with piece-wise linear and nonlinear transfer functions are shown in Figure 3.6(a) and (b), respectively. An ideal clipping function (shown in Figure 3.6(a)) can be described as:

$$\tilde{v} = T(v) = \begin{cases} v - v_{SET}, & v \geq v_{SET} \\ 0, & v_{RESET} < v < v_{SET} \\ v - v_{RESET}, & v \leq v_{RESET} \end{cases} \tag{3.19}$$

where v_{SET} and v_{RESET} are the threshold voltages for SET and RESET, respectively, and they may have different magnitudes. Equation (3.19) sets $\tilde{v}$ to zero when v is in between v_{SET} and v_{RESET}, so that R_m (or G_m) remains unchanged. Thus the non-disruptive read scheme can be modeled where the read voltage is below the thresholds so that the

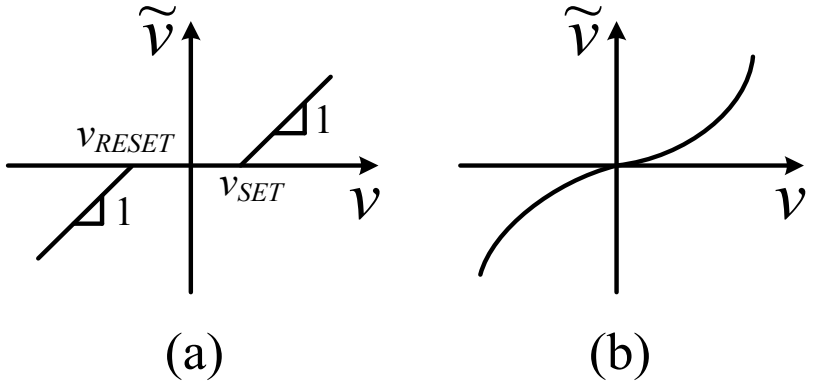


Figure 3.6 Linear (a) and nonlinear (b) clipping functions.

state of the device can be retrieved without being disturbed. On the other hand, modeling the disruptive read, where the device experiences finite change or even complete switching of its state even when v is well below thresholds [17, 25, 44, 57], is essential in applications where a memory cell is frequently read but seldom written. The realistic clipping function (shown in Figure 3.6(b)) can adopt nonlinear transfer functions to model disruptive read, for example:

$$\tilde{v} = T(v) = \begin{cases} A \cdot (e^{mv} - 1), & v \geq 0 \\ B \cdot (e^{nv} - 1), & v < 0 \end{cases} \tag{3.20}$$

where A, B, m and n are coefficients. Suppose the device is at LRS and it is read by a negative voltage pulse with an amplitude of v_R ($v_{RESET} < v_R < 0$) for a duration of $\Delta\tau$, the non-zero value of $\tilde{v}$ shifts the device towards HRS every time a read command is issued. The number of consecutive reads allowed until the device still holds the valid state is jointly determined by v_R, B and $\Delta\tau$.

Another feature seen from recent thin-film memristive devices is a nonlinear *i*-*v* relationship exists even when the device resides at one of the memristance boundaries [6, 18, 55, 73-75]. This nonlinearity is primarily originated from the MIM junction in the device and is further engineered to reduce sneak currents in selection-device-less passive RRAM arrays [25]. The nonlinear *i*-*v* relationship can be captured by adding a dedicated block $N(\cdot)$, e.g., for a voltage-actuated memristor, the following function can be realized:

$$\hat{v} = N(v) = \begin{cases} v, & \text{linear i-v relationship} \\ \sinh(v), & \text{nonlinear i-v relationship} \end{cases} \tag{3.21}$$

$$i = G_m(\varphi) \cdot \tilde{v} \tag{3.22}$$

where (3.21) calculates a generalized actuated voltage $\hat{v}$ that is multiplied by G_m in (3.22) to yield the current. Depending on the content of $N(\cdot)$, memristive devices with linear or nonlinear *i*-*v* relationships can both be modeled.

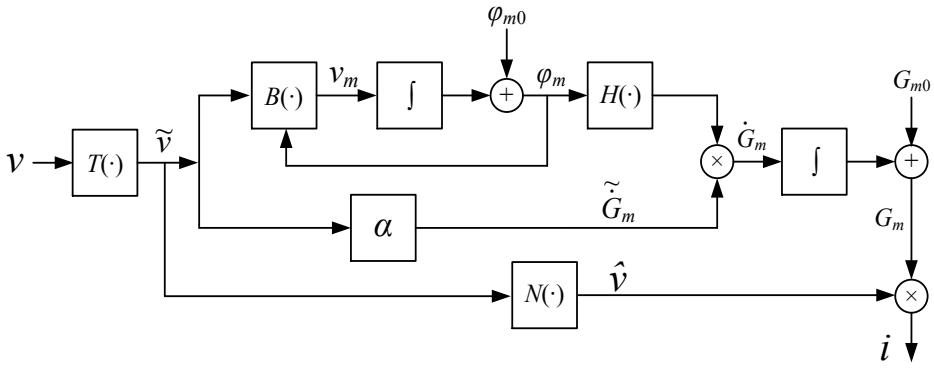


Figure 3.7 Modular model for memristors with threshold voltages and nonlinear *i*-*v* relationships.

As an example, voltage-actuated memristive devices with thresholds and nonlinear *i*-*v* characteristics are modeled by the schematic shown in Figure 3.7. Current-actuated memristive devices are modeled with the similar schematic except all the functions are defined in current/charge/memristance domains.

3.2.4 Memristor with device parameter variations

It is commonly observed that state-of-the-art memristive devices exhibit temporal (cycle to cycle) and spatial (device to device) variations. Temporal variations, which include voltage-dependent switching time, multiple-level resistance programing via compliance current or external series resistor [57] can be modeled by adopting nonlinear functions in $T(\cdot)$ and $N(\cdot)$ in Figure 3.7. Spatial variations are usually stochastic due to the nature of the fabrication process [22, 76, 77]. When a number of memristive devices are present on a chip, the value for the same parameter (e.g., HRS, LRS, v_{SET}, etc.) is likely to vary from device to device under certain distributions. Here we build a statistical feature into the model so that each device is instantiated with different parameter values but collectively they obey pre-defined distributions. For example, if 20% of variation on HRS (nominal value = 100kΩ) is observed from fabricated devices and suppose that accounts for 3 sigma of deviation from the mean value, a Gaussian distribution of μ_{HRS} = 100kΩ and σ_{HRS} = 6.7kΩ can be established to

statistically assign HRS values to all devices. The modeling of both temporal and spatial variations on memristive devices helps to analyze systems comprehensively in much longer time scale and with much larger device count. This is especially valuable in building high-density storage and computing systems with memristors when the fabrication of the devices is far from being mature.

3.3 Proposed modular compact memristor model

The schematics to model both voltage- and current-actuated memristive devices are shown in Figure 3.8. Given the input voltage (or current) of the device, the model is able to calculate the instantaneous current (or voltage) at any given time. The function of each consisting block is a nutshell is: $T(\cdot)$ models the thresholds for SET/RESET; $B(\cdot)$ sets the boundaries of the memristive flux (or charge), which determines the limited memristance switching range; $H(\cdot)$ is a window function that modulates the change rate of the state variable; $N(\cdot)$ is employed to capture the highly nonlinear *i*-*v* characteristic. The statistical feature discussed in Section 3.2.4 is not shown in Figure 3.8, but it is used in the initial setup before the simulation starts.

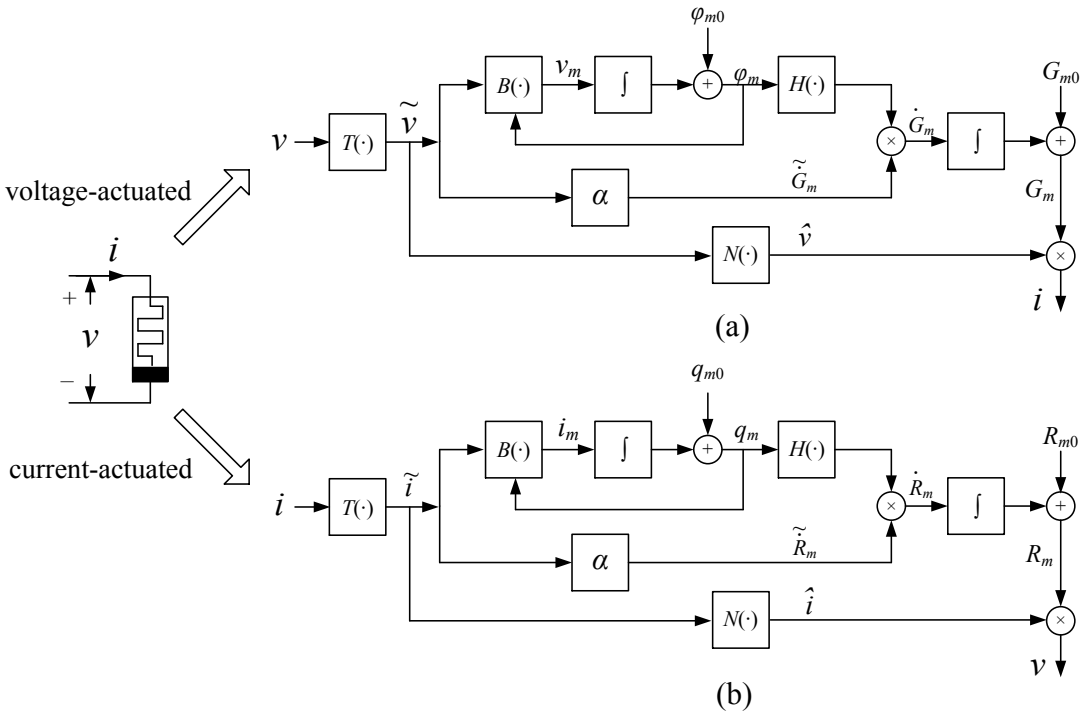
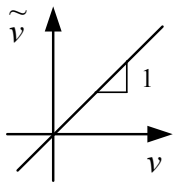
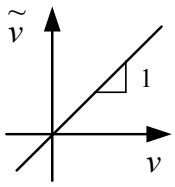
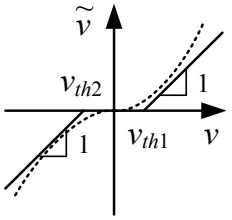
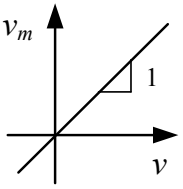
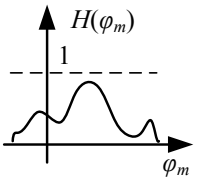
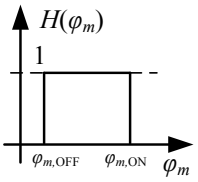
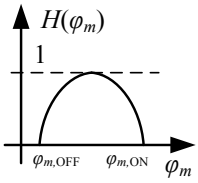
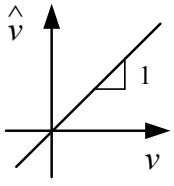
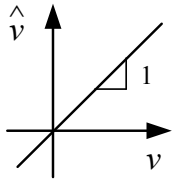
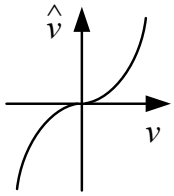


Figure 3.8 Proposed modular compact model for (a) voltage-actuated, and (b) current-actuated memristive devices.

The functions adopted by each block are illustrated in Table 3.1 over three types of voltage-actuated memristors. For theoretical memristors, $H(\cdot)$ needs to be a non-negative-valued function with a maximum of unity across the entire φ_m domain, while the other blocks have linear transfer functions with a unity slope. To model HP's memristor [1], all the blocks remain the same except $B(\cdot)$ adopts (3.13), and $H(\cdot)$ is defined within a bounded φ_m range. RRAM devices in [25] can be described by setting $T(\cdot)$ to the nonlinear transfer function shown in Figure 3.6 and using (3.20) in $N(\cdot)$.

Table 3.1 Summary on modules for different types of memristors.

	Theoretical Memristor	HP Memristor [1]	RRAM Device [25]
$T(\cdot)$			
$B(\cdot)$		(3.13)	(3.13)
$H(\cdot)$			
$N(\cdot)$			

3.4 Simulation results

The proposed compact model shown in Figure 3.8 has been implemented in SPICE-compatible environments such as PSpice and Verilog-A. Simulations are carried out on three types of memristors to demonstrate the capabilities and features of the proposed compact memristors model.

3.4.1 Theoretical memristor

A theoretical memristor is modeled by adopting unity-slope linear transfer functions in $T(\cdot)$, $B(\cdot)$, $N(\cdot)$ and setting $H(\cdot) = 1$, $\alpha = 3e{-}4$. As the memristor is applied

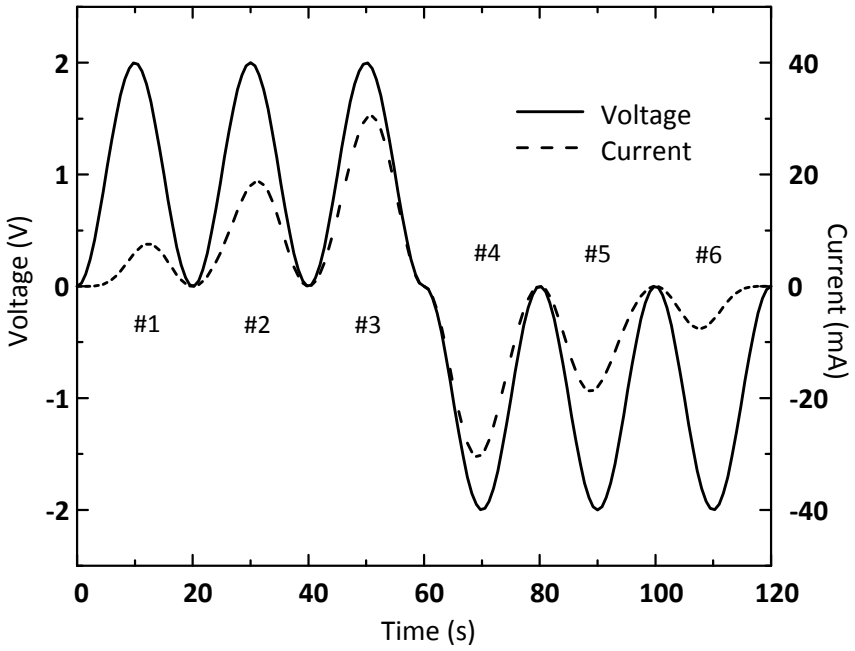


Figure 3.9 Unbalanced voltage waveform and corresponding current waveform of a theoretical memristor.

with a voltage source with unbalanced waveforms, the transient current and voltage waveforms of the device are shown in Figure 3.9. The voltage across the device is expressed as $v(t) = 1 + \sin(\omega_0 t + 3\pi/2)$ with $\omega_0 = 2\pi \cdot 0.05$ rad/s for the first three cycles. For the next three cycles, the same waveform repeats but with opposite polarity. The amplitude of current from cycle #1 to cycle #3 continuously increases, indicating the memductance (memristance) continuously increases (decreases) when the flux is increased by the positive voltage. As soon as the voltage negates its polarity, the memductance (memristance) decreases (increases), leading to smaller current amplitude for each cycle. The corresponding *i*-*v* characteristic of the ideal memristor is

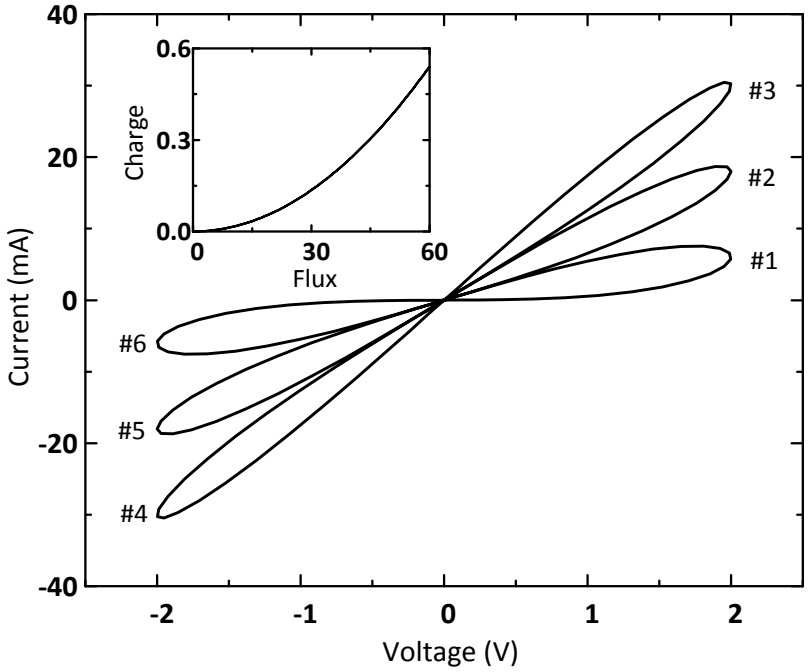


Figure 3.10 *i*-*v* characteristics of the device simulated in Figure 3.9. Inset: charge-flux relationship.

presented in Figure 3.10. The pinched hysteresis manifests the footprint of a memristor. The gradual change in memductance is clearly observed here from cycle #1 to cycle #6. From the charge-flux characteristic shown in the inset of Figure 3.10, the constitutive relationships are successfully obtained. Note that absolute values of charge and flux depend on the initial conditions of the simulation. It is the relative change in these quantities that conveys meaningful information about the device dynamics. For example, it takes 60 V·s of flux for the device to change its memductance from 10μS to 15mS.

3.4.2 Memristor with bounded memristance

The HP memristor presented in [1] is simulated. The memristance switching range is bounded between LRS = 1kΩ and HRS = 100kΩ by using (3.13) in $B(\cdot)$ to set boundary values in φ_m. The linear switching rate of the state variable is realized by adopting (3.14) and (3.15) in the window function $H(\cdot)$. All the other parameters remain the same as described in Section 3.4.1. The device is applied with an external voltage source $v(t) = \sin(\omega_0 t)$, where the frequency of the source has three options: 0.05Hz, 0.2Hz and 1Hz. The transient voltage and current waveforms under $\omega_0 = 2\pi \cdot 0.05$ rad/s are illustrated in Figure 3.11. Under the sinusoidal voltage stimulus, the irregular current waveform shows the device changes its memristance accordingly. Both the

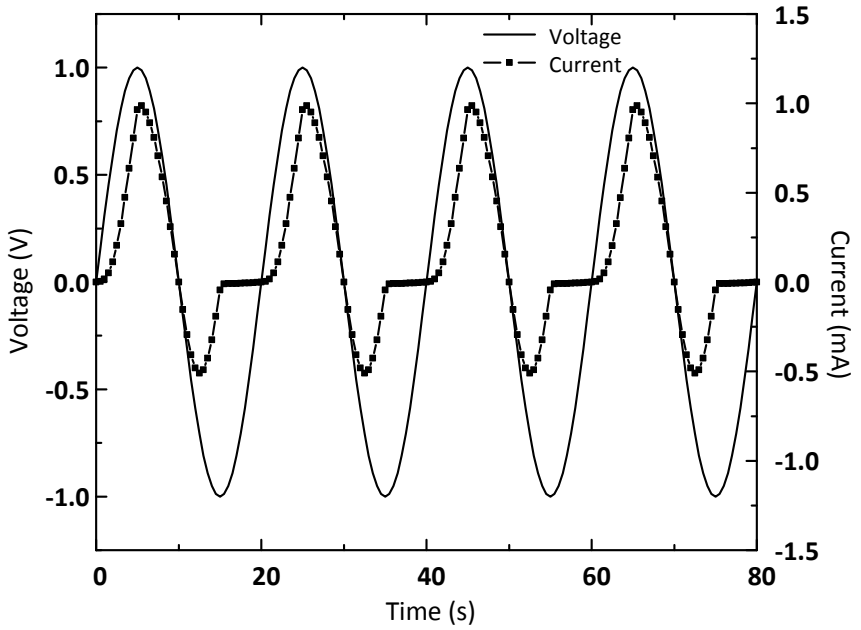


Figure 3.11 Sinusoidal voltage waveform and corresponding current waveform of a memristor with bounded memristance.

voltage and the current exhibit periodic behaviors over time, indicating that the device returns to its original state with a zero net flux input. The *i*-*v* characteristics under three different frequencies are presented in Figure 3.12. Under the frequency of 0.05Hz, the device performs hard switching (where both memristance boundaries are reached) without experiencing any backing problem. Soft switching (at least one memristance boundary is not reached) occurs when the frequency is increased to 0.2Hz and 1Hz. Note that the pinched hysteresis gradually collapses as frequency increases, indicating that the memristive device becomes a linear resistor at the limit of infinity frequency. Regardless of the operating frequency, a single trace of memristive charge-memristive flux constitutive relationship can be observed in the inset of Figure 3.12.

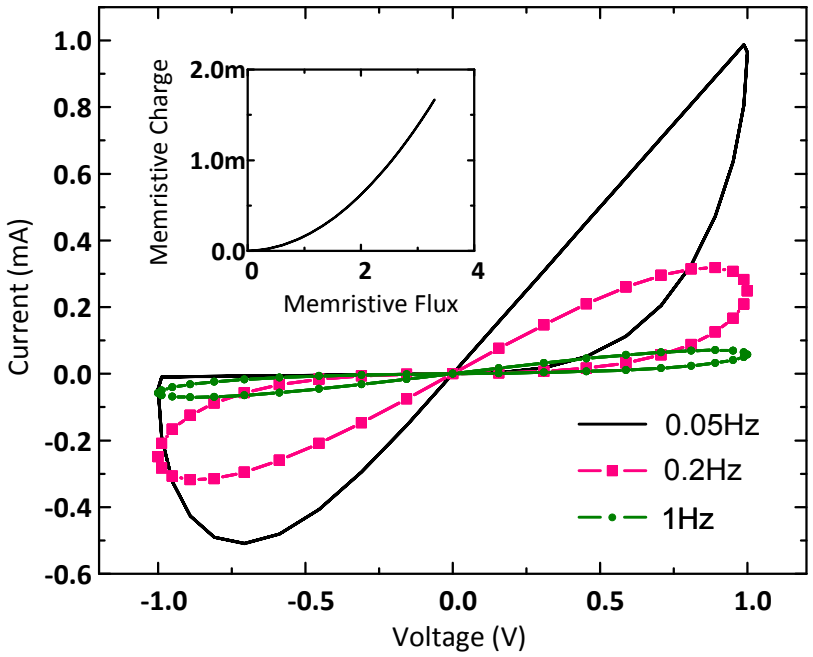


Figure 3.12 *i-v* characteristics of the device simulated in Figure 3.11. Inset: charge-flux relationship.

3.4.3 RRAM device

The threshold voltages for SET/RESET are set as +/− 0.5V by using (3.19) in $T(\cdot)$. The nonlinear switching rate of the state variable is realized by using (3.14) and (3.16) in $H(\cdot)$. All the other parameters are same as Section 3.4.2 except α = 5e−2. Device-to-device variations are included by assigning Gaussian distributions to LRS and HRS values with μ_{LRS} = 1kΩ, σ_{LRS} = 100Ω and μ_{HRS} = 100kΩ, σ_{HRS} = 40kΩ, respectively. A hundred random devices are applied with an external voltage source of $v(t) = \sin(\omega_0 t)$ with a frequency of 0.01Hz, and the *i*-*v* characteristics are overlapped in Figure 3.13. The memristance of each device is held unchanged when the voltage stays below +/−

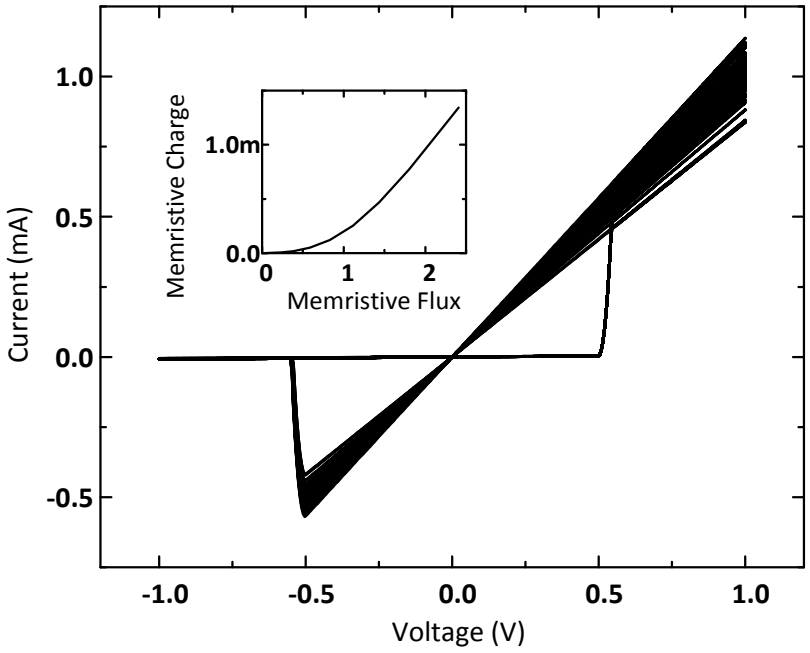


Figure 3.13 *i*-*v* characteristics of 100 randomly chosen RRAM devices with process variations on LRS and HRS values. Inset: the relationship between memristive-charge and memristive-flux of one of the 100 devices.

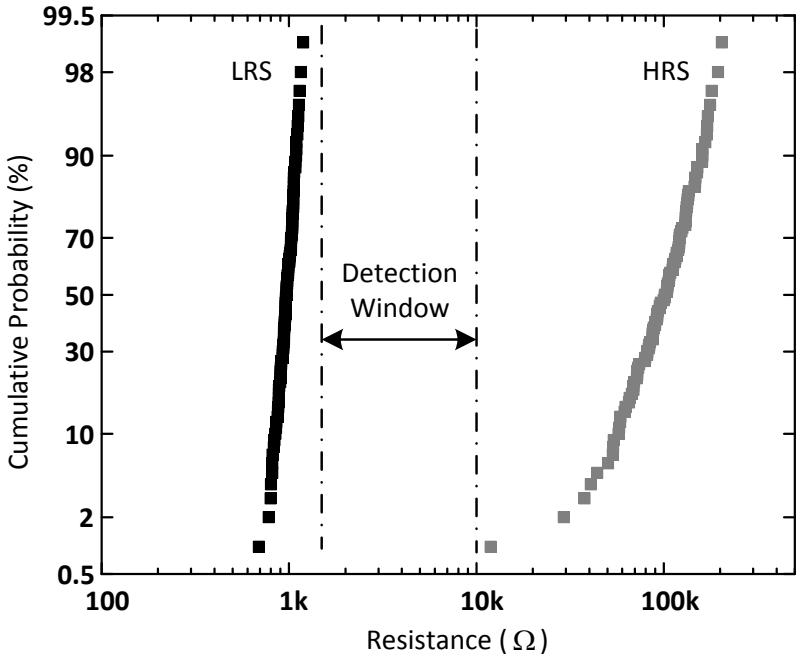


Figure 3.14 Cumulative probability of LRS and HRS values of the 100 RRAM devices in Figure 3.13.

0.5V. The device switches from HRS to LRS near +0.5V and switches from LRS to HRS near −0.5V. Note the actual switching voltage deviates slightly from the two threshold voltages due to the finite switching rate. The memristive charge-memristive flux relationship of one of the hundred devices is depicted in the inset of Figure 3.13. The cumulative probabilities of LRS and HRS values of the hundred devices are illustrated in Figure 3.14. The proposed model is able to assign different distributions to LRS and HRS. Here HRS exhibits larger variations compared with LRS, which is often observed from practical RRAM devices [22, 76, 77]. The study on the variations

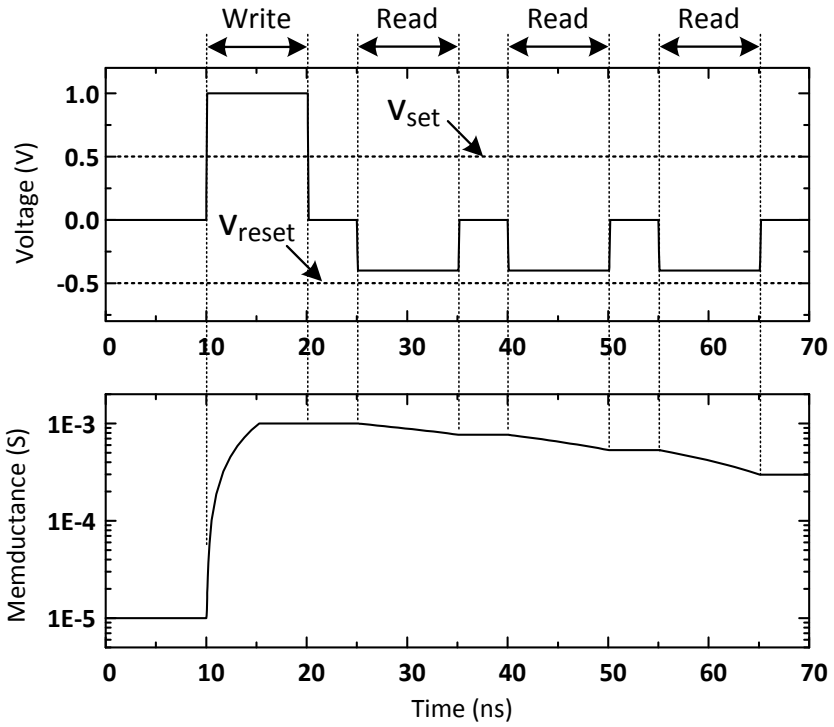


Figure 3.15 Transient voltage waveform and corresponding memductance waveforms of a RRAM device in the disruptive read process. $T(\cdot)$ adopts (3.20) with $A = B = 0.05$, $m = n = 1.5$, $v_{SET} = 0.5\text{V}$, $v_{RESET} = -0.5\text{V}$, $\alpha = 40\text{e}4$.

on LRS and HRS helps to identify the detection window labeled in Figure 3.14, which is a crucial parameter in designing and analyzing read schemes of RRAMs.

To model the disruptive read of the RRAM device, $T(\cdot)$ uses the nonlinear transfer function from (3.20). The applied voltage waveform and the corresponding memductance of the device are shown in Figure 3.15. The first positive voltage pulse (amplitude = 1V, duration = 10ns) sets the device from HRS to LRS in 5ns. Although the following negative voltage pulses (amplitude = 0.45V, duration = 10ns) are below the RESET threshold (−0.5V), the memductance slowly drifts away from LRS due to

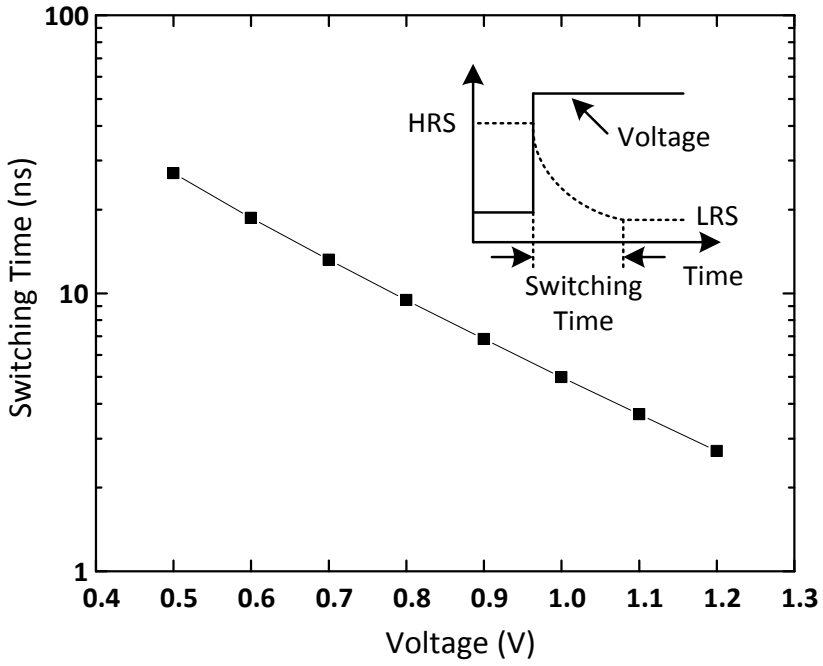


Figure 3.16 Voltage-dependent switching time of the device in Figure 3.15.

the non-zero output of $T(\cdot)$ when v is below thresholds. The shape of $T(\cdot)$ determines the number of consecutive read operations allowed before the device has to be refreshed to its original state. The voltage-dependent switching rate is drawn in Figure 3.16 where the device switches from HRS to LRS under a voltage step. The switching time is defined as the period from the onset of the voltage step to the completion of the switching. Due to the adopted nonlinear function in $T(\cdot)$, the switching time decreases exponentially as the applied voltage linearly increases from 0.5V to 1.2V, which is similar to the observations in [57].

A RRAM device with a nonlinear *i*-*v* relationship is modeled by setting $N(v)$ = $\sinh(5 \cdot v)$, α = 5e-5 and the other parameters remain the same as in Section 3.4.2. The device is driven by an external voltage source $v(t) = \sin(\omega_0 t)$ with a frequency of 0.01Hz. From the *i*-*v* characteristic shown in Figure 3.17, the hard switching is relatively difficult to identify since even at HRS or LRS, the slope of the curve keeps changing with the voltage. Comparing the current level at 0.5V and 1V, it can be observed that the device exhibits more than 5x of change in current given 2x of change in voltage. By tuning β to different values and using nonlinear functions in $T(\cdot)$, various degrees of nonlinearities can be achieved by the proposed model.

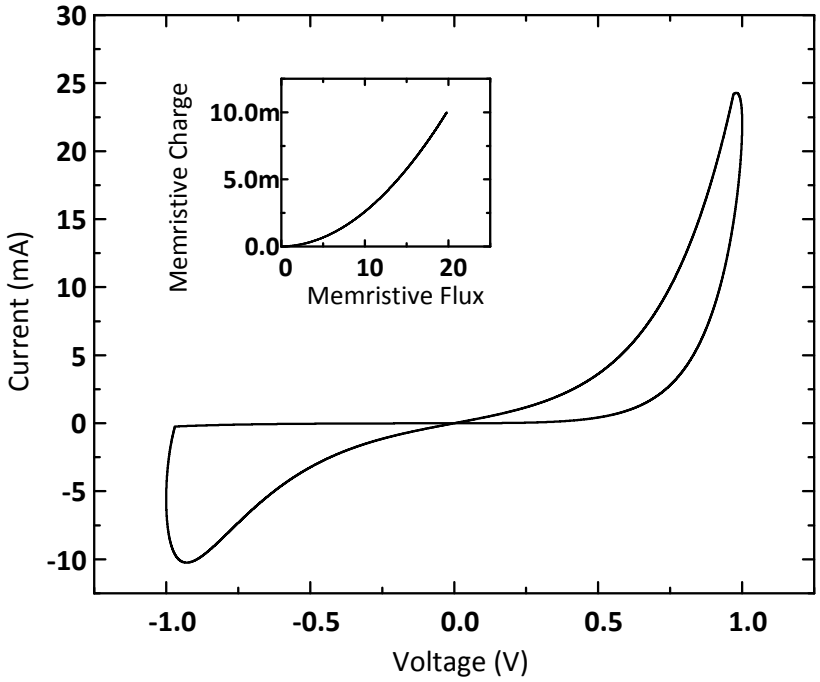


Figure 3.17 *i*-*v* characteristics of a RRAM device. Inset: the relationship between memristive-charge and memristive-flux.

3.5 Comparisons with existing memristor models

The proposed model is compared with existing memristor models in terms of the breadth of common device characteristics. As seen from the comparison result shown in Table 3.2, the proposed model covers a wide range of device behaviors including bounded memristance, threshold voltages for SET/RESET, nonlinear state variable change rate, nonlinear *i*-*v* characteristics, statistical variations on device parameters, etc. The backing problem is mitigated adopting a window function that is directly defined in the space of flux (or charge). The modular structure of the proposed model enables convenient reconfiguration of the model to reflect distinct types of memristors.

Table 3.2 Comparisons with existing memristor models.

Model	SET/RESET Voltages	Finite Resistance Range	Backing Problem	Switching Rate	Charge-Flux	Nonlinear Resistance Dynamics
HP [1]	No	No	N/A	Linear	Yes	No
Modified HP [1]	No	Yes	Exist	Both	Yes	No
Joglekar [3]	No	Yes	Exist	Both	Yes	No
Biolek [4]	No	Yes	Solved	Both	No	No
Shin [5]	No	Yes	Solved	Both	Yes	No
Yakopcic [6]	Yes	No	N/A	N/A	No	Yes
Pershin [7]	Yes	Yes	Solved	Linear	Yes	No
Corinto[8]	Yes	Yes	Solved	Linear	Yes	No
Proposed[9]	**Yes**	**Yes**	**Solved**	**Both**	**Yes**	**Yes**

3.6 Discussion

Although thermistors, Hodgkin-Huxley model of the neuron, discharge tubes can be categorized as memristive devices/systems [15], this work focuses on modeling voltage- and current-controlled memristors, and resistive memory devices widely referred as RRAM devices. As mentioned in Chapter 1, the devices of interest are envisioned to replace or complement current CMOS technology in circuits and systems to achieve new functionalities or better performance.

Ever since the first nanoscale memristor was discovered, numerous memristors have been fabricated and a broad range of device behaviors have been reported. Models based on detailed physical mechanisms generally offer more accurate presentations of device dynamics but they are restricted to specific types of devices. Models with more flexibility are also reported but either stability issues are present or they are less capable of capturing nonlinear device characteristics. The major contribution of this work is to establish a more generalized framework to model a wide range of memristors. With the model's SPICE compatibility and modular structure, the memristors are incorporated as a standard device library in circuit design environments and circuit designers will only need to determine the parameter values of the embedded modules to design around various types of memristors. Moreover, the stability issue observed from previous models is circumvented in this work. Finally the proposed model reveals that an

equivalent charge-flux constitutive relationship can always be obtained from memristive devices.

An interesting topic for the future research is to design an optimal strategy to determine the content of each module in the proposed model and associated parameter values, assuming both static and dynamic measurements of the target device are obtained, including: *i*-*v* characteristics under different sweep rates, switching time vs. stimulus amplitude, distributions of LRS and HRS, etc. Although each module of the proposed model is designated to independently reflect different aspects of device behavior, eventually an extensive multi-variable optimization process is needed. A custom optimization algorithm that achieves good matching between simulations and measurement results deserves more attention in developing future memristor models.

Chapter 4 CAM/TCAM for data-intensive applications

This chapter discusses fundamental technical aspects of the content addressable memory. First the motivation to investigate content addressable memory systems in the context of data-intensive applications will be discussed. Second general CAM systems will be explained in the levels of architectures, match line structures and bit cell topologies. Finally the tradeoff between power and speed in designing CAM systems will be discussed, leading to various match line sensing schemes, search line driving schemes and power-efficient CAM architectures.

4.1 Motivation

Data-intensive computing applications such as data mining, search engine, scientific computing, cloud storage, and computing, etc., pose stringent latency and power requirements on current computing infrastructures [79-81]. However CMOS-based computing systems continuously find themselves insufficient to meet the challenges despite aggressive scaling over the years [82-85]. One of the bottlenecks in current computing systems is the power-hungry and high-latency memory access that

has become a bigger obstacle in high-performance distributed applications where large amounts of data must be stored, retrieved, manipulated, and transferred [10, 11].

Because CAM implements associative lookups with high throughputs, it is promising to incorporate CAM systems into existing computing fabrics to improve the computation performance. For example, applications that require searching a large amount of data such as biological genome study, facial recognition, and search engine can benefit from using CAM systems without frequently resorting to expensive memory accesses. Moreover, it is also proposed that certain applications/algorithms can be mapped to associative lookups to take advantage of CAM systems to realize fast and energy-efficient computations.

4.2 CAM/TCAM systems

CAM is a special type of associative memory where the input data is compared against a table of stored data, and returns the address of the matched entry [48]. The architecture of a typical CAM is shown in Figure 4.1 where the memory contains an array of memory bit cells that are arranged in entries of words with a specific word width. In a search operation, the search word is broadcast onto the vertical searchlines (SLs) and compared with all the stored words simultaneously. Each stored word has a horizontal matchline (ML) to indicate the comparison result. At the end of the search,

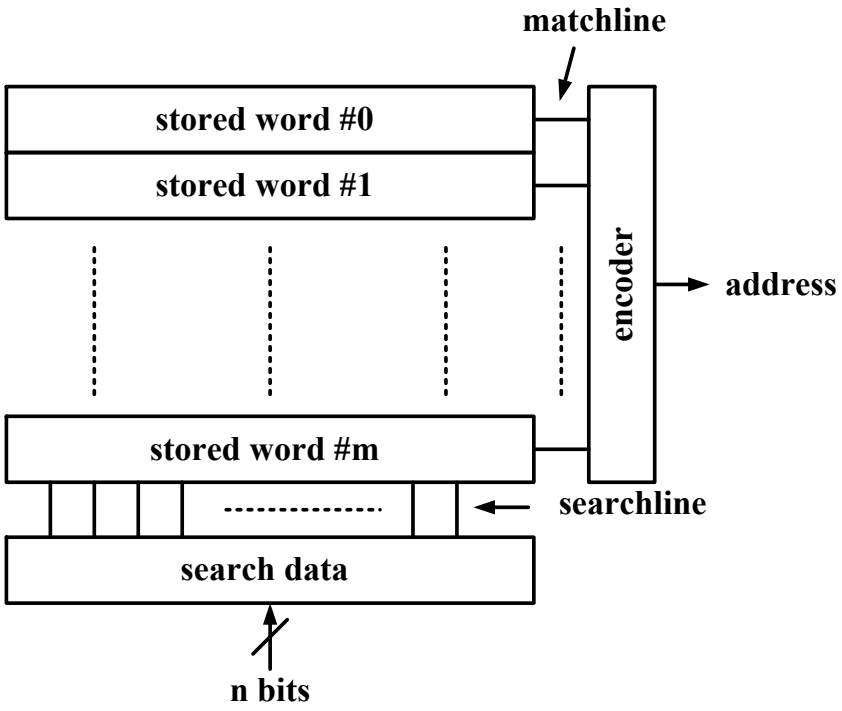


Figure 4.1 Conceptual architecture of a content addressable memory.

only the matchline of the match entry will change its voltage, which is interpreted by the encoder to generate the output address. Ternary content addressable memory (TCAM) is a special class of CAM that allows storing and searching a wildcard (X) in addition to binary data. As a wildcard matches zero, one, and itself, TCAM offers flexible search operations that are especially valuable in certain applications.

4.2.1 CAM cell structure

A CAM cell has two basic functions: bit storage and bit comparison. Generally SRAM is adopted as the basic storage element of each CAM cell. Additional circuitries

are added to program the stored data and allow bit comparison between the stored data and the search data. Depending on the relationship between the search result from each cell and ML of the entire word, two types of cell structures are discussed: NOR cell and NAND cell.

4.2.1.1 NOR cell

The schematic of a typical NOR cell is shown in Figure 4.2. The main storage element is a SRAM cell, which contains two inverters. In addition, four transistors are used as access transistors to implement bit comparison. Each pair of the transistors, M_1/M_3 and M_2/M_4, serves as a pull-down path from ML to the ground. During the search operation, the search data will be applied to two complementary SLs. The comparison result between the stored data D and the search data SL determines whether a pull-down path is enabled. For example, when D matches SL, both pull-down paths

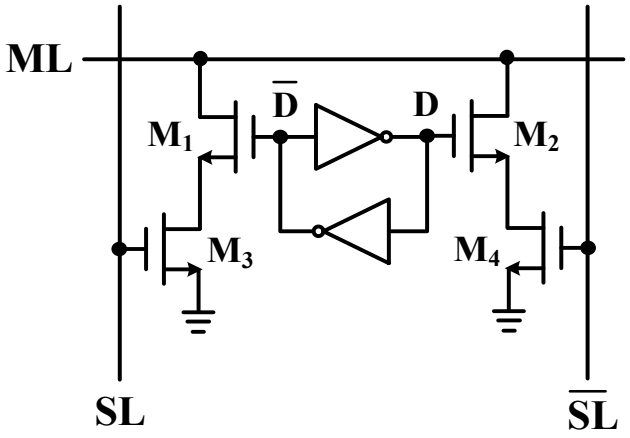


Figure 4.2 Schematic of a NOR CAM cell.

are disabled and ML retains its original voltage; whereas if D and SL are different, one of the pull-down path is enabled and discharge ML to ground. As cells are appended horizontally to form a word, ML will retain its state when all the bits of a word is matched with the search content otherwise a single mismatch will change ML's state.

4.2.1.2 NAND cell

The schematic of a NAND cell is shown in Figure 4.3. Similar to a NOR cell, a SRAM cell is used as the storage element. Three access transistors are included in the cell the implement the search operation. Depending on the search result, the voltage at node B controls M_1 to determine the connection between two neighboring MLs: ML_{n+1} and ML_n. Consider a match case where D = 0, SL = 0 (or $\overline{D}$ = 1, $\overline{SL}$ = 1), transistor M_2 is ON and passes $\overline{SL}$ = 1 to node B, which in turns ON transistor M_1, connecting ML_{n+1}

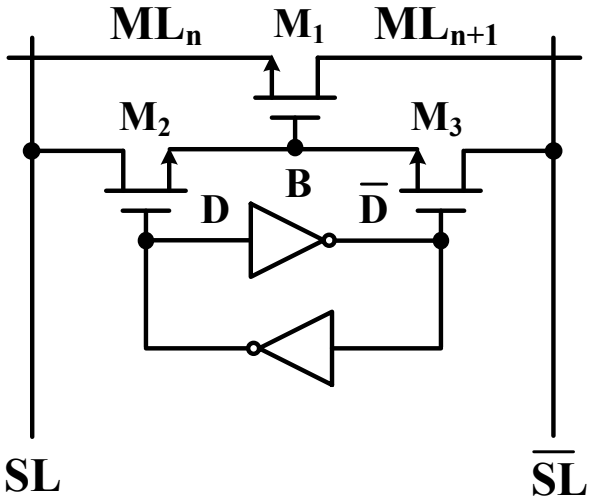


Figure 4.3 Schematic of a NAND CAM cell.

and ML_n. Note that ML_{n+1} and ML_n are connected as well in the other match case where D = 1, SL = 1 (or $\overline{D}$ = 0, $\overline{SL}$ = 0). A mismatch pulls the voltage at node B to ground, disabling M_1, and disconnecting the matchline. As the NAND cells are appended horizontally to form a word, each segment of the matchline is controlled by an individual bit. The operation principle of the NAND cell is such that only when all the bits in a word match the search content, a pull-down path is established to discharge the matchline, otherwise the state of the matchline is retained.

An important property of NOR cell is full rail voltage is available at every gate of access transistors. In contrast, a voltage headroom reduction occurs in NAND cell because the gate voltage of M_1 in Figure 4.3 is supplied by access transistor M_2 or M_3 thus can only be V_{DD}-V_{th}. As will be discussed later in this chapter, the reduction in gate voltage headroom results in diminished noise margin when sensing MLs.

4.2.2 TCAM cell structure

The CAM cells presented above store binary data: logic '0' or logic '1'. Ternary CAM cells, however, store an additional logic 'X', or can be referred as a 'wildcard'. The wildcard is a 'don't care' bit that matches 0, 1, or itself. Note that a wildcard can occur in the stored data or the search content. In either case, the wildcard generates a 'match' regardless of the input signal. The truth table of the ternary symbol

representation and the search results in different scenarios are presented in Table 4.1. Ternary data is encoded by two bits D, $\overline{D}$. Logic '0' or '1' is represented when the two bits store complementary information. Logic 'X' is represented when both of them store '0'. Note that the state of D = 1, $\overline{D}$ = 1 is not allowed in a ternary symbol.

Table 4.1 reveals that to store a ternary value, two storage bits are required. In other words, TCAM cells should contain two SRAM cells. NOR-type and NAND-type TCAM cells are illustrated in Figures 4.4 and 4.5 respectively. In a NOR-type TCAM cell, the pull-down path is controlled by a SRAM cell and a SL. When '0' or '1' is stored, the cell operates in the same way as the NOR-type CAM cell presented before. When 'X' is stored in the cell (D = 0, $\overline{D}$ = 0), SRAM cells present a 'low' voltage level at the gates of M_1 and M_2 respectively, thus both pull-down paths are disabled to retain the original state of ML. When the search content is 'X', both SLs are set to '0', thus

Table 4.1 Truth table of TCAM

Value	**Stored Bit**		**Search Bit**	
	D	**$\overline{D}$**	**SL**	**$\overline{SL}$**
0	**0**	**1**	**0**	**1**
1	**0**	**1**	**1**	**0**
X	**1**	**1**	**0**	**0**

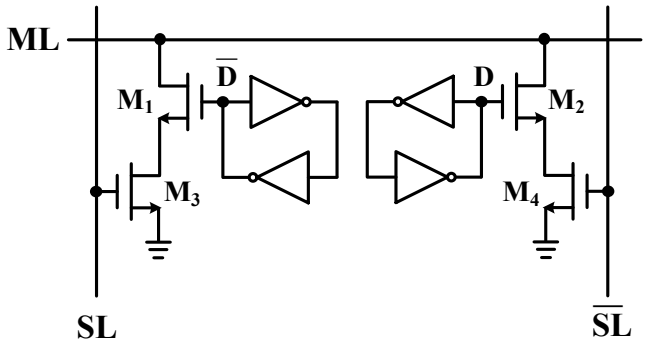


Figure 4.4 Schematic of a NOR-type TCAM cell.

M_3 and M_4 are turned OFF to disable both pull-down paths. The NOR-type TCAM cell thus supports both storing and searching ternary bits of 0, 1, and X. A NAND-type TCAM cell is modified from its CAM counterpart by adding another SRAM to represent the mask bit. When the stored mask bit is 0, M_{mask} is turned OFF, thus the cell operates as a CAM cell; if the mask bit is 1, M_{mask} is turned ON, discharging ML

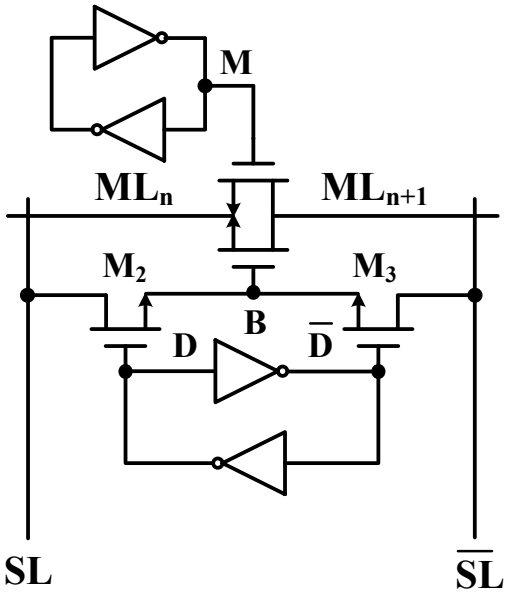


Figure 4.5 Schematic of a NAND-type TCAM cell.

to ground regardless of the states of D and $\overline{D}$ and the voltages on SLs. The TCAM cell also supports searching for X by setting both SLs to '1'. The truth table to encode storing and searching operations is listed in Table 4.1.

Further modifications on CAM/TCAM cells include reducing the number of transistors per cell [86, 87], mixing parts of NAND and NOR cells, alternating the logic level of the pull-down path in the NOR-type cell, among many others [88-90].

4.2.3 Matchline structure

The matchline is a key component in a CAM/TCAM system. Match/mismatch is represented by the final state of ML at the end of the search operation. Thus its arrangement directly affects the speed and energy consumption of the memory. As a result, many techniques have focused on designing matchline schemes that achieve better tradeoff between latency and power. This section briefly explains operating principles of two major matchline schemes: NOR-type and NAND type.

4.2.3.1 NOR-type matchline

A NOR-type ML is depicted in Figure 4.6, where individual NOR-type bit cells share a single ML and are connected to it in parallel. This scheme applied to both CAM and TCAM systems.

A search operation of a NOR-type ML has three phases: SL precharge, ML precharge, and evaluate. First, SLs are set to low to disable pull-down paths in each bit cell. Second, with ML disconnected to ground, precharge signal $\overline{pre}$ is asserted to charge ML to V_{DD} via transistor M_{pre}. Finally, $\overline{pre}$ is de-asserted and search content is broadcast to SLs to trigger the evaluation phase. In the case of a match, ML's voltage is retained as high since there is no discharge path to ground. In the case of a miss, at least one discharge path to ground is established between ML and ground. At the end of each search cycle, the voltage of ML is sensed by the matchline sense amplifier (SA) and match/miss is represented by the logic level at SA's output. One of the major advantages of NOR-type ML is its high-speed operation. In the worst case, ML is discharged through a path that contains two pull-down transistors. As will be analyzed below, this discharging path is significantly shorter than that in NAND-type ML schemes.

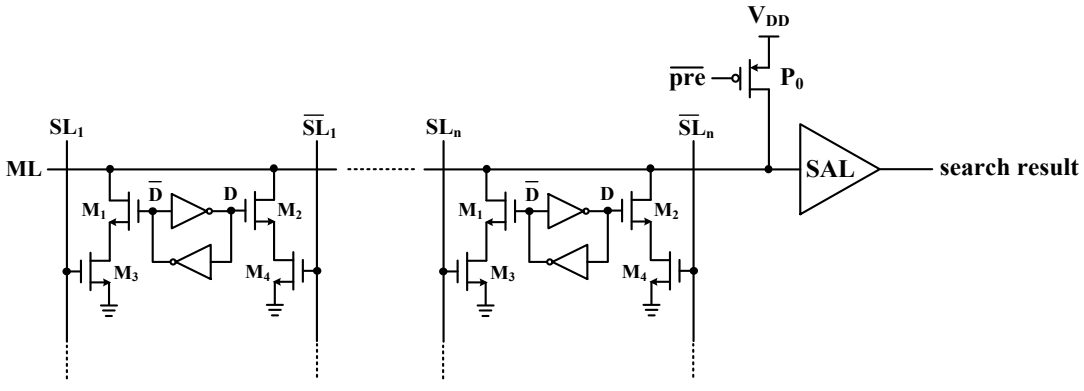


Figure 4.6 NOR-type matchline scheme.

4.2.3.2 NAND-type matchline

The schematic of a NAND-type matchline is drawn in Figure 4.7. As can be seen, the entire matchline is segmented into n+1 sections by n bit cells. In other words, all the bit cells in a stored word are cascaded to form a discharge path from node ML to ground. Similarly, the search operation has three phases: searchline precharge, matchline precharge and evaluate. First, all SLs are set to low to prevent all bit cells from connecting neighboring ML segments. Second, both the precharge signal $\overline{pre}$ and the evaluation signal eval is set to low so that the voltage at node ML is charged to V_{DD}. Finally, both $\overline{pre}$ and eval are asserted high, and ML's voltage is determined by the search result. In the case of a match, a discharge path is established between ML and ground since each bit cell connects its neighboring matchline segments. In the case of a miss, at least one bit cell disconnects matchline segments so that ML's voltage is retained since there is no discharge path.

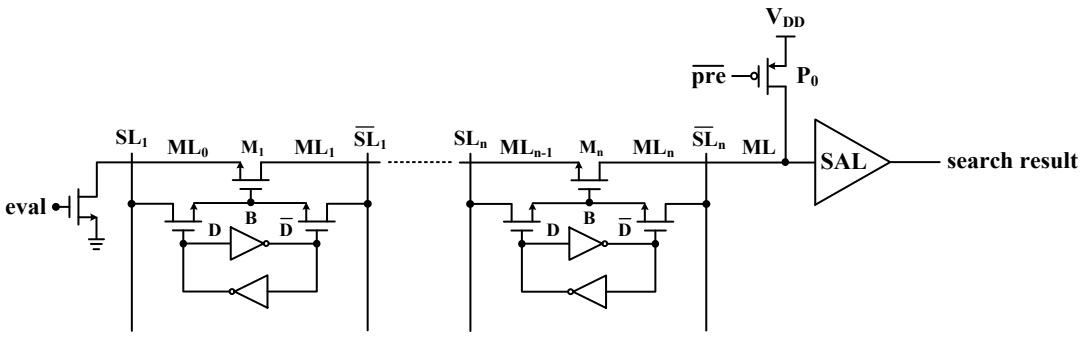


Figure 4.7 NAND-type matchline scheme.

A potential charge-sharing problem exists in NAND-type matchlines. Consider a case where every bit cell matches the search content except for the leftmost cell in Figure 4.7, during the evaluation phase, the charge at node ML will be shared with nodes from ML_{n-1} to ML_1, which may lower the voltage at ML so significantly that results a false detection at SA. One remedy to this issue is precharging all the intermediate nodes on the matchline to VDD during the phase of matchline precharge by setting all SLs to high to force all the transistors in the chain, i.e., M_1 to M_n, to turn ON. This eliminates the charge sharing since ML and all intermediate nodes are shorted initially. However, excessive power consumption is expected from charging additional capacitances.

NAND-type matchline scheme offers potential savings on power consumption since a single mismatch stops ML from discharging. Major drawbacks of NAND-type matchline scheme include a quadruple delay dependence on the number of bit cells per word and the diminished noise margin along the matchline. Each bit cell not only contributes a series resistance but also a parasitic capacitance to the signal chain. The RC product determines the time required to charge/discharge the matchline. The diminished noise margin is due to the fact that NMOS transistor can pass a voltage as high as the gate voltage deducted by V_{th} (where V_{th} is the threshold voltage of the device). For example, since the gate voltage of M_1-M_n is limited at V_{DD}-V_{th}, the highest voltage that is passed on the matchline is V_{DD}-$2V_{th}$. The reduction in the sensing

window causes it more difficult for SA to distinguish different states of ML. This issue becomes more severe when low supply voltage is used to reduce power consumption in CAM/TCAM systems.

4.3 Speed and power tradeoffs in CAM/TCAM systems

Although both NOR-type cell and NAND-type cell have their perspective advantages and shortcomings, NOR-type cell is more prevalent in current CAM systems due to its high speed and less susceptible to low power supply voltage [48]. Techniques have been developed in various levels of designs to significantly reduce the power consumption of NOR-type CAM systems. We thus focus our discussions in this section on NOR-type CAM systems where applicable.

4.3.1 Reduced matchline swing voltage

To reduce the power consumption on ML and potentially increase the sensing speed, the voltage swing on ML is reduced [91]. Since the dynamic power of ML is described as,

$$P_{ML} = m \cdot C_{ML} V_{DD} V_{swing} f \tag{4.1}$$

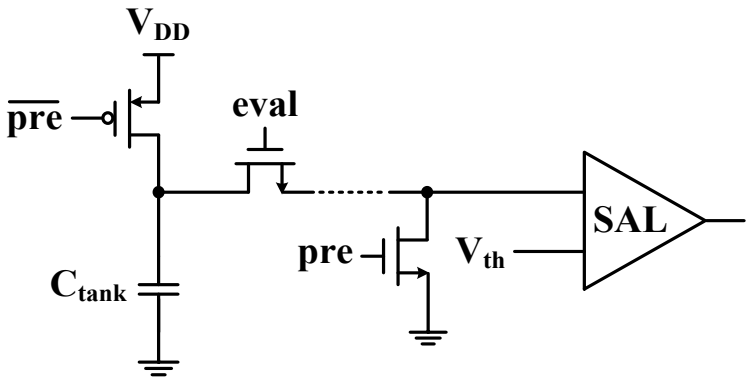


Figure 4.8 Low-swing matchline scheme.

where m is the number of MLs, C_{ML} is the total capacitance on a ML, V_{swing} is the amplitude of voltage swing occurred on ML and f is the operating frequency. The reduction of power consumption is thus linearly proportional to the voltage swing. A simplified circuit to reduce swing on ML is shown in Figure 4.8. First C_{tank} is charged to V_{DD} in precharge phase, and the charge is shared among all the cells on ML during the evaluation phase. As a result, the highest voltage on ML will be smaller than V_{DD}, depending on the ratio between C_{tank} and C_{ML}. Note that this scheme requires a lower reference voltage for SA, and also C_{tank} will occupy extra area for each ML. The sensing accuracy of the scheme is compromised by the uncertainty when estimating the exact value of C_{ML} even when post layout extraction tools are available.

4.3.2 Current-race sensing scheme

The schematic of the current-race scheme is illustrated in Figure 4.9 [92]. In the precharge phase, ML is discharged to low while the input to SA is charged to high. In the evaluation phase, pre is de-asserted and en is asserted. As the current source I_{ML} starts to charge ML, the final voltage of ML depends on the search result. In the case of a match, every cell presents a high impedance on ML so that ML will be charged to V_{DD}; in the case of a miss, ML's voltage will be $I_{ML} \times R_{ML}/k$, where R_{ML} is the equivalent resistance of an enabled pull-down path and k is the number of bit-wise misses in a word. By setting the maximum voltage of a miss to be small enough, SA can easily differentiates between a match and a miss. The SA shown in Figure 4.9 adopts a sense transistor to detect the voltage level of ML and a half-latch to retain its result.

The advantage of this scheme over the conventional precharge-high scheme is that the precharge-low scheme eliminates the need to separately precharge SLs to low as

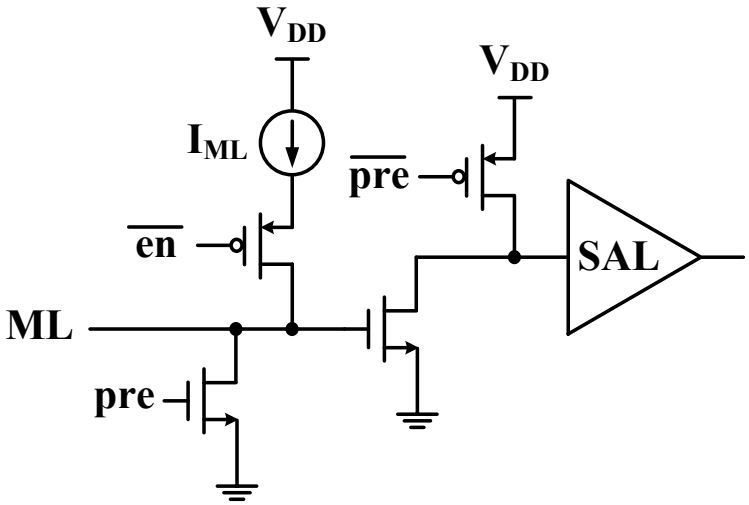


Figure 4.9 Current-racing scheme.

described in Section 4.2.3.1, thus saving power. One of the challenges is, however, to determine the values of I_{ML} and R_{ML} to yield an optimal sensing margin. Particularly the transistors in the pull-down path in Figure 4.9 should be large enough to have enough margin between R_{ON} and R_{OFF}, which increases the area of the bit cell.

4.3.3 Selective-precharge scheme

This scheme performs an initial search operation on the first few bits of a word before activating the search on the remaining bits [93]. For example, if a search is done on the first 2 bits of a word, assuming a uniform distribution of the data in the memory, only 1/4 of the entries will be searched again, saving about 75% of the power. In practice, the amount of power saving depends on the distribution uniformity of the data and the power consumed in the initial search stage.

A conceptual schematic of the selective-precharge scheme is shown in Figure 4.10. The example uses the first bit in the initial search and the remaining n-1 bits for the

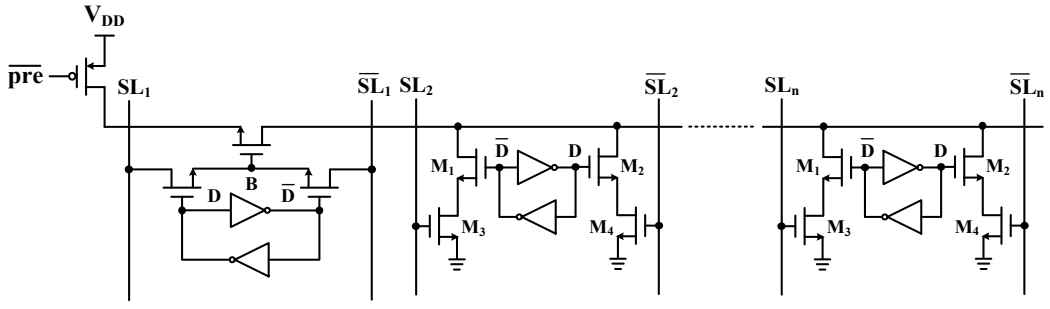


Figure 4.10 Selective-precharge scheme.

secondary search. NAND cell and NOR cell are used in initial search and secondary search respectively. The ML is precharged through M_1, which is controlled by the initial search result. If a match occurs at the first bit, M_1 is turned ON and the rest of the ML will be precharged to high and secondary search is carried out. Note that the ML of the NOR cells should be pre-discharged to low to allow proper operations.

Selective-precharge scheme is widely adopted among CAM systems [86, 94-98] since it reduces the dynamic power significantly and does not introduce excessive hardware complexity.

4.3.4 Pipeline scheme

The basic idea of pipeline scheme is from the selective-precharge scheme [99]. Instead of splitting the ML into two segments, the pipeline scheme divides the ML into multiple segments that offers more granularity to optimize the power saving. The concept of the pipeline scheme is shown in Figure 4.11 where the ML is comprised of four pipeline stages and NOR cells in each stage share the same local ML. The search of the word is carried out sequentially from the bit cells on the left to those on the right. A match from the previous stage enables the search in the succeeding stage; where a

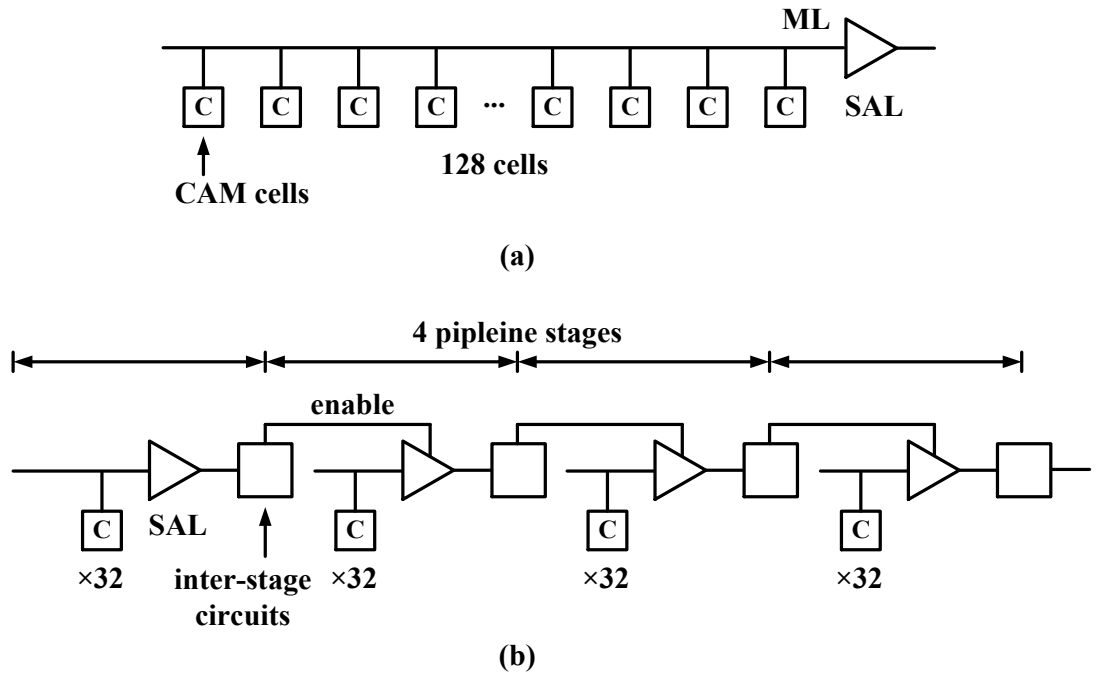


Figure 4.11 Conceptual diagram of conventional NOR-type matchline (a) and pipelined matchline scheme (b).

miss from one stage stops the entire search operation and generates the result at the end of the ML. Additional SAs and logic circuits are added in between stages to allow proper operations. The pipeline scheme saves power by shutting down the following stages when there is already a miss. The drawbacks of this scheme are the increased latency and area overhead from the inter-stage circuitries.

4.3.5 Power-efficient CAM architectures

The power-saving circuit techniques discussed so far have focused on local matchline structures. On a global level, there have been research activities on developing power-efficient CAM architectures.

The first architectural technique is the bank-selection scheme where only a subset of the CAM is active on any given cycle [100, 101]. The basic concept of bank-selection is drawn in Figure 4.12. The CAM is divided into multiple banks and extra data bits are introduced to determine which bank to activate. The example in Figure 4.12 shows that two bank-selection bits are added to partition the CAM into four banks. The bank-selection bits determine which bank is selected when storing or searching data. The decoder implements the bank-selection by providing enabling signals to each bank. The bank-selection scheme reduces the overall power consumption by 75% since only one fourth of the CAM is activated at any given time. The correct operation of the

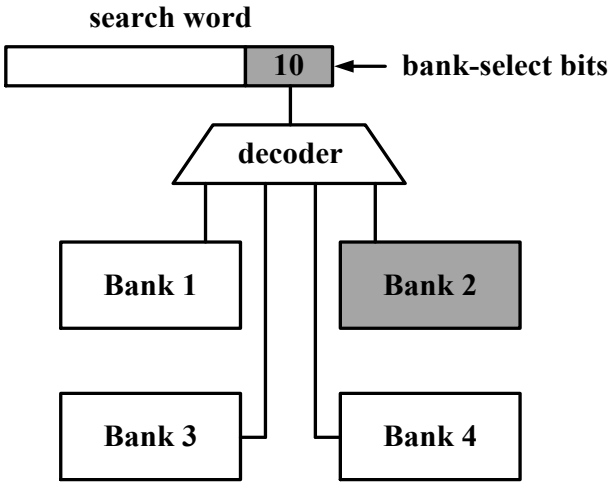


Figure 4.12 Bank-selection scheme where a CAM is divided into four banks.

bank-selection scheme relies on a pre-defined algorithm that determines the location to search depending on the search content.

The second technique is the pre-computation scheme, which is used in binary CAMs [102]. In this scheme, each entry has a feature that is computed based on the data stored in the entry. During the search phase, the feature of the search content is computed and then searched against the features of all the entries with in the CAM. If a match is found in the features, only those matched entries will be compared with the search content. Since the pre-computation already pre-selects the entries, only a fraction of the CAM is activated, thus saving overall power. The concept of the pre-computation scheme is presented in Figure 4.13. In addition to the main CAM, an extra smaller-sized CAM that stores the features of entries is included. The feature in this particular example is the number of '1's in the content. The search content of '10111'

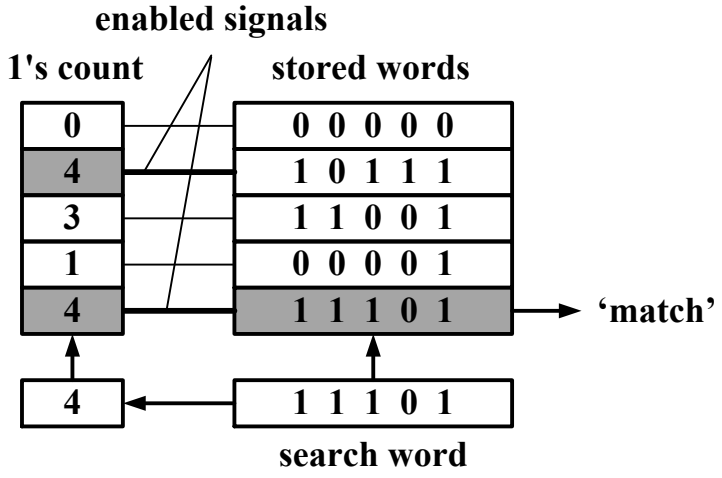


Figure 4.13 Pre-computation scheme where the computed feature is the number of '1's in the entries.

generates a feature of ‘4’, which is looked up in the CAM on the left. Two entries are returned with the matched feature and both are activated for the subsequent search operation. Similar to previously discussed power-saving techniques, pre-computation scheme is a type of ‘hierarchical search’ where a subset of data is searched first to narrow the scope of power-hungry searches. However pre-computation is unique in that it computes certain feature of the data based on pre-defined algorithm and utilizes the feature to reduce the scope of search. The saving on power comes at the cost of additional latency and area overheads from the additional ‘feature’ CAM and supporting logic circuits that implement the feature computation.

Chapter 5 Memristor-based TCAM

So far we discussed the motivation of incorporating CAM systems into existing computation fabrics for data-intensive applications. Important concepts and operating principles of CAM systems are introduced and popular techniques of building energy-efficient CAM systems are reviewed. As will be pointed out in this chapter, due to their limited storage density, conventional CMOS-based CAM systems have continuously found themselves insufficient in applications where a large amount of data is present. On the other hand, CAM systems that are based on emerging non-volatile memory (NVM) technologies are actively investigated to achieve superior storage density and competitive latency and power performance. After briefly reviewing existing NVM-based CAM systems, this chapter proposes a novel memristor-based TCAM (mTCAM) that intends to mitigate known issues from current NVM-based CAMs. The design of the mTCAM is presented and the operating principles along with design issues are discussed. The functionality and performance of the proposed mTCAM are demonstrated by simulation results.

5.1 Motivation

Ternary content addressable memory (TCAM) is a type of associative storage systems that implements high performance content lookups. Compared to a binary content addressable memory (CAM), the TCAM offers additional search flexibility by storing and searching for the wildcard, or 'X'. When associative lookups are implemented by TCAM, power-hungry and slow instruction processing, and data movement workloads that occur in conventional RAM-based computing systems are reduced or eliminated, greatly improving the energy-efficiency of the computations [49, 103]. Thus TCAM is an attractive alternative solution to curb both power dissipation and off-chip bandwidth demand from a wide range of data-applications.

Despite the promises and advantages of the TCAM, the usage of CMOS-based TCAMs have been largely limited in high-performance network routers due to the large area of the bit cell. For example, a typical SRAM-based TCAM cell consists of 14 transistors [104, 105]. Large cell area results low storage capacity and density, thus high cost-per-bit. Large cell area also increases the power consumption since it determines the length of ML and SL that are frequently charged and discharged during the operations. Compared with random access memories (e.g., DRAM and SRAM), state-of-the-art CMOS-based TCAMs have much higher cost-per-bit and the typical storage capacity is limited to tens of Mb, which prevents the wide usage of TCAMs [50].

Various cell structures have been proposed to reduce the TCAM cell size. A 12T SRAM-based TCAM saves cell area by eliminating access and driver transistors, but the net power saving is compromised by the hardware overheads [92]. A DRAM-based TCAM has been proposed to reduce the cell to 6 transistors and 2 capacitors, but hardware complexity to improve the sense margin and handle the frequent refresh could outweigh the benefits [106]. Recently emerging non-volatile memory devices have become promising candidates to build next-generation CAM systems with low cost and high storage density. Different cell structures have been proposed using NVM devices such as phase change memory, spin transfer torque magnetoresistive memory, memristor, etc. While basic functionalities have been demonstrated, drawbacks are observed from existing NVM-based TCAMs. First, the resistive device directly connects ML to GND via an access transistor [49, 103, 107-110]. By detecting the current (or the impedance) between ML and GND, 'match' and 'miss' can be discerned. This method not only presents large parasitics on ML, but the sensing margin also diminishes when the word width increases [49, 103]. Second, memristors are used as switches to control the gates of transistors [111, 112]. The sensing sensitivity in this case is compromised since the gate always presents a high impedance so that different resistance states of memristors cannot be reliably distinguished. Third, the previously reported cell structure contains a large number of transistors [107, 113], which prevents

a TCAM from achieving high storage density. Finally, the method of writing data to the TCAM cell has not yet been described in details [49, 113].

We present a novel memristors-based TCAM (mTCAM) that allows both storing and searching for zero, one, and the wildcard with energy and area efficiencies. Each mTCAM cell contains five transistors and two memristors, i.e., 5T2M. The memristors in the cell can be programmed individually so that a high impedance is always present between searchlines, greatly reducing the static current during write and search modes. A novel two-step write scheme has been proposed to guarantee the programming of the cell regardless of its initial memory state. Practical design issues such as voltage compliances to ensure reliable write and search operations, parameter-dependent sensing margin and device variations have been analyzed in details. Simulations on mTCAM arrays demonstrate functionalities as well as performance such as search latency and energy consumption.

5.2 mTCAM cell structure

The proposed mTCAM cell uses two memristors to store the ternary information. The data definition of each memristor and the cell is summarized in Table 5.1. The data stored by a memristor is represented by its memristance R_M, i.e., D = 0 when R_M = HRS and D = 1 when R_M = LRS. The content of an mTCAM cell is represented by

Table 5.1 Data definition of the mTCAM cell.

C	D	$\overline{D}$	SL	$\overline{SL}$	Result
0	0	1	0	1	Match
0	0	1	1	0	Miss
1	1	0	1	0	Match
1	1	0	0	1	Miss
X	0	0	Any	Any	Match
Any	Any	Any	0	0	Match

different combinations of states of the two memristors. The cell stores 0 or 1 when the two memristors store the opposite data. A wildcard is represented by storing 0 to both memristors. It is important to define the data in such way that at least one HRS state is present for any data pattern. As will be shown later, this arrangement significantly reduces the amount of direct current flowing across SLs during write and search modes.

The schematics of the 5T2M mTCAM cell during write and search modes are shown in Figure 5.1, with irrelevant devices greyed-out. Memristors M_1 and M_2 store the data D and $\overline{D}$ respectively. While M_1 and M_2 are connected together at their positive ports, their negative ports are driven by complementary searchlines SL and through access transistors T_1 and T_2. The voltage level on WL enables write/search operations for the cell by controlling the gates of T_1 and T_2. Two additional searchlines SX and SX' are introduced to overdrive the voltage at the common node G by controlling T_3

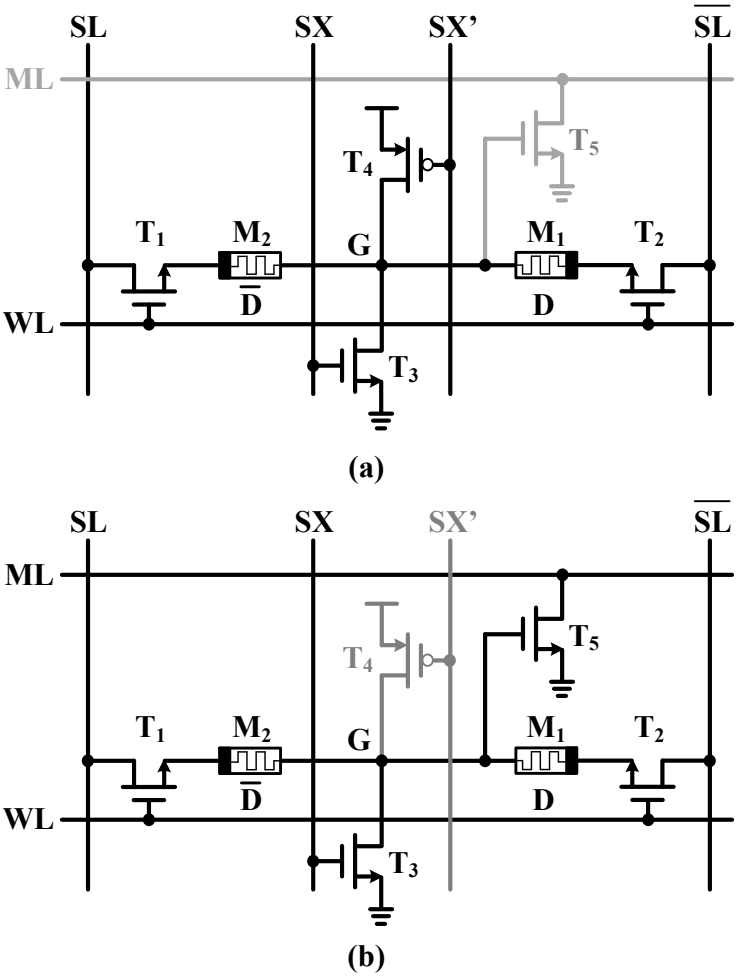


Figure 5.1 Schematic of the mTCAM cell in write mode (a) and search mode (b).

and T_4. Note that V_G is an important voltage within the cell since it determines the discharging of ML by controlling the pull-down transistor T_5. The write sequence starts by discharging ML to GND via an external switch (not shown in Figure 5.1) to avoid excessive current flowing through T_5. Then T_1 and T_2 are turned ON by setting WL to high. To write data into the cell, SL, $\overline{SL}$, SX and SX' are controlled such that suitable voltages are dropped across M_1 and M_2 to program them to the desired states. The

negative ports of M_1 and M_2 are set by SL and $\overline{SL}$ respectively. The positive nodes of M_1 and M_2 are set by SX and SX'. For example, setting SL, SX, SX' and $\overline{SL}$ to GND, high, GND and high respectively results zero and negative voltage being supplied on M_1 and M_2 respectively, which maintains M_1's state, and reset M_2 to HRS.

It is beneficial that each memristor in the mTCAM cell can be individually programmed. First, the state of (M_1 = HRS, M_2 = HRS) can be reliably achieved so that the direct current flowing across SLs is suppressed in the search mode regardless of the data pattern. On the contrary, in complementary resistive switches (CRS) [114], (M_1 = HRS, M_2 = HRS) is not available as a stable state. If the intermediate state of (LRS, LRS) is adopted instead to represent X, the direct current between SLs would be increased by the factor of HRS/LRS (usually $\approx$100) times, which prevents building high-capacity TCAM arrays. Second, the maximum voltage compliance required to program the cell is on the order of $|V_{SET}|$ or $|V_{RESET}|$, whereas in CRS, at least $2\times$ $|V_{RESET}|$ is needed. The reduction in the voltage compliance helps to save power consumption during the write mode.

Table 5.2 Two-step write scheme.

Data	D $\overline{D}$	Step 1	Step 2
C = 0	0 1	Reset D; Hold $\overline{D}$	Hold D; Set $\overline{D}$
C = 1	1 0	Set D; Hold $\overline{D}$	Hold D; Reset $\overline{D}$
C = X	0 0	Reset D; Hold $\overline{D}$	Hold D; Reset $\overline{D}$

A novel two-step write scheme is described in Table 5.2 where in each step only one memristor is programmed and the other one is isolated. Suppose C = 0 (D = 0, $\overline{D}$ = 1) is to be stored in the cell. In the first step, a negative voltage is applied to M_1 to reset it to HRS and M_2 holds its state with zero voltage drop. In the second step, M_1 maintains its state and a positive voltage is applied to M_2 to set it to LRS. The proposed write scheme programs the cell in only two clock cycles regardless of the cell's initial content. This is especially favorable in the power-on-reset of the memory where the previously stored data is usually unknown. Assuming the voltage drops on T_1 and T_2 are negligible, the maximum voltage applied on M_1 or M_2, V_{write}, should satisfy the following requirement,

$$V_{write} > |V_{SET}| \; and \; |V_{RESET}| \tag{5.1}$$

In the search mode, ML is precharged to high via an external switch and is driven by the cell depending on the search result. SX' is asserted high to deactivate T_4.

To search for X, SX is set to high so that T_3 is turned ON and V_G is tied to GND. As a result, T_5 is turned OFF and ML retains its voltage, indicating a 'match' is obtained regardless of the cell's content. To eliminate the direct current between searchlines, SL and $\overline{SL}$ are set to GND.

To search for 0 (or 1), SL and $\overline{SL}$ are set to GND (or high) and high (or GND) and SX is set to GND to turn OFF T_3. With both T_3 and T_4 deactivated, V_G is a result of a resistor divider which should turn ON/OFF T_5 depending on the search result. Thus the voltage different between SL and $\overline{SL}$, i.e., V_{search} should be analyzed as follows to guarantee the correct operation of the cell. Suppose the cell stores C = 0 (D = 0, $\overline{D}$ = 1). If the search data is 0, a 'match' is achieved and the following should be met,

$$V_G = V_{search} \cdot LRS/(LRS + HRS) \tag{5.2}$$

$$V_G < V_{th} \tag{5.3}$$

where V_{th} is the threshold voltage of T_5. Equation (5.2) describes the resistor divider and (5.3) states that V_G should be low enough to turn OFF T_5. If the search data is 1, a 'miss' is detected, as,

$$V_G = V_{search} \cdot HRS/(LRS + HRS) \tag{5.4}$$

$$V_G > V_{th} \tag{5.5}$$

Equation (5.4) describes the resistor divider and (5.5) states that V_G should be high enough to turn ON T_5 to discharge ML. Suppose the cell stores C = X (D = 0, $\overline{D}$ = 0), a 'match' should always occur when the search data is either 0 or 1. That is,

$$V_G = V_{search} \cdot \frac{HRS}{HRS + HRS} = V_{search}/2 \tag{5.6}$$

$$V_G < V_{th} \tag{5.7}$$

(5.6) describes the resistor divider and (5.7) states V_G should be low enough to turn OFF T_5. The analysis for V_{search} when the stored content is 1 is similar to (5.2) - (5.5).

Given the existing technology, it is realistic to assume the memristor exhibits a high HRS/LRS ratio with values larger than 10. Thus approximations can be safely applied to (5.2), (5.4) and the above analysis can be summarized as,

$$0 < V_{th} \tag{5.8}$$

$$V_{search} > V_{th} \tag{5.9}$$

$$\frac{V_{search}}{2} < V_{th} \tag{5.10}$$

where (5.8), (5.9) and (5.10) describe the conditions of 'exact match', 'miss' and 'wildcard match' respectively. Equation (5.8) is trivial. The amplitude of V_{search} determines the sensing margin of (5.9) and (5.10). To ensure operating conditions for

all three cases, the optimum amplitude of V_{search} is defined such that (5.9) and (5.10) have the same sensing margin, i.e.,

$$V_{search} - V_{th} = V_{th} - V_{search}/2 \tag{5.11}$$

As a result, the optimum V_{search} is obtained as,

$$V_{search} = 1.33 \cdot V_{th} \tag{5.12}$$

In addition, to avoid possible accidental change of the stored cell content, additional requirement on V_{search} is applied as follows,

$$V_{search} < |V_{SET}| \; or \; |V_{RESET}| \tag{5.13}$$

We define the sensing window as the difference between the maximum and minimum values of V_G during all search scenarios, i.e.,

$$\text{Sensing Window} = V_{search} \cdot \frac{HRS - LRS}{HRS + LRS} \tag{5.14}$$

and the normalized sensing window as,

$$\text{Normalized Sensing Window} = \frac{HRS - LRS}{HRS + LRS} \tag{5.15}$$

As can be seen from (5.15), the normalized sense window is only dependent on HRS, LRS values and is thus subject to variations of actual fabricated devices. With device variations accounted in our compact model [12], simulations are carried out to

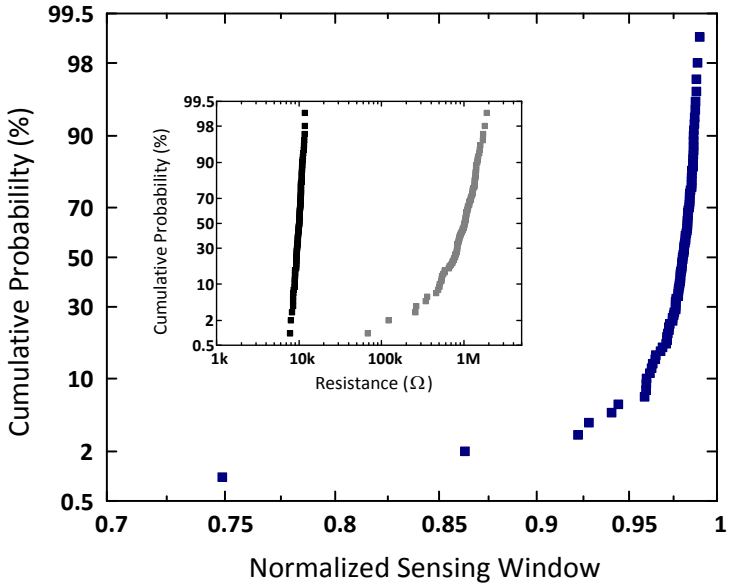


Figure 5.2 Distributions of the normalized sensing window.

study the statistical impact of the sensing window. Assuming both LRS and HRS experience Gaussian distributions [22, 57, 76, 77], e.g., μ_{LRS} = 10kΩ, σ_{LRS} = 1kΩ, μ_{HRS} = 1MΩ, σ_{HRS} = 400kΩ, the cumulative probabilities of the normalized sensing window and LRS, HRS are shown in Figure 5.2 and its inset respectively. It can be seen that most of the distributions of the normalized sensing window are located between 0.95 and 1. In other words, the sensing window can be properly approximated as V_{search} in designing the mTCAM cells.

5.3 mTCAM matchline structure

The proposed mTCAM adopts a NOR-type ML structure since it offers high speed operations due to the single-transistor pull-down path in the worst case [48]. The detailed ML scheme is shown in Figure 5.3 where all the cells of a word are connected to ML in parallel. Transistors P_0 and N_0 are added to precharge and discharge ML during different stages of the operation. The sense-and-latch (SAL) is placed at the end of ML to sense its voltage and determine the search result.

The write mode begins with asserting write_en so that ML is tied to ground to avoid excessive current flowing through T_5 in each cell. With ML grounded, WL is asserted to enable the entire row. The write data is then placed on SLs according to the two-step write scheme. Since the entire row shares the same WL, it is possible all the cells are programmed simultaneously.

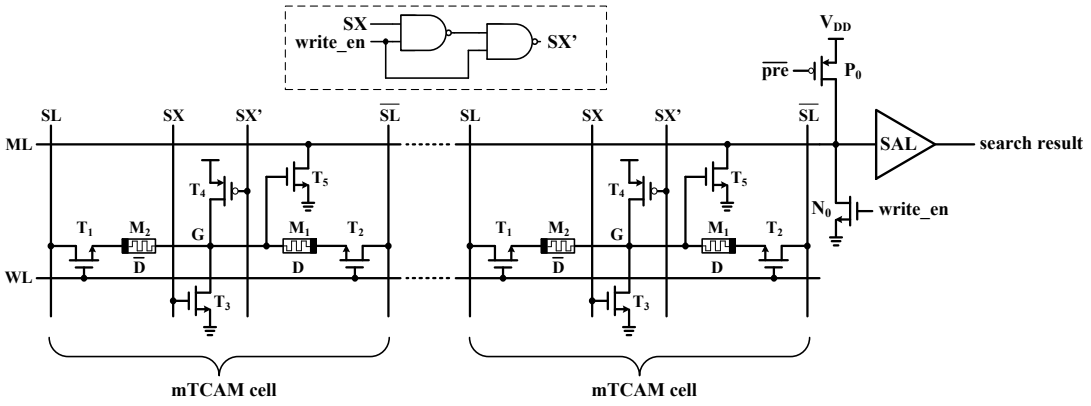


Figure 5.3 NOR-type matchline of the mTCAM.

A typical search cycle operates in two phases: ML precharge and ML evaluation. The operation starts by asserting $\overline{pre}$ to charge ML to high via P_0. Then $\overline{pre}$ is de-asserted and the ML evaluation phase begins by placing the search word on SLs. If at least one bit is missed, a path to GND through T_5 discharges ML, representing a 'miss' for the entire word. If all the bits are matched, ML maintains its potential, indicating a 'match'. At the end of the search cycle, the voltage on ML is processed by SAL to generate the search result.

The circuit shown in dash lines in Figure 5.3 is used to generate signals on SX and SX' such that in the write mode, SX' = $\overline{SX}$ and in the search mode, SX' = V_{DD}. Since this circuit is required for every column, it should be included in the searchline driver circuitry, as will be discussed in the next Section.

The schematic of SAL is shown in Figure 5.4(a). The inverter and N_2 form a half-latch where the input to the latch is controlled by P_1 and N_1. P_1 is used to sense ML voltage whereas N_1 is used to reset SAL. The operation of SAL starts by asserting SAL_reset to reset SAL's output, which is kept by the half-latch. If the search result is 'match', ML stays high and SAL's output remains high; whereas a 'miss' will discharge ML to low such that SAL's output switches to low. Note that P_1, N_1, and N_2 are carefully sized to guarantee correct operations of SAL. The timing diagram of the signals 'pre' and 'SAL_reset' are shown in Figure 5.4(b). It is important to have longer

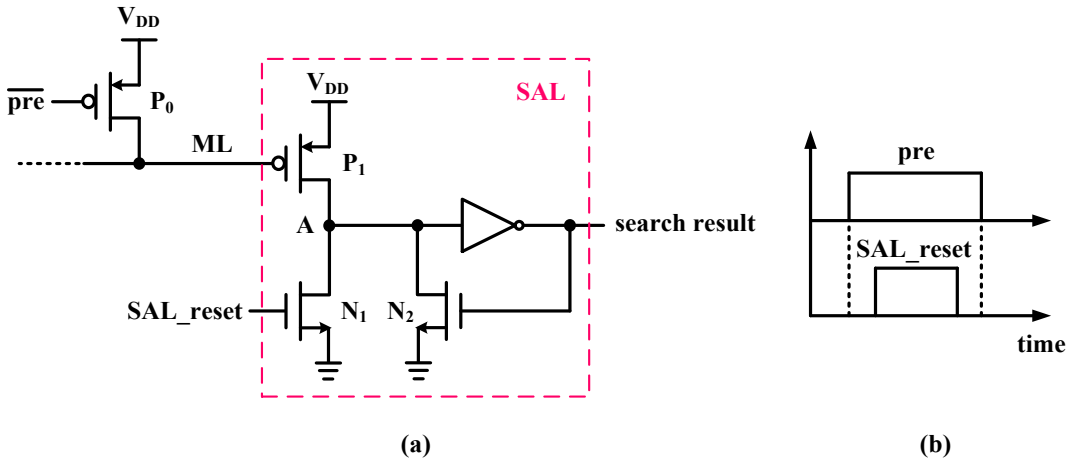


Figure 5.4 Schematic of the matchline sense amplifier and the required timing diagram.

duration on 'pre' than 'SAL_reset' such that ML is exempted from potential disturbances when node A is reset to low by N_1.

5.4 mTCAM array architecture

The schematic of an m×n mTCAM macro is shown in Figure 5.5. The macro consists of the core storage, the searchline drivers, the wordline drivers, the sense-and-latches, the local memory controller, and the encoder. The core storage area contains m words with a word width of n bits. The cells are placed a grid with horizontal MLs, WLs and vertical SLs. The WLs are driven by wordline drivers and the SLs are controlled by write/search drivers. The operations of the horizontal and vertical drivers are controlled by the local memory controller which takes the command from the data bus. At the end of each ML, SAL presents the search result to the encoder which

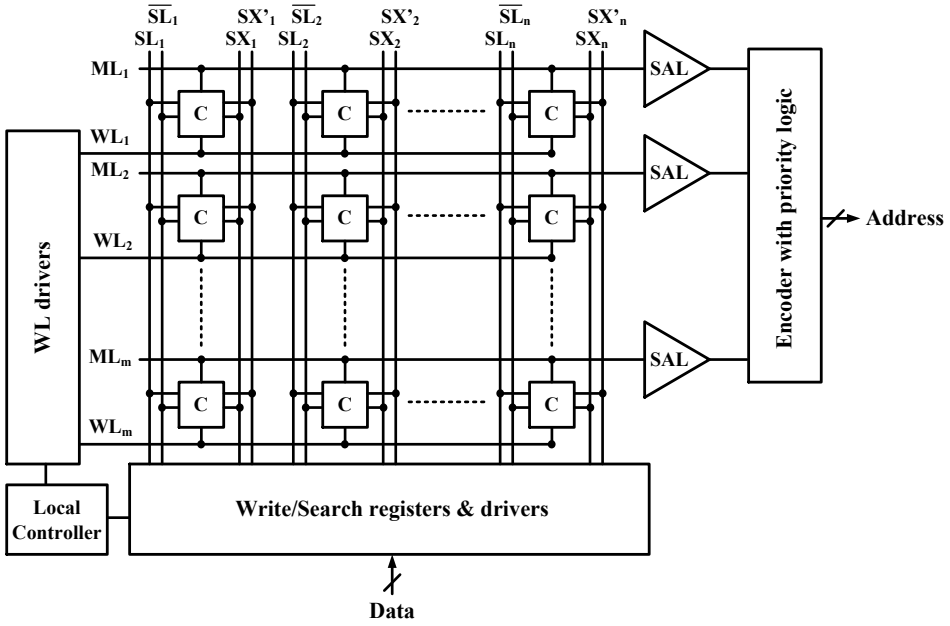


Figure 5.5 Array architecture of the mTCAM.

produces the corresponding address of the matched entry. Since the mTCAM array may contain zero, one or multiple matched entries, the encoder is built with priority logic that calculates the correct address.

5.5 Latency and energy modeling

Search latency and energy consumption are the two most important performance figures of a CAM system. It is beneficial to develop energy and latency models of the mTCAM such that: 1) performance predictions of mTCAMs with different sizes are possible without actual implementations; 2) performance projections to different CMOS technology nodes are possible; 3) the mTCAM can be extracted as a memory

module with latency/energy specifications and can be incorporated in high-level computer architecture simulators to investigate the impact of a CAM system in the context of data-intensive applications.

5.5.1 Latency

The total search latency τ_S is modeled as follows,

$$\tau_S = \tau_{SL} + \tau_{bit} + \tau_{ML} \tag{5.16}$$

where τ_{SL}, τ_{bit} and τ_{ML} are latencies from the searchline, the bit cell, and the matchline respectively.

5.5.1.1 Searchline latency

The search line latency τ_{SL} is defined as the time delay from the searchline driver's output to the farthest bit cell's input. As can be imagined, τ_{SL} is jointly determined by the driving capability of the driver and the total parasitic capacitance on the searchline. The minimum searchline latency is achieved by systematically designing of the searchline driver. Greater search depths result longer latencies due to the increase of the total parasitic capacitance.

Simulations have shown that the searchline latency is dominated by the gate delay of the searchline driver. The parasitic RC delay of the searchline can be neglected. Therefore τ_{SL} is proportional to the gate delay of an inverter with a fan-out of 4 (which is a technology-dependent parameter), i.e.,

$$\tau_{SL} \propto FO_4 \tag{5.17}$$

5.5.1.2 Bit cell latency

The bit cell latency τ_{bit} is defined as the time delay from the searchline input of the cell to the switching of the pull-down transistor (T_5 in Figure 5.1). To the first order approximation, τ_{bit} is dominated by the charging/discharging time at node G of the bit cell. Assuming G is initially discharged to GND, its transient voltage waveforms under all miss/match scenarios are conceptually depicted in Figure 5.6 and τ_{bit} is proportional to the RC time constant in the case of miss, i.e.,

$$\tau_{bit} \propto (R_{LRS} \parallel R_{HRS}) \cdot C_G \tag{5.18}$$

where C_G represents the total parasitic capacitance appearing at node G. Assuming $R_{LRS} << R_{HRS}$, (5.18) is reduced to,

$$\tau_{bit} \propto R_{LRS} \cdot C_G \tag{5.19}$$

As more advanced CMOS technologies are adopted, it's reasonable to assume R_{LRS} stays relatively constant whereas C_G scales accordingly. C_G is dominated by the gate capacitance of T_5 in Figure 5.1, i.e.,

$$C_G \simeq C_{ox}WL = \frac{\varepsilon}{t_{ox}}WL \tag{5.20}$$

where C_{ox} is the gate capacitance per unit-area, W is the channel width, L is the channel length, ε is the dielectric constant of the gate oxide, and t_{ox} is the thickness of the gate oxide. To the first order approximation, when the CMOS technology scales by F, the same scaling factor applies to t_{ox}, W and L. Therefore C_G and τ_{bit} both scale by F.

5.5.1.3 Matchline latency

The matchline latency (τ_{ML}) is defined as the time period it takes to discharge the matchline in the case of miss. The worst-case τ_{ML} is obtained when there is only one mismatched bit in the stored word, i.e.,

$$\tau_{ML} \propto R_{ON}C_{ML} \tag{5.21}$$

where R_{ON} is the ON resistance of the pull-down transistor and C_{ML} is the total parasitic capacitance on the matchline. R_{ON} and C_{ML} are expressed as follows,

$$R_{ON} = \frac{1}{\mu C_{OX}\frac{W}{L}(V_{GS} - V_{th})} \tag{5.22}$$

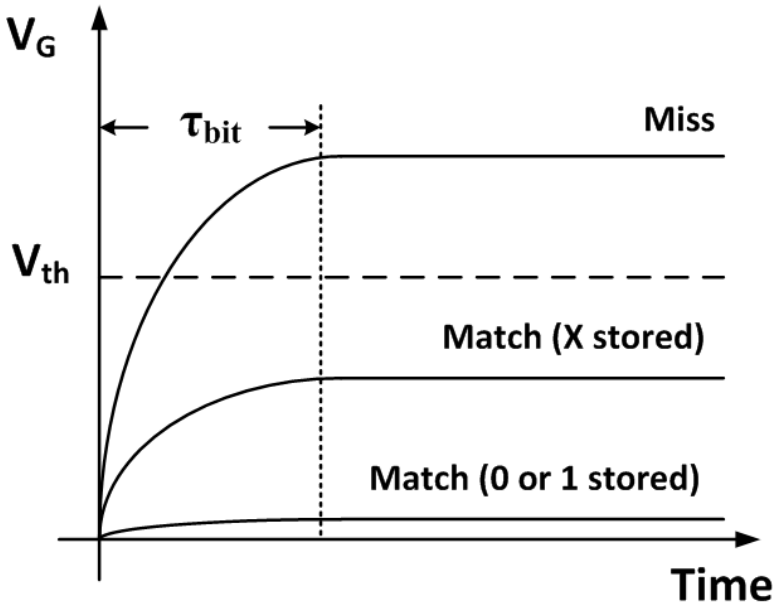


Figure 5.6 Conceptual illustration of the voltage at node G in all miss/match scenarios, where V_{th} is the threshold voltage of T_5 in Figure 5.1.

$$C_{ML} = C_{wire} + N \cdot C_{D,ML} + C_{SA} \tag{5.23}$$

where μ is a technology-related constant, V_{GS} is the gate voltage of the pull-down transistor, C_{wire} is the parasitic capacitance of the matchline wire, N is the word width, $C_{D,ML}$ is the drain capacitance of the pull-down transistor, and C_{SA} is the input capacitance of the sense amplifier.

τ_{ML}'s dependency on technology nodes is discussed below. As previously derived, C_{ox}, W and L all scale by a factor of F. $V_{GS}-V_{th}$ can be assumed as a constant because: 1) as discussed in Chapter 5.2, V_{GS} (or V_{search}) is determined by V_{th}, i.e., $V_{GS} \propto V_{th}$; 2) V_{th} can be assumed to be constant across difference CMOS technologies. Therefore

Table 5.3 Scaling rules for latencies of searchline, bit cell, and matchline.

Latency	Scaling rule
Searchline (τ_{SL})	$\propto FO_4$
Bit cell (τ_{bit})	Factor: F
Matchline (τ_{ML})	Factor: F^2

R_{ON} scales by a factor of F. Assuming C_{wire} and C_{SA} can be neglected comparing with $N \cdot C_{D,ML}$, (5.23) is reduced to,

$$C_{ML} = N \cdot C_{D,ML} \tag{5.24}$$

Similar to C_{ox}, the scaling factors of $C_{D,ML}$ and C_{ML} are both F. As a result, τ_{ML} scales by F^2.

Table 5.3 summarizes the scaling rules for τ_{SL}, τ_{bit} and τ_{ML}.

5.5.2 Energy

The total energy consumed by of the mTCAM per search cycle (E_S) consists of energy consumption from the searchline (E_{SL}), the bit cell (E_{bit}) and the matchline (E_{ML}), i.e.,

$$E_S = E_{SL} + E_{bit} + E_{ML} \tag{5.25}$$

The average energy consumption ($\overline{E_S}$) is defined as the energy consumed per bit per search cycle, i.e.,

$$\overline{E_S} = \frac{E_S}{M \cdot N} \tag{5.26}$$

where M and N are the search depth and the word width of the mTCAM, respectively. $\overline{E_S}$ has a unit of (fJ/bit/search) and is commonly used to evaluate the energy performance of a CAM system.

5.5.2.1 Searchline energy consumption

Although each bit cell is connected with three searchlines (SL, $\overline{\text{SL}}$ and SX), only one search line is asserted in the search mode according to the search protocol described in Chapter 5.2. The searchline energy consumption (E_{SL}) is dominated by charging/discharging the parasitic capacitances, i.e.,

$$E_{SL} = N \cdot (C_{SL} V_{search}^2) \tag{5.27}$$

where C_{SL} is the total parasitic capacitance on the searchline and V_{search} is the voltage applied to the searchline during the search mode. Note that,

$$C_{SL} = C_{wire} + M \cdot C_{D,SL} \tag{5.28}$$

where $C_{D,SL}$ is the drain capacitance of T_1 (or T_2) in Figure 5.1. Assuming C_{wire} can be neglected, (5.28) is reduced to,

$$E_{SL} = N \cdot M \cdot C_{D,SL} V_{search}^2 \tag{5.29}$$

Based on previous discussions, when the technology scales by a factor of F, $C_{D,SL}$ scales by F whereas V_{search} is constant. Thus E_{SL} scales by a factor of F.

5.5.2.2 Bit cell energy consumption

The bit cell energy consumption (E_{bit}) is caused by the static current flow between SL and $\overline{SL}$ when searching for 0 or 1. The worst-case E_{bit} is obtained when the stored bit is 0 or 1, which results a minimum resistance of R_{HRS} between SL and $\overline{SL}$. E_{bit} is derived as follows,

$$E_{bit} = \left(\frac{V_{search}}{R_{HRS}}\right)^2 \cdot (\tau_{bit} + \tau_{ML}) \cdot M \cdot N \tag{5.30}$$

Note that the time period during which SL (or $\overline{SL}$) is asserted is $\tau_{bit} + \tau_{ML}$ to ensure the matchline reflects the search result. Previous discussions reveal that V_{search} and R_{HRS} are constant during technology scaling. Therefore E_{bit} follows the same scaling rule of $\tau_{bit} + \tau_{ML}$ listed in Table 5.3.

5.5.2.3 Matchline energy consumption

Similar to E_{SL}, the energy consumed on the matchline (E_{ML}) is dominated by charging/discharging parasitic capacitances. Before the search begins, all the matchlines are precharged to V_{ML}. As a result, E_{ML} can be expressed as,

$$E_{ML} = M \cdot (C_{ML} V_{ML}^2) \tag{5.31}$$

where C_{ML} is the total parasitic capacitance on the matchline and can be approximated as,

$$C_{ML} = N \cdot C_{D,ML} \tag{5.32}$$

where $C_{D,ML}$ is the drain capacitance of the pull-down transistor T_5. Thus (5.31) is reduced to,

$$E_{ML} = M \cdot N \cdot C_{D,ML} V_{ML}^2 \tag{5.33}$$

The optimal value of V_{ML} is determined by the tradeoff between the power and the speed, and also depends on the sensing scheme of the sense amplifier. To the first order approximation, the following can be assumed,

$$V_{ML} \propto V_{th} \tag{5.34}$$

According to previous discussions, as the CMOS technology scales, V_{ML} is constant and $C_{D,ML}$ scales by a factor of F. Consequently E_{ML} scales by a factor of F as well.

In summary, the average energy consumption $\overline{E_S}$ can be expressed as follows,

Table 5.4 Scaling rules for energy consumptions of searchline, bit cell and matchline.

Energy consumption	Scaling Rule
Search line (E_{SL})	Factor: F
Bit cell (E_{bit})	Same as $\tau_{bit}+\tau_{ML}$
Match line (E_{ML})	Factor: F

$$\overline{E_S} = \frac{1}{M \cdot N}(E_{SL} + E_{bit} + E_{ML}) \\ = C_{D,SL}V_{search}^2 + \left(\frac{V_{search}}{R_{HRS}}\right)^2 \cdot (\tau_{bit} + \tau_{ML}) + C_{D,ML}V_{ML}^2 \quad (5.35)$$

Table 5.4 summarizes the scaling rules for E_{SL}, E_{bit} and E_{ML}.

5.6 Functionality verification

An mTCAM array is designed using 0.18μm CMOS technology. The memristors inside the mTCAM are represented by our SPICE-compatible model [5, 12, 78]. Conservative estimations on wire parasitics are also included in the design. Simulations in both write and search modes are conducted to verify the functionalities of the mTCAM.

5.6.1 Write mode

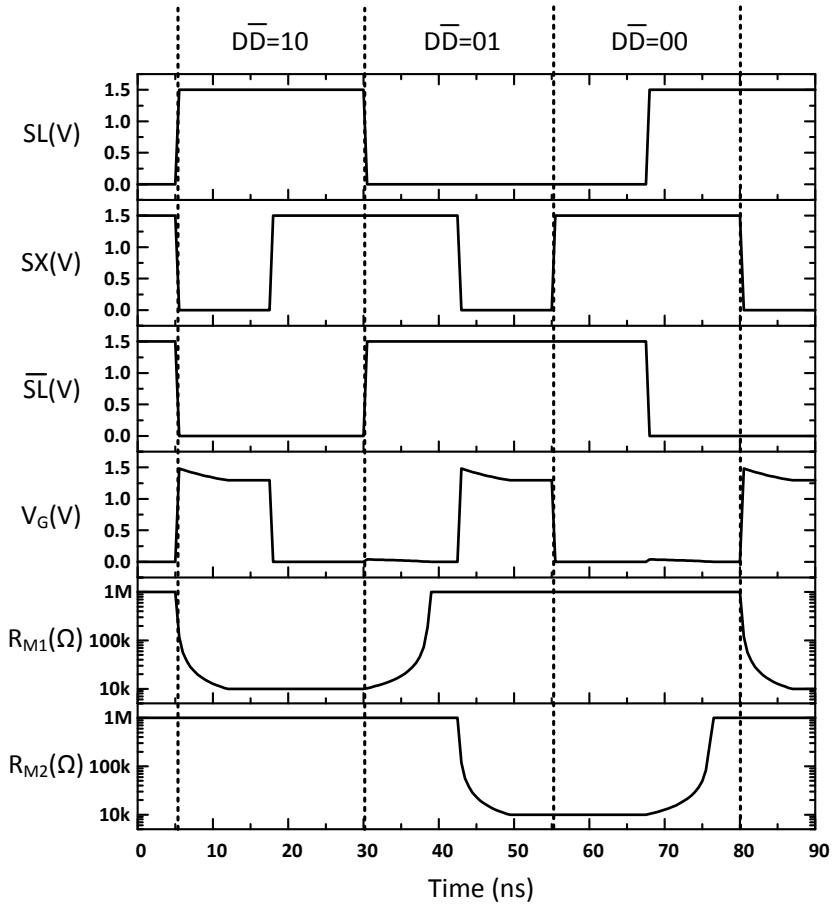


Figure 5.7 Simulation results of the mTCAM in write mode.

In the write mode, data of C = 1, C = 0, and C = X are written to the mTCAM cell sequentially. The V_{write} is set to 1.5V to provide about 0.5V of voltage compliance for the device to switch. The transient voltage waveforms and corresponding memristances of a cell are shown in Figure 5.7. For example, to write C = 1 (D = 1, $\overline{D}$ = 0) to the cell, M_1 is programmed to LRS followed by setting M_2 to HRS. It can be observed that at the end of each write cycle, the data has been correctly written to the cell regardless of its previously stored content. Each write cycle has a duration of ~25ns to accommodate

the finite switching time of the memristors. Note that the deviation of V_G from its initial value is due to the finite ON resistances of access transistors.

5.6.2 Search mode

To verify the functionality of the search operation, an mTCAM of 4-bit word width is simulated. The power supply used to precharge ML and drive SAL is set to 1V. V_{search} is set to 640mV according to (5.12). The content of C = {1, X, 1, 0} is programmed to a random word. The search word is set to S_1 = {1, 0, X, 1} and S_2 = {1,

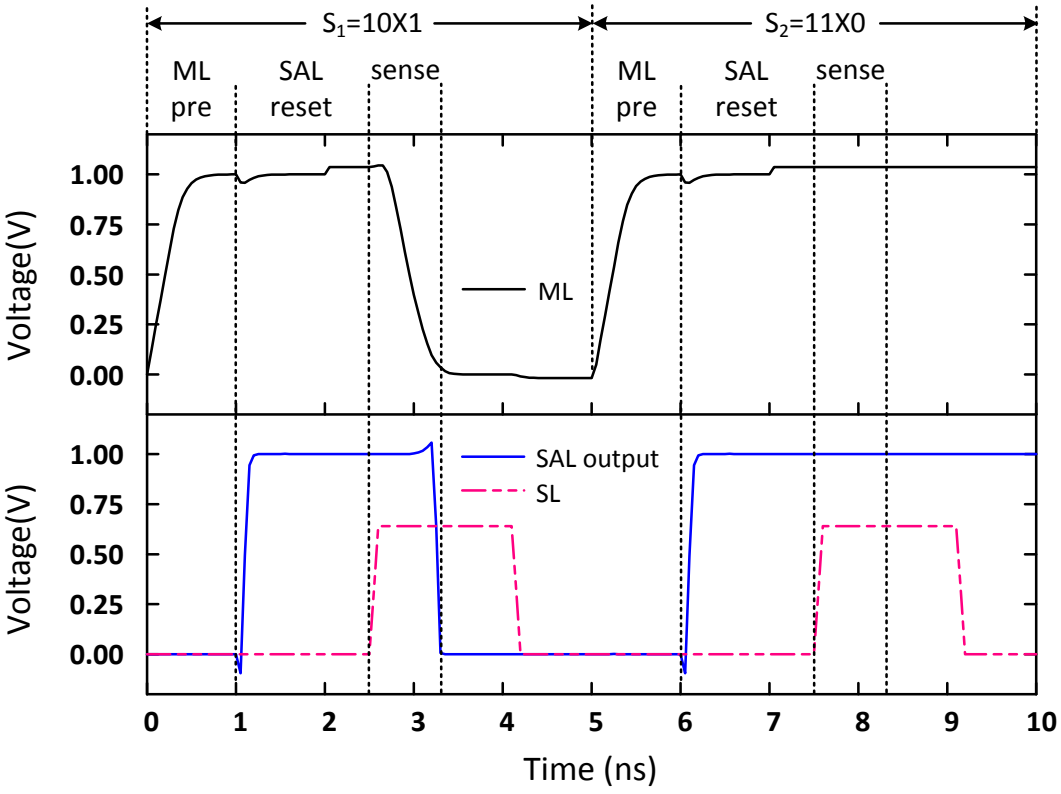


Figure 5.8 Simulation results of the mTCAM in search mode.

1, X, 0} sequentially. The voltage waveforms of ML and the output of SAL are depicted in Figure 5.8. In the precharge phase, both ML and SAL are set to their desired states. After the search word is placed on SLs, the voltage of ML changes according to the search result. In the first search cycle, since there is a one-bit 'miss' between S_1 and C, ML is discharged and SAL's output switches from high to low. In the second search cycle, both ML and SAL stay at high, indicating a 'match' has been found between S_2 and C.

5.7 Performance evaluation and projection

The performance of the proposed mTCAM is evaluated by characterizing search latency and energy consumption of the system. Simulations on mTCAM systems of different search depths and word widths are conducted to analyze search latency and energy consumption's dependencies on the size of the system. Simulation data are extrapolated to estimate the performance of mTCAMs in a wide range of search depth and word width configurations. The search depth (or the number of rows) of the mTCAM is varied from 16 to 1024. The search width (or the number of columns) of the mTCAM is varied from 16 to 128. Simulation data in 0.18μm node are used to project the mTCAM's performance in more advanced technologies (e.g., 90nm and 45nm) based on the formulations in Tables 5.3 and 5.4. The data from ITRS

(International Technology Roadmap for Semiconductors) are also obtained to facilitate the projection.

The search latency's dependencies on search width and search depth are illustrated in Figures 5.9 and 5.10 respectively. As search width and search depth increase, the search latency increases accordingly. The search latency is a stronger function of the word width than the search depth due to the fact that the matchline latency increases linearly with the word width according to (5.21) whereas the searchline latency can always optimized by re-designing the searchline driver. Nonetheless the search latency decreases as more advanced technologies are adopted. For example, an mTCAM array

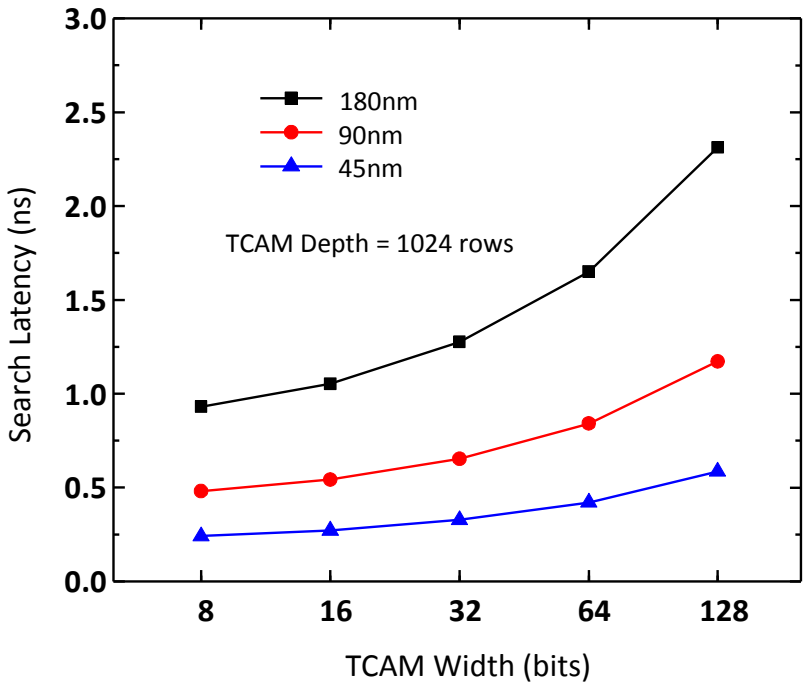


Figure 5.9 Simulation results of the search latency's dependency on word width.

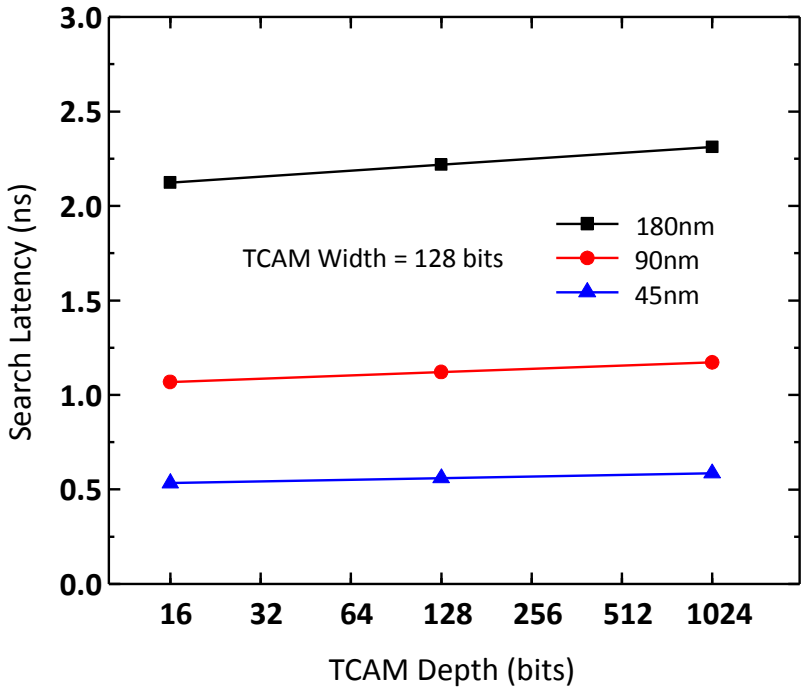


Figure 5.10 Simulation results of the search latency's dependency on search depth.

with a size of 1024 by 128 exhibits search latencies of 2.3ns, 1.2ns, and 0.59ns in 0.18μm, 90nm, and 45nm nodes respectively.

The energy consumption at different word widths and search depths are presented in Figures 5.11 and 5.12 respectively. When the word width increases from 8 bit to 128 bit, a local minimum value for search energy exists due to the following reason. $\overline{E_{bit}}$ increases with bigger word width due to the longer τ_{ML}. $\overline{E_{ML}}$ decreases with bigger word width since the parasitic capacitances of the metal trace and the sense amplifier's input become less significant than the total drain capacitance of all pull-down transistors. The search energy is less dependent on the search depth given the same

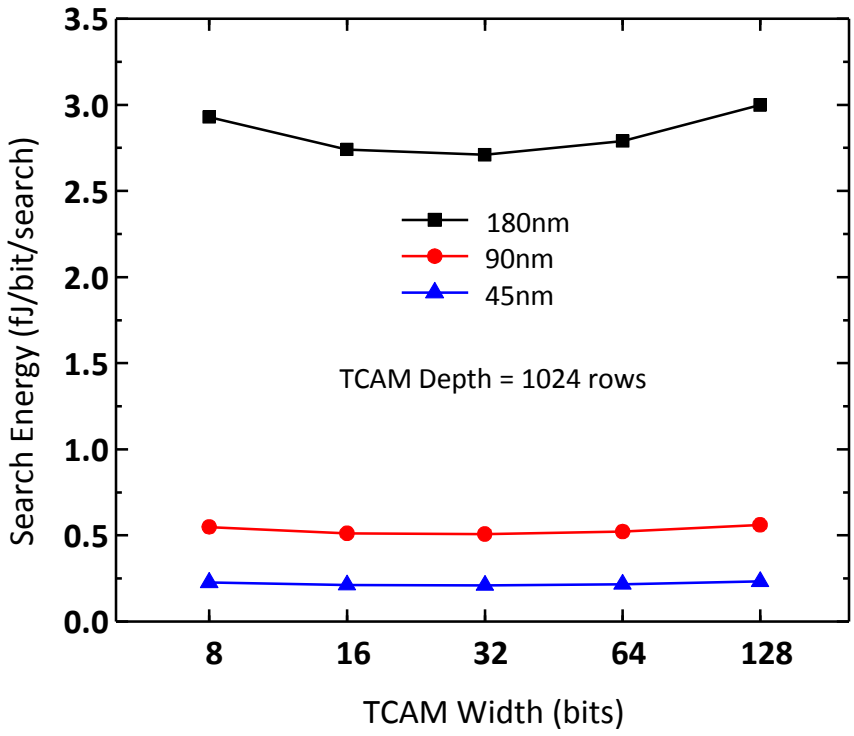


Figure 5.11 Simulation results of the search energy's dependency on word width.

word width, as observed from Figure 5.12. The search energy of an mTCAM with a size of 1024 by 128 is 3fJ/bit/search, 0.56fJ/bit/search, and 0.23fJ/bit/search in 0.18μm, 90nm, and 45nm nodes respectively.

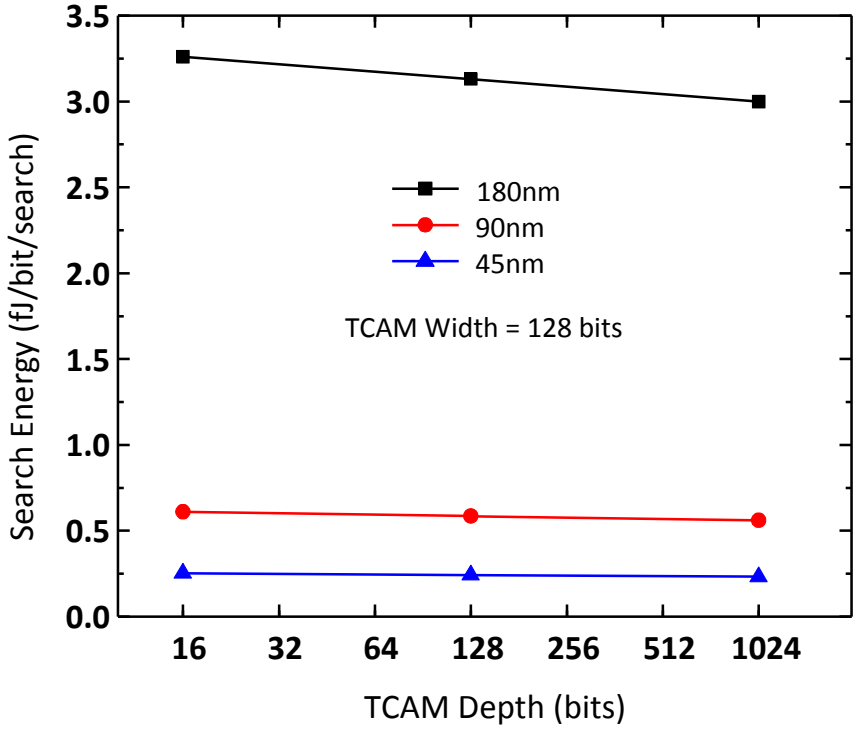


Figure 5.12 Simulation results of the search energy's dependency on search depth.

5.8 Comparisons with existing CAM/TCAM systems

The proposed mTCAM is compared with existing CAM/TCAM systems that use either SRAM or NVM devices as the storage elements. The comparisons focus on storage density, search latency and energy consumption across technology nodes from 0.18μm to 45nm, as shown in Figures 5.13, 5.14, and 5.15 respectively. The compact 5T2M bit cell helps the proposed mTCAM achieve a storage density that is larger than all reported SRAM-based CAM systems and is competitive to other NVM-based alternatives in each technology node. It is estimated that the proposed mTCAM attains a storage density of 0.2Mb/mm^2, 0.8Mb/mm^2 and 3.3Mb/mm^2 at 0.18μm, 90nm, and

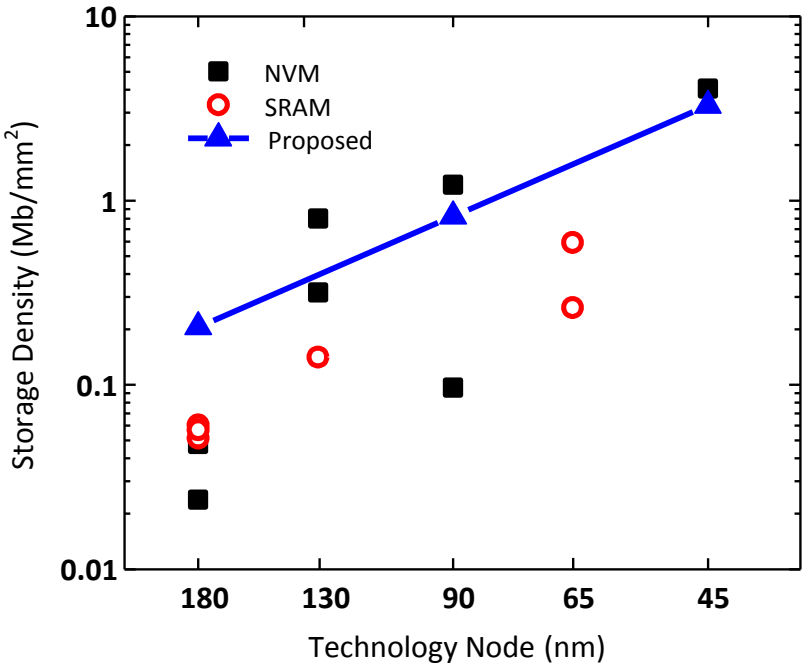


Figure 5.13 Comparison with existing CAM/TCAM systems on storage densities.

45nm technology nodes respectively. The search latency of the proposed mTCAM is superior to other NVM-based alternatives and can directly compete with SRAM-based CAM systems. Latencies of 2.3ns, 1.1ns, and 0.58ns are estimated at 0.18μm, 90nm, and 45nm nodes. Similarly, the proposed mTCAM is more energy-efficient than existing NVM-based alternatives and has similar energy performance as SRAM-based counterparts. Energy consumption of 3.0fJ/bit/search, 0.56fJ/bit/search, 0.23fJ/bit/search are estimated at 0.18μm, 90nm, and 45nm nodes.

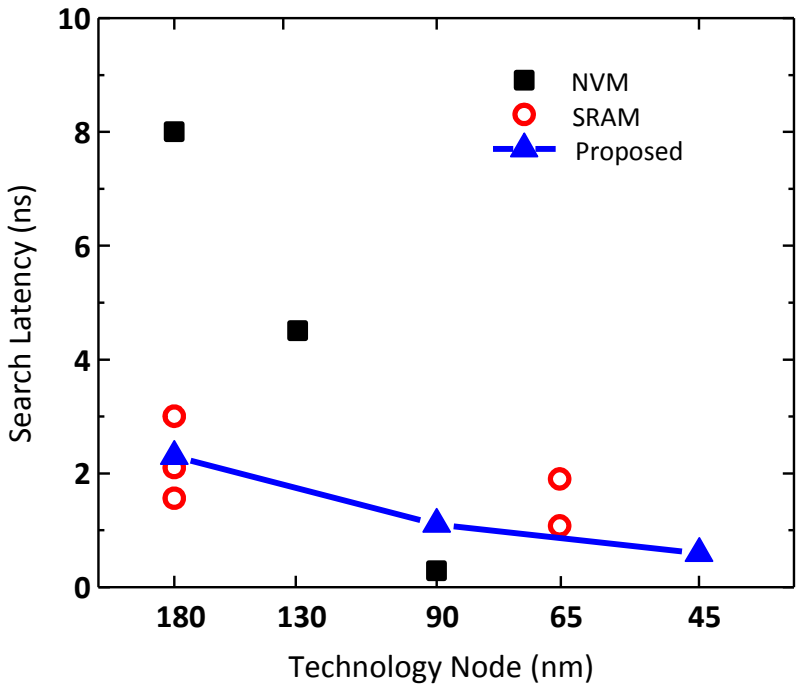


Figure 5.14 Comparison with existing CAM/TCAM systems on search latencies.

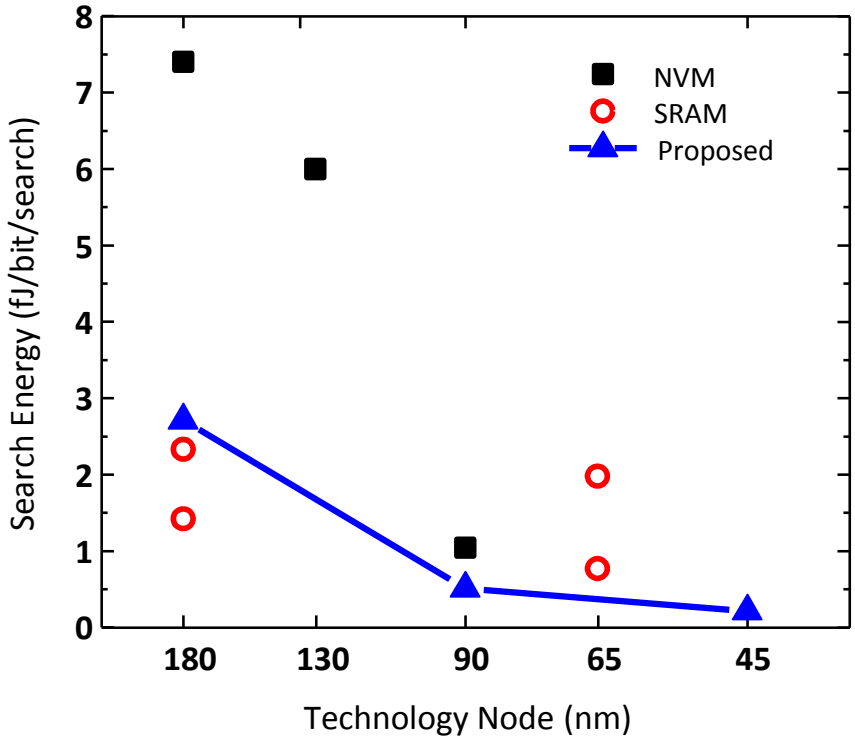


Figure 5.15 Comparison with existing CAM/TCAM systems on energy consumptions.

In Table 5.5, the specifications of the proposed mTCAM are summarized and compared with existing CAM/TCAM systems. The proposed mTCAM demonstrates the feasibility to implement a non-volatile TCAM by offering a compact cell size (5T2M) as well as competitive storage density (0.2Mb/mm^2), search latency (2.3ns) and search energy (3fJ/bit/search). Higher storage density is achievable if a more advanced technology node is chosen. Even more promising latency and power performance can be expected when architectural techniques (e.g., bank selection, pre-computation matchline, etc. [48]) are adopted.

Table 5.5 Comparisons with existing CAM/TCAM systems.

	Arsovski, JSSC'13	**Hayashi, JSSC'13**	**Li, JSSC'14**	**Matsunaga, VLSIC'11**	**Matsunaga, JJAP '11**	**Xu, TVLSI'10**	**This work**
Technology	32nm	65nm	90nm	90nm	0.14μm	0.18μm	**0.18μm**
CAM/TCAM	TCAM	TCAM	TCAM	TCAM	CAM	TCAM	**TCAM**
Word Length (bit)	640	72	64	32	64	144	**128**
Cell Structure	16T	16T	2T2PCM	6T2MTJ	1T1MTJ	11T3MTJ	**5T2M**
Supply Voltage (V)	0.7−1.1	1	1.2	1.2	1.0 or 1.5	1.8	**1.5/1.0/0.63**
Storage Density (Mb/mm^2)	0.84	0.59	1.22	0.10	0.80	0.02	**0.21**
Search Latency	1GHz*	1.9ns	1.9ns	0.29ns	1ns	8ns	**2.3ns**
Search Energy (fJ/bit/search)	0.58	1.98	N/A	1.04	6	7.4	**3**

Chapter 6 Conclusion

As the computing systems have continuously benefited from the scaling of CMOS technology for the past few decades with ever-improving performance and efficiency, the advantage of scaling has started to diminish when the 'power wall' is hit. Hardware and software innovations are called to build next-generation computing systems. Recently emerging NVM technologies have attracted significant attention since they are promising candidates for future memory products that could lead to power-efficient computing systems with better performance.

As a class of NVM devices, memristor was first postulated as the fourth fundamental two-terminal passive circuit element to establish a direct link between flux and charge. Ever since the first nanoscale memristor was identified, extensive research activities have focused on exploiting new possibilities of using 'resistance' instead of 'charge' as the new state variable. Due to the non-volatility, nonlinear dynamics, and compact dimension of the devices, memristors are widely explored in applications such as resistive random access memory, reconfigurable nanoelectronic systems, ultra-high density Boolean logic, non-volatile VLSI computing, neuromorphic computing, etc.

A compact circuit model of memristors is demanded to help analyze and design future electronic systems. A modular compact model for a broad range of memristors

has been proposed in this dissertation. The proposed model is flexible enough to capture a wide variety of device behaviors, including bounded memristance, threshold voltage for SET/RESET, nonlinear switching rate, nonlinear *i*-*v* relationship, statistical distributions of device parameters, etc. The proposed model not only mitigates stability issues from previous works, it also reveals that an equivalent flux-charge constitutive relationship can always be obtained.

Because CAM offers high-throughput associative lookups, it has been investigated to improve the performance of computing systems for data-intensive applications. TCAM is a special class of CAM that allows storing and searching for a wildcard. Such flexibility provides TCAM with greater advantages in applications such as data compression, pattern recognition, genome analysis/diagnosis, search engine, scientific computing, etc. Basic concepts of CAMs are reviewed in this dissertation, including cell topologies, matchline structures, and array architectures. The tradeoffs between speed and power in designing CAMs are also discussed.

The usage of CAMs nowadays has largely been limited to high-performance network routers due to high cost-per-bit and low storage capacity (~tens of Mb). Recently NVM-based CAMs have shown promises to increase the storage capacity/density and reduce the cost of manufacturing. Various cell structures based on different NVM technologies (e.g., PCM, STT-MRAM, memristor) have been proposed

and their advantages and disadvantages are identified. We propose a novel memristor-based TCAM with a 5T2M bit cell structure. Design procedures are explained in details on the levels of bit cell structure, matchline structure, and array architecture. A novel two-step write scheme is introduced to eliminate programming uncertainties. The search scheme is designed to ensure optimal sensing margins given statistical variations on memristor devices. A scalable model is developed to estimate the latency and energy consumption of mTCAMs with various sizes. The model is also technology-dependent so that performance projections into more advanced technologies are supported. The proposed mTCAM is implemented in 0.18μm CMOS technology. Simulations demonstrate functionalities in both write and search modes. Comparing with existing CAM/TCAM systems, the proposed mTCAM achieves superior storage density, and exhibits competitive latency and energy performance across a range of technology nodes from 0.18μm to 45nm.

While this dissertation investigates the feasibility of incorporating memristors into CAM systems, the future work is substantially more challenging. On the computer architecture level, a high-level simulator is required to effectively emulate the performance of the mTCAM via the developed latency/energy model, and evaluate the system against performance benchmarks. A compatible command interface is needed to allow the correct representation of mTCAM circuits in the simulator. Emulation results from the simulator are analyzed to explore new computing architectures.

Emulation results are also continuously fed back to circuit designs for improvements in the areas of storage density, latency, energy consumption, etc. On the circuit level, it is crucial to develop novel circuits and array architectures to leverage unique properties of memristors and demonstrate greater promises of using memristor-based TCAMs in high-performance computing systems for data-intensive applications.

Reference

[1] D. B. Strukov*, et al.*, "The missing memristor found," *Nature,* vol. 453, pp. 80-83, 2008.

[2] Y. V. Pershin, S. La Fontaine, and M. Di Ventra, "Memristive model of amoeba learning," *Phys. Rev. E,* vol. 82, 2010.

[3] Y. N. Joglekar and S. J. Wolf, "The elusive memristor: properties of basic electrical circuits," *Eur. J. Phys.,* vol. 30, pp. 661-675, Jul 2009.

[4] Z. Biolek, D. Biolek, and V. Biolkova, "SPICE model of memristor with nonlinear dopant drift," *Radioengineering,* vol. 18, pp. 210-214, Jun 2009.

[5] S. Shin, K. Kim, and S. M. S. Kang, "Compact models for memristors based on charge-flux constitutive relationships," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.,* vol. 29, pp. 590-598, Apr 2010.

[6] C. Yakopcic*, et al.*, "Generalized memristive device SPICE model and its application in circuit design," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.,* vol. 32, pp. 1201-1214, Aug 2013.

[7] Y. V. Pershin and M. Di Ventra, "SPICE model of memristive devices with threshold," *arXiv:1204.2600,* 2012.

[8] F. Corinto and A. Ascoli, "A boundary condition-based approach to the modeling of memristor nanostructures," *IEEE Trans. Circuits and Syst. I, Reg. Papers,* vol. 59, pp. 2713-2726, Nov 2012.

[9] L. Zheng, S. Shin, and S. M. S. Kang, "Unified modeling for memristive devices based on charge-flux constitutive relationships," in *IEEE ISCAS*, 2013, pp. 213-216.

[10] M. Horowitz, "Computing's energy problem (and what we can do about it)," in *IEEE ISSCC*, 2014, pp. 10-14.

[11] D. Perlmutter, "Sustainability in silicon and systems development," in *IEEE ISSCC*, 2012, pp. 31-35.

[12] L. Zheng, S. Shin, and S. M. S. Kang, "Modular structure of compact models for memristive devices," *IEEE Trans. Circuits Syst. I, Reg. Papers,* vol. 61, pp. 1390-1399, 2014.

[13] M.-K. Tsai, "Cloud 2.0 clients and connectivity—technology and challenges," in *IEEE ISSCC*, 2014, pp. 15-19.

[14] L. O. Chua, "Memristor-the missing circuit element," *IEEE Trans. Circuit Theory,* vol. 18, pp. 507-519, 1971.

[15] L. O. Chua and S. M. S. Kang, "Memristive devices and systems," *Proc. IEEE,* vol. 64, pp. 209-223, 1976.

[16] A. L. Hodgkin and A. F. Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve," *J. Physiol.,* vol. 117, p. 500, 1952.

[17] J. J. Yang, *et al.*, "Memristive switching mechanism for metal/oxide/metal nanodevices," *Nat. Nanotechnol.,* vol. 3, pp. 429-433, 2008.

[18] M. D. Pickett, *et al.*, "Switching dynamics in titanium dioxide memristive devices," *J. Appl. Phys.,* vol. 106, 2009.

[19] S. H. Jo, K.-H. Kim, and W. Lu, "High-density crossbar arrays based on a Si memristive system," *Nano Lett.,* vol. 9, pp. 870-874, 2009.

[20] S. H. Jo, *et al.*, "Nanoscale memristor device as synapse in neuromorphic systems," *Nano Lett.,* vol. 10, pp. 1297-1301, 2010.

[21] H. Y. Lee, *et al.*, "Evidence and solution of over-RESET problem for HfOx based resistive memory with sub-ns switching speed and high endurance," in *IEEE IEDM*, 2010, pp. 19.7. 1-19.7. 4.

[22] S. Yu, X. Guan, and H. S. P. Wong, "On the switching parameter variation of metal oxide RRAM part II: model corroboration and device design strategy," *IEEE Trans. Electron Devices,* vol. 59, pp. 1183-1188, 2012.

[23] S.-S. Sheu, *et al.*, "A 4Mb embedded SLC resistive-RAM macro with 7.2ns read-write random-access time and 160ns MLC-access capability," in *IEEE ISSCC*, 2011, pp. 200-202.

[24] Y. S. Chen, *et al.*, "Challenges and opportunities for HfOx based resistive random access memory," in *IEEE IEDM*, 2011, pp. 31.3.1-31.3.4.

[25] J. J. Yang, *et al.*, "Engineering nonlinearity into memristors for passive crossbar applications," *Appl. Phys. Lett.,* vol. 100, p. 113501, Mar 2012.

[26] M.-J. Lee*, et al.*, "A fast, high-endurance and scalable non-volatile memory device made from asymmetric Ta2O5−x/TaO2−x bilayer structures," *Nat. Mater.,* vol. 10, pp. 625-630, 2011.

[27] S. M. S. Kang and S. Shin, "Energy-efficient memristive analog and digital electronics," in *Advances in Neuromorphic Memristor Science and Applications*. vol. 4, R. Kozma, R. E. Pino, and G. E. Pazienza, Eds., ed: Springer Netherlands, 2012, pp. 181-209.

[28] R. Waser and M. Aono, "Nanoionics-based resistive switching memories," *Nat. Mater.,* vol. 6, pp. 833-840, 2007.

[29] H. S. P. Wong*, et al.*, "Metal-oxide RRAM," *Proc. IEEE,* vol. 100, pp. 1951-1970, 2012.

[30] S. Shin, K. Kim, and S. M. S. Kang, "Analysis of passive memristive devices array: data-dependent statistical model and self-adaptable sense resistance for RRAMs," *Proc. IEEE,* vol. 100, pp. 2021-2032, 2012.

[31] A. Kawahara*, et al.*, "An 8Mb multi-layered cross-point ReRAM macro with 443MB/s write throughput," in *IEEE ISSCC*, 2012, pp. 432-434.

[32] M.-F. Chang*, et al.*, "Embedded 1Mb ReRAM in 28nm CMOS with 0.27-to-1V read using swing-sample-and-couple sense amplifier and self-boost-write-termination scheme," in *IEEE ISSCC*, 2014, pp. 332-333.

[33] R. Fackenthal, *et al.*, "A 16Gb ReRAM with 200MB/s write and 1GB/s read in 27nm technology," in *IEEE ISSCC*, 2014, pp. 338-339.

[34] G. S. Rose, *et al.*, "Designing CMOS/molecular memories while considering device parameter variations," *ACM JETC,* vol. 3, p. 3, 2007.

[35] J. Rajendran, *et al.*, "An energy-efficient memristive threshold logic circuit," *IEEE Trans. Comput.,* vol. 61, pp. 474-487, 2012.

[36] P. E. Gaillardon, *et al.*, "GMS: Generic memristive structure for non-volatile FPGAs," in *IEEE VLSI-SoC*, 2012, pp. 94-98.

[37] J. Cong and B. Xiao, "mrFPGA: A novel FPGA architecture with memristor-based reconfiguration," in *IEEE NANOARCH*, 2011, pp. 1-8.

[38] S. Shin, K. Kim, and S. M. S. Kang, "Resistive computing: memristors-enabled signal multiplication," *IEEE Trans. Circuits Syst. I, Reg. Papers,* vol. 60, pp. 1241-1249, 2013.

[39] S. Shin, K. Kim, and S. M. S. Kang, "Reconfigurable stateful nor gate for large-scale logic-array integrations," *IEEE Trans. Circuits Syst. II, Exp. Briefs,* vol. 58, pp. 442-446, 2011.

[40] K. Kim, S. Shin, and S. M. S. Kang, "Field programmable stateful logic array," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.,* vol. 30, pp. 1800-1813, 2011.

[41] J. Borghetti, *et al.*, "'Memristive' switches enable 'stateful' logic operations via material implication," *Nature,* vol. 464, pp. 873-876, 2010.

[42] P. J. Kuekes, D. R. Stewart, and R. S. Williams, "The crossbar latch: Logic value storage, restoration, and inversion in crossbar circuits," *J. Appl. Phys.,* vol. 97, p. 034301, 2005.

[43] S. Shin, K. Kim, and S. M. S. Kang, "Memristive computing - multiplication and correlation," in *IEEE ISCAS*, 2012, pp. 1608-1611.

[44] J. J. Yang, D. B. Strukov, and D. R. Stewart, "Memristive devices for computing," *Nat. Nanotechnol.,* vol. 8, pp. 13-24, 2013.

[45] S. Shin*, et al.*, "Neuronal spike event generation by memristors," in *Cellular Nanoscale Networks and Their Applications (CNNA), 2012 13th International Workshop on*, 2012, pp. 1-4.

[46] H. Kim*, et al.*, "Neural synaptic weighting with a pulse-based memristor circuit," *IEEE Trans. Circuits and Syst. I, Reg. Papers,* vol. 59, pp. 148-158, 2012.

[47] Y. V. Pershin and M. Di Ventra, "Experimental demonstration of associative memory with memristive neural networks," *Neural Netw.,* vol. 23, pp. 881-886, 2010.

[48] K. Pagiamtzis and A. Sheikholeslami, "Content-addressable memory (CAM) circuits and architectures: A tutorial and survey," *IEEE J. Solid-State Circuits,* vol. 41, pp. 712-727, 2006.

[49] Q. Guo*, et al.*, "A resistive TCAM accelerator for data-intensive computing," in *IEEE/ACM MICRO*, 2011, pp. 339-350.

[50] A. Goel and P. Gupta, "Small subset queries and bloom filters using ternary associative memories, with applications," in *ACM SIGMETRICS*, 2010, pp. 143-154.

[51] T. Prodromakis, C. Toumazou, and L. O. Chua, "Two centuries of memristors," *Nat. Mater.,* vol. 11, pp. 478-481, 2012.

[52] Z. Wei*, et al.*, "Highly reliable TaOx ReRAM and direct evidence of redox reaction mechanism," in *IEEE IEDM*, 2008, pp. 1-4.

[53] T. Prodromakis*, et al.*, "A versatile memristor model with nonlinear dopant kinetics," *IEEE Trans. Electron Devices,* vol. 58, pp. 3099-3105, Sep 2011.

[54] Á. Rák and G. Cserey, "Macromodeling of the memristor in SPICE," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.,* vol. 29, pp. 632-636, 2010.

[55] H. Abdalla and M. D. Pickett, "SPICE modeling of memristors," in *IEEE ISCAS*, 2011, pp. 1832-1835.

[56] T. Chang*, et al.*, "Synaptic behaviors and modeling of a metal oxide memristive device," *Appl. Phys. A,* vol. 102, pp. 857-863, 2011.

[57] P. Sheridan*, et al.*, "Device and SPICE modeling of RRAM devices," *Nanoscale,* vol. 3, pp. 3833-3840, 2011.

[58] E. Lehtonen*, et al.*, "Time-dependency of the threshold voltage in memristive devices," in *IEEE ISCAS*, 2011, pp. 2245-2248.

[59] C. Yakopcic*, et al.*, "A memristor device model," *IEEE Electron Device Lett.,* vol. 32, pp. 1436-1438, 2011.

[60] V. Biolkova, *et al.*, "Memristor modeling based on its constitutive relation," in *Proceedings of the European conference of systems, and European conference of circuits technology and devices, and European conference of communications, and European conference on Computer science*, 2010, pp. 261-264.

[61] M.-F. Chang, *et al.*, "A 0.5V 4Mb logic-process compatible embedded resistive RAM (ReRAM) in 65nm CMOS using low-voltage current-mode sensing scheme with 45ns random read time," in *IEEE ISSCC*, 2012, pp. 434-436.

[62] M.-F. Chang, *et al.*, "A high-speed 7.2-ns read-write random access 4-Mb embedded Resistive RAM (ReRAM) macro using process-variation-tolerant current-mode read schemes," *IEEE J. Solid-State Circuits,* vol. 48, pp. 878-891, 2013.

[63] J.-J. Huang, *et al.*, "One selector-one resistor (1S1R) crossbar array for high-density flexible memory applications," in *IEEE IEDM*, 2011, pp. 31.7.1-31.7.4.

[64] M. Liu and W. Wang, "Application of nanojunction-based RRAM to reconfigurable IC," *IET Micro & Nano Lett.,* vol. 3, pp. 101-105, 2008.

[65] J. Borghetti, *et al.*, "A hybrid nanomemristor/transistor logic circuit capable of self-programming," *PNAS,* vol. 106, pp. 1699-1703, 2009.

[66] K. Kim, S. Shin, and S. M. S. Kang, "Stateful logic pipeline architecture," in *IEEE ISCAS*, 2011, pp. 2497-2500.

[67] S. Shin, K. Kim, and S. M. S. Kang, "Memristive XOR for resistive multiplier," *Electronics Letters,* vol. 48, pp. 78-80, 2012.

[68] M. Laiho, *et al.*, "Memristive synapses are becoming reality," *The Neuromorphic Engineer,* 2010.

[69] R. Pino, "Computational intelligence and neuromorphic computing architectures," in *Advances in Neuromorphic Memristor Science and Applications*. vol. 4, R. Kozma, R. E. Pino, and G. E. Pazienza, Eds., ed: Springer Netherlands, 2012, pp. 77-88.

[70] S. Benderli and T. Wey, "On SPICE macromodelling of TiO2 memristors," *Electronics letters,* vol. 45, pp. 377-379, 2009.

[71] D. Batas and H. Fiedler, "A memristor SPICE implementation and a new approach for magnetic flux-controlled memristor modeling," *IEEE Trans. Nanotechnol.,* vol. 10, pp. 250-255, 2011.

[72] A. Ascoli, *et al.*, "PSpice switch-based versatile memristor model," in *IEEE ISCAS*, 2013, pp. 205-208.

[73] S. Kvatinsky, *et al.*, "TEAM: threshold adaptive memristor model," *IEEE Trans. Circuits and Syst. I, Reg. Papers,* vol. 60, pp. 211-221, 2013.

[74] L. Zhang, *et al.*, "A compact modeling of TiO2-TiO2–x memristor," *Appl. Phys. Lett.,* vol. 102, p. 153503, 2013.

[75] P. R. Mickel, *et al.*, "A physical model of switching dynamics in tantalum oxide memristive devices," *Appl. Phys. Lett.,* vol. 102, 2013.

[76] X. Guan, S. Yu, and H. S. P. Wong, "On the switching parameter variation of metal-oxide RRAM part I: physical modeling and simulation methodology," *IEEE Trans. Electron Devices,* vol. 59, pp. 1172-1182, 2012.

[77] X. Guan, S. Yu, and H. S. P. Wong, "A SPICE compact model of metal oxide resistive switching memory with variations," *IEEE Electron Device Lett.,* vol. 33, pp. 1405-1407, 2012.

[78] S. Shin*, et al.*, "Compact circuit model and hardware emulation for floating memristor devices," *IEEE Circuits Syst. Mag.,* vol. 13, pp. 42-55, 2013.

[79] M. Armbrust*, et al.*, "A view of cloud computing," *Commun. ACM,* vol. 53, pp. 50-58, 2010.

[80] G. Bell, T. Hey, and A. Szalay, "Beyond the data deluge," *Science,* vol. 323, pp. 1297-1298, 2009.

[81] I. Gorton*, et al.*, "Data-intensive computing in the 21st century," *Computer,* vol. 41, pp. 30-32, 2008.

[82] K. J. Kuhn, "Considerations for ultimate CMOS scaling," *IEEE Trans. Electron Devices,* vol. 59, pp. 1813-1828, Jul 2012.

[83] K. J. Kuhn, "CMOS scaling beyond 32nm: challenges and opportunities," in *IEEE/ACM DAC*, 2009, pp. 310-313.

[84] R. Chau*, et al.*, "Integrated nanoelectronics for the future," *Nat. Mater.,* vol. 6, pp. 810-812, Nov 2007.

[85] T.-C. Chen, "Where CMOS is going: trendy hype vs. real technology," in *IEEE ISSCC*, 2006, pp. 1-18.

[86] A. Roth*, et al.*, "Advanced ternary CAM circuits on 0.13 μm logic process technology," in *IEEE CICC*, 2004, pp. 465-468.

[87] C.-C. Wang, J.-S. Wang, and C. Yeh, "High-speed and low-power design techniques for TCAM macros," *IEEE J. Solid-State Circuits,* vol. 43, pp. 530-540, 2008.

[88] H. Miyatake, M. Tanaka, and Y. Mori, "A design for high-speed low-power CMOS fully parallel content-addressable memory macros," *IEEE J. Solid-State Circuits,* vol. 36, pp. 956-968, 2001.

[89] S. Liu, F. Wu, and J. B. Kuo, "A novel low-voltage content-addressable-memory (cam) cell with a fast tag-compare capability using partially depleted (pd) soi cmos dynamic-threshold (dtmos) techniques," *IEEE J. Solid-State Circuits,* vol. 36, pp. 712-716, 2001.

[90] G. Thirugnanam, N. Vijaykrishnan, and M. J. Irwin, "A novel low power CAM design," in *IEEE ASIC/SOC*, 2001, pp. 198-202.

[91] G. Kasai*, et al.*, "200MHz/200MSPS 3.2 W at 1.5 V Vdd, 9.4 Mbits ternary CAM with new charge injection match detect circuits and bank selection scheme," in *IEEE CICC*, 2003, pp. 387-390.

[92] I. Arsovski, T. Chandler, and A. Sheikholeslami, "A ternary content-addressable memory (TCAM) based on 4T static storage and including a current-race sensing scheme," *IEEE J. Solid-State Circuits,* vol. 38, pp. 155-158, 2003.

[93] C. A. Zukowski and S.-Y. Wang, "Use of selective precharge for low-power content-addressable memories," in *IEEE ISCAS*, 1997, pp. 1788-1791.

[94] I. Y.-L. Hsiao, D.-H. Wang, and C.-W. Jen, "Power modeling and low-power design of content addressable memories," in *IEEE ISCAS*, 2001, pp. 926-929.

[95] A. Efthymiou and J. D. Garside, "An adaptive serial-parallel CAM architecture for low-power cache blocks," in *ACM ISLPED*, 2002, pp. 136-141.

[96] A. Efthymiou and J. D. Garside, "A CAM with mixed serial-parallel comparison for use in low energy caches," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.,* vol. 12, pp. 325-329, 2004.

[97] N. Mohan and M. Sachdev, "Low power dual matchline ternary content addressable memory," in *IEEE ISCAS*, 2004, pp. II-633-6 Vol. 2.

[98] K.-H. Cheng, C.-H. Wei, and S.-Y. Jiang, "Static divided word matching line for low-power content addressable memory design," in *IEEE ISCAS*, 2004, pp. II-629-32 Vol. 2.

[99] K. Pagiamtzis and A. Sheikholeslami, "A low-power content-addressable memory (CAM) using pipelined hierarchical search scheme," *IEEE J. Solid-State Circuits,* vol. 39, pp. 1512-1519, 2004.

[100] M. Motomura*, et al.*, "A 1.2-million transistor, 33-MHz, 20-b dictionary search processor (DISP) ULSI with a 160-kb CAM," *IEEE J. Solid-State Circuits,* vol. 25, pp. 1158-1165, 1990.

[101] K. J. Schultz and P. G. Gulak, "Fully parallel integrated CAM/RAM using preclassification to enable large capacities," *IEEE J. Solid-State Circuits,* vol. 31, pp. 689-699, 1996.

[102] C.-S. Lin, J.-C. Chang, and B.-D. Liu, "A low-power precomputation-based fully parallel content-addressable memory," *IEEE J. Solid-State Circuits,* vol. 38, pp. 654-662, 2003.

[103] J. Li*, et al.*, "1 Mb 0.41um2 2T-2R cell nonvolatile TCAM with two-bit encoding and clocked self-referenced sensing," *IEEE J. Solid-State Circuits,* vol. 49, pp. 896-907, Apr 2014.

[104] I. Hayashi*, et al.*, "A 250-MHz 18-Mb full ternary CAM with low-voltage matchline sensing scheme in 65-nm CMOS," *IEEE J. Solid-State Circuits,* vol. 48, pp. 2671-2680, Nov 2013.

[105] I. Arsovski*, et al.*, "A 32 nm 0.58-fJ/Bit/Search 1-GHz ternary content addressable memory compiler using silicon-aware early-predict late-correct sensing with embedded deep-trench capacitor noise mitigation," *IEEE J. Solid-State Circuits,* vol. 48, pp. 932-939, Apr 2013.

[106] H. Noda*, et al.*, "A cost-efficient high-performance dynamic TCAM with pipelined hierarchical searching and shift redundancy architecture," *IEEE J. Solid-State Circuits,* vol. 40, pp. 245-253, 2005.

[107] S. Matsunaga, *et al.*, "Fully parallel 6T-2MTJ nonvolatile TCAM with single-transistor-based self match-line discharge control," in *IEEE VLSIC*, 2011, pp. 298-299.

[108] S. Matsunaga, *et al.*, "Standby-power-free compact ternary content-addressable memory cell chip using magnetic tunnel junction devices," *Appl. Phys. Express,* vol. 2, p. 023004, 2009.

[109] B. Rajendran, *et al.*, "Demonstration of CAM and TCAM using phase change devices," in *IEEE IMW*, 2011, pp. 1-4.

[110] O. Kavehei, *et al.*, "Non-volatile complementary resistive switch-based content addressable memory," *arXiv:1108.3716,* 2011.

[111] P. Junsangsri and F. Lombardi, "A memristor-based TCAM (ternary content addressable memory) cell: design and evaluation," in *IEEE GLSVLSI*, 2012, pp. 311-314.

[112] K. Eshraghian, *et al.*, "Memristor MOS content addressable memory (MCAM): Hybrid architecture for future high performance search engines," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.,* vol. 19, pp. 1407-1417, 2011.

[113] W. Xu, T. Zhang, and Y. Chen, "Design of spin-torque transfer magnetoresistive RAM and CAM/TCAM with high sensing and search speed," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.,* vol. 18, pp. 66-74, 2010.

[114] E. Linn, *et al.*, "Complementary resistive switches for passive nanocrossbar memories," *Nat. Mater.,* vol. 9, pp. 403-406, 2010.